



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

THIAGO FRANCISCO FERREIRA

**DESENVOLVIMENTO DE UM AMBIENTE VIRTUAL 3D PARA ENSINO A
DISTÂNCIA**

ASSIS
2008



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

DESENVOLVIMENTO DE UM AMBIENTE VIRTUAL 3D PARA ENSINO A DISTÂNCIA

THIAGO FRANCISCO FERREIRA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientadora: Profa. Dra. Marisa Atsuko Nitto

Analisador (1): Luiz Ricardo Begosso

Analisador (2): Alexandre Charles Cassiano

ASSIS
2008



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

THIAGO FRANCISCO FERREIRA

DESENVOLVIMENTO DE UM AMBIENTE VIRTUAL 3D PARA ENSINO A DISTÂNCIA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientadora: Profa. Dra. Marisa Atsuko Nitto

Área de Concentração: Realidade Virtual

ASSIS
2008

"Não se mede o valor de um homem pelas suas roupas ou pelos bens que possui. O verdadeiro valor de um homem é o seu caráter, suas idéias e a nobreza dos seus ideais".

Charles Spencer Chaplin

DEDICATÓRIA

Dedico este trabalho a Deus, por ter me dado o dom da vida e por me permitir viver nesta época, junto com pessoas maravilhosas, amigos da faculdade, trabalho, familiares, pessoas que sempre ajudam a toda hora, sempre dando a força necessária para lutar e vencer!

AGRADECIMENTOS

A Profa. Dra. Marisa Atsuko Nitto pela orientação, pela dedicação e pelo estímulo transmitido durante as horas boas e ruins no decorrer deste trabalho, e por não ter me deixado desistir no meio do caminho.

Aos amigos Lucas, Ângelo, Pedro, Tiago, Renan e Mateus que me acompanharam nesta jornada.

Á todos os amigos da classe, em especial Thiago, Bruno, Kleber, Danilo e Andréia.

Aos amigos de trabalho Sara, Eduardo, Ederson, Fernanda, André, Juliano e Ana Carolina.

Agradeço também ao Ênio, pela importante colaboração no auxílio para encontrar a ferramenta adequada para desenvolver este trabalho.

Aos familiares, principalmente a minha mãe Eliana pela orientação na escolha do curso, e pelo apoio durante o curso, e a todos que colaboraram, direta ou indiretamente na execução deste trabalho.

RESUMO

Este trabalho tem por objetivo a pesquisa e desenvolvimento de uma ferramenta computacional, que simule um ambiente de ensino a distância utilizando as técnicas de Realidade Virtual em 3D. Esta Realidade Virtual é composta por um cenário, mais especificamente, uma sala de aula, além de objetos, tais como móveis, lousa, iluminação, além de um personagem.

O personagem (*avatar*) pode ser controlado pelo usuário, podendo interagir com alguns objetos, como por exemplo, ao se aproximar de uma porta, abri-la automaticamente ao pressionar uma tecla, ou até mesmo acionar uma vídeo-aula na lousa, que se torna um “telão”. O intuito disso é mostrar um protótipo modelo de uma ferramenta que pode auxiliar no ensino a distância.

Todo o desenvolvimento prático deste trabalho foi feito utilizando a *engine* 3D Game Studio. Esta poderosa ferramenta é muito utilizada no desenvolvimento de jogos, e foi crucial no desenvolvimento dos cenários em 3D.

Palavras-chave: Realidade Virtual. Ensino a Distância. 3D Game Studio.

ABSTRACT

The purpose of this work is the research and development of a computational tool that simulates an environment of the distance education using the techniques of 3D virtual reality. This virtual reality is a scenario, more specifically, a classroom, as well as objects such as furniture, blackboard, lighting, in addition to the characters.

The character (*avatar*) can be controlled by the user and may interact with certain objects, such as when approaching a door, open it automatically by pressing a button, or even trigger a video lesson on the board, which is makes a "painting". The purpose of this is showing a prototype model of a tool that can assist in distance learning. All the practical development of this work was done using the 3D Game Studio *engine*. This powerful tool is widely used in the development of games, and was crucial in the development of the scenarios in 3D.

Keywords: Virtual Reality. Distance Education. 3D Game Studio.

LISTA DE ILUSTRAÇÕES

Figura 1 - Modelo com concepção de cursos em equipe	20
Figura 2 - Plataforma de comunicação EAD	21
Figura 3 – Modelo de Mundo Virtual	22
Figura 4 - RV Imersiva	25
Figura 5 – Simulador de cockpits	26
Figura 6 - O simulador de vôo da Boeing 777	26
Figura 7 - Trator virtual baseado em Projeção de Tela	27
Figura 8 - Visor baseado em Projeção de Tela	27
Figura 9 - Realidade Realçada.....	28
Figura 10 - Realidade Realçada.....	28
Figura 11 - Esquema de luva de dados baseada em sensores de fibra ótica.	29
Figura 12 - Exemplo de sistema de feedback de força para mão e braço	30
Figura 13 - Protótipo de um telerobô utilizado em explorações espaciais.....	31
Figura 14 - Telerobô "Pathfinder" da NASA criado para explorar Marte em 1999.....	32
Figura 15 - HMD.....	33
Figura 16 - BOOM	33
Figura 17 - Sistema de RV de Mesa	33
Figura 18 - Ponto de vista do usuário dentro do mundo gorila.....	36
Figura 19 - Mundo de Newton.....	36
Figura 20 - Exemplo de criação de um cenário no WED.....	39
Figura 21 - Exemplo de criação de modelos.	40
Figura 22 - Modelo de Sala Virtual.....	42
Figura 23 - Relacionamentos do Sistema.	44
Figura 24 - Propriedades do mapa.....	52

Figura 25 - Adicionando bloco.....	53
Figura 26 - Bloco oco.	54
Figura 27 – Texturas.	55
Figura 28 - Subtraindo Objeto.	56
Figura 29 - Adicionando luzes.	56
Figura 30 - Adicionando Sprite.....	57
Figura 31 - Adicionando modelos.....	58
Figura 32 - Adicionando prefabs.	59
Figura 33 - Imagem de um <i>Sky-cube</i>	60
Figura 34 - Propriedades do Objeto.	60
Figura 35 - Acessando Script pelo WED.....	61
Figura 36 - Cenário com texturas.....	63
Figura 37 - Adicionando portas e janelas.....	64
Figura 38 - Ambiente Virtual interno.....	65
Figura 39 - Ambiente Virtual externo.....	66
Figura 40 - Ambiente Virtual com <i>avatar</i>	67
Figura 41 - <i>Sky-cube</i> utilizado.....	67
Figura 42 - <i>Sky-cube</i> no ambiente virtual.....	68
Figura 43 - Adicionando script.....	72
Figura 44 - Associando script ao personagem.	73
Figura 45 - Adicionando script de Câmera Livre.	75
Figura 46 - Ação porta.....	78
Figura 47 - Criação de uma lousa virtual.....	79

LISTA DE ABREVIATURAS E SIGLAS

EAD	Ensino a Distância ou Educação a Distância
RV	Realidade Virtual
NPC	<i>Non Player Character</i>
3DSG	<i>3D Studio Game</i>
MED	<i>Model Editor</i>
WED	<i>World Editor</i>
SED	<i>Script Editor</i>

SUMÁRIO

1. INTRODUÇÃO	15
1.1. OBJETIVO.....	15
1.2. MOTIVAÇÃO.....	16
1.3. ESTRUTURA DO TRABALHO	16
2. CONCEITOS BÁSICOS E TEÓRICOS.....	17
2.1. ENSINO Á DISTÂNCIA.....	17
2.1.1. Modelo inicial de comunicação: via de mão-dupla	19
2.2. REALIDADE VIRTUAL.....	22
2.2.1. Modelo de um mundo virtual.....	22
2.2.1.1. Modelagem geométrica	23
2.2.1.2. Restrições cinemáticas.....	23
2.2.1.3. Modelagem física	23
2.2.1.4. Comportamento do objeto	24
2.2.2. Realidade Virtual Imersiva e Não-Imersiva	24
2.2.3. Aplicabilidades da Realidade Virtual.....	25
2.2.3.1. Realidade Virtual de simulação	26
2.2.3.2. Realidade Virtual de Projeção	26
2.2.3.3. Augmented Reality (Realidade Realçada).....	27
2.2.3.4. Dispositivos Físicos	28
2.2.3.5. Telepresença.....	30
2.2.3.6. Displays visualmente casados.....	32
2.2.3.7. RV de Mesa (Desktop RV)	33
2.2.4. Realidade Virtual na Educação	34
2.3. 3D STUDIO GAME.....	38
2.3.1. WED (World Editor)	39
2.3.2. MED (Model Editor)	40

2.3.3. SED (Script Editor)	41
3. MODELAGEM DO PROBLEMA	42
3.1. DESCRIÇÃO DO PROBLEMA	42
3.2. MODELAGEM DO PROBLEMA	44
3.3. DESCRIÇÃO GERAL DOS RELACIONAMENTOS.....	44
3.3.1. Cenário	44
3.3.2. Objetos	45
3.3.3. Personagens	45
3.3.4. Avatares	45
3.3.5. Permissões	46
3.3.6. Comunicação	46
3.3.7. Áudio	47
3.3.8. Texto.....	47
3.3.9. Lousa virtual	47
3.4. CASOS DE INTERAÇÕES	47
3.4.1. Interação Professor – Alunos.....	48
3.4.2. Interação Alunos – Professor.....	49
3.4.3. Interação Alunos – Alunos	50
4. DESENVOLVIMENTO DO AMBIENTE VIRTUAL 3D	51
4.1. COMANDO BÁSICOS ÚTEIS DO 3D STUDIO GAME.....	51
4.1.1. Configurações do sistema.....	51
4.1.2. Adicionado objeto	52
4.1.3. Bloco oco	53
4.1.4. Texturas	54
4.1.5. Subtração de objetos	55
4.1.6. Iluminação.....	56
4.1.7. Sprites	57

