

# DESENVOLVIMENTO DE JOGOS ELETRÔNICOS

Felipe Castanheira Morais<sup>1</sup>

Cristiano Maciel Silva – Orientador<sup>2</sup>

<sup>1</sup>felipeshaman@gmail.com; <sup>2</sup>cristianomsilva@yahoo.com.br;  
Curso de Ciência da Computação – Uni-BH (www.unibh.br)

**Resumo** – Neste trabalho são apresentadas técnicas para o desenvolvimento de jogos eletrônicos, uma área da computação em larga expansão. Também são apresentadas definições sobre jogos eletrônicos e seu impacto na sociedade enquanto meio de entretenimento.

**Palavras-chave** – Jogos Eletrônicos.

**Abstract** – This paper presents relevant techniques related to game-development, an expanding area of computer science. This paper also presents definitions and concepts related to electronic games and its impact in the society as means of entertainment.

**Keywords** – Computer games

## 1 INTRODUÇÃO

O desenvolvimento de jogos eletrônicos tem se tornado, cada vez mais, uma área predominante da engenharia de software. Com a ampliação do mercado de entretenimento virtual, cada vez mais demanda-se profissionais nesta área, que exige conhecimento técnico específico e diferente daquele encontrado em processos de engenharia bem difundidos. Este trabalho demonstra este processo utilizando-se de ferramentas computacionais simples para se desenvolver um jogo. No capítulo 1 será discutido o que é um jogo eletrônico, trazendo desde sua origem até o mundo atual juntamente com a sua fundamentação teórica. No capítulo 2 são discutidos os tipos de jogos, as diferenças entre eles, tecnologias utilizadas e os elementos encontrados nos jogos, desde os primórdios da era Atari até os dias presentes. No capítulo 3 será discutido o processo de desenvolvimento de um jogo, desde a concepção do projeto até os testes necessários, além da etapa de *game design*, que envolve todo o processo de desenvolvimento do jogo. No capítulo 4 será apresentada a proposta de desenvolvimento em que será utilizada a ferramenta *Game Maker*,

demonstrando suas capacidades como motor (engine) de jogo e os benefícios que trás, limitações e aspectos técnicos do software e de sua linguagem, GML, que é utilizada massivamente no projeto proposto. No capítulo 5 tem-se as conclusões obtidas do trabalho desenvolvido, as dificuldades encontradas, as soluções completas, além de trabalhos futuros.

### 1.1 O QUÊ É UM JOGO

O primeiro jogo de computador foi criado nos Estados Unidos, em 1958, em um computador analógico. O jogo era conhecido como “Tênis para Dois” sendo mostrado na tela de um osciloscópio. Pequenas barras representavam as raquetes enquanto um ponto representava a bola. Naturalmente, este jogo nunca saiu do laboratório. O primeiro jogo comercial foi criado em 1972, chamado “PONG”, e possuía uma premissa similar à do “Tênis Para Dois”. Ele foi criado pelo engenheiro Ralph Baer e foi comprado pela Magnavox, que construiu o primeiro *videogame* doméstico: o Odyssey. Na sequência, surgiram os primeiros arcades, e também o Atari, que foi o grande marco dos jogos eletrônicos, por ser mais acessível, barato e simples. Alguns anos mais tarde, a Nintendo, empresa japonesa que até então trabalhava no ramo de cartas de baralho, entra no mundo dos jogos e cria o Famicom (conhecido como *Nintendo Entertainment System* no resto do mundo), que revoluciona com jogos como *Super Mario Bros*, *The Legend of Zelda* e *Donkey Kong*, grandes marcos e inspirações para os futuros jogos. A partir daí, a evolução neste mercado foi incessante, passando por evoluções em hardware e software. Os jogos passaram a ter um visual mais refinado.

CRAWFORD(1982) define jogos eletrônicos como um sistema fechado, formal, que,

subjetivamente, demonstra um subconjunto do mundo real. Entende-se como fechado, o fato do jogo possuir seu próprio mundo, ou seja, possui uma estrutura auto-suficiente e completa – sem a necessidade de acesso ao mundo exterior para sua evolução (em desenvolvimento e em prática). A quebra dessa regra é comum e jogos que forcem o jogador a estender as regras do jogo ao mundo real tendem ao fracasso. Por formal, diz-se que o jogo tem suas próprias regras e leis, onde o jogador as segue em direção à vitória (objetivo final de todos os jogos). Alguns jogos possuem um conceito conhecido como sandbox, onde as regras são vagas e livres, e o jogador pode tomar decisões baseadas no contexto em que se encontra. Em suma, jogos eletrônicos são “sub-mundos” da nossa realidade com o objetivo de entreter, divertir, ensinar e evoluir. Um jogo cria uma representação simples, subjetiva e deliberada da realidade. A fantasia criada pelo jogador é a chave para o sucesso real de um jogo.

