

**UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA
DEPARTAMENTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

HEVERTON SANTOS QUEIROZ

DESENVOLVENDO JOGOS COM O GAME MAKER

Vitória da Conquista – BA
Abril de 2012

HEVERTON SANTOS QUEIROZ

DESENVOLVENDO JOGOS COM O GAME MAKER

Trabalho de Conclusão de Curso apresentada a Universidade Estadual do Sudoeste da Bahia, como requisito para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Fábio Moura Pereira.

Vitória da Conquista - BA
Abril de 2012

HEVERTON SANTOS QUEIROZ

DESENVOLVENDO JOGOS COM O GAME MAKER

Banca Examinadora:

Prof. Dr. Fábio Moura Pereira

Prof^a. Ms. Maísa Soares dos Santos Lopes

Prof^a. Ms. Maria Sílvia Barbosa

Aos meus pais, Jesuino e Aldir,
pela eterna confiança.

AGRADECIMENTO

Agradeço:

Aos meus pais, Jesuino Carlos Silva Queiroz e Aldir Santos Queiroz, pelo apoio que sempre ofereceram para mais uma conquista, mostrando desde cedo que com perseverança e dedicação os objetivos são alcançados

Ao meu irmão, João Gabriel Santos Queiroz, por me ajudar no andamento deste trabalho final.

À minha namorada, Heloísa Cintra Alves Pinto, pela paciência, inteligência, compreensão e força para mais uma vitória.

Ao meu orientador, professor e auxiliador, Fábio Moura Pereira, que sempre teve paciência e atenção nos momentos em que necessitei de ajuda; além da confiança depositada em mim.

A todos os professores, do curso de ciência da computação pela contribuição do conhecimento transmitido nestes últimos anos de estudos, e a todos que colaboraram de forma direta ou indireta neste trabalho.

"Tudo o que a mente humana pode
conceber, ela pode conquistar".
(Napoleon Hill)

RESUMO

O ensino de jogos de computadores em um curso de computação é um fator estimulante à motivação do discente. Fora da sala de aula, o mercado mundial de jogos tem mobilizado bilhões de dólares gerando grandes investimentos em mão-de-obra qualificada e em novas tecnológicas. Para um profissional que deseja ingressar na área de desenvolvimento de jogos é necessário conhecer os tipos de jogos, as plataformas onde os jogos são executados, os aspectos de desenvolvimento e as tecnologias utilizadas na fase de implementação. Este trabalho tem como objetivo o desenvolvimento de um jogo utilizando um motor de jogos e conhecimentos de Inteligência Artificial, de Programação Orientada a Objetos e de Engenharia de Software voltados à área de jogos, como por exemplo, o Game Design.

PALAVRAS-CHAVES: Desenvolvimento de jogos, Motor de Jogos, Inteligência Artificial, Game Design.

ABSTRACT

The teaching of computer games at a computer course is a factor stimulating the motivation of students. Out of the classroom, the global market for games has mobilized billions of dollars creating large investments in skilled labor and new technology. For professionals who want to join in the area of game development is necessary to know the types of games, platforms where games are played, the aspects of development and the technologies used in the implementation phase. This work has as objective the development of a game using a game engine and knowledge of Artificial Intelligence, Object Oriented Programming and Software Engineering focused on the area of games, such as Game Design.

KEYWORDS: Game Development, Game Engine, Artificial Intelligence, Game Design.

LISTA DE FIGURAS

Figura 1. Uncharted 3, jogo com melhor gráfico no ano de 2011 (ABRIL, 2012).....	23
Figura 2. Raquete para tênis que simula movimentos (WEBCHEATS, 2012).....	26
Figura 3. Estrutura geral de um motor de jogo (GOMES & PAMPLONA, 2005).	30
Figura 4. Mapa principal de God Of Arms.	34
Figura 5. Barra de contadores de God Of Arms.	35
Figura 6. Minigun.....	35
Figura 7. Armadilhas: Machado duplo, serra e espinhos.....	35
Figura 8. Godzilla, chefe da primeira fase.	36
Figura 9. Diagrama de navegação de God Of Arms	37
Figura 10. Menu Pausa do Mapa Principal	37
Figura 11. Linguagem de <i>script</i> GML.....	40
Figura 12. <i>Drag-and-drop</i> , interface do Game Maker.	42
Figura 13. Gráfico da avaliação de God Of Arms.	44

LISTA DE ABREVIATURAS E SIGLAS

2D	Duas dimensões.
3D	Três dimensões.
ABRAGAMES	Associação Brasileira das Desenvolvedoras de Jogos Eletrônicos
ACM	(<i>Association for Computing Machinery</i>) – Maior sociedade do mundo da informática educativa e científica.
API	(<i>Application Programming Interface</i>) – Interfaces de Programação de Aplicativos.
ARB	(<i>OpenGL Architecture Review Board</i>) – Conjunto de empresas de que supervisionava o OpenGL.
CAD	(<i>Computer-aided design</i>) – Ferramenta de computador para auxiliar em desenhos.
CPU	(<i>Central Processing Unit</i>) – Unidade Central de Processamentos.
Dark AI	(<i>Dark Artificial Intelligence</i>) – Motor de Inteligência Artificial do motor de jogo Dark Basic.
Dark NET	(<i>Dark Network</i>) – Motor de rede do motor de jogo <i>Dark Basic</i> .
DBPro Free	(<i>Dark Basic Professional Free</i>) – Versão profissional livre do motor de jogo <i>Dark Basic</i> .
ENIAC	(<i>Electronic Numerical Integrator and Computer</i>) – Computador e Integrador Numérico Eletrônico.
FPS	(<i>First Person Shooter</i>) – Perspectiva de ter o mesmo ponto de vista do mundo do personagem controlado.
GDC	(<i>Computer Game Developers Conference</i>) – Conferência de desenvolvedores de jogos de computador.
GDD	(<i>Game Designer Document</i>) – Documento base do jogo para os desenvolvedores.
GM	(<i>Game Maker</i>) – Motor de jogo.
GML	(<i>Game Maker Language</i>) – Linguagem <i>script</i> do Game Maker.
GUI	(<i>Graphic User Interface</i>) – Interface gráfica do usuário.
IA	Inteligência Artificial.
IHC	Interação humano-computador.

MIT	(<i>Massachusetts Institute of Technology</i>) – Universidade privada de pesquisa.
NPC	(<i>Non-player character</i>) – personagens controlados pelo computador
OpenAL	(<i>Open Audio Library</i>) – Biblioteca de Áudio Livre.
OpenGL	(<i>Open Graphics Library</i>) – Biblioteca Gráfica Livre.
PC	(<i>Personal computer</i>) – Computador pessoal.
RPG	(<i>Role Playing Game</i>) – Jogo de interpretação de personagens.
SDL	(<i>Simple DirectMedia Layer</i>) – Biblioteca multimídia e de multiplataforma que cria uma abstração em várias plataformas de gráficos, sons, dispositivos de entrada.
SGI	(<i>Silicon Graphics, Inc.</i>) – Empresa de soluções de alto desempenho para computação.
TI	Tecnologia da Informação.
XP	(<i>Extreme Programming</i>) – Metodologia ágil de desenvolvimento de <i>software</i> .

SUMÁRIO

1.	INTRODUÇÃO	12
1.1	Objetivos.....	13
1.1.1	Objetivo geral	13
1.1.2	Objetivos específicos	13
1.2	Justificativa.....	13
1.3	Metodologia	14
1.4	Estrutura do documento.....	14
2.	DESENVOLVIMENTO DE JOGOS.....	16
2.1	O mercado de jogos eletrônicos.....	16
2.1.1	Jogos em geral.....	16
2.1.2	Jogos eletrônicos	16
2.1.3	Tipos de jogos.....	17
2.1.4	Plataformas	19
2.1.5	Mercado de jogos eletrônicos.....	20
2.2	Aspectos de desenvolvimento.....	21
2.2.1	Processo (Game Design)	22
2.2.2	Arte gráfica.....	23
2.2.3	Sons	24
2.2.4	IA para jogos	24
2.2.5	Interação	25
2.3	Desenvolvimento de jogos para computador.....	26
2.3.1	Histórico	26
2.3.2	Plataformas de desenvolvimento	27
2.3.3	Ferramentas/ linguagens de desenvolvimento	28

2.4	Motores para desenvolvimento de jogos	29
2.4.1	Características essenciais.....	30
2.4.2	Características importantes.....	31
2.4.3	Motores de jogos para PC.....	31
3.	ESTUDO DE CASO	34
3.1	Fluxo do jogo	34
3.2	Elementos do gameplay	35
3.3	Inimigos e armadilhas	35
3.4	Telas de apoio e elementos de interação off-game	36
3.5	História	38
3.6	Fases do jogo	38
4.	DESENVOLVIMENTO E IMPLEMENTAÇÃO	39
4.1	Desenvolvimento de <i>God Of Arms</i>	39
4.2	Avaliação do Game Maker	41
4.2.1	Características essenciais.....	41
4.2.2	Características importantes.....	42
4.3	Testes do jogo	44
5.	CONCLUSÃO	46
5.1	Trabalhos futuros	47
6.	BIBLIOGRAFIA	48
	APÊNDICE A - DOCUMENTO GAME DESIGN	52
	APÊNDICE B QUESTIONÁRIO DO JOGO	74

1. INTRODUÇÃO

Os jogos de computador têm encantado o mundo. Adultos e jovens estão conectados às rápidas e atraentes emoções que um jogo pode proporcionar à mente. Os jogos de computador vêm sendo utilizados em diferentes áreas em todo o mundo, seja no campo da diversão; da educação, por meio de software utilizado para o desenvolvimento intelectual de crianças, como os jogos matemáticos e lógicos; empresarial, através do auxílio ao desenvolvimento e aperfeiçoamento dos recursos humanos; e militar, através de simuladores para treinar indivíduos em diferentes circunstâncias.

O ensino de jogos de computadores em um curso de computação ou similar tem como caráter significativo despertar no discente um maior empenho e motivação com as matérias proporcionadas pelo seu curso (DOS SANTOS, 2007). Além disso, fora da sala de aula, nota-se que o mercado de trabalho para desenvolvedores de *games* está aumentando e conquistando importância, pois o mercado mundial mobiliza bilhões de dólares para o desenvolvimento deste tipo de programa. Assim, empresas de Tecnologias da Informação (TI) e de entretenimento estão fazendo grandes investimentos em mão-de-obra qualificada e em desenvolvimento de mecanismos computacionais sofisticados (PERUCIA, BALESTRIN & VERSCHOORE, 2011).

A área de jogos é multidisciplinar, se a intenção de quem quer trabalhar nesse campo é ser um *designer* de jogos, então, é importante conhecer arte e cultura; se a finalidade for animação, é importante ter habilidades artísticas; por último, se o design for à área de implementação do jogo, é fundamental ter raciocínio lógico, além de conhecimentos físicos, matemáticos e de programação.

O intuito deste trabalho é mostrar que através do desenvolvimento de jogos é possível empregar conceitos de diferentes áreas do curso de Ciência da Computação. Por exemplo, partindo de princípios básicos de programação e matemáticos poderão ser criados a física e os movimentos do jogo, baseado nos fundamentos de Teoria da Computação e de Inteligência Artificial (IA) poderão ser criados agentes inteligentes para aumentar a atratividade e realidade do jogo. Cada

disciplina pode ser interligada a um diferente aspecto do desenvolvimento de jogos.

Foi utilizado como estudo de caso um jogo de aventura/ação com uma visão lateral de duas dimensões (2D). Esses jogos já fascinavam a todos desde o início da década de 80. O jogo tem um tema de guerra e segue a uma história fictícia. O processo de criação do jogo utilizou técnicas de Game Design e foi utilizado um motor de jogo para todo o processo de implantação.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Desenvolvimento de um jogo de aventura/ação empregando a técnica de Game Design, com o uso do motor de jogo Game Maker para a plataforma Windows.

1.1.2 Objetivos específicos

- ✓ Estudar e compreender o ambiente e as características do motor de jogo para utilização como ferramenta base para o desenvolvimento de jogos para a plataforma Windows.
- ✓ Utiliza técnica de Engenharia de Software, como o Game Design, voltada ao desenvolvimento de jogos.
- ✓ Avaliar a empregabilidade da ferramenta Game Maker para o desenvolvimento de jogos.
- ✓ Testar a atratividade e a jogabilidade do jogo com base num grupo de usuários.

1.2 JUSTIFICATIVA

Tendo em vista o crescimento do mercado de jogos no Brasil e no mundo, é fácil perceber que cada vez mais será necessária mão-de-obra qualificada nesse campo. Por causa dessa falta de profissionais, a área de criação de jogos se oferece

como uma grande oportunidade de trabalho. Apesar das universidades fornecerem profissionais capacitados ao mercado, o problema é que eles não são voltados para a área de *games*.

O curso de Ciência da Computação oferece conhecimentos técnicos e teóricos necessários para a criação de jogos, seja em programação ou fornecendo metodologias e técnicas adequadas. Empregar esse conhecimento em um caso real justificou o desenvolvimento desse trabalho.

1.3 METODOLOGIA

O presente trabalho apresenta uma pesquisa bibliográfica a respeito do desenvolvimento de jogos. Em seguida, utiliza a técnica de estudo de caso que demonstra o uso de um motor para o desenvolvimento de jogos que é classificada como estudo exaustivo e profundo de um ou mais objetos de maneira que se permita o seu amplo e detalhado conhecimento (MINAYO, 2007). Para a confecção deste projeto foram realizadas pesquisas nos seguintes meios de comunicação: sites, livros e artigos.

1.4 ESTRUTURA DO DOCUMENTO

Esta monografia está dividida em seis capítulos. O Capítulo 1 (Introdução) contém a apresentação resumida deste trabalho, exibindo, de forma breve, a contextualização do desenvolvimento de jogos. Além dos objetivos e da motivação relativos ao desenvolvimento deste trabalho.

O Capítulo 2 (Fundamentação Teórica) é um estudo sobre o processo de desenvolvimento de jogos, apresentando as características sobre o desenvolvimento e mercado de jogos. Este capítulo ainda trabalha com os aspectos do processo de criação do jogo e seus elementos, além do histórico dos jogos de computadores e as tecnologias utilizadas na fase de desenvolvimento dos mesmos. Por fim, o capítulo aborda os motores de jogos de computadores e suas respectivas características.

O Capítulo 3 (Estudo de Caso) apresenta um resumo do Game Design, do jogo desenvolvido neste trabalho, utilizado durante toda a fase de projeto e de

desenvolvimento do jogo.

O Capítulo 4 contém o desenvolvimento e a implementação do jogo construído, além da avaliação do motor de jogos utilizado e o resultado de testes de atratividade e jogabilidade do jogo realizado por um grupo de usuários.

O Capítulo 5 contém a conclusão do trabalho, além dos trabalhos futuros que podem ser realizados a partir deste.

O Capítulo 6 apresenta as referências bibliográficas que foram utilizadas no referencial teórico deste trabalho.

No apêndice A pode ser visto o documento do Game Design utilizado para desenvolver o jogo do estudo de caso. No apêndice B pode ser visto o questionário utilizado para avaliar o jogo do estudo de caso.

2. DESENVOLVIMENTO DE JOGOS

Neste capítulo são abordados diferentes aspectos do desenvolvimento de jogos, incluindo a situação do mercado de jogos eletrônicos, os aspectos de desenvolvimento, o desenvolvimento de jogos para como computadores pessoais (PC's) e motores para desenvolvimento de jogos.

2.1 O MERCADO DE JOGOS ELETRÔNICOS

O profissional que atua no ramo de desenvolvimento de jogos eletrônicos tem uma perspectiva de mercado em grande crescimento, o que pode ser comprovado com a expansão da indústria de jogos no Brasil e em vários outros países (GAMEVICIO, 2011). O desenvolvimento de jogos depende diretamente do mercado, por isso o desenvolvedor, ao pensar em criar jogos, deve, fundamentalmente, refletir sobre qual o tipo de jogo poderá ser aceito neste mercado e qual o tipo de plataforma em que este jogo será executado. A seguir são apresentadas características sobre o desenvolvimento e mercado de jogos.

2.1.1 Jogos em geral

O homem sempre teve o anseio de conhecer o universo, desta curiosidade usou-se a ação, representação e imitação como formas de se expressar, assim, produzindo maneiras de viver melhor com a natureza. Esta necessidade, fez com que surgisse, desde os começos das sociedades humanas e de suas culturas, o jogo. O jogo permite entrar em um mundo imaginário, fazendo com que o jogador viva esse mundo como verdade absoluta. Basicamente, o jogo é uma atividade não-produtiva em uma realidade alternativa que abrange a diversão e a improbabilidade conduzida por regras (OLIVEIRA, 2010).

2.1.2 Jogos eletrônicos

O jogo eletrônico é como um jogo qualquer, porém jogado em um ambiente digital, por exemplo, computador, videogame, dispositivo portátil etc. O jogo digital

não é somente um software que processa a entrada do usuário e devolve a este uma saída, além disso, este tipo de jogo deve realizar estas funções em um ambiente interativo e esta interação apenas acaba até que seja alcançado um fim por alguma regra determinada ou pelo próprio usuário. (OLIVEIRA, 2010).

