



**UNIFACS**  
**UNIVERSIDADE SALVADOR**  
LAUREATE INTERNATIONAL UNIVERSITIES\*

**UNIFACS UNIVERSIDADE SALVADOR**  
**MESTRADO EM SISTEMAS E COMPUTAÇÃO**

**MARCOS ANTÔNIO PACHECO DOS SANTOS**

**UNLOCK-C: UM JOGO SÉRIO 3D PARA APOIO AO ENSINO E APRENDIZADO  
DA LINGUAGEM DE PROGRAMAÇÃO C**

Salvador  
2016

**MARCOS ANTÔNIO PACHECO DOS SANTOS**

**UNLOCK-C: UM JOGO SÉRIO 3D PARA APOIO AO ENSINO E APRENDIZADO  
DA LINGUAGEM DE PROGRAMAÇÃO C**

Dissertação apresentada ao Mestrado em Sistemas e Computação da UNIFACS Universidade Salvador, Laureate International Universities como requisito parcial para a obtenção do título de Mestre

Orientador: Prof. Dr. Sérgio Martins Fernandes.

Salvador  
2016

FICHA CATALOGRÁFICA

Elaborada pelo Sistema de Bibliotecas da UNIFACS Universidade Salvador, Laureate International Universities)

Santos, Marcos Antônio Pacheco dos

UNLOCK-C: um jogo sério 3D para apoio ao ensino e aprendizado da linguagem de programação C./ Marcos Antônio Pacheco dos Santos.- Salvador: UNIFACS, 2016.

199 f. : il.

Dissertação Programa de Pós-Graduação em Sistemas e Computação de UNIFACS Universidade Salvador, Laureate International Universities como requisito parcial à obtenção do título de Mestre.

Orientador: Prof. Dr. Sérgio Martins Fernandes.

1. Jogos educativos. 2. Jogos eletrônico. I. Fernandes, Sérgio Martins, orient. II. Título.

CDD: 004



**UNIFACS**  
**UNIVERSIDADE SALVADOR**  
LAUREATE INTERNATIONAL UNIVERSITIES

**MARCOS ANTONIO PACHECO DOS SANTOS**

**UNLOCK-C: UM JOGO SÉRIO 3D PARA APOIO AO ENSINO E APRENDIZADO DA  
LINGUAGEM DE PROGRAMAÇÃO C.**

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Sistemas e Computação, Universidade Salvador - UNIFACS, do Curso de Mestrado Acadêmico em Sistemas e Computação, pela seguinte banca examinadora:

Prof. Dr. Sérgio Martins Fernandes - Orientador  
Universidade Salvador – UNIFACS

Prof. Dr. Artur Henrique Kronbauer  
Universidade Salvador – UNIFACS

Profa. Dra. Mônica de Souza Massa  
Universidade do Estado da Bahia

Salvador, 21 de outubro de 2016

Dedico este trabalho à minha esposa Tatiana Elvas por ser uma crucial apoiadora e incentivadora para o desenvolvimento deste trabalho e também a meu filho Thales por me prover inspiração, paz e equilíbrio.

## **AGRADECIMENTOS**

Registro aqui meus sinceros agradecimentos a Unifacs e a Rede Laureate pelo incentivo a minha formação docente. Também agradeço aos discentes e docentes da Unifacs que de alguma forma ajudaram no processo de desenvolvimento deste trabalho.

Agradeço enormemente ao Prof. Dr. Sérgio Martins Fernandes, pela sua contribuição precisa e efetiva na definição de todos os aspectos pertencentes a esta dissertação, pela sua compreensão e paciência e, principalmente, pela sua consultoria no que diz respeito a pesquisa científica. Sem isso este trabalho não teria sido desenvolvido.

## RESUMO

A expansão tecnológica e o aumento da influência da computação na sociedade humana faz com que o ensino de programação de computadores em cursos de tecnologia ou graduação da área de computação torne-se cada vez mais importante. Apesar disso, a baixa motivação do aluno, a complexidade do assunto e a forma tradicional de ensino, se configuram como barreiras ao aprendizado nestas disciplinas. Com intuito de mitigar essas barreiras, aparecem nesse contexto um tipo de software denominado jogo sério. Jogos sérios são jogos digitais que possuem um propósito específico que não apenas o entretenimento. Jogos sérios têm recebido uma atenção significativa entre os pesquisadores. Em especial têm ocorrido pesquisas na direção de desenvolver jogos sérios que enriqueçam o processo de ensino e aprendizado das disciplinas de programação de computadores e linguagens de programação. No entanto, muitos jogos sérios educacionais desenvolvidos possuem narrativas maçantes e jogabilidade que deixam a desejar se comparados com os jogos 3D sofisticados usualmente jogados pelos alunos. Com relação a construção desses jogos sérios educacionais, apesar de existirem metodologias de desenvolvimento de software aplicáveis a este tipo de jogo, não existe consenso sobre qual metodologia utilizar. Este trabalho de dissertação propôs a utilização de um jogo sério educacional com representação tridimensional (3D) sofisticado, de gênero mais alinhado com os jogos digitais 3D, jogados pelos alunos nativos digitais. O jogo foi desenvolvido através de uma metodologia de desenvolvimento de software customizada e otimizada a partir de outras metodologias de desenvolvimento de jogos sérios educacionais. O jogo foi utilizado em salas de aulas de cursos de graduação em computação, como ferramenta de ensino e aprendizado de conceitos relacionados a disciplinas de programação de computadores, com foco na linguagem C. Os resultados preliminares mostraram um aumento da motivação, e interesse dos alunos em relação ao aprendizado dos conteúdos das disciplinas. Desta forma, este jogo sério poderá ser utilizado como ferramenta de ensino e aprendizado como forma de obter mais dados sobre o problema e concomitantemente tentar melhorar o nível de motivação e interesse dos alunos acerca da linguagem C e conceitos de programação de computadores.

**Palavras-chave:** Jogos sérios. Jogos Sérios de Ensino a Programação. Jogos Digitais Educacionais.

## ABSTRACT

The growing influence of computing on human society makes it increasingly important the teaching of computer programming in computer sciences undergraduate courses or technological courses. Nevertheless, the low level of motivation of many students, the complexity of the subject and the traditional way of teaching are barriers for learning these disciplines. In order to mitigate these barriers ,in this context, a type of software called serious game was developed, as a tool to help teach programming languages. Serious games are digital games that have a specific purpose other than just entertainment. Serious games have received significant attention among researchers. In particular, there have been research toward developing serious games that enrich the teaching and learning process of programming disciplines of computers and programming languages. However, many developed educational games have serious dull narrative and gameplay that are lacking compared to the sophisticated 3D games usually played by students. Regarding the construction of these educational serious games, although there are software development methodologies applicable to this type of game, there is no consensus on which methodology to use. This dissertation proposed the use of an educational serious game with three-dimensional (3D) sophisticated, more in line with gender digital 3D games, played by digital native students. The game was developed by a custom software development methodology optimized from other educational serious games development methodologies. The game was used in undergraduate classrooms, as a teaching and learning tool related to computer programming disciplines, focusing on the C language. Preliminary results showed an increase in motivation and interest of students in relation to the learning of the content. Thus, this serious game can be used as a teaching and learning tool as a way to get more data on the problem and simultaneously trying to improve the level of motivation and interest of students about the C language and computer programming concepts.

**Keywords:** Serious Games. Serious Games for Programming Education. Educational Digital Games.



## LISTA DE FIGURAS

Figura 1 - Trecho dos resultados da aplicação formulário pesquisa sobre jogos digitais e linguagem C aplicado a alunos de graduação de computação .....	26
Figura 2 - Classificação dos jogos sérios .....	33
Figura 3 - Uso dos jogos sérios na indústria .....	34
Figura 4 - Classificação dos jogos sérios .....	35
Figura 5 - Relações entre jogos, jogos digitais, jogos sérios educativos .....	38
Figura 6 - Exemplo de tela e sub-tela de um rascunho da interface de um jogo.....	47
Figura 7 - Exemplo de storyboards. ....	55
Figura 8 - Design de Interface e Interação: storyboards. ....	56
Figura 9 - Unreal Engine editor – UDK - Tiro em primeira pessoa – FPS.....	58
Figura 10 - Ambiente da engine UDK com visualização do ambiente do jogo sério desenvolvido nesta dissertação. ....	59
Figura 11 - BomberMan sobre linguagem C .....	61
Figura 12 - Tela do jogo de arrastar e soltar .....	62
Figura 13 - Interface gráfica Block-C.....	63
Figura 14 - Tela do jogo CROCOTILE. ....	64
Figura 15 - Um tela do jogo Robot ON.....	65
Figura 16 - Classificação das Metodologias.....	70
Figura 17 - Fluxo de criação da metodologia customizada .....	71
Figura 18 - Resultado análise do mapeamento sistemático em relação às disciplinas e atividades executadas nas mesmas. ....	72
Figura 19 - Descrição processo concebido para criar uma metodologia customizada de desenvolvimento do jogo sério.....	73
Figura 20 - Execução etapa A processo de geração da metodologia customizada .....	77
Figura 21 - Características específicas disciplina requisitos.....	78
Figura 22 - Características específicas disciplina projeto .....	78
Figura 23 - Características específicas disciplina implementação.....	79
Figura 24 - Características específicas disciplina testes .....	79
Figura 25 - Características específicas disciplina avaliação.....	80
Figura 26 - Execução etapa B processo de geração da metodologia customizada .....	90
Figura 27 - Atividades da disciplina requisitos da metodologia customizada.....	99
Figura 28 - Lista de atividades da disciplina projeto da metodologia customizada .....	99
Figura 29 - Atividades da disciplina implementação da metodologia customizada.....	100
Figura 30 - Lista atividades da disciplina testes da metodologia customizada .....	100
Figura 31 - Lista atividades da disciplina testes da metodologia customizada .....	101
Figura 32 - Metodologia customizada resultante da execução do processo 3.4.....	102
Figura 33 - Desenvolvimento Incremental do Jogo sério.....	103
Figura 34 - Fases e atividades da metodologia após aplicação das melhorias .....	117
Figura 35 - Atividades da fase de pré-produção.....	119
Figura 36 - Trechos das perguntas e opções do questionário discente.....	121
Figura 37 - Tela extraída do jogo com um desafio de análise de códigos .....	125
Figura 38 - Fábrica abandonada representada no jogo sério desenvolvido .....	128
Figura 39 – Personagem do jogo.....	128
Figura 40 - Robô inimigo e um desafio de abrir a porta com o código correto.....	129
Figura 41 - Torre com a fonte de energia principal a ser destruída pelo jogador.....	129
Figura 42 - Código no chão que não estava sendo corretamente visualizado.....	137

Figura 43 - Tratamento visual de elemento que não estava sendo corretamente percebido pelo jogador.....	138
Figura 44 - Storyboards desenvolvidos para o jogo e o resultado final .....	140
Figura 45 - Atividades da fase de produção.....	144
Figura 46 - Construção do mundo virtual .....	144
Figura 47 - Alunos testando o protótipo do jogo.....	153
Figura 48 - Exemplo de programação através do Kismet da UDK .....	156
Figura 49 - Atividades da fase de pós-produção .....	157
Figura 50 - Aplicação jogo laboratório informática com alunos de computação .....	158
Figura 51- Alunos jogando o jogo sério desenvolvido .....	160

## LISTA DE TABELAS

Tabela 1 - Comparação motores de jogos.....	26
Tabela 2 - Papeis no desenvolvimento de um jogo digital .....	43
Tabela 3 - Exemplo de tela e sub-tela de um rascunho da interface de um jogo. ....	49
Tabela 4 - Tópico de um documento de design.....	54
Tabela 5 – Resumo dos jogos sérios sobre linguagem C .....	65
Tabela 6 - Sumário de um documento de design .....	85
Tabela 7 - Verificação da presença da CG na disciplina requisitos.....	91
Tabela 8 - Verificação da presença da CG na disciplina projeto .....	92
Tabela 9 - Tabela verificação da presença da CG na disciplina implementação.....	93
Tabela 10 – Verificação da presença da CG na disciplina testes.....	94
Tabela 11 - Verificação da presença da CG na disciplina avaliação .....	95
Tabela 12 - Verificação da presença características específicas de requisitos .....	96
Tabela 13 - Verificação da presença características específicas de projeto .....	96
Tabela 14 - Verificação da presença características específicas de implementação	97
Tabela 15 - Verificação da presença características específicas de testes.....	97
Tabela 16 - Verificação da presença características específicas de requisitos .....	98
Tabela 17 - Descrição das atividades selecionadas para a disciplina requisitos.....	104
Tabela 18 - Descrição das atividades selecionadas para a disciplina projeto .....	107
Tabela 19 - Descrição atividades selecionadas para a disciplina implementação...	110
Tabela 20 - Descrição das atividades selecionadas para a disciplina testes .....	111
Tabela 21 - Descrição das atividades selecionados para a disciplina avaliação .....	112
Tabela 22 - Lista de melhorias efetuadas na metodologia concebida capítulo 3.....	114
Tabela 21 - Resultados da aplicação 93 questionários discentes .....	122
Tabela 22 - Gêneros de jogos digitais .....	127
Tabela 23 - Documento de requisitos do jogo.....	131
Tabela 24 - Conceito do jogo.....	132
Tabela 25 - Documento de design do jogo .....	133
Tabela 26- Simulação do jogo através de storyboards .....	142
Tabela 27 - Respostas as perguntas acerca da simulação do jogo .....	143
Tabela 28 - Lista de elementos motivacionais inseridos no jogo.....	145

Tabela 29 - Resultados da verificação da presença das heurísticas de Nielsen .....	151
Tabela 30 - Transcrições das falas dos alunos após a aplicação do think-aloud ....	153
Tabela 31 - Transcrição das entrevistas realizados com os alunos .....	154
Tabela 33 - Lista erros técnicos, falhas usabilidade, ludicidade, valor educacional	159
Tabela 34 - Resultados da avaliação do jogo sério UNLOCK-C .....	161
Tabela 35 – Questões de avaliação de ganho educacional e projeto do jogo sério	162
Tabela 36 - Percepção dos alunos após jogarem o jogo sério desenvolvido .....	164

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	15
1.1 MOTIVAÇÃO .....	15
1.2 OBJETIVO.....	19
<b>1.2.1 Objetivo Geral</b> .....	19
<b>1.2.2 Objetivos Específicos</b> .....	20
1.3 JUSTIFICATIVA .....	20
1.4 METODOLOGIA.....	27
1.5 ESTRUTURA DO TRABALHO.....	27
<b>2 JOGOS SÉRIOS</b> .....	29
2.1 VISÃO GERAL SOBRE JOGOS SÉRIOS.....	29
<b>2.1.1 Conceitos fundamentais sobre jogos sérios</b> .....	29
<b>2.1.1.1 Jogo</b> .....	29
<b>2.1.1.2 Jogos digitais</b> .....	30
<b>2.1.1.3 Jogos sérios</b> .....	31
<b>2.1.2 Classificação dos jogos sérios</b> .....	32
<b>2.1.3 Jogos sérios no contexto educacional</b> .....	36
2.2 DESENVOLVIMENTO DE JOGOS SÉRIOS.....	40
<b>2.2.1 Conceitos envolvidos</b> .....	41
<b>2.2.1.1 Game design</b> .....	41
<b>2.2.1.2 Jogabilidade</b> .....	44
<b>2.2.1.3 Prototipagem</b> .....	44
<b>2.2.1.4 Conceito do Jogo (game concept)</b> .....	45
<b>2.2.1.5 Interface</b> .....	46
<b>2.2.2 Principais papéis envolvidos</b> .....	48
<b>2.2.2.1 Especialista de domínio</b> .....	49
<b>2.2.2.2 Designer de jogos</b> .....	50
<b>2.2.2.3 Testadores de software</b> .....	50
<b>2.2.2.4 Programador</b> .....	51
<b>2.2.3 Principais artefatos gerados</b> .....	52
<b>2.2.3.1 Documento de requisitos</b> .....	52
<b>2.2.3.2 Documento de design</b> .....	52

2.2.3.3 Storyboards .....	55
2.2.4 Ferramentas de apoio a criação de jogos.....	56
2.2.4.1 Ferramentas de autoria.....	56
2.2.4.2 Motores de jogos (Engines) .....	57
2.2.5 Fases do desenvolvimento de jogos .....	59
2.3 JOGOS SÉRIOS COMO APOIO AO ENSINO E APRENDIZAGEM DE LINGUAGENS DE PROGRAMAÇÃO .....	60
2.3.1 Jogos sérios no ensino de programação.....	61
2.4 CONCEITOS SOBRE LINGUAGEM DE PROGRAMAÇÃO.....	66
2.4.1 Linguagem de Programação C.....	66
<b>3 PROCESSO DE DEFINIÇÃO DA METODOLOGIA DE DESENVOLVIMENTO DO JOGO SÉRIO EDUCACIONAL .....</b>	<b>68</b>
3.1 INTRODUÇÃO .....	68
3.2 NECESSIDADE DEFINIÇÃO DE METODOLOGIA CUSTOMIZADA.....	68
<b>3.2.1 Classificação das atividades em um conjunto de disciplinas comuns da Engenharia de Software.....</b>	<b>71</b>
3.3 PROCESSO PARA A GERAÇÃO DA METODOLOGIA.....	72
<b>3.3.1 Descrição do processo para geração da metodologia .....</b>	<b>73</b>
3.4 EXECUÇÃO DO PROCESSO DE GERAÇÃO DA METODOLOGIA CUSTOMIZADA .....	76
3.4.1 Disciplina requisitos .....	90
3.4.2 Disciplina projeto .....	92
3.4.3 Disciplina implementação .....	93
3.4.4 Disciplina testes .....	94
3.4.5 Disciplina avaliação .....	95
3.4.6 Disciplina requisitos .....	96
3.4.7 Disciplina projeto .....	96
3.4.8 Disciplina implementação .....	97
3.4.9 Disciplina testes .....	97
3.4.10 Disciplina avaliação .....	98
3.5 MELHORIAS NA METODOLOGIA CONCEBIDA .....	113
<b>4 DESENVOLVIMENTO DO JOGO SÉRIO COM BASE NA METODOLOGIA CUSTOMIZADA.....</b>	<b>119</b>

4.1 ATIVIDADES DE PRÉ-PRODUÇÃO .....	119
4.1.1 Atividades acerca do público alvo e domínio do problema .....	119
4.2 ATIVIDADES PEDAGÓGICAS.....	123
4.2.1 Atividades sobre o conceito do jogo.....	125
4.2.2 Atividades de criação e gerenciamento de documentos do projeto.....	130
4.2.3 Atividades de concepção de elementos do jogo.....	137
4.3 ATIVIDADES DE PRODUÇÃO .....	143
4.3.1 Atividades de criação de protótipos e desenvolvimento do jogo até a sua versão final.....	150
4.4 ATIVIDADES DE PÓS-PRODUÇÃO.....	157
4.4.1 Atividades de testes de pós-produção.....	157
4.4.2 Atividades de avaliação do jogo .....	161
5 CONSIDERAÇÕES FINAIS .....	167
5.1 CONCLUSÕES .....	167
5.2 CONTRIBUIÇÕES DO TRABALHO.....	169
5.3 TRABALHOS FUTUROS .....	170
REFERÊNCIAS .....	172
APÊNDICE A – METODOLOGIA DE DESENVOLVIMENTO DE UM JOGO SÉRIO EDUCACIONAL: VERSÃO COM APLICAÇÃO DAS MELHORIAS .....	181
APÊNDICE B – QUESTIONÁRIO PESQUISA DISCENTE .....	186
APÊNDICE D – DOCUMENTO COM VOCABULÁRIO COMPARTILHADO .....	188
APÊNDICE E – DOCUMENTO DE REQUISITOS .....	189
APÊNDICE F – DOCUMENTO DE DESIGN E CONCEITO DO JOGO.....	190
APÊNDICE G – ITENS PRONTOS DO MOTOR DE JOGO UDK UTILIZADOS NO JOGO SÉRIO.....	195

## 1 INTRODUÇÃO

Este capítulo tem como objetivo contextualizar o cenário no qual a proposta desta dissertação se situa, apresentar e justificar o objetivo desta dissertação, e apresentar sucintamente a metodologia de trabalho e a estrutura da dissertação.

### 1.1 MOTIVAÇÃO

A expansão tecnológica e o aumento da influência da computação na sociedade humana, faz com que cursos da área de computação tornem-se cada vez mais importantes (BURKE, 2016). Outra referência que destaca a importância estratégica dos cursos de computação são os resultados obtidos pelo estudo do observatório da Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), em 2013.

Estes dados projetam que haverá, no Brasil, um déficit de cerca de 408 mil profissionais de Tecnologia da Informação(TI) em 2022. Indicam uma demanda estimada de profissionais de cerca de 1,669 milhões contra 1,261 milhões efetivamente contratados. O relatório da SOFTEX (2013) identifica que devem haver ações urgentes em relação ao déficit, para mitigação deste obstáculo ao crescimento da Indústria Brasileira de Software e Serviços de TI.

Tendo como referência esse cenário, a formação de novos profissionais de TI, se configura como um ponto chave para tentar diminuir o déficit projetado pelos resultados do relatório da pesquisa realizada pela SOFTEX (2013). Essa formação passa geralmente pela academia. Na formação acadêmica de um profissional de TI, frequentemente estão presentes, nas grades curriculares dos cursos de Bacharelado em Ciência da Computação, Engenharia de Computação, Bacharelado em Sistemas de Informação e Cursos Superiores de Tecnologia: Tecnologia em Redes e Tecnologia em Desenvolvimento para WEB, disciplinas de programação de computadores e disciplinas relacionadas a linguagens de programação.

Essas disciplinas relacionadas com programação de computadores são reconhecidas, como indicam Berry e Kölling, (2016) como elementos fundamentais em Ciência da Computação. É o processo pelo qual as ideias conceituais são mapeadas para instruções que podem ser compreendidas e interpretadas por uma máquina. No entanto, é justamente o aprendizado ineficiente de programação que Azevedo e Paula (2011) indicaram como o principal problema que os alunos



enfrentam em cursos de computação. Esta ineficiência de aprendizado contribui para as altas taxas de desistência (evasão) dos cursos da área de computação.

Dificuldades referentes ao processo de ensino e aprendizado dessas disciplinas contribuem para esse cenário de ineficiência de aprendizado de programação. Algumas destas dificuldades estão relacionadas com a área educacional. Khenissi, Essalmi e Jemni (2013), relatam que, no ensino e aprendizado de uma linguagem de programação, a baixa motivação, a complexidade do assunto e a diferença existente entre as gerações de professores e alunos, se configuram como barreiras ao aprendizado. Já para os autores Bittencourt et al., (2013) os professores da área de computação geralmente utilizam estratégias de ensino excessivamente formais, o que torna o aprendizado relativamente árduo.

A falta de motivação também é apontada por Soares e Zaluar (2015), como importante causa do baixo desempenho e das altas taxas de reprovação nas disciplinas, e de desistência do curso de computação por parte dos alunos. Os autores indicam que esta situação ocorre geralmente no início do curso, pois os alunos encontram dificuldades e não sabem como lidar com elas. Penteado e Damasceno (2014) alertam, inclusive, que a disciplina de programação de computadores possui um elevado índice de reprovação e desistência o que dificulta ou até mesmo impede a continuidade dos alunos em cursos de computação.

Outros autores corroboram com esta questão e indicam novas ou ratificam a presença dessas dificuldades. Kinnunen e Malmi (2006) declaram que os alunos novatos geralmente indicam a dificuldade no aprendizado da programação de computadores, como principal motivo de sua reprovação nas disciplinas relacionadas, ou na sua desistência do curso. Bittencourt et al., (2013) sinalizam que o período de tempo que os alunos novatos possuem para estudar diversos conceitos complexos é muito curto. Os alunos precisam aprender a utilizar ferramentas de programação, como IDEs (ambientes integrados de desenvolvimento) e compiladores, estudar conceitos de algoritmos e estrutura de dados, aprender uma linguagem de programação, e ainda desenvolver habilidades relacionadas com a lógica computacional e abstração.

No que se refere as IDEs citadas anteriormente, Penteado e Damasceno (2014) comentam que essas ferramentas, que são utilizadas pelos alunos para programar em alguma linguagem de programação, possuem muitos conceitos e

características que dificultam o aprendizado do aluno novato. Os ambientes integrados de desenvolvimento (IDEs) são geralmente desenvolvidos para atender as necessidades profissionais dos programadores e não de seu aprendizado. Desta forma, muitos conceitos presentes nestas ferramentas são complexos para alunos iniciantes, e as mensagens de erros e avisos dos compiladores integrados a IDE, podem ser difíceis de entender para um aluno que está iniciando na programação.

Outras dificuldades estão relacionadas com o conceito de algoritmos. Construir algoritmos faz parte da trajetória de aprendizado de um aluno de programação. Falckembach e Araujo (2003) relatam que a aprendizagem de algoritmos é uma área recente e o processo mental requerido ao aluno iniciante na construção destes algoritmos representa uma dificuldade a seu aprendizado. Eles argumentam também que um grande desafio do professor de algoritmos e programação é fornecer ao aluno, um atendimento personalizado e adaptado em sala de aula em relação ao seu perfil de aprendizado.

Em uma Educação que segue geralmente o processo de ensino tradicional, a padronização é quem rege a metodologia do professor. Assim esta personalização e adaptação são difíceis de realizar e a metodologia de ensino do conteúdo de algoritmos acaba sendo a mesma para todos os alunos. Portanto, o nível de conhecimento de cada aluno, as expectativas, preferências e suas necessidades não são considerados (FALCKEMBACH; ARAUJO, 2003).

As linguagens de programação são parte integrante deste processo de ensino e aprendizado de algoritmos e programação. A linguagem de programação C é uma das linguagens frequentemente utilizadas nos cursos de Bacharelado em Ciência da Computação, Engenharia de Computação, Bacharelado em Sistemas de Informação e Cursos Superiores de Tecnologia: Tecnologia em Redes e Tecnologia em Desenvolvimento para WEB. Portanto, dificuldades de aprendizado de programação de computadores utilizando a linguagem C se configuram como importantes barreiras ao êxito dos alunos nas disciplinas de programação.

Charalampos, Nektarios e Stavros (2015) acreditam que as características inerentes à linguagem de programação C fazem com que os alunos novatos enfrentem diversos obstáculos ao programar utilizando esta linguagem. A orientação da linguagem para baixo nível de abstração, sua sintaxe obscura, e mensagens de

erros de compilação e execução muitas vezes ambíguas e abstratas são alguns destes obstáculos.

Estas dificuldades relatadas anteriormente contribuem para diminuir o número de alunos concluintes da área de computação e conseqüentemente diminuí também o número de profissionais para atuar na área da tecnologia da informação (TI). Nas aulas ministradas, pelo autor deste trabalho, desde 2010, na Universidade Salvador (Unifacs), nos cursos de Bacharelado em Ciência da Computação e Bacharelado em Sistemas de Informação, foram percebidas estas dificuldades apontadas pelos autores citados anteriormente. Esta situação permitiu identificar uma oportunidade de pesquisa e motivou o pesquisador deste trabalho, a escolher o tema desta dissertação. A tentativa é contribuir com uma ferramenta no formato de jogo sério digital 3D para ser utilizado por professores e alunos na tentativa de mitigar os riscos de reprovação dos alunos novatos nas disciplinas de programação dos cursos de computação, no que se refere às dificuldades de aprendizado de programação de computadores com linguagem de programação C.

No entanto, desenvolver um jogo sério digital de cunho educacional é um projeto complexo, que demanda conhecimentos em diversas áreas, dentre as quais destacam-se a engenharia de software e a pedagogia. Em relação a engenharia de software, “não se trata apenas do programa em si, mas de toda a documentação associada e dados de configurações necessários para fazer esse programa operar corretamente”.(SOMMERVILLE, 2011, p.3). Já a pedagogia irá nortear a forma como os conteúdos de aprendizagem serão inseridos e avaliados no jogo sério. A pedagogia deverá responder o que será abordado, como será esta abordagem, por que, para que e para quem estes conteúdos estão sendo incluídos no jogo. Os conteúdos educacionais necessitam ter clareza em relação ao que se almeja ensinar, desenvolver e avaliar. Em adição, é necessário conhecimento específico de desenvolvimento de jogos eletrônicos. Um desafio na criação de jogos sérios é “a dificuldade em manter um equilíbrio entre aprendizagem, diversão e motivação”. (FRANZWA; TANG; JOHNSON, 2013 apud OLIVEIRA; HOUNSELL; KEMCZINSKI, 2014, p.6).

A utilização de uma metodologia especializada no desenvolvimento de jogos sérios educacionais poderá contribuir para gerenciar a complexidade inerente ao processo de desenvolvimento desses jogos. Adicionalmente, ampliam-se de forma

significativa as chances de que o produto gerado seja efetivo sob o aspecto das boas práticas da engenharia de software e suporte pedagógico. A metodologia, cuja definição é descrita no capítulo 3, será utilizada para desenvolver o jogo sério educacional.

## 1.2 OBJETIVO

Esse trabalho envolve a concepção, desenvolvimento e utilização de um jogo sério educacional com representação tridimensional(3D) sofisticado, de gênero mais alinhado com os jogos digitais 3D jogados pelos alunos nativos digitais. O desenvolvimento desse jogo utilizou uma metodologia de desenvolvimento customizada e otimizada a partir de metodologias pré-existentes de desenvolvimento de jogos sérios educacionais. O jogo sério será utilizado como ferramenta de ensino e aprendizado de conceitos relacionados a disciplinas de programação de computadores, com foco na linguagem C. Espera-se através da abordagem de utilização deste jogo sério em sala de aula, um aumento da motivação, e interesse dos alunos em relação ao aprendizado dos conteúdos das disciplinas. É esperado adicionalmente que esta abordagem contribua diretamente para permitir que o aluno exercite seu conhecimento em relação à linguagem C.

O jogo é sério, em 3D, sofisticado e desafiador, e trabalha conceitos relacionados à programação de computadores utilizando a linguagem C. O aluno poderá praticar seu conhecimento e aprender com seus erros. O jogo poderá ser utilizado como ferramenta de apoio ao ensino pelo professor da disciplina. Pressupõe-se que o jogo poderá deixar mais atrativo, para os alunos novatos, os conceitos relacionados com programação utilizando linguagem C, e que estimule estes alunos a querer aprender os tópicos conceituais das disciplinas de programação de computadores. A medição dos resultados esperados será realizada através de pós-testes, questionários de percepção e entrevistas com alunos e professores.

### 1.2.1 Objetivo geral

Desenvolver um jogo sério educacional 3D através de uma metodologia de desenvolvimento de jogos customizada para jogos sérios, que possua gráficos avançados e recursos aprimorados, para investigar a possibilidade de utilização

deste jogo como ferramenta de apoio ao ensino e aprendizado de linguagem de programação C, em disciplinas de programação de computadores.

### **1.2.2 Objetivos Específicos**

Os objetivos específicos deste trabalho são:

a) Conceber uma metodologia customizada a partir de outras metodologias de desenvolvimento de jogos sérios já existentes para embasar o desenvolvimento do jogo sério educacional.

b) Criar uma ferramenta que permita ao especialista de domínio (professor) apoiar seu processo de ensino de programação que utilize linguagem C.

c) Desenvolver um software (jogo sério) utilizando esta metodologia customizada e produzir os artefatos através da execução das atividades desta metodologia.

d) Analisar e avaliar o jogo sério desenvolvido sob o ponto de vista dos alunos e professores e apresentar os resultados.

### **1.3 JUSTIFICATIVA**

A utilização de software como ferramentas de apoio ao ensino e aprendizagem tem sido um importante foco de estudo. Em especial diversos software, da categoria jogos digitais, foram desenvolvidos com este propósito. Apesar disso, Santaella (2012), destaca que o mercado pedagógico tem explorado pouco os jogos digitais educacionais. Sendo assim, Clua e Bittencourt (2005) comentam que torna-se interessante incentivar o desenvolvimento de novos jogos digitais de qualidade, para serem utilizados em sala de aula e em outros ambientes de aprendizagem.

A adoção, conjuntamente, de estratégias pedagógicas e inovações tecnológicas, podem ajudar na melhoria do processo de ensino e aprendizagem. É possível aumentar a assimilação de um conteúdo através da utilização de recursos tecnológicos que permitam o aluno estudar no seu ritmo e seus objetivos (FALCKEMBACH; ARAUJO, 2003). Penteado e Damasceno (2014), comentam que realmente a área de Educação foi impactada diretamente com a revolução tecnológica, e isso está modificando em alguns casos, ou ainda pode modificar em outros, o processo de ensino tradicional. O professor não está sendo mais

considerado o centro do ensino, e o foco deve está agora no aluno e em suas capacidades e possibilidade de absorver melhor o conteúdo da disciplina.

Os motivos que incentivam ao desenvolvimento destes jogos digitais parecem estar inseridos nos problemas existentes em ensinar conceitos tecnológicos através de formas de ensino tradicionais. Segundo Vygotsky (1988 apud COELHO; COSTA, 2016), para que o processo de ensino e aprendizagem possa ocorrer e seja estabelecido de forma eficaz, é importante que aprendizagem seja significativa. Desta forma, esta aprendizagem deve incentivar no aluno, uma disposição para apreender o novo conteúdo, “uma base de informações que sirva como material de apoio durante o processo, o papel do professor como um facilitador de transmissão, transformação e geração de objetos de saber”.

Esta disposição para aprender o novo conteúdo pode ser atrelada a ferramentas tecnológicas, já que, os alunos que na sua maioria pertencem a nova geração, já possuem uma inclinação, segundo Prensky (2007), para descobrir o novo principalmente nos elementos de tecnologia. Prensky, inclusive cunhou e utiliza o termo nativo digital para categorizar estes indivíduos nascidos a partir da geração Y, que possuem características relacionadas com a tecnologia da informação.

Para Prensky esses indivíduos são nativos, pois já nasceram na era digital. Já os imigrantes digitais tiveram que migrar da era analógica para a digital. Um nativo digital é capaz de ser multitarefa e gosta de experimentar novas tecnologias, mesmo diante de desafios e barreiras de sua utilização. O autor relata que esta coragem para descobrir o novo, principalmente nos elementos de tecnologia, precisa ser canalizada na educação para que as novas ferramentas de tecnologia possam estar inseridas no contexto de ensino e aprendizagem das disciplinas. Na mesma direção, Coelho e Costa (2016) destacam que é necessário desenvolver e estimular novas ferramentas que auxiliem no processo de ensino e aprendizagem de alunos desta nova geração. Por isso, cada vez mais os jogos são inseridos no contexto escolar, em especial os jogos digitais.

Os alunos nativos digitais parecem mostrar insatisfação com relação à utilização de abordagens de ensino tradicionais. Mayo (2007, apud OLIVEIRA, 2016), realizou um estudo em cursos de Engenharia dos Estados Unidos e constatou que 98% dos alunos que desistiram do curso, indicaram estar insatisfeitos com a pedagogia de aulas expositivas adotada nos cursos de Engenharia. A mesma

insatisfação com esta pedagogia tradicional foi declarada por 86% dos graduandos destes cursos. Estes alunos relataram que consideram inadequados, os métodos de ensino atuais, e que poderiam estar mais engajados se existisse formas inovadoras e com mais entretenimento, como o presente nos jogos digitais.

Consideradas as barreiras de aprendizagem dos alunos e dificuldades de ensino dos professores de programação relatados na seção 1.1, é relevante considerar o desenvolvimento de alguma ferramenta tecnológica para tentar ultrapassar estas barreiras e dificuldades. Já que o público alvo são nativos digitais, criar uma ferramenta a qual eles estejam acostumados a utilizar em seu cotidiano tem possivelmente potencial para auxiliar nesta tarefa.

Neste sentido, os jogos educacionais que “são aqueles que ensinam enquanto entretêm.” (GATES, 2004), se configuram como iniciativas desenvolvidas, como destaca Gebremichael (2016), para auxiliar o jogador no aprendizado de um assunto específico. No que se refere especificamente as disciplinas de programação de computadores, esta necessidade de desenvolvimento e pesquisa de novas ferramentas também é colocada como importante por parte dos pesquisadores. Novos jogos são concebidos a partir de pesquisas para que possam tentar mitigar os riscos dos alunos relacionados com a reprovação nestas disciplinas e tentar evitar até mesmo, a desistência dos mesmos dos cursos de computação.

De fato, nos últimos anos, como destacam Paralič e Pietriková (2014), ocorreu uma popularização da ideia de usar jogos digitais em cursos de introdução à programação. Assim alguns trabalhos têm sido desenvolvidos com o foco na criação de jogos digitais, com o objetivo de deixar mais interessante e efetivo a aprendizagem e ensino de programação de computadores. Apesar disso, este recurso ainda não é muito utilizado na grande maioria dos cursos de programação existentes.

Neste sentido, a utilização de jogos digitais em atividades de ensino e aprendizagem de disciplinas de programação de computadores pode ser eficaz e oferecer benefícios (PESSINI et al., 2014; PIETRUCHINSKI; C. NETO; REINEHR, 2011; PARALIČ, PIETRIKOVÁ, 2014). Os autores Malliarakis, Satratzemi e Xinogalos, (2014), comentam que já que os alunos estão familiarizados com o ambiente gráfico e cenário dos jogos digitais, a adoção destes jogos digitais tornará

os alunos mais ativos em relação a sua aprendizagem. Desta forma, está sendo integrada esta nova tendência de utilização de jogos digitais na educação.

Hakulinen (2011) indica que cursos relacionados com programação e que utilizam linguagem de programação, são favoráveis à utilização de jogos digitais que envolvam tarefas ligadas a programação. Meeklai et al., (2015) relatam que se comparado com o ensino através de programação baseada em texto, a utilização de jogos digitais, poderá ocasionar um aumento na motivação do aluno e uma diminuição na curva de aprendizado dos conceitos pertencentes às disciplinas de programação.

Neste sentido, Mayo (2007, apud OLIVEIRA, 2016) comenta que a utilização de jogos digitais educacionais pode ser uma possível solução para diminuir a alta taxa de evasão de alunos. Diante disso, faz-se necessário propor ferramentas para apoiar o ensino de programação e tentar mitigar os riscos envolvidos na trajetória de aprendizado do aluno iniciante.

Contudo Paula e Valente (2016) alertam que embora os jogos digitais se configurem como uma possibilidade de ferramenta para uma transformação pedagógica positiva do ensino, a utilização destes jogos não é uma solução mágica. O uso destes jogos para atingir o objetivo de integrar jogos e Educação requer muito trabalho e não é uma atividade simples de realizar. O fato de inserir jogos como elementos pedagógicos não garante que os alunos ficarão motivados e que aprendam melhor os conteúdos.

É importante destacar que desenvolver jogos digitais educacionais se configura como um processo que envolve uma ampla e complexa gama de tecnologias de desenvolvimento e enfoques conceituais que ainda estão em evolução. Este desenvolvimento envolve a área de engenharia de software e outras áreas de conhecimento, em especial a pedagogia. Além destas áreas de conhecimento, diversas áreas de estudo relacionados ao tema, necessitam de novos estudos, para preencher as lacunas de pesquisa em aberto.

Apesar destes alertas, percebe-se, no entanto, como destacam Soares e Zaluar (2015), que um efeito possivelmente favorável é efetuar uma mudança na metodologia de ensino do conteúdo pertencente às ementas das disciplinas. Neste contexto de mudanças na metodologia de ensino, estão presentes uma categoria de jogos digitais denominada jogos sérios (em inglês, serious games). Um jogo sério,



como definido por Zyda (2005), é um jogo digital que utiliza o entretenimento para promover, entre outros, a educação. A função principal do jogo sério não é o entretenimento e sim o objetivo o qual ele se propõe.

Uma pesquisa publicada pelo BNDES (2014) indicou que no ano de 2013, 133 das empresas desenvolvedoras de jogos no Brasil, levantadas pela pesquisa, produziram 1.417 títulos de jogos. Destes 678 jogos, 47,9% do total dos jogos desenvolvidos pertencem a categoria de serious games (jogos sérios). Ou seja, a maior parte dos jogos desenvolvidos pertence à categoria de jogos sérios o que revela uma demanda crescente por este tipo de jogo.

Na tentativa de utilizar jogos digitais como ferramentas de apoio ao ensino e aprendizagem de disciplinas relacionadas com programação de computadores, alguns jogos sérios foram desenvolvidos. Apesar de existirem um número expressivo de pesquisas que criam jogos digitais para o ensino de programação, o número de pesquisas que criaram jogos sérios com foco específico em linguagem de programação C é pequeno. Conforme comentado no capítulo 2, alguns jogos sérios educacionais com objetivo de ensino de programação utilizando linguagem C foram desenvolvidos. Contudo, estes jogos não utilizam gráficos 3D e nem são semelhantes aos jogos digitais que os alunos dos cursos de computação estão acostumados a jogar.

Corroborando com esta questão, Clua e Bittencourt (2005) alertam que para que estes jogos se diferenciem dos jogos didáticos padrões, eles necessitam ser mais atraentes e imersivos. Para criar jogos divertidos, atrativos, lúdicos, podem ser aplicadas técnicas de criação de jogos digitais 3D como forma de tentar atingir os objetivos pedagógicos estabelecidos para o jogo.

Considerando, como já comentado pelos autores supracitados, que o aprendizado de programação não é tarefa simples. Considerando que a linguagem C está presente em muitas disciplinas introdutórias de programação, é necessária uma atenção especial na tentativa de criar métodos, processos ou ferramentas que possam auxiliar o professor, e principalmente o aluno, na possibilidade de obtenção do sucesso no ensino ou aprendizado de programação utilizando a linguagem C.

Ao analisar os trabalhos acerca de jogos digitais com foco na linguagem de programação C, detectou-se uma oportunidade de pesquisa em relação ao desenvolvimento de um jogo sério no padrão 3D, pertencente a algum gênero de

jogo digital que seja mais próximo do cotidiano de uso dos alunos dos cursos de computação, do que os já concebidos em pesquisas anteriores.

Já que os jogos criados para esta finalidade não possuem recursos aprimorados e nem estão próximos dos gêneros dos jogos digitais jogados pelos alunos, pressupõe-se que ao criar um jogo digital sofisticado, ocorrerá uma melhor imersão dos alunos frente aos desafiadores conteúdos de programação. Desta forma, é justificável o desenvolvimento de um jogo sério para auxiliar no ensino e aprendizado de programação que utiliza a linguagem C como suporte.

Assim, presume-se que um jogo sério educacional com gráficos 3D aprimorados, pertencente a um gênero de jogos conhecido e frequentemente jogado pelos alunos de computação, possa ajudar no aumento da motivação e no engajamento destes alunos em relação às disciplinas de programação que utilizam a linguagem C.

Um jogo de tiro em primeira pessoa (do inglês first-person shooter, FPS) é um gênero de jogo digital, que o jogador utiliza armas de fogo e visualiza o cenário do jogo a partir do ponto de vista do protagonista, como se o jogador e personagem do jogo fossem o mesmo observador (WIKIPEDIA, 2016).

Escolheu-se nesta dissertação focar neste gênero específico de jogo digital 3D de tiro em primeira pessoa (FPS – First Person Shooter), pelos seguintes motivos:

- a) Uma pesquisa realizada com 93 alunos em formulário objetivo indicou, conforme relatado na figura 1, que o gênero FPS é o preferido dentre os demais gêneros pesquisados e que a maioria dos alunos jogaria um jogo 3D FPS sobre linguagem C.
- b) O software de desenvolvimento(motor de jogo) UDK, escolhido para criar o jogo sério 3D facilita o desenvolvimento de jogos do tipo FPS.

Figura 1 - Trecho dos resultados da aplicação formulário pesquisa sobre jogos digitais e linguagem C aplicado a alunos de graduação de computação

<b>Que tipo de jogo mais gosta de jogar?</b>	
(55,91%) Tiro primeira/terceira pessoa (FPS) (44,08%) RPG (43,01%) Estratégia (38,70%) Ação/Aventura	(33,34%) Multiusuário (32,26%) Futebol (26,88%) Luta (23,66%) Esportes
<b>Você gosta de jogos do tipo FPS (tiro em primeira pessoa)?</b>	
(81,72%) Gosto ( 3,22% ) Não gosto	
<b>Você jogaria um jogo FPS (tiro em primeira pessoa) sobre linguagem C?</b>	
(78,49%) Sim (17,20% ) Não (2,15%) Sim se não tiver sangue ou muita violência	
<b>Prefere jogos em:</b>	
( 20,43% ) 3D ( 77,42% ) AMBOS (4,3%) 2D	

O resultado da pesquisa com os alunos viabilizou a utilização do software UDK(Unreal Development Kit) para o desenvolvimento do jogo sério. Apesar do software facilitar a criação de jogos FPS 3D, ele somente poderia ser utilizado se os alunos indicassem inclinação para jogar este tipo de jogo, o que fora observado na pesquisa citada anteriormente. A escolha do motor de jogo UDK também baseou-se em comparações com outros motores de jogos mais utilizados. A tabela a seguir mostra estas comparações.

Tabela 1 - Comparação motores de jogos

Componente	Unity3D	UDK	CryEngine 3	Torque 3D
Suporte Gráfico	3D	3D	3D	3D
Animação	Sim	Sim	Sim	Sim
Áudio	2D & 3D	2D & 3D	2D & 3D	2D & 3D
Scripting	Javascript; C#.NET	UC	LUA	TS
Inteligência Artificial	Scripts	Scripts; Pathfinding; Decision Mak- ing	Scripts; Pathfinding; Decision Mak- ing	Scripts
Motor de física	Detecção de Colisões; Rigid Body; Vehicle Physics	Detecção de Colisões; Rigid Body; Vehicle Physics	Detecção de Colisões; Rigid Body; Vehicle Physics	Detecção de Colisões; Rigid Body; Vehicle Physics
Importação de conteúdo CAD	3ds max; maya	3ds max; maya	3ds max; maya	3ds max; maya
Editor de ambiente integrado	Sim	Sim	Sim	Sim
Código Fonte	Não	Não	Sim	Sim
Curva de aprendizagem	Média	Elevada	Média	Média
Documentação e tutoriais	Sim	Sim	Sim	Sim
Licença	Gratuita	Gratuita	Gratuita <sup>a</sup>	\$99
Suporte de Rede	Cliente/Servidor	Sim	Sim	Sim
Integração em web browser	Sim	Não	Não	Sim
Plataformas	PC; Mac; XBox; iOS	PC; iOS	PC; PSX; XBox; Game- cube	PC; Mac

Fonte: Gonçalves (2011, p.13).

A UDK é gratuita, possui itens de física aprimorados, é focada em jogos 3D, possui integração com outros softwares de criação de itens gráficos do jogo, etc. O conjunto destas características aliado ao fato da UDK privilegiar a criação de um jogo FPS culminou na decisão de usar este motor de jogo para o desenvolvimento do jogo Unlock-C.

#### 1.4 METODOLOGIA

O trabalho foi realizado através das seguintes atividades:

- a) Revisão bibliográfica referente ao desenvolvimento de jogos digitais educacionais.
- b) Análise de trabalhos específicos relacionados com jogos sérios.
- c) Análise de trabalhos relacionados com metodologia de desenvolvimento de jogos sérios.
- d) Customização de uma metodologia de desenvolvimento de jogos sérios a partir de outras 17 metodologias compiladas no mapeamento sistemático de (OLIVEIRA; HOUNSELL; KEMCZINSKI, 2014)
- e) Aplicação da metodologia customizada na construção do jogo sério.
- f) Aplicação do jogo sério desenvolvido em sala de aula.

#### 1.5 ESTRUTURA DO TRABALHO

Este trabalho de dissertação está estruturado em capítulos e será desenvolvido da seguinte forma:

O capítulo 1 da dissertação apresenta motivação, objetivo, justificativa, metodologia e estrutura da dissertação.

O capítulo 2 apresenta o estado da arte dos temas abordados pela dissertação: jogos, jogos digitais, jogos sérios, desenvolvimento de jogos sérios, exemplos de jogos sérios, uso educacional de jogos sérios, linguagem de programação, linguagem C, desafios e dificuldades de ensino e aprendizagem de programação e linguagem C.

O capítulo 3 define um processo de geração de uma metodologia de desenvolvimento de um jogo sério educacional. Posteriormente o capítulo apresenta

a execução desse processo para gerar a metodologia customizada que foi utilizada no desenvolvimento do jogo sério educacional.

O capítulo 4 descreve as melhorias efetuadas na metodologia concebida no capítulo 3 e mostra a construção do jogo sério educacional, utilizando a metodologia aprimorada neste capítulo. No capítulo também é relatado os resultados da aplicação em sala de aula, do jogo sério desenvolvido e suas características.

O capítulo 5 apresenta considerações finais, contribuições da dissertação, e trabalhos futuros.

Apêndice A: mostra o resultado final das melhorias realizadas na metodologia, que embasou o desenvolvimento do jogo sério educacional.

Apêndice B: Mostra o questionário aplicado na pesquisa com os alunos de Sistemas de Informação e Ciência da Computação.

Apêndice C: Mostra os exercícios de pós-testes que foram realizados com os alunos após jogarem o jogo sério desenvolvido neste trabalho.

Apêndice D: Contém os termos utilizados no vocabulário compartilhado pelas partes interessadas durante o projeto de desenvolvimento do jogo.

Apêndice E: Contém uma lista de requisitos funcionais e não funcionais levantados durante o desenvolvimento do jogo sério.

Apêndice F: Descreve em sua totalidade todo o documento de design e conceito do jogo escrito no decorrer do projeto de desenvolvimento do jogo sério.

## 2 JOGOS SÉRIOS

Este capítulo tem o objetivo de apresentar uma visão geral sobre jogos sérios (do inglês serious games). São abordados conceitos fundamentais sobre este tipo de jogo digital, seus tipos, algumas metodologias de desenvolvimento e sobre a utilização de jogos sérios com objetivo educacional. Com maior abrangência são analisados os estudos acerca destes jogos sérios.

Já que o objetivo deste trabalho de dissertação é desenvolver um jogo sério, a grande parte dos conceitos deste capítulo estará relacionada a este tema. Notadamente há conceituações sobre jogos e jogos digitais que darão embasamento para a posterior discussão acerca dos jogos sérios.

### 2.1 VISÃO GERAL SOBRE JOGOS SÉRIOS

Nesta seção define-se o que é um jogo, o que é um jogo digital e o que é um jogo sério.

#### 2.1.1 Conceitos Fundamentais Sobre Jogos Sérios

##### 2.1.1.1 Jogo

De acordo com Kishimoto (1997, p.13), definir o conceito de jogo não é uma empreitada fácil, pois o entendimento do assunto pode ser abordado por diversos pontos de vista. “Pode-se estar falando de jogos políticos, de adultos, crianças, animais ou amarelinha, xadrez, ...”.

Crawford (1982), indica que os jogos são uma parte fundamental da existência humana, e a ampla penetração dos jogos em todo o espectro da experiência humana, nos apresenta duas barreiras potenciais para a compreensão dos jogos. Primeiro, nosso livre uso de termos sobre jogos promove uma percepção exagerada da nossa própria compreensão de jogos, e sistematicamente ignoramos as complexidades acerca do projeto de um jogo.

O segundo obstáculo é a ambiguidade. Nós aplicamos os princípios e conceitos de jogos tão amplamente que talvez tenha-se diluído os seus significados originais. Não há mais um foco claro para os conceitos que procuramos entender. Os criadores de jogos não têm bem definido um conjunto de termos comuns para comunicação uns com os outros. As discussões sobre design de jogos

frequentemente desintegram-se em argumentos sobre semântica (CRAWFORD, 1982, p.5).

De acordo com Huizinga (2007), um pré-requisito para que exista cultura, é a existência de uma sociedade humana. Sendo assim, já que o jogo esteve presente na história da humanidade muito antes do surgimento de antigas e remotas sociedades humanas, ele é, portanto, mais antigo que a própria cultura. O autor indica que jogo pode ser definido como uma atividade livre e lúdica que extrapola limites das atividades puramente físicas ou biológicas. Sendo assim, o jogo não pode ser considerado meramente um fenômeno fisiológico ou um reflexo psicológico. O jogo está fora da vida habitual e é limitado pelo tempo e espaço em que se joga o jogo, até alcançar um certo fim. Enquanto está sendo jogado ocorre no jogo movimento, mudança, alternância, sucessão, associação, separação (HUIZINGA, 2007, p.3-4).

Já Zyda (2005, p.25) define um jogo como “uma competição física ou mental, jogado de acordo com regras específicas, com o objetivo de diversão ou de premiar o participante” do jogo.

### **2.1.1.2 Jogos digitais**

Schuytema (2008) declara que um jogo eletrônico (ou digital) é uma atividade que possui uma condição final, que pode ser obtida por um conjunto lúdico de ações e decisões. Estas ações e decisões são restringidas por um universo e um conjunto de regras ambos regidos por um software. As regras estabelecem o que é e o que não é permitido ser realizado e possuem a função de dificultar ou evitar que o jogador alcance os objetivos estabelecidos ao jogo. O universo criado e concebido para o jogo contextualiza e ambienta o jogo quando da execução de ações e tomada de decisão do jogador.

Battaiola (2000), descreve que enredo, interface interativa e motor são os componentes básicos de um jogo digital. Ele comenta que o enredo estabelece os objetivos do jogo a serem alcançados pelas ações do jogador. No enredo são definidos a sequência dos acontecimentos, o tema e a trama do jogo, sendo necessários, além da criatividade e pesquisa sobre o assunto, a interação entre especialistas do domínio, pedagogos e psicólogos para que estas definições ocorram. Battaiola indica ainda, que deve existir um sistema, denominado motor (do

inglês engine), que controla e dispara reações do jogo em função de ações que o jogador executa. Cada novo estado do jogo é criado e gerenciado pela interface interativa que controla e permite a comunicação entre o motor e o jogador. Já Zyda (2005, p.25), indica sua definição de jogo relatando que um jogo digital “é uma disputa mental, jogado com um computador de acordo com certas normas de diversões, recreação, ou de como ganhar o jogo.

### 2.1.1.3 Jogos Sérios

Os autores Djaouti, Alvarez e Jessel (2011, p.119) evidenciam que existem várias definições de jogos sérios e que “a primeira definição formal do conceito parece ter sido introduzido por Abt (1970)”. Em 1970, no seu livro *serious games*, Clark Abt utilizou o termo jogos sérios (*serious games*) pela primeira vez. Definia que os jogos sérios possuem uma finalidade educacional explícita e bem estruturada. Apesar de não se destinarem “prioritariamente para diversão, isso não significa que os jogos sérios, não são, ou não deveriam ser, divertidos”. (ABT, 1970, p.9). Abt indica ainda, que o termo jogos sérios é um oximoro, uma figura de linguagem, que combina duas palavras que “seriam aparentemente contraditórias mas que juntas em uma mesma expressão acabam assumindo um significado novo e interessante”. (ABT, 1970 apud MATTAR, 2010, p.20).

Apesar de não haver consenso sobre uma definição sobre o termo, muitos autores definem os jogos sérios como “um jogo desenvolvido para um fim específico que não apenas o entretenimento.” (RODRIGUES; MACHADO; VALENÇA, 2010, p.1). Cardenas (2014) ratifica esta definição e baseado nas definições contidas no artigo de Djaouti, Alvarez e Jessel (2011) indica que “um jogo sério é qualquer software que combine um propósito que não seja de entretenimento”. Além disso, o autor destaca que “um jogo sério, portanto, tem duas dimensões: a “séria” e a de “jogo”. Um jogo digital de entretenimento tem apenas a dimensão “jogo”, segundo essa concepção.” (CARDENAS, 2014, p.37).

Em adição Uliosack (2010, p.24), argumenta que desde a criação de jogos digitais, tem havido muita discussão tentando separar o conceito de jogos sérios de jogos de lazer. A autora faz posteriormente a seguinte pergunta sobre esta questão: “por que uma distinção é necessária?”. Como resposta a autora indica que o professor “Bartle argumenta que, através deste nome fica aceitável usá-los nos sistemas de educação formal e para ter financiamento para suas pesquisas”.



Outros autores corroboram com este conceito de jogos sérios com objetivo de ensino/aprendizagem ou treinamento e não apenas de entretenimento. Zyda (2005, p.26) conceitua um jogo sério como sendo "uma disputa mental, jogado com um computador de acordo com regras específicas, que usa entretenimento para promover treinamentos corporativos, educação, saúde, políticas públicas, [...]". Em seu livro os autores Michael e Chen, (2005) declaram que jogos sérios são uma recente classe de jogos cujo principal propósito não está no entretenimento e na diversão em si.

### **2.1.2 Classificação dos jogos sérios**

Antes de indicar uma classificação para os jogos sérios, será abordada a classificação acerca dos jogos digitais. Assim como a conceituação de jogos é diversa, a classificação de jogos digitais possui também algumas classificações distintas. Para Prensky (2007, p.130-131) existem 8(oito) tipos de jogos: ação, aventura, esportes e estratégia, luta, quebra-cabeças, RPG (do inglês Role-Playing Game ou jogo de interpretação de papéis) e simulações.

Bergeron (2006 apud MATTAR, 2010, p.79), apresenta a seguinte classificação para jogos digitais: ação, aventura, arcade, combate, tiro em primeira pessoa, tiro militar, multiusuário simultâneo, puzzle(quebra-cabeça), simulação em tempo real, RPG, tiro, simulação, sne aker (tipo de tiro), esportes, estratégia, tiro em terceira pessoa, trívia e multiusuário (um por vez).

A classificação de jogos sérios também possui algumas vertentes e diferentes abordagens. "Os primeiros sistemas de classificação baseavam-se em apenas um critério os quais podem ser organizados em duas vertentes: classificações baseadas em mercado e [...] em propósito" (SAWYER; SMITH, 2008 apud CARDENAS, 2014, p.39). Por exemplo, Zyda (2005, p.31) classifica os jogos sérios em 7(sete) tipos de mercados: saúde, políticas públicas, comunicação estratégica, engenharia de performance humana, treinamento e simulação, educação e avaliação de jogo. A figura 2 mostra esta classificação.

Figura 2 - Classificação dos jogos sérios



Fonte: Zyda (2005, p.30).

Em seu livro Michel e Chen (2006), classificam os jogos sérios nas seguintes categorias: jogos militares, jogos de governo, jogos educativos, jogos corporativos, jogos de cuidados com a saúde, jogos sobre política, religião e arte. A figura 3 mostra exemplos de jogos sérios e as suas respectivas categorias. Contudo, Cadernas (2014, p.40), indica que existem problemas nestes tipos de classificação por mercado, e informa que apesar de ser uma classificação útil, não existe informação sobre o conteúdo presente no jogo sério. Outra limitação destacada pelo autor, é sobre o dinamismo inerente aos mercados e o fato de sempre surgirem novos mercados para os jogos sérios.

Figura 3 - Uso dos jogos sérios na indústria



Fonte: Smith ( 2009, p.4).

Outros trabalhos classificam os jogos sérios baseados em seu propósito. Por exemplo, Bergeron (2006) categorizou os jogos sérios nos seguintes propósitos: jogos de ativismo, jogos de publicidade, jogos de negócios, exergames (de exercícios, sejam mentais ou físicos), jogos de saúde e medicina, newsgames (jornalismo de jogos digitais), jogos sobre política. Outros autores também fizeram suas classificações baseadas em propósitos, mas foi em 2011, que um sistema de classificação geral de jogos sérios foi proposta por Djaouti, Alvarez e Jessel (2011). Os autores introduziram uma nova classificação denominada modelo Gameplay/Purpose/Scope (G/P/S), que classifica os jogos sérios e combina os pontos fortes de vários sistemas de classificação anteriores. Esta classificação está presente e descrita na figura 4 a seguir.

Figura 4 - Classificação dos jogos sérios



Fonte: Djaouti, Alvarez e Jessel (2011, p.130).

No geral a classificação de Presky(2007) é amplamente difundida e aceita. Ele fez uma classificação baseada no propósito dos jogos digitais. A classificação de Bergeron (2006) envolve a criação de mais gêneros de jogos digitais. Ele classifica os jogos digitais mesclando a classificação em relação ao propósito e também acerca da atuação dos jogos digitais.

Da mesma forma a classificação dos jogos sérios foi realizada através de diferentes abordagens. Sawyer e Smith(2008) focaram na classificação dos jogos baseado em mercado alvo de atuação e propósito. Michel e Chen (2006) também efetuaram a classificação dos jogos sérios baseado nos mercados de atuação. Cardenas(2014) indicou que a classificação por mercados é estática e que novos mercados de jogos sérios podem surgir, inviabilizando a classificação. Além disso, Cardenas comenta que esta classificação por mercados não inclui a informação sobre o conteúdo presente no jogo sério.

A classificação que se mostra mais ampla e pode ser utilizada como uma classificação geral foi a proposta por Djaouti, Alvarez e Jessel (2011). Os autores combinaram pontos fortes e fracos de vários sistemas de classificação anteriores.

Eles categorizaram as classificações dos jogos sérios em gameplay(jogabilidade), purpose (proposta) e scope (escopo). A maneira como o jogo é jogado está relacionada com a jogabilidade(gameplay). Assim Djaouti, Alvarez e Jessel (2011), dividiram este gameplay em dois: game-based: Jogo com muitos objetivos e play-base: Jogo sem objetivos.

A proposta(purpose) indica qual é o objetivo ao jogar o jogo. É um treinamento mental? físico? São trocas de informações? Ou ainda, o jogo foi desenvolvido para transmitir mensagens educacionais? Já o escopo informa o mercado onde o jogo irá atuar, educacional, militar, dentre outros.

A próxima seção discorre acerca do contexto de utilização de jogos sérios para fins educacionais.

### **2.1.3 Jogos sérios no contexto educacional**

Quando jogos são desenvolvidos e utilizados para auxiliar o jogador no aprendizado de um assunto específico, na aquisição de novas habilidades ou ainda no reforço de seu desenvolvimento, eles são referidos como jogos educacionais. Os jogos sérios estão inseridos nesta categoria, (GEBREMICHAEL, 2016). Também chamados de edutainment os jogos educacionais “são aqueles que ensinam enquanto entretém”. (GATES, 2004, p.11). Entretenimento educativo (em inglês: edutainment) é uma maneira de tentar ensinar ou socializar através de elementos tais como filmes, jogos eletrônicos, música, sites, software multimídia, dentre outros. Esta forma de entretenimento objetiva educar divertindo o público alvo. (WIKIPEDIA, 2015).

Geralmente, estes jogos educacionais são destinados a um público muito mais jovem do que a maioria dos produtos comerciais. Os desenvolvedores dos jogos tipicamente trabalham em estreita colaboração com especialistas no assunto(especialistas do domínio), para garantir que o conteúdo é apropriado para o público-alvo. Os autores Pessini et al., (2014, p.537), comentam que, de forma semelhante a outras áreas, a educação se favorece da evolução tecnológica e assim, a utilização de jogos em atividades de ensino e aprendizagem pode ser algo promissor. De forma análoga, Pietruchinski, C. Neto e Reinehr (2011, p.2), indicam que quando da utilização como meio de favorecer o processo de ensino, os jogos podem se constituir como uma forma efetiva para que se consiga um ensino de qualidade.

Corroborando com este argumento Cheng e outros (2015, p.354), relatam ainda que existe, nos últimos anos, um crescimento das pesquisas e desenvolvimento de novas e inovadoras ferramentas com o objetivo de apoiar a aprendizagem. Neste cenário, os jogos sérios "têm sido utilizados de forma crescente, explorando atividades lúdicas que possuem objetivos pedagógicos específicos, com a finalidade de desenvolver o raciocínio e o aprendizado." (PESSINI et al., 2014, p. 537).