## 1.2 FUNDAMENTAÇÃO TEÓRICA

A proposta deste trabalho é demonstrar a importância das etapas de desenvolvimento de jogos tratando-as com maturidade, ressaltando o valor dessa área da computação no mundo atual, onde jogos eletrônicos fazem parte do cotidiano de grande parte das pessoas. Este trabalho será baseado na visão de CRAWFORD(1982), que entende que o desenvolvimento de jogos eletrônicos é uma arte e que deve ser levada com seriedade, pois exige grande dedicação, inspiração, criatividade e paciência, as principais virtudes de um artista.

## 2 TIPOS DE JOGOS ELETRÔNICOS

Jogos são subdivididos em diversos tipos, cada um com uma premissa diferente para o jogador. O projeto de um jogo deve começar pela definição do tipo de jogo.

### a) Jogos de Aventura

Um jogo de aventura é aquele em que o jogador toma o papel de um herói em busca de um objetivo, passando por diversos obstáculos. Um clássico exemplo desse estilo de jogo é Super Mario Bros. (figura 1), em que o jogador toma controle de um simples encanador italiano em busca de salvar a princesa de seu reino, sequestrada por um dragão. Jogos de aventura geralmente têm histórias simples, e uma jogabilidade fácil, pois o principal público alvo são os jogadores iniciantes (geralmente crianças).



Figura 1 – Super Mario Bros., clássico jogo de aventura

### b) Jogos de Ação

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. Este tipo de jogo geralmente abrange outros tipos, como jogos de luta, combate, tiro, etc. Um exemplo de jogo de ação, bastante conhecido é *Doom* (figura 2), onde o jogador controla um soldado em uma missão em Marte sendo confrontado por monstros.



Figura 2 – Doom, exemplo do jogo de ação

### c) Jogos de RPG

RPG é um estilo de jogo que surgiu em jogos de mesa e tabuleiro. Consiste de uma história complexa, longa, em que as ações do jogador influenciam o decorrer do jogo. RPG significa “*Role Playing Game*”, ou seja, um jogo em que o jogador recebe o papel do personagem, interagindo com os personagens não-jogadores no mundo do jogo. Os personagens são customizáveis pelo jogador, o que gera uma imersão maior, tornando o jogador parte do mundo do jogo, adquirindo experiência e poder, para chegar ao objetivo final do jogo. Exemplos de jogos de RPG são diversos, como a série *Final Fantasy* (figura 3), *Dragon Quest*, *The Legend of Zelda*, dentre outros.



Figura 3 – Final Fantasy, exemplo de jogo de RPG

### d) Jogos de Estratégia

Assim como os jogos de RPG, os jogos de estratégia também são derivados dos jogos de mesa e tabuleiros. Em um jogo de estratégia, o jogador toma controle de um exército de personagens, executando ações como coletar recursos ou dominar territórios. Um exemplo de jogo de estratégia é *Starcraft* (figura 4), jogo de grande sucesso em que o jogador toma controle de um exército espacial, e que hoje em dia existem até competições oficiais.



Figura 4 – Starcraft, exemplo de jogo de estratégia

### e) Jogos Esportivos

Jogos esportivos são aqueles que simulam esportes do mundo real, como futebol, basquete, tênis, etc. Esse tipo de jogo é bastante popular no Brasil, principalmente aqueles de futebol. Exemplos são *Winning Eleven*, *NBA Street*, *Virtua Tennis*, etc.

### f) Jogos Online

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 na grande rede de computadores. Isso permite ao jogador uma imersão maior no mundo do jogo, pois os outros personagens encontrados também são pessoas (ao invés de personagens controlados pelo computador). Exemplos de jogos online são *World of Warcraft* (figura 5),

*The Duel Online*, *Counter-Strike*, *Team Fortress*, entre outros.