Os jogos eletrônicos não são simples passatempo dos jovens, atualmente, eles estão ganhando cada vez mais espaço no cotidiano de toda a sociedade, pois eles vêm trazendo uma nova reestruturação do mundo do entretenimento, da informação e da educação. O grande faturamento deste mercado propicia o alto investimento que, por sua vez, possibilita a criação de jogos cada vez mais fascinantes. A atração por este ambiente de entretenimento traz ao usuário uma quantidade significativa de informações, conhecimentos e valores de uma maneira divertida (TAVARES, 2006).

2.1.3 Tipos de jogos

O projeto de um jogo deve começar pela definição do tipo de jogo que será desenvolvido. Jogos são subdivididos em diversos tipos. Os principais tipos de jogos em relação ao gênero são: aventura, ação, estratégia, lógico, passatempo, educação e entretenimento, RPG, esportivo, online, simulação. Abaixo seguirá uma pequena síntese sobre cada tipo de jogo.

- Jogos de aventura: caracterizados por incluir raciocínio e atividades lúdicas, ele ainda estimula a exploração exaustiva de cada cena do jogo. Normalmente, um jogo de aventura é aquele em que o jogador atua como um herói em busca de um objetivo, passando por diversos obstáculos. Alguns exemplos desse estilo são *Super Mario World*, *Donkey Kong Country*, *The Legend of Zelda* e *Tomb Raider*.
- Jogos de ação: a característica principal deste estilo de jogo é exigir habilidade do jogador com os controles e comandos, envolvendo ações de curta duração e atividades de causa e efeito. Este tipo de jogo geralmente abrange outros tipos, como jogos de luta, combate, tiro etc. Um jogo de ação é aquele em que o jogador emerge em um mundo caótico, e seu primeiro objetivo é sobreviver para chegar ao final. Fazem parte deste tipo, jogos como *Doom*, *Max Payne*, *Street Fighter*, *Counter Strike* etc.

- Jogos de Estratégia: têm como foco principal a habilidade do jogador. São idealizados a fim do jogador cumprir objetivos usando estratégias e táticas. Cada jogador controla uma base, constituída de várias instalações e estruturas, e um exército de unidades de guerra. Vence o jogo aquele que demonstrar sua supremacia, através do desenvolvimento de sua base ou da destruição das bases dos outros jogadores. Exemplos de jogo de estratégia seriam o *Warcraft*, *Age Of Empires*, *Commandos* e *Starcraft*.
- Jogos lógicos: desafiam muito mais a mente do que os reflexos e podem ter um limite de tempo dentro do qual o usuário deve finalizar a tarefa. Aqui podem ser incluídos clássicos, tais como xadrez e damas.
- Jogos passatempo: são jogos simples como caça-palavras, palavras-cruzadas, quebra-cabeças rápidos e sem nenhuma história. Este tipo de jogo pode ser exemplificado por jogos como Paciência e MahJong.
- Jogos para educação: são jogos com as características de qualquer um dos jogos anteriores, porém, com fins educacionais. Jogos como *Jogo das cores-Colors Game*¹ e *Tabuada do Dino*².
- Jogos *Role Playing Game* (RPG): jogos RPG têm como característica o desenvolvimento gradativo dos personagens, ou seja, avançando em níveis através da obtenção de experiência, executando ações como combater ou resolver problemas (*quests*). Geralmente são feitos com uma linha de estória principal, com problemas que permitem múltiplas soluções, além de vários objetivos secundários opcionais. Para resolver problemas é importante a interação com outros personagens. O jogador pode customizar os personagens, tornando o jogador parte do mundo do jogo. Alguns exemplos desse gênero são *Chrono Trigger*, *Final Fantasy*, *Diablo*, *Fable* e *Dragon Age*.
- Jogos esportivos: são simuladores de esportes do mundo real, como futebol, basquete, tênis etc. Os jogos *Pro Evolution Soccer*, *F1*, *NBA Street* e *International Volleyball* são alguns exemplos deste tipo.

¹ http://www.educacaodinamica.com.br/games/colors/jogo_colors_cores.html

² <http://www.escolagames.com.br/jogos/tabuadaDino/>

- Jogos online: não possuem um tipo distinto, alguns são de RPG, outros de ação, outros de aventura, estratégia, ou esportes. O grande diferencial é o fato de ser jogado por meio da rede de computadores. Exemplos de jogos online são *World Of Warcraft*, *The Crims* e *Lineage*.
- Jogos de Simulação: são jogos que requerem reflexo do usuário na realização de habilidades num ambiente limitado que tenta retratar a realidade, ou parte dela. Os simuladores são utilizados tanto para diversão como para treinamento de profissionais. Para uso profissional, os ambientes simulados testam reflexos, aplicam e avaliam conhecimentos previamente adquiridos, e medem as reações dos profissionais em situações críticas. Fazem parte desse gênero, jogos como *The Sims 3*, *Flight Simulator*, *SimCity*, *Second Life* etc.

2.1.4 Plataformas

É de extrema importância para um desenvolvedor conhecer a plataforma de *hardware* sobre a qual o jogo será executado. Geralmente, os games são desenvolvidos para serem jogados em apenas uma plataforma (JUNIOR, NASSU, JONACK, 2002). As plataformas serão as tecnologias em que o jogo irá ser executado, existem basicamente quatro classificações de plataformas para jogos. Os jogos podem ser executados nas plataformas: arcade, console, PC ou móvel.

- Arcade: uma das plataformas mais conhecidas, antigas e distintas pelo grande porte, chamada de fliperama no Brasil. Os fliperamas executam jogos de diversos gêneros. Essas plataformas tiveram o seu sucesso entre as décadas de 80 e 90 quando eram usadas em comércios ou em casas de jogos. Atualmente, os arcades são pouco utilizados, pois o seu uso foi diminuindo com o crescimento dos consoles e dos *games* de computadores (PEREIRA, 2006).
- Console: também conhecido como videogame, é uma das plataformas mais vendidas, por causa do seu preço acessível, portabilidade e a enorme quantidade de jogos que são lançados para esta plataforma. No começo, os videogames tinham gráficos pobres, exemplo disso seria o clássico Atari.

Hoje em dia, com a importância que as pessoas dão a essas plataformas e com os avanços nas áreas tecnológicas seus recursos gráficos foram aperfeiçoados, competindo com os computadores pessoais. Alguns exemplos de videogames são *Super Nintendo*, *PlayStation*, *Dreamcast*, *Xbox* e *Nintendo Wii*.

- Computador pessoal (PC): Após surgir os computadores pessoais e as *Graphic User Interface* (GUI) que permitem usuários utilizarem os computadores de modo mais simplificado, a indústria de jogos eletrônicos investiu no mercado de jogos para este tipo de tecnologia. Os PC's são os aparelhos que mais passaram por evoluções tecnológicas, e atinge um enorme número de usuários (FELDMAN, 2004). Apesar disso, eles ainda perdem para os consoles em relação ao entretenimento, pois alguns fatores como a dificuldade de manipular os jogos, custo e pirataria são culpados por usuários deixarem os computadores para os videogames. Apesar desse aspecto negativo relacionado aos computadores, o mercado de jogos para PC's obtém uma rentabilidade altíssima. Pois, há jogos que rodam somente em computador por causa do elevado requisito de hardware que o jogo exige, pode-se afirmar que os jogos são os próprios motivadores ao surgimento e aprimoramento das tecnologias de computadores, são eles, os principais responsáveis pelo progresso das placas gráficas e de áudio. (LOWOOD, 2004)
- Plataforma Móvel: A utilização desta plataforma deu-se início na década de 90, com o aparecimento dos *minigames*. Apesar das primeiras versões não apresentarem gráficos de alta definição obtiveram grande sucesso devido a sua fácil mobilidade e por apresentarem diversas opções de jogo. Atualmente, apresentam uma enorme capacidade de processamento de imagem e áudio (OUTERSPACE, 2011). Exemplos de dispositivos móveis são os *celulares*, *smartphones*, *Gameboy* e *iPad*.

2.1.5 Mercado de jogos eletrônicos

A área de jogos eletrônicos é uma área de grande interesse financeiro principalmente no segmento de entretenimento digital podendo ser encontrado em diversos dispositivos, PC's, consoles domésticos e dispositivos móveis. Além disso,

a indústria dos *games* é uma das maiores do mundo, ela compete com a bilionária indústria do cinema em termos de arrecadamento, uso de efeitos especiais, gastos com publicidade. (PORTALMULTIRIO, 2011).

No mercado mundial, a indústria de jogos eletrônicos já ultrapassou e muito o lucro de outros setores de entretenimento. O vice-presidente de Relações Institucionais da ABRAGAMES mostrou dados que comparavam o faturamento entre elas nas primeiras 24h, visando a diferença significativa entre os valores do jogo *Call Of Duty: Modern Warfare 2* com US\$ 400 milhões, o livro *Harry Potter e as Relíquias da Morte* com US\$ 220 milhões e *Cinema: Lua Nova* com US\$ 73 milhões (TECHTUDO, 2011).

Mesmo com o mercado mundial de jogos superando os de outros setores do entretenimento, no Brasil este mercado ainda é pequeno, profissionais brasileiros criativos acabam trabalhando em outros países (GAMEVICIO, 2011). De acordo com o diretor executivo da distribuidora NC Games o mercado nacional movimentou aproximadamente R\$ 950 milhões em 2010. O Brasil, nos últimos anos, demonstrou ser um grande consumidor de entretenimento digital e, por isso, está sendo considerado como prioridade para grandes empresas deste negócio. Porém, a alta tributação, a pirataria e a importação ilegal são os maiores obstáculos para o crescimento do mercado de jogos brasileiro (METODISTA, 2011).

2.2 ASPECTOS DE DESENVOLVIMENTO

A imaginação é um aspecto importantíssimo ao desenvolvimento de jogos, pois exige muita criatividade para chamar a atenção de quem o joga. Outro aspecto imprescindível à criação de jogos é envolver uma metodologia de desenvolvimento, pois, um jogo nada mais é que um software com arte, inteligência artificial e interatividade. Jogos com métodos de desenvolvimento muito pobre tendem a serem mais dispendiosos, demorarem mais tempo e não atenderem aos requisitos iniciais (PEREIRA, 2006).

Contudo, os jogos eletrônicos possuem uma arquitetura mais complexa do que um *software* comum e por isso as metodologias utilizadas nestes programas apresentam limitações acerca do desenvolvimento de jogos. Existe uma metodologia voltada excepcionalmente ao processo de criação de jogos, o Game Design, que

aborda, do melhor modo possível, os elementos envolvidos nos ciclos do desenvolvimento de um jogo (PEREIRA, 2006). A seguir é apresentada a metodologia Game Design.

2.2.1 Processo (Game Design)

O Game Design é visto como um processo de imaginar um jogo, definição de como o jogo vai funcionar, comunicação da equipe e descrição dos itens que irão criar o jogo. Porém, mesmo com uma metodologia específica para o desenvolvimento de jogos, pode-se destacar a enorme dificuldade em coletar os requisitos do sistema, por causa das influências sofridas pelos jogos, como por exemplo, os avanços tecnológicos (LUZ, 2004).

O processo de Game Design começa na conceituação do jogo. O conceito do jogo consiste na criação, refinação ou formalização da idéia inicial do jogo. A idéia inicial do jogo é imaginar de qual tipo o jogo será, em qual plataforma o jogo será executado, como ele será visto pelo jogador, entre outros. Um *game designer* experiente já trabalha na criação de idéias apurando o jogo. Após a refinação deve-se construir o documento que irá formalizar o jogo, o Game Concept. O Game Concept deverá conter informações importantes em relação ao jogo, por exemplo, o gênero, plataforma, descrição etc. (ROUSE, 2005). A partir do Game Concept começa-se a desenvolver os elementos e as características do jogo, essa etapa é chamada de projeto do jogo. Nessa fase é definido, por exemplo, o modo com os personagens irão raciocinar, agir e sentir, as regras do jogo, o mundo do jogo, o modo como o jogo é visto pelo jogador e a interação jogador-mundo baseada nas regras (ROLLINGS & MORRIS, 2004).

Após ter finalizado a fase de projeto é iniciada a etapa de desenvolvimento do jogo que tem como objetivo implementar o *game* de acordo ao que foi especificado na etapa de projeto, além de escolher as ferramentas de suporte que serão usadas no desenvolvimento (PEREIRA, 2006). Durante a fase de desenvolvimento, todos os elementos da fase de projeto e as ferramentas de suporte serão reunidos em um documento chamado Game Designer Document (GDD). Este documento é a principal fonte para tirar dúvidas e comunicação da equipe, fornecendo assim, um desenvolvimento do jogo padronizado. (ANDROID, 2012).

Depois do desenvolvimento é iniciada a etapa de testes, que envolve a procura de falhas de sistema, aceitação dos jogadores e a jogabilidade que é a facilidade na qual o jogo pode ser entendido e jogado (PEREIRA, 2006). Os testes de jogabilidade são analisados pelo *playtest* que poderá ser feito por qualquer pessoa (JUNIOR et al., 2002).

2.2.2 Arte gráfica

É a parte visual do jogo, em que se definem texturas, modelagem dos personagens, objetos e cenário. É um componente importantíssimo nos jogos, pois por ser o primeiro elemento do jogo que vai interagir com o jogador, deverá fazer com que o jogador se sinta imerso no universo virtual do jogo através da atração visual. Para isso é necessário que o cenário e personagens estejam de acordo à história e ao contexto em que o jogo se baseia (BETHKE, 2003).

Inicialmente, os jogos costumavam ter um único artista para o desenho de personagens e para os cenários do mundo. Nos anos 90, após o lançamento do jogo Quake, em três dimensões (3D), os requisitos de arte e seu orçamento cresceram substancialmente no desenvolvimento de jogos (BETHKE, 2003). Na figura 1 pode ser visto *Uncharted 3*, escolhido pela *Video Game Awards* como o jogo com melhor gráfico no ano de 2011.



Figura 1. Uncharted 3, jogo com melhor gráfico no ano de 2011.
Fonte: (ABRIL, 2012).

A fase inicial do desenho é a criação dos rascunhos de objetos, cenários e personagens que vai interatuar com o jogador. A partir deste momento, os esboços serão usados pelo time de computação gráfica que usarão motores gráficos com bibliotecas voltadas ao desenvolvimento de aplicações de duas e três dimensões, de modelagem e de animações (BETHKE, 2003).

2.2.3 Sons

O Áudio é tão importante quanto a arte gráfica e a interatividade, o que é provado pelo fato de um *game* se tornar monótono, ou até mesmo não ser jogado, se este não tiver som de qualidade. Entretanto, a qualidade do som é diretamente proporcional ao seu tamanho, por isso a equipe de áudio deve dar atenção à preferência entre o tamanho e a qualidade dos sons, a fim de proporcionar uma boa qualidade sonora ao jogo sem sobrecarregar o computador. Atualmente, a arte de áudio é basicamente dividida em duas partes: o efeito sonoro e o efeito musical (PEREIRA, 2006).

Igualmente ao mundo real, os efeitos sonoros são responsáveis por alertar algum acontecimento que ocorreu ou que está próximo de ocorrer, por isso eles se tornam um grande responsável pelo *feedback* a quem joga. Como exemplos de efeitos sonoros, poderiam ser citados uma pessoa ofegante, o latir de um cão, som de um tiro etc. Habitualmente, o efeito musical de um *game* é correspondente ao tema do jog. Além disso, a música pode ser o maior fator de imersão do jogador em determinados desafios, por exemplo, é a música que dá o tom de tensão quando o jogador enfrentar o chefe da fase (PEREIRA, 2006).

2.2.4 IA para jogos

A Inteligência Artificial é o estudo e o projeto de agentes inteligentes, sistemas que percebem os seus ambientes e executam ações que maximizam as suas chances de sucesso (RUSSELL & NORVIG, 2004). Dentro deste estudo, o campo de jogos eletrônicos é um dos mais fascinantes, pois inúmeros fatores causam este deslumbre, dentre eles, o mercado competitivo dos jogos exige continuamente

técnicas aperfeiçoadas de IA, o faturamento deste mercado (SILVA, 2005).

A Inteligência Artificial utilizada em jogos, chamada de Game AI, é diferente da IA acadêmica que estuda a maneira como o sistema vai pensar na resolução de problemas específicos, busca atender a um mercado de consumidores que procuram por uma diversão associada ao contexto do jogo, ou seja, a dificuldade é como o sistema vai agir no mundo virtual (SCHWAB, 2009) (FUNGE, 2004). A Inteligência Artificial em jogos não pode agir perfeitamente, não deixando o jogador vencer, porque irá desestimulá-lo e também não pode ser fácil a ponto de desanimá-lo (LUZ, 2004).

2.2.5 Interação

A afinidade entre o homem e a máquina aumentou desde o primeiro computador digital eletrônico, o ENIAC. Porém, o uso de interfaces que facilitaram a interação entre eles deu início somente com o surgimento dos microcomputadores e o advento da WEB, respectivamente nas décadas de 80 e 90 (AGNER, 2002). Segundo a *Association for Computing Machinery* (ACM), o termo interação humano-computador (IHC) é definido como uma área que diz respeito a projetar, avaliar e implementar sistemas de computador interativos para uso humano e ao estudo dos elementos centrais que os abrangem (CARVALHO, 2003).