4.1.8. Modelos	58
4.1.9. Prefabs	58
4.1.10. Sky-cube	59
4.2. DESENVOLVIMENTO DO AMBIENTE VIRTUAL 3D	61
4.2.1. Ambiente Virtual (cenário)	62
4.3. SCRIPTS	68
4.3.1. Movimentos e colisões	72
4.3.2. Câmera em primeira pessoa	73
4.3.3. Câmera em terceira pessoa	74
4.3.4. Câmera livre	74
4.3.5. Abrir porta	75
4.4. CRIAÇÃO DE UMA LOUSA VIRTUAL	78
5. CONCLUSÃO	81
REFERÊNCIAS BIBLIOGRÁFICAS	83

CAPÍTULO 1

INTRODUÇÃO

Atualmente, o Ensino a Distância (EAD) vem ganhando mais ênfase, à medida que seus métodos e ferramentas evoluem. O que começou na antiguidade, quando a Grécia e Roma passaram a transmitir conhecimento à longa distância via correspondência, hoje continua com o auxílio de ferramentas modernas, como computador e a internet. As ferramentas mais conhecidas para EAD hoje em dia são o *Moodle*, *AdaptWeb*, *TelEduc*, *Amadeus Lms*, além dos softwares de ambiente virtual de aprendizagem. Todas elas são ferramentas que auxiliam o EAD, incluindo em suas funcionalidades: agenda, mural de avisos, bate-papo (chat), correio eletrônico, material de apoio, funções administrativas e de configuração, entre outros. As técnicas de Realidade Virtual (RV) também têm sido utilizadas para o desenvolvimento de ambiente virtual com a finalidade de facilitar o ensino a Distância. Existem no mercado muitas ferramentas computacionais para o desenvolvimento de ambientes 3D, os mais conhecidos são *Blender*, *Milkshap*, *Maya*, *Ogre*, *3D Game Studio*, etc.

1.1. OBJETIVO

O objetivo principal é o desenvolvimento e pesquisa de uma nova ferramenta de apoio ao EAD, através da simulação de um cenário que constitui o ambiente virtual em 3D. Este cenário é composto por uma sala de aula com objetos, tais como, móveis, lousa, iluminação, além de um personagem. A proposta deste trabalho é desenvolver uma nova maneira de fazer EAD, mostrando um ambiente atrativo e dinâmico para esta realização.

1.2. MOTIVAÇÃO

A principal motivação para o desenvolvimento deste trabalho deve-se ao fato de criar uma nova perspectiva para o Ensino a Distância utilizando os conceitos de Realidade Virtual. A existência de ferramentas computacionais cada vez mais poderosas para a construção de cenários complexos tem impulsionado o desenvolvimento de pesquisa nesta área. Sem estas ferramentas, levar-se-ia anos para construir um simples cenário, além de milhares de linhas de códigos que deveriam ser implementadas. Outra motivação é a contribuição pioneira que este trabalho pode oferecer para o desenvolvimento do Ensino a Distância.

1.3. ESTRUTURA DO TRABALHO

O capítulo 1 aborda a introdução do trabalho, englobando o objetivo do trabalho, a motivação de levar o tema adiante e a estrutura do trabalho.

O capítulo 2 trata dos conceitos básicos e teóricos sobre o Ensino a Distância, Realidade Virtual, e a *engine 3D Studio Game* que foi utilizada para o desenvolvimento do projeto.

Já o capítulo 3 trata da modelagem do problema, envolvendo a descrição do problema em si, a descrição geral dos relacionamentos e os casos de interação entre os atores.

O capítulo 4 está mais relacionado ao desenvolvimento do ambiente virtual, citando primeiramente os comandos e funcionalidades básicas do 3DSG, seguido da seqüência de desenvolvimento do cenário, além dos scripts de funções e ações adicionados aos objetos do cenário, finalizando com uma breve explicação sobre a lousa virtual.

O capítulo 5 aborda a conclusão do trabalho, mostrando uma clara visão dos resultados obtidos e alguns possíveis trabalhos futuros.

Finalizando são apresentadas as referências bibliográficas utilizadas no trabalho.

CAPÍTULO 2

CONCEITOS BÁSICOS TEÓRICOS

Neste capítulo, será abordada a fundamentação teórica dos conceitos utilizados no desenvolvimento do trabalho. Será feita uma descrição resumida sobre ensino a distância, Realidade Virtual, e conceitos técnicos sobre o 3D *Studio Game*.

2.1. ENSINO À DISTÂNCIA

O ensino a distância, ou educação a distância é um processo de ensino onde professores ou alunos estão espacialmente ou temporalmente separados. Este método é possível graças ao uso de tecnologias modernas.

É um sistema de aprendizagem, onde professores e alunos, embora não estejam fisicamente próximos, podem estar conectados, interligados por tecnologias, normalmente através da internet. Além desta, podem ser usados outros meios de auxílio à EAD, tais como correio, rádio, televisão, vídeo, CD-ROM, telefone, fax, e tecnologias semelhantes.

O método de EAD tem sua ênfase voltada para o professor, que é supostamente o “gerente” de todo o processo.

Segundo Gava (2002), além da EAD, temos mais dois métodos de aprendizagem:

- Educação presencial: é o ensino convencional, onde professores e alunos se encontram sempre em um mesmo local físico, chamado sala de aula.
- Semi-presencial: Esta é a união da educação presencial, juntamente com outra parte a distância através de tecnologias.

O EAD pode ou não ter momentos presenciais, mas está focalizado na situação onde professores e alunos estão separados pelo tempo ou espaço, podendo estar juntos através de tecnologias de comunicação.

O EAD pode ser aplicado nos mesmos níveis que o ensino regular, podendo abranger o ensino fundamental, médio, superior e até mesmo a pós-graduação.

Este método é mais direcionado para adultos, que já tenham experiência consolidada de aprendizagem individual e pesquisa.

Com o avanço das tecnologias de comunicação virtual, como internet com redes de alta velocidade, videoconferência e telecomunicação, o conceito de EAD começa a mudar de imagem no sentido de ensino presencial. De acordo com Moran (2002), no futuro será possível ter professores externos compartilhando determinadas aulas, ou um professor “externo” compartilhando com voz e imagem na aula de outro professor, ou até mesmo um sistema mais imersivo baseado em Realidade Virtual, onde todos os participantes tenham a sensação de estar participando de uma aula presencial, mesmo estando separados fisicamente. Será possível assim um intercâmbio maior de conhecimentos, tornando cada vez mais dinâmico e “confortável” os sistemas de EAD.

Com isso, o conceito de aula e curso também muda. Hoje, entendemos aula como um espaço e tempo determinado, e com os avanços tecnológicos, esse tempo e espaço tornam-se cada vez mais flexíveis. As aulas continuarão existindo, mas com uma independência de tempo e espaço, enriquecendo o processo de aprendizado e aproveitando de melhor forma o poder das tecnologias de comunicação e interação, tornando mais prático as trocas de perguntas e respostas entre alunos e professores, criação de listas de discussões e alimentação dos debates e pesquisas com textos, páginas de internet, independente do horário da aula. A possibilidade de estarmos todos presentes em diferentes espaços de tempo torna-se cada vez mais acentuada.

Assim, professores e alunos estarão cada vez mais motivados, entendendo o termo “aula” como pesquisa e intercâmbio. Nesse processo, o papel do professor tende a ser cada vez mais próximo ao de supervisor, motivador, e incentivador dos alunos na busca pelo conhecimento.

As crianças necessitam de interação com outras crianças e pessoas, para seu próprio desenvolvimento pessoal e social, e por isso não podem prescindir do contato físico. Mas para os jovens e adultos, o sistema de ensino virtual irá superar o presencial. Haverá, no entanto, uma reorganização nas escolas, com edifícios

menores, menos salas de aulas, mais salas de pesquisas, de encontros. Além da escola, também a casa ou escritório serão importantes locais de aprendizagem (Moran, 2002).

2.1.1. Modelo inicial de comunicação: via de mão-dupla

De acordo com Silva, Silveira e Balbinot (2004),

A Educação representa um processo interativo, um conjunto de ações responsáveis por toda articulação entre as entidades envolvidas no processo de transferência do conhecimento.

Toda transferência, por sua vez, diz respeito à comunicação, à troca de dados e informações entre elementos. Na Educação, entendemos que a transferência do conhecimento, ou da informação, deve ser tratada como uma comunicação interativa em uma via de mão-dupla, onde o conhecimento trafega nos dois sentidos, num processo de troca que proporciona a interação entre as entidades que se comunicam.

Em concordância com o estudo realizado, observa-se a existência de três entidades comuns a todas as propostas de modelo educacional: Aprendiz, Professor e Conhecimento. De forma bem simples, pode-se dizer que Aprendiz é a entidade interessada na sedimentação e aquisição de novos conhecimentos para si e que Professor é a entidade predisposta a disponibilizar o seu tempo, sabedoria e experiência em favor dos Aprendizes. Quanto ao Conhecimento, podemos entendê-lo como a informação que é transmitida. A representação deste último como uma entidade possibilita sua modelagem e implementação de forma a permitir sua reutilização por vários Aprendizes e Professores (Hanna, 2000), como mostra a Figura 1.