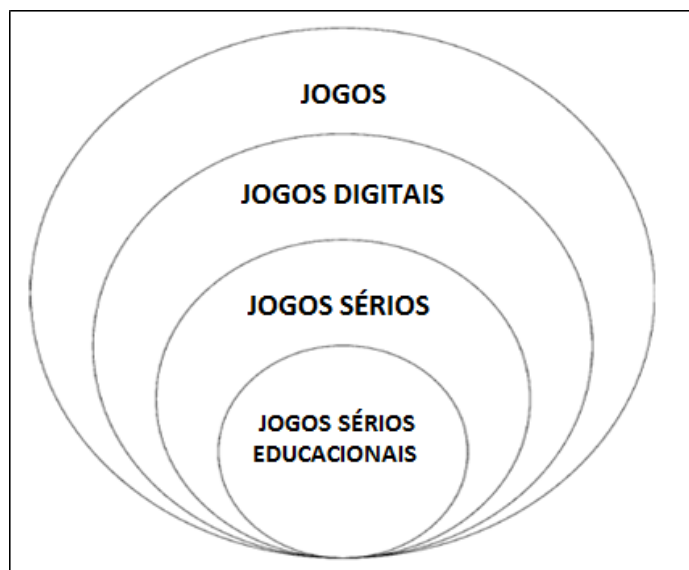
Essa nova tendência está sendo integrada no domínio da educação, e estes jogos tem como objetivo fazer com que os alunos participem ativamente da aprendizagem, e se sintam mais confortáveis por estarem familiarizados com o ambiente gráfico e cenários dos jogos. (MALLIARAKIS; SATRATZEMI; XINOGALOS, 2014, p.296).

Outro fator interessante é que para avançar no jogo, o aluno deverá necessariamente compreender completamente cada tópico do assunto abordado. Já na sala de aula tradicional, o aluno pode não entender algum conceito fornecido pelo professor, e ainda assim, avançar para o próximo assunto sem ter consolidado o tópico anterior (BNDES, 2014).

Apesar de existirem pesquisas e iniciativas na área de jogos, a autora Santaella (2012) ressalta que "atualmente, os jogos educacionais ainda são uma área pouco explorada no mercado pedagógico." A autora indica a dificuldade de criar um jogo educacional como um dos motivos para esta questão. Além disso, o grande público é atraído por jogos de entretenimento da grande indústria, que não são focados no aprendizado explícito do jogador. Adiciona-se a este cenário o fato "da maioria dos jogos educacionais apenas atraírem aqueles que já têm um interesse em determinada área e não o público em geral." (SANTAELLA, 2012, p.193).

Para situar a conexão entre os conceitos a figura 5 a seguir, exibe a relação entre jogos, jogos digitais, jogos sérios e jogos sérios educativos.

Figura 5 - Relações entre jogos, jogos digitais, jogos sérios educacionais



Fonte: Cheng et al. (2015, p. 355).

O cenário do mundo atual e tecnológico, cria a necessidade de muitos “indivíduos precisarem entender as tecnologias de computação em suas vidas, e saber construir tecnologias de computação que atendam às suas necessidades.” (MARGULIEUX; CATRAMBONE; GUZDIAL, 2016, p.44). Neste aspecto Margulieux, Catrambone e Guzdial (2016) apontam ainda, que esta necessidade de compreender a construção de tecnologias de computação, culminou no crescimento rápido “na demanda por ensino de computação o que deixou muitos sistemas de ensino lutando para se adaptar”.

A utilização de jogos de computador no processo de ensino pode oferecer vários benefícios. Embora existam outras abordagens, os jogos digitais aparecem como uma grande fonte de ideias inspiradoras e tornam-se populares com aceitação positiva (PARALIČ; PIETRIKOVÁ, 2014).

No entanto Battistella e Wangenheim (2016), alertam que “embora, os jogos pareçam aproximar-se bem das respectivas áreas de conhecimento, ainda existem lacunas a serem preenchidas, em especial, na área de conhecimento educacional”. Assim, estes autores argumentam que as pesquisas sobre jogos sérios, nesta área de ensino e aprendizagem, devem embasar e concentrar seu projeto em teorias educacionais e de jogos, e com isso, tentar “assegurar os resultados de aprendizagem pretendidos.” (BATTISTELLA; WANGENHEIM, 2016, p.1).

Em relação especificamente a jogos que ensinam, Santos e Souto (2014, p.1), após analisarem alguns trabalhos, concluem que mesmo com todos os

benefícios trazidos pelos jogos de ensino de programação, é preciso cautela na sua adoção. Já Navarro, Van Der Hoek (2007 apud SANTOS; SOUTO, 2014), afirmam que uma compreensão mais precisa a respeito dos resultados do uso desse tipo de recurso permite saber se compensam os custos e esforços envolvidos para adotá-los.

Ainda que os jogos digitais sejam os mais propostos e pesquisados para apoiar o ensino e aprendizagem em disciplinas de computação, é importante salientar que outras abordagens também são estudadas e utilizadas. O pesquisador Hakulinen (2011) relata que "jogos de computador não são a única maneira de usar jogos na educação". Ele indica em seu trabalho um estudo de caso em que o ensino das disciplinas de estruturas de dados e algoritmos, era feito através de um conjunto de cartas ou ainda praticado por um jogo realizado em um quadro negro (HAKULINEN, 2011, p.83).

Com relação ao ensino e aprendizagem de programação, Hakulinen (2011) chama atenção que no ensino de ciência da computação, os jogos podem ser aplicados de várias maneiras diferentes. Notadamente os cursos relacionados com a programação de computadores utilizando alguma linguagem de programação, proporcionam "um ambiente natural para jogos relacionados com tarefas de programação." Contudo Hakulinen (2011, p.2) indica que "as tarefas de programação não precisam ser a única maneira de utilização dos jogos".

Mattar (2010), destaca que nem sempre os games e seus elementos devem ser utilizados. Ele indica que os jogos devem ser utilizados se as pessoas estão desmotivadas em relação ao objeto de aprendizado. Caso contrário, havendo pessoas plenamente motivadas para aprender o conteúdo o mais breve possível, a utilização dos elementos pertencentes aos games é desnecessária (MATTAR, 2010, p.54).

Já Silveira (1999, p.102) relata que "de nada adianta um software educativo repleto de recursos gráficos se não existe um conteúdo que disponha de relevância pedagógica, ou seja, que seja realmente útil para os alunos".

Outra questão importante é indicada por Kashiwakura (2008, p.21) em que ele reflete que o desafio que surge juntamente com esta nova mídia para educação, é o de incentivar os educadores que modifiquem sua visão, que algumas vezes desqualifica os jogos digitais, a passarem a incentivar o uso de jogos em sala de aula. É neste aspecto que o jogo sério pode se inserir, pois "o jogador de um jogo



sério é motivado a jogar o jogo, e ao fazê-lo continua a lição muito mais tempo e com maior atenção do que ele ou ela fariam ao utilizar técnicas de aprendizagem tradicionais." (ADAMO-VILLANI; OANIA; COOPER, 2012-2013, p.110).

Outras considerações importantes são fornecidas por Clua e Bittencourt (2005), que indicam que uma vez compreendida a importância dos jogos digitais, com ou sem fins pedagógicos, no processo de ensino e aprendizagem, torna-se interessante incentivar o desenvolvimento de novos jogos digitais de qualidade, para serem utilizados em sala de aula e em outros ambientes de aprendizagem.

Os autores chamam atenção que estes novos jogos digitais devem ser mais atraentes e imersivos e desta forma, irão se diferenciar dos jogos didáticos padrões. Adicionalmente eles indicam que: "técnicas de desenvolvimento de games 3D podem ser bem utilizadas para criar jogos que sejam lúdicos, divertidos, atrativos e que possam atender [...] uma série de objetivos pedagógicos." (CLUA; BITTENCOURT, 2005, p. 45).

Embora a área de educação se beneficie da evolução tecnológica, os autores citados anteriormente deixam nítido que é necessária uma evolução nas pesquisas sobre desenvolvimento deste tipo de jogo. A utilização de teorias educacionais formais poderia ajudar neste processo. Em adição, a adoção de uma metodologia consolidada de desenvolvimento específica para jogos sérios, também poderia melhorar a qualidade dos jogos desenvolvidos.

Desta forma, apesar das pesquisas na área de jogos sérios terem evoluído, ainda se faz necessário aprofundar estas pesquisas, na busca de soluções que se tornem referência para esta importante área, tanto do ponto de vista educacional quanto da área de engenharia de software. Outras áreas afins como a psicologia, sociologia, linguística, semiótica, artes, design, IHC (interação homem-computador), ergonomia, ciência da computação, também devem continuar a fornecer contribuições para a área de jogos sérios.

Uma explanação sobre técnicas de desenvolvimento de jogos sérios será realizada no tópico a seguir.

## 2.2 DESENVOLVIMENTO DE JOGOS SÉRIOS

Atualmente é necessário gerenciar projetos de desenvolvimento de jogos que seguem padrões diversificados e podem ser de longa duração. Os autores Rodrigues, Machado e Valença (2010), comentam que as equipes envolvidas neste

processo são heterogêneas e que o conteúdo e aspectos pedagógicos envolvidos são complexos. Os autores abordam ainda que existem alguns modelos de desenvolvimento de software (Game Waterfall Process - GWP, Extreme Game Development – XGD e Game Unified Process – GUP) que foram criados em anos anteriores e que foram moldados especificamente para o desenvolvimento de jogos. Contudo, estes modelos são no geral superficiais e não adaptáveis aos objetivos de jogos com cunho pedagógico (RODRIGUES; MACHADO; VALENÇA, 2010, p.1-2).

Para um melhor entendimento sobre os aspectos que envolvem o desenvolvimento dos jogos sérios, faz-se necessário analisar quais são os conceitos-chaves envolvidos neste processo, os papéis, principais artefatos gerados e as ferramentas de apoio à produção destes jogos. As próximas seções irão discorrer sobre todos estes itens.

### **2.2.1 Conceitos envolvidos**

Nesta seção define-se conceitos referentes ao desenvolvimento de jogos considerados mais relevantes para esta dissertação.

#### **2.2.1.1 Game Design**

Game design é um termo utilizado com frequência em livros, sites e pesquisas relacionadas com o desenvolvimento de jogos digitais. O termo design pode ser traduzido como projeto ou desenho, contudo é possível observar a utilização deste termo mesclando termos em inglês e português: design de jogos.

Outra utilização para o termo refere-se ao profissional responsável pela concepção, criação e coordenação de todo o jogo, e neste caso a palavra ganha o sufixo “er” tornando-se game designer. O game designer também é usualmente chamado de designer do jogo. O papel desempenhado por este profissional é discutido na seção 2.2.2.2

Mastrocola (2012) inclusive, alerta que muitas pessoas não possuem uma compreensão da essência do significado da palavra design, e desta forma, é muito frequente elas simplesmente relacionarem o processo de game design apenas como a etapa de conceber a forma gráfica do jogo, o leiaute, o cenário, a trilha sonora e a forma como os elementos 3D são exibidos na tela, etc. O autor destaca que na realidade “o grande desafio do game designer é criar as mecânicas, as regras e objetivos de um jogo”. (MASTROCOLA, 2012, p.34-35).

Alguns autores como Clua e Bittencourt (2005), estabelecem que game design está relacionado à conceituação artística do jogo, abrangendo “quais as principais características dos cenários, esboços de personagens, descrição das texturas fundamentais, mapas e descrições das fases (também denominado de level design)”.

Já para Crawford (1982, p.45), o game design se constitui em um processo artístico e também é um processo técnico, já que o game designer muitas vezes está trabalhando arduamente em montanhas de código, orientado a grandes objetivos artísticos.

O design do jogo é efetuado por um papel denominado designer do Jogo. Gates (2004), indica que mesmo que o design do jogo seja bem recente, os princípios de bom design de jogos (ou game design), já devem estar bem estabelecidos para tentar garantir que o jogo seja excelente. É importante ter a capacidade de avaliar cada item do jogo e antecipar qual será a reação do jogador para cada elemento. Como esta capacidade de previsão não é exata, a função dos testadores de jogos será crucial para prover informações sobre itens do jogo em que não houve resposta satisfatória na sua execução (GATES, 2004, p.17).

Observa-se que Crawford(1982) orienta o trabalho do game designer para a codificação do jogo. Já Gates(2004) cita outros papéis relacionados com o desenvolvimento de jogos. Embora um game designer possa desempenhar o papel de programador de códigos de um jogo, atualmente existe uma maior divisão de papéis a serem efetuados no processo de desenvolvimento de um jogo digital.

Para melhorar a compreensão sobre estes papéis, a seguir são listadas na tabela 1, algumas destas novas divisões de papéis desempenhados por profissionais quando do desenvolvimento de jogo digital. É importante salientar que em projetos menores um profissional frequentemente executa vários papéis. Já em projetos de grandes jogos os profissionais geralmente executam apenas um único papel.

Tabela 2 - Papéis no desenvolvimento de um jogo digital

<b>PAPEL</b>	<b>DESCRIÇÃO</b>
<b>Animador de Jogos</b>	Animadores de jogos são responsáveis pela interpretação de movimentos e comportamentos dentro de um jogo.
<b>Artista de jogos</b>	Artistas de jogos criam os elementos visuais de um jogo, como personagens, cenário, objetos, veículos, texturas de superfície e de vestuário.
<b>Artista Principal</b>	O artista principal é responsável pela aparência geral de um jogo.
<b>Artista Técnico</b>	Artista técnico age como uma ponte entre os artistas e programadores durante o desenvolvimento de um jogo.
<b>Assistente de produção</b>	Assistente de produção trabalha com a equipe de produção de um jogo para garantir que a entrega do produto ocorra no prazo estabelecido.
<b>Engenheiro de áudio</b>	Engenheiros de áudio criam a trilha sonora para um jogo, incluindo música, efeitos sonoros, vozes dos personagens e instruções de voz.
<b>Designer de jogos</b>	Designers de jogos definem do que um jogo consiste e como ele é jogado. Define também todos os elementos centrais do jogo.
<b>Diretor de Criação</b>	O diretor de criação de jogos é responsável pela aparência geral de um jogo digital.
<b>Editor de Nível (ou Level designer)</b>	O editor de nível define e cria a arquitetura interativa para um nível (fase) de um jogo, incluindo a paisagem, edifícios e objetos.
<b>Gerente de produto</b>	O gerente de produto ajuda a criar e desenvolve campanhas de marketing para maximizar as vendas de jogos.
<b>Gerente de projetos</b>	O gerente de projetos garante a entrega bem-sucedida do desenvolvimento de um jogo.
<b>Produtor externo</b>	O produtor externo trabalha para garantir a entrega bem-sucedida de um jogo e trabalha externamente a equipe de desenvolvimento do jogo.
<b>Programador de IA(inteligência artificial)</b>	Responsável por programar os algoritmos de inteligência artificial usados no jogo. Dentre estes algoritmos estão o de planejamento estratégico de grupo num atirador de primeira pessoa, a montagem da estratégia do time controlado pela CPU num jogo de esportes, o planejamento de caminho para levar um personagem de um local ao outro, etc.
<b>Programador de jogos</b>	O programador escreve o código da programação que executa e controla um jogo.
<b>Programador de redes</b>	Responsável pela criação de uma infraestrutura on-line para um jogo, e garantia da estabilidade e segurança dos serviços da web. O programador de redes é o responsável por fazer os códigos que possibilitam o jogo ser jogado por mais de uma pessoa simultaneamente através de uma rede. Esta rede pode ser tanto local, como uma rede LAN, quanto global, como a internet.
<b>Programador Líder</b>	O programador líder lidera a equipe de programação responsável pela criação de todo o código de programação do jogo.
<b>Redator</b>	Projeta a estrutura narrativa de um jogo e escreve o diálogo, história e descrições de itens e tutoriais.
<b>Roteirista</b>	Descreve como é a trama do jogo, o perfil psicológico dos personagens e a interação entre eles, sendo a interação do tipo social e/ou psicológica, a qual não necessariamente corresponde àquela entre o jogador e os personagens do jogo.
<b>Testadores</b>	Testam e depuram um jogo, sugerindo melhorias que garantem sua qualidade e capacidade de reprodução, garantindo qualidade em um jogo e encontram falhas antes que se tornem públicas.

Fonte: Creativeskillset ( 2016).

### **2.2.1.2 Jogabilidade**

Um dos conceitos importantes acerca dos jogos digitais chama-se *gameplay* (ou jogabilidade). De acordo com Vannucci e Prado (2009, p.138), *gameplay* se relaciona e “emerge das interações do jogador com o ambiente, a partir da manipulação das regras e mecânicas do jogo, pela criação de estratégias e táticas que tornam interessante e divertida a experiência de jogar”. Outra definição é fornecida por Schuyttema (2008, p.7-8), em que ele relata que *gameplay* está relacionado com a experiência, o percurso no qual o usuário aprende habilidades básicas, até atingir a vitória ou derrota, do início ao fim do jogo. Esta experiência pode ser lúdica, instigante e emocionante.

Como já foi relatado neste trabalho, um jogo sério precisa ser motivador e interessante, para que possa atingir seus objetivos e garantir a imersão apropriada do jogador ao ambiente alvo do jogo. Neste sentido fazer com que os jogos sérios possuam boa jogabilidade torna-se um elemento fundamental no alcance destes importantes objetivos.

### **2.2.1.3 Prototipagem**

Com o objetivo de tentar garantir que o jogo seja interessante e divertido, são criados modelos de trabalho que sirvam para testá-lo, e também verificar se a mecânica do jogo está impecável. Estes modelos podem ser de baixa fidelidade (rascunhos) e feitos em papel, ou podem ser algum item funcional de software que mostre na tela a essência que torna o jogo especial.(NOVAK, 2010, p. 342).

Fullerton, Swain e Hoffman (2008), acrescentam que "prototipagem é a criação de um modelo de trabalho, de sua ideia, que lhe permite testar a sua viabilidade e fazer melhorias a ele". O protótipo se parece com desenhos e às vezes já é “jogável”. O objetivo do protótipo é verificar como está o funcionamento de um determinado conjunto de características e mecânicas do jogo.

Alguns profissionais iniciantes descartam a utilização da prototipagem e seguem diretamente para a construção do jogo. Contudo, o design de um jogo eficaz tem como ponto central a prototipagem. Para aprimorar a jogabilidade de seu jogo e descobrir se seu design é sólido, é necessário investir tempo em um processo de prototipagem eficiente. Neste processo não existirão preocupações em já possuir uma interface e ambiente bem elaborados para o jogo. Os itens a serem analisados

no protótipo devem se concentrar nos mecanismos do jogo e se o jogo prende a atenção dos jogadores testadores (FULLERTON; SWAIN; HOFFMAN, 2008, p.175).

Novak (2012, p.354) afirma que protótipos devem servir de ferramenta para tornar o jogo bem-sucedido. Ela considera o protótipo como sendo “um pedaço de trabalho de software, que captura na tela a essência do que faz o seu jogo especial, o que o diferencia do resto, e o que vai torná-lo bem-sucedido”.

A autora esclarece que anteriormente a construção de um protótipo digital é benéfico criar um protótipo utilizando cartões, placas, blocos, ou miniaturas de papel. Este protótipo em baixa fidelidade deverá ser testado para verificar se o jogo está divertido e atraente, e se a mecânica do jogo está adequada. A utilização de protótipos em baixa fidelidade antes dos protótipos digitais evita que os designers percam o foco e se distraiam sobre a jogabilidade do jogo e seus fundamentos. Como exemplo destas distrações a autora destaca as preocupações que existem com o estilo visual que podem ocorrer quando da avaliação de protótipos já no formato digital (NOVAK, 2012, p.354).

#### **2.2.1.4 Conceito do Jogo (game concept)**

Uma das primeiras coisas que os designers de jogos fazem é pensar que tipo de jogo eles estão concebendo. Para desenvolver um jogo, os designers precisam chegar ao conceito do jogo (game concept). Este conceito é uma ideia detalhada de como o jogo é. O conceito descreve os elementos que fazem parte do jogo e como estes elementos se relacionam entre si. É necessário definir estes itens e suas relações, para criar a experiência que será vivenciada pelo jogador ao jogar o jogo (HARTEVELD, 2011, p.71).

Gates (2004) chama atenção que a concepção do conceito do jogo não é uma tarefa simples. É importante previamente conhecer o público-alvo e os objetivos delineados para o jogo. “Antes de começar o trabalho de design em um jogo, deve-se ter uma ideia geral do jogo, sobre o que o jogo é, em que gênero ele se encaixa, e quais os objetivos.” (GATES, 2004, p.3). Assim Gates mostra e ratifica a importância desta atividade de criar o conceito do jogo. Outros autores também chamam atenção que criar o conceito do jogo, não é uma atividade simples de ser executada. Fullerton, Swain e Hoffman (2008, p.148), relatam que para muitas pessoas ter ideias é difícil e ainda mais complicado é conseguir ter excelentes

ideias. Ter a ideia é só o início, pois o processo é iterativo e irá auxiliar na evolução e no aprimoramento do conceito original do jogo (game concept).

Existe, no entanto, uma pergunta que Crawford (1982, p.46) nos apresenta: Como você seleciona um objetivo correto? Segundo ele, não existe uma resposta objetiva para esta questão, pois existe muita subjetividade no conceito e na arte do design de jogos digitais. Selecionar um bom tema para o jogo pode demorar. É importante analisar cada tópico conceitual, para avaliar a sua capacidade de realizar com sucesso os objetivos do jogo. Vários tópicos possuem um excesso de bagagem emocional que podem interferir nos objetivos do jogo.

### **2.2.1.5 Interface**

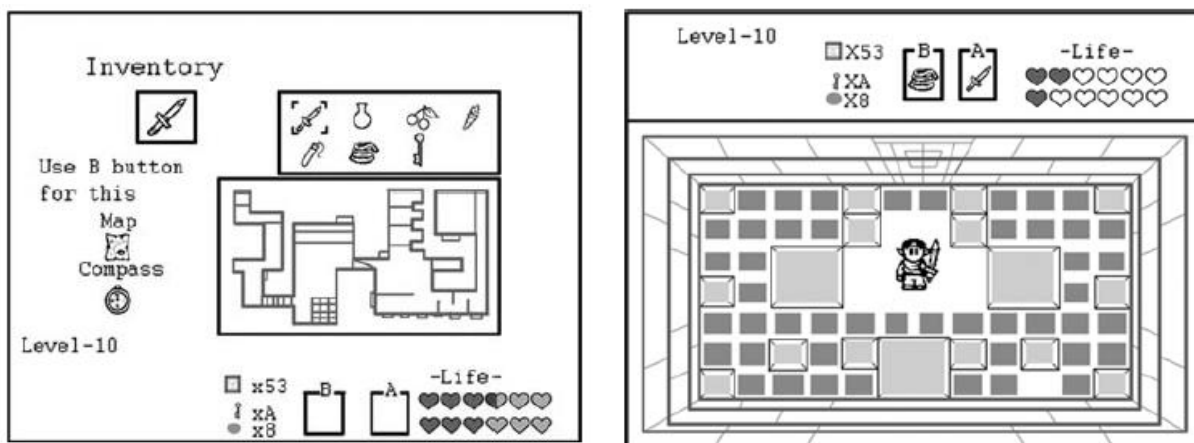
Interface pode ser descrita como um local em que o contato entre duas entidades ocorre. Este conceito dá suporte a ideia de que uma interface é um contexto compartilhado de ação no qual tanto o computador como o humano são agentes. Um exemplo é a tela do computador. Apesar de ser um termo frequentemente atribuído a ícones, menus, barras de rolagem ou talvez, linhas de comando e cursores piscando, a interface não é só isso. Existem diversos outros exemplos de interfaces: a maçaneta de uma porta, uma torneira, a direção de um carro, etc. Assim as qualidades físicas das partes presentes nesta interação são refletidas pela interface. As atuais interfaces gráficas explicitamente representam o que vem a ser esse meio de significados comum, pela aparência e comportamento dos objetos na tela (ROCHA;BARANAUSKAS, 2003, p.8-9)

Um exemplo clássico é a conhecida área de trabalho do computador(desktop). Este recurso “permite a interação por meio de tela vertical, visto como um espaço, onde estão [...] entidades passivas, que reagem com a vontade do usuário. O desktop serviu para introduzir uma primeira noção de interface gráfica aos usuários” (LOBO, p.51). Com relação aos jogos digitais, a interatividade entre o jogador e o jogo é proporcionada pela interface. É a interface que permite ao jogador alcançar seus objetivos, sem frustrações, através de escolhas realizadas durante o jogo (NOVAK, 2011, p. 235).

Desta forma a interface é um elemento muito importante no projeto de desenvolvimento de um jogo. Apesar desta importância Gates (2004), alerta que a interface não deve fornecer todas as dicas referente a como avançar no jogo. Contudo, ela deve apresentar as informações de forma rápida e instantânea. Assim

para determinados jogos é criado um HUD (heads-up display) que exibe na tela informações de ação do jogador. Em outros jogos pode ser melhor a informação não estar o tempo todo visível ou não aparecer o tempo inteiro. Podem adicionalmente serem exibidos os botões de controle e status ao redor de uma área ativa da tela. É importante que seja fácil para o jogador executar um controle, manipular o teclado ou mouse e que estas ações sejam otimizadas em relação a quantidade de cliques, teclas e apertos de botões (GATES, 2004, p.26).

Figura 6 - Exemplo de tela e sub-tela de um rascunho de uma interface de um jogo



Algumas áreas de conhecimento contribuem para a obtenção de uma boa interface. A semiótica por exemplo. “De um modo geral, semiótica é o estudo de como os significados são criados. A questão do que os sinais representam, ou denotam, é uma preocupação central para esta área.” (SALEN; ZIMMERMAN, p.4).

Outro conceito relacionado com a obtenção de uma boa interface é o da usabilidade. A usabilidade é conhecida “como a capacidade do software de ser facilmente usado, compreendido, aprendido e apreciado, quando usados nas condições especificadas.” (LOBO, 2013, p.62).

Lobo (2013) nos fornece ainda, a definição de uma outra área importante e relacionada com a interface do jogo. Segundo ele, a interface homem-computador (IHC), “compreende em uma disciplina preocupada com o design, avaliação e implementação de sistemas computacionais interativos para o uso humano, considerando o estudo dos principais fenômenos que o cercam”. Ele discorre e evidencia que assim, compreender estes aspectos e conceitos que possuem relacionamentos não somente com a área de jogos, é importante para responder as



“questões relacionadas à compreensão do usuário e suas capacidades mentais”. (Lobo, 2013, p.41).

Para que a interface possa ser avaliada outro conceito conhecido como **heurística** é utilizado. Com o objetivo de identificar erros de design que possam comprometer a usabilidade, alguns especialistas de design avaliam a interface, de forma não concomitante, "confrontando-as com regras conhecidas como heurísticas." (MACHADO NETO, 2013, p.18, grifo nosso).

Já Novak (2012) observa que em relação a interface de um jogo, a sua funcionalidade é mais importante que a sua estética, e ela deve fornecer excelente usabilidade. Assim a usabilidade deve ser pensada mais como engenharia do que como arte. Seria contraproducente produzir uma interface esteticamente atraente, mas que não seja funcional para jogadores e desenvolvedores. Permitir que o jogador jogue o jogo é a função mais importante de uma interface. Ser esteticamente atraente é desejável, porém se a interface não atingir este objetivo ela será inútil.

Esta interface deve ter gráficos nítidos e evitar uma estética com excessos que podem ocupar toda a tela e obstruir o jogador. Deve conter controles intuitivos sem obrigar ao jogador a ter que interagir sempre muitas vezes para chegar ao seu objetivo. As opções fornecidas ao jogador devem ser em pequeno número sem ser, no entanto, simplista (NOVAK, 2012, p.259)

### **2.2.2 Principais papéis envolvidos**

Anteriormente a definição dos papéis envolvidos no desenvolvimento de um jogo sério, serão descritos os conceitos relacionados com papéis oriundos do processo unificado(RUP). A metodologia concebida no capítulo 3 utiliza estas nomenclaturas e conceitos do RUP. A tabela 2 lista e define estes conceitos.

Tabela 3 - Exemplo de tela e sub-tela de um rascunho da interface de um jogo.

<b>DISCIPLINA</b>
Uma disciplina é um conjunto de atividades relacionadas a uma área de interesse importante em todo o projeto. O principal objetivo do agrupamento de atividades em disciplinas é ajudar a compreender o projeto a partir de uma perspectiva em cascata tradicional. Por exemplo, geralmente é mais comum executar determinadas atividades de requisitos em coordenação direta com as atividades de análise e de design. A separação dessas atividades em disciplinas distintas facilita a compreensão.
<b>ATIVIDADE</b>
Uma atividade é uma unidade de trabalho desempenhada por um papel inseridas no contexto de uma disciplina. Uma atividade é algo que um papel faz e produz um resultado significativo no contexto do projeto. Uma atividade é uma unidade de trabalho que um indivíduo, desempenhando o papel descrito, pode ser chamado a realizar. A atividade tem uma finalidade clara, normalmente expressa em termos da criação ou atualização de alguns artefatos como um modelo, uma classe, um plano. Toda atividade é atribuída a um papel específico. As atividades estão fortemente relacionadas aos artefatos. Os artefatos fornecem a entrada e a saída para as atividades e o mecanismo pelo qual as informações são transmitidas entre as atividades.
<b>TAREFA</b>
Uma tarefa é uma ação desempenhada por alguma pessoa visando a realização ou monitoramento do projeto. Não representa uma evidência de progresso no desenvolvimento, e consome recursos: esforço (tempo de pessoa), equipamento, financeiro.
<b>ARTEFATO</b>
Um artefato é um resultado de uma atividade. As atividades possuem artefatos de entrada e saída. Os artefatos são utilizados como entradas e/ou saídas na execução das atividades. Os artefatos podem ser: modelos, documentos, código fonte, executáveis, etc. Os papéis usam os artefatos de entrada para executar atividades e produzem artefatos de saída ao executarem as atividades. Os artefatos são responsabilidade de um único papel e promovem a ideia de que todas as informações no processo devem ser responsabilidade de uma pessoa específica. Embora um artefato "pertença" a uma pessoa, muitas outras podem utilizá-lo e, talvez, até atualizá-lo se tiverem permissão.
<b>PAPEL</b>
Um papel é uma definição abstrata de um conjunto de atividades executadas e dos respectivos artefatos. Normalmente os papéis são desempenhados por uma pessoa ou um grupo de pessoas que trabalham juntas em equipe. Um membro da equipe do projeto geralmente desempenha muitos papéis distintos. Os papéis não são pessoas; pelo contrário, eles descrevem como as pessoas se comportam no negócio e quais são as responsabilidades que elas têm. Os papéis têm um conjunto de atividades coerentes por eles executadas. Essas atividades são estreitamente relacionadas e combinadas em termos de funcionalidade, e é recomendável que elas sejam executadas pela mesma pessoa.

Fonte: Wthreex ( 2015)

### 2.2.2.1 Especialista de Domínio

Uma pessoa que possua um conhecimento especial ou habilidades em determinada área de atuação é denominada especialista de domínio. Um contador, por exemplo, é um especialista no domínio da contabilidade. Caso seja desenvolvido um software contábil será exigido o conhecimento dos domínios da contabilidade e de software (CAMPBELL, 2013).

### **2.2.2.2 Designer de jogos**

O designer de jogos (do inglês game designer), é o profissional responsável por criar a mecânica do jogo, de fornecer ao jogador do jogo digital, a experiência de ultrapassar obstáculos e vencer ameaças presentes no jogo. Estes obstáculos e ameaças devem estar em consonância com a história e cenário. Os designers devem garantir a existência destes elementos desde o início do jogo. O designer de jogos é quem provavelmente irá analisar as novas ideias para decidir se elas irão ajudar ou atrapalhar o desenvolvimento. O importante documento de design é criado por este profissional que deve atualizá-lo durante todo o desenvolvimento do jogo. (GATES, 2004, p.97; p.159-160).

Adicionalmente Salen e Zimmermann (2004) indicam que o designer de jogos tem como foco trabalhar no projeto, na concepção e na elaboração de regras e estruturas que darão ao jogador uma experiência quando jogar o jogo desenvolvido. Assim como um designer industrial, um designer gráfico, o designer de jogos é um tipo particular de designer. Contudo este profissional não é necessariamente um programador, designer visual ou gerente de projeto. Apesar disso, em alguns casos ele pode atuar nestas funções na criação de um jogo. Um designer de jogos pode projetar não apenas jogos digitais. Ele pode criar jogos de cartas, jogos sociais ou qualquer outro tipo de jogo.

Crawford (1982) indica e converge com o que foi relatado anteriormente, e mais uma vez destaca que o jogador deve ser a principal preocupação do profissional de designer de jogos. A sua finalidade é dar entretenimento, educar o jogador. Para se obter um bom design de jogo é fundamental responder as seguintes perguntas: por que as pessoas jogam jogos? O que os motiva? O que ocasiona o divertimento de jogos? (CRAWFORD, 1982, p.14).

### **2.2.2.3 Testadores de software**

Um Jogo digital é um software e sendo um software ele necessita ser testado para atingir sua confiabilidade. O software deve ser testado para se tornar suficientemente bom e adequado para seu uso. Os clientes e desenvolvedores devem ser convencidos após o processo de testes, que o software atingiu sua confiabilidade. Esta é a principal meta do processo de testes (SOMMERVILLE, 2007, p.356).

Canteri (2014) indica que "testes de software são essenciais nessa área de jogos, pois, além de testar falhas, deve-se testar também se o jogo não ficou desequilibrado ou maçante para o jogador." Na fase alfa de testes, funcionários da empresa desenvolvedora do jogo (equipe de desenvolvimento), realizam testes utilizando casos de testes criados pela equipe de testadores e analistas de sistemas. Já na fase beta de testes, o jogo, ainda incompleto, é disponibilizado para alguns jogadores(beta-testers) do público-alvo do jogo, e que não pertencem a equipe de desenvolvimento. O objetivo é que estes "beta-testers" testem o jogo fora da empresa e indiquem possíveis defeitos (CANTERI (2014, p.12).

Assim, uma função crucial no ciclo de desenvolvimento de um jogo e de qualquer software é a do testador. Gates (2004) relata, inclusive, que são os testadores que de certa forma determinam quando o jogo será realmente lançado. Para ele, testar não é simplesmente efetuar uma rápida verificação do jogo antes dele ir para o mercado. Quando o projeto atingir a fase alfa (a fase em que o jogo é mais ou menos jogável do início ao fim), deve-se elaborar o plano de teste juntamente com a equipe de desenvolvimento. Na fase beta, o desenvolvimento será parado e apenas correções de defeitos serão realizadas. O objetivo será retirar o maior número de erros possíveis antes do lançamento do jogo e torná-lo estável (GATES, 2004, p.176-177).

#### **2.2.2.4 Programador**

O papel do programador codifica em alguma linguagem de programação ou script visual o design de um jogo. Ele torna real e aplicável digitalmente o que um designer imaginou para o projeto. Este profissional trabalha em conjunto com os designers de jogos que, muitas vezes, não entendem as implicações técnicas do que eles criam. Caberá ao programador orientar e guiar a construção do jogo em relação a camada de código. É ele quem irá apontar onde devem ser realizadas intervenções que podem gerar melhorias internas no código, que resultarão em redução da complexidade, maior eficiência, aumento da velocidade de visualização de elementos no jogo, e um menor número de defeitos (GATES, 2004, p.165).

## **2.2.3 Principais artefatos gerados**

### **2.2.3.1 Documento de requisitos**

O autor Sommerville (2007, p.63), explica que um documento de requisitos de software, “é uma declaração oficial do que os desenvolvedores do sistema devem implementar”. Este documento é também denominado: Especificação de Requisitos de Software (do inglês: Software Requirements Specification). O processo de engenharia de requisitos produz este documento, que deve satisfazer as partes interessadas no projeto do software. As partes interessadas também são referenciadas pelo termo stakeholders.

Sommerville (2007) destaca ainda, que este documento tem dois níveis de abstração: um para os usuários finais e clientes e outro para os desenvolvedores do sistema. No primeiro caso os clientes e usuários finais precisam apenas de uma declaração de requisitos em alto nível. Em contraste os desenvolvedores necessitam de uma especificação mais detalhada de requisitos. O documento de requisitos é parte integrante e resultado da engenharia de requisitos. É um processo crítico, pois, se forem cometidos erros nesta fase, estes erros irão causar efeitos negativos também na fase de projeto e implementação do sistema (SOMMERVILLE, 2007, p.23-24).

### **2.2.3.2 Documento de design**

Começar a codificar um jogo sem antes possuir um documento de design é um erro que muitas equipes de desenvolvedores de jogos comete. A pressão para programar o jogo desde o início do projeto contribui para isso. Assim é importante que o projeto possua um documento de design. Este documento contém as características que o jogo deve possuir, descreve as interfaces do usuário, define os requisitos de arte, etc. O documento de design do jogo é semelhante a um documento de análise de requisitos que é utilizado em um desenvolvimento padrão de software (GATES, 2004, p.219).

Com o objetivo de ajudar na comunicação de ideias de design aos programadores, artistas e toda a equipe envolvida no desenvolvimento do jogo, é criado um documento balizador de todo o processo de desenvolvimento. O documento de design do jogo contém, em detalhes todas as interfaces e mecânicas do jogo que está sendo desenvolvido (SCHELL, 2008, p.384).

A autora Novak (2012), também apresenta a ideia do documento de design ser um documento guia de referência de desenvolvimento de todo jogo, desde o seu início até o fim de sua criação. Este documento foca na jogabilidade, enredo, personagens, regras e interface. Estão detalhadas as regras do jogo, sendo até possível simular o jogo sem mesmo ter alguma versão ou protótipo digital (NOVAK 2012, p.391).

O autor Schuytema (2008) nos propõe a seguinte definição e indica que “o documento de design do jogo é o coração e a alma de todos os documentos que giram em torno de um jogo em desenvolvimento”. Ele faz uma analogia deste documento relatando que é como se o documento de design fosse o documento de planta baixa do jogo. O objetivo de um documento de design em relação a um jogo, “é ilustrar como se deve jogá-lo e apresentar uma descrição abrangente de todos os aspectos, para que a equipe de desenvolvimento possa de fato, criar o game.” (SCHUYTEMA, 2008, p.100).

Mattar (2010), também relata que o documento de design é um dos mais importantes documentos gerados durante o desenvolvimento de um jogo. Ele deve ser elaborado na fase de pré-produção do jogo e pode conter diversos outros documentos dentro dele. Este documento servirá de referência e será consultado com frequência pelos membros da equipe de desenvolvimento. O documento de design deve conter todos os detalhes de implementação do jogo, desde o seu conceito, além das especificações técnicas e funcionais. Adicionalmente também devem estar presentes neste documento informações sobre a codificação, a arte, a logística de produção, a lista de tarefas, as características do jogo, os personagens, a sua interface, gráficos, animações, vídeos, áudios, músicas, enredo, descrição das fases e demais elementos específicos ao jogo que está sendo desenvolvido (MATTAR, 2010, p.90)

Adicionalmente Schell (2008), chama atenção que, apesar de ser um documento fundamental, não existe um formato padrão para criar um documento de design. Existe, porém, um consenso que este documento necessita possuir todas as informações necessárias para construir o jogo (SCHELL, 2008, p.384).

Uma sugestão de tópicos é fornecida por Schuytema (2008, p.101). Esta sugestão de sumário de um documento de design do game é indicada a seguir:

Tabela 4 - Tópico de um documento de design

<ul style="list-style-type: none"> <li>I. Visão geral essencial <ul style="list-style-type: none"> <li>a. Resumo</li> <li>b. Aspectos fundamentais</li> <li>c. Golden nuggets</li> </ul> </li> <li>II. Contexto do game <ul style="list-style-type: none"> <li>a. A história do game</li> <li>b. Eventos Anteriores</li> <li>c. Principais Jogadores</li> </ul> </li> <li>III. Objetos essenciais do game <ul style="list-style-type: none"> <li>a. Personagens</li> <li>b. Armas</li> <li>c. Estruturas</li> <li>d. Objetos</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>IV. Conflitos e soluções</li> <li>V. Inteligência artificial</li> <li>VI Fluxo do game</li> <li>VII Controles</li> <li>VIII Variações do jogo</li> <li>IX Definições</li> <li>X Referências</li> </ul>
---	---

Fonte: Schuytema (2008, p.101).

Contudo Schell (2008) alerta que não existem modelos prontos e mágicos. Os documentos são parte importante do desenvolvimento do jogo, mas existem documentos diferentes para cada jogo e equipe. Anteriormente a criação dos documentos do projeto do jogo, é necessário entender qual o propósito deste jogo a ser desenvolvido. (SCHELL, 2008, p.382).

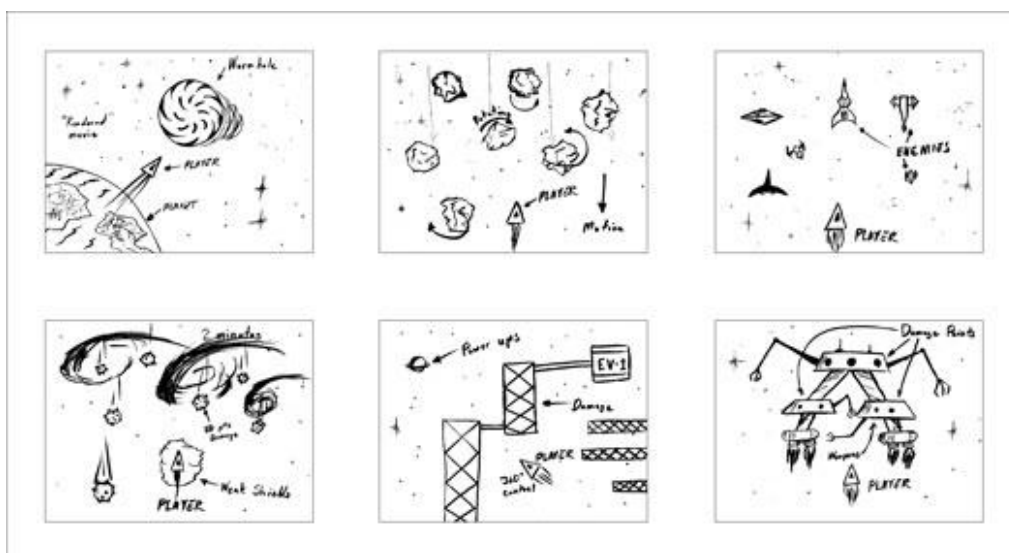
Fullerton, Swain e Hoffman (2008), também apresentam a importância do documento de design. Diante de um projeto cada vez mais complexo, com equipes grandes, cronogramas e orçamentos apertados, tornou-se imprescindível possuir uma documentação clara e abrangente. Os autores consideram importante existir um documento norteador para balizar os trabalhos que envolvem o desenvolvimento dos jogos sérios. O documento de design contém os objetivos do jogo que será desenvolvido e um plano detalhado de como atingir estes objetivos. Este documento é gerenciado pelo designer de jogos e contribui para uma comunicação mais eficaz, diante de equipes e projetos cada vez maiores (FULLERTON; SWAIN; HOFFMAN, 2008, p.395)

Desta forma o documento de design é um documento “vivo”, em constante atualização, que pode estar em uma wiki ou documento de arquivo e que contém toda a visão do design do jogo (BRATHWAITE; SCHREIBER, 2009, p.14).

### 2.2.3.3 Storyboards

Uma definição de storyboard é fornecida por Lobo (2013): “um documento utilizado [...] para organizar uma série de gráficos, com o propósito de pré-visualizar um filme, uma animação ou elemento interativo, de forma a servir de suporte para seu desenvolvimento.” (LOBO, 2013, p.20). Fullerton, Swain e Hoffman (2008, p.216), indicam que os storyboards são bastante utilizados para mostrar como serão filmadas, as cenas pertencentes à produção de um filme. Eles são desenhos que exibem um esboço de uma sequencia visual para determinar como estas cenas serão filmadas. Da mesma forma, os storyboards são válidos e de boa utilidade para mapear o potencial de jogabilidade dentro de um nível(level) de um jogo. As figuras 7 e 8 a seguir mostram exemplos de storyboards.

Figura 7 - Exemplo de storyboards



Fonte: Lamothe ( 2015).



Figura 8 - Design de Interface e Interação: storyboards



Fonte: Rocha e Araujo (2012,, p.71).

## 2.2.4 Ferramentas de apoio a criação de jogos

Existem muitas ferramentas para criação completa e para a construção de protótipos de um jogo. Novak (2012) apresenta alguns exemplos destas ferramentas: Neverwinter Nights (Aurora Toolset), Half-Life (Valve Hammer Editor), e Unreal (Editor Unreal) e Far Cry (Editor Sandbox do CryENGINE) (NOVAK, 2012, p.342).

As ferramentas são variadas sendo que as que se destacam na produção de jogos são os motores ou engines. A seguir serão citados alguns grupos de outros tipos de ferramentas e posteriormente são relatados conceitos acerca dos motores de jogos.

### 2.2.4.1 Ferramentas de autoria

As ferramentas de autoria são utilizadas no desenvolvimento de projetos multimídia. Este tipo de ferramenta organiza os elementos do projeto (sons, gráficos,

animações, vídeos) e permite a edição destes elementos. É oferecido ao desenvolvedor um ambiente integrado que permite criar, editar, importar dados brutos e transformar estes dados em uma sequência de reprodução. Além disso, permitem responder, via linguagem estruturada ou métodos, as entradas do usuário. Estas ferramentas são utilizadas para criar a interatividade e interface com o usuário, além de agrupar elementos multimídia no projeto e apresentar este projeto na tela de forma coesa (SILVEIRA, 1999, p.31).

#### **2.2.4.2 Motores de jogos (ENGINES)**

O motor é “o programa principal que executa o jogo” (BRATHWAITE; SCHREIBER, 2009, p.14). Motor de jogo ou game engine “é o coração do game. Normalmente ele inclui um motor gráfico, um motor de física, suporte a animação, sons e inteligência artificial, dentre várias outras funcionalidades.” (MATTAR, 2010, p.91).

Mattar (2010) destaca um relatório da Education Arcade que indica que apesar de existirem muitas opções de motores de jogos comerciais no mercado, muitos projetos para jogos 3D costumam construir seu próprio motor de jogos customizado. Ele ainda afirma que existem motores de jogo de código aberto, mas que são ultrapassados e não se adequam as necessidades dos jogos educacionais (MATTAR, 2012, p.91).

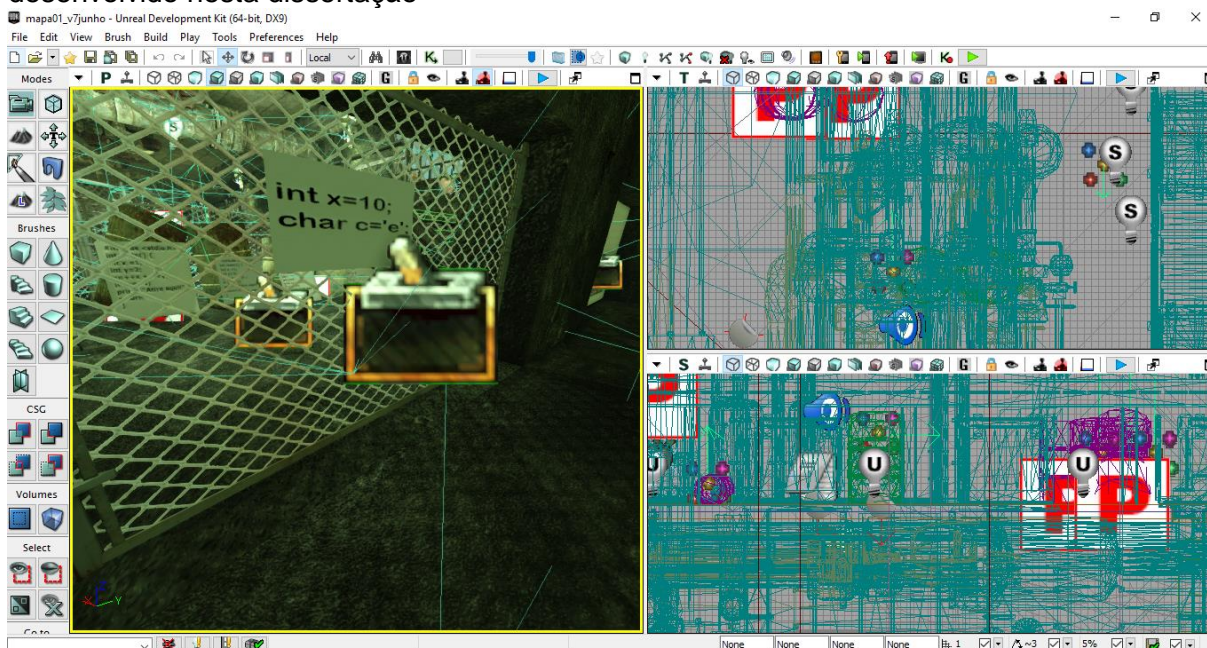
Apesar dos motores de jogos serem ferramentas que podem fornecer produtividade ao desenvolvimento de jogos, Fullerton, Swain e Hoffman, (2008), alertam que utilizar um motor de jogos pode otimizar o tempo de trabalho, porém pode amarrar seus projetos se ele tiver como base apenas os recursos que o motor de jogos pode oferecer. Apesar das limitações os motores de jogos permitem a criação, por exemplo, de protótipos de forma rápida e eficaz. Os autores indicam ainda que provavelmente um dos mais poderosos motores de jogo seja a Unreal Engine (FULLERTON; SWAIN; HOFFMAN, 2008, p.239). A figura 9 mostra o ambiente de desenvolvimento da Unreal Engine.

Figura 9 - Unreal Engine editor – UDK - Tiro em primeira pessoa – FPS



O jogo construído neste trabalho de dissertação utilizou a versão gratuita da Unreal Engine 3. Esta versão é denominada de UDK (Unreal Development KIT). O uso do UDK para fins não comerciais é gratuito. A figura 10 exibe o ambiente de produção do jogo na UDK.

Figura 10 - Ambiente da engine UDK com visualização do ambiente do jogo sério desenvolvido nesta dissertação



### 2.2.5 Fases do desenvolvimento de jogos

Novak (2012) nos apresenta as seguintes fases no processo de desenvolvimento de um jogo: conceito, pré-produção, protótipo, produção, alfa, beta, ouro e pós-produção. Cada uma destas fases irá focar em objetivos específicos e envolver a participação seletiva de membros da equipe.

A fase de conceito define qual é a ideia principal do jogo e documenta essa ideia em um documento de conceito, que será gerenciado por uma equipe ainda pequena, formada geralmente por um designer, um programador, um artista e um produtor. Esta fase irá terminar quando é decidido iniciar o planejamento do projeto (fase de pré-produção).

Na fase de pré-produção a equipe utiliza o conceito do jogo e começa a desenvolver o planejamento. Documentos como guia de estilo, de arte e plano de produção são criados. O mais importante documento criado nesta fase é o documento de design do jogo. A fase de protótipo cria protótipos para avaliar o andamento do desenvolvimento do jogo. Uma vez que o protótipo foi aprovado inicia-se a fase de produção em que a equipe de desenvolvimento, de fato, começa a desenvolver o jogo.

As fases a seguir podem diferir a depender do projeto. A fase alfa é um marco que indica que toda codificação do jogo teoricamente está completa. Neste ponto já existe uma versão jogável do início ao fim. Contudo nesta fase o jogo ainda possui

muitos erros e é uma versão instável. A equipe de testes deve criar um plano de testes para testar os módulos e demais componentes do jogo, registrando os erros, os resultados dos testes de desempenho e criando um banco de dados com os erros encontrados. Esta fase inaugura o primeiro contato do jogo com pessoas de fora da equipe de desenvolvimento.

A fase beta também é um marco e indica que o jogo está em uma versão mais estável e que todos os conteúdos estão no jogo. Nesta fase defeitos e questões identificadas na versão alfa são analisadas e corrigidas. O objetivo principal da fase beta é retirar a maior quantidade possível de erros e estabilizar o projeto antes do lançamento do jogo. Desta forma, pessoas de fora da equipe, denominados de testadores beta são recrutados para testar o jogo. Após o jogo ultrapassar a fase beta, finalmente entra na versão ouro. Nesta fase o jogo final está pronto e é lançado e liberado ao mercado e público-alvo. (NOVAK, 2012, p.352-362).

### 2.3 JOGOS SÉRIOS COMO APOIO AO ENSINO E APRENDIZAGEM DE LINGUAGENS DE PROGRAMAÇÃO

Nos últimos anos, ocorreu uma popularização da ideia de usar jogos de computador em cursos de introdução à programação e, portanto, esta não é uma ideia nova. Apesar disso, este recurso ainda não é muito utilizado na grande maioria dos cursos de programação existentes hoje (PARALIČ; PIETRIKOVÁ, 2014). Em relação ao aprendizado de programação nas universidades Mitamura, Suzuki e Oohori (2012, p.1), declaram que este “nível de aprendizagem de programação difere significativamente dependendo da formação acadêmica do estudante”. Os autores ainda destacam que existem “problemas com a aprendizagem no ensino de programação tradicional porque o aluno não se sente engajado e não está interessado”.

Diante disso, faz-se necessário propor ferramentas para apoiar o ensino de programação e tentar mitigar os riscos envolvidos na trajetória de aprendizado do aluno iniciante. A seguir serão relatados diversos trabalhos relacionados com o tema de ensino de programação utilizando jogos sérios. O foco, no entanto, será relatar trabalhos relacionados com o tema desta dissertação, que é a criação de jogos sérios com o objetivo de ensino da linguagem de programação C, para o público de universidades do curso de computação.

### 2.3.1 Jogos sérios no ensino de programação

Muitos trabalhos têm sido desenvolvidos com o foco na criação de jogos digitais, com o objetivo de deixar mais interessante e efetivo o estudo e ensino de programação de computadores. Contudo o número de pesquisas que criam jogos para o ensino específico da linguagem de programação C é limitado. Nesta seção serão relatados alguns destes trabalhos considerados relevantes e convergentes ao tema desta dissertação.

Um ambiente de ensino e aprendizagem formal colaborativo acerca de conceitos introdutórios da linguagem C, foi proposto por Chang, Chou e Chen (2010). Este ambiente possui itens interativos e também um jogo, mostrado na figura 11 a seguir, baseado na solução de problemas com código C, para controlar o movimento de um bomberman. Bomberman é um jogo de estratégia baseado em labirinto, que tem fases que são completadas depositando bombas em lugares estratégicos para destruir obstáculos e inimigos. Este ambiente tem como objetivo auxiliar os alunos em atividades de aprendizagem se configurando como um complemento as aulas ministradas.

Figura 11 - BomberMan sobre linguagem C



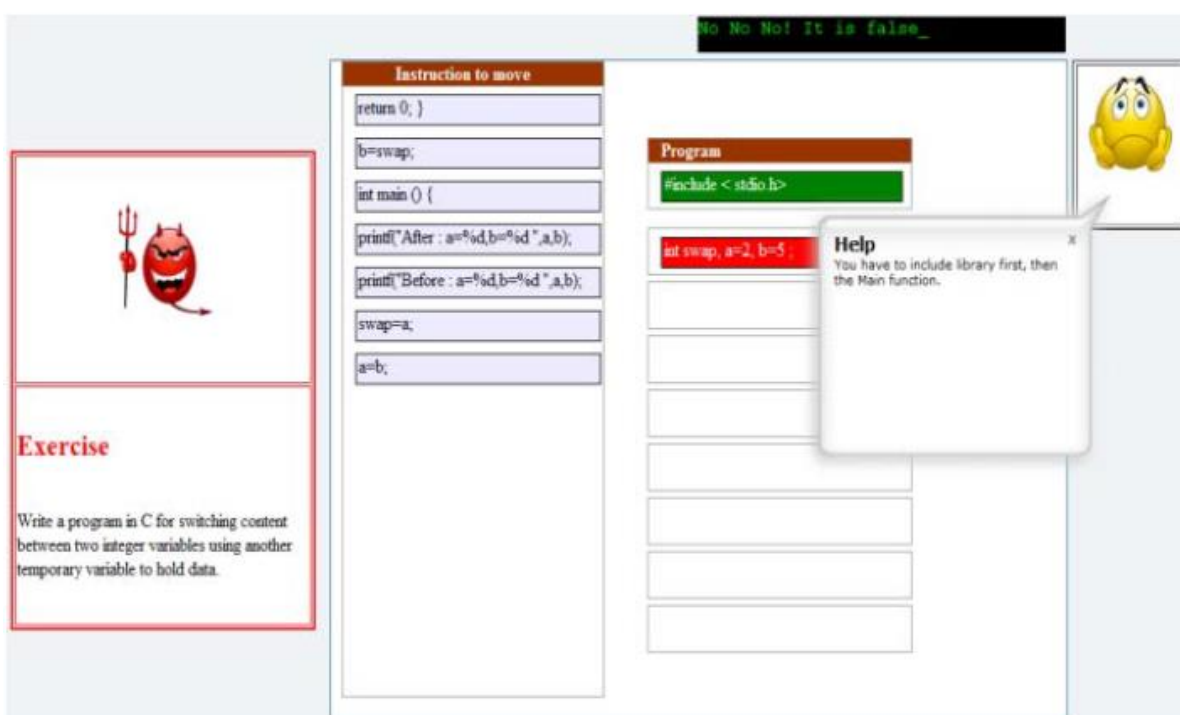
Fonte: Chang , Chou e Chen (2010, p.706).

Neste trabalho, Chang, Chou e Chen (2010) proporcionaram um ambiente interativo com aprendizagem colaborativa via internet. No entanto, graficamente o

jogo é simples e não proporciona um ambiente gráfico sofisticado. Jogos do tipo bomberman não são 3D, que é o tipo de jogo usualmente jogado pelos alunos universitários.

Um outro jogo de aprendizagem no ensino superior, com objetivo de ensinar a linguagem de programação C, foi concebido no trabalho de Khenissi, Essalmi e Jemni (2013). O jogo utiliza o drag and drop(arrastar e soltar) como recurso central. A figura 12 mostra uma tela do jogo.

Figura 12 - Tela do jogo de arrastar e soltar



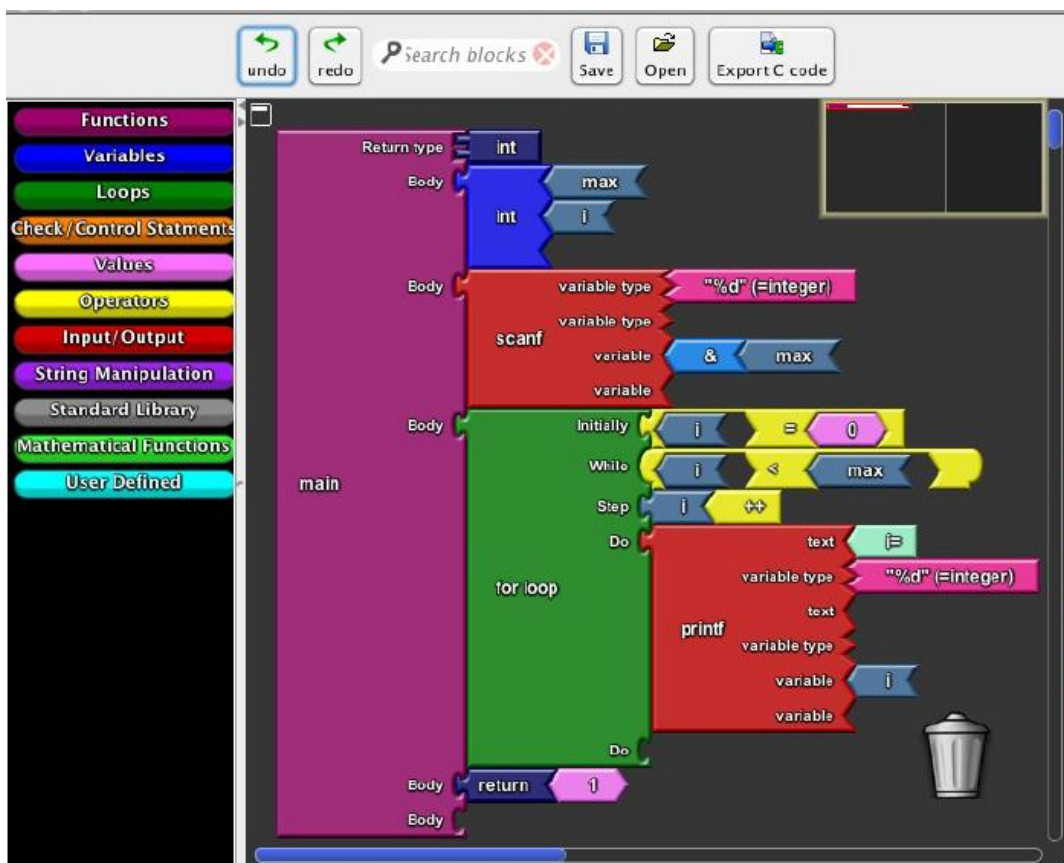
Fonte: Khenissi, Essalmi e Jemni (2013, p.326).

No jogo proposto no trabalho de Khenissi, Essalmi e Jemni (2013) existe a possibilidade de se utilizar outras linguagens de programação. O nível de customização é alto pois, permite que professores criem seus próprios desafios e perguntas no jogo. Além disso, ao jogar o jogo o aluno recebe feedbacks quanto comete erros. Apesar disso, o jogo ainda carece de desafios mais elaborados e sua interface e gráficos também não são sofisticados.

Outro trabalho mais recente, nesta mesma linha de pesquisa, foi desenvolvido por Charalampos, Nektarios e Stavros (2015). Eles propuseram um jogo denominado Block-C, que se configura como um ambiente em que iniciantes na

programação C, podem aprender a programar nesta linguagem. O ambiente fornece orientação e consegue evitar erros frequentes de sintaxe e codificação.

Figura 13 - Interface gráfica Block-C



Fonte: Charalampos, Nektarios e Stavros (2015, p.3).

O jogo baseado no ambiente em blocos proposto por Charalampos, Nektarios e Stavros (2015), figura 13, fornece uma forma de codificar em linguagem C, através do encaixe de blocos coloridos de códigos. Além disso, após o encaixe dos blocos o código fonte em texto, correspondente a arrumação destes blocos pode ser gerada.