Na área de jogos, o elemento interativo é a maneira de como o jogador vê, escuta e executa ações no mundo virtual do jogo (ROLLINGS & MORRIS, 2004). Algumas dicas podem ser utilizadas por desenvolvedores de jogos para obter um *game* com boa interação, por exemplo, a consistência das telas e menus, oferecendo ao jogador a padronização de ações, teclas de atalhos para comando, proporcionando praticidade, efeitos sonoros correspondentes a eventos do jogo e músicas de fundo que trabalhe com o emocional do jogador (LUZ, 2004). Na figura 2 pode ser vista a raquete de tênis, desenvolvida pela companhia Nintendo, que simula os movimentos do jogador.



Figura 2. Raquete para tênis que simula movimentos.

Fonte: (WEBCHEATS, 2012).

2.3 DESENVOLVIMENTO DE JOGOS PARA COMPUTADOR

A abordagem deste assunto contém informações relativas ao histórico dos jogos para computador e aos elementos que auxiliam o processo de desenvolvimento de jogos de computador, como a apresentação de plataformas gráficas, ferramentas e linguagens de desenvolvimento.

2.3.1 Histórico

O primeiro jogo foi desenvolvido pelo físico Willy Higinbotham, em meados da década de 50. Era um jogo de tênis simples desenvolvido por ele, visualizado em um osciloscópio e processado por um computador analógico, que logo fez sucesso e começou a ser cobiçado pelo público que freqüentava o laboratório a fim de poder jogá-lo. Em 30 de julho de 1961, um grupo de estudantes do *Massachusetts Institute of Technology* (MIT) testava o jogo eletrônico *Spacewar*, desenvolvido em um enorme computador que custava 120.000 dólares. (OUTERSPACE, 2011).

Em março de 1986, houve a primeira *Computer Game Developers Conference* (GDC), em que Chris Crawford, um *game designer* famoso, juntou-se com poucos amigos desenvolvedores de jogos e discutiram sobre Game Design (REIS, 2009).

Um fato marcante na história dos jogos para computador foi o surgimento do *game Wolfstein 3D*, desenvolvido pela ID Software em 1991. Ele foi o primeiro jogo de três dimensões, além de também ser o pioneiro em utilizar o *First Person Shooter*

(FPS) que permite ao jogador a perspectiva de visualizar o mundo através dos olhos do personagem que ele controla (REIS, 2009).

2.3.2 Plataformas de desenvolvimento

Enquanto o mercado ainda era dominado por games que utilizavam gráficos 2D simples e a quantidade de placas de vídeo distintas era pequena os programadores de jogos acessavam diretamente os recursos da placa. Porém, com o início da era 3D isso ficou inviável, por causa da dificuldade do trabalho e a enorme variedade de placas. Esta situação originou o aparecimento das Interfaces de Programação de Aplicativos (API) para a criação de gráficos 3D, que simplificam o trabalho do desenvolvedor, permitindo a utilização de elementos gráficos padronizados, para a utilização de qualquer placa 3D que seja compatível. Atualmente, o OpenGL e o Direct3D são as plataformas que lideram o mercado (MORIMOTO, 2012).

O *Open Graphics Library* (OpenGL) é um programa de baixo nível que fornece ao programador uma interface para *hardware* gráfico. A principal vantagem que o OpenGL tem em relação a seus concorrentes é que ele é executado em muitas plataformas diferentes, como o Windows, Linux, Mac OSX e dispositivos portáteis. OpenGL foi projetado para renderizações 3D com aceleração de *hardware*. O OpenGL é usado em muitos tipos de aplicações nas diversas áreas, como *Computer-aided design* (CAD), criação de conteúdo, energia, entretenimento, realidade virtual, simulações científicas que usam modelagem 3D, assistência médica e, principalmente, desenvolvimento de jogos (BENSTEAD, 2009).

Assim como a OpenGL, o DirectX também é uma API porém, mais completa. DirectX é um conjunto de APIs utilizadas para desenvolvimento de jogos 2D ou 3D, vídeo ou áudio. O programa foi criado pela Microsoft, por isso o pacote é produzido apenas para o sistema operacional Windows. O DirectX se divide em três componentes principais (TECMUNDO, 2012).

- Direct3D - Possibilita a animação tridimensional no monitor do computador.
- DirectDraw - Ajuda a produzir efeitos visuais bidimensionais (2D).
- DirectSound - Melhora o desempenho dos efeitos de áudio no computador e possibilita muitos efeitos sutis na mixagem e reprodução de áudio.

O Direct 3D é a API mais usada para o desenvolvimento de jogos por ser constantemente atualizada, além de fazer parte do pacote DirectX, que oferece serviços para conexões em rede com o DirectPlay, sons no formato MIDI utilizado em músicas, principalmente de jogos, com o DirectMusic e controla todos os dispositivos de entrada, tais como teclado, mouse e *joystick* com o DirectInput (ALMEIDA, 2004).

2.3.3 Ferramentas/ Linguagens de desenvolvimento

A programação do jogo segue o desenvolvimento desde o princípio, ainda que o jogo não possua a arte gráfica definida ou até mesmo a história completa. Atualmente, a linguagem de programação mais empregada para implementação de jogos eletrônicos é o C/C++, porém, um jogo pode ser escrito, praticamente, por qualquer linguagem de programação. Os jogos também podem ser escritos com linguagens de *script*. Comumente, estas linguagens são utilizadas em partes do jogo em que não é exigido maior desempenho, normalmente, as linguagens de programação trabalham em conjunto com uma API gráfica, como o OpenGL e o DirectX (JUNIOR et al., 2002).

A linguagem de programação C++, sucessora da linguagem C, possui características peculiares para o desenvolvimento de jogos, como suportar objetos, assim, admitindo que um problema seja decomposto com facilidade em subgrupos. Além disso, é simples de ser entendida e por ser uma linguagem próxima da linguagem de máquina, baixo nível, possui um bom desempenho ao rodar os jogos (BATES, 2003).

Java é uma linguagem de programação que atualmente está sendo utilizada mais frequentemente nos jogos, sobretudo para dispositivos portáteis. Ela também é orientada a objetos e é considerada mais próxima da linguagem humana do que C++, proporcionando assim, uma compreensão mais fácil do que esta. Além disso, Java é multi-plataforma, ou seja, pode ser executada em qualquer sistema operacional (PEREIRA, 2006).

Linguagens de Script geralmente são linguagens interpretadas que trabalham embutidas em um programa codificado em alguma linguagem compilada, assim, o programa que hospeda os *scripts* é o que vai admitir quais processos estes poderão

acessar (STEIN, 2009).

Um exemplo de linguagem *script* é Lua, completamente brasileira, que é poderosa, leve. Além disso, oferece a facilidade de programar os comportamentos de um objeto a partir de sua descrição (LUA, 2012). Outro exemplar de linguagem *script* é Python que é interativa, orientada a objetos, portátil, livre e, atualmente, facilmente usada por processadores multi core (PYTHON, 2012).

Macromedia Flash é a ferramenta mais usada na Web por aceitar o desenvolvimento animações por vetores. Esta popularidade se deve a inúmeras vantagens que o Flash oferece como a plataforma ser independente, possuir acesso simples e livre ao *plug-in*, cria arquivos leves com gráfico de boa qualidade e possuir um enorme número de usuários (ADOBE, 2012).

2.4 MOTORES PARA DESENVOLVIMENTO DE JOGOS

Existem *softwares* que facilitam as diversas etapas da criação de jogos digitais, estes programas são chamados *game engine* (motor de jogo). Os motores são bibliotecas de desenvolvimento, que dão suporte à linguagem e à API gráfica, na maioria das vezes tem a obrigação de organizar a estrutura do jogo, gerenciar imagens, processar entrada de dados e outras funções (GOMES & PAMPLONA, 2005). Os primeiros motores de jogos ficaram famosos com jogos como Doom e Quake que utilizaram programas com ferramentas que facilitaram e deram suporte ao processo de criação destes jogos. (REIS, 2009).

O motor é responsável pelo funcionamento interno do jogo, retirando esta complexidade de quem está desenvolvendo o *game*, proporcionando recursos que sejam reutilizáveis para a maior quantidade de jogos (REIS, 2009). O procedimento de desenvolvimento de jogos é feito agilmente quando um mesmo *game engine* é usado para criar diversos jogos. (WARD, 2012). Atualmente, os motores têm grande importância no desenvolvimento de jogos, ajudando a controlar o arcabouço e o ciclo de vida dos *games* (GOMES & PAMPLONA, 2005). A estrutura de um típico motor de jogo pode ser vista na Figura 3.



Figura 3. Estrutura geral de um motor de jogo.

Fonte: (GOMES & PAMPLONA, 2005).

2.4.1 Características essenciais

Pode-se dizer que uma *engine* é constituída por diversos sub motores e cada um é responsável por gerenciar uma função específica do jogo (ALMEIDA, 2009). Motores de jogo são qualificados por fornecer algumas características essenciais, tais como: motor gráfico e animações, motor físico, módulo de rede, sons e suporte a inteligência artificial (WARD, 2012).

O Motor gráfico é responsável pelo carregamento e renderização de imagens. Possui o encargo de desenhar os elementos gráficos do jogo formados por objetos em 2D e 3D, sem ou com movimento (PEREIRA, 2006) (REIS, 2009).

O Motor físico é de grande importância para aumentar a realidade dos jogos eletrônicos, pois, ele é encarregado de simular colisões em objetos, corpos que possuem movimentos, rigidez, aceleração, impulsão, resistência, gravidade (BITTENCOURT & OSÓRIO, 2006) (REIS, 2009).

Motor de som é o encarregado pelo carregamento, renderização e reprodução das músicas e efeitos especiais (SILVA & MACEDO, 2009). Geralmente, os arquivos de áudio são manipulados por uma API própria, como, por exemplo, o DirectSound (BITTENCOURT & OSÓRIO, 2006).

O Motor de lógica ou motor de Inteligência Artificial é responsável pelo

elemento lógico do *game*, quais as regras, os comportamentos das entidades virtuais, de modo específico, ele codifica os algoritmos de IA que serão utilizadas ao *game* (PEREIRA, 2006).

O motor de Rede deve fornecer a comunicação entre máquinas que compartilham uma simulação ao mesmo tempo, ou seja, oferece ao usuário a opção de desenvolver um jogo *multi-player* (múltiplos jogadores) em rede (REIS, 2009).

2.4.2 Características importantes

Além das funções essenciais que um motor de jogo deve possuir, há outras características que podem ser oferecidas por uma *engine*, como publicação que deve gerar o código fonte final do *game* baseado com a plataforma desejada e, possivelmente, oferecer a publicação do jogo na internet; a GUI que possui a função de gerar e manusear ações dos menus do *game*; manipulador de eventos que tem a função de controlar os acontecimentos durante o jogo; linguagem *script* e compilador embutido (PEREIRA, 2006) (REIS, 2009).

Outras características importantes que um motor de jogo poderia oferecer são o gerenciador de dados do jogo que possui a responsabilidade de armazenar todas as informações contidas no jogo; o mecanismo de ajuda que gera a documentação do jogo e o sistema de ajuda do jogo; a configuração do sistema que é encarregada de disponibilizar alternativas para que o desenvolvedor configure o *game* baseado no equipamento físico do computador; gerenciamento da memória; a ferramenta de reconhecimento de dispositivos que tem o papel de monitorar e captar dados dos equipamentos utilizados pelo usuário para interagir com a aplicação. (PEREIRA, 2006) (REIS, 2009).

2.4.3 Motores de jogos para PC

Existem diferentes motores de jogos para computadores, pode-se citar como exemplos o Game Maker que tem como foco a criação de jogos de 2 dimensões, o RPG *Maker* que tem como principal objetivo o desenvolvimento de jogos RPG, o DarkBasic que possui técnicas para um desenvolvimento acelerado de jogos de 3 dimensões e motores como o OGRE3D4, Id Tech 4, CryEngine3, Unreal Engine 3,

CrystalSpace, Torque, TrueVision 3D para desenvolver jogos profissionais de 3 dimensões (BITTENCOURT & OSÓRIO, 2006).

A família de motor de jogos Id Tech foi projetada pela id Software e é bastante famosa por causa dos motores anteriores, tais como os de Doom e Quake. Este conjunto de motores também fez sucesso com outras séries de jogos, como: Hexen, Heretic, Soldier of Fortune, Medal of Honor, Star Wars Jedi Knight . Alguns jogos usando o atual Id Tech 5 é o Rage e Doom 4. O id Tech 5 fornece recursos de iluminação, sombras, texturização, malhas, rede, IA, física, músicas, editores e *scripting* (IDSOFTWARE, 2012) (DEVMMASTER, 2012).

DarkBasic Pro é um motor proprietário desenvolvida pelo TheGameCreators para o sistema operacional Windows. O DarkBasic Pro é uma ferramenta de desenvolvimento rápido de jogos e pode ser utilizada para desenvolver diversos gêneros de jogos, por exemplo, Room War um jogo de ação, Axes Of Evil do estilo aventura, Blasteroids do tipo arcade, são jogos feitos por este motor de jogo. Esta *engine* oferece diversos recursos para o desenvolvimento de *games*, dentre eles: manipulação de objetos em formatos 3D, visualizações de 2 e 3 dimensões, interação via dispositivos de entrada comuns em jogos, funções de áudio e de rede, simuladores de física e de inteligência Artificial, editores de mapas, modeladores de objetos e bibliotecas prontas de modelos 3D (THEGAMECREATORS, 2012).

Um motor interessante a se apresentado é o Game Maker (GM) que foi utilizado no estudo de caso deste trabalho por oferecer suporte para desenvolvedores iniciantes. O GM é um motor de jogo que permite ao usuário desenvolver *games* com princípios básicos de programação. Com ele pode ser criado jogos, animações, músicas e efeitos sonoros (YOYOGAMES, 2012). É uma *engine* completa por possuir todos os sub motores padrões e um grande número recursos. O Game Maker possui o recurso *drag-and-drop* (arrastar e soltar), que converte os comandos em ícones a sua própria linguagem de programação embutida. A Linguagem Game Maker (GML) proporciona ao usuário opções mais avançadas e flexíveis para o desenvolvimento dos *games* do que a interface *drag-and-drop*.

Genesis3D é um motor para desenvolvimento de jogos com renderização para todas os programas em tempo real 3D. Esta *game engine* é totalmente gratuita para aplicações não comerciais. Alguns jogos de sucesso mediano foram

desenvolvidos usando Genesis 3D como o de aventura Dragon's Lair 3D: Return to the Lair e o militar de primeira pessoa Special Force (Hezbollah). Para desenvolver algum *software* com Genesis3D é necessário que o usuário tenha um compilador de C++, um pacote de modelagem 3D para fazer personagens e um programa Level Editor para a construção das fases ou cenários do jogo (GENESIS3D, 2012).

Um motor de jogo reconhecido mundialmente é o Unreal Engine 3, uma *game engine* completa para o desenvolvimento de jogos profissionais, permitindo visualizações avançadas e simulações em 3D. Unreal Engine 3 tem sido usado desde desenvolvedores de jogo a estúdios de televisão. Este motor possui motor de física, gráfico e animação, de rede, de inteligência artificial, de áudio e possui ferramentas de suporte a iluminação, sombras, texturização, malhas, terreno, gerenciamento de cenas e interface para programadores através de editores, bibliotecas e da linguagem Unrealscript (DEVMASTER, 2012). O motor Unreal tem uma lista de jogos famosos em seus projetos, tais como: Unreal Tournament, Deus Ex, Gears of War, Batman: Arkham Asylum, as séries dos jogos Rainbow Six e BioShock, entre outros (UNREALENGINE, 2012).

CryENGINE3 é um motor proprietário da Crytek que dá suporte ao desenvolvimento de jogos para computadores pessoais, Xbox 360 e PlayStation. A CryENGINE é bastante famosa pelo desenvolvimento de jogos de sucesso, como: Warface, Ryse e as séries de jogos Crysis e Far Cry. O CryENGINE3 proporciona suporte a iluminação, sombras, texturização, malhas, gerenciamento de cenas, edição de terrenos, física, IA, músicas, scripting e dispositivos de entradas (CRYTEK, 2012) (DEVMASTER, 2012).

3. ESTUDO DE CASO

O estudo de caso deste projeto foi o desenvolvimento do jogo *God Of Arms*, um *game* de aventura/ação do tipo plataforma em duas dimensões. O processo de desenvolvimento se deu através da aplicação dos fundamentos teóricos adquiridos, além do uso de um motor de jogos para auxiliar na fase de implementação do jogo. Apenas a primeira fase do jogo foi implementada. A seguir segue um resumo do Game Design do jogo.

3.1 FLUXO DO JOGO

O jogador irá percorrer cenários ao redor do mundo do jogo através do mapa principal que contém 23 fases. Os níveis de ação são constituídos por solo e plataformas que o jogador deve alcançar e para isso ele deverá passar por inimigos e armadilhas. O jogador achará itens, como armas, primeiro socorros e vidas que o ajudaram a passar estes obstáculos. O jogador poderá atirar em seus inimigos para atacá-los, além disso, para desviar dos ataques e dos perigos do mapa ele poderá pular ou abaixar, além de se mover para a direita e para a esquerda. Em cada nível do jogo, o jogador deverá buscar pelas Orbitais Titans que o ajudam a entrar na fortaleza do chefe daquela área. Godzilla é o chefe da primeira fase, ele possui Inteligência Artificial baseada em uma máquina de estados finitos. O mapa principal que contém as fases pode ser visualizado na figura 4.



Figura 4. Mapa principal de *God Of Arms*.