Figura 5 – World of Warcraft, exemplo de jogo online

#### g) Jogos de Simulação

Este é um outro tipo de jogo que abrange uma grande gama de variações. Simuladores de vôo, corrida, e até mesmo da vida real são exemplos de tipos de jogos de simulação. Estes jogos tomam conceitos da vida real e os coloca dentro do jogo, gerando uma sensação de realidade para o jogador maior do que os outros estilos. Exemplos desse tipo de jogo são *The Sims*, *Flight Simulator*, *Need for Speed*, etc.

Além desses, sempre existem jogos inovadores que criam seu próprio estilo. Estes jogos geralmente ganham destaque no mercado, e podem abranger áreas pouco exploradas pelos estilos anteriores. Existem também jogos que abrangem diversos estilos em um só, desde um jogo de aventura com cenas de corrida, ou dividido em diversos *minigames*, onde cada um possui seu próprio estilo, como visto na série *Warioware* (figura 6), por exemplo.



Figura 6 – Diversas telas de minigames do Warioware.

### 2.1 ELEMENTOS DE UM JOGO ELETRÔNICO

Por trás do cenário de um jogo existem regras que definem a jogabilidade, os desafios que o jogador irá encontrar e as condições de vitória ou derrota. Todo esse contexto de regras é definido em um mundo específico, separado do real, dentro do jogo. Como exemplo, em um jogo de simulação de corrida, o “mundo” é a pista e a interação do jogador com esse mundo é que ditará se o jogador vence ou não. Com a escolha do mundo, deve-se escolher a perspectiva de visão do jogador, e a forma em que o mundo será representando – em 2D ou 3D. Jogos como *Super Mario Bros*, *Sonic* e outros do estilo, começaram em 2D antes de pular para o mundo 3D, onde a liberdade de movimento é maior, porém, torna o jogo mais complexo. Em um jogo 2D, é comum o uso da técnica de *Side-Scrolling*, que consiste em visualizar o mundo do jogo visto de lado com as únicas direções possíveis de se caminhar sendo esquerda, direita, cima e baixo (sem noções de profundidade). Pode-se utilizar a visualização *top-down* (figura 7), que consiste em ver o mundo de cima, tornando a liberdade maior. Este tipo de visão é mais comum em jogos de RPG como *Final Fantasy*. Em um mundo 3D, estas restrições são anuladas, e o mundo é mais “livre”. Depois de definida a perspectiva do jogo, deve-se pensar o quão realista este será. Alguns jogos possuem caráter cômico, portanto não precisam de gráficos realistas. Já um jogo que simula o

mundo real, quanto mais próximo da realidade, melhor, o que cria uma imersão maior do jogador no mundo do jogo.



Figura 7 – Exemplo de Visão *Top-Down*

Finalmente, após definir o realismo do jogo, define-se o contexto do mundo em que o jogador vai encontrar seus desafios em busca da vitória. Dependendo do estilo de jogo, essa parte pode ser a mais ou a menos importante. Em um jogo de aventura, a jogabilidade simples e a diversão proporcionada são mais importantes que a história, porém, em um jogo de RPG, a história é um fator muito importante no desenvolvimento do jogador no mundo do jogo.

### 3 PROCESSO DE DESENVOLVIMENTO DE JOGOS ELETRÔNICOS

O processo de desenvolvimento de jogos eletrônicos difere em diversas partes do desenvolvimento de software convencional. De acordo com CLUA (2004), a estrutura de um projeto de desenvolvimento de jogos segue como a figura 8 demonstra. Game Design seria a etapa em que o jogo é criado em si, ou seja, anexa-se a história aos personagens no mundo com sua fantasia própria, além de envolver diversos outros passos como criação da IA, por exemplo. Na etapa de rascunho dos objetos e cenários, começa-se a desenhar, graficamente e textualmente, os personagens, ambientes, objetos

e etc, para auxiliar a próxima etapa, que é a modelagem, onde são criados os desenhos definitivos que estarão presentes e farão parte do jogo final. Já a etapa de engenharia de software envolve mais o sistema de computação do jogo, dando o apoio necessário para a etapa de programação, que vem logo a seguir, onde o jogo é efetivamente desenvolvido para funcionar em uma determinada plataforma. No caso do projeto proposto neste trabalho, a plataforma é o próprio PC.

