

**UNIVERSIDADE FEDERAL DE GOIÁS – UFG**

**CAMPUS CATALÃO – CaC**

**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO – DCC**

Bacharelado em Ciência da Computação

Projeto Final de Curso

**Jogos sérios: tecnologia de jogos por computador  
aplicada ao ensino e aprendizagem.**

Autor: Gustavo Pereira de Cuba

Orientador: Alexsandro Santos Soares

**Gustavo Pereira de Cuba**

**Jogos sérios: tecnologia de jogos por computador aplicada ao ensino e aprendizagem.**

Monografia apresentada ao Curso de  
Bacharelado em Ciência da Computação da  
Universidade Federal de Goiás Campus Catalão  
como requisito parcial para obtenção do título de  
Bacharel em Ciência da Computação

**Área de Concentração:** Softwares como Ferramentas Educacionais

**Orientador:** Alexsandro Santos Soares

P. de Cuba, Gustavo

**Jogos sérios: tecnologia de jogos por computador aplicada ao ensino e aprendizagem./Alexsandro Santos Soares- Catalão - 2009**

Número de páginas: 79

Projeto Final de Curso (Bacharelado) Universidade Federal de Goiás, Campus Catalão, Curso de Bacharelado em Ciência da Computação, 2009.

Palavras-Chave: 1. Jogos Eletrônicos. 2. Computação Gráfica. 3. Software Educacional

**Gustavo Pereira de Cuba**

**Jogos sérios: tecnologia de jogos por computador aplicada ao ensino e aprendizagem.**

Monografia apresentada e aprovada em \_\_\_\_\_ de \_\_\_\_\_

Pela Banca Examinadora constituída pelos professores.

---

Alexsandro Santos Soares – Presidente da Banca

---

Professor 1

---

Professor 2

Dedico esse trabalho aos meus pais, irmãos, colegas e todos aqueles que, direta ou indiretamente, contribuíram para a realização desse trabalho.

## AGRADECIMENTOS

Agradeço primeiramente a Deus, por criar as condições necessárias para que eu desenvolvesse esse trabalho.

Aos meus pais, José e Lenir, pela força e o incentivo, sem o qual teria sido difícil concluir esse trabalho.

A meus irmãos, Augusto e Frederico , pela ajuda que me deram.

A todos os meus colegas de turma principalmente aos meus amigos e companheiros Diego, Eustáquio e Hugo.

Ao professor Alexandro, pela oportunidade que me deu de conhecer coisas novas.

## RESUMO

**Cuba, G. Jogos sérios: tecnologia de jogos por computador aplicada ao ensino e aprendizagem..** Curso de Ciência da Computação, Campus Catalão, UFG, Catalão, Brasil, 2009, 79p.

O desenvolvimento de jogos digitais conta hoje com recursos sofisticados de animação e interação, as quais conseguem atrair a atenção e o interesse de pessoas de todas as idades. O desafio proporcionado pela trama, a sofisticação das cenas e a facilidade de interação são fatores críticos para o sucesso de um jogo atual. Além disso, características, como estimular a repetição de exercícios, exigir concentração entre outros, abrem espaço aos jogos eletrônicos para a educação como uma ferramenta de auxílio à aprendizagem. Este trabalho descreve sobre o desenvolvimento de um jogo educativo voltado ao ensino das leis de trânsito, que tem como base critérios envolvidos nos jogos eletrônicos e conceitos e teorias da aprendizagem.

**Palavras-Chaves:** Jogos Eletrônicos, Computação Gráfica, Software Educacional

# *Sumário*

## **Lista de Figuras**

<b>1</b>	<b>Introdução</b>	p. 1
1.1	Contexto . . . . .	p. 1
1.2	Objetivo . . . . .	p. 2
1.3	Estrutura do Trabalho . . . . .	p. 3
<b>2</b>	<b>Jogos Eletrônicos: possibilidades de desenvolvimento e aprendizagem</b>	p. 4
2.1	Todos os jogos são sérios . . . . .	p. 4
2.2	Jogos Eletrônicos, a arte eletrônica . . . . .	p. 7
2.3	Tipo de Jogo . . . . .	p. 8
2.4	Jogos Sérios . . . . .	p. 11
2.4.1	Definindo Jogos Sérios . . . . .	p. 12
2.4.2	A importância do divertimento . . . . .	p. 13
2.5	Jogos na Educação, como se expressam na prática. . . . .	p. 14
<b>3</b>	<b>Jogos Eletrônicos Educativos, critérios de desenvolvimento.</b>	p. 18
3.1	Critérios de Desenvolvimento de Jogos Eletrônicos . . . . .	p. 18
3.1.1	Interface dos Jogos de Computador . . . . .	p. 19
3.1.2	Imersividade: elemento chave para um jogo . . . . .	p. 22
3.1.3	Inteligência Artificial . . . . .	p. 23
3.1.3.1	Personagens Não Jogadores - PNJ (NPC - non players characters) . . . . .	p. 23



3.1.3.2	Adversário artificial . . . . .	p. 24
3.1.3.3	Interface . . . . .	p. 24
3.2	Software educacional . . . . .	p. 25
3.2.1	Base Pedagógica de um Software Educativo . . . . .	p. 25
3.2.1.1	Construcionismo . . . . .	p. 26
3.2.2	Jogos, um tipo de Software Educacional . . . . .	p. 27
<b>4</b>	<b>Desenvolvimento de Jogos 3D</b>	p. 30
4.1	Etapas do processo de elaboração de um jogo . . . . .	p. 31
4.1.1	Design Bible . . . . .	p. 31
4.1.2	Produção de Áudio e Imagens 2D . . . . .	p. 32
4.1.3	Modelagem 3D . . . . .	p. 33
4.2	Motor de Jogos (Game Engine) . . . . .	p. 35
4.2.1	Engine de Renderização . . . . .	p. 36
4.2.2	Engine de Física . . . . .	p. 38
4.2.3	Engine de som . . . . .	p. 39
4.2.4	A Escolha do Engine . . . . .	p. 39
4.3	A Ferramenta jMonkeyEngine . . . . .	p. 40
4.3.1	Características . . . . .	p. 42
4.3.2	Criando a primeira aplicação . . . . .	p. 43
<b>5</b>	<b>Desenvolvimento</b>	p. 46
5.1	Game Design . . . . .	p. 46
5.1.1	Visão geral do Jogo . . . . .	p. 46
5.1.1.1	O Jogo e as Leis de Trânsito . . . . .	p. 46
5.2	Ferramentas Utilizadas . . . . .	p. 48
5.3	Definição dos Requisitos . . . . .	p. 48
5.3.1	Requisitos Funcionais . . . . .	p. 48

5.3.2	Requisitos não funcionais . . . . .	p. 49
5.4	Definição dos Diagramas . . . . .	p. 49
5.4.1	Casos de uso . . . . .	p. 49
5.4.2	Diagrama de Classes . . . . .	p. 50
5.4.3	Diagrama de Sequência . . . . .	p. 53
5.4.4	Diagrama de Estado . . . . .	p. 54
5.5	Arquitetura . . . . .	p. 55
5.5.1	Restrições . . . . .	p. 55
5.6	Metodologia . . . . .	p. 56
5.7	Implementação . . . . .	p. 56
<b>6</b>	<b>Conclusão</b>	p. 61
	<b>Referências</b>	p. 63
	<b>Apêndices</b>	p. 67
	<b>Apêndice A – Código Fonte</b>	p. 68
A.1	Classe Principal “InGameState” . . . . .	p. 68
	<b>Apêndice B – Informações utilizadas no jogo</b>	p. 78
B.1	Informações sobre multas de trânsito . . . . .	p. 78

# *Lista de Figuras*

1	Jogo America's Amy. . . . .	p. 12
2	Áreas mais utilizadas para HUD's . . . . .	p. 21
3	Interface do GIMP. . . . .	p. 33
4	Um mesmo modelo com quantidades de polígonos diferentes. . . . .	p. 34
5	Algumas screenshots do Blender3D. . . . .	p. 35
6	Etapas mais importantes do pipeline gráfico. . . . .	p. 37
7	Exemplos de jogos comerciais construídos em jME . . . . .	p. 41
8	Arquitetura da engine jMonkeyEngine. . . . .	p. 41
9	Exemplo da estrutura de um SceneGraph. . . . .	p. 44
10	Representação do processo de repetição (loop) de um jogo em jME. . .	p. 45
11	Modelo tridimensional da cidade. . . . .	p. 47
12	Modelo tridimensional da moeda e da caixa. . . . .	p. 47
13	Casos de Uso . . . . .	p. 50
14	Diagrama de Classes . . . . .	p. 51
15	Complemento do Diagrama de Classes . . . . .	p. 52
16	Diagrama de Sequência: Jogar . . . . .	p. 53
17	Complemento Diagrama de Sequência : Jogar . . . . .	p. 54
18	Diagrama de Estado: Jogar . . . . .	p. 54
19	Arquitetura Interna . . . . .	p. 55
20	Menu Principal do Jogo . . . . .	p. 57
21	Jogando o Jogo . . . . .	p. 57
22	Infrações que podem ser cometidas no jogo. . . . .	p. 58

23	Mapa da Cidade 3D . . . . .	p. 59
24	Mostrando as informações sobre multas. . . . .	p. 60

# 1 *Introdução*

## 1.1 Contexto

Conforme afirmativa de Alves (2007), os jogos eletrônicos não são uma novidade passageira mas estão presentes na vida de grande número de pessoas e já fazem parte mesmo da cultura humana. Ainda há aqueles que os entendem como brincadeiras infantis, mas o fato que chama a atenção é o crescimento estatístico que a indústria desses jogos vem tendo. De acordo com os dados da empresa de consultoria de mercado NPD <sup>1</sup>, a venda de jogos de video game teve um crescimento de 11% no ano de 2008, totalizando 409.9 milhões de unidades contra 367.7 milhões de unidades vendidas em 2007. Para Ramalho e Corruble (2007) a indústria dos jogos tornam-se cada vez mais importantes por razões, culturais, educativas, estratégicas, dentre outros.

Alguns pais e educadores devem estar atentos para o futuro próximo, por serem os maiores interessados e talvez os que oferecem ainda alguma resistência à entrada de novas tecnologias na educação, sobre o movimento de mais uma nova inserção de instrumento de ensino. Da mesma maneira que o cinema e a televisão em um passado próximo demoraram a ser assimilados, o fenômeno agora se repete em crescimento notável em relação a novas didáticas, porém, desta vez, com os computadores e com os videogames (TAVARES, 2007b).

Como se pode observar na literatura universal, os jogos estão rodeados de fatores importantes o qual Alves (2007) descreve o jogo como um "elemento da cultura que contribui para o desenvolvimento social, cognitivo e afetivo dos sujeitos". Com características próprias eles instigam, envolvem e divertem, imprimindo uma nova maneira de comunicação que permite a atribuição de novos significados a diferentes conceitos (ALVES, 2007). Recentemente, os jogos encontraram o terreno fértil das novas tecnologias, constituindo assim o predomínio dos jogos digitais que permitem, pela utilização de plataformas computacionais, que um ou mais jogadores se enfrentem em situações de desafios porém sem riscos (CRAWFORD, 2003 apud RAMALHO; CORRUBLE, 2007).

---

<sup>1</sup> Disponível em [http://www.npd.com/press/releases/press\\_090202.html](http://www.npd.com/press/releases/press_090202.html) Acesso em 3 de dezembro de 2009.

Em relação a educação, uma visão interessante sobre a condição humana e o aprendizado é apresentada pela revista *Trainning Magazine*. De acordo com pesquisas publicadas em outubro de 2002, o ser humano consegue se lembrar de 15% do que escuta, 25% do vê e mais de 60% do que interage. Notando-se assim a importância da interação no processo de aprendizagem onde o aprendiz, como ser ativo na interação com os demais e com o próprio mundo, torna-se o responsável pelas escolhas, decisão, compreensão e pela direção e significado do aprendido (AZEVEDO et al., 2005).

Apesar de alguns autores, os quais compartilho da ideia, criticarem que os jogos eletrônicos causam alienação e levam ao vício, tais jogos em razão das possibilidades de interação e recepção, vêm sendo aplicados como instrumentos factíveis de melhorar a aprendizagem. Nesse sentido, empresas e instituições diversas têm investido nos denominados *Serious Games* (jogos de aprendizagem), onde segundo Tavares (2007a) o entretenimento é utilizado com uma finalidade de passar conhecimentos, informações, valores e atitudes.

Abordar educação e contemporaneidade é sempre um desafio. Talvez um desses desafios se encontre em questões que envolvem o que é sério do que é apenas diversão ou entretenimento, conforme mostrado por Bittencourt e Giraffa (2003), “a sociedade atual ainda está muito presa aos valores e processos da era industrial, quando se defendia que trabalho e diversão eram campos distintos” .

Tendo como base a importância dos jogos para a pessoa e para a sociedade, este trabalho tem por objetivo buscar uma nova visão sobre os jogos, como mecanismo de aprendizagem, pois, através dos mesmos é possível contemplar a possibilidade de uma forma lúdica, prazerosa, autônoma, além de interativa e criativa, de comunicar, agir, sentir, interagir, socializar e num processo colaborativo, contribuir para a troca e a construção de novos saberes, “saberes de experiência feita” (MOITA, 2006).

## 1.2 Objetivo

Jogos Eletrônicos voltados à educação é um assunto que ganha cada vez mais espaço na sociedade, nesse intuito este trabalho preocupa-se em entender este fenômeno, e como ele pode influenciar as atuais gerações. De forma prática, este trabalho será destinado a criação de um jogo eletrônico educacional voltado principalmente a jovens, o qual tem como tema a abordagem educativa das Leis de Trânsito.

## 1.3 Estrutura do Trabalho

Este trabalho está dividido em cinco capítulos. No capítulo dois são apresentadas as características dos jogos que podem ser estendidas aos jogos eletrônicos, alguns conceitos sobre jogos eletrônicos, os diferentes tipos de jogos eletrônicos, particularidades dos jogos voltados para a educação e como estes estão sendo aplicados. No capítulo três são discutidos os critérios de desenvolvimento de jogos eletrônicos educativos. No capítulo quatro é mostrado como desenvolver um jogo eletrônico e quais ferramentas utilizar. No capítulo cinco é apresentado o desenvolvimento de um jogo voltado a questão do trânsito.

## ***2 Jogos Eletrônicos: possibilidades de desenvolvimento e aprendizagem***

O desenvolvimento desse trabalho com foco nos jogos eletrônicos é justificado por várias razões, especialmente pelas características inerentes ao uso da informática. A abordagem sobre jogos eletrônicos torna-se relevante exatamente pelas possibilidades serem potencializadas pela utilização dos recursos da informática.

Nesse enfoque e ainda com o que diz respeito a educação, Tarouco et al. (2004) destaca que “os games podem se tornar ferramentas instrucionais eficientes, pois eles divertem e motivam, facilitando o aprendizado, pois aumenta a capacidade de retenção do que foi ensinado”.

Assim, o capítulo é apresentado com destaque nos jogos eletrônicos como potenciais objetos de auxílio a aprendizagem, onde são identificados as características dos jogos que podem ser estendidas aos jogos eletrônicos, alguns conceitos sobre jogos eletrônicos, os diferentes tipos de jogos eletrônicos, particularidades dos jogos voltados para a educação e como estes estão sendo aplicados.

### **2.1 Todos os jogos são sérios**

Normalmente as pessoas já tem uma idéia sobre o que são os jogos e que estão muitas vezes associada ao conceito de vitória, competição, divertimento, dentre outros. No entanto, muitas outras características podem ser identificadas, como alguns teóricos mostram, as quais estão ligadas a assuntos relacionados às virtudes inerentes ao ser humano, o desenvolvimento de valores humanos, especialmente na fase infantil e a conceitos que se relacionam ao aprendizado.



Como se pode observar na literatura, os jogos são tão antigos quanto o próprio homem e são atividades que por diversas razões já pesquisadas, remetem a questões psicológicas e sociais (RAMALHO; CORRUBLE, 2007). Apresenta Huizinga (1996) em seu trabalho que uma interessante característica dos jogos é destacada, onde “mesmo em suas definições menos rigorosas, pressupõe sempre a sociedade humana”, o que se amplia a sua essência na medida em que os próprios animais já realizam atividades lúdicas (ALVES, 2007).

Para Huizinga (1996) todo jogo tem um significado, tendo nele algum que transcende as necessidades imediatas da vida e que confere sentido à ação. Assim o autor identifica e apresenta seis propriedades :

1. Atividade voluntária, isto é, livre, ninguém é obrigado a jogar, deve ser uma escolha do jogador.
2. Jogo não é vida real e a criança, o adolescente e o adulto quando jogam sabem disso.
3. A imersão no jogo leva o jogador a uma completa atenção, impulsionada pela imprevisibilidade, incerteza e o acaso.
4. O jogo se dá em certos limites de tempo e espaço, portanto, tem um início e um fim assim como, uma fronteira espacial deste com a vida.
5. Através de regras, é baseado na ordem, o que talvez por isso o jogo esteja ligado à estética, ao que é belo, referentes a duas qualidades muito nobres: o ritmo e a harmonia.
6. Atividade social, ou seja, criam-se grupos sociais de jogadores, no qual estão envolvidos atividades como socialização e organização.

Quando o jogador entra em um jogo, deverá ser capaz de seguir regras para conquistar objetivos, aspecto de grande importância pelo fato da correspondência com o que acontece no mundo e na vida real, constituindo o desenvolvimento de novas competências que certamente serão úteis na vida escolar do aluno e outros que o acompanharão na vida adulta.

Dentre as competências e habilidades, estão (MENEZES et al., 2003): o planejamento de uma ação; a seleção de informações seguindo critérios estabelecidos; a percepção e organização de elementos para atingir objetivos definidos; o desenvolvimento de aptidão intelectual para relacionar e interpretar dados e informações em diferentes formas e linguagens; a capacidade de tomar decisões com rapidez diante de um conjunto limitado de

dados e o enfrentamento de situações problema, com competência para socializar situações e agir de forma cooperativa com o parceiro do jogo.

Naturalmente motivador e envolvente, o jogo possui características que quando se fala de educação são muito interessantes, o que induz a gerar a análise e até mesmo um novo conceito na opinião de muitos pais e professores que argumentam que quando as crianças e adolescentes jogam, estão fazendo algo que está relacionado apenas à diversão. Para muitos, há apenas lazer, o que é visto como perda de tempo. Nesse aspecto, o fato de que brincar preenche as necessidades essenciais especialmente das crianças é constantemente esquecido (ALVES, 2005).

Essas reflexões permitem esclarecer alguns dos assuntos envolvidos nos jogos. Segundo Alves (2005) no que se refere aos teóricos e especialistas, existe uma unanimidade em torno das contribuições cognitivas <sup>1</sup> e sociais, Piaget (1978), afetivas Freud (1976) e culturais Huizinga (1996), potencializadas pelos diferentes jogos.

Vygotsky (apud ALVES, 2005) afirma que os jogos atuam como elementos mediadores entre o conhecimento já construído, aquilo que o sujeito já é capaz de fazer sozinho sem a ajuda do outro e as suas possibilidades de ampliar o seu desenvolvimento e aprendizagem.

Lèvy (apud ALVES, 2005) ainda destaca que o jogo é capaz de promover a construção ou reorganização de funções cognitivas, como a memória, a atenção, a criatividade, a imaginação, e contribuir para determinar o modo de percepção e inteligência pelo qual o sujeito conhece o objeto.

Por fim, Alves (2005) destaca a importância do papel da repetição no jogo, sendo ele um fundamento do brincar. Segundo ela o prazer vivenciado pelos jogadores é atingido na medida em que revivem a ação lúdica anterior, resignificando e elaborando sentimentos, emoções, através da imitação e/ou criação da vida cotidiana, o que para o aprendiz está ligada a memorização e a reconstrução de novas idéias.

Todas essas considerações remetem ao que se refere às possibilidades do saber, do conhecer, que através dos jogos eletrônicos permite às crianças, adolescentes e adultos a descobrir novas formas de aprender, o que hoje ocorrem por meio da simulação de novos mundos. Professores e empresas começam a ficar atentos para o surgimento desses novos caminhos que emergem cotidianamente na vida das pessoas como atores do processo de produção e construção de conhecimento (ALVES, 2005).

Sem dúvida os jogos conseguem imprimir um grande fascínio nas pessoas. Atentas a

---

<sup>1</sup>Segundo o dicionário aurélio 2007, a palavra cognitivo é relativo a cognição que significa ato ou ação de conhecer; aquisição de um conhecimento; percepção; atenção

esta qualidade, as empresas de entretenimento preocupados, sobretudo, com o mercado, investem em diferentes artefatos, principalmente nos jogos eletrônicos, que possam seduzir as crianças, os adolescentes e até os adultos para o prazer de jogar. No entanto, com o passar do tempo essas ferramentas amplamente usadas para a diversão mudam de foco e passam a receber uma aplicação mais séria, como por exemplo, aplicados a treinamentos ou ao aprendizado de assuntos específicos.

## 2.2 Jogos Eletrônicos, a arte eletrônica

Para qualquer jogador de videogame seria impossível a idéia de viver num mundo sem arte, esta como forma de estimular o pensamento, a criatividade, a motivação e o envolvimento em uma atividade prazerosa. No entanto a produção artística do jogo ainda é muito escassamente fundamentada quando comparada as vastas bibliografias sobre cinema, por exemplo. (BOBANY, 2008)

Por outro lado, videogames, tal como se apresentam hoje, possivelmente se constituem na manifestação artística mais dominante do planeta. A indústria do videogame cresce em uma escala vertiginosa ano após ano, estando cada vez mais na vida das pessoas (BOBANY, 2008). Mas é preciso mais do que uma indústria crescente para podermos considerá-los como uma forma de arte.

A arte pode ser considerada como uma forma de expressão, que através de impulsos criativos possibilita construir o ambiente da vida sob formas que reflitam uma concepção positiva do mundo. Sob essa análise os jogos podem se encaixar no seletor universo das artes (BOBANY, 2008). Um exemplo de como isso aconteceria seria encontrado na série “Fallout”, que retrata um mundo após um holocausto nuclear, onde encontramos fortes referências visuais aos anos 1950, através de um jogo interessante e divertido que possibilita ao jogador explorar as consequências dramáticas desta catástrofe. (BOBANY, 2008)

Os videogames, como destaca Turkle (apud ALVES, 2005), também podem ser vistos como uma janela para um novo tipo de intimidade com máquinas, mais especificamente com os computadores. Essa cultura é particularizada por formas de pensamento não lineares que envolvem negociações, oferecem coisas diferentes a pessoas diferentes e conduzem os adultos criados em uma outra lógica a percorrer esses novos caminhos (ALVES, 2005).

A relação com os jogos se dá por meio da interatividade, que nos permite sonhar, ser e fazer algo virtual, que, no campo das idéias, torna-se real. Um personagem não necessita

ter a forma de um ser humano, ele pode ser um animal, uma planta, um alienígena, uma máquina, ou outra espécie qualquer. E esse é um tipo de experiência que se dá não de forma individual, mas coletiva, pois cada vez mais o ambiente da vida social encontra desenvolvimento no mundo virtual da internet e dos jogos (BOBANY, 2008).