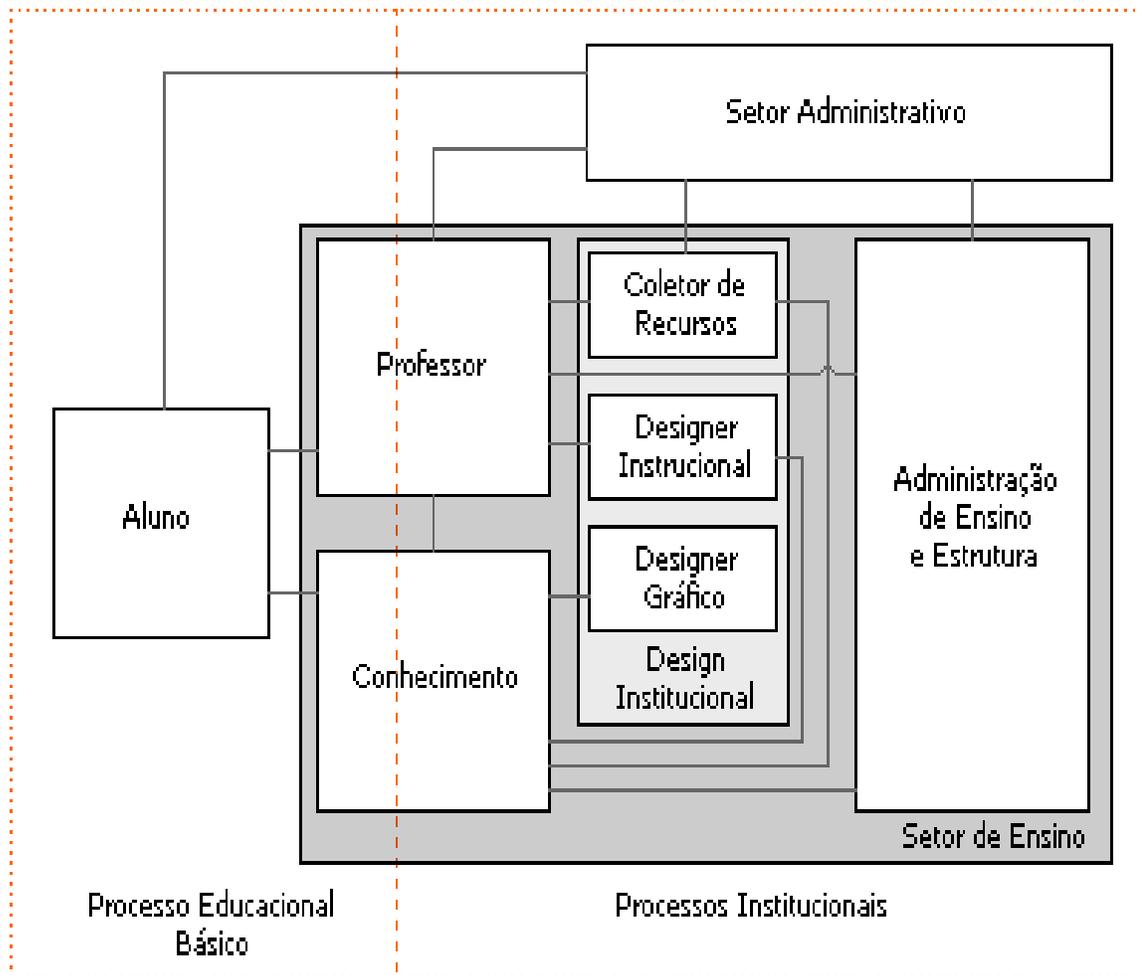


Figura 1 - Modelo com concepção de cursos em equipe. (Silva e Balbinot , 2004)

A comunicação sob o aspecto prático de implantação, permite a identificação dos usuários e a prestação de serviços para os mesmos. Além disso, dado o contexto "virtual" de implementação do sistema (computador, Internet,...), faz-se necessária a adição de especialistas ao sistema. Neste ponto, contempla-se a viabilização prática da implementação de cursos e aulas em equipe (Silva, Silveira e Balbinot, 2004).

A Figura 2 mostra a plataforma de comunicação do EAD tendo o aluno como o centro do processo educacional.

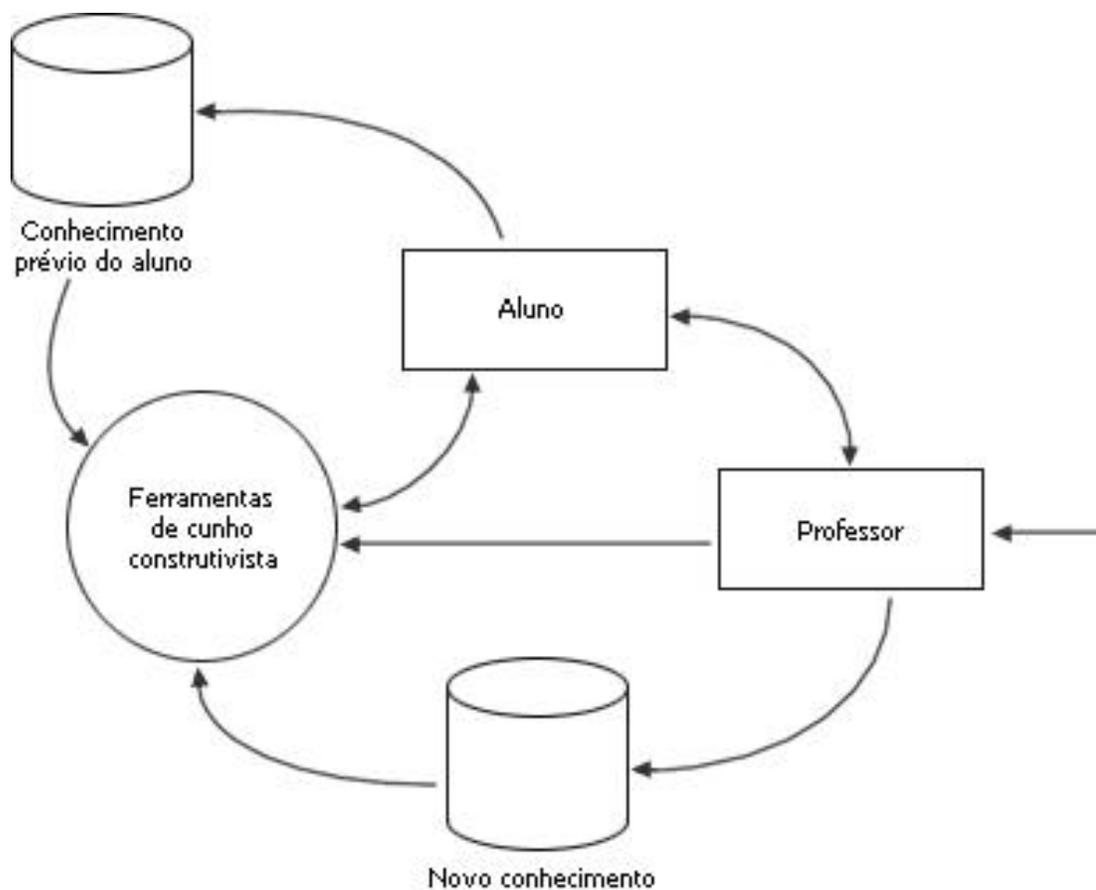


Figura 2 - Plataforma de comunicação EAD tendo o aluno como centro do processo educacional. (Silva e Balbinot , 2004)

2.2. REALIDADE VIRTUAL

De acordo com Kirner (1995),

Realidade Virtual (RV) pode ser definida de uma maneira simplificada como sendo a forma mais avançada de interface do usuário de computador até agora disponível. Com aplicação na maioria das áreas do conhecimento, senão em todas, e com um grande investimento das indústrias na produção de hardware, software e dispositivos de E/S especiais, a Realidade Virtual vem experimentando um desenvolvimento acelerado nos últimos anos e indicando perspectivas bastante promissoras para os diversos segmentos vinculados com a área.

Uma definição um pouco mais refinada de Realidade Virtual é a seguinte: "Realidade Virtual é uma forma das pessoas visualizarem, manipularem e interagirem com computadores e dados extremamente complexos". Agrupando algumas outras definições de Realidade Virtual, pode-se dizer que Realidade Virtual é uma técnica avançada de interface, onde o usuário pode realizar imersão, navegação e interação em um ambiente sintético tridimensional gerado por computador, utilizando canais multi-sensoriais.

2.2.1. Modelo de um mundo virtual

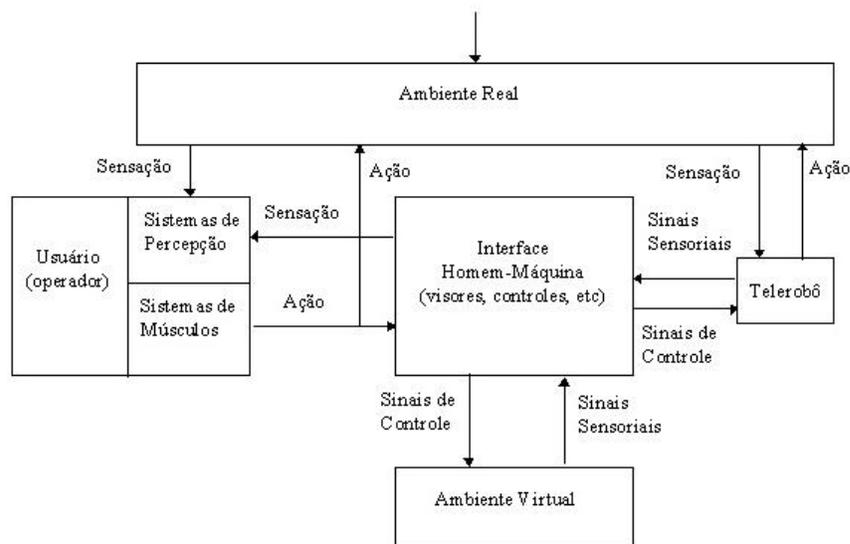


Figura 3 – Modelo de Mundo Virtual. (Silva e Balbinot, 2004)

A Figura 3 representa um mundo virtual. Por se tratar de um sistema, possui seus módulos que funcionam de forma independente, mas todos interligados.