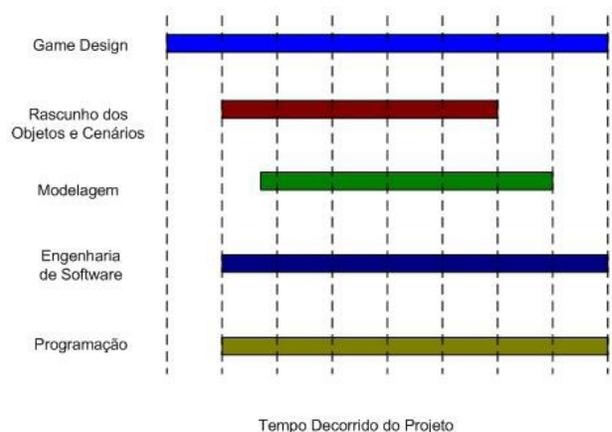


Figura 8 – Estrutura de um projeto de desenvolvimento de jogos.

Como mostrado na figura 8, a maioria destas etapas são feitas em paralelo, e geralmente, a etapa de engenharia pode ser dividida em partes, dependendo do progresso do desenvolvimento.

#### 3.1 GAME DESIGN

A proposta de um jogo, seja eletrônico ou não, é fazer com que os jogadores se encorajem a pensar, raciocinar, e tomar as devidas ações que o tornem o vencedor. Estas ações que deverão ser tomadas são ditas pelas regras do jogo, que por sua vez, são as primeiras coisas criadas na etapa de *game design*. O *game design* é uma etapa em que se deve:

- Imaginar o jogo no futuro;

- Definir sua jogabilidade, ou seja, como ele deve se comportar com o jogador;
- Descrever seus elementos, sejam estes gráficos ou conceituais);
- Tornar importante o desenvolvimento do jogo como um produto.

Cabe dizer também que, o game design, é a etapa mais importante do processo de desenvolvimento de jogos, pois engloba todas as áreas necessárias para a finalização do projeto, e, caso algum destes momentos fujam do controle, todo o projeto pode ser comprometido. Um design mal feito vai gerar um jogo ruim, sem dúvidas, e este problema deve-se ao fato de que o game design ainda é uma idéia muito nova na indústria. Um *game design* inovador geralmente traz um bom retorno, e este é um dos pontos principais do projeto proposto neste trabalho.

### 3.2 CONCEITUAÇÃO

Antes de se começar o processo de desenvolvimento, a idéia deve ser colocada em pauta: como o jogo deve ser, quais os objetivos do jogador e que desafios ele irá encontrar para alcançá-los. Nesta etapa se tem uma idéia vaga do que o jogo será, porém, não é necessariamente aquilo que o jogo se tornará no final, pois pode haver mudanças necessárias para melhoria da jogabilidade que, por mais que no conceito a idéia parecesse boa, na prática não seria assim, portanto, é o processo onde as idéias começam a serem pensadas.

O conceito de idéia do jogo é bastante abstrato, pois pode vir de todos os lugares, como qualquer processo artístico. O game designer pode ver em qualquer coisa algo que gere uma grande idéia para o projeto, assim como um artista sempre busca inspiração em outras coisas fora de seu ambiente de trabalho. A partir dessas idéias, começam a ser desenvolvidas as

características do jogo, desde a personalidade dos personagens até a ambientação dos cenários.

### 3.3 IMERSÃO

Um jogo eletrônico propõe um novo mundo ao jogador, e para que o mesmo tenha máximo proveito do que o produto tem a oferecer, ele deve imergir no mundo fantasioso criado. Quanto maior a ilusão de realismo gerada do jogo em relação ao jogador, maior a imersão e mais ele estará interessado em avançar e conseguir alcançar seus objetivos, e tudo isso é um importante fator que deve ser alcançado na própria etapa de game design. O mundo de um jogo eletrônico deve ser harmônico, ou seja, todas as partes do jogo devem fazer parte de único contexto. A imersão, além de envolver o ambiente, também envolve os personagens do jogo, que devem fazer com que o jogador se sinta a vontade com eles. Estes personagens podem assumir diversos esteriótipos, desde o herói, que é o centro da história, aquele que o jogador terá o maior contato, até o vilão, aquele que é o antagonista da história e geralmente é o criador dos obstáculos encontrados pelo herói. Um mundo estático com personagens mal planejados e sem personalidade, com uma história mal elaborada, geralmente termina em um jogo ruim.