A criação de um jogo atraente lida com o que é estético, relacionado ao desenvolvimento de recursos gráficos atraentes, principalmente quando se fala de arquitetar o ambiente do jogo que também deve atender às expectativas de um público cada vez mais exigente por jogos que desafiam a inteligência. Apoiado pela tecnologia a diversidade de interesses, objetivos, saberes e estratégias que podem ser utilizadas em seus processos de desenvolvimento é muito grande, como também é grande as formas como estas informações podem ser passadas ao jogador (BOBANY, 2008).

Os jogos se aliam à interatividade e à virtualidade para oferecer novas fronteiras a serem exploradas, comportando roteiros e imagens com a versatilidade e o potencial do cinema, da pintura e da literatura. Há além da vantagem de serem interativos e reativos ao jogador, penetrando em seu lar, onde livre de críticas sociais ele pode ser o mocinho, um tirano ou um bandido, existe principalmente a busca por evocar impulsos, sentimentos e filosofias, tendo nestes a força e a capacidade de estimular e modificar o ser humano (BOBANY, 2008).

## 2.3 Tipo de Jogo

Os jogos podem ser classificados a partir de diferentes critérios ou abordagens, dependendo de cada estilo.

Para Caillois (1990) as categorias de jogo são estabelecidas a partir das sensações e experiências que proporcionam: Agôn, Alea, Mimicry e Ilinx. Agôn (quando o jogo tem caráter predominante competitivo, como exemplo o xadrex ou o futebol), Alea: quando o que predomina é a sorte como o poker ou dados, Ilynx: vertigem como o paraquedismo e, finalmente, Minicry: simulação como jogos de RPG (*Role Playing Game*). Caillois (1990) ainda polariza esses quatro tipos em dois extremos antagônicos: Paidia, manifestação espontânea do instinto do jogo, onde reina um princípio de diversão, turbulência, fantasia e improviso e Ludus, onde essa aparente indisciplina alegre é contida por uma necessidade primordial de submeter-se a regras convencionais e até incômodas, exigindo um número crescente de tentativas, de persistência, de habilidade ou de artifício.

Essas categorias estão presentes, de uma forma ou de outra, nos jogos eletrônicos, pois

eles apresentam tanto o desejo de vitória, como a ação do acaso, a representação ilusória de um mundo paralelo e a sensação de tensão causadora da vertigem e do êxtase, que transportam o indivíduo para outra esfera.

Sobre os jogos eletrônicos o tipo de um jogo define algumas de suas características de interface e de motor, com implicações claras nas técnicas de implementação. . Battaiola (2000) apresenta as principais descrições dos tipos de jogos, como podemos observar:

**Estratégia:** São jogos idealizados com o objetivo do usuário tomar decisões de grandes conseqüências que possibilitam um desafio intelectual ao invés de apenas reflexo, o qual o jogador deve ter análise crítica da situação para conquistar objetivos. Como exemplo o *SimEarth* e *StarCraft*.

**Simuladores:** São, normalmente, jogos de âmbito tático, com uma visão em primeira pessoa. São jogos que, salvo os de ficção científica, buscam levar uma consideração física do ambiente, sendo seu principal objetivo, a imersão do usuário em um ambiente previamente proposto. O jogador vê o que é possível para seu personagem ver. Enquadra-se neste tipo, os simuladores de carros, de avião, os “*3D Shooters*” e qualquer outro simulador que tente modelar o mundo real e ponha o jogador em perspectiva de primeira pessoa.

**Aventura:** São jogos que combinam ações baseadas em raciocínio e reflexo. O objetivo central é uma resolução de problemas, enigmas e quebra-cabeças para se chegar ao final do jogo. Podemos citar como exemplos: Onde está *Carmem SanDiego?*, *Monkey Island* e *The Day of The Tentacle* .

**Infantil:** São jogos que visam, como público alvo, as crianças e enfocam quebra cabeças ou histórias simples, com o objetivo de divertir ou educar. São jogos caracterizados por imagens coloridas e atrativas, tendendo uma proximidade aos desenhos animados. O jogador atua, geralmente, em terceira pessoa, auxiliando um personagem principal. Como exemplo, podemos citar dois jogos nacionais, *Gustavinho* e *O Mestre*.

**Passatempo:** São programas simples, com quebra cabeças rápidos e sem nenhuma história relacionada, cujo objetivo essencial é atingir uma pontuação alta ou terminar o jogo em menos tempo possível. Geralmente, são representações computacionais de jogos (de tabuleiro ou cartas) já consagrados. Como exemplos, podemos citar o xadrez, damas, paciência, poker, tetris.

**RPG:** *Role Playing Game* É uma implementação dos tradicionais RPG's. Assim como nos jogos de Estratégia, o objetivo do jogo é a realização de uma tarefa através de uma análise crítica da situação e que possibilite um desafio mais intelectual do que com o reflexo. Com a própria notação do RPG podemos diferenciar esses dois tipos. No RPG, o jogador assume papel de um (ou mais) personagem, enquanto a máquina movimenta todo o universo que o ronda. A máquina faria o papel de “Mestre”. Nos jogos de Estratégia, geralmente, o jogador faz esse papel. Ele comanda um grande número de pessoas, cidades, mundos e circunstâncias e a máquina de um possível adversário.

**Esporte:** São programas que simulam jogos populares, como futebol, vôlei, basquete, tênis e etc. O jogador pode comandar times inteiros ou um atleta isolado.

**Luta:** Jogos que simulam a agressão física entre dois ou mais personagens. Muitos dos jogos classificados aqui também poderiam ser enquadrados no tipo Esporte (boxe e outros), mas esse tipo envolve ainda jogos com o objetivo não meramente competitivo, como jogos onde o personagem é o mocinho que enfrenta uma multidão de bandidos para salvar sua namorada. Coordenação motora e reflexos rápidos são valorizados nesse tipo de jogo.

**Educação/Treinamento:** Jogos educacionais, geralmente evocam características de um dos tipos anteriores. Por exemplo, um simulador de vôo pode ser utilizado para treinamento de pilotos de avião ou ainda um jogo de aventura onde o personagem deve decifrar um enigma relacionado às pirâmides, pode ser utilizado para o ensino de história antiga. A diferença é que esse tipo de jogo leva em conta, na sua confecção, critérios pedagógicos associados aos conceitos que procura difundir.

Envolvendo ainda jogos com características semelhantes aos de Educação/Treinamento, os chamados Jogos Sérios (*Serious Game*) são, de acordo com Julian et al. (2007) classificados em 5 categorias.

**Edutainment:** Mistura das palavras inglesas *Educational* = educacional e *Entertainment* = entretenimento. O objetivo do *edutainment* é a de transmitir conhecimento, ou treinamento de uma forma lúdica. Alguns jogos criados para facilitar a interação de crianças com museus são considerados “*edutainment*” onde crianças jogam em um atrativo ambiente, e aprendem durante esse processo. Um exemplo é o *Strong National Museum of Play*<sup>2</sup>.

---

<sup>2</sup> Disponível em <http://www.museumofplay.org/> Acesso em 3 de outubro de 2009.

**Advergaming:** Fusão das palavras inglesas *Advertise* = propaganda e videogame = jogo eletrônico. São jogos usados para anunciar um produto, organização ou ponto de vista. Eles têm sido aplicados a vários jogos online grátis, comissionados por grandes empresas. Nos Estados Unidos o MMORPG *EverQuest II*, é um exemplo onde um anúncio para a Empresa *on-line Pizza Hut Company* aparece no jogo.

**Edumarket Games:** Este reúne aplicações com uma proposta educativa, ou pelo menos aplicações destinadas a tornar os seus utilizadores sensíveis a uma mensagem educativa através dos jogos. Um exemplo é o jogo *Food Force*<sup>3</sup> que combina notícia, persuasão e aspectos do Edutainment.

**Jogos Políticos:** O objetivo deste *Serious Game* é o de denunciar de modo direto algum tipo de problema. Para isto buscam-se mobilizar, de um modo divertido, mecanismos lúdicos do vídeo-game dentro de uma situação politicamente engajada. Esta diversão pode ser feito de dois modos: modificando as regras do jogo, como por exemplo, substituir estratégias de destruição e conquista por desenvolvimentos sociais; e transformando os gráficos e sons do jogo de forma a permitir aos jogadores essa modificação. Um exemplo é modificar as paredes do *Counter Strike* por pichações pacifistas.

**Treinamento e Jogos de Simulação:** a proposta aqui não é vencer, mas simplesmente ter diversão ou chegar a algum objetivo criado pelo usuário. Exemplos destes tipos de jogos são *Sim city*, *The Sims* e *Flight Simulator*.

## 2.4 Jogos Sérios

Para a maioria das pessoas o termo “jogos sérios” soa como palavras de sentido contraditório. As duas palavras parecem mutuamente exclusivas. Uma questão que surge é, exatamente a respeito de que um jogo possa ser ao mesmo tempo algo sério.

Para compreender o que são os jogos sérios, uma simples explicação que alguns profissionais usam, é que “Jogos Sérios são jogos no qual educação (de várias formas) é o alvo principal, em vez do entretenimento” (MICHAEL; CHEN, 2006).

Esta definição ajuda a esclarecer o sentido contraditório, mas por outro lado levanta certa desconfiança de pessoas (desenvolvedores de jogos e educadores) que veem entretenimento e educação, como duas coisas que se contradizem. Em Michael e Chen (2006) vemos que “educação e entretenimento não estão em conflito, mas existem lugares onde

---

<sup>3</sup> Disponível em <http://food-force.educacional.com.br/> Acesso em 8 de novembro de 2009.

os dois se sobrepõem e onde cada lado pode usar as ferramentas dos outros para alcançar seus objetivos”.

Dessa maneira serão mostradas algumas considerações sobre os jogos sérios em relação à importância destes, para atividades de educação e aprendizado.

### 2.4.1 Definindo Jogos Sérios

Com a idéia de jogo já introduzida e tendo em mente que a maioria dos jogos são apresentados aos potenciais jogadores com uma divertida, agradável e prazerosa maneira de passar o tempo ou interagir com outros jogadores, pode-se ainda questionar qual seria uma outra proposta para os jogos que vão além disto.

Abt (1987) descreve Jogos Sérios como tendo um “Explícito e cuidadoso planejamento com o propósito educacional”

Jogos podem ser utilizados seriamente ou casualmente. Estamos preocupados com os jogos sérios no sentido que estes jogos tem um explícito e cuidadoso planejamento com proposta educacional e não são destinados a serem jogados principalmente como diversão. Isto não significa que os jogos sérios não são, ou não podem ser divertidos.

A definição mais simples dos jogos sérios, então, são jogos que utilizam do entretenimento e diversão, ou prazer para veicular propósitos, como no presente trabalho apresento a questão de sua utilização no ensino e aprendizagem das Leis de Trânsito como foco principal.

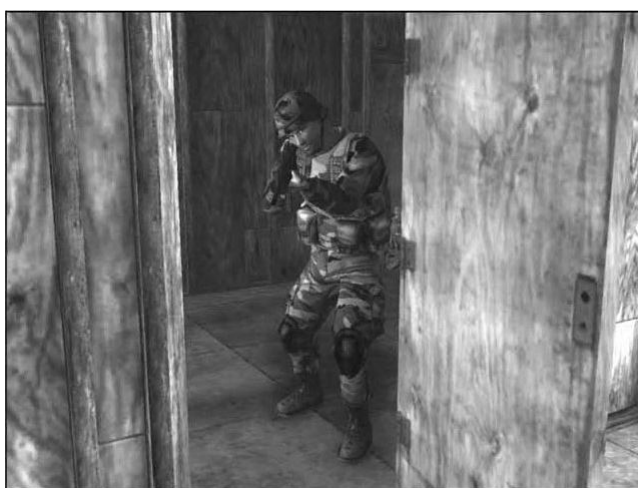


Figura 1: Jogo America's Amy.

*America's Amy* (veja Figura 1) é um exemplo de Jogo Sério. Desenvolvido pelo exército dos Estados Unidos como uma ferramenta de recrutamento, quando jogado por



adolescentes ou outros civis podem ainda ser uma experiência divertida, uma oportunidade de brincar de soldado. Ainda o mesmo jogo com suas qualidades foi reutilizado no Exército como uma forma de treinamento e de ambiente de testes para o ensaio de missões, o treinamento de habilidades, primeiros socorros e treinamento de sobrevivência, dentre outros. Por esta razão, jogos de entretenimento reaplicados para outros propósitos podem também ser considerados jogos sérios.

Esta tarefa de mídia de entretenimento aplicada para outros fins não é limitada aos jogos como mostrado em Michael e Chen (2006). Muitos livros ao longo dos séculos, e filmes do século passado, foram escritos e produzidos com mensagens ou intenções semelhante-mente serias. O romance de John Steinbeck 1942, “*The Moon is Down*”, é um exemplo do que poderia ser chamado de um “livro sério” apresentando a história de uma pequena cidade norueguesa retomada pelos nazistas. Apesar deste motivo oculto, “*Moon is Down*” se apresenta ainda como um grande romance no final das contas. O livro “*Chronicles of Narnia*” de C.S. Lewis são parábolas, que ensinam lições religiosas através de aventuras fictícias de crianças e criaturas mágicas, dentre muitos outros.

Por vezes a arte, por si só, não leva a credibilidade, interpretada apenas como lúdico, sendo que, em sua essência a arte imita a realidade podendo transformá-la e motivar mudança do comportamento.

Algumas pessoas podem argumentar que este tipo de motivo escondido deprecia o jogo, da mesma forma como dizem que tais motivos desvirtuam livros e filmes. Como uma forma de arte, porém, todos os jogos, desde os mais simples, como tetris, aos mais complexos, com histórias interessantes, sons 3D, com um ou múltiplos jogadores, tem algo a dizer. A arte é expressão, não é sempre uma expressão profunda, mas é expressão apesar de tudo (MICHAEL; CHEN, 2006).

Nem todos os jogos sérios abrangem material sério, contudo Bem Sawyer (apud MICHAEL; CHEN, 2006), co-fundador da *Serious Games Initiative*, afirmou que o “sério” em “jogos sérios” destina-se a refletir sobre a proposta do jogo, porque ele foi criado, não tendo nenhuma influência sobre o conteúdo do jogo em si.

Assim, jogos sérios são jogos que utilizam o meio artístico dos jogos para entregar uma mensagem, ensinar uma lição, ou fornecer uma experiência.

## 2.4.2 A importância do divertimento

Divertimento pode ser definido como uma atividade que proporciona satisfação e prazer. Contudo, divertimento não é um ingrediente ou algo que pode ser colocado, mas

sim um resultado, como pode ser percebido quando Raph Koster (apud MICHAEL; CHEN, 2006) define divertimento como um efeito colateral da aprendizagem de algo novo, algo que nós “obtemos”. O sentimento de diversão é um mecanismo essencial de feedback positivo para nos chamar a repetir a atividade.

Um jogo pode ser divertido somente se o jogador gosta de jogar o jogo, dado que jogos são atividades voluntárias. Existe uma implicação do prazer, seja no que se espera de um jogo ou com base na experiência do passado. Na ausência desses dois aspectos o jogador pode optar por jogar ou não ou ainda procurar outra coisa para fazer. Isto é, se um jogador não encontrar um jogo divertido, ele provavelmente não o escolherá novamente para jogá-lo.

Assim, Jogos Sérios precisam ser divertidos ou podem ser divertidos, este é um assunto de grande debate. Em uma pesquisa apresentada em Michael e Chen (2006) entre desenvolvedores, educadores, e pesquisadores, pode-se verificar que mais de 80% das respostas consideram o divertimento como um elemento importante ou muito importante.

Mas nem sempre os Jogos Sérios tem a característica de ser uma atividade voluntária, como proposto por Huizinga (1996). Estagiários podem ser ordenados a jogar um jogo específico como parte de sua formação, o que não significa que o jogo não pode ser divertido. Outro ponto sobre os Jogos Sérios é que, nem sempre o que é divertido ou atrativo em um jogo de simulação para uma pessoa, será também igualmente para outras. Por exemplo, um médico tentando aprender a melhor maneira para abordar um procedimento cirúrgico em uma simulação não estará tentando se divertir. Este estará dentro do objetivo de salvar uma vida, por exemplo, causando o mínimo de danos para o tecido afetado, na busca de uma cura. Neste caso, o divertimento assume uma posição insignificante ou praticamente inexistente para a precisão na prática da simulação (MICHAEL; CHEN, 2006).

Assim também se pode considerar em relação aos jogos educacionais que em dado momento assumem uma relevância muito maior dentro do objetivo que está sendo buscado do que o esparecimento em um momento de diversão.

## **2.5 Jogos na Educação, como se expressam na prática.**

Atrás de um simples jogo eletrônico há expressiva quantidade de informações e conhecimentos, além de valores. Profissionais das mais diversas áreas estão implicados na produção que envolve conhecimentos multidisciplinares tais como a participação de pesquisadores, designers, analistas de computação, além de educadores e psicólogos, para

citar alguns. O conteúdo de um jogo pode ser facilmente compreendido e compartilhado entre os jogadores porque é o resultado de um grande trabalho anterior que gera a possibilidade dos participantes serem ativos e interativos, não simplesmente integrantes neutros ou passivos quando se propõe a jogar (TAVARES, 2007a).

A teoria de Mikhail Bakhtin (apud TAVARES, 2007a) reforça a idéia de que as crianças, assim como os adultos, são indivíduos com bagagem própria, pessoal de experiências que envolvem gostos, preferências, comportamentos, reações além de culturas que são resignificadas em contato com outras experiências, gostos, enfim, com outras realidades, como acontecem com os jogos eletrônicos.

Contudo, como objetos complexos que são, estes só podem ser entendidos através de um pensamento aberto, abrangente e flexível, que leve em consideração não somente os pontos positivos como também pontos negativos.

Lowell Monke professor de Educação da Universidade de Witternberg em um artigo na revista americana Orion questiona o acesso, cada vez mais cedo, de crianças e adolescentes à mídia digital. Segundo ele, há um paradoxo na relação entre a criança e a tecnologia, onde ele ressalta:

Quanto mais poder externo a criança tiver à sua disposição, mais dificuldade ela terá para desenvolver a sua capacidade interna de utilizar esse poder externo com sabedoria. Durante duas décadas em que ensinei crianças a usar as tecnologias digitais, descobri que o poder dos computadores pode levar as crianças à alienação, à insensibilidade e à manipulação das relações sociais no mundo (MONKE, 2005).

É inegável que os jogos podem vir a exercer uma influência negativa sobre seus jogadores (crianças, jovens e adultos), originando condutas indesejáveis para a sociedade, como por exemplo a violência, é uma suposição que não pode ser desconsiderada. Mas exatamente porque trata-se de suposição, deve ser estudada e aprofundada com total seriedade e imparcialidade (TAVARES, 2007a).

Ainda sobre os possíveis efeitos dos jogos sobre os indivíduos Lynn Alves traz em seu trabalho (ALVES, 2005) uma nova forma de se pensar sobre os jogos eletrônicos considerados violentos, que tanto atraem a atenção dos jovens principalmente. A autora entende a violência não como um fruto dos jogos, mas sim de uma condição social desequilibrada. Segundo ela:

A violência vende por favorecer um efeito terapêutico, possibilitando aos sujeitos uma catarse, na medida em que canaliza seus medos, desejos e frustrações para o outro, identificando-se ora com o vencedor ora com o perdedor das batalhas ... penso que a interação com os jogos eletrônicos não produz comportamentos violentos nos jovens. A violência emerge como um sintoma que sinaliza questões afetivas (desestruturação familiar, ausência de limites etc.) e socioeconômicas (queda do poder aquisitivo, desemprego etc.)

De qualquer maneira os estudiosos reafirmam o papel pedagógico dos jogos eletrônicos que já vem sendo utilizado com sucesso nas áreas empresariais e educacionais dentre outras.

### **Jogo e a Fome Mundial**

Um exemplo da utilização de jogos que ensinam vem da ONU. O Programa Alimentar Mundial das Nações Unidas, a maior organização humanitária do mundo, desenvolveu o jogo *Food Force* (2009) que tem o objetivo de conscientizar a sociedade sobre a questão da fome mundial. O jogo, acessível pela web, é inclusive indicado para ser utilizado em sala de aula (TAVARES, 2007a).

### **Jogo na Medicina**

Pam Omidyar é a bióloga que teve a idéia de criar um jogo eletrônico capaz de elevar a auto-estima de crianças cancerosas. Desenvolvido pela ONG HopeLab (2009), localizada na Califórnia, EUA, o jogo *Re-Mission* (2009), considerado o primeiro jogo de ajuda a crianças e jovens vítimas da doença, é protagonizado pela personagem Roxxi, metade robô e metade humana. Roxxi leva o jogador a participar de uma expedição pelo corpo humano de jovens pacientes com diferentes tipos de câncer. O jogo apresenta 20 estágios e em cada um deles, a personagem tem que diagnosticar a doença e iniciar uma verdadeira luta pela vida, através da destruição das células cancerosas, combatendo as infecções além de ter que controlar os efeitos colaterais da doença (TAVARES, 2007a).

Tavares (2007a) cita que cerca de 375 pacientes entre 13 e 29 anos, de 34 centros médicos dos Estados Unidos, Canadá e Austrália, receberam e testaram o jogo. Os resultados foram apresentados na 4ª Conferência Internacional de Medicina do Câncer de Adolescentes e Jovens Adultos, que aconteceu em Londres em 2006.

O resultado foi positivo apresentando melhoras significativas na qualidade de vida

dos pacientes. Os jovens mantiveram boas taxas de sangue após o tratamento de quimioterapia, mostraram uma boa receptividade na utilização de antibióticos, demonstraram saber mais sobre a doença e se dispuseram a participar de terapias. O jogo foi criado com a ajuda de pesquisadores, especialistas na patologia, biólogos, desenvolvedores de jogos eletrônicos, animadores e contou, inclusive, com jovens pacientes (TAVARES, 2007a).

### **Projeto Games-To-Teach**

O projeto *Games To Teach* (2003), desenvolvido em parceria com o *Massachusetts Institute Technology*, tem como objetivo melhorar a visão de como os jogos podem ser usados como suporte ao aprendizado, bem como à investigação das questões atrás do desenvolvimento e comercialização da próxima geração de jogos. É proposto uma série de novos estilos de jogos, como o jogo *Revolution* que aborda a história da Guerra Civil Americana na forma de um MMORPG (*Massive Multiplayer Online Role-Playing Games*)

### **Rede Brasileira de Jogos e Educação**

No Brasil, nestas últimas décadas, os jogos vêm sendo alvo dos estudos de investigações por parte de pesquisadores. Uma manifestação da preocupação com os games foi a formação, em outubro de 2005, da Rede Brasileira de Jogos e Educação (RBJE), cujo objetivo principal é o de se constituir num espaço de intercambio de saberes e interesses de pesquisadores, profissionais e estudantes que desejem ampliar e socializar suas investigações e experiências sobre games (MOITA, 2007).