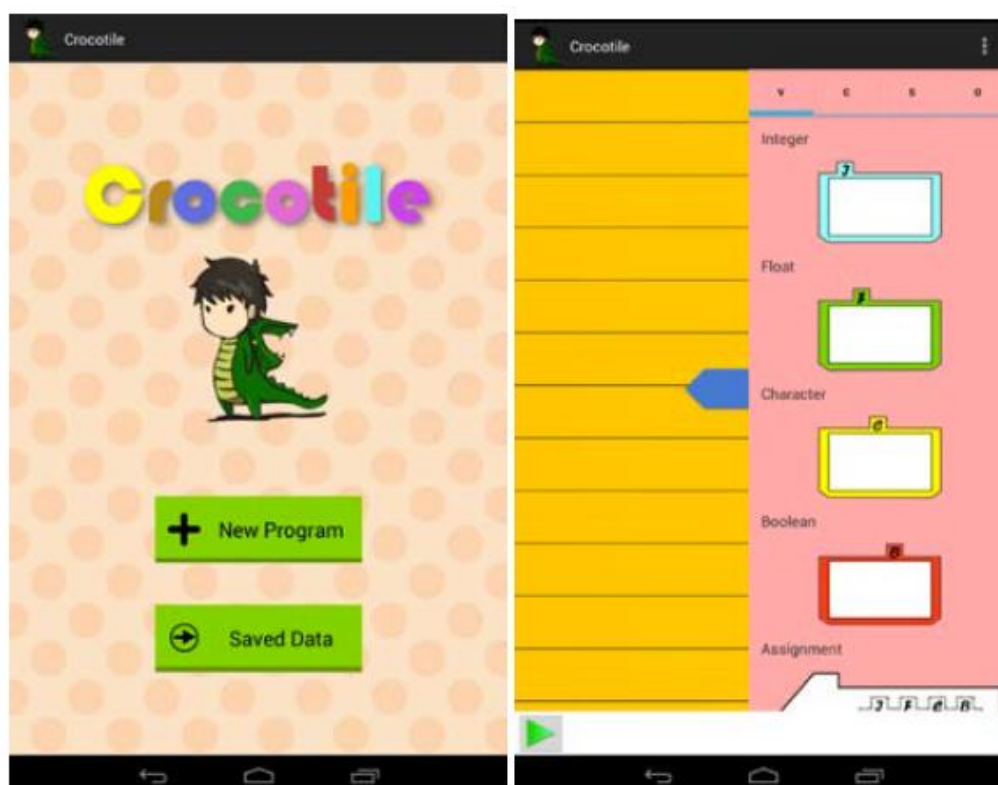
A programação utilizando blocos com recurso de arrastar e soltar pode eliminar erros de digitação e isso pode ocasionar menos erros de compilação oriundos de erros de sintaxe (MEEKLAI et al., 2015, p.2). Isto de fato evita erros como a falta de ponto e vírgula e erros na sintaxe da linguagem C. Contudo o aprendizado através de erros é um dos pilares do aprendizado por jogos. “Erros fazem parte do progresso no jogo e os jogadores aprendem a aprender, dentre outros pontos”. (ALDRICH 2007 apud MATTAR 2010, p.27). Apesar do Block-C permitir que o jogador erre, estes erros se limitam a semântica (significado) contido



nos códigos montados pelo jogador. Mas uma vez este jogo não possui interface gráfica avançada pois é baseado na interface de jogos de encaixe em blocos.

Meeklai et al., (2015), criaram em seu trabalho o jogo crocotile, que auxilia alunos do ensino médio do Japão no aprendizado de conceitos simples da linguagem C. O jogo foi desenvolvido para mobile e também possui o estilo arrastar e soltar(drag and drop) como recurso principal utilizado nos desafios do jogo.

Figura 14 - Tela do jogo CROCOTILE



Fonte: Meeklai et al. (2015, p.2).

O jogo proposto por Meeklai et al., (2015), novamente não possui interface gráfica sofisticada e também tenta evitar, através do encaixe dos blocos(tiles), que os alunos cometam erros comuns de sintaxe da linguagem C. Novamente no jogo não é dado muitas possibilidades de aprendizagem por erros, e os conceitos envolvidos são apenas introdutórios da linguagem C.

Outra proposta foi oferecida por Miljanovic, e Bradbury (2016), que conceberam o jogo sério Robot ON!, O jogo tem foco na melhora acerca da compreensão da programação e é destinado a jogadores que nunca tenham programado antes. Apesar de ser um jogo com foco na linguagem C++, ele é semelhante às propostas anteriores, e como a linguagem de programação C++

possui sintaxe algumas vezes semelhante a linguagem C, este jogo foi relatado nesta seção. Neste jogo, se os jogadores não seguirem as instruções do jogo eles falham, e tem que repetir um nível caso não demonstrarem que compreenderam o programa.

Figura 15 - Um tela do jogo Robot ON



Fonte: Miljanovic e Bradbury (2016).

A tabela a seguir indica um resumo das características dos jogos e adiciona o jogo Unlock-c proposto neste trabalho de dissertação.

Tabela 5 – Resumo dos jogos sérios sobre linguagem C

Autores	Jogo	Recursos Gráficos Avançados?	Gênero	Plataforma	Nível de customização
Chang, Chou e Chen (2010)	Boombberman	Não	Estratégia	Internet	Baixo
Khenissi, Essalmi e Jemni (2013)		Não	Quebra-Cabeça	PC	Alto
Charalampos, Nektarios e Stavros (2015).	Block-C	Não	Quebra-Cabeça	PC	Baixo
Meeklai et al., (2015)	Crocotile	Não	Quebra-Cabeça	Mobile	Baixo
Miljanovic, Bradbury (2016), Santos(2016)	Unlock-C	Sim	Aventura	PC	Baixo
			<b>FPS</b>	<b>PC</b>	<b>Baixo</b>

Outros trabalhos envolvendo jogos foram desenvolvidos com foco no ensino e aprendizagem de disciplinas do curso de computação na educação superior. Uma

revisão sistemática de Battistella e Wangenheim (2016), identificou 107 jogos com o foco principalmente para o ensino de engenharia de software, fundamentos de programação, redes, algoritmos e complexidade e segurança.

Em relação ao ensino de linguagem de programação, o volume de trabalhos é bem maior para outras linguagens (Java, Python, etc) do que se comparado ao número de trabalhos com foco na linguagem C. Além disso, apesar de serem relatados poucos trabalhos em que foram concebidos jogos sofisticados, não foram encontrados trabalhos com jogos sérios 3D, com gráficos e efeitos aprimorados com o objetivo de ensino e aprendizagem da linguagem de programação C.

## 2.4 CONCEITOS SOBRE LINGUAGEM DE PROGRAMAÇÃO

A sintaxe de uma linguagem de programação é a forma de suas expressões, sentenças e unidades de programas. Sua semântica é o significado dessas expressões, sentenças e unidades de programas. Como exemplos Sebesta (2011), nos oferece o seguinte trecho de código: **while (expressão booleana) sentença**. Ele indica que a semântica deste trecho é que quando for verdadeiro o valor da expressão booleana, a sentença dentro da estrutura será executada. Se o valor da expressão booleana for falso a execução do programa continua a partir da linha imediatamente inferior, sem entrar na estrutura while, e desta forma não irá executar a sentença. (SEBESTA, 2011, p.136-137, grifo nosso).

A utilização de um conjunto de regras sintáticas e semânticas para definir um programa de computador é denominada linguagem de programação. A linguagem de programação, portanto configura-se como sendo um método padronizado para comunicar instruções para um computador. “Permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias.” (WIKEPEDIA, 2015)

### 2.4.1 Linguagem de Programação C

A linguagem de programação C está presente em diversos cursos de graduação em computação ao redor do mundo. Como destacam Parreira, Petersen e Craig (2015, p.347) “a linguagem C é uma parte importante de muitos programas de graduação em computação, uma vez que fornece um ambiente para interagir diretamente com a memória e explorar conceitos de sistemas e programação.”

A linguagem C é uma linguagem utilizada para desenvolver muitos software. É uma linguagem procedural, que utiliza rotinas, sub-rotinas, denominadas funções. Desta forma é uma linguagem de programação muito importante para ser utilizada como base para aprender programação (CHANG; CHOU; CHEN, 2010, p.704)

Na sociedade da informação moderna é fundamental melhorar a eficiência no ensino das habilidades de programação de computadores. (MEEKLAI, et a., 2015, p.1). Contudo, existem alguns obstáculos no ensino e aprendizagem de programação que necessitam ser superados. A baixa motivação, a complexidade do assunto e a diferença existente entre as gerações de professores e alunos são algumas das dificuldades apontadas por Khenissi, Essalmi e Jemni (2013), no ensino de uma linguagem de programação.

Em destaque está a linguagem de programação C, na qual “existem muitas barreiras para programadores iniciantes quando confrontados com esta linguagem.”(CHARALAMPOS; NEKTARIOS; STAVROS, 2015, p.1). Estes mesmos autores Charalampos, Nektarios e Stavros (2015), relatam estas barreiras como sendo: a orientação da linguagem para baixo nível de abstração, sua sintaxe obscura, e mensagens de erros de compilação e execução muitas vezes ambíguas e abstratas.

Desta forma, os alunos que começaram a programar após entrar na universidade, podem se formar sem aprender programação suficientemente. Apesar deste cenário, a utilização de jogos digitais pode aumentar a motivação do aluno e diminuir a curva de aprendizado se comparado com o ensino através de programação baseada em texto (MEEKLAI, et al., 2015, p.1-2).

O trabalho desta dissertação criou um jogo sério 3D, concebido através da execução de uma metodologia criada especificamente para a produção de jogos sérios educacionais. A forma como foi criada esta metodologia será descrita no próximo capítulo.

### **3 PROCESSO DE DEFINIÇÃO DA METODOLOGIA DE DESENVOLVIMENTO DO JOGO SÉRIO EDUCACIONAL**

#### **3.1 INTRODUÇÃO**

Este capítulo tem o objetivo de apresentar a definição de uma metodologia adequadamente embasada que norteia a concepção, criação e avaliação de um jogo sério de cunho educacional. Será mostrada a definição de um processo para a geração desta metodologia em que será elucidado de que forma a metodologia foi gerada. Um processo de software “inclui todas as atividades envolvidas no desenvolvimento do software [...] especificação, desenvolvimento, validação e evolução são parte de todos os processos de software”. (SOMMERVILLE, 2011, p.16). Neste caso o produto de software será o jogo sério. Além disso, será exposta neste capítulo a descrição em detalhes de todas as disciplinas e atividades pertencentes a metodologia customizada concebida.

#### **3.2 NECESSIDADE DEFINIÇÃO DE METODOLOGIA CUSTOMIZADA**

Existem inúmeros estudos sobre avaliação e desenvolvimento de jogos sérios. Apesar disso, “[...] a aplicação de teorias adequadas para a concepção de ferramentas educacionais eficazes baseadas em jogos não foi suficientemente estudada.” (SHIRATUDDIN, 2011, p.2).

Outros estudos indicam metodologias para desenvolver um jogo sério utilizando elementos da engenharia de software. Ocasionalmente estão inseridos nestas metodologias de desenvolvimento, quando da construção de um jogo sério educacional, elementos de outras áreas do conhecimento, em especial da pedagogia.

Contudo Catalano, Luccini e Mortara (2014) argumentam que apesar de existir uma crescente consciência sobre o potencial de uso dos jogos sérios para a educação, ainda ocorre uma falta de metodologias, diretrizes e boas práticas sobre como desenvolver jogos sérios eficazes e como integrá-los nos processos de aprendizagem e de formação reais. Acrescentando-se a este cenário, Barbosa (2014, et al.) alertam que os jogos sérios têm sido utilizados numa vasta gama de setores, tais como: militar, saúde, educação, e corporativa, entre outros, mas que, no entanto, apenas algumas metodologias foram propostas para orientar a concepção de jogos sérios.

Além destas argumentações anteriores Pessini, Oliveira e Kemczinski (2014) relataram em um mapeamento sistemático que “quanto ao uso de metodologias de desenvolvimento na produção dos jogos sérios, 30 jogos (90,9%) não mencionaram o uso de metodologias”.(PESSINI; OLIVEIRA; KEMCZINSKI, 2014, p.540) Este mapeamento sistemático objetivou identificar como estão sendo desenvolvidos os jogos sérios na área de educação em informática no Brasil.

Desta forma ao serem estudados diversos trabalhos sobre metodologias existentes acerca de construção jogos sérios, foi percebido um cenário diversificado no que tange as especificidades e objetivos desta construção. Observou-se ainda que:

- a) Não existe uma metodologia que se destaque de forma clara de outras existentes, e que seja aceita de forma consensual.
- b) Dentre os trabalhos estudados foi encontrado um mapeamento sistemático realizado por Oliveira, Hounsell e Kemczinski (2014), que tem o foco analisar metodologias de desenvolvimento de software direcionadas a jogos sérios e que selecionou 17 metodologias para o desenvolvimento de jogos sérios educacionais.
- c) Uma análise dessas 17 metodologias identificou que possuíam pontos fortes e fracos distintos.
- d) Identificou-se também que a partir delas poderia ser montada uma metodologia consistente que incorporasse pontos fortes das metodologias usadas como referência.

As metodologias de desenvolvimento de jogos sérios propostas nestes 17 artigos foram classificadas pelos autores do mapeamento sistemático quanto ao seu tipo de ciclo de vida (sequencial, iterativo ou híbrido).

Além desta classificação, foi identificado se existia alguma forma de participação de usuários no processo e em quais disciplinas esta participação ocorria. Esta classificação é mostrada na figura 16.

Figura 16 – Classificação das Metodologias

	WorkFlow	Tipo WorkFlow	Particip. User
MARCOS e ZAGALO - 2011	Pg 5	Sequencial	N
BORRO-ESCRIBANO et al. - 2014	Pg 6	Iterativo	S
LAZAROU - 2011	Pg 4 e 5	Sequencial	S
RUPPEL e SCHATZ - 2011	Pg 6	Híbrido (Sequencial e Iterativo)	N
VAN DER ZEE et al. - 2012	Pg 3	Sequencial	N
AMANATIADOU e VAN DER WEERD - 2009	Pg 4	Sequencial	S
ISMAILOVIC et al - 2010	Pg 2	Iterativo	S
ROCHA et al. - 2012	Pg 4	Iterativo	S
SHIRATUDDIN - 2011	Pg 4	Iterativo	N
TORRENTE et al. - 2014	Pg 2	Iterativo	S
ASUNCION et al. - 2011	Pg 2	Iterativo	S
BROWN et al - 2009	Pg 4	Iterativo	S
COENEN et al - 2013	Pg 6	Iterativo	S
DE ABREU et al. - 2011	Pg 4	Sequencial	S
EAGLE e BARNES - 2012	Pg 2	Sequencial	S
RANKIN et al - 2008	Pg 1	Iterativo	S
LUKOSCH et al - 2012	Pg 3	Iterativo	S

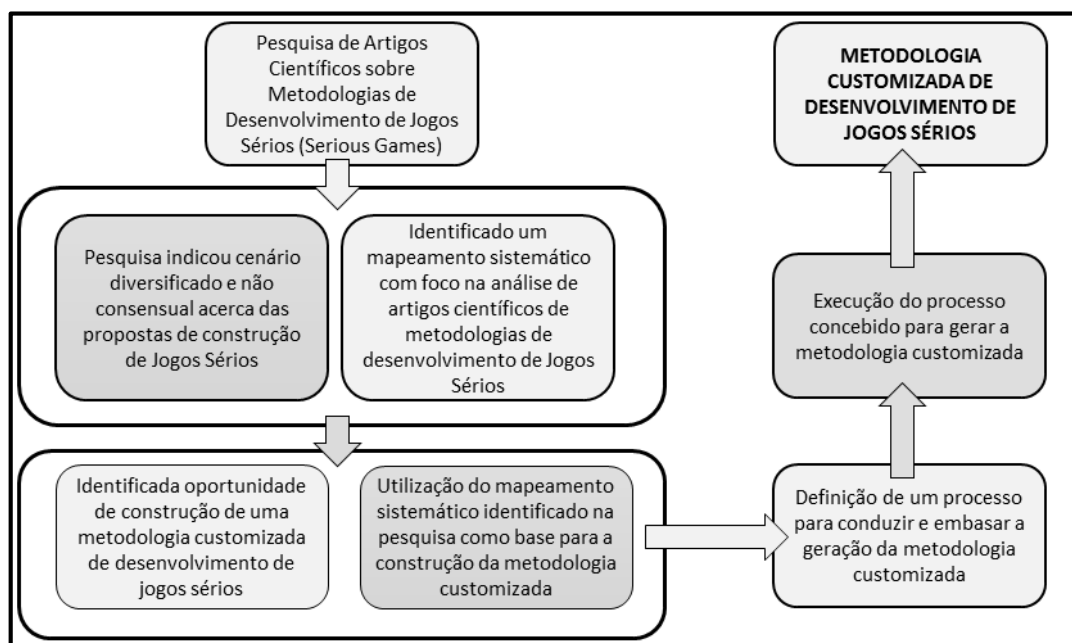
Fonte: Oliveira, Hounsell e Kemczinski (2014).

O produto desse mapeamento sistemático citado acima foi a fonte de informação fundamental para a construção da metodologia de desenvolvimento de jogos digitais sérios aqui proposta. Uma análise inicial das 17 metodologias identificadas pelo mapeamento sistemático foi efetuada, com vistas à seleção de uma metodologia para ser usada nesta dissertação.

Essa análise identificou que cada uma das 17 metodologias apresentadas possui pontos específicos fortes e fracos; em alguns casos as metodologias são sucintamente mencionadas, em outras há descrições mais detalhadas; nessas, há atividades descritas em detalhes e outras de forma muito sucinta; foco maior em engenharia de software em alguns casos, e em outros, foco em aspectos didáticos; características semelhantes entre determinadas metodologias, e marcadamente distintas em outros casos. Como consequência dessa análise inicial das metodologias, identificou-se que não era possível selecionar uma metodologia específica. Havia a necessidade de se efetuar a construção de uma metodologia customizada a partir dos aspectos e atividades mais interessantes das 17

metodologias oriundas do mapeamento sistemático. A abordagem utilizada para conceber esta metodologia customizada está descrita na figura 17:

Figura 17 - Fluxo de criação da metodologia customizada



### 3.2.1 Classificação das atividades em um conjunto de disciplinas comuns da engenharia de software

Os autores do mapeamento sistemático Oliveira, Hounsell e Kemczinski, (2014), adotaram como disciplinas da engenharia de software as disciplinas de requisitos, projeto, implementação, testes e avaliação. Os autores, inclusive resumiram um quadro geral, descrito na figura 18, o qual descrevem para cada uma das metodologias dos 17 artigos, quais são as atividades relacionadas com cada disciplina.

Diante disso percebeu-se uma oportunidade de se utilizar este quadro geral como base para avaliação de todas estas atividades, visto que as mesmas foram agrupadas pelos autores do mapeamento sistemático de acordo com sua presença em cada uma das disciplinas de requisitos, projeto, implementação, testes e avaliação. Assim estas disciplinas também irão compor a nova metodologia customizada a ser concebida.



Figura 18 - Resultado da análise do mapeamento sistemático em relação às disciplinas e atividades executadas nas mesmas

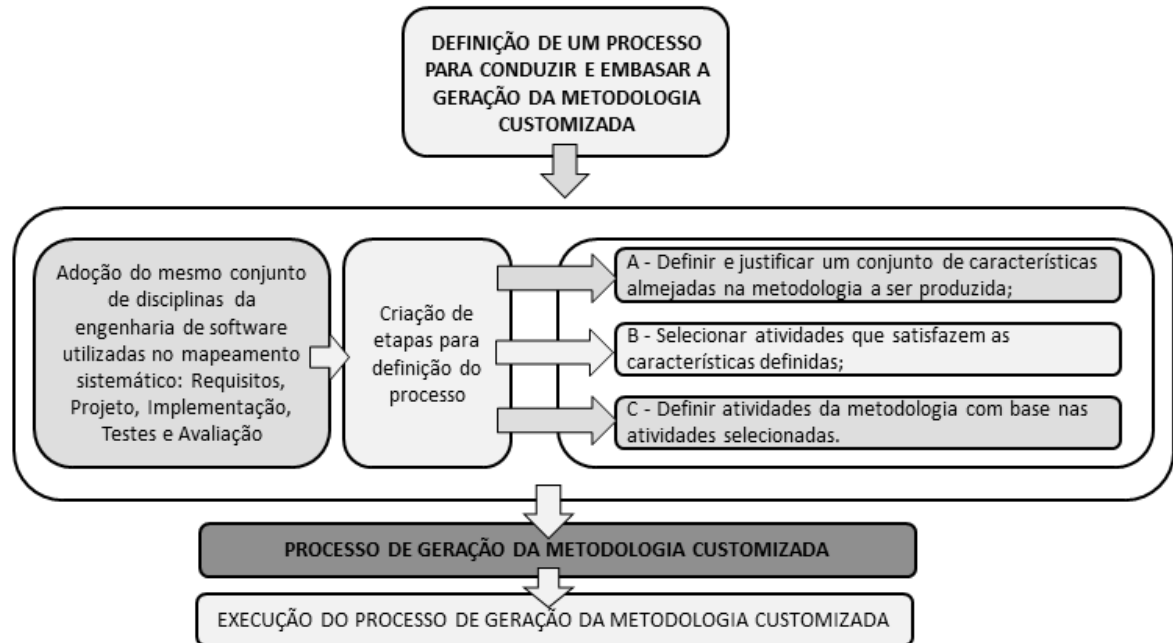
	Requirements	Design	Implementation	Test	Evaluation
MARCOS e ZAGALO - 2011	Starting Concept	Concept Design Narrative Design Experience Design Aesthetic Musing Game Design	Game Implementation	Game Deployment Planning	
BORRO-ESCRIBANO et al. - 2014	Specification	Game Design	Simulation Development		Quality Assurance
LAZAROU - 2011	<i>Identify and define Activity Systems</i> Identify personal requirements Solve contradictions	Development Pedagogy Design	Development		<i>Usability Testing</i> Implementation and Data Collection <i>Evaluation</i>
RUPPEL e SCHATZ - 2011	Model of Reality Value Proposal Game Concept Use-Cases Planning Requirements	Analysis Design	Implementation	Testing	Evaluation
VAN DER ZEE et al. - 2012	Setting objectives and parameters	Model development Decisions about representation	Construction and Modification		Preparation for use
AMANIATIADOU e VAN DER WEERD - 2009	Concept	Pre-production	Production	Post-production	
ISMAILOVIC et al - 2010	Design an Idea		Generate a Prototype		<i>Evaluate with Users</i>
ROCHA et al. - 2012	Plan Analysis	Design	Implementation	Evaluation Configuration Integration and Test	<i>Execution</i> <i>Evaluation</i>
SHIRATUDDIN - 2011	Concept	Concept Pre-Production	Prototype Production		Post-Production
TORRENTE et al. - 2014	Analysis	Game Design	Implementation		<i>Quality Assurance</i>
ASUNCION et al. - 2011	Pre-Production		Production		<i>Final Test</i>
BROWN et al - 2009	Design		<i>Development</i>		Release Evaluation
COENEN et al - 2013	Problem Formulation	Building Intervention			<i>Data Gathering</i> Reflection and Learning
DE ABREU et al. - 2011	Study of Process	Study of Technology Modelling	Development		Assessment <i>Evaluation</i>
EAGLE e BARNES - 2012	<i>Identify users and needs</i> <i>Observation</i>	Design-Prototype	Translate		<i>Evaluate</i>
RANKIN et al - 2008	<i>Observational Studies</i> Conceptualization	Conceptualization	Prototyping		<i>Usability Playtesting</i>
LUKOSCH et al - 2012	Analysis	Synthesis		Simulation	<i>Evaluation</i>

Fonte: Oliveira, Hounsell e Kemczinski( 2014)

### 3.3 PROCESSO PARA A GERAÇÃO DA METODOLOGIA

Para embasar a geração da metodologia customizada a partir das 17 analisadas no mapeamento sistemático referenciado, foi definido, como parte do trabalho desta dissertação, um processo específico para conceber a metodologia. Nesta seção é apresentado em detalhes esse processo presente na figura 19. Seções subsequentes descrevem a execução deste processo e a metodologia gerada a partir dele, que será utilizada para o desenvolvimento do jogo sério educacional.

Figura 19 - Descrição do processo concebido para criar uma metodologia customizada de desenvolvimento do jogo sério



### 3.3.1 Descrição do processo para geração da metodologia

Uma vez identificada a necessidade de geração de uma metodologia customizada, foi definido o processo para construção dessa metodologia. Este processo possui as seguintes etapas:

- a) **Definir e justificar um conjunto de características almejadas na metodologia a ser produzida;**
- b) **Selecionar atividades que satisfazem as características definidas;**
- c) **Definir atividades da metodologia com base nas atividades selecionadas.**

Neste contexto característica deve ser avaliada como sendo um requisito em alto nível. Um requisito é definido como "uma condição ou uma capacidade com a qual o sistema deve estar de acordo." (WTHREEX, 2015). Também neste cenário uma disciplina é "um conjunto de atividades relacionadas a uma área de interesse importante em todo o projeto" No que diz respeito a atividades o Processo Unificado da Rational (RUP) define uma atividade como o trabalho realizado por um papel, quer seja usando e transformando artefatos de entrada ou produzindo artefatos de saída novos e alterados (WTHREEX, 2015).

O papel define o comportamento e as responsabilidades de um indivíduo ou de um conjunto de indivíduos que trabalham juntos como uma equipe, no contexto de uma organização de engenharia de software. Em relação aos artefatos eles são produtos de trabalho finais ou intermediários produzidos e usados durante os projetos (WTHREEX, 2015).

A seguir serão explanadas as etapas A, B e C listadas anteriormente. Para cada destas etapas, será apresentada sua descrição e a justificativa da sua existência.

#### **A. Definir e justificar um conjunto de características almeçadas na metodologia a ser produzida;**

##### **Descrição**

O objetivo desta etapa é definir um conjunto de características que serão utilizadas para avaliar cada uma das atividades das respectivas disciplinas das metodologias utilizadas como referência aqui denominadas metodologias fonte. A meta é selecionar atividades que contemplem estas características. Desta forma será possível verificar se a descrição da atividade relata, no próprio artigo, as subatividades e ferramentas geradas para chegar até os artefatos e com isso garantir as chances que a mesma possa ser replicada.

Outras avaliações deverão ser realizadas para incluir tarefas e elementos de cunho pedagógico e permitir a realização de boas práticas da engenharia de software. O produto desta etapa serão quadros comparativos que indicarão a existência de determinadas características em cada atividade de cada disciplina. Além da definição de características, outra tarefa desta etapa é justificar a relevância destas características.

##### **Justificativa**

A definição de características desejadas para as atividades das disciplinas da metodologia a ser produzida, poderá conduzir a uma escolha mais consistente e coerente de atividades. Desta forma estas características servirão de base para permitir mapear e selecionar atividades. Ao ser efetuado um mapeamento será

possível traçar um panorama que vai indicar de que forma estas características estão relacionadas entre as atividades. Isso permitirá também avaliar e decidir em próximas etapas como a disciplina customizada será composta.

## **B. Selecionar atividades que satisfazem as características definidas;**

### **Descrição**

Esta etapa consiste mapear em cada atividade a presença das características a serem definidas quando da execução da etapa A do processo. Eventualmente se uma ou mais atividades contemplarem todas as características listadas em A, estas atividades poderão compor em sua totalidade a disciplina customizada. Caso contrário uma atividade customizada será configurada mesclando elementos e tarefas de todas as atividades de modo que abranja todas as características definidas para a sua respectiva disciplina. O produto desta etapa serão quadros comparativos que indicarão a existência de determinadas características em cada atividade de cada disciplina.

### **Justificativa**

Ao ser efetuado este mapeamento será possível traçar um panorama que vai indicar de que forma estas características estão relacionadas entre as atividades das metodologias usadas como referência. Isso permitirá avaliar e decidir em próximas etapas como a disciplina customizada será composta. Após esta análise será possível criar de forma mais sólida disciplinas customizadas que abranjam todas estas características definidas. Assim o processo de compor uma atividade que contemplem estas características específicas poderá ser mais preciso.

## **C. Definir atividades da metodologia com base nas atividades selecionadas.**

### **Descrição**

Esta etapa consiste em selecionar alguma atividade que possua todas as características mapeadas em B ou, o que se pressupõe que vai ocorrer, configurar uma nova atividade mesclando elementos de todas as demais atividades mapeadas. Uma vez efetuada a seleção e mescla de elementos e artefatos das atividades

mapeadas em B é necessário efetuar uma síntese e posteriormente uma descrição textual detalhada da nova atividade composta. O produto desta etapa será uma lista de atividades selecionadas ou customizadas para cada uma das disciplinas de requisitos, projeto, implementação, testes e avaliação. Além desta lista deverá ser realizada uma síntese acerca destes elementos e posteriormente uma descrição textual concisa sobre cada uma das atividades a serem executadas.

### **Justificativa**

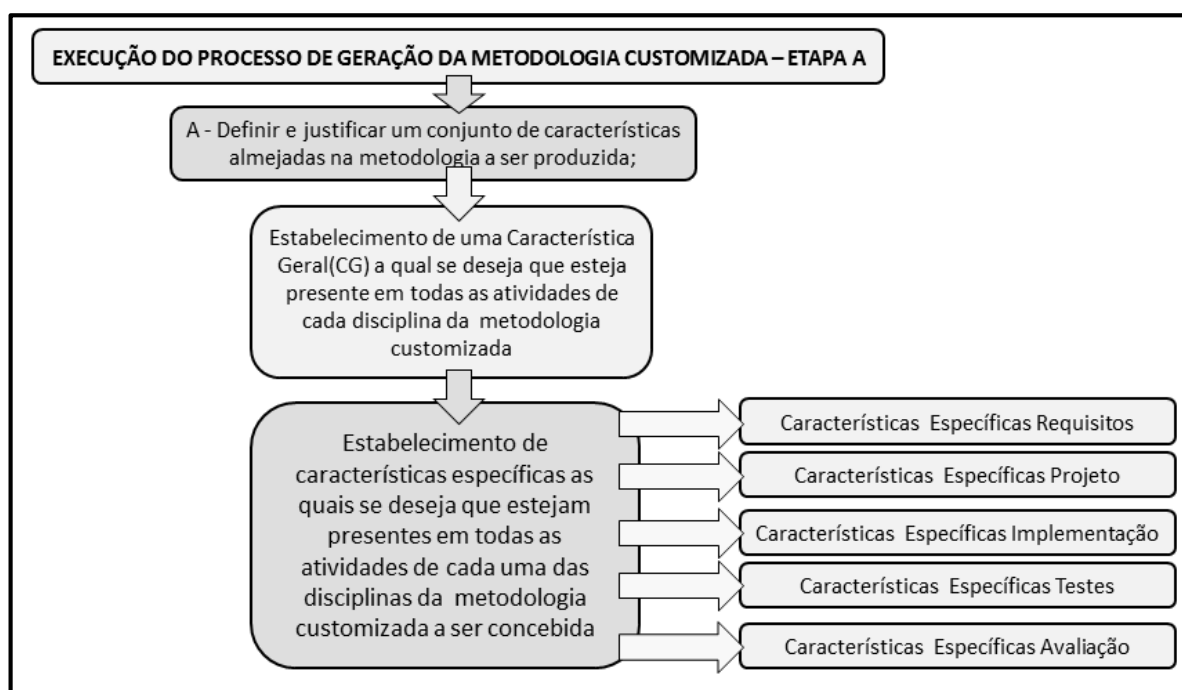
No caso de não existir alguma atividade que possua todas as características mapeadas em B. Cada disciplina com suas atividades customizadas será diferente das contidas nas metodologias fonte, pois seu conjunto de características não será possuído por nenhuma das outras disciplinas das metodologias fonte quando analisadas isoladamente.

Assim a seleção de elementos de atividades fonte é parte importante no processo de composição das atividades customizadas, pois a mesmas irão contemplar todas as características almejadas e definidas para cada disciplina da metodologia a ser construída. A síntese desta análise realizada irá consolidar os elementos presentes em cada atividade customizada. Uma descrição textual precisa de cada atividade e seus elementos é fundamental para permitir sua posterior execução.

### **3.4 EXECUÇÃO DO PROCESSO DE GERAÇÃO DA METODOLOGIA CUSTOMIZADA**

Esta seção descreve a execução do processo de geração da metodologia customizada, utilizando o processo apresentado na seção 3.4.1. A concepção da metodologia será iniciada com a execução da etapa A que está descrita na figura 20.

Figura 20 - Execução da etapa A do processo de geração da metodologia customizada



**A - Definir e justificar um conjunto de características almejadas na metodologia a ser produzida.**

Deste conjunto de características foi estabelecida inicialmente uma característica geral (CG). A característica geral(CG) foi estabelecida da seguinte forma:

- A atividade possui uma descrição suficientemente clara e, além disso, esta descrição é suficientemente detalhada para que possa ser executada.

Além da característica geral (CG) foram também estabelecidas características específicas para cada uma das disciplinas de requisitos, projeto, implementação, testes e avaliação. Essas características específicas estão definidas nas figuras a seguir:

Figura 21 - Características específicas disciplina requisitos

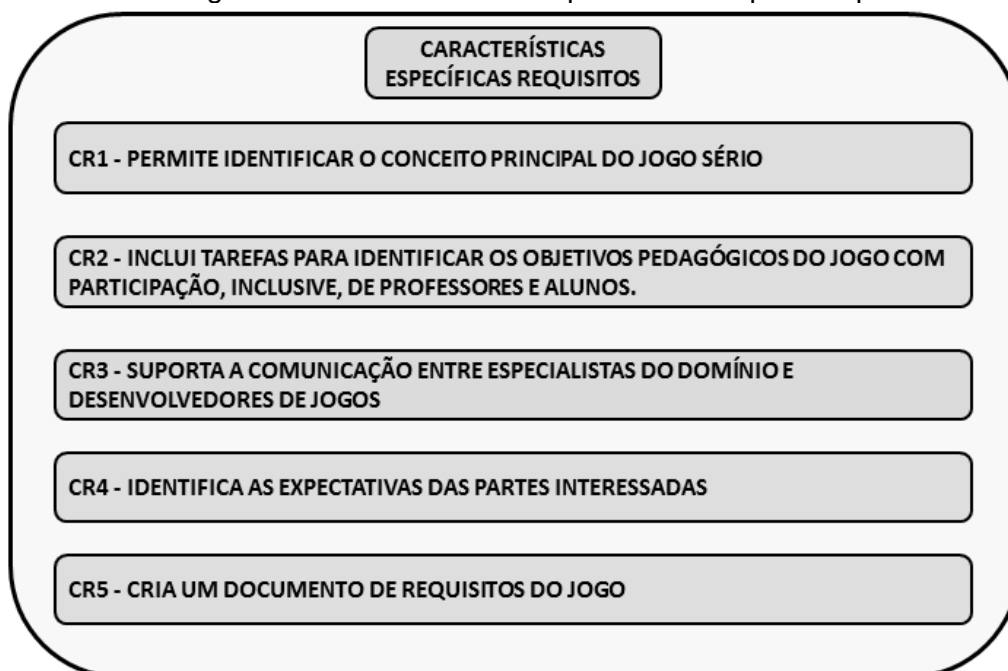


Figura 22 - Características específicas disciplina projeto

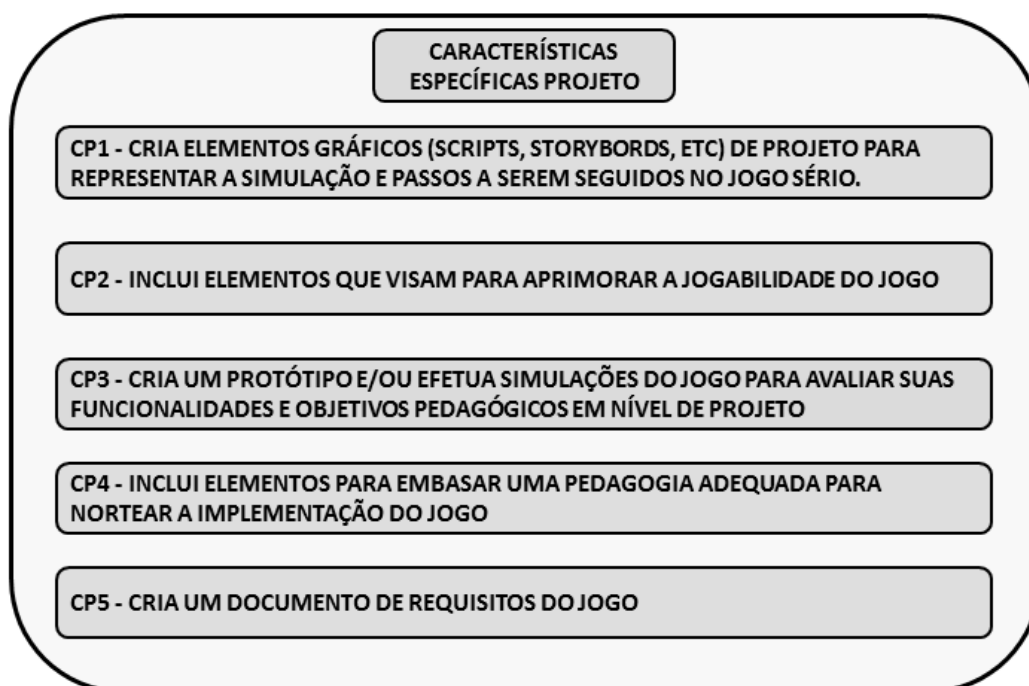


Figura 23 - Características específicas disciplina implementação

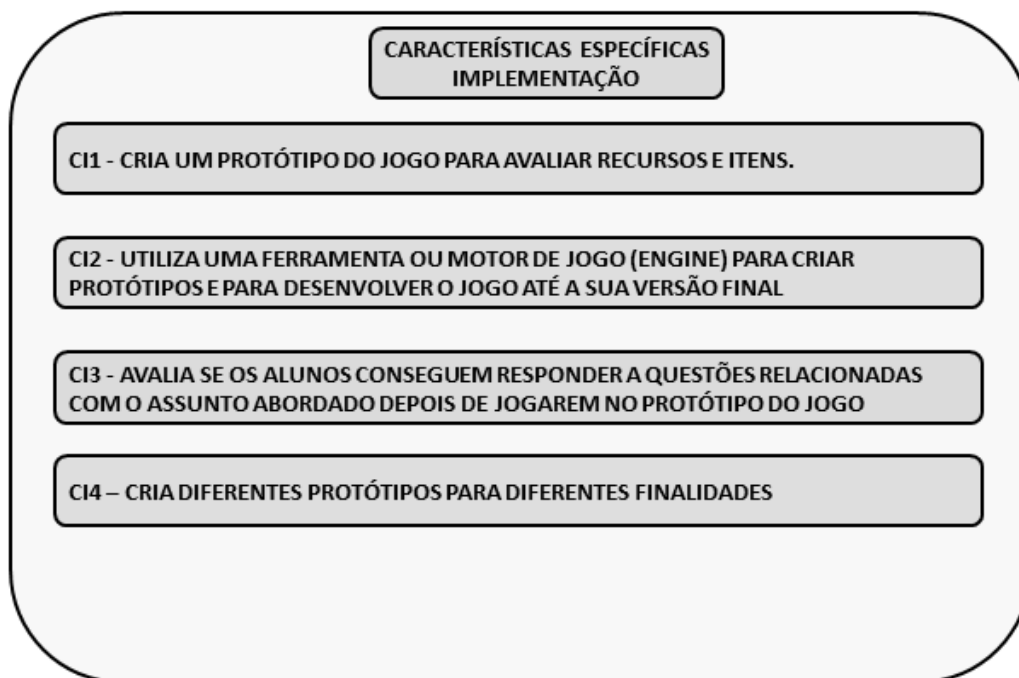
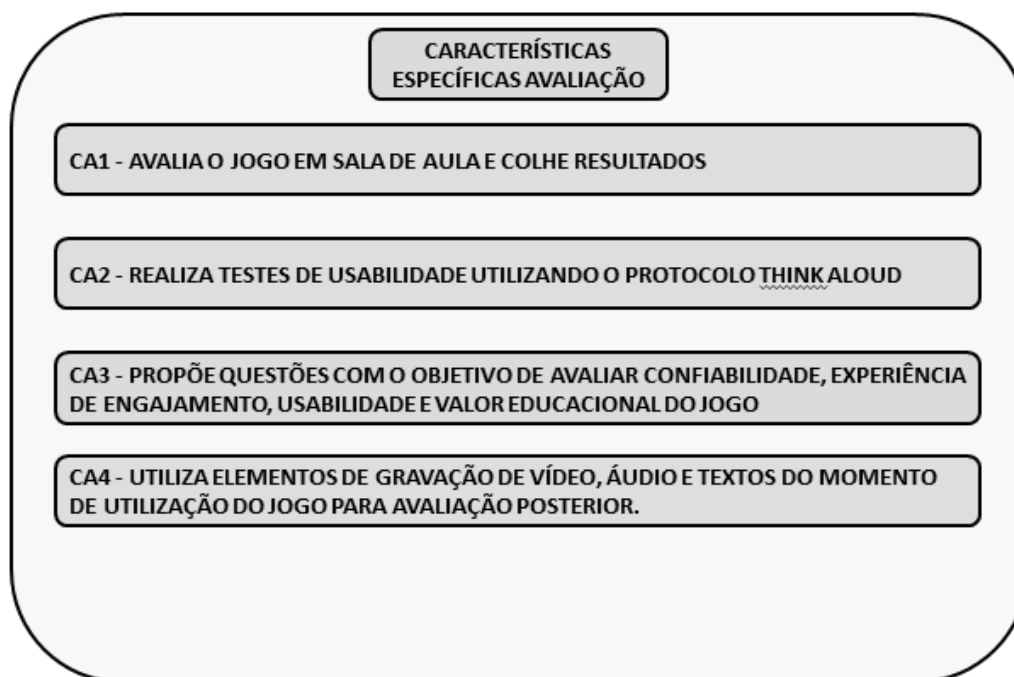


Figura 24 - Características específicas disciplina testes





Figura 25 - Características específicas disciplina avaliação



A característica geral(CG) será utilizada para avaliar a descrição de todas as atividades de cada disciplina presentes nas 17 metodologias fonte. Após esta análise um conjunto de atividades será selecionado e servirão de base para etapas posteriores do processo.

Uma rápida análise das 17 metodologias utilizadas como referência identificou que algumas atividades das disciplinas que as compõem poderiam não estar descritas num nível de detalhe suficiente para que possam ser adequadamente compreendidas e executadas. Quando do início de uma análise mais aprofundada das 17 metodologias fonte, essa característica geral(CG) terá como objetivo avaliar as descrições destas atividades e selecionar apenas as atividades que estão suficientemente detalhadas nos artigos das 17 metodologias fonte.

A seguir, as características específicas definidas são descritas, detalhadas e justificadas.

## **DISCIPLINA REQUISITOS**

### **Característica CR1 - Permite identificar o conceito principal do jogo sério**

#### **Justificativa**

O conceito de um jogo esta relacionado com “ideia do jogo, público, recursos, objetivo, finalidade do jogo, e condições de vitória” (ROCHA; ARAUJO, 2013). A identificação clara do conceito principal do jogo permitirá efetuar direcionamentos mais nítidos acerca das tarefas pertencentes ao levantamento de requisitos do jogo a ser desenvolvido.

### **Característica CR2: Inclui tarefas para identificar os objetivos pedagógicos do jogo com participação, inclusive, de professores e alunos.**

#### **Justificativa**

Como se trata de um jogo sério de escopo educacional é fundamental a identificação dos seus objetivos pedagógicos. Objetivos pedagógicos são metas que se pretende atingir quando da elaboração de aulas. No contexto de jogos estas metas pretendem ser atingidas com utilização de elementos contidos no jogo. Estes objetivos pedagógicos “podem ser gerais e envolverem competências que se constroem ao longo de muito tempo, ou específicos e envolverem habilidades e aprendizagens desenvolvidas em prazos mais curtos.” (EDUCARED PRONIÑO, 2008). Assim é necessário deixar claro que conteúdo se deseja ensinar, desenvolver e avaliar com o jogo sério. A participação efetiva de professores(especialistas do domínio), alunos e especialistas de desenvolvimento de jogos na identificação destes objetivos educacionais, poderão aumentar as chances de alcance das finalidades do jogo. Além disso, um entendimento mais profundo do papel que o jogo deverá desempenhar pode permitir delinear melhor seu escopo e tornar o desenvolvimento mais objetivo.

**Característica CR3: Suporta a comunicação entre especialistas do domínio e desenvolvedores de jogos****Justificativa**

Uma comunicação efetiva entre os educadores e desenvolvedores de jogos pode evitar a falta de integração e entendimento relacionado com as características e tarefas pertinentes com o desenvolvimento do jogo sério. Os educadores são os especialistas do domínio, ou seja, os que entendem de forma ampla as características e os problemas que o jogo pretende fornecer soluções. O domínio conforme declara a (WIKIPEDIA, 2015) é o conjunto de requisitos, funcionalidades e terminologia comuns que “descrevem uma família de problemas para os quais uma determinada aplicação pretende dar solução”.

**Característica CR4: Identifica as expectativas das partes interessadas****Justificativa**

Uma parte interessada é “um indivíduo, grupo ou organização que pode afetar, ser afetada ou sentir-se afetada por uma decisão, atividade ou resultado de um projeto” (PMBOK, 2013, p.30). A identificação clara das expectativas das partes interessadas pode aumentar a probabilidade de sucesso do jogo. Esta ação pode prevenir futuros problemas de falta de entendimento dos requisitos do projeto e, portanto torna-se uma característica relevante para uma atividade de levantamento de requisitos do jogo sério.

**Característica CR5: Cria um documento de requisitos do jogo****Justificativa**

Um documento que explicita os requisitos é uma referência fundamental para propiciar que eles sejam incorporados ao jogo e para validar sua implementação. Esse documento deve ser revisado pelas partes interessadas antes da implementação do jogo e deve-se garantir que todos concordem que essa relação de requisitos é a mais adequada. A equipe de desenvolvimento do jogo deve implementar esses requisitos e não será cobrada por nenhum outro.

## **DISCIPLINA PROJETO**

**Característica CP1 - Cria elementos gráficos (scripts, storyboards, etc) de projeto para representar a simulação e passos a serem seguidos no jogo sério.**

### **Justificativa**

Através da criação de elementos gráficos do jogo será possível uma análise em nível de projeto de sua jogabilidade também conhecida do termo em inglês *gameplay*. Após uma análise de diversas definições de variados autores e dicionários acerca do conceito de *gameplay* Vannucci e Prado (2009) indicaram que o *gameplay* “emerge das interações do jogador com o ambiente, a partir da manipulação das regras e mecânicas do jogo, pela criação de estratégias e táticas que tornam interessante e divertida a experiência de jogar” (VANNUCCI;PRADO, 2009, p.138).

Desta forma serão avaliados e simulados, pelos especialistas de domínio juntamente com os designers de jogos, os elementos do jogo na busca de possíveis correções e ajustes necessários. Saliente-se ainda que estes elementos servirão como forma de comunicação entre as partes interessadas no jogo. Além disso, cria-se a possibilidade de confirmar se o jogo sério está convergindo para os objetivos estabelecidos ao mesmo. Estes elementos pouco refinados serão os primeiros protótipos com finalidade específica de testar de forma ainda rudimentar a jogabilidade e demais itens do jogo sério.

**Característica CP2 - Inclui elementos que visam aprimorar a jogabilidade do jogo**

### **Justificativa**

Apesar de um jogo sério não focar prioritariamente o entretenimento, incluir tarefas que convirjam para esta meta poderá aprimorar a experiência do jogador durante a execução do jogo sério. Caso o jogador execute um jogo que possua uma boa interação e fluxo de acontecimentos, as chances de sucesso e alcance aos objetivos podem aumentar.

**Característica CP3 - Cria um protótipo e/ou efetua simulações do jogo para avaliar suas funcionalidades e objetivos pedagógicos em nível de projeto.**

### **Justificativa**

A criação de protótipos em diversas fases do processo de desenvolvimento do jogo sério, servirá para efetuar uma análise do jogo a ser desenvolvido e se este poderá atender a seus propósitos. Através de um desenvolvimento iterativo e incremental diversos protótipos serão produzidos em diferentes níveis de abstração e detalhamento.

Como destaca Schuytema (2008, p.24). “Um componente fundamental do trabalho do designer é criar o protótipo de uma experiência de jogo. Um protótipo é uma aproximação funcional de como o produto final pode ser”. Este protótipo servirá para testar a dinâmica e jogadas do jogo. “Para um designer de games, criar protótipos significa fazer a parte funcional do game de modo a testar a dinâmica das jogadas. Existem muitas maneiras de se criar um protótipo de um pedacinho do gameplay.” Schuytema (2008, p.24). O objetivo de construir um protótipo nesta fase do projeto é testar elementos propostos para o jogo sério antes do início da implementação e desta forma permitir a obtenção da aprovação do projeto pelas partes interessadas. Com isso, já será possível identificar se os objetivos pedagógicos que foram listados até esta fase, estão sendo obtidos e também possíveis erros e ajustes a serem realizados.

**Característica CP4 - Inclui elementos para permitir a criação de uma pedagogia adequada que irá nortear a implementação do jogo.**

#### **Justificativa**

Para aproveitar o conhecimento de especialistas de domínio é importante que o projeto de um jogo sério educacional possua elementos da área pedagógica. Estes elementos poderão direcionar o desenvolvimento do jogo alinhado com os objetivos educacionais estabelecidos.

**Característica CP5 - Cria o documento de design do jogo.**

#### **Justificativa**

Schuytema (2008) indica que “o documento de design do game é o coração e a alma de todos os documentos que giram em torno de um game em desenvolvimento”. Ele faz uma analogia deste documento relatando que é como se o documento de design fosse o documento de planta baixa do jogo. O objetivo de um documento de design em relação a um jogo, “é ilustrar como se deve jogá-lo e

apresentar uma descrição abrangente de todos os aspectos, para que a equipe de desenvolvimento possa de fato, criar o game.” (SCHUYTEMA, 2008, p.100).

Esta característica CP5 assegura que estará presente na metodologia customizada a ser concebida, uma atividade em que declara que será criado um documento de design do jogo sério. Este documento irá nortear todo o desenvolvimento do jogo. Através deste documento balizador pode-se assegurar, com uma boa documentação, que o planejamento foi aprovado por todos os envolvidos no processo. Assim haverá uma visão nítida do conceito do jogo e, portanto podem-se evitar desvios de rotas garantindo que a execução do projeto corresponda ao trabalho esperado.

Uma sugestão de sumário de um documento de design do game é indicado a seguir na tabela 4, por Schuytema (2008, p.101).

Tabela 6 - Sumário de um documento de design

I. Visão geral essencial a. Resumo b. Aspectos fundamentais c. Golden nuggets	IV. Conflitos e soluções
II. Contexto do game a. A história do game b. Eventos Anteriores c. Principais Jogadores	V. Inteligência artificial
III. Objetos essenciais do game a. Personagens b. Armas c. Estruturas d. Objetos	VI Fluxo do game
	VII Controles
	VIII Variações do jogo
	IX Definições
	X Referências

Fonte: Schuytema (2008, p.101).

## DISCIPLINA IMPLEMENTAÇÃO

### Característica C11 - Cria um protótipo do jogo para avaliar recursos e itens.

#### Justificativa

A criação de um protótipo permite tornar concreta e apresentar na prática as ideias e elementos previstos para o jogo. Além disso, a experiência do jogador poderá ser avaliada e os itens do jogo ganham forma permitindo que as partes interessadas possam visualizar de forma nítida a jogabilidade do jogo sério e proporem ajustes a um baixo custo de alterações. Será possível também realizar

testes de usabilidade e obter o feedback dos usuários e projetistas facilitando portanto a comunicação entre os integrantes das equipes.

**Característica CI2 - Utiliza uma ferramenta ou motor de jogo (engine) para criar protótipos e para desenvolver o jogo até a sua versão final**

**Justificativa**

A criação de protótipos é parte importante para avaliar possíveis erros e ajustes a serem realizados no jogo sério. A utilização de alguma ferramenta ou motor de jogo (engine) poderá aumentar a produtividade, rapidez e realismo das avaliações a serem feitas. A premissa é que, ao ser utilizado um recurso mais próximo do jogo final para construir os protótipos e realizar as simulações, será fornecida aos especialistas de domínio e designers uma visão mais ampla do jogo a ser desenvolvido através de um protótipo navegável. Além disso, o motor de jogo(engine) será utilizado para desenvolver todo o jogo até sua versão final.

**Característica CI3 - Avalia se os alunos conseguem responder a questões relacionadas com o assunto abordado depois de jogarem no protótipo do jogo**

**Justificativa**

Considerou-se, ainda, ser importante uma análise sobre o conhecimento adquirido pelo aluno ao jogar o jogo. Através de um protótipo pode-se criar uma primeira versão (preliminar) do jogo e inserir algum assunto com o propósito de avaliar a aquisição deste conhecimento. Com estes resultados os especialistas poderão avaliar se é provável que o jogo atinja os objetivos estabelecidos. Da mesma forma algumas estratégias pedagógicas do jogo poderão ser ajustadas ou ratificadas.

**CI4 - Cria diferentes protótipos para diferentes finalidades**

**Justificativa**

Para que sejam testados diferentes recursos em diferentes momentos do jogo faz-se necessário a construções de diferentes tipos de protótipos. Inicialmente protótipos de baixa definição são construídos. Depois protótipos em nível intermediário permitem outro tipo de visão. Depois um protótipo em alta definição e digital poderá fornecer uma visão ampla e realista do jogo.

## **DISCIPLINA TESTES**

### **Característica CT1- Simula o jogo para efetuar testes**

#### **Justificativa**

A simulação começa a testar a primeira versão jogável do jogo. Esta etapa irá permitir obter um feedback de aprovação sobre o leiaute, o roteiro, os sons e outros elementos do jogo. Além disso, poderão ser identificados bugs e providenciado a correção destes erros.

### **Característica CT2- Cria protótipos mock-up (ou seja, simulações com recursos de arte provisórios) para efetuar os testes**

#### **Justificativa**

No início a utilização de protótipos mais simples ajudam na rapidez do processo de prototipação a um baixo custo. Estes recursos de arte provisórios ajudarão nos primeiros testes mesmo sem ter sido construído qualquer tipo de elemento computacional do jogo.

### **Característica CT3- Cria testes com membros do público alvo para assegurar que o jogo é uma tradução eficaz das metas específicas de aprendizagem**

#### **Justificativa**

A condição de estar sendo desenvolvido um jogo sério de cunho educacional já justificaria a presença desta característica. Contudo é importante estabelecer ferramentas efetivas para verificar se as metas de aprendizagem estão sendo alcançadas no jogo sério.

### **Característica CT4- Realiza testes passo a passo para verificar os requisitos do jogo**

#### **Justificativa**

Uma importante ação no processo de desenvolvimento é aumentar as chances de cumprimento aos requisitos estabelecidos para o jogo. Assim um processo que possua atividades que verificam passa a passo os requisitos do jogo é de grande relevância. Alguns problemas que podem estar ocorrendo poderão ser ignorados caso estes testes não sejam realizados.



**Característica CT5- Realiza beta-teste em busca de erros técnicos e maiores falhas de usabilidade****Justificativa**

Durante a evolução do desenvolvimento do jogo e posteriormente a realização de testes anteriores é fundamental a realização de testes com os jogadores do público alvo do jogo. Estes testes verificarão as funcionalidades do produto, a jogabilidade e os seus elementos pedagógicos.

**DISCIPLINA AVALIAÇÃO****Característica CA1- Avalia o jogo em sala de aula e colhe resultados****Justificativa**

O processo de avaliação em ambiente aonde se localiza o público alvo é tarefa crucial no processo. Analisar e efetuar testes in loco permitirá colher informações importantes e validar a jogabilidade e objetivos do jogo sério educacional.

**Característica CA2 - Realizar testes de usabilidade utilizando o método think aloud****Justificativa**

Segundo Villanueva (2004 apud RENZI; FREITAS, 2010, p.2) , a técnica think-aloud protocol ocorre em um ambiente específico em que um moderador observa usuários efetuando ações e atividades específicas que devem ser descritas em voz alta. O moderador então grava as ações do participante e registra através de gravação de voz, filmagem e anotações escritas. Através da aplicação deste método poderão ser colhidas várias informações sobre o jogo no momento em que ele é jogado. O feedback realizado pelos alunos e professores será fundamental para consolidar os objetivos pedagógicos e de engenharia de software do jogo sério.

**Característica CA3 - Propõe questões com o objetivo de avaliar confiabilidade, experiência de engajamento, usabilidade e valor educacional do jogo****Justificativa**

Ao serem criadas questões específicas que ajudem na obtenção de respostas acerca da usabilidade, experiência do jogador e valor educacional do jogo, o cumprimento dos objetivos estabelecidos ao jogo sério estarão sendo mais uma vez avaliados. Estas questões serão importantes ferramentas para obter respostas que permitirão esta importante avaliação.

#### **Característica CA4 - Utiliza elementos de gravação de vídeo, áudio e textos do momento de utilização do jogo.**

##### **Justificativa**

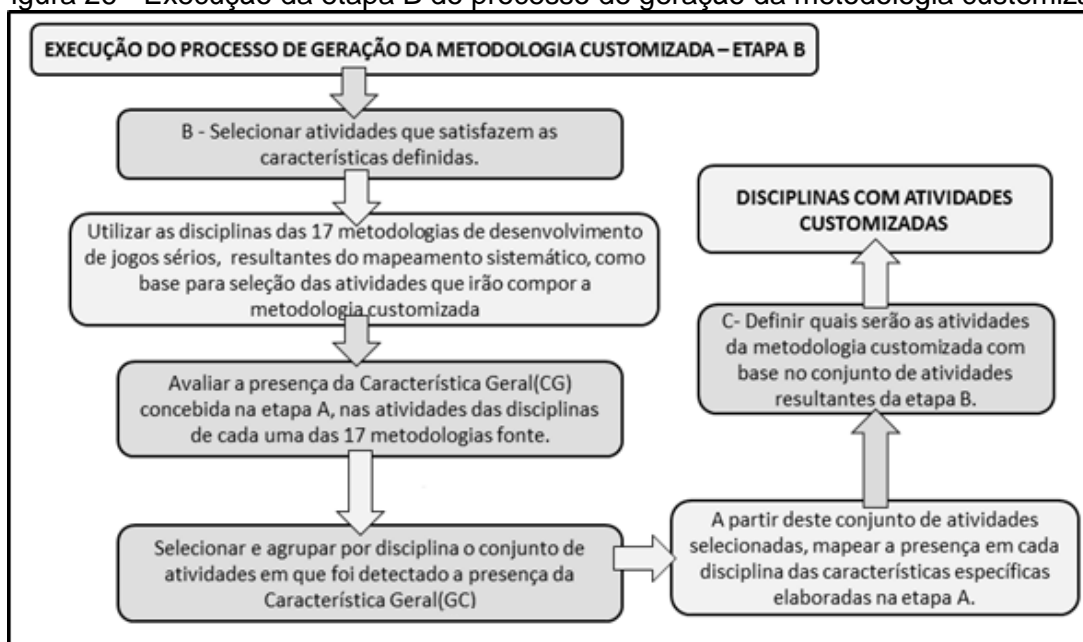
O teste beta será o momento em que os alunos e professores poderão vivenciar todas as funcionalidades propostas e desenvolvidas para o jogo sério educacional. Algumas vezes no momento de coleta de informações os observadores podem perder algum momento importante e deixar de registrar fatos que culminariam em ajustes importantes no jogo. Por isso ter um registro perene em vídeo, áudio ou texto do evento de teste beta irá ajudar a revisar o teste e encontrar novos itens de avaliação do jogo.

Após a execução da etapa A, será efetuada a execução da etapa B. A descrição deste processo está descrito a seguir.

#### **B. Selecionar atividades que satisfazem as características definidas.**

O objetivo desta etapa é selecionar, dentre as atividades que compõem as disciplinas das 17 metodologias de referência, aquelas que tenham presentes as características geral e específicas definidas na etapa A. Inicialmente serão avaliadas a presença nas atividades da característica geral(CG) e neste caso um conjunto de atividades serão selecionados. A partir deste conjunto será iniciada posteriormente a verificação da presença das características específicas também definidas na etapa A. Este processo está descrito na figura 26 a seguir.

Figura 26 - Execução da etapa B do processo de geração da metodologia customizada



A avaliação da descrição das atividades através da característica geral(CG) será realizada para cada disciplina e está descrita em seguida.

### 3.4.1 Disciplina Requisitos

O primeiro conjunto de atividades a serem verificados com o objetivo de avaliar a presença da característica geral(CG) é o pertencente a disciplina requisitos. O resultado final obtido após esta avaliação está presente na tabela 5 a seguir. A presença da característica geral é destacada através do sombreado das células nas tabelas a seguir.

Tabela 7 - Verificação da presença da CG na disciplina requisitos

<b>DISCIPLINA REQUISITOS (REQUIREMENTS)</b>		
<b>ARTIGO</b>	<b>ATIVIDADE</b>	<b>PRESENÇA DA CARACTERÍSTICA GERAL (CG)</b>
<b>MARCOS e ZAGALO - 2011</b>	Starting Concept	
<b>BORRO-ESCRIBANO et al. - 2014</b>	Specification	
<b>LAZAROU - 2011</b>	Identify and define Activity Systems	
	Identify personal requirements	
	Solve contradictions	
<b>RUPPEL e SCHATZ – 2011</b>	Model of Reality	
	Value Proposal	
	Game Concept	
	Use-Cases	
	Planning	
	Requirements	
<b>VAN DER ZEE et al. - 2012</b>	Setting objectives and parameters	
<b>AMANATIADOU e VAN DER WEERD - 2009</b>	Concept	
<b>ISMAILOVIC et al - 2010</b>	Design an Idea	
<b>ROCHA et al. – 2012</b>	Plan	
	Analysis	
<b>SHIRATUDDIN - 2011</b>	Concept	
<b>TORRENTE et al. - 2014</b>	Analysis	
<b>ASUNCION et al. - 2011</b>	Pre-Production	
<b>BROWN et al - 2009</b>	Design	
<b>COENEN et al - 2013</b>	Problem Formulation	
<b>DE ABREU et al. - 2011</b>	Study of Process	
<b>EAGLE e BARNES - 2012</b>	Identify users and needs	
	Observation	
<b>RANKIN et al - 2008</b>	Observational Studies	
	Conceptualization	
<b>LUKOSCH et al – 2012</b>	Analysis	

De forma análoga o mesmo processo foi aplicado as demais disciplinas, conforme exposto a seguir.

### 3.4.2 Disciplina Projeto

Tabela 8 - Verificação da presença da CG na disciplina projeto

DISCIPLINA PROJETO (DESIGN)		
ARTIGO	ATIVIDADE	PRESENÇA DA CARACTERÍSTICA GERAL (CG)
MARCOS e ZAGALO - 2011	Concept Design	
	Narrative Design	
	Experience Design	
	Aesthetic Musing	
	Game Design	
BORRO-ESCRIBANO et al. - 2014	<b>Game Design</b>	
LAZAROU - 2011	<b>Development</b>	
	Pedagogy Design	
RUPPEL e SCHATZ - 2011	<b>Analysis</b>	
	<b>Design</b>	
VAN DER ZEE et al. - 2012	Model development	
	Decisions about	
AMANATIADOU e VAN DER WEERD - 2009	Pre-production	
ISMAILOVIC et al - 2010	Design an Idea	
ROCHA et al. - 2012	Design	
SHIRATUDDIN - 2011	<b>Concept</b>	
	<b>Pre-Production</b>	
TORRENTE et al. - 2014	<b>Game Design</b>	
ASUNCION et al. - 2011	Pre-Production	
BROWN et al - 2009	Design	
COENEN et al - 2013	Building	
	Intervention	
DE ABREU et al. - 2011	Study of Technology	
	Modelling	
EAGLE e BARNES - 2012	<b>Design-Prototype</b>	
RANKIN et al - 2008	Conceptualization	
LUKOSCH et al - 2012	Synthesis	

### 3.4.3 Disciplina Implementação

Tabela 9 - Tabela verificação da presença da CG na disciplina implementação.