2.2.1.1. Modelagem geométrica

Modelagem geométrica diz respeito a tudo que está ligado ao visual, tudo que os olhos podem ver no mundo virtual. Ela abrange a forma dos objetos como polígonos, vértices, textura, aparência, reflexão da superfície, posições no mapa, etc.

2.2.1.2. Restrições cinemáticas

A modelagem geométrica de um objeto não é suficiente para conseguir uma animação, como por exemplo, agarrar um objeto, alterar sua posição, mudar a escala, detectar colisões e produzir deformações na superfície. A utilização de coordenadas locais dos objetos e de coordenadas gerais, juntamente com matrizes de transformação, permitirá a alteração das posições e as mudanças de escala.

2.2.1.3. Modelagem física

Visando a obtenção de realismo nos mundos virtuais, os objetos virtuais, incluindo a imagem do usuário precisam comportar-se como se fossem reais. No mínimo, os objetos sólidos não poderão passar um pelo outro e as coisas deverão mover-se de acordo com o esperado, quando puxadas, empurradas, agarradas, etc. Nesse sentido, os objetos virtuais também deverão ser modelados fisicamente pela especificação de suas massas, pesos, inércia, texturas (lisas ou ásperas), deformações (elásticas ou plásticas), etc. Essas características, juntas com a modelagem geométrica e com as leis de comportamento, determinam uma modelagem virtual próxima da realidade. A simulação mecânica do mundo virtual, para ser realista, deverá ser executada de maneira confiável, contínua, automática e em tempo real.

2.2.1.4. Comportamento do objeto

As modelagens anteriores limitaram-se à modelagem matemática das propriedades cinemáticas e físicas dos objetos, visando uma resposta realista as ações do usuário. Também é possível modelar o comportamento de objetos independentes do usuário, como relógio, calendário, termômetro e outros agentes inteligentes independentes, acessando quando necessário alguns sensores externos.

2.2.2. Realidade Virtual Imersiva e Não-Imersiva

A RV também pode ser caracterizada pela coexistência integrada de três idéias básicas: imersão, interação e envolvimento (Morie, 1994). A idéia de imersão está ligada ao sentimento de fazer parte do ambiente. Normalmente, um sistema imersivo é obtido com o uso de capacete de visualização, ou cavernas; sistemas imersivos baseados em salas com projeções das visões nas paredes, teto, e piso (Cruz-Neira, 1992). Além do fator visual, dispositivos ligados aos demais sentidos também são importantes para o sentimento de imersão, como o som (Begault, 1994; Gradecki, 1994), o posicionamento automático da pessoa e dos movimentos da cabeça, controles reativos, etc. A visualização de uma cena 3D em um monitor é considerada não-imersiva. Dessa forma, tem-se a conceituação de RV imersiva e não-imersiva. De modo geral, do ponto de vista da visualização a RV imersiva utiliza capacete ou cavernas, enquanto a RV não-imersiva utiliza monitores. Entretanto, dispositivos baseados nos demais sentidos podem introduzir algum grau de imersão à RV que usa monitores (Robertson, 1993). Os monitores ainda apresentam alguns pontos positivos, como o baixo custo e a facilidade de uso, evitando as limitações técnicas e problemas decorrentes do uso do capacete. Porém, a tendência deve ser a utilização da RV imersiva na grande maioria das aplicações futuras. A interação está ligada à capacidade do computador detectar as entradas do usuário e modificar instantaneamente o mundo virtual em função das ações efetuadas sobre ele (capacidade reativa). As pessoas são cativadas por uma boa simulação em que as cenas mudam em resposta aos seus comandos, que é característica mais marcante

dos vídeo games. Para que um sistema de RV pareça mais realista, o ambiente virtual inclui objetos simulados. A Figura 4 mostra a Realidade Virtual imersiva.

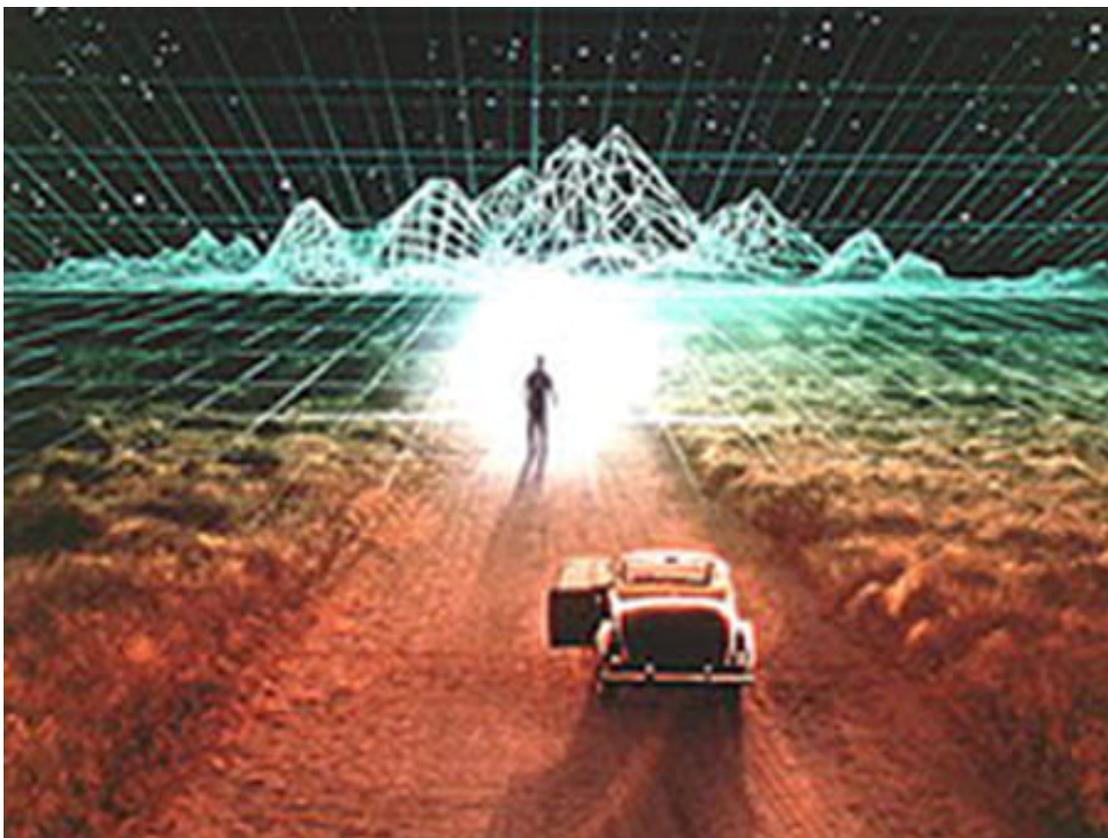


Figura 4 - RV Imersiva. (Hosn e Costa, 1999)

2.2.3. Aplicabilidades da Realidade Virtual

Será apresentada a seguir uma demonstração das diversas áreas onde a RV já é atualmente aplicada.

2.2.3.1. Realidade Virtual de simulação

A RV foi inicialmente utilizada como simuladores de vôo pelo militares americanos durante o período da segunda guerra mundial. (Jacobson, 1994).



Figura 6 - O simulador de vôo da Boeing 777 considerado a mais sofisticada aplicação de RV. (Hosn e Costa, 1999)



Figura 5 – Simulador de cockpits. (Hosn e Costa, 1999)

2.2.3.2. Realidade Virtual de Projeção

Segundo Jacobson (1994), “a Realidade Virtual de Projeção, foi criada na década de 70 por Myron Krueger, caracterizando-a pelo usuário estar fora do mundo virtual, mas que por outro lado, poderia se comunicar com personagens ou objetos dentro dele”.

Krueger também em 1994, cria a RV de projeção denominada VIDEOPLACE, ou “sala de projeção”, que capturava imagens de um ou mais usuários e projetava-as numa grande tela que representava um mundo virtual, onde era possível à interação destes usuários uns com os outros ou com objetos.



Figura 7 - Trator virtual baseado em Projeção de Tela. (Hosn e Costa, 1999)

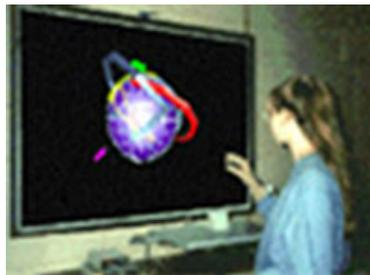


Figura 8 - Visor baseado em Projeção de Tela. (Hosn e Costa, 1999)

2.2.3.3. Augmented Reality (Realidade Realçada)

A Augmented Reality utiliza dispositivos visuais transparentes presos à cabeça nos quais os dados são projetados. Pelo fato desses displays serem transparentes, o usuário pode ver dados, diagramas, animações e gráficos 3D sem deixar de enxergar o mundo real, tendo informações sobrepostas ao mundo real. Estes displays transparentes são chamados Heads Up Displays (HUDs) por permitirem essa visão através das informações geradas pelo computador. O usuário pode, por exemplo, estar consertando algo e visualizando nos óculos os dados necessários a esta operação (Pimentel e Teixeira, 1995).