Esse grupo vem se preocupando também em fomentar a criação, o desenvolvimento e a popularização de jogos que possam mediar os processos de ensinar e aprender, nos diferentes níveis e modalidades de ensino, bem como servir de canal de referência para diversos grupos que estejam interessados na temática (MOITA, 2007).

### ***3 Jogos Eletrônicos Educativos, critérios de desenvolvimento.***

Como descrito no capítulo anterior, os próprios jogos eletrônicos possuem características que são interessantes dentro de um contexto educacional. Mas, para alcançar melhores resultados no processo de ensino-aprendizagem, jogos educativos devem ser construídos a partir de um planejamento criterioso, o qual devido a sua essência se concentram características de duas áreas de estudo, a dos jogos eletrônicos e a dos softwares educativos.

Dessa forma, neste capítulo aspectos importantes relacionados a criação de jogos educativos serão expostos, o qual formarão a base teórica para o desenvolvimento do jogo proposto nesta monografia, tendo em vista que “um jogo bem estruturado e corretamente aplicado proporciona resultados muito ricos, em termos de assimilação ou reformulação de conceitos” (MILITÃO; MILITÃO, 2000).

#### **3.1 Critérios de Desenvolvimento de Jogos Eletrônicos**

A maior parte dos jogos digitais compartilha características com os clássicos tais como xadrez ou gamão: um conjunto de regras que definem o estado do jogo, as ações possíveis do jogador e as condições de vitória. Entretanto, os jogos digitais adicionam diversos elementos, especialmente em torno da noção de interatividade (RAMALHO; CORRUBLE, 2007). As interações são efetivamente muito ricas e visam criar, via elementos gráficos, sonoros e personagens inteligentes, uma experiência imersiva. Nesse sentido (GREENFIELD, 1988) diz que os jogos computadorizados são atraentes para as crianças em consequência ao desenvolvimento destes por preferências pelas imagens visuais dinâmicas e animadas devido à experiência televisiva e ainda reforça a noção da interatividade, pois a pessoa pode agir no jogo.

Na busca de verificar quais as razões que tornam os jogos computadorizados tão atrativos para os jovens e o que eles pensam sobre os jogos educacionais, Clua, Junior e

Nabais (2002) realizaram uma pesquisa com 80 jovens de 10 a 17 anos, de classe média e moradores da cidade do Rio de Janeiro. Entre os jovens, 85% preferem ambientes imersivos com histórias ricas, jogos com qualidade gráfica e com recursos oriundos da aplicação de técnicas de Inteligência Artificial. Destes jovens, 68% consideram os jogos educativos ruins e ninguém considera estes jogos ótimos. Segundo os entrevistados isto se deve ao baixo grau de imersão, baixa qualidade gráfica e muitas vezes a falta de desafios grandes e motivadores. Segundo Bittencourt e Giraffa (2003) evidencia-se assim a necessidade de desenvolver jogos educativos atrativos, que desenvolvam habilidades de forma explícita ou não e não seja uma forma alienante de ensinar.

A partir da necessidade de se construir jogos educativos atraentes, algumas particularidades importantes do desenvolvimento de jogos devem ser analisadas. Assim dentro do contexto de jogos digitais podemos identificar alguns requisitos importantes como descreve Ramalho e Corruble (2007): restrições de tempo e de memória, diretrizes de qualidade de software, gestão do custo e prazo final do projeto. Apesar da importância destes requisitos, o foco deste tópico está notadamente naqueles que se articulam com a noção chave de jogabilidade, ou seja, a maneira pelo qual o jogador interage e controla o jogo e onde o objetivo último é o de seduzir o usuário através de um ambiente gratificante e motivador, para que este queira jogar por horas ou re-jogar o mesmo jogo várias vezes. Complementando, em Battaiola et al. (2002) é acrescentado que ao contrário do produto, as características técnicas dos jogos de computador são pouco conhecidas e que o sucesso de um jogo será determinado pela combinação perfeita entre os três principais fatores: enredo ou trama, motor (software de desenvolvimento de jogos) e interface interativa.

### **3.1.1 Interface dos Jogos de Computador**

Uma boa interface não é apenas aquela bonita e com um som legal quando se interage com ela, mas sim aquela que disponibiliza ao jogador a melhor forma de interação para com o mundo do jogo, logo a interface interativa deve apresentar o estado corrente do jogo e viabiliza a interação entre o jogo e o usuário (ROLLINGS; ADAMS, 2003). Segundo Battaiola et al. (2002) as características relevantes da interface de um jogo são o atrativo visual, a compatibilidade com a trama e o alto nível de jogabilidade, o qual representa a capacidade do usuário de fácil e rapidamente se movimentar pelo ambiente do jogo e acionar os recursos necessários para realizar uma nova jogada.

Também em Battaiola et al. (2002) diz que os critérios para se medir o nível de compatibilização de uma interface a trama de um jogo envolvem a definição dos diversos tipos de jogos e de ambientes que eles usualmente requerem, no entanto, jogos de pas-

satempo como cartas, xadrez, damas e etc ., usualmente requerem interfaces 2D simples e jogos de simulação como corrida de carro ou moto, pilotagem de avião e etc., usualmente requerem realismo, por isto utilizam interfaces baseadas em ambientes 3D e objetos poligonais sofisticados. Battaiola et al. (2002) ainda destaca que a sofisticação visual, o nível de realismo e a flexibilidade de interação baseada em diversas formas de navegação no ambiente, fazem das interfaces 3D as preferidas para o desenvolvimento dos jogos modernos.

Em um estudo feito por Battaiola et al. (2002) através da análise dos jogos mais famosos foi estabelecido os seguintes critérios gerais para a classificação de interfaces 3D: visão do jogador, ambiente e projeção.

A visão do jogador é um dos parâmetros utilizados para se classificar os diferentes gêneros de jogos, o que dependendo da trama ela pode ser em primeira ou terceira pessoa, conforme mostrado por Battaiola et al. (2002):

**Games com visão em primeira pessoa:** são aqueles em que a tela do computador é o olho do usuário, ou seja, o jogador tem a mesma visão do personagem do jogo. Por isso, o jogador quase nunca vê o personagem em si, a não ser quando fica em frente de um espelho durante o jogo. Jogos em primeira pessoa tendem a ser mais imersivos devido a essas características. Exemplos de jogos que utilizam esse tipo de visão são os de combate em ambientes 3D fechados;

**Games com visão em terceira pessoa:** são aqueles em que a visão do jogador não é a mesma da do seu personagem. O jogador vê o game como platéia se enxergando no jogo sendo ele, apenas um soldado, um exército ou um veículo. Muito utilizado em jogos 2D onde personagens e ambientes são limitados. Para interface do jogo é um grande atrativo além de facilitar a interação, pois o usuário tem visão total de sua localização no ambiente do jogo podem ter um maior conhecimento da área de interesse do ambiente.

Dentre os tipos de ambientes dos jogos, Battaiola et al. (2002) os classifica em: internos, externos e siderais, podendo ser criados ambientes mistos formados pelo uso alternado dos tipos básicos.

**Ambientes internos:** são aqueles fechados, constituídos por casas, edifícios, prédios e etc., onde a iluminação é geralmente estática porque provêm de fontes de luz como lâmpadas e tochas. São mais encontrados em jogos em primeira pessoa.



**Ambientes externos:** são amplos espaços criados a partir de um gerador de terreno, ou um mapa de altitude onde geralmente a iluminação é natural e o céu é simulado através de uma caixa mapeada com texturas denominada de SkyBox. Este é muito utilizado em jogos de simulação de carros e aviões.

**Ambientes siderais:** são representados por meio da exibição de estrelas, planetas, nebulosas dentre outros, tendo o sol como fonte principal de iluminação.

Ambientes tridimensionais são construídos a partir de dados 3D que suportam esse tipo de representação, mas que devem ser convertidos em dados 2D que podem ser desenhados na tela do monitor através de um processo chamado projeção. Em jogos, segundo Battaiola et al. (2002), dois tipos de projeções se destacam: a perspectiva e a axonométrica.

**Projeção Perspectiva:** esta é feita de forma que a representação do mundo 3D seja mais adaptada a percepção humana de profundidade, pois um centro de projeção fixado no infinito emite raios projetores que promovem a perda de paralelismo entre linhas. Por exemplo, andando por uma estrada ou uma rua reta o observador verá um estreitamento destas quanto mais longe tentar enxergar. Muito utilizada em jogos de 1ª pessoa.

**Projeção Axonométrica:** ela é gerada por raios projetores os quais são emitidos de um centro de projeção fixado no infinito e que atingem paralelamente ao plano de projeção. Por se perder a noção de profundidade ela é mais utilizadas em jogos 3D onde a sua trama se dá em um plano que é visualizado de forma inclinada e de uma certa altura.



Figura 2: Áreas mais utilizadas para HUD's

Com relação à idéia de jogabilidade de interfaces, atributos importantes são destacados nos próximos subtópicos, mas dentre estes estão os chamados HUD's (Head Up Displays), elementos estáticos muito utilizados para mostrar a saúde do personagem, itens, indicações do progresso do jogo entre outros. A sua disposição na interface pode variar muito de

jogo para jogo, mas as partes mais utilizadas são (ROLLINGS; ADAMS, 2003): a lateral esquerda do cenário, uma tira bem fina na parte superior e a parte inferior da tela, como pode ser visto na Figura 2.

### 3.1.2 Imersividade: elemento chave para um jogo

Um assunto muito discutido quando se fala em ambientes virtuais é sobre a imersividade. O termo é muito usado em realidade virtual e em vídeo games, tendo fundamental importância em destaque aos jogos voltados para fins educativos, pois está relacionado com o grau de envolvimento do usuário com o game. A imersão está ligada a três principais elementos (ROLLINGS; ADAMS, 2003): o visual, o auditivo e o interativo. Mas tratar de imersividade não é nada fácil, pois mesmo considerando que ambientes 3D são potencialmente mais imersivos que ambiente 2D, Battaiola et al. (2002) diz que a sensação de imersão proporcionada por um bom jogo 2D pode ser maior do que a oferecida por um 3D, por permitir um envolvimento mais intenso entre o jogador e o ambiente. Nesse sentido, Battaiola et al. (2002) aborda alguns fatores para a implementação de interfaces.

**Exploração e descoberta:** a interface do jogo deve proporcionar uma navegação intuitiva para que o jogador seja capaz de explorar e descobrir recursos e funções;

**Controles ao Alcance do Usuário:** O jogador tem que ter controle sobre o ambiente do jogo para que ele saiba qual o próximo passo tomar. A sensação de controle do usuário em jogos de 3ª pessoa por exemplo, advém da capacidade dele se ver na cena e determinar como se orientar em relação a outro objetos;

**Guias:** O personagem é orientado por guias, como os mapas presentes em determinados jogos de corrida, o qual conduz a lugares específicos do ambiente.

**Limites do Ambiente:** É fundamental que os jogos de ambientes abertos possuam lugares visitáveis atraentes e os não visitáveis fiquem fora dos limites do mundo;

**Movimentação versus Animação:** Os objetos utilizados na realização de uma ação pelo usuário são os mais importantes. Assim não existe motivo para associar um visual sofisticado e animações complexas em objetos sem nenhuma função para a movimentação do usuário;

**Nível de Complexidade da Reação:** Por motivos psicológicos deve haver um limite de objetos na cena, pois em grande quantidade pode desorientar ou ainda desmotivar o jogador devido ao volume de informações.

**Visão do jogador:** Existem características que diferenciam jogos de 1<sup>a</sup> e 3<sup>a</sup> pessoa.

Em jogos de 1<sup>a</sup> pessoa existe uma facilidade de implementação e um maior nível de assimilação por parte do jogador, enquanto em jogos de 3<sup>a</sup> pessoa a experiência e a interação do jogador é mais complexa.

### 3.1.3 Inteligência Artificial

A interface de jogos tem como papel principal facilitar a maneira de interagir com o jogo, mas a interação propriamente dita inclui outros elementos que visam aumentar a jogabilidade, ou seja, a motivação para continuar jogando e aprofundando-se no jogo, e é nesse sentido que a IA toma importância.

Dentre as diversas características dos diferentes tipos de jogos, indo desde os que exigem mais habilidades motoras, aos que exigem mais raciocínio, na área de IA se concentram importantes desafios para o desenvolvedor. Partindo da idéia de que não se é obrigado a jogar, na relação do usuário com o jogo deve-se atender as demandas mais sofisticadas e desafiadoras, buscando assim seduzir e incentivar o usuário a jogar (RAMALHO; CORRUBLE, 2007).

A exigência de jogos mais inteligentes, a importância e o espaço da IA tem sido crescente. Ela pode intervir em vários níveis ou em vários papéis durante a execução do jogo a fim de torná-lo mais rico e estimulante, propondo níveis de desafios adequados ao jogador, ações coerentes dos personagens entre outros. Segundo Ramalho e Corruble (2007) alguns dos elementos envolvidos no desenvolvimento de jogos que poderiam se beneficiar de técnicas de IA são: os PNJs, o adversário artificial, o mestre do jogo e a interface.

#### 3.1.3.1 Personagens Não Jogadores - PNJ (NPC - non players characters)

PNJ são personagens controlados pela máquina. É no seu comportamento e nas suas decisões onde a IA é mais facilmente notada. Os PNJ podem aparecer nos jogos como elementos do ambiente (aves, transeuntes, etc) que reforçam o realismo, personagens que fornecem informações ao jogador ou que ajudam ativamente e os adversários (RAMALHO; CORRUBLE, 2007).

A capacidade de tomar decisões independentemente do jogador tem na modelagem como “agente inteligente” uma das mais interessantes abordagens para a implementação de um PNJ. Além disso, em muitos jogos os PNJ não devem atuar isoladamente, mas em grupo e de maneira organizada, o que tem solução na modelagem multiagente de seus

comportamentos e decisões.

A autonomia em relação ao jogador é a característica mais importante em PNJ, mas outras características também podem ser desejáveis como a capacidade de se adaptar, de demonstrar uma personalidade e a de promover um diálogo em linguagem natural. Ainda dentre os PNJs a autonomia dita parcial, é aquela onde os personagens tem a capacidade a capacidade de interpretar e adaptar as ordens do jogador de acordo com a circunstâncias do jogo. Existem também os PNJs semi-autônomos que podem de vez em quando tomar iniciativas, evitando que o jogador perca tempo dando todas as ordens aos personagens, como em jogos de esporte do tipo futebol.

### **3.1.3.2 Adversário artificial**

A IA também esta presente nos chamados “adversários artificiais” que são os oponentes contra o qual o jogador humano joga. De certa forma, eles devem estar acima do nível do PNJs, tendo acesso as mesmas ações do jogador humano. Suas decisões por estar a nível do jogador compreendem, por exemplo, a gestão de recursos, a análise do adversário humano (para poder antecipar suas ações e decisões) e dar ordens aos personagens PNJs que compõem a equipe adversária (RAMALHO; CORRUBLE, 2007).

Dentro de uma abordagem multiagente para PNJ e adversário artificial são definidos dois casos: o centralizado e o distribuído. No centralizado (do alto para baixo), é possível ter um único agente (o adversário artificial) que toma todas as decisões, abrangendo as que envolvem cada PNJ. No distribuído (de baixo para o alto) os PNJs tomam decisões de alto nível (RAMALHO; CORRUBLE, 2007).

PNJ e agentes artificiais devem por fim ser implementados a agir de forma autônoma, inteligente a ponto de tomar iniciativa e adaptar seu comportamento e suas decisões em função do desenrolar do jogo e do oponente humano, o que representa um bom nível de desafio, além de evitar trapagens que podem ser detectadas pelo jogador.

### **3.1.3.3 Interface**

A IA pode também ser aplicada no jogo através de sua interface propriamente dita. Seu objetivo seria o de facilitar a utilização do jogo por meio de sistemas de ajuda nos quais um agente (visível ou não) pode, segundo o contexto do jogo e o nível do jogador, dar conselhos, sugestões e fazer avaliações. A IA ainda pode ser aplicada na criação automática de mapas que sejam coerentes geograficamente quando requerido, o que para o jogador repercute no interesse em jogar o jogo mais vezes (RAMALHO; CORRUBLE, 2007).

## 3.2 Software educacional

Uma visão mais ampla de se compreender e discutir jogo educativo é através da ótica do chamado software educacional, onde se concentram importantes conceitos que envolvem naturalmente tanto a área de software como especialmente a área do aprendizado. É importante destacar que o software educacional explore as verdadeiras potencialidades pedagógicas, para que ele não seja utilizado apenas como um programa divertido e agradável.

De acordo com Lacerda (2007) software educacional é qualquer software que possa ser utilizado na educação, muito embora não tenha sido desenvolvido para tal finalidade. Neste sentido, professores e alunos em contato com a informática, podem se beneficiar dela, pois oferece diversos tipos de softwares educacionais e programas aplicados à educação, como os indicados por Seabra (1994): exercitação, programas tutoriais e aplicativos, linguagem, programas de autoria, editores de textos, simulações e é claro os jogos.

Cada vez mais o software educacional ganha importância como uma ferramenta educacional, o que pode ser percebido em uma pesquisa feita por Marisa Lucena (LUCENA, 1992 apud TEIXEIRA; BRANDAO, 2003), que ao analisar fatores determinantes da utilização do computador como ferramenta educacional, em Diretrizes para a Capacitação do Professor na Área de Tecnologia Educacional, conclui-se que “o processo de informatização da sociedade brasileira é irreversível e que se a escola também não se informatizar, correrá o risco de não ser mais compreendida pelas novas gerações”. Isto mostra a necessidade da utilização de meios educacionais que se adaptam a realidade das pessoas, o que hoje é centrada na tecnologia

Todo software educacional tem o seu valor pedagógico, mas nenhum software é, em termos absolutos, um bom software (MEIRA, 1998), por isso mais importante que o software, em si, é o modo como ele é utilizado, o que requer analisar como um software pode ter um uso educacional e como ele pode ajudar o aprendiz a construir o seu conhecimento. Nesse sentido, serão descritos a seguir alguns aspectos importantes do desenvolvimento de softwares educativos centrada nos jogos.

### 3.2.1 Base Pedagógica de um Software Educativo

Segundo (SILVA; MARCHELLI, 2009) um dos critérios de análise de softwares educativos é a identificação da abordagem pedagógica, pois um software para ser educativo deve ser pensado segundo uma teoria sobre como o sujeito aprende, como ele constrói o seu

conhecimento. Dentre as diferentes teorias baseadas em estudos sobre o desenvolvimento cognitivo (construtivista, behaviorista, construcionista, ou outra) será tratada aqui a construcionista.

### 3.2.1.1 Construcionismo

O construcionismo é uma teoria do Seymour Papert um dos principais pensadores sobre as formas pelas quais a tecnologia pode modificar a aprendizagem. Segundo (VALENTE, 1998b) na abordagem construcionista o computador é usado como meio para propiciar a construção do conhecimento pelo aluno, ou seja, o aluno interagindo com o computador na resolução de problemas, tem a chance de construir o seu saber. Ele ainda complementa que esse é o paradigma que enfatiza a aprendizagem ao invés de destacar o ensino; a construção do conhecimento e não a instrução.

Dentro da concepção construcionista, um software para ser educativo deve ser um ambiente interativo que proporcione ao aprendiz investigar, levantar hipóteses, testá-las e refinar suas idéias iniciais, dessa forma o aprendiz estará desenvolvendo o seu próprio aprendizado. Para Valente (1998a), a realização do ciclo descrição - execução - reflexão - depuração - descrição é de extrema importância na aquisição de novos conhecimentos por parte do aprendiz, onde ele propõe:

**Descrição da resolução do problema:** a descrição permite ao aprendiz representar e explicitar o nível de compreensão que possui sobre os diferentes aspectos envolvidos na resolução do problema.

**Execução dessa descrição pelo computador:** o computador executa fielmente a descrição fornecida. Esse “feedback” fiel e imediato para o aprendiz. O resultado obtido é fruto somente do que foi solicitado à máquina.

**Reflexão sobre o que foi produzido pelo computador:** o resultado obtido através do computador permite ao aprendiz refletir sobre o que foi solicitado ao computador, o que pode provocar alterações na sua estrutura mental. Desse processo de reflexão pode surgir a necessidade de uma nova depuração quando o resultado é diferente das idéias iniciais.

**Depuração dos conhecimentos por intermédio da busca de novas informações ou do pensar:** se o resultado não corresponde ao que era esperado, o aprendiz tem que depurar a idéia original através da aquisição de conteúdos ou de estratégias, onde o aprendiz busca informações em outros locais e essa informação é assimilada

pela estrutura mental, passando a ser conhecimento e as utiliza no programa para modificar a descrição anteriormente definida. Nesse momento, repete-se o ciclo descrição - execução - reflexão - depuração - descrição.

Levando em consideração esse ciclo, o software facilita a análise do programa de modo que o aprendiz possa achar seus erros. Para Valente (1998b) o processo de achar e corrigir o erro constitui uma oportunidade única para o aluno aprender sobre um determinado conceito envolvido na solução do problema ou sobre estratégias de resolução de problemas. O aluno pode experimentar alterar o conteúdo ou as estratégias e verificar se essas alterações são mais ou menos efetivas na busca da solução do problema. Mas isso só é possível com um "feedback" fiel, pois se houver problema no funcionamento do programa, esse é produto do pensamento do aprendiz.

### **3.2.2 Jogos, um tipo de Software Educacional**

Cada um dos diferentes softwares usados na educação apresenta características que podem ajudar, de maneira mais clara, o processo de construção do conhecimento. Segundo Valente (1998a) estes podem ser classificados em algumas categorias, de acordo com seus objetivos pedagógicos: tutoriais, programação, aplicativos, exercícios e práticas, multimídia e Internet, simulação e modelagem e jogos.

Geralmente os jogos são desenvolvidos com a finalidade de desafiar e motivar o aprendiz, envolvendo-o em uma competição com a máquina ou com os colegas. Valente (1998a) ressalta que o aprendiz deve jogar o jogo e, com isso, espera-se que ele esteja elaborando hipóteses, usando estratégias e conhecimentos já existentes ou elaborando conhecimentos novos. Assim os jogos podem também ser analisados do ponto de vista do ciclo descrição - execução - reflexão - depuração - descrição, dependendo da ação do aprendiz em descrever suas idéias para o computador.

Dentre os jogos, Falkembach, Geller e Silveira (2006) evidencia no trabalho de Stahl (1991) e Bongiolo, Braga e Silveira (1998) características importantes que devem estar presentes em um jogo educativo digital, das quais se destacam as seguintes:

- os participantes devem entender os objetivos do jogo;
- o jogo deve ser atraente para que haja interesse e entusiasmo;
- os efeitos auditivos e visuais devem ser explorados para atingir o objetivo educacional buscado no jogo;

- explorar a competição;
- o jogador deve se capaz de repetir segmentos e controlar a taxa de avanço do jogo;
- deve oferecer reforço positivo nos momentos adequados;
- desafios devem ser colocados, de forma que um determinado problema possa ser solucionado em diferentes níveis;
- deve estar presente o nível de desempenho durante o jogo, fornecendo resumos do desempenho global ao final;
- utilizar mecanismos para corrigir possíveis erros de quem está jogando para melhorar o desempenho do mesmo;
- oferecer um ambiente rico e complexo para resolução de problemas, através da aplicação de regras lógicas, da experimentação de hipóteses e antecipação de resultados e planejamento de estratégias.

