

# **GAMES INDEPENDENTES**



**Fundamentos metodológicos  
para criação, planejamento e  
desenvolvimento de jogos digitais**

**DAVID DE OLIVEIRA LEMES**



Associação Commercial, Industrial e Cultural de Games



PUC-SP



**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO • PUC-SP**

**Programa de Pós-Graduação em  
Tecnologias da Inteligência e Design Digital**

**David de Oliveira Lemes**

[dolemes@pucsp.br](mailto:dolemes@pucsp.br)

**GAMES INDEPENDENTES:**  
Fundamentos metodológicos para criação,  
planejamento e desenvolvimento de jogos digitais

**São Paulo**

**2009**



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO • PUC-SP

**David de Oliveira Lemes**

**Games Independentes:**

Fundamentos metodológicos para criação,  
planejamento e desenvolvimento de jogos digitais

Dissertação apresentada à Banca Examinadora  
como exigência parcial para obtenção do título  
de MESTRE em Tecnologias da Inteligência e De-  
sign Digital, pela Pontifícia Universidade Católica  
de São Paulo, sob orientação da Profº Dr. Luís  
Carlos Petry.

**São Paulo**

**2009**



## BANCA EXAMINADORA

---

---

---



## DEDICATÓRIA

A todos aqueles que sonham em um dia em desenvolver um game.

Ao tio Otávio Lemes e avó Alice Dias (*in memorian*).

Aos meus pais: Inês de Oliveira Lemes e Osvaldo Lemes, por tudo!

A todos os leitores do GameReporter.

Para Andréa e Pedro.

## AGRADECIMENTOS

A toda minha família, que sempre me apoiou durante minha trajetória acadêmica, especialmente Andréa e Pedro, pela compreensão e presença constante durante o tempo em que me dediquei a este trabalho.

Aos meus pais Osvaldo e Inês e meus irmãos Matheus, Lízia e Samaria, que sempre estiveram presentes em todos os momentos. Felipe, pelas perguntas que nunca param.

Ao meu orientador, Prof. Dr. Luís Carlos Petry, pela paciência, dedicação, inspiração e postura acadêmica na qual eu me espelho todos os dias. Obrigado por me guiar.

Aos colegas do Curso Tecnologia em Jogos Digitais e Tecnologia e Mídias Digitais da PUC-SP: Luís Carlos Petry, Rogério Cardoso, Sérgio Basbaum, Alexandre Braga, Fábio Musarra, Sérgio Nesteriuk, Victor Emmanuel, Donizette Louro, Eliseu Lopes, Maurício Pontuska, Lúcia Leão, Flávio Sogawa, Vicente Gosciola e todos aqueles que participaram e continuam participando no meu percurso acadêmico.

À secretaria do TIDD, em especial a Edna Conti, pelas orientações sempre pontuais. Aos alunos de Jogos Digitais e Mídias Digitais da PUC-SP. Aos amigos do Insper, que sempre se preocuparam com meu trabalho acadêmico. A todos os leitores do GameReporter: foi lá que este trabalho começou.

À PUC-SP, por fazer toda a diferença em minha vida.

## RESUMO

LEMES, David de Oliveira. **Games Independentes - Fundamentos metodológicos para criação, produção e desenvolvimento de jogos digitais.** 158 f. Dissertação (Mestrado) - Pontifícia Universidade Católica de São Paulo: São Paulo, 2009.

A presente pesquisa reúne e analisa os fundamentos metodológicos para criação, planejamento e desenvolvimento de jogos digitais com foco no produtor independente de *games*.

Transcorrendo sobre as características fundamentais dos jogos digitais, investiga-se nesse trabalho relatar brevemente a história dos *games* independentes no Brasil e organizar a forma como deve ser pensados a argumentação narrativa, a organização de ideais, a mecânica de jogo e os gêneros de *game* e personagens quando relacionadas à criação de *games*.

Busca-se também sistematizar as principais questões relacionadas ao planejamento de jogos digitais, como o *game design* e o gerenciamento de projetos, a criação e o *design* de personagens, progressão, dificuldade, jogabilidade, roteiro aplicado ao universo dos *games*, linguagens de programação e bibliotecas de software.

A produção de jogos digitais enfoca aspectos tecnológicos e práticos, começando com o *design* de interface para *games*, a modelagem 3D em seus principais aspectos, como objetos, cenários e mundos tridimensionais desenvolvidos para *games*, a produção de imagens 2D como texturas e seus similares, busca situar a ambientação sonora no universo dos jogos digitais até chegar ao motor de jogo, relatando suas principais funcionalidades para a produção de *games*.

**Palavras-chave:** desenvolvimento; *games*; jogos digitais; metodologia.

## ABSTRACT

LEMES, David de Oliveira. **Independent Games - Methodological elements for creation, production and development of digital games.** 2009. 158 p. Essay (Master's Degree) - Pontifícia Universidade Católica de São Paulo: São Paulo, 2009.

The following research gathers and analyses the methodological elements for creation, production and development of digital games with a focus on the independent game producer.

It elapses over the fundamental characteristics of digital games to underlie the theme to be studied, it seeks to report briefly the history of games and characters when talking about questions related to game creation.

It pursues to systematize the main questions over planning digital games as the game design and the project management, the creation and the character design, progression, difficulty, playability, applied script to the game universe, programming languages and software libraries.

The production of game design focus practical and technological aspects, starting on interface design for games, 3D modelling in its main aspects, as objects, scenarios and game developed tridimensional worlds, the production of 2D images as textures and similarities, it seeks to situates the sound environment in the digital game universe until it reaches the game engine, reporting its main features for game production.

**Keywords:** development; games; digital games; methodology.

## LISTA DE FIGURAS

<b>Figura 01</b> - Atari 2006	19
<b>Figura 02</b> - Controle <i>Dual Shock 2</i> para PS2	24
<b>Figura 03</b> - Tela do game <i>Amazônia</i>	29
<b>Figura 04</b> - Tela do game <i>TriLinea</i>	31
<b>Figura 05</b> - Imagem de abertura do game <i>God of War</i>	33
<b>Figura 06</b> - Tela do game <i>Mario Bros</i>	35
<b>Figura 07</b> - Rascunho de um mapa mental	38
<b>Figura 08</b> - Tela do game <i>Myst</i>	48
<b>Figura 09</b> - <i>Spacewar!</i>	49
<b>Figura 10</b> - <i>Space Invaders</i>	51
<b>Figura 11</b> - Tela do software <i>Openproj</i>	57
<b>Figura 12</b> - <i>O Homem Vitruviano</i>	65
<b>Figura 13</b> - <i>Model sheet</i> - figura humana	66
<b>Figura 14</b> - Site <i>All the Human Photo References</i>	67
<b>Figura 15</b> - <i>Model sheet</i> - personagem caricato	69
<b>Figura 16</b> - <i>Model sheet</i> - carro caricato	69
<b>Figura 17</b> - Tabela esquema de personagem	70
<b>Figura 18</b> - Projeto inicial de uma fase	77
<b>Figura 19</b> - Rascunhos	84
<b>Figura 20</b> - Estrutura de jogo	85
<b>Figura 21</b> - Interface <i>Macintosh</i>	92

<b>Figura 22</b> - Interface do jogo <i>Warcraft</i>	97
<b>Figura 23</b> - Interface do jogo <i>Need for Speed Most Wanted</i>	97
<b>Figura 24</b> - Tela de abertura do jogo <i>Wolfenstein 3-D</i>	99
<b>Figura 25</b> - Tela de abertura do jogo <i>Doom</i>	100
<b>Figura 26</b> - Interface do <i>Autodesk 3ds Max</i>	103
<b>Figura 27</b> - Interface do <i>Autodesk Maya</i>	104
<b>Figura 28</b> - Interface do <i>Blender</i>	105
<b>Figura 29</b> - Textura em ambiente 3D	108
<b>Figura 30</b> - Interface do software <i>GIMP</i>	110
<b>Figura 31</b> - <i>Wii Sports</i>	111
<b>Figura 32</b> - Interface do software <i>Audacity</i>	116
<b>Figura 33</b> - Esquema de funcionamento de um motor de jogo	118
<b>Figura 34</b> - Interface do <i>3D Gamestudio</i>	120
<b>Figura 35</b> - Interface do <i>Neoaxis</i>	121
<b>Figura 36</b> - Interface do <i>Blender Game Engine</i>	123
<b>Figura 37</b> - Interface do <i>Unity 3D</i>	125

# SUMÁRIO

<b>Introdução: vamos jogar?</b>	14
<b>1 - Criação de jogos digitais</b>	17
1.1 - Características fundamentais dos jogos digitais	19
1.2 - O <i>game</i> e a cena independente	27
1.3 - Rascunhos, argumento e organização de ideias	33
1.4 - Mecânica de jogo, gêneros e narrativa	43
1.5 - Papéis e personagens	49
<b>2 - Planejamento de jogos digitais</b>	53
2.1 - <i>Game design</i> e gerenciamento de projetos	53
2.2 - Criação e <i>design</i> de personagens	63
2.3 - Níveis: progressão, dificuldades e jogabilidade	73
2.4 - Roteiro: formatando o documento de <i>game design</i>	81
2.5 - Linguagem de programação e bibliotecas	87
<b>3 - Desenvolvimento de jogos digitais</b>	91
3.1 - Interfaces para <i>games</i>	91
3.2 - Modelagem 3D: objetos, cenários e mundos	99
3.3 - Produção de imagens 2D: texturas e similares	107
3.4 - Ambientação e produção sonora	111
3.5 - Motor de jogo: engine e suas funcionalidades	117
<b>Conclusão</b>	127
<b>Apêndices</b>	137
<b>Referências bibliográficas</b>	153

## Introdução: vamos jogar?

Nos dias de hoje é muito simples criar e desenvolver produções digitais das mais variadas. Ferramentas, *softwares*, tutoriais, *plug-ins* e aplicativos estão disponíveis aos montes na grande rede. Quer montar um site na internet? É possível fazer isso em poucos minutos com apenas alguns cliques. Precisa fazer uma apresentação e está cansado dos modelos presentes em seu computador? Existem incontáveis *templates* disponíveis para *download* na internet, necessitando apenas que seja preenchido com conteúdo. Essa mesma linha de raciocínio, que existe muita ‘coisa pronta’ na *web*, se aplica para diversos temas, em voga no mundo digital.

Contudo, quando o assunto é o *game*, essa questão pode ficar um pouco mais complicada, pois não existe a possibilidade de se criar um jogo digital em poucos minutos e com alguns cliques do *mouse*. A mística que envolve o desenvolvimento de *games* faz parecer com que esta produção digital específica seja intangível e não acessível ao grande público e aos interessados no tema.

Mas não é. Apesar de não ser tão simples quanto criar um *blog*, a criação e produção de *games* não são tão difíceis quanto parece. A criação de jogos requer método, disciplina, ideias, roteiro, desenho, recursos tecnológicos, *softwares* específicos e muita vontade de se fazer um jogo.

A presente dissertação visa descomplicar o que parecer ser complicado e que pretende refletir sobre as questões permeiam os fundamentos metodológicos para a criação, planejamento e desenvolvimento de *games*, com foco no produtor independente. A jornada não é longa e tem tudo para ser divertida. Só resta perguntar: vamos jogar?

O primeiro capítulo aborda as características fundamentais dos jogos digitais, busca relatar a história dos *games* independentes no Brasil e organizar a forma como deve ser pensada a argumentação narrativa, a organização de ideais, a mecânica de jogo, os gêneros de *game* e personagens quando são tratadas as questões relacionadas à criação de *games*.

O capítulo dois busca sistematizar as principais questões relacionadas ao planejamento de jogos digitais como o *game design* e o gerenciamento de projetos, a criação e o *design* de personagens, progressão, dificuldade, jogabilidade, roteiro aplicado ao universo dos *games*, linguagens de programação e bibliotecas de software.

O capítulo três aborda a produção de jogos digitais com foco nos aspectos tecnológicos e práticos como: *design* de interface, modelagem 3D em seus principais aspectos, como objetos, cenários e mundos tridimensionais desenvolvidos para *games*, a produção de imagens 2D como texturas e seus similares, ambientação sonora no universo e motor de jogo, relatando suas principais funcionalidades para a produção de *games*.



## 1 - Criação de jogos digitais

**Fase 1.** Uma nave exploratória pousa em um planeta desconhecido repleto de *pixels* e parte para sua missão: descobrir como o universo digital, lúdico e interativo foi criado e investigar aspectos relativos a sua evolução. O caminho não será demasiadamente longo. Começa a jornada: no presente capítulo será apresentado tópicos relativos à criação de *games* tendo como foco central o produtor independente de jogos digitais. Contudo, antes de serem tratados questões que contextualizem pontos chaves sobre a criação de *games*, o jogo digital passará por uma breve apresentação de suas principais características.

Ao navegar por um *game*<sup>1</sup>, o neófito em jogos digitais poderá, em um dado momento, descobrir alguma vocação relacionada com a atividade da concepção e produção de jogos eletrônicos e/ou digitais. É caso de inúmeros jovens que prestam hoje o vestibular para cursos de graduação que tem por linha mestra a formação na produção de jogos digitais. Ao entrar por esta via, nosso jogador que está prestes a se converter em um produtor, iniciando o contato com outras atividades e interesses para além da função usuário-jogador.

Este é um dos inúmeros caminhos que alguém pode percorrer até tornar-se um produtor de *games* e, especialmente, um produtor independente. Para que isto ocorra, será necessário adquirir competências específicas, todas elas nascidas dentro da formação e na pesquisa do tema.

É caso da competência criativa. Descreveremos aqui os passos dessas competências, que envolvem desde a percepção das características fundamen-

---

<sup>1</sup> Segundo a Wikipédia, *game* é um jogo eletrônico no qual o jogador interage com imagens enviadas a um dispositivo que as exibe geralmente uma televisão ou um monitor. O termo *videogame* também é amplamente utilizado para se referir ao console de *videogame*. Mais detalhes em <http://pt.wikipedia.org/wiki/Game> (acessado em 27/03/2009).

tais que todo o *game* não somente deve ter, mas igualmente fomentar, bem como a capacidade de definir e dar vida às personagens.

Começamos então o nosso percurso de pesquisa mestral pelas características fundamentais que devem animar todo e qualquer jogo digital.

## 1.1 - Características fundamentais dos jogos digitais

Os jogos digitais, comumente conhecidos como *games*, fazem parte da cultura de massa<sup>2</sup> há pelo menos 30 anos, desde a popularização dos consoles de *videogames*, como o *Atari Video Computer Systems* (*Atari VCS*) e a consequente invasão de lares por todo o mundo.

Lançado em 1977, o console que seria mais tarde chamado de *Atari 2600*, chegou a vender 8 milhões de unidades até 1983. De acordo com Vicente (2005), no início da década de 80, o *Atari* já era um grande sucesso. O mercado então é inundado por centenas de novos jogos. Acessórios e periféricos revolucionários prometiam ao usuário controlar as ações do jogo apenas com o seu pensamento, como no caso raríssimo do *MindLink*<sup>3</sup>.



Figura 01 - *Atari 2600*: mais de 8 milhões de unidades vendidas de 1977 até 1983

---

<sup>2</sup> Cultura de massa é toda cultura produzida para a população em geral – a despeito de heterogeneidades sociais, étnicas, etárias, sexuais ou psicológicas – e veiculada pelos meios de comunicação de massa. Mais em: [http://pt.wikipedia.org/wiki/Cultura\\_de\\_massa](http://pt.wikipedia.org/wiki/Cultura_de_massa) (acessado em 21/02/2009).

<sup>3</sup> Mais detalhes podem ser encontrados em: [http://en.wikipedia.org/wiki/Atari\\_Mindlink](http://en.wikipedia.org/wiki/Atari_Mindlink).

Apesar do grande sucesso comercial do *Atari 2600*, os jogos digitais não ficaram confinados aos consoles de *videogame* e são classificados (ou agrupados) por gêneros. Segundo Santaella (2007), os *games* dividem-se em três grandes tipos, reunidos a partir do suporte utilizado: jogos para consoles ocorrem em um console acoplado ao monitor de uma televisão à parte, como o *Atari* ou o *Playstation*<sup>4</sup>; os jogos para computador ocorrem no monitor do computador a partir de seu próprio *hardware*; e os jogos para arcades, que alguns chamam de equivocadamente de ‘fliperama’, são grandes máquinas integradas (o console com um monitor), geralmente dispostas em lugares públicos.

Como pudemos ver, a partir da década de 1970, os *games* por intermédio dos consoles de *videogame* passaram a fazer parte da cultura de massa. Assim sendo, como podemos definir um *game*?

Um game é uma atividade lúdica composta por uma série de ações e decisões, limitada por regras e pelo universo do game, que resultam em uma condição final. As regras do universo do game são apresentadas por meios eletrônicos controlados por um programa digital. As regras e o universo do game existem para proporcionar uma estrutura e um contexto para as ações de um jogador. As regras também existem para criar situações interessantes com o objetivo de desafiar e se contrapor ao jogador. As ações do jogador, suas decisões, escolhas e oportunidades, na verdade, sua jornada, tudo isso compõe a “alma do game”. A riqueza do contexto, o desafio, a emoção e a diversão da jornada de um jogador, e não simplesmente a obtenção da condição final, é que determinam o sucesso do game (SCHUYTEMA, 2008, pg. 7)

---

<sup>4</sup> *PlayStation* é um console *videogame* da Sony, lançado em 3 de dezembro de 1994 no Japão e em 9 de setembro de 1995 nos Estados Unidos. Desde o seu lançamento até 2006 (quando sua produção fora extinta), o *PlayStation* vendeu mais de 100 milhões de unidades, superado apenas pelo seu sucessor, o *PlayStation 2* de acordo com a Wikipédia. - <http://pt.wikipedia.org/wiki/PlayStation> (acessado em 25/02/2009). Mais informações em: <http://pt.playstation.com/>.

Schuytema (2008) apresenta uma definição de quem está inserido na indústria dos *games*. Porém, questões relativas à definição de jogo estão presentes na literatura há muito tempo. Por este motivo, segundo Ranhel (2009), a ideia de criar um conjunto de definições de jogo em um painel único parece um caminho sensato. Esse trabalho foi realizado por Jesper Juul e, uma vez montado, tenta extrair uma definição que contemple todos os aspectos ou elementos destacados no conjunto de definições agrupadas no painel. Juul parte das definições de Huizinga, Caillois, Bernard Suits, David Kelley, Avedon & Sutton-Smith, Chris Crawford e Salen & Zimmerman presentes na seguinte tabela:

FONTE	DEFINIÇÃO
Johan Huizinga	“... uma atividade livre, conscientemente tomada como ‘não-séria’ e exterior à vida habitual, mas ao mesmo tempo capaz de absorver o jogador de maneira intensa e total. É uma atividade desligada de todo e qualquer interesse material, com a qual não se pode obter qualquer lucro, praticada dentro de limites espaciais e temporais próprios, seguindo uma certa ordem e certas regras. Promove a formação de grupos sociais com a tendência a rodearem-se de segredos e a sublinharem sua diferença em relação ao resto do mundo por meio de disfarces ou outros meios semelhantes.”
Roger Caillois	“(o jogo) é uma atividade que é essencialmente: livre (voluntária), separada (no tempo e espaço), incerta, improdutiva, governada por regras, fictícia (faz-de-conta).”
Bernard Suits	“Jogar um jogo é se engajar em uma atividade dirigida para causar um estado específico de ocorrências, usando somente meios permitidos por regras, onde as regras proíbem meios mais eficientes em favor de meios menos eficientes, e onde tais regras são aceitas apenas porque elas tornam possível tal atividade.”
Avedon & Sutton-Smith	“No seu nível mais elementar podemos definir jogo como um exercício de sistemas de controle voluntário, nos quais há uma posição entre forças confinado por um procedimento e regras, a fim de produzir um resultado não estável.”

<b>Chris Crawford</b>	“Eu percebo quatro fatores comuns: representação (um sistema formal fechado, que subjetivamente representa um recorte da recorte da realidade), interação, conflito e segurança (o resultado do jogo é sempre menos severo do que as situações que o jogo modela).”
<b>David Kelley</b>	“Um jogo é uma forma de recreação constituída por um conjunto de regras que especificam um objeto (objetivo) a ser almejado e os meios permissíveis de consegui-lo.”
<b>Salen &amp; Zimmerman</b>	“Um jogo é um sistema no qual jogadores engajam-se em um conflito artificial, definido por regras, que resultam em um resultado quantificável.”

Vicente Gosciola, em entrevista concedida ao projeto Profissão *Gamer*<sup>5</sup> em 2005, afirma que o *game* é uma hipermídia por excelência. Mas o que é a hipermídia? Segundo Santaella (2001), antes da era digital, os suportes estavam separados por serem incompatíveis: o desenho, a pintura e a gravuras nas telas, o texto e as imagens no papel, a fotografia e o filme na película química, o som e o vídeo na fita magnética. Depois da passagem pela digitalização, todos esses campos tradicionais de produção de linguagem e processos de comunicação humanos juntaram-se na constituição da hipermídia.

Para ampliar o contexto sobre as características dos jogos digitais, é sensato que olhemos para o passado para entender que, antes do advento dos *video-games* como conhecemos hoje, existiam simplesmente os jogos (como visto na tabela acima). E estamos falando de jogos de qualquer natureza. Sobre estas questões que Johan Huizinga chegou a definir o homem como o ser que brinca e, nesse sentido, afirma que:

---

<sup>5</sup> Profissão *Gamer*: trabalho de conclusão de curso para obtenção do título de Bacharel em Tecnologias e Mídias Digitais pela PUC-SP realizado em 2005 sob orientação do Prof. Dr. Luis Carlos Petry. O Prof. Vicente Gosciola foi um dos entrevistados do projeto. O software de pesquisa acadêmica, que também pode ser chamado de *game*, pode ser encontrado para download em: <http://www.dolemes.org/2007/09/10/profissao-gamer/>.

O jogo é um traço essencial talvez o mais importante das sociedades humanas. Diferentemente dos outros animais que brincam, o homem é o único que faz conscientemente e durante a vida para obter prazer (HUIZINGA, 2004, pg. 47).

Se pela ótica de Huizinga (2004), o jogo é um traço essencial da sociedade, não é difícil entender o grande sucesso comercial dos jogos digitais, que começou com o lançamento do *Atari 2600* em 1977 e segue até os dias de hoje. Contudo, existe outro elemento que, dado a evolução tecnológica em curso, foi determinante para o casamento do lúdico com o tecnológico: a interatividade mediada por aparatos tecnológicos.

Para Santaella (2004), uma definição básica de interatividade nos diz que se trata de um processo pelo qual duas ou mais coisas produzem um efeito uma sobre a outra ao trabalharem juntas. E continua:

No *videogame*, por exemplo, em que o jogo fica mais difícil conforme o jogador alcança alguns pontos (...) pode-se considerá-lo interativo porque o jogador está recebendo, respostas em tempo real da pessoa que criou o *software* para o jogo (SANTAELLA, 2004, pg. 153).

Analizando uma das definições de interatividade proposta por Santaella, vemos que, jogos de todos os tipos, inclusive os predecessores dos jogos digitais, são interativos. E estas interações podem acontecer das mais diversas formas, sobretudo nas competições diretas entre jogadores. Além a interatividade, a imersividade (chamada também de imersão por diversos autores) é um ponto que, por ser inerente a este novo meio, que são os jogos digitais, anda junto com a interatividade nas relações diretas e indiretas.

Em computação gráfica existem muitos graus de imersividade, seja num cenário 3D ou não. O conceito de imersividade está relacionado com o grau de interatividade que um usuário é capaz de ter numa aplicação. Esta interatividade não está apenas relacionada à capacidade de “andar” num cenário, mas também com a capacidade de interagir com objetos e outros personagens dentro deste mundo virtual. Outros fatores que permitem aumentar o grau de imersividade de uma aplicação são o seu foto-realismo (semelhança com o mundo real) e estímulos sensoriais, que podem ser dados por joysticks e diversos dispositivos de entrada (como por exemplo, no joystick Dual Shock 2, do Playstation 2 (PS2), pode-se perceber até 128 níveis de pressão no controle, além de ser capaz de fazer com que o controle vibre de acordo com situações do jogo, como por exemplo quando um carro passa raspando na parede ou entra no banco de areia da pista) (CLUA e BITTENCOURT 2004, pg. 6).



**Figura 02 - Dual Shock 2 para PS2:** controle vibra de acordo com as situações de jogo

Percebemos que a questão do jogo, tão presente na condição humana, ganhou o suporte digital em função da evolução das tecnologias que permearam o século XX e continuam presentes no século XXI. Contudo, se a interatividade já estava presente antes dos jogos chegarem aos ambientes digitais

e a imersividade se faz presente nos ambientes tridimensionais, o que diferencia os jogos do passado com os *games* do presente? De acordo com Nesteriuk (2002), ao analisar este novo universo, é notório que a principal diferença entre os *videogames* e seus precursores não eletrônicos é que os *videogames* acrescentaram automação e complexidade - eles podem sustentar e calcular regras do jogo por si só permitindo, por meio disso, mundos de jogos (*gameworlds*) mais profundos; além de permitir a manutenção do ritmo do jogo. Assim, *videogames* criaram novos mundos, mais tempos-reais e mais jogos individuais (*single player*) que os jogos não eletrônicos .

A característica interativa, presente em todos os jogos, segundo Rabelo (2005), é a dependência de comandos sobre uma interface digital<sup>6</sup> que faz com que o projeto desta natureza não seja um filme ou uma animação, mas um *game*. Nos jogos digitais, vemos que, a interatividade e a imersividade, são fatores determinantes que, aliados ao conjunto de tecnologias que permitem os *games*, foram primordiais para construir as características básicas e fundamentais desse universo em constante evolução. E sendo digital por natureza, concluímos que para um seu pleno funcionamento, é necessário um suporte computadorizado, seja este em forma de um computador (PC ou MAC) ou de um console de *videogame*.

Sob a ótica da presente pesquisa, todas as características supracitadas dos jogos digitais são plenamente aplicadas no desenvolvimento de um *game* independente, como veremos no tópico a seguir.

---

<sup>6</sup> O conceito de Interface se expressa pela presença de uma ou mais ferramentas para o uso e movimentação de qualquer sistema de informações, seja ele material, seja ele virtual. Pode significar um circuito eletrônico que controla a interligação entre dois dispositivos *hardwares* e ajuda-os a trocar dados de maneira confiável. Mais detalhes em <http://pt.wikipedia.org/wiki/Interface> (acessado em 12/08/2009).



## 1.2 - O *game* e a cena independente

Qual a primeira coisa que nos vem em mente quando pensamos em um *game* independente? Uma equipe produzindo um jogo em uma garagem ou um garoto produzindo um *game*, isoladamente, em seu quarto?

Na verdade, *game* independente é um conceito amplo que pode ser ilustrado de várias formas. Na presente pesquisa, o *game* independente será tratado com um projeto a ser desenvolvido sem aportes financeiros de grandes empresas. Independente, por assim dizer, é um *game* que é desenvolvido por uma pequena equipe, ou individualmente, por pura paixão sobre o assunto ou simplesmente pelo fato de querer um dia ganhar dinheiro e fazer carreira na área de criação e desenvolvimento de jogos digitais.

Nos dias de hoje, existem diversas iniciativas, inclusive de grandes empresas como a *Microsoft*<sup>7</sup>, que apostam na cena independente, criando, desenvolvendo e fornecendo ferramentas para desenvolvedores de *games* (de pequeno ou grande porte) a fim que de eles criem grandes sucessos com as soluções da empresa.

Voltando os olhos para o cenário nacional, os movimentos em direção ao desenvolvimento de jogos digitais no Brasil existem há mais de vinte anos, ou seja, diversas iniciativas que começaram independentes continuam independentes até hoje. Ora, porque nunca conseguiu atingir o estrelato, ou porque o cenário de um país como o Brasil ainda não propiciou oportunidades reais para os desenvolvedores independentes de *games*.

---

<sup>7</sup> De olho no mercado de desenvolvimento de *games*, a Microsoft lançou o *XNA*, um *framework* de desenvolvimento para criação de jogos para PCs com *Windows* e também para console *Xbox 360*.

O *Game Brasilis*, catálogo de jogos eletrônicos brasileiros, publicado em 2003 pela Editora Senac, traz informações sobre 32 jogos produzidos e comercializados no Brasil e no exterior. Interessante notar que grande parte dos *games* presentes no catálogo são compostos por equipes pequenas, isso quando não são desenvolvidos por uma única pessoa.

Lançado em agosto de 1983, *Amazônia*<sup>8</sup> é um jogo para computador criado e desenvolvido por Renato Degiovani. O jogo de aventura (*adventure*) demorou 3 meses para ser produzido e foi comercializado no Brasil, Portugal, Estados Unidos e África do Sul. *O Game Brasilis* traz uma sinopse do jogo *Amazônia* que ajuda a entender como tudo começou:

*Amazônia* foi um dos primeiros jogos publicados em revista técnica no Brasil. Grande exemplo do início da produção de jogos no país, nesta época os jogos ainda não eram vendidos em embalagens, eram encontrados em revistas. A programação vinha impressa e o usuário precisava digitá-la no seu computador e executá-la, para só então, jogar. *Amazônia* é um jogo de aventura de texto. A história se desenrola a partir dos comandos digitados pelo jogador. Cada comando dado dispara um evento relacionado, o objetivo é sobreviver a um acidente aéreo e escapar dos perigos da selva amazônica. O jogo já ganhou várias atualizações apesar dos seus mais de 20 anos, ainda está a venda no mercado, via internet.

Como se pode ver, Renato Degiovani<sup>9</sup> é o desbravador do mercado brasileiro de desenvolvimento de *games* e o pai dos jogos independentes no Brasil.

---

<sup>8</sup> O jogo *Amazônia* ainda é vendido nos dias de hoje e pode ser adquirido no site de Renato Degiovani em <http://www.tilt.net/amazonia>.

<sup>9</sup> Segundo a Wikipédia, Renato Degiovani é o primeiro projetista de jogos brasileiro a criar e produzir profissionalmente um jogo de computador em língua portuguesa, no início da década de 1980. Foi colaborador e diretor técnico da primeira revista brasileira de microcomputadores, a *Micro Sistemas*. Atualmente é editor e produtor do site TILT online, onde escreve artigos técnicos de progra-

Depois do jogo *Amazônia*, inúmeras iniciativas começaram a se desenvolver, tanto em pequenas empresas, quanto em computadores de apaixonados pelo tema. Foi em 1998 que um lançamento fez os jogos independentes feitos no Brasil serem comparados com os melhores *games* desenvolvidos em outras partes do mundo.



Figura 03 - *Amazônia*: game desenvolvido em 1983 por Renato Degiovani

O *Incidente em Varginha*<sup>10</sup>, um jogo de tiro em primeira pessoa, foi desenvolvido pelas empresas *Perceptum*<sup>11</sup> e pela *Cia Software* e comercializado no Brasil, Argentina, Europa e Ásia. O tempo de produção do *game* foi de 15 meses e contou com uma equipe de 6 pessoas.

Com uma ambientação muito realista, em *Incidente em Varginha* o jogador é convidado a desvendar o mistério do suposto aparecimento alienígena o-

---

mação e *design* de jogos, bem como cria os jogos comercializados pelo site. Mais detalhes podem ser encontrados em [http://pt.wikipedia.org/wiki/Renato\\_Degiovani](http://pt.wikipedia.org/wiki/Renato_Degiovani) (acessado em 03/02/2009).

<sup>10</sup> A versão de demonstração do *game* *Incidente em Varginha* pode ser baixa no seguinte endereço na web: <http://www.perceptum.com/indexp.htm> (acessado em 25/02/2009).

<sup>11</sup> Apesar do site da empresa *Perceptum* ainda estar ativo na internet, nota-se que nenhum novo projeto está em desenvolvimento. Mais detalhes em <http://www.perceptum.com>.

corrido na cidade mineira de Varginha. As missões do jogo se passam em locais conhecidos do Brasil como a Praça da Sé (SP), Baía de Guanabara (RJ) e São Tomé das Letras (MG) e além de procurar pistas nestes locais, o jogador precisa também fugir das forças inimigas.

Analizando a ficha técnica do jogo é fácil notar que, apesar de o *Incidente em Varginha* contar com 6 pessoas em sua equipe de desenvolvimento, nota-se que duas pessoas estavam presentes em todas as etapas de produção. São eles Marcos F. Cuzziol e Odair Gaspar.