3.2 ELEMENTOS DO GAMEPLAY

Como elementos de apoio e medidores/contadores são utilizados Contador de Vidas, Contador de Pontos, Contador de Balas, Contador de Granadas, Contador de Saúde e Contador de Saúde do chefe final. A barra de contadores do jogo God Of Arms pode ser visualizada na figura 5.



Figura 5. Barra de contadores de *God Of Arms*.

Os itens utilizados no jogo são Vida Extra, Primeiros Socorros (pequeno), Primeiros Socorros (grande), Armas e Orbitais Titans. As armas utilizadas pelo jogador serão as Pistola, BioGun, Minigun, Rifle, Granada de Gás Letal, Lightgun, Granada Explosiva e Bazuca. Um exemplo de item que pode ser encontrado em God Of Arms pode ser visto na figura 6.



Figura 6. Minigun.

3.3 INIMIGOS E ARMADILHAS

Existirá vários inimigos no jogo, dentre eles, podem ser citados, como exemplos, Alien, Harpie, Phantom, Mech Dragon, Scorpion X, Andros, Diamond Head, SkyWorm, Wendigo, Demon e StinkFly. Além dos inimigos, haverá armadilhas pelas fases do jogo, como a serra, espinhos, machado duplo, lança, lava etc. Algumas armadilhas que estão presentes em God Of Arms podem ser vistas na figura 7.

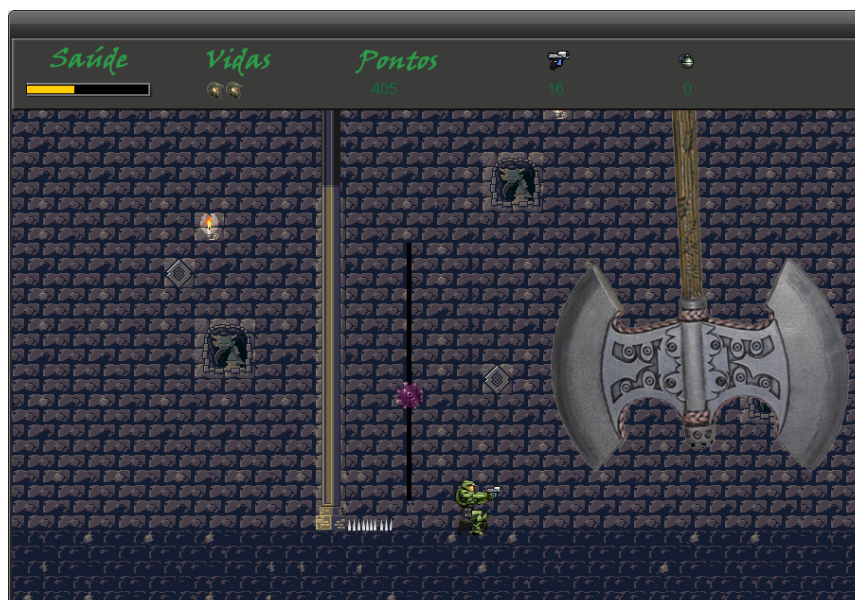


Figura 7. Armadilhas: Machado duplo, serra e espinhos.

O chefe da primeira fase é Godzilla, um robô criado com nome e aparência do monstro Godzilla, ele possui uma grande armadura metálica que tem uma enorme resistência aos danos de qualquer arma, além de ter uma forte defesa, ele também possui grande poder de fogo. Todos os ataques dele são capazes de matar vários alvos apenas com um tiro. Além de todas estas características, este inimigo também tem a característica de recuperar a saúde, permitindo-o durar mais tempo no campo de batalha. Os ataques de Godzilla sempre são à distância para preservar a sua vida, dentre eles podem ser destacados *Light Beam*, *Missile*, *Energy Beam* que possuem danos extraordinários. Godzilla pode ser visto na figura 8.

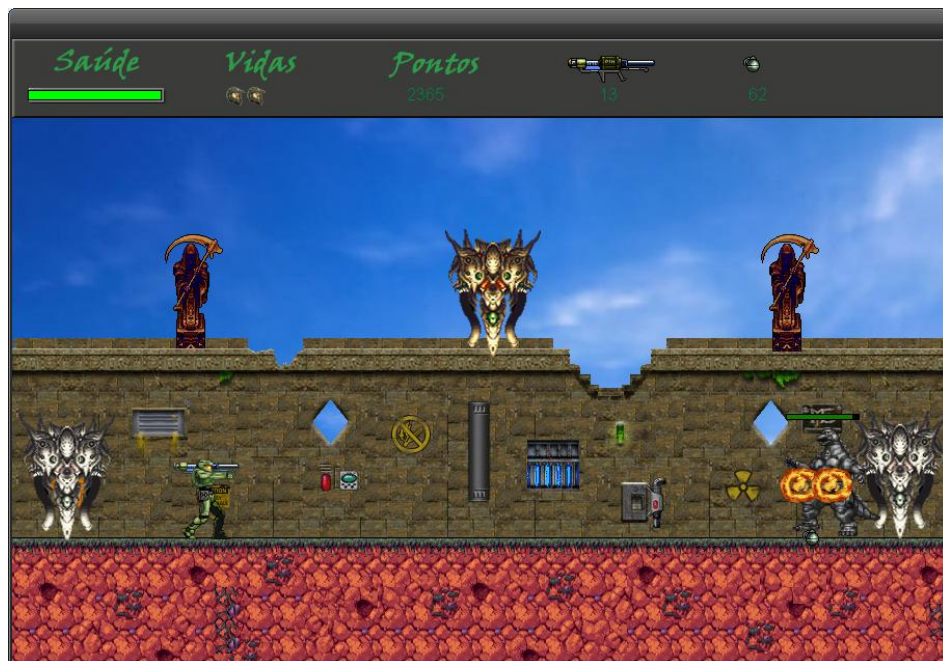


Figura 8. Godzilla, chefe da primeira fase.

3.4 TELAS DE APOIO E ELEMENTOS DE INTERAÇÃO OFF-GAME

As telas do jogo são: Tela do Menu Principal, Tela de Perfis, Tela de Saves, Tela de Pausa do Mapa Principal, Tela de Pausa, Tela de Ajuda e Tela de Game Over. Toda navegação de uma tela para outra depende de uma ação específica do usuário, por exemplo, se o jogador quiser ir da Tela do Menu Principal para a Tela de Ajuda, então, ele deverá clicar no botão do menu da Tela Principal que irá redirecionar o jogo para a Tela de Ajuda. O diagrama de navegação do jogo pode ser visto na figura 9.

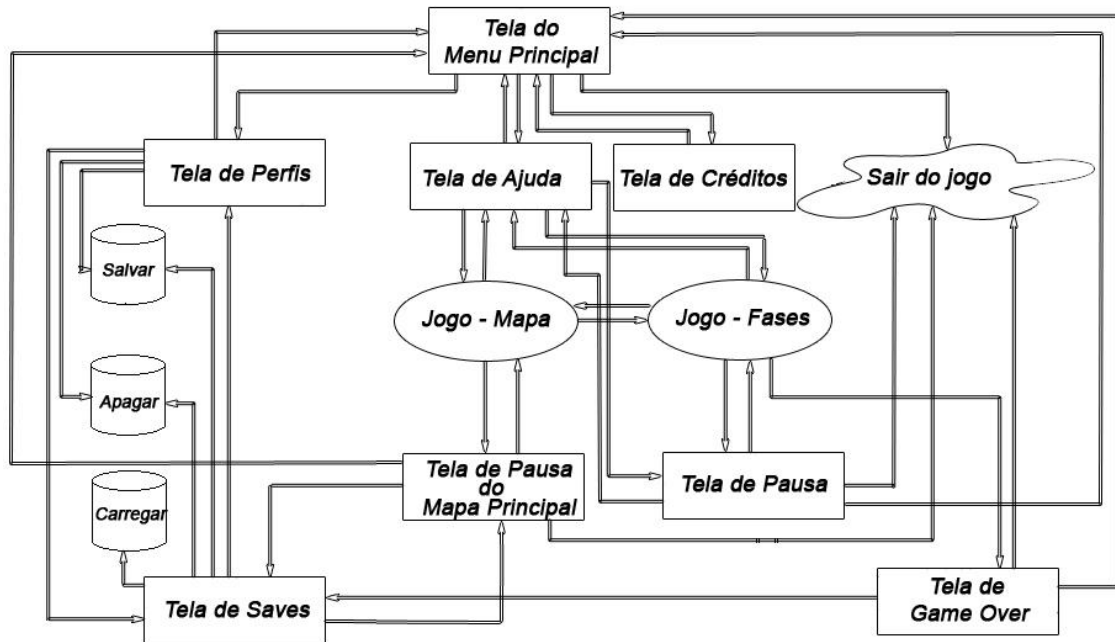


Figura 9. Diagrama de navegação de *God Of Arms*.

Cada uma destas telas possuirá o seu próprio menu. Elas utilizarão um esquema padrão de fonte e botões. As opções serão representadas por botões que ao entrar em contato com o mouse ficarão com as arestas posterizadas. Se o mouse sair do contato com o botão, este voltará às características padrões. Perfis e Saves serão representados por retângulos com dados do jogo, quando selecionados ficaram com a imagem padrão misturada com azul claro. Um exemplo de tela e menu pode ser visualizado na figura 10.



Figura 10. Menu Pausa do Mapa Principal.

3.5 HISTÓRIA

O jogador se encontrará dentro de uma história fictícia no ano de 2053 em que a Terra vivia em uma época de caos, a falta de recursos naturais, principalmente a água, tornou-se a fonte de busca de novos conhecimentos e experimentos. Era uma disputa pelo poder mundial. Esta disputa ocasionou muitas guerras. Para complicar ainda mais a situação, mutantes e robôs com Inteligência Artificial gerados pelas novas descobertas rebelaram-se juntos contra a raça humana, dominaram o mundo e construíram bases em todo o planeta, eles se denominavam *Titans*. Graças ao seu treinamento para a guerra, o personagem *God Of Arms* foi um dos poucos da raça humana que sobreviveu a guerra e talvez o único com habilidade suficiente para tentar um último ataque contra estes poderosos inimigos.

3.6 FASES DO JOGO

A primeira fase do jogo é realizada em uma área dominada por Godzilla, um robô que possui uma avançada técnica de Inteligência Artificial. Depois dos poderosos Titans dominarem o planeta, eles construíram bases em todos os territórios. Godzilla ordenou aos monstros com menor ataque, como Harpies e Phantoms, que fossem para a primeira linha de defesa localizada na floresta em frente ao castelo de Godzilla. Para auxiliar estes monstros apenas com ataques corporais foram enviados os Skyworms para a realização de ataques aéreos. São usados para a defesa de dentro do castelo monstros como Mech Dragon, Stinkfly, Elemental Fire, Nergal, Draconis e Stubborn que possuem uma maior resistência aos danos, além de ataques mais fortes. Godzilla se encontra em um santuário em um pequeno deserto após o castelo, já que esta fortaleza seria alvo para possíveis ataques. Após o castelo, o jogador irá se deparar com monstros de nível superior, com saúde e danos maiores, o Demon e o Lizard serão os inimigos mais fortes a serem combatidos antes de enfrentar o tão temível Godzilla. Por toda a fase são encontradas diversas armadilhas, desde pequenas lâminas cortantes até machados duplos.

4. DESENVOLVIMENTO E IMPLEMENTAÇÃO

Este capítulo apresenta o processo de desenvolvimento do estudo de caso deste trabalho considerando o referencial teórico apresentado no mesmo.

4.1 DESENVOLVIMENTO DE *GOD OF ARMS*

O jogo *God Of Arms*³ é do tipo aventura/ação, pois ele faz o jogador controlar o herói em busca dos seus objetivos, focados na história, enfrentando inimigos e ultrapassando obstáculos que estimulam o raciocínio lógico. Além disso, ele exige do jogador reação e coordenação entre a visão e as mãos em situações de pressão, assim, fazendo com que o jogador tenha a sobrevivência como principal objetivo.

God Of Arms foi desenvolvido para a plataforma de computador, o sistema operacional escolhido foi o Windows. Apesar de haver fatores negativos com esta plataforma em relação aos jogos, como a pirataria e o alto custo, os *games* de computadores são os que mais evoluem e possuem a maior quantidade de usuários por causa do suporte oferecido por esta plataforma.

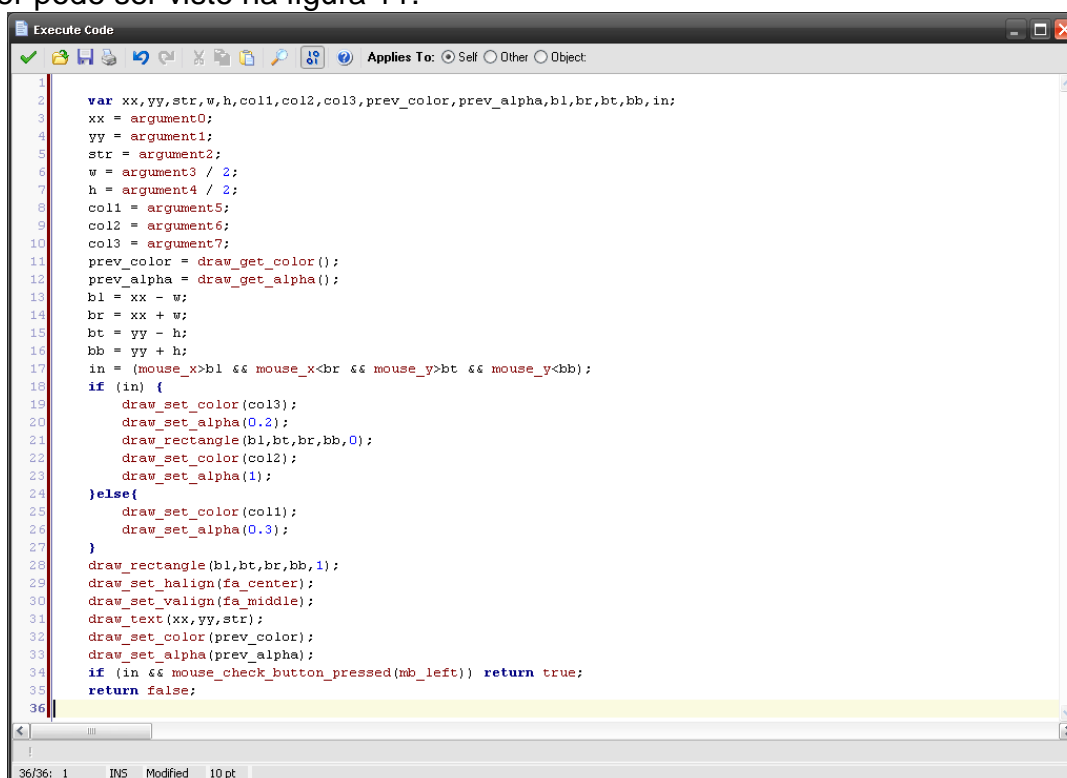
O primeiro passo do processo de desenvolvimento foi o planejamento e execução do Game Design que relaciona todos os aspectos de desenvolvimento de jogos da melhor maneira possível. O Game Design de *God Of Arms* está disponível no apêndice A.

Por ser um jogo de caráter acadêmico e não comercial foram utilizadas *sprites* da internet para o elemento gráfico 2D, além de músicas e efeitos sonoros já existentes para a parte sonora. A Inteligência Artificial do jogo foi programada baseada em uma máquina de estados finitos⁴, IA para controlar alguns agentes *non-player character* (NPC), personagens controlados pelo computador. O jogo foi projetado para ter uma boa interação com o usuário, possuindo telas e menus padronizados, teclas de atalhos para praticidade, utilização de sons que caracterizam eventos e balanceamento do jogo que permite o jogador ter a sensação de conduzir o jogo. Foram inseridos no GDD todos os elementos do jogo e a partir deste documento foi iniciado o desenvolvimento do jogo.

³Disponível em: https://rapidshare.com/files/963888905/God_Of_Arms.exe

⁴Disponível em: https://rapidshare.com/files/2957695034/Estados_Chefe3.jpg.png

Em relação às plataformas gráficas e sonoras, para executar o jogo *God Of Arms*, é necessário ter a API gráfica DirectX versão 8.0 (ou superior) compatível tanto com a placa gráfica quanto com a placa de som. A respeito da linguagem de implementação⁵ do jogo, foi utilizada apenas a linguagem de *script* embutida no motor de jogos utilizado. Um exemplo de código da linguagem de *script* do Game Maker pode ser visto na figura 11.



```

1  var xx,yy,str,w,h,col1,col2,col3,prev_color,prev_alpha,b1,br,bb,in;
2  xx = argument0;
3  yy = argument1;
4  str = argument2;
5  w = argument3 / 2;
6  h = argument4 / 2;
7  col1 = argument5;
8  col2 = argument6;
9  col3 = argument7;
10 prev_color = draw_get_color();
11 prev_alpha = draw_get_alpha();
12 b1 = xx - w;
13 br = xx + w;
14 bt = yy - h;
15 bb = yy + h;
16 in = (mouse_x>b1 && mouse_x<br && mouse_y>bt && mouse_y<bb);
17 if (in) {
18     draw_set_color(col3);
19     draw_set_alpha(0.2);
20     draw_rectangle(b1,bt,br,bb,0);
21     draw_set_color(col2);
22     draw_set_alpha(1);
23 }else{
24     draw_set_color(col1);
25     draw_set_alpha(0.3);
26 }
27 draw_rectangle(b1,bt,br,bb,1);
28 draw_set_halign(fa_center);
29 draw_set_valign(fa_middle);
30 draw_text(xx,yy,str);
31 draw_set_color(prev_color);
32 draw_set_alpha(prev_alpha);
33 if (in && mouse_check_button_pressed(mb_left)) return true;
34 return false;
35
36

```

Figura 11. Linguagem de *script* GML

O motor de jogos utilizado foi o Game Maker 8.0 Lite, pois este motor de jogo permite o usuário desenvolver jogos a partir de princípios básicos de programação. Na essência, foram utilizados do GM os sub motores de física, animação, colisão, músicas e efeitos sonoros. Outras características do Game Maker usadas para o auxílio da estrutura do jogo foi a publicação, a GUI, o manipulador de eventos, a linguagem de *script* e o compilador embutido, o gerenciador de dados, o mecanismo de ajuda e a ferramenta de reconhecimento de dispositivos. A seguir é apresentada a avaliação do motor de jogos Game Maker.