Para Passerino (1998) os jogos educativos computadorizados são criados com a finalidade de entreter e possibilitar a aquisição de conhecimento, devendo tentar explorar o processo completo de ensino-aprendizagem. É reforçado que basicamente bons jogos educativos apresentam algumas das seguintes características:

- trabalhar com representações virtuais de maneira coerente;
- oferecer grandes quantidades de informações que podem ser mostradas de maneiras diversas (imagens, texto, sons, filmes, etc.), de uma maneira objetiva e lógica;
- exigir uma certa coordenação e organização por parte do usuário;
- possibilitar que o usuário veja o resultado de sua ação de maneira imediata facilitando a auto-correção;
- permitir um envolvimento homem-máquina gratificante;
- estimular a repetição de exercícios;
- incentivar a criatividade do usuário, instigando-o a crescer, tentar, sem se preocupar com os erros.



Falkembach, Geller e Silveira (2006) distingue ainda em Bongiolo, Braga e Silveira (1998) que, além das características apresentadas anteriormente, um jogo educativo digital precisa observar alguns requisitos de qualidade didático-pedagógica e também de qualidade técnica, tais como:

**Requisitos de qualidade didático-pedagógica:** objetivos bem definidos, encadeamento lógico do conteúdo, adequação do vocabulário, possibilidade de formação de conceitos, ortografia e gramática corretas, feedback apropriado, clareza e concisão dos textos apresentados, possibilidade de acesso direto a diferentes níveis do programa;

**Requisitos de qualidade técnica:** execução rápida e sem erros, resistência a respostas inadequadas, interface amigável, tempo suficiente de exibição das telas, possibilidade de acesso à ajuda, possibilidade de trabalho interativo, possibilidade de controle do usuário sobre a sequência de execução do jogo, possibilidade de correção das respostas, possibilidade de sair do sistema a qualquer momento e uso de telas com diagramação seguindo um modelo único de organização.

Por fim, Passerino (1998) destaca que para utilização de um jogo computadorizado dentro de um processo de ensino e aprendizagem devem ser considerados não apenas o seu conteúdo senão também a maneira como o jogo o apresenta, relacionada é claro à faixa etária que constituirá o público alvo. E Valente (1998a) alerta que os jogos têm a função de envolver o aprendiz em uma competição e essa competição pode dificultar o processo da aprendizagem uma vez que, enquanto estiver jogando, o interesse do aprendiz está voltado para ganhar o jogo e não em refletir sobre os processos e estratégias envolvidos no mesmo. Sem essa consciência é difícil uma melhora do nível mental.

## 4 *Desenvolvimento de Jogos 3D*

Um jogo 3D não é feito como muitos outros programas. A manipulação de sons, imagens e objetos 3D o tornam softwares especiais que envolve em seu desenvolvimento diferentes áreas da computação como: a computação gráfica, a inteligência artificial, redes, multimídia, etc. Além disso jogos conduzem para a programação de aplicações em tempo real ou de simulações, ao invés de programas que tem uma lógica sequencial. Outro fator de fundamental importância destacado em Clua e Bittencourt (2005) é que o jogo esteja baseado sobre diversas API's<sup>1</sup>, como OpenGL, DirectX, OpenAL, etc.

Segundo (LAMOTHE, 2003) existem poucas áreas da engenharia de software que exigem os limites de hardware, software e do próprio programador tal como a programação de jogos. Existem peças essenciais que devem trabalhar juntas (matemática, física, imagens, sons, estrutura de dados) de forma que o jogo seja divertido e agradável de utilizar, fundamental para que as pessoas se interessem em usá-lo.

Neste capítulo, é apresentado o processo de desenvolvimento de um Jogo 3D. Para tal, aspectos gerais serão descritos, como também é exposto o motor de jogos (game engine) jMonkeyEngine, ferramenta utilizada no desenvolvimento do jogo educativo mostrado nesta monografia. Por motor de jogos entende-se um componente de software projetado para facilitar o desenvolvimento de jogos eletrônicos.

Este capítulo foi inspirado no trabalho de Esteban Walter Gonzalez Clua e João Ricardo Bittencourt, *Desenvolvimento de Jogos 3D: Concepção, Design e Programação*, apresentado no XXV Congresso da Sociedade Brasileira da Computação (CLUA; BITTENCOURT, 2005).

---

<sup>1</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

## 4.1 Etapas do processo de elaboração de um jogo

Softwares tais como os jogos, seja ele 2D ou 3D, requerem que um processo de desenvolvimento seja adotado. Além disso os jogos se enquadram nas aplicações de entretenimento digital, o que significa que aspectos artísticos devem ser tratados nesse processo (CLUA; BITTENCOURT, 2005). Em Rollings e Morris (2004) são descritos as seguintes etapas no desenvolvimento de um jogo 3D:

1. Confecção do Design Bible;
2. Produção de audio e imagens 2D;
3. Modelagem 3D;
4. Escolha ou desenvolvimento do motor de jogos;
5. Integração dos aspectos artísticos com os aspectos computacionais.

Nas próximas subseções serão descritos estas etapas no processo de desenvolvimento de um jogo.

### 4.1.1 Design Bible

Não é possível criar um jogo sem um documento com todos os seus detalhes. Este documento costuma ser chamado de Design Bible. A documentação explica qual o conceito do jogo, como ele pode ser jogado, o que ele tenta alcançar e como isto pode ser atingido. Com esse documento as regras e as características do jogo podem ser melhor implementadas no plano de software (ROLLINGS; MORRIS, 2004). Segundo Clua e Bittencourt (2005) é importante que este documento esteja pronto no início do processo de desenvolvimento, onde devem estar presentes os seguintes elementos:

#### **Roteiro**

Elaborados sobre regras similares a dos roteiros de filmes ou de séries de televisão, este para o jogo documenta o enredo (trama), define os personagens, as suas ações, os cenários e qualquer detalhe adicional importante. Diferentemente dos roteiros de filmes, a característica que deve ser acrescentada ao roteiro de um jogo é a interatividade. Assim, ramificações da trama devem levar em conta as ações do jogador, o que acarreta múltiplas cenas e vários finais para o jogo (BATTAIOLA, 2000). O roteiro deve ser elaborado considerando o estilo do jogo que será desenvolvido.

## Game Design

A conceituação artística do jogo é a chamada Game Design. No início da história dos jogos eletrônicos, esta parte do documento muitas vezes era feita pelo próprio programador, mas hoje, devido a complexidade das histórias e dos cenários, é importante que esta parte do documento seja escrita por um artista. O conceito artístico complementa a escrita do documento que por si só pode não ser suficiente para entender a visão do jogo, pois envolve a descrição dos cenários, esboços de personagens, descrição das texturas fundamentais, mapas e descrições das fases (também denominado de *level design*).

## Game Play

Nesta parte do documento deve ser descrito como será a relação entre jogo e jogador ou seja a jogabilidade. A jogabilidade abrange as regras do jogo e o balanceamento das regras (*game balancing*). A extensão da jogabilidade para os jogos eletrônicos é muito grande, pois é possível recriar virtualmente qualquer situação, como por exemplo em uma partida de futebol que você pode ser o técnico, o goleiro ou o atacante. Assim, nessa descrição deve ficar claro que o jogo proporciona divertimento e desafios interessantes.

Com a ideia do jogo definida na etapa de conceituação, passa-se para a fase da criação artística (imagens e sons) e a de programação. Nesta parte do desenvolvimento de um jogo 3D são necessários: um compilador (Java ou C), um programa de manipulação de imagens 2d, um programa de processamento de som, e algum programa de modelagem 3d.

### 4.1.2 Produção de Áudio e Imagens 2D

Um motor é o responsável principal no funcionamento de um jogo, mas de forma a incrementá-lo outros aspectos tecnológicos devem ser considerados na fase de produção, entre eles o áudio e as imagens 2D. Para a produção de áudio uma ferramenta interessante é o Audacity (AUDACITY, 2009), um software livre que permite a criação de áudio. Suas capacidades incluem alteração da velocidade ou do timbre de uma gravação, cortar, copiar e juntar sons e faixas de áudio nos formatos Ogg Vorbis, MP3 e WAV. Ele é muito útil, mas não possui todos os recursos oferecidos pelo SoundForge, que é uma ferramenta comercial.

Os jogos 3D são criados com objetos tridimensionais que ganham uma aparência interessante quando compostos com imagens bidimensionais. Tais imagens também podem ser usadas como texturas, na elaboração de botões, janelas, barras de energia e outros elementos gráficos.

Para gerar estas imagens existem muitas ferramentas para a manipulação gráfica cujo Adobe Photoshop é um dos softwares mais utilizados para essa atividade. Outra opção que não deixa a desejar é o GIMP (GNU Image Manipulation Program)(GIMP, 2009), uma ferramenta multilplataforma e livre . A figura 3 ilustra a caixa de ferramentas usada para criar e editar as imagens no GIMP.

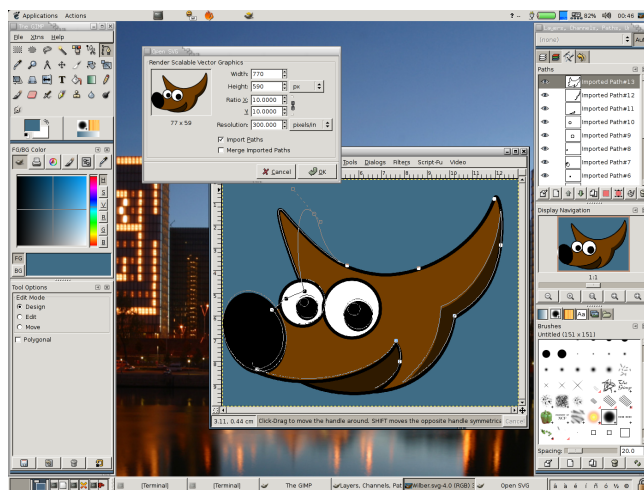


Figura 3: Interface do GIMP.

O GIMP oferece as mesmas funções encontradas no Photoshop, permitindo retocar, compor ou criar imagens. Para isso ele dispõe de uma série de pincéis, permite trabalhar com inúmeras camadas e máscaras, exporta e importa diferentes formatos de imagens (BMP, PNG, JPG, TIF, TGA, PCX, GIF, entre outros), edita textos, disponibiliza uma grande quantidade de filtros ( desfocar, realçar , detecção de bordas, distorções, efeitos de luz, entre outros), permite que novos recursos sejam adicionados (filtros e formatos de arquivos) através de plug-ins construídos em C.

### 4.1.3 Modelagem 3D

Os objetos geométricos do jogo serão desenvolvidos na fase de modelagem. Os softwares de modelagem utilizam do conceito básico de projeções planares (BATTIOLA, 2000), que tomam a tela do computador como o plano de projeção para possibilitar a visualização do objeto 3D em vários pontos de vista. Nessa fase, serão modelados objetos estáticos e dinâmicos (CLUA; BITTENCOURT, 2005), havendo essa diferenciação pelo fato do motor de jogos tratar cada um destes de forma diferente. Como por exemplo, os objetos estáticos podem ser pré-processados para a otimização do processo de renderização.

De acordo com (CLUA; BITTENCOURT, 2005), o software de modelagem deve ter 4 características principais: 1 - a ferramenta deve possibilitar que a modelagem seja feita

por polígonos; 2 - não devem haver dificuldades para a aplicação de texturas sobre os modelos; 3 - devem ser oferecidos recursos que possibilitem a redução da quantidade de polígonos dos objetos; 4 - uma interface que permita visualizar como o objeto ficará dentro do jogo, ainda no processo de modelagem.

Os objetos que constituem um jogo 3D são produzidos a partir de polígonos. O nível de detalhes de cada objeto é proporcional a quantidade de polígonos que o forma, mas cada polígono exige de um processamento. Assim, os modeladores devem ser capazes de desenvolver objetos com uma quantidade de polígonos razoável, para que não se perca a qualidade, mas que possibilite a criação de cenários maiores ou que novos objetos sejam adicionados. Veja figura 4.

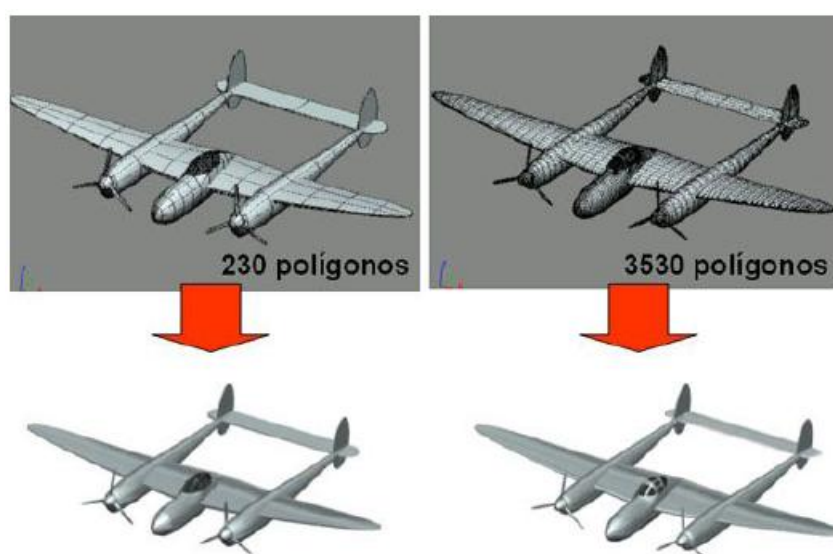


Figura 4: Um mesmo modelo com quantidades de polígonos diferentes.

A figura 4 mostra o mesmo modelo construído com quantidades de polígonos distintos (uma diferença de 15 vezes), mas que no entanto são muito semelhantes.

Uma opção livre e gratuita para a criação de modelos 3D, animações e até jogos profissionais é o Blender3D (BLENDER3D, 2009). Trata-se de uma ferramenta completa que oferece tantas funcionalidades quanto os grandes concorrentes pagos, mas que tem como principal desvantagem a interface, que não é fácil de ser usada. O Blender3D suporta importação e exportação de diversos formatos incluindo 3ds Max (.3ds), Collada(.dae), MilkShape 3D (.ms3d), Wavefront (.obj), entre outros. Ele ainda suporta o Python como linguagem de desenvolvimento de scripts de novos plug-ins aumentando as funcionalidade básicas da ferramenta.

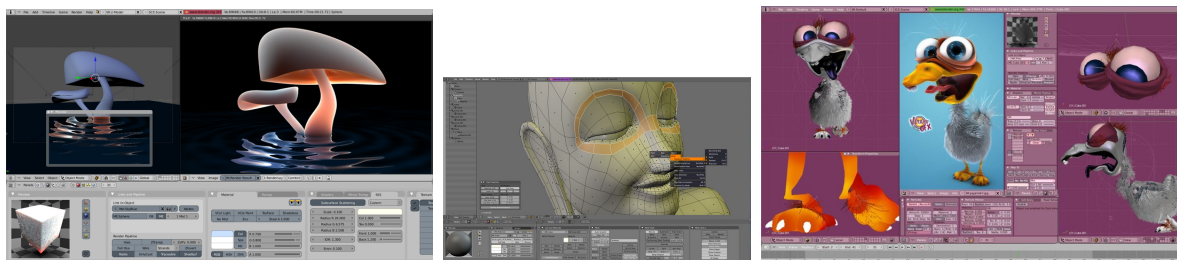


Figura 5: Algumas screenshots do Blender3D.

A figura 5 mostra algumas screenshots que ilustram a capacidade do Blender3D para modelagem e animação 3D.

## 4.2 Motor de Jogos (Game Engine)

O motor do jogo é o responsável por juntar todos os recursos, de forma a dar uma funcionalidade a aplicação. Assim, dentro do processo de desenvolvimento do jogo a primeira e principal tarefa é a criação da historia do jogo e a definição da jogabilidade. O próximo passo é construir o conteúdo do jogo, a parte artística que inclui a modelagem. Finalmente os programadores desenvolvem, através do motor de jogos, uma aplicação para carregar o conteúdo necessário, gerenciar os dados do mundo e renderizar eles sobre a tela do computador, integrando a historia e a jogabilidade.

Dessa forma, assim como o carro não pode andar sem o seu motor, um jogo tem em seu motor o sistema principal capaz de controlar toda a “mecânica” do jogo. Ele dispõe de diversas funcionalidades tais como, a alocação de memória, a aquisição de recursos, o carregamento de dados do disco, o processamento de entradas, entre muitas outras que permitem ao programador concentrar esforços apenas na lógica do jogo. Sua função principal é a de controlar a reação do jogo, efetuando toda logica incluindo a movimentação de personagens, em função de uma ação do usuário na interação com dispositivos de entradas como teclas pressionadas ou mouse arrastado, apresentando em um dispositivo de vídeo a imagem do estado atual do jogo (CLUA; BITTENCOURT, 2005).

É desejável para um motor de jogos algumas características (CLUA; BITTENCOURT, 2005), dentre elas : 1 - permitir que diferentes jogos sejam desenvolvidos sobre o mesmos motor; 2 - possibilitar que códigos de projetos anteriores sejam reutilizados; 3 - ter a manipulação de API's abstraída, facilitando a troca de uma API sem precisar mudar o código do jogo; 4 - permitir uma integração entre código e modelagem 3D sem grandes dificuldades.

Normalmente, um engine é composto por diversas ferramentas, cada uma responsável

por alguma etapa do processo de criação de um jogo. Os componentes mais comuns (ZERBST; DUVEL, 2004 apud CLUA; BITTENCOURT, 2005) de se encontrar em engines são os seguintes:

- **Engine Core:** Camada sobre o qual o jogo será construído, responsável pela manipulação dos dados e renderização da cena;
- **Engine SDK:** Código fonte do engine core. Alguns empresas, se caso não comprado, não disponibilizam o SDK de seu motor de jogos, limitando a quantidade de aplicações que podem ser desenvolvidas.
- **Level Editors:** Componente de auxílio a programação, que permite testar códigos sobre os modelos desenvolvidos em outros softwares. Alguns editores também são capazes de criar modelos 3D.
- **Conversores/Exportadores:** O motor de jogos deve fornecer algum recurso que permita importar modelos construídos em algum software de modelagem 3D. Uma opção é a instalação de plug-ins conversores nos softwares de modelagem.
- **Linguagens Script:** Uma área da codificação de jogos que não é muito explorada é a de script. Script's são utilizados para tratar a lógica do jogo e a Inteligência Artificial dos elementos dinâmicos. A implementação de um script não é feita diretamente sobre o motor, o que preserva todo o código principal do jogo e permite por exemplo, que o próprio jogador escreva extensões para o jogo. As linguagens script mais comuns são: *JavaScript*, *Python* e *Lua*.

Segundo Battaiola (2000) as técnicas envolvidas na implementação de um motor são muito abrangentes e incluem: estruturação e classificação de dados, mecanismos de comunicação, métodos sofisticados de controle dos personagens e do mundo, de detecção de colisão, de sincronização de animações gráficas, imagens, som entre outros.

De acordo com Clua e Bittencourt (2005), um motor que é desenvolvido para ser usado em projetos de video-game contém diversos sub-motores. Exemplos de sub-motores são o de renderização, o de física e o de som.

### 4.2.1 Engine de Renderização

Esse motor cuida do pipeline gráfico, que é o processo de renderização do próximo frame a partir do tratamento das entradas do jogador e da lógica do jogo. Com isso, imagens 2D são geradas dos modelos 3D para serem mostrada na tela do computador.



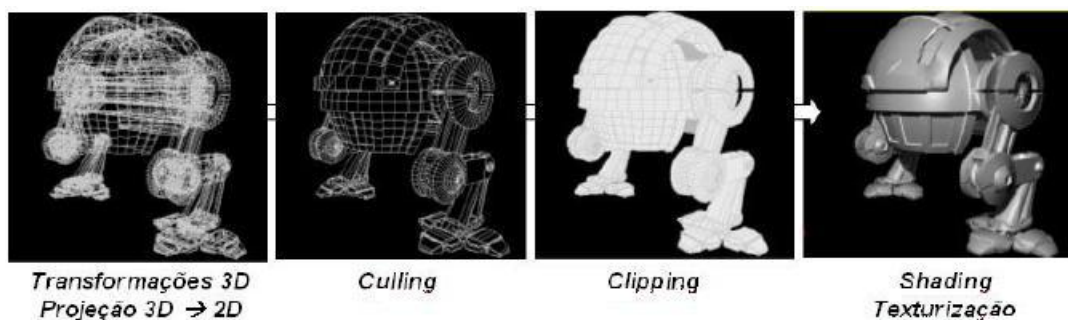


Figura 6: Etapas mais importantes do pipeline gráfico.

A figura 6 ilustra as principais etapas do processo de renderização.

Algumas das etapas mais importantes do pipeline gráfico são (LAMOTHE, 2003):

- **Transformações 3D:** Nesta etapa aplica-se o movimento aos modelos 3D que incluem translação, variação de escala e rotação. As transformações de entidades tridimensionais devem ser pensados segundo um sistema de coordenadas. A cada passo do jogo uma matriz irá acumular o resultado do movimento de um objeto e irá calcular sobre este valor o novo posicionamento do objeto na cena.
- **Projeção 3D → 2D:** Projeções deste tipo levam em consideração o sistema de coordenadas da câmera, os vértices que compõem os objetos 3D e o plano de projeção da câmera( plano de visão). O plano de visão é usado para gerar uma imagem virtual do mundo 3D, que será mostrada na tela do computador. Para renderizar os objetos 3D que estão dentro do campo de visão da câmera, matematicamente os vértices que formam o objeto são projetados para o plano de visão. Todo esse processo é executado após a aplicação das transformações 3D;
- **Culling:** Esta é uma técnica de otimização do processamento gráfico. Ela consiste na remoção de objetos ou de polígonos que não importam para a visualização da cena. Uma forma usada para a remoção de objetos utiliza do conceito de *BoundingBox* onde o objeto é envolvido por uma caixa que o representará. Assim com poucos testes é possível determinar se um dado objeto faz parte do campo de visão da câmera. O culling reduz a quantidade de processamento ao retirar polígonos desnecessários, sendo assim interessante que seja aplicado o quanto antes dentro do pipeline gráfico;
- **Clipping:** Refere-se a prática do computador de desenhar apenas coisas que são visíveis para a tela do computador. Polígonos ao serem projetados podem cair totalmente ou parcialmente dentro do plano de visão. Este problema é solucionado aplicando-se o clipping (recorte), que cria novas arestas e vértices que eliminam

partes desnecessárias do polígono, o que dependendo da forma que é aplicado evita o processamento da rasterização sobre essas partes;

- **Rasterização (iluminação e texturização):** A último passo do processo da renderização é o de desenhar os polígonos aplicando o material (iluminação e textura) correspondente a ele. Desenhar um polígono é como desenhar um conjunto de linhas que unidas tomam uma aparência. Neste ponto, é requerido uma interpolação (rasterização), que colori cada linha atribuindo uma iluminação. Para garantir um processo de renderização rápido a rasterização calcula a iluminação para cada linha, onde perde-se alguns efeitos de iluminação. Mesmo assim a demanda por processamento é grande cabendo a este processo ser realizado por um hardware específico, que é a placa gráfica, ou GPU - *Graphic Processor Unit*.