Por ocasião do lançamento de o *Incidente em Varginha*, Renato Degiovani, criador do jogo *Amazônia* e editor do *Tilt Online*<sup>12</sup>, entrevistou Marcos F. Cuzziol em janeiro de 1999. Apesar de ser uma entrevista técnica, fica claro que a estrutura para o desenvolvimento do jogo se encaixa em um contexto de produção de um *game* independente.<sup>13</sup>

Apesar dos quase 10 anos dessa entrevista, o método até hoje no que diz respeito ao *modus operanti* do produtor independente: esforço contínuo de poucas pessoas. Fazendo uma analogia com o nascimento do microcomputador criado por Steve Wozniak e Steve Jobs<sup>14</sup>, seria o mesmo que desenvolver os próprios jogos na garagem de casa.

*Amazônia* e *Incidente em Varginha* são dois pontos marcantes na história do desenvolvimento de *games* independentes no Brasil. Contudo, são exemplos

---

<sup>12</sup> O primeiro site sobre desenvolvimento de *games* brasileiro, mantido por Renato Degiovani. Pode ser acessado no endereço: <http://www.tilt.net>.

<sup>13</sup> A entrevista citada pode ser lida na íntegra no Apêndice 1 desta dissertação.

<sup>14</sup> Steve Wozniak e Steve Jobs criaram a Apple Computers em 1976 na garagem da casa dos pais de Steve Jobs.

do século passado. Um jogo desenvolvido nos anos 1980 e outro no final dos anos 1990. De lá para cá muita coisa mudou no que diz respeito à tecnologia e o *Adobe Flash* e o *XNA* da *Microsoft*, hoje ferramentas fundamentais para os produtores independentes, ainda não existiam. Ser do século passado não quer dizer que jogos velhos são *games* ruins. A leitura aqui é do ponto de vista da evolução técnica das ferramentas de desenvolvimento.

Com o advento de novas ferramentas para o desenvolvimento de jogos digitais, é natural novos *games* surgirem periodicamente e os melhores ganhem as páginas da grande imprensa. E foi justamente isso que aconteceu com o *TriLinea*<sup>15</sup> game desenvolvido por brasileiros com o *XNA*, *framework* de desenvolvimento de jogos criado pela *Microsoft*.

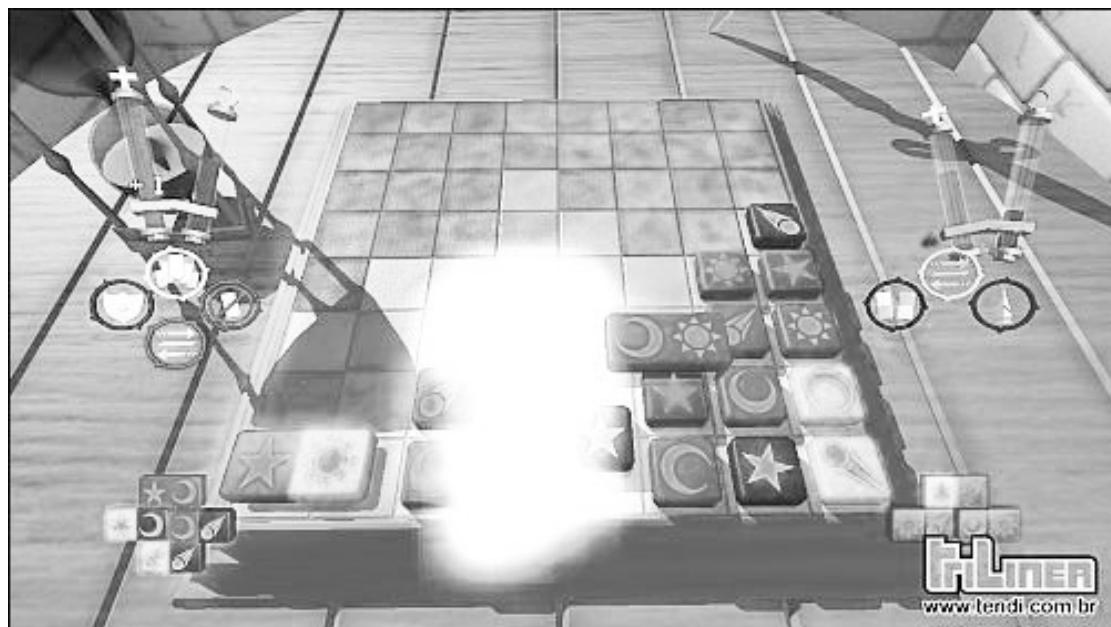


Figura 04 - *TriLinea*: game desenvolvido por brasileiros usando o *XNA* da *Microsoft*

---

<sup>15</sup> O jogo *Trilinea* pode ser encontrado em <http://www.tendi.com.br/trilinea/>

Criado por Davi da Silva Prata e Edson da Silva Prata Jr. e Renato Pelizzari, o jogo *TriLinea* teve uma boa aceitação na rede *Xbox Live*<sup>16</sup> da *Microsoft*, sendo apresentado inclusive em uma palestra oficial da empresa, o que impulsionou os *downloads* da versão de demonstração do jogo na rede. Até o presente momento a versão final do jogo ainda não foi oficialmente lançada para comercialização.

Quando olhamos para a história dos *games*, podemos observar que no início da história do desenvolvimento de jogos digitais, um *game* era criado praticamente por um único autor. Como um livro ou uma pintura, a história do desenvolvimento de jogos digitais, no seu início, era um trabalho autoral, quando no máximo, desenvolvido em duplas. Foi assim no nascimento do *videogame* e persiste até hoje, quando olhamos para os *games* independentes.

---

<sup>16</sup> *Xbox Live* é o serviço de jogos online do console de jogos *Xbox* e *Xbox 360* da *Microsoft*, no qual permite aos jogadores conectarem e jogarem entre si com muitas funcionalidades, tais como: download de bônus para jogos, criação de perfis, torneios e *chat* por voz. Mais informações podem ser encontradas em [http://pt.wikipedia.org/wiki/Xbox\\_Live](http://pt.wikipedia.org/wiki/Xbox_Live) (acessado em 10/10/2008).

### 1.3 - Rascunhos, argumento e organização de ideias

Ao jogar pela primeira vez *God of War*<sup>17</sup>, lançado pela Sony em 2003, não é difícil se deparar com a seguinte pergunta: como é que este jogo foi feito? Como conseguiram reunir tantas ideias boas em um único *game*? O roteiro parece de um filme!

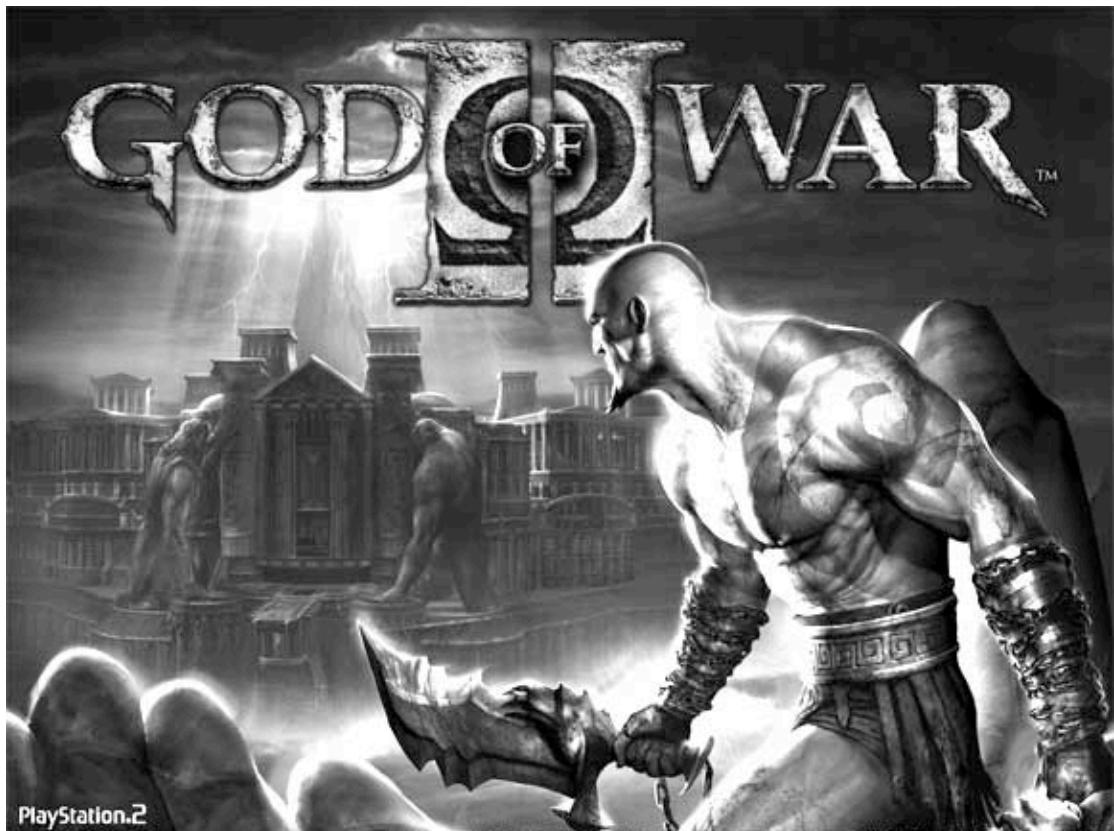


Figura 05 - *God of War*: considerado por muitos críticos de *games* uma obra de arte

O processo de criação de um *game* não difere de outros processos de criação, seja este processo coletivo, quando nos referimos a uma equipe, ou individual, quando uma única pessoa decide que fará um *game* do começo ao fim.

---

<sup>17</sup> *God of War*: [http://us.playstation.com/PS2/Games/God\\_of\\_War](http://us.playstation.com/PS2/Games/God_of_War) (acessado em 22/12/2008).

De acordo com Perucia *et al* (2005), o desenvolvimento de uma ideia de jogo geralmente surge de um pequeno conceito, que deve ser expandido com técnicas de *brainstorm*<sup>18</sup>. Muitas vezes bons conceitos para jogo surgem de pensamentos que em um primeiro momento pareciam ridículos. Por esse motivo, deve-se cuidar para não eliminar a ideia enquanto estiver pouco madura. Afinal, como um bombeiro poderia tornar-se herói em um mundo com tartarugas e cogumelos que andam? Se observar o bem-sucedido Mario<sup>19</sup>, poderá obter algumas respostas.

Mas que respostas são essas? De onde pode vir uma boa ideia para um *game*? Será que todas as boas ideias já não se transformaram em algum tipo de jogo digital? Certamente não. Basta observar os principais sites especializados em *games* para ver que todos os meses são lançados dezenas de novos títulos. Isso se levarmos em conta apenas os jogos oriundos das grandes indústrias, sem contar as inúmeras iniciativas independentes. Para Schuytema (2008), os *games* nascem de ideias simples. Mas como posso ter uma boa ideia simples? Como se dá esse processo?

Nós temos ideias a todo o momento: ao acordar, ao nos deslocar de um ponto para outro na cidade e, sobretudo, quando estamos tomando banho, não é? Como identificar uma boa ideia?

Young (1994) afirma que a produção de ideias é um processo tão consolidado como a produção de um carro; que corre por uma linha de montagem. Nessa produção a mente utiliza uma técnica operativa que pode ser aprendida e

---

<sup>18</sup> *Brainstorming* (tempestade cerebral) é uma atividade desenvolvida para explorar a potencialidade criativa de um indivíduo ou grupo, colocando-a a serviço de seus objetivos. Mais detalhes em: <http://pt.wikipedia.org/wiki/Brainstorming> (acessado em 22/02/2009).

<sup>19</sup> Super Mario Bros. Lançado pela Nintendo em 1985, é um caso de sucesso até os dias de hoje.

controlada, e seu uso efetivo é no fundo uma questão de prática e técnica, da mesma forma como no uso de qualquer ferramenta.

O famoso encanador Mario, protagonista da série de *games* mais bem sucedida da *Nintendo*, *Super Mario Bros.*, foi criado por Shigeru Miyamoto<sup>20</sup> em 1985. Hoje em dia, com o alto grau de evolução técnica dos *videogames*, podemos até dizer que o jogo *Super Mario Bros.* é um jogo simples, contudo, marcou época por diversos motivos, entre eles o de consolidar um personagem que transcendeu o mundo dos *videogames*. Um personagem simples, um jogo simples e sucesso inegável e absoluto.

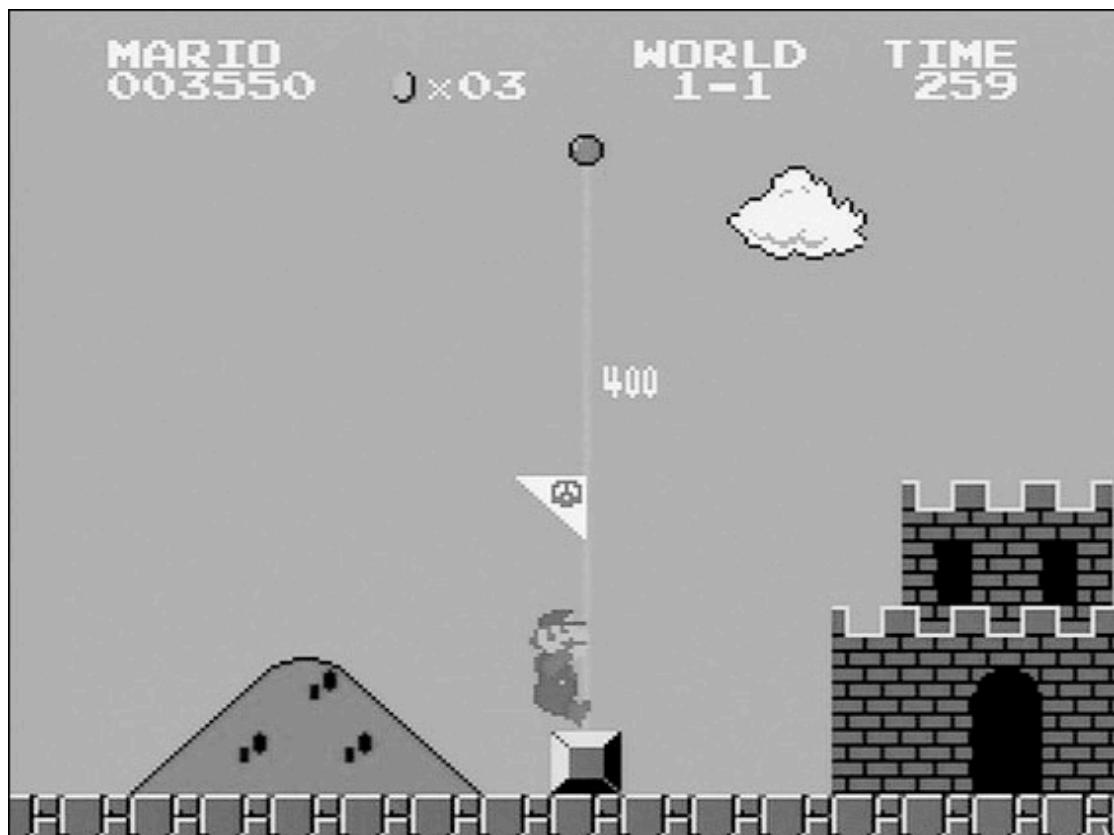


Figura 06 - *Mario Bros.*: uma ideia simples cercada por um mundo complexo criado em 1985

<sup>20</sup> Shigeru Miyamoto é uma das personalidades do mundo dos jogos mais idolatrados no Japão e no mundo. Como principal designer de jogos da *Nintendo*, ele foi responsável pelo nascimento de vários personagens mundialmente conhecidos como *Mario*, *Link* e *Yoshi*. Mais informações em: [http://pt.wikipedia.org/wiki/Shigeru\\_miyamoto](http://pt.wikipedia.org/wiki/Shigeru_miyamoto) (acessado em 22/02/2009).

Mas por onde passam as ideias simples? E como elas surgem? Segundo Young (1994), elas nascem a partir do abastecimento da mente com informações de qualquer natureza (aqui, podemos nos referir aos jogos digitais), e descreve, resumidamente, um método dividido em 5 passos:

- **Primeiro passo:** coletar material puro, ou seja, materiais referentes ao seu problema imediato e materiais adicionais para o enriquecimento cultural da ideia. Por exemplo, quando o criador do jogo organiza, em sua sala de trabalho, um grande Mapa de Parede no qual cola fotos, recorte, escreve ideias, desenha esboços e os organiza lógicamente por meio de setas-vínculos utilizando a técnica do mural de ideias;
- **Segundo passo:** trabalhar estes materiais em sua mente. Estude estes materiais e deixe seu cérebro absorver tudo o que puder sobre o assunto. O que significa estudar a fundo o problema;
- **Terceiro passo:** estágio de incubação. Nesta fase você deixa o problema de lado para se concentrar em outras tarefas. A mente consciente faz o trabalho de síntese do problema;
- **Quarto passo:** depois de um processo de amadurecimento, vem o nascimento real da ideia. Eureka!
- **Quinto passo:** o contorno final e desenvolvimento do conceito para uso prático. Aqui começa a formatação e a consolidação da ideia.

A partir destes pontos de vista e sugestões é fácil de entender que simplicidade é a palavra-chave quando se fala em organização de ideias para a produção de jogos digitais. E não é difícil entender a questão da simplicidade: quanto mais simples, maior o público a ser atingido pela ideia e logo em seguida pelo *game* produzido.

Após o nascimento da grande ideia, rascunhar os seus pontos centrais é fundamental. Santee (2005) nos aconselha a ordenar as ideias do jogo e passá-las ao papel<sup>21</sup> levantando detalhes, mas mantendo a simplicidade ao mesmo tempo, para que então sejam desenvolvidas. O que temos aqui é uma das características que todo bom designer de jogos que deve seguir à risca por todos, sejam amadores, profissionais.

Os esboços são de suma importância para criação de obras das mais diversas naturezas, tanto no caso dos *games* quanto da hipermídia. Notamos que, ao esboçar uma ideia, começa-se também a ser esboçado um roteiro. Gosciola (2003) nos diz que o esboço pode ser a criação de uma hipermídia, mas, certamente é um instrumento eficiente de criação do roteirista.

Como parte do processo de esboçar ideias, uma técnica interessante é a construção de mapas mentais. Um mapa mental<sup>22</sup> (ou *mind-map*) é um diagrama ou gráfico usado para gestão de informações, conhecimentos e para a compreensão e solução de problemas. De acordo com Leão (1999) fazer um mapa é construção: construção que evidencia as interconexões, os arranjos em pleno movimento, em metamorfose constante. Esse recurso é muito útil em processos de *brainstorm* e elaboração de conceitos.

---

<sup>21</sup> Apesar desta dissertação tratar de um produto de natureza digital, ou seja, o *game*, o papel ainda é o principal ponto de partida para criações. É no papel que se começa todo o planejamento de um jogo digital.

<sup>22</sup> Mapas mentais podem ser feitos diretamente em folhas de papel ou em softwares específicos que ajudam na organização de ideias, conceitos, cruzamentos, interconexões e vínculos dos mais diversos. Uma relação destes softwares auxiliares podem ser encontrados na Wikipédia no seguinte endereço: [http://pt.wikipedia.org/wiki/Mapa\\_mental](http://pt.wikipedia.org/wiki/Mapa_mental) (acessado em 22/02/2009).



Figura 07 - Mapa mental: técnica para organização de ideias e roteiros para games

Além das possibilidades citadas, temos o exemplo de Scott McCloud<sup>23</sup>, autor de diversos livros teóricos sobre histórias em quadrinhos, que aponta um caminho interessante para o desenvolvimento de novas ideias. McCloud (1994) aponta um caminho de seis passos, seguindo um método semelhante de Young (1994), para estruturar e dar forma a uma boa ideia. O autor foca suas técnicas em histórias em quadrinhos, mas que podem ser plenamente usadas quando falamos em jogos digitais. Ao invés de citá-lo simplesmente, reproduzo a seguir duas páginas de seu livro *Desvendando os Quadrinhos*.

---

<sup>23</sup> Além de teórico das Histórias em Quadrinhos, Scott McCloud também é desenhista e quadrinista. Seus trabalhos podem ser encontrados em <http://www.scottmccloud.com/>





É notório que, por trás de um grande *game*, existe uma boa ideia. Mas esta ideia não é nada sem uma sistematização para seu desenvolvimento transformando algo abstrato em algo concreto e real. No caso dos *games*, a formulação de uma ideia constitui em reunir subsídios e ordená-las a fim de transformar esta ideia inicial em um grande projeto de jogo.

Nos pontos apresentados por McCloud (1994), vemos que os passos apresentados podem e devem ser usados quando estamos começando a formatar uma ideia para um projeto de *game*. A primeira etapa diz respeito à ideia e objetivo. Qual é o teor do *game* a ser desenvolvido? Quais são os impulsos, filosofias e objetivos? É justamente neste começo que pode delinear a forma do *game*, que é a etapa seguinte proposta por McCloud. Qual será a forma do *game*? 3D ou 2D? Em idioma, que é a terceira etapa, nada mais natural que se produzir um *game* em seu idioma nativo e criar opções para outros idiomas, a fim de criar-se oportunidades de internacionalização do projeto. A etapa que se segue diz respeito à estrutura. Este ponto trata da composição. Fazendo um paralelo com o universo dos jogos digitais, notamos que a composição pode ser aplicada ao método de trabalho para seu desenvolvimento. A quinta etapa proposta diz respeito à habilidade, ou seja, como o desenvolvedor está preparado para sua criação. Está diretamente ligado ao conhecimento técnico e prático. A superfície, que é a última etapa, trata do produto final acabado, ou seja, o *game* propriamente dito, em seu estado final do jogo.



## 1.4 - Mecânica de jogo, gêneros e narrativa

Formatar e detalhar uma boa ideia compõe a primeira parte do longo processo de criação e desenvolvimento de um *game*. Entendemos por formatação a organização sistematizada de uma ideia com foco no desenvolvimento de um *game*.

O universo dos jogos digitais é composto por um sem número de *games*, dos mais variados estilos. Contudo, por mais diversos que sejam, cada jogo é classificado de acordo com um gênero. É interessante observar que por trás de cada gênero de jogo existe uma mecânica e um gênero narrativo que faz com que o *game* funcione como tal. Mas como podemos entender o que é mecânica de jogo? Para Sato (2009), compõe a mecânica de um jogo o seu conjunto de regras, possibilidades de ações e decisões, e variedades de respostas do sistema do *game*.

Villas Bôas (2005) faz uma análise interessante em sua classificação de tipos e gêneros de jogos como podemos ver na íntegra na tabela a seguir:

GÊNERO	DESCRIÇÃO	MECÂNICA	QUALIDADES	EXEMPLOS
Adventure	Jogos baseados em histórias, geralmente voltados em solucionar enigmas para seguir seu curso.	Nem sempre são em tempo real (híbrido com ação), usam mais o cérebro e menos a destreza e os reflexos.	Um mundo grande e complexo para se explorar, com PCs interessantes e uma boa história.	<i>Monkey Island</i>
Ação	Jogos em tempo real, nos quais o jogador deve responder com velocidade ao que está ocorrendo na tela.	Dominado pelos FPS ( <i>first person shooter</i> ), menos intelectual que jogos de <i>puzzles</i> , estratégia, adventure e outros.	Explosões de adrenalina e ação que exigem rápidas escolhas e bons reflexos.	<i>Half-Life</i>

<b>RPG</b>	Geralmente o jogador dirige um grupo de personagens em alguma missão, em diversas tramas e cenários	Grande universo do jogo com história não-linear. Sistemas de evolução dos poderes e forças dos personagens.	Gerenciamento dos personagens: escolha de equipamentos e armas; sistema de magias amplo e complexo.	<i>Ultima Online</i>
<b>Estratégia</b>	Gerenciamento de recursos para atingir um determinado objetivo. Esses recursos são geralmente usados para construção de unidades de combate.	Dois estilos: turn-based (onde o adversário esperas suas ações para agir, sendo o jogo 100% de estratégia mental / RTS - real time strategy (a ação ocorre simultânea, exigindo reflexos e agilidade no controle de tropas, além de estratégia)	O principal fator é o balanceamento das 'variáveis' - os recursos, como obtê-los, como adquirir unidades e qual o custo-benefício de cada uma delas.	<i>Civilization / Starcraft</i>
<b>Simuladores</b>	Jogos que simulam condições do mundo real, principalmente operações de máquinas complexas, como aviões ou carros.	Quanto mais 'sério' o simulador, mais próximo à realidade espera-se que seja. Jogos estilo 'arcade sims' são menos realistas, com controles simplificados.	O realismo e a precisão dos controles das máquinas e de sua operação são os principais fatores.	<i>Flight Simulator 2002</i>
<b>Esportes</b>	Jogos que representam os esportes 'reais' coletivos ou individuais.	Podem tanto simular o esporte pelo lado do atleta praticando o esporte, ou pelo lado do técnico gerenciando sua equipe.	Espera-se a completa reprodução das regras e das principais peculiaridades de cada esporte.	<i>FIFA 2002</i>
<b>Luta</b>	Jogos para dois jogadores onde cada um controla um personagem que usa uma combinação de movimentos e manobras para ataque e defesa contra o oponente.	Jogos geralmente de perspectiva lateral e de curta duração, com ampla variedade de personagens e manobras.	Têm um conjunto básico de ataques, defesas e contra-ataques de rápida aprendizagem, e um grupo de manobras e combinações mais complexas que exigem mais prática.	<i>Tekken 3</i>
<b>Casuais</b>	Adaptações dos jogos tradicionais como xadrez, gamão e paciência. Inclui também jogos dos shows de TV.	Jogos de interface simples, geralmente com uma baixa curva de aprendizagem.	Jogadores esperam que as regras sejam exatamente as mesmas dos jogos em suas versões 'reais'.	<i>Chess / Show do Milhão</i>

<b>'God' Games</b>	Também chamados de 'softwares toys', jogos que não possuem um real objetivo além do passatempo.	Geralmente não há critérios de vitória e derrota, ou de erros e acertos.	Como um brinquedo simples, espera-se apenas e simplesmente que seja divertido.	<i>The Sims</i>
<b>Educacionais</b>	Jogos cujo objetivo é ensinar enquanto se diverte jogando.	Geralmente voltados para o público infantil, utilizam uma estrutura semelhante à de desenhos animados.	O conteúdo deve ser muito bem elaborado em conjunto com especialistas, para que realmente atinja o objetivo de ensinar brincando.	<i>Coelho Sabido</i>
<b>Puzzle</b>	Jogos puramente voltados para o desafio intelectual na solução de problemas.	Esse gênero é de jogos de enigmas e problemas propriamente ditos, sem nenhum contexto de história, cenário ou outro objetivo além da solução de problemas.	Problemas de lógica, de matemática ou mesmo enigmas filosóficos são apreciados.	<i>The Incredible Machine</i>
<b>Online / Massive Multiplayer</b>	Jogos que podem ser de qualquer gênero anterior, com a diferença de ser jogado na internet.	Comunidades inteiras estão surgindo em torno desses jogos, que são desenvolvidos com o objetivo de favorecer o surgimento dessas comunidades.	Um gênero novo e com diversas características de jogabilidade ainda em fase de transformação.	<i>Everquest</i>

A tabela apresentada por Villas Boas (2005) mostra que o universo dos *games* é repleto por um grande número de gêneros, mecânicas e possibilidades narrativas que sem o apoio de uma boa ideia, de nada servem para criar um *game* original, e possivelmente de sucesso.

Tendo a narrativa como um dos elementos essenciais de um jogo digital, quais são os elementos desta narrativa? Sob o olhar da narrativa clássica (ou tradicional), o narrador de um acontecimento (ou história) é uma voz fictícia que descobre e narra os fatos não sendo o autor da história. Quando narrado em 1ª pessoa do singular, percebe-se que o narrador é um dos personagens e descreve sua versão da história; e quando descrito na 3ª pessoa do

singular, o narrador coloca-se ao lado do leitor com um simples expectador e pode ainda, perceber detalhes mais profundos dos personagens e ainda tem acesso aos seus pensamentos.

E para entender como funciona a narrativa clássica, devemos entender que a narrativa é composta por cinco partes:

- **Enredo:** conjunto sucessivos de fatos que podem ou não criar algum tipo de conflito. Segue a ordem início, meio e fim que, com a finalidade de impressionar o expectador (leitor ou interator) pode ser alterada;
- **Personagens:** elementos fictícios que estão presentes na obra enquanto esta é lida ou assistida. O personagem pode incorporar as mais diversas formas;
- **Espaço:** local onde é construído todo o cenário e os acontecimentos da história;
- **Tempo:** momento, cronológico ou psicológico, em que acontece o fato ou a história narrada;
- **Clímax:** momento da trama ou história onde aparece um momento de tensão e suspense criando expectativas para acontecimentos desencadeados no decorrer da história. Pode ser também um ponto de decisão.

Quando direcionamos nosso olhar para o universo do mundo dos jogos digitais, vemos que é justamente neste campo que a narrativa avança.

No campo da narrativa digital, os maiores esforços criativos e sucesso comercial têm se concentrado, até agora, na área dos jogos para computador. Muito desse empenho foi dedicado ao desenvolvimento de ambientes visuais mais elaborados e de tempos de reação mais rápidos; avanços que proporcionaram aos joga-

dores desafios mais variados à sua rapidez no gatilho contra oponentes visualmente mais convincentes. O conteúdo narrativo desses jogos é escasso, sendo frequentemente emprestado de outros meios ou suprido por personagens esquemáticos e estereotipados (MURRAY, 2003, pg. 61).

E para compreender melhor esta narrativa que nasce com os *games*, é preciso, de acordo com Gomes (2003) e Santaella (2007), levar em consideração os meios de interação que são usados pelo jogador. Nos jogos gráficos que, graças à sofisticação tecnológica cada vez mais acentuada, são hoje processados em animações tridimensionais, o jogador interage através de um avatar, uma personagem gráfica que o usuário escolhe e com a qual se identifica para representá-lo no interior do jogo. É essa identificação encarnada que se responsabiliza pela intensificação da competitividade e pelo envolvimento emocional e afetivo do jogador.

Esta transposição do real para o virtual, ou seja, a entrada no ambiente de *game*, não é mérito exclusivo dos jogos digitais. Para McLuhan (2005) o jogo é uma máquina que começa a funcionar só a partir do momento em que os participantes consentem em se transformar em bonecos temporariamente. Vemos que o ato de jogar, independente do suporte, vem acompanhado da entrega do indivíduo ao jogo. Uma entrega emocional.

E quando falamos de envolvimento emocional e afetivo em um *game*, a questão do som é fundamental. O projeto sonoro de um *game* trabalha a ambientação de cenários e situações que ajudam na imersão e participação do interator no *game*. Janet Murray ilustra bem a situação descrevendo o jogo *Myst*:

CD-ROM *Myst* (1993) deve muito do seu poder de imersão a seu sofisticado projeto de som. Cada uma das áreas dos jogo é caracterizada por um som ambiental

te distinto, como o assobio do vento por entre as árvores ou o rebentar das ondas no litoral, reforçando a realidade dos mundos imaginários, que são, na verdade, uma sucessão de imagens estáticas (MURRAY, 2003, pg. 63).



Figura 08 - *Myst*: game com sofisticado projeto sonoro

Estando um jogador imerso em um jogo digital e vivendo aquele momento de conflitos e decisões narrativas, cabe aqui perguntar: o jogador é também autor história? Murray (2003) no diz que a habilidade de se locomover por paisagens virtuais pode ser prazerosa em si mesma, independentemente do conteúdo dos espaços. Já Santaella (2007) afirma que o *game* é um mundo possível porque nele o jogador e o jogo são inseparáveis, um exercendo controle sobre o outro. É neste âmbito de controle de jogo que nasce da imersão, começando com a narrativa e o gênero do jogo. Este gênero, por sua vez trás consigo sua mecânica nativa.

## 1.5 - Papéis e personagens

Analisando atentamente a história dos jogos digitais<sup>24</sup>, vemos que os personagens quase sempre estiveram presentes no *games*. Desde *Spacewar!*, jogo criado em 1961 nos laboratórios do Massachusetts Institute of Technology<sup>25</sup> (MIT) por Steve Russell auxiliado por Dan Edwards, Alan Kotok, Peter Sampson e Martin Graetz, onde uma nave espacial tinha como missão derrotar seus inimigos, vemos que a existência de um protagonista já estava presente no universo narrativo dos criadores do jogo. Nesse caso, era uma nave espacial e seu piloto.