### 3.4 DESENHO

A parte gráfica de um jogo é onde o usuário vê o novo mundo que ele vai interagir, portanto, este deve ser bem desenhado, modelado e harmônico, o que quer dizer, os cenários onde os personagens irão interagir possuem grande relevância à história e o contexto em que o jogo se realiza, portanto, não pode-se fugir deste padrão. Os personagens também devem fazer parte deste mundo, e geralmente, o jogador deve se identificar com eles, principalmente com o herói, não só na aparência, mas também na personalidade.

A primeira etapa do desenho é a criação dos rascunhos de objetos e cenários em que o jogador irá interagir. Estes rascunhos, então são utilizados pela equipe de computação gráfica para gerar os sprites, no caso de um jogo 2D, ou os modelos, no caso de um jogo 3D. Só então, estes são inseridos no jogo, que, mesmo sem os gráficos, pode-se utilizar gráficos substitutos, conhecidos como “*placeholders*”, enquanto a versão final não está pronta, com intuito de testes que não envolvem a parte gráfica do jogo, como *scripts* de eventos.

### 3.4 INTELIGÊNCIA ARTIFICIAL

O mundo de um jogo não é apenas um cenário e o jogador, mas também, existem os personagens não-jogadores, os quais serão aqueles com o que o jogador irá interagir em sua aventura. Alguns destes personagens, além da personalidade criada através da história para ele, devem possuir uma forma de interagir com o jogador dependendo das ações tomadas por este, como por exemplo, um inimigo atacá-lo em momentos oportunos ou se esconder quando em perigo.

Existem diversas formas de se utilizar conceitos de inteligência artificial em jogos, desde as máquinas de estado finito, onde que é um modelo básico de IA, em que em um determinado instante, dependendo do estado atual, o personagem irá tomar uma decisão em relação ao ambiente à sua volta e às ações do jogador. Máquinas de estado finito são fáceis de ser implementadas e são muito eficientes para a IA básica de um jogo. Outra forma de IA para jogos que é bastante utilizada são os algoritmos de busca clássicos, como A\* e Dijkstra, sendo estes utilizados principalmente para calcular caminhos mais curtos para desviar de um possível obstáculo. Uma das formas mais utilizadas de IA para jogos, inclusive utilizada no projeto proposto por este trabalho, é o *scripting*, em que comandos de código são interpretados em tempo real no jogo, dando mais

flexibilidade às opções tomadas pelo personagem.

Alguns jogos possuem sistemas complexos de inteligência artificial. The Sims, demonstrado na figura 9, utiliza um conceito conhecido como *A-Life* ou *Artificial Life*, onde o conceito do jogo é participar da vida de uma família, simulando o desenvolvimento das pessoas de forma natural, incluindo a interação entre dois ou mais personagens não-jogadores.



Figura 9 – The Sims, que possui um complexo sistema de IA

### 3.5 PROGRAMAÇÃO

A programação do jogo acompanha o desenvolvimento desde o início, mesmo que o jogo ainda não possua gráficos definidos ou uma história completa, pode-se adiantar muitas coisas que virão a ser necessárias no futuro do desenvolvimento. Um jogo pode ser programado em virtualmente qualquer linguagem de programação, apesar de que, hoje em dia, a maioria dos jogos eletrônicos é feito em C/C++. Em conjunto da linguagem de programação, deve ser utilizada uma API (*Application Programming Interface*) que trabalhe com gráficos, como o DirectX e o OpenGL. Para dar suporte à linguagem e à API, utilizam-se os motores de jogo, ou *Game Engines*, que geralmente já possuem funções ou módulos já próprios para o desenvolvimento de um jogo,

tais como cálculo de colisão, física de um ambiente, tratamento de sons, de vídeo, e que dá toda a base para o início do desenvolvimento.

#### 4 PROPOSTA DE DESENVOLVIMENTO

O jogo proposto para ser desenvolvido para este trabalho acadêmico é do estilo aventura, como descrita no capítulo 2, cuja visão será aquela utilizada em vários jogos do estilo, como em *Super Mario Bros.*, a visão de *Side-Scrolling*, já citada anteriormente. O jogo propõe uma nova iniciativa no mundo dos jogos: o jogo é feito para ser extremamente difícil, como visto na figura 10, porém, sem penalidades ao jogador, ou seja, você aprende a jogar o jogo ao perder. Com a experiência da derrota anterior, o jogador passa a ter conhecimento do ambiente e consegue, de alguma forma, evitar o acontecimento, conseguindo assim, prosseguir com seu objetivo. Esta iniciativa é um fator inovador no momento do game design, e geralmente, faz com que o jogo chame mais a atenção dos jogadores, prendendo-os mais ao jogo, pois foge dos padrões dos jogos normais, os quais já podem, geralmente, serem considerados “enjoativos”.