<b>DISCIPLINA IMPLEMENTAÇÃO (IMPLEMENTATION)</b>		
<b>ARTIGO</b>	<b>ATIVIDADE</b>	<b>PRESENÇA DA CARACTERÍSTICA GERAL (CG)</b>
<b>MARCOS e ZAGALO - 2011</b>	Game Implementation	
<b>BORRO-ESCRIBANO et al. - 2014</b>	Simulation Development	
<b>LAZAROU - 2011</b>	Development	
<b>RUPPEL e SCHATZ - 2011</b>	Implementation	
<b>VAN DER ZEE et al. - 2012</b>	Construction and Modification	
<b>AMANATIADOU e VAN DER WEERD - 2009</b>	Production	
<b>ISMAILOVIC et al - 2010</b>	Generate a Prototype	
<b>ROCHA et al. - 2012</b>	Implementation	
<b>SHIRATUDDIN - 2011</b>	Prototype	
	Production	
<b>TORRENTE et al. - 2014</b>	Implementation	
<b>ASUNCION et al. - 2011</b>	Production	
<b>BROWN et al - 2009</b>	Development	
<b>COENEN et al - 2013</b>	Building	
	Intervention	
<b>DE ABREU et al. - 2011</b>	Development	
<b>EAGLE e BARNES - 2012</b>	Translate	
<b>RANKIN et al - 2008</b>	Prototyping	
<b>LUKOSCH et al - 2012</b>	Synthesis	

### 3.4.4 Disciplina Testes

Tabela 10 – Verificação da presença da CG na disciplina testes

<b>DISCIPLINA TESTES (TEST)</b>		
<b>ARTIGO</b>	<b>ATIVIDADE</b>	<b>PRESENÇA DA CARACTERÍSTICA GERAL (CG)</b>
<b>MARCOS e ZAGALO - 2011</b>	Game Deployment Planning	
<b>BORRO-ESCRIBANO et al. - 2014</b>	Simulation Development	
<b>LAZAROU - 2011</b>	Development	
<b>RUPPEL e SCHATZ - 2011</b>	Testing	
<b>VAN DER ZEE et al. - 2012</b>	Construction and Modification	
<b>AMANATIADOU e VAN DER WEERD - 2009</b>	Post-production	
<b>ISMAILOVIC et al - 2010</b>	Generate a Prototype	
<b>ROCHA et al. - 2012</b>	Evaluation Configuration	
	Integration and Test	
<b>SHIRATUDDIN - 2011</b>	Prototype	
	Production	
<b>(TORRENTE et al., 2014)</b>	Implementation	
<b>(ASUNCION et al., 2011)</b>	Production	
<b>(BROWN et al., 2009)</b>	Development	
<b>(COENEN et al., 2013)</b>	Building	
	Intervention	
<b>(DE ABREU et al., 2011)</b>	Development	
<b>(EAGLE e BARNES, 2012)</b>	Translate	
<b>(RANKIN et al., 2008)</b>	Prototyping	
<b>(LUKOSCH et al., 2012)</b>	Simulation	

### 3.4.5 Disciplina Avaliação

Tabela 11 - Verificação da presença da CG na disciplina avaliação

DISCIPLINA AVALIAÇÃO (EVALUATION)		
ARTIGO	ATIVIDADE	PRESENÇA DA CARACTERÍSTICA GERAL (CG)
(MARCOS; ZAGALO, 2011)	Game Deployment Planning	
(BORRO; ESCRIBANO et al., 2014)	Quality Assurance	
(LAZAROU, 2011)	Usability Testing	
	Implementation and Data Collection	
	Evaluation	
(RUPPEL; SCHATZ, 2011)	Evaluation	
(VAN DER ZEE et al., 2012)	Preparation for use	
(AMANATIADOU; VAN DER WEERD, 2009)	Post-production	
(ISMAILOVIC et al., 2010)	Evaluate with Users	
(ROCHA et al., 2012)	Execution	
	Evaluation	
SHIRATUDDIN - 2011	Post-Production	
(TORRENTE et al., 2014)	Quality Assurance	
(ASUNCION et al., 2011)	Final Test	
(BROWN et al., 2009)	Release	
(COENEN et al., 2013)	Evaluation	
	Data Gathering	
	Reflection and Learning	
(DE ABREU et al., 2011)	Assessment	
	Evaluation	
(EAGLE e BARNES, 2012)	Evaluate	
(RANKIN et al., 2008)	Usability Playtesting	
(LUKOSCH et al., 2012)	Evaluation	

Uma vez selecionadas as atividades que contemplam a característica geral(CG) dar-se início a avaliação da presença nestas atividades das características específicas definidas em na etapa A. O passo seguinte, portanto, é mapear a presença ou ausência nas atividades destas características específicas. Foi realizado este estudo que culminou no seguinte panorama:



### 3.4.6 Disciplina Requisitos

Tabela 12 - Verificação da presença características específicas de requisitos

DISCIPLINA REQUISITOS (REQUIREMENTS)						
ARTIGO	ATIVIDADE	PRESENÇA DA CARACTERÍSTICA				
		CR1	CR2	CR3	CR4	CR5
BORRO-ESCRIBANO et al. - 2014	Specification					
LAZAROU - 2011	Identify personal requirements					
SHIRATUDDIN - 2011	Concept					
TORRENTE et al. - 2014	Analysis					
EAGLE e BARNES - 2012	<i>Identify users and needs</i>					
	<i>Observation</i>					
DISCIPLINA CUSTOMIZADA	REQUISITOS (REQUIREMENTS)					

Observa-se na tabela 5 que nenhuma das atividades contemplam todas as características. Desta forma uma nova disciplina de requisitos será customizada inserindo elementos de cada uma das atividades avaliadas acima. Assim a nova disciplina irá contemplar todas as características específicas: CR1, CR2, CR3, CR4 e CR5. De forma análoga este processo será executado em todas as demais disciplinas, como se segue.

### 3.4.7 Disciplina Projeto

Tabela 13 - Verificação da presença características específicas de projeto

DISCIPLINA PROJETO(DSIGN)						
ARTIGO	ATIVIDADE	PRESENÇA DA CARACTERÍSTICA				
		CP1	CP2	CP3	CP4	CP5
BORRO-ESCRIBANO et al. - 2014	Game Design					
LAZAROU - 2011	Development					
ROCHA et al. - 2012	Design					
SHIRATUDDIN - 2011	Concept					
	Pre-Production					
TORRENTE et al. - 2014	Game Design					
EAGLE e BARNES - 2012	Design-Prototype					
DISCIPLINA CUSTOMIZADA	PROJETO(DSIGN)					

Mais uma vez observa-se na tabela que nenhuma das atividades contempla todas as características. Assim a nova disciplina de projeto irá aglutinar todas as características específicas: CP1, CP2, CP3, CP4 e CP5.

### 3.4.8 Disciplina Implementação

Tabela 14 - Verificação da presença características específicas de implementação

Disciplina Implementação (Implementation)		PRESENÇA DA CARACTERÍSTICA			
ARTIGO	ATIVIDADE	CI1	CI2	CI3	CI4
BORRO-ESCRIBANO et al. - 2014	Simulation Development				
LAZAROU - 2011	Development				
ROCHA et al. - 2012	Implementation				
TORRENTE et al. 2014	Implementation				
EAGLE e BARNES - 2012	Translate				
DISCIPLINA CUSTOMIZADA	IMPLEMENTAÇÃO				

Nenhuma das atividades contemplam todos as características desejadas para compor uma atividade da disciplina de implementação. A nova disciplina irá contemplar todos os critérios estabelecidos para a construção do jogo sério educacional em relação a disciplina de implementação.

### 3.4.9 Disciplina Testes

Tabela 15 - Verificação da presença características específicas de testes

Disciplina Testes (Test)		CT1	CT2	CT3	CT4	CT5
(BORRO-ESCRIBANO et al. 2014)	Simulation Development					
EAGLE e BARNES - 2012	Translate					
ROCHA et al. - 2012	Evaluation Configuration Integration and Test					
TORRENTE et al. 2014	Implementation					
DISCIPLINA CUSTOMIZADA	TESTES					

Nenhuma das atividades contemplam todos as características desejadas para compor uma atividade da disciplina de testes. A nova disciplina irá contemplar todos os critérios estabelecidos para a construção do jogo sério educacional em relação a disciplina de testes.

### 3.4.10 Disciplina Avaliação

Tabela 16 - Verificação da presença características específicas de requisitos

Disciplina Avaliação (Evaluation)		CA1	CA2	CA3	CA4
<b>BORRO-ESCRIBANO et al. - 2014</b>	Quality Assurance				
<b>LAZAROU - 2011</b>	Usability Testing				
	Implementation and Data Collection				
<b>TORRENTE et al.</b>	Quality Assurance				
<b>DISCIPLINA CUSTOMIZADA</b>	<b>AVALIAÇÃO</b>				

A próxima etapa será a de definir quais serão as atividades que irão compor a metodologia com base na seleção efetuada na etapa B.

### C. Definir atividades da metodologia com base nas atividades selecionadas.

O próximo passo é executar a definição final de quais serão as atividades que irão compor cada disciplina da metodologia customizada. Esta etapa utiliza como base a etapa anterior que selecionou atividades que contemplam a característica geral e alguma ou algumas das características específicas definidas em B. As descrições de cada uma foram retiradas das atividades das metodologias fontes resultantes da etapa B. Portanto as atividades customizadas terão elementos que permitem que todas as características desejadas, geral e específicas, estejam presentes em cada uma das disciplinas da metodologia customizada.

A seguir o resultado do processo expõe uma lista destas atividades customizadas para cada disciplina da metodologia concebida. Além disso, figuras também irão mostrar cada uma destas atividades.

Inicialmente serão listadas as atividades que foram selecionadas dos artigos em relação à disciplina requisitos e posteriormente serão exibidas as atividades das demais disciplinas de projeto, implementação, testes e avaliação.

Figura 27 - Lista de atividades da disciplina requisitos da metodologia customizada

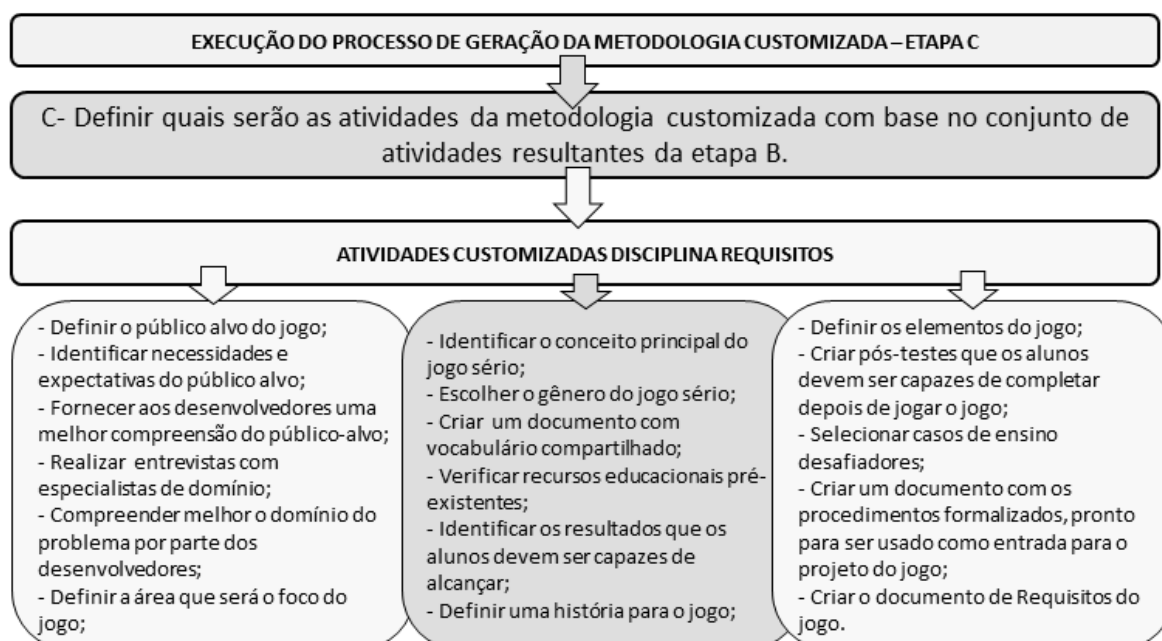


Figura 28 - Lista de atividades da disciplina projeto da metodologia customizada

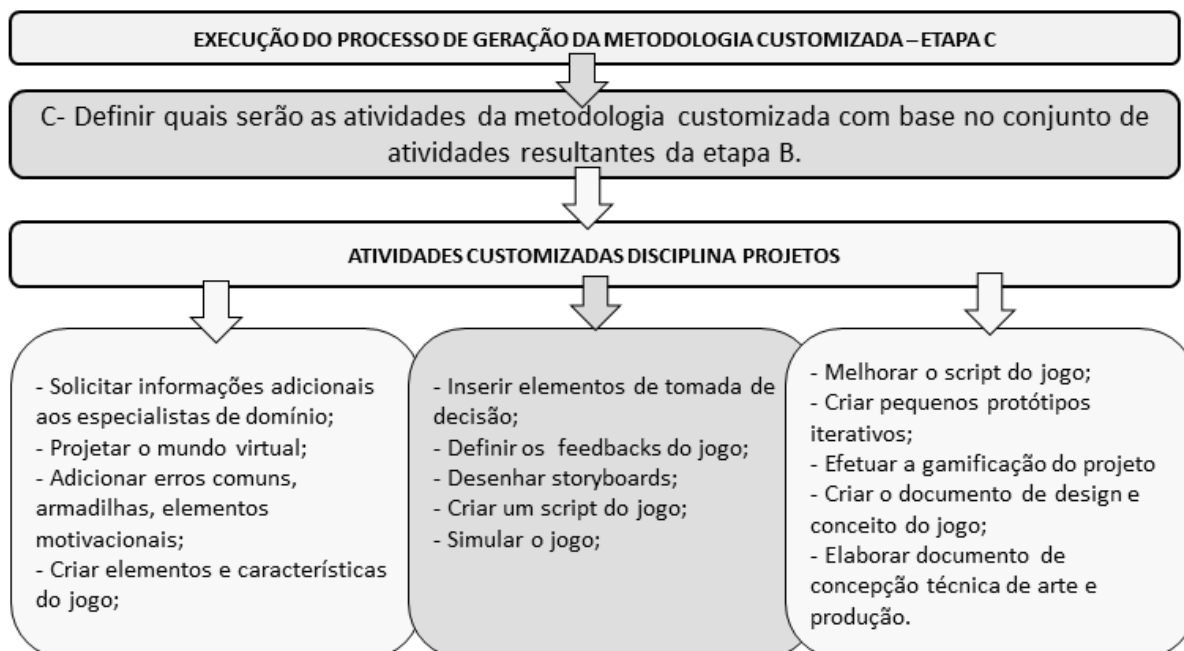


Figura 29 - Lista atividades da disciplina implementação da metodologia customizada

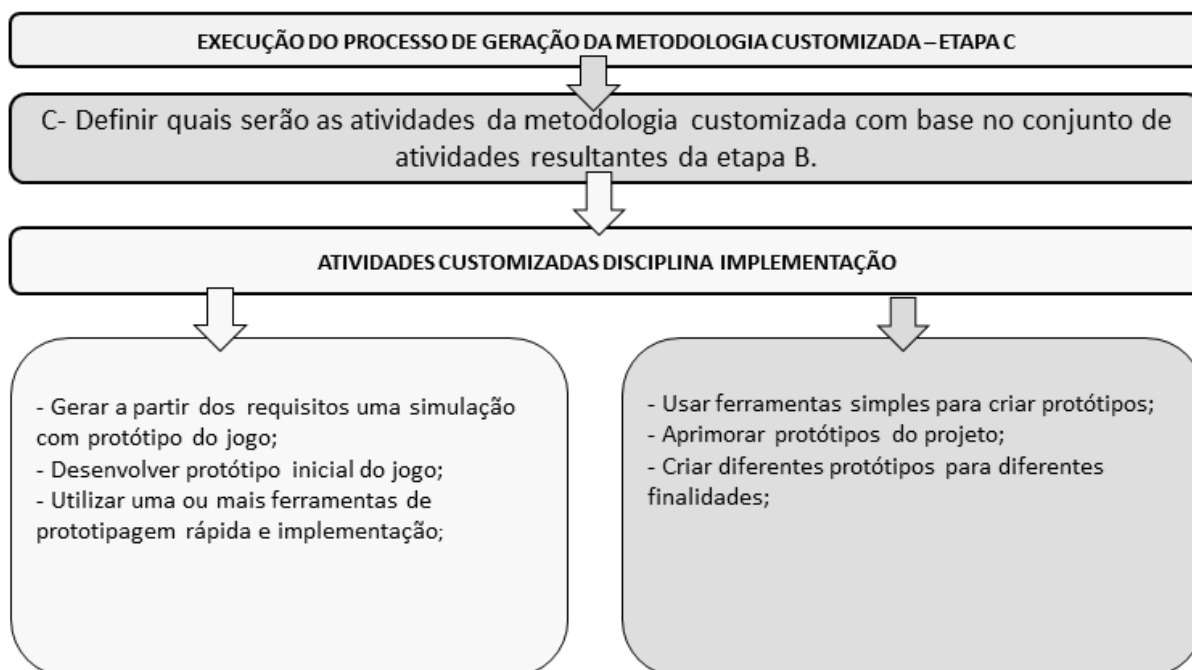


Figura 30 - Lista atividades da disciplina testes da metodologia customizada

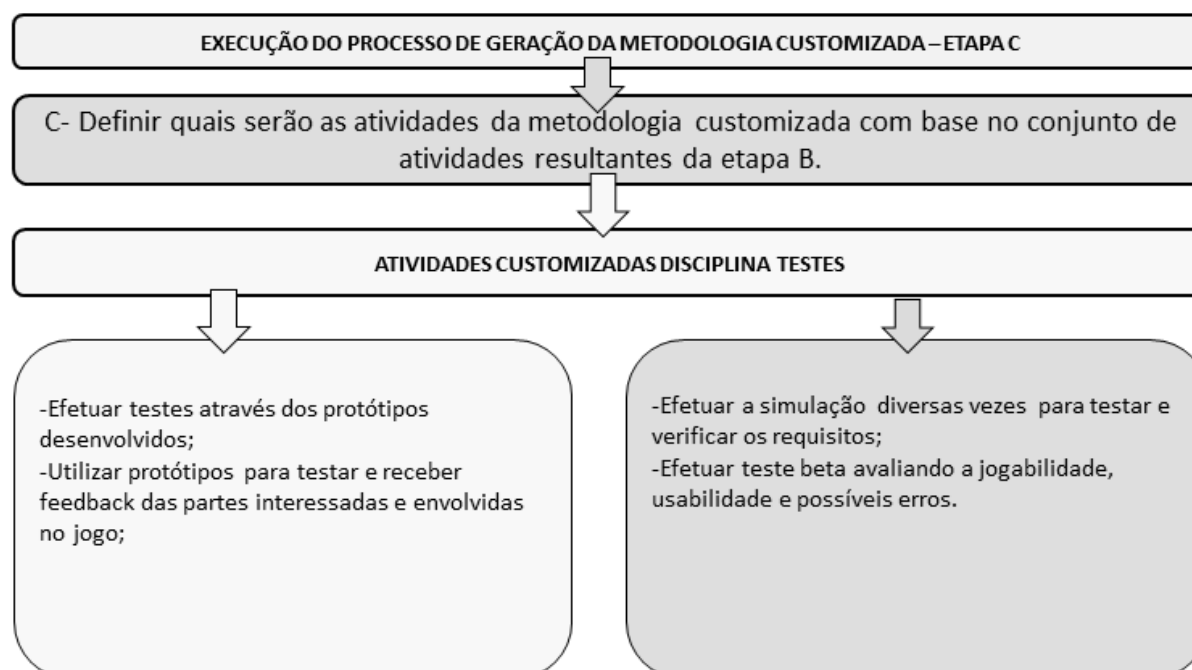
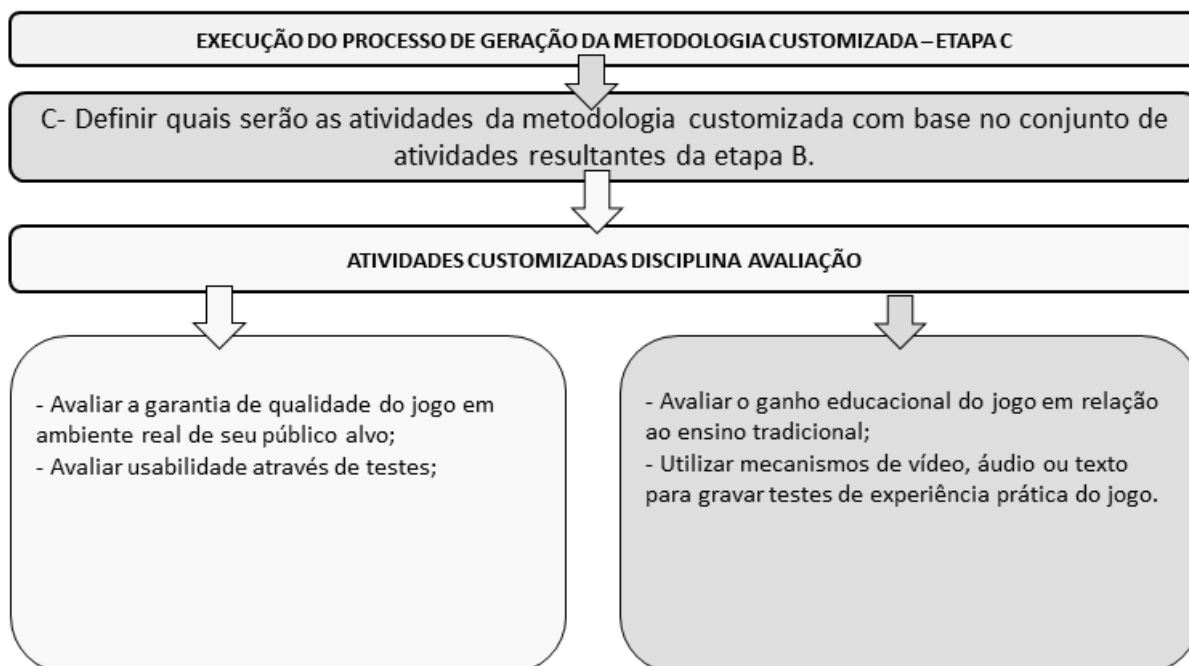


Figura 31 - Lista atividades da disciplina testes da metodologia customizada



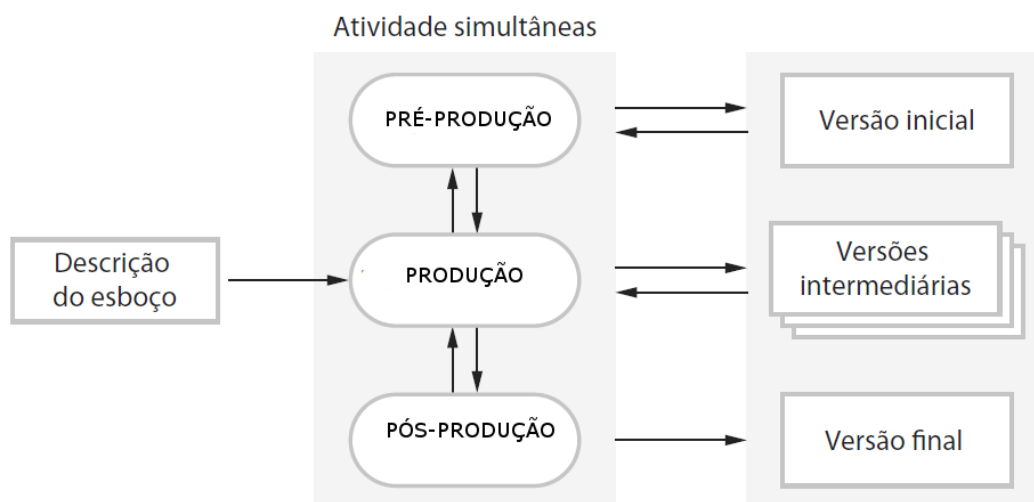
As atividades e suas respectivas disciplinas da metodologia customizada concebida no processo anterior é mostrada através da figura 32. A figura representa em linhas gerais um agrupamento de atividades que segue um ciclo de desenvolvimento incremental de um jogo que engloba as fases de pré-produção, produção e pós-produção. Estas fases são utilizadas por Schuytema (2008) e são amplamente utilizadas quando do desenvolvimento de jogos. As fases de pré-produção, produção e pós-produção foram utilizadas para agrupar o sequenciamento das atividades das disciplinas.

Figura 32 - Metodologia customizada resultante da execução do processo 3.4



Em relação ao desenvolvimento incremental ele é embasado na perspectiva de “desenvolver uma implementação inicial, expô-la aos comentários dos usuários e continuar por meio da criação de várias versões até que um sistema adequado seja desenvolvido.” (SOMMERVILLE, 2011, p.21). A metodologia concebida também irá fazer uso deste cenário e assim vários protótipos em diversos níveis de completude do jogo serão gerados. A figura 33 mostra a abordagem incremental que foi escolhida para guiar a metodologia de desenvolvimento do jogo sério produzida nas seções anteriores deste trabalho.

Figura 33 - Desenvolvimento Incremental do Jogo sério



Fonte: Adaptado de Sommerville (2011, p.22).

A próxima etapa será realizar a descrição completa de cada uma destas atividades contidas na figura anterior. As atividades e ferramentas utilizadas serão amplamente descritas. O detalhamento textual das atividades pertencentes a cada disciplina será exposto a seguir nas tabelas e são oriundos dos artigos fonte.



Tabela 17 - Descrição das atividades selecionadas para a disciplina requisitos

<b>ATIVIDADE CUSTOMIZADA - DISCIPLINA REQUISITOS</b>																		
<b>COD</b>	<b>ATIVIDADE SELECIONADA</b>	<b>DESCRIÇÃO</b>																
<b>CR1</b>	Identificar o conceito principal do jogo sério (SHIRATUDDIN, 2011)	<p>A fim de estabelecer uma abordagem pedagógica, o professor da disciplina deve efetuar um estudo sobre:</p> <ul style="list-style-type: none"> <li>• O conteúdo do jogo</li> <li>• Objetivos</li> <li>• Resultados esperados,</li> <li>• Como os alunos serão avaliados,</li> <li>• As estratégias de ensino</li> <li>• O sequenciamento do aprendizado</li> <li>• As tarefas e atividades que serão realizados no jogo e</li> <li>• As habilidades que os alunos irão adquirir.</li> </ul>																
	Escolher o gênero do jogo sério.	<p>A escolha de um gênero adequado deve ser selecionada para coincidir com as estratégias de ensino e aprendizagem estabelecidas. Os seguintes tipos de gêneros devem ser analisados e um escolhido:(SHIRATUDDIN, 2011, p.4)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Gênero</th> <th style="text-align: center;">Tipos de atividades e desafios</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Jogos de ação</td> <td style="text-align: center;">Desafios Físicos</td> </tr> <tr> <td style="text-align: center;">Jogos de estratégia em tempo real</td> <td style="text-align: center;">Estratégia, táticas e desafios logísticos</td> </tr> <tr> <td style="text-align: center;">Jogos de RPG</td> <td style="text-align: center;">Táticas, logística, experiências e desafios econômicos</td> </tr> <tr> <td style="text-align: center;">Simulações do mundo real (jogos esportivos e simulações de veículos simulações)</td> <td style="text-align: center;">Desafios físicos e táticos</td> </tr> <tr> <td style="text-align: center;">Construção e gestão de jogos</td> <td style="text-align: center;">Desafios econômicos e conceituais</td> </tr> <tr> <td style="text-align: center;">Jogos de Aventura</td> <td style="text-align: center;">Exploração e desafios de resolução de quebra-cabeças</td> </tr> <tr> <td style="text-align: center;">Jogos de Quebra Cabeças</td> <td style="text-align: center;">Lógica e desafios conceituais</td> </tr> </tbody> </table> <p>SUMMARY OF GAME GENRES AND THEIR RELATED ACTIVITIES AND CHALLENGES (SHIRATUDDIN, 2011, p.4)</p>	Gênero	Tipos de atividades e desafios	Jogos de ação	Desafios Físicos	Jogos de estratégia em tempo real	Estratégia, táticas e desafios logísticos	Jogos de RPG	Táticas, logística, experiências e desafios econômicos	Simulações do mundo real (jogos esportivos e simulações de veículos simulações)	Desafios físicos e táticos	Construção e gestão de jogos	Desafios econômicos e conceituais	Jogos de Aventura	Exploração e desafios de resolução de quebra-cabeças	Jogos de Quebra Cabeças	Lógica e desafios conceituais
	Gênero	Tipos de atividades e desafios																
	Jogos de ação	Desafios Físicos																
	Jogos de estratégia em tempo real	Estratégia, táticas e desafios logísticos																
	Jogos de RPG	Táticas, logística, experiências e desafios econômicos																
	Simulações do mundo real (jogos esportivos e simulações de veículos simulações)	Desafios físicos e táticos																
Construção e gestão de jogos	Desafios econômicos e conceituais																	
Jogos de Aventura	Exploração e desafios de resolução de quebra-cabeças																	
Jogos de Quebra Cabeças	Lógica e desafios conceituais																	
Definir os elementos do jogo	Indicar quais serão as atividades básicas que os alunos estarão fazendo durante o jogo. Se estarão coletando, perseguindo, combatendo, esquivando-se, construindo, etc. (SHIRATUDDIN, 2011)																	
Definir a história para o jogo	Definir se o jogo terá uma história e se esta história será básica ou complexa e se os alunos poderão influenciar a mesma. (SHIRATUDDIN, 2011)																	
Definir o público alvo do jogo	Deve ser decidido para quem o jogo sério é destinado. (SHIRATUDDIN, 2011)																	
<b>CR2</b>	Definir a área que será o foco do jogo	Deverão ser definidos os objetivos de aprendizagem do jogo em relação à área foco estabelecida. Conforme destaca (SHIRATUDDIN, 2011) existem diferentes áreas que um jogo sério pode ser utilizado para complementar o ensino convencional e cada área terá seus próprios objetivos de aprendizagem e o projeto do jogo deve considerá-los.																
	Verificar recursos educacionais pré-existentes	Verificar com educadores a existência de recursos compartilhados para os problemas e exercícios que atendam aos objetivos de aprendizagem importantes para área alvo do jogo sério. (EAGLE ; BARNES, 2012)																

<b>CR2</b>		
	Compreender melhor o domínio do problema por parte dos desenvolvedores.	Efetuar uma melhor compreensão do domínio do problema pelos desenvolvedores do jogo para definir um conteúdo que seja significativo e desafiador para o aprendizado dos alunos. Realizar entrevistas com estudantes e professores para que eles reflitam sobre a aprendizagem anterior, pesquisa bibliográfica, observação de aulas e métodos de ensino. Idealmente, os desenvolvedores devem se envolver em aulas individuais, observando o que funciona e o que não funciona com atenção especial para áreas onde está ocorrendo maior confusão no aprendizado dos alunos através da identificação de lacunas no conhecimento. Utilizar protocolo think-aloud onde os alunos pensem em voz alta sobre o seu processo de resolução de problemas. Depois efetuar uma observação da resolução de problemas efetuando uma identificação de lacunas no conhecimento, que irão fornecer informações sobre o projeto e recursos do jogo. (EAGLE; BARNES, 2012)
	Identificar os resultados que os alunos devem ser capazes de alcançar	Desenvolvedores e professores deverão escrever os objetivos de aprendizagem para as aulas, onde os resultados devem ser observáveis e mensuráveis. (EAGLE; BARNES, 2012)
	Criar pós-testes que os alunos devem ser capazes de completar depois de jogar o jogo.	Projetar pós-testes usando exercícios de livros didáticos, ou questões de prova. As perguntas devem tentar medir a profundidade do conhecimento do aluno. Recomenda-se o desenvolvimento de questões que consideram a dificuldade com base na Taxonomia de Objetivos Educacionais de Bloom. Após o pós-teste ser criado, o mesmo deve ser verificado consultando professores antes de ser utilizado. Este requisito também ajuda a garantir que as atividades no jogo são válidas, isto é, que o jogo ensina o que ele pretende. Os desenvolvedores devem procurar jogos, atribuições, e materiais de aprendizagem online que sejam eficazes para o material alvo. Métodos comuns para a resolução de problemas devem ser anotados para incorporação no projeto. (EAGLE; BARNES, 2012)
	Selecionar casos de ensino desafiadores	Selecionar casos de ensino para incluir no jogo sério que não sejam óbvios para desafiar o aluno de alguma forma especial. (BORRO-ESCRIBANO et al. 2014)
<b>CR3</b>	Criar um documento com vocabulário compartilhado	Criar um vocabulário compartilhado de termos e de conhecimentos relacionados com o projeto do jogo sério e que seja entendido por todas as partes interessadas. O objetivo é facilitar a comunicação entre os especialistas com diferentes formações. (BORRO-ESCRIBANO et al. 2014) O vocabulário pode ser representado usando uma notação formal ou uma representação visual (diagramas, fluxogramas, linguagem visual específica). Reuniões informais entre os participantes devem ser realizadas para revisar documentos. Na reunião, especialistas de domínio devem demonstrar como procedimentos são executados, mostrar vídeos ou quaisquer outros materiais que ilustram o domínio do jogo e fornecer outros materiais para os especialistas de jogos. Os especialistas de jogos devem mostrar um jogo relevante com exemplos para ajudar os especialistas de domínio entender a gama de recursos expressivos que podem ser usados para transformar procedimentos em um jogo. (TORRENTE et al. 2014)

	Realizar entrevistas com especialistas de domínio	Realizar entrevistas com os especialistas de domínio em que o jogo sério está inserido para melhorar e compreender o processo. (BORRO;ESCRIBANO et al. 2014)
<b>CR4</b>	Identificar necessidades e expectativas do público alvo.	Realizar entrevistas com professores e alunos para levantar as expectativas dos mesmos em relação ao jogo sério. (LAZAROU, 2011)
	Fornecer aos desenvolvedores uma melhor compreensão do público-alvo	Os desenvolvedores deverão observar instrutores e participar de sessões de tutoria para ver que metáforas e métodos são usados atualmente para ensinar, com especial atenção para as lacunas no conhecimento do público-alvo.. (EAGLE; BARNES, 2012)
<b>CR5</b>	Criar o documento de Requisitos do jogo	<p>Obter um documento de especificação detalhado que descreve as características do público alvo, o conhecimento relacionado com a área que o jogo sério deve contemplar e como ele vai ser representado, o ambiente e as configurações a que o jogo se destina e quaisquer outras informações que sejam necessárias para formalizar a área do jogo . (BORRO;ESCRIBANO et al. 2014)</p> <p>Descrever os aspectos técnicos do jogo:</p> <ul style="list-style-type: none"> <li>• Os requisitos de sistema, como o hardware / dispositivos que terão que ser utilizados para executar o jogo.</li> <li>• Eventuais restrições que podem se aplicar ao produto final, por exemplo, licenciamento, comercial, de código aberto, idade, etc.</li> <li>• Requisitos técnicos para os aspectos visuais do jogo.</li> <li>• O conteúdo de áudio. (SHIRATUDDIN, 2011)</li> </ul>
	Criar um documento com os procedimentos formalizados, pronto para ser usado como entrada para o projeto do jogo	A experiência dos especialistas do domínio é fundamental, pois eles sabem como os alunos normalmente interagem com o ambiente e podem apontar elementos que normalmente criam confusão entre os estudantes para serem usados como "pistas falsas" durante o jogo. Descrever de forma detalhada o ambiente, os elementos e os materiais utilizados, e os participantes e potenciais distratores. Capturar e formalizar a sequência de passos para os procedimentos. A sequência dos passos é descrita pela primeira vez em todos os detalhes. Esta sub-tarefa requer mais interação entre especialistas de jogo e especialistas de domínio, devido à dificuldade inerente de tornar o conhecimento tácito dos especialistas explícito. (TORRENTE et al., 2014)

Uma vez finalizada a montagem da disciplina customizada de requisitos e descritas quais serão as suas atividades, será executado o mesmo procedimento para a disciplina de projeto e demais disciplinas conforme exposto a seguir.

Tabela 18 - Descrição das atividades selecionadas para a disciplina projeto

ATIVIDADE CUSTOMIZADA - DISCIPLINA PROJETO		
CARACTERÍSTICA	TAREFAS	DESCRIÇÃO
<b>CP1</b>	Criar um script do jogo.	O script do jogo indica a transformação do documento de especificação de requisitos para os elementos de projeto do jogo O game designer deve compreender o processo, desenhar o storyboard da simulação do jogo e verificar se ele fornece o entretenimento adequado e valor educativo. O game designer deve saber como integrar as habilidades de design, conhecimentos e valores para permitir que o jogador possa construir um modo de pensar e de resolução de problemas que seja semelhante à abordagem dos peritos na área. (BORRO;ESCRIBANO et al. 2014)
	Desenhar storyboards.	A equipe tem que desenhar storyboards que identificam os elementos importantes, interfaces e interações que devem ser apresentadas ao usuário. Isso inclui cenário, painel de instrumentos, Objetos 2D e 3D, sons, interfaces interativas e da história, entre outros. Estes elementos devem fornecer ao usuário interface intuitiva que os permita desempenhar suas funções de acordo com os procedimentos executados no mundo real. (ROCHA et al. 2012)
	Melhorar script do jogo	Um script não é apenas uma sequência de passos; é necessário melhorá-lo para fornecer ao aluno uma experiência que promova imersão. Quando o jogo começa, os alunos precisam estar situados. A falta de contextualização adequada pode danificar seriamente a experiência de jogo, fazendo com que os alunos se sintam perdidos, e diminuindo o rendimento educacional de jogar o jogo. Primeiro, os alunos precisam ter uma compreensão clara dos principais objetivos do jogo (isto é, como ter sucesso) e as regras gerais (o que você deve e não deve fazer para ter sucesso). Durante o jogo, o aluno será confrontado com diferentes situações ou problemas que tem de ser resolvido. (TORRENTE et al. 2014)
	Projetar o mundo virtual	A descrição de alto nível do ambiente é traduzida em elementos do jogo. O modelo de ambiente de jogo pode ser produzido também com a ferramenta WEEV ou usando qualquer outra representação gráfica (TORRENTE et al. 2014)
<b>CP2</b>	Adicionar erros comuns, armadilhas e elementos motivacionais	Os erros comuns devem ser progressivamente incluídos como elementos do jogo. O objetivo é conduzir os alunos a atenção os aspectos que são de maior importância. Também é necessário adicionar armadilhas e entrelaçar elementos motivacionais para aumentar o engajamento. (BORRO;ESCRIBANO et al. 2014)
<b>CP2</b>	Solicitar informações adicionais aos especialistas de domínio	No entanto, em contraste com o processo de especificação, a iniciativa aqui deve ser liderada pelos designers de jogos, que são responsáveis por solicitar informações adicionais, filtrando os aspectos que não são compatíveis com o gameplay(jogabilidade) e identificar aspectos que ainda não foram capturados. (BORRO;ESCRIBANO et al, 2014)
<b>CP2</b>	Criar elementos e características do jogo	Criar elementos e características do jogo derivados de diretrizes de design sugeridos através de investigação relacionada com a psicologia, a semiótica e a interface, dentro do campo de interface homem computador(IHC). Avaliar os elementos do jogo principalmente para o projeto da interface do jogo. Realizar entrevista para conhecer as práticas dos alunos quando eles jogam com

		computador em casa e considerar possíveis aspectos para conceituar características do jogo a ser construído. Estas características colhidas junto aos alunos podem ser a de preferência em jogos em primeira ou terceira pessoa, enredo com a presença de mistérios com elementos escondidos, resolver problemas, perguntas e respostas pertinentes, desafios para serem capazes de continuar com as etapas do jogo, etc. (LAZAROU, 2001)
<b>CP3</b>	Simular o jogo	Simular o jogo e colher informações relevantes para possíveis aprimoramentos. Recomendamos rastrear cada ação relevante (correta ou incorreta) que é tomada pelo jogador e usá-lo para gerar um relatório, entregue ao aluno para autoavaliação no final da simulação do jogo, ou entregue ao instrutor para monitorar o desempenho dos alunos. O relatório inclui comentários detalhados sobre como o estudante usa o procedimento. Esta informação também pode ser usada para identificar potenciais pontos fracos do jogo simulado, corrigir a formalização do procedimento, se necessário e reforçar a aprendizagem sem a necessidade de intervenção do professor (BORRO ; ESCRIBANO et al. 2014)
	Criar pequenos protótipos iterativos	Um estudo realizado anteriormente de práticas pedagógicas, e processos mentais dos estudantes quando da resolução de problemas que envolvem a área do jogo sério devem ser utilizados como base para gerar esses pequenos protótipos. Recomendamos especial atenção à criação de interações que podem fazer um processo abstrato mais concreto para os alunos. A oportunidade para a manipulação direta de artefatos em jogos é um benefício potencial do uso de jogos para educação. O público alvo deve usar os protótipos e os investigadores devem documentar onde a mecânica foi bem e mal sucedida. Usar o protocolo think-aloud pode ajudar a identificar problemas com a mecânica do jogo e regras que precisam de ser adicionadas para melhorar a aprendizagem. (EAGLE; BARNES, 2012)
<b>CP4</b>	Inserir elementos de tomada de decisão	O jogo deve: (1) Orientar o usuário na descoberta de informações disponíveis relacionada com a decisão e as diferentes opções disponíveis; (2) Proporcionar ao aluno uma forma simples de executar essas ações no jogo. A maneira mais simples de apresentar opções é o uso questões de múltipla escolha apresentadas no texto, apesar destas opções poderem também ser apresentadas visualmente utilizando animações (a um ligeiro maior custo de produção). O jogo também deve fornecer mecanismos para a execução de decisões. (TORRENTE et al. 2014)
<b>CP4</b>	Definir os feedbacks do jogo	Indicar que uma interação desejada do jogador foi realmente executada. Comentários também devem apoiar e facilitar a reflexão sobre os acontecimentos que ocorrem no jogo e, especialmente, destacar os efeitos sobre decisões erradas dos procedimentos. O feedback também contribui para criar uma percepção contínua do progresso, uma característica inerente de bons jogos que evita frustração e encoraja o jogador a ir em frente, entre outros benefícios. Na maioria dos casos, o feedback deve ser simples (limitado a áudio ou pistas visuais simples), curto e não-intrusivo a fim de evitar a quebra do ritmo do jogo. O feedback deve ser mais explícito e amplo em partes do processo, que são especialmente complexo. Isso ajuda os alunos refletirem sobre os conceitos subjacentes de forma mais aprofundada e transferirem os conhecimentos adquiridos a situações do mundo real. Um pequeno vídeo apresentando um aspecto particular de um procedimento pode ser mais útil e envolvente. Ao lidar

		com o feedback para ações incorretas, o tempo pode ser ajustado de modo a refletir as consequências de manipulações válidas e manipulações inválidas, tanto no próprio processo e nos seus possíveis subprodutos. É importante para assegurar que o jogo baseia-se nestas situações, fazer os alunos "pagar um preço" por erros (por exemplo, ter que começar de novo) e ajudá-los a evitar o mesmo erro próxima vez. Quando o jogo for concluído, o estudante pode ser apresentado com feedback definitivo para fins de autoavaliação, incluindo uma lista de todos os erros e ações incorretas realizadas. Esta informação pode ser utilizada para identificar potenciais fraquezas no jogo, ajustar a formalização do procedimento e reforçar a aprendizagem sem a necessidade de intervenção de professores. (TORRENTE et al. 2014)
<b>CP4</b>	Efetuar a gamificação do projeto	Aplicar técnicas para aumentar o engajamento e a motivação do estudante e reforçar a aprendizagem. Intriga e curiosidade, oportunidades para o desafio, estratégia e solução de problemas, a concorrência (auto-dirigida ou com seus pares) ou humor podem ser implementados de forma eficaz em termos de custos. Tecnologia muito avançada ou gráficos 3D são atraentes e podem contribuir para criar uma experiência imersiva no ambiente, mas são comparativamente muito mais caros. A utilização de heurísticas quantificáveis para mostrar o progresso dos alunos e realizações na forma de uma nota pode fornecer uma forma visível e eficaz de desafio. Estas heurísticas podem ser complementadas como emblemas ou outros elementos visuais que podem fornecer status de exibição ou competências adquiridas pelo aluno. Outras fontes potenciais de desafio incluem o número total de objetivos cumpridos, o tempo necessário para completar cada procedimento, ou o número total de erros. (TORRENTE et al. 2014)
<b>CP5</b>	Criar o documento de design do jogo	Este documento descreve a visão para o jogo, conteúdo, horários e estratégia de implementação. Ele fornece diretrizes para o projeto como um todo, e as instruções para a equipe de desenvolvimento do jogo (produtores, designers de jogos, artistas e programadores) (SHIRATUDDIN, 2011)
<b>CP5</b>	Elaborar documento de concepção técnica de arte e produção .	Elaborar documento de concepção técnica de arte e produção do jogo e: Definir Prêmio / Recompensa Definir se jogo Single-Player ou Multi-Player Definir Plataforma do jogo Definir Interface do usuário Definir Estória Definir Concepção de arte Áudio, som, iluminação Arquitetura do jogo ( SHIRATUDDIN, 2011)

Tabela 19 - Descrição atividades selecionadas para a disciplina implementação

<b>DISCIPLINA IMPLEMENTAÇÃO</b>		
<b>CARACTERÍSTICA</b>	<b>TAREFAS</b>	<b>DESCRIÇÃO</b>
<b>CI1</b>	Gerar a partir dos requisitos uma simulação com protótipo do jogo	Esta simulação pode ser criada usando uma ferramenta de autoria de jogos como eAdventure ou outra ferramenta. Isso reduz o custo e aumenta a participação de especialistas de conteúdo no processo de desenvolvimento. Além disso, permite e facilita uma prototipagem rápida, o que acelera os processos de verificação de conteúdo, pois, no protótipo as revisões são mais produtivas, ao invés de desenhos ou documentos. Esta abordagem também permite que os designers de jogos testem rapidamente a mecânica do jogo e ideias, além de identificar as principais dificuldades no projeto. Em uma primeira iteração criar protótipos mock-up (ou seja, simulações com recursos de arte provisórios), que permite que o especialista de domínio verifique se o conhecimento explícito e tácito da área do jogo foram devidamente capturados. Em iterações intermédias gerar protótipos que estão orientados para a avaliação dos elementos do jogo que estão incluídos. As iterações finais têm versões estáveis, incluindo os recursos de arte finais para avaliar o engajamento com os usuários finais. (BORRO-ESCRIBANO et al, 2014)
<b>CI1</b>	Desenvolver protótipo inicial do jogo	Um protótipo inicial do jogo deve ser desenvolvido principalmente com base nas características e elementos, derivados a partir de propostas e recomendações feitas por alunos e professores durante as entrevistas. O protótipo do jogo deve também ser formado por estratégias complementares de design incluindo diretrizes de projeto decorrentes da pesquisa Interface Homem computador(IHC), as diretrizes de design derivados do cotidiano dos alunos e experiências e hábitos ao jogar videogames em casa, e diretrizes de projeto derivadas da área alvo do jogo sério. (LAZAROU, 2011)
<b>CI2</b>	Utilizar uma ou mais ferramentas de prototipagem rápida e implementação.	Utilizar uma ou mais ferramentas de criação para prototipagem rápida e implementação do jogo até a versão final. Para a apresentação de implementação, a equipe de desenvolvimento deve aplicar os elementos (modelos 3D, sons, imagens, texturas, animações) ou reutilização de bibliotecas existentes, como Google Warehouse para os modelos 3D. O cenário e as interfaces precisa agora serem desenvolvidos. Para isso, vários motores de jogo e editores podem ser utilizados, tais como, Unreal, Unity3D, Blender e 3D Studio (Max. ROCHA et al. 2012)
<b>CI2</b>	Usar ferramentas simples para criar protótipos	Utilizar uma ferramenta de autoria de alto nível mais simples, o que simplifica um processo de gerar protótipos funcionais do jogo. Utilizar como entrada o documento de design produzido na tarefa anterior. O principal resultado é um conjunto de protótipos de trabalho para ser utilizado para diferentes fins (TORRENTE et al. 2014)
<b>CI3</b>	Aprimorar	Protótipos dos métodos de ensino criados na fase de projeto devem ser

	protótipos do projeto	transformados em jogo. Para ser eficaz para a aprendizagem, o ambiente do jogo deve promover os mesmos comportamentos que são necessários para resolver o pós-teste que os alunos devem ser capazes de completar depois de jogar. Desenvolvedores iterativamente devem construir partes testáveis do protótipo, com o feedback frequente dos alunos, dos outros desenvolvedores e instrutores. (EAGLE; BARNES, 2012)
<b>CI4</b>	Cria diferentes protótipos para diferentes finalidades	Efetuar uma prototipagem iterativa que permita uma avaliação gradativa e testes planejados para facilitar análise e tarefas de projeto. É mais simples para os especialistas de domínio encontrar imprecisões ou aspectos negligenciados no protótipo do que analisar desenhos ou documentos. Permite também aos especialistas de jogo testar rapidamente a mecânica do jogo e identificar possíveis inconsistências. Diferentes protótipos podem ser criados para diferentes fins. Protótipos mock-up permitem a avaliação rápida da exatidão do procedimento formalizado. Protótipos intermediários devem ser utilizados principalmente para suscitar comentários sobre o design do jogo e das próximas iterações. Protótipos finais estáveis devem ser usados para a avaliação do usuário final. (TORRENTE et al, 2014)

Tabela 20 - Descrição das atividades selecionadas para a disciplina testes

<b>DISCIPLINA TESTES</b>		
<b>CARACTERÍSTICA</b>	<b>TAREFAS</b>	<b>DESCRIÇÃO</b>
<b>CT1</b>	Efetuar testes através dos protótipos desenvolvidos	A partir da simulação desenvolvida na etapa de implementação, testes deverão ser efetuados. A simulação que foi criada no processo de desenvolvimento deve testar a mecânica do jogo e identificar as principais dificuldades no projeto. Em uma primeira iteração criar protótipos mock-up (ou seja, simulações com recursos de arte provisórios), que permite que o especialista de domínio verificar se o conhecimento explícito e tácito da área do jogo foram devidamente capturados.
<b>CT2</b>		Em iterações intermediárias gerar protótipos que estão orientados para a avaliação dos elementos do jogo que estão incluídos. As iterações finais têm versões estáveis, incluindo os recursos de arte finais para avaliar o engajamento com os usuários finais. (BORRO-ESCRIBANO et al. 2014)
<b>CT3</b>	Utilizar protótipos para testar e receber feedback das partes interessadas e envolvidas no jogo	Usar o protótipo para efetuar testes e utilizar o protocolo think-aloud com membros do público alvo para assegurar que o jogo é uma tradução eficaz do especificado nas metas de aprendizagem. Os desenvolvedores devem fazer alterações na interface do jogo e decisões usando o feedback obtido a partir da observação de jogadores, tendo em mente que a tomada de decisão "mais fácil" poderia prejudicar a aprendizagem. (EAGLE; BARNES. 2012)
<b>CT4</b>	Efetuar a simulação diversas vezes para testar e verificar requisitos.	A simulação deve ser executada várias vezes para testar e verificar os requisitos. (ROCHA et al. 2012)
<b>CT5</b>	Efetuar teste beta avaliando a jogabilidade, usabilidade e possíveis erros.	Os membros da equipe de desenvolvimento, colegas, ou estudantes, podem ser recrutados para executar beta testes com protótipos. Os beta-testadores devem explorar o jogo, experimentando todas as ações e opções possíveis, em busca de bugs escondidos, erros técnicos, falhas de usabilidade e devem fornecer feedback sobre o valor educacional e experiência de diversão esperada. (TORRENTE et al. 2014)



Tabela 21 - Descrição das atividades selecionados para a disciplina avaliação

<b>DISCIPLINA AVALIAÇÃO</b>																										
<b>CARACTERÍSTICA</b>	<b>TAREFAS</b>	<b>DESCRIÇÃO</b>																								
<b>CA1</b>	Avaliar a garantia de qualidade do jogo em ambiente real de seu público alvo.	<p>Verificar de alguns ou todos os seguintes aspectos:</p> <p>(1) teor de validade (isto é, o processo tem sido representado com precisão);</p> <p>(2) confiabilidade (ou seja, a simulação do jogo é estável e livre de erros de programa);</p> <p>(3) facilidade de utilização (ou seja, a simulação jogo é agradável e fácil de usar);</p> <p>(4) O valor educativo (isto é, o os alunos podem atingir os objetivos de aprendizagem pretendidos)</p> <p>(5) o envolvimento (ou seja, o jogo é apelativo e envolvente e desafia as habilidades do jogador.</p> <p>A importância de cada aspecto varia dependendo do contexto específico.</p> <p>A validação do conteúdo só pode ser feito por especialistas do domínio através de uma avaliação dos diferentes mock-ups (KEBRITCHI et al. 2010 apud BORRO-ESCRIBANO et al. 2014)</p>																								
<b>CA2</b>	Avaliar usabilidade através de testes	Realizar testes de usabilidade com pares de estudantes e colher informações sobre alterações necessárias a serem feitas no protótipo do jogo. Deve-se eliminar problemas de usabilidade e avaliar a utilidade do jogo na realização dos seus objetivos pretendidos. O teste de usabilidade deve usar o protocolo think aloud a fim de que os alunos expressem suas reações e façam recomendações ao interagir com o jogo. Realizar testes também com os professores com o método think aloud como ferramenta de obtenção de dados, principalmente comentários sobre se o jogo pode ter sucesso em fazer que os alunos aprendam o conteúdo. (LAZAROU, 2011)																								
<b>CA3</b>	Avaliar aspectos técnicos e educacionais do jogo	<p>Efetuar as seguintes questões de avaliação:</p> <table border="1"> <thead> <tr> <th>Questão</th> <th>Quem responde?</th> <th>Relacionado a</th> </tr> </thead> <tbody> <tr> <td>Existem erros ou comportamentos inesperados no jogo?</td> <td>Beta-Testers</td> <td>Confiabilidade</td> </tr> <tr> <td>O jogo funciona em todas as plataformas-alvo?</td> <td>Beta-Testers</td> <td>Confiabilidade</td> </tr> <tr> <td>O jogo simula fielmente o processo e ambiente proposto?</td> <td>Especialistas de Domínio</td> <td>Valor educacional</td> </tr> <tr> <td>O conteúdo incorporado no jogo (por exemplo, materiais de referência, diálogos, texto) são precisos e adequados para o público-alvo?</td> <td>Especialistas de Domínio</td> <td>Valor educacional</td> </tr> <tr> <td>Os objetivos do jogo e as regras são claras para o usuário iniciante?</td> <td>Estudantes, especialistas de domínio, especialistas de Jogos</td> <td>Jogabilidade, Valor educacional, Experiência de uso</td> </tr> <tr> <td>O jogo é atraente para o público-alvo?</td> <td>Estudantes, especialistas de Jogos</td> <td>Jogabilidade, Experiência de uso</td> </tr> <tr> <td>O contexto e as condições são claras</td> <td>Estudantes,</td> <td>Jogabilidade, Valor</td> </tr> </tbody> </table>	Questão	Quem responde?	Relacionado a	Existem erros ou comportamentos inesperados no jogo?	Beta-Testers	Confiabilidade	O jogo funciona em todas as plataformas-alvo?	Beta-Testers	Confiabilidade	O jogo simula fielmente o processo e ambiente proposto?	Especialistas de Domínio	Valor educacional	O conteúdo incorporado no jogo (por exemplo, materiais de referência, diálogos, texto) são precisos e adequados para o público-alvo?	Especialistas de Domínio	Valor educacional	Os objetivos do jogo e as regras são claras para o usuário iniciante?	Estudantes, especialistas de domínio, especialistas de Jogos	Jogabilidade, Valor educacional, Experiência de uso	O jogo é atraente para o público-alvo?	Estudantes, especialistas de Jogos	Jogabilidade, Experiência de uso	O contexto e as condições são claras	Estudantes,	Jogabilidade, Valor
Questão	Quem responde?	Relacionado a																								
Existem erros ou comportamentos inesperados no jogo?	Beta-Testers	Confiabilidade																								
O jogo funciona em todas as plataformas-alvo?	Beta-Testers	Confiabilidade																								
O jogo simula fielmente o processo e ambiente proposto?	Especialistas de Domínio	Valor educacional																								
O conteúdo incorporado no jogo (por exemplo, materiais de referência, diálogos, texto) são precisos e adequados para o público-alvo?	Especialistas de Domínio	Valor educacional																								
Os objetivos do jogo e as regras são claras para o usuário iniciante?	Estudantes, especialistas de domínio, especialistas de Jogos	Jogabilidade, Valor educacional, Experiência de uso																								
O jogo é atraente para o público-alvo?	Estudantes, especialistas de Jogos	Jogabilidade, Experiência de uso																								
O contexto e as condições são claras	Estudantes,	Jogabilidade, Valor																								

		para o usuário iniciante?	especialistas de domínio, especialistas de Jogos	educacional, Experiência de uso
		É fácil aprender a usar o jogo?	Beta-Testers, Estudantes, especialistas de Jogos	Usabilidade
		Os objetivos de aprendizagem são contemplados?	Especialistas de Domínio	Valor educacional
		Quão bem os alunos alcançam as metas de aprendizagem?	Estudantes	Valor educacional
		Será que os alunos aprendem mais alguma coisa (conhecimento incorreto ou correto)	Estudantes	Valor educacional
		Avaliação formal é o tipo mais demorado de avaliação e, portanto, apenas uma ou duas experiências no final do processo são recomendadas, orientada para avaliar o ganho educacional em relação ao ensino tradicional. O resultado é um conjunto de propostas de modificação a qualquer um dos os sub-produtos das outras tarefas (análise e projeto documentos ou protótipos). (TORRENTE et al. 2014)		
<b>CA4</b>	Utilizar mecanismos de vídeo, áudio ou texto para gravar testes de experiência prática do jogo.	Uma câmara de vídeo capturando a interação de um par de estudantes com o jogo, outra câmera de vídeo em movimento para capturar as discussões durante a aula, as ações do professor, e a experiência prática realizada. Gravadores de áudio também devem ser usados para gravar interações de estudantes enquanto se envolvem com o jogo. Entradas de texto dos alunos dentro do jogo devem ser automaticamente salvos em arquivos de texto externo ao jogo, e suas ações dentro do jogo devem ser gravadas por uma gravador de tela. (LAZAROU, 2011)		

### 3.5 MELHORIAS NA METODOLOGIA CONCEBIDA

Durante o início da execução da metodologia de desenvolvimento do jogo sério educacional concebida no capítulo 3, foram identificadas oportunidades de melhorias de itens desta metodologia. Após uma nova análise, as seguintes melhorias foram realizadas:

- a) Adequação do fluxo de execução de algumas atividades
- b) Eliminação de atividades superpostas
- c) Melhoria da descrição de atividades
- d) Melhor identificação dos produtos de trabalho(artefatos) gerados

A tabela 20 descreve as melhorias efetuadas na metodologia. As atividades, as ações e motivos destas melhorias estão explanadas.

Tabela 22 - Lista de melhorias efetuadas na metodologia concebida capítulo 3

ATIVIDADE(S)	AÇÃO(ÕES)	MOTIVO(S)
Fornecer aos desenvolvedores uma melhor compreensão do público-alvo	Exclusão da atividade e inserção do seu texto na atividade <u>compreender melhor o domínio do problema por parte dos desenvolvedores.</u>	As tarefas da atividade complementam a da atividade <u>compreender melhor o domínio do problema por parte dos desenvolvedores.</u>
Realizar entrevistas com especialistas de domínio	Exclusão da atividade e transformação em uma tarefa da <u>atividade compreender melhor o domínio do problema por parte dos desenvolvedores</u>	A atividade possui uma configuração mais próxima de uma tarefa pertencente à atividade a qual foi inserida.
Elaborar documento de concepção técnica de arte e produção.	Exclusão e inserção do texto desta atividade na atividade <u>criar o documento de design e conceito do jogo.</u>	A atividade possui tarefas que são subconjuntos do conjunto de tarefas pertencente a <u>criar o documento de design e conceito do jogo.</u>
Criar o documento de design e conceito do jogo	A atividade descreve itens incompletos de um documento de design e conceito de um jogo. Desta forma, foi adicionada ao texto desta atividade, seções de um modelo completo de documento de design e conceito de um jogo, bem difundido na literatura e pertencente a Schyutema (2008).	O documento descrito na atividade não possuía muitos elementos pertencentes a um documento de design completo.
Definir os elementos do jogo	Exclusão da atividade e inserção do texto desta atividade na atividade <u>criar elementos e características do jogo.</u>	A atividade possui uma configuração mais próxima de uma tarefa pertencente à atividade a qual foi inserida.
Criar o Script o do jogo e melhorar o script do jogo	Fusão destas atividades em uma nova atividade denominada: <u>criar e melhorar um script do jogo.</u>	As atividades fundidas se complementam.
Criar o documento de requisitos do jogo	Foram acrescentadas, para estruturar melhor o documento, as seguintes seções baseadas em Sommerville (2007): INTRODUÇÃO: objetivos, convenções, termos e abreviações, escopo, visão geral. DESCRIÇÃO GERAL: requisitos funcionais, requisitos não funcionais.	A atividade <u>criar o documento de requisitos do jogo</u> descreve um documento de requisitos, mas carecia de uma estrutura mais formal e usual de um documento de requisitos.
Criar o documento de requisitos do jogo	A atividade foi movida no fluxo para ser executada anteriormente à atividade <u>criar e melhorar o script do jogo.</u>	A atividade de <u>criar e melhorar o script do jogo</u> utiliza o documento de requisitos como artefato de entrada.
Simular o jogo	Exclusão da atividade e inserção do seu texto na atividade <u>gerar a partir dos requisitos uma simulação com protótipos evolucionários do jogo.</u>	A atividade possui texto que complementa a outra atividade a qual foi inserida.
Gerar a partir dos requisitos uma simulação com protótipos do jogo	A atividade foi dividida em duas <u>gerar a partir dos requisitos protótipos evolucionários do jogo</u> e <u>simular o jogo com os protótipos</u>	A atividade possui tarefas que pertencem a dois tipos de atividades: a de prototipação e a de simulação.
Simular o jogo com protótipos	Esta atividade criada no item anterior foi inserida na atividade <u>efetuar a simulação diversas vezes para testar e verificar os requisitos.</u>	As atividades possuem textos complementares sendo possível criar uma única atividade com suas tarefas.
Gerar a partir dos requisitos protótipos evolucionários do jogo	Esta atividade oriunda da divisão anteriormente citada foi renomeada para <u>gerar a partir dos requisitos e documento de design, protótipos evolucionários do jogo.</u>	Além do documento de requisitos o documento de design do jogo foi colocado como um dos artefatos de entrada, por possuir diversos elementos relacionados ao jogo.
Criar um documento com os procedimentos formalizados, pronto para ser usado como entrada para o projeto do jogo.	Supressão de algumas tarefas da atividade deixando apenas os elementos genéricos, e exclusão da atividade com adição do seu texto resultante na atividade <u>criar elementos e características do jogo.</u>	Algumas tarefas eram destinadas especificamente a área de saúde e foram suprimidas. O texto resultante ajudará a complementar o texto da atividade <u>criar elementos e características do jogo.</u>
Adicionar erros comuns, armadilhas e elementos motivacionais; Inserir elementos	Fusão dos textos destas 3 atividades em uma única atividade intitulada <u>inserir elementos motivacionais, de decisão, armadilhas e feedbacks.</u>	As atividades possuem textos complementares sendo possível criar uma única atividade com suas tarefas.