Figura 09 - *Spacewar!* Considerado um dos primeiros videogames da história

---

<sup>24</sup> Na Wikipédia pode ser encontrado uma excelente linha do tempo com a história dos videogames em português. - [http://pt.wikipedia.org/wiki/Hist%C3%B3ria\\_dos\\_videogames](http://pt.wikipedia.org/wiki/Hist%C3%B3ria_dos_videogames)

<sup>25</sup> [www.mit.edu](http://www.mit.edu)

Desenvolvido em um *PDP-1 (Programmable Data Processor)* da *Digital Equipment (DEC)*<sup>26</sup>, este equipamento era considerado um aparelho pequeno para a época, sendo mais ou menos do tamanho de um carro. O PDP-1 é um dos primeiros computadores a valorizar a interatividade com o usuário, tendo como acessórios um teclado semelhante a uma máquina de escrever e um monitor de vídeo, peças que foram fundamentais para a criação do primeiro *game* da história. O jogo de Russell não chegou ao grande público, pois nunca foi comercializado e ficou restrito a comunidade científica tendo feito muito sucesso entre os *nerds* da época. Contudo, vale ressaltar que em *Spacewar!*, o personagem central, a nave espacial, combate uma série de asteroídes inanimados, ou seja, é um *game* de um personagem só.

Para Prado e Stelko (2005) a história dos personagens em jogos começou no jogo *Space Invaders*<sup>27</sup> em 1978. Mesmo não tendo as atuais características e recursos dos personagens de hoje, o jogo possui um conjunto diversificado de inimigos.

O exemplo de *Space Invaders* nos ensina que um passo significativo para a criação de um personagem consiste na definição de sua necessidade dramática. Para isso, devemos perguntar: qual a função do personagem na história? Qual a função do personagem no jogo que estou desenvolvendo? Sabendo que, depois de formatar a ideia central de um jogo digital, o passo seguinte é a formatação de um roteiro preliminar (também conhecido como *story line*), antes de começar a produção propriamente dita; faz-se necessário entender então que o personagem deve ser parte integrante do roteiro.

---

<sup>26</sup> *Digital Equipment Corporation*: companhia dos Estados Unidos pioneira na indústria de computadores. Mais detalhes em: [http://pt.wikipedia.org/wiki/Digital\\_Equipment\\_Corporation](http://pt.wikipedia.org/wiki/Digital_Equipment_Corporation) (acessado em 25/02/2009).

<sup>27</sup> Por ser um jogo muito popular, não é difícil encontrar diversas releituras do *games* na internet. Mais detalhes em [http://pt.wikipedia.org/wiki/Space\\_Invaders](http://pt.wikipedia.org/wiki/Space_Invaders) (acessado em 22/02/2009).

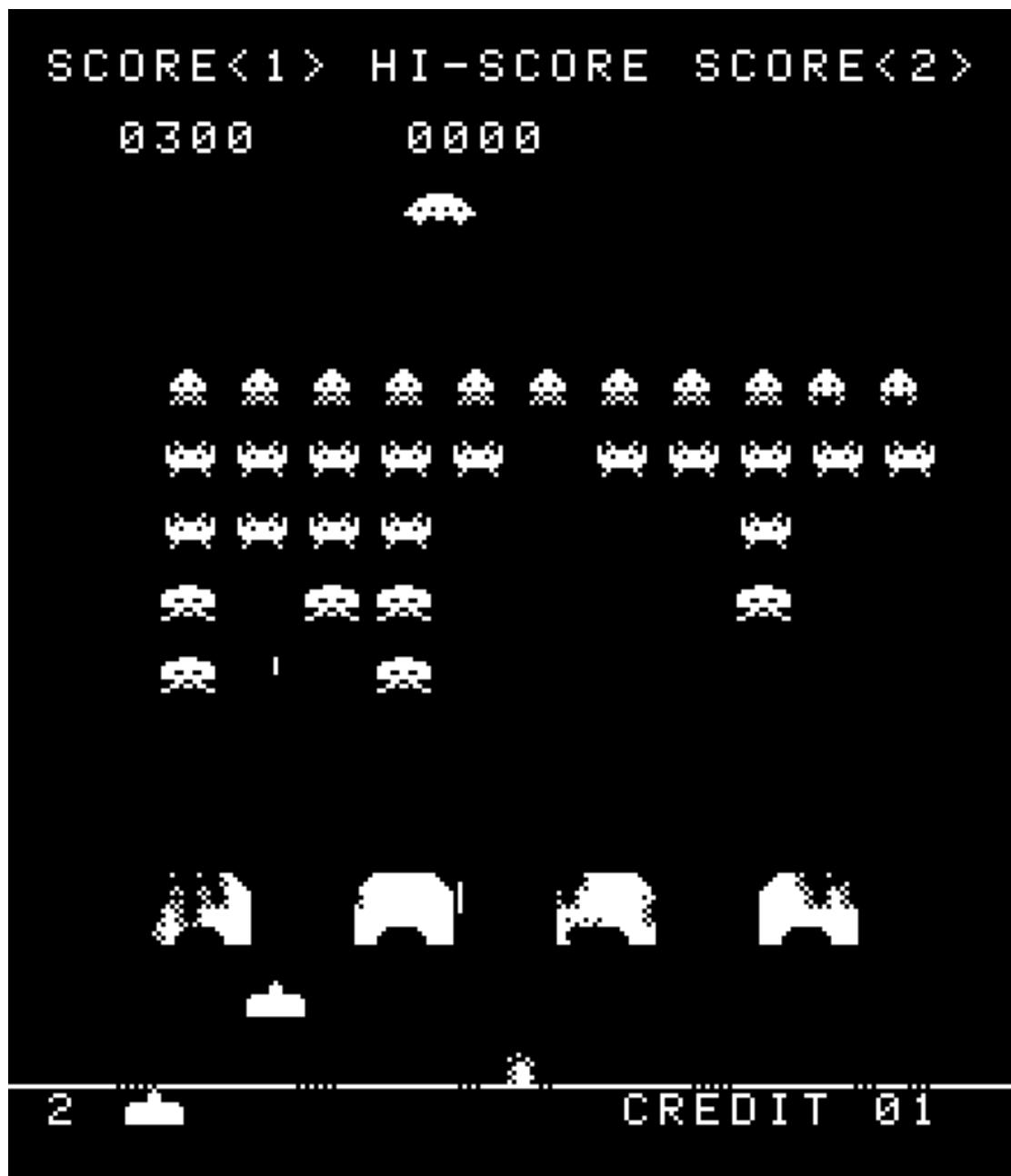


Figura 10 - *Space Invaders*: conjunto diversificado de inimigos

Para Field (1995), o personagem é o fundamento essencial de um (seu) roteiro. É o coração, alma e sistema nervoso de uma (sua) história. Antes de colocar uma palavra no papel, tem que se conhecer o personagem.

Segundo Rabelo (2005) a complexidade dos jogos atuais acostumou o usuário de jogos a passar por uma experiência de escape mental<sup>28</sup> durante o ato de jogar. O *game designer* (como são definidas as pessoas que projetam os *games*) deve pensar nas criações dos personagens com características psicológicas bem delimitadas de forma a construir um ambiente experimental em que o jogador acredite momentaneamente estar em diálogo ou imerso efetivamente no personagem. Expressões faciais, diálogos, vestimentas, movimentação devem ser desenvolvidos com base em peculiaridades para a formação das identidades dos personagens. Essa delimitação é fundamental para o sucesso de um novo *game*.

Todas as características dos um personagem precisam estar alinhadas com o universo criado para o *game*. Suas características físicas e psicológicas devem estar totalmente ligadas aos desdobramentos narrativos. Tudo isto, claro, em *games* que possuem personagens como elemento principal do jogo. *Puzzles*, geralmente, são jogos que não possuem um protagonista principal como em *games* de aventura, esportes ou tiro. Pensando na condição de *puzzles*, como observam Bairon e Petry (2000), no jogo, há um se deixar levar sem qualquer objetivo ou finalidade. Essa é a premissa fundamental do ato de jogar: todo jogo é um ser jogado.

Ao abordar as características fundamentais dos jogos digitais, comentar a história dos *games* independentes no Brasil e organizar a forma como deve ser pensada a argumentação narrativa, a organização de ideais, a mecânica de jogo, os gêneros de *game* e personagens o presente capítulo, que aqui termina, buscou organizar e a sistematizar um método para a criação e concepção de jogos digitais, nunca perdendo de vista o produtor independente de *games*. Esta organização introduz os meios pelos quais será pensada produção de jogos digitais no capítulo a seguir.

---

<sup>28</sup> O escape mental sugerido por Rabelo nada mais é do que a imersão.

## 2 - Planejamento de jogos digitais

**Fase 2.** Em sua missão exploratória no planeta repleto de *pixels*, a nave coletou informações gerais sobre o processo de criação e evolução do universo digital, lúdico e interativo. De posse das informações sobre o processo de criação, a exploração segue rumo ao planejamento. A jornada continua: no presente capítulo analisaremos temas relativos ao gerenciamento de projetos, *game design*, criação e *design* de personagens para *games*, níveis, roteiro e linguagens de programação.

### 2.1 - *Game design* e gerenciamento de projetos

Com as ideias já formuladas no capítulo 1 sobre: gênero, narrativa, personagens e outros itens esboçados em mãos, o próximo passo no processo de criação e desenvolvimento de *games* é agrupar todos os elementos que se tem em mãos em um único documento, conhecido como Documento de *Game Design* (*Game Design Document* - *GDD* em inglês)<sup>29</sup>.

Este documento é a espinha dorsal de todo e qualquer projeto de um *game*. É o que define todos os pontos de um jogo digital e guia todas as equipes envolvidas no processo de produção de um *game*. Como o foco deste trabalho é o *game* independente, é o *GDD* que guiará a produção de todo e qualquer *game* solo produzido.

O profissional responsável por esse documento e por todo o projeto de um jogo é chamado de *game designer*, aquele que fazer o *game design*. O que é o *game design*? Autores como Perucia *et al.* (2005) pensam o *game design* como o processo a partir do qual são descritas as características principais do jogo, como jogabilidade (palavra que provém ou deriva a expressão in-

---

<sup>29</sup> Um bom exemplo de um documento de *game design* (*GDD*) já formatado pode ser baixado em [http://www.runawaystudios.com/articles/chris\\_taylor\\_gdd.asp](http://www.runawaystudios.com/articles/chris_taylor_gdd.asp) (acessado em 25/02/2009).

glesa *gameplay*), controles, interfaces, personagens, armas, golpes, inimigos, fases e todos os aspectos gerais do projeto. Durante esta fase do desenvolvimento ou trabalho, é elaborado o *design document* (ou *game design document*), um documento que descreve as características citadas em detalhes, a saber o *design document* é inspirado na metodologia da produção cinematográfica, tendo como exemplo um roteiro de cinema. Com base nesse conjunto de informações, os artistas criam o visual do *game* e os programadores desenvolvem os recursos necessários para a interatividade do produto.

Além de ser baseado em um roteiro técnico de cinema, o documento de *game design* também é o documento de projeto de um jogo. De acordo com Barbaro (1983) pode-se definir o roteiro técnico (de um filme) como uma sistemática e ordenada tentativa de previsão do futuro filme em todos os seus detalhes; previsão que, praticamente, se concretiza num manuscrito que contém a descrição, sequência por sequência e plano por plano, das ações, dos diálogos, a indicação dos sons e da música de acompanhamento, e a solução de todos os problemas técnicos e artísticos que se apresentarão durante a realização do filme.

É no documento de *game design* que tudo se concentra antes da produção do jogo começar. Schuytema (2008) afirma que o este documento usa palavras, tabelas e diagramas para explicar o funcionamento de um *game*, a partir da história do mundo ficcional do *game* até a organização de botões em uma interface e ao modo como um arqueiro luta contra um espadachim.

O *game designer*, de acordo com Rabelo (2005), deve compreender o jogo não apenas como uma simulação da realidade objetiva, com intenção da imitação óbvia, mas como um simulacro que tende à transcendência da intenção secundária além da simples imitação. O *designer* de jogos deve aproveitar a expectativa do consumidor pelo simulacro oferecido pelos jogos de

forma em que objetos e seres passam a ter funções diferentes das expectativas objetivas.

Ao planejar o desenvolvimento de um *game*, este deve ser encarado como um produto e consequentemente, como um projeto. Aqui se faz necessário entender questões chaves sobre gerência de projetos.

No modo de ver de Cleland e Ireland (2002) um projeto consiste em uma combinação de recursos para criar algo que não existia anteriormente. Quando aplicada à questão de desenvolvimento de jogos independentes, essa combinação consiste na criação de um *game*, do início ao fim. Apesar de o *game* independente ser tratado como uma iniciativa individual nessa pesquisa, com o advento das redes de colaboração via internet, um jogo digital também pode ser produzido por um conjunto de indivíduos que usem a plataforma web para centralizar todas as informações necessárias ao desenvolvimento do projeto.

A gerência dos recursos de um projeto, independente de sua natureza, sendo ele individual ou coletivo, é executada por meio de um processo de administração composto por 5 etapas de acordo com a metodologia proposta por Cleland e Ireland (2002):

- **Planejamento:** desenvolvimento dos objetivos, metas e estratégias que proporcionem o compromisso de recursos para apoiar o projeto;
- **Organização:** identificação dos recursos humanos e materiais necessários, fornecendo uma distribuição adequada dos mesmos, e o estabelecimento de papéis individuais e coletivos dos membros das equipes dos projetos, que agem como um ponto focal para o emprego desses recursos;

- **Motivação:** o processo de estabelecimento de um sistema cultural que faça vir à tona o melhor que as pessoas pode fazer em seu projeto de trabalho;
- **Direção:** proporcionar a competência necessária de liderança para garantir a tomada e a execução de decisões que envolvem o projeto;
- **Controle:** monitoração, avaliação e controle do emprego de recursos no projeto que sejam coerentes com ele e com os planos do projeto.

Gerenciar um projeto de um *game* é gerenciar um projeto de uma hipermídia. E para gerenciar um projeto desta natureza, a melhor ferramenta que os designers de *games* podem utilizar é o documento de *game design*. Existem softwares que podem ajudar no processo de gerenciamento de projetos, como o *Mind Manager*<sup>30</sup>, *Microsoft Project* e também o *Open Proj*<sup>31</sup>. Este último possui mais funcionalidades que o *Microsoft Project* com a grande vantagem de ser um software código aberto e gratuito.

Ferramentas de gerenciamento de projetos ajudam no desmembramento das atividades relacionadas à produção de um produto (aqui discutimos a criação e produção de *games*), listagem de itens a serem criados, analisados e desenvolvidos, além de montar um cronograma com todas as etapas de produção. Também ajuda na definição de datas para a criação de um cronograma de trabalho bem como a divisão de tarefas entre os membros de uma mesma equipe de trabalho. Softwares desta natureza devem ser vistos, dentro de um projeto de produção de um *game*, como uma ferramenta auxiliar do documento de *game design*.

---

<sup>30</sup> O *Mind Manager*, além de ser uma ferramenta que auxilia no processo de construção de mapas mentais, é útil também em questões relacionadas ao gerenciamento de projetos. Mais detalhes do software em <http://www.mindjet.com>.

<sup>31</sup> O software *Open Proj* pode ser baixado gratuitamente no site <http://openproj.org>.

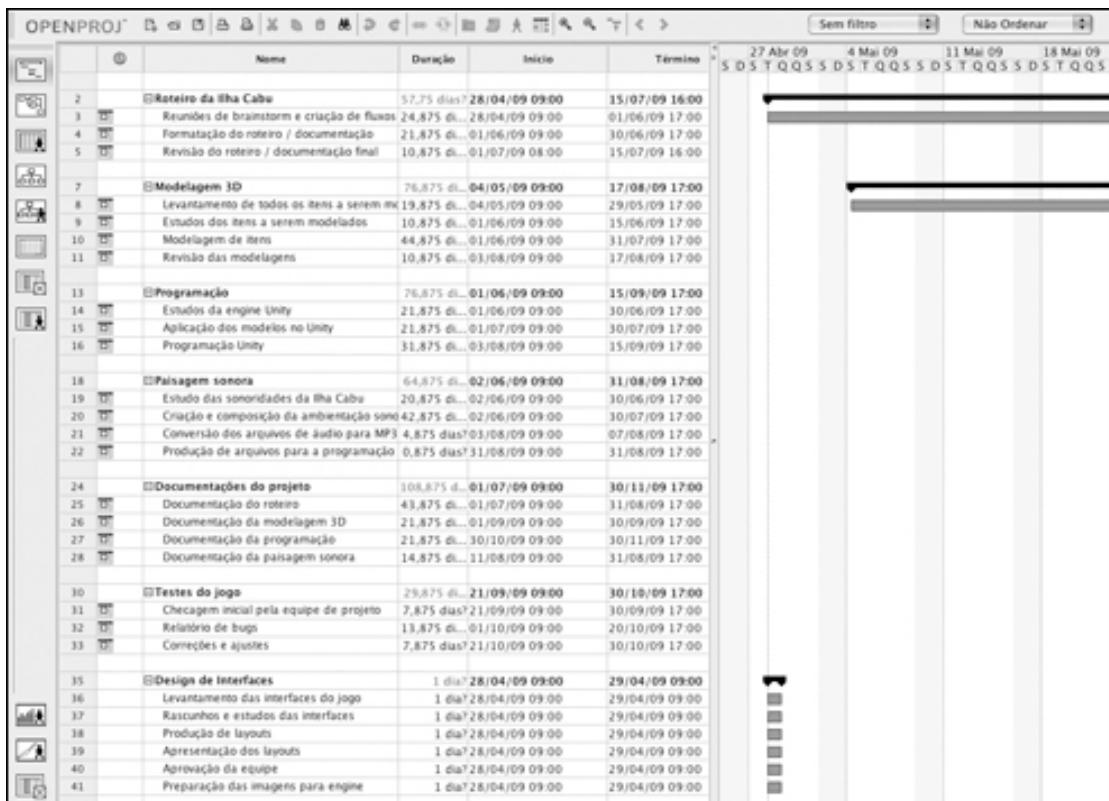


Figura 11 - Openproj: cronograma de trabalho de um projeto de desenvolvimento de jogo

Mas o que deve conter em um documento de *game design*? Quais os tópicos obrigatórios? Perucia *et al.* (2005) listam alguns itens que podem ser abordados em um documento de *game design* (*GDD*):

- Conceito:** nome do jogo; apresentação resumida do jogo; público-alvo; estilo de jogo; história; principais regras do jogo.
- Especificações técnicas:** hardware; sistema operacional; hardware mínimo; requerimentos de software; gráficos.
- Especificações do jogo:** número de fases; níveis de dificuldade; vidas; descrição dos tipos ou modos de jogo; sistema de pontuação; sistema de ranking (ou *high scores*); opções de configuração; número de jogadores; recursos de carga e gravação (*load* e *save*); sistema de câmera; personagens; itens de jogo; itens de cenário; tabela de itens evolução de fases; tabela de mensagens.

- **Dispositivos de entrada:** suporte para mouse; dispositivos de entrada para os menus; dispositivos de entrada do jogo; definição de teclas e botões.
- **Design gráfico e arte:** abertura; descrição de *layout* de menus e telas; descrição de *layout* do jogo na fase; definição de fases; definição do jogo.
- **Sonorização:** definição das músicas nos menus; definição das músicas nas fases; definição dos efeitos sonoros de menu e outros; definição dos efeitos sonoros de jogo (nas fases).
- **Desenvolvimento:** tempo de desenvolvimento; alocação de pessoal; metas.

O documento de *game design* (*GDD*) deve ser encarado com um documento de projeto. Quando estamos em um processo de criação e desenvolvimento de um *game*, nada mais natural do que ver tal processo sob o olhar do gerenciamento de projetos para atribuir questões profissionais a projetos individuais de caráter independente.

Dessa forma, a partir dos autores pesquisados, somos levados a visualizar que as principais características do jogo a ser projetado devem ser documentadas no documento de *game design*. De acordo com o ponto de vista de Santee (2005), tais informações devem ser acessíveis a todos os membros da equipe de desenvolvimento, como artistas e programadores, quanto a pessoas interessadas no projeto, mas que não estão envolvidas diretamente com a produção. Tais informações podem estar divididas em dois grandes grupos de categorias:

## Informações de *Design*

- **Tipo de jogo:** do que se trata o jogo. Ação em terceira ou primeira pessoa, RPG, estratégia, simulador de voo ou de automóveis, etc. Nessa parte do documento podem ser explicados os detalhes específicos do projeto, exemplos de como seriam determinados elementos e demais detalhes do *gameplay* (jogabilidade). Se o jogo apresenta alguma proposta de objetivo ou desafio ao jogador que o caracterize de modo geral, essas informações devem ser bem definidas;
- **Enredo e roteiro:** se o jogo possuir enredo e roteiro, devem ser documentados também. Normalmente, jogos com características de *role-playing game* possuem enredos que se desenvolvem no decorrer do jogo. Todos os acontecimentos e reviravoltas relevantes devem ser registrados.
- **Características gráficas do projeto:** como serão os cenários e o modo de renderização em geral. Aqui estariam inclusas informações como: “as luzes serão elementos-chave no jogo”, ou “os cenários e personagens do jogo terão estilo cartum, com características que lembram desenhos animados” etc;
- **Conceito:** arte conceitual do que precisa ser passado para a tela do computador. Tudo pode ser criado, desde objetos, armas e personagens à arquitetura, vegetação e ambientação do universo, planeta, país ou cidade onde se passa o jogo.

## Informações de Programação

- **Configuração mínima:** capacidade de processamento mínimo do computador em que o jogo deve rodar com bom desempenho. Isso inclui memória RAM e de vídeo, processador mínimo e o mais importante, modelo de placa de vídeo;
- **Tipos de mapa:** qual será o sistema de mapa que o jogo vai exigir. O que vai possuir relação direta com o tipo de jogo. Devem ser escolhi-

- dos tipos como ambientes externos com árvores e nuvens para simuladores de vôo, cenas pré-renderizadas para RPGs, sistema baseado em tiles 3D para jogos de ação/RPG e estratégia ou mapas mais elaborados para jogos de ação em primeira pessoa;
- **Propriedades de entidade:** aqui são especificadas as propriedades obrigatórias de cada entidade do jogo. Uma entidade pode ser um objeto, arma, personagem, inimigos, figurante e até todo o mapa. Por exemplo: uma arma deve possuir variáveis que especifiquem sua força, quantidade de balas no pente, máximo de munição e precisão. Já um inimigo pode possuir como propriedades obrigatórias resistência, velocidade de corrida, campo de visão e armadura. As propriedades contidas em cada entidade dependerão muito do que for definido em tipo de jogo.

Com todas estas informações, existe algo que ainda deva constar no documento de *game design*? Se todos estes itens estiverem contemplados, podemos dizer que o documento estará muito bem elaborado. Contudo, algo inerente à criação e não apenas à formatação técnica deve estar presente. Para Tavares e Neves (2006) um bom *game design* deve ser balanceado, ou seja, não pode ser muito fácil para que o jogador se entedie e perca o interesse nele, nem tão difícil a ponto do jogador se desmotivar e dessa maneira, desistir. O balanceamento não deve ser confundido com os ajustes de dificuldade particular de cada jogo, como fácil, médio, ou difícil.

O balanceamento deve ser independente dos níveis de dificuldade. Da mesma maneira, não confunda o balanceamento com a linguagem inerente de cada jogo. Caso nunca tenha jogado um jogo de estratégia em tempo real<sup>32</sup> (RTS), por

---

<sup>32</sup> Estratégia em tempo real (RTS, do inglês *real-time strategy*) é um tipo de estratégia de jogos de computador que ao contrário do antigo gênero de estratégia não é baseado em turnos para cada jogador, podendo assim jogarem todos ao mesmo tempo. Mais detalhes em [http://pt.wikipedia.org/wiki/Estratégia\\_em\\_tempo\\_real](http://pt.wikipedia.org/wiki/Estratégia_em_tempo_real) (acessado em 22/02/2009).

exemplo, primeiro habitue-se ao mecanismo comum a todos esses jogos, como as interfaces complexas e manipulação das diversas unidades. Para isso basta jogar um ou outro jogo mais conhecido, para depois da devida identificação desses elementos, partir para a análise do balanceamento deles. Um exemplo bastante ilustrativo de problemas no balanceamento são alguns RPGs que apresentam minigames ou *puzzles* dentro da história principal, cuja resolução é obrigatória para poder se avançar na narrativa (*Final Fantasy XI*, Square Enix, 2003). Embora a maioria das soluções desses puzzles possa ser encontrada no próprio jogo ou mesmo através de dicas na internet, muitas pessoas, além de desistirem, ficam tão desestimuladas que evitam jogos semelhantes. Outro exemplo menos comum é possível ser encontrado em uma classe de adultos que não tem mais velocidade ou coordenação motora suficientes para jogarem certos jogos que dependem de reflexos rápidos (TAVARES e NEVES, 2006, pg. 7).

O documento de *game design* pode ser criado de diversas formas. Geralmente são produzidos em *softwares* editores de texto, como *BrOffice.org*, *Microsoft Word* ou *Google Docs*<sup>33</sup>, sendo que este último permite que mais de um usuário, ao mesmo tempo, pode trabalhar em um documento. Além dos editores de textos tradicionais um documento de *game design* pode ser criado em um ambiente *wiki*, *blog* ou qualquer outra ferramenta online de colaboração. A colaboração permite que mais de uma cabeça trabalhe em um projeto simultaneamente e potencializa assim a capacidade de criação e organização do projeto.

---

<sup>33</sup> O *Google Docs* pode ser acessado em <http://docs.google.com/>



## 2.2 - Criação e *design* de personagens

A criação de personagem para um jogo digital inspira-se na mesma dinâmica da criação de personagens para TV, animação, cinema, histórias em quadrinhos e outros meios que também contam com a presença do personagem no produto final. O personagem é de suma importância no processo de construção de um *game*, pois nos jogos digitais, assim como em outras mídias, é um fator de lembrança marcante. De acordo com McCloud (1993), os criadores sabem que os personagens compõem um dos elementos fundamentais para garantir o envolvimento do público.

E como garantir este envolvimento com o público? Field (1995) apresenta as diretrizes de seu método usado na criação de personagens para o cinema e TV. Ele nos indica primeiramente, estabelecer o personagem principal. Em um segundo momento, somos levados a separar e elaborar os componentes da vida dele / dela em duas categorias básicas: interior e exterior. A vida interior de seu personagem acontece a partir do nascimento até o momento em que o filme (jogo) começa. É um processo que forma o personagem. A vida exterior do personagem acontece desde o momento em que o filme (jogo) começa até a conclusão da história. É um processo que revela o personagem.

A revelação nada mais é do que a personalidade do personagem que pode ser composta por uma grande variedade de características, desde físicas até psicológicas. Todas estas informações devem ser relacionadas em um documento chamado ficha de personagem. Para Gurgel e Padovani (2006) a ficha de personagem deve conter a maior quantidade possível de informações do mesmo antes de desenhá-lo. As informações irão variar de acordo com o tipo de personagem, sua finalidade e possíveis restrições ou limitações do projeto onde esteja sendo desenvolvido.

Trabalhar questões relativas à personalidade do personagem pode acontecer paralelamente à sua concepção visual. Ao começar o *design* e sua formatação visual, deve se levar em conta como estes personagens podem ser construídos de forma bidimensional (2D), como na animação tradicional ou tridimensional (3D) modelados com recursos de computação gráfica.

A opção entre o 2D e o 3D deve acontecer nos momentos iniciais de concepção do *game*, e o mesmo deve acontecer com o estilo que os personagens (ou personagens) assumirão, que podem ser realísticos ou estilizados. Os personagens realísticos, com um *design* mais próximo da realidade, são concebidos de acordo com sua natureza, mantendo proporções, características e forma. Sua proporção é o que faz a ligação visual com a realidade. Um personagem humanizado é medido pelo tamanho da cabeça, tendo sete ou oito cabeças de altura.

O Homem Vitruviano<sup>34</sup>, famoso desenho de Leonardo da Vinci<sup>35</sup>, apresenta a figura de um homem com aproximadamente 8 cabeças de altura. Esta imagem tornou-se então o modelo clássico no estudo do desenho anatomia.

---

<sup>34</sup> O Homem Vitruviano é um desenho que acompanhava as notas que Leonardo da Vinci fez por volta do ano 1490 em um dos seus diários. Descreve uma figura masculina desnuda separadamente e simultaneamente em duas posições sobrepostas com os braços inscritos num círculo e num quadrado. A cabeça é calculada como sendo um oitavo da altura total. Detalhes de todos os pontos de proporção e também a tradicional e famosa ilustração podem ser encontrados no endereço a seguir [http://pt.wikipedia.org/wiki/Homem\\_Vitruviano](http://pt.wikipedia.org/wiki/Homem_Vitruviano) e também site da Universidade de Stanford <http://leonardodavinci.stanford.edu/submissions/clabaugh/history/leonardo.html> (ambos acessados em 25/02/2009).

<sup>35</sup> Apesar de o famoso desenho ser feito por Leonardo da Vinci, o criador do conceito do homem vitruviano foi apresentado na obra Os dez livros da Arquitetura, de Marco Vitruvio Polião.

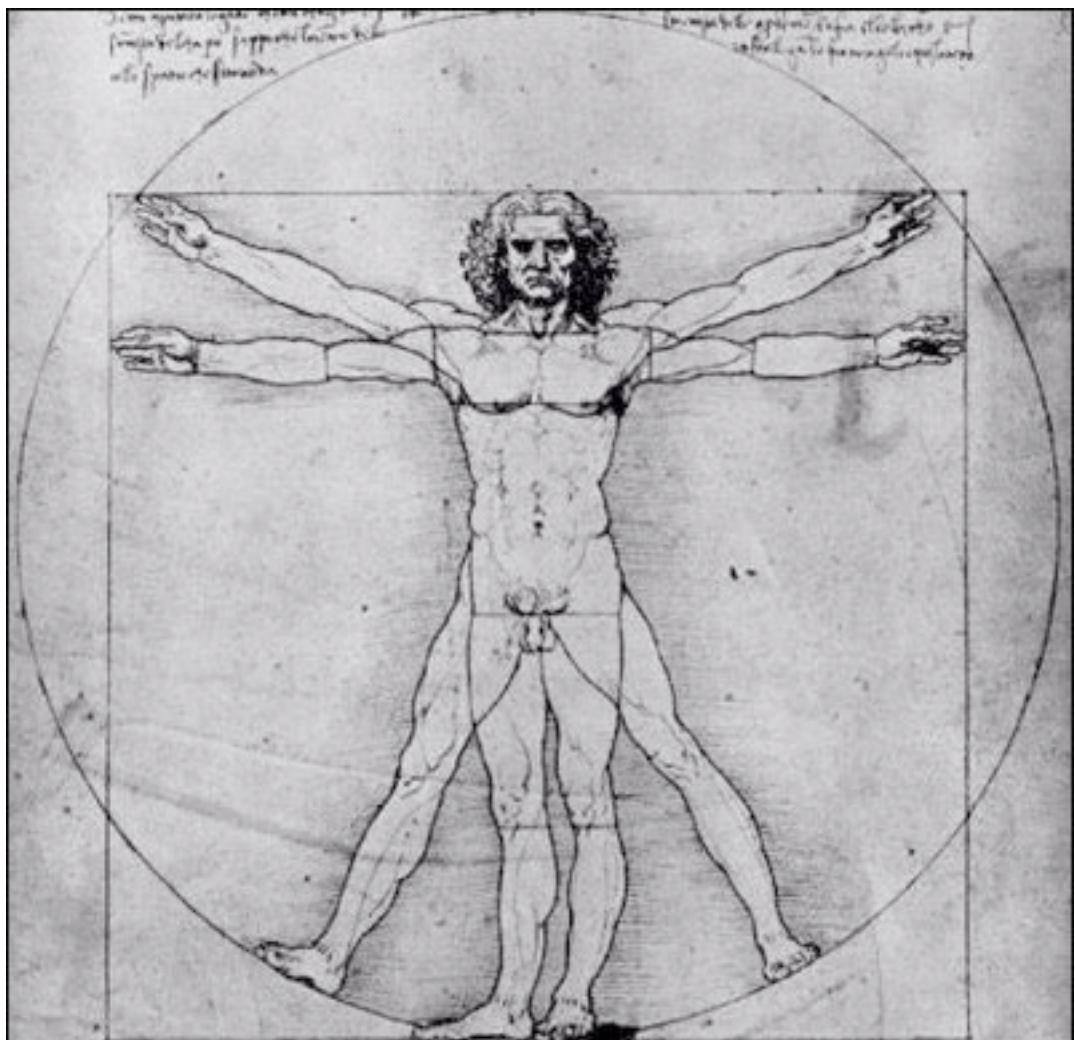


Figura 12 - O Homem Vitruviano de Leonardo da Vinci

Os personagens estilizados, por criarem uma ruptura com a realidade, não seguem a mesma estrutura de um personagem realístico, podendo assim ter qualquer proporção. Alguns possuem a altura de duas ou três cabeças e não é raro possuírem uma cabeça maior que o normal, o que torna o desenho mais agradável e divertido.