Figura 10 – Um exemplo de obstáculo no jogo.  
Tocar qualquer um dos espinhos significa perder.

O jogo é uma sátira em relação às antigas gerações dos jogos eletrônicos, utilizando-se de gráficos e trilhas sonoras dos mesmos, e também personagens com suas características, trazendo

assim, um sentimento de nostalgia aos jogadores que, em algum momento de sua infância, se divertiram com aqueles jogos, como mostrado na figura 11.



Figura 11 – Cenário baseado no clássico Pitfall

#### 4.1 METODOLOGIA DO DESENVOLVIMENTO

O jogo foi feito na linguagem GML, que está atribuída ao ambiente de desenvolvimento conhecido como *Game Maker 7.0*, e, neste capítulo, será explicado como funciona a linguagem e o ambiente de desenvolvimento, além de exemplos do código para maior compreensão.

De acordo com sua documentação oficial, o *Game Maker 7.0* é um programa que possui uma *game engine*, ou um motor de jogo, e um grupo de bibliotecas que facilitam a abstração do desenvolvimento de um jogo eletrônico ou outras ferramentas computacionais. O *Game Maker* é um software proprietário e comercial, ou seja, não possui seu código livre e sua versão completa deve ser comercialmente adquirida. A versão utilizada para o desenvolvimento do jogo proposto é a versão LITE, distribuída gratuitamente pela empresa desenvolvedora do *Game Maker*.

O *Game Maker* possui como maior característica a utilização de uma linguagem própria, a GML (*Game Maker Language*). É uma linguagem de sintaxe simples e de fácil

aprendizagem, além de já possuir itens práticos exclusivos para o desenvolvimento de jogos eletrônicos. É uma linguagem orientada a objetos e flexível, ou extensível, ou seja, pode-se atribuir novas funções que não estão naturalmente embutidas na linguagem. Sua sintáxe é parecida com a da linguagem C, como mostrado no exemplo do código 1, embora o *Game Maker* tenha sido desenvolvido em Delphi, o qual é o motivo de apenas existir para a plataforma Windows.

**Exemplo de Código 1 – Faz um objeto se mover na horizontal caso um trigger(gatilho) seja ativado:**

```
if (trigger1.triggered){ hspeed=5; }  
else{ hspeed=0; }
```

O *Game Maker*, além de trabalhar com a GML, também utiliza um sistema facilitador, chamado de *drag'n'drop*, em que não é necessário utilizar linhas de código para programar as ações do jogo, mas sim, utiliza-se de comandos pré-definidos, de fácil entendimento, que atuam como o código GML. Este tipo de interface é recomendada aos desenvolvedores iniciantes, que não possuem um conhecimento aprimorado de linguagens de programação, porém desejam desenvolver o seu próprio jogo, que porém, não poderá alcançar todas as possibilidades de customização abertas pela GML pura. O projeto proposto neste trabalho utiliza-se de ambas as possibilidades, para demonstrar todas as formas em que o *Game Maker* pode trabalhar para um desenvolvedor.

Para gerenciar recursos gráficos e sonoros, o *Game Maker* utiliza o sistema clássico de árvore de arquivos, onde cada recurso se encontra em seu respectivo estilo, por exemplo, um efeito sonoro se encontra na pasta *Sounds*, enquanto uma imagem de fundo se encontra na pasta *Backgrounds*. Os arquivos são importados diretamente do disco-rígido computador portador do software, e são armazenados internamente quando o jogo é compilado. Este processo pode se tornar custoso em relação à memória, pois, se o jogo possuir

um grande número de recursos, mais memória será demandada no momento de carregar o jogo, e, para isso, o *Game Maker* disponibiliza ações de importar e remover recursos em tempo de execução. Dessa forma, a quantidade de recursos carregados em um momento é limitada, economizando memória para quando recursos desnecessários não estão sendo utilizados.