<b>de tomada de decisão; Definir os feedbacks</b>		
<b>Criar diferentes protótipos para diferentes finalidades</b>	Exclusão da atividade e inserção do seu texto na atividade <u>gerar a partir dos requisitos e documento de design, protótipos evolucionários do jogo.</u>	As atividades possuem textos complementares sendo possível criar uma única atividade com suas tarefas.
<b>Desenvolver o protótipo inicial do jogo</b>	Exclusão da atividade e inserção da mesma como tarefa da atividade <u>gerar a partir dos requisitos e documento de design, protótipos evolucionários do jogo.</u>	A atividade possui uma configuração mais próxima de uma tarefa pertencente à atividade a qual foi inserida.
<b>Utilizar uma ou mais ferramentas de prototipagem rápida e implementação</b>	A atividade foi renomeada para <u>desenvolver e codificar o jogo até a versão final.</u>	O nome da atividade ficou mais nítido em relação a seu propósito.
<b>Aprimorar o protótipo do projeto</b>	Esta atividade foi excluída e foi inserida como tarefa na atividade <u>gerar a partir dos requisitos e documento de design, protótipos evolucionários do jogo.</u>	A atividade aprimorar o protótipo do projeto possui tarefas que complementam a <u>atividade gerar a partir dos requisitos e documento de design, protótipos evolucionários do jogo.</u>
<b>Utilizar protótipos para testar e receber feedback das partes interessadas e envolvidas no jogo</b>	Esta atividade foi inserida na atividade <u>efetuar testes através dos protótipos desenvolvidos.</u>	As atividades possuem textos complementares sendo possível criar uma única atividade com suas tarefas.
<b>Criar e melhorar o script do jogo</b>	Além da especificação de requisitos, foi inserido como artefato de entrada para esta atividade o documento de design do jogo.	O documento de design irá oferecer mais elementos para a construção e melhoria do script do jogo.
<b>Efetuar a gamificação do projeto</b>	Esta atividade foi excluída. Contudo as tarefas listadas a seguir foram inseridas na atividade <u>inserir elementos motivacionais, de decisão, armadilhas e feedbacks.</u> <ul style="list-style-type: none"> <li>• Aplicar técnicas para aumentar o engajamento e a motivação do estudante.</li> <li>• Reforçar a aprendizagem utilizando tecnologia muito avançada ou gráficos 3D atraentes para contribuir para criar uma experiência imersiva no ambiente do jogo;</li> <li>• Utilização de heurísticas quantificáveis para mostrar o progresso dos alunos</li> </ul>	Foi excluída por ser semelhante à atividade <u>inserir elementos motivacionais, de decisão, armadilhas e feedbacks.</u> As partes não semelhantes foram então aproveitadas como tarefas.
<b>Gerar a partir dos requisitos uma simulação com protótipo do jogo</b>	A tarefa criar protótipos mock-up (ou seja, simulações com recursos de arte provisórios), foi retirada da atividade e colocada como tarefa da atividade <u>criar e simular um script para o jogo.</u>	A tarefa está alinhada com a atividade de criar e simular um script para o jogo, através da criação de scripts com recursos de arte provisórios.
<b>Desenvolver e codificar o jogo até a versão final</b>	Esta atividade foi deslocada para o final da fase de produção ficando após as atividades de <u>efetuar testes através dos protótipos desenvolvidos</u> e <u>efetuar a simulação diversas vezes para testar e verificar os requisitos.</u>	O fluxo de criação dos protótipos, testes, simulação e desenvolvimento do jogo completo ficou mais adequado ao ciclo de desenvolvimento do jogo.
<b>Efetuar a simulação diversas vezes para testar e verificar os requisitos.</b>	A atividade foi excluída.	A atividade é semelhante a atividade de <u>efetuar testes através dos protótipos desenvolvidos.</u>
<b>Utilizar mecanismos de vídeo, áudio ou texto para gravar testes de experiência prática do jogo</b>	A atividade foi excluída e foi inserida como tarefa na atividade <u>efetuar teste beta avaliando a jogabilidade, usabilidade e possíveis erros.</u>	A atividade possui uma configuração mais próxima de uma tarefa pertencente à atividade a qual foi inserida.
<b>Avaliar usabilidade através de testes</b>	A atividade foi excluída e foi inserida como tarefa na atividade <u>efetuar teste beta avaliando a jogabilidade, usabilidade e possíveis erros.</u>	A atividade possui uma configuração mais próxima de uma tarefa pertencente à atividade a qual foi inserida.
<b>Avaliar a garantia de qualidade do jogo em ambiente real de seu público alvo.</b>	O texto descrito como: a validação do conteúdo só pode ser feito por especialistas do domínio através de uma avaliação dos diferentes mock-ups foi modificado por: o conteúdo educacional trabalhado no jogo é válido? (só pode ser feito por especialistas do domínio através de uma	A modificação textual foi efetuada para adequar a tarefa da atividade para o estilo de jogo sério educacional. O texto original tem foco em jogo sério de simulação.

	avaliação de todos os conceitos trabalhados no jogo)	
<b>Avaliar a garantia de qualidade do jogo em ambiente real de seu público alvo.</b>	O item teor de validade (isto é, o processo é representado com precisão) presente no texto da atividade foi suprimido.	A atividade indica que a importância de cada elemento varia dependendo do contexto específico. Neste caso este item refere-se especificamente a jogos sérios de simulação e o foco do trabalho é um jogo sério educacional. Desta forma o item foi retirado da fase de avaliação.

A descrição das atividades da metodologia já com as melhorias, poderá ser consultada no apêndice A. Para o desenvolvimento do jogo foi utilizado esta versão melhorada da metodologia. A seguir a figura 34 mostra o fluxo final das atividades da metodologia já com as melhorias.

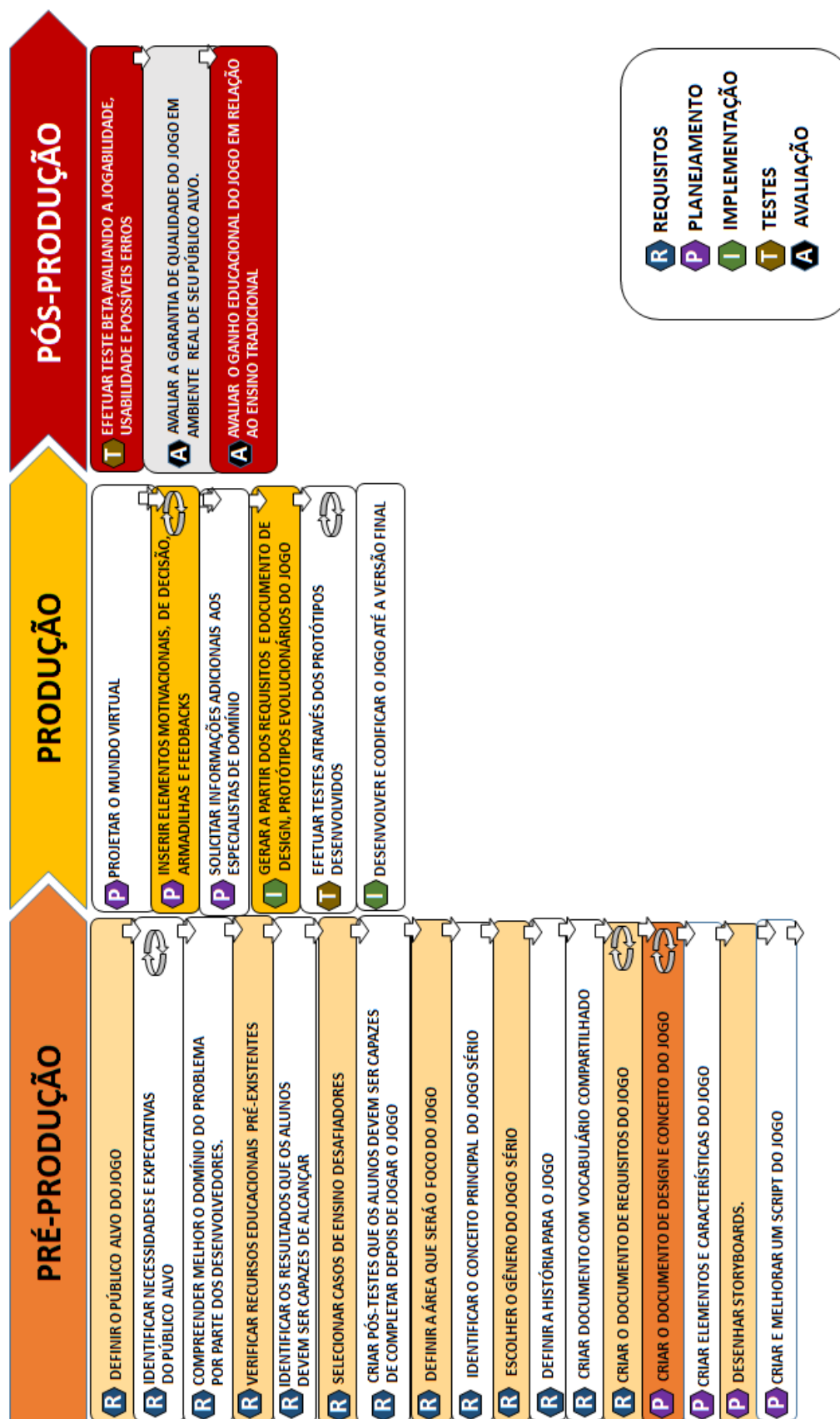


Figura 34 - Fases e atividades da metodologia após aplicação das melhorias

A partir da próxima seção todas as atividades deste fluxo serão executadas e os resultados serão descritos.

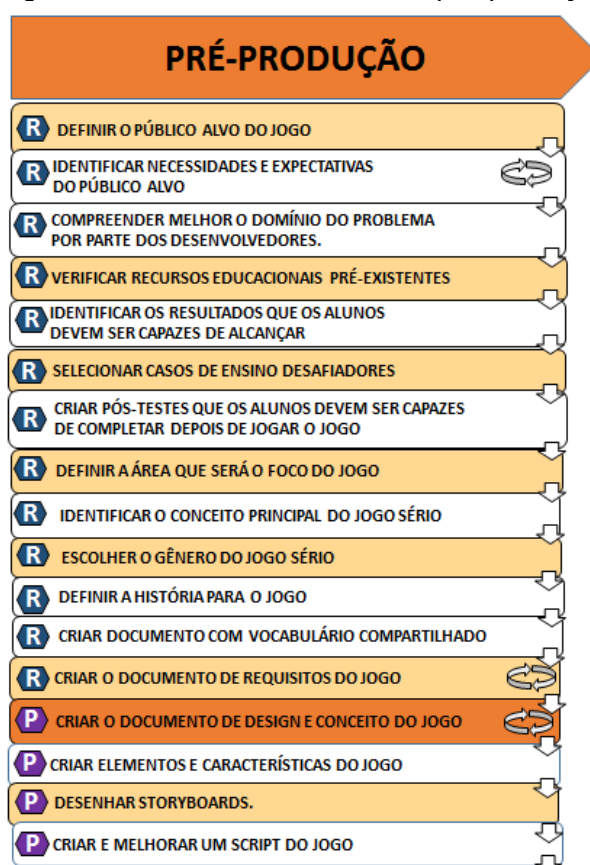
## 4 DESENVOLVIMENTO DO JOGO SÉRIO COM BASE NA METODOLOGIA CUSTOMIZADA

Este capítulo descreve a construção do jogo sério educacional, utilizando a metodologia concebida no capítulo 3. Aqui se descreve brevemente a execução das atividades pertencentes à metodologia customizada e os produtos de trabalho resultantes (artefatos).

### 4.1 ATIVIDADES DE PRÉ-PRODUÇÃO

As atividades a seguir são parte integrante da fase de pré-produção, da metodologia de desenvolvimento do jogo sério. Estas atividades, indicadas na figura 35, serão executadas e seus produtos de trabalho (artefatos) serão indicados.

Figura 35 - Atividades da fase de pré-produção



#### 4.1.1 Atividades acerca do público alvo e domínio do problema

A execução da metodologia começa com a execução das atividades relacionadas com o público alvo do jogo sério a ser desenvolvido. A primeira



atividade é a de definir o público alvo do jogo<sup>1</sup>. Na execução desta atividade foi estabelecido o seguinte perfil para o público alvo:

- a) Alunos do primeiro ao terceiro semestre de graduação dos cursos de bacharelado em Ciência da Computação e bacharelado em Sistemas de Informação;
- b) Que cursam disciplinas que envolvem linguagem de programação C;
- c) Que não possuem algum tipo de deficiência física ou mental.

Desta forma o jogo sério educacional sobre linguagem de programação C, será desenvolvido com foco neste perfil de usuário. Além disso, foi definido que o aluno deverá jogar o jogo de forma dedicada e que o jogo visa proporcionar benefícios de apoio a sua aprendizagem relacionada a linguagem C.

Com objetivo de identificar as necessidades e expectativas deste público alvo, foi criado um questionário relacionado com o ensino e aprendizado da linguagem de programação C. O objetivo da aplicação desta entrevista via questionário, foi tentar mapear itens relacionados com o software (jogo), e adicionalmente itens relacionados com a pedagogia envolvida no processo de ensino da linguagem C.

Um total de 93(noventa e três) alunos responderam a este questionário. Os itens presentes se constituíam em perguntas de múltipla escolha que tentou identificar:

- a) Os maiores erros cometidos pelos alunos em relação à sintaxe da linguagem C;
- b) Que conceitos sobre a linguagem de programação C, cada aluno considerava que mais tem ou teve dificuldade de entendimento;
- c) Os motivos que acreditavam não terem logrado êxito na disciplina;
- d) Que tipos de jogos digitais que mais gostam de jogar;
- e) Se gostam ou jogariam um jogo do tipo atirador em primeira pessoa FPS(First Person Shooter);
- f) E qual tipo de jogo que mais gosta de jogar: 3D, 2D ou ambos.

Trechos deste questionário podem ser observados na figura 36, e o questionário completo está descrito no apêndice B.

---

<sup>1</sup> O estilo da fonte em itálico e sublinhado tem o objetivo de destacar os nomes das atividades da metodologia.

Figura 36 - Trechos das perguntas e opções do questionário discente

<b>1) Quais foram os ERROS MAIS FREQUENTES que você cometeu ao cursar disciplina(s) com linguagem C?</b>							
1.1 ( )	Na lógica da programação. De como resolver o problema proposto no exercício						
1.2 ( )	Esquecer o ponto e vírgula ; no final de comandos, etc.						
1.3 ( )	Uso incorreto do número de parênteses. Ex: <code>if ((x==1) &amp;&amp; (x==2))    ((y==0) /* aqui falta parênteses*/</code>						
1.4 ( )	Uso inadequado da vírgula. Ex: <code>printf("numero=",%d,"num)</code> sendo o correto <code>printf("numero=%d",num);</code>						
1.5 ( )	Esquecer o <code>&amp;</code> no <code>scanf</code> travando o sistema com tela de erro. Ex: <code>scanf("%d", num);</code> ao invés de <code>scanf("%d", &amp;num);</code>						
1.6 ( )	Esquecer de fechar chaves <code>{ }</code> de início e fim em algum lugar do código						
<b>2) Quais os comandos e/ou conceitos da linguagem C que você teve MAIOR DIFICULDADE de aprendizado?</b>							
2.1 ( )	Comando <code>if-else</code> (se então)	2.14 ( )	Loops(repetições) <code>while</code> , <code>for</code>				
2.2 ( )	Estruturas( <code>Struct</code> )	2.15 ( )	<code>Strings - char nome[80];</code>				
2.3 ( )	<code>scanf("%d", &amp;num)</code>	2.16 ( )	inclusão bibliotecas <code>#include &lt;string.h&gt;</code>				
2.4 ( )	Declaração de variáveis	2.17 ( )	Operador <code>% MOD</code> (resto da divisão inteira)				
2.5 ( )	Uso de parênteses. ex: <code>if ((achou)==1)</code>	2.18 ( )	Manipulação arquivos texto e binário				
<b>3) Que elementos ATRAPALHARAM no seu aprendizado da disciplina com linguagem C?</b>							
3.1 ( )	A Linguagem C é complexa						
3.2 ( )	Falta de prática em sala de aula						
3.3 ( )	Falta de prática em casa						
3.4 ( )	Falta de tempo para programar						
3.5 ( )	Falta de contato diário com a linguagem C						
<b>4) Se existisse um jogo sobre linguagem C que elementos considera importante que o jogo possua?</b>							
4.1 ( )	Que tenha boa jogabilidade	4.6 ( )	Que possua Gráfico 3D e boa interface gráfica				
4.2 ( )	Que seja de fácil compreensão	4.7 ( )	Que tenha bom cenário				
4.3 ( )	Que ensine comandos da linguagem	4.8 ( )	Que seja bem simples com muita decisão				
<b>5) Que tipo de jogo mais gosta de jogar?</b>							
5.1 ( )	Ação e aventura	5.3 ( )	Tiro em primeira/terceira pessoa(FPS)	5.5 ( )	Multiusuário	5.7 ( )	Futebol
5.2 ( )	RPG	5.4 ( )	Esportes	5.6 ( )	Estratégia	5.8 ( )	Luta
<b>OBS: NAS QUESTÕES 6,7,8 ABAIXO MARQUE COM UM X APENAS UMA DAS ALTERNATIVAS</b>							
<b>6) Você gosta de jogos do tipo FPS (tiro em primeira pessoa)?</b>							
6.1 ( )	Gosto	6.2 ( )	Não gosto				
<b>7) Você jogaria um jogo FPS (tiro em primeira pessoa) sobre linguagem C?</b>							
7.1 ( )	Sim	7.2 ( )	Sim se não tiver sangue ou muita violência	7.3 ( )	Não		
<b>8) Prefere jogos em:</b>							
8.1 ( )	3D	8.2 ( )	2D	8.3 ( )	AMBOS		

Um item do questionário poderia ser ou não marcado pelo aluno, e era possível escolher mais de uma opção de resposta, excetuando-se os itens das perguntas 5, 6 e 7, que possuíam uma única possibilidade de marcação. A seguir estão descritos na tabela 21, em ordem decrescente de porcentagem de marcação, os resultados da aplicação do questionário. Foram obtidos os seguintes dados:

Tabela 23 - Resultados da aplicação 93 questionários discentes

<b>Quais foram os erros mais frequentes que você cometeu ao cursar disciplina(s) com linguagem c?</b>	
( 46,23% ) Esquecer o ponto e vírgula ; no final de comandos, etc. ( 35,48% ) Esquecer & no scanf travando o sistema com tela de erro. Ex: scanf("%d", num); ( 32,26% ) Na lógica da programação. De como resolver o problema proposto no exercício ( 30,10% ) Usar o especificador incorreto( %d, %f, %c, %lf, %s, %2f ) ( 26,88% ) Não usar aspas simples ' ' em um caractere. Ex: char letra = "A"; ( 26,88% ) Declarar variáveis e não inicializá-las ex: int contador; ao invés de int contador=0; ( 20,43% ) Errar o nome de comandos causando erros de compilação. Ex: printf, switch, double, etc ( 20,43% ) Usar atribuição = ao invés de == em uma comparação. Ex: if (num=2) ( 19,35% ) Não declarar variáveis utilizadas no programa. ( 19,35% ) Errar divisão / (inteira e real) double meio = 1/2; <- 0(zero). double meio = 1.0/2; <- 0,5 ( 17,20% ) Uso inadequado da vírgula. Ex: printf("numero=", %d, num) ( 17,20% ) Esquecer o break final do comando case do switch ( 16,13% ) Usar índices fora do limite do vetor. Ex: int vetor[10]; vetor[10]=34; <-vai de vetor[0] a [9] ( 15,05% ) Errar uso da barra ex: /n ao invés de colocar \n ( 13,98% ) Separar o comando for por vírgula for(i=0,i<5,i++) ao invés do correto com ponto e vírgula; ( 13,98% ) Errar retorno função main( ) e sua declaração ( int main( int argc, char *argv[ ] ) ( 12,90% ) Esquecer de incluir biblioteca com #include. Ex: system("PAUSE"); esquecer <stdlib.h> ( 11,83% ) Uso incorreto do número de parênteses. Ex: if ((x==1) && (x==2))    ((y==0) ( 11,83% ) Colocar ponto e vírgula ; onde não deveria. Ex: depois de if ( ); etc ( 10,75% ) Tentar colocar palavras em um char e só cabe 1 caractere. Ex: char c = "teste"; ( 8,60% ) Esquecer de fechar chaves { } de início e fim em algum lugar do código. ( 6,45% ) Usar de forma errada o caractere nulo '\0' sendo o correto é '\0'	
<b>Quais os comandos e/ou conceitos da linguagem C que você teve maior dificuldade de aprendizado?</b>	
(34,40%) Vetores e Matrizes ex: int vetor[10]; (32,26%) funções string.h strcpy, strlen, strcmp (27,96%) typedef (26,88%) Loops(repetições) while, for (26,88%) Estruturas(Struct) (22,58%) Uso de variável char com tabela (20,43%) Operadores lógicos: &&    If (letra >= 65 && achou) (19,35%) Break e Continue (18,28%) Pré e pós incremento ++i i-- i++ --i (17,20%) Strings - char nome[80]; (16,12%) Comando switch-case(escolha caso) (13,98%) fflush(stdin) – limpeza buffer	( 9,68% ) Operador % (resto da divisão inteira) ( 7,52% ) Variáveis Locais e Globais ( 7,52% ) Especificadores %d %f %lf %s %c ( 6,45% ) Uso da vírgula – ex: printf("%d, %d", num1, num2) ( 6,45% ) Comando if-else (se então) ( 5,38% ) scanf("%d", &num) ( 5,38% ) Declaração de variáveis ( 5,38% ) Uso de parênteses. ex: if ((achou)==1) ( 5,38% ) inclusão bibliotecas #include <string.h ( 4,30% ) printf("%d %c %2f", num, letra, media) ( 3,22% ) Uso aspas simples e duplas ' ' " "
<b>Que elementos atrapalharam no seu aprendizado da disciplina com linguagem c?</b>	
(61,29%) Falta de contato diário com a linguagem C (60,21%) Falta de prática em casa (22,58%) Falta de estudo aprofundado de cada conceito da linguagem C (17,20%) A Linguagem C é complexa (13,98%) Baixa execução de programas e de análise dos erros de compilação (12,90%) Falta de material para iniciantes totais na linguagem C	(12,90%) Didática do professor insuficiente (12,90%) Dificuldades em entender erros de compilação em inglês e seus significados ( 8,60% ) Não consegui entender como computador executa o programa e o manipula na memória ( 7,52% ) Dificuldades no uso das IDE's (DEV++ , CODE BLOCKS, Visual Studio) ( 7,52% ) Falta de prática em sala de aula ( 0,0% ) Falta de aulas utilizando o compilador
<b>Se existisse um jogo sobre linguagem C que elementos considera importante que o jogo possua?</b>	
(79,57%) Que tenha boa jogabilidade (74,19%) Que tenha boa relação entre usuário e o jogo (73,12%) Que não seja chato (56,99%) Que seja de fácil compreensão (54,84%) Que ensine comandos da linguagem	(47,32%) Que revele os erros cometidos na linguagem C (40,86%) Que tenha bom cenário (39,78%) Que possua enigmas (33,34%) Que seja 3D e com boa interface gráfica ( 8,60% ) Que seja bem simples sem muito design
<b>Que tipo de jogo mais gosta de jogar?</b>	
(55,91%) Tiro primeira/terceira pessoa(FPS) (44,08%) RPG (43,01%) Estratégia (38,70%) Ação/Aventura	(33,34%) Multiusuário (32,26%) Futebol (26,88%) Luta (23,66%) Esportes
<b>Você gosta de jogos do tipo FPS (tiro em primeira pessoa)?</b>	
(81,72%) Gosto ( 3,22% ) Não gosto	
<b>Você jogaria um jogo FPS (tiro em primeira pessoa) sobre linguagem C?</b>	
(78,49%) Sim (17,20% ) Não (2,15%) Sim se não tiver sangue ou muita violência	
<b>Prefere jogos em:</b>	
( 20,43% ) 3D ( 77,42% ) AMBOS (4,3%) 2D	

Estes resultados indicaram quais seriam os itens conceituais da linguagem de programação C, mais abordados no jogo sério a ser desenvolvido. Foram inseridos com maior frequência no jogo os erros e conceitos que obtiveram maior porcentagem na aplicação dos questionários.

Alguns professores foram entrevistados e a maioria deles indicou que a falta de prática dos alunos fora do ambiente de sala de aula, aumentam as dificuldades de aprendizado. Este fato relatado pelos docentes apareceu nos resultados obtidos pela aplicação do questionário citado anteriormente. 61,29% dos alunos indicaram a falta de contato diário com a linguagem C, como sendo um entrave para seu aprendizado e 60,21% indicaram que a barreira era a falta de prática em casa.

Finalizadas as atividades relacionadas com público alvo, o domínio do problema foi estabelecido. Através da execução da atividade compreender melhor o domínio do problema por parte dos desenvolvedores foi identificado que o apoio ao ensino e aprendizagem da linguagem C, é o alvo em que o conceito do jogo deveria estar imerso.

Assim começaram a ser definidos conteúdos significativos e desafiadores para o aprendizado dos alunos. Após entrevistas e observação nas aulas, foi estabelecido que o jogo iria abordar não somente conceitos básicos, mas também avançados da linguagem. Algumas lacunas no conhecimento foram identificadas após a aplicação do protocolo think-aloud em algumas aulas:

- a) Dificuldade no entendimento das mensagens de erros do compilador C;
- b) Dificuldade em estabelecer a lógica de resolução do problema;
- c) Dificuldade em entender como um programa é executado na CPU e como ele manipula a memória;
- d) Dificuldade com diversos itens da sintaxe da linguagem C;
- e) Falta de motivação para o estudo da linguagem C.

#### 4.2 ATIVIDADES PEDAGÓGICAS

A execução da atividade verificar recursos educacionais pré-existents, concentrou esforços na busca de itens pré-existent

s utilizados nas aulas envolvendo linguagem de programação C. Muitos professores utilizam listas de exercícios gerais e pontuadas como um dos recursos de aula. Outro recurso é a

elaboração de trabalhos maiores com desafios a serem construídos em linguagem C. Assim a solução de problemas de código utilizando a linguagem de programação C, foram inseridos como elementos centrais do jogo sério desenvolvido.

A próxima atividade objetiva identificar os resultados que os alunos devem ser capazes de alcançar. Neste caso foram estabelecidos os seguintes objetivos:

- a) Os alunos devem ser capazes de diminuir ou excluir a ocorrência de erros de sintaxe durante a programação de aplicativos utilizando a linguagem C.
- b) Deve-se aumentar a motivação dos alunos em relação ao aprendizado da linguagem C
- c) Os alunos devem ser capazes de responder corretamente ao mínimo 80% das questões dos pós-testes após jogarem o jogo e completarem todos os desafios
- d) Os alunos devem ser capazes de direcionar melhor seus estudos, focando nos conceitos que obtiveram maior dificuldade na linguagem C durante o jogo.

Para que o jogo se apresentasse como desafiador para os alunos foi executada a atividade selecionar casos de ensino desafiadores. Alguns exercícios mais elaborados de listas de exercícios e de livros foram utilizados como base para criar desafios mais complexos ao jogador. Detecção de erros em códigos, erros semânticos e erros que somente ocorreriam em tempo de execução, foram inseridos como desafios para o jogador analisar e identificar para prosseguir no jogo.

Um exemplo é referenciado na figura 37, que aborda um assunto que geralmente causa confusão entre os alunos: os operadores de incremento( ++ ) e decremento( -- ).

Figura 37 - Tela extraída do jogo com um desafio de análise de códigos



Outra atividade importante foi a de criar pós-testes que os alunos devem ser capazes de completar depois de jogar o jogo. Nesta atividade foi criado um conjunto de 10(dez) questões que o aluno deve ser capaz de responder após ter jogado o jogo e realizado as análises necessárias quanto ao aprendizado e nuances da linguagem C. Este pós-teste é apresentado no apêndice C.

#### 4.2.1 Atividades sobre o conceito do jogo

O próximo passo foi definir a área que será o foco do jogo. Foi estabelecido que o jogo sério servirá de apoio ao ensino e aprendizado da linguagem de programação C. Com isso, erros de sintaxe e semântica, de execução, estruturas da linguagem e lógica da programação, serão trabalhados no jogo.

Os objetivos definidos para o jogo são:

- Ser um instrumento motivador e de incentivo ao aprendizado da linguagem de programação C;
- Ser uma ferramenta que aborda e tenta diminuir a incidência dos principais erros de sintaxe no código da linguagem C, relatados pelos alunos e professores em pesquisa;
- Melhorar a experiência do aluno com programação de computadores proporcionando um ambiente lúdico e sofisticado na forma de um jogo 3D;

- d) Melhorar o entendimento do aluno acerca da lógica de programação de computadores melhorando a sua eficiência na resolução de problemas e desenvolvimento de algoritmos;
- e) Desmistificar a complexidade da linguagem de programação C fazendo com que exista mais contato com a linguagem.

Após reuniões entrevistas com alunos e alguns professores, foi possível identificar o conceito principal do jogo sério e estabelecer uma abordagem pedagógica. O jogo não será substituído das aulas tradicionais e o professor terá papel fundamental no processo de utilização do jogo e alcance dos objetivos pretendidos.

O conteúdo do jogo será composto por códigos incorretos e corretos relacionados a erros de sintaxe e semântica, erros em tempo de compilação e execução, erros mais frequentes de acordo com relatado por alunos e professores. Ficou estabelecido que o jogo deverá tentar:

- a) Diminuir a incidência de erros de sintaxe e semântica quando o aluno for desenvolver códigos da linguagem C;
- b) Elevar o conhecimento do aluno em relação às estruturas básicas da linguagem C;
- c) Aumentar a motivação do aluno em relação ao estudo da linguagem de programação C;
- d) Proporcionar ao aluno maior contato com a linguagem C.

A estratégia de ensino será alternar aulas expositivas com aulas de laboratório e uso do jogo. A cada aula no laboratório o aluno irá ter contato com cada elemento dos cenários pertencentes ao jogo. Os cenários presentes no jogo, como salões, subsolo, área aberta, entre outros, foram concebidos de forma que os conceitos abordados partem dos mais básicos e vai avançando a cada novo cenário até chegar a cenários com conceitos mais complexos.

Desta forma, após o aluno vencer uma etapa do jogo, ele irá visualizar na tela assuntos e conceitos que serão abordados em aulas futuras e será incentivado a estudar o assunto para poder progredir no jogo. Assim o sequenciamento do aprendizado será realizado através da abordagem de conceitos de forma gradual no jogo e que seguem o plano de ensino da disciplina.

As tarefas e atividades que serão realizados no jogo envolvem o aluno analisar códigos e escolher códigos sem erro. Desta forma ele poderá atravessar portas, portões, explodir barris sem ser explodido, desabilitar inimigos, receber feedback de erros, assistir mini vídeos com respostas corretas, etc. As habilidades que os alunos irão adquirir são:

- a) Saber escrever um programa em linguagem C com poucos ou nenhum erros de sintaxe e semântica;
- b) Conseguir identificar erros de execução sem necessariamente executar códigos;
- c) Fazer análise de códigos em busca de erros.

A próxima atividade objetiva escolher o gênero do jogo sério. As opções de escolha estão listadas na tabela a seguir:

Tabela 24 - Gêneros de jogos digitais

Gênero	Tipos de atividades e desafios
Jogos de ação	Desafios Físicos
Jogos de estratégia em tempo real	Estratégia, táticas e desafios logísticos
Jogos de RPG	Táticas, logística, experiências e desafios econômicos
Simulações do mundo real (jogos esportivos, simulações de veículos e simulações)	Desafios físicos e táticos
Construção e gestão de jogos	Desafios econômicos e conceituais
Jogos de Aventura	Exploração e desafios de resolução de quebra-cabeças
Jogos de Quebra Cabeças	Lógica e desafios conceituais

O gênero escolhido foi o de jogos de ação e o jogo possui vários desafios e explora recursos que envolvem o jogador em um ambiente de ações constantes. Após a definição do gênero foi executada a atividade definir a história para o jogo. A história do jogo ocorrerá em um ambiente de uma fábrica de produtos químicos abandonada. A figura 38 a seguir mostra um dos cenários desta fábrica concebido para o jogo.



Figura 38 - Fábrica abandonada representada no jogo sério desenvolvido



O personagem principal do jogo, de nome C-GUARD, terá a missão de vencer os desafios colocados em seu caminho e avançar até o fim do jogo em que terá que destruir a fonte de energia que alimenta os robôs inimigos e com isso vencer o jogo. A figura 39 mostra este personagem.

Figura 39 – Personagem do jogo



A missão do personagem é enfrentar os inimigos robôs e vencer os desafios sobre linguagem C. A figura 40 mostra um destes robôs inimigos e um dos desafios de linguagem C presentes no jogo.

Figura 40 - Robô inimigo e um desafio de abrir a porta com o código correto



Durante a trajetória do personagem C-GUARD, vários desafios relacionados com a linguagem C foram colocados para que ele os vença e siga seu caminho até chegar à fonte de energia representada na figura 41.

Figura 41 - Torre com a fonte de energia principal a ser destruída pelo jogador



#### 4.2.2 Atividades de criação e gerenciamento de documentos do projeto

Através da atividade criar um documento com vocabulário compartilhado, foi criado um documento de termos e conhecimentos relacionados com o projeto do jogo sério. O objetivo foi facilitar a comunicação entre especialistas de diferentes formações para que estes termos pudessem ser entendidos por todas as partes interessadas. O vocabulário foi representado textualmente e contém algumas imagens para melhorar o entendimento de alguns itens mais complexos. Este documento foi atualizado durante toda execução da metodologia e pode ser visualizado no apêndice D.

A próxima atividade é a de criar o documento de requisitos do jogo. Este documento contém os tópicos sugeridos na atividade, e como ocorreu em outros documentos ele também foi atualizado durante toda execução da metodologia. Desta forma, foram definidos neste documento a especificação que descreve:

- a) As características do público alvo
- b) O conhecimento relacionado com a área que o jogo sério deve contemplar e como ele vai ser representado;
- c) O ambiente e as configurações a que o jogo se destina;
- d) Os requisitos de sistema: como o hardware e dispositivos que terão que ser utilizados para executar o jogo.
- e) Eventuais restrições que podem se aplicar ao produto final: licenciamento, comercial, de código aberto, idade, etc.
- f) Requisitos técnicos para os aspectos visuais do jogo.
- g) O conteúdo de áudio
- h) Quaisquer outras informações que sejam necessárias para formalizar a área do jogo.

Para uma melhor estruturação do documento de requisitos descrito na atividade, os conteúdos dos tópicos citados anteriormente foram inseridos as seguintes seções adicionadas ao documento de requisitos:

- INTRODUÇÃO
  - Objetivos
  - Convenções, Termos e Abreviações
  - Escopo
- DESCRIÇÃO GERAL
  - Requisitos Funcionais

- Requisitos Não Funcionais

A tabela 23 a seguir indica estes tópicos do documento de requisitos e suas descrições. O documento completo está disponível no apêndice E.

Tabela 25 - Documento de requisitos do jogo

<b>DOCUMENTO DE REQUISITOS</b>	
<b>INTRODUÇÃO</b>	
Este documento contém a especificação de requisitos para o jogo digital: UNLOCK-C, que será utilizado como sistema de apoio ao ensino e aprendizado da linguagem de programação C, para alunos de cursos de computação no ensino superior.	
<b>Objetivos</b>	
Este projeto visa desenvolver um jogo que objetiva auxiliar no processo de ensino e aprendizagem de disciplinas de graduação em computação, que utilizam a linguagem de programação C como suporte. O jogador experimentará fases que irão abordar conceitos acerca da sintaxe, semântica e lógica de programação utilizando a linguagem C.	
<b>Convenções, Termos e Abreviações.</b>	
<b>Termo</b>	<b>Significado</b>
RNF	Requisitos Não Funcionais: são os requisitos relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenibilidade e tecnologias envolvidas.
RF	São os requisitos relacionados às funcionalidades do sistema.
<b>Escopo</b>	
O jogo digital UNLOCK-C é um jogo de ação e aventura para plataforma PC e destinado a alunos universitários dos cursos de computação. O jogo possui a necessidade de instalação prévia, e deve ser jogado por um único jogador sem necessidade de conexão à internet. UNLOCK-C é uma ferramenta para apoiar e contribuir no processo de ensino e aprendizagem dos conceitos envolvidos com a linguagem de programação C. O jogo deve fornecer uma interface 3D aprimorada, intuitiva, prazerosa e divertida que irá envolver o jogador com desafios relacionados com programação.	
Utilizando o jogo, o professor poderá permitir ao seu aluno explorar elementos relacionados com a linguagem C, principalmente os erros frequentes de sintaxe e semântica. Ao jogar o jogo o aluno poderá percorrer um ambiente 3D com estilo de jogo de tiro em primeira pessoa(FPS) e decidir os caminhos que deseja seguir de acordo com desafios de programação apresentados a ele.	
<b>DESCRIÇÃO GERAL</b>	
As principais funções associadas a este produto serão descritas nesta seção.	
<b>Requisitos funcionais</b>	
RF1.O sistema deve possibilitar que o jogador pause o jogo.	
RF2. O sistema deve possibilitar que o jogador reinicie o jogo após ter pausado.	
RF3. O sistema deve possibilitar que o jogador saia do jogo	
RF4. O sistema deve possibilitar que o jogador mude do modo em primeira pessoa para o modo em terceira pessoa.	
RF4. O sistema deve permitir ao jogador obter feedbacks sobre suas ações no jogo.	
RF5. O sistema deve fornecer uma forma de registrar erros que o jogador cometeu durante o jogo.	
RF6. O sistema deve gerar inimigos que atacam o jogador em momentos determinados.	
RF7. O sistema deve emitir mensagens de feedback em texto e vídeo para o jogador.	
RF8. O sistema deve disponibilizar ao jogador possibilidade de recarregar munições de suas armas.	
RF9. O sistema deve disponibilizar ao jogador possibilidade de alternar entre armas disponíveis.	
RF10. O sistema deve ser capaz de exibir informações na tela sobre a saúde do jogador, a quantidade de munição das armas e qual a sua arma atual.	
RF11. O sistema deve mostrar na tela uma mensagem caso o jogador seja abatido e indicar o nome do inimigo que o abateu.	

RF12. O sistema deve fornecer ao jogador a possibilidade de recarregar suas armas através de itens coletados durante o jogo.

RF13. O sistema deve fornecer ao jogador a possibilidade de aumentar sua saúde através de itens coletados durante o jogo.

RF14. O sistema deve gerenciar o número de vezes que o jogador morreu.

RF15. O sistema deve ser capaz de mostrar a pontuação(score) e número de vezes que o jogador morreu no jogo.

RF16. O sistema deve ser capaz de indicar a quantidade de tempo que resta para o jogador conseguir vencer o jogo.

RF17. O sistema deve mostrar ao final do jogo informações sobre a partida.

#### Requisitos não funcionais

RNF1. O sistema deverá ser executado em Plataforma PC com no mínimo sistema operacional Windows XP SP3 2.0 GHz velocidade processador, 2 GB de memória RAM e placa de vídeo com suporte a modelo de sombras 3.0

RNF2. O sistema deverá ser de fácil utilização com controles de movimentação e interação intuitivos

RNF3. O sistema deve apresentar adequação de conteúdo, clareza no conteúdo, aprendizagem com encadeamento de ideias e relação entre a teoria e a prática.

RNF4. O sistema deve possuir interatividade, presença de elementos gráficos e atratividade.

RNF5. O sistema deve possuir interface gráfica 3D.

RNF6. O sistema deve ser desenvolvido na categoria tiro em primeira pessoa(FPS)

A próxima atividade executada foi a de criar o documento de design e conceito do jogo. A primeira tarefa desta atividade é a de definir o conceito do jogo.

Este conceito foi desenvolvido e é exibido nos tópicos da tabela 24 a seguir:

Tabela 26 - Conceito do jogo

<b>CONCEITO DO JOGO</b>	
<b>Nome do Jogo:</b>	UNLOCK-C
<b>Plataforma:</b>	PC/Windows
<b>Jogadores:</b>	Single Player;
<b>Gênero:</b>	atirador em primeira pessoa FPS (First Person Shooter);;
<b>High Concept:</b>	O jogo consiste em vencer desafios relacionados com programação utilizando a linguagem C. O jogador terá que analisar códigos e escolher o caminho correto para passar ao próximo desafio e ambiente. O jogador terá que enfrentar ao longo de sua jornada inimigos robôs(bots), posicionados em diversas partes do cenário, bombas que explodem no caminho, locais fatais, torres que atiram, nave inimiga que atira e demais itens.
<b>Objetivo:</b>	Para vencer o jogo o jogador deve ultrapassar todos os desafios, conseguir abrir todas as portas e alcançar uma torre com uma fonte de energia que alimenta os robôs inimigos e destruí-la.
<b>Recursos:</b>	O jogador poderá atirar através de três tipos de armas, recarregar armas, obter mais saúde, sofrer danos de tiros, sofrer danos de choque, explodir barris com tiros, movimentar barris com tiros, modificar o modo do jogo de primeira pessoa para terceira pessoa. O jogo irá evoluir de um ponto inicial de uma fábrica de produtos químicos abandonada e o jogador vai passar por várias salas e prédios desta fábrica até o final do jogo. Diversos sons estarão presentes no jogo, desde som ambiente, sons de vento, sons de tiros, sons de armas, de passos do personagem e inimigos. O jogo contará com simulação da luz do sol e iluminação com lâmpadas comuns e direcionais.

Após ser aprovado pelo designer de jogos, o conceito do jogo começou a ser desenvolvido o documento de design do game. Os tópicos e texto deste documento estão exibidos na tabela 25 e também estão disponíveis no apêndice F.

Tabela 27 - Documento de design do jogo

<p><b>Eventos anteriores</b> C-Guard é um especialista em linguagem de programação C e robôs e por isso, ele foi o escolhido para ser enviado a fábrica. Além disso, o local da fábrica fica próximo a uma comunidade onde vive a mãe de C-Guard e sua família. Caso ele falhe a vida de seus familiares correrá risco, pois caso os robôs organizem um exército irão sair da fábrica e eliminar todos no caminho.</p> <p><b>Principais Jogadores</b> <b>C-Guard</b> - É o personagem principal e seu objetivo é entrar na fábrica, abater os robôs e destruir a fonte de energia que os alimenta. Ele terá a sua disposição três tipos de armas e munição para enfrentar os desafios. O personagem terá que possuir e também adquirir ao longo de sua trajetória no jogo conhecimentos sobre linguagem C, para ultrapassar os desafios sobre a linguagem colocados no jogo. <b>BOT Inimigo</b> - O robô inimigo é um agente movido a inteligência artificial programado para abater qualquer intruso. Os robôs terão armas diferentes e serão mais fortes ou mais fracos de acordo com as armas e inteligência artificial programada.</p> <p><b>OBJETOS ESSENCIAIS DO GAME</b> <b>Personagens</b></p> <p><b>C-GUARD</b> É o atirador que utiliza armas e é o alterego do jogador. Este jogador possui saúde que pode ser restabelecida com recarregadores de saúde coletados no chão. O C-Guard pode alternar entre três tipos de armas e recarregá-las coletando munição espalhada pelo cenário. Ele pode sofrer danos com os tiros dos robôs inimigos, com bombas que explodem em alguns locais de passagem, com choques se cair em um curto circuito, se entrar em locais errados e fatais.</p> <p><b>Robô(Bot) inimigo</b> O bot inimigo é um agente de inteligência artificial que possui armas, atira e persegue o personagem principal C-GUARD. Ele pode sofrer danos e morrer ao ser atingido pelos tiros de C-GUARD.</p> <p><b>Armas</b> <b>LinkGun</b> Arma padrão dos personagens e alguns robôs. Tem poder de fogo limitado, mas efetivo. Dispara tiros e também pode disparar um tiro longo em forma de raio que consome muita munição, mas causa mais danos ao alvo. Possui boa quantidade de munição e pode ser recarregada através de coleta de munição.</p> <p><b>ShockRifle</b> Arma mais poderosa que a linkgun e gera lasers azuis com alto poder de dano ao bot inimigo ou ao jogador. Pode disparar também um laser contínuo com maior poder de fogo.</p> <p><b>Rocket Laucher</b> Arma mais potente do jogo que dispara mísseis que explodem ao colidirem com o alvo. Possui munição limitada e pode ser recarregada através de coleta de munição.</p> <p><b>Nave Inimiga</b> Nave que dispara tiros via dois canhões presentes nas asas do jogador. A nave se locomove após um robô entrar e assumir o controle. A nave recebe danos de tiros que devem ser em grande volume para conseguir destruí-la</p> <p><b>Veículo Scorpion</b> Veículo que pode ser utilizado pelo jogador e que atira dois tiros longos via canhão localizado em seu teto.</p> <p><b>Torres que atiram</b> É uma arma imóvel que atira contra o jogador e só pode ser destruída se receber diversos tiros do personagem C-GUARD.</p> <p><b>Bombas</b> As bombas irão detonar após o jogador ter chegado em determinado local e ficarão explodindo aleatoriamente causando dano ao jogador.</p>
---

**Objetos****Base de captura de armas**

Imagem: Base luminosa com arma em rotação suspensa por uma luz

Onde pode ser encontrado: Em diversas partes do cenário

Comportamento: Quando o jogador passa pela base recebe a arma correspondente e sua munição

Tabela 25 – Documento de design do jogo. (Continuação)

**Base de captura de armaduras**

Imagem: Base com armadura um pouco acima, em rotação e suspensa pelo ar.

Onde pode ser encontrado: Em diversas partes do cenário

Comportamento: Quando o jogador captura a armadura fica com mais poder e sua saúde será menos atingida quando for atingido por tiros ou qualquer elemento que reduza sua vida.

**Recarregador de Munição**

Imagem: Objeto luminoso em formato diferenciado para cada um dos três tipos de armas

Onde pode ser encontrado: Em diversas partes do cenário

Comportamento: Quando o jogador colidir a arma correspondente terá sua munição recarregada

**Recarregador de Saúde**

Imagem: Base de metal que sustenta uma esfera iluminada de diversos tamanhos a depender do ganho de saúde proporcionado

Onde pode ser encontrado: Em diversas partes do jogo.

Comportamento: Quando o jogador colidir sua saúde será revigorada em X pontos a depender do tamanho do recarregador de saúde.

**Barril**

Imagem: Barril com sinais de corrosão

Onde pode ser encontrado: Em algumas partes do cenário, principalmente nas primeiras salas.

Comportamento: Explode se atingido por tiros do jogador ou robôs inimigos. Caso o jogador esteja próximo do barril ele irá ser abatido.

**Teleporter**

Imagem: Base que sustenta uma elipse com áurea animada que mostra no centro o destino de ida

Onde pode ser encontrado: Em poucas partes do cenário.

Comportamento: Quando o jogador colidir será teletransportado para o destino estabelecido para o teleporter

**Portas automáticas**

Imagem: Duas portas de ferro com corrosão

Onde pode ser encontrado: Nas saídas das salas da fábrica

Comportamento: Quando o jogador se aproximar elas abrem de forma animada e com som de abertura. Quando o jogador se afasta elas fecham de forma animada e com som de fechamento.

**Jump pad(Impulsionadores)**

Imagem: Base luminosa com feixes de luz quadrados em suspensão

Onde pode ser encontrado: em poucos prédios do cenário

Comportamento: Quando o jogador fica em cima ele é impulsionado com um salto e vai parar no ponto de destino especificado para o jump pad.

**Triggers(Gatilhos)**

Imagem: é um elemento invisível

Onde pode ser encontrado: Na maior parte de todos os ambientes do cenário

Comportamento: Dispara eventos indicando a programação do jogo que o jogador tocou ou passou por aquele local. Diversos eventos diferentes são acionados por diversos tipos de triggers.

**Checkpoint (ponto de checagem)**

Imagem: é um elemento invisível

Onde Pode ser encontrado: Em alguns locais do jogo.

Comportamento: é um elemento invisível criado via gatilho(trigger) que estabelece um ponto de checagem. Caso o jogador seja abatido o jogo irá recomeçar deste ponto.

**Base com mini fonte de energia**

Imagem: Base elíptica com partícula de luz animada no centro e suspensa pelo ar

Onde pode ser encontrado: No subsolo da primeira sala do jogo

Comportamento: Ao ser capturada e tendo o jogador eliminado 3 inimigos, uma porta do primeiro salão principal será aberta.

**Fonte de energia dos bots inimigos**

Imagem: Uma esfera vazada formada por anéis superpostos e, com efeito, animado luminoso no centro indicando energia.

Onde pode ser encontrado: No teto do prédio que é o final do jogo

Comportamento: Quando o jogador atira é destruída e uma animação de robôs inimigos gigantes explodindo e caindo para trás é apresentada. Um texto indicando que o jogador venceu o jogo aparece e começa um passeio virtual por todo o cenário do jogo.

Tabela 25 – Documento de design do jogo. (Continuação)

### CONFLITOS E SOLUÇÕES

Existem alguns sistemas de conflitos no jogo.

#### **Portas que se abrem e explodem o jogador**

Existem diversas portas que se o jogador ultrapassar ele será explodido. Para que isso não ocorra ele deverá escolher a porta livre de erros de programação e com o conceito correto sobre a linguagem C.

#### **Barril que explode**

Ao longo do caminho existirão barris fechando a passagem do jogador. Se o jogador atirar o barril explode. Caso ele esteja próximo ele irá morrer.

#### **Barril com física**

Este barril não explode e será movido com a força do impacto de tiros. Em alguns momentos códigos indicando atire aqui estarão presos a barris que explodem e de física. O jogador só sobreviverá se atingir o código correto anexado ao barril de física. Caso contrário toda a área ao seu redor será atingida pela explosão dos barris com códigos incorretos.

#### **Nave inimiga**

Esta nave irá atacar o jogador com tiros em dois canhões. Para o jogador vencer terá que conseguir atirar em uma quantidade alta. Ele poderá coletar armaduras e munição próxima a nave para aumentar suas chances de sucesso.

#### **Local Fatal**

Local que existe uma killzone(zona de morte). Caso o jogador faça uma escolha errada, ele será imediatamente abatido e o jogo recomeça a partir do último checkpoint.

### Gerador de energia em curto circuito

Um gerador de energia que se o jogador cair em cima dele será eletrocutado e morrerá instantaneamente. Ele precisa garantir que não seja empurrado em buracos que possuem o gerador no centro. Ele poderá ser empurrado por disparos feitos por bots inimigos ou ao escorregar e cair durante o jogo.

#### **Porta que abre em condições especiais**

Esta porta só irá abrir se o jogador conseguir abater 3 inimigos e depois capturar uma mini fonte de energia no chão.

#### **Queda de local muito alto**

Se o jogador cair de local muito alto ele irá morrer e deve evitar isso.

#### **Bots inimigos**

Vários robôs inimigos serão espalhados pelo cenário com diversas armas. O jogador terá que atirar neles até reduzir a saúde do robô a zero. O jogador deve evitar ser atingido pelos tiros dos bots pois, se sua saúde chegar a zero ele é eliminado.

#### **Torres que atiram**

Torres que atiram e tiram saúde do jogador. Ele deverá conseguir chegar até elas após passar por campo minado por bombas e atirar nas torres para destruí-las.

### INTELIGÊNCIA ARTIFICIAL

UNLOCK-C terá inteligência artificial aprimorada. Os principais comportamentos são o seguinte:

#### **Bots inimigos**

Os robôs serão gerados em locais de desova estabelecidos e irão percorrer caminhos criados e estabelecidos para os mesmos. Os bots irão perseguir o jogador de forma agressiva e até mesmo pular obstáculo para conseguir atirar no jogador

#### **Nave inimiga**

A nave inimiga será acionada após o jogador ativar um trigger que fará com que um bot entre na nave e comece a atirar no jogador. A nave irá se movimentar e será agressiva em relação à quantidade de tiros.

### FLUXO DO JOGO

#### **Cinemática de apresentação**

O jogo começa com uma cena que irá mostrar o jogador surgindo na fábrica, explosões e um passeio dramático pelo cenário.



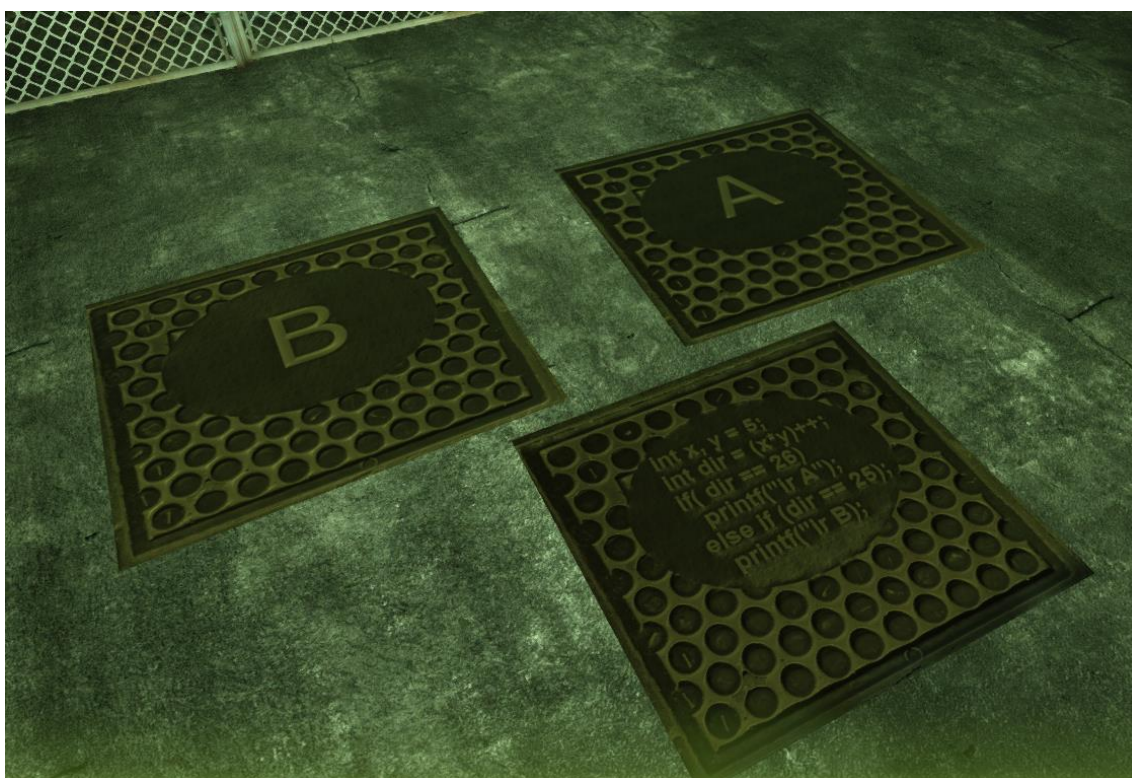
Tabela 25 – Documento de design do jogo. (Continuação)

<p><b>Nível salão principal</b> O jogador irá começa a partir desta sala e terá pela frente portas a serem abertas, robôs a serem destruídos e desafios a serem vencidos. As portas com códigos C serão abertas pelo jogador que irá explodir caso abra a porta errada. Para sair do salão principal o jogador deverá conseguir passar pelas portas e vencer 2 inimigos no primeiro andar e 3 inimigos que estão no subsolo. Neste salão o jogador também terá a interação com 2 inimigos que estão no fosso de água um de cada lado.</p> <p><b>Nível Sala de máquinas</b> A sala de máquinas existe um robô inimigo com arma a laser de difícil atingimento. O jogador deverá abater este inimigo e poderá contar com munição e arma extra escondida atrás de cilindros grandes de produtos químicos.</p> <p><b>Nível sala de resfriamento.</b> Caso o jogador opte por entrar na sala de máquinas e pegar o elevador ele vai até esta sala de resfriamento. Lá existirão três inimigos e uma saída para uma ponte que fica acima da sala de máquinas e leva a sala cilindro. Uma animação dramática será acionada com um inimigo aparecendo e pulando em frente ao jogador perto da porta de saída.</p> <p><b>Nível ponte suspensa 1</b> Neste nível as três torres começarão a atirar no jogador que terá que correr até as torres do outro lado da ponte e destruí-las. Para isso poderá contar com bases de saúde e armadura ao longo da ponte, para que aumente as suas chances de sucesso. Quando o jogador conseguir destruir as torres uma porta se abrirá para o campo aberto e esta abertura será mostrada via cinemática com passeio da câmera pelo cenário da sala cilindro. Aqui teremos um checkpoint.</p> <p><b>Nível ponte suspensa 2</b> Neste nível o jogador irá enfrentar campo minado em que bombas começarão a explodir aleatoriamente. Ele terá que ultrapassar correndo esta ponte até chegar ao outro lado.</p>
<p><b>Nível sala cilindro.</b> Nesta sala o jogador terá que enfrentar três inimigos ao mesmo tempo e terá que derrotá-los. Ele contará com base de saúde e armaduras para tentar fazer isso.</p> <p><b>Nível campo aberto</b> Ao entrar neste nível o jogador irá estabelecer um checkpoint. Neste ambiente irá aparecer a sua frente uma grande quantidade de itens de armadura, saúde e armas que podem ser coletadas. Assim que entrar o jogador verá uma animação de um bot inimigo entrando em uma nave robô inimiga que começará a atirar nele. Ele deverá atirar muitas vezes até conseguir destruir a nave. Neste mesmo campo aberto após a nave existem 5 inimigos guardando a porta do prédio da fonte de energia dos bots. O jogador poderá entrar em um veículo jipe-tanque atirando nos inimigos robôs. O jogador deve evitar ser atingido por tiros, pois se o veículo explodir com ele dentro ele morre.</p> <p><b>Nível prédio da fonte de energia dos bots: térreo e andares superiores</b> Este prédio possui uma porta que o jogador precisa adivinhar que é necessário atirar nela para destruí-la. Assim que entra existe outro checkpoint. Uma rampa e um passadiço envolvem um gerador de energia que está em curto circuito e um robô inimigo patrulha o local. O jogador deverá eliminar o robô e evitar tocar no gerador em curto, pois senão ele morre. Depois da batalha com o robô o jogador irá saltar para o andar acima através de um impulsionador(jump pad). Lá existirá um robô inimigo que ele deve eliminar e também evitar cair no vão central para não ser atingido pelo gerador em curto embaixo e morrer eletrocutado. Após eliminar o robô o jogador irá novamente utilizar um jump pad e para subir ao próximo andar. Lá existirá uma rampa que dá acesso a cobertura decorada com flamulas suspensas em mastros. Esta rampa será protegida por outro bot e o jogador deve eliminá-lo para conseguir chegar a cobertura.</p> <p><b>Nível prédio da fonte de energia dos bots: cobertura com flamulas</b> Neste nível três bots inimigos aparecem atirando imediatamente após a chegada do jogador. O jogador terá que eliminar estes três inimigos para que uma porta que dá acesso a um elevador seja aberta. Este elevador dá acesso ao teto onde fica a fonte principal de energia que alimenta os robôs inimigos (bots).</p> <p><b>Nível prédio da fonte de energia dos bots: Teto da fonte de energia</b> Este é o local final do jogo. O jogador receberá um feedback escrito na tela indicando que precisará atirar na fonte de energia. Após atirar o jogo acaba e ele vence. Uma cinemática com cena que irá passear por todos os cenários anteriores será mostrada e um texto indicando que o jogador venceu.</p>
<p><b>CONTROLES</b> O jogador irá movimentar o personagem C-GUARD através do teclado com as clássicas teclas WASD de movimentação de jogos FPS. Ele irá atirar com as armas através dos botões esquerdo (tiro normal) e direito do mouse (tiro longo potente). Além disso, também poderá usar o mouse para posicionar sua visão através de rotações e translações. Em adição o jogador poderá teclar a tecla tab e digitar BehindView para alternar para o modo de visão em terceira pessoa se preferir.</p>

### 4.2.3 Atividades de concepção de elementos do jogo

Diversos elementos foram criados para o jogo baseados em desafios acerca da linguagem de programação C, e alinhados com o estilo de jogo FPS adotado. Ao criar elementos e características do jogo, diversos itens foram criados e avaliados em relação à interface do jogo. Foram identificados itens que poderiam comprometer a usabilidade, causar confusão ou que não estavam permitindo ao jogador descobrir o objetivo do item no fluxo do jogo. Como exemplo a figura 42 mostra um código escrito no chão que não estava sendo observado pelos jogadores durante o jogo.

Figura 42 - Código no chão que não estava sendo corretamente visualizado



Desta forma uma atenção especial for fornecida aos feedbacks ao jogador e estes aspectos foram tratados na execução da atividade inserir elementos motivacionais, de decisão, armadilhas e feedbacks com o resultado exibido na figura 43.

Figura 43 - Tratamento visual de elemento que não estava sendo corretamente percebido pelo jogador



Os questionários respondidos pelos alunos e descritos anteriormente ajudaram na definição das características do jogo. Os resultados revelaram uma preferência dos alunos por jogos em primeira pessoa e jogos 3D. Assim os aspectos que norteiam um jogo 3D e FPS foram considerados no jogo. Como consequência foi estabelecido que o jogador terá de vencer desafios e decifrar problemas envolvendo linguagem C, além de precisar desabilitar robôs atirando nos mesmos.

Outra característica importante é que o jogador somente poderá avançar se escolher as opções corretas das situações problemas relacionadas com os conceitos da linguagem C. Em caso de erros na escolha o jogador morre e serão exibidos feedbacks textuais ou em vídeos para que o mesmo perceba onde errou e aprenda o conceito. Esta característica permite que o jogador somente avance para a próxima fase após ter avaliado seus erros e acertos e compreendido os conceitos trabalhados em determinada fase do jogo.

Na execução desta atividade a experiência de professores foi capturada para revelar os elementos que normalmente ocasionam confusão entre os alunos. Estes elementos foram transformados em pistas falsas para deixar o jogo ainda mais desafiador e para trabalhar os conceitos indicados pelos especialistas de domínio (professores), em que os alunos no geral possuem dificuldades. Potenciais distratores foram listados e inseridos.

Muitos elementos do jogo foram definidos aproveitando elementos do motor de jogo utilizado UDK. Foi realizada uma identificação de que elementos prontos, que a UDK poderia oferecer ao desenvolvimento de um jogo. A seguir são listados estes elementos que foram escolhidos como candidatos a compor o jogo.