A rasterização também pode ser feita com o cálculo da iluminação para cada um dos pixels que formam um polígono (CLUA; BITTENCOURT, 2005), o que permite que uma série de efeitos de iluminação (blur, bump-mapping, especularidade) possa ser utilizada. Mas essa solução só pode ser aplicada com o suporte das placas gráficas para a tecnologia de pixel shaders, que trata cada pixel separadamente.

## 4.2.2 Engine de Física

O corrente estado da arte da computação gráfica nos permite explorar a interatividade dos dados em três dimensões. Nos jogos, um dos desafios desta área é a parte da simulação física, essencial para que o jogo se torne envolvente. Assim um carro não pode começar a andar com toda a velocidade, ele tem que ir acelerando aos poucos; ou um personagem não pode atravessar uma parede.

De acordo com Clua e Bittencourt (2005) os cálculos básicos de física num jogo são:

- **Colisão:** A utilidade principal da física nos jogos é a detecção de colisão. Esse processo físico não é trivial exigindo de avançados algoritmos para detectar e responder a colisões de forma realista. Para isto, os motores podem realizar os seus cálculos basicamente sobre duas formas: os polígonos ou *BoundVolumes* (são objetos simples como uma caixa ou esfera utilizados para representar objetos mais complexos). *BoundVolumes* são muito usados por exigir uma menor quantidade de processamento, apesar de tornar a simulação menos realística (EBERLY, 2003);
- **Resultante de forças:** É a responsável pela simulação de movimento nos objetos. O movimento pode ser visto como uma configuração de um objeto que muda sobre

o tempo como o resultado da alteração da posição, da orientação ou da deformação. Outra funcionalidade da engine física é calcular a resultante da força que resolva uma colisão (BERGEN, 2003).

Uma biblioteca de simulação física livre é a *Open Dynamics Engine* (ODE). Suas duas principais funcionalidades são: a dinâmica de corpos rígidos e a detecção de colisão. Ela é útil para a simulação de veículos, ambientes de realidade virtual e criaturas. Desenvolvida em C++ ela pode ser facilmente usada junto a um motor de jogos. Também é possível que o seu código fonte seja mudado de acordo com a necessidade ou ainda que partes do seu código seja utilizado para criar um engine personalizado. Com ele é possível simular um mundo com propriedades físicas usando variáveis como massa, velocidade e fricção.

### 4.2.3 Engine de som

A percepção sonora nos jogos é de fundamental importância para aumentar a imersividade. Nesse sentido o engine de som é o componente responsável pela manipulação de áudio de um motor de jogos. Uma alternativa é a API OpenAL, um software livre multiplataforma utilizado para adicionar sons 3D em alguns aplicativos e principalmente em jogos. OpenAL é capaz de gerenciar uma coleção de fontes de áudios em movimento no espaço 3D que são apreciados por um único ouvinte naquele espaço. Suas funcionalidades incluem a criação de efeitos de atenuação de acordo com a distância ou distorções resultantes da velocidade do ouvinte (OPENAL, 2009).

### 4.2.4 A Escolha do Engine

Existem muitos motores de jogos, tendo em Game Programming Wiki (WIKI, 2009) uma excelente lista. Mas para a escolha de um motor diversos fatores devem ser considerados (CLUA; BITTENCOURT, 2005):

- **Orçamento disponível:** Engines podem ter diversos recursos, sendo os mais caros, os mais completos além de normalmente terem uma equipe de apoio responsável por desenvolver plug-ins específicos, adaptar funcionalidades do engine para as necessidades particulares, entre outros;
- **Tipo de jogo a ser desenvolvido:** Motores de jogos podem ser criados para o desenvolvimento de diversos tipos de jogos ou para um determinado tipo. Por exemplo, o *Quake Engine* permite desenvolver jogos no estilo FPS (*First Person Shooter*).

- **Tempo:** A escolha do motor também é determinado pelo tempo disponível para a produção do jogo. O desenvolvimento de um jogo pode ser feito em pouco tempo quando utilizado soluções padrões que um motor oferece. No entanto, tais jogos se assemelharam a outros jogos construídos sobre a mesma condição.
- **Plataforma:** Um jogo pode ser construído para rodar em plataformas do tipo: PC, X-Box, GameCube, PlayStation, MacOS, entre outras. Normalmente quanto maior a quantidade de plataformas que o jogo pode rodar maior será o custo do motor.
- **Documentação oferecida:** Para que o programador utilize adequadamente os recursos proporcionados pelo motor é preciso uma documentação eficiente.
- **Ferramentas disponíveis:** Os conversores e exportadores contidos em cada motor permitem utilizar outros softwares para programar seu SDK ou para modelar os objetos 3D. Cada ferramenta necessita de ferramentas específicas cabendo ao desenvolvedor determiná-las.

O reaproveitamento de componentes já previamente projetados é mais eficaz do que recriar frequentemente as mesmas soluções. Para Rollings e Morris. (2004), à medida que a complexidade dos jogos aumenta, mais difícil se tornará para que se cumpram prazos em um tempo hábil. Nesse sentido Clua e Bittencourt (2005) recomenda que tanto no setor industrial como no meio acadêmico se empregue o reaproveitamento de soluções existentes.

### 4.3 A Ferramenta jMonkeyEngine

O jMonkeyEngine é um engine de abstração de nível médio, isto é, possibilita que o desenvolvimento de um jogo seja realizado necessitando do conhecimento de programação orientada a objeto, pois esse motor foi desenvolvido sobre a linguagem de programação Java. O desenvolvimento de jogos sobre ele é relativamente simples e rápida, contando com a disponibilidade de diversos exemplos que mostram muitas das suas funcionalidades. O jME é totalmente open source (código aberto) sobre a licença BSD, o que significa que ela pode ser usada para hobby ou comércio. Todo o código, como também diversos tutoriais podem ser encontrados no site oficial (JMONKEYENGINE, 2009). Adicionalmente existe outra engine criada sobre o jME para a simulação física, a jMEphysics. Por ser desenvolvida em Java, jogos criados são executáveis nas plataformas Windows, Linux e Mac.

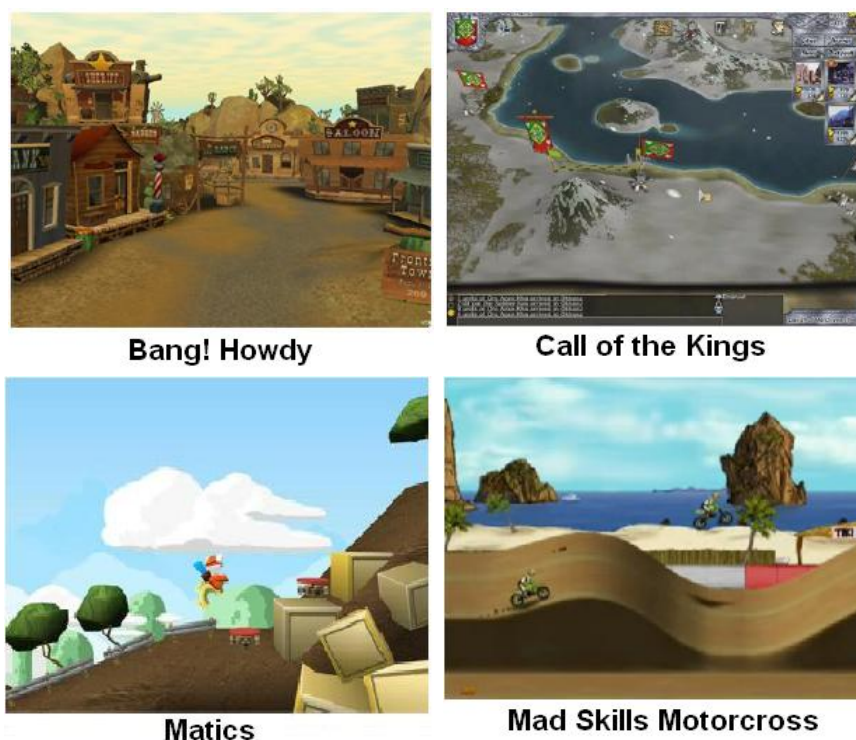


Figura 7: Exemplos de jogos comerciais construídos em jME

A figura 7 mostra alguns exemplos de jogos comerciais construídos em jME que revela a capacidade dessa ferramenta para o desenvolvimento de jogos.



Figura 8: Arquitetura da engine jMonkeyEngine.

A figura 8 mostra a arquitetura do jME, onde observa-se módulos responsáveis pelo tratamento do áudio, de colisão, de importação entre outros.

A engine preferencialmente pode ser utilizada através de softwares de programação que suportam o Java, como o NetBeans (NETBEANS, 2009) e o Eclipse (ECLIPSE, 2009).

O jMonkeyEngine é então uma aplicação de manipulação de cenas em 3D baseadas

em grafo que permite criar jogos e aplicações de alta qualidade com o apoio de gráficos e som. Para isso tendo o Java como plataforma suporte é utilizada uma camada JNI para acessar bibliotecas nativas e comunicar diretamente com o áudio, vídeo e o dispositivo de entrada de hardware.

### 4.3.1 Características

Há diversos elementos que devem ser entendidos para se começar a programar (JMONKEYENGINE, 2009):

- **Modularidade:** desenvolvida sobre esse conceito, permite a facilidade de utilização do jME, além de assegurar uma rápida capacidade da engine de fazer uso de novas características.
- **SceneGraph**(Cenas baseadas em Grafo) : é a estrutura de dados principal. Ela consiste em um grupo de nodos hierárquicos onde estão informações sobre a construção do ambiente do jogo. O relacionamento entre os dados do jogo (geometria, som, física, etc) estão principalmente em uma estrutura de árvore, que tem nos nodos folha a essência dos elementos do jogo (cubos, esferas, terrenos e etc). Estes elementos são renderizados na tela do monitor ou reproduzidos pela placa de som. Tipicamente, o scene graph é representado por um grande conjunto de dados (contidos na árvore) que podem ser encontrados facilmente a partir de uma única peça (nodo raiz ou algum nodo pai da árvore ). Essa organização do scene graph é muito importante, pois uma grande quantidade de processamento pode ser removida apenas estruturando a árvore de forma diferente.
- **Contexto Gráfico:** um jogo não pode ter início sem que uma janela, um sistema de entradas, uma câmera e um sistema de repetição (responsável por verificar as entradas, calcular e retornar algum resultado) tenha sido criados. Em jME o nível de abstração desse motor permite que esses parâmetros sejam iniciados facilmente, como também sejam modificados de acordo com as necessidades.
- **Básicas Construções:** Todos os elementos gráficos são originários de uma classe chamada Spatial, desta surgem dois tipos: a Geometry, estendida por outras classes que podem criar cubos, esferas ou terrenos na cena; e a Node, que abstrai características gerais de classes responsáveis pela geração de luz, mas que também é utilizada como um nodo pai que armazena informações sobre seus nodos filhos. Outra construção básica é a classe InputAction que irá tratar todas as entradas e suas ações associadas.

- **Os nodos da árvore (Node e Geometry):** Todo o nodo do tipo Node pode conter filhos (outros Node ou Geometry), enquanto Geometry não contem filhos. Todos os nodos filhos herdam as características (posicionamento, escala, iluminação, textura e etc) dos seus nodos pais, ou até mesmos na renderização se o nodo pai não é mostrado, então o nodo filho também não. Todo nodo contem informações importantes sobre si mesmo: - Transformação: define a orientação, a localização e a escala do nodo; - BoundingBox: define a area minima que contem um nodo particular. No Geometry o volume contem todos os seus vertices. No Node, o volume contém todos os vértices de seus filhos; - render state: esta relacionado a textura, definindo como o Geometry será mostrado na tela; - Controllers: utilizado para definir alterações nos nodos sobre um periodo de tempo. Ele inclui coisas como animação de um modelo, atributos fisicos (lançamento de uma bola), etc.

### Exemplo de SceneGraph

O seguinte exemplo (Fig. 9) <sup>2</sup> mostra uma possível organização da cena de um jogo. As três partes essenciais feitas sob a classe Geometry são: a espaçonave, a terra e a lua. A espaçonave é livre para voar no universo e está ligada ao nodo raiz. A lua rotaciona sobre a terra, então ela está ligada ao nodo da terra para permitir que ela sempre rotacione ao redor dela não importando onde ela esteja.

O interessante é que o Scene Graph não tem que ser arrumado parte por parte separadamente. Algumas vezes é melhor organizar os objetos de forma que eles compartilhem estados. Por exemplo, se você tem um conjunto de árvores que tem a mesma textura, é possível colocar estas sobre um único nodo (atribuindo o estado de renderização para aquele nodo pai) em vez de colocá-la em cada nodo de cada árvore. Assim é possível tentar encontrar um balanço das propriedades na árvore (JMONKEYENGINE, 2009).

### 4.3.2 Criando a primeira aplicação

O jMonkeyEngine foi desenvolvido para ser uma rápida engine de gráficos em tempo real. Para cumprir esta necessidade ela prove um alto desempenho na criação de cenas baseadas em grafos através de um sistema de renderização (JMONKEYENGINE, 2009).

A essência do sistema jME e o scene graph. O scene graph como já foi explicado é a estrutura de dados principal para a construção do jogo. Os elementos desta estrutura serão basicamente ou renderizados na cena, ou tocados por algum aparelho de som.

---

<sup>2</sup> Figura adaptada. Disponível em [http://www.jmonkeyengine.com/wiki/doku.php/what\\_is\\_a\\_scene\\_graph?s\[\]=ship](http://www.jmonkeyengine.com/wiki/doku.php/what_is_a_scene_graph?s[]=ship) Acesso em 8 de novembro de 2009.

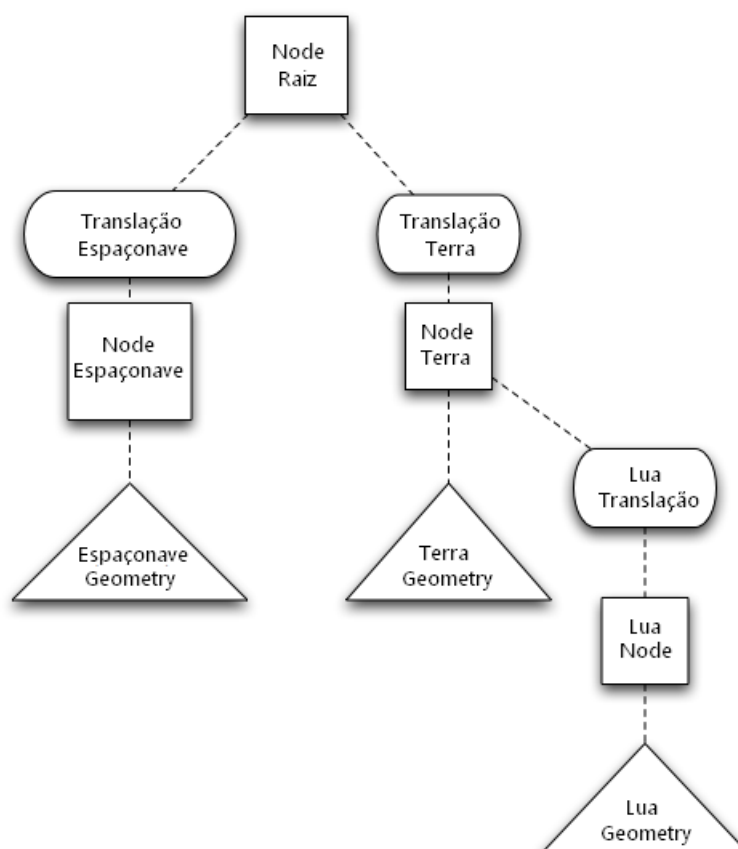


Figura 9: Exemplo da estrutura de um SceneGraph.

Mas enquanto o scene graph é a essência dos elementos, o núcleo do jogo está no loop (processo de repetição), parte do código que é repetidamente executada durante a vida do jogo. Este loop, é o responsável por coordenar as atualizações (sons e imagens que sofreram alguma alteração) a cada ciclo (JMONKEYENGINE, 2009).

Todas as subclasses de AbstractGame são responsáveis pela criação deste loop que só terá início quando o método *start* é chamado (JMONKEYENGINE, 2009). Embora cada subclasse varie ligeiramente este laço, todas elas seguem o mesmo estilo básico. (Fig. 10)<sup>3</sup>

<sup>3</sup> Figura adaptada. Disponível em [http://www.jmonkeyengine.com/wiki/doku.php/main\\_game\\_loop?s\[\]=loop](http://www.jmonkeyengine.com/wiki/doku.php/main_game_loop?s[]=loop) Acesso em 8 de novembro de 2009.





Figura 10: Representação do processo de repetição (loop) de um jogo em jME.

## 5 *Desenvolvimento*

Este capítulo apresenta a especificação e o desenvolvimento do sistema, descrevendo seus requisitos e funcionalidades. O projeto do jogo foi pensado segundo os critérios mostrados no capítulo 3, onde são descritos formas adequadas para inserção de conteúdos em jogos educativos, sem esquecer o aspecto da diversão. Ao mesmo tempo, na parte técnica foi utilizada as informações do capítulo 4, que fala sobre a concepção das idéias e ferramentas empregados no desenvolvimento dos jogos comerciais. Por fim será mostrada a especificação dos requisitos do sistema, englobando os diagramas UML.

### 5.1 Game Design

Esta seção tem por objetivo descrever a etapa de concepção e design do jogo.

#### 5.1.1 Visão geral do Jogo

O jogo proposto aborda a aprendizagem das leis de trânsito. O objetivo é informar as pessoas sobre questões do trânsito na busca da conscientização das diversas responsabilidades (muitas vezes pouco conhecidas) que um motorista deve ter para poder dirigir de forma a garantir a sua segurança e a das outras pessoas.

O funcionamento do jogo se dá através de uma interface interativa e imersiva, operando com um ambiente 3D com visão em 3ª pessoa, onde uma pessoa dirige um carro. Para garantir que este sistema tenha o almejado valor pedagógico, a sua interface foi definida considerando-se os tópicos 3.1.1 e 3.1.2 abordados no capítulo 3.

##### 5.1.1.1 O Jogo e as Leis de Trânsito

O ambiente do jogo foi definido como uma cidade em 3D (Figura 11), onde se fazem presentes elementos da sinalização do trânsito como semáforos, placas de trânsito, faixa de pedestre e etc.

O jogo é baseado sobre a pontuação dentro de um determinado tempo e utiliza o conceito de que as multas de trânsito são classificadas de acordo com a sua gravidade (os tipos de multa utilizados no jogo são classificadas em: Leve, Média e Grave).

A idéia do jogo é fazer com que o usuário primeiramente conheça sobre uma multa para depois tentar fazer algumas combinações, pré-estabelecidas de acordo com o tipo de multa, levando assim o usuário através do processo de conhecer e exercitar, à aprender. Dessa forma se consegue atingir a proposta apresentada no capítulo 3 de que o aprendizado se dá através do comportamento de: descrição - execução - reflexão - depuração.

Para poder conhecer uma multa existem algumas caixas coloridas, como mostrado na figura 12, espalhadas pela cidade que, quando encontradas, mostram a informação sobre a multa. Para poder fazer as combinações e assim ganhar pontos existem caixas brancas que mostram informações sobre a multa, mas não falam sobre qual o tipo dela. Para ajudar nesse processo o jogador pode acessar um mapa da cidade, que mostram os lugares onde as caixas se encontram. Na cidade também existem moedas como mostra a figura, que dão pontos extras ao usuário e torna o jogo mais atrativo.

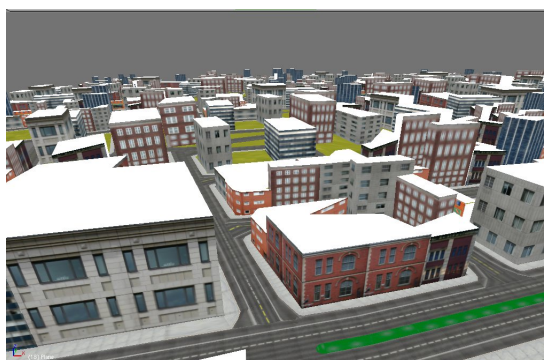


Figura 11: Modelo tridimensional da cidade.



Figura 12: Modelo tridimensional da moeda e da caixa.

## 5.2 Ferramentas Utilizadas

No capítulo 4 é discutido sobre o desenvolvimento de jogos 3D, onde são apresentadas algumas ferramentas que podem ser utilizadas para o desenvolvimento de jogos, sendo neste trabalho utilizado o Blender, o GIMP e o jME.

## 5.3 Definição dos Requisitos

A definição de requisitos de um jogo compreende em explicar as características respectivas a ele, como por exemplo, quais os desafios que serão proporcionados ao usuário. Além da definição do jogo já apresentada anteriormente segue abaixo a descrição dos requisitos referentes ao sistema.

### 5.3.1 Requisitos Funcionais

A descrição dos Requisitos Funcionais do sistema mostram as diversas funções buscadas para o desenvolvimento do software, sendo elas:

**Mensagens:** o software deverá apresentar mensagens mostrando algumas infrações de trânsito, que podem ser cometidas como, por exemplo, ao passar por um sinal vermelho. O sistema deverá também mostrar informações sobre as leis de trânsito e as possíveis punições;

**Refazer uma Ação:** dentro do desafio do jogo proposto deve existir uma possibilidade de que o usuário refaça uma ação caso ele tenha cometido um erro;

**Instruções:** o software deve disponibilizar em algum momento do jogo, as suas instruções;

**Ranque:** o sistema deverá armazenar e mostrar a pontuação atualizada dos jogadores;

**Estado do Jogo:** o sistema deve possibilitar ao jogador ver o estado do jogo (Ex. pontuação, tempo para terminar) em um dado momento;

**Novo Jogo:** o jogador poderá, a qualquer momento, solicitar um novo jogo;

**Termino do Jogo:** o jogo terminará depois de um determinado tempo;

### 5.3.2 Requisitos não funcionais

Os Requisitos não Funcionais expressam como o sistema deve ser feito, sendo eles:

**Ambiente do Jogo:** o ambiente do jogo deve ser construído sobre três dimensões;

**Assunto:** o assunto do jogo deve estar relacionado as leis de trânsito;

**Usabilidade (jogabilidade):** dentro da proposta de um jogo educativo, ele deve ter uma interface fácil e atrativa;

**Confiabilidade:** por se tratar de um jogo voltado a educação, existe a necessidade da confiabilidade das informações;

**Desempenho:** garantir quantidade de quadros-por-segundo adequada (no mínimo 25 frames por segundo);

**Escalabilidade:** para este software pode-se modularizar o desenvolvimento da quantidade de informação disponível, pois existe a possibilidade de adicioná-los ao jogo sem a necessidade de se refazer o programa completo.