Foram implementados no jogo de estudo de caso todas as ações que o jogador pode executar e eventos que irão ocorrer durante o jogo. As ações dos botões dos menus *off-game*, ações do jogador, ações dos inimigos, ações das

⁵Disponível em: https://rapidshare.com/files/985885346/Codificação_God_Of_Arms.rtf

armadilhas, ações de eventos fixos ou aleatórios. Além da implementação das ações também foi construído as telas e menus *off-game*, tela de mapa principal e a primeira fase do jogo.

4.2 AVALIAÇÃO DO GAME MAKER

A avaliação de um motor de jogo é baseada nas próprias funcionalidades dele. Foram analisados os sub motores e outras funcionalidades do motor que foram utilizadas durante o desenvolvimento do estudo de caso e, então, foram definidos os níveis básico, médio e avançado para ponderar cada função do motor.

4.2.1 Características essenciais

O Motor Gráfico do GM oferece suporte gráfico de nível médio. Por ser um programa destinado ao desenvolvimento de jogos de 2 dimensões é um motor excepcional para a criação e edição de imagens em 2D, porém possui um rendimento muito baixo em relação aos jogos com gráficos de três dimensões. Diversos formatos de imagens são aceitos pelo Game Maker, por exemplo, JPG, PNG e GIF. Junto ao editor de imagens, o GM também proporciona animação aos *sprites*. Além disso, esta *engine* possui um editor de salas (telas e cenários).

O Game Maker possui um sub motor físico com nível avançado, pois possui as principais funções para deixar o jogo mais realístico em relação à física, como simulação de colisões em objetos, corpos que possuem movimentos, rigidez, aceleração, impulsão, gravidade. Todas estas funções do motor físico foram utilizadas no *God Of Arms* sem nenhum problema.

O Game Maker tem um suporte de conexões de rede avançado. O GM oferece quatro diferentes tipos de conexões: IPX, TCP/IP, modem e Serial. A conexão IPX é usada para jogar em rede local. TCP / IP é o protocolo para jogar pela internet, para utilizá-lo é necessário os jogadores saberem entre si os endereços IP. A conexão do modem é feita através do modem por meio de configurações como uma sequência de inicialização e um número de telefone. O cabo serial que é a conexão direta entre os computadores.

O GM possui um motor de IA básico. A única base ao Game AI neste motor é

a função de um objeto seguir outro pela posição (com a opção de desviar de objetos sólidos). Se o desenvolvedor quiser utilizar alguma instância com IA em seu jogo, ele terá que implementar todo o código sem o auxílio do motor.

O sistema de áudio do GM é de um nível avançado, pois este motor suporta a maioria dos formatos de áudio, apesar de ser recomendado o uso dos formatos *wave* para efeitos sonoros e *midi* para a música de fundo. Há quatro tipos diferentes de som no GM: sons normais utilizadas para efeitos sonoros, música de fundo, sons 3D que tem uma posição no espaço (são ouvidos de acordo a distância entre o som e personagem principal) e os sons que devem ser tocadas através do *media player*.

4.2.2 Características importantes

A função de publicação do GM possui um nível avançado, pois gera para o usuário o executável (*stand-alone*) do jogo e oferece a publicação do jogo na internet através do site da desenvolvedora do motor.

A interface ao usuário do Game Maker tem um nível avançado, uma vez que a GUI utilizada é simples, o desenvolvedor não tem problemas em achar as funções da *engine*. A geração de menus é igual ao desenvolvimento dos cenários do jogo, os botões são instâncias e as telas são as salas. O desenvolvedor pode criar jogos a partir de dois comandos que são as interfaces *drag-and-drop* e GML. A interface *drag-and-drop* pode ser vista na figura 12.

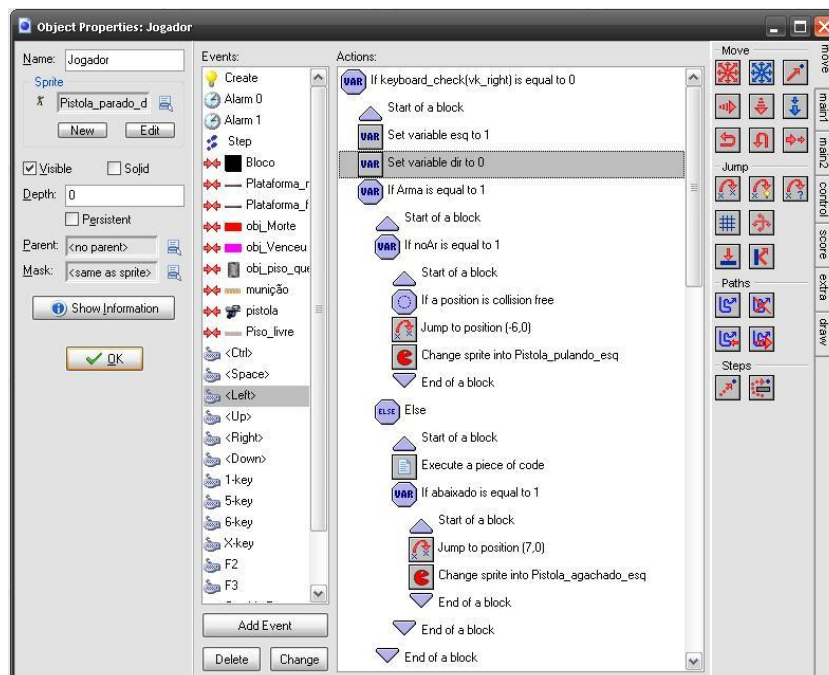


Figura 12. *Drag-and-drop*, interface do Game Maker.

O manipulador de eventos do Game Maker é qualificado como nível avançado, pois pode ser dito que o GM é orientado a evento, ou seja, sempre que algo acontece no jogo as instâncias dos objetos recebem mensagens que dizem que algo aconteceu (eventos). O Desenvolvedor pode de maneira simples determinar para quais eventos as instâncias reagem e quais ações eles irão executar quando o evento ocorrer.

A linguagem de script e o compilador embutido são considerados como nível avançado. GML é orientada a objetos, derivada de C++, e há duas maneiras de usá-la, a primeira por meio de *scripts* (pedaços de código que podem ser reutilizados) e a segunda através da adição de um código de ação para algum evento. A codificação entre os dois modos é a mesma e o que diferencia as duas maneiras é que o código de ações não tem um nome e não pode usar argumentos, além de ser exclusivo para um determinado objeto.

O gerenciador de dados do jogo é de nível básico. O GM oferece suporte ao gerenciamento de dados em um jogo, porém depende do desenvolvedor. A função de salvar o jogo feita pelo GM faz com que todas as informações atuais do jogo sejam gravadas e isso pode gerar dificuldades ao desenvolvedor, já que em jogos somente as informações relativas ao jogador atual deveriam ser salvas. Então, se o desenvolvedor quiser armazenar uma pequena quantidade de informação, ele deve usar um arquivo externo, como por exemplo, o registro do Windows.

O mecanismo de ajuda é caracterizado com nível médio. O sistema de ajuda do GM é limitado ao gerador de informações das instâncias do jogo (nome, eventos, códigos etc.).

A configuração do sistema é qualificada como de nível médio. O Game Maker disponibiliza algumas alternativas básicas para configurar o jogo compatível ao equipamento físico do computador, por exemplo, a resolução, as cores e interpolar cores entre *pixels* e frequência de atualização.

O reconhecimento de dispositivos do GM é classificado como de nível médio. Para usar esta função devem ser colocadas ações em eventos de mouse, teclado e *joysticks*.

4.3 TESTES DO JOGO

Os testes de funcionalidade, atratividade e a jogabilidade do jogo foram feitos por um grupo de usuários escolhidos aleatoriamente. No total, 14 pessoas jogaram *God Of Arms*, dentre elas 3 mulheres e 11 homens. A média de idade dos indivíduos foi de 20,8 anos, eles possuem, aproximadamente, 10 anos como jogadores e uma média de 7,5 *games* jogados por ano.

No geral, todos os usuários disseram gostar do jogo. Foi notado que os indivíduos que são jogadores há mais tempo ou que jogavam uma quantidade maior de jogos por ano não tiveram problemas com a jogabilidade, porém usuários que não tinham estas características sentiram um pouco de dificuldade até se habituarem aos controles. Todos os jogadores sentiram-se desafiados pelo jogo, principalmente dentro do castelo onde existem diversas armadilhas.

Após a análise do questionário em relação aos elementos do *game* e ele próprio, pode-se visualizar através do gráfico da figura 13 o resultado de cada elemento do jogo, como Interface (Gráfico), Jogabilidade (Controles), Áudio, Interação, e por fim, o jogo em geral. O questionário de avaliação de *God Of Arms* está disponível no apêndice B.

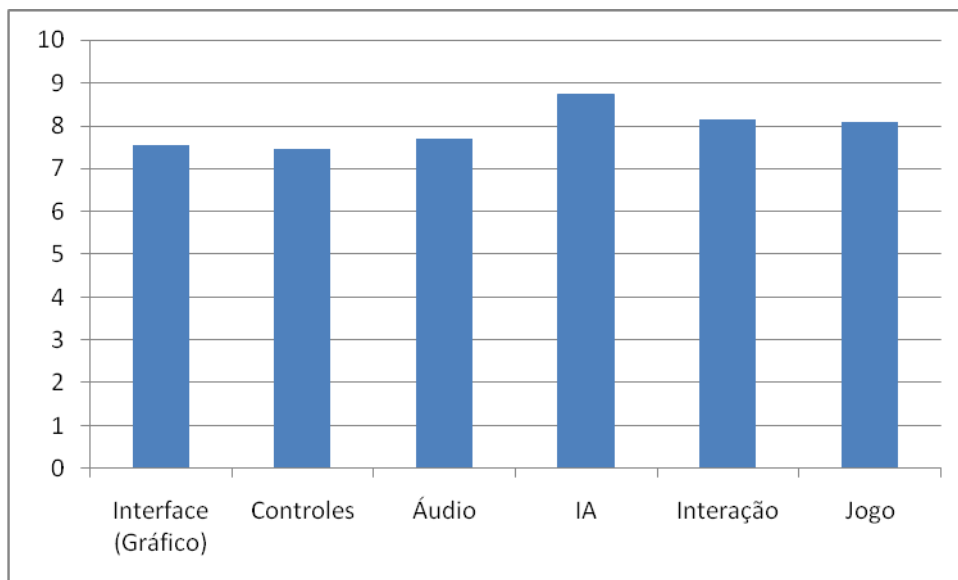


Figura 13. Gráfico da avaliação de God Of Arms.

Com a análise da avaliação do questionário relativo aos elementos do jogo foi notado que por não ser um jogo com gráficos modernos, como os jogos 3D, a interface gráfica não obteve uma nota mais alta. O elemento de jogabilidade teve a

nota mais baixa pelo fato de haver muitos jogadores com pouca experiência. O áudio teve uma nota média, pois o elemento de áudio foi escolhido conexo ao tema do jogo que é de guerra e, por isso, nem todos os jogadores gostaram do estilo de áudio do jogo. O item de IA recebeu a maior nota, pois todos os jogadores acharam os agentes inteligentes bons adversários. O elemento interativo obteve uma boa nota, pois os jogadores acharam que os componentes sonoros e visuais que caracterizavam eventos que ocorreram ou que ainda iriam ocorrer ajudaram no *feedback* do jogo para com o jogador. Por fim, os jogadores avaliaram o jogo completo e demonstraram, com uma boa nota, que o jogo é aceitável e jogável.

5. CONCLUSÃO

O objetivo deste trabalho foi realizar um estudo sobre o processo de criação de jogos, assim como os componentes que fazem parte desde a concepção da idéia do jogo até a fase de desenvolvimento. Em concordância ao estudo realizado, pode-se verificar que ao desejar desenvolver um jogo, deve-se escolher primeiramente o tipo do jogo e a plataforma na qual este deverá ser executado.

Pode-se concluir que por ser um jogo, um *software* com características específicas, o Game Design é o primeiro passo mais adequado que oferece suporte a todas as fases de desenvolvimento. Além disso, pode-se afirmar que é necessário a todo *game designer* conhecer as partes de um jogo: arte gráfica, arte sonora, implementação, inteligência artificial e interatividade.

As tecnologias utilizadas são de grande importância na fase de desenvolvimento, pois garantem sistematização e agilidade ao processo. Dentre estes aparatos tecnológicos, devem ser destacados os motores de jogos que fornecem módulos reutilizáveis, podendo assim, serem aproveitados para o desenvolvimento de distintos jogos.

Através da análise do Game Maker, pode-se dizer que ele é um excelente motor de jogos para desenvolvedores iniciantes. Apesar do GM não oferecer suporte em alguns aspectos, como o gerenciamento de dados e a Inteligência Artificial, por outro lado, ele possui interface amigável, identificação de colisão e o manipulador de eventos. Estas funcionalidades são as que mais simplificam ao usuário o desenvolvimento do jogo nesse motor, assim, permitindo ao usuário que não tem muita experiência criar jogos simples com facilidade.

O desafio acadêmico deste trabalho foi utilizar fundamentos das disciplinas relacionadas ao desenvolvimento de jogos em um único trabalho e ao mesmo tempo atender aos requisitos exclusivos dos jogos.

Por fim, o desenvolvimento de jogos é um mercado lucrativo e, por isso, oferece grandes oportunidades. Contudo, para conseguir ao final do desenvolvimento de um jogo a possibilidade da aceitação dele no mercado, cada vez mais exigente, são exigidos métodos e ferramentas adequados para as etapas de projeto e de execução da criação do jogo.

5.1 TRABALHOS FUTUROS

Como sugestão de trabalhos futuros propõe-se a continuidade do desenvolvimento do projeto *God Of Arms*, pois foi desenvolvida somente a primeira fase do jogo. Seria interessante terminar o jogo de acordo com o projeto que consta nesse trabalho avaliando a flexibilidade da metodologia de Game Design.

Outro item que poderia ser realizado é a avaliação de outros motores de jogos para comparação com as características do Game Maker.

6. BIBLIOGRAFIA

- ADOBE, Flash Player. Disponível em <<http://www.adobe.com/br/products/flashplayer>>. Acesso em 20 de janeiro, 2012.
- AGNER, L. C. **Otimização do Diálogo Usuários – Organizações na World Wide Web: Estudo de Caso e Avaliação Ergonomica de Usabilidade de Interfaces Humano-Computador**. Tese de mestrado, PUC-Rio. 2002.
- ALMEIDA, G. R. DE. **Desenvolvimento de Motores de Jogos Casuais voltados para Usuário**. Trabalho de Conclusão de Curso. Ciência da Computação, Universidade Estadual de Londrina. 2009
- ALMEIDA, M. R. G. DE. **Programação de jogos 2D usando o mrdx e fundamentos de jogos 3D**. Trabalho de Conclusão de Curso. Ciência da Computação, Universidade Federal do Pará. 2004.
- ANDROID. Disponível em: <http://www.androidbrasilprojetos.org/projetos/game-design-document/>. Acesso em 23 de fevereiro, 2012.
- BATES, B. **Game developers market guide**. Thomson course technology. 1 edição. 2003.
- BENSTEAD, Luke. **Beginning OpenGL game programming**. 2 edição. Course technology, cengage learning. 2009.
- BETHKE, Erik. **Game development and production** (Wordware Game Developer's Library). 2003.
- BITTENCOURT, J. R.; OSÓRIO, F. S. **Motores para Criação de Jogos Digitais: Gráficos, Áudio, Interação, Rede, Inteligência Artificial e Física**. UNISINOS. 2006.
- CARVALHO, J. O. F. **O papel da interação humano-computador na inclusão digital**. In: Revista Transinformação, V.15, N.3. edição especial. Campinas. 2003.
- CRYTEK, CryEngine. Disponível em: <<http://www.crytek.com/cryengine>>. Acesso em 24 de fevereiro, 2012.
- DEVMMASTER, CryEngine. Disponível em: <<http://devmaster.net/devdb/engines/cryengine-3>>. Acesso em 22 de fevereiro, 2012.
- DEVMMASTER, Id-Tech. Disponível em: <<http://devmaster.net/devdb/engines/id-tech-4>>. Acesso em 24 de fevereiro, 2012.
- DEVMMASTER, Unreal Game Engine. Disponível em: <<http://devmaster.net/devdb/engines/unreal-development-kit-udk>>. Acesso em 23 de fevereiro, 2012.
- DOS SANTOS, N. dos S. R. S. et al. **Jogos Educacionais - ferramentas para o ensino de programação**. Seminário de Informática – RS. 2007.