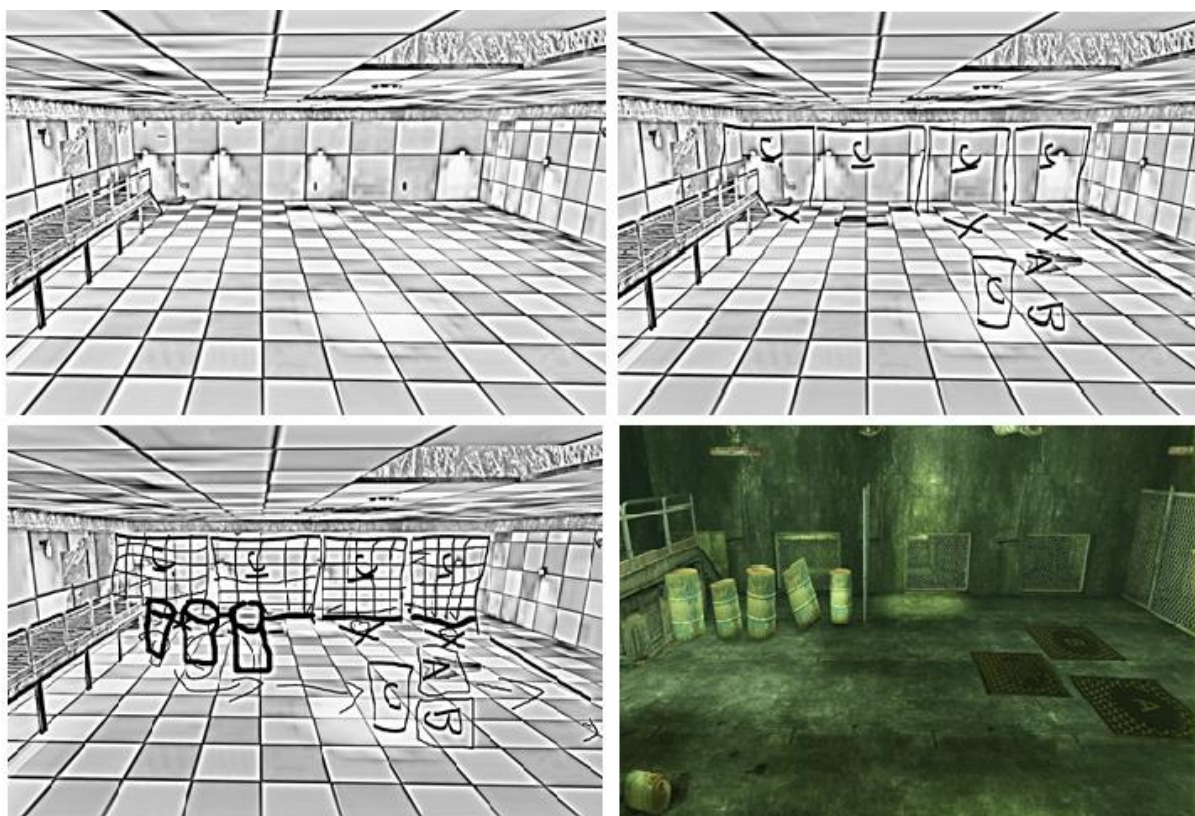
- a) Portões com códigos pendurados com erros de sintaxe;
- b) Códigos desenhados no chão em forma de bueiro com fluxo do caminho correto e incorreto;
- c) Minas que explodem, caso o jogador erre o desafio do código;
- d) Locais fatais que o jogador irá entrar caso erre a análise do código;
- e) Vídeos explicativos de itens conceituais da linguagem C, acionados quando o jogador erra algum desafio;
- f) Feedbacks constantes em formato texto e sons sobre êxito ou fracasso do jogador em relação aos desafios acerca de códigos da linguagem C;
- g) Nave inimiga que atira no jogador;
- h) Jump-pads que impulsionam o jogador fazendo-o voar pelo cenário, caindo de forma segura em determinado local;
- i) Pick-ups de armas;
- j) Pick-ups de energia para revitalizar vida do jogador;
- k) Barris que explodem e matam jogador se ele estiver próximo da explosão;
- l) Névoa;
- m) Sistema de partículas de fumaça, fogo, água, folhas caindo;
- n) Munição para as armas;
- o) Animações de portas abrindo e fechando;
- p) Animações com câmeras que passeiam pelo cenário;
- q) Diversas cinemáticas sofisticadas;
- r) Sons diversos;
- s) Luz do sol, céu, relevo com montanhas, lago, árvores, pássaros;
- t) Nós de caminhos (path nodes) para trajetória dos agentes(robôs) de inteligência artificial;
- u) Materiais, texturas;
- v) Triggers(gatilhos) que são disparados pelo jogador, por tiros, etc; Teletransporte;
- w) Elevador;
- x) Jipe que atira;
- y) Simulação de tecidos em forma de bandeira;
- z) Simulação de vento.

A próxima atividade executada desenhar storyboards necessita de um profissional com habilidades em desenho. Não existia a disponibilidade deste profissional no desenvolvimento do jogo sério. Apesar disso, foi possível criar telas

rascunho que funcionaram como storyboards do jogo. O motor de jogo UDK permitiu criar ambientes ainda sem muita estruturação, mas que permitiu montar de forma rápida as ideias e analisar elementos importantes, interfaces e interações a serem exibidas ao usuário.

O cenário foi montado a partir das ideias da fábrica de produtos químicos abandonada já estabelecida na história do jogo. O objetivo foi deixar os elementos 2D e 3D do jogo o mais intuitivos possível. A figura 44 mostra alguns storyboards do jogo e também o resultado final produzido na fase de produção já com interface aprimorada.

Figura 44 - Storyboards desenvolvidos para o jogo e o resultado final


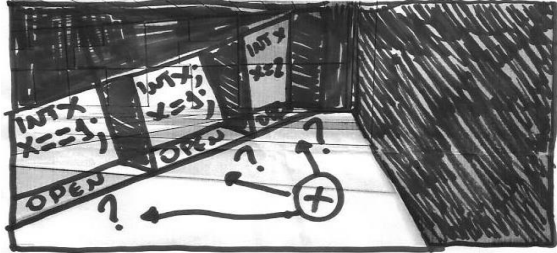
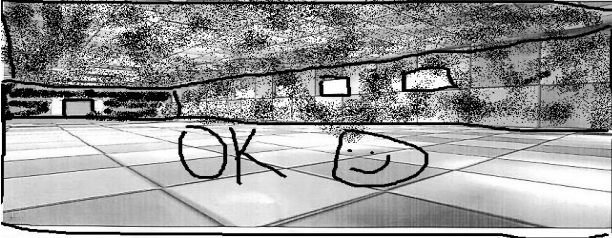




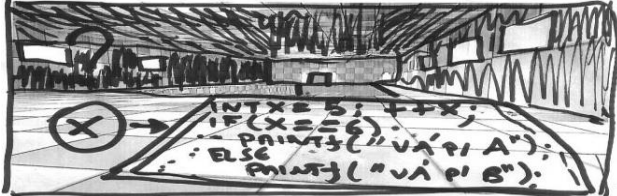
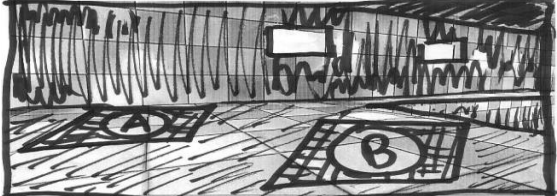
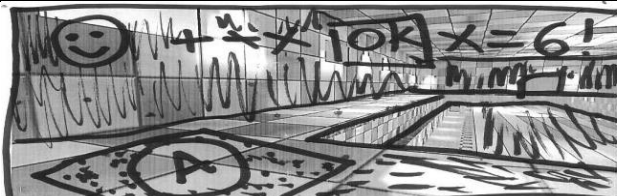
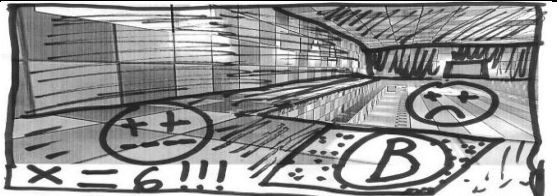


A próxima atividade foi criar e melhorar o script do jogo. Uma sequência de passos (script), foi criada com recursos de arte provisórios (mock-ups) e utilizou como artefatos de entrada o documento de requisitos e o documento de design. Foram criados e melhorados os storyboards de simulação do jogo e verificado se ele fornece o entretenimento adequado e valor educativo. Essa verificação pode ser realizada através das respostas as seguintes perguntas:

- a) A forma de pensar e resolver problemas pelo jogador durante o jogo é semelhante à abordagem utilizada pelos especialistas de domínio no ensino da linguagem C?
- b) Os alunos possuem uma compreensão clara dos principais objetivos do jogo e de como ter sucesso?
- c) Os alunos percebem quais são as regras gerais e o que deve e não deve fazer para ter sucesso?
- d) Os alunos se sentem situados em relação ao jogo?
- e) Durante o jogo o aluno é confrontado com diferentes situações ou problemas que precisam ser resolvidos?

Posteriormente foram desenvolvidos os storyboards de simulação do jogo. Estes storyboards foram utilizados e mostrados a alguns alunos para que as perguntas anteriormente formuladas pudessem ser avaliadas. Mas uma vez o motor de jogos UDK auxiliou no processo, pois, permitiu a construção da geometria do jogo e alguns elementos rascunho. Foram exibidos aos alunos cada um dos desafios e ambientes inicialmente projetados para o jogo. Ações eram requeridas e os resultados de sucesso ou fracasso eram mostrados. Estes storyboards com a simulação do jogo estão exibidos na tabela 26 a seguir:

Tabela 28- Simulação do jogo através de storyboards

<b>ESCOLHA UMA PORTA</b>	
	
<b>O RESULTADO DE SUA AÇÃO</b>	
<b>SUCESSO</b>	<b>FRACASSO</b>
	
<b>ATIRE EM UM BARRIL</b>	
	
<b>SUCESSO</b>	<b>FRACASSO</b>
	
<b>ANALISE O CÓDIGO E TOME SUA DECISÃO</b>	
	
<b>SUCESSO</b>	<b>FRACASSO</b>
	

Após a construção da simulação do jogo via storyboards, foi efetuada uma simulação mediada por um especialista de domínio (professor) com alguns alunos para este fluxo de acontecimentos e foram avaliadas as perguntas formuladas, obtendo-se as seguintes respostas presentes na tabela 27.

Tabela 29 - Respostas as perguntas acerca da simulação do jogo

<p><b>A forma de pensar e resolver problemas pelo jogador durante o jogo é semelhante à abordagem utilizada pelos especialistas de domínio no ensino da linguagem C?</b></p> <p>Sim. Os alunos erraram no início alguns aspectos relacionados a construção de códigos, sintaxe e semântica na linguagem C. Apesar do jogo não ter como objetivo ser um elemento completo de ensino, a simulação mostrou que o apoio ao ensino estava ocorrendo. Este fato foi observado quando alunos perceberam como o jogo funcionava e conseguiram seguir o fluxo entendendo seus objetivos. Apesar de ter sido uma simulação em papel com elementos rudimentares foi percebido que a ideia do jogo poderia seguir em frente, ainda sendo necessário, no entanto muito mais testes em outros níveis de abstração (como protótipos, etc)</p> <p><b>Os alunos possuem uma compreensão clara dos principais objetivos do jogo e de como ter sucesso?</b></p> <p>Inicialmente os alunos não entendem do que se trata o jogo. Contudo após experimentarem a uma simulação de escolha de portas para abrir, eles foram avisados se passariam a porta ou morreriam. Neste momento um feedback foi fornecido indicando os motivos do aluno ter passado ou morrido. Eles foram informados da falta de ponto e vírgula em código, de palavras dentro de variáveis que só cabem um caractere, etc.</p> <p><b>Os alunos percebem quais são as regras gerais e o que deve e não deve fazer para ter sucesso?</b></p> <p>Inicialmente não, mas com o andamento da simulação as regras foram ficando claras à medida que os alunos cometeram erros. Isso foi importante, pois o aprendizado por erros é um dos pontos chave do aprendizado por jogos conforme já relatado no capítulo 2 desta dissertação.</p> <p><b>Os alunos se sentem situados em relação ao jogo?</b></p> <p>Sim. Eles perceberam que necessitariam exercitar seus conhecimentos acerca da linguagem de programação C.</p> <p><b>Durante o jogo o aluno é confrontado com diferentes situações ou problemas que precisam ser resolvidos?</b></p> <p>Sim. Diferentes elementos foram colocados para o aluno durante a simulação. Neste caso foram três. Portas com código pendurados que se abrem quando o jogador se aproxima barris que devem receber tiros, mensagens de códigos no chão com decisão de que caminho seguir. Como se tratava de uma simulação ainda não estavam presentes diversos outros elementos criados no decorrer do ciclo de desenvolvimento do jogo.</p>
---

### 4.3 ATIVIDADES DE PRODUÇÃO

A partir desta fase o jogo começa a ser desenvolvido e as atividades estão mais relacionadas em construir, de fato, os elementos planejados e estruturados na fase de pré-produção. A figura 45 a seguir indica as atividades que foram executadas nesta fase.

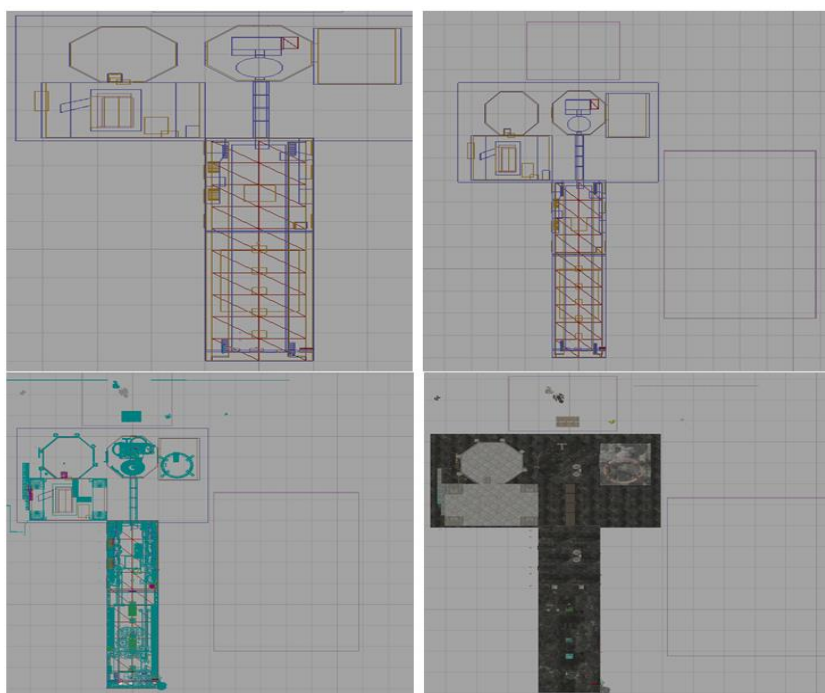


Figura 45 - Atividades da fase de produção



Atividades de projeto e inserção de elementos do jogo. A descrição de alto nível do ambiente projetado foi traduzida para elementos do jogo. A ferramenta UDK mais uma vez foi utilizada para desenvolver o mundo virtual do jogo. Inicialmente foi projetado o mapa do jogo, depois cada local do mapa foi sendo construído. O resultado é mostrado na figura 46 a seguir.

Figura 46 - Construção do mundo virtual



Notadamente ainda era necessário aplicar todos os elementos de textura, materiais, efeitos, sons, e objetos de decoração dos ambientes no mundo virtual. Neste ponto da execução da metodologia começou a se inserir elementos motivacionais, de decisão, armadilhas e feedbacks. Nesta atividade foram criados ou reutilizados gráficos 3D atraentes, para contribuir no reforço a aprendizagem e criar uma experiência imersiva do jogador, no ambiente do jogo.

Os elementos motivacionais foram criados aproveitando itens e recursos presentes no motor de jogo UDK. Estes elementos motivacionais estão listados na tabela 28 a seguir.

Tabela 30 - Lista de elementos motivacionais inseridos no jogo

<b>ELEMENTOS MOTIVACIONAIS</b>
<ul style="list-style-type: none"> <li>• Sons com frases divertidas de efeito indicando um acerto do jogador</li> <li>• Explosões que ocorrem e fazem o jogador morrer caso erre a interpretação de códigos C ou caso não consiga interpretar e vencer os desafios e que estimulam o jogador a querer vencer.</li> <li>• Um ambiente de jogo com interface 3D aprimorada com elementos de jogos no estilo FPS (inimigos, armas, bombas, animação, fogo, fumaça, névoa, munição, nave inimiga, tele transporte, zonas fatais, luz do sol, luz ambiente, sons ambiente, sons de objetos e tudo mais que promove imersão do jogador dentro do jogo.)</li> <li>• Um objetivo final de conseguir chegar até o teto do prédio onde reside o sistema de alimentação de energia dos robôs inimigos e destruí-lo.</li> <li>• Objetivos intermediários de conseguir sair de uma sala da fábrica após vencer desafios.</li> <li>• Conquistas pontos de checagem (check points) após avançar no jogo e não precisar mais voltar para o ponto de partida inicial do jogo toda vez que morrer.</li> <li>• Sons e um texto escrito (double kill) em inglês do próprio motor UDK são ouvidos. Caso o jogador consiga eliminar dois inimigos com um único tiro.</li> </ul>

Conforme orientação de uma tarefa da atividade foram aplicadas técnicas para aumentar o engajamento e a motivação do estudante. Os elementos de decisão foram inseridos no jogo aproveitando estruturas de decisão da própria linguagem de programação C. Comandos se então ( if-else) e escolha caso ( switch-case) foram utilizados para garantir a experiência de tomada de decisão por parte do jogador. Como exemplo estão listados a seguir alguns elementos de tomada de decisão:

- a) Alguns barris receberam etiqueta com código C, indicando uma decisão atire aqui. No entanto, somente um barril com a decisão correta não explode. Caso o jogador tome a decisão errada o barril que ele atirou explode causando sua morte.

- b) Uma expressão matemática que aborda a precedência de operadores deve ser realizada pelo jogador. O resultado é então selecionado pela estrutura switch-case. O jogador deve ir para o teletransporte com o número que corresponde ao resultado correto ou morrerá.

Diversas outras estruturas de decisão foram criadas ao longo do jogo de forma análoga mudando apenas o contexto, o código e objetos envolvidos.

Em relação às armadilhas, os erros comuns dos alunos que foram identificados na seção 4.2.1, foram gradativamente sendo inseridos no jogo. Os erros que obtiveram maior porcentagem na pesquisa com os alunos foram os mais explorados. O momento em que cada erro aparece no jogo foi definido baseado na ordem dos assuntos e conceitos presentes no plano de ensino da disciplina específica, de ensino de programação utilizando a linguagem de programação C.

Outra parte importante foram as definições dos feedbacks. Proporcionar ao jogador (aluno) ser informado em quais itens de cada conceito ele errou foi um dos pontos chaves para tentar garantir o apoio à aprendizagem deste aluno. Os erros de maior incidência relatados pelos alunos foram considerados os aspectos de maior importância da linguagem C. Assim estes aspectos foram os que receberam maior atenção no tratamento dos feedbacks ao aluno. Para fornecer feedbacks ao jogador foram utilizados recursos de áudio, texto, imagem, câmera e vídeo. Como exemplo de feedbacks:

- a) Caso o jogador abra um portão errado e morra o jogo congela e um vídeo é mostrado indicando o código correto, os motivos e em quais itens existe o erro.
- b) Outros feedbacks aparecem sob forma de texto indicando acerto e motivos.
- c) Áudios são reproduzidos ao jogador parabenizando-o pelo seu acerto e ratificando os motivos deste acerto.

Este último item de feedback foi importante pois, sempre existe a possibilidade do jogador tentar adivinhar o caminho e acertar por acaso um desafio. Desta forma o jogo vai indicar os motivos do acerto e também porque os demais estavam incorretos. Todos estes elementos criados nesta atividade tentaram

aumentar o engajamento do aluno no jogo e motivá-lo a estudar os conceitos acerca da linguagem de programação C.

Os feedbacks criados seguiram a orientação da atividade da metodologia e na maioria dos casos, o feedback foi simples, curto e não-intrusivo, a fim de evitar a quebra do ritmo do jogo. No entanto em situações de abordagem de conceitos mais complexos o feedback foi mais explícito e amplo, para ajudar aos alunos a refletirem sobre os conceitos subjacentes de forma mais aprofundada, e transferirem os conhecimentos adquiridos a situações do mundo real. No caso deste jogo sério, o mundo real é o universo da programação de softwares utilizando a linguagem C, e das linguagens cuja algumas estruturas foram derivadas da mesma (como PHP, C++, Java, entre outras).

A orientação ao usuário na descoberta de informações e opções disponíveis relacionadas com tomada de decisão foi desenvolvida usando e alternando momentos com:

- a) Elementos de mistério;
- b) Foco da câmera;
- c) Áudios;
- d) Imagens;
- e) Textos para chamar atenção do jogador;
- f) Imagens com setas indicando caminhos;
- g) Áudios para orientar o jogador quanto as suas opções.

No início do jogo foi estabelecido que o jogador deveria aprender sobre as regras e objetivos do mesmo. Este mistério inicial ocasionou um maior desafio ao jogador. Desta forma em outros momentos este mistério foi aplicado ao jogo. Notadamente em outros pontos do cenário, regras claras é o que regia a dinâmica do jogo. Esta alternância foi bem avaliada pelos alunos nas atividades de testes e avaliação do jogo.

As ações de tomada de decisão e a mecânica do jogo foram simplificadas pelos elementos pertencentes ao motor de jogo UDK. As múltiplas escolhas disponíveis aos alunos foram representadas de forma visual. Alguns comentários em texto aparecem para o jogador para que ele reflita sobre seus erros e as

consequências geradas pelos mesmos. Um exemplo disso é o erro de esquecer o símbolo de & (e comercial) no comando scanf.

Quando o jogador erra no jogo neste aspecto ele recebe um comentário que embora o compilador C não apresente um erro de sintaxe, a execução do código irá ocasionar um erro em tempo de execução. Com todos estes tipos de elementos de feedback o jogador obtém uma indicação que uma interação desejada por ele foi realmente executada e em adição contribuiu conforme ratifica, (TORRENTE et al., 2014), "para criar uma percepção contínua do progresso, uma característica inerente de bons jogos que evita frustração e encoraja o jogador a ir em frente, entre outros benefícios."

Por isso, conforme orientado em uma tarefa da atividade, os vídeos de feedback apresentando um aspecto particular de um procedimento do código C, foram criados para serem mais úteis e envolventes em situações mais complexas. Uma das tarefas desta atividade relata que é importante fazer o aluno "pagar um preço" por erros e ajudá-lo a evitar o mesmo erro na próxima vez.

Durante todo o jogo foram inseridos elementos que fizeram o aluno pagar um preço pelo seu erro, seja morrendo, seja voltando ao início do jogo. Além disso, os feedbacks criados tentaram ajudar o jogador a não repetir o erro e aprender com estes erros.

Para finalizar esta atividade foi criado um feedback final para fins de autoavaliação do aluno. Quando o aluno, erra este erro é colocado dentro de um arquivo texto no diretório de instalação do jogo. Neste ponto uma dificuldade foi observada na criação deste arquivo via o motor de jogo UDK. Apesar de tentativas não foi possível resolver o problema de criar apenas um único arquivo texto. Assim a cada erro um novo arquivo texto é criado.

O inconveniente é que o aluno precisa abrir cada arquivo texto separadamente para visualizar seus erros. Contudo, apesar disso, as informações de cada arquivo texto foram descritas pelos alunos e professores como importantes para o objetivo de ensino e aprendizagem do jogo e para reforçar o estudo dos conceitos que o aluno mais errou.

Conforme orientado na atividade executada, esta informação também foi utilizada para identificar potenciais fraquezas no jogo. Prioritariamente, contudo os

arquivos com os erros dos alunos foram utilizados para reforçar a aprendizagem dos alunos sem a necessidade de intervenção de professores.

Já a tarefa que indica a utilização de heurísticas quantificáveis para mostrar o progresso dos alunos foi apenas parcialmente atendida. Elementos visuais que fornecem status de exibição ou competências adquiridas pelos alunos não foram explicitamente desenvolvidos. O tempo limitado para o desenvolvimento do jogo e a complexidade de programação destes itens foram alguns dos motivos da falta de produção dos mesmos. Em contrapartida os seguintes elementos listados a seguir foram desenvolvidos ou aproveitados e ajudaram na percepção do jogador em relação ao seu progresso no jogo. Estes elementos são:

- a) Pontos de checagem(checkpoint): O aluno ao passar por estes pontos não irá precisar mais voltar ao início do jogo quando errar e for morto.
- b) Tela de tempo e pontuação: Uma tela de tempo restante do jogo e pontuação aparece ao ser pressionada de forma contínua a tecla F1.
- c) Tempo necessário para cumprir o objetivo do jogo aparece ao teclar F1
- d) Os erros cometidos pelo aluno, foram gravados em arquivos de texto dentro da pasta em que o jogo está instalado.

Após a execução da atividade acima o jogo ficou em um nível menor de abstração em relação a sua estrutura. Isso permitiu, solicitar informações adicionais aos especialistas de domínio com mais clareza. Foram solicitadas informações adicionais sobre aspectos importantes de programação com linguagem C aos especialistas de domínio. Em entrevistas informais os professores relataram a falta de prática constante de construção de códigos C, como barreira para aprendizagem dos alunos. Os professores indicaram que a maioria dos alunos somente pratica programação C durante as aulas de laboratório. Apesar de ser uma ação importante para o jogo a possibilidade de construir códigos durante o jogo não foram compatíveis com jogabilidade do jogo (gameplay).

Embora os recursos do motor de jogos UDK sejam sofisticados e aprimorados, a construção de códigos durante o jogo mostrou-se limitado e esta funcionalidade não está presente no jogo. Espera-se, no entanto, que ao jogar o jogo o aluno se sinta mais motivado e compelido a exercitar mais a construção de códigos em C.

Outro aspecto que foi relatado pelos professores foi a falta de entendimento dos alunos das mensagens de erros de compilação. Este aspecto ainda não havia sido tratado no jogo. Este tratamento se deu sobre forma de questões de múltipla escolha e de verdadeiro e falso projetados em paredes do cenário do jogo.

O aluno deve interpretar estas questões e decidir que caminho seguir a depender de sua resposta. Ele poderá escolher suas respostas indo em direção a portas etiquetadas com A B C D ou portas com etiquetas Verdadeiro e Falso. Desta forma alguns erros de compilação foram tratados nestes desafios principalmente em relação à sintaxe da linguagem C.

#### **4.3.1 Atividades de criação de protótipos e desenvolvimento do jogo até a sua versão final**

Após o desenvolvimento e criação de elementos do jogo, iniciou-se a construção dos protótipos através da execução da atividade gerar a partir dos requisitos e documento de design, protótipos evolucionários do jogo. O desenvolvimento dos protótipos foi evolucionário sendo, portanto, incremental. Inicialmente a tarefa desenvolver protótipo inicial do jogo foi executada. Nesta atividade um ambiente do jogo, com jogabilidade completa, foi desenvolvido e testado pelos alunos. Este desenvolvimento seguiu todo o planejamento criado para um jogo sério. Isso inclui principalmente, utilizar como base às características e elementos derivados das propostas e recomendações feitas por alunos e professores durante as entrevistas.

Algumas das propostas foram capturadas através de perguntas presentes no questionário dos alunos sobre suas experiências e hábitos de jogar jogos eletrônicos em casa. Conforme indicado em tarefas desta atividade, estratégias complementares de design relacionadas com diretrizes de projeto da área de interface Homem-computador(IHC) foram utilizadas. Heurísticas de Nielsen (1994) foram utilizadas para avaliação de usabilidade das interfaces do protótipo.

A tabela 29 a seguir, indica cada uma das heurísticas utilizadas, suas descrições e as análises da presença das mesmas na interface do protótipo:

Tabela 31 - Resultados da verificação da presença das heurísticas de Nielsen

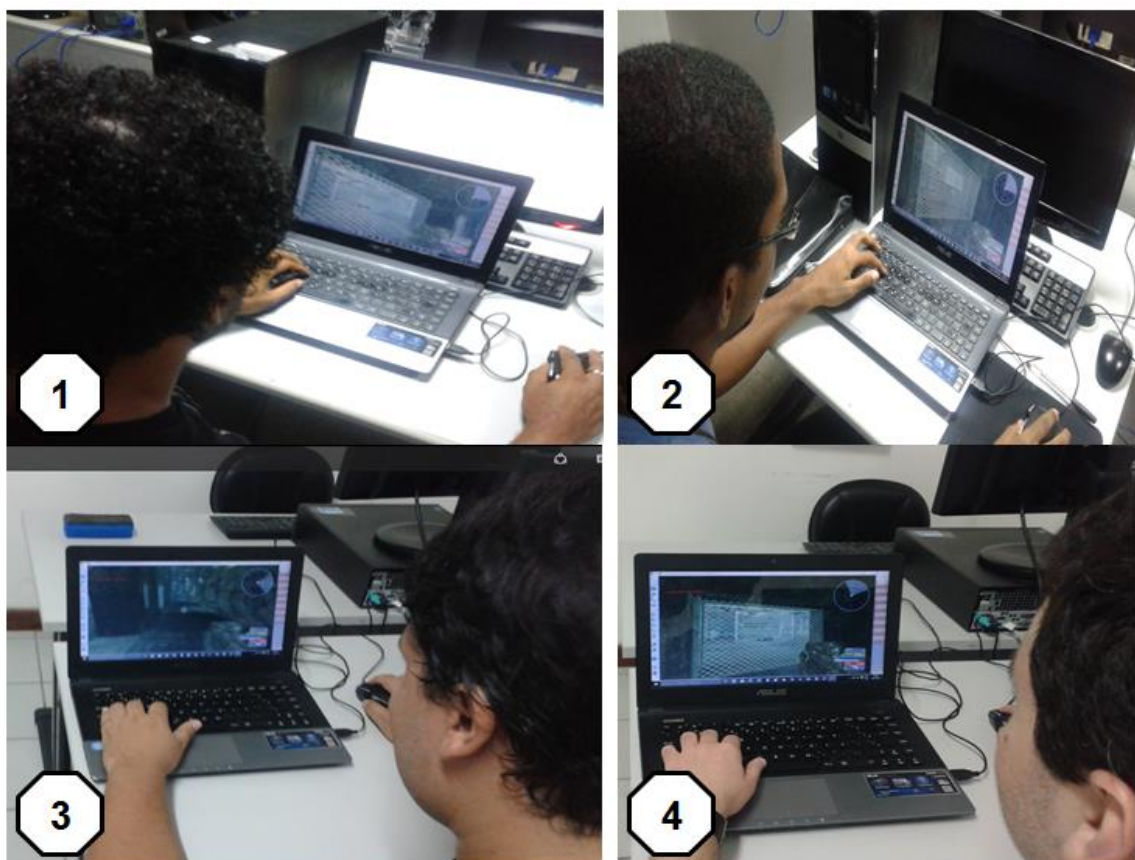
<b>Heurística</b>	<b>1. Visibilidade do status do sistema.</b>	<b>Heurística Satisfeita: SIM</b>
<b>Descrição</b>	O usuário deve estar completamente informado do que está acontecendo, por meio de feedback imediato da interface.	
<b>Análise</b>	No protótipo estão presentes elementos de tela que informam a quantidade de munição e armas. Além disso, mensagens são exibidas se o jogador for abatido ou abater inimigo, se errar algum conceito dos desafios sobre a linguagem C. Em adição se o jogador pressionar a tecla F1 irá visualizar o seu score(pontuação), o tempo que resta para ele acabar o jogo, o número de vezes que foi abatido e abateu inimigos.	
<b>Heurística</b>	<b>2. Compatibilidade do sistema com o mundo real.</b>	<b>Heurística Satisfeita: SIM</b>
<b>Descrição</b>	A terminologia deve ser adequada à linguagem do usuário e não orientada ao sistema. As informações devem ser organizadas de acordo com o modelo mental do usuário.	
<b>Análise</b>	As mensagens e objetos de feedback ao jogador foram criados de forma lúdica e de acordo com os recursos de tela oferecido pelo estilo de jogo FPS. Desta forma durante os testes com o protótipo observou-se que a terminologia utilizada não estava causando dúvidas e assim estava orientada ao jogador e não ao sistema.	
<b>Heurística</b>	<b>3. Controle e liberdade do usuário.</b>	<b>Heurística Satisfeita: SIM</b>
<b>Descrição</b>	Disponibilize "saídas de emergência" ao usuário, para que ele possa desfazer ou refazer ações, a fim de que ele se situe em um ponto recente da interação.	
<b>Análise</b>	Durante o jogo caso o jogador seja abatido ele poderá recomeçar o jogo do ponto de partida ou checkpoint. Se desejado é permitido ao jogador voltar e refazer desafios anteriores.	
<b>Heurística</b>	<b>4. Consistência e padrões.</b>	<b>Heurística Satisfeita: SIM</b>
<b>Descrição</b>	Nunca identifique uma mesma ação por ícones ou metáforas diferentes. Elementos similares devem ser usados para propósitos semelhantes, assim como funcionalidades semelhantes devem possuir uma sequência de ações semelhantes.	
<b>Análise</b>	Para cada ação foram utilizados e programados itens disponíveis no motor de jogos UDK e foram também criados itens adicionais. Dentre estes itens estão sons, textos, imagens, efeitos que aparecem para determinado tipo de ação e funcionalidade. Desta forma a heurística foi atendida.	
<b>Heurística</b>	<b>5. Prevenção de erros.</b>	<b>Heurística Satisfeita: Parcialmente</b>
<b>Descrição</b>	Idealmente, interfaces não precisam de mensagens de erro por serem capazes de prevenir que erros ocorram.	
<b>Análise</b>	A estrutura do motor de jogos já possui um sistema de prevenção de erros. Se algum erro de programação ocorre é emitido na tela em forma de texto informações sobre o erro. Esta informação desaparece rapidamente da tela e o jogador continua o jogo. Entretanto não foram tratados possíveis erros de código e exceções.	
<b>Heurística</b>	<b>6. Reconhecimento ao invés de lembrança.</b>	<b>Heurística Satisfeita: Parcialmente</b>
<b>Descrição</b>	O usuário não deve precisar memorizar o que está realizando. Permita que a interface atue como um meio de dialogar com o usuário, em tempo de execução.	
<b>Análise</b>	O fluxo do jogo e os elementos visuais indicam ao jogador a todo instante as ações que está realizando. Além disso, os demais itens de design foram utilizados ou criados para estabelecer um entendimento claro de seu objetivo no jogo. Durante o teste com o protótipo os jogadores entenderam as funções destes itens de design. Notadamente o estilo de jogo FPS em que os alunos estão acostumados a jogar parece ter auxiliado neste processo de entendimento. Contudo comandos de voz não estão presentes e em alguns momentos do jogo o jogador deverá lembrar de ações necessárias para vencer desafios parecidos.	
<b>Heurística</b>	<b>7. Flexibilidade e eficiência de uso.</b>	<b>Heurística Satisfeita: Parcialmente</b>
<b>Descrição</b>	O sistema deve ser fácil de ser operado por usuários novatos, mas também robusto o suficiente para permitir eficiência de uso a usuários avançados. Teclas de atalho e de comandos por voz podem ser alternativas para tornarem a interface flexível. Em outras palavras, flexibilidade implica em permitir que uma mesma funcionalidade seja realizada por comandos distintos.	
<b>Análise</b>	Com relação a usuários novatos, foi observado que alunos sem prática em jogos do estilo FPS tiveram certa dificuldade com os controles do jogo. No entanto após orientações do professor os alunos conseguiram avançar no jogo. Já alunos com experiência em jogos FPS e considerados usuários avançados não tiveram dificuldades com os controles e foram eficientes durante o jogo. A flexibilidade da interface esta presente nos comandos de jogo digitados e acionados via tecla TAB. É possível modificar a visão do jogador de primeira para terceira pessoa, modificar o nome do jogador, entre outros. Apesar de existirem estes comandos não existem muitas alternativas que tornam a interface flexível.	
<b>Heurística</b>	<b>8. Estética e design minimalista.</b>	<b>Heurística Satisfeita: SIM</b>
<b>Descrição</b>	As informações devem ser sucintas e não devem informar mais do que os usuários necessitam para realizar a funcionalidade corrente. Os diálogos do sistema precisam ser diretos e naturais e devem aparecer nos momentos adequados.	
<b>Análise</b>	Esta heurística foi seguida e recebeu atenção especial principalmente por trata-se de um jogo sério educacional. O planejamento pedagógico realizado em atividades anteriores, permitiu estabelecer feedbacks diretos e naturais de forma sucinta, para a realização das funcionalidades e ações relacionados com os desafios impostos ao jogador.	
<b>Heurística</b>	<b>9. Ajuda para usuário identificar, diagnosticar e corrigir erros.</b>	<b>Heurística Satisfeita: SIM</b>
<b>Descrição</b>	As mensagens de erros devem ser claras e simples e não podem intimidar o usuário. Ao contrário, devem estimulá-lo a oferecer formas de corrigir o erro.	
<b>Análise</b>	Quando o jogador erra algum conceito ele no geral morre ou perde pontos. Este tipo de ação funciona no jogo	



	como uma mensagem indicando que ele errou. Além disso, mensagens textuais mais explícitas aparecem explicando os motivos do erro e ajudando o jogador a não mais cometê-los. Outro recurso importante foi a adição de vídeos explicativos de códigos da linguagem C que aparecem para o jogador caso ele erre e/ou acerte o desafio. Estes recursos já estavam presentes no protótipo inicial do jogo e os alunos indicaram boa interação com os mesmos.	
<b>Heurística</b>	<b>10. Ajudas (Help) e documentação.</b>	<b>Heurística Satisfeita: Não</b>
<b>Descrição</b>	Um bom design evita que o usuário tenha que usar opções de ajuda com frequência. Entretanto, é fundamental que o sistema possua telas específicas de ajuda, para orientar o usuário em casos de dúvidas.	
<b>Análise</b>	Em alguns momentos do jogo itens de ajuda foram inseridos, pois, percebeu-se dificuldades no entendimento do desafio, No protótipo inicial um código no chão com comandos de decisão não estava sendo percebido pelos alunos. Foi colocada uma seta indicando o aluno olhar para este código e analisar o desafio. Outros efeitos foram utilizados em determinados pontos de confusão do jogo com os mesmos objetivos. Contudo estes foram elementos paliativos e o jogo não possui um help e nem documentação. Embora importantes a ideia é deixar que o jogador descubra as regras através da experiência de tentativa e erro.	

Após o desenvolvimento do protótipo foi executada a atividade de efetuar testes através dos protótipos desenvolvidos. O teste com este protótipo inicial serviu para analisar se o conceito estabelecido para o jogo estava adequado e alinhado aos objetivos de ensino criados para o jogo sério. A jogabilidade(gameplay) do jogo também foi avaliada e mostrou uma boa interação com o jogador. O pós-teste criado na seção 4.3 e descrito no apêndice C, foi transformado em elementos do jogo. Os conceitos de identificação de erros de sintaxe e semântica, tomada de decisão e lógica de programação foram estabelecidos como itens do jogo. A figura 47 a seguir mostra alunos testando o protótipo produzido.

Figura 47 - Alunos testando o protótipo do jogo



O protocolo think-aloud foi utilizado durante os testes do protótipo e a seguir existem algumas transcrições textuais do que os alunos comentaram durante o jogo:

Tabela 32 - Transcrições das falas dos alunos após a aplicação do think-aloud

1	"da até pena de morrer" – Tem que ser esquerda – Não é para direita – Tem que ver o incremento. Eu atirei em qual a primeira vez? Esse vai ser 1 aqui não vai incrementar- Foi!!! – Derrubou aqui o barril e aí? – Ah ele passa ele passa – Por aqui vamos lá – Morri- Já sei como é já - Só tem esse código – Ah tem vários códigos nas portas – Entendi , Entendi, Volta, Volta, Volta. Ah ai é a resposta né – Ah apenas um caractere em um char – Pô show de bola! – Se chegar perto ele já abre – char c = 'e' – Acertei!.
2	Obs: Para este aluno o vídeo foi corrompido não sendo possível recuperar a transcrição.
3	Eu estou meio perplexo com as variáveis – Levarei tempo para me acostumar com a mecânica do jogo. – Entendi que estava morrendo por que isso aqui (código) devia tá errado - eu imagino que tenho que procurar a entrada certa – Pra eu puder avançar no jogo - Não é aqui por que está sem o ponto e vírgula – Tava sem aspas duplas também – Huum – Ah legal depois que entendi que depois que você morre ele lhe mostra qual é o correto – Então por eliminação – Opa – Agora estou feliz por que passei – Agora o que é isso aqui? – é um comando? – Mas eu tenho que descobrir o que é para fazer – Deixa eu fazer um teste atirando em um barril – Ohhh morri.
4	Agora entendi este lugar está errado – Ah este está certo – Eu entendi a instrução – aqui eu posso entrar – Opa não??? – [risos com o efeito do som] – Qual é o erro deixa eu ver – AAHHHH não é uma cadeia de caracteres entendi – que massa!!!! – Não era uma string – Ah ótimo – aqui não posso entrar – aqui não posso entrar – Esse aqui só podia ter um caractere – Esse aqui, Esse aqui dá para entrar – Deixa eu ver se tem mais algum – Esse aqui não posso entrar porque o C dá dizendo que é igual a aspas duplas – Vamos ver – Exato - entendi

Após jogar o jogo o aluno foi entrevistado. A tabela 31 a seguir indica as transcrições dos áudios das entrevistas para cada aluno identificados pelos números nas figuras.

Tabela 33 - Transcrição das entrevistas realizados com os alunos

1	"O jogo é bom, de fato é didático, poderia usar da seguinte forma. A partir que você morre ele modificar o código para você acertar. Como está você pode ir por eliminação e às vezes pode não precisar pensar."
2	Obs: Para este aluno o áudio foi corrompido não sendo possível recuperar a transcrição
3	"Interessante quando você erra alguma coisa no jogo ele mostra no código como deveria ser e você aprende. A única coisa que fiquei assim foi que se acabasse morrendo eu iria perder no jogo e quando estou estudando eu não tenho muito problema, pois quando não compila o código eu simplesmente volto vejo o que tá errado e faço de novo, mas essa sensação de dentro do jogo, eu vou perder, ela me incomodou um pouquinho. Tirando isso achei a proposta muito boa. É importante quando se tem modos alternativos ao convencional de se ensinar, que possa fazer esta aproximação maior do aluno com o material a ser estudado."
4	"Eu acho que ele tá muito legal quanto à usabilidade, só que quanto a destacar o que você tem que fazer precisa ser aprimorado. Porque quando a gente chega no desafio não existe nenhum indicativo que é preciso pensar naquela fórmula para fazer. Mais instruções do tipo tente resolver isso, tente resolver aquilo seria melhor para a proposta do jogo. Ficaria mais fácil, mais direto. Eu adorei pois eu aprendi errando, errei a primeira vez e já sabia que não podia ir por aquele caminho e ai fui pra outra alternativa até pegar o jeito da coisa. Quando a gente está em casa estudando de verdade a gente tá errando e procurando uma solução para aquele problema que a gente não conseguiu resolver. A mesma coisa foi aqui no jogo. Normalmente em sala de aula a gente não está 100% no ritmo do professor. Fazendo a gente quebrar a cabeça para conseguir avançar, achei fantástico, adorei."

Baseado na observação e nos feedbacks dos alunos os desenvolvedores efetuaram alterações na interface do jogo e nos desafios de tomada de decisões. Este protótipo foi evoluído até a versão final do jogo. Durante o desenvolvimento novos testes foram realizados para a analisar os elementos do jogo e engajamento dos usuários finais. Percebeu-se que, de fato, foi mais simples para os especialistas de domínio encontrar imprecisões ou aspectos negligenciados no protótipo do que analisar desenhos ou documentos. Os desenvolvedores puderam testar rapidamente a mecânica do jogo e identificar possíveis inconsistências. Assim a próxima atividade foi desenvolver e codificar o jogo até a versão final.

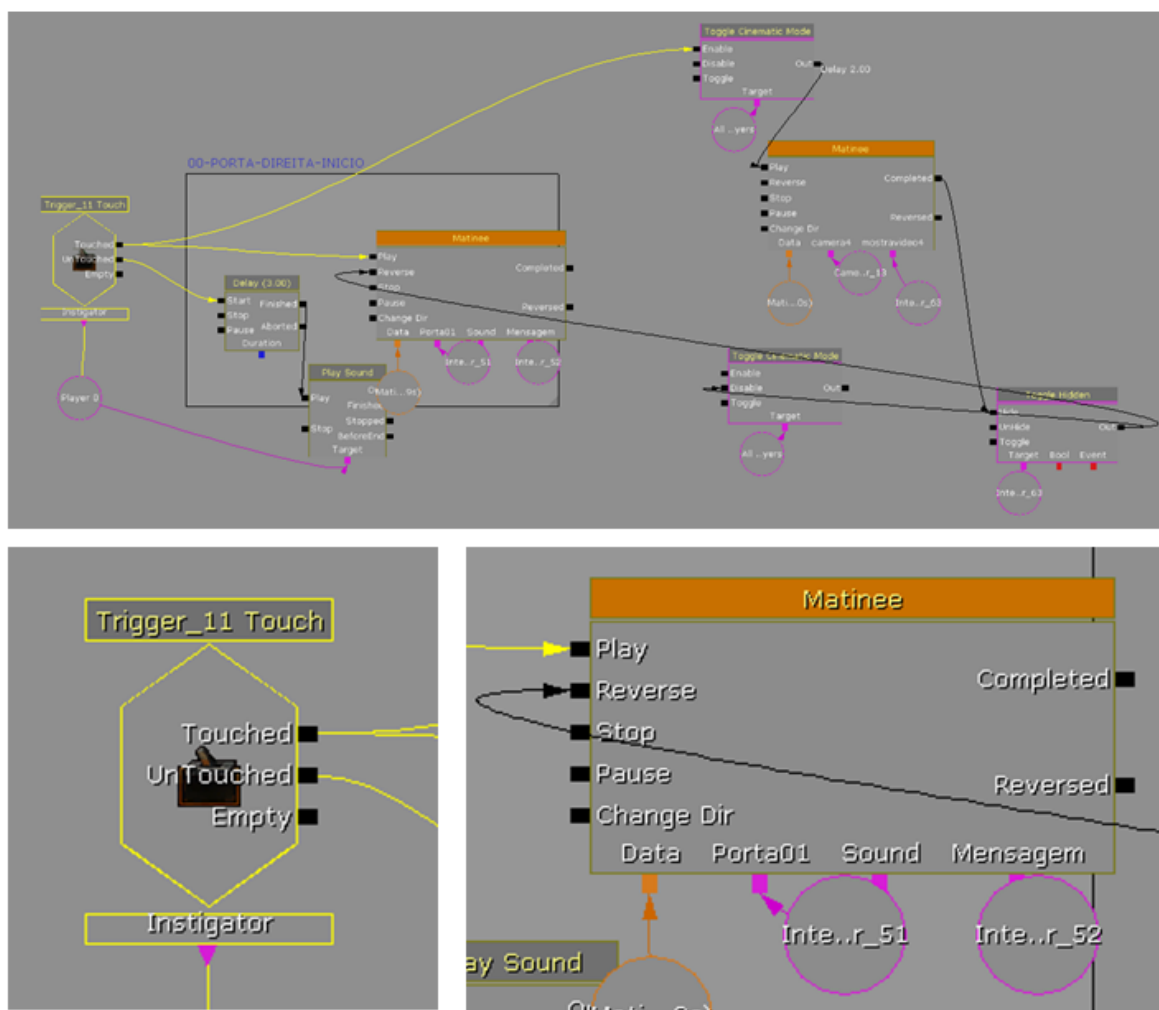
O jogo foi desenvolvido na categoria atirador em primeira pessoa (FPS First Person Shooter). O motor de jogo(engine) UDK escolhido para o desenvolvimento do jogo, proporciona muitos itens prontos para apoiar o desenvolvimento de um jogo FPS:

- a) Armas
- b) Robôs(bots);
- c) Materiais e texturas;
- d) Elementos gráficos;

- e) Inteligência artificial pré-programada;
- f) Física e sistema de colisão facilitada por meio de configurações visuais;
- g) sistema de partículas que simulam fogo fumaça água, etc;
- h) Editor de cinemática permitindo criar cenas com câmera;
- i) Animações já programadas;
- j) Diversos sons;
- k) Facilidade em criar elementos geométricos e aplicar materiais e texturas;  
Possibilidade de programar o jogo de forma visual utilizando o Unreal kismet;
- l) Possibilidade de criar materiais e texturas;
- m) Possibilidade de importar vídeos e inseri-los no jogo.

Para a codificação foi utilizado um recurso do motor de jogo UDK que permite a criação da lógica de programação do jogo de forma visual. Através de setas e elementos que representam estruturas de programação o recurso denominado kismet foi amplamente utilizado. Isso ocasionou um aumento da produtividade em relação à programação do jogo. A figura 48 a seguir mostra um exemplo desta programação desenvolvida para o jogo UNLOCK-C.

Figura 48 - Exemplo de programação através do Kismet da UDK



A programação está relacionada com um elemento denominado gatilho (trigger). Este elemento é invisível ao jogador, mas pode ser tocado por ele. Neste caso ele foi posicionado no cenário em frente a uma porta e quando o jogador tocar (touched) no gatilho(trigger) ele dispara um fluxo de ações que seguirão as setas programadas nas ligações mostradas na figura 48. Em específico ao disparar o gatilho será chamada uma animação programada no módulo de matinee do motor de jogos. Esta animação faz com que a porta abra. Ao mesmo tempo é esperado 3 segundos de atraso(delay) e um som de porta abrindo é disparado. A outra seta deste mesmo fluxo acionado o modo de cinemática fazendo com que o jogador e outros elementos do jogo como inimigos sejam congelados e a cena fique focada na

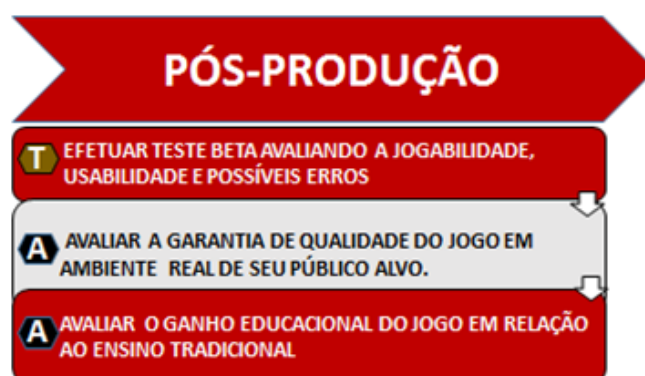
animação do matinee. Isso evita problemas do jogador invadir a cena durante uma animação.

Outros recursos do motor de jogos UDK foram amplamente utilizados no desenvolvimento do jogo sério. Estes elementos estão listados no apêndice G.

#### 4.4 ATIVIDADES DE PÓS-PRODUÇÃO

Após as fases de pré-produção e produção do jogo, iniciou-se a execução das atividades relacionadas com a pós-produção do jogo listadas na figura 49 a seguir.

Figura 49 - Atividades da fase de pós-produção



##### 4.4.1 Atividades de testes de pós-produção

Após o desenvolvimento do jogo, uma versão beta foi testada através da atividade: efetuar teste beta avaliando a jogabilidade, usabilidade e possíveis erros. Alunos e professores foram recrutados para testar o jogo conforme mostrado nas figuras 50 e 51. Eles exploram o jogo, experimentando todas as suas ações conforme orientado nas tarefas da atividade da metodologia. Durante e após cada testador jogar o jogo foram colhidas informações sobre:

- a) Erros técnicos(bugs)
- b) Maiores falhas de usabilidade
- c) Feedback sobre o valor educacional do jogo
- d) Experiência de diversão esperada.

As interações foram gravadas em áudio e vídeo através da câmera de um smartphone. O objetivo foi eliminar problemas de usabilidade e avaliar a utilidade do jogo na realização dos seus objetivos pretendidos. Foram também realizados testes com os professores com o método think-aloud como ferramenta de obtenção de

dados, principalmente comentários sobre se o jogo pode ter sucesso em fazer com que os alunos aprendam o conteúdo pretendido. A lista de erros técnicos, falhas de usabilidade, experiência de diversão esperada, valor educacional esperado é mostrada posteriormente na tabela 33.

Figura 50 - Aplicação jogo laboratório informática alunos de computação



Tabela 34 - Lista erros técnicos, falhas usabilidade, ludicidade, valor educacional

<b>Erros técnicos (bugs)</b>	Algumas portas eram fechadas muito rapidamente impedindo o jogador de prosseguir obrigando-o a voltar a abrir a porta novamente. Foram realizados ajustes nas animações e cinemática.
	Após a compilação da luz do jogo alguns elementos ficaram em áreas de sombra, impedindo o jogador de visualizar o código. Os ajustes foram realizados e as sombras deixaram de existir.
	Alguns códigos estavam representados em imagens e elementos pequenos o que dificultava a leitura do jogador. Estes itens foram redimensionados.
	Alguns desafios não eram percebidos. Animações com foco da visão do jogador no código foram usadas para chamar atenção do jogador.
	O material aplicado nas árvores, arbustos e plantas do cenário ficou cinza após a criação da versão final instalável do jogo, ocasionando efeito estético desagradável. A correção foi efetuada e uma nova compilação do jogo foi realizada.
	Criação de muitos arquivos texto com logs de erros conceituais da linguagem C cometidos pelo jogador. São criados no diretório de instalação do jogo, um arquivo de log para cada erro. Apesar de tentativas este erro técnico não foi corrigido. Contudo, ao final do jogo dentro do diretório é possível com um pouco mais de trabalho, visualizar os erros que o jogador cometeu.
	Algumas poucas mensagens de erro da programação com script visual(kismet) apareceram na tela. Estas mensagens não impediam e nem afetavam as ações do jogo. Elas foram tratadas e ocorriam pela falta do objeto instanciado e referenciado no código durante a execução do jogo.
<b>Falhas de usabilidade</b>	Em alguns momentos do jogo os objetivos e desafios do jogador estavam em nível de dificuldade muito acima do planejado. Dicas e itens foram colocados para ajudar na descoberta das ações necessárias por parte do jogador para vencer os desafios.
	Os comandos de tela para aparecer às opções de armas, munição e equipamentos do jogador não estão disponíveis de forma explícita. Os alunos e professores receberam um mini manual com as instruções do jogo, comandos de teclas F1, F10, etc, de como verificar status e tempo de jogo, e o menu de armas, equipamentos e munição.
<b>Experiência de diversão esperada</b>	Para não se tornar um efeito muito repetitivo, alguns elementos não indicavam se o jogador errou ou acertou através de explosões e sons. No entanto percebeu-se frustrações do aluno ao não visualizar ou ouvir estes elementos. Estes elementos foram colocados e apesar de repetitivos agradaram na experiência de diversão do jogador.
	Alguns sons engraçados estavam presentes quando o jogador acertava desafios. Contudo durante os testes foi percebido que os alunos se motivavam e riem muito com estes sons de YEAH!. A quantidade de sons com esta ludicidade foi aumentada.
	Os alunos relataram que em algumas salas deveria haver mais inimigos. Foram colocados mais inimigos nestes ambientes.
	Os alunos questionaram se era possível mudar o nome do jogador que aparecia como PLAYER. Não era possível modificar o nome do jogador via menu de opções, mas após pesquisa foi oferecido aos alunos o comando setName que muda o nome do jogador através da tecla TAB.
<b>Valor educacional esperado</b>	Alguns itens do jogo apenas citavam via texto o feedback sobre o acerto e erro do jogador. Os alunos observaram que achavam melhor usar vídeos para explicar os motivos do erro.
	Os nomes dos inimigos eram genéricos e foram modificados para nomes de barreiras ao estudo de programação. Como: Falta de tempo, não vou conseguir, é difícil, não sei programar, etc. Assim quando o jogador abate um inimigo uma mensagem aparece na tela indicando por exemplo: <nome do jogador> killed "falta de tempo".
	Foi observado por professores que o aluno poderia chutar e avançar no jogo por eliminação de possibilidades. Ou seja, em alguns elementos ele poderia atirar aleatoriamente e não receberia um feedback sobre seu erro. Este item foi corrigido colocando-se feedbacks aos jogadores quando atirava em algum item(barril por exemplo) e morria. Desta forma todos os itens de erro ou acerto do jogador receberam feedbacks quanto ao seu erro. Além disso, o professor foi orientado a incentivar o aluno a jogar o jogo com o objetivo de aprendizado. Ele deve informar ao aluno que seus erros conceituais estão sendo coletados para que ele no final possa obter uma lista de conceitos da linguagem C que mais necessita estudar.
	Outro professor indicou que seria interessante aparecer no final do jogo uma lista de erros na tela indicando onde os jogadores erraram. Foi explicado ao professor que este item seria interessante, mas que demandaria muito mais tempo de programação e aprendizado por parte dos desenvolvedores. Foi desenvolvido um item em código unrealscript em uma classe do jogo que escreve arquivos de texto com os erros do jogador.
	Outro item observado por professor foi a questão da modularidade do jogo em fases. Como o jogo tem como objetivo ser uma ferramenta de ensino e aprendizagem seria interessante usar apenas parte do jogo a cada conceito trabalhado em sala de aula e depois avançar para



<p>módulos com outros conceitos avançados. Esta funcionalidade não foi implementada de forma técnica, mas foi oferecida ao professor a oportunidade de realizar esta ação de forma operacional. Ele poderia mediar as interações do jogo indicando aos alunos até onde ele deveria ir no jogo. O elemento de curiosidade também poderia ser utilizado quando o aluno observasse conceitos ainda não estudados e que deveria estudar para ir bem no jogo em próxima rodada.</p> <p>Professor indicou que o pós-teste desenvolvido para o jogador ser capaz de responder após jogar o jogo, deveria ser dividido por conceitos. Existiria um pós-teste para cada um dos conceitos de cada ambiente do jogo e um pós-teste final para o jogo completo. Foi informado ao professor que ele poderia elaborar estes pós-testes intermediários como elementos pedagógicos de sua aula. E que posteriormente ele poderia utilizar o pós-teste final criado para o jogo e disponível no apêndice C.</p>
--

Figura 51- Alunos jogando o jogo sério desenvolvido



Além dos resultados citados anteriormente foi observado que embora o jogo, tenha sido concebido para ser jogado por apenas um jogador (single player), a dinâmica de jogo em grupo mostrou-se uma alternativa interessante. Equipes de alunos podem ser formadas e o professor pode mediar uma competição entre as mesmas.

#### 4.4.2 Atividades de avaliação do jogo

As próximas atividades executadas objetivaram avaliar a garantia de qualidade do jogo em ambiente real de seu público alvo. Conforme indicado na tarefa da atividade foram verificados os aspectos a seguir:

- Confiabilidade (o jogo é estável e livre de erros de programa);
- Facilidade de utilização (ou seja, o jogo é agradável e fácil de usar);
- O valor educativo (isto é, o os alunos podem atingir os objetivos de aprendizagem pretendidos)
- O envolvimento (ou seja, o jogo é apelativo e envolvente e desafia as habilidades do jogador)
- O conteúdo educacional trabalhado no jogo é válido? (só pode ser feito por especialistas do domínio através de uma avaliação de todos os conceitos trabalhados no jogo)

Os resultados da avaliação são apresentados na tabela 34.

Tabela 35 - Resultados da avaliação do jogo sério UNLOCK-C

<b>Confiabilidade (O jogo é estável e livre de erros de programa)</b>
R. Sim, em todos os testes o jogo não travou. Apenas surgiram algumas poucas mensagens de erro da programação com script visual(kismet), que apareceram na tela e não impediam e nem afetavam as ações do jogo.
<b>Facilidade de utilização (ou seja, o jogo é agradável e fácil de usar)</b>
R. Sim, por ter sido construído com elementos 3D e itens de design avançados o jogo foi facilmente manipulado pelos jogadores. Além disso, por se tratar de um jogo do tipo tiro em primeira pessoa(FPS), a maioria dos alunos já estava acostumada com os comandos e dinâmica do jogo. Alguns alunos que não tinham o costume de jogar jogos FPS, tiveram dificuldade no início. Contudo, com o passar do tempo de jogo eles conseguiram avançar.
<b>O valor educativo (isto é, o os alunos podem atingir os objetivos de aprendizagem pretendidos).</b>
R. Sim, os pós-testes realizados pelos alunos tiveram resultados satisfatórios e os alunos indicaram que aumentou a motivação em estudar a linguagem de programação C. Apesar disso, vale ressaltar que esta avaliação não foi realizada com métodos formais de pedagogia.
<b>O envolvimento, ou seja, o jogo é apelativo e envolvente e desafia as habilidades do jogador.</b>
R. Sim. Diversos elementos do jogo criaram um ambiente imersivo o que ficou evidenciado nos vídeos gravados dos alunos jogando o jogo.
<b>O conteúdo educacional trabalhado no jogo é válido? (só pode ser feito por especialistas do domínio através de uma avaliação de todos os conceitos trabalhados no jogo)</b>
R. Sim, os conceitos trabalhados no jogo seguiram o plano de ensino da disciplina de programação com linguagem C. Estes conceitos foram gradativamente trabalhados durante o jogo. Alunos e professores relataram que gostaram do conteúdo educacional trabalhado.

Após a execução da atividade anterior outra avaliação foi efetuada para avaliar o ganho educacional do jogo em relação ao ensino tradicional. A tabela 35

indicada na atividade foi utilizada para esta avaliação. Conforme é destacada na atividade uma avaliação formal é o tipo mais demorado de avaliação e, portanto, apenas uma ou duas experiências no final do processo são recomendadas para avaliar o ganho educacional em relação ao ensino tradicional. Assim uma experiência foi realizada e ao final de uma sessão de execução do jogo completo as perguntas foram realizadas para os respectivos papéis. Os resultados estão descritos na tabela 35.