Figura 9 - Realidade Realçada. (Hosn e Costa, 1999)



Figura 10 - Realidade Realçada. (Hosn e Costa, 1999)

2.2.3.4. Dispositivos Físicos

Segundo Gradecki (1994), os dispositivos físicos procuram estimular as sensações físicas, como o tato, tensão muscular e temperatura. Diferente dos dispositivos de saída de visão e audição, os dispositivos físicos requerem uma sofisticada interação eletromecânica com o corpo do usuário. A tecnologia existente atualmente, não é capaz de estimular os sentidos físicos com o nível de realismo que atinge os sentidos visuais e auditivos: o problema está além da criação de dispositivos de feedback, pois envolve também a compreensão e simulação das forças apropriadas (Pimentel, 1995).

- **Feedback Tátil**

Feedback tátil é o nome dado a sistemas que transmitem sensações que atuam sobre a pele. O feedback tátil (Figura 11) deve fornecer não apenas a sensação do toque mas também permitir ao usuário perceber se está tocando uma superfície lisa ou rugosa (Gradecki, 1994). Existem atualmente duas diferentes formas de fazer essa simulação tátil: através de pressão de ar e através de vibrações (Pimentel, 1995).

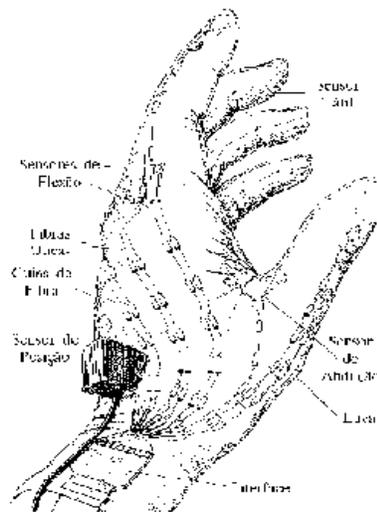


Figura 11 - Esquema de luva de dados baseada em sensores de fibra ótica.
(Hosn e Costa, 1999)

- **Feedback de Força**

Sistemas que permitem as sensações de pressão ou peso oferecem feedback de força. Uma maneira de construção de um sistema de feedback de força seria através de uma espécie de exoesqueleto mecânico que se encaixa no corpo do usuário, fazendo com que determinados movimentos possam permitir-lhe sentir o peso ou a resistência do material de um objeto no mundo virtual (Gradecki, 1994).

Alguns sistemas transmitem feedback de força apenas para as mãos e braços, como na Figura 12. Através do uso de pistões, por exemplo, é possível controlar a quantidade de resistência do braço e/ou da mão do usuário. No entanto, este tipo de sistema limita a faixa de possíveis situações de feedback, além de ser demasiadamente caro (Gradecki, 1994).

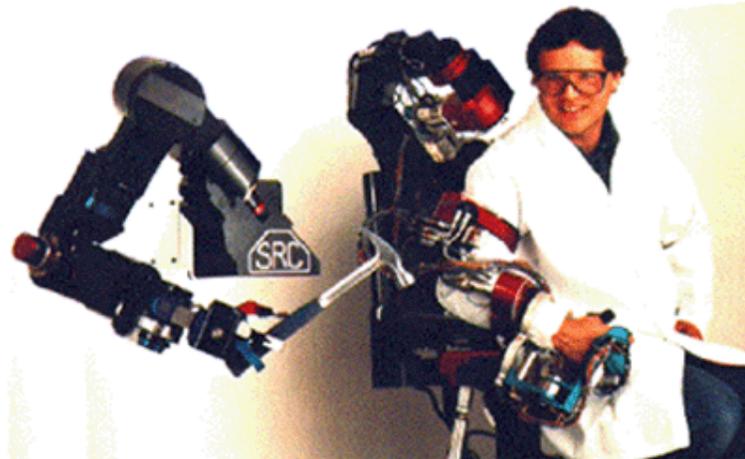


Figura 12 - Exemplo de sistema de feedback de força para mão e braço. O usuário pode "sentir" o peso e o cabo do martelo, enquanto um braço mecânico executa a mesma ação com o martelo real. (Hosn e Costa, 1999)

- **Feedback de Térmico**

Um tipo de feedback que também pode ser fornecido por um sistema de RV é o feedback térmico. Este feedback poderia ser fornecido, por exemplo, quando o usuário se aproximasse de uma fogueira no mundo virtual (Gradecki, 1994).

O feedback térmico não é muito utilizado em sistemas de RV devido ao seu alto custo, mas já existem algumas pesquisas neste campo sendo desenvolvidas. Uma dessas pesquisas fez uma empresa do Texas desenvolver um sistema que aquece parte do corpo através de um pequeno dispositivo que reúne um aquecedor, um sensor de temperatura e um inversor de calor (Gradecki, 1994).

2.2.3.5. Telepresença

Enquanto a Telepresença faz com que a interface atue sobre o telerobô que vai agir sobre o mundo real, o sistema de RV faz com que a interface opere diretamente sobre o computador, que vai interagir sobre um mundo virtual ou um mundo real

simulado. Em telepresença e em outros casos, onde possa haver dificuldades de transferência ou tratamento em tempo real de imagens reais complexas, a substituição do mundo real por um mundo virtual equivalente pode resolver o problema, na medida em que as imagens podem ser geradas localmente. As transferências de informações podem ser reduzidas a dados de posicionamento.

Por sua vez, o propósito fundamental de um sistema de Telepresença, é estender as capacidades motoras e sensoriais de um operador, para um ambiente remoto, para que tarefas complexas possam ser realizadas. O operador usa dispositivos de rastreamento que detectam seus movimentos e enviam estas informações ao telerobô, que, por sua vez, repete as ações do operador no ambiente remoto. Tais dispositivos, mais comuns para Telepresença são: capacete de RV, que produz uma visão estereoscópica do ambiente dando a sensação de imersão, sistema de som tridimensional e luvas ou dispositivos que possibilitam ao usuário manipular objetos. Este tipo de RV utiliza câmeras de vídeo e microfones remotos para criar a imersão e a projeção do usuário profundamente no mundo virtual. Controle de robôs e exploração planetária são exemplos de pesquisas em desenvolvimento. No entanto, existe um grande campo de pesquisa no uso de telepresença em aplicações médicas. Em intervenções cirúrgicas, médicos já utilizam câmeras de vídeo e cabos de fibra óptica para visualizar os corpos de seus pacientes. Através da RV eles podem literalmente "entrar" no paciente indo direto ao ponto de interesse e/ou vistoriar a operação feita por outros (Pimentel e Teixeira, 1995).



Figura 13 - Protótipo de um telerobô utilizado em explorações espaciais.

(Hosn e Costa, 1999)

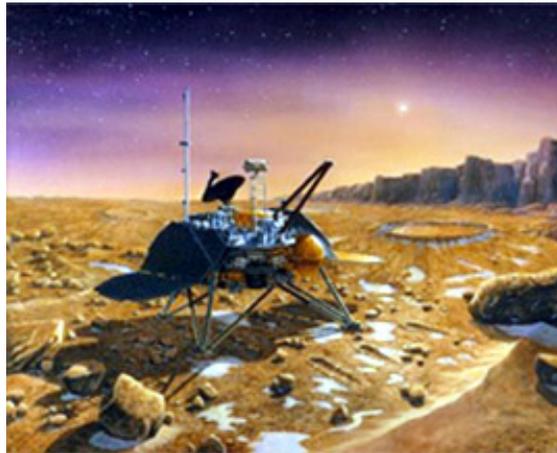


Figura 14 - Telerobô "Pathfinder" da NASA criado para explorar Marte em 1999.
(Hosn e Costa, 1999)

2.2.3.6. Displays visualmente casados

Esta é a classe de sistemas geralmente mais associada à RV. Nos sistemas desta classe as imagens são exibidas diretamente ao usuário, sendo que o usuário está olhando em um dispositivo que deve acompanhar os movimentos de sua cabeça. Este dispositivo geralmente permite imagens e sons em estéreo, além de conter sensores especiais ligados a ele que detectam a movimentação da cabeça do usuário, usando esta informação para realimentação da imagem exibida (Pimentel e Teixeira, 1995).

Desta forma, "Displays Visualmente Casados" são formas de RV que podem ser representados por dispositivos HMD's AMD's, BOOM e outros como mostram a Figura 15 e Figura 15.



Figura 16 - BOOM. (Hosn e Costa, 1999)



Figura 15 - HMD. (Hosn e Costa, 1999)

2.2.3.7. RV de Mesa (Desktop RV)

Pimentel e Teixeira (1995) vêem a RV de Mesa como um subconjunto dos sistemas tradicionais de RV. Ao invés do uso de "Head Mounted Displays" (HMD's), são utilizados grandes monitores ou algum sistema de projeção para apresentação do mundo virtual. Alguns sistemas permitem ao usuário ver imagens 3D no monitor através do uso de óculos obturadores, polarizadores ou com filtros coloridos.



Figura 17 - Sistema de RV de Mesa. (Hosn e Costa, 1999)

2.2.4. Realidade Virtual na Educação

Atualmente, várias ferramentas em Realidade Virtual para a educação estão sendo pesquisadas. Certamente, uma das mais conhecidas aplicações educacionais é o treinamento de pilotos comerciais em simuladores de vôo. Os militares também estão bastante interessados na utilização da Realidade Virtual para o treinamento de combatentes, e vem investindo pesadamente em pesquisas na área.