## 5.4 Definição dos Diagramas

Nesta seção, são descritos os seguintes diagramas referentes ao sistema: diagrama de casos de uso, diagrama de classes, diagrama de sequência e o diagrama de estado.

### 5.4.1 Casos de uso

O diagrama de caso de uso descreve as funcionalidades propostas e que foram desenvolvidas para o sistema. A figura 13 mostra cada caso de uso, logo abaixo uma descrição de cada uma é dada.

Segue abaixo algumas informações sobre os casos de uso.

**Mostrar Instruções:** é o processo em que o ator visualiza as instruções do jogo;

**Jogar:** processo em que o ator passa a jogar o jogo.

**Visualizar ranking:** processo no qual o ator passa a visualizar o ranking de recordes do jogo

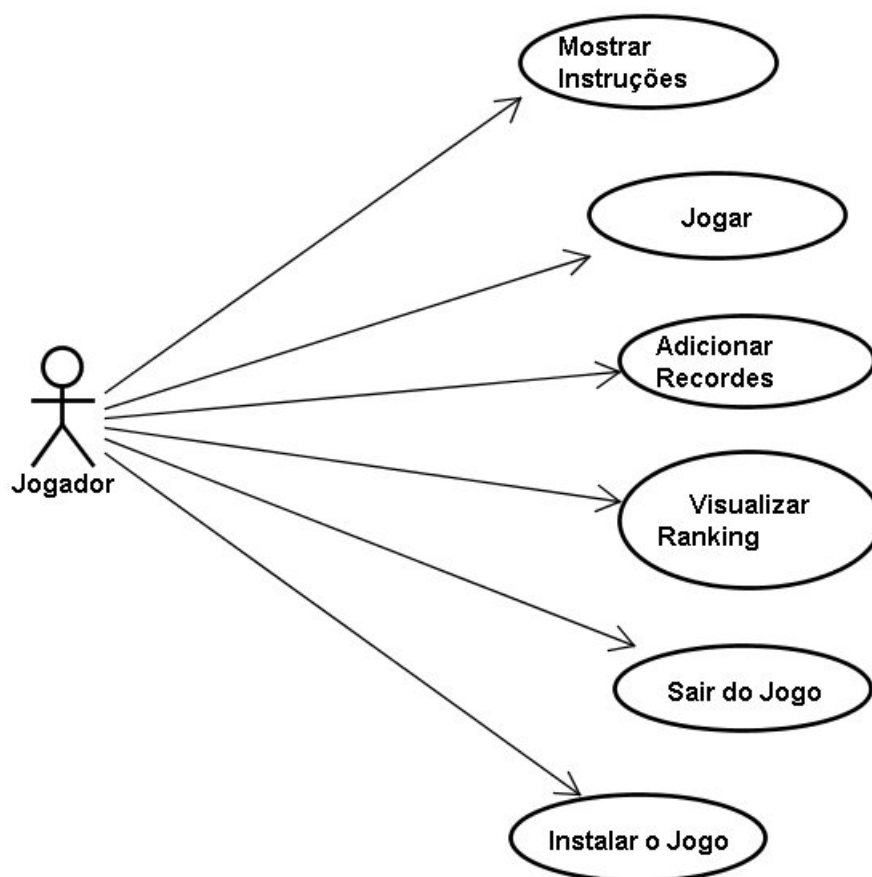


Figura 13: Casos de Uso

**Adicionar Recordes:** trata o processo de inserção do nome do ator no ranking de recordes do jogo. Para tanto o ator terá que ter jogado o jogo e ter feito uma pontuação mínima para a entrada no ranking

**Sair do jogo:** processo no qual o ator enquanto está jogando sai do jogo

## 5.4.2 Diagrama de Classes

Um jogo é composto por estados, como por exemplo a janela principal ou o jogo em si. Cada estado é responsável por inicializa os seus objetos, carregar imagens e sons, gerenciar entradas e saídas, desenhando e atualizando os quadros do jogo. Os estados do jogo são aqueles que implementam a interface `InterfaceGameState`

Cada estado deve ser gerenciado de forma que quando um estiver ativo o outro estará desativado. A classe responsável por este gerenciamento é a `GameManager`. A ele está associado o conceito de Singleton, que é um padrão de projeto que garante a existência de apenas uma instância de uma classe.

`InterfaceGameState` é a classe abstrata que permite alocar os estados do jogo a

uma única camada para que GameManager consiga fazer o gerenciamento do jogo. São definidos os métodos enter (mostrar o estado na tela), pause (parar a execução de um estado), exit (retirar o estado) e resume (voltar o estado pausado) .

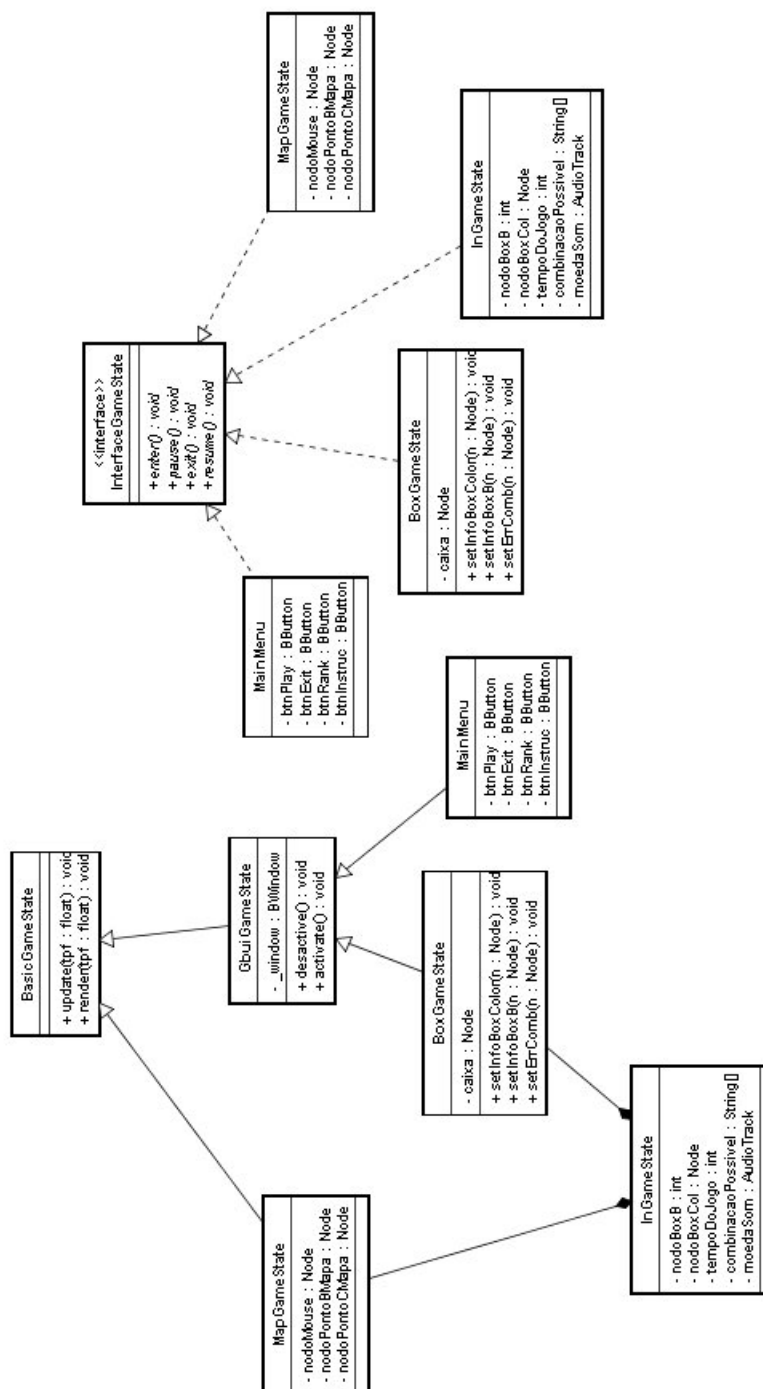


Figura 14: Diagrama de Classes

ClassLoader é a classe responsável por armazenar objetos e fazer-os acessíveis para a aplicação agindo como um container. A ela também está associado o conceito de Singleton. GbuiGameState é a classe abstrata que permite alocar os estados do jogo destinados a

uma janela a uma única camada.

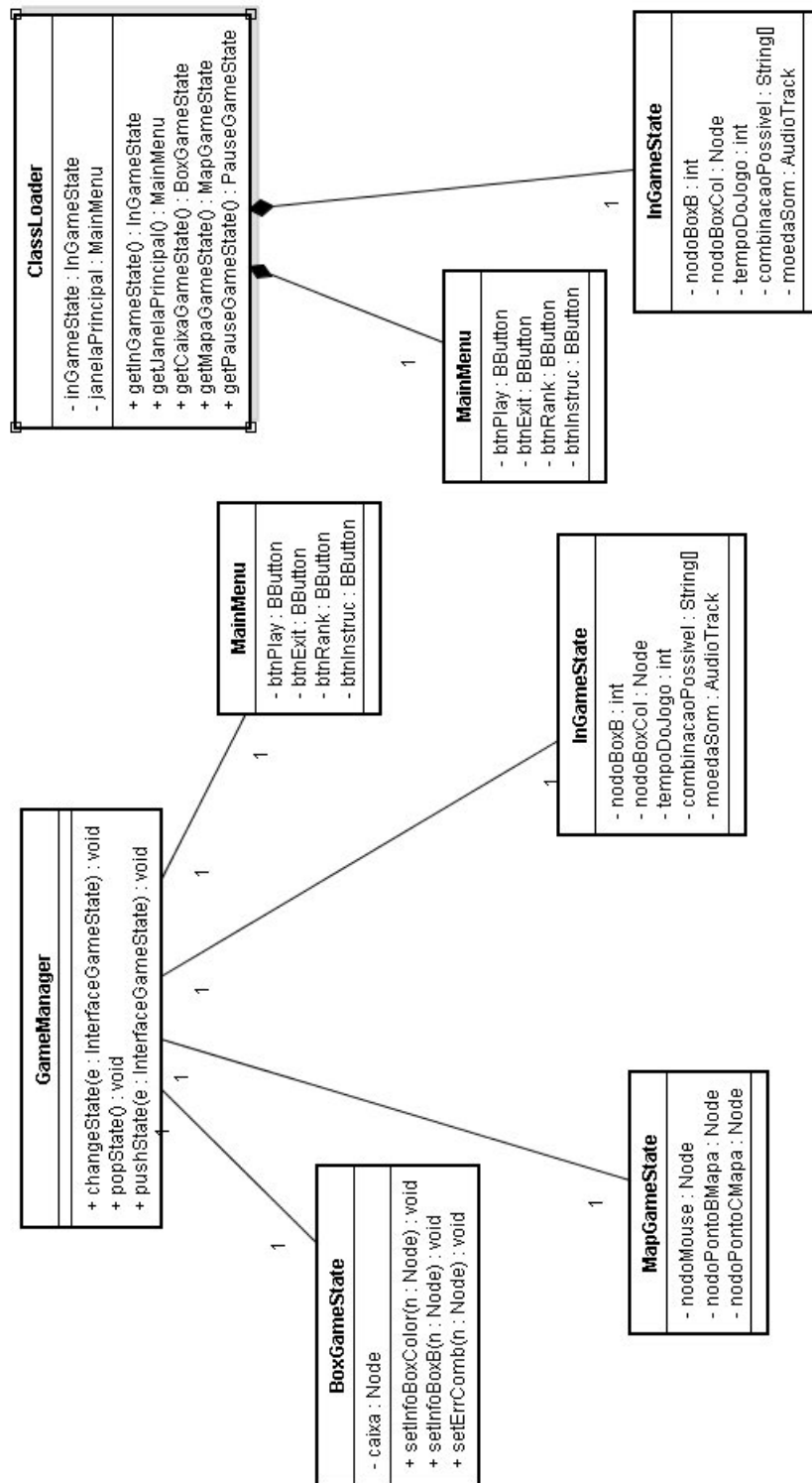


Figura 15: Complemento do Diagrama de Classes

A figura 14 e 15 mostram os diagramas de classes onde são identificadas as diferentes classes que compõem o sistema e como elas se relacionam.

JanelaPrincipal, BoxGameState e MapGameState são as classes que possuem os itens



e as ações de cada menu e sub-menu que utilizam dos recursos da classe `GbuiGameState`. A classe `JanelaPrincipal` é a responsável por gerar o menu principal do jogo. `BoxGameState` tem por objetivo exibir as informações relacionadas as multas de trânsito. `MapGameState` cuidará de mostrar o mapa da cidade, dentre outras funcionalidades descritas na seção 7.

`GbuiGameState` defini uma interface para classes destinadas a criação de GUI's que utilizam a API GBUI.

### 5.4.3 Diagrama de Sequência

O diagrama de seqüência, apresentado na fig. 16 mostra o fluxo do jogo completo. Esse fluxo demonstra a seqüência de telas e as ações requeridas para que ocorra a transição entre as classes. Toda a seqüência é gerenciada por `GameManager`, iniciando pela janela principal `MainMenu` indo para o jogo `InGameState` onde são atualizadas as saídas de acordo com as entradas, seguindo para salvar a pontuação do jogo e retornando a janela principal.

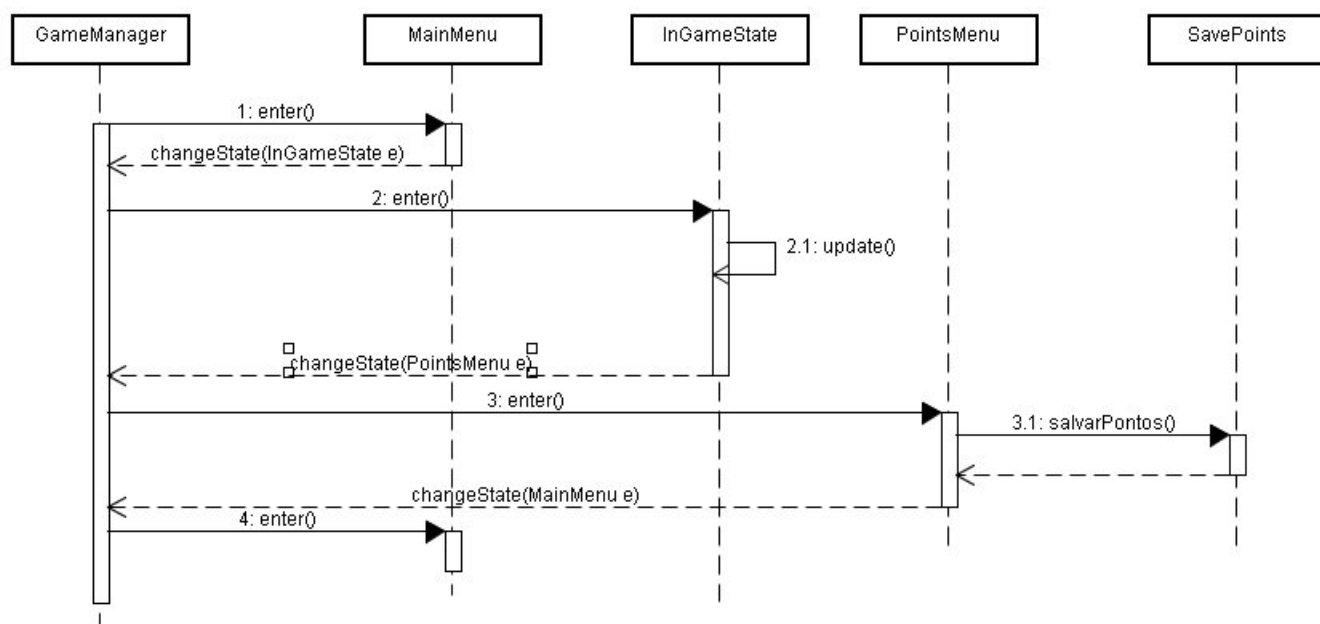


Figura 16: Diagrama de Sequência: Jogar

O diagrama de seqüência da fig. 17, complementa o diagrama anterior mostrando as ações de ver o mapa e ver informações sobre as caixas. Neste caso, representa-se o caso onde o jogo ao se encontra no estado `InGameState` pode seguir para o estado `InBoxGameState` (mostra a informação da caixa) ou para o estado `InMapState` (mostra o mapa).



a pontuação ao encontrar uma moeda ou uma combinação for realizada com sucesso; mostrar um alerta quando uma infração for detectada (andar na contra mão, subir na calçada, sinal vermelho), mostrar mapa quando solicitado pelo usuário.

## 5.5 Arquitetura

A arquitetura do sistema foi planejada sobre o framework jME. Na figura 19 é mostrada a arquitetura interna do jogo.

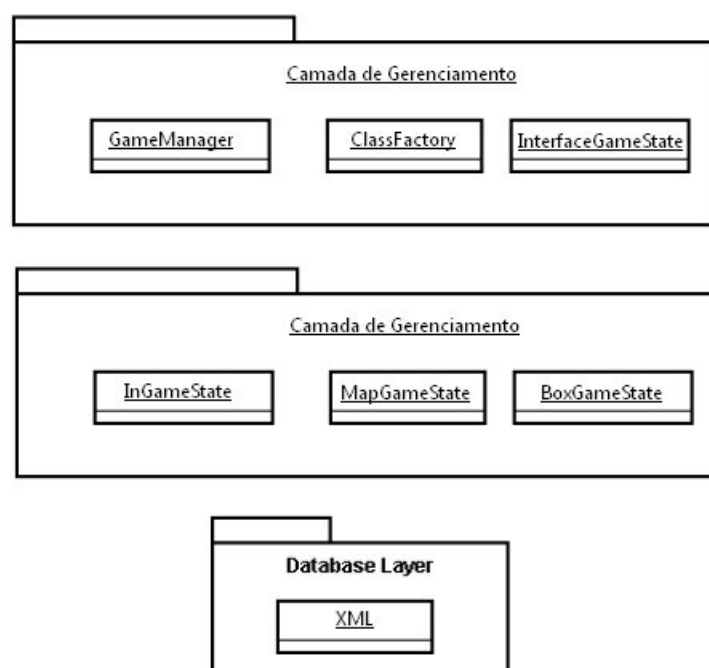


Figura 19: Arquitetura Interna

A arquitetura foi dividida em três camadas. A primeira camada é de gerenciamento (Camada de Gerenciamento), a responsável por controlar os estados do jogo. A segunda camada é onde estão os estados do jogo, responsáveis por tratar as entradas e o gerenciamento da renderização das imagens. A terceira e última camada é a responsável pelo gerenciamento do banco de dados, inserindo e capturando dados do rank do jogo.

### 5.5.1 Restrições

O objetivo principal é uma versão para computadores para um único jogador. A API jME possibilita que o mesmo código seja executado nos ambientes Windos, Linux e Max OS X, mas requer alguns requisitos para a correta execução, como: CPU - 2 Ghz; Memória - 1GB; Placa de Video - NVIDIA GeForce 7600, ATI Radeon X700 ou melhor.

## 5.6 Metodologia

Este projeto adota o modelo incremental, que de acordo com Pressman (2006) é um modelo de desenvolvimento onde o software é projetado, implementado e testado incrementalmente (a cada passo um pouco mais do sistema é construído) até o produto estar terminado. Este modelo combina os elementos do modelo em cascata com a filosofia interativa da prototipação.

O sistema foi decomposto em um número de componentes, cada um sendo projetado e construído separadamente. Depois que um requisito era identificado, um componente era construído o que possibilitava analisar estratégias alternativas de projeto e decidir qual a melhor para o projeto.

Este modelo foi importante também no desenvolvimento desse projeto, pois com o não conhecimento inicial da ferramenta jME, a criação de cada componente separadamente ajudou no aprendizado da ferramenta e na criação do jogo.

No desenvolvimento do sistema foi priorizado o desenvolvimento de componentes que possibilitassem a reutilização de código. Dentre as classes que foram construídas com este objetivo são: GameManager, BasicGameState, GbuiGameState.

## 5.7 Implementação

A papel do jogo é o de mostrar algum conhecimento que esteja relacionado às leis de trânsito, o que foi implementado na forma de um desafio relacionado as informações sobre as multas e de “obstáculos” relacionados a infrações de trânsito. Como descrito anteriormente, o desafio criado para o jogo busca fazer com que o usuário faça uma associação entre a informação de uma multa e o seu determinado tipo. Isto para a implementação consistiu basicamente no desenvolvimento de um carro que possibilita ao usuário andar pela cidade para que ele encontre alguma informação afim de cumprir o desafio do jogo. Mas o usuário enquanto anda com o carro deve observar alguns “obstáculos”, pois dentro da cidade existem algumas leis que devem ser seguidas, o que nesse sentido foi implementado a detecção de: andar na contra-mão, subir na calçada, não respeitar o sinal vermelho. A seguir são descritos maiores detalhes sobre a implementação.

O primeiro contato que o jogador tem com o jogo é através do menu principal mostrado na figura 20. Este menu é gerado pela classe MainMenu mostrada no diagrama de classe. Quando esta janela é exibida o programa aguarda por uma ação do usuário. A partir daí

as possíveis opções são: jogar o jogo; ver as instruções do jogo, ver o rank ou sair do jogo.

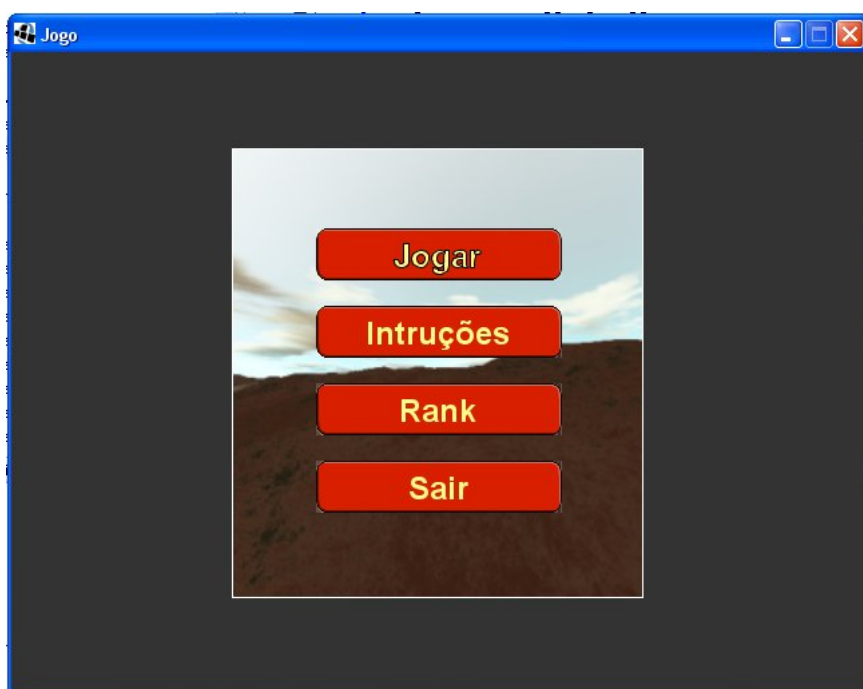


Figura 20: Menu Principal do Jogo



Figura 21: Jogando o Jogo

A figura 21 apresenta a primeira imagem que é mostrada quando o jogo é iniciado. Nela estão presentes a pontuação do jogo, as combinações que estão sendo feitas e o tempo restante para terminar o jogo. O jogo é gerado pela classe `InGameState`, que quando iniciado carrega os cubos, as moedas, o carro e toda a cidade. Depois de iniciada

esta classe fica esperando por alguma ação que pode ser: movimentar o carro através das setas do teclado, mostrar o mapa da cidade ao clicar na tecla M, ou ainda mostrar as informações sobre multas ao encontrar alguma caixa. Desconsiderando a movimentação do carro todas as outras ações irão chamar outras classes responsáveis por gerar novas janelas. Outras tarefas também feitas por esta classe é contabilizar os pontos e as combinações corretamente.