Tabela 36 – Questões de avaliação de ganho educacional e projeto do jogo sério

Questão	Quem responde?	Relacionado a
<b>Existem erros ou comportamentos inesperados no jogo?</b>	Beta-Testers	Confiabilidade
R. Não, em todos os testes o jogo não travou. Apenas surgiram algumas poucas mensagens de erro da programação com script visual(kismet) que apareceram na tela e não impediam e nem afetavam as ações do jogo.		
<b>O jogo funciona em todas as plataformas-alvo?</b>	Beta-Testers	Confiabilidade
R. Sim, contudo a única plataforma alvo é a PC 2.0 GHz velocidade processador, 2 GB de memória RAM e placa de vídeo com suporte modelo de sombras 3.0. Para esta plataforma o jogo foi instalado e funcionou corretamente.		
<b>O jogo simula fielmente o processo e ambiente proposto?</b>	Especialistas de Domínio	Valor educacional
R. Apesar de não ser um jogo sério de simulação, o ambiente proposto relacionado a um jogo FPS que acontece em uma fábrica de produtos químicos abandonada foi estabelecido. O processo de ensino e apoio a aprendizagem ocorreu através dos diversos tipos de feedbacks fornecidos ao jogador, em especial os vídeos explicativos sobre códigos em linguagem C. Os professores aprovaram a dinâmica do jogo e apenas sugeriram questões pontuais.		
<b>O conteúdo incorporado no jogo (por exemplo, materiais de referência, diálogos, texto) são precisos e adequados para o público-alvo?</b>	Especialistas de Domínio	Valor educacional
R. Sim. O público alvo(alunos de computação) já está, em sua maioria, acostumada com os elementos de design utilizados em um jogo do gênero FPS. Os elementos de códigos utilizados seguiram os conceitos utilizados nas aulas sobre linguagem C e se mostraram adequados para o público-alvo que não relatou maiores empecilhos durante os testes e avaliação do jogo.		
<b>Os objetivos do jogo e as regras são claras para o usuário iniciante?</b>	Estudantes, especialistas de domínio, especialistas de Jogos	Jogabilidade, Valor educacional, Experiência de uso
R. Parcialmente. O usuário iniciante no jogo e quem não tem experiência em jogos FPS relataram dificuldades no início do jogo. Contudo após algumas orientações de comandos os usuários iniciantes conseguiram avançar. Uma observação importante é que um dos elementos de desafio e mistério planejado para o jogo é que inicialmente o jogador descubra quais são as regras e objetivos do jogo, por meio de tentativa e erro.		
<b>O jogo é atraente para o público-alvo?</b>	Estudantes, especialistas de	Jogabilidade, Experiência de uso

	Jogos	
R. Sim, durante as gravações dos alunos jogando o jogo, os comentários foram positivos em relação a esta atratividade e os alunos se mostraram engajados.		
<b>O contexto e as condições são claras para o usuário iniciante?</b>	Estudantes, especialistas de domínio, especialistas de Jogos	Jogabilidade, Valor educacional, Experiência de uso
R. Não, mas como já observado, um dos elementos de desafio e mistério planejado para o jogo é que inicialmente o jogador descubra quais são as regras e objetivos do jogo, por meio de tentativa e erro. Aprender errando é um dos pilares de ensino através de jogos e já referenciado no capítulo 2 desta dissertação.		
<b>É fácil aprender a usar o jogo?</b>	Beta-Testers, Estudantes, especialistas de Jogos	Usabilidade
R. Parcialmente. Alguns itens foram projetados para serem mais complexos e aumentar os desafios. Contudo, com a importante mediação do professor não existiram maiores dificuldades no aprendizado da dinâmica do jogo.		
<b>Os objetivos de aprendizagem são contemplados?</b>	Especialistas de Domínio	Valor educacional
R. Sim, o jogo tem como principal objetivo ser um instrumento de apoio ao ensino e aprendizagem de conceitos relacionados a disciplinas de cursos de computação que utilizam a linguagem de programação C como suporte. Este apoio foi verificado com o relato do aumento do engajamento e motivação dos alunos em relação a seu aprendizado da linguagem de programação C. Os especialistas de domínio conseguiram visualizar os conceitos ministrados em suas aulas e a possibilidade de utilização do jogo como ferramenta de apoio nas aulas.		
<b>Quão bem os alunos alcançam as metas de aprendizagem?</b>	Estudantes	Valor educacional
R. De forma satisfatória. Os alunos responderam com bom nível de aproveitamento ao pós-teste desenvolvido para ser respondido após jogarem o jogo. A média de aproveitamento dentre 52 alunos que responderam os testes foi de 87,9%		
<b>Será que os alunos aprendem mais alguma coisa (conhecimento incorreto ou correto)</b>	Estudantes	Valor educacional
R. Sim, o jogo fornece diversos feedbacks em relação a conceitos sobre programação da linguagem C, o que fez com que os jogadores aprendessem diversos detalhes sobre a linguagem. Outro ponto importante é que o aluno só consegue avançar no jogo após aprender com seus erros e ratificar seus acertos. Este ritmo de aprendizado muitas vezes não é conseguido em sala de aula. No jogo cada aluno avança e aprende no seu ritmo.		

Além da tabela anterior os alunos foram estimulados a escrever abertamente e de forma anônima sobre sua percepção em relação ao jogo. Alguns resultados estão transcritos na tabela 36 a seguir.

Tabela 37 - Percepção dos alunos após jogarem o jogo sério desenvolvido

<p>EXCELENTE! O JOGO VOLTADO PARA O SEGMENTO EDUCACIONAL, SEM DÚVIDA UMA ÓTIMA PROPOSTA PARA A GALERA QUE ESTÁ DESSTIMULADA.</p>
<p>"Excelente! o jogo voltado para o segmento educacional, sem dúvida uma ótima proposta para a galera que está desestimulada"</p>
<p>O jogo testa os conhecimentos de programação para que o jogador consiga avançar no mapa/missão, fazendo assim uma revisão para os alunos de linguagem de programação. Jogabilidade boa, dificuldade depende da selecionada, gráficos bons, pode ser uma boa ideia fazer um no estilo RPG.</p>
<p>"O jogo testa os conhecimentos de programação para que o jogador consiga avançar no mapa/missão, fazendo assim uma revisão para os alunos de linguagem de programação. Jogabilidade boa, dificuldade depende da selecionada, gráficos bons, pode ser uma boa ideia fazer um no estilo RPG."</p>
<p>JOGO INTERESSANTE, EM QUE ALÉM DE PROPORCIONAR DIVERSÃO, INDUZ O JOGADOR À APRENDIZAGEM DA LINGUAGEM C.</p>
<p>"Jogo interessante, em que além de proporcionar diversão induz o jogador a aprendizagem da Linguagem C"</p>
<p>GAME BASTANTE DIDÁTICO, INSCUTIVA O APRENDIZADO NA LINGUAGEM C. SEGUE UM LINHA DE GAME QUE ATÉ QUIS NÃO <del>FINHA</del> TINHA VISTO</p>
<p>"Game bastante didático, incentiva o aprendizado na Linguagem C. Segue uma linha gamer que até então não tinha visto."</p>
<p>O jogo tem um papel educativo excepcional. Ele nos instiga a buscar onde está o erro, tornando o aprendizado mais sólido.</p>
<p>"O jogo tem um papel educativo excepcional. Ele nos instiga a buscar onde está o</p>

erro tornando o aprendizado mais sólido”

Adorei pois é uma forma diferenciada de ensinar, principalmente para alunos de computação.

Jogo bem feito "nota 10".

“Adorei pois é uma forma diferenciada de ensinar, principalmente para alunos de computação. Jogo bem feito nota 10.”

Percepções sobre o game:

- Estimula a aprendizagem do aluno com resolução de questões;
- Estimula a memória do aluno nas questões já resolvidas nas escolhas das portas corretas;
- Torna o aprendizado divertido, evita o desinteresse e a desmotivação.

Percepções sobre o game: Estimula a aprendizagem do aluno com resolução de questões. Estimula a memória do aluno nas questões já resolvidas, nas escolhas das portas corretas. Torna o aprendizado divertido, evita o desinteresse e desmotivação.

Inicialmente o jogo me pareceu confuso. Porém, na base da tentativa e erro o jogo te faz entender seu objetivo e particularmente, gostei disso.

A questão de tentar avançar o máximo possível contra o tempo também me agradou. Achei um jogo divertido e, até certo ponto, viciante.

“Inicialmente o jogo me pareceu confuso. Porém, na base da tentativa e erro o jogo te faz entender seu objetivo e particularmente gostei disso. A questão de tentar avançar o máximo possível com tempo também me agradou. Achei um jogo divertido e, até certo ponto, viciante.”

Ótimo.

Não entendo muito de jogos, mas é algo muito interessante. Viciante e acaba chamando a atenção para aprender e entender.

“Ótimo. Não entendo muito de jogos, mas é algo muito interessante. Viciante e acaba chamando atenção para aprender e entender.”

Com isso, todas as atividades da metodologia foram executadas resultando no jogo sério educacional UNLOCK-C. Um exemplo de jogabilidade (gameplay) deste jogo pode ser visualizado no vídeo disponível para download em: <https://www.dropbox.com/s/pounnd7my3a400c/Unlock-c%20Trechos%20jogo%20mestrado.mp4?dl=0>. O jogo sério pode ser baixado e instalado no link <https://dl.orangedox.com/ooOGAvA1N2d4k9DF2u>. A sua utilização é gratuita para uso educacional e deve seguir a licença de uso da Unreal Development Kit (UDK) disponível em: <https://www.unrealengine.com/previous-versions/udk-licensing-resources>.

No capítulo seguinte as conclusões sobre este trabalho serão relatadas.

## 5 CONSIDERAÇÕES FINAIS

Este capítulo tem como objetivo apresentar um conjunto de conclusões oriundas da execução deste trabalho, explicitar contribuições e propor trabalhos futuros que deem continuidade à pesquisa efetuada e aos resultados obtidos.

### 5.1 CONCLUSÕES

A utilização pelos alunos do jogo sério educacional desenvolvido neste trabalho representa uma alternativa ao formato de aprendizagem tradicional com aulas expositivas e mecânicas. Baseado em tecnologia digital sob a forma de um jogo sério, esta alternativa de ensino torna-se cada vez mais relevante para lidar com os alunos nativos digitais. Buscou-se neste trabalho tentar mitigar os problemas da falta de motivação e interesse dos alunos novatos dos cursos de computação, em relação às disciplinas de programação, com foco na linguagem C.

Para isso, foi apresentada uma proposta de abordagem para apoiar o processo de ensino e aprendizagem, acerca de conceitos de programação de computadores e linguagem C. Esta proposta sugeriu a utilização do jogo sério educacional, concebido para servir de ferramenta de apoio às aulas ministradas. A tentativa era aumentar a motivação e interesse dos alunos em relação às disciplinas de programação que utilizam a linguagem C.

Para ampliar as chances de sucesso do jogo em relação às boas práticas da engenharia de software e suporte pedagógico, foi identificada a necessidade de utilização de uma metodologia especializada no desenvolvimento de jogos sérios educacionais. No entanto, observou-se que não existia na literatura uma metodologia de desenvolvimento de jogos sérios educacionais que se destacasse de forma clara, de outras existentes, e que fosse aceita de forma consensual. Assim, foi concebida, customizada e otimizada uma nova metodologia.

Através da execução desta nova metodologia customizada, o jogo sério educacional foi desenvolvido. O jogo sério foi concebido com representação tridimensional (3D) do gênero: tiro em primeira pessoa (FPS – First Person Shooter), para que possuísse características que se aproximassem dos jogos digitais comerciais que os alunos estão acostumados a jogar. A proposta foi criar um jogo que se diferenciasse dos jogos sérios anteriormente criados para ensino de



programação com linguagem C (MILJANOVIC; BRADBURY, 2016; MEEKLAI et al., 2015; CHARALAMPOS; NEKTARIOS; STAVROS, 2015; KHENISSI; ESSALMI; JEMNI, 2013; CHANG; CHOU; CHEN, 2010) e dos jogos educacionais padrões. Criou-se para o jogo sério, uma jogabilidade aprimorada e avançada para tentar deixar o aluno motivado a jogar o jogo e conseqüentemente ficar empenhado a estudar os conceitos de programação com linguagem C.

A aplicação do jogo em sala de aula mostrou-se útil na melhoria do engajamento e motivação dos alunos frente aos conteúdos de programação com linguagem C. Nos questionamentos acerca da percepção do jogo a maioria dos alunos relatou que ficaram mais interessados em estudar a sintaxe e conceitos da linguagem C, após terem jogado o jogo. Eles relataram também que isso ajudou no seu desempenho na disciplina.

Um dos pontos positivos apontados pelos alunos foram os vídeos explicativos que aparecem na tela quando o jogador erra algo relacionado a linguagem C. Desta forma eles gostaram de aprender com seus erros, e acharam esta forma mais interessante que as aulas tradicionais pois, nem sempre tiravam dúvidas sobre seus erros. Assim se for levado em consideração as pesquisas preliminares e os pós-testes realizados, o jogo desenvolvido atendeu aos objetivos educacionais definidos ao mesmo.

Como lição aprendida destaca-se que durante a aplicação do jogo em sala de aula com computadores, percebeu-se que embora o jogo tivesse sido concebido prioritariamente para ser jogado por apenas um jogador, havia potencial para jogá-lo em grupo. A experiência do jogo em grupo foi interessante, pois a competição entre equipes dinamizou o processo. Contudo foi percebido que apenas um ou dois integrantes da equipe é quem realmente operava o jogo e conhecia os conceitos. Foi concluído que para o atendimento dos objetivos educacionais é recomendado manter a prática do jogo por apenas um jogador. Isso individualiza o aprendizado e dar mais subsídios para que cada aluno observe seus pontos fracos acerca dos conceitos de programação e foque neste estudo.

Já que foi estabelecido ao jogo objetivos educacionais, conclui-se que estes objetivos parecem terem sido atendidos. É importante destacar a necessidade de avaliações pedagógicas mais formais para que, de fato, possa ser afirmado que os

objetivos educacionais do jogo sério desenvolvido tenham sido rigorosamente atingidos.

Assim destacam-se as seguintes conclusões:

- a) Os primeiros resultados do uso do jogo já indicam que ele foi bem aceito pelos alunos e por outros professores da instituição do autor.
- b) Ainda não é possível avaliar uma contribuição efetiva do jogo como ferramenta que alavanca o aprendizado.
- c) A metodologia customizada contribuiu para o sucesso do trabalho, pelos seguintes aspectos:
  - o Agregou aspectos de engenharia de software a aspectos da área educacional.
  - o A metodologia também contribuiu para a especificação e detalhamento concepção e design do jogo, em toda a sua complexidade, conforme pode ser visto nas descrições e produtos de trabalho do capítulo 4.
- d) Foi percebido que o desenvolvimento de um jogo necessita de criatividade e uma dose de fantasia, algo que não pode ser obtido através de metodologia. Há, portanto, uma interseção com o campo artístico.
- e) Foi observado que a experiência em jogos eletrônicos, tanto sérios quanto os que visam meramente entretenimento, é importante para a concepção de jogos que sejam efetivos do ponto de vista educacional e atrativos ao público alvo.

## 5.2 CONTRIBUIÇÕES DO TRABALHO

Como contribuições deste trabalho destacam-se:

- a) Uma metodologia de desenvolvimento concebida para desenvolver o jogo sério educacional que se mostrou efetiva na prática, e evoluiu ao ser utilizada. Foram estudadas inúmeras metodologias de desenvolvimento de jogos com objetivos não inteiramente equivalentes e com graus de detalhamentos os mais diversos. Foi identificado um conjunto de requisitos desejados para a metodologia alvo e a metodologia foi sintetizada em um grau de detalhamento uniforme e razoável, aglutinando aspectos da engenharia de software e educacionais.

- b) O desenvolvimento de um jogo sério educacional com características que o distinguem de outros com mesmo objetivo e que se diferencia destas propostas (MILJANOVIC; BRADBURY, 2016; MEEKLAI et al., 2015; CHARALAMPOS; NEKTARIOS; STAVROS, 2015; KHENISSI; ESSALMI; JEMNI, 2013; CHANG; CHOU; CHEN, 2010) de jogos sérios educacionais de ensino a programação de computadores com linguagem C. Esta diferenciação foi realizada através do desenvolvimento de um jogo 3D do gênero FPS, com recursos e efeitos gráficos avançados e com a adoção de uma mecânica de utilização do jogo mais próxima da dos jogos digitais comerciais jogados pelos alunos.

### 5.3 TRABALHOS FUTUROS

Com base nos resultados obtidos por esta pesquisa, surgem novas possibilidades para trabalhos futuros. É importante destacar que não é possível afirmar com rigor científico que o jogo atingiu seus objetivos educacionais estabelecidos. Apesar das avaliações preliminares indicarem neste sentido é necessário a utilização de teorias e técnicas formais da pedagogia e psicologia para garantir que o resultado foi alcançado. São, portanto, avaliações preliminares que requerem um aprimoramento e possivelmente um estabelecimento de padrões da psicologia para aferir e avaliar a questão motivacional.

Também pode ser posto como melhoria ao processo a utilização direta de teorias formais de aprendizagem oriundas da pedagogia. Indiretamente a metodologia customizada para o jogo, utilizou algumas destas teorias, como a teoria da aprendizagem de Vygotsky, utilizada por um dos autores dos artigos científicos que serviram de base para a geração da metodologia customizada. Apesar disso, será importante aprimorar a metodologia inserindo de forma mais contundente alguma teoria formal de aprendizagem ou mesclando outras, para que estas teorias estejam presentes em todo o ciclo de desenvolvimento do software e permitir uma maior abrangência dos resultados.

A metodologia customizada pode ser melhorada em relação a sua adequação ao processo unificado (RUP). Apesar da metodologia ter sido customizada e estruturada baseada em grande parte nos conceitos de disciplinas, atividades, papéis, produtos de trabalho (artefatos) e tarefas do RUP, é importante uma análise

mais aprofundada desta metodologia para identificar com maior precisão estes conceitos e propor melhorias.

Outro elemento que pode ser explorado é a criação de uma abordagem que insira o aluno como parte integrante da equipe de desenvolvedores do jogo sério educacional. Desta forma, será interessante criar maneiras que permitam ao aluno criar um jogo sério sofisticado e que durante este processo ele aprenda sobre conceitos de programação.

## REFERÊNCIAS

- ABREU, P. F. de. et al. Integrating multi-agents in a 3D serious game aimed at cognitive stimulation. In: INTERNATIONAL CONFERENCE ON VIRTUAL AND MIXED REALITY: SYSTEMS AND APPLICATIONS, 2011. **Proceedings...** 2011. p. 217-226.
- ABT, C. **Serious Games**. New York: The Viking Press, 1970.
- ADAMO-VILLANI, Nicoletta; OANIA, Marcus; COOPER, Stephen. Using a serious game approach to teach secure coding in introductory programming: development and initial findings. **Journal educational technology systems**, v. 41, n.2, p.107-131, 2012-2013.
- AMANATIADOU, A.; VAN DE WEERD, I. Extending the reference method for game production: A situational approach. In: CONFERENCE IN GAMES AND VIRTUAL WORLDS FOR SERIOUS APPLICATIONS. 2009. **Proceedings...** 2009. p. 20-27.
- ANÁLISE DE DOMÍNIO. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2015. Disponível em: <[https://pt.wikipedia.org/w/index.php?title=An%C3%A1lise\\_de\\_dom%C3%ADnio&oldid=42172462](https://pt.wikipedia.org/w/index.php?title=An%C3%A1lise_de_dom%C3%ADnio&oldid=42172462)>. Acesso em: 21 mar. 2016.
- ASUNCION, H. et al. Serious game development as an iterative user-centered agile software project. In: INTERNATIONAL WORKSHOP ON GAMES AND SOFTWARE ENGINEERING, 1., 2011. **Proceedings...** 2011. p. 44-47.
- AZEVEDO, Rafael Faria de; PAULA, Bruno Campagnolo. Proposta de metodologia de aprendizado da programação de computadores através da recontextualização de jogos sérios no estilo Game & Watch. In: SBGAMES, 2011. **Proceedings...** 2011.
- BARBOSA, A. F. S. et al. A new methodology of design and development of serious games. **International Journal of Computer Games Technology**, 2014. Disponível em: <<http://dx.doi.org/10.1155/2014/817167>>. Acesso em: 1 out. 2015.
- BATTAIOLA, A. L. Jogos por computador: histórico, relevância tecnológica e mercadológica, tendências e técnicas de implementação. In: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA, 19., 2000. **Anais...** 2000. p. 83-122.
- BATTISTELLA, P.; WANGENHEIM, C. Gresse Von. Games for Teaching Computing in Higher Education: A Systematic Review. **IEEE Technology and Engineering Education (ITEE) Journal**, v.9, n.1, p.8-30, 2016.
- BERGERON, Bryan. **Developing serious games**. Hingham: Charles River Media, 2006.

BERRY, Michael; KÖLLING, Michael. Novis: A notional machine implementation for teaching introductory programming. In: FOURTH INTERNATIONAL CONFERENCE ON LEARNING AND TEACHING IN COMPUTING AND ENGINEERING - LATICE, 3., Mumbai, India. 2016. **Proceedings...** 2016.

BNDES - BANCO NACIONAL DE DESENVOLVIMENTO ECONÔMICO E SOCIAL. **Mapeamento da Indústria Brasileira e Global de Jogos Digitais. Relatório Final.** 2014.

\_\_\_\_\_. **I Censo da Indústria Brasileira de Jogos Digitais, com Vocabulário Técnico da IBDJ.** GEDIGames, NPGT, Escola Politécnica. São Paulo, 2014.

BORRO-ESCRIBANO, B.; BLANCO, A. del; TORRENTE, J.; ALPUENTE, I. M.; FERNÁNDEZ-MANJÓN, B. **Developing game-like simulations to formalize tacit procedural knowledge: the ONT experience.** Educational Technology Research and Development, v. 62, n. 2, p. 227-243, 2014.

BRATHWAITE, Brenda; SCHREIBER, Ian. **Challenges for game designers: non-digital exercises for video game designers,** 2009, Charles River Media, Boston, MA. 317p.

BROWN, Q.; LEE, F.; ALEJANDRE, S. **Emphasizing soft skills and team development in an educational digital game design course.** In: 4th international Conference on Foundations of Digital Games. Proceedings... 2009. p. 240-247.

BURKE, Kathleen T. **Computing Language and Thinking: Analysis, Design, and Assessment of Introductory Computer Science.** Workshops in the Liberal Arts Experience. 2016. Senior Projects Spring.

CAMPBELL, Greg. **What is a Domain Expert?** 2013. Disponível em: <<https://www.spiekerpoint.com/2013/02/26/what-is-a-domain-expert>>, Acesso em: 12/12/2015.

CANTERI, Rafael dos p. **Diretrizes para o design de aplicações de jogos eletrônicos para educação infantil de surdos.** 2014. 64 f. Dissertação (Mestrado) - Programa de Pós-graduação em Informática, Universidade Federal do Paraná - UFPR, 2014.

CARDENAS, Y. G. **Modelo de ontologia para representação de jogos digitais de disseminação do conhecimento.** 2014. 149 f. Dissertação (Mestrado) - Engenharia e Gestão do Conhecimento - Universidade federal de Santa Catarina - UFSC Florianópolis, 2014.

CATALANO, Chiara Eva; LUCCINI, Angelo Marco; MORTARA, Michela. **Guidelines for an effective design of serious games.** The International Journal Of Serious Games. Genoa, 2014. p. 8-43.

CHANG, Wen-Chih; CHOU Yu-Min; CHEN, Kuen-Chi. **Game-based digital learning system assists and motivates C programming language learners**. Networked Computing and Advanced Information Management (NCM), 2010, 6, International Conference on Date of Conference. p.704-709

CHARALAMPOS, Kyfonidis; NEKTARIOS, Moumoutzis; STAVROS, Christodoulakis. **Block-C: A block-based visual environment for supporting the teaching of C programming language to novices**. NHIBE - International Conference: on New Horizons in Industry, Business and Education, 9, Grécia, 2015.

CHENG, M.-T.; CHEN, J.-H.; CHU, S.-J.; & CHEN, S.-Y. **The use of serious games in science education: a review of selected empirical research from 2002 to 2013**. Journal of Computers in Education, 2015, 2(3), p.353–375.

CLUA, Esteban Walter Gonzalez; BITTENCOURT, J. R. **Desenvolvimento de Jogos 3D: Concepção, Design e Programação**. In: Antônio A. F. Loureiro; Marinho P. Barcellos;. (Org.). Jornadas de Atualização em Informática. São Leopoldo: Unisinos, 2005, v.1 , p.1-49.

COELHO, P. M. F.; COSTA, M. R. M. **Uma ferramenta digital que faz games educativos: o contexto brasileiro de ensino e aprendizagem**. RIED.Revista Iberoamericana de Educación a Distância, 19(2), 2016, p. 53-70.

COENEN, T.; MOSTMANS, L; NAESSENS, K. **MuseUs: Case study of a pervasive cultural heritage serious game**. Journal on Computing and Cultural Heritage (JOCCH), v. 6, n. 2, p. 8, 2013.

CRAWFORD, C. (1982). **The art of digital game design**. Washington State University, Vancouver, 1982.

CREATIVESKILLSET. **Lista de profissionais envolvidos no desenvolvimento de um jogo digital**. Disponível em:

<<http://creativeskillset.org/search/59?q=&tags%5B%5D=Games&tags%5B%5D=&tags%5B%5D=>>. Acesso em: 27/06/2016.

DJAOUTI, Damien; ALVAREZ, Julian; JESSEL, Jean-Pierre. **Classifying serious games: the G/P/S model**. In Handbook of Research on Improving Learning and Motivation Through Educational Games: Multi-disciplinary Approaches, P. Felicia, Ed., IGI Global, 2011. p. 118-136.

EAGLE, M. J.; BARNES, T. **A learning objective focused methodology for the design and evaluation of game-based tutors**. In: 43rd ACM Technical Symposium on Computer Science Education. Proceedings... 2012. p. 99-104.

EDUCARED PRONIÑO. Grupo de formação de dinamizadores. 2008. Disponível em: <<https://sites.google.com/site/webquestweb2/objetivospedag%C3%B3gicos>>. Acesso em: 11 set. 2015.

ENTRETENIMENTO EDUCATIVO (edutainment). In: WIKIPÉDIA a enciclopédia livre. Flórida: Wikimedia Foundation, 2015. Disponível em: <[https://pt.wikipedia.org/wiki/Entretenimento\\_educativo](https://pt.wikipedia.org/wiki/Entretenimento_educativo)>. Acesso em: 21 maio. 2016.

FALCKEMBACH, Gilse A. Morgental; ARAUJO, Fabrício Viero de. **Aprendizagem de algoritmos: dificuldades na resolução de problemas**. Santa Maria, RS: ULBRA, 2003.

FULLERTON, Tracy; SWAIN, Christopher; HOFFMAN, Steven S. **Game design workshop: a playcentric approach to creating innovative games**. 2. ed. Massachusetts: Elsevier, 2008, 496p.

GATES, Bob, **Game design**. 2. ed. Boston, MA : Thomson Course Technology, 2004, 377 p.

GEBREMICHAEL, D. **An evaluation of gamification to assess students' learning on their understanding of first year computer science programming module**. 2016. 121 f. Dissertação (Mestrado) - Computing, Advanced Software Development. Dublin Institute of Technology, Dublin, 2016.

GONÇALVES, Ricardo E. Ferreira. **Jogo digital para o ensino dos fundamentos da programação**. Dissertação (Mestrado) - Engenharia Informática e Computação – FEUP. Porto, 2011.

HAKULINEN, L. Using serious games in computer science education. In: KOLI CALLING INTERNATIONAL CONFERENCE ON COMPUTING EDUCATION RESEARCH, 11., 2011, Koli, Finlândia. **Proceedings...** Nova Iorque, USA: ACM. 2011. p.83-88.

HARTEVELD, C. **Triadic game design**. Londres: Springer-Verlag, 2011. 316p.

HUIZINGA, Johan. **Homo ludens: o jogo como elemento da cultura**. 5. ed. São Paulo: Perspectiva, 2007. 162p.

ISMAILOVIĆ, D. et al. Adaptive serious game development. In: INTERNATIONAL WORKSHOP ON GAMES AND SOFTWARE ENGINEERING. 2., 2012. **Proceedings...** 2012. p. 23-26.

LAZAROU, D. Using cultural-historical activity theory to design and evaluate na educational game in science education. **Journal of Computer Assisted Learning**, v.27, n.5, p. 424–439, 2011.



LUKOSCH, H.; RUIJVEN, T. van; VERBRAECK, A. The participatory design of a simulation training game. In: WINTER SIMULATION CONFERENCE. 2012. **Proceedings...** 2012. p. 142.

KASHIWAKURA, Eduardo Yukio. **Jogando e aprendendo**: um paralelo entre vídeo games e habilidades cognitivas. 2008. 100 f. Dissertação (Mestrado) - Tecnologias da Inteligência e Design Digital. Pontifícia Universidade Católica de São Paulo – PUC, São Paulo, 2008.

KHENISSI, M. A; ESSALMI, F.; JEMNI M. Presentation of a Learning Game for Programming Languages Education. In: INTERNATIONAL CONFERENCE ON ADVANCED LEARNING TECHNOLOGIES, 13, 2013. **Proceedings...** 2013. p.324-326.

KINNUNEN, P.; MALMI, L. Why students drop out cs1 course? In: ICER'06: INTERNATIONAL WORKSHOP ON COMPUTING EDUCATION RESEARCH, 2., 2006. **Proceedings...** Nova Iorque, USA: ACM, 2006. p.97-108.

KISHIMOTO, T. M. **O jogo e a educação infantil**. São Paulo: Pioneira, 1997. 62 p.

LAMOTHE, Andre. **Designing video games**. Disponível em: <<http://www.dummies.com/how-to/content/designing-video-games.html>>. Acesso em: 30 nov. 2015.

LINGUAGEM DE PROGRAMAÇÃO. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2015. Disponível em: <[https://pt.wikipedia.org/wiki/Linguagem\\_de\\_programa%C3%A7%C3%A3o](https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o)>. Acesso em: 11 abr. 2016.

LOBO, Ronilson M. **Metodologia para design de jogos tabletop baseada em fotografia**: Lucas's toys® - interface lúdica para jogos. 2013. 109 f. Dissertação (Mestrado)- Universidade Salvador - UNIFACS, Salvador, 2013.

MACHADO NETO, Olibário J. **Usabilidade da interface de dispositivos móveis**: heurísticas e diretrizes para o design. 2013. 118 f. Dissertação (Mestrado) - Ciências de Computação e Matemática Computacional - Universidade de São Paulo - USP, São Carlos, 2013.

MARCOS, A.; ZAGALO, N. Instantiating the creation process in digital art for serious games design. **Entertainment Computing**, v. 2, n. 2, p. 143-148, 2011.

MARGULIEUX, Lauren E.; CATRAMBONE, Richard; GUZDIAL, Mark. Employing subgoals in computer programming education. **Computer Science Education Journal**, 2016, p.1–24.

MASTROCOLA, Vicente Martin. **Ludificador**: um guia de referências para o game designer brasileiro. São Paulo, 2012. Disponível em:  
<<http://www.ludificador.com.br/ludificador.pdf>> Acesso em:8 set.2015.

MATTAR, J. **Games em educação**: como os nativos digitais aprendem. São Paulo: Pearson Prentice Hall, 2010. 208p.

MALLIARAKIS, C.; SATRATZEMI, M.; XINOGALOS, S. Designing educational games for computer programming: a holistic framework. **Academic Journal | Electronic Journal of e-Learning**, v.12 n.3 p.281-298, 2014.

MEEKLAI, Satjapong et al. CROCOTILE: a learning environment for C language learners with tile programming. In: ICT INTERNATIONAL STUDENT PROJECT CONFERENCE (ICT-ISPC2015), 2015, 4., Tokyo, Japan. **Proceedings...** 2015.

MICHAEL, D.; CHEN, S. **Serious games**: games that educate, train, and inform. Canada: Thomson Course Technology PTR, 2006, 312 p.

MILJANOVIC, Michael A.; BRADBURY, Jeremy S. **Robot on!**: a serious game for improving programming comprehension. Proceedings, International Workshop on Games and Software Engineering, 5, 2016, Austin, Texas, USA. p.33-36.

MITAMURA, Tamotsu; SUZUKI, Yasuhiro; OOHORI, Takahumi. Serious games for learning programming languages. In: SYSTEMS, MAN, AND CYBERNETICS (SMC), 2012 IEEE INTERNATIONAL CONFERENCE ON. IEEE, 2012. **Proceedings...** 2012.

NOVAK, Jeannie. **Desenvolvimento de games**. São Paulo: Cengage Learning, 2010.

\_\_\_\_\_. **Desenvolvimento de games**. 2. ed. São Paulo: Cengage Learning, 2011.472p.

\_\_\_\_\_. **Game development essentials**: an introduction. 3. ed. São Paulo: Cengage Learning, 2012. 512p.

OLIVEIRA, H. C.; HOUNSELL, M. S.; KEMCZINSKI, Avaniilde. Mapeamento sistemático de metodologias de desenvolvimento centrado no usuário para jogos sérios. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO(SBIE), 25., 2014, Dourados - MS. **Anais...** Porto Alegre (RS): SBC - Sociedade Brasileira de Computação, 2014. v. 1. p. 1-10.

OLIVEIRA, Helder Cognaco de. **Uma metodologia participativa para o desenvolvimento de jogos sérios**. 2016. 185 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação Aplicada. Universidade do Estado de Santa Catarina – UDESC. Joinville, Santa Catarina, 2016.

PARALIČ, Marek; PIETRIKOVÁ, Emilia. Learning by game creation in introductory programming course: 5-year-long study In: ICETA 2014 : IEEE INTERNATIONAL CONFERENCE ON EMERGING ELEARNING TECHNOLOGIES AND APPLICATIONS: STARÝ SMOKOVEC, 12., 2014. **Proceedings...** 2014.p.391-396.

PARREIRA, Daniel M.; PETERSEN, A.; CRAIG, M. PCRS-C: helping students Learn C. In: ACM CONFERENCE ON INNOVATION AND TECHNOLOGY IN COMPUTER SCIENCE EDUCATION. ITICSE .15., 2015, New York, USA. **Proceedings...** 2015. p. 347-347.

PAULA, B. H. de; VALENTE, José A. Jogos digitais e educação: uma possibilidade de mudança da abordagem pedagógica no ensino formal. **Revista Ibero-americana de Educação**, v.70, n.1, p.9-28, 2016.

PENTEADO,C. S.; DAMASCENO,E. F. Contribuições da aplicação de jogos digitais no ensino de lógica de programação para o ensino médio integrado em informática. **Revista E.T.C**, 1. Ed. 10, 2014.

PESSINI, A. et al. O uso de jogos sérios na educação em informática: um mapeamento sistemático. In: CONFERÊNCIA INTERNACIONAL SOBRE INFORMÁTICA NA EDUCAÇÃO - TISE, 19., 2014, Fortaleza. **Proceedings....** Chile: Universidad de Chile, 2014. v. 10. p. 537-541.

PIETRUCHINSKI, M. Hoeldtke; NETO, João C.; REINEHR, Sheila. A. Os jogos educativos no contexto do SBIE: uma revisão sistemática de Literatura. In: SBIE – XVII WIE, 22., 2011. Aracaju. **Anais...** 2011. Disponível em: <[http://www.br-ie.org/sbie-wie2011/SBIE-Trilha3/93164\\_1.pdf](http://www.br-ie.org/sbie-wie2011/SBIE-Trilha3/93164_1.pdf)> Acesso em: 24 mar. 2015.

PMBOK. **Um guia do conhecimento em gerenciamento de projetos**. 5. ed. EUA: Project Management Institute, 2013.

PRENSKY, Marc. **Digital game -based learning: practical ideas for the application of digital game -based learning**. St. Paul: Paragon House, 2007.

\_\_\_\_\_. **Don't bother me, mom,I'm learning!**: how computers and video games are preparing your kids for 21<sup>st</sup> century success and how you can help! St. Paul: Paragon House Publishers. 2008.

RANKIN, Y. A. et al. User centered game design: evaluating massive multiplayer online role playing games for second language acquisition. In: 2008 ACM SIGGRAPH SYMPOSIUM ON VIDEO GAMES, 2008. **Proceedings...** 2008. p. 43-49.

RENZI, Adriano B.; FREITAS S. Aplicação de think-aloud protocol em teste de usabilidade na procura de livros em livrarias online: recomendações de leitores. In:

CONGRESSO INTERNACIONAL DE ERGONOMIA E USABILIDADE DE INTERFACES HUMANO-COMPUTADOR (USIHC),10., 2010, Rio de Janeiro. **Anais...** Rio de Janeiro: PUC-RJ, 2010.

ROCHA, Heloisa Vieira da; BARANAUSKAS, Maria Cecília C. **Design e avaliação de interfaces humano-computador**. Campinas: UNICAMP–NIED, 2003. 241p.

ROCHA, Rafaela V. da; ARAUJO, Regina B. de. Metodologia de design de jogos sérios para treinamento. In: SIMPÓSIO BRASILEIRO DE JOGOS E ENTRETENIMENTO DIGITAL (SBGAMES), 12., 2013. **Anais...** 2013. v. 1. p. 63-72.

ROCHA, R. V. From behavior modeling to communication, 3d presentation and interaction: an M&S life cycle for serious games for training. In: IEEE/ACM INTERNATIONAL SYMPOSIUM ON DISTRIBUTED SIMULATION AND REAL TIME APPLICATIONS – DS-RT, 16., 2012, Dublin, Irlanda. **Proceedings...** 2012. p. 132-139.

RODRIGUES, H. F, MACHADO, L. d. S., VALENÇA, A. M. G. Definição e aplicação de um modelo de processo para o desenvolvimento de serious games na área de saúde. In: CSBC, 2010. **Proceedings...** 2010. 1532-1541.

RÜPPEL, U.; SCHATZ, K. Designing a BIM-based serious game for fire safety evacuation simulations. **Advanced Engineering Informatics**, v. 25, n. 4, p. 600-611, 2011.

SALEN, Katie; ZIMMERMAN, Eric. **Rules of play - game design fundamentals**. The Massachusetts Institute of Technology (MIT) Press, Londres, 2004, 688 p.

SANTOS, Mayara W. A. dos; SOUTO, Analiz. **Jogos voltados para o ensino-aprendizagem de programação: uma revisão sistemática**. [S.l.]: SBIE 2014.

SANTAELLA, Lucia. O papel do lúdico na aprendizagem. **Revista Teias**, ProPEd/UERJ, Rio de Janeiro, v. 13, n. 30, p. 163-173. set./dez. 2012.

SCHELL, Jesse. **The art of game design: a book of lenses**. Burlington, MA: Elsevier, 2008. 489p.

SCHUYTEMA, Paul. **Design de games: uma abordagem prática**. São Paulo: Cengage Learning, 2008. 472 p.

SHIRATUDDIN, M. F. Integrating computer game-based learning into construction education. In: CONFERENCE ON INFORMATION TECHNOLOGY AND MULTIMEDIA. 2011. **Proceedings...** 2011. p. 1-6.

SILVEIRA, Renato S. **Estudo de uma ferramenta de autoria multimídia para a elaboração de jogos educativos**. Porto Alegre: Instituto de informática, UFRS, 1999.

SMITH, Roger. **A history of serious games roger**. 2009. Disponível em: <[http://www.modelbenders.com/papers/RSmith\\_History\\_Serious\\_Games.pdf](http://www.modelbenders.com/papers/RSmith_History_Serious_Games.pdf)>. Acesso em: 12 jan. 2015.

SOARES, Cecília de A.; ZALUAR, Pedro Lamy. **Uma abordagem baseada em projetos para o ensino de programação**: investigações sobre a tecnologia Android. 2015. 84 f. (Graduação) – Sistemas de Informação, Universidade Federal do Rio de Janeiro - UFRJ, Rio de Janeiro, 2015.

SOFTEX - ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. **Cadernos temáticos do observatório**: mercado de trabalho e formação de mão de obra em TIC. 2013. Disponível em: <[http://www.softex.br/wp-content/uploads/2015/10/Cadernos\\_Tematicos\\_Mercado\\_de\\_trabalho.pdf](http://www.softex.br/wp-content/uploads/2015/10/Cadernos_Tematicos_Mercado_de_trabalho.pdf)>. Acesso em: 12/04/2016.

SOMMERVILLE, Ian. **Engenharia de software**. 8. ed. São Paulo: Pearson Addison-Wesley, 2007. 552 p.

\_\_\_\_\_. **Engenharia de software**. 9.ed. São Paulo: Pearson Education, 2011. 529 p.

TIRO em primeira pessoa. In: WIKIPÉDIA a enciclopédia livre. Flórida: Wikimedia Foundation, 2016. Disponível em: <[https://pt.wikipedia.org/wiki/Tiro\\_em\\_primeira\\_pessoa](https://pt.wikipedia.org/wiki/Tiro_em_primeira_pessoa)>. Acesso em: 9 mar. 2016.

TORRENTE, J. et al. Development of game-like simulations for procedural knowledge in healthcare education. **IEEE Transactions on Learning Technologies**, v. 7, n. 1, p. 69-82, 2014.

VAN DER ZEE, D. J.; HOLKENBORG, B.; ROBINSON, S. Conceptual modeling for simulation-based serious gaming. **Decision Support Systems**, v. 54, n. 1, p. 33-45, 2012.

WTHREEX. Rational Software Corporation. **Disciplina**. Disponível em: <[http://www.wthreex.com/rup/manuals/intro/kc\\_discipline.htm](http://www.wthreex.com/rup/manuals/intro/kc_discipline.htm)>. Acesso em: 11 set. 2015.

ZYDA, M. From Visual Simulation to Virtual Reality to Games. **IEEE Computer**, v. 38, n.9, p. 25-32, 2005.

## APÊNDICE A – METODOLOGIA DE DESENVOLVIMENTO DE UM JOGO SÉRIO EDUCACIONAL: VERSÃO COM APLICAÇÃO DAS MELHORIAS

<b>PRÉ-PRODUÇÃO</b>	
<b>DEFINIR O PÚBLICO ALVO DO JOGO</b>	
Deve ser decidido para quem o jogo é destinado. Se é para alunos do ensino médio, de graduação ou pós-graduação, alunos de formação para desenvolvimento profissional e de carreira. O público-alvo também determina os benefícios e o tipo de experiência que será apresentado aos estudantes. Considerações devem ser dadas em questões tais como se aluno deve jogar o jogo de forma dedicada ou poderá ser apenas um jogador ocasional. O grau de instrução, a idade dos estudantes, se os alunos possuem alguma deficiência ou são de outras culturas também deve ser observado. (SHIRATUDDIN, 2011)	
<b>IDENTIFICAR NECESSIDADES E EXPECTATIVAS DO PÚBLICO ALVO.</b>	
Identificar necessidades e expectativas do público alvo. Realizar entrevistas com professores e alunos para levantar as expectativas dos mesmos em relação ao jogo sério. (LAZAROU, 2011)	
<b>COMPREENDER MELHOR O DOMÍNIO DO PROBLEMA POR PARTE DOS DESENVOLVEDORES</b>	
Efetuar uma melhor compreensão do domínio do problema pelos desenvolvedores do jogo para definir um conteúdo que seja significativo e desafiador para o aprendizado dos alunos. Realizar entrevistas com estudantes e professores para que eles reflitam sobre a aprendizagem anterior, pesquisa bibliográfica, observação de aulas e métodos de ensino. Idealmente, os desenvolvedores devem se envolver em aulas individuais, observando o que funciona e o que não funciona com atenção especial para áreas onde está ocorrendo maior confusão no aprendizado dos alunos através da identificação de lacunas no conhecimento. Utilizar protocolo think-aloud onde os alunos pensem em voz alta sobre o seu processo de resolução de problemas. Depois efetuar uma observação da resolução de problemas efetuando uma identificação de lacunas no conhecimento, que irão fornecer informações sobre o projeto e recursos do jogo. (EAGLE; BARNES, 2012) Os desenvolvedores deverão observar instrutores e participar de sessões de tutoria para ver que metáforas e métodos são usados atualmente para ensinar, com especial atenção para as lacunas no conhecimento do público-alvo.. (EAGLE; BARNES, 2012). Realizar entrevistas com os especialistas de domínio em que o jogo sério está inserido para melhorar e compreender o processo. (BORRO;ESCRIBANO et al. 2014)	
<b>VERIFICAR RECURSOS EDUCACIONAIS PRÉ-EXISTENTES</b>	
Verificar com educadores a existência de recursos compartilhados para os problemas e exercícios que atendam aos objetivos de aprendizagem importantes para área alvo do jogo sério. (EAGLE ; BARNES, 2012)	
<b>IDENTIFICAR OS RESULTADOS QUE OS ALUNOS DEVEM SER CAPAZES DE ALCANÇAR</b>	
Desenvolvedores e professores deverão escrever os objetivos de aprendizagem para as aulas, onde os resultados devem ser observáveis e mensuráveis. (EAGLE; BARNES, 2012)	
<b>SELECIONAR CASOS DE ENSINO DESAFIADORES</b>	
Selecionar casos de ensino para incluir no jogo sério que não sejam óbvios para desafiar o aluno de alguma forma especial. (BORRO-ESCRIBANO et al. 2014)	
<b>CRIAR PÓS-TESTES QUE OS ALUNOS DEVEM SER CAPAZES DE COMPLETAR DEPOIS DE JOGAR O JOGO.</b>	
Projetar pós-testes usando exercícios de livros didáticos, ou questões de prova. As perguntas devem tentar medir a profundidade do conhecimento do aluno. Recomenda-se o desenvolvimento de questões que consideram a dificuldade com base na taxonomia de objetivos educacionais de Bloom. Após o pós-teste ser criado, o mesmo deve ser verificado consultando professores antes de ser utilizado. Este requisito também ajuda a garantir que as atividades no jogo são válidas, isto é, que o jogo ensina o que ele pretende. Os desenvolvedores devem procurar jogos, atribuições, e materiais de aprendizagem online que sejam eficazes para o material alvo. Métodos comuns para a resolução de problemas devem ser anotados para incorporação no projeto. (EAGLE; BARNES, 2012)	
<b>DEFINIR A ÁREA QUE SERÁ O FOCO DO JOGO</b>	
Deverão ser definidos os objetivos de aprendizagem do jogo em relação à área foco estabelecida. Conforme destaca (SHIRATUDDIN, 2011) existem diferentes áreas que um jogo sério pode ser utilizado para complementar o ensino convencional e cada área terá seus próprios objetivos de aprendizagem e o projeto do jogo deve considerá-los.	
<b>IDENTIFICAR O CONCEITO PRINCIPAL DO JOGO SÉRIO</b>	
A fim de estabelecer uma abordagem pedagógica, o professor da disciplina deve efetuar um estudo sobre: O conteúdo do jogo – Objetivos - Resultados esperados - Como os alunos serão avaliados - As estratégias de ensino – O sequenciamento do aprendizado - As tarefas e atividades que serão realizados no jogo - As habilidades que os alunos irão adquirir. (SHIRATUDDIN, 2011)	
<b>ESCOLHER O GÊNERO DO JOGO SÉRIO</b>	
A escolha de um gênero adequado deve ser selecionada para coincidir com as estratégias de ensino e aprendizagem estabelecidas. Os seguintes tipos de gêneros devem ser analisados e um escolhidos:(SHIRATUDDIN, 2011, p.4)	
<b>Gênero</b>	<b>Tipos de atividades e desafios</b>
<b>Jogos de ação</b>	Desafios Físicos
<b>Jogos de estratégia em tempo real</b>	Estratégia, táticas e desafios logísticos
<b>Jogos de RPG</b>	Táticas, logística, experiências e desafios econômicos
<b>Simulações do mundo real</b> (jogos esportivos, simulações de veículos e simulações)	Desafios físicos e táticos
<b>Construção e gestão de jogos</b>	Desafios econômicos e conceituais
<b>Jogos de Aventura</b>	Exploração e desafios de resolução de quebra-cabeças
<b>Jogos de Quebra Cabeças</b>	Lógica e desafios conceituais

## APÊNDICE A – METODOLOGIA DE DESENVOLVIMENTO DE UM JOGO SÉRIO EDUCACIONAL: VERSÃO COM APLICAÇÃO DAS MELHORIAS (CONTINUAÇÃO)

<b>DEFINIR A HISTÓRIA PARA O JOGO</b>
Definir se o jogo terá uma história e se esta história será básica ou complexa e se os alunos poderão influenciar a mesma. (SHIRATUDDIN, 2011)
<b>CRIAR UM DOCUMENTO COM VOCABULÁRIO COMPARTILHADO</b>
<p>Criar um vocabulário compartilhado de termos e de conhecimentos relacionados com o projeto do jogo sério e que seja entendido por todas as partes interessadas. O objetivo é facilitar a comunicação entre os especialistas com diferentes formações. (BORRO-ESCRIBANO et al. 2014)</p> <p>O vocabulário pode ser representado usando uma notação formal ou uma representação visual (diagramas, fluxogramas, linguagem visual específica). Reuniões informais entre os participantes devem ser realizadas para revisar documentos. Na reunião, especialistas de domínio devem demonstrar como procedimentos são executados, mostrar vídeos ou quaisquer outros materiais que ilustrem o domínio do jogo e fornecer outros materiais para os especialistas de jogos. Os especialistas de jogos devem mostrar um jogo relevante com exemplos para ajudar os especialistas de domínio entender a gama de recursos expressivos que podem ser usados para transformar procedimentos em um jogo. (TORRENTE et al. 2014)</p>
<b>CRIAR O DOCUMENTO DE REQUISITOS DO JOGO</b>
<p>Obter um documento de especificação detalhado que descreve as características do público alvo, o conhecimento relacionado com a área que o jogo sério deve contemplar e como ele vai ser representado, o ambiente e as configurações a que o jogo se destina e quaisquer outras informações que sejam necessárias para formalizar a área do jogo. (BORRO;ESCRIBANO et al. 2014). Descrever os aspectos técnicos do jogo:</p> <ul style="list-style-type: none"> <li>• Os requisitos de sistema, como o hardware / dispositivos que terão que ser utilizados para executar o jogo.</li> <li>• Eventuais restrições que podem se aplicar ao produto final, por exemplo, licenciamento, comercial, de código aberto, idade, etc.</li> <li>• Requisitos técnicos para os aspectos visuais do jogo.</li> <li>• O conteúdo de áudio. (SHIRATUDDIN, 2011)</li> </ul> <p>Estruturar o documento da seguinte forma:  <b>INTRODUÇÃO:</b> objetivos, convenções, termos e abreviações, escopo, visão geral.  <b>DESCRIÇÃO GERAL:</b> requisitos funcionais, requisitos não funcionais.</p>
<b>CRIAR O DOCUMENTO DE DESIGN E CONCEITO DO JOGO</b>
<p>Elaborar documento de concepção técnica de arte e produção do jogo e:</p> <p>Definir Prêmio / Recompensa  Definir se jogo terá apenas um jogador durante o jogo (Single-Player) ou vários jogadores durante o jogo (Multi-Player)  Definir Plataforma do jogo  Definir Interface do usuário  Definir Estória  Definir Concepção de arte  Áudio, som, iluminação  Arquitetura do jogo  ( SHIRATUDDIN, 2011)</p> <p>Este documento descreve a visão para o jogo, conteúdo, horários e estratégia de implementação. Ele fornece diretrizes para o projeto como um todo, e as instruções para a equipe de desenvolvimento do jogo (produtores, designers de jogos, artistas e programadores) (SHIRATUDDIN, 2011)</p> <p>Os documentos de conceito e design do jogo devem estar estruturados segundo Schyutema (2008).  Documento de conceito do jogo:  Nome do jogo, plataforma, jogadores, Gênero, High Concept, Objetivo e recursos  Documento de design:  Introdução. Visão geral essencial:Resumo, Aspectos Fundamentais, Golden Nuggets. Contexto do jogo: História do game, eventos anteriores, principais jogadores. Objetos essenciais do game, Conflitos e Soluções, Inteligência artificial, Fluxo do jogo, Controles</p>
<b>CRIAR ELEMENTOS E CARACTERÍSTICAS DO JOGO</b>
<p>Criar elementos e características do jogo derivados de diretrizes de design sugeridos através de investigação relacionada com a psicologia, a semiótica e a interface, dentro do campo de interface homem computador(IHC). Avaliar os elementos do jogo principalmente para o projeto da interface do jogo. Realizar entrevista para conhecer as práticas dos alunos quando eles jogam com computador em casa e considerar possíveis aspectos para conceituar características do jogo a ser construído. Estas características colhidas junto aos alunos podem ser a de preferência em jogos em primeira ou terceira pessoa, enredo com a presença de mistérios com elementos escondidos, resolver problemas, perguntas e respostas pertinentes, desafios para serem capazes de continuar com as etapas do jogo, etc. (LAZAROU, 2001)</p> <p>Indicar quais serão as atividades básicas que os alunos estarão fazendo durante o jogo. Se estarão coletando, perseguindo, combatendo, esquivando-se, construindo, etc. (SHIRATUDDIN, 2011)</p> <p>A experiência dos especialistas do domínio é fundamental, pois eles sabem como os alunos normalmente interagem com o ambiente e podem apontar elementos que normalmente criam confusão entre os estudantes para serem usados como "pistas falsas" durante o jogo. Esta tarefa requer mais interação entre especialistas de jogo e especialistas de domínio, devido à dificuldade inerente de tornar o conhecimento tácito dos especialistas explícito. (TORRENTE et al., 2014)</p>

## APÊNDICE A – METODOLOGIA DE DESENVOLVIMENTO DE UM JOGO SÉRIO EDUCACIONAL: VERSÃO COM APLICAÇÃO DAS MELHORIAS (CONTINUAÇÃO)

<p><b>DESENHAR STORYBOARDS.</b></p> <p>A equipe tem que desenhar storyboards que identificam os elementos importantes, interfaces e interações que devem ser apresentadas ao usuário. Isso inclui cenário, painel de instrumentos, Objetos 2D e 3D, sons, interfaces interativas e da história, entre outros. Estes elementos devem fornecer ao usuário interface intuitiva que os permita desempenhar suas funções de acordo com os procedimentos executados no mundo real. (ROCHA et al. 2012)</p>
<p><b>CRIAR E MELHORAR UM SCRIPT DO JOGO</b></p> <p>O script(modelo de simulação) do jogo indica a transformação do documento de especificação de requisitos e do documento de design, para os elementos de projeto do jogo O game designer deve compreender o processo, desenhar o storyboard da simulação do jogo e verificar se ele fornece o entretenimento adequado e valor educativo. O game designer deve saber como integrar as habilidades de design, conhecimentos e valores para permitir que o jogador possa construir um modo de pensar e de resolução de problemas que seja semelhante à abordagem dos peritos na área. (BORRO;ESCRIBANO et al. 2014)</p> <p>Um script não é apenas uma sequência de passos; é necessário melhorá-lo para fornecer ao aluno uma experiência que promova imersão. Quando o jogo começa, os alunos precisam estar situados. A falta de contextualização adequada pode danificar seriamente a experiência de jogo, fazendo com que os alunos se sintam perdidos, e diminuindo o rendimento educacional de jogar o jogo. Primeiro, os alunos precisam ter uma compreensão clara dos principais objetivos do jogo (isto é, como ter sucesso) e as regras gerais (o que você deve e não deve fazer para ter sucesso). Durante o jogo, o aluno será confrontado com diferentes situações ou problemas que tem de ser resolvido. (TORRENTE et al. 2014)</p> <p>Em uma primeira iteração criar protótipos mock-up (ou seja, simulações com recursos de arte provisórios), que permite que o especialista de domínio verifique se o conhecimento explícito e tácito da área do jogo foram devidamente capturados. (BORRO;ESCRIBANO et al. 2014)</p>
<p><b>PRODUÇÃO</b></p>
<p><b>PROJETAR O MUNDO VIRTUAL</b></p> <p>A descrição de alto nível do ambiente é traduzida em elementos do jogo. O modelo de ambiente de jogo pode ser produzido também com a ferramenta WEEV ou usando qualquer outra representação gráfica (TORRENTE et al. 2014)</p>
<p><b>INSERIR ELEMENTOS MOTIVACIONAIS, DE DECISÃO, ARMADILHAS E FEEDBACKS.</b></p> <p>Os erros comuns devem ser progressivamente incluídos como elementos do jogo. O objetivo é conduzir os alunos a atenção os aspectos que são de maior importância. Também é necessário adicionar armadilhas e entrelaçar elementos motivacionais para aumentar o engajamento. (BORRO;ESCRIBANO et al. 2014)</p> <p>O jogo deve:</p> <p>(1) Orientar o usuário na descoberta de informações disponíveis relacionada com a decisão e as diferentes opções disponíveis;</p> <p>(2) Proporcionar ao aluno uma forma simples de executar essas ações no jogo. A maneira mais simples de apresentar opções é o uso questões de múltipla escolha apresentadas no texto, apesar destas opções poderem também ser apresentadas visualmente utilizando animações (a um ligeiro maior custo de produção). O jogo também deve fornecer mecanismos para a execução de decisões. (TORRENTE et al. 2014)</p> <p>Indicar que uma interação desejada do jogador foi realmente executada. Comentários também devem apoiar e facilitar a reflexão sobre os acontecimentos que ocorrem no jogo e, especialmente, destacar os efeitos sobre decisões erradas dos procedimentos. O feedback também contribui para criar uma percepção contínua do progresso, uma característica inerente de bons jogos que evita frustração e encoraja o jogador a ir em frente, entre outros benefícios.</p> <p>Na maioria dos casos, o feedback deve ser simples (limitado a áudio ou pistas visuais simples), curto e não-intrusivo a fim de evitar a quebra do ritmo do jogo. O feedback deve ser mais explícito e amplo em partes do processo, que são especialmente complexo. Isso ajuda os alunos refletirem sobre os conceitos subjacentes de forma mais aprofundada e transferirem os conhecimentos adquiridos a situações do mundo real. Um pequeno vídeo apresentando um aspecto particular de um procedimento pode ser mais útil e envolvente. Ao lidar com o feedback para ações incorretas, o tempo pode ser ajustado de modo a refletir as consequências de manipulações válidas e manipulações inválidas, tanto no próprio processo e nos seus possíveis subprodutos. É importante para assegurar que o jogo baseia-se nestas situações, fazer os alunos "pagar um preço" por erros (por exemplo, ter que começar de novo) e ajudá-los a evitar o mesmo erro próxima vez. Quando o jogo for concluído, o estudante pode ser apresentado com feedback definitivo para fins de autoavaliação, incluindo uma lista de todos os erros e ações incorretas realizadas. Esta informação pode ser utilizada para identificar potenciais fraquezas no jogo, ajustar a formalização do procedimento e reforçar a aprendizagem sem a necessidade de intervenção de professores. (TORRENTE et al., 2014)</p> <p>Aplicar técnicas para aumentar o engajamento e a motivação do estudante, Reforçar a aprendizagem utilizando tecnologia muito avançada ou gráficos 3D atraentes para contribuir para criar uma experiência imersiva no ambiente no jogo, Utilização de heurísticas quantificáveis para mostrar o progresso dos alunos.(TORRENTE et al., 2014)</p>
<p><b>SOLICITAR INFORMAÇÕES ADICIONAIS AOS ESPECIALISTAS DE DOMÍNIO</b></p> <p>No entanto, em contraste com o processo de especificação, a iniciativa aqui deve ser liderada pelos designers de jogos, que são responsáveis por solicitar informações adicionais, filtrando os aspectos que não são compatíveis com o gameplay(jogabilidade) e identificar aspectos que ainda não foram capturados. (BORRO;ESCRIBANO et al, 2014)</p>



## APÊNDICE A – METODOLOGIA DE DESENVOLVIMENTO DE UM JOGO SÉRIO EDUCACIONAL: VERSÃO COM APLICAÇÃO DAS MELHORIAS (CONTINUAÇÃO)

<p><b>GERAR A PARTIR DOS REQUISITOS E DOCUMENTO DE DESIGN, PROTÓTIPOS EVOLUCIONÁRIOS DO JOGO.</b></p> <p>Efetuar uma prototipagem iterativa que permita uma avaliação gradativa e testes planejados para facilitar análise e tarefas de projeto. É mais simples para os especialistas de domínio encontrar imprecisões ou aspectos negligenciados no protótipo do que analisar desenhos ou documentos. Permite também aos especialistas de jogo testar rapidamente a mecânica do jogo e identificar possíveis inconsistências. Diferentes protótipos podem ser criados para diferentes fins. Protótipos mock-up permitem a avaliação rápida da exatidão do procedimento formalizado. Protótipos intermediários devem ser utilizados principalmente para suscitar comentários sobre o design do jogo e das próximas iterações. Protótipos finais estáveis devem ser usados para a avaliação do usuário final. (TORRENTE et al, 2014)</p> <p>Um protótipo inicial do jogo deve ser desenvolvido principalmente com base nas características e elementos, derivados a partir de propostas e recomendações feitas por alunos e professores durante as entrevistas. O protótipo do jogo deve também ser formado por estratégias complementares de design incluindo diretrizes de projeto decorrentes da pesquisa Interface Homem computador(IHC), as diretrizes de design derivados do cotidiano dos alunos e experiências e hábitos ao jogar videogames em casa, e diretrizes de projeto derivadas da área alvo do jogo sério. (LAZAROU, 2011)</p> <p>Protótipos dos métodos de ensino criados na fase de projeto devem ser transformados em jogo. Para ser eficaz para a aprendizagem, o ambiente do jogo deve promover os mesmos comportamentos que são necessários para resolver o pós-teste que os alunos devem ser capazes de completar depois de jogar. Desenvolvedores iterativamente devem construir partes testáveis do protótipo, com o feedback frequente dos alunos, dos outros desenvolvedores e instrutores. (EAGLE; BARNES, 2012)</p>
<p><b>EFETUAR TESTES ATRAVÉS DOS PROTÓTIPOS DESENVOLVIDOS</b></p> <p>Usar o protótipo para efetuar testes e utilizar o protocolo think-aloud com membros do público alvo para assegurar que o jogo é uma tradução eficaz do especificado nas metas de aprendizagem. Os desenvolvedores devem fazer alterações na interface do jogo e decisões usando o feedback obtido a partir da observação de jogadores, tendo em mente que a tomada de decisão "mais fácil" poderia prejudicar a aprendizagem. (EAGLE; BARNES, 2012)</p> <p>Simular o jogo e colher informações relevantes para possíveis aprimoramentos. Recomendamos rastrear cada ação relevante (correta ou incorreta) que é tomada pelo jogador e usá-lo para gerar um relatório, entregue ao aluno para autoavaliação no final da simulação do jogo, ou entregue ao instrutor para monitorar o desempenho dos alunos. O relatório inclui comentários detalhados sobre como o estudante usa o procedimento. Esta informação também pode ser usada para identificar potenciais pontos fracos do jogo simulado, corrigir a formalização do procedimento, se necessário e reforçar a aprendizagem sem a necessidade de intervenção do professor (BORRO ; ESCRIBANO et al. 2014)</p> <p>Esta simulação pode ser criada usando uma ferramenta de autoria de jogos como eAdventure ou outra ferramenta. Isso reduz o custo e aumenta a participação de especialistas de conteúdo no processo de desenvolvimento. Além disso, permite e facilita uma prototipagem rápida, o que acelera os processos de verificação de conteúdo, pois, no protótipo as revisões são mais produtivas, ao invés de desenhos ou documentos. Esta abordagem também permite que os designers de jogos testem rapidamente a mecânica do jogo e ideias, além de identificar as principais dificuldades no projeto.. Em iterações intermédias gerar protótipos que estão orientados para a avaliação dos elementos do jogo que estão incluídos. As iterações finais têm versões estáveis, incluindo os recursos de arte finais para avaliar o engajamento com os usuários finais. (BORRO-SCRIBANO et al, 2014)</p>
<p><b>DESENVOLVER E CODIFICAR O JOGO ATÉ A VERSÃO FINAL</b></p> <p>Utilizar uma ou mais ferramentas de criação para prototipagem rápida e implementação do jogo até a versão final. Para a apresentação de implementação, a equipe de desenvolvimento deve aplicar os elementos (modelos 3D, sons, imagens, texturas, animações) ou reutilização de bibliotecas existentes, como Google Warehouse para os modelos 3D. O cenário e as interfaces precisa agora serem desenvolvidos. Para isso, vários motores de jogo e editores podem ser utilizados, tais como, Unreal, Unity3D, Blender e 3D Studio, (ROCHA et al. 2012)</p>
<p><b>PÓS-PRODUÇÃO</b></p>
<p><b>EFETUAR TESTE BETA AVALIANDO A JOGABILIDADE, USABILIDADE E POSSÍVEIS ERROS.</b></p> <p>Uma câmara de vídeo capturando a interação de um par de estudantes com o jogo, outra câmara de vídeo em movimento para capturar as discussões durante a aula, as ações do professor, e a experiência prática realizada. Gravadores de áudio também devem ser usados para gravar interações de estudantes enquanto se envolvem com o jogo. Entradas de texto dos alunos dentro do jogo devem ser automaticamente salvos em arquivos de texto externo ao jogo, e suas ações dentro do jogo devem ser gravadas por uma gravador de tela . (LAZAROU, 2011)</p> <p>Realizar testes de usabilidade com pares de estudantes e colher informações sobre alterações necessárias a serem feitas no protótipo do jogo. Deve-se eliminar problemas de usabilidade e avaliar a utilidade do jogo na realização dos seus objetivos pretendidos. O teste de usabilidade deve usar o protocolo think aloud a fim de que os alunos expressem suas reações e façam recomendações ao interagir com o jogo. Realizar testes também com os professores com o método think aloud como ferramenta de obtenção de dados, principalmente comentários sobre se o jogo pode ter sucesso em fazer que os alunos aprendam o conteúdo. (LAZAROU, 2011)</p>
<p><b>AVALIAR A GARANTIA DE QUALIDADE DO JOGO EM AMBIENTE REAL DE SEU PÚBLICO ALVO.</b></p> <p>Verificar de alguns ou todos os seguintes aspectos:</p> <ul style="list-style-type: none"> <li>Confiabilidade (ou seja, a simulação do jogo é estável e livre de erros de programa);</li> <li>Facilidade de utilização (ou seja, a simulação jogo é agradável e fácil de usar);</li> <li>O valor educativo (isto é, o os alunos podem atingir os objetivos de aprendizagem pretendidos)</li> <li>O envolvimento (ou seja, o jogo é apelativo e envolvente e desafia as habilidades do jogador.</li> <li>A importância de cada aspecto varia dependendo do contexto específico.</li> <li>O conteúdo educacional trabalhado no jogo é válido? (só pode ser feito por especialistas do domínio através de uma avaliação de todos os conceitos trabalhados no jogo) (KEBRITCHI et al. 2010 apud BORRO-ESCRIBANO et al. 2014)</li> </ul>

## APÊNDICE A – METODOLOGIA DE DESENVOLVIMENTO DE UM JOGO SÉRIO EDUCACIONAL: VERSÃO COM APLICAÇÃO DAS MELHORIAS (CONTINUAÇÃO)

### AVALIAR ASPECTOS TÉCNICOS E EDUCACIONAIS DO JOGO

Efetuar as seguintes questões de avaliação:

Questão	Quem responde?	Relacionado a
Existem erros ou comportamentos inesperados no jogo?	Beta-Testers	Confiabilidade
O jogo funciona em todas as plataformas-alvo?	Beta-Testers	Confiabilidade
O jogo simula fielmente o processo e ambiente proposto?	Especialistas de Domínio	Valor educacional
O conteúdo incorporado no jogo (por exemplo, materiais de referência, diálogos, texto) são precisos e adequados para o público-alvo?	Especialistas de Domínio	Valor educacional
Os objetivos do jogo e as regras são claras para o usuário iniciante?	Estudantes, especialistas de domínio, especialistas de Jogos	Jogabilidade, Valor educacional, Experiência de uso
O jogo é atraente para o público-alvo?	Estudantes, , especialistas de Jogos	Jogabilidade, Experiência de uso
O contexto e as condições são claras para o usuário iniciante?	Estudantes, especialistas de domínio, especialistas de Jogos	Jogabilidade, Valor educacional, Experiência de uso
É fácil aprender a usar o jogo?	Beta-Testers, Estudantes, , especialistas de Jogos	Usabilidade
Os objetivos de aprendizagem são contemplados?	Especialistas de Domínio	Valor educacional
Quão bem os alunos alcançam as metas de aprendizagem?	Estudantes	Valor educacional
Será que os alunos aprendem mais alguma coisa (conhecimento incorreto ou correto)	Estudantes	Valor educacional

Avaliação formal é o tipo mais demorado de avaliação e, portanto, apenas uma ou duas experiências no final do processo são recomendadas, orientada para avaliar o ganho educacional em relação ao ensino tradicional. O resultado é um conjunto de propostas de modificação a qualquer um dos os sub-produtos das outras tarefas (análise e projeto documentos ou protótipos). (TORRENTE et al. 2014)

## APÊNDICE B – QUESTIONÁRIO PESQUISA DISCENTE

QUESTIONÁRIO DISCENTE LINGUAGEM C DATA: \_\_\_\_/\_\_\_\_/2016 CURSO(s): ( ) Sistemas de informação ( ) Ciência da computação

OBS: RESPONSA SEM SE IDENTIFICAR! FIQUE À VONTADE PARA SER SINCERO NAS RESPOSTAS! Marque (X) em um ou mais opções das perguntas!