De acordo com Lopes Filho (2007), não existe uma regra que determine quantas cabeças um personagem deva ter. O corpo humano realista tem a proporção de 7 a 8 cabeças. Esta, talvez, seja a única proporção estabelecida para um tipo de personagem.

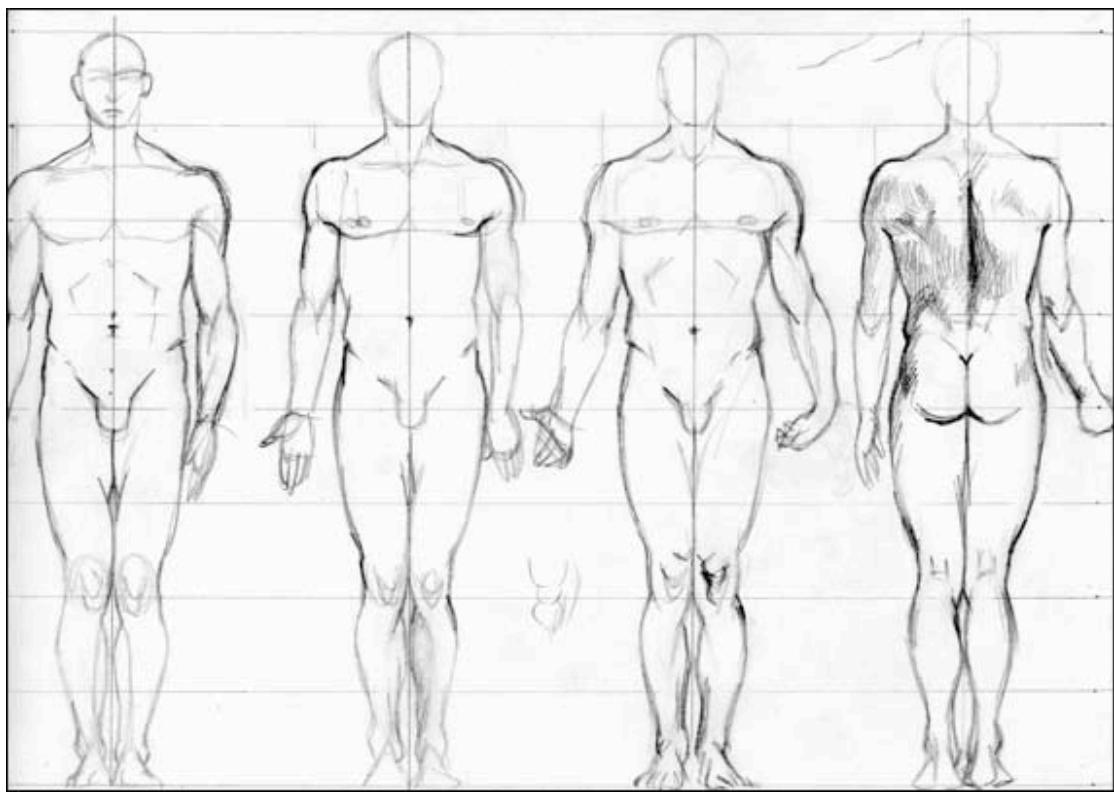


Figura 13 - Personagem humanizado com oito cabeças de altura

Sobre as possibilidades gráficas a serem exploradas, McCloud (1993) defende que o ser humano possui uma maior afinidade com personagens icônicos, ou seja, construídos à sua imagem e semelhança. Com a Ficha do Personagem em mãos, o próximo passo é começar uma pesquisa de referência para começar a dar forma ao personagem.

É justamente neste sentido que Prado e Stelko (2005) ressaltam que a pesquisa de referência é uma grande aliada em qualquer produção de personagem, pois com ela é possível conseguir uma grande quantidade de material sobre o tema que está sendo pesquisado. Mas onde buscar tais referências? Nos dias de hoje é impossível pensarmos em pesquisa de referência sem pensarmos na internet. Um site interessante que pode ajudar na caracterização de um personagem é o *All the Human Photo References and Textures*

*3D Artist Wants*<sup>36</sup>. O site é um grande repositório de imagens de referência que podem ajudar durante o processo de construção de um personagem, seja para um *game* ou para outro produto de natureza digital.



Figura 14 - Imagem do site *All the Human Photo References and Textures 3D Artist Wants*

Uma vez que já foi definida se o personagem será realístico ou caricato, uma grande pesquisa de referência foi feita e todas as informações importantes estão devidamente inseridas da ficha do personagem, chega o momento de começar a arte conceitual, também chamado de desenho conceitual, etapa do processo que é responsável por definir a forma visual do personagem.

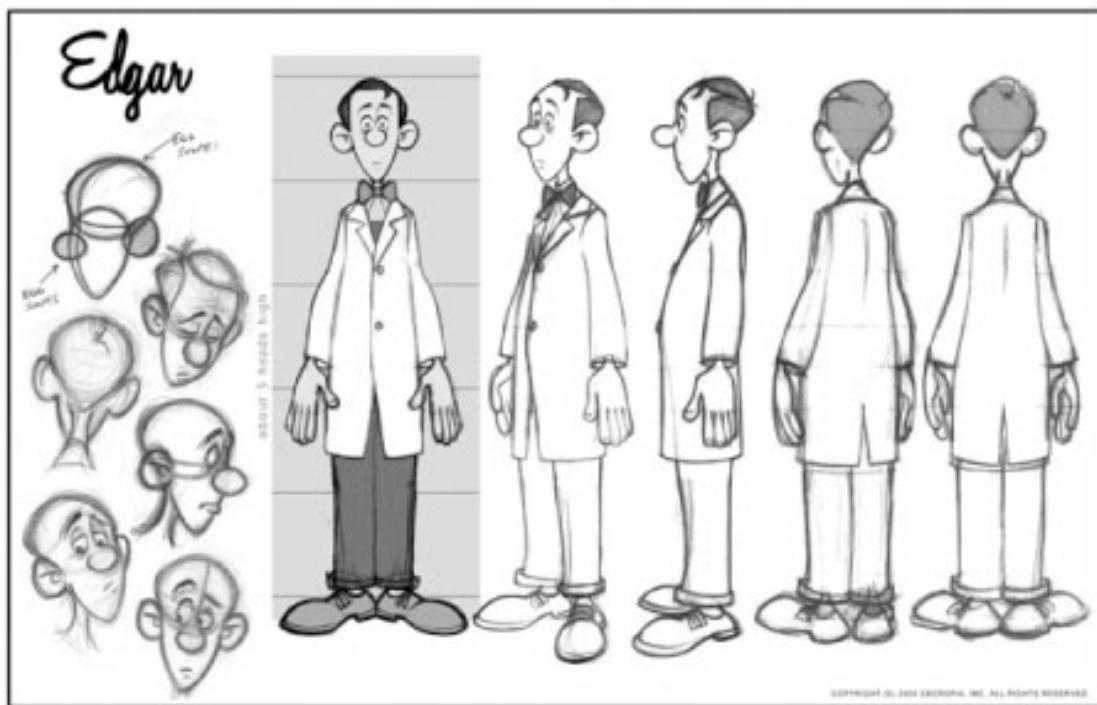
<sup>36</sup> Mais detalhes no endereço: <http://www.3d.sk> (acessado em 21/01/2009).

Para Prado e Stelko (2005), a arte conceitual de um personagem para *games* pode ser simples como um esboço ou complexo como uma escultura altamente detalhada. Servindo como um mapa, a arte conceitual do personagem é um projeto para um personagem 2D ou 3D bem-sucedido. Durante o processo, devem ser decididas quais as características físicas do personagem, texturas, vestimentas e cores. As decisões estão inter-relacionadas.

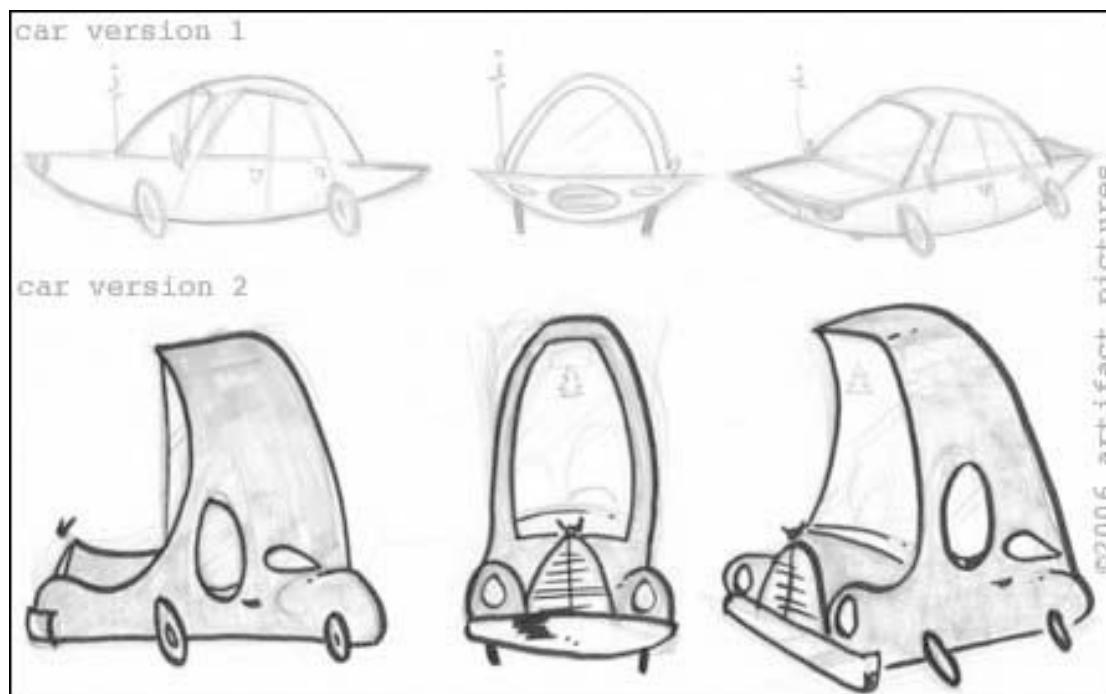
Segundo Blair (1995), os personagens não são meramente representações gráficas, mas também expressam uma personalidade. Esta personalidade é construída na fase inicial do processo, quando são definidas as questões relativas à parte psicológica do personagem.

Com todos os elementos reunidos, a próxima fase é construir um projeto técnico do personagem, chamado de *model sheet*. O *model sheet* geralmente é usado para apresentar, nas mais diversas posições, modos e feições o personagem que está sendo trabalhado. Contudo, o personagem também pode ser usado para um objeto qualquer como armas, acessórios, carros entre outros. Prado e Stelko (2005) ressaltam que o projeto técnico fornece a estrutura, proporção e *design* de cada personagem.

Em algumas ocasiões, quando diversos pontos do personagem já estão definidos, um *model sheet* pode ser construído diretamente em um *software* 3D, contudo, a recomendação de diversos especialistas é que tudo seja começado com lápis e papel. As ideias devem fluir de forma analógica antes de tomar a forma digital. Isso ajuda com que os estudos não sigam apenas o que já está pré-definido pela máquina. A mente trabalha melhor e a criatividade flui.



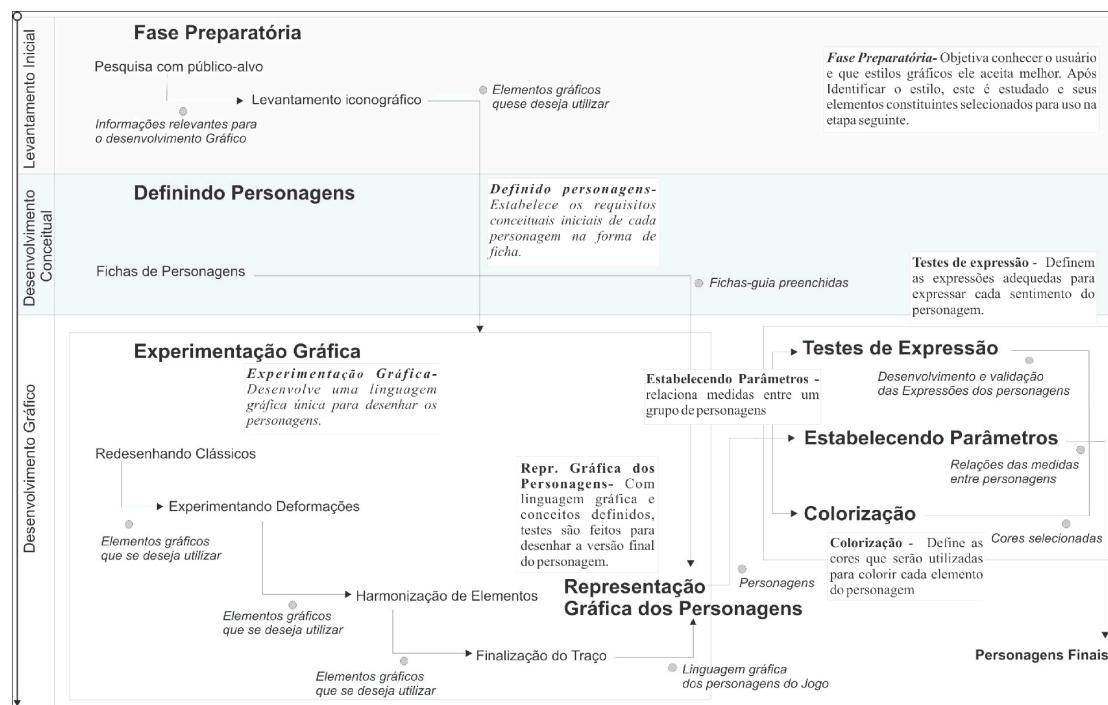
**Figura 15 - Model sheet de personagem caricato com cinco cabeças de altura**



**Figura 16 - Model sheet de um carro caricato (estilizado)**

A partir do exposto, vemos que, durante o processo de criação de um personagem, são necessárias duas etapas cruciais: a primeira é construir a Ficha do Personagem, com todos os requisitos apresentados e em seguida, o *model sheet*. São dois documentos que devem estar presentes no Documento de Game Design (GDD).

Gurgel e Padovani (2006) criaram um processo de construção de personagens para *games* que pode ser visto no gráfico a seguir. Muito útil para que o momento da criação e formatação do personagem não se perca no meio das diversas tarefas que compõem a criação de um *game*.



**Figura 17:** Fluxo de criação de personagem para *games*

O esquema apresentado acima define as etapas de criação de personagens para *games* desde a fase preparatória, que tem como objetivo conhecer o usuário do *game* (público-alvo) e definir o estilo gráfico a ser usado. O segundo passo é o desenvolvimento conceitual que estabelece os requisitos

conceituais de cada personagem na forma de ficha. O desenvolvimento gráfico é a fase final e tem como objetivo realizar experimentações gráficas para o desenvolvimento da linguagem visual dos personagens para assim passar para a representação gráfica dos personagens, testes de expressão, parâmetros de medida e colorização.



## 2.3 - Jogabilidade, níveis, progressão, dificuldades e enigmas

Ao começar o projeto de um *game*, devemos sempre nos perguntar: o que torna um *game* divertido? O que o público espera de um bom jogo? Ao fazer alguma destas perguntas para um grupo *gamers*, não será difícil escutar a seguinte resposta: “espero que o jogo tenha uma boa jogabilidade”.

Mas o que é a jogabilidade? Jogabilidade está diretamente ligada ao modo de jogar um *game* e é composta por um conjunto de características e componentes que influenciam na mecânica do *game* e nas reações do jogador. Também conhecido como *gameplay*, podemos dizer que jogabilidade é a habilidade de jogar. Schuytema (2008) afirma que *gameplay* é o que acontece entre o início e o final de um *game* - desde o momento em que você aprende quais são seus objetivos até atingir a vitória ou o fracasso final. Os desafios ao longo dessa jornada de *gameplay* e as pequenas e grandes vitórias de um jogador é que criam uma experiência lúdica instigante e emocionante.

Tendo em vista que a jogabilidade está diretamente ligada com a experiência do jogador, podemos nos perguntar: o que os jogadores querem? Perucia *et al.* (2005) sugerem uma lista com o que os jogadores realmente querem e de que gostam em um jogo:

- **Desafio:** é a verdadeira motivação de um jogo. Os desafios sempre servem como experiência de aprendizado e geram emoção ao serem superados;
- **Socializar:** jogos em geral (não só *videogames*, mas jogos de tabuleiros ou cartas) provocam uma experiência social com amigos ou família. *Videogames* oferecem opção *singleplayer* (um único jogador) e *multiplayer* (mais de um jogador) e ambos são sociáveis. Quantos já

não se reuniram em casa de amigos ou familiares para passar horas se divertindo com jogos?

- **Experiência solitária:** embora se oponha ao item anterior, jogadores também gostam de jogarem sozinhos ou quando estão desacompanhados, apenas para passar o tempo. É como ler um bom livro;
- **Respeito:** jogadores gostam de ganhar e obter, com isso, respeito. Quando estão entre os melhores do ranking de um jogo, sentem-se orgulhosos. Esses fatores criaram disputas em torno dos jogos;
- **Experiência emocional:** todos procuram sempre um tipo de emoção. Adrenalina e tensão no *Quake*. Suspense e medo no *7th Guest*. Heroísmo em um jogo de aventura como *Mário* ou *Sonic*, que tem de salvar o mundo dos vilões;
- **Fantasia:** jogadores querem escapar para uma realidade ou mundo diferente do nosso. O jogador poderá voar, atirar, mergulhar, salvar pessoas, matar alienígenas. Poderá aplicar ações proibidas em nosso mundo, como no *Grand Theft Auto (GTA)*, em que se pode ser um ladrão de carro e assassino. Quem não gostaria de ser um Homem-Aranha ou Batman e combater o crime? Ou comandar a melhor seleção de futebol do mundo ou seu time do coração em um campeonato?

Focalizando o lado técnico de produção de um *game*, jogabilidade, segundo Rabelo (2005), é o momento em que os programadores transformam toda a conceituação e arte do jogo em um ambiente interativo. Todas as possibilidades e coordenadas de jogo são determinadas nesta fase, assim como as formas de jogar de acordo com as possibilidades de controle do usuário.

A jogabilidade está diretamente ligada com o *design* de níveis (*level design*). Segundo Silva (2005) o *level design* é a construção dos cenários de um jogo e está relacionado tanto com artes gráficas quanto com programação e mesmo com *game design*. Artes porque trabalha com imagens, programação

porque depende de implementação em linguagem de programação e *game design* porque envolve diretamente a ideia do jogo.

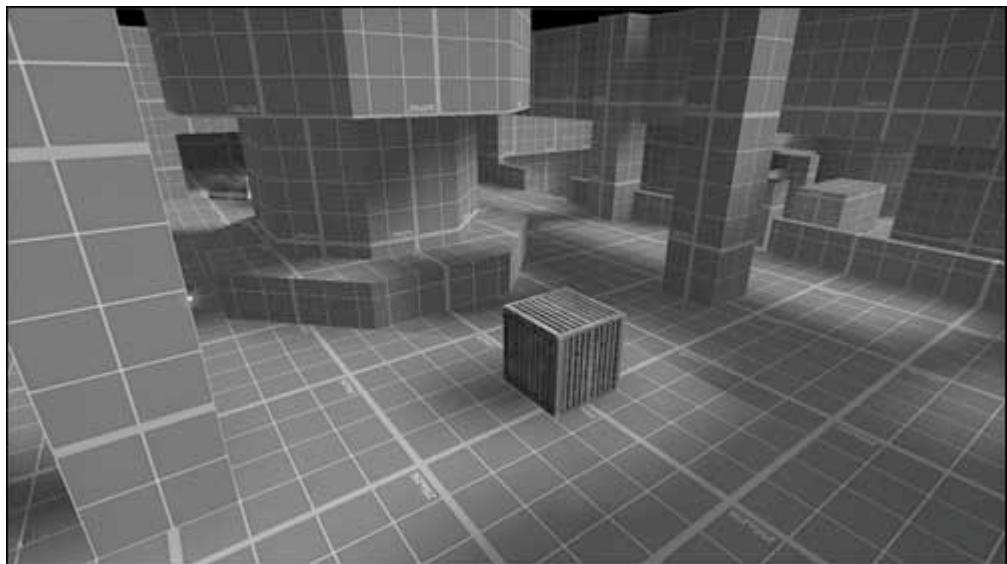
Entendendo que o *level design* é a construção de cenários e ambientes, podemos concluir que um cenário é construído sobre um mapa. Mas o que deve conter este cenário do jogo? Todos os itens de interatividade entre o jogador e o jogo. E para se desenvolver um mapa, são necessárias ferramentas específicas que disponibilizem recursos para sua construção. Tal qual o desenvolvimento e a construção de um personagem, um cenário passa pelas mesmas etapas, ou seja, rascunhos, organização de ideias e referências visuais.

Para Schuytema (2008) os ambientes de jogos são mais que apenas níveis estáticos repletos de detalhes arquitetônicos - eles são a arena na qual o jogo acontece - e, muitas vezes, os próprios ambientes orientam e conduzem a *gameplay* (jogabilidade). É essencial manter isso em foco à medida que um ambiente, mapa, cenário ou mundo é projetado. Schuytema ainda propõe maneiras de como pensar os ambientes dos jogos digitais como entidades dinâmicas que contribuem para a experiência total do jogador no *game* e lista uma série de recursos que ajudam na construção de um ambiente de jogo:

- **Arte conceitual:** use a arte conceitual para definir o clima e o fluxo dos ambientes de *games*;
- **Realização de um rascunho do seu mundo:** mesmo que não seja um grande artista, faça um rascunho antes da implementação;
- **Assegure a continuidade:** não se esqueça de como um único ambiente se encaixa no todo;
- **Guia de estilos:** em grandes *games*, crie um guia estilos de ambientes para os desenvolvedores;

- Dê significado ao clima:** use os ambientes ou níveis para dar significado ao clima do *game*;
- Liberdade de fachada:** faça que os jogadores pensem que tem opção, mesmo que não tenham;
- Luta por recursos:** sempre faça o jogador lutar por recursos;
- Concentre-se em um tema:** tenha um tema para cada nível que quer projetar;
- Arquitetura significativa:** os elementos arquitetônicos de um nível devem ser significativos e ter impacto;
- Conheça sua tecnologia:** ainda que não seja programador, conheça a tecnologia que será usada;
- Adote uma nova ideia:** tire vantagem de uma nova ideia como foco de um nível;
- Evite distrações sem sentido:** não envie o jogador em caminhadas sem sentido em áreas inúteis;
- Tire vantagem da revisitação:** revisitar o mesmo lugar a partir de outro ponto de observação é considerado um bom recurso de *level design*;
- Evite layouts sem originalidade:** não permita que seu nível caia em um padrão entediante e repetitivo;
- Pense em vários jogadores ou em um só:** quanto estiver projetando um nível, pense em como ele será usado no *game*.

A nossa pesquisa bibliográfica nos mostrou que um fator que sempre devemos ter presentes, o *level design* é parte do fundamento do *game design* e do processo de criação de produção de um jogo digital. Desenhar e projetar um nível é como projetar um jogo. Cada nível pode ser concebido como um *game* independente que interliga com o próximo para a construção do todo.



**Figura 18:** O projeto inicial de uma fase (*level*)

Jogabilidade e *level design* estão diretamente ligados aos enigmas, também chamados de *puzzles* ou minigames, que estão presentes e espalhados por diversos pontos de uma determinada fase. Tudo depende do projeto do jogo. Enigma, no contexto do jogo digital, é um (ou mais) desafio apresentado ao jogador no desenrolar de uma partida. Os enigmas estão diretamente ligados ao estilo e tipo de jogo, ou seja, um enigma apresentado em um jogo de tiro possui uma dinâmica diferente de um enigma presente em um RPG. De acordo com o ponto de vista de Schuytema (2005) a marca registrada de um enigma é a exigência de que a pessoa que queira resolvê-lo use suas habilidades de raciocínio para avaliar e determinar uma solução potencial para o desafio, e não simplesmente habilidades de reflexo físicas ou mentais.

Durante o processo de criação de um *game* e definição de enigmas, algumas perguntas devem ser feitas: o enigma é suficientemente atraente para o jogador? Como este enigma é resolvido? É simples ou fácil de resolver? Uma lista de apoio desenvolvida por Schuytema (2005) ajuda a responder estas perguntas:

- Faça um enigma se encaixar:** um enigma deve parecer natural no *game*, como se fizesse parte dele;
- Apóie a história:** um enigma deve amplificar a história ou tema do *game*;
- Faça que o enigma seja óbvio:** deixe claro que um obstáculo é, na verdade, um enigma a ser resolvido;
- Varie o escopo:** varie o escopo e a profundidade dos enigmas em um *game*;
- Concentre-se no “como”:** não faça os jogadores lutarem para saber o que fazer, mas sim em como fazê-lo;
- Mantenha um enigma ativo:** não permita que os jogadores esqueçam de que estão trabalhando para resolver um enigma;
- Misture atividades do jogador:** quando possível, misture enigmas com atividades que exigem habilidades de ação;
- Evite pressão artificial:** não crie pressão artificial para os jogadores quando eles estiverem resolvendo um enigma;
- Tome cuidado com enigmas cronometrados:** seja criterioso quando apresentar ao jogador enigmas cronometrados;
- Não mate o jogador:** não force a morte do jogador como um requisito para aprender a resolver um enigma;
- Faça que a solução pareça possível:** faça que a solução do enigma pareça possível para o jogador;
- Use microenigmas:** quando possível, apresente microenigmas que ensinam algo sobre enigmas maiores futuros no *game*;
- Não deixe o jogador perder algo vital:** não destrua a chance do jogador resolver um enigma porque ele perdeu uma pista ou peça vital do inventário anteriormente;
- O fracasso deve ser interessante:** se o jogador tem possibilidade de fracassar ao resolver um enigma, torne o fracasso interessante;
- Reinic peace o enigma:** se um enigma pode ser completado incorretamente, ofereça um modo sensato de reiniciar o *game*;

- **Evite enigmas binários:** não apresente ao jogador um enigma com um resultado imediato de sucesso ou fracasso.

Todo enigma deve ser bem definido pelo projetista do *game* (*game designer*) e deve ficar claro o que exige do jogador e documentado em detalhes de forma narrativa no documento de *game design*.

O conjunto destes elementos buscam o que Marcelo e Perscuite (2009) definem como condição de vitória, ou seja, um conjunto de objetivos que permitem finalizar um jogo; e devemos entender por objetivo os elementos necessários para alcançar a vitória em um *game*, como por exemplo, uma sequência de desafios, pois em um *game*, salvo raras exceções, além de jogar, todo jogador busca a vitória.



## 2.4 - Roteiro: formatando o documento de *game design*

Retomando: o que é um documento de *game design*? Ele é a espinha dorsal de todo e qualquer projeto de *game* e é também o aglutinador de todos os documentos adjacentes que poderão nascer durante a produção de um jogo digital. É nele que estarão todas as instruções relativas ao roteiro, personagem, cenário, níveis, arte, *design*, programação e até detalhes comerciais quando necessário. Estes documentos podem assumir diversas formas, desde documentos formatados em um editor de textos, como o *Microsoft Word* ou *OpenOffice.org* até *wikis* e *blogs*.

Formatar um documento de *game design* é dar forma ao projeto de um *game*, ou seja, torná-lo comprehensível aos outros membros de sua equipe. O primeiro passo, segundo Schuytema (2005), é delinear o conceito inicial em um breve documento de visão geral que explica os princípios mais importantes do *game* e alguns de seus principais recursos. Este conceito inicial é acompanhado do *high concept*, uma sentença simples - ou duas - que descreve a essência de um *game*. Com frequência, é assim que o *design* de um *game* começa - pode ser a semente a partir da qual o *game* se desenvolverá.

Documentos desta natureza seguem uma estrutura que podem variar de acordo com o projeto, mas em geral, acaba apresentado o mesmo conteúdo, ou seja, o que é básico e determinante para a produção de um jogo digital. A produção destes documentos é baseada em equipes médias de desenvolvimento de *games*, mas que podem ser plenamente adaptadas ao produtor independente. Tavares e Neves (2006) adaptaram a estrutura de uma equipe de desenvolvimento de Rolling e Morris (2004) apresentada a seguir:

<b>Gerenciamento e design</b>
<i>Game Designer</i>
<i>Level Designer</i> (Designer de níveis ou fases)
<i>Character Designer</i> (Designer de Personagens)
Gerente de Projeto
Gerente de <i>Software</i>
<b>Programação</b>
<i>Lead programmer</i> (Programador de conceito e condutor do projeto)
Programadores
<b>Arte visual</b>
<i>Lead artist</i> (Artista de conceito)
Artistas visuais (modeladores, ilustradores etc)
<b>Música</b>
Músico
Efeitos sonoros e diálogos
Programador de áudio
<b>Controle de qualidade</b>
<i>Q.A. Lead</i> (Condutor de controle de qualidade)
<i>Q.A. Technicians</i> (Controladores de qualidade)
<i>Playtesters</i> (Jogadores avaliadores)
<b>Outros</b>
Especialistas em outras áreas (educadores, consultores etc.)
Técnicos em áreas diretamente relacionadas (captura de movimento, roteiristas etc.)

Analizando a tabela apresentada vemos que uma equipe de desenvolvimento de um *game* é uma equipe multidisciplinar. Esta multi-disciplinaridade deve estar presente no documento de *game design* (*GDD*), documento este que deve contemplar todas as etapas de desenvolvimento de um *game* em todos os seus aspectos, sejam eles de natureza narrativa, de programação, de *design*, de música ou de qualquer outra.

O documento de *game design*, dependendo do autor, recebe outras denominações, como *gamedoc*, *game design bible* ou *gamespec* (especificações de *game*). De acordo com Assis (2007), o *gamedoc* é o documento que ancora

todas as ideias que deverão nortear gráficos, interfaces, ações, diálogos e premiações. Um típico documento de *game design* descreve quais características distintivas do *game* (gênero, habilidades que se esperam do jogador); qual a sua mecânica; quais as interfaces usadas e esboços, os mais detalhados que for possível neste estágio, dos ambientes em que o ação acontecerá.

Durante o processo de confecção de um documento desta natureza, o principal fator é transpor e ordenar as ideias e passá-las para o papel. Devemos sempre lembrar que o papel, seja ele uma folha em branco ou de caderno, é a melhor forma de começar a construir um documento de *game design*. Os rascunhos podem ser de detalhes do *game* ou simplesmente sentenças descriptivas do projeto.

De acordo com Marcelo e Pescuite (2009) é nesta fase do projeto que são elaborados rascunhos dos vários elementos definidos nas fases anteriores como personagens, cenários, objetos do jogo, componentes de ambientes (plantas, objetos, veículos etc.), menus, telas de configuração, apresentação e *storyboards*<sup>37</sup>. A arte deve ser trabalhada até que consiga expor de maneira clara toda a personalidade dos elementos que compõe o jogo, em coerência com a identidade visual do projeto.