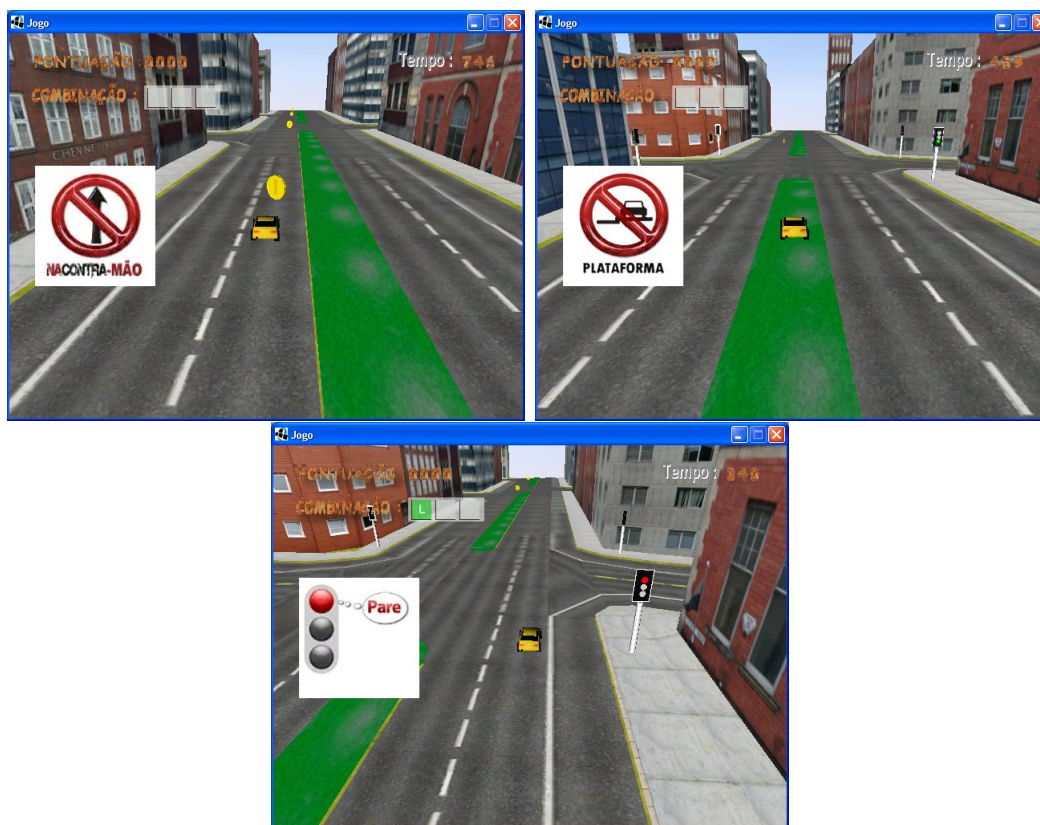


Figura 22: Infrações que podem ser cometidas no jogo.

Outra funcionalidade da classe InGameState é identificar e mostrar alguma infração relacionada a andar na contra mão, subir na calçada ou não respeitar o sinal vermelho. Estas ações podem ser vistas na figura 22

O mapa mostrado na figura 23 permite ao jogador localizar na cidade onde estão as caixas coloridas e as caixas brancas, representados respectivamente por pontos coloridos e brancos, enquanto a localização do carro na cidade pode ser identificada através do ponto preto. O mapa tem uma grande importância em relação a idéia base do jogo, que é tentar fazer com que o usuário pense um pouco sobre multas, mas que neste caso está relacionado a associar uma multa a seu determinado tipo. Ao ser iniciado o mapa, o tempo do jogo pára e o jogador pode então movimentar o mouse e passar pelos pontos brancos que mostra a informação que está contida ali. Os pontos coloridos só indicam uma localização e mais nada, pois para saber qual o tipo da multa o jogador deve encontrar



na cidade uma caixa colorida enquanto que as caixas brancas devem ser encontradas para fazer as combinações. Sobre o mapa ainda são encontradas as informações que indicam qual ordem que as caixas devem ser encontradas para poder pontuar. O mapa e suas funcionalidades são geradas pela classe MapGameState.



Figura 23: Mapa da Cidade 3D

A classe BoxGameState é a responsável por mostrar as informações contidas nas caixas, como mostrado na figura 24. A representação da informação de cada caixa é diferente de acordo com a sua funcionalidade no jogo, a cor de fundo sempre estará associada à cor da caixa e conseqüentemente ao tipo de multa. Para as caixas brancas, existe a opção de aceitar ou não fazer uma combinação com o que foi apresentado. Ao aceitar se a combinação estiver errada uma mensagem de erro é lançada.

Dentro dos desafios para a construção do jogo estavam: a manipulação de árvores, pois como descrito no capítulo “Desenvolvimento de Jogos 3D” a ferramenta jME se baseia em árvores para a renderização de cenas; fazer funcionar as texturas e as entradas corretamente; manusear variáveis importantes para o funcionamento do sistema sem que hajam sobrecargas de memória e de execução. Essas questões estão representadas no Apêndice A que mostra o código do jogo referente a parte de atualizações da classe InGameState. Ainda no Apêndice B podem ser encontradas as informações sobre multas utilizadas neste trabalho.



Figura 24: Mostrando as informações sobre multas.



## 6 *Conclusão*

Ao longo desse trabalho, discutiu-se o uso dos jogos eletrônicos como ferramenta de complementação do aprendizado, onde foi proposto um jogo voltado as Leis de Trânsito. Para isso foi desenvolvido um ambiente tridimensional (uma cidade) dentro da qual o jogador, controlando um carro, se destina a fazer algumas tarefas propostas, possibilitando assim a descoberta de informações relacionadas a multas de trânsito.

Esta aplicação usa como ferramenta principal o jME, um framework livre para desenvolvimento de jogos, que permite a criação de jogos para computadores. Outra ferramenta utilizada foi o Blender, destinada a construção de objetos 3D. Ainda para o objetivo deste trabalho um estudo sobre desenvolvimento de jogos educativos foi realizado a fim de garantir ao sistema desenvolvido o devido valor educacional.

Quis-se frisar que os jogos eletrônicos recebem o rótulo de divertimento, de algo aprazível, mas que pode-se transferir esse prazer para o aprendizado, pois o mesmo contribui com aspectos sociais e psicológicos, provendo benefícios à sociedade graças às características que lhe são inerentes. Com relação as possibilidades acadêmicas, o desenvolvimento de jogos eletrônicos pode motivar o interesse do estudo de áreas da computação como: engenharia de software, redes de computadores e inteligência artificial, computação gráfica.

Dentro do aspecto da educação os jogos eletrônicos ganham importância na medida em que o processo de informatização da sociedade brasileira é irreversível e que se a escola também não se informatizar, correrá o risco de não ser mais compreendida pelas novas gerações, mostrando a necessidade da utilização de meios educacionais que se adaptam a realidade das pessoas, o que hoje é centrada na tecnologia onde se encontram os jogos.

Expor sobre a aplicabilidade dos jogos na educação formal não é tarefa fácil (CLUA; JUNIOR; NABAIS, 2002), já que no Brasil, é, infelizmente, um dos setores que mais desperta a preocupação, devido à baixa qualidade de ensino que oferece aos seus alunos, principalmente quando consideramos o ensino de crianças e adolescentes. Implementar jogos educativos eletrônicos na escola seria uma inovação, que pode ser muito bem explorada e

aproveitada.

Como objetivo do projeto foi desenvolvido o jogo eletrônico como software gerado por um projeto de engenharia, onde a intenção era o de mostrar dentro desta área que ainda está em desenvolvimento e que não existe muito embasamento teórico como jogos e educação podem se unir.

Seria interessante que a aplicabilidade do jogo desenvolvido fizesse parte dos estudos de temas transversais, que já são uma realidade legal nos Parâmetros Curriculares Nacionais (PCNs), pois muitas vezes com palavras não se consegue atingir o objetivo do jogo, que não é mudar o conceito que a pessoa tem, e sim instigar o usuário a se interessar pelo assunto, mostrando informações relacionadas ao trânsito.

Apesar da simplicidade do desafio proposto, o jogo foi implementado de forma a explorar recursos de imagem, sons e interatividade a fim de garantir um envolvimento sobre o assunto que se queria explorar. Nesse sentido, podendo sim ser utilizado para o aprendizado, mesmo não tendo sido usado na prática, justificado pela falta de tempo.

Cabe ainda ressaltar que este jogo pode além de ser utilizado como objeto de aprendizado, ainda aqui referindo-me ao código do programa, ser utilizado, reutilizado e estudado como ponto de partida para aqueles que queiram desenvolver projetos no mesmo sentido, ou ainda, queiram implementar novas idéias, novos desafios dentro do conceito do jogo.

Como trabalhos futuros o jogo pode receber versões que se destinem a ser jogado pela internet. Pode-se também investir em jogos que apresentem uma qualidade gráfica superior à atual, tanto em questões artísticas como na concepção de modelos tridimensionais melhores. Existe ainda a possibilidade de abranger a jogabilidade para se comportar múltiplos jogadores, uma vez que a versão atual permite apenas o jogo a um único jogador. Em relação à aplicabilidade do jogo à educação, pode-se elaborar planos de ação e testes, juntamente com profissionais da área para validar a sua aplicabilidade.

Por fim, destaca-se aqui a importância dos jogos eletrônicos na visão de Esteban Clua (apud MOITA, 2007):

Devido a repercussão dos jogos na nossa sociedade, não é preciso muito esforço para se concluir que os jogos eletrônicos vão exercer, nos próximos anos, uma forte influência na forma de pensar, no comportamento social, psicológico e educacional dos jovens.

## *Referências*

- ABT, C. C. *Serious Games*. [S.l.]: University Press of America, 1987.
- ALVES, L. *GAME OVER: jogos eletrônicos e violência*. [S.l.]: Editora Futura, 2005.
- ALVES, L. Jogos eletrônicos e screenagens: possibilidades de desenvolvimento e aprendizagem. In: \_\_\_\_\_. *Moita, F. M. G. S. C. e Silva, E. M. e Souza, R. P. (org.) Jogos Eletrônicos: Construindo Novas Trilhas*. 1. ed. Campina Grande: EDUEP, 2007. p. 63–81.
- AUDACITY. 2009. Disponível em: <http://audacity.sourceforge.net/> Acesso em: 13 de junho de 2009.
- AZEVEDO, E. et al. *Desenvolvimento de Jogos 3D e Aplicações em Realidade Virtual*. [S.l.]: Editora Campus: Rio de Janeiro, 2005.
- BATTAIOLA, A. L. Jogos por computador - histórico, relevância tecnológica e mercadológica, tendências e técnicas de implementação. In: *Anais da XIX Jornada de Atualização em Informática do XIX Congresso Nacional da SBC, Curitiba: PUCPR*. [S.l.]: XIX Jornada de Atualização em Informática, 2000. Disponível em: <http://www.design.ufpr.br/lai/arquivos/JogosComputadores.pdf>. Acesso em: 27 de março de 2009.
- BATTAIOLA, A. L. et al. Desenvolvimento de um software educacional com base em conceitos de jogos. *XIII Simpósio Brasileiro de Informática na Educação - SBIE - UNISINOS*, 2002.
- BERGEN, G. V. D. *Collision Detection in Interactive 3D Environments (Morgan Kaufmann Series in Interactive 3D Technology)*. [S.l.]: Morgan Kaufmann, 2003.
- BITTENCOURT, J. ao R.; GIRAFFA, L. M. M. *A utilização dos role-playing games digitais no processo de ensino aprendizagem*. [S.l.], 2003. Disponível em: <[http://www.pucrs.br/inf/pos/mestdout/rel\\_tec/tr031.pdf](http://www.pucrs.br/inf/pos/mestdout/rel_tec/tr031.pdf)>. Acesso em: 27 mar. 2009.
- BLENDER3D. 2009. Disponível em: [www.blender.org/](http://www.blender.org/) Acesso em: 13 de junho de 2009.
- BOBANY, A. *Video Game Arte*. [S.l.]: Teresópolis, RJ : Novas Idéias, 2008.
- BONGIOLO, C. E. F.; BRAGA, E. R.; SILVEIRA, M. S. *Subindo e Escorregando: jogo para introdução do conceito de adição de números inteiros*. [S.l.]: IV Congresso Congresso Ibero-Americano de Informática na Educação-RIBIE, Brasília: Universidade de Brasília, 1998.
- CAILLOIS, R. *Os jogos e os homens: a máscara e a vertigem*. [S.l.]: Lisboa: Cotovia, 1990.

- CLUA, E. W. G.; BITTENCOURT, J. R. Desenvolvimento de jogos 3d: Concepção, design e programação. In: *Desenvolvimento de Jogos 3D: Concepção, Design e Programação*. [S.l.]: XXV Congresso da Sociedade Brasileira de Computação, 2005.
- CLUA, E. W. G.; JUNIOR, C. L. de L.; NABAIS, R. J. de M. Importância e impacto dos jogos educativos na sociedade. *Workshop Brasileiro de Jogos e Entretenimento Digital*. SBC: Fortaleza, 2002.
- CRAWFORD, C. *Chris Crawford on Game Design*. [S.l.]: New Riders Publishers, 2003.
- EBERLY, D. H. *Game Physics (Interactive 3d Technology Series)*. [S.l.]: Morgan Kaufmann, 2003.
- ECLIPSE. 2009. Disponível em: <http://www.eclipse.org/> Acesso em: 13 de junho de 2009.
- FALKEMBACH, G. A. M.; GELLER, M.; SILVEIRA, S. R. *Desenvolvimento de Jogos Educativos Digitais utilizando a Ferramenta de Autoria Multimídia: um estudo de caso com o ToolBook Instructor*. [S.l.]: Revista Novas Tecnologias na Educação, Porto Alegre, v. 4, n. 1, 2006.
- Food Force. *Programa Alimentar das Nações Unidas*. [S.l.]: Disponível em: <http://food-force.educacional.com.br/>. Acesso em 7 de junho de 2009., 2009.
- FREUD, S. *Além do princípio do prazer e outros trabalhos*. [S.l.]: Imago: Rio de Janeiro, 1976.
- Games To Teach Project. *Desenvolvido em parceria com o MIT*. [S.l.]: Disponível em: <http://www.educationarcade.org/gtt/proto.html>. Acesso em 7 de junho de 2009., 2003.
- GIMP. 2009. Disponível em: <http://www.gimp.org/> Acesso em: 13 de junho de 2009.
- GREENFIELD, P. M. *O Desenvolvimento do Raciocínio na Era da Eletrônica - Os Efeitos da TV, Computadores e Videogames*. [S.l.]: Summu, 1988.
- HopeLab. 2009. Disponível em: <http://www.hopelab.org/>. Acesso em: 7 de julho de 2009.
- HUIZINGA, J. *Homo ludens: o jogo como elemento da cultura*. [S.l.]: São Paulo: Perspectiva, 1996.
- JMONKEYENGINE. 2009. Disponível em: <http://www.jmonkeyengine.com/> .Acesso em: 13 de junho de 2009.
- JULIAN, A. et al. Serious game: just a question of posture? In: *AISB'07 Symposium: AI and Narrative Games for Education, Newcastle University*. [S.l.: s.n.], 2007.
- LACERDA, R. D. A. *Proposta de um modelo para análise de requisitos de software educativo*. [S.l.]: Programa de Pós-Graduação em Educação - UnB, 2007.
- LAMOTHE, A. *Tricks of the 3D Game Programming Gurus-Advanced 3D Graphics and Rasterization*. [S.l.]: Pearson Education, 2003.

LUCENA, M. *A Gente é uma Pesquisa: Desenvolvimento Cooperativo da Escrita Apoiado pelo Computador*: Dissertação de mestrado; departamento de educação. PUC-Rio, Rio de Janeiro, 1992.

LÈVY, P. *A ideografia dinâmica: rumo a uma imaginação artificial*. [S.l.]: Loyola : São Paulo., 1998.

MEIRA, L. Making sense of instructional devices : The emergence of transparency in mathematical activity. *Journal for Research in Mathematics Education*, vol. 29, n. 2., 1998.

MENEZES, C. S. de et al. *Informática Educativa II - Linguagens para Representação do Conhecimento*. Vitória: UFES, 2003. (Fascículo de livro).

MICHAEL, D.; CHEN, S. *Serious Games: Games That Educate, Train, and Inform*. [S.l.]: Thomson Course Technology, 2006.

MILITÃO, A.; MILITÃO, R. *Jogos, Dinâmicas & Vivências Grupais*. [S.l.]: Rio de Janeiro: Qualitymark Editora, 2000.

MOITA, F. M. G. da S. C. Games: locus de aprendizagens colaborativas e novas sociabilidades. In: \_\_\_\_\_. *LIMA JR, A. S.; Hetkowski, T. M. (Orgs.). Educação e contemporaneidade: desafios para a pesquisa e a pós-graduação*. 1. ed. Rio de Janeiro: Quartet, 2006. p. 217–237.

MOITA, F. M. G. da S. C. Games: Contexto cultural e curricular de "saberes de experiências feitas". In: \_\_\_\_\_. *Moita, F. M. G. S. C. e Silva, E. M. e Souza, R. P. (org.) Jogos Eletrônicos: Construindo Novas Trilhas*. 1. ed. Campina Grande: EDUEP, 2007. p. 43–61. Disponível em: <http://www.comunidadesvirtuais.pro.br/seminario2/trabalhos/filomena.pdf>. Acesso em: 27 de março de 2009.

MONKE, L. Why children shouldn't have the world at their fingertips. *Charlotte's Webpage*, 2005. Disponível em: <http://www.orionmagazine.org/index.php/articles/article/159/>. Acesso em: 8 out. 2009.

NETBEANS. 2009. Disponível em: <http://www.netbeans.org/> Acesso em: 13 de junho de 2009.

OPENAL. 2009. Disponível em: <http://connect.creativelabs.com/openal/default.aspx> Acesso em: 13 de junho de 2009.

PASSERINO, L. M. *Avaliação de jogos educativos computadorizados*. [S.l.]: Taller Internacional de Software Educativo (TISE98). Santiago, Chile., 1998.

PIAGET, J. *Seis estudos de psicologia*. [S.l.]: Editora Forense Universitária, 1978.

PRESSMAN, R. S. *Engenharia de Software - (6ª edição)*. São Paulo: McGrawHill, 2006.

RAMALHO, G.; CORRUBLE, V. Introdução sobre o papel da inteligência artificial nos jogos digitais. In: *Moita, F. M. G. S. C. e Silva, E. M. e Souza, R. P. (org.) Jogos Eletrônicos: Construindo Novas Trilhas*. Campina Grande: EDUEP. [S.l.: s.n.], 2007.

Re-Mission. 2009. Disponível em: <http://www.re-mission.net/>. Acesso em: 7 de julho de 2009.

ROLLINGS, A.; ADAMS, E. *Andrew Rollings and Ernest Adams on Game*. [S.l.]: Nova Iorque: New Riders, 2003.

ROLLINGS, A.; MORRIS, D. *Game Architecture and Design: A New Edition*. [S.l.]: New Riders Publishers, 2004.

ROLLINGS, A.; MORRIS., D. *Game Architecture and Design: A New Edition*. [S.l.]: New Riders Publishers, 2004.

SEABRA, C. *Software educacional e telemática: novos recursos para a escola*. [S.l.]: Disponível em: <http://cseabra.wordpress.com/1994/09/01/software-educacional-telematica/>. Acesso em: 27 de outubro de 2009., 1994.

SILVA, D. D.; MARCHELLI, P. S. *Informática e Ensino: visão crítica dos softwares educativos e discussão sobre as bases pedagógicas adequadas ao seu desenvolvimento*. [S.l.]: Grupo de Estudos e Pesquisas Sobre o Emprego de Novas Tecnologias Educacionais - Grupo Gente, UFMT, 2009. Disponível em: [http://www.ufmt.br/ufmtvirtual/textos/se\\_visao.htm](http://www.ufmt.br/ufmtvirtual/textos/se_visao.htm). Acesso em: 25 de agosto de 2009.

STAHL, M. M. *Ambientes e Ensino-Aprendizagem Computadorizados: da sala de aula convencional ao mundo da fantasia*. [S.l.]: Rio de Janeiro: COPPE-UFRJ, 1991.

TAROUÇO, L. M. R. et al. *Jogos educacionais*. [S.l.]: Novas Tecnologias na Educação. CINTED-UFRGS, 2004. Disponível em: <http://www.cinted.ufrgs.br/renote/mar2004/artigos/30-jogoseducacioanis.pdf>. Acesso em: 30 de outubro de 2009.

TAVARES, M. T. D. S. Jogos eletrônicos: Educação e mídia. In: *Moita, F. M. G. S. C. e Silva, E. M. e Souza, R. P. (org.) Jogos Eletrônicos: Construindo Novas Trilhas. Campina Grande: EDUEP*. [S.l.: s.n.], 2007. Disponível em: <http://www.comunidadesvirtuais.pro.br/seminario2/trabalhos/marcostavares.pdf>. Acesso em: 13 de novembro de 2009.

TAVARES, R. Do livro ao game: paralelos entre pornografia e violência. In: *Moita, F. M. G. S. C. e Silva, E. M. e Souza, R. P. (org.) Jogos Eletrônicos: Construindo Novas Trilhas. Campina Grande: EDUEP*. [S.l.: s.n.], 2007.

TEIXEIRA, A. C.; BRANDÃO, E. J. R. *Software educacional: difícil começo*. [S.l.]: Revista Novas Tecnologias na Educação. Vol 1. n 1. CINTED/UFRGS, 2003. Disponível em: [http://www.cinted.ufrgs.br/renote/fev2003/artigos/adriano\\_software.pdf](http://www.cinted.ufrgs.br/renote/fev2003/artigos/adriano_software.pdf). Acesso em: 27 de outubro de 2009.

VALENTE, J. A. Análise dos diferentes tipos de softwares usados na educação. *III Encontro Nacional do PROINFO - MEC, Pirenópolis - GO*, 1998.

VALENTE, J. A. A telepresença na formação de professores da área de informática em educação: Implantando o construcionismo contextualizado. *IV Congresso Congresso Ibero-Americano de Informática na Educação-RIBIE, Brasília*, 1998. Disponível em: <http://lsm.dei.uc.pt/ribie/docfiles/txt200342421644232.PDF>. Acesso em: 27 de março de 2009.