DISCIPLINAS CURSADAS COM LINGUAGEM C: ( ) Algoritmos e programação ( ) Linguagem de programação ( ) Linguagens e técnicas de programação I ( ) Estrutura de dados I Outra(s): \_\_\_\_\_

### 1) Quais foram os ERROS MAIS FREQUENTES que você cometeu ao cursar disciplina(s) com linguagem C?

- 1.1 ( ) Na lógica da programação. De como resolver o problema proposto no exercício
- 1.2 ( ) Esquecer o ponto e vírgula ; no final de comandos, etc.
- 1.3 ( ) Uso incorreto do número de parênteses. Ex: `if ((x==1) && (x==2)) || ((y==0) /* aqui falta parênteses*/`
- 1.4 ( ) Uso inadequado da vírgula. Ex: `printf("numero=",%d,"num)` sendo o correto `printf("numero=%d",num);`
- 1.5 ( ) Esquecer o `&` no `scanf` travando o sistema com tela de erro. Ex: `scanf("%d", num);` ao invés de `scanf("%d", &num);`
- 1.6 ( ) Esquecer de fechar chaves { } de início e fim em algum lugar do código.
- 1.7 ( ) Esquecer de incluir biblioteca com `#include`. Ex: usar função `system("PAUSE");` e esquecer incluir biblioteca `<stdlib.h>`
- 1.8 ( ) Erro no nome de comandos causando erros de compilação. Ex: `printf, swithc, doble,etc`
- 1.9 ( ) Usar índices errados no vetor e fora do limite do mesmo. Ex: `int vetor[10]; vetor[10]=34;` <-só que vai de vetor[0] a [9]
- 1.10 ( ) Erro uso da barra ex: `/n` ao invés de colocar `\n`
- 1.11 ( ) Colocar ponto e vírgula ; onde não deveria. Ex: depois de declarar função `int mensagem( );` depois de `if ( );` etc
- 1.12 ( ) Erro no retorno da função `main( )` e na sua declaração (`int main(void)` ou pode ser `int main(int argc, char *argv[ ] )`
- 1.13 ( ) Usar atribuição = ao invés de == em uma comparação. Ex: `if (num=2)` quando deveria ser `if(num==2)`
- 1.14 ( ) Não declarar variáveis utilizadas no programa.
- 1.15 ( ) Não usar aspas simples ' ' em um caractere. Ex: `char letra = "A";` ao invés do uso correto `'A'`
- 1.16 ( ) Tentar colocar palavras em um char que só cabe um caractere. Ex: `char c = "teste";` o correto seria `char c[ ] = "teste";`
- 1.17 ( ) Separar o comando `for` por vírgula `for(i=0,i<5,i++)` ao invés do correto com ponto e vírgula; `for(i=0;i<5;i++)`
- 1.18 ( ) Declarar variáveis e não inicializá-las ex: `int contador;` ao invés de `int contador=0;`
- 1.19 ( ) Esquecer o `break` no final de cada comando case do `switch ( int x = 2; switch(x) { case 2: printf("dois\n"); }`
- 1.20 ( ) Usar o especificador incorreto (`%d, %f, %c, %lf, %s, %2f`)
- 1.21 ( ) Usar de forma errada o caractere nulo `'\0'` sendo o correto é `'\0'`
- 1.22 ( ) Erro na divisão / (inteira e real) `double meio = 1/2;` <-resultado zero. Ao invés `double meio = 1.0/2;` <-resultado 0,5

Outro(s): \_\_\_\_\_

### 2) Quais os comandos e/ou conceitos da linguagem C que você teve MAIOR DIFICULDADE de aprendizado?

- |  |  |
|--|--|
| 2.1 ( ) Comando if-else (se então)   | 2.14 ( ) Loops (repetições) while, for   |
| 2.2 ( ) Estruturas (Struct)  | 2.15 ( ) Strings - char nome[80];  |
| 2.3 ( ) <code>scanf("%d", &amp;num)</code>   | 2.16 ( ) inclusão bibliotecas <code>#include &lt;string.h&gt;</code>               |
| 2.4 ( ) Declaração de variáveis  | 2.17 ( ) Operador <code>% MOD</code> (resto da divisão inteira)                    |
| 2.5 ( ) Uso de parênteses. ex: <code>if ((achou)==1)</code>  | 2.18 ( ) Manipulação arquivos texto e binário                                      |
| 2.6 ( ) Uso da vírgula - ex: <code>printf("%d,%d",num1, num2)</code>   | 2.19 ( ) Uso aspas simples e duplas ' ' " "  |
| 2.7 ( ) Uso de variável char com tabela ASCII  | 2.20 ( ) funções <code>string.h</code> <code>strcpy, strcat, strlen, strcmp</code> |
| 2.8 ( ) Operadores lógicos: <code>&amp;&amp;   </code> ex: <code>if (letra &gt;= 65 &amp;&amp; letra &lt;= 90)   </code> | 2.21 ( ) Variáveis Locais e Globais  |
| 2.9 ( ) Break e Continue   | 2.22 ( ) <code>fflush(stdin)</code> - limpeza de buffer teclado                    |
| 2.10 ( ) Especificadores <code>%d %f %c %lf %s %2f</code>  | 2.23 ( ) Comando <code>switch-case</code> (escolha caso)                           |
| 2.11 ( ) <code>typedef</code>  | 2.24 ( ) Vetores e Matrizes ex: <code>int vetor[10];</code>                        |
| 2.12 ( ) Pré e pós incremento <code>++i --i ++i --i</code>   | 2.25 ( ) <code>printf("%d %c %.2f", num, letra, media)</code>                      |

Outro(s): \_\_\_\_\_

### 3) Que elementos ATRAPALHARAM no seu aprendizado da disciplina com linguagem C?

- 3.1 ( ) A Linguagem C é complexa
- 3.2 ( ) Falta de prática em sala de aula
- 3.3 ( ) Falta de prática em casa
- 3.4 ( ) Falta de tempo para programar
- 3.5 ( ) Falta de contato diário com a linguagem C
- 3.6 ( ) Falta de resolução frequente de exercícios
- 3.7 ( ) Baixa execução de programas e de análise dos erros de compilação
- 3.8 ( ) Não consegui entender como o computador executa um programa na CPU e o manipula a memória
- 3.9 ( ) Dificuldades no uso das IDE's - ambiente de desenvolvimento (DEV++ , CODE BLOCKS, Visual Studio)
- 3.10 ( ) Falta de aulas utilizando o compilador
- 3.11 ( ) Falta de material para iniciantes totais na linguagem C
- 3.12 ( ) Didática do professor insuficiente
- 3.13 ( ) Dificuldades em entender erros de compilação em inglês e seus significados
- 3.14 ( ) Falta de estudo aprofundado de cada conceito da linguagem C

Outro(s): \_\_\_\_\_

## PARTE 2 JOGOS

### 4) Se existisse um jogo sobre linguagem C que elementos considera importante que o jogo possua?

- |  |   |
|--|---|
| 4.1 ( ) Que tenha boa jogabilidade                   | 4.6 ( ) Que possua Gráfico 3D e boa interface gráfica |
| 4.2 ( ) Que seja de fácil compreensão                | 4.7 ( ) Que tenha bom cenário                         |
| 4.3 ( ) Que ensine comandos da linguagem             | 4.8 ( ) Que seja bem simples sem muito design         |
| 4.4 ( ) Que revele os erros cometidos na linguagem C | 4.9 ( ) Que possua enigmas                            |
| 4.5 ( ) Que tenha boa relação entre usuário e o jogo | 4.10 ( ) Que não seja chato                           |

### 5) Que tipo de jogo mais gosta de jogar?

- 5.1 ( ) Ação e aventura 5.3 ( ) Tiro em primeira/terceira pessoa (FPS) 5.5 ( ) Multiusuário 5.7 ( ) Futebol
- 5.2 ( ) RPG 5.4 ( ) Esportes 5.6 ( ) Estratégia 5.8 ( ) Luta

OBS: NAS QUESTÕES 6,7,8 ABAIXO MARQUE COM UM X APENAS UMA DAS ALTERNATIVAS

### 6) Você gosta de jogos do tipo FPS (tiro em primeira pessoa)?

- 6.1 ( ) Gosto 6.2 ( ) Não gosto

### 7) Você jogaria um jogo FPS (tiro em primeira pessoa) sobre linguagem C?

- 7.1 ( ) Sim 7.2 ( ) Sim se não tiver sangue ou muita violência 7.3 ( ) Não

### 8) Prefere jogos em:

- 8.1 ( ) 3D 8.2 ( ) 2D 8.3 ( ) AMBOS

Outras considerações que achar importante: (OPCIONAL) \_\_\_\_\_

## APÊNDICE C – PÓS-TESTE APLICADO APÓS O ALUNO JOGAR O JOGO

Nas questões de 1 a 10 assinale de caneta azul ou preta uma única alternativa (letra) que considerar correta

**Questão 01) Qual o valor das variáveis v, x, y e z após a execução do seguinte trecho de código**

<pre>int v = 0, x = 1, y = 2, z = 3; v += x+y; x *= y = z + 1; z %= v + v + v; v += x += y += 2;</pre>	V = _____ X = _____ Y = _____ Z = _____
--	--

**Questão 02) Diga o resultado das variáveis x, y e z depois da seguinte sequência de operações:**

<pre>int x,y; int a = 14, b = 3; float z; x = a/b; y = a%b; z = y/x;</pre>	a. x = 4.66666, y = 2, z = 0.4286 b. x = 5, y = 2, z = 0.4 c. x = 5, y = 2, z = 0. d. x = 4, y = 2, z = 0.5 e. x = 4, y = 2, z = 0. f. Nenhuma das opções anteriores
--	---

**Questão 03) Diga o resultado das variáveis x, y e z depois da seguinte sequência de operações:**

<pre>int x,y,z; x=y=10; z=++x; x=-x; y++; x=x+y-(z--);</pre>	a. x = 11, y = 11, z = 11    b. x = -11, y = 11, z = 10 c. x = -10, y = 11, z = 10    d. x = -10, y = 10, z = 10 e. Nenhuma das opções anteriores
--	---

**Questão 04) Qual o valor de x após a seguinte sequência de comandos:**

<pre>a = 10; b = 20; x = 0; x = (b &gt; a) ? b : a;</pre>	a)0    b)2    c)10    d)20    e)40
---	------------------------------------

**Questão 05) Considere o código a seguir**

<pre>#include&lt;stdio.h&gt; int main() { int x; for(x=-1; x&lt;=10; x++) { if(x &lt; 5) continue; else break; printf("UNLOCK-C"); } return 0; }</pre>	Quantas vezes UNLOCK-C é impresso na tela? A. infinitas vezes B. 11 vezes C. nenhuma vez D. 10 vezes E. Uma vez F. 9 vezes G. 8 vezes
--	--

**Questão 06) Analise o código a seguir:**

<pre>#include &lt;stdio.h&gt; int main () { printf ("Hello World\n"); }</pre>	O código ao lado: a) Não compila devido a falta de ponto e vírgula em int main(); b) Compila, mas ocorre um erro de execução c) Compila normalmente e mostra na tela ("Hello World\n"); d) Compila normalmente e mostra na tela Hello World e) Compilaria se existisse o comando return 0; antes da chave } final
---	--

**Questão 07) Analise o código a seguir:**

<pre>#include &lt;stdio.h&gt; int main(){ int i=0; for(i=0;i&lt;=50;i++) printf("%d",++i); }</pre>	O código ao lado: a) exibirá na tela os números de 0 até 50 b) exibirá na tela os números de 1 até 50 c) exibirá na tela os números de 0 até 49 d) exibirá na tela os números de 1 até 49
--	---

**Questão 08) Analise o código a seguir:**

<pre>#include &lt;stdio.h&gt; int main(){ int continua=1, cont=0; while(continua) printf("%d", cont++); } continua=0;//falso</pre>	O código ao lado a) Irá executar até cont =100 b) irá entrar em loop infinito c) exibirá tela o valor de cont de 1 até 100; d) irá executar até continua se tornar zero.
--	--

**Questão 09) Analise o código a seguir:**

<pre>#include &lt;stdio.h&gt; int main(){ int x=0; int y=0; x=++y + ++x; printf("%d",++x); }</pre>	O código ao lado vai exibir na tela o valor: a) 2 b) 1 c) 0 d) 3
--	--

**Questão 10) Analise o código a seguir:**

<pre>#include &lt;stdio.h&gt; int main(){ int i=0; for(i=0;i&lt;=50;i++) if(i%2 ==0) printf("%d", i); }</pre>	O código ao lado a) imprimir na tela os números ímpares entre 0 a 50 b) imprimir na tela os números pares entre 0 a 50 c) imprimir na tela os números ímpares entre 1 a 50 d) imprimir na tela os números pares entre 1 a 50
---	--

## APÊNDICE D – DOCUMENTO COM VOCABULÁRIO COMPARTILHADO

<p>           Ambiente            Arquivo            Artefato            Atividade            Bot            Cenário            Codificar            Código            Compilador            Conceito do jogo            CPU            Desenvolvedor            Design            Documento de design            Documento de requisitos            Escopo            Especialista de domínio            Feedback            FPS            Gameplay            Golden Nuggets            IHC(interface homem-computador)            Interface            Implementação            Inteligência artificial            Jogabilidade            Jogo sério            Heurística         </p>	<p>           High Concept            Linguagem C            Linguagem de programação            Log            Lúdico            Kismet            Memória            Metodologia            Mock-up            Motor de jogo            Papel            Plataforma            Protótipo            Prototipagem            Requisito            Script            Semântica            Sintaxe            Storyboard            Tarefas            Testes            Think-Aloud            UDK            Unreal Engine            Usabilidade            Versão final         </p>
--	---

## APÊNDICE E – DOCUMENTO DE REQUISITOS

<b>DOCUMENTO DE REQUISITOS</b>	
<p>Este documento contém a especificação de requisitos para o jogo digital: UNLOCK-C, que será utilizado como sistema de apoio ao ensino e aprendizado da linguagem de programação C para alunos de cursos de computação no ensino superior</p>	
<p><b>OBJETIVOS</b> Este projeto tem como objetivo, desenvolver um jogo que objetiva auxiliar no processo de ensino e aprendizagem de disciplinas de graduação que utilizam a linguagem de programação C como suporte. O jogador experimentará fases que irão abordar conceitos acerca da sintaxe, semântica e lógica de programação utilizando a linguagem C.</p>	
<p><b>CONVENÇÕES, TERMOS E ABREVIações</b></p>	
Termo	Significado
RNF	Requisitos Não Funcionais: são os requisitos relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenibilidade e tecnologias envolvidas.
RF	São os requisitos relacionados às funcionalidades do sistema.
<p><b>ESCOPO</b> O jogo digital UNLOCK-C é um jogo de ação e aventura para plataforma PC e destinado a alunos universitários dos cursos de computação. O jogo será destinado à plataforma PC, com necessidade de instalação prévia, e jogado por um único jogador sem necessidade de conexão a internet. UNLOCK-C é uma ferramenta para apoiar e contribuir no processo de ensino e aprendizagem dos conceitos envolvidos com a linguagem de programação C. O jogo deve fornecer uma interface 3D aprimorada, intuitiva, prazerosa e divertida que irá envolver o jogador com desafios relacionados com programação. Utilizando o jogo, o professor poderá permitir ao seu aluno explorar elementos relacionados com a linguagem C, principalmente os erros frequentes de sintaxe e semântica. Ao jogar o jogo o aluno poderá percorrer um ambiente 3D de jogo estilo tiro em primeira pessoa(FPS) e decidir os caminhos que deseja seguir de acordo com desafios de programação apresentados a ele.</p>	
<p><b>DESCRIÇÃO GERAL</b> As principais funções associadas a este produto serão descritas nesta seção</p>	
<p><b>REQUISITOS FUNCIONAIS</b> RF1. O sistema deve possibilitar que o jogador pause o jogo pressionando a tecla ESC do teclado RF2. O sistema deve possibilitar que o jogador reinicie o jogo clicando com o botão esquerdo do mouse na opção de tela resume RF3. O sistema deve possibilitar que o jogador saia o jogo clicando com o botão esquerdo do mouse na opção de tela exit RF4. O sistema deve possibilitar que o jogador mude do modo em primeira pessoa para o modo em terceira pessoa RF4. O sistema deve permitir ao jogador obter feedbacks sobre RF5. O sistema deve fornecer uma forma de registrar erros que o jogador cometeu durante o jogo RF6. O sistema deve gerar inimigos bots que atacam o jogador em momentos determinados RF7. O sistema deve emitir uma mensagens de feedback em texto e vídeo para o jogador RF8. O sistema deve disponibilizar ao jogador possibilidade de recarregar munições de suas armas RF9. O sistema deve disponibilizar ao jogador possibilidade de alternar entre armas disponíveis RF10. Sistema a deve ter mostrar um HUD para o jogador com informações na tela da sua saúde e quantidade de munição da arma atual. RF11. O sistema deve mostrar na tela mensagem em texto indicando que o jogador foi abatido por um inimigo indicando seus respectivos nomes RF12. O sistema deve fornecer ao jogador a possibilidade de recarregar suas armas através de itens coletados durante o jogo RF13. O sistema deve fornecer ao jogador a possibilidade de aumentar sua saúde através de itens coletados durante o jogo. RF14. O sistema deve gerenciar o numero de vezes que o jogador morreu. RF15. O sistema deve mostrar ao final do jogo informações sobre o mesmo.</p>	
<p><b>REQUISITOS NÃO FUNCIONAIS</b> RNF1. O sistema deverá ser executado em ambiente Plataforma PC com no mínimo sistema operacional Windows XP SP3 2.0 GHz velocidade processador, 2 GB de memória RAM e placa de vídeo com suporte modelo de sombras 3.0 RNF2. O sistema deverá ser de fácil utilização com controles de movimentação e interação intuitivos RNF3 O sistema deve apresentar adequação de conteúdo, clareza no conteúdo, aprendizagem com encadeamento de ideias e relação entre a teoria e a prática. RNF4. O sistema deve possuir interatividade, presença de elementos gráficos e atratividade. RNF5. O sistema deve possuir interface gráfica 3D RNF6. O sistema deve ser desenvolvido na categoria tiro em primeira pessoa(FPS)</p>	

## APÊNDICE F – DOCUMENTO DE DESIGN E CONCEITO DO JOGO

<b>CONCEITO DO JOGO</b>	
<p><b>Nome do Jogo:</b> UNLOCK-C</p> <p><b>Plataforma:</b> PC/Windows</p> <p><b>Jogadores:</b> Single Player;</p> <p><b>Gênero:</b> FPS;</p> <p><b>High Concept:</b> O jogo consiste em vencer desafios relacionados com programação utilizando a linguagem C. O jogador terá que analisar códigos e escolher o caminho correto para passar ao próximo desafio e ambiente. O jogador terá que enfrentar ao longo de sua jornada inimigos robôs(bots), posicionados em diversas partes do cenário, bombas que explodem no caminho, locais fatais, torres que atiram, nave inimiga que atira e demais itens.</p> <p><b>Objetivo:</b> Para vencer o jogo o jogador deve ultrapassar todos os desafios, conseguir abrir todas as portas e alcançar uma torre com uma fonte de energia que alimenta os robôs inimigos e destruí-la.</p>	
<p><b>Recursos:</b> O jogador poderá atirar através de três tipos de armas, recarregar armas, obter mais saúde, sofrer danos de tiros, sofrer danos de choque, explodir barris com tiros, movimentar barris com tiros, modificar o modo do jogo de primeira pessoa para terceira pessoa. O jogo irá evoluir de um ponto inicial de uma fábrica de produtos químicos abandonada e o jogador vai passar por várias salas e prédios desta fábrica até o final do jogo. Diversos sons estarão presentes no jogo, desde som ambiente, sons de vento, sons de tiros, sons de armas, de passos do personagem e inimigos. O jogo contará com simulação da luz do sol e iluminação com lâmpadas comuns e direcionais.</p>	
<b>DOCUMENTO DE DESIGN – UNLOCK-C</b>	
<p><b>INTRODUÇÃO</b> Este documento tem como objetivo apresentar uma concepção descritiva do design (documento de design) do jogo UNLOCK-C.</p>	
<p><b>VISÃO GERAL ESSENCIAL</b> <b>Resumo</b> UNLOCK-C colocará o jogador no controle de um personagem inserido dentro de uma fábrica abandonada de produtos químicos tomada por robôs inimigos. O jogador passará POR diversas salas e prédios da fábrica tentando vencer desafios sobre a linguagem de programação C e combatendo os robôs inimigos até conseguir chegar a fonte de energia que alimenta os robôs. Assim o Unlock-C é um jogo 3D FPS com interface gráfica marcante, com diversos desafios relacionados com a linguagem de programação C, e com ações e aventuras do gênero de tiro em primeira pessoa.</p>	
<p><b>Aspectos fundamentais</b> O jogador poderá controlar e mover pelo cenário um guarda armado com armadura e utilizar armas para atirar nos robôs inimigos. Ao longo do jogo vários robôs inimigos serão gerados para atacar o jogador. O jogador só poderá avançar no jogo se decodificar os desafios sobre linguagem de programação C e conseguir sobreviver aos ataques, abatendo os robôs inimigos.</p>	
<p><b>Golden Nuggets</b> UNLOCK-C se passa "dentro" de um fábrica abandonada com cenário e interface gráfica aprimorada e com diversos itens que irão provocar uma excelente imersão do jogador ao clima do jogo. O jogador terá ao longo do jogo muita ação e aventura garantidos pelos diversos desafios de um jogo FPS. Existirão também diversas animações, cinemática, elementos atmosféricos(névoa, vento, fogo) que irão fornecer um clima de tensão AO jogo. Além disso, o jogador terá a oportunidade de aprender mais sobre a linguagem de programação C de forma divertida e atraente.</p>	
<p><b>CONTEXTO DO JOGO</b> <b>História do Game</b> O personagem C-GUARD foi "plantado" como agente em uma fábrica de produtos químicos abandonada. A fábrica foi dominada por robôs inimigos que são alimentados por uma torre de energia que fica no teto em um dos prédios da fábrica. Ele precisa destruir esta fonte de energia pois, os robôs ameaçam se multiplicar e invadir outros territórios. C-GUARD foi enviado para esta missão e para ter êxito terá que enfrentar robôs inimigos armados e também analisar códigos em linguagem de programação C para conseguir avançar no jogo. Depois de ter abatido os robôs inimigos, desviado de bombas, sobrevivido a barris que explodem, a nave inimiga, a locais fatais, a choques, a luta com os robôs, C-Guard terá que chegar até a torre de energia e destruí-la.</p>	

## APÊNDICE F – DOCUMENTO DE DESIGN E CONCEITO DO JOGO (CONTINUAÇÃO)

### Eventos anteriores

C-Guard é um especialista em linguagem de programação C e robôs e por isso, ele foi o escolhido para ser enviado a fábrica. Além disso, o local da fábrica fica próximo a uma comunidade onde vive a mãe de C-Guard e sua família. Caso ele falhe a vida de seus familiares correrá risco, pois caso os robôs organizem um exército irão sair da fábrica e eliminar todos no caminho.

### Principais Jogadores

**C-Guard** - É o personagem principal e seu objetivo é entrar na fábrica, abater os robôs e destruir a fonte de energia que os alimenta. Ele terá a sua disposição três tipos de armas e munição para enfrentar os desafios. O personagem terá que possuir e também adquirir ao longo de sua trajetória no jogo conhecimentos sobre linguagem C, para ultrapassar os desafios sobre a linguagem colocados no jogo.

**BOT Inimigo** - O robô inimigo é um agente movido a inteligência artificial programado para abater qualquer intruso. Os robôs terão armas diferentes e serão mais fortes ou mais fracos de acordo com as armas e inteligência artificial programada.

### OBJETOS ESSENCIAIS DO GAME

#### Personagens

##### C-GUARD

É o atirador que utiliza armas e é o alívio do jogador. Este jogador possui saúde que pode ser restabelecida com recarregadores de saúde coletados no chão. O C-Guard pode alternar entre três tipos de armas e recarregá-las coletando munição espalhada pelo cenário. Ele pode sofrer danos com os tiros dos robôs inimigos, com bombas que explodem em alguns locais de passagem, com choques se cair em um curto circuito, se entrar em locais errados e fatais.

##### Robô(Bot) inimigo

O bot inimigo é um agente de inteligência artificial que possui armas, atira e persegue o personagem principal C-GUARD. Ele pode sofrer danos e morrer ao ser atingido pelos tiros de C-GUARD.

#### Armas

##### LinkGun

Arma padrão dos personagens e alguns robôs. Tem poder de fogo limitado, mas efetivo. Dispara tiros e também pode disparar um tiro longo em forma de raio que consome muita munição, mas causa mais danos ao alvo. Possui boa quantidade de munição e pode ser recarregada através de coleta de munição.

##### ShockRifle

Arma mais poderosa que a linkgun e gera lasers azuis com alto poder de dano ao bot inimigo ou ao jogador. Pode disparar também um laser contínuo com maior poder de fogo.

##### Rocket Launcher

Arma mais potente do jogo que dispara mísseis que explodem ao colidirem com o alvo. Possui munição limitada e pode ser recarregada através de coleta de munição.

##### Nave Inimiga

Nave que dispara tiros via dois canhões presentes nas asas do jogador. A nave se locomove após um robô entrar e assumir o controle. A nave recebe danos de tiros que devem ser em grande volume para conseguir destruí-la

##### Veículo Scorpion

Veículo que pode ser utilizado pelo jogador e que atira dois tiros longos via canhão localizado em seu teto.

##### Torres que atiram

É uma arma imóvel que atira contra o jogador e só pode ser destruída se receber diversos tiros do personagem C-GUARD.

#### Bombas

As bombas irão detonar após o jogador ter chegado em determinado local e ficarão explodindo aleatoriamente causando dano ao jogador.

#### Objetos

##### Base de captura de armas

Imagem: Base luminosa com arma em rotação suspensa por uma luz

Onde pode ser encontrado: Em diversas partes do cenário

Comportamento: Quando o jogador passa pela base recebe a arma correspondente e sua munição



## APÊNDICE F – DOCUMENTO DE DESIGN E CONCEITO DO JOGO (CONTINUAÇÃO)

### **Base de captura de armaduras**

Imagem: Base com armadura um pouco acima, em rotação e suspensa pelo ar.

Onde pode ser encontrado: Em diversas partes do cenário

Comportamento: Quando o jogador captura a armadura fica com mais poder e sua saúde será menos atingida quando for atingido por tiros ou qualquer elemento que reduza sua vida.

### **Recarregador de Munição**

Imagem: Objeto luminoso em formato diferenciado para cada um dos três tipos de armas

Onde pode ser encontrado: Em diversas partes do cenário

Comportamento: Quando o jogador colidir a arma correspondente terá sua munição recarregada

### **Recarregador de Saúde**

Imagem: Base de metal que sustenta uma esfera iluminada de diversos tamanhos a depender do ganho de saúde proporcionado

Onde pode ser encontrado: Em diversas partes do jogo.

Comportamento: Quando o jogador colidir sua saúde será revigorada em X pontos a depender do tamanho do recarregador de saúde.

### **Barril**

Imagem: Barril com sinais de corrosão

Onde pode ser encontrado: Em algumas partes do cenário, principalmente nas primeiras salas.

Comportamento: Explode se atingido por tiros do jogador ou robôs inimigos. Caso o jogador esteja próximo do barril ele irá ser abatido.

### **Teleporter**

Imagem: Base que sustenta uma elipse com áurea animada que mostra no centro o destino de ida

Onde pode ser encontrado: Em poucas partes do cenário.

Comportamento: Quando o jogador colidir será teletransportado para o destino estabelecido para o teleporter

### **Portas automáticas**

Imagem: Duas portas de ferro com corrosão

Onde pode ser encontrado: Nas saídas das salas da fábrica

Comportamento: Quando o jogador se aproximar elas abrem de forma animada e com som de abertura. Quando o jogador se afasta elas fecham de forma animada e com som de fechamento.

### **Jump pad(Impulsionadores)**

Imagem: Base luminosa com feixes de luz quadrados em suspensão

Onde pode ser encontrado: em poucos prédios do cenário

Comportamento: Quando o jogador fica em cima ele é impulsionado com um salto e vai parar no ponto de destino especificado para o jump pad.

### **Triggers(Gatilhos)**

Imagem: é um elemento invisível

Onde pode ser encontrado: Na maior parte de todos os ambientes do cenário

Comportamento: Dispara eventos indicando a programação do jogo que o jogador tocou ou passou por aquele local. Diversos eventos diferentes são acionados por diversos tipos de triggers.

### **Checkpoint (ponto de checagem)**

Imagem: é um elemento invisível

Onde Pode ser encontrado: Em alguns locais do jogo.

Comportamento: é um elemento invisível criado via gatilho(trigger) que estabelece um ponto de checagem. Caso o jogador seja abatido o jogo irá recomeçar deste ponto.

### **Base com mini fonte de energia**

Imagem: Base elíptica com partícula de luz animada no centro e suspensa pelo ar

Onde pode ser encontrado: No subsolo da primeira sala do jogo

Comportamento: Ao ser capturada e tendo o jogador eliminado 3 inimigos, uma porta do primeiro salão principal será aberta.

### **Fonte de energia dos bots inimigos**

Imagem: Uma esfera vazada formada por anéis superpostos e, com efeito, animado luminoso no centro indicando energia.

Onde pode ser encontrado: No teto do prédio que é o final do jogo

Comportamento: Quando o jogador atira é destruída e uma animação de robôs inimigos gigantes explodindo e caindo para trás é apresentada. Um texto indicando que o jogador venceu o jogo aparece e começa um passeio virtual por todo o cenário do jogo.

## APÊNDICE F – DOCUMENTO DE DESIGN E CONCEITO DO JOGO (CONTINUAÇÃO)

### CONFLITOS E SOLUÇÕES

Existem alguns sistemas de conflitos no jogo.

#### Portas que se abrem e explodem o jogador

Existem diversas portas que se o jogador ultrapassar ele será explodido. Para que isso não ocorra ele deverá escolher a porta livre de erros de programação e com o conceito correto sobre a linguagem C.

#### Barril que explode

Ao longo do caminho existirão barris fechando a passagem do jogador. Se o jogador atirar o barril explode. Caso ele esteja próximo ele irá morrer.

#### Barril com física

Este barril não explode e será movido com a força do impacto de tiros. Em alguns momentos códigos indicando atire aqui estarão presos a barris que explodem e de física. O jogador só sobreviverá se atingir o código correto anexado ao barril de física. Caso contrário toda a área ao seu redor será atingida pela explosão dos barris com códigos incorretos.

#### Nave inimiga

Esta nave irá atacar o jogador com tiros em dois canhões. Para o jogador vencer terá que conseguir atirar em uma quantidade alta. Ele poderá coletar armaduras e munição próxima a nave para aumentar suas chances de sucesso.

#### Local Fatal

Local que existe uma killzone(zona de morte). Caso o jogador faça uma escolha errada, ele será imediatamente abatido e o jogo recomeça a partir do último checkpoint.

#### Gerador de energia em curto circuito

Um gerador de energia que se o jogador cair em cima dele será eletrocutado e morrerá instantaneamente. Ele precisa garantir que não seja empurrado em buracos que possuem o gerador no centro. Ele poderá ser empurrado por disparos feitos por bots inimigos ou ao escorregar e cair durante o jogo.

#### Porta que abre em condições especiais

Esta porta só irá abrir se o jogador conseguir abater 3 inimigos e depois capturar uma mini fonte de energia no chão.

#### Queda de local muito alto

Se o jogador cair de local muito alto ele irá morrer e deve evitar isso.

#### Bots inimigos

Vários robôs inimigos serão espalhados pelo cenário com diversas armas. O jogador terá que atirar neles até reduzir a saúde do robô a zero. O jogador deve evitar ser atingido pelos tiros dos bots pois, se sua saúde chegar a zero ele é eliminado.

#### Torres que atiram

Torres que atiram e tiram saúde do jogador. Ele deverá conseguir chegar até elas após passar por campo minado por bombas e atirar nas torres para destruí-las.

### INTELIGÊNCIA ARTIFICIAL

UNLOCK-C terá inteligência artificial aprimorada. Os principais comportamentos são o seguinte:

#### Bots inimigos

Os robôs serão gerados em locais de desova estabelecidos e irão percorrer caminhos criados e estabelecidos para os mesmos. Os bots irão perseguir o jogador de forma agressiva e até mesmo pular obstáculo para conseguir atirar no jogador

#### Nave inimiga

A nave inimiga será acionada após o jogador ativar um trigger que fará com que um bot entre na nave e comece a atirar no jogador. A nave irá se movimentar e será agressiva em relação à quantidade de tiros.

### FLUXO DO JOGO

#### Cinemática de apresentação

O jogo começa com uma cena que irá mostrar o jogador surgindo na fábrica, explosões e um passeio dramático pelo cenário.

## APÊNDICE F – DOCUMENTO DE DESIGN E CONCEITO DO JOGO (CONTINUAÇÃO)

### Nível salão principal

O jogador irá começar a partir desta sala e terá pela frente portas a serem abertas, robôs a serem destruídos e desafios a serem vencidos. As portas com códigos C serão abertas pelo jogador que irá explodir caso abra a porta errada. Para sair do salão principal o jogador deverá conseguir passar pelas portas e vencer 2 inimigos no primeiro andar e 3 inimigos que estão no subsolo.

Neste salão o jogador também terá a interação com 2 inimigos que estão no fosso de água um de cada lado.

### Nível Sala de máquinas

A sala de máquinas existe um robô inimigo com arma a laser de difícil atingimento. O jogador deverá abater este inimigo e poderá contar com munição e arma extra escondida atrás de cilindros grandes de produtos químicos.

### Nível sala de resfriamento.

Caso o jogador opte por entrar na sala de máquinas e pegar o elevador ele vai até esta sala de resfriamento. Lá existirão três inimigos e uma saída para uma ponte que fica acima da sala de máquinas e leva a sala cilindro. Uma animação dramática será acionada com um inimigo aparecendo e pulando em frente ao jogador perto da porta de saída.

### Nível ponte suspensa 1

Neste nível as três torres começarão a atirar no jogador que terá que correr até as torres do outro lado da ponte e destruí-las. Para isso poderá contar com bases de saúde e armadura ao longo da ponte, para que aumente as suas chances de sucesso. Quando o jogador conseguir destruir as torres uma porta se abrirá para o campo aberto e esta abertura será mostrada via cinemática com passeio da câmera pelo cenário da sala cilindro. Aqui teremos um checkpoint.

### Nível ponte suspensa 2

Neste nível o jogador irá enfrentar campo minado em que bombas começarão a explodir aleatoriamente.

Ele terá que ultrapassar correndo esta ponte até chegar ao outro lado.

### Nível sala cilindro.

Nesta sala o jogador terá que enfrentar três inimigos ao mesmo tempo e terá que derrotá-los. Ele contará com base de saúde e armaduras para tentar fazer isso.

### Nível campo aberto

Ao entrar neste nível o jogador irá estabelecer um checkpoint. Neste ambiente irá aparecer a sua frente uma grande quantidade de itens de armadura, saúde e armas que podem ser coletadas. Assim que entrar o jogador verá uma animação de um bot inimigo entrando em uma nave robô inimiga que começará a atirar nele. Ele deverá atirar muitas vezes até conseguir destruir a nave.

Neste mesmo campo aberto após a nave existem 5 inimigos guardando a porta do prédio da fonte de energia dos bots. O jogador poderá entrar em um veículo jipe-tanque atirando nos inimigos robôs. O jogador deve evitar ser atingido por tiros, pois se o veículo explodir com ele dentro ele morre.

### Nível prédio da fonte de energia dos bots: térreo e andares superiores

Este prédio possui uma porta que o jogador precisa adivinhar que é necessário atirar nela para destruí-la. Assim que entra existe outro checkpoint. Uma rampa e um passadiço envolvem um gerador de energia que está em curto circuito e um robô inimigo patrulha o local. O jogador deverá eliminar o robô e evitar tocar no gerador em curto, pois senão ele morre.

Depois da batalha com o robô o jogador irá saltar para o andar acima através de um impulsor (jump pad). Lá existirá um robô inimigo que ele deve eliminar e também evitar cair no vão central para não ser atingido pelo gerador em curto embaixo e morrer eletrocutado.

Após eliminar o robô o jogador irá novamente utilizar um jump pad e para subir ao próximo andar.

Lá existirá uma rampa que dá acesso a cobertura decorada com flamulas suspensas em mastros. Esta rampa será protegida por outro bot e o jogador deve eliminá-lo para conseguir chegar a cobertura.

### Nível prédio da fonte de energia dos bots: cobertura com flamulas

Neste nível três bots inimigos aparecem atirando imediatamente após a chegada do jogador. O jogador terá que eliminar estes três inimigos para que uma porta que dá acesso a um elevador seja aberta. Este elevador dá acesso ao teto onde fica a fonte principal de energia que alimenta os robôs inimigos (bots).

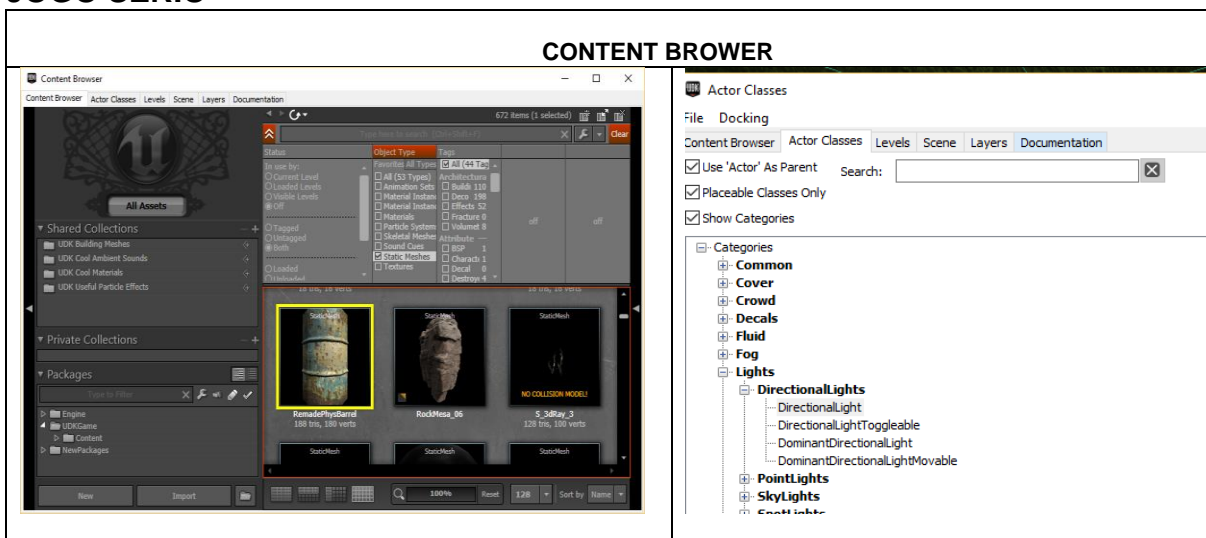
### Nível prédio da fonte de energia dos bots: Teto da fonte de energia

Este é o local final do jogo. O jogador receberá um feedback escrito na tela indicando que precisará atirar na fonte de energia. Após atirar o jogo acaba e ele vence. Uma cinemática com cena que irá passear por todos os cenários anteriores será mostrada e um texto indicando que o jogador venceu.

### CONTROLES

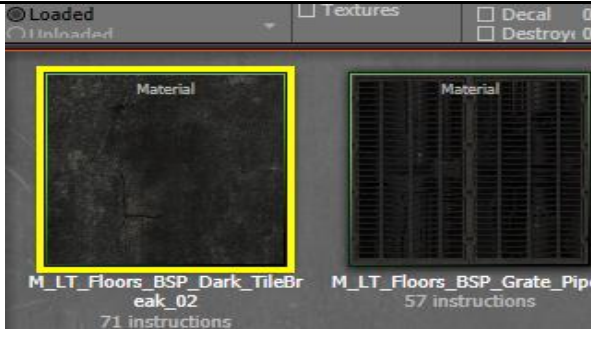
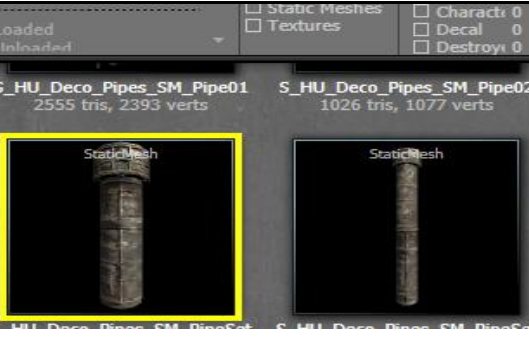
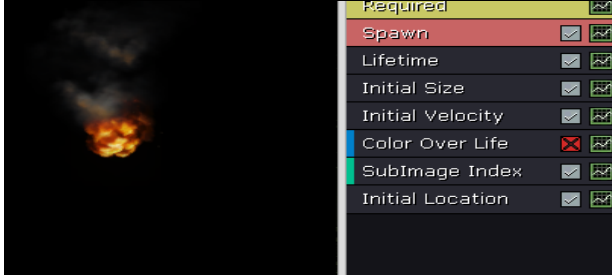

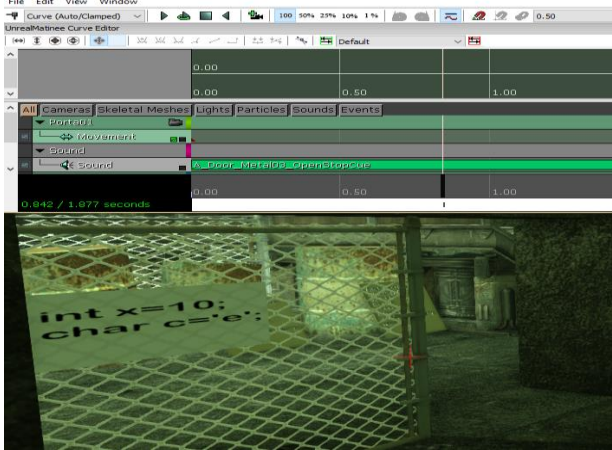
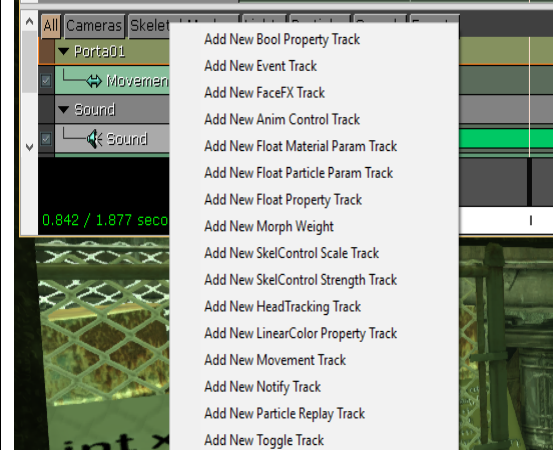
O jogador irá movimentar o personagem C-GUARD através do teclado com as clássicas teclas WASD de movimentação de jogos FPS. Ele irá atirar com as armas através dos botões esquerdo (tiro normal) e direito do mouse (tiro longo potente). Além disso, também poderá usar o mouse para posicionar sua visão através de rotações e translações. Em adição o jogador poderá teclar a tecla tab e digitar BehindView para alternar para o modo de visão em terceira pessoa se preferir.

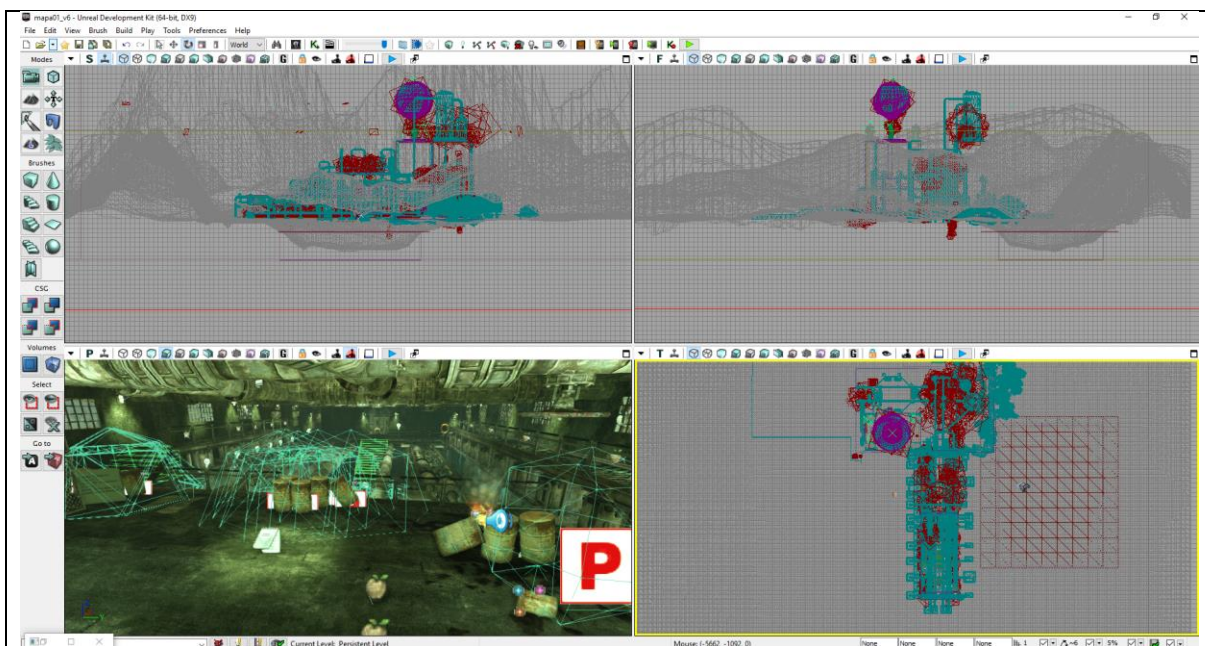
## APÊNDICE G – ITENS PRONTOS DO MOTOR DE JOGO UDK UTILIZADOS NO JOGO SÉRIO



O navegador de conteúdo é utilizado para criar, importar, organizar, visualizar e modificar ativos de conteúdo dentro da Unreal Editor. Diversos elementos prontos de design e programação estão a disposição dos desenvolvedores que podem arrastá-los para o cenário ou inserir através de menus de contexto. Dentre esses elementos destacam-se alguns que foram frequentemente inseridos no jogo UNLOCK-C, tais como: sons, personagens, materiais, texturas, sistema de partículas, etc. Além disso, é possível inserir classes de elementos pré-programados na linguagem da UDK denominada unrealscript, como luzes, veículos, vento, névoa, triggers, partículas de fogo, água, etc. A seguir são indicadas algumas destas estruturas utilizadas.

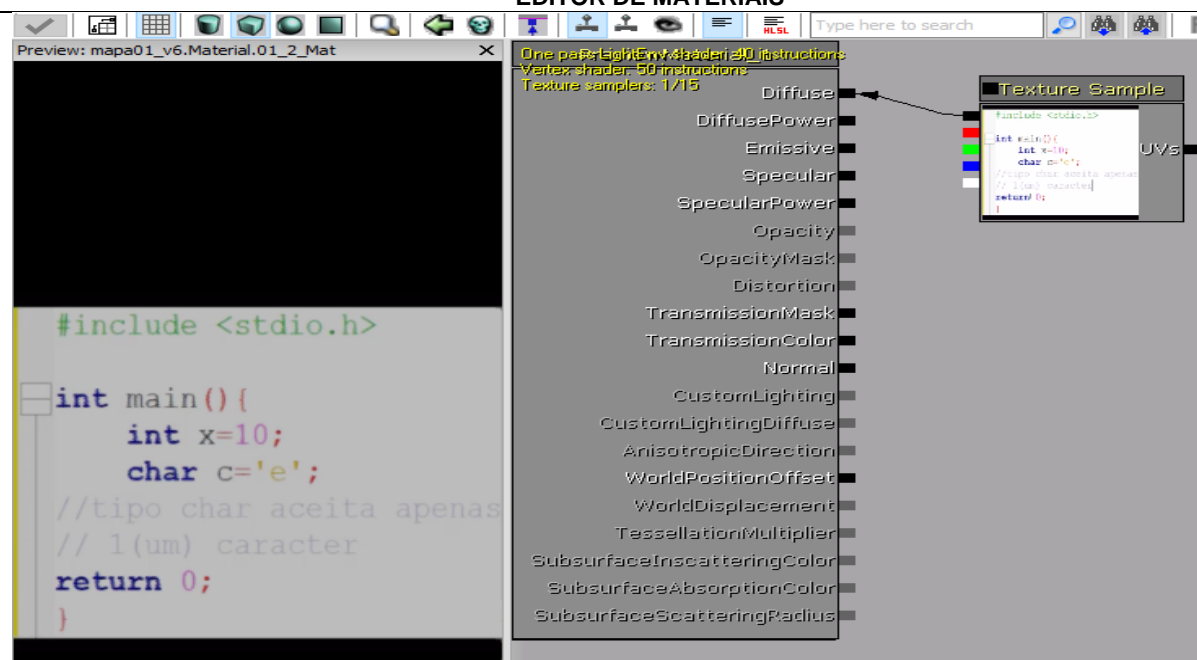
<p style="text-align: center;"><b>Sons</b></p>	<p style="text-align: center;"><b>Personagens</b></p> <pre>#include &lt;stdio.h&gt; int main() {     int x = 1;     int y = 2;     y += x + ++y;     if (y == 4)         printf("Atire aqui!");     return 0; }</pre>	<p style="text-align: center;"><b>Triggers</b></p>	
<p style="text-align: center;"><b>Luzes</b></p>	<p style="text-align: center;"><b>Partículas</b></p>	<p style="text-align: center;"><b>Teletransporte</b></p>	
<p style="text-align: center;"><b>Materiais</b></p>		<p style="text-align: center;"><b>Statis Meshs</b></p>	

	
<p style="text-align: center;"><b>Sistema de Partículas</b></p> 	<p style="text-align: center;"><b>Veículos</b></p> 
<b>UNREAL MATINEE</b>	
	
<p>A ferramenta de animação UnrealMatinee fornece a capacidade de animar as propriedades de atores ao longo do tempo, para criar tanto jogabilidade dinâmica ou sequências cinemáticas no jogo. O sistema baseia-se no uso de faixas de animação especializados em que você pode colocar keyframes para definir os valores de certas propriedades dos atores no nível. O Editor Matinee é semelhante aos editores usados para edição de vídeo, tornando-se familiar para os profissionais de vídeo. Diversas animações do jogo UNLOCK-C foram criadas através desta ferramenta. Como exemplos destacam-se:</p> <ul style="list-style-type: none"> <li>• Animação de início do jogo com passeio virtual pelo primeiro cenário e som ambiente</li> <li>• Animação de portas abrindo e fechando com som</li> <li>• Animação de câmeras focando códigos fonte em linguagem C para chamar atenção ao vídeo explicativo mostrado ao jogador como forma de corrigir seus erros.</li> </ul> <p style="text-align: center;"><b>VIEWPORTS</b></p>	



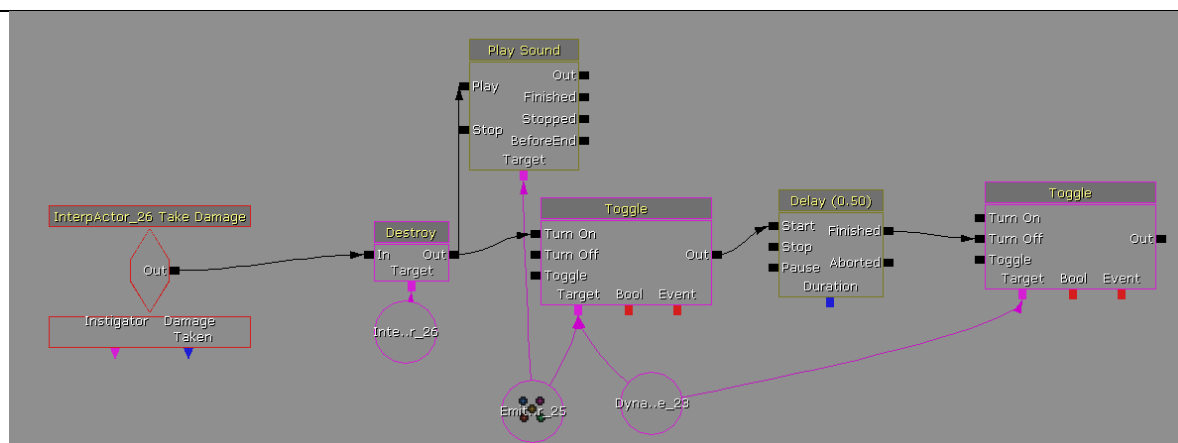
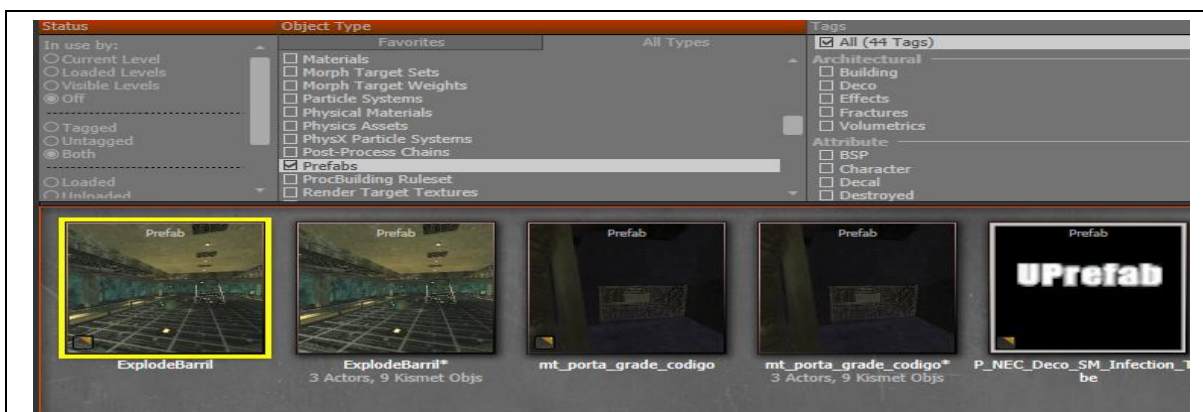
As viewports permitem visualizar a construção e manipulação de elementos do jogo sobre diversos tipos de visões. A visão de perspectiva, top, side e front foram utilizadas para ajustes finos e um melhor enquadramento dos itens de design inseridos durante a construção do jogo.

#### EDITOR DE MATERIAIS

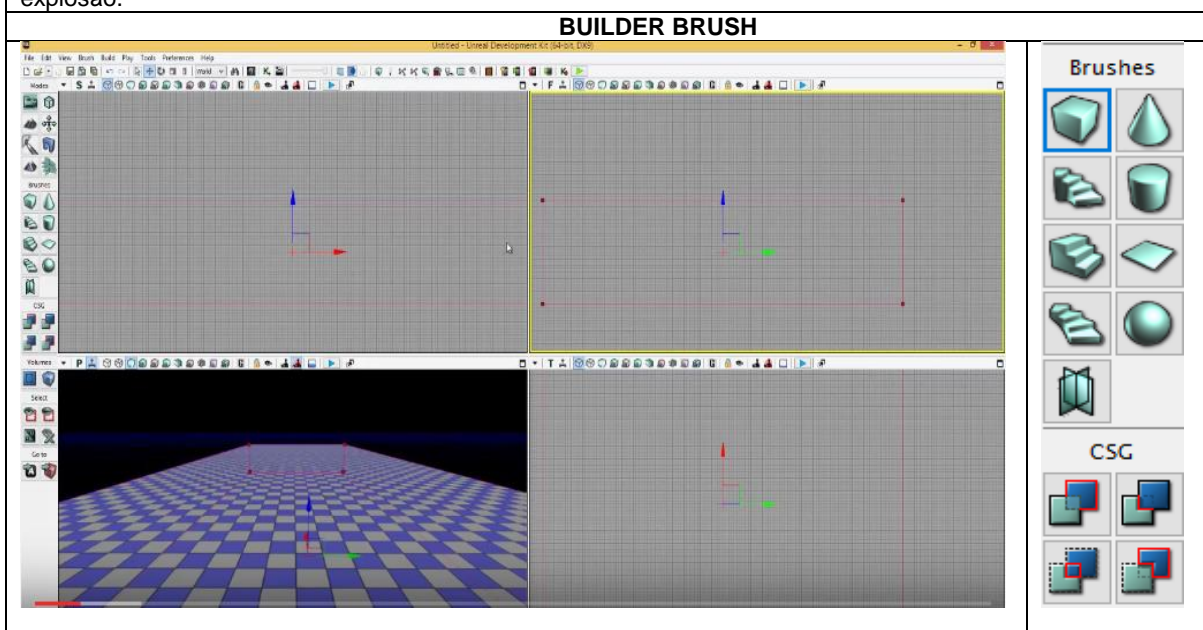


O editor de materiais permite a criação de materiais a partir de texturas. Este recurso permitiu criar materiais que representavam códigos em linguagem C sobre a forma de textura e vídeos.

#### PREFABS

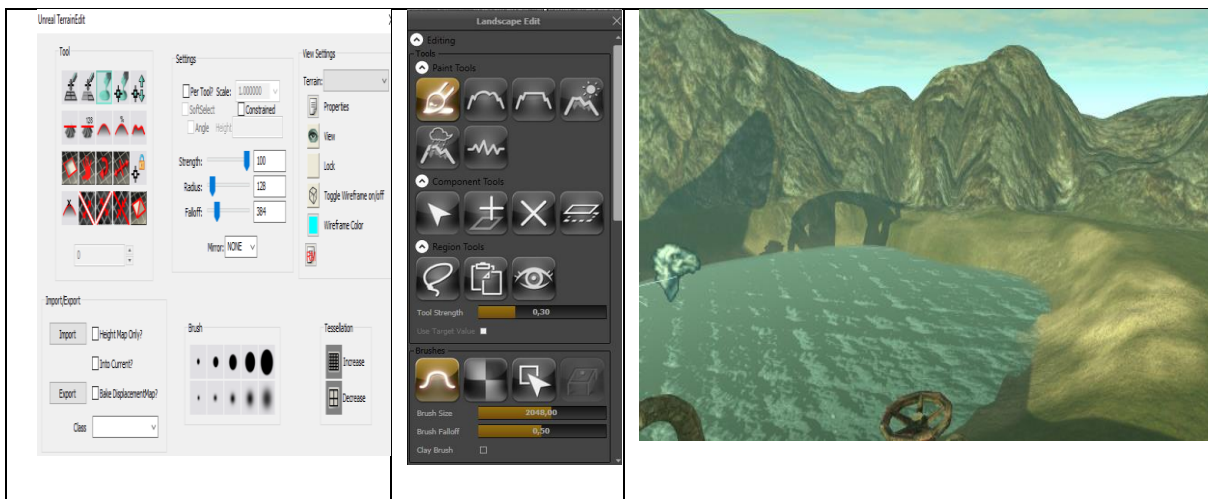


Prefabs são conjuntos de atores e programação(Kismet) associados que podem ser armazenados em um pacote e instanciados em um nível. Eles ajudam na produtividade de desenvolvimento do jogo, pois elementos complexos poderiam ser programados apenas uma vez e depois replicados. Como exemplo os barris que explodem ao serem atingidos por tiros do jogador e que caso esteja próximo ele irá morrer com a explosão.



Os brushes de geometria permitiram criar toda a geometria do mundo virtual. As salas, paredes, vão de portas, escadas, buracos, torres, prédios e outros elementos complexos foram construídos com estas ferramentas.

#### EDITOR DE TERRENOS



O editor de terrenos permitiu modelar o terreno, criar montanhas, lago e outros elementos.

### FOLIAGE



O sistema de foliagem permitiu decorar o terreno com boa produtividade permitindo inserir gramas, árvores, arbustos, etc.