---

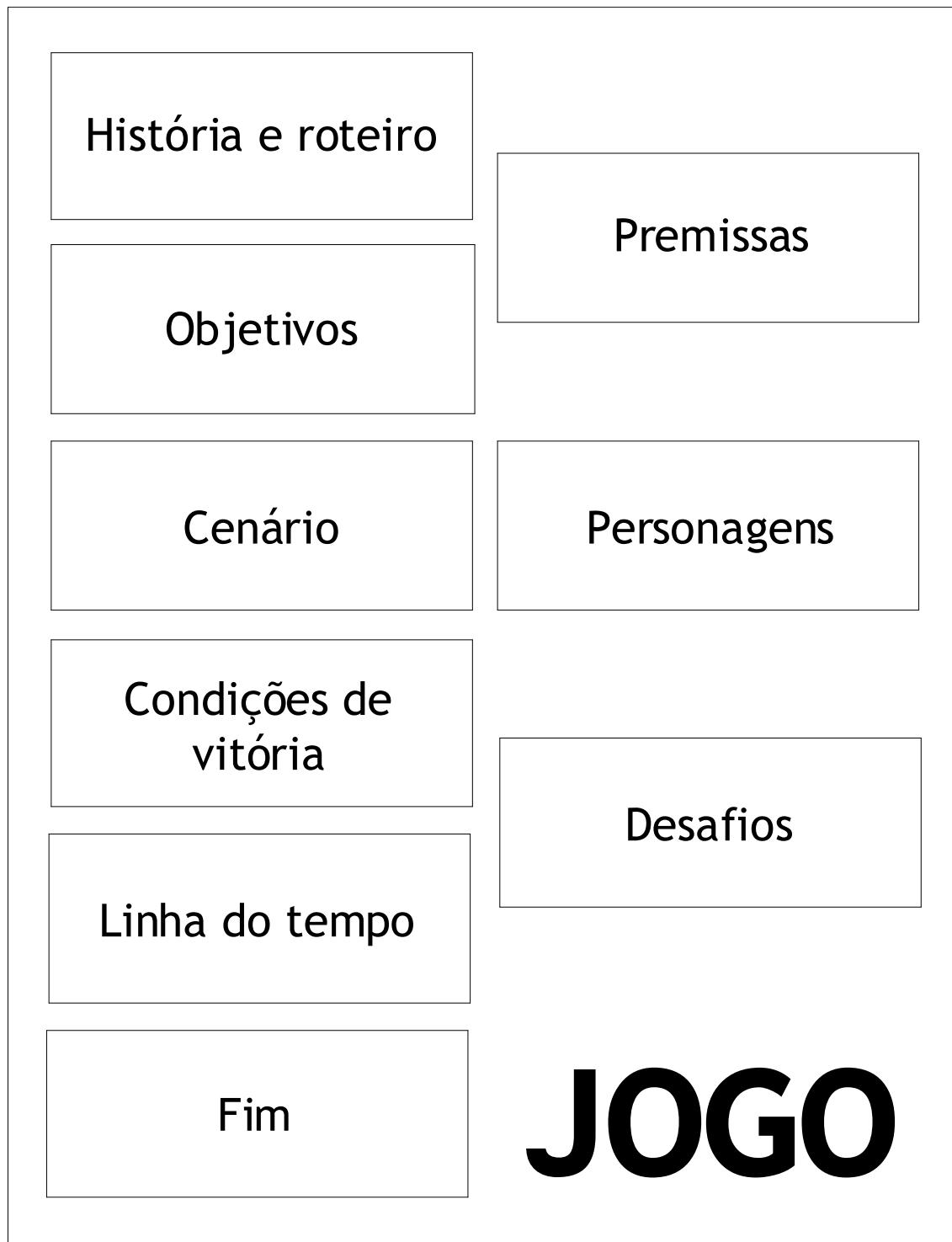
<sup>37</sup> *Storyboard* é um projeto de uma sequência de cenas cinematográficas muito utilizado na publicidade, animação, cinema e *games*. À primeira vista um *Storyboard* parece uma história em quadrinhos. Apesar de o *Storyboard* não ser uma HQ propriamente dita, por não possuir balões nem se destinar à reprodução, reserva as características de divisão de ação em quadros. Mais detalhes pode ser obtidos em <http://pt.wikipedia.org/wiki/Storyboard> (acessado em 21/07/2009)



**Figura 19:** Rascunhos: o começo de um documento de *game design*

A partir destes estudos iniciais constrói-se então a estrutura inicial de um jogo; um elemento essencial que dá as linhas gerais do *game* e consequentemente a estruturação do documento de projeto, aqui chamada de documento de *game design*.

Marcelo e Pescuite (2009) propõe que no momento em que o jogo é pensado deve-se então criar uma estruturação inicial e apresentam o gráfico que ajuda a entender como se dá uma estruturação inicial de jogo.



**Figura 20:** Estrutura de como pensar um *game* proposta por Marcelo e Pescuite

Um fator importante e determinante apontado por Assis (2007) é que não basta que o documento detalhe todas as características técnicas dos *games*,

é preciso que também haja argumentação. Com esta afirmação fica claro uma detalhe: é preciso haver uma fio condutor, uma história a ser contada no *game*. A vivência da narrativa.

No **apêndice 2** desta dissertação apresento um modelo de documento de *game design* desenvolvido por Elinaldo Azevedo<sup>38</sup> que pode servir de modelo para o desenvolvimento de qualquer *game* independente.

---

<sup>38</sup> O documento de *game design* desenvolvido por Azevedo está disponível em no endereço [http://br.geocities.com/elinaldoazevedo3/arquivos/DesignBible\\_Elinaldo.doc](http://br.geocities.com/elinaldoazevedo3/arquivos/DesignBible_Elinaldo.doc) (acessado em 21/07/2009).

## 2.5 - Linguagem de programação e bibliotecas

No processo de construção de um jogo digital, em um determinado momento do projeto, será necessário determinar a linguagem de programação a ser usada no desenvolvimento do *game*. De acordo com Diverio e Menezes (2000), linguagem é um conceito fundamental no estudo da Teoria da Computação, pois trata-se de uma forma precisa de expressar problemas, permitindo um desenvolvimento formal ao estudo da computabilidade.

Os autores citados definem programa como um conjunto estruturado de instruções que capacitam uma máquina a aplicar sucessivamente certas operações básicas e testes sobre os dados iniciais fornecidos, com o objetivo de transformar estes dados. Assim, podemos entender que a máquina, ou seja, o computador, faz exatamente tudo o que quisermos, desde que as orientações sejam de fato, bem explicadas:

A magia de um computador está em sua capacidade de se tornar quase qualquer coisa que você possa imaginar, desde que você explique exatamente o quê. O problema é aplicar o que você quer. Com a programação certa, um computador pode se transformar em um teatro, um instrumento musical, um livro de referência um adversário no jogo de xadrez. Nenhuma outra entidade no mundo, exceto o ser humano, possui uma natureza tão adaptável e universal. Em essência, todas essas funções são implementadas pelos blocos lógicos booleanos e máquinas de estado finito. (HILLS, 2000, pg. 47).

A linguagem de programação pode estar incorporada ao *engine*<sup>39</sup> usado ou não. *Engine*, ou motor de jogo, são softwares que simplificam a construção

---

<sup>39</sup> *Engine*, *game engine* ou motor de jogo é um software composto por um conjunto de bibliotecas, que ajudam a simplificar e abstrair o desenvolvimento de jogos para *videogames* ou computadores rodando sistemas operacionais. A funcionalidade tipicamente apresentadas por um motor de jogo

de um jogo. Contudo, apesar da grande diversidade de *engines* presentes hoje mercado, as linguagens de programação pouco se alteram.

No ambiente de desenvolvimento de *games*, existem um grande número de linguagens disponíveis muitas bibliotecas de funções, denominadas *API's*<sup>40</sup> de programação. Marcelo e Pescuite (2009) apontam algumas linguagens comuns no ambiente de desenvolvimento de *games*, como veremos a seguir:

- **C++:** um grande número de jogos foram desenvolvidos em C++, como *Doom* e *Quake* esta linguagem continua sendo a base de muitos títulos. É uma linguagem complexa, orientada a objetos e requer grande dedicação por parte do desenvolvedor.
- **Java:** linguagem de média complexidade, orientada a objetivos e desenvolvida pela *Sun Microsystems*<sup>41</sup> na década de 1990. Em função de sua grande popularidade, existe grande suporte e muitas referências no mercado e principalmente na internet.
- **ActionScript:** com a popularização da internet, a *Adobe Flash* se tornou a principal ferramenta de desenvolvimento de sites multimídia e jogos para a *web*. O *ActionScript* é uma linguagem orientada a objetos e já está em sua versão 3.0. Ideal para jogos *games* casuais desenvolvidos especialmente e a internet.

---

incluir: um motor gráfico para renderizar gráficos 2D e/ou 3D, um motor de física para simular a física ou simplesmente para fazer detecção de colisão, suporte a animação, sons, inteligência artificial, *networking*, gerência de memória, gerência de arquivos, gerência de linha de execução, um gráfico de cena e, suporte a uma linguagem de *script*. Mais detalhes sobre motor de jogos com exemplos comentados em [http://pt.wikipedia.org/wiki/Motor\\_de\\_jogo](http://pt.wikipedia.org/wiki/Motor_de_jogo) (acessado em 15/06/2009).

<sup>40</sup> API, de *Application Programming Interface* (ou Interface de Programação de Aplicativos) é um conjunto de rotinas e padrões estabelecidos por um *software* para a utilização das suas funcionalidades por programas aplicativos, isto é: programas que não querem envolver-se em detalhes da implementação do *software*, mas apenas usar seus serviços disponíveis. Mais detalhes em <http://pt.wikipedia.org/wiki/API> (acessado em 21/07/2009).

<sup>41</sup> Mais detalhes em <http://www.sun.com>

- **OpenGL**: o *OpenGL* não é uma linguagem e sim um conjunto de diversas funções que fornecem acesso aos recursos do *hardware* de vídeo. É uma *API* livre muito utilizada em computação gráfica para o desenvolvimento de aplicações gráficas em ambientes 3D, e sobretudo, em *games* de pequenas produtoras e também *games* independentes.
- **DirectX**: como o *OpenGL*, o *DirectX* não é uma linguagem e sim um conjunto de *APIs* que padronizam a comunicação entre *software* e *hardware*, fazendo a interpretação das instruções gráficas, sobretudo em tarefas relacionadas a programação de jogos para o sistema operacional *Windows*, da *Microsoft*.
- **Lua<sup>42</sup>**: linguagem de programação interpretada, de *script*, leve projetada para expandir aplicações. Criada em 1993 na PUC-Rio, já foi utilizada pela *LucasArts<sup>43</sup>* no *game Escape from Monkey Island<sup>44</sup>*.

O produtor independente, ao escolher uma linguagem de trabalho, precisa, antes de tudo, estudar o assunto enquanto trabalha no seu projeto de *game*. Começar a desenvolver um *game* ao mesmo tempo em que aprender uma linguagem não é recomendado em nenhum momento por um simples motivo: a medida em que se aprende a desenvolver em determinada linguagem, o conhecimento torna a resolução de problemas mais fácil e por consequência, evita-se o re-trabalho. Deve-se então aprender e estudar a linguagem de programação adequada para depois começar a programar.

Sistematizar as principais questões relacionadas ao planejamento de jogos digitais como o *game design* e o gerenciamento de projetos, a criação e o *design* de personagens, progressão, dificuldade, jogabilidade, roteiro apli-

---

<sup>42</sup> Mais detalhes sobre a linguagem em <http://www.lua.org/>

<sup>43</sup> Site oficial da Lucas Arts: <http://www.lucasarts.com/>

<sup>44</sup> Outras informações sobre o em [http://pt.wikipedia.org/wiki/Monkey\\_Island](http://pt.wikipedia.org/wiki/Monkey_Island)

cado ao universo dos *games*, linguagens de programação e bibliotecas de software foi o objetivo do capítulo que termina aqui.

Estas questões são fundamentais para o planejamento de *games* e constroem a base conceitual e prática para o desenvolvimento de jogos digitais, o tema do próximo capítulo desta dissertação.

.

## 3 - Desenvolvimento de jogos digitais

**Fase 3.** O final pode significar o começo da missão: nossa nave exploratória agora coletou informações gerais sobre o processo de criação e evolução do universo digital, lúdico e interativo, gerenciamento de projetos, *game design*, criação e *design* de personagens para *games*, níveis, roteiro e linguagens de programação. O presente capítulo trata da fase final de exploração, ou seja, o desenvolvimento de *games*, onde serão tratados temas como interfaces para *games*, ambientação sonora, modelagem 3D e motor de jogo.

### 3.1 - Interfaces para *games*

Antes de pensar sobre *design* de interfaces para *games*, temos que levantar a seguinte questão: o que é interface? A interface gráfica, e as novas formas de interação homem-máquina, foram decisivas para a apropriação social dos micro-computadores. A noção de interface gráfica (a manipulação de ícones pelo intermédio de um apontador - o *mouse*) foi popularizada com o *Apple Macintosh*. O objetivo do *Macintosh* era trazer ao grande público um sistema de manipulação de informações de fácil manuseio fazendo analogias com os objetos do nosso dia a dia (pastas, arquivos, lixeiras). O *Macintosh*, através de sua interface gráfica, instaura um diálogo entre o homem e o computador de forma quase orgânica (LEMOS, 2002, pg. 117).

Segundo o Dicionário Houaiss (2001), o termo interface significa: elemento que proporciona uma ligação física ou lógica entre dois sistemas ou partes de um sistema que não poderiam estar conectados diretamente. Meio pelo qual um usuário interage com um programa ou sistema operacional que emprega ou não recursos gráficos. Para Radfahrer (2000) a interface gráfica é o ponto de contato de um ser humano com uma máquina. O pesquisador Lev Manovich<sup>45</sup> afirma que em um ambiente digital, a interface é aquele ele-

---

<sup>45</sup> Os textos e estudos de Lev Manovich podem ser encontrados em <http://www.manovich.net/>.

mento que separa dois espaços absolutamente diferentes e que de alguma maneira coexistem.

Esta coexistência se dá pelo contato entre o seu humano e a máquina. No universo dos *games*, é o ambiente gráfico do produto digital (o jogo), o canal de comunicação do usuário final com o conteúdo, roteiro do *game*, história, *puzzles* e todos os elementos presentes em um jogo digital. É na interface gráfica de usuário que acontece a relação do jogador com o sistema, sendo a interface o elemento que faz a transição entre o real e o digital.

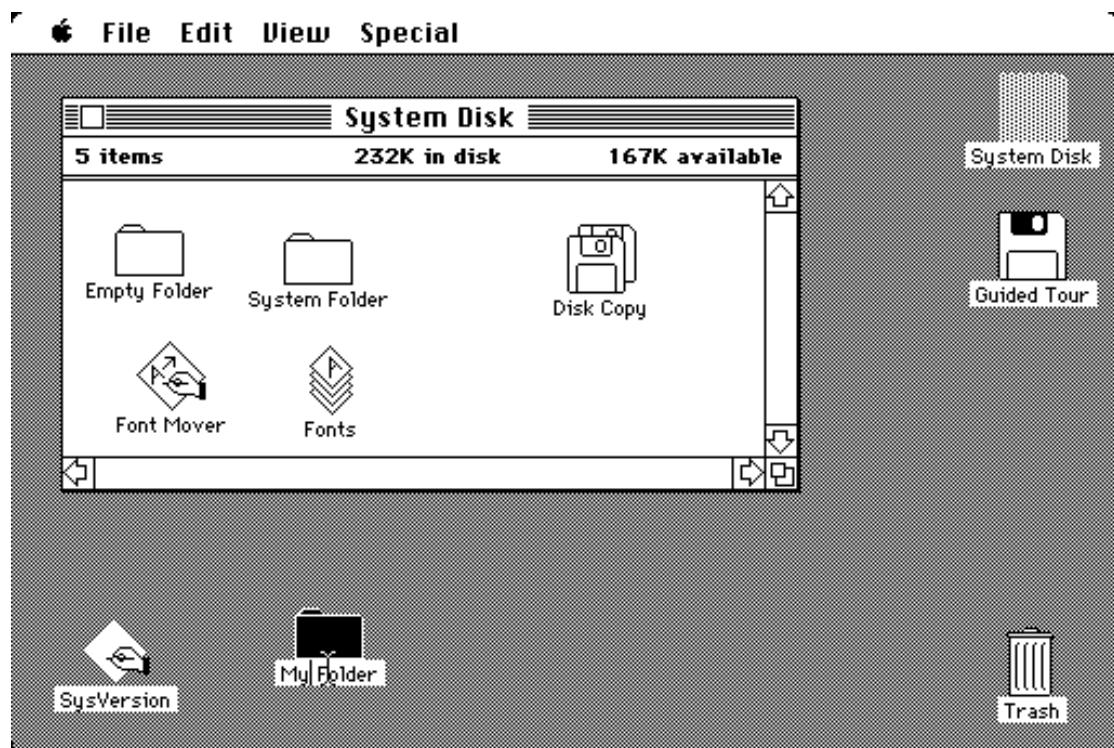


Figura 21: Interface de uma das primeiras versões do sistema operacional do *Macintosh*

De acordo com Bonsiepe (1997), a interface é o domínio do acoplamento estrutural entre ferramenta e usuário. No âmbito do *game*, pode-se afirmar que é o interface é o elemento responsável pelo acoplamento estrutural en-

tre jogo e jogador. Bonsiepe (1997) também analisa que a interface é um meio que:

- pode frustrar e irritar;
- pode facilitar ou dificultar a aprendizagem;
- pode ser divertida ou chata;
- pode revelar relações entre informações ou deixas-las confusas;
- pode abrir ou excluir possibilidades de ação efetiva-instrumental e comunicativa.

A interface gráfica de usuário (*GUI - graphic user interface*) é o meio pelo qual o *game* se comunica com o jogador e vice-versa; e também um sistema de navegação para orientação do jogador dentro do *game*. Esta navegação precisa ser clara e objetiva para que se minimize a frustração do jogador em se perder pelos caminhos da interface. É fundamental neste ponto ter a noção de que tudo está baseado em um sistema de navegação, que segundo Leão (1999), é a arte de encontrar um caminho que leve de um local a outro.

Este importante elemento de conexão homem-máquina pode melhorar a experiência de jogo ou frustrar os jogadores a tal ponto que eles abandonem o *game*; e não importa como é a interface, o objetivo é sempre o mesmo, ou seja, fornecer informações vitais para o jogador. Schuytema (2000) analisa diversos pontos relacionados à interface para *games* que são discutidos e apresentados a seguir:

- **A interface deve agir conforme o esperado:** uma interface de *game* é essencialmente visual e seus elementos devem funcionar de modo óbvio para o jogador. A criação de ícones ajuda na compreensão de ações. Não há necessidade de criar ícones diferentes e criativos para

tarefas óbvias com o objetivo de exibir elementos artísticos. Como a interface é o ponto de contato com o *game*, é fundamental que os jogadores tenham uma ideia clara do que fazem os botões, menus e controles mais óbviros;

- **A interface deve permanecer coerente:** o jogador, depois que descobre como algo funciona, quer sempre que este “algo” funcione da mesma maneira. Um ponto importante a ser observador em um projeto de interface para *games* é não confundir o jogador apresentando diferentes maneiras de fazer essencialmente a mesma coisa em diferentes partes do *game*.
- **A interface não deve pedir ao jogador para se lembrar de algo:** interfaces complexas devem lembrar o jogador do que ele fez por último e apresentar esse valores quando o jogador acessar novamente na interface. Não force o jogador digitar um valor que foi mostrado em uma interface anterior. Se for apresentado uma tarefa em uma interface nova ou complexa ao jogador, faça com que a tela de ajuda apareça apenas com um único clique. Simplifique.
- **A interface deve informar ao jogador situação do mundo do game:** sendo os ambientes de *games* muito dinâmicos, normalmente o jogador é capaz de entender a situação do mundo do *game* a partir da área da tela principal. Contudo, em determinadas situações, essa informação pode não ser tão evidente ou facilmente representada apenas por imagens do mundo do *game*. Nesses casos, a interface com o usuário pode ser usada para informar ao jogador sobre o que está acontecendo no *game*. A interface com o usuário é uma ferramenta para receber informações do jogador e para apresentar os acontecimentos do mundo virtual para quem está jogando e este deve ser capaz de entender rapidamente o que está acontecendo no momento do jogo.

- **A interface deve oferecer camadas de informações:** a interface deve apresentar uma visão geral do que está acontecendo no *game*, oferecendo uma opção de detalhamento que permita com que o jogador, por sua própria escolha, veja as informações mais detalhadas acerca da situação do mundo do *game*. Exemplos: Qual é o nível de saúde do jogador? A que distância está o próximo nível? Quais são suas reversas de ouro e pedras preciosas? Qual a pontuação? Quanto dinheiro se tem disponível?
- **A interface deve alertar o jogador sobre alterações vitais:** o mundo de um *game* é um sistema dinâmico e, à medida que os jogadores superam obstáculos, eles alteram o mundo e o estado do *game*. Em um *game*, pequenas informações vitais refletem os aspectos mais urgentes das mudanças no mundo do *game*, e eles devem ser relatadas ao jogador de modo dinâmico. Informações importantes que exigem a atenção imediata do jogador devem destacar-se, seja por um recurso auditivo ou visual. Deve assim assegurar-se de que a interface se comunica com os jogadores para ajudá-los em decisões importantes.
- **A interface deve evitar que o jogador cometa erros:** a interface deve verificar todas as informações que recebe (*input*) e rejeitar tudo o que possa causar um erro no jogo. Os jogadores são conhecidos por clicar loucamente e apenas explorar a interface, mesmo quando os componentes da interface não estão sendo usados naquele momento.
- **A interface deve refletir o tom do *game*:** a interface é um componente do *game* que estará na tela na maior parte do tempo em que o *game* é jogado. O mundo do *game* pode mudar drasticamente, mas a interface permanecerá, e deve assim representar o estilo visual e o

tom do jogo, servindo como ferramenta para sustentar o clima dos *games*, das texturas da interface até os efeitos sonoros.

- **A interface não deve dominar a tela do game:** mantenha a interface pequena, mas usável. Mantenha as informações vitais na frente e no centro, e não domine os dados ou a tela com excesso de ornamentos ou elementos gráficos. Mais uma vez, simplifique.
- **Os elementos da interface devem fornecer feedback se forem ativados:** quando jogadores interagirem com a interface, forneça um feedback para que eles saibam que fizeram alguma coisa. Efeitos sonoros em botões ajudam neste tipo de *feedback*. Jogadores executam muitas ações no decorrer de um jogo, sendo assim, assegure-se de que eles saibam que o *game* recebeu uma entrada para que possam seguir em frente para a próxima tarefa.

A interface de um *game* é tão importante quanto ele próprio, pois é a partir deste elemento que o jogador interage e entra no mundo do *game*. Apesar dos tópicos aqui apresentados focarem na produção de interface para *games*, existem que devem ser observados e nunca ignorados: não reinventar a roda, ou seja, os ícones e informações presentes na interface precisam ser óbvios e naturais e sobretudo, manter a simplicidade. Vale lembrar a máxima criada na famosa escola de *design* Bauhaus<sup>46</sup>: menos é mais.

---

<sup>46</sup> Escola de *design*, artes plásticas e arquitetura de vanguarda que funcionou entre 1919 e 1933 na Alemanha. Mais informações em: <http://pt.wikipedia.org/wiki/Bauhaus> (acessado em 16/06/2009).



Figura 22: Interface do jogo *Warcraft* da produtora *Blizzard*



Figura 23: Interface do jogo *Need for Speed Most Wanted* da produtora *Electronic Arts*



### 3.2 - Modelagem 3D: objetos, cenários e mundos

Se é para encontrar um culpado, ele tem nome: John Carmack<sup>47</sup> é o homem que em 1991 fundou id Software (empresa desenvolvedora de *games*) e teve a brilhante ideia de usar o processamento gráfico presente nos computadores da época para criar *games* tridimensionais. Sua primeira criação foi *Wolfenstein 3D*. *Wolf 3D*, como é conhecido, não é o primeiro jogo 3D da história, mas é o *game* que popularizou os jogos de tiro em primeira pessoa e consequentemente fez com que toda a indústria de jogos para computador ficasse de olho no que Carmack e sua equipe estavam produzindo.



Figura 24: Tela de abertura do game *Wolfenstein 3-D* da *id Software*

---

<sup>47</sup> John D. Carmack II nasceu em 20 de agosto de 1970 e é uma das pessoas mais importantes na indústria dos *games*. Programador muito talentoso, Carmack tem feito contribuições significantes no campo de programação de gráficos 3D, é co-fundador da *id Software*, uma das maiores empresas no mercado de desenvolvimento de jogos. Mais informações podem ser encontradas em [http://pt.wikipedia.org/wiki/John\\_Carmack](http://pt.wikipedia.org/wiki/John_Carmack) (acessado em 16/07/2009).

Os *games* de tiro em primeira pessoa (FPS, *first person shooter*, em inglês) não foi criação de Carmack, mas a tecnologia criada na *id Software*, além de popularizar este gênero de jogo, fez com que diversas outras empresas investissem no tema. Além de criar o *Wolf 3D*, a empresa criou também os *games* *Doom*<sup>48</sup> e *Quake*<sup>49</sup>, verdadeiros clássicos da indústria e figuraram entre os mais jogados de todos os tempos.



Figura 25: Tela de abertura do *game* *Doom*, da *id Software*

Para fazer com que suas criações funcionassem, a *id* criou um motor de jogo (*engine*), software que simplifica a produção e o desenvolvimento de *games*, e licenciou seu programa para a *Valve Corporation*, que com o criou o *Half-*

---

<sup>48</sup> *Doom* é um jogo de computador lançado em 1993 pela *id Software*, e um dos títulos seminais do gênero tiro em primeira pessoa. Combinando gráficos 3D com violência gráfica, ele tornou-se tanto controverso quanto imensamente popular, com um lançamento em versão shareware que estima-se ter sido jogada por 15 milhões de pessoas. Mais detalhes em <http://pt.wikipedia.org/wiki/Doom> (acessado em 16/07/2009).

<sup>49</sup> Foi lançado em 1996 e chegou a vender 4 milhões de cópias.

*Life*, e também para a *Electronic Arts*, que criou, juntamente com Steven Spielberg, o game *Medal of Honor*. Do *Half-Life* original nasceu o *Counter-Strike*, game quase que onipresente em *lan houses* tupiniquins.

Da década de 1990 para cá, muita coisa mudou. A capacidade de processamento gráfico dos computadores aumentou muito, diversos consoles surgiram e o 3D dominou então a indústria dos jogos digitais. Em dezembro de 1994 a *Sony* lançou o *Playstation* e o 3D saiu do computador e passou a dominar também a sala de estar em função de processador gráfico presente no equipamento sendo capaz de processar 300.000 polígonos por segundo. Algo muito avançado para a época.

O que antes só era possível ser jogado em um computador de ponta, passou a ser jogado em uma televisão. Junto com o alto poder de processamento dos *PlayStation* surgiram jogos como: *Resident Evil*, *Tomb Raider*, *Gran Turismo*, *Final Fantasy* e muitos outros. Depois da revolução causada pelo *PlayStation*, outros consoles com características semelhantes invadiram o mercado: *X-Box*, *PlayStation 2*, *Nintendo Game Cube*, *X-Box 360*, *PlayStation 3*, *Nintendo Wii* entre outros.

Apesar do *games* em 2D terem presença dominante nas décadas de 70, 80 e 90, o 3D, nos dias de hoje, domina o cenário de grande parte da produção de jogos, sejam eles comerciais ou independentes. Mas no mundo dos *games*, onde está presente o 3D, a tridimensionalidade pode ser encontrada em objetos, personagens e mapas.

A modelagem tridimensional para *games* se difere da modelagem para animação em cinema e diversos pontos, sendo o principal deles o número reduzidos de polígonos. Segundo Silva (2005), polígonos são as linhas e vértices que formam um objeto, personagem ou mapa. Quanto maior o

que formam um objeto, personagem ou mapa. Quanto maior o número de polígonos, melhor a qualidade gráfica e maiores os requisitos de *hardware* para que o jogador possa jogar o *game*.

Para Bairon e Petry (2000), a modelagem 3D, seja ela para construção de mapas, objetos ou personagens, permite a representação de pensamentos formais que a linguagem escrita torna proibitiva ou quase impossível. Esta representação só é possível, em uma produção de *games*, com *softwares* específicos para esta finalidade.

Dentre os *softwares* comerciais para modelagem 3D, o *Cinema 4D*, *Autodesk 3ds Max*<sup>50</sup> e o *Autodesk Maya* merecem destaque em função da sua larga utilização entre os desenvolvedores. O site da *Autodesk*, destaca as funcionalidades do *3ds Max*, como vemos a seguir:

Essa solução de modelagem, animação e renderização 3D repleta de recursos é usada para a produção dos jogos mais vendidos e de premiados filmes e conteúdo de vídeo. Trata-se da ferramenta recomendada para a rápida geração de personagens realísticos, efeitos de computação gráfica perfeitos, jogos de cair o queixo ou filmes e conteúdo de televisão de qualidade superior. O aprimorado conjunto de ferramentas permite que você crie seu ambiente 3D como quiser, além de possibilitar o gerenciamento de cenas complexas e tirar proveito da melhor interoperabilidade de *softwares* e do maior suporte para integração em canais.

Apesar do discurso mercantilista do texto acima, é possível entender que é uma ferramenta que atende as necessidades de diversos públicos. Tamanho desempenho acompanha um alto valor de licenciamento, o que torna praticamente inviável para quem está começando, como é o caso do produtor

---

<sup>50</sup> Mais detalhes sobre o *3ds Max* pode ser encontrado em <http://www.autodesk.com.br/> (acessado em 16/07/2009).

independente que como o fanzineiro<sup>51</sup> do passado, precisa de maneiras alternativas e mais econômicas para viabilizar suas criações ainda em começo de carreira.

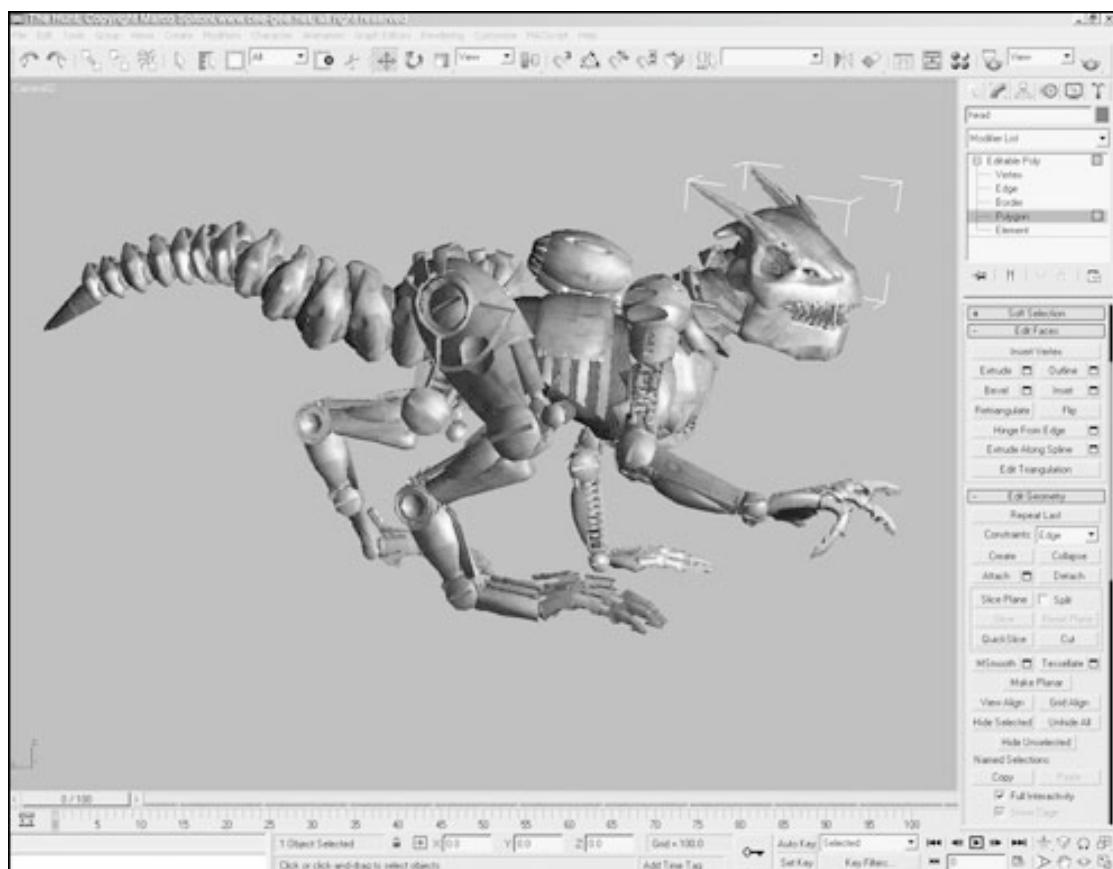


Figura 26: Interface do Autodesk 3ds Max

Além do Autodesk 3ds Max, um outro produto de alto desempenho e muito popular é o Autodesk Maya. Segundo Marcelo e Pescuite (2009), esse é um dos mais populares e poderosos pacotes de modelagem, animação e rendering<sup>52</sup> 3D que existem no mercado. Ganhou diversos prêmios e é utilizado por vários estúdios de Hollywood e muitas produtoras de *games*. É um paco-

<sup>51</sup> Fanzineiro é a pessoa responsável pela edição de um fanzine, revista independente editada por fã de determinado assunto. Mais em <http://pt.wikipedia.org/wiki/Fanzine> (acessado em 16/07/2009).