Porém, dentro do escopo deste estudo, estamos mais interessados na utilização das técnicas de Realidade Virtual na sala de aula (ou sala de aula virtual). Serão apresentados alguns exemplos de projetos de pesquisas na área que estão sendo conduzidos por instituições ao redor do mundo:

- **Projeto Professor Virtual:** em desenvolvimento pelo Grupo de Realidade Virtual do Departamento de Computação da Universidade Federal de São Carlos. Trata-se de um sistema que permite que o professor seja transportado para uma sala de aula remota e receba informações da classe, podendo interagir com os alunos como se estivesse presente. Utiliza técnicas de Realidade Virtual e tele-presença. O sistema oferecerá ferramentas adicionais que aumentarão as possibilidades de interação como por exemplo, a identificação dos alunos (crachá virtual), prontuários e experimentos (laboratórios virtuais) (Kirner, 1999).
- **Projeto REVIR:** em desenvolvimento pelo Grupo de Realidade Virtual do Departamento de Computação da Universidade Federal de São Carlos. O projeto enfoca o uso integrado da Realidade Virtual e técnicas de visualização científica, visando o desenvolvimento de ferramentas e programas computacionais aplicados ao ensino fundamental, dando ênfase nas séries iniciais do primeiro grau (primeira a quarta séries), para as quais se implementará inicialmente o Ambiente Virtual para Educação no Trânsito, com simulações, animações e ferramentas de navegação (Bugatti, 1999).

- **Aprendendo Astronomia: O Sistema Solar:** disciplina experimental oferecida pela Universidade de Indiana para alunos de astronomia no estudo do sistema solar. A abordagem é utilizar modelos em Realidade Virtual criados no computador pelos próprios alunos, utilizando os software CosmoWorlds e VRCreator. As tarefas de aula requerem que os alunos incorporem em seus trabalhos fatos, conceitos e princípios associados à astronomia em geral, e ao sistema solar em específico. Os trabalhos finais ficam disponíveis na Internet e podem ser visualizados pelos alunos em dispositivos do tipo CAVE (Universidade de Indiana, 2000).

- **Projeto Gorila Virtual:** é uma ferramenta educacional sendo desenvolvida por pesquisadores do Laboratório de Computação Gráfica, Visualização e Usabilidade e do programa de Doutorado da Faculdade de Arquitetura do Instituto de Tecnologia da Geórgia, em cooperação com os pesquisadores e funcionários do Zoológico de Atlanta. Neste projeto, a Realidade Virtual ajudará visitantes do zoológico a entender e apreciar o comportamento dos gorilas. O objetivo do projeto é fazer com que os visitantes, por meio de técnicas de Realidade Virtual, ao entrarem num Mundo Gorila e fazerem parte de uma família de gorilas virtuais, possam pensar e se comportar assim como um gorila o faria, de modo a melhor entender as necessidades conversacionistas desta espécie ameaçada de extinção (Hodges 2000). A Figura 18 mostra o ponto de vista do usuário dentro do mundo gorila.



Figura 18 - Ponto de vista do usuário dentro do mundo gorila. (Hodges, 2000)

- **Projeto Science Space:** desenvolvido conjuntamente pela Universidade George Mason, Universidade de Houston e Centro Espacial Johnson da NASA, tem como objetivo explorar as vantagens e desvantagens da Realidade Virtual na Educação. O projeto é uma coleção de mundos virtuais imersivos projetados para auxiliar estudantes em compreender e dominar difíceis conceitos de ciências. Alguns mundos virtuais já implementados são: O Mundo de Newton (Figura 19), O Mundo de Maxwell e O Mundo de Pauling (Dede, 2000).

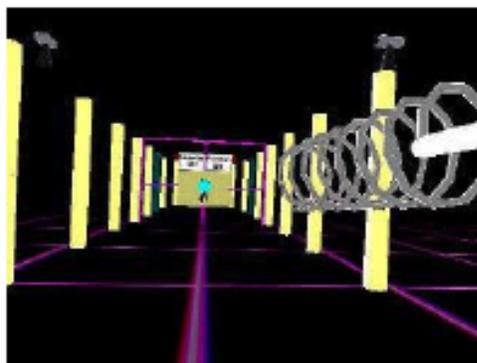


Figura 19 - No Mundo de Newton, o usuário aprende conceitos básicos de cinemática e dinâmica. (Dede, 2000)

- **Projeto Starship:** desenvolvido pelo Laboratório de Tecnologia de Interface Humana da Universidade de Washington (HITL) e o Museu do Voo de Seattle, o projeto Starship é um programa educacional que visa desenvolver abordagens inovadoras nas quais as mais modernas tecnologias de interface, entre elas a Realidade Virtual, serão utilizadas para ensinar grupos de estudantes, adultos e famílias sobre as ciências espaciais (HIT Lab, 2000).

2.3. 3D STUDIO GAME

A *engine 3D Studio Game* é uma ferramenta específica para a criação de jogos 3D. Ela vem acompanhada de um editor de modelos (MED), um editor de mapas (WED) e uma IDE para a geração de códigos script (SED). A linguagem de programação própria da ferramenta é a WDL (*World Definition Language*), também chamada de “C-Script” pela semelhança com a linguagem C.

A versão utilizada para o desenvolvimento do trabalho é o *3D Game Studio Version 6.20.2.0*, devido à vasta quantidade de materiais explicativos e tutoriais. A última versão existente do *3D Game Studio* é o A7. Por ser uma versão muito recente, a quantidade de materiais para a mesma é ainda muito escassa. Em função disso foi utilizada a versão anterior.

2.3.1. WED (World Editor)

O WED é o editor de mundos virtuais no 3D Game Studio. Com ele é possível criar ambientes em três dimensões desde os mais simples como pequenas salas até os mais complexos como cidades inteiras.

Ele possui uma interface simples, com as paletas de ferramentas e os comandos básicos de qualquer modelador 3D. Os arquivos gerados pelo WED tem a extensão “.WMP” que é restrito ao 3D *Studio Game*, mas ele permite a importação de arquivos com extensões criados em vários outros editores.

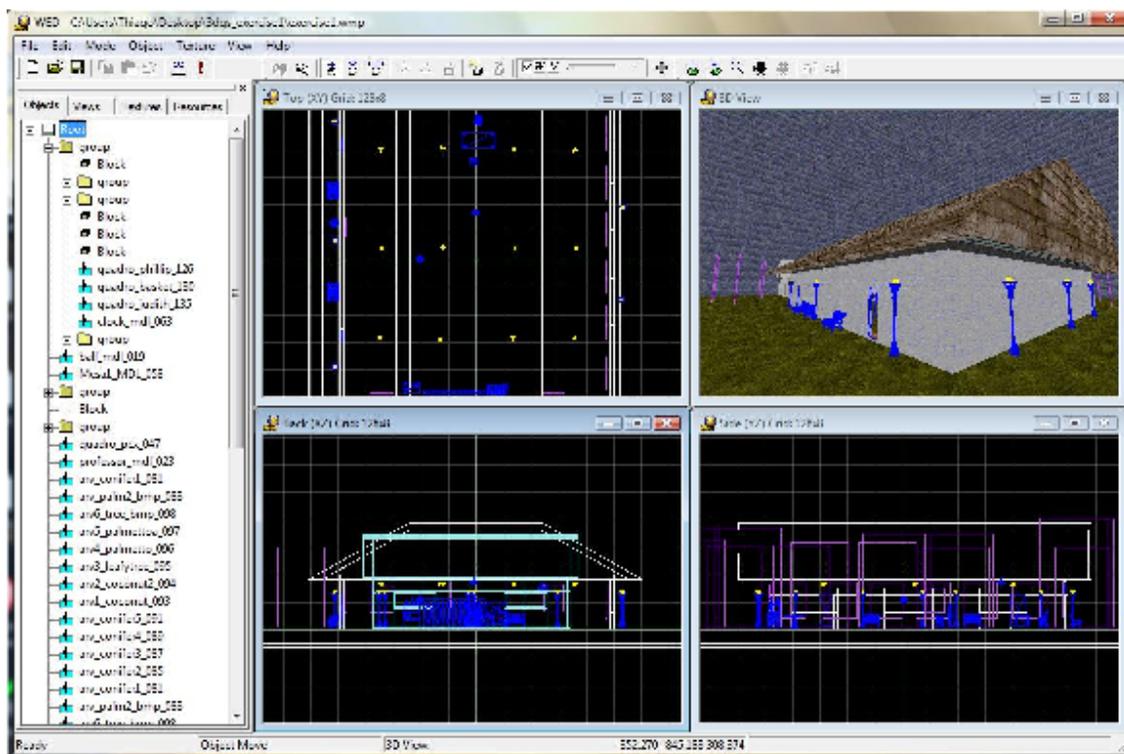


Figura 20 - Exemplo de criação de um cenário no WED

2.3.2. MED (Model Editor)

O MED é o editor de modelos do 3D *Studio Game*. Nele são criados os modelos que vão compor o ambiente dentro do mundo virtual. Esses modelos podem ser figuras humanas ou animais, plantas e objetos. Esse editor de modelos é muito simples e fácil de usar, mas deixa um pouco a desejar na criação de modelos mais complexos.