- FELDMAN, G. **Game industry study predicts massive growth ahead**. 2004. Disponível em: <http://www.gamespot.com/news/2004/05/28/news_6099584.html>. Acesso em 20 de setembro, 2005.
- FUNGE, J. D. **Artificial Intelligence for computer games: an introduction**. Natick: ak peters. 2004.
- GAMEVICIO, Mercado brasileiro. Disponível em: <<http://www.gamevicio.com.br/i/noticias/79/79775-mercado-brasileiro-de-games-cresce-28-em-tres-anos-e-ganha-adesao-de-empresas/index.html>>. Acesso em 26 de agosto, 2011.
- GENESIS3D, Genesis3D Game Engine. Disponível em: <<http://www.genesis3d.com/>>. Acesso em 22 de fevereiro, 2012.
- GOMES, P. C. R.; PAMPLONA, V. F. **M3ge: um motor de jogos 3D para dispositivos móveis com suporte a mobile 3D graphics**. FURB/BCC. 2005.
- IDSOFTWARE, Id tech4. Disponível em: <<http://www.idsoftware.com>>. Acesso em 21 de fevereiro, 2012.
- ABRIL, Info. Disponível em: <<http://info.abril.com.br/noticias/blogs/info-games/sony/uncharted-3-ptbr/>>. Acesso em 11 de abril, 2012.
- JUNIOR, A. DE S. R.; NASSU, B. T.; JONACK, M. A. **Um estudo sobre os processos de desenvolvimento de jogos eletrônicos (games)**. Departamento de Informática. UFPR - Universidade Federal do Paraná. Curitiba. 2002.
- LOWOOD, H. **Playing History with Games: Steps towards Historical Archives of Computer Gaming**. Dissertação (Annual Meeting of the American Institute for Conservation of Historic and Artistic Works) — Stanford University. 2004.
- LUA. Disponível em:<<http://www.lua.org/press.html>>. Acesso em 18 de janeiro, 2012.
- LUZ, Mairlo Hideyoshi Guibo Carneiro Da. **Desenvolvimento de Jogos de Computadores**. Faculdade de Ciência da Computação. Dissertação (graduação). Universidade Federal de Itajubá. 2004.
- METODISTA. Mercado nacional de jogos eletrônicos. Disponível em: <<http://www.metodista.br/rronline/noticias/tecnologia/2011/mercado-nacional-de-jogos-eletronicos-movimenta-r-950-mi-ao-ano>>. Acesso em 26 de agosto, 2011.
- MINAYO, M. C. S. **O desafio do conhecimento: pesquisa qualitativa em saúde**. 10 edição. São Paulo: Hucitec. 2007.
- MORIMOTO, C. E. Disponível em: <<http://www.hardware.com.br/tutoriais/abc-placas-3d3/pagina4.html>>. 2010. Acesso em 15 de janeiro, 2012.
- OLIVEIRA, Yanko Gitahy. **Jogos Eletrônicos como Ferramenta de Ensino**. Universidade do Brasil. 2010.

- OUTERSPACE, Histórico dos games. Disponível em: <<http://outerspace.terra.com.br/retrospace/materias/consoles/historiadosconsoles1.htm>>. Acesso em 22 de Outubro, 2011.
- PEREIRA, Gean Alex. **Projeto E Desenvolvimento De Jogos Computacionais**. Trabalho de Conclusão de Curso. Faculdade de Ciência da Computação. Universidade do Estado de Santa Catarina – UDESC. 2006.
- PERUCIA, Alexandre; BALESTRIN, Alsones; VERSCHOORE, Jorge. **Coordenação das atividades produtivas na indústria brasileira de jogos eletrônicos: hierarquia, mercado ou aliança?**. UNISINOS. 2011.
- PORTALMULTIRIO. Disponível em: http://portalmultirio.rio.rj.gov.br/riosummit2004/noticias_full.asp?id_noticias=401&idioma=por&forum=>. Acesso em 15 de agosto, 2011.
- PYTHON. Disponível em: <<http://www.python.org.br/wiki>>. Acesso em 18 de janeiro, 2012.
- REIS, F. V. DOS. **Jogo da cabanagem: projeto e implementação**. Instituto de Tecnologia. Universidade Federal do Pará. Belém – Pará. 2009.
- ROLLINGS, A.; MORRIS, D. **Game architecture and design: a new edition**. United States of America: New Riders. 2004.
- ROUSE, R. **Game Design: Theory & Practice**. Edição 2. Wordware Publishing. 2005.
- RUSSELL, Stuart; NORVIG, Peter. **Inteligência Artificial**. Tradução: Vanderberg D. De Souza. Rio de Janeiro: Elsevier. 2004.
- SCHWAB, Brian. **AI Game Engine Programming**. Boston, MA: Course Technology. Cengage Learning. 2009.
- SILVA, A. S.; MACEDO, H. T. **Um motor para jogos cross-media**. Scientia plena. 2009.
- SILVA, Filipe Correa Lima da. **Tópicos em Ciência da Computação - Agentes Inteligentes em Jogos de Computador**. Faculdade de Informática. Dissertação (graduação) – Universidade de São Paulo. 2005.
- STEIN, Daniel Barbosa. **Implementação de Linguagens de Script Para Jogos**. 2009.
- TAVARES, M. T. de S. **JOGOS ELETRÔNICOS: EDUCAÇÃO E MÍDIA**. II Seminário Jogos eletrônicos, educação e comunicação: construindo novas trilhas. Universidade Federal do Estado do Rio de Janeiro – UNIRIO. 2006.
- TECHTUDO, Mercado de jogos. Disponível em: <<http://www.techtudo.com.br/jogos/noticia/2011/02/o-mercado-de-games-e-aplicativos-no-brasil.html>>. Acesso em 26 de agosto, 2011.

TECMUNDO, Directx. Disponível em: <<http://www.tecmundo.com.br/900-o-que-e-directx-.htm>>. Acesso em 8 de fevereiro, 2012.

THEGAMECREATORS, Dark Basic Pro. Disponível em: <http://www.thegamecreators.com/?m=view_product&id=2000&page=index>. Acesso em 22 de fevereiro, 2012.

UNREALENGINE. Disponível em: <<http://www.unrealengine.com/features>>. Acesso em 22 de janeiro, 2012.

WARD, JEFF. Disponível em <http://gamecareerguide.com/features/529/what_is_a_game_.php?>. Publicado em 2008. Acesso 26 de janeiro, 2012.

WEBCHEATS. Disponível em: <<http://www.webcheats.com.br/forum/wii-dicas-e-tutoriais/872709-mega-topico-nitenndo-wii.html>>. Acesso em 12 abril, 2011.

YOYOGAMES, Game Maker. Disponível em: <www.yoyogames.com/make>. Acesso em 21 de fevereiro, 2012.

APÊNDICE A - DOCUMENTO GAME DESIGN

God Of Arms

Game Design:

Mecânica do Jogo

***GamePlay* Central**

Jogo de plataforma 2D com visão lateral.

Tipo

Aventura.

Fluxo

O jogador irá controlar um especialista em armas, percorrendo um cenário onde se encontram vários inimigos e armadilhas. Ao decorrer do jogo, o jogador será testado cada vez mais em suas habilidades de atirar nos seus inimigos e ao mesmo tempo desviar de ataques dos mesmos. As armas ficarão mais fortes, porém, assim também serão os inimigos, além disso, o jogador contará com uma quantidade cada vez mais limitada de munição ao decorrer do jogo.

O jogador irá percorrer cenários ao redor do mundo, terá o mapa principal que vai conter várias fases. Os níveis de ação são constituídos por solo e plataformas que o jogador deve alcançar e para isso deverá transpor obstáculos e armadilhas. Estes, que podem ser transpostos pelo jogador, são definidos pelos itens que já ele já possui.

O jogador poderá atirar em seus inimigos para atacá-los e para desviar dos ataques e dos perigos do mapa o jogador poderá pular ou abaixar, além de, se mover para a direita e esquerda.

Em cada nível do jogo, o jogador deverá buscar pelas Orbitais Titans que ajudam o jogador a entrar na fortaleza do chefe daquela área.

Personagens

God Of Arms (jogador) – Um atirador com grande destreza na utilização de diversas armas controlado pelo jogador. Sua perícia da condição para uso de armas como revolver, rifle, espingarda, mini metralhadora, bazuca, raio laser, raio gama, raio de luz, granada explosiva e de granada de gás letal. Jogador ideal para enfrentar diversos inimigos. O Especialista poderá se mover para frente e para trás, pular e abaixar.

Godzilla (I.A) – É o chefe das primeiras fases, uma grande fera metálica, com grande poder de fogo, com o objetivo explícito de matar o jogador. Foi um robô criado por países da América do Sul para resistir a batalhas longas e com alvos distantes. Feito para atacar sempre com o máximo de energia disponível. Após a energia se acabar ele se carregará para executar um próximo ataque e se por acaso sofrer determinada quantidade de danos ele poderá recuperar sua saúde.

Elementos do *GamePlay*

Elementos de apoio e medidores / contadores

Contador de Vidas – Contabiliza o número de Vidas de God Of Arms, caso atinja zero o jogador sofrerá *Game Over* (Ao chegar a Game over o jogador poderá dar load no jogo salvo do mapa principal).

Contador de Pontos – Mostra os pontos que God Of Arms tem (No início o jogador irá ter 0 ponto e será acrescentado, relativamente, após matar algum inimigo).

Contador de Saúde – Mostra quanto de saúde God Of Arms tem (No início o jogador irá contar com 100 (máximo) de saúde e será reduzido, relativamente, ao receber um dano). Cada saúde ao máximo é representada por uma barra horizontal verde e ao perder saúde ela se tornará vermelha. Uma barra de saúde vazia significa uma vida a menos.

Contador de Balas – Cada arma terá seu próprio contador de balas, porém, só será visualizado o contador da arma que o jogador estiver usando. Ao chegar a zero, o jogador não poderá atirar com aquela arma até achar mais munição.

Contador de Granadas – conta o número de granadas que o jogador possui, assim como a arma, se o contador chegar a zero o jogador não poderá lançar granadas até achar mais.

Contador de Orbitais Titans – Contabiliza o número de Orbitais conseguidos pelo jogador, cada fortaleza exige um número mínimo de Orbitais para permitir a entrada.

Contador de Saúde dos inimigos – Apenas os chefes terão esse contador, ele simboliza a quantidade atual de saúde do chefe. No início o chefe irá começar com 100 (máximo). Quando o contador estiver totalmente vazio o chefe morre.

Contador de Energia dos inimigos – Apenas os chefes terão esse contador, ele simboliza a quantidade atual de energia do chefe. No início o chefe irá começar com 100 (máximo). Quando o contador estiver totalmente vazio o chefe não poderá atacar.

Itens

Vida Extra – Simbolizado pela cabeça de God Of Arms, quando tocada, acrescentará uma vida ao jogador.

Primeiros Socorros (pequeno) – Uma pequena maleta pequena de primeiros socorros que acrescenta mais 25 de saúde a God Of Arms.

Primeiros Socorros (grande) – Uma pequena maleta grande de primeiros socorros que acrescenta mais 50 de saúde a God Of Arms.

Armas – Com uma imagem de uma arma utilizada por God Of Arms, quando tocada, dá a possibilidade do jogador de usar aquela arma. Se a arma já puder ser utilizada pelo jogador, acrescenta um determinado número de munição para aquela arma.

Munição – Item em forma de determinada munição, cada arma tem uma munição diferente, adiciona 5 balas à munição de determinada arma. Também poderão ser conseguidas após a morte de um inimigo.

Orbitais Titans – Item que possibilita à entrada nas bases médias, fortaleza dos chefes, localizados no fim de cada fase. Cada fortaleza exige uma quantidade mínima de Orbitais.

Armas, proteções e Ataques

Pistola – Arma principal de God Of Arms, arma com a qual o jogador inicia, causa dano x 1 podendo destruir pequenos obstáculos e inimigos mais fracos.

BioGun – Arma biológica que causa dano mais forte do que a pistola, possui um dano x 3.

Granada Explosiva – Granadas de mão que podem atingir inimigos que estão atrás de obstáculos e possui dano x 5.

Bazuca – Lança foguetes de grande danos, a arma mais forte do jogador e causa um dano x 10.

Rifle – O Rifle é uma arma precisa, atira balas com uma distância impressionante. Além disso, possui um dano alto, porém, com isso, ela possui um delay alto para atirar novamente

Minigun – a Minigun é a arma perfeita a média distância, pois atira balas em rajada com danos medianos. O seu delay é o menor do jogo.

LightGun – arma muito parecida com a Biogun, porém, ela emite raios de energia ao invés de biológicos, possui um dano igual ao da Biogun, porém, em certos inimigos ganha um dano extra.

Laser – dispara raios lasers em seus inimigos. Possui um dano altíssimo, pois realiza cortes profundos em seus inimigos. O seu delay é baixo.

Granada de Gás letal – Granadas de mão que deixam o inimigo atordoado, além de tirar a saúde do inimigo bem devagar.

Movimentos Personagem principal

Correr – God Of Arms poderá mover-se correndo para frente e para trás.

Pulo – Pulo normal com distância média que desloca God Of Arms sobre obstáculos e pequeno inimigos, além de desviar de ataques inimigos.

Abaixar – God Of Arms poderá abaixar para desviar de ataques inimigos e para atirar.

Levantar – God Of Arms poderá levantar-se após ter se abaixado

Atirar – God Of Arms poderá atirar em seus inimigos

Inimigos

Monstros comandados por Godzilla:

Mutações e robôs com pouca inteligência utilizados para realizar e manter a ocupação da área Sul da América. São a linha de frente do exército inimigo.

Alien – Mutação de humanos recém-nascidos com o objetivo de aumentar a inteligência de humanos através de plantas sintéticas. Além de não conseguir seres humanos inteligentes, o experimento fundiu a melanina humana e das plantas dando uma pigmentação esverdeada a mutação. Apesar da experiência não dar certo, ele é utilizado para trabalhos de guarda e exploração. Não possui qualquer tipo de ataque, apenas utilizado para vigiar. [Vida: 1][Ataque: 3] ;

Harpie – São pequenas aves mutantes. Com o rosto de humano e corpo de ave. Criadas para entregar mensagens e controlar uma parte do espaço aéreo. Foi criada com larga escala no continente sul americano. Não possui ataque. [Vida: 1][Ataque: 3] ;

Phantom – Mutante voador, originado por acidente, quando um cientista usou uma fórmula nova em humanos. A fórmula fazia os homens expelir ectoplasma de dentro para fora do corpo, essa forma o consumia e a partir desse momento, surgiu essa mutação que assustava muitos homens na Terceira Guerra Mundial. Possui apenas um ataque direto, mais utilizado para amedrontar do que como um ataque. [Vida: 1][Ataque: 5];

Mech Dragon – Apesar de ser parecido e apelidado com Dragão, foi um robô construído com aparência inofensiva. Um ponto forte do Mech Dragon é o inimigo o superestimar. Foi um dos primeiros robôs a irem para o campo de batalha. Ele lança como ataque uma pequena chama. [Vida: 1][Ataque: 8];

Scorlion X – Um monstro criado a partir da mistura de genes de escorpião, leão e outro animal desconhecido (o que justifica o X no nome). Ele tem a aparência de um leão com a cauda de um escorpião. Visualmente não é possível distinguir algo de outro animal, mas após sua morte é possível ver o sistema respiratório de outro ser vivo, com um dragão, explicando o fato de ele lançar bolas de fogo. [Vida: 1][Ataque: 10];

Andros – Um robô pequeno convencional. Não possui uma forma assustadora para um campo de batalha. Com o raio laser embutido em seus olhos consegue atingir alvos a longa distância. [Vida: 1][Ataque: 12];

Diamond Head – Uma criatura monstruosa, criada a partir de experiências com a dureza do diamante em seres vivos, o que dava ao exército aliado a grande vantagem da resistência contra os inimigos. Lança ataques de diamantes. [Vida: 1][Ataque: 15];

SkyWorm – Um monstro voador que lança projéteis na posição do seu alvo terrestre. Um verme que consegue voar e identificar a presença de inimigos no espaço terrestre pelo céu. Então é feito um cálculo entre a distância do alvo, a gravidade e a velocidade normal do tiro do Skyworm para achar o ângulo do lançamento do projétil. [Vida: 1][Ataque: 10];

StinkFly – Um monstro aperfeiçoado por experiências com insetos, devido a grande força dos insetos e sua carapaça, a experiência foi um sucesso. Stinkfly pode alcançar grandes velocidades. Possui um ataque de toxina em formato gosmento com grande dano. [Vida: 1][Ataque: 15];

Nergal – Minotauro que utiliza seu porte físico para atacar os inimigos. Apesar de não lançar ataques, possui muita resistência e vai em direção ao inimigo para derrubá-lo. [Vida: 1][Ataque: 30];

Stubborn – Monstro voador que possui uma fisionomia assustadora. Possui somente ataques corporais, também utiliza seu porte físico para atacar os inimigos. Assim como o Nergal, possui muita resistência e vai em direção ao inimigo para derrubá-lo. [Vida: 1][Ataque: 35];

Gyodai – O monstro voador que lança raios pelos olhos. Seus ataques são impossíveis de esquivar, por isso, a única maneira de não ser atingido é correr.. Possui ataques poderosos. [Vida: 1][Ataque: 35];

Draconis – Um monstro voador que lança cuspes de fogo na diagonal no seu alvo terrestre. Tem a fisionomia de um pequeno morcego. Seu ataque não é tão forte, porém não é possível matá-lo com ataques terrestres. [Vida: 1][Ataque: 15];

Lizard – Monstro mais forte controlado por Godzilla. Possui ataques poderosos, além disso, possui muita resistência a danos. O seu ataque é uma bola de fogo que vai ao encontro dos seus inimigos. [Vida: 1][Ataque: 45];

Demon – O monstro voador mais forte. É encontrado próximo ao Lizard. Possui ataques poderosos, além disso, possui muita resistência a danos. O seu ataque é uma gosma azul que é produzida pelo seu corpo. [Vida: 1][Ataque: 40];

Carnage – Mutante humano que se atrofiou um pouco devido às experiências. Possui uma grande velocidade que muitas vezes é usada como ataque mortal. Apesar de uma aparência não ameaçadora, eles receberam esse nome por causa do grande número de morte das linhas inimigas quando estavam em ação. [Vida: 1][Ataque: 12].