<sup>52</sup> Rendering é o processo de gerar uma imagem à partir de um modelo 3D por meio de softwares específicos para tal finalidade.

te caro e rico em diversos recursos, com uma grande quantidade de *plugins*<sup>53</sup> e ferramentas de apoio muito grande. Roda em ambiente *Windows* e *Macintosh*, mas é uma opção fora da realidade para muitas empresas e, sobretudo, para o produtor independente. Contudo, no site da Autodesk há uma versão para *download* gratuito, mas com redução de recursos.

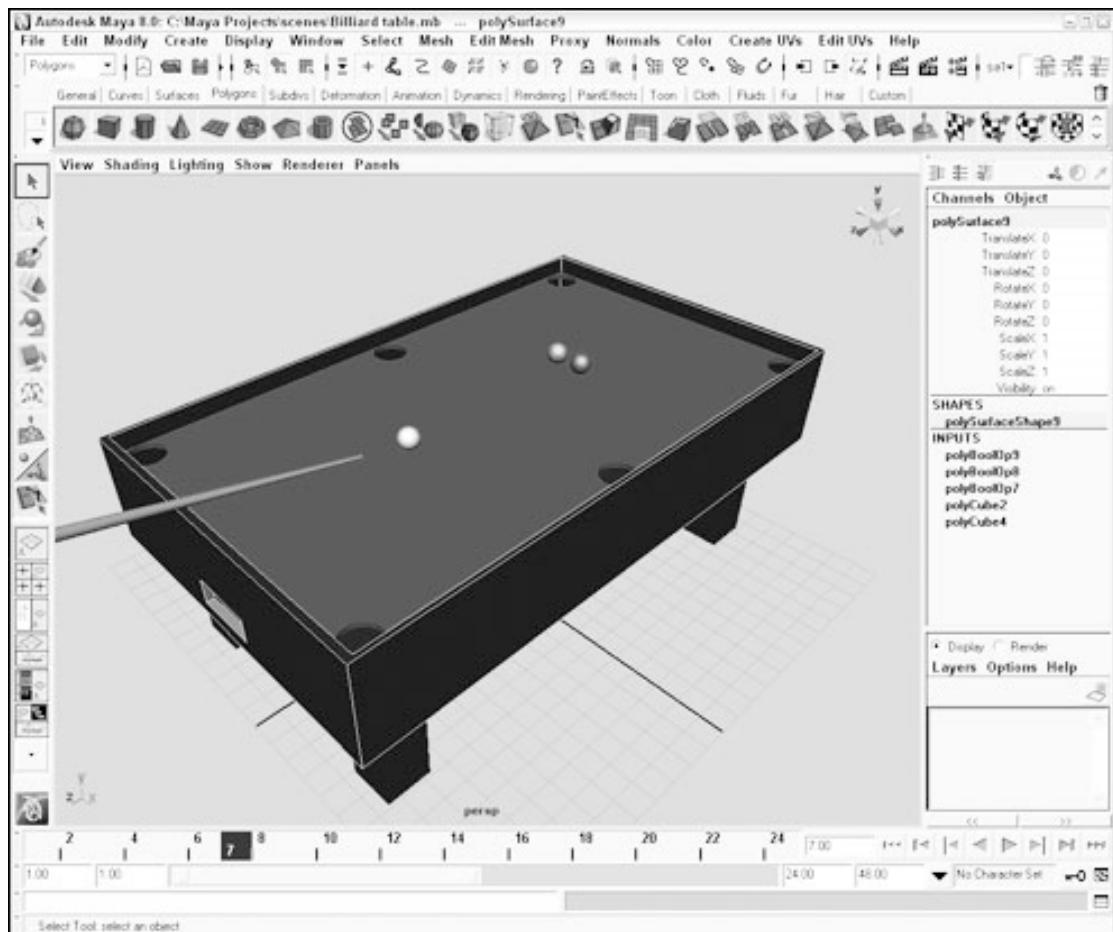


Figura 27: Interface do Autodesk Maya

<sup>53</sup> *Plug-in* é um *software* que serve normalmente para adicionar funções a outros programas maiores, provendo alguma funcionalidade especial ou muito específica. Mais detalhes em <http://pt.wikipedia.org/wiki/Plugin> (acessado em 13/08/2009).

Um *software* relativamente novo dentro universo das ferramentas de modelagem 3D é o *Rhinoceros 3D*<sup>54</sup>, ou simplesmente *Rhino 3D*, que conta com um grande conjunto de ferramentas de apoio. Apesar de ser um *software* comercial, professores e estudantes contam com descontos significativos.

Tendo como foco o produtor independente de *games*, a ferramenta ideal é o *Blender*. Este é um *software* de criação 3D multiplataforma que roda em *Windows*, *Linux*, *Mac OS X* e possui recursos de modelagem tridimensional, animação, renderização e pós-produção. O programa pode ser utilizado em diversas áreas que utilizem modelagem 3D, como arquitetura, publicidade, desenho industrial, engenharia, vídeo, animação, e sobretudo, desenvolvimento de *games*.

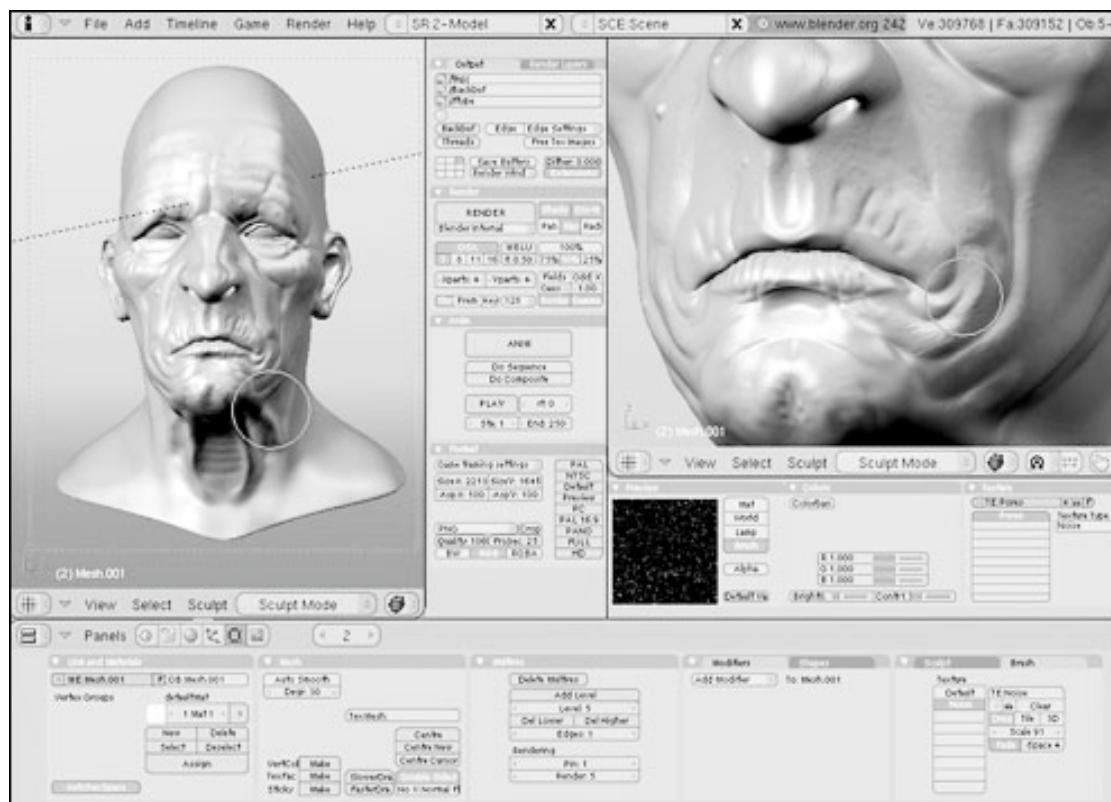


Figura 28: Interface do *Blender*

<sup>54</sup> Mais informações podem ser obtidas em <http://www.rhino3d.com>

A versão 2.49 conta também com um mecanismo de jogo (*game engine*) baseado na linguagem de programação *Python*<sup>55</sup>, que permite criar cenas interativas, objetos manipuláveis que podem responder às ações do mouse, comandos de teclado e interagir com outros objetos e eventos da mesma cena.

Sendo o *Blender* um *software* de código aberto, sua evolução é constante. Apesar de o programa já possuir um longa-metragem intitulado *Elephants Dream*<sup>56</sup>, de animação, feito com todos os seus recursos, o *Blender* ainda não pode ser comparado com softwares comerciais, como o *Maya* ou o *3ds Max*, apesar de possuir recursos semelhantes e muito bem definidos. A grande vantagem do produto é que ele está disponível gratuitamente para *download*<sup>57</sup> e, além de contar com recursos de modelagem tridimensional, conta também com um motor de jogo nativo no produto, o que aumenta ainda mais sua atratividade para o produtor independente de *games*.

---

<sup>55</sup> Python é uma linguagem de programação de alto nível, interpretada, imperativa, orientada a objetos, de tipagem dinâmica e forte. Mais detalhes em <http://pt.wikipedia.org/wiki/Python> (acessado em 14/08/2009)

<sup>56</sup> *Elephants Dream* está disponível para *download* em <http://www.elephantsdream.org>

<sup>57</sup> O *Blender* pode ser baixado em <http://www.blender.org/>. Uma boa compilação de dicas do *Blender* pode ser encontradas aqui <http://blendertotal.wordpress.com/10-dicas-para-aprender-blender-3d>.

### 3.3 - Produção de imagens 2D: texturas e similares

Ao olhar para qualquer cenário de um jogo 3D, o que vamos ver, além das formas devidamente modeladas, são texturas bem desenhadas. Mas o que é uma textura? Como ela é construída? Qual a melhor forma de aplicação?

Para Dondis (1991) textura é o elemento visual que com frequência serve de substituto para as qualidades de outro sentido, o tato. Na verdade, porém, podemos apreciar e reconhecer a textura tanto através do tato quanto da visão, ou ainda mediante uma combinação de ambos. Como estamos aqui trabalhando com um universo de conceitos ligados ao mundo digital, nos ateremos, quando falarmos de sobre reconhecimento de textura, a algo ligado à visão. Ao navegar em um ambiente tridimensional, no caso aqui estudado, um *game*, e se deparar com uma pedra, o que se vê é a simulação de uma pedra em um ambiente virtual. Ao pegar a pedra, você sabe que é uma pedra, porém, não pode sentí-la.

Em um ambiente 3D a textura, combinada com os polígonos de um objeto modelado em um *software* de composição tridimensional, como o por exemplo, o *Blender*, é o elemento responsável por dar vida e beleza visual à construções poligonais aproximando visualmente, quando necessário, este objeto modelado de uma imagem real.

Em qualquer ambiente digital, uma textura é, antes de tudo, uma imagem. E esta imagem, no ambiente computacional, é formada por um mapa de *bits*, ou seja, imagens formadas por *pixels*<sup>58</sup>, onde cada um destes contém

---

<sup>58</sup> *Pixel* é o menor elemento num dispositivo de exibição (como por exemplo um monitor), ao qual é possível atribuir-se uma cor

uma descrição específica de cor. Este mapa de *bits*, por sua vez, é composto por formas, cores e tamanhos diferentes.

As texturas podem ajudar a representar todo e qualquer elemento do mundo real, como por exemplo, um céu repleto de nuvens, uma montanha rochosa, folhas de uma árvore e até mesmo um copo de água.



**Figura 29:** céu, montanha e água. As texturas constroem a beleza do ambiente 3D

Como a textura é também um arquivo digital e uma imagem, os formatos mais comuns atualmente presentes neste arquivos são .JPG, .TIF e .BMP. É importante observar que, uma textura pode ser produzida digitalmente, em um software de edição de imagens utilizando-se as ferramentas específicas para tal finalidade. Outro recurso extremamente útil e funcional são as imagens fotográficas. Fotografar elementos e transformar estas imagens em texturas que serão usados em uma produção tridimensional pode dar um to-

que especial de realismo ao trabalho, desde que a fotografia tenha uma qualidade minimamente aceitável, como foco, cor e nitidez.

*Softwares* de edição de imagem bidimensionais estão presentes no mercado de editoração eletrônica e manipulação de imagens desde a década de 1980, com o lançamento do *Adobe Photoshop* em 1988. Sinônimo de edição de imagens, o *Photoshop*, como é popularmente conhecimento, atualmente está em sua versão 11.0, denominada CS4. Do ponto de vista do produtor independente de *games*, o *Photoshop* seria a melhor ferramenta a ser usada em função do grande número de recursos e ferramentas, mas o fator custo pode dificultar a aquisição de um produto desta natureza pelo desenvolvedor em início de carreira.

Contudo, o *Adobe Photoshop* não é a única opção quando se trata em manipulação de imagens bidimensionais e produção de texturas. Uma alternativa é o *GIMP*, programa de código aberto voltado para a criação e edição de imagens.

*GIMP (GNU Image Manipulation Program)*<sup>59</sup> foi criado em 1995 por dois estudantes tendo com premissa básica ser uma alternativa ao *Adobe Photoshop*. Apesar de algumas limitações com relação ao seu concorrente comercial, o *software* apresenta um conjunto de ferramentas que tornam o produto atraente e com recursos suficientes para o produtor independente.

Como é um *software* livre, conta com uma grande quantidade de *plugins* disponíveis desenvolvidos pela comunidade criada em torno do programa, e conta também com suporte a diversos formatos de arquivos de imagem como .JPG, .GIF, .BMP, .TIF, .PIX, .WMF e .XWD.

---

<sup>59</sup> Mais informações em <http://www.gimp.org/>

Com tamanha versatilidade o *GIMP* faz com que seja a ferramenta ideal para o produtor de *games* independentes. Além da criação de texturas para modelagens 3D, é possível usar o *software* para:

- Criação de logotipo do *game*;
- Criação e manipulação de interface gráfica de usuário;
- Criação de menus e botões;
- Manipulação de imagens fotográficas;
- Criação e edição de texturas.

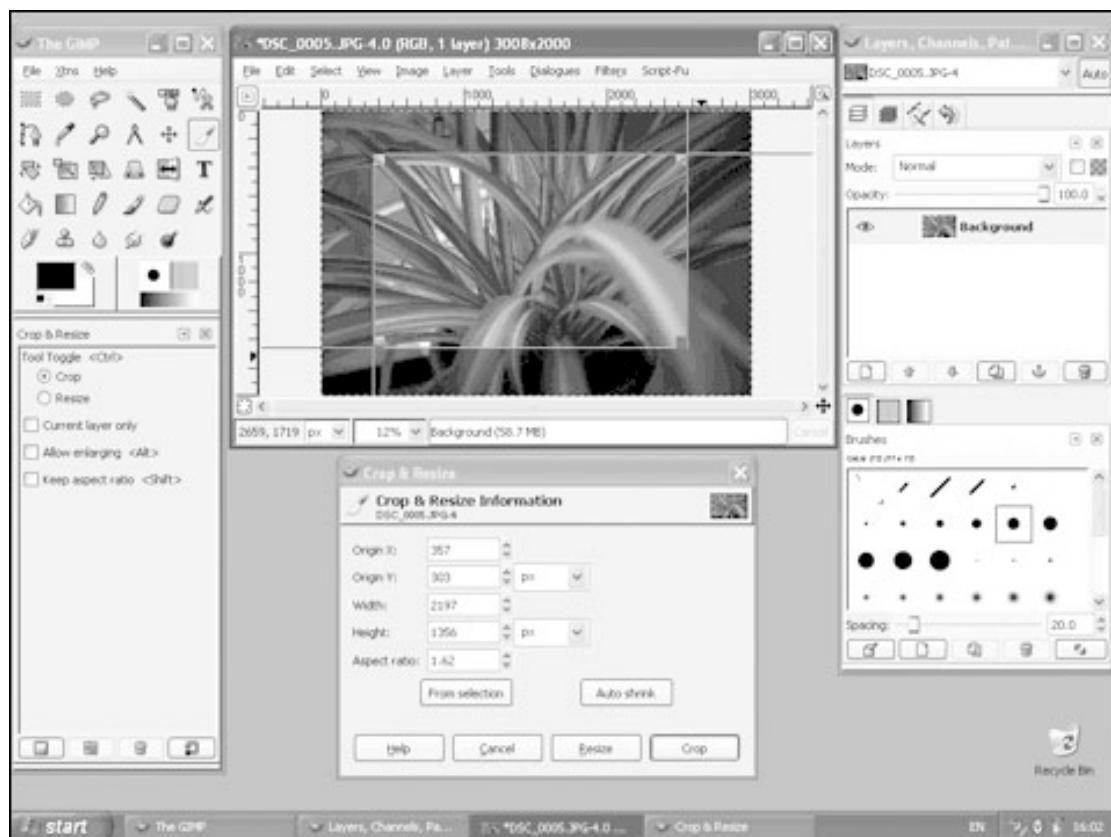


Figura 30: Interface de trabalho do *GIMP*

Tal versatilidade em suas funções faz do *GIMP* a ferramenta ideal para o produtor independente de *games*.

### 3.4 - Ambientação e produção sonora

Você já imaginou jogar um *game* sem som? Como seria jogar qualquer modalidade do *game* *WiiSports*<sup>60</sup> sem os sons emitidos pelo *Wii Remote*<sup>61</sup> que ajudam a construir no processo de sentir o jogo? No *WiiSports*, assim como em muitos jogos para a plataforma *Wii*<sup>62</sup>, da Nintendo, usam o som como um elemento sensorial que ajuda nas ligações do conjunto jogador -> controle (*joystick*) -> console -> jogo. O som é um elemento essencial na aproximação do usuário ao *game* e contribui com o processo de imersão.



Figura 31: Gamers jogando *WiiSports*. O controle do *Wii* também emite sons

<sup>60</sup> Mais detalhes em <http://us.wii.com/wiisports/>

<sup>61</sup> *Wii Remote*, também conhecido com *Wiimote*, é o principal controle do *Wii*, console da *Nintendo*.

Mais detalhes em [http://pt.wikipedia.org/wiki/Wii\\_Remote](http://pt.wikipedia.org/wiki/Wii_Remote)

<sup>62</sup> <http://www.nintendo.com/wii>

O conjunto de sons presentes em um *game* desempenham papel fundamental no processo de imersão no jogo; é também chamado de paisagem sonora, que segundo Schafer (1997), é qualquer porção do ambiente sonoro visto como um campo de estudos. O termo pode referir-se a ambientes reais ou construções abstratas, como musicais e montagem de fitas, em particular consideradas como um ambiente.

Sendo o *game* um produto digital, concebido, montado e gerado em um computador, como podemos entender e definir os elementos sonoros presentes em um jogo? Como é a paisagem sonora de um *game*?

De acordo com Shum (2009), a paisagem sonora de um filme ou *game* pode variar entre cenas e/ou fases diferentes, e é composta pelos sons *on-screen*, cujas fontes sonoras podem ser vistas na tela, e *off-screen*, aqueles cujas fontes não são mostradas na tela. Os sons *on-screen* normalmente são provenientes de objetos, pessoas, seres ou máquinas e costumam estar sincronizados com suas fontes causadoras. Já os sons *off-screen* são responsáveis pela extensão sonora do ambiente.

Como definir os elementos de uma paisagem sonora de um *game*? É fundamental, antes de se pensar nos sons presentes em um jogo digital, realizar um inventário, ou seja, um levantamento de todos os itens, passagens, interações e situação de jogo onde o som pode estar presente. É justamente o conjunto destes sons, presentes neste inventário, que ajudam a compor a paisagem sonora de um *game*.

Este grande conjunto de sons, combinados de diversas maneiras durante o ato de jogar criam as mais diversas percepções e interpretações que o jogador tem ao interagir com um *game*. Shum (2009) ajuda a entender esta

questão afirmando que a criação de diferentes paisagens sonoras para uma mesma tela, cena ou fase de um *game* estimula a irrupção de novas leituras e interpretações dos elementos visuais e da própria experiência de jogar.

Mas e o produtor independente? Quais elementos sonoros o autor de um *game* independente precisa produzir, ou atentar para a produção? Higgin (2005) identifica sete elementos fundamentais para a produção de áudio para *games*, são eles:

- **Silêncio:** uma das ferramentas mais efetivas para criação de uma ambientação sonora efetiva é o silêncio. Em filmes de terror o silêncio é um elemento chave da narrativa. Em um *game*, o silêncio é usado como um elemento adicional e não como falta de elementos.
- **Atmosfera ou ambiente:** produtores de áudio para *games* geralmente iniciam a produção dos sons a partir de um tom que reflete a atmosfera de um quarto comum. Esses sons ficam como um *background* contínuo de cada ambiente e são geralmente simulações do som que ouvimos quando estamos em silêncio absoluto.
- **Sons externos:** os sons externos são todos aqueles que ocorrem aleatoriamente no ambiente em que estamos imersos. Como por exemplo: buzinas, sirenes, pássaros, vento, assobios, cornetas... Estes sons são considerados narrativas adicionais.
- **Sons elementares:** são elementos realistas associados às ações do personagem principal do *game* ou presente em cenários, como por exemplo: no andar de um personagem, o som de seus passos devem estar de acordo com a velocidade de seus passos e de acordo com o piso onde este personagem está andando.
- **Sons associados:** todos aqueles gerados a partir de movimentos, como por exemplo, o som de uma armadura presente em um personagem que não existe na vida real.

- **Sons de atrito:** colisões, choques, empurrações, carga de arma, galho quebrando ou portal batendo...
- **Efeitos sonoros:** qualquer coisa que possa chamar a atenção, gratificar ou simplesmente iludir deve ter um efeito sonoro associado à ação.

De posse de todos estes conceitos elementares presentes na produção de áudio para *games*, é fundamental escolher um formato de áudio. Este formato, que geralmente não é único, pode variar de acordo com a situação presente na produção. Contudo, Higgin (2005) aponta dois fatores fundamentais na escolha do formato a ser utilizado: qualidade e tamanho.

Mas quais são os formatos de áudio digital mais comuns? Velho e Nigro (2002) fazem uma classificação dos formatos de áudio digital e apresentam uma tabela muito útil para quem trabalha com áudio, sobretudo com *games*, e podem ser vistos na tabela a seguir:

Formatos de áudio digital mais comuns		
TIPO	CODEC	EXTENSÃO DO ARQUIVO
AIFF	PCM (aceita outros codecs)	.aif, .aiiff
AU	u-Law	.au
CD Audio (CDDA)	PCM	(não se aplica)
MP3	MPEG Audio Layer III	.mp3
WMA	Proprietário (Microsoft)	.wma
QuickTime	Proprietário (Apple)	.qt
RealAudio	Proprietário (Real Networks)	.ra, .ram
Sound Designer II	PCM	.sd2
WAV	PCM (aceita outros codecs)	.wav

Os formatos presentes na tabela acima são criados em *softwares* específicos para produção sonora digital. Dentre os *softwares* comerciais comumente usados por produtoras profissionais, podemos citar: *Protocols*, *Sound Forge*, *Cakewalk*, *Vegas*, *Aware Studio*, *Cubase VST* entre outros.

O produtor independente de *games*, não dispõe de recursos financeiros para comprar *softwares* proprietários de alto valor, tem que recorrer a *softwares* gratuitos e que sejam, no mínimo, de qualidade. Para produção e edição de sons, um *software* gratuito, fácil de usar e altamente recomendado é o *Audacity*<sup>63</sup>, programa livre e gratuito, de código aberto e disponível para os mais diversos sistemas operacionais, como *Windows*, *Linux* e *Mac OS*. Atualmente na versão 1.3, conta com recursos profissionais, podendo ser usado na produção de todo e qualquer *game* independente.

Com o *Audacity* é possível editar sons dos mais diversos, fazer cortes, combinar elementos sonoros, mixar faixas de áudio e muito mais. O *software* é um poderoso aliado das bibliotecas sonoras disponíveis na *web*. Estas bibliotecas possuem uma grande gama de sons dos mais variados tipos e são extremamente úteis na produção de um jogo digital independente. Existem diversos bibliotecas pagas, contudo, existe um sem número de opções gratuitas que podem complementar todos os quesitos sonoros existentes na produção de um jogo digital independente.

---

<sup>63</sup> O *Audacity* está disponível para download em <http://audacity.sourceforge.net/?lang=pt>

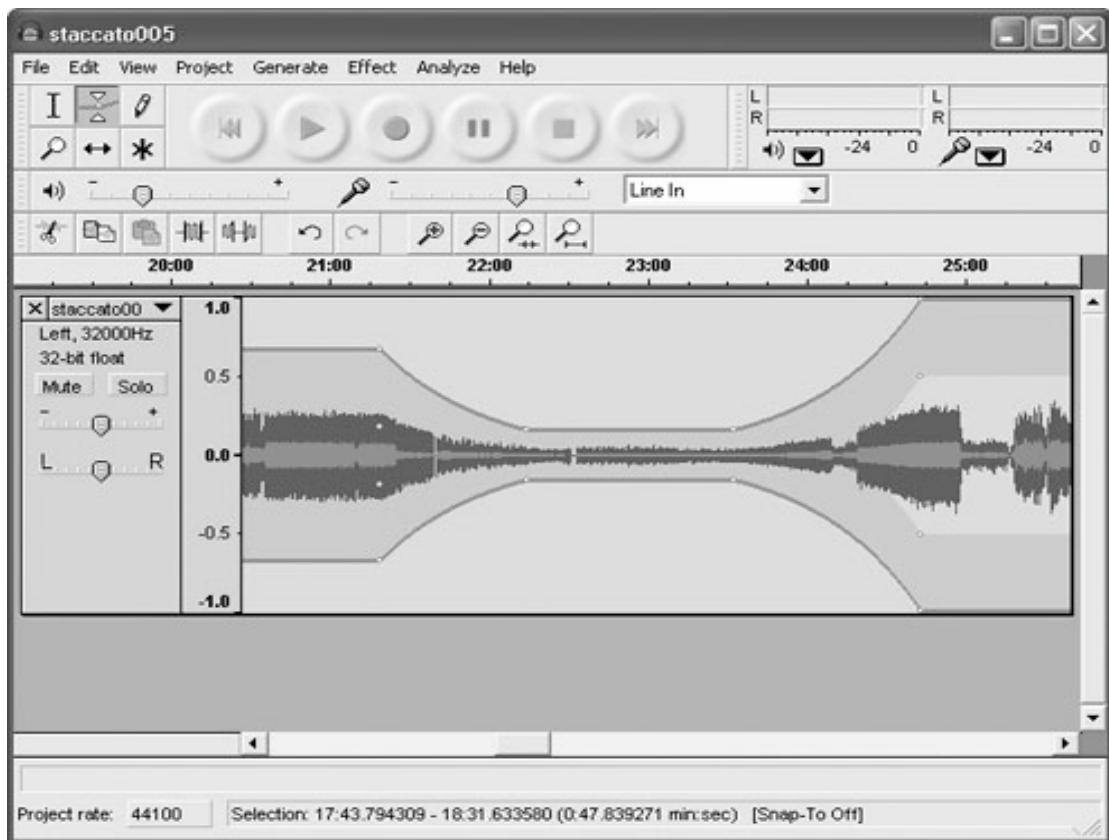


Figura 32: Interface do Audacity

### 3.5 - Motor de jogo: *game engine* e suas funcionalidades

Depois de entender os conceitos fundamentais para a produção de *games* apresentados até este momento da dissertação, chega o momento de analisar e entender questões relacionadas ao motor de jogo (*engine* ou *game engine*) de um *game* e suas respectivas funcionalidades.

O presente capítulo da dissertação tem como objetivo principal, comparar, analisar e discutir tipos de *engines* de *games*, classificados dentro das usuais categorias livres que são utilizadas na comunidade técnica e usuários no ciberespaço, a saber, os tipos *Indie*, *Open Source* e *Comerciais*.

De acordo com Clua (2008) o motor de jogo é um ambiente que permite realizar a integração de toda a diversidade de componentes de um *game*. Existem centenas de motores de *games*, alguns de código aberto, outros que custam dezenas de dólares e ainda outros na custam centenas de milhares de dólares. Tal diversidade no mundo dos *engines* faz com que o produtor independente realize uma escolha criteriosa para seu projeto de jogo digital e justamente por isso que o pleno entendimento das funcionalidades de um motor de *engine* é extremamente importante.

Gomes e Pamplona (2005) complementam as ideias de Clua (2008) e afirmam que os motores são bibliotecas de desenvolvimento responsáveis pelo gerenciamento do jogo, das imagens, do processamento de entrada de dados e outras funções. A ideia é que os motores implementem funcionalidades e recursos comuns a maioria dos jogos, permitindo que esses recursos sejam reutilizados a cada novo *game* criado. Os motores são tão importantes que estão em praticamente todos os jogos para micro-computadores, controlando a estrutura do jogo e seu ciclo de vida.

Podemos afirmar então que um motor de jogo é o coração e o cérebro de um *game*, ou seja, controla tudo o que acontece em um jogos e é responsável também por sua vida. A figura abaixo mostra um esquema de funcionamento de um moto de jogo.

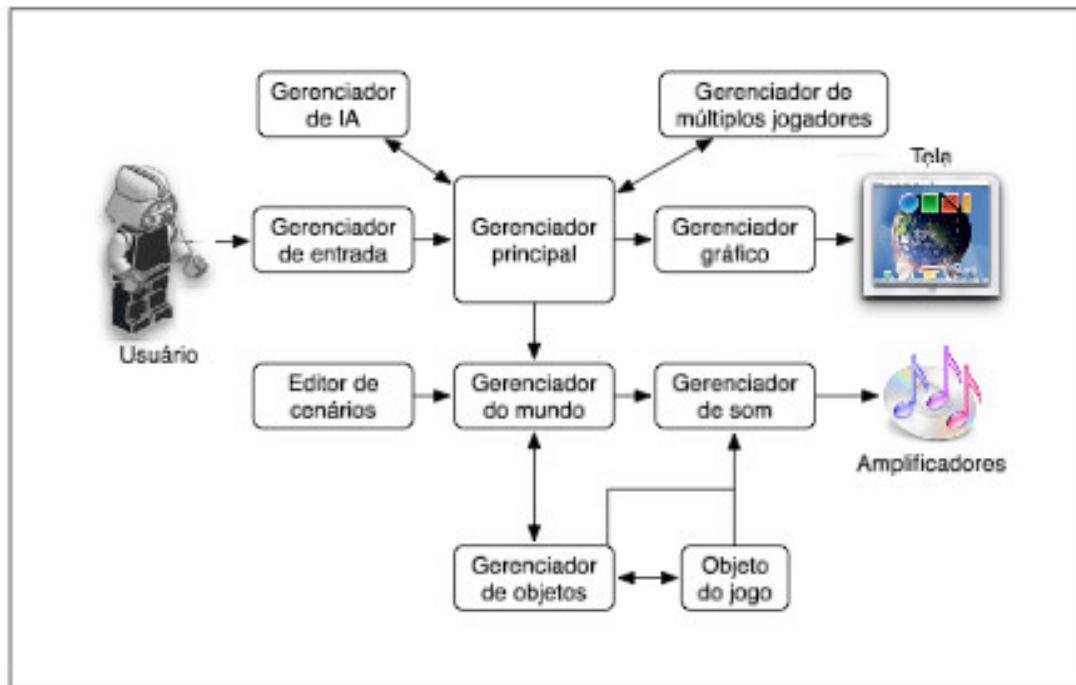


Figura 33: Esquema de funcionamento de motor de jogo

O motor de jogo é uma ferramenta de autoria que permite que o criador de jogos junte todos os componentes presentes em um projeto de jogo, empacote estes elementos e transforme tudo isso num *game* jogável. Mas como funciona um motor de jogo? Quais as funcionalidades de cada um dos elementos presentes em um *game engine*?

O gerenciador de entrada recebe e identifica os eventos de entrada e os envia para o gerenciador principal. O gerenciador gráfico transforma o modelo que define o estado atual do jogo em uma visualização para o usuário. O gerenciador de inteligência artificial gerencia o comportamento dos objetos desenvolvi-

dos pelo *designer* do jogo. O gerenciador de múltiplos jogadores trata da comunicação dos jogadores, independentemente do meio físico em que se encontram. O gerenciador de objetos carrega, controla o ciclo de vida, salva e destrói um grupo de objetos do jogo. Em geral, um jogo possui vários gerenciadores de objetos que, além de suas funções normais, ainda precisam se comunicar. O objeto do jogo possui dados relevantes para uma entidade que faça parte do jogo (como “avião”, “monstro”, etc). Esta parte do motor controla a posição, velocidade, dimensão, detecção de colisão, entre outros. O gerenciador do mundo armazena o estado atual do jogo e para isso utiliza os gerenciadores de objetos. Em geral, uma ferramenta externa, o editor de cenários, descreve um estado inicial do jogo para cada um de seus níveis. O gerenciador principal é responsável pela coordenação entre os demais componentes (GOMES E PAMPLONA, 2005, pg. 56).