VYGOTSKY, L. S. *A formação social da mente: o desenvolvimento dos processos psicológicos superiores*. [S.l.]: Martins Fontes, 1994.

WIKI, G. P. 2009. Disponível em: [http://gpwiki.org/index.php/Game\\_Engines](http://gpwiki.org/index.php/Game_Engines)  
Acesso em: 13 de junho de 2009.

ZERBST, S.; DUVEL, O. *3D Game Engine Programming*. [S.l.]: Thomson Course Technology, Premier press, 2004.

## ***APÊNDICE A – Código Fonte***

### **A.1 Classe Principal “InGameState”**

*/\*Classe responsavel por inicializar o jogo, carregando texturas  
objetos 3D. Esta classe tambem é responsavel por gerenciar as entradas (comandos do usuario) e as  
saidas.*

```
public class InGameState extends DebugGameState implements InterfaceGameState
```

```
/*Variavel responsável por pausar o jogo*/  
protected boolean pause;
```

```
/*Variavel que armazena o tempo total do o jogo*/  
private final int TEMP_MAX_JOGO =300;
```

```
/* Variavel que conta o tempo total do o jogo */  
private float tempoDoJogo=0;
```

```
/*Variaveis que mostram o tempo corrente do jogo*/  
/*Cada variavel representa um digito do tempo*/  
private Quad tempoJogoH,tempoJogoM,tempoJogoS;
```

```
/*Varivel que guarda as texturas de cada numero. */  
private TextureState[] texture0123456789 = new TextureState[10];
```

```
/*Guarda os pontos atuais do jogo. */  
private String guardarPontos=”0”;
```

```
/*Variavel que gerencia a simulação da fisica do jogo */  
private PhysicsSpace physicsSpace;
```

```
/* A raiz da arvore do jogo */  
protected Node rootNode;
```

```
/* Nodo que contém o semaforo */
```



```

protected Node semaforo;

/* Nodo que contém todas as caixas e moedas como nodos filhos */
private Node nodoCaixa,nodoMoeda;

/* Carro do jogo */
protected Car car;

/* Nodo que contém respectivamente todas as caixas coloridas e brancas */
Node [] caixaColorida,caixaBranca;

/*Atualiza o jogo de acordo com a interação do jogador*/
@Override
public void update(float interpolation) {
    //Chama a classe herdada da ferramenta jME
    super.update(interpolation);
    //Utilizado quando é chamado as classes para Mostrar o mapa ou informação da caixa
    if (!pause) {
        //Tempo do jogo diminui a cada iteração
        tempoDoJogo = tempoDoJogo - interpolation;
        //Atualiza o tempo mostrado na tela Cada variavel representa um digito do tempo mostrado
        tempoJogoH.setRenderState(texture0123456789[(int) tempoDoJogo / 60]);
        tempoJogoH.updateRenderState();
        tempoJogoM.setRenderState(texture0123456789[((int) tempoDoJogo % 60) / 10]);
        tempoJogoM.updateRenderState();
        tempoJogoS.setRenderState(texture0123456789[((int) tempoDoJogo % 60) % 10]);
        tempoJogoS.updateRenderState();
        //Se o tempo acabar uma nova classe é chamada
        if (tempoDoJogo <= 0) {
            ClassFactory.getInPontuacaoGameState().setMostrarPontuacaoJogo(guardarPontos);
            GameManager.getInstance().changeState(ClassFactory.getInPontuacaoGameState());
        }
        //Atualiza toda a parte do jogo que simula a fisica
        physicsSpace.update(interpolation * physicsSpeed);
        //O audio referente a ganhar pontos é atualizado
        AudioSystem.getSystem().update();
        //Toda a estrutura do jogo é atualizada a partir do nodo raiz da árvore
        rootNode.updateGeometricState(interpolation, true);
        //Avalia se a tecla M foi apertada para poder mostrar o mapa do jogo
        if (KeyBindingManager.getKeyBindingManager().isValidCommand("mapa", true)) {
            GameManager.getInstance().pushState(ClassFactory.getInPauseGameState());
        }
        //Determina se alguma caixa teve colisão com o carro
        if(nodoCaixa.hasAncestor(rootNode) && nodoCaixa.hasCollision(car, false)){

```

```

//Encontra qual caixa colorida teve colizão com o carro
for(i = 0 ; i< caixaColorida.length; i++){
    if (caixaColorida[i].hasAncestor(nodoCaixa) && caixaColorida[i].hasCollision(car, false)) {
        //Retira da arvore e consequentemente do jogo
        //a caixa colidida com o carro
        caixaColorida[i].removeFromParent();
        //Chama a classe responsavel por mostrar as informações sobre as caixas
        ClassFactory.getBoxGameState().setInformacaoCaixaColorida(caixaColorida[i]);
        GameManager.getInstance().pushState(ClassFactory.getBoxGameState());
        //Retorna depois de um tempo a caixa colorida ao jogo
        caixaControle = new Node();
        nodoCaixa.attachChild(caixaControle);
        caixaControle.addController(new Controller() {
            private static final long serialVersionUID = 1L;
            private float tempo = 0;
            public Node nodo = caixaColorida[i];
            public Node controle = caixaControle;
            @Override
            public void update(float tpf) {
                tempo += tpf;
                if ((int) tempo == 3) {
                    tempo = 0;
                    nodoCaixa.attachChild(nodo);
                    controle.removeFromParent();
                }
            }
        }
    }
    break;
}
} //Fim for
//Encontra qual caixa branca teve colizão com o carro
for(i = 0 ; i< caixaBranca.length; i++){
    if(caixaBranca[i].hasAncestor(nodoCaixa) && caixaBranca[i].hasCollision(car, false))
        //Chama a classe responsavel por mostrar as informações sobre as caixas
        ClassFactory.getBoxGameState().setInformacaoCaixaBranca(caixaBranca[i]);
        GameManager.getInstance().pushState(ClassFactory.getBoxGameState());
    }
} //Fim for
} //Fim if
updateMoedas();
updateSemaforo(interpolation);
}
}

```

```

/*Retira a moeda da arvore (do jogo) caso colisão com o carro
inicia o som e atualiza os pontos
/
protected void updateMoedas() {
    //Determina se alguma moeda teve colisão com o carro
    if (nodoMoeda.hasAncestor(rootNode) && nodoMoeda.hasCollision(car, false))
        //Encontra qual moeda teve colisão com o carro
        for (int i = 0; i < nodoMoeda.getQuantity(); i++) {
            if (nodoMoeda.getChild(i).hasCollision(car, false)) {
                //Atualiza a pontuação mostrada na tela
                guardarPontos(20, true);
                //Utilizado para reiniciar as moedas caso o jogo seja iniciado
                //pela segunda vez, evitando assim que todo o jogo seja carregado
                //novamente
                resetarJogoMoedas.attachChild(nodoMoeda.detachChildAt(i));
                break;
            }
        }
    }
}

/*O semaforo é atualizado a fim de mostrar uma mensagem caso
detecte que o carro passo por um sinal vermelho
/
protected void updateSemaforo(float interpolation) {
    //Encontra se o semaforo teve colisão com o carro
    if (semaforo.getChild(4).hasCollision(car.getChassis().getChild(0), false)) {
        //Mostra uma mensagem alertando sobre o sinal vermelho
        if (!temMensagemSemaforoVermelho) {
            temMensagemSemaforoVermelho = true;
            timeMensagemSemaforoVermelho = interpolation;
            semaforoHUD.setCullHint(Spatial.CullHint.Never);
        }
    }
    //Retira a mensagem da tela depois de 3 segundos
    if (temMensagemSemaforoVermelho) {
        timeMensagemSemaforoVermelho += interpolation;
        if ((int) timeMensagemSemaforoVermelho == 3) {
            timeMensagemSemaforoVermelho = 0;
            temMensagemSemaforoVermelho = false;
            semaforoHUD.setCullHint(Spatial.CullHint.Always);
        }
    }
}
}

```

```

/*Guarda e atualiza a pontuação na tela*/
public void guardarPontos(int pontos, boolean isGuardar){
    //Quando o jogo é iniciado pela segunda vez, os pontos
    //são zerados não necessitando de som
    if(isGuardar){
        //Inicia o som de que ganhou pontos
        moedaSom.setWorldPosition(_cam.getLocation());
        moedaSom.stop();
        moedaSom.play();
    }
    //A pontuação do jogo está guardada em uma String, então
    //esta é passada para um int para poder atualizar a pontuação
    //e mostra-la na tela
    pontosInteirosAux = Integer.parseInt(guardarPontos) + pontos;
    if(pontosInteirosAux > 0)
        guardarPontos = "0";
    else
        guardarPontos = pontosInteirosAux + ;
    //Mostra a pontuação na tela do jogo, onde cada
    //posição da variável pontUDCM[] representa um dígito
    //da pontuação mostrada
    for (pontosInteirosAux = guardarPontos.length() - 1; pontosInteirosAux >= 0; pontosInteirosAux--) {
        switch (Integer.parseInt(guardarPontos.charAt(pontosInteirosAux) + )) {
            case 0:
                pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].setRenderState(texture0123456789[0]);
                pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].updateRenderState();
                break;
            case 1:
                pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].setRenderState(texture0123456789[1]);
                pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].updateRenderState();
                break;
            case 2:
                pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].setRenderState(texture0123456789[2]);
                pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].updateRenderState();
                break;
            case 3:
                pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].setRenderState(texture0123456789[3]);
                pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].updateRenderState();
                break;
            case 4:
                pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].setRenderState(texture0123456789[4]);
                pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].updateRenderState();
                break;

```

```

case 5:
    pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].setRenderState(texture0123456789[5]);
    pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].updateRenderState();
    break;
case 6:
    pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].setRenderState(texture0123456789[6]);
    pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].updateRenderState();
    break;
case 7:
    pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].setRenderState(texture0123456789[7]);
    pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].updateRenderState();
    break;
case 8:
    pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].setRenderState(texture0123456789[8]);
    pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].updateRenderState();
    break;
case 9:
    pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].setRenderState(texture0123456789[9]);
    pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].updateRenderState();
    break;
default:
    pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].setRenderState(texture0123456789[0]);
    pontUDCM[guardarPontos.length() - 1 - pontosInteirosAux].updateRenderState();
    break;
    }
}
}

/*
Chamado pela classe BoxGameState. Utilizado como o retorno de
uma informação de uma caixa Branca. Seu objetivo é o de atualizar
a combinação ou pontuação, isso de acordo com a interação do usuario
/
protected void setRetornoCaixaBranca(boolean isCancel, Node caixaBranca) {
    //O usuario cancelou uma possivel combinação com a caixa branca
    if (isCancel) {
        retireTemporariamenteCaixaBranca(caixaBranca);
    }
    else {
        //Identifica se é o primeiro item selecionado da combinação com tres itens
        if (combinacaoJogador[0].isEmpty()) {
            //Retira a caixa branca e a caixa colorida e os pontos do mapa
            // relacionados a ela
            caixaBranca.removeFromParent();
            ((NodoCaixa) caixaBranca.getChild(2)).getCaixaColorida().removeFromParent();

```

```

((NodoCaixa) caixaBranca.getChild(2)).getPontoBrancoMapa().removeFromParent();
((NodoCaixa) caixaBranca.getChild(2)).getPontoColMapa().removeFromParent();
armazenaCaixaBranca.attachChild(caixaBranca);
//Armazena o primeiro item da combinaçao do jogador
combinacaoJogador[0] = ((NodoCaixa) ((NodoCaixa) caixaBranca.getChild(2)).getCaixaColorida().getChild(2)).g
//Atualiza na tela o primeiro item da combinação do jogador
if (combinacaoJogador[0].equals(ClassFactory.getCorCaixaVerde())) {
    combinacaoSelecionada123[0].setRenderState(textureCombSelectWGYR[1]);
} else if (combinacaoJogador[0].equals(ClassFactory.getCorCaixaAzul())) {
    combinacaoSelecionada123[0].setRenderState(textureCombSelectWGYR[2]);
} else if (combinacaoJogador[0].equals(ClassFactory.getCorCaixaVermelha())) {
    combinacaoSelecionada123[0].setRenderState(textureCombSelectWGYR[3]);
}
combinacaoSelecionada123[0].updateRenderState();
//Identifica qual das tres possiveis combinações que o jogador
// pode fazer
if (combinacaoPossivel[0].equals(combinacaoJogador[0])) {
    combinacaoPossivelJogador = combinacaoPossivel;
} else if (combinacaoPossivel2[0].equals(combinacaoJogador[0])) {
    combinacaoPossivelJogador = combinacaoPossivel2;
} else if (combinacaoPossivel3[0].equals(combinacaoJogador[0])) {
    combinacaoPossivelJogador = combinacaoPossivel3;
}
//Identifica se é o segundo item selecionado da combinação com tres itens
} else if (combinacaoJogador[1].isEmpty()) {
    //Armazena o segundo item da combinação do jogador
    combinacaoJogador[1] = ((NodoCaixa) ((NodoCaixa) caixaBranca.getChild(2)).getCaixaColorida().getChild(2)).g
    //Verifica se o jogador acertou o segundo item da combinação
    if (combinacaoJogador[1].equals(combinacaoPossivelJogador[1])) {
        //Retira a caixa branca e a caixa colorida e os pontos do mapa relacionados a ela
        caixaBranca.removeFromParent();
        ((NodoCaixa) caixaBranca.getChild(2)).getCaixaColorida().removeFromParent();
        ((NodoCaixa) caixaBranca.getChild(2)).getPontoBrancoMapa().removeFromParent();
        ((NodoCaixa) caixaBranca.getChild(2)).getPontoColMapa().removeFromParent();
        armazenaCaixaBranca.attachChild(caixaBranca);
        //Atualiza na tela o segundo item da combinação do jogador
        if (combinacaoJogador[1].equals(ClassFactory.getCorCaixaVerde())) {
            combinacaoSelecionada123[1].setRenderState(textureCombSelectWGYR[1]);
        } else if (combinacaoJogador[1].equals(ClassFactory.getCorCaixaAzul())) {
            combinacaoSelecionada123[1].setRenderState(textureCombSelectWGYR[2]);
        } else if (combinacaoJogador[1].equals(ClassFactory.getCorCaixaVermelha())) {
            combinacaoSelecionada123[1].setRenderState(textureCombSelectWGYR[3]);
        }
        combinacaoSelecionada123[1].updateRenderState();
    }
}

```

```

//Errou a segunda parte da Combinacao
} else {
    retireTemporariamenteCaixaBranca(caixaBranca);
    //Retorna ao jogo a caixa branca, a caixa colorida e os pontos do mapa relacionados a ela
    Node nod = (Node) armazenaCaixaBranca.detachChildAt(0);
    nodoCaixa.attachChild(nod);
    nodoCaixa.attachChild(((NodoCaixa) nod.getChild(2)).getCaixaColorida());
    nodoPontoBranco.attachChild(((NodoCaixa) nod.getChild(2)).getPontoBrancoMapa());
    nodoPontoCol.attachChild(((NodoCaixa) nod.getChild(2)).getPontoColMapa());
    //Mostra na tela a informação de que o jogador errou a combinação
    ClassFactory.getBoxGameState().setErroDeCombinacao(true);
    GameManager.getInstance().pushState(ClassFactory.getBoxGameState());
    //Atualiza na tela que nenhuma combinação foi iniciada
    combinacaoSelecionada123[0].setRenderState(textureCombSelectWGYR[0]);
    combinacaoSelecionada123[0].updateRenderState();
    //Retira 30 pontos do jogador
    guardarPontos(-30, false);
    //Reinicia a combinação do jogador
    combinacaoJogador[0] = "";
    combinacaoJogador[1] = "";
}
//Terceiro item selecionado pelo jogador da combinação com tres itens
} else {
    //Armazena o terceiro item da combinação do jogador
    combinacaoJogador[2] = ((NodoCaixa) ((NodoCaixa) caixaBranca.getChild(2)).getCaixaColorida().getChild(2)).g
    //o jogador acertou o terceiro item da combinação
    if (combinacaoJogador[2].equals(combinacaoPossivelJogador[2])) {
        //Atualiza na tela que nenhuma combinação foi iniciada
        combinacaoSelecionada123[0].setRenderState(textureCombSelectWGYR[0]);
        combinacaoSelecionada123[1].setRenderState(textureCombSelectWGYR[0]);
        combinacaoSelecionada123[0].updateRenderState();
        combinacaoSelecionada123[1].updateRenderState();
        //Adiciona 100 pontos ao jogador
        guardarPontos(100, true);
        //Retira a caixa branca e a caixa colorida e os pontos do mapa relacionados a ela
        caixaBranca.removeFromParent();
        ((NodoCaixa) caixaBranca.getChild(2)).getCaixaColorida().removeFromParent();
        ((NodoCaixa) caixaBranca.getChild(2)).getPontoBrancoMapa().removeFromParent();
        ((NodoCaixa) caixaBranca.getChild(2)).getPontoColMapa().removeFromParent();
        armazenaCaixaBranca.attachChild(caixaBranca);
        //Guardas as informações retiradas do jogo para serem recuperadas ao reiniciar o jogo
        //pela segunda vez
        resetarJogoCaixaBranca.attachChild(armazenaCaixaBranca.detachChildAt(0));
        resetarJogoCaixaBranca.attachChild(armazenaCaixaBranca.detachChildAt(0));
    }
}

```

```

    resetarJogoCaixaBranca.attachChild(caixaBranca);
    //Reinicia a combinação do jogador
    combinacaoJogador[0] = "";
    combinacaoJogador[1] = "";
    combinacaoJogador[2] = "";
    //Errou a terceira parte da Combinacao
} else {
    //Atualiza na tela que nenhuma combinação foi iniciada
    combinacaoSelecionada123[0].setRenderState(textureCombSelectWGYR[0]);
    combinacaoSelecionada123[1].setRenderState(textureCombSelectWGYR[0]);
    combinacaoSelecionada123[0].updateRenderState();
    combinacaoSelecionada123[1].updateRenderState();
    //Retira 30 pontos do jogador
    guardarPontos(-30, false);
    //Retorna ao jogo as caixas brancas, coloridas e os pontos do mapa
    // relacionados
    retireTemporariamenteCaixaBranca(caixaBranca);
    Node nod = (Node) armazenCaixaBranca.detachChildAt(1);
    nodoCaixa.attachChild(nod);
    nodoCaixa.attachChild(((NodoCaixa) nod.getChild(2)).getCaixaColorida());
    nodoPontoBranco.attachChild(((NodoCaixa) nod.getChild(2)).getPontoBrancoMapa());
    nodoPontoCol.attachChild(((NodoCaixa) nod.getChild(2)).getPontoColMapa());
    nod = (Node) armazenCaixaBranca.detachChildAt(0);
    nodoCaixa.attachChild(nod);
    nodoCaixa.attachChild(((NodoCaixa) nod.getChild(2)).getCaixaColorida());
    nodoPontoBranco.attachChild(((NodoCaixa) nod.getChild(2)).getPontoBrancoMapa());
    nodoPontoCol.attachChild(((NodoCaixa) nod.getChild(2)).getPontoColMapa());
    //Mostra na tela a informação de que o jogador errou a combinação
    ClassFactory.getBoxGameState().setErroDeCombinacao(true);
    GameManager.getInstance().pushState(ClassFactory.getBoxGameState());
    //Reinicia a combinação do jogador
    combinacaoJogador[0] = "";
    combinacaoJogador[1] = "";
    combinacaoJogador[2] = "";
}
}
}

/*Utilizado quando o jogo é inicializado pela segunda vez
evitando assim todo o carregamento do jogo novamente*/
public void resetarJogo(){
    //Retorna o carro a posição inicial
    car.setPosition( 0, 50, 0 );

```



```

//Zera a pontuação do jogo, o -1000 e utilizado pois a pontuação
// não pode ser menor do que zero, se for é atribuido a ela zero
guardarPontos(-1000,false);
//Reinicia o tempo
tempoDoJogo = TEMP_MAX_JOGO;
//Auxilia a retornar as caixas ao jogo
Node cBranca;
//Retorna ao jogo as caixas brancas, coloridas e os
//pontos do mapa retirados no jogo anterior
while (resetarJogoCaixaBranca.getQuantity() != 0) {
    cBranca = (Node) resetarJogoCaixaBranca.detachChildAt(0);
    nodoCaixa.attachChild(cBranca);
    nodoCaixa.attachChild(((NodoCaixa) cBranca.getChild(2)).getCaixaColorida());
    nodoPontoCol.attachChild(((NodoCaixa) cBranca.getChild(2)).getPontoColMapa());
    nodoPontoBranco.attachChild(((NodoCaixa) cBranca.getChild(2)).getPontoBrancoMapa());
}
//Retorna as moedas ao jogo
while(resetarJogoMoedas.getQuantity() != 0){
    nodoMoeda.attachChild(resetarJogoMoedas.detachChildAt(0));
}
//Limpa da tela as possiveis combinações iniciadas no jogo anterior
combinacaoSelecionada123[0].setRenderState(textureCombSelectWGYR[0]);
combinacaoSelecionada123[1].setRenderState(textureCombSelectWGYR[0]);
}

}

```

## *APÊNDICE B – Informações utilizadas no jogo*

### **B.1 Informações sobre multas de trânsito**

Foram utilizadas nove multas <sup>1</sup> para o desenvolvimento do jogo. São elas:

<b>Multa</b>	<b>Infração</b>	<b>Penalidade</b>	<b>Pontos na Carteira</b>	<b>Medida administrativa</b>
Conduzir veículo sem os documentos de porte obrigatório: carteira de habilitação e documentos do veículo .	Leve	Multa de 53,20	3	Retenção do veículo até a apresentação do documento
Ter seu veículo imobilizado na via por falta de combustível.	Média	Multa de 85,13	4	Remoção do veículo
Deixar o condutor ou passageiro de usar o cinto de segurança.	Grave	Multa de 127,69	5	Retenção do veículo até colocação do cinto pelo infrator.
Parar sobre faixa destinada a pedestres.	Leve	Multa de 53,20	3	
Conduzir o veículo com defeito no sistema de iluminação, de sinalização ou com lâmpada queimadas.	Média	Multa de 85,13	4	
Transitar pela contramão de direção.	Grave	Multa de 127,69	5	

<sup>1</sup>Todas as informações foram retiradas dos sites <http://detranet.mg.gov.br/detran/tbinfr.asp>; <http://www.manualdepericias.com.br/infracoestransito.asp>

<b>Multa</b>	<b>Infração</b>	<b>Penalidade</b>	<b>Pontos na Carteira</b>	<b>Medida administrativa</b>
Deixar de atualizar o cadastro de registro do veículo ou de habilitação do condutor.	Leve	Multa de 53,20	3	
Dirigir o veículo utilizando-se de fones nos ouvidos conectados a aparelhagem sonora ou de telefone celular.	Média	Multa de 85,13	4	
Usar no veículo equipamento com som em volume ou frequência que não sejam autorizados pelo CONTRAN.	Grave	Multa de 127,69	5	Retenção do veículo para regularização.