Chefes

Godzilla – Um grande robô criado com nome e aparência do Godzilla. Possui uma grande armadura metálica que resiste bem aos danos, com grande poder de fogo. Com inúmeros e poderosos ataques a distância é uma criatura destrutiva, capaz de matar vários alvos apenas com 1 ataque. Foi criado com a característica de recuperar a saúde para durar mais tempo em batalha. [Vida: 1] [Ataque Light Beam: 50] [Ataque Missile: 40] [Ataque Energy Beam: 30] [Ataque Assault:20] [Ataque Fire Kick: 15] [Ataque Launch Pad:10].

Armadilhas

Serra – Instalado no chão ou em paredes. Colocadas durante a fase para dificultar a passagem de inimigos no território mutante/robô. Não pode ser destruída. [Ataque: 20].

Machado duplo – Normalmente instalados no teto dos castelos dos chefes, além de enorme causa uma grande quantidade de dano. Possui face afiada para os dois lados. Não pode ser destruído. [Ataque: 25].

Lança – Instalados na base do castelo de Godzilla, não possui movimento, porém causam um enorme dano. Não pode ser destruída. [Ataque: 35].

Pequenas lâminas cortantes – Instalado no chão. Colocadas em pontos estratégicos para dificultar a passagem de inimigos no território dos Titans. Não pode ser destruída. [Ataque: 10].

Piso que cai – Instalado por Godzilla para matar seus inimigos despercebidos. Este piso cai ao ser tocado. Se houver algo em cima dele também sofrerá a queda e com isso danos. [Ataque: 0 a 100].

Física

Movimento: A movimentação do jogador de corrida e ataque acontecerá sobre plataformas móveis ou não, dessa forma será necessário análise de gravidade sobre os atores que andam, atores que flutuam ou voam podem ignorar essa força. Tiros não sofrerão influência da gravidade seguindo em linha reta até o fim da tela, desde o ponto inicial ou até atingir algum corpo.

Colisão - As colisões ocorrerão entre objetos. Os objetos serão o jogador, inimigos, tiros, paredes, itens, etc.

Objetos podem apresentar áreas com dano nulo, ou seja, colisões naquela área não acarretaram dano, como colisão de um tiro com uma parede, a parede não sofrerá danos.

Como, o jogador pode fazer inúmeros movimentos por todo o mapa, a sua máscara de colisão não é simétrica a sua imagem. A máscara de colisão é retangular e menor do que a imagem, dando uma sensação de realidade tanto nos movimentos quanto nas colisões com o mesmo.

A colisão será representada por explosões, efeito de sangramento ao ser atingido e destruição de objetos.

A máscara de colisão dos inimigos dependerá da posição do mesmo na fase, pois, em alguns casos há necessidade de mudar a sua máscara de colisão. Inimigos que não se move por todos os pontos do mapa e não entra muito em contato com a física do jogo terá uma máscara de colisão simétrica a imagem. Inimigos que poderão ocasionar erros de colisão devidos sua localização no mapa terão uma máscara de colisão retangular.

Os tiros sofrerão anulação quando colidirem e explodirão no ar não ocasionando qualquer tipo de dano.

Combate – Os combates normalmente acontecerão à longa, média e pequena distância. A maioria dos inimigos lança projéteis ao invés de atacar com o próprio corpo, porém o jogo contém monstros que não atacam ou usam seu corpo para realizar ataques.

Inteligência Artificial

O uso de inteligência artificial será somente para alguns monstros, entre eles todos os chefes. A maior parte dos monstros executa movimentos independentes da aproximação ou ação do jogador, implementados por código comportamental padrão que irá descrever o seu movimento pela fase.

Será utilizada uma máquina de estados para representar o movimento da IA aos chefes do jogo. Sua definição simples facilita a implementação. É decidido o estado inicial q_0 da máquina de estados. A cada iteração um dado de entrada d é lido, se no estado atual x houver uma transição $\delta(x,d) = z$ que ao receber esse dado de entrada levar para outro estado z , então o estado atual será z .

A máquina de estados será implementada com a linguagem GML (Game Maker Language), linguagem orientada a objetos derivada de C.

O chefe da terceira fase (Godzilla) será implementado com uma máquina de estados finita que iniciará parado até o momento que receberá como entrada a distância mínima entre ele e o jogador, então entrará em um estado de alerta. Em outros estados ele realizará ataques, recarregará energia quando este estiver pouca e voará para recuperar energia quando for muito atingido pelo jogador e estiver com pouca saúde. Sempre realiza o melhor ataque possível dependendo, respectivamente, da quantidade de energia disponível.

Interface do Usuário

I – Telas de apoio e elementos de interação Off-game:

1 - **Tela de Título e Menu Principal:** Deve apresentar o nome do jogo e o “Menu principal”, é a primeira tela a ser apresentada ao usuário;

2 - **Tela de Perfis:** Mostra ao jogador perfis salvos na memória do jogo. Apresentará o título Perfil, o “Menu Perfil”;

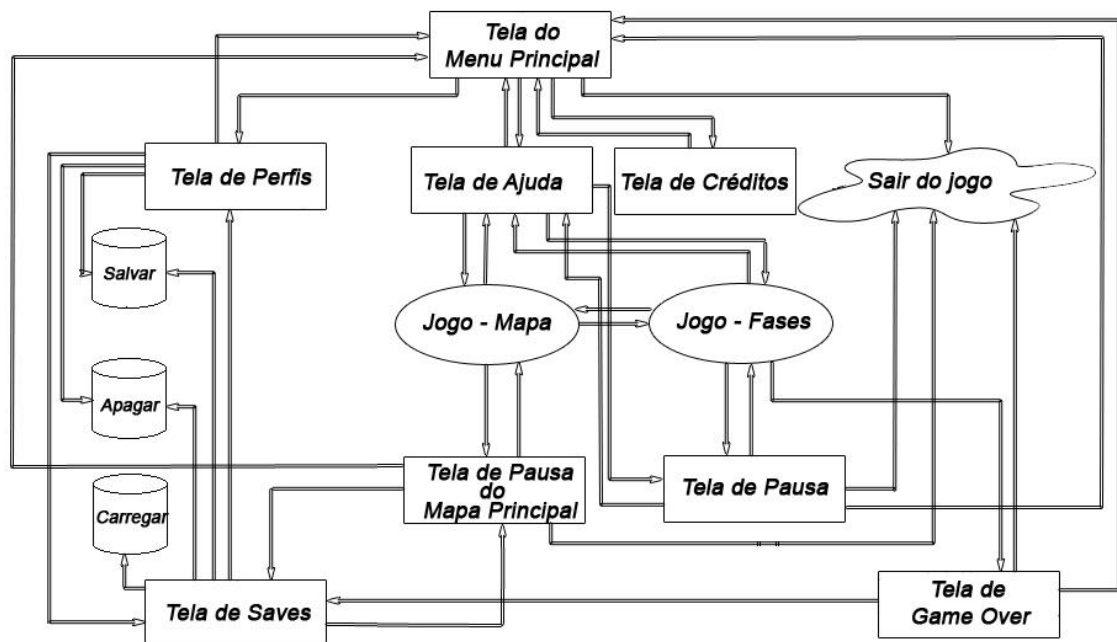
3 - **Tela de Saves:** Exibe ao jogador os saves do determinado perfil. Apresentará o Título Saves e o “Menu Saves”;

4 - **Tela de Pausa do Mapa Principal:** Apresentada quando o jogador para o jogo no Mapa Principal. Deverá apresentar o título Pausa e o “Menu de Pausa do Mapa Principal”;

5 - **Tela de Pausa:** Apresentada quando o jogador para o jogo em uma fase. Deverá apresentar o título Pausa e o “Menu de Pausa”;

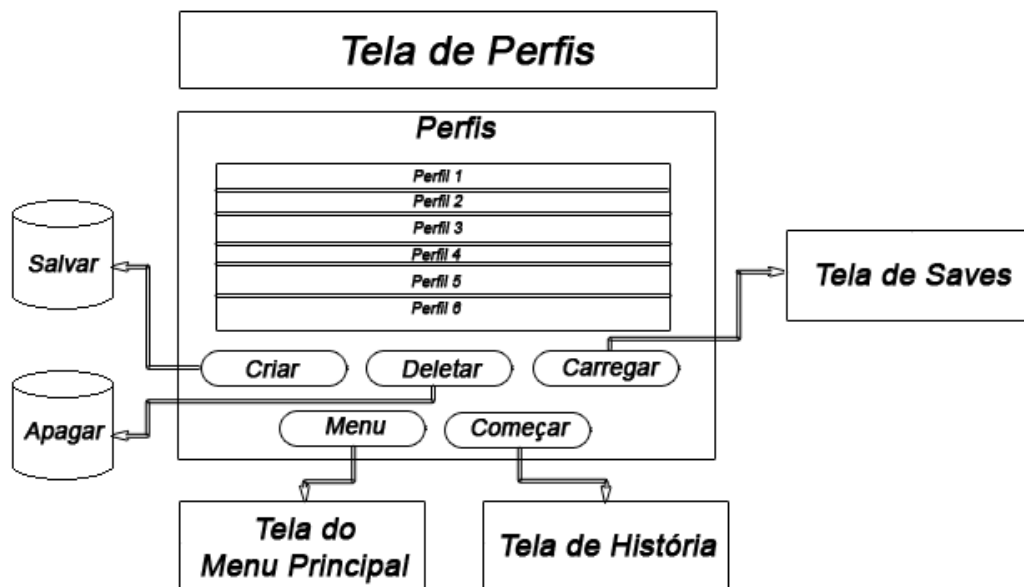
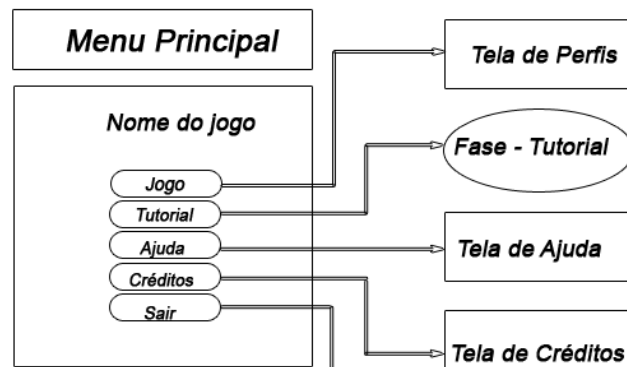
6 - **Tela de Ajuda:** Mostra ao jogador o título Ajuda e diversos comandos do teclado que controlam o jogo;

7 - **Tela de Game Over:** Mostrada quando o jogador perde todas as vidas, apresenta o nome Game Over e uma imagem ou pequena animação e o “Menu Fim de Jogo”;



II – Requisitos e imagem de telas e menus:

1 - **Tela de Título e Menu Principal:** O jogo irá iniciar com a tela de título, onde irá estarão listadas cinco opções: “Jogo” – Passa para a Tela de Perfis, “Tutorial” – Direciona para a Fase Tutorial, “Ajuda” – Exibe a Tela de Ajuda, “Créditos” – Passa para a Tela de Créditos, “Sair” – Encerra o jogo.

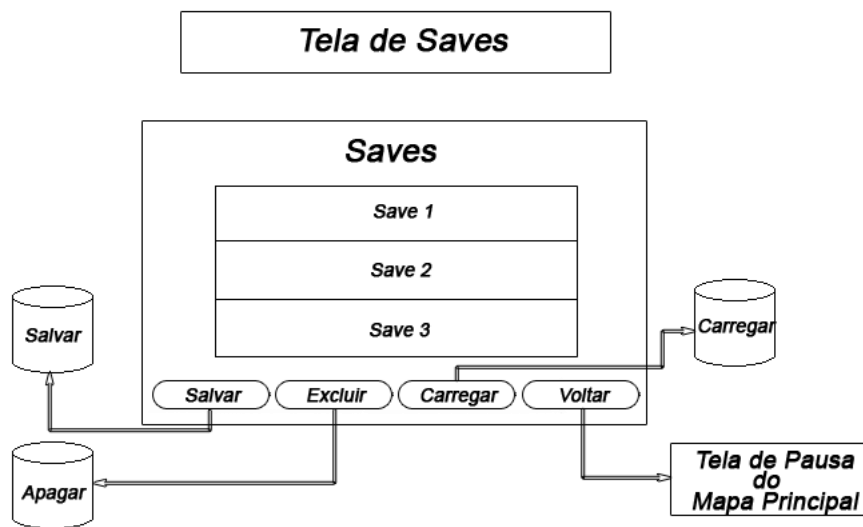


2 - **Tela de Perfis:** A Tela de Perfis é exibida quando o jogador escolhe a opção “Jogo” na Tela de Título e Menu Principal, estará disponível aos jogadores o máximo de seis perfis. Estarão listadas cinco opções na Tela de Perfis: “Criar” – Cria perfis dos jogadores, “Deletar” – Apaga algum perfil selecionado da memória, “Carregar” – Direciona o jogador para a Tela de Saves, após o jogador selecionar algum o perfil e se este tiver algum save armazenado memória, “Menu” – Volta a

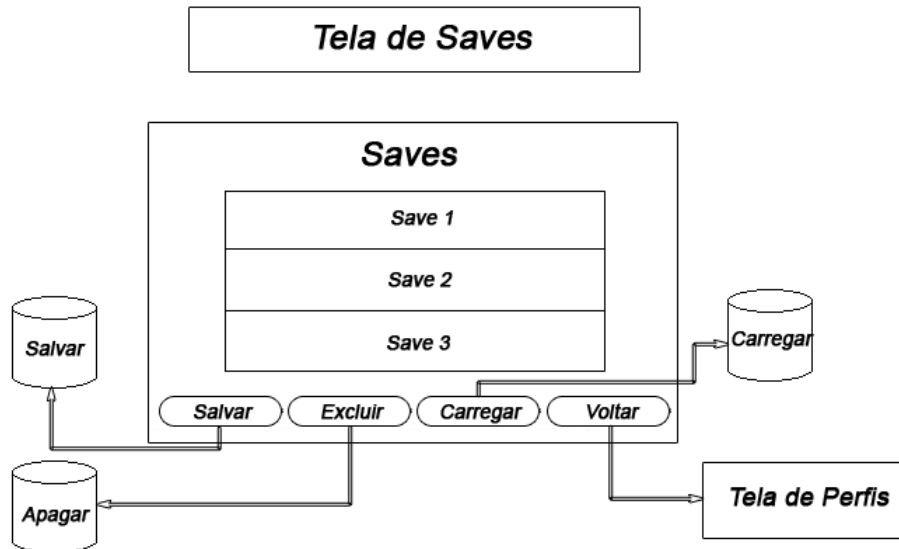
Tela de Título e Menu Principal, “Começar” – Após selecionar algum perfil, exibe a Tela de História do jogo.

3 - **Tela de Saves:** Será exibida a tela após o jogador escolher a opção “Saves” da Tela de Pausa do Mapa Principal. A memória é verificada e, então, se houver saves na memória, eles serão exibidos. Estará disponível o máximo de três saves por perfil. Terá quatro opções: “Salvar” – Salva o jogo atual na memória, “Excluir” – Apaga o jogo salvo selecionado da memória, “Carregar” – Após selecionar um save, retorna ao Mapa Principal onde o jogo foi gravado, “Voltar” – Retorna a tela anterior, existem dois casos possíveis, retornar para a Tela de Perfis e retornar para a Tela de Pausa do Mapa Principal.