A lista de motores de jogos existentes nos dias de hoje é extensa e vale ressaltar que o produtor independente deve analisar todas as suas funcionalidades antes de escolher o motor mais adequado para o seu projeto. Estas funcionalidades devem estar de acordo com o projeto de jogo a ser desenvolvido. Um produto popular, fácil de usar e que não conta com um custo tão alto é o *3D Gamestudio*<sup>64</sup>. Com diversos títulos já publicados desenvolvidos com a ferramenta, este é um aplicativo para autoração de projetos multimídia 2D e 3D focado em jogos para computador. Mas o programa não é apenas um *game engine*, e sim um sistema completo de desenvolvimento que combina a linguagem de programação *C-Script* com recursos avançados de modelagem 3D.

O *software* conta com *engine* de física (gravidade, fenômenos da natureza, entre outros), editor de níveis, modelador de terrenos e uma grande biblioteca de objetos 3D. Segundo os desenvolvedores do programa, nunca foi tão fácil criar *games* de tiro em primeira pessoa, personagens 3D, *RPGs*, simula-

---

<sup>64</sup> Mais informações em <http://www.3dgamestudio.com/>

dores de voo, jogos de tabuleiro, *games* de esportes e outros gêneros. No site, além conferir vários demos de jogos feitos com o *3D GameStudio*, pode-se baixar uma versão de teste do produto, que pesa só 35 MB. Ele é um *engine* comercial de baixo custo e indicado para projetos de *games* independentes.

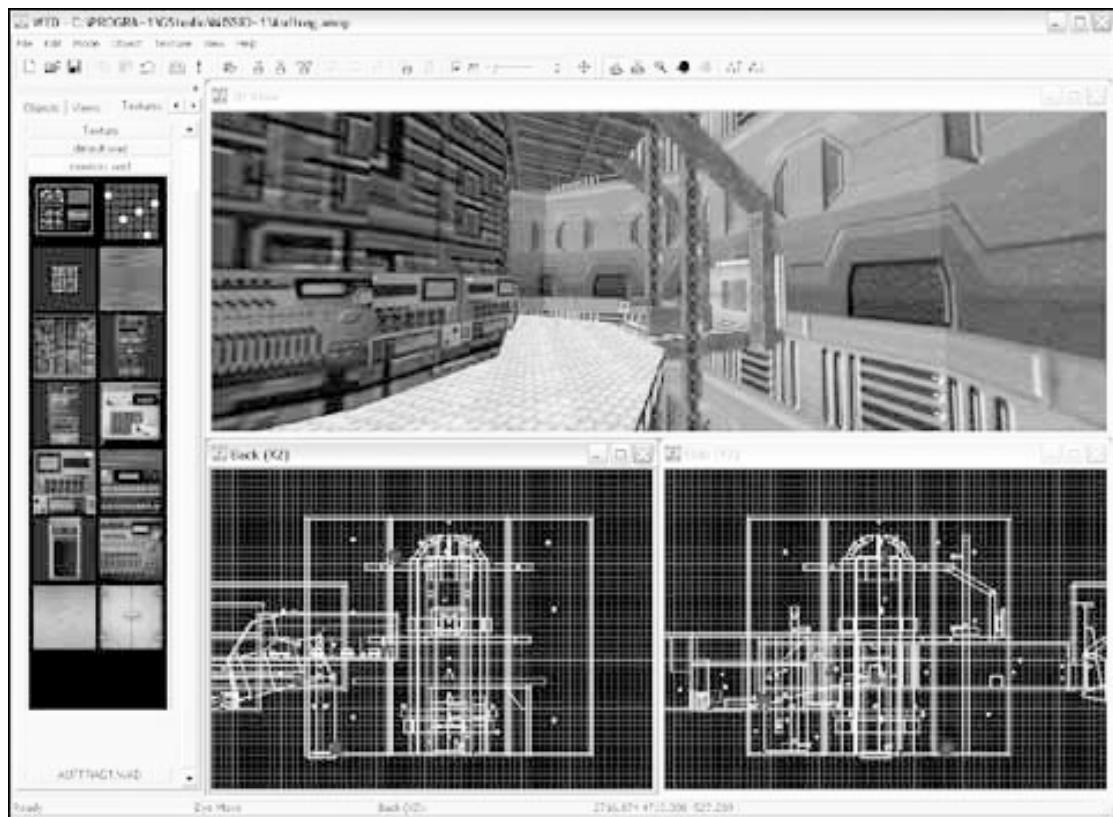


Figura 34: tela de trabalho do *3D Gamestudio*

Assim como o *3D Gamestudio*, um motor de jogo que indicado para produtores independentes é o *NeoAxis Engine*<sup>65</sup>. Este é um ambiente completo e integrado de desenvolvimento de aplicações 3D interativas, incluindo aqui mundos virtuais, *games* e simulações realistas.

---

<sup>65</sup> Mais detalhes em <http://www.neoaxisgroup.com/>

O sistema do *Neoaxis Engine* é composto por um motor de jogo 3D em tempo real, ou seja, você monta uma cena de *game* e já consegue ver, testar e jogar; e também um conjunto de ferramentas de programação flexível que pode ser adaptado a diferentes projetos. O produto conta também com um editor de mapas, editor de materiais, editor de interface gráfica de usuário, editor de física (para colisões, gravidade e afins), sistema editor de partículas e um editor de terrenos, todos plenamente integrados e com suporte de programação .NET Framework 2.0<sup>66</sup>, da Microsoft, aceitando ainda pacotes de objetos 3D modelados na principais ferramentas 3D do mercado, como 3D Studio Max, Maya e Blender.

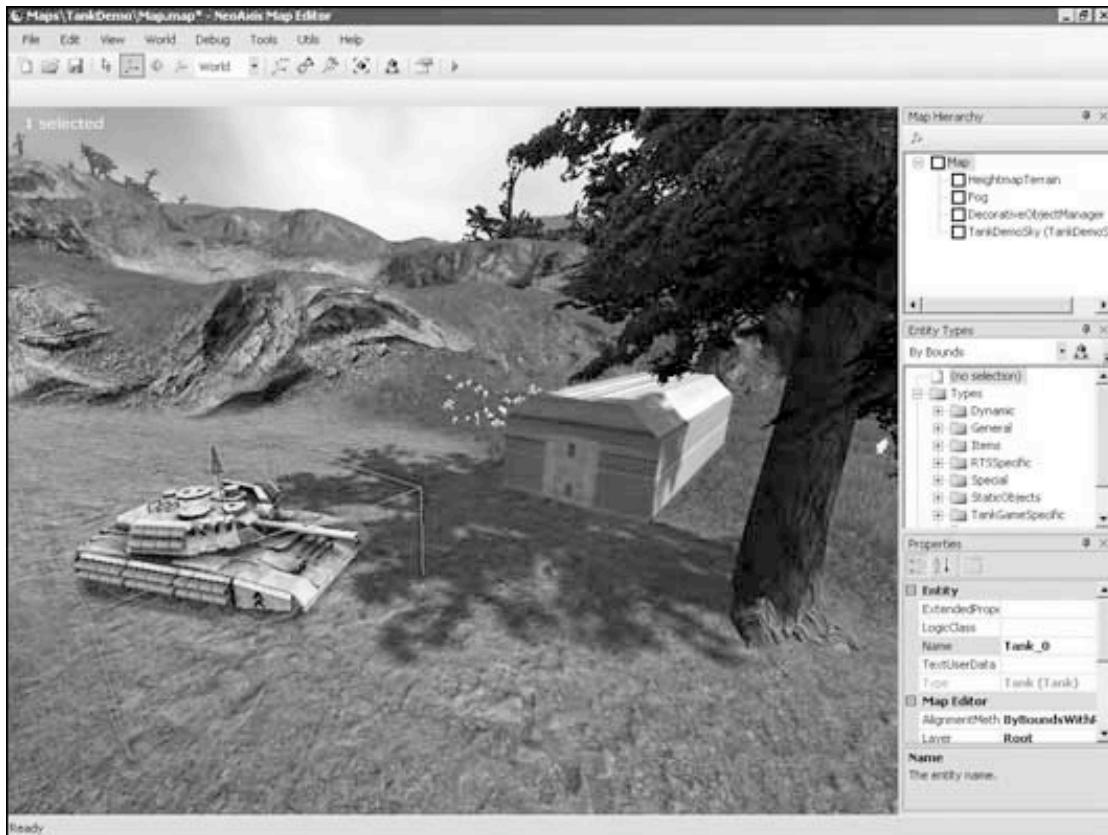


Figura 35: tela de trabalho do *Neoaxis*

---

<sup>66</sup> Mais informações em [http://pt.wikipedia.org/wiki/.NET\\_Framework](http://pt.wikipedia.org/wiki/.NET_Framework)

Dentro do universo de *engines* livre, gratuitos e de código aberto, o *Ogre 3D*<sup>67</sup> é uma ferramenta popular e usada por diversos desenvolvedores, seja por ideologia, ou seja, a opção preferencial por produtos livre e com código aberto ou seja simplesmente por sua funcionalidades.

No campo do código aberto, o *Ogre 3D* (sigla para *Object-oriented Graphics Rendering Engine*) é o motor predominante. Desenvolvido por um pequeno time colaborativo, atualmente está na versão 1.4.3. A principal linguagem de desenvolvimento do *Ogre 3D* é o C++, contudo, por ser um projeto de código aberto e colaborativo, já existem versões em testes para rodar nas linguagens *Python*, *Java* e *.NET*.

Também no campo do código aberto, está o *Blender*<sup>68</sup> que, além de ser um *software* de modelagem tridimensional, conta com um motor de jogo nativo chamado *Blender Game Engine*, que usa a linguagem *Python* para *scripts* de funcionamento de jogo. *Scripts* automatizam e ampliam as ferramentas do Blender quando o assunto é motor de jogo. A grande versatilidade do motor de jogo do *Blender* faz com que seja possível realizar, além de um projeto de um jogo digital, projetos de realidade virtual, planejamento arquitetônico, auxílio em processo de animação e apresentações.

Estudar e aprender toda a sintaxe de funcionamento do *Python* é fundamental para trabalhar com o *Blender Game Engine*. Uma das grandes vantagens deste moto de jogo, além da sua gratuidade, é o fato de sua fácil integração com o sistema de motor gráfico do *Ogre 3D*, o que faz a ferramenta ideal para os partidários do *software* livre.

---

<sup>67</sup> Mais informações em <http://www.ogre3d.org>

<sup>68</sup> Mais informações em <http://pt.wikipedia.org/wiki/Blender>

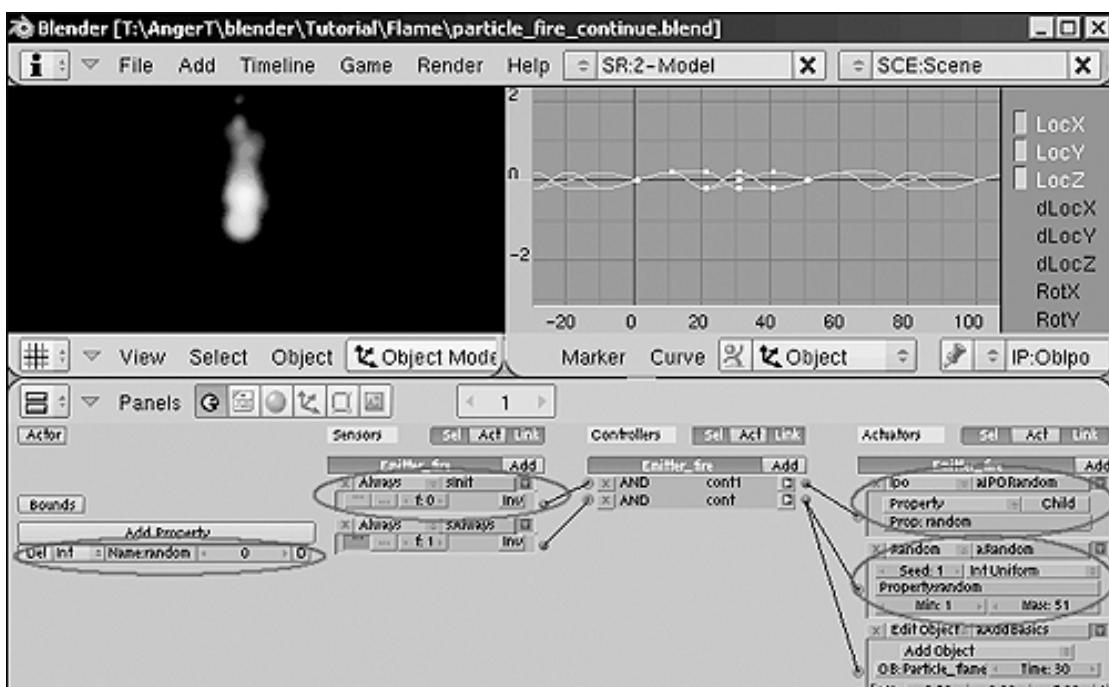


Figura 36: tela de trabalho do *Blender Game Engine*

Uma funcionalidade do *Blender Game Engine* que torna o *software* uma boa opção para desenvolvedores iniciantes é seu sistema de configuração de ações de programação de forma visual, ligando comando de programação e funções em *Python* visualmente.

A Microsoft, gigante do *software*, não ficou de fora na briga por criar ambientes de desenvolvimento de *games* e lançou, em 2004, o XNA<sup>69</sup>, um framework<sup>70</sup> de desenvolvimento de jogos para PC com Windows e XBox 360, o console de videogame da empresa. A ferramenta pode ser baixada gratuitamente no site da companhia e conta com um grande conjunto de ferramenta

<sup>69</sup> Detalhes podem ser encontrados em <http://www.xna.com>.

<sup>70</sup> Em desenvolvimento de *software*, um framework ou arcabouço é uma abstração que une códigos comuns entre vários projetos de *software* provendo uma funcionalidade genérica. Um framework pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação. Mais em <http://pt.wikipedia.org/wiki/Framework> (acessado em 16/08/2009).

e uma forte e organizada comunidade<sup>71</sup> de desenvolvedores. A plataforma de desenvolvimento XNA é composta pelos seguintes componentes:

- **XNA Game Studio:** ambiente de desenvolvimento baseado no *Visual Studio*, também da *Microsoft*;
- **XNA FrameWork:** conjunto de classes para executar um jogo feito em XNA;
- **XNA Content Pipeline:** componente de gerência de conteúdo e artefatos de um projeto de jogo;
- **XACT (Audio Authoring Tool):** ferramenta para organização de arquivos de áudio.

O XNA não é um *software* de código aberto, mas conta com ampla documentação desenvolvida e sempre atualizada pela *Microsoft*, o que torna o produto interessante para desenvolvedores que querem focar suas produções em jogos para PC (ambiente *Windows*) e *XBox 360*. Contudo, estas não são as únicas plataformas de jogos, o que torna a ferramenta limitada.

Uma verdadeira mudança de paradigma vem tomando o ambiente de desenvolvimento de jogos. E isso se deve graças ao *Unity 3D*<sup>72</sup>, uma ferramenta versátil, com uma interface intuitiva e com uma infinidade de recursos. Não é uma ferramenta gratuita, mas analisando toda a sua versatilidade, o seu custo se torna baixo. E a famosa relação custo x benefício. E o produtor independente de *games* pode se beneficiar disso.

O ambiente de desenvolvimento do *Unity 3D* integra as principais ferramentas de um motor de jogo em um único ambiente. A interface intuitiva do *software* com recursos de arrastar / soltar. Ou seja, é possível mudar

---

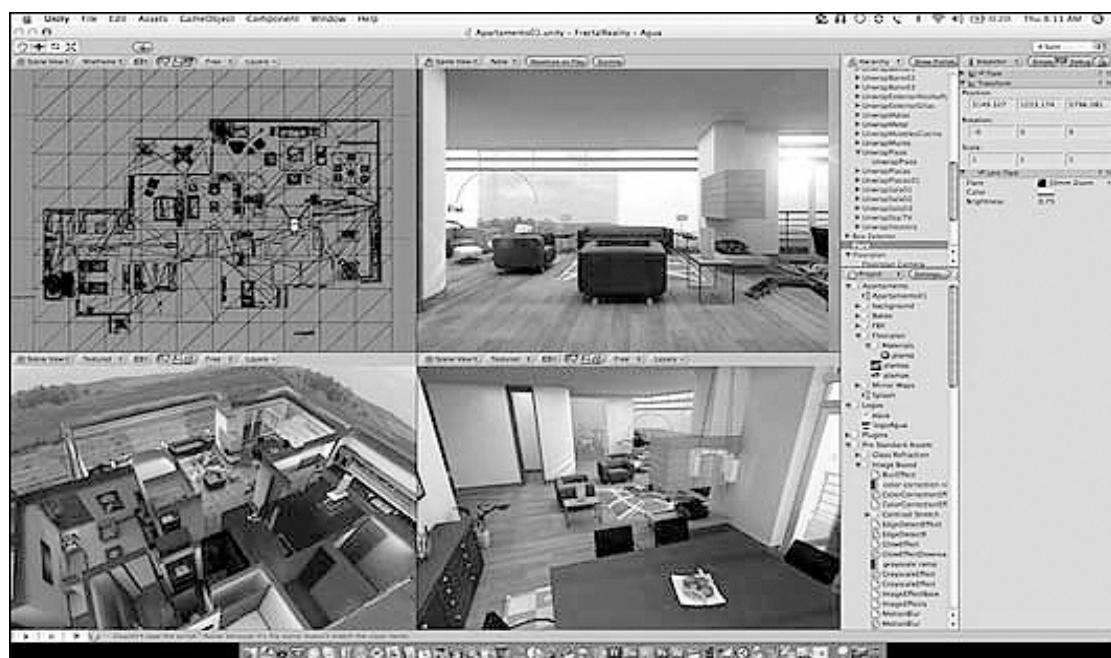
<sup>71</sup> A comunidade XNA está presente aqui: <http://creators.xna.com>

<sup>72</sup> Mais detalhes em <http://unity3d.com/>

texturas de objetos tridimensionais, alteração de áudios específicos, *scripts* e variáveis de jogo visualmente. Só apontar e clicar.

A versatilidade do *Unity 3D* também está aplicada ao seu processamento gráfico que dispõe de integrações com *DirectX* e *OpenGL*. Tal versatilidade está presente no gerenciamento do sistema de partículas, para as mais diversas finalidades e também em sombras e iluminação processadas e renderizadas em tempo real.

O motor de jogo também conta um sistema de gerenciamento de física avançado, que pode ser usado com facilidade em jogos de corrida, personagens, forças, vento, colisão e tudo o mais que um bom *game designer* projetar. Vídeos podem ser reproduzidos em qualquer ambiente do jogo e a codificação de áudio do *software* aceita os principais formatos comumente usados em produções digitais.



**Figura 37:** tela de trabalho do *Unity 3D*

Mas o que faz do *Unity 3D* a ferramenta ideal para o desenvolvedor independente de *games* é sua capacidade de compilar e gerar jogos para as mais diversas plataformas, como: *iPhone*, *Nintendo Wii*, jogos para *web*, jogos em rede, o que amplia a capacidade de atuação destes desenvolvedores iniciantes. Apesar de ser pago, o motor de jogo conta com preços atrativos para qualquer criador de jogos.

Ao abordar o desenvolvimento de jogos digitais focando os aspectos tecnológicos e práticos como: *design* de interface, modelagem 3D em seus principais aspectos, como objetos, cenários e mundos tridimensionais desenvolvidos para *games*, a produção de imagens 2D como texturas e seus similares, ambientação sonora no universo e motor de jogo, relatando suas principais funcionalidades para a produção de *games* fechamos o presente capítulo e um ciclo que começa na criação de jogos digitais, segue para o planejamento e termina aqui, com a produção de *games*.

## CONCLUSÃO

O *game* independente é antes de tudo diretamente ligado à motivação do seu realizador. Sem motivação pessoal, um projeto de um jogo digital independente pode não sair do papel, ou de sua tela inicial. Tendo a motivação de se criar um jogo, o produtor precisa então sistematizar sua produção. Nesta sistematização está presente os fundamentos metodológicos para criação, planejamento e desenvolvimento de jogos digitais.

Aprendemos que o jogo digital se constitui em uma atividade lúdica composta por uma série de ações e decisões, limitada por regras e pelo universo do *game*, que resultam em uma condição final. O *game* é uma hipermídia por excelência e tem na interatividade mediada por aparatos tecnológicos, seu papel fundamental. Esta característica interativa é a dependência de comandos sobre uma interface digital, que faz com que o projeto digital desta natureza não seja um filme ou uma animação, e sim um *game*.

O *game* independente, tratado neste trabalho acadêmico, é um projeto desenvolvido sem aportes financeiros oriundos de grandes empresas. A evolução tecnológica nos mostrou que, à medida que novos recursos técnicos são apresentados ao grande público, é natural que novos *games* independentes surjam periodicamente e os melhores ganhem as páginas da grande imprensa. Estes recursos permitem que, em tempos de produções milionárias na indústria dos *games*, projetos capitaneados por um único autor ainda seja possível. É o *game* autoral.

O desenvolvimento de uma ideia de jogo (seja ele autoral ou produzido por uma grande equipe) geralmente surge de um pequeno conceito a ser expandido. Esta expansão pode se tornar mais clara a medida que se entende a

produção de ideias como um processo tão consolidado quanto a produção de um carro, ou seja, que passa por uma linha de montagem. Em essa linha de montagem, um recurso essencial que ajuda na sistematização e na visão geral do todo são os mapas mentais. Os mapas mentais constituem um conjunto de recursos que ajudam a organizar ideias, argumentos, esboços e tudo o que pode existir durante o processo de produção e organização do pensamento.

Organizar as ideias é o primeiro passo para começar a produção de um *game*. Mas a ideia é só o começo. Esta ideia de jogo se enquadra em um gênero de *game*. É interessante notar que por trás de cada gênero de jogo existe uma mecânica e um gênero narrativo que faz com que o *game* funcione como tal. A mecânica de um jogo é o seu conjunto de regras, possibilidades de ações e decisões, e variedades de respostas do sistema do *game*.

Ora, a partir daí temos que a narrativa clássica é composta por enredo, personagens, espaço, tempo e clímax. E são estes os elementos que estão presentes nas narrativas digitais presentes no *games* que possibilitam que a imersão seja um recurso presente em produções de boa qualidade. E quando o assunto é imersão, o projeto sonoro é peça fundamental, pois trabalha a ambientação de cenários e situações que ajudam na imersão e participação do jogador.

Essa participação geralmente se dá por intermédio de um personagem. E um passo significativo para a criação de um personagem consiste na definição de sua necessidade dramática e deve ser peça marcante em um roteiro. O personagem é o fundamento essencial de um roteiro. O coração, a alma e o sistema nervoso de uma história. É a formatação da ideia do jogo.

Deve-se pensar nas criações de personagens com características psicológicas bem delimitadas de forma a construir um ambiente experimental em que o jogador acredite estar em diálogo ou imerso efetivamente no personagem. E todas as características de um personagem precisam estar alinhadas e descritas em uma ficha de personagem e alinhadas com o universo criado para o jogo digital. Personagens não são meramente representações gráficas, e sim representações gráficas de uma personalidade.

A formatação de uma ideia para um projeto de *game* acontece no documento de *game design* (*game design document - GDD*). Este documento é a espinha dorsal de todo e qualquer projeto de um *game*. É o que define todos os pontos de um jogo digital e guia todas as equipes envolvidas no processo de produção de um jogo digital e o processo dentro e a partir do qual são descritas as características principais do jogo como jogabilidade (palavra que provém ou deriva a expressão inglesa *gameplay*), controles, interfaces, personagens, armas, golpes, inimigos, fases e todos os aspectos gerais do projeto.

É no documento de *game design* que tudo se concentra antes da produção do jogo começar. Este documento usa palavras, tabelas e diagramas para explicar o funcionamento de um *game*, a partir da história do mundo ficcional do *game* até a organização de botões em uma interface e ao modo como um arqueiro luta contra um espadachim onde é descrita também a jogabilidade do *game*. Jogabilidade está diretamente ligada ao modo de jogar um *game* e é composta por um conjunto de características e componentes que influenciam na mecânica do *game* e nas reações do jogador. É no *GDD* também que nasce a estrutura inicial do jogo, composto por história, roteiro, objetivos, premissas, cenário, personagens, condições de vitória, desafios e linha do tempo.

Com o jogo planejado, chega o momento execução do projeto. E a definição da linguagem de programação é muito importante nesta etapa do processo. A linguagem de programação pode estar incorporada ao *engine* usado ou não. *Engine*, ou motor de jogo, são softwares que simplificam a construção de um jogo. Contudo, apesar da grande diversidade de *engines* presentes hoje mercado, as linguagens de programação pouco se alteram.

Linguagem definida, parte-se então para o desenho de interface. A interface, no universo dos *games*, é o ambiente gráfico do produto digital (o jogo), o canal de comunicação do usuário final com o conteúdo, roteiro do *game*, história, *puzzles* e todos os elementos presentes em um jogo digital. É na interface gráfica de usuário que acontece a relação do jogador com o sistema, sendo a interface o elemento que faz a transição entre o real e o digital. A interface de um *game* é tão importante quanto o próprio *game*, pois é a partir deste elemento que o jogador interage e entra no mundo do *game*.

Os elementos tridimensionais, peças que se acoplam à interface no processo de produção de um *game*, se difere da modelagem para animação e cinema e diversos pontos, sendo o principal deles o número reduzidos de polígonos. Polígonos são as linhas e vértices que formam um objeto, personagem ou mapa. Quanto maior o número de polígonos, melhor a qualidade gráfica e maiores os requisitos de *hardware* para que o jogador possa jogar o *game*.

A modelagem 3D, seja ela para construção de mapas, objetos ou personagens permite a representação de pensamentos formais que a linguagem escrita torna proibitiva ou quase impossível. Esta representação só é possível, em uma produção de *games*, com softwares específicos para esta finalidade. A modelagem 3D ganha vida com texturas que são aplicadas ao objetos modelados. Em um ambiente 3D, a textura, combinada com os polígonos de um objeto modelado em um software de composição tridimensional, é o ele-

mento responsável por dar vida e beleza visual às construções poligonais aproximando visualmente, quando necessário, este objeto modelado de uma imagem real. E pode ser produzida com fotografias e *softwares* específicos.

Não menos importante é a produção sonora do *game*. O conjunto de sons presente em um *game* também chamado de paisagem sonora desempenham papel fundamental no processo de imersão no jogo. É nesta paisagem que trabalha-se o silêncio, atmosfera, ambiente, sons externos, sons elementares, sons associados, sons de atrito e efeitos sonoros. E o produtor independente pode usar bibliotecas sonoras prontas para tal fim.

Com tudo em mãos, a fase final é montar todos estes elementos em um motor de jogo, que é o ambiente que permite realizar a integração de toda a diversidade de componentes de um *game*. Existem centenas de motores de *games*, alguns de código aberto, outros que custam dezenas de dólares e ainda outros na custam centenas de milhares de dólares. Tal diversidade no mundo dos *engines* faz com que o produtor independente realize uma escolha criteriosa para seu projeto de jogo digital e é justamente por isso que o pleno entendimento das funcionalidades de um motor de *engine* é extremamente importante. E qual o futuro dos *games*? Para onde andaremos? Quais os próximos passos na área de criação e desenvolvimento?

Para muitos a palavra Holodeck pode parecer estranha e sem sentido, para outros, uma saudosa lembrança da série Stark Trek, ou Jornada nas Estrelas, como ficou conhecida no Brasil. Mas o fato é que: os *games* estão chegando cada vez mais perto do Holodeck, uma tecnologia de projeção 3D, misturada com dispositivos sensoriais, apresentada nos anos 1960 na série televisiva de ficção científica.

Com o avanço da tecnologia computacional, o Holodeck está se aproximando cada vez mais da nossa realidade. Existem hoje tecnologias em desenvolvimento que, uma vez combinadas, poderão levar o jogador definitivamente para dentro do *game*.

Tudo começa com os projetos de realidade virtual, desenvolvido nas décadas de 1970 e 1980, que tem como objetivo recriar a sensação de realidade para o usuário de um sistema computacional. Projetos de realidade virtual geralmente usam um conjunto de dispositivos como luva digital, óculos esteroscópicos, capacete de imersão, teclado, mouse, monitor. Muitos destes projetos estão geralmente associados a *games* e projetos acadêmicos, como é o caso da Caverna Digital, criado na Escola Politécnica da Universidade de São Paulo.

O projeto Natal, apresentado pela *Microsoft* na E3<sup>73</sup> de 2009, consite em sistema de controle que utiliza detecção de movimentos e reconhecimento de voz e transforma o corpo do jogador no controle do *videogame*. Desenvolvido para trabalhar junto com o console da companhia, o *XBox 360*, o dispositivo combina uma câmera, sensor de profundidade, microfone e um processador especial.

Mas o projeto Natal é só o começo. Nomes de peso da indústria cinematográfica como Steven Spielberg e James Cameron, que já trabalham em produções cinematográficas 3D, vêem que o próximo passo é a produção de *games* com esta tecnologia. O próximo filme de Cameron, *Avatar*, com estreia prevista para 18 de dezembro deste ano, utilizará a tecnologia de projeção esteroscópica 3D.

---

<sup>73</sup> E3: maior feira de *games* do mundo.

Esta mesma tecnologia está sendo usada no jogo baseado no filme de Cameron, *Avatar: The Game*<sup>74</sup>. O jogo está sendo desenvolvido paralelamente ao filme pela *Ubisoft*, e é um dos primeiros *games* a contar com imagens esteroscópicas 3D desde o seu desenvolvimento. Este tipo de tecnologia ajuda a aumentar a sensação de imersão, realismo e profundidade, ou seja, o jogador terá a sensação de estar dentro do *game*. O jogo, cercado de segredos, chega às lojas em novembro deste ano.

O 3D, nova mania do cinema, está chegando aos *games* e promete mudar a concepção de jogo. Os jogadores ainda estarão presos a consoles, controles, monitores (ou televisores de alta resolução) e ganharão um novo acessório: um óculos que decodificará as imagens esteroscópicas para recriar a sensação de realidade virtual em sua sala de estar. O realismo será ampliado e continuará sendo projetado em um monitor, mas com a ajuda deste novo acessório, trará a realidade para mais perto: monstros ganharão mais vida, cenários mais profundidade, explosões mais realismo. É o próximo passo rumo ao *Holodeck*.

Contudo, este tipo de tecnologia ainda necessita de alto poder de processamento gráfico. Apenas os consoles de última geração e PC's equipados com placas gráficas de alta qualidade poderão exibir as imagens esteroscópicas que permitem ampliar o conceito de jogo.

A *NVIDIA*, fabricante de placas de vídeo para PC e consoles de *videogame*, já comercializa o *NVIDIA 3D Vision*, um sistema de processamento de imagem que promete, uma vez combinado todos os elementos técnicos, transformar os jogos para PC em ambientes 3D esteroscópicos. Para isso, é preciso o kit da empresa, composto por óculos especiais sem fio, um transmissor infra-

---

<sup>74</sup> Mais detalhes em <http://avatargame.us.ubi.com/>

vermelho, *software* especial, monitor específico no padrão 120 MHz para uso da tecnologia, placa *GeForce* compatível e um PC com *Windows Vista* ou *Windows 7*. Se você for um proprietário de uma placa GeForce da NVIDIA, é possível fazer um teste no site da empresa e verificar sua compatibilidade com o *3D Vision*<sup>75</sup>.

A empresa afirma que mais de 350 *games* disponíveis no mercado são compatíveis com a tecnologia *NVIDIA 3D Vision*, que não exige nenhuma atualização dos jogos. Títulos como *Age of Empires 3*, *Call of Duty 4*, *Guitar Hero 3*, *Spore* e *World of Warcraft* já podem ser jogados com todo o realismo e a sensação de imersão que a tecnologia oferece. Uma lista completa dos *games*, com uma classificação de qualidade criada pela empresa, pode ser encontrada em uma página criada no site da *NVIDIA* especialmente para gamermaníacos<sup>76</sup>.

De olho na crescente onda 3D, a empresa californiana *iZ3D* desenvolveu um monitor especialmente para os gamers de plantão. Com 22 polegadas e resolução de 1680 x 1050 *pixels*, o *iZ3D Monitor* pode ser comprado por US\$ 399,00 nos Estados Unidos. As especificações técnicas do produto afirmar que o monitor oferece suporte para mais de 600 jogos, contudo, no site da empresa, pouco mais de 170 jogos, como por exemplo *Painkiller*, *Quake 3*, *Counter Strike: Source* entre outros<sup>77</sup>.