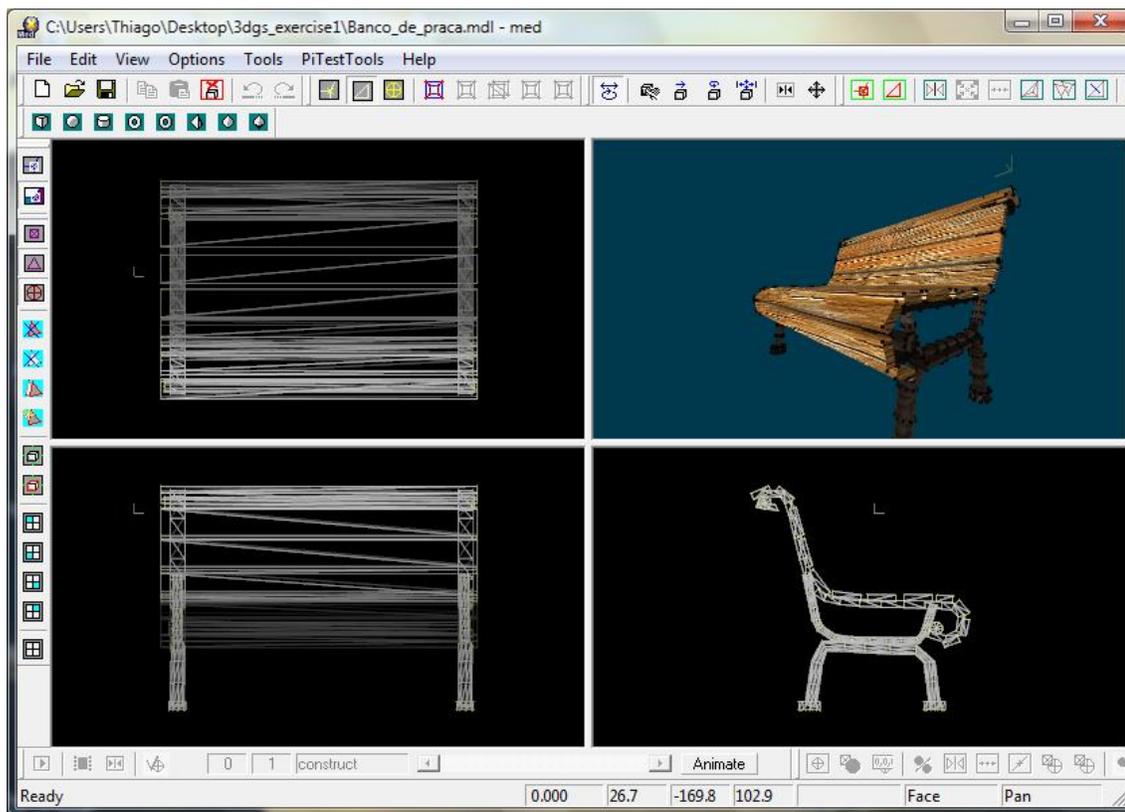


Figura 21 - Exemplo de criação de modelos.

2.3.3. SED (Script Editor)

O SED é o editor de códigos *script* suportado pelo 3D *Studio Game*. É usada a linguagem WDL (*World Definition Language*), também chamado de “*C-Script*” pela semelhança com a linguagem C. Ele possui todas as características típicas de um compilador completo como correção de sintaxe em tempo real, auto-complementação do código, etc. Mesmo assim ele é dispensável para quem quiser, nada impede que seja usado outro editor.

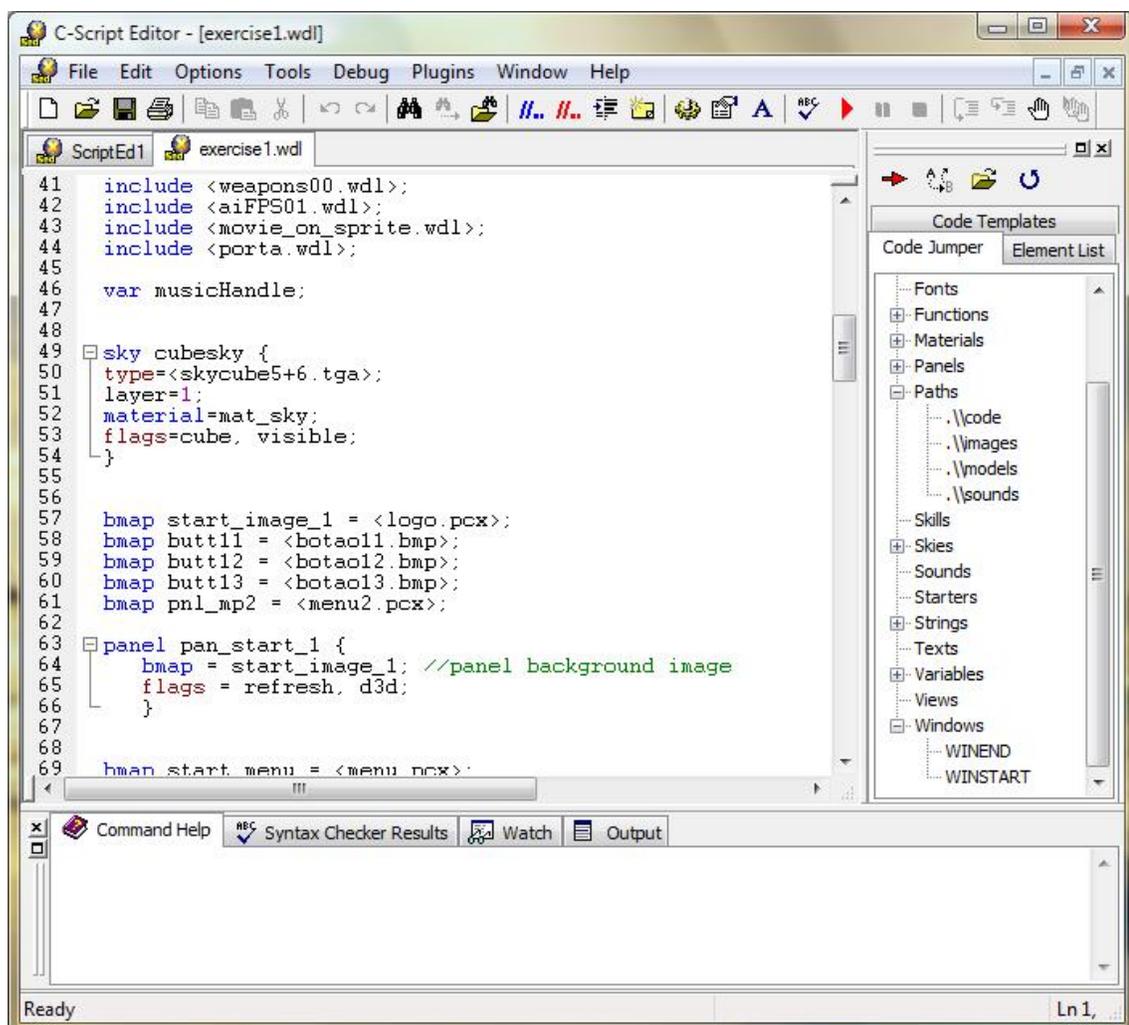


Figura 21 - Exemplo de criação de um código.

CAPÍTULO 3

MODELAGEM DO PROBLEMA

Neste capítulo será apresentada toda a modelagem da classe virtual proposta, com suas regras de Realidade Virtual, regras de interação entre usuários, e informações sobre cada componente do cenário.

3.1. DESCRIÇÃO DO PROBLEMA

Com base em toda a teoria descrita anteriormente, este trabalho tem como proposta desenvolver uma sala de aula virtual em 3D, utilizando as técnicas de Realidade Virtual. Este ambiente será dotado das regras de interação que são aluno – professor, professor – aluno, e aluno – aluno. A implementação de todas estas regras dentro do ambiente virtual é o que designa que este sistema seja mais voltado para o EAD. A Figura 22 mostra o modelo da sala virtual 3D e suas interações.



Figura 22 - Modelo de Sala Virtual.

O problema será dividido em três fases:

- **Primeira fase**

A primeira fase é composta basicamente da criação do cenário e dos personagens. Para isso será utilizado a ferramenta 3D *Studio Game*, que é uma *engine* muito utilizada para desenvolvimento de jogos em 3D. Este problema já é resolvido em partes, pois ao se utilizar uma *engine* para desenvolver um ambiente de RV, o trabalho do programador já se reduz consideravelmente, uma vez que a própria *engine* trata internamente problemas gráficos como renderização, posições de polígonos, linhas, e vértices, posição dos objetos no mapa e texturas.

- **Segunda fase**

A segunda fase do problema será a animação do personagem. Nesta fase, devido a dificuldade em criar a animação real dos personagens ela se limitará apenas em fazer com que o personagem possa andar, movendo braços e pernas. Para isso foi utilizado um personagem já criado, que vem junto com o pacote de objetos gráficos do 3D *Studio Game*. No futuro, será feito um aperfeiçoamento na animação dos personagens.

- **Terceira fase**

A terceira fase é a mais importante de todas pois trata-se da interação entre duas máquinas conectadas em rede. Uma destas máquinas será o servidor, onde estará rodando todo o cenário. A máquina servidor deverá conectar-se ao cenário com um personagem, ao passo que a outra máquina conectada na rede, deverá acessar este mesmo cenário. Desta forma, os dois personagens poderão encontrar-se simultaneamente, em um mesmo cenário.

3.2. MODELAGEM DO PROBLEMA

A Figura 23 mostra a arquitetura da sala de aula virtual que será desenvolvido neste trabalho.