## 4.2 ESPECIFICAÇÕES DO PROJETO

O jogo desenvolvido para este trabalho segue o estilo plataforma 2D, onde o personagem é visto de lado, e a profundidade do cenário não é acessível em termos de jogabilidade, apenas visibilidade. O nome do jogo é “*I Wanna be the Tribute*”, por ser um jogo baseado em um jogo já conhecido, chamado “*I Wanna Be The Guy*” (O'REILLY, 2008), que é mostrado na figura 12. O jogo será distribuído gratuitamente na internet, porém, não possuirá código livre. Ele possui quatro níveis de dificuldade e cabe ao jogador escolher qual a lhe convém. O jogo se passa em um mundo abstrato que é uma mistura de vários ambientes de outros jogos famosos, estes ambientes variando desde os personagens encontrados até às músicas e efeitos sonoros.



Figura 12 – *I Wanna Be The Guy*, jogo que inspirou o desenvolvimento deste projeto

## 5 CONCLUSÃO

Com este trabalho, conclui-se que o projeto de desenvolvimento de jogos difere-se em diversos aspectos dos processos de engenharia de software convencionais, principalmente por existir a necessidade da inspiração artística para um bom game design, por este ser um ponto especialmente responsável pelo sucesso e pela qualidade do produto final. Um bom game design não só gera um bom jogo, mas também um bom game designer, que são aqueles que se dedicam ao estudo das técnicas propostas neste trabalho, desde a concepção das idéias até o desenho do ambiente, da interface homem-máquina, e do balanceamento do jogo.

### 5.1 TRABALHOS FUTUROS

Um jogo sempre deve ser atualizado de alguma forma, seja com sequências, que são novos jogos seguindo a mesma linha do anterior, porém, como alguma novidade que irá atrair jogadores, ou seja com atualizações de expansão, ou seja, cria-se novos ambientes e objetivos para o jogador, utilizando-se do projeto atual, para que o jogo nunca fique obsoleto e sem novidades.

Portanto, é proposto como trabalho futuro, uma atualização de expansão para o jogo criado neste trabalho, com adição de novos cenários, personagens e obstáculos, assim, aqueles que já terminaram de concluir os seus objetivos na versão atual possam voltar a jogar para desafiar o que houver de novo nessa expansão, que também poderá incluir novos projetos envolvendo inteligência artificial, computação gráfica e a própria engenharia de software, para que o projeto não seja apenas mais do mesmo.

### AGRADECIMENTOS

Agradeço principalmente aos meus pais, pela educação que me deram e que, graças a ela, pude chegar onde estou. Agradeço aos professores do DCET do UNI-BH os quais me lecionaram, pois

sem o conhecimento e a paciência que eles têm, nada do que fiz até hoje academicamente seria possível. E agradeço à minha querida Júlia pelo apoio que me deu, e por sempre estar ao meu lado quando preciso.

### REFERÊNCIAS

ABRAGAMES Homepage. Plano Diretor da Promoção da Indústria de Desenvolvimento de Jogos Eletrônicos no Brasil – Diretrizes Básicas; Comitê de Promoção da Indústria de Desenvolvimento de Jogos Eletrônicos no Brasil. 2004. Disponível em: [http://www.abragames.org/docs/pd\\_diretrizesbasicas.pdf](http://www.abragames.org/docs/pd_diretrizesbasicas.pdf), acessado em: 23/09/2008.

CLUA, Esteban. Ferramentas necessárias para desenvolvimento de jogos. Campinas, Unicamp, 2004.

CRAWFORD, Chris. The Art of Computer Game Design. 1982. Disponível na Web <http://www.vancouver.wsu.edu/fac/peabody/game-book/Coverpage.html>, acessado em 23/09/2008

GAME MAKER 7.0 DOCUMENTATION, Disponível em [yoyogames.com/gamemaker/docs](http://yoyogames.com/gamemaker/docs) acessado em 10/03/09.

KOSTER, Raph. A Theory of Fun for Game Design. Palygraph Press, 2004.

PERUCIA, Alexandre Souza. Desenvolvimento de Jogos Eletrônicos. São Paulo, 2007

O'REILLY, Michael. I Wanna be the Guy, Disponível em <http://kayin.pyoko.org/iwbtg/> acessado em 10/04/09.

ROLLINGS, Andrew; ADAMS, Ernest. Andrew Rollings and Ernest Adams on Game Design. 1ª edição. New Riders, 2003

ROUSE, Richard. Game Design: Theory and Practice, 2nd Edition. Wordware Publishing, 2000 Rollings, Andrew; Morris, Dave. Game Architecture and Design. New Riders Games, 1999