Fora do mundo dos PCs, a *Sony* já começa a fazer testes com a tecnologia para uso no seu console, o *PlayStation 3*, com os títulos *Gran Turismo* e *Motors-torm: Pacific Rift*. Para *Wii*, o console da *Nintendo*, o game *Battle Rage: The Robot Wars* já está disponível para quem quiser jogar. Para *Xbox 360* já

---

<sup>75</sup> Detalhes aqui: [http://www.nvidia.com/object/GeForce\\_3D\\_compatibility.html](http://www.nvidia.com/object/GeForce_3D_compatibility.html)

<sup>76</sup> Lista de *games*: [http://www.nvidia.com/object/3D\\_Vision\\_3D\\_Games.html](http://www.nvidia.com/object/3D_Vision_3D_Games.html)

<sup>77</sup> Lista de *games*: <http://www.iz3d.com/t-supsupportedgames.aspx>

é possível adquirir simples, rápido e fluído *Invincible Tiger: The Legend of Han Tao* disponível na rede *Xbox Live*.

Não seria um grande exercício de futurologia imaginar que no Natal de 2010 empresas como *Microsoft*, *Sony* e *Nintendo* não estejam comercializando seus famosos consoles com placas de vídeo atualizado com suporte a imagens esteroscópicas 3D. Tudo depende, claro, da popularização das TV's que suportem a tecnologia.

E para onde vamos? A combinação de tecnologias como as imagens esteroscópicas 3D, óculos especiais e um sistema de reconhecimento de movimentos como o projeto Natal prometem ser a nova barreira a ser quebrada rumo a imersão total nos *videogames*. E isso não vai demorar quase nada.

A pesquisa para nossa dissertação de mestrado levou para todas estas questões relacionadas à esterografia e ao *games*, o que pretende agora pesquisar em uma pesquisa de doutoramento.

*Game over?* Não! A jornada até aqui foi só um começo.



## APÊNDICES

**Apêndice 1:** entrevista realizada em 1999 por Renato Degiovani por ocasião do lançamento do game brasileiro *Incidente em Varginha*.

### Os alienígenas estão chegando

*Por Renato Degiovani*

Jogos brasileiros são raros. Com temática brasileira então, nem se fala. Então é um duplo acontecimento travar contato com o pessoal do *Incidente em Varginha* que, independente de qualquer outra análise, é um jogaço que não perde em funcionalidade ou qualidade gráfica para nenhum jogo 3D, baseado na primeira pessoa.

A equipe do Incidente, Marcos Cuzziol, Odair Gaspar, Fábio Cardelli, Hans Veríssimo, Rod Reis e Rogério Vilela está de parabéns por esta bela produção. Digna de se tornar um clássico nacional, não apenas pelas suas qualidades técnicas, mas pelo oportunismo e interesse que o tema desperta: yes, nós também temos nossos Arquivos-X.

Marcos Cuzziol nos concedeu uma entrevista e os leitores da TILT poderão agora conhecer um pouco mais sobre os bastidores deste incrível jogo. Especialmente aqueles que pretendem lançar-se à produção de jogos nacionais.

### TILT - Quais as ferramentas e programas que mais foram usados, na construção do jogo?

Para o jogo propriamente dito, licenciamos o mecanismo gráfico *Acknex*, que utiliza uma linguagem de programação do tipo *script*, parecida com o *C*. Outro produto licenciado foi o *Smacker*, um compactador e *player* para animações e vídeo. Para a modelagem 3D e renderização das animações utilizamos o *TrueSpace*, enquanto que gráficos e texturas foram elaborados no *PhotoStyler* e *PaintShop*. Também utilizamos o *Poser*, para gerar *sprites* de

personagens. Todos são *softwares* de bom desempenho (uns mais, outros menos) e preço bastante acessível, fatores que levamos em consideração desde o princípio.

Outra ferramenta muito utilizada foi o compilador *Watcom C/C++*. Geramos vários programinhas paralelos com ele, para otimização de paletas de cores e treinamento da inteligência dos personagens, por exemplo.

**TILT - Sendo um jogo baseado na primeira pessoa, é impossível não ligá-lo aos jogos famosos, baseados nesta estrutura funcional. O que o distingue desses seus "concorrentes", além da temática brasileira, já que pelo menos teoricamente ele concorre na mesma faixa que *Quakes* e assimelhados?**

Honestamente, acho que a maior diferença está no volume de investimentos e de recursos para o desenvolvimento. Veja bem, não digo que o *Incidente em Varginha* deva figurar lado a lado com *Quake* e afins em nível de tecnologia, sabemos que alguns aspectos técnicos do jogo exigem melhorias, e estamos trabalhando nisso. Entretanto, tenho certeza de que a nossa relação entre custo de desenvolvimento e qualidade final foi muito mais eficiente, e isso nos deixa extremamente confiantes para o futuro. Outra diferença, talvez não tão evidente a primeira vista, é o tipo de inteligência artificial empregada. Uma mistura de algoritmos genéticos e máquinas de estados finitos permitem aos personagens alguns comportamentos emergentes como cercar o jogador, atacar em grupo ou armar emboscadas, por exemplo.

**TILT - Numa entrevista ao *The Kingmob Journal* vocês afirmam que a resolução adotada (320 x 400) foi usada em decorrência de uma maior compatibilidade com sistemas mais antigos, baseados em DOS. Fale-nos um pouco desta decisão e suas consequências para o desenvolvimento técnico do jogo - facilitou ou complicou a parte técnica; como é o rela-**

cionamento com a "parceira" *Conitec Datensysteme*, responsável pelo *Acknex*?

Realmente, essa não foi uma decisão fácil. Para fazer uso do *DirectX* teríamos que adotar uma resolução padrão mínima de 320 x 240, ou de 640 x 480, que exigiria uma máquina muito mais "pesada". Além disso, o *Acknex* tinha suporte para *DirectDraw*, e não *Direct3D*, o que limita qualquer aceleração por *hardware*. Então optamos por uma resolução intermediária, sem exigir *hardware* especial, pois esse era um perfil mais compatível com os PCs brasileiros da época. Em termos de desenvolvimento isso simplificou bastante alguns detalhes, permitindo inclusive que o jogo rodasse diretamente do CD, sem nenhuma configuração ou instalação.

O relacionamento com a *Conitec* foi sempre muito profissional. Nesse aspecto o projeto correu bem tranquilo, sem grandes surpresas e sem *bugs* inesperados no mecanismo, que exigissem correção imediata.

**TILT - Uma das críticas mais ouvidas, dos nossos assessores e colaboradores que tiveram acesso ao demo, é a de que o movimento dos soldados deixa transparecer sua condição de sprites (aliás uma característica bastante comum nos sistemas 3D, para agilizar a renderização das cenas). Isto decorreu da estrutura interna do *Acknex* ou foi uma opção conceitual da equipe?**

A versão do *Acknex* que utilizamos é o que chamamos de um mecanismo 2,5 D - não é exatamente um mecanismo 3D real, pois falta um grau de liberdade na movimentação do jogador (são cinco no total, ao invés de seis) e não existe perspectiva vertical, apenas uma projeção paralela. Tecnicamente, ele é um *raycaster* como o do *Duke Nukem 3D*, porém com suporte a modelos tipo *Quake*. O resultado de tudo isso é uma velocidade não muito otimizada para os modelos poligonais. Utilizamos atores poligonais onde eles eram mais necessários: automóveis, orelhões, mobília, luminárias, naves, etc. Os personagens tipo sprite 3D foram criados para que pudéssemos em-

pregá-los em grande número sem comprometer a velocidade de execução. No nível da Praça da Sé, por exemplo, podemos observar mais de 50 personagens simultaneamente, de um total de 212. Isso não seria viável sem a utilização os *sprites 3D*.

**TILT - Demos uma passadinha no site da *Cia do Software* (o distribuidor do jogo) e notamos que, além da venda virtual, o jogo pode ser encontrado em algumas lojas de São Paulo e umas poucas do Rio Grande do Sul e somente uma loja do Piauí e de Minas Gerais. Isto não dificulta a comercialização do programa? Até que ponto os distribuidores, do seu ponto de vista, estão aptos a efetivamente responder à comercialização desse tipo de produto? Fale-nos um pouco sobre essa relação.**

Além da *Cia do Software*, o jogo também é distribuído pela *Top* no interior de São Paulo, pela *Unibyte* no sul do Brasil e pela *Paradigma* no Rio. Mas, apesar da boa vontade de todos, a distribuição ainda é nosso maior problema. Muita gente nos procura dizendo que não consegue encontrar o jogo em lugar algum.

Em minha opinião, a mentalidade de distribuição desse tipo de produto no Brasil ainda não se adaptou à nossa nova realidade. Acho que não podemos mais viver apenas da exploração de nichos de mercado criados no exterior, das migalhas que caem da mesa de países mais abastados, de produtos que já tem demanda garantida aqui no Brasil simplesmente porque fizeram sucesso lá fora. Mesmo as grandes empresas estrangeiras já perceberam isso, e algumas esforçam-se para criar produtos diferenciados para o Brasil, apostando no nosso mercado. Está mais do que na hora de criarmos também os nossos próprios caminhos, de competirmos internacionalmente, de fazermos uso de todo o nosso potencial. Agora, é óbvio que essa atitude exige dedicação extrema, e que o retorno comercial é muito baixo, pelo menos no início. Ainda assim, essa é a opção mais sensata a médio prazo. Nossa história já provou várias vezes que o "extrativismo" pode render muito dinheiro no

começo, mas provou também que essa festa dura pouco tempo e que, inviavelmente, termina em prejuízo.

### **TILT - Que expectativas tinham ou tem em relação ao desempenho comercial do jogo?**

Antes de colocarmos as mãos na massa, fizemos uma pesquisa sobre os resultados da venda de jogos 3D no Brasil. Ainda que pouco confiáveis, os números eram, em geral, desalentadores: 5 mil cópias no máximo para um ótimo produto, 10 mil para um autêntico *best seller*, e isso apenas para os que tinham uma distribuição muito forte. Outro ponto desagradável: estima-se que a pirataria represente mais de 65% do número total de cópias - em outras palavras, 65% das cópias - ou mais - são furtadas, pura e simplesmente (e muita "gente boa" ainda se orgulha disso!).

Por esses motivos, não apostamos num grande desempenho comercial. Tentamos alcançar o máximo de qualidade possível com um custo de desenvolvimento que pudesse ser recuperado dentro de um número relativamente modesto de cópias vendidas. Acima de tudo, nosso objetivo não se limita ao primeiro produto. Os números que citei ainda são muito baixos, é verdade, mas estão crescendo num ritmo alucinante. Crises à parte, a verdade é que hoje em dia já se fala em alguns poucos jogos ultrapassando a casa de 30 ou 50 mil cópias vendidas.

### **TILT - Valeu a pena a experiência?**

Sem dúvida. Além de todo o aprendizado e da experiência, é reconfortante saber que criamos alguma coisa nova. É ótimo receber emails de incentivo de todos os cantos do Brasil, e saber que o jogo está cumprindo o seu papel principal, que é o de entreter as pessoas.

Fizemos também vários contatos importantes, e alguns estão se convertendo em novos projetos. Cópias preliminares em inglês do *Incidente em Varginha* foram solicitadas e estão agora sendo avaliadas por *publishers* na Alemanha, França, Dinamarca, Suécia, Holanda e Grécia. O *demo* esta sendo publicado em revistas especializadas nos EUA e na Suécia. Nada mal para um primeiro projeto, diga-se de passagem. Tudo isso nos incentiva a continuar nosso trabalho, ainda que comercialmente os resultados não sejam muito animadores aqui no Brasil.

**TILT - Já tem algum novo projeto em andamento?**

É claro! Afinal, somos uma empresa nova e estamos apenas esquentando os motores, tentando aprender o máximo possível e evoluir rapidamente para poder - quem sabe? - competir com as gigantes do ramo. Temos algumas pequenas surpresas preparadas para 1999. Não estou tentando fazer suspense, adoraria contar tudo sobre os novos projetos. O caso é que, para viabilizarmos comercialmente novos produtos, tivemos que buscar o apoio de outras empresas. Por força desses acordos, ainda não posso divulgar detalhes. Mas garanto que as surpresas virão. E logo.

# Título do Jogo

## Subtítulo do jogo

### Sumário

### 1- Histórico do Projeto

Escreva uma breve descrição das versões e mudanças ocorridas durante o projeto desde o início.

### 2 - Resumo do Projeto

#### 2.1 - Conceito do Jogo

Descreva um resumo do principal conceito do jogo(Ex.: O jogo baseia-se em...).

#### 2.2 - Conjunto de características

Descreva um resumo das plataformas adotadas, número de níveis, tipos de jogabilidade(Ex. Modo Arcade, Estória etc.), modos de visualização(Ex. 2D, 3D, Sonoro etc), características de cores(Ex. 16 bits, 32 bits etc), Física (Ex. Básica, intermediária, Avançada).

#### 2.3 - Gênero

Descreva um resumo do gênero do jogo(Ex. RPG, FPS, RTS, Aventura, Ação, Simulação etc).

#### 2.4 - Público-alvo

Descreva qual o público que você quer atingir (Ex. Jovens entre 12 e 17 anos, Adultos do sexo feminino etc).

#### 2.5 - Resumo do Fluxo do Jogo

Descreva resumidamente o processo do jogo como um todo, se perguntando: Como o jogador se move pelo jogo? Como ele pega os itens? Como ele ganha vida / energia? etc.

## 2.6 - Olhar e Sentir

Descreva qual será a visão do Jogador, os momentos em que a visão muda e o estilo visual do jogo.

## 2.7 - Escopo do Projeto

Descreva um resumo do escopo do jogo:

- 2.7.1 - Número de cenários;
- 2.7.2 - Número de níveis;
- 2.7.3 - Número de *NPCs*;
- 2.7.4 - Número de armas;
- 2.7.5 - Etc

# 3 - Jogabilidade e Mecânica

## 3.1 - Jogabilidade

### 3.1.1 - Progressão do Jogo

### 3.1.2 - Estrutura das Missões / Desafios

### 3.1.3 - Estruturas dos Quebra-cabeças

### 3.1.4 - Objetivos

Descreva os objetivos do jogo

### 3.1.5 - Fluxo do Jogo

Descreva como o jogo flui para o jogador.

## 3.2 - Mecânicas

Descreva as regras do jogo (implícitas e explícitas). Este é o modelo do universo no qual o jogo funciona. Pense em simulação do mundo do jogo e como todos os pedaços interagem entre si. Geralmente este é a parte mais longa desta seção. Esta seção é referente ao(s) personagem(ns) do jogo e seu universo.

### 3.2.1 - Física do jogo

Descreva como a física afeta o universo e os objetos no jogo.

### **3.2.2 - Movimentos**

- 3.2.2.1 - Movimentos Gerais
- 3.2.2.2 - Movimentos específicos
- 3.2.2.3 - Outros movimentos

### **3.2.3 - Objetos**

- 3.2.3.1 - Pegando objetos
- 3.2.3.2 - Movendo objetos
- 3.2.3.3 - Descartando objetos
- 3.2.3.4 - Modificando objetos

### **3.2.4 - Ações**

- 3.2.4.1 - Interruptores, alavancas e botões
- 3.2.4.2 - Pegando, Carregando e Soltando
- 3.2.4.3 - Falando e Conversando
- 3.2.4.4 - Lendo e Pensando

### **3.2.5 - Combates**

Descreva como são os combates ou eventuais conflitos. Modele especificamente o fluxo do início ao fim.

### **3.2.6 - Economia**

Descreva qual a economia do jogo e como ela funciona.

### **3.2.7 - Planilha de Fluxo de Telas**

Descreva graficamente como uma tela se relaciona com as demais.

### **3.2.8 - Descrição de Telas**

Descreva a proposta de cada tela.

- 3.2.8.1 - Tela de instalação
- 3.2.8.2 - Tela Principal do jogo
- 3.2.8.3 - Tela Opções
- 3.2.8.4 - Etc

## **3.4 - Opções do jogo**

Quais as opções e como elas afetam a jogabilidade e a mecânica?

### **3.5 - Re-jogando e Salvando o jogo**

### **3.6 - Códigos de trapaça(*Cheat-codes*) e procedimentos escondidos(*Easter-eggs*)**

## **4 - Enredo, Universo e Personagens**

### **4.1 - Enredo e Narrativa**

Descreva resumidamente o enredo do jogo. Detalhes específicos como *Script* e corte de cenas, são descritos em uma *Story Bible*.

#### **4.1.1 - Prelúdio**

#### **4.1.2 - Elementos do enredo**

#### **4.1.3 - Progressão do Jogo**

#### **4.1.4 - Corte de Cenas**

##### **4.1.4.1- Corte de cena 1**

Atores

Descrição

*Storyboard*

*Script*

##### **4.1.4.2 - Corte de cena 2**

...

### **4.2 - Universo do Jogo**

#### **4.2.1 - Impressões gerais do universo do jogo**

#### **4.2.2 - Área 1**

Descrição Geral

Características físicas (Ex.: visuais, sonoros etc)

Níveis utilizados na Área

Conexões com outras Áreas

#### **4.2.3 - Área 2**

...

### **4.3 - Personagens**

#### **4.3.1 - Personagem 1**

##### **4.3.1.1 - Prelúdio**

- 4.3.1.2 - Personalidade
- 4.3.1.3 - Aparência
  - Características físicas
  - Animações
- 4.3.1.4 - Habilidades especiais
- 4.3.1.5 - Relevância no Enredo do Jogo
- 4.3.1.6 - Relacionamentos com outros personagens
- 4.3.1.7 - Estatísticas (Ex.: Frequência em que o personagem aparece)

#### 4.3.1 - Personagem 2

...

### 4.4 - Referências

Coloque documentos, livros, filmes, vídeos, apêndices deste documento, outros jogos ou atas de reunião como referência da narrativa e enredo.

## 5 - Níveis

### 5.1 - Nível 1

- 5.1.1 - Resumo
- 5.1.2 - Material introdutório (Ex.: Cortes de cena, *Breafing* de missão, vídeos ou textos)
- 5.1.3 - Objetivos
- 5.1.4 - Descrição física
- 5.1.5 - Mapa
- 5.1.6 - Caminho crítico (Ex.: Caminho em que o personagem pode encontrar um inimigo)
- 5.1.7 - Encontros
- 5.1.8 - Nível passo-a-passo
- 5.1.9 - Finalização do material (Ex.: Cortes de cena, check de objetivos, vídeos ou textos)

### 5.2 - Nível 2

...

### 5.n - Nível de treinamento

## 6 - Interface

### 6.1 - Sistema Visual

- 6.1.1 - *HUD(Head-Up Display)* - O que controlar?
- 6.1.2 - Menus
- 6.1.3 - Sistema de Renderização
- 6.1.4 - Câmera
- 6.1.5 - Modelos de Iluminação

### 6.2 - Sistema de Controle

Descreva como o jogador pode controlar o jogo. Existem comandos específicos? Existem comandos ocultos para o jogador?

### 6.3 - Sistema de Áudio

Descreva se o áudio é em Mono, Stereo, 2D ou 3D.

- 6.3.1 - Músicas
- 6.3.2 - Efeitos sonoros

### 6.4 - Sistema de Ajuda

## 7 - Inteligência Artificial

### 7.1 - IA de Oponentes

Descreva os oponentes ativo que interagem contra o jogador, no qual, reagem às ações estratégicas feitas por ele(Ex.: Xadrez, jogo-da-velha etc).

### 7.2 - IA de Inimigos (Vilões e Monstros)

### 7.3 - Personagens Não-Combatentes

### 7.4 - Personagens Amigáveis

### 7.5 - IA de suporte

- 7.5.1 - Colisões do jogador e objetos
- 7.5.2 - Melhor caminho (*Pathfinding*)

## 8 - Projeto Técnico

Os termos técnicos podem ser descritos em formato de glossário no *Technical Bible*.

### 8.1 - Equipamento-alvo

Descreva o ambiente recomendado em que o jogo deverá ser instalado depois de produzido, indicando, também, os requisitos mínimos de *hardware* e *software* do jogador.

### 8.2 - Ambiente desenvolvido (*Hardware e Software*)

Descreva detalhadamente em que ambiente o jogo é desenvolvido, inclusive, Sistema Operacional, Memória, *Hard Disk* e suas versões.

### 8.3 - Procedimentos e padrões de Desenvolvimento

### 8.4 - Motor do Jogo (*Engine*)

Descreva qual a engine utilizada para criar o jogo e sua versão.

### 8.5 - Rede

Descreva o ambiente de rede em que o jogo está, expondo servidores (no caso de *Multiplayers* e *MMOs*), se será via internet, apenas *intranet* ou *VPN*, entre outros.

### 8.6 - Linguagem de programação

O código-fonte comentado é inserido na *Script Bible* produzido pela equipe de programadores.

## 9 - Projeto Artístico

Os termos artísticos podem ser descritos em formato de glossário no *Art Bible*. Para projetos musicais é necessário criar um documento específico. Nesta seção podem ser inseridos arquivos de imagens.

### 9.1 - Arte conceitual

### 9.2 - Guias de Estilo

### 9.3 - Personagens

### 9.4 - Ambientes

### 9.5 - Equipamentos

### 9.6 - Cortes de Cena

### 9.7 - Miscelânea

## **10 - Softwares Secundários**

**10.1 - Editores** (Ex.: Modelagem 2D ou 3D, sons, músicas)

**10.2 - Instaladores**

**10.3 - Atualização de programas**

**10.4 - Miscelânea**

## **11 - Gerenciamento**

**11.1 - Detalhes do Cronograma**

**11.2 - Orçamento**

**11.3 - Considerações de Licença**

**11.4 - Análise de Risco**

**11.5 - Plano de Locação** (Lugar a ser vendido)

**11.6 - Plano de Teste**

## **12 - Equipe**

Coloque os créditos da equipe do projeto, identificando o que cada pessoa ou empresa terceirizada faz.

## **13 - Apêndices**

É recomendado escrever o caminho do arquivo com extensão para referências, inclusive.

### **13.1 - Ativos de Arte**

Lista de Modelos e Texturas

Lista de animações

Lista de efeitos

Lista de interface artística

Lista de cortes de cena

### **13.2 - Ativos de Som**

Sons de ambiente

Sons de armas

Sons de interface

### **13.3 - Ativos de Música**

Ambiente

“Ação”

Vitória

Derrota

#### **13.4 - Ativos de Vozes**

Ator 1 - Linha de voz

Ator 2 - Linha de voz

Ator 3 - Linha de voz

Em nn / nn / nnnn

---

**Nome do Gerente do Projeto**

Gerente do Projeto

Em nn / nn / nnnn

---

**Nome do Coordenador Técnico**

Coordenador Técnico

Em nn / nn / nnnn

---

**Nome do Coordenador Artístico**

Coordenador Artístico



## **REFERÊNCIAS**

### **Bibliografia**

ASSIS, Jesus de Paula. **Artes do videogame: conceitos e técnicas.** São Paulo, Alameda, 2007.

AZEVEDO, Eduardo (org.). **Desenvolvimento de jogos 3D e aplicações em realidade virtual.** Rio de Janeiro, Elsevier, 2005.

BAIRON, Sérgio. **Multimídia.** São Paulo, Global, 1995.

BARBARO, Umberto. **Argumento e Roteiro.** São Paulo, Global Editora, 1983.

BLAIR, Preston. **Cartoon animation.** California, Walter Foster Publishing, 1995.

BEARID, Jason. **Princípios do web design maravilhoso.** Rio de Janeiro, Alta Books, 2008.

BONSIEPE, Gui. **Design: do material ao digital.** Florianópolis, FIESC, SEBRAE-SC, 1997.

CLUA, Esteban Walter Gonzalez. **Como funciona um game 3-D?** In: BOBANY, Arthur. **Vídeo Game Arte.** Teresópolis, RJ, Novas Ideias, 2008.

CLELAND, David I. e IRELAND, Lewis R. **Gerência de projetos.** Rio de Janeiro, Reichmann & Affonso Editores.

DIVERIO, Tiarajú Asmuz. e MENEZES, Paulo Blauth. **Teoria da computação: máquinas universais e computabilidade.** Porto Alegre, Editora Sagra Luzzatto, 2000.

DONDIS, Donis A. **Sintaxe da linguagem visual**. São Paulo, Martins Fontes, 1997.

FIELD, Syd. **Manual do roteiro**. Rio de Janeiro, Objetiva, 1995.

**Game Brasilis: catálogo de jogos eletrônicos brasileiros**. São Paulo, Faculdade Senac de Comunicação e Artes, 2003.

GOSCIOLA, Vicente. **Roteiro para as novas mídias: do game à TV interativa**. São Paulo, Editora Senac São Paulo, 2003.

GOMES, Renata. **Imersão e participação nos jogos eletrônicos**. PUC-SP, 2003. Dissertação de Mestrado em Comunicação e Semiótica.

GOMES, Paulo César Rodacki. e PAMPLONA, Vitor Fernando. **M3GE: um motor de jogos 3D para dispositivos móveis com suporte a Mobile 3D Graphics API**. In: Workshop Brasileiro de Jogos e Entretenimento Digital, 2005, Porto Alegre. Anais WJOGOS, 2005.

GURGEL, Ivannoka. e PADOVANI, Stephania. **Processo de Criação de Personagens: Um Estudo de Caso no Jogo Sério SimGP**. In: Simpósio Brasileiro de Jogos para Computador e Entretenimento Digital - SBGames, 2006, Recife. Simpósio Brasileiro de Jogos para Computador e Entretenimento Digital, 2006.

HIGGIN, Brian C. Áudio. In: AZEVEDO, Eduardo (org.). **Desenvolvimento de jogos 3D e aplicações em realidade virtual**. Rio de Janeiro, Elsevier, 2005.

HILLS, Daniel. **O padrão gravado na pedra: as ideias simples que fazem os computadores funcionarem**. Rio de Janeiro, Rocco, 2000.

HOUAISS, Antônio. **Dicionário Houaiss da língua portuguesa**. Rio de Janeiro, Objetiva, 2001.

HUIZINGA, J. **Homo Ludens**. Tradução de J.P. Monteiro. São Paulo, Perspectiva, 2004, 1938.

HURLBURT, Allen. **Layout: o design da página impressa**. São Paulo, Nobel, 2002.

LEÃO, Lucia. **O Labirinto da hipermídia - arquitetura e navegação no ciberspaço**. São Paulo, Iluminuras, 1999.

LEMOS, André. **Cibercultura, tecnologia e vida social na cultura contemporânea**. Porto Alegre, Sulina, 2002.

LOPES FILHO, Eliseu de Souza. **Animação e hipermídia - Trajetória da luz e sombra aos recursos midiáticos**. PUC-SP, 2007. Dissertação de Mestrado em Comunicação e Semiótica.

MARCELO, Antonio. e PESCUITE, Julio. **Design de jogos - fundamentos**. Rio de Janeiro, Brasport, 2009.

MCLUHAN, Marshall. **Os meios de comunicação como extensões do homem**. São Paulo, Editora Cultrix, 2005.

MCCLOUD, Scott. **Desvendando os Quadrinhos**. São Paulo, Makron Books, 1995.

\_\_\_\_\_. **Reinventando os Quadrinhos**. São Paulo, M. Books, 2006.

MEMÓRIA, Felipe. **Design para a Internet: projetando a experiência perfeita**. Rio de Janeiro, Elsevier, 2005.

MURRAY, Janet H. **Hamlet no Holodeck: o futuro da narrativa no ciberespaço**. São Paulo, Itaú Cultural : Unesp, 2003.

MUSBURGER, Robert B. **Roteiro para mídia eletrônica**. Rio de Janeiro, Elsevier, 2008.

NESTERIUK, Sérgio. **A narrativa do jogo na hipermídia: a interatividade como possibilidade comunicacional.** PUC-SP, 2002. Dissertação de Mestrado em Comunicação e Semiótica.

PERUCIA, Alexandre Souza.; BERTHÊM, Antônio Córdova de.; BERTSCHINGER, Guilherme Lage e MENEZES, Roberto Ribeiro Castro. **Desenvolvimento de Jogos Eletrônicos - Teoria e Prática.** São Paulo, Novatec, 2005.

PLAZA, Julio. **Tradução Intersemiótica.** São Paulo, Perspectiva, 1987.

PRADO, Sandro Luiz Nhaia de. e STELKO, Michelle. **Modelagem de Personagens.** In: AZEVEDO, Eduardo (org.). **Desenvolvimento de jogos 3D e aplicações em realidade virtual.** Rio de Janeiro, Elsevier, 2005.

PREECE, Jennifer.; ROGERS, Yvone e SHARP, Helen. **Design de interação: além da interação homem-computador.** Porto Alegre, Bookman, 2005.

RADFAHRER, Luli. **Design/Web/Design: 2.** São Paulo, Market Press, 2000.

RANHEL, João. **O conceito de jogo e os jogos computacionais.** In: SANTAELLA, Lucia. e FEITOSA, Mirna. (orgs.). **Mapa do Jogo - A diversidade cultural dos games.** São Paulo, Cengage Learning, 2009.

RABELO, Cláudio. Game Design. In: AZEVEDO, Eduardo (org.). **Desenvolvimento de jogos 3D e aplicações em realidade virtual.** Rio de Janeiro, Elsevier, 2005.

REINICKE, José Fernando. **Modelando personagens com o Blender 3D.** São Paulo, Novatec Editora, 2008.

ROLLINGS, Andrew e MORRIS, Dave. **Game Architecture and Design.** A New Edition. Indianapolis, USA, New Riders, 2004.

SATO, Adriana Kei Ohashi. **Do mundo real ao mundo ficcional: a imersão no jogo.** In: SANTAELLA, Lucia. e FEITOSA, Mirna. (orgs.). *Mapa do Jogo - A diversidade cultural dos games*. São Paulo, Cengage Learning, 2009.

SCHAFER, Raymond Murray. **Afinação do mundo.** São Paulo, Fundação Editora da Unesp, 1997.

SHUM, Lawrence Rocha. **Paisagens sonoras nos games.** In: SANTAELLA, Lucia. e FEITOSA, Mirna. (orgs.). *Mapa do Jogo - A diversidade cultural dos games*. São Paulo, Cengage Learning, 2009.

SANTAELLA, Lúcia. **Linguagens líquidas na era da mobilidade.** São Paulo, Paulus, 2007.

\_\_\_\_\_. **Navegar no ciberespaço: o perfil cognitivo do leitor imersivo.** São Paulo, Paulus, 2004.

\_\_\_\_\_. **Matrizes da Linguagem e Pensamento - Sonora Visual Verbal.** São Paulo, Iluminuras, 2001.

SANTEE, André. **Programação de Jogos com C++ e DirectX.** São Paulo, Nocatec, 2005.

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

SILVA, Luciano Augusto da. **Level Design.** In: AZEVEDO, Eduardo (org.). **Desenvolvimento de jogos 3D e aplicações em realidade virtual.** Rio de Janeiro, Elsevier, 2005.

SIQUEIRA, João Hilton Sayeg. **Organização textual da narrativa.** São Paulo, Selinunte, 1992.

TAVARES, Roger. e NEVES, Felipe. **Introdução ao Game Design através da modificação de Unreal 2004.** In: Simpósio Brasileiro de Jogos para Compu-

tador e Entretenimento Digital - SBGames, 2006, Recife. Simpósio Brasileiro de Jogos para Computador e Entretenimento Digital, 2006.

VELHO, João. e NIGRO, Márcio. **Vídeo & Áudio Digital no Macintosh**. São Paulo, Editora Bookmakers, 2002.

VICENTE, Victor. **Análise das Estruturas Interativas em Jogos Multiplayer: Caso Counter-Strike**. PUC-SP, 2002. Dissertação de mestrado em comunicação e semiótica.

VILLAS BÔAS, Rafael. **Mercado de Jogos**. In: AZEVEDO, Eduardo (org.). Desenvolvimento de jogos 3D e aplicações em realidade virtual. Rio de Janeiro, Elsevier, 2005.

VAUGHAN, Tay. **Multimídia na prática**. São Paulo, Makron Books, 1994.

WATTS, Harris. **On camera: o curso de produção de filme e vídeo da BBC**. São Paulo, Summus, 1990.

YOUNG, James Webb. **Técnica para produção de idéias**. São Paulo, Nobel, 1994.

### **Webografia**

DEGIOVANI, Renato. **Os alienígenas estão chegando**. Tilt Online ([http://www.tilt.net/wiki/index.php/Os\\_alienígenas\\_estão\\_chegando](http://www.tilt.net/wiki/index.php/Os_alienígenas_estão_chegando)) - acessado em 13 de outubro de 2008.