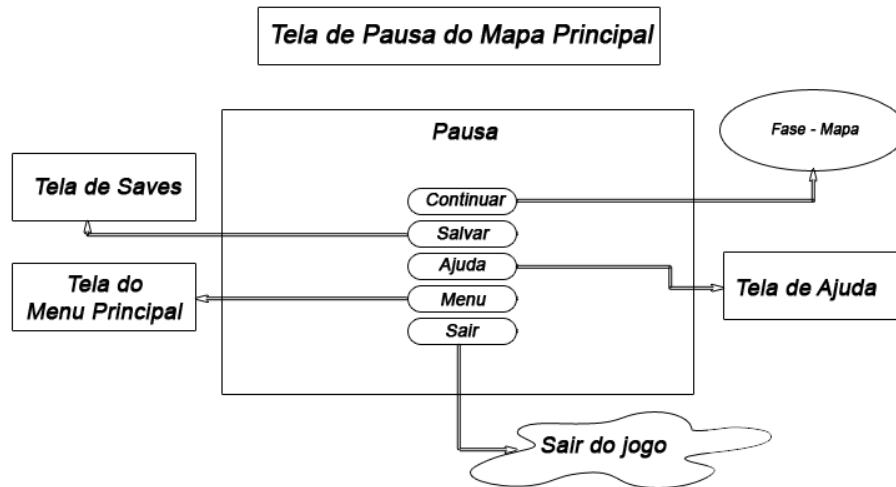
Caso 1 – Quando a última tela visitada foi a Tela de Pausa do Mapa Principal



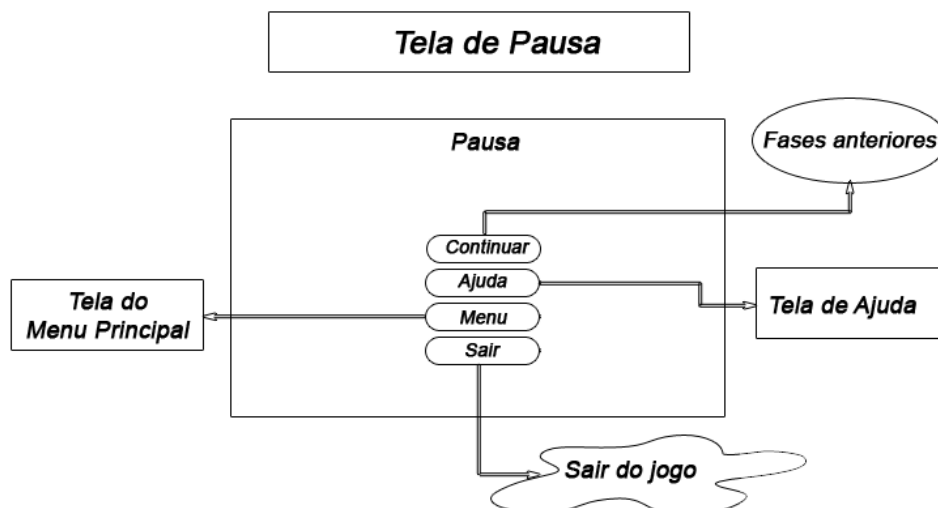
Caso 2 – Quando a última tela visitada foi a Tela de Perfis.



4 - **Tela de Pausa do Mapa Principal:** Mostrada após o jogador apertar a tecla <Esc> no Mapa Principal. Apresentará cinco opções: “Continuar” – Continua o jogo atual, “Salvar” – Direciona o jogador para a Tela de Saves do respectivo perfil, “Ajuda” – Exibe a Tela de Ajuda, “Menu” – Volta a Tela de Título e Menu Principal “Sair” – Encerra o jogo.



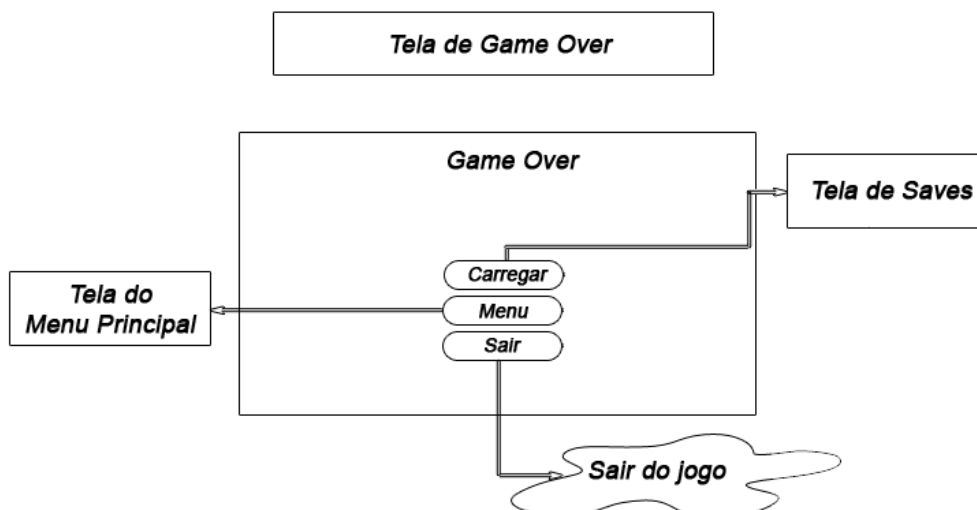
5 - **Tela de Pausa:** Mostrada após o jogador apertar a tecla <Esc> em uma fase do jogo. Apresentará cinco opções: “Continuar” – Continua o jogo atual, “Ajuda” – Exibe a Tela de Ajuda, “Menu” – Volta a Tela de Título e Menu Principal, “Sair” – Encerra o jogo.



6 - **Tela de Ajuda:** Mostra ao jogador os controles do jogo. É apresentada após o jogador escolher a opção “Ajuda” da Tela de Título e Menu Principal, da Tela de Pausa do Mapa Principal ou da Tela de Pausa dos diversos cenários. Dividida em três partes: Controles Primários, Controle Secundários, Teclas de Função.



7 - **Tela de Game Over:** Mostra após o jogador perder todas as vidas em uma fase do jogo. Apresentará três opções: “Carregar” – Direciona o jogador para a Tela de Saves do respectivo perfil, “Menu” – Volta a Tela de Título e Menu Principal, “Sair” – Encerra o jogo.



III – Objetos da GUI

As telas de apoio utilizarão um esquema padrão de fonte. As cores serão diferenciadas, apesar de seguirem a uma mistura de verde e preto, cores padrões do jogo. As opções serão representadas por botões que ao entrar em contato com o mouse ficarão com as arestas posterizadas. Se o mouse sair do contato com o botão, este voltará às características padrões. Perfis e Saves serão representados por retângulos com dados do jogo, quando selecionados ficaram com a cor padrão misturada com azul claro.

Os comandos nas telas serão todos realizados pelo mouse, utilizando apenas o botão esquerdo. Na tela de 'Ajuda' o jogador poderá voltar para a tela anterior se apertar <Esc>, no cenário do "Mapa Principal" e das fases ao apertar <Esc> será mostrado, respectivamente, as Tela de Pausa do Mapa Principal e Tela de Pausa.

IV – Relação Geral dos Objetos de Vídeo e arte do jogo

1. Aspectos gerais da arte:

God Of Arms é um jogo de gráficos 2D e scroll horizontal, o jogo é simples graficamente. O jogador, os inimigos, armadilhas e itens serão produzidos por gráficos 2D.

As animações do personagem, efeitos especiais, armadilhas e inimigos devem ser produzidas por sprites 2D.

Para o cenário, serão utilizadas imagens backgrounds ou tiles, para dar ao jogo uma maior realidade e, além disso, para não deixar o jogo pesado demais.

2. Animação e Arte 2D

Todos os elementos gerais do jogo se encontrarão aqui.

2.1. Elementos da GUI Offgame

- 1 – Tela de Título e Menu Principal;
- 2 – Tela de Perfis;
- 3 – Tela de Saves;
- 4 – Tela de História;
- 5 – Tela de Pausa do Mapa Principal;
- 6 – Tela de Pausa;
- 7 – Tela de Ajuda;
- 8 – Tela de Game Over.

2.2. Cenários

Terra no ano de 2053;
 Floresta;
 Castelo;
 Deserto.

2.3. Itens

Vida Extra;
 Primeiros Socorros;
 Armas;
 Munição;
 Orbitais Titans.

2.4. Efeitos Especiais

Tiros;
 Danos a God Of Arms;
 Danos a inimigos;
 Explosões (granadas, inimigos);

3. Elementos de arte 3D

A maioria dos sons feitos por inimigos serão 3D. Sons 3D referem-se ao som que tem uma posição (e velocidade) com respeito ao ouvinte. Embora seu uso seja mais proeminente em jogos 3D, ele será utilizado em God Of Arms, jogo 2D. A idéia é que um som tem uma posição relativa no espaço (0,0). O sistema calcula quanto o ouvinte iria ouvir o som e adaptá-lo em conformidade.

Sons e músicas

I. Músicas

A idéia das músicas de fundo é empolgar o jogador, serão utilizadas músicas que servirão para inspirar o jogador. As músicas serão aleatórias, mas também poderão ser a critério do jogador ao utilizar a tecla <M>.

Algumas músicas terão temas de guerra e outras serão apenas para deixar o jogador mais animado e envolvido com o jogo.

Sons serão utilizados para diversas situações, tais como: atirar, pegar itens, passar de nível.

Os sons de armas tentarão deixar o momento em que o jogador atira mais real. Cada arma terá o som próprio, o mais próximo possível da realidade.

O som ao pegar itens e ao passar de nível será apenas animações para deixar o jogo mais interativo.

Os sons dos inimigos serão 3D. Os sons 3D terão características que serão adaptadas de acordo com a distância do jogador ao inimigo.

As músicas terão formato MP3 e os sons terão formatos MID ou WAV.

II. Sons e efeitos

Efeitos especiais

Tiros das armas;

Explosões (granadas);

Explosões do Míssil

Characters

Danos a God Of Arms;

Morte de God Of Arms;

Danos aos inimigos;

Movimentos de robôs;

Sons realizados pelos monstros.

Game Play Elements

Coleta de armas;

Coleta de munições;

Coleta de vida extra;

Coleta de primeiros socorros;

Coleta de Orbitais Titans.

História (Roteiro)

No ano de 2053 a Terra vivia em uma época de caos. A humanidade estava à beira da extinção. Todas as buscas de poder e conhecimento da humanidade foram conseguidos para o colapso dela própria. Tudo isso começou no ano de 2036, a falta de recursos naturais, principalmente a água, tornou-se a fonte de busca de novos conhecimentos e experimentos. Era uma disputa pelo poder mundial e pelos recursos naturais, essa disputa foi chamada Corrida Tecnológica.

A Corrida Tecnológica foi o motivo para avanços impressionantes e aterrorizantes de todas as áreas da guerra. Esses avanços foram desde o aperfeiçoamento de armas existentes e a criação de novas armas até o surgimento de andróides com Inteligência Artificial e de mutações em seres vivos.

Em 2045, cem anos depois da Primeira Guerra Mundial, visto que, a paz não era mais uma aliada, foi dada início a Terceira Guerra Mundial, a guerra científica e tecnológica.

Três anos após o começo da guerra, aconteceu algo terrível, as esmagadoras forças dos mutantes se voltaram contra os seus próprios criadores, os andróides ao verem as forças humanas caindo rapidamente, se rebelaram e se juntaram as mutações, dando início a aliança Titans.

Após essa tragédia, as nações não tiveram outra escolha a não ser se unirem. A Terceira Guerra Mundial foi interrompida e a ONU – Organização das Nações Unidas declarou a Guerra Planetária.

Depois de anos e anos de luta, as forças e as armas humanas ficavam cada vez mais fracas contra o poderio dos robôs e dos mutantes. Os ataques humanos, cada vez com menor intensidade, já não faziam efeito às forças inimigas. Então, após as terríveis forças Titans dominarem o mundo, os países deixaram de existir e apenas algumas resistências humanas sobraram ao redor do mundo.

Um robô chamado Godzilla se proclama o líder das forças Titans, ele tem uma enorme quantidade de poder e sempre lidera a maior parte dos inimigos humanos para o campo de batalha. Nenhum monstro se propõe a lutar contra Godzilla e atualmente nenhum humano também teve essa coragem e sobreviveu. E ele está no topo da hierarquia da aliança Titans.

Godzilla junto com alguns robôs super inteligentes e mutantes poderosos

criaram técnicas astuciosas para assumir o controle da Terra. Eles se dividiram em grupos ao redor do planeta. Cada grupo pertence a áreas do mundo. Nos grupos há áreas pequenas e médias. As bases foram construídas com posição estratégica próxima a uma suposta localização de resistência. Em cada área do mundo há uma base média, uma fortaleza, onde se encontram os monstros chefes da área. Cada base média dá ordens para as pequenas. E as bases médias recebem ordens da base maior, onde se encontra Godzilla e os mais fortes e inteligentes inimigos.

Para esse sistema dar certo, os Titans criaram as Orbitais Titans, itens criados pelos robôs mais inteligentes do exército Titans. As Orbitais Titans eram feitas por uma estrutura hexagonal de átomos de carbono, chamada grafeno e programadas com linguagem de máquina criptografadas com chaves de 256 bits. As Orbitais Titans são o segredo da ordem e submissão dos poderes Titans.

Cada Orbital fica localizado no fim das bases, onde se encontram os monstros mais fortes das mesmas. Os Orbitais possuem o conhecimento necessário para abrir o caminho até Godzilla, pois eles têm códigos que são utilizados para poder passar às fortalezas em que ficam os chefes. Diferente das bases pequenas, nas bases médias os Orbitais são guardados com o seu chefe.

Godzilla, apesar de forte, estará disponível somente após o jogador enfrentar as bases gêmeas, cada um contendo um dos chefes mais poderosos. O homem que conseguir derrotar as bases gêmeas ganhará o respeito de Godzilla.

Nos dias atuais, restam poucas forças humanas em atividade, foi decidido pela maioria dos humanos que nenhuma resistência irá atacar mais os Titans e que o restante da população se esconderá para dar continuidade à espécie.

Nas resistências estão soldados treinados que foram capazes de sobreviver aos ataques dos inimigos. Mas nenhum deles é igual a você (God Of Arms), treinado com todos os tipos de armas e com uma inteligência tática de guerra. Ele clama por sangue inimigo e, apesar de estar em uma grande desvantagem numérica, ele pede para as resistências atacarem com forças totais aos inimigos. Porém, ele não consegue o que quer. Então, Sozinho, ele decide ir atrás do inimigo sozinho, todos sabem que apesar do perigo eminente e de ser um ataque suicida, se alguém pode vencer esta guerra, esse alguém é você. Você é conhecido pelos humanos como o Deus das armas por sua destreza com as mesmas. Mostre aos inimigos por que você é chamado assim.

Níveis

I. Fluxo

O jogo God of Arms contém 23. Em cada fase haverá um chefe, que somente será enfrentado após o jogador passar por todas as armadilhas e inimigos da fase. O jogador deverá coletar o máximo de Orbitais Titans possíveis, pois estes serão necessários para enfrentar o chefe final.

Ao derrotar um chefe o jogador irá passar de fase e poderá ir para o próximo cenário. A dificuldade dos níveis será equilibrada com o poder de fogo do jogador e com a destreza necessária para passar do nível. Os problemas serão aumentados gradualmente, para haver o balanceamento e o jogador não sentir maiores dificuldades quando se deparar com níveis extremamente difíceis.

A maior dificuldade do jogo será deixada para o final, quando o jogador estiver com armas melhores e estiver apto a enfrentar maiores perigos.

II. Descrição geral dos níveis e suas alocações

Nível 1 – Atacar!

Local: Fortzilla.

A primeira fase do jogo é realizada em uma área dominada por Godzilla, um robô que possui uma avançada técnica de Inteligência Artificial. Depois dos poderosos Titans dominarem o planeta, eles construíram bases em todos os territórios. Godzilla ordenou aos monstros com menor ataque, como Harpies e Phantoms, que fossem para a primeira linha de defesa localizada na floresta em frente ao castelo de Godzilla. Para auxiliar estes monstros apenas com ataques corporais foram enviados os Skyworms para a realização de ataques aéreos. São usados para a defesa de dentro do castelo monstros como Mech Dragon, Stinkfly, Elemental Fire, Nergal, Draconis e Stubborn que possuem uma maior resistência aos danos, além de ataques mais fortes. Godzilla se encontra em um santuário em um pequeno deserto após o castelo, já que esta fortaleza seria alvo para possíveis

ataques. Após o castelo, se encontram monstros de nível superior, sendo o Demon e o Lizard os inimigos mais fortes a serem combatidos antes de Godzilla. Por toda a fase são encontradas diversas armadilhas, desde pequenas lâminas cortantes até gigantescos machados duplos.

APÊNDICE B QUESTIONÁRIO DO JOGO

Avaliação do jogo God Of Arms Questionário referente ao nível de satisfação do usuário

1. Perfil do Jogador

- a) Sexo: Masc. () Fem. () b) Idade: _____
- c) Há quanto tempo você joga? _____
- d) Quantos jogos você joga por ano? _____
- e) Você já jogou um jogo do tipo aventura (Ex: Super Mario World)? _____

2. Jogo em geral

- a) Qual destes momentos foi o último que você passou no jogo?
- () Entrou no castelo () Entrou no 1º portal () Saiu do castelo
- () Enfrentou o chefe () Derrotou o chefe () Nenhum momento anterior

- b) Você gostou do jogo? Por quê?
- _____

- c) Você encontrou dificuldades no jogo? Quais?
- _____

- d) Você se sentiu desafiado pelo jogo? Em quais momentos?
- _____

3. Marque um xis (x) em apenas uma nota para:

3.1. Elementos do jogo

Itens	Notas de zero a dez (0 a 10)										
	0	1	2	3	4	5	6	7	8	9	10
Interface (Gráfico)											
Controles											
Áudio											
IA ⁶ (chefe)											
Interação											

3.2. Jogo

Item	Notas de zero a dez (0 a 10)										
	0	1	2	3	4	5	6	7	8	9	10
Jogo											

⁶ Inteligência Artificial