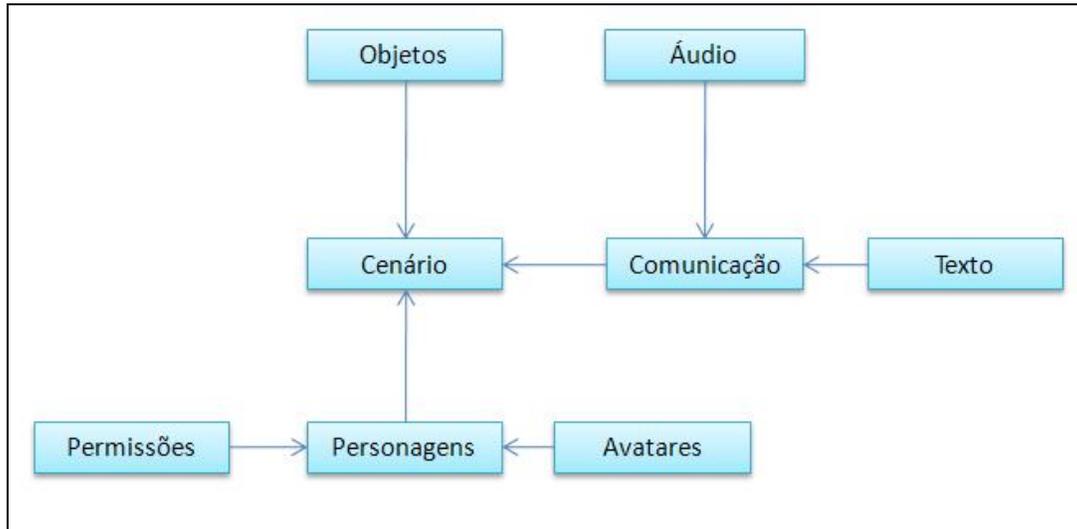


Figura 23 - Relacionamentos do Sistema.

3.3. DESCRIÇÃO GERAL DOS RELACIONAMENTOS

3.3.1. Cenário

O cenário é o ambiente onde todas as partes deste projeto se conectam. Este ambiente virtual tem por finalidade simular uma sala de aula em 3D.

A sala de aula é uma sala comum, possuindo uma porta, uma lousa e móveis, tais como cadeiras e mesas para os alunos e uma mesa de professor. Também deverá haver luzes no local, permitindo uma total iluminação do cenário.

3.3.2. Objetos

Os objetos estão dentro do cenário. São responsáveis por proporcionar uma maior sensação de realidade ao cenário. Os objetos também são responsáveis por identificar cada cenário. No caso deste trabalho, os objetos que identificam uma sala de aula são as mesas e cadeiras, uma lousa, janelas, lâmpadas, etc.

3.3.3. Personagens

Todos os personagens podem andar pelo cenário usando as teclas direcionais do teclado para mover seu corpo, e movimentando o mouse para mover a cabeça. Personagens também podem interagir com o cenário, por exemplo, ao se aproximar de uma porta, esta deve se abrir com um comando do usuário.

3.3.4. Avatares

Avatares são as “caras” de cada personagem, a forma de sua visualização. Cada pessoa conectada ao ambiente deverá ter um *avatar*. *Avatar* é o que identifica cada pessoa.

Exemplo:

- **Professor:** pode possuir vestimentas únicas, para que se destaque entre os alunos, podendo ser uma roupa mais social, ou jaleco branco.
- **Aluno:** pode haver dois tipos de alunos, homem e mulher. Uma vez escolhido o sexo do personagem, será possível escolher a cor das roupas, formas da face, cabelo, peso, altura, etc.

3.3.5. Permissões

As permissões são relacionamentos entre professor e aluno.

- **Professor:** o personagem do tipo “Professor” deverá possuir várias habilidades especiais, como por exemplo, a possibilidade de cortar a comunicação entre os alunos da sala, desta forma, os demais poderiam apenas ouvir o professor. Outra habilidade é a possibilidade de diminuir o áudio dos demais presentes, desta maneira a voz do professor poderá facilmente sobressair sobre a voz dos alunos. Quando a comunicação dos alunos estiver cortada, o professor poderá reabilitar a comunicação do aluno desejado, selecionando seu personagem e escolhendo esta opção.
- **Aluno:** ao selecionar outro personagem, tem-se a opção de comunicar apenas com ele, independente da sua distância dentro do ambiente. Uma vez que um Aluno estiver com sua comunicação cortada pelo Professor, este terá uma opção que, quando usada, o personagem levantará uma das mãos, mostrando assim que ele deseja falar, neste caso o professor irá desbloqueá-lo, se assim desejar.

3.3.6. Comunicação

É a maneira como a comunicação entre os usuários deverá ser estabelecida. Vale tanto para comunicação no sentido de conexão, como de conversação. Poderá haver conexão de cliente para cliente (sem professor), podendo ser utilizada para grupos de estudo, além da conexão de cliente para servidor, onde todas as informações de alunos e arquivos utilizados nas aulas estarão armazenadas.

3.3.7. Áudio

A comunicação por Áudio será possível apenas para usuários que possuam microfones, e saídas de som (fones de ouvido ou caixas acústicas). Quando um usuário falar no seu microfone, todos os usuários próximos que tiverem caixas acústicas deverão receber a mensagem de voz que foi dita.

3.3.8. Texto

Uma opção de comunicação para quem não possui os equipamentos de multimídia é a comunicação por texto.

Na parte inferior da janela haverá um chat apenas para conversação através de textos.

3.3.9. Lousa virtual

Uma das mais importantes ferramentas para este sistema é a lousa virtual. Nela poderão ser executados vídeos, tele-aulas, exibição de textos, ou até mesmo a imagem de webcam do professor. Tudo o que for mostrado na lousa virtual será visível para todos que estiverem conectados ao cenário.

3.4. CASOS DE INTERAÇÕES

As maneiras como os personagens se interagem são de suma importância para que este cenário virtual fique mais próximo do cenário real. Nesta etapa, o propósito é simular as regras que existem em uma sala de aula real. Estas regras são especificamente o balaço de autoridade entre professor e aluno.

3.4.1. Interação Professor – Alunos

Esse estudo compreende a detecção das possíveis ações efetuadas pelo professor para relacionar-se com seus alunos.

Levando-se em conta que o sistema deva garantir ao professor um grande controle sobre a sala de aula, foram definidas as seguintes ações:

- **Identificar Alunos:** para a identificação dos alunos dentro do ambiente virtual é utilizado o que se denominou no projeto como crachá virtual. Cada aluno possui um crachá desse tipo, contendo seu primeiro nome. Clicando nesse crachá, uma ficha completa do aluno é exibida, contendo informações do aluno, tais como dados pessoais, foto, histórico acadêmico, etc.
- **Escolher com quem se comunicar:** supondo que o professor queira chamar a atenção de um aluno em particular, ou fazer uma pergunta (avaliação oral) somente a um aluno (ou grupo de alunos) sem que o restante da sala tenha conhecimento. Dessa forma, o sistema deve permitir ao professor dirigir a palavra a quem ele bem entender.
- **Restringir a comunicação entre alunos:** quando um aluno conecta-se no sistema (efetua *login*) ele pode comunicar-se com quem ele quiser. Porém, em algumas ocasiões seria conveniente ao professor escolher com quem o aluno poderá se comunicar. Tal ação, assim como em uma aula real, visa preservar a ordem da sala e atenção dos alunos. Além disso, a restrição de comunicação pode ser útil na formação de grupos de estudo pelo professor, como será visto adiante.

- **Desconectar alunos da sala:** o professor possui um grande controle sobre a sala de aula. Sendo assim, ele tem a liberdade de desconectar alunos do sistema, em casos extremos.
- **Controlar presença:** uma das tarefas do professor na sala de aula é referente ao controle da frequência dos alunos. Esse controle pode ser feito automaticamente pelo sistema, da seguinte maneira: quando o aluno se conecta ao sistema, sua presença é confirmada em uma ficha (arquivo), na qual podem ser armazenadas informações tal como a data atual e horário de entrada e de saída na aula. A partir desta ficha, o professor pode informar-se sobre a frequência de seus alunos.

3.4.2. Interação Alunos – Professor

Serão definidas aqui ações do aluno em seu relacionamento com o professor durante a aula.

- **Esclarecer dúvidas:** como citado anteriormente, as ações de um usuário dentro de um ambiente virtual são refletidas através de seu *avatar*. Portanto, quando um aluno tiver alguma dúvida sobre a matéria, ele poderá fornecer um comando específico ao sistema (clicar num botão, por exemplo) e seu *avatar* efetuará alguma ação (como levantar o braço). Dessa forma, o aluno ganhará a atenção do professor para fazer a pergunta.
- **Trocar anotações:** nessa parte do estudo analisam-se formas de como o aluno pode enviar/receber anotações ao professor. Tais anotações compreendem desde exercícios (que podem ser entregues na forma de arquivos), até rascunhos para esclarecimento de perguntas durante a aula.

3.4.3. Interação Alunos – Alunos

Será feito um estudo sobre o relacionamento entre os alunos dentro de uma sala de aula. Portanto, serão estudadas ações tais como a formação de grupos de estudos, como um aluno se comunicará com outro em particular, etc.

- **Escolher com quem se comunicar:** tal como na interação professor - alunos, onde o professor escolhe com quem se comunicar, o sistema deve permitir também a um determinado aluno a comunicação em particular. Para isso, o aluno tem duas opções: selecionar o(s) aluno(s) para que a comunicação seja restrita entre eles, ou desativar a comunicação com outros alunos.
- **Formar grupos de estudo:** uma atividade frequente entre alunos em uma sala de aula é se reunir em grupos, sendo os componentes desse grupo escolhidos pelo professor ou entre os próprios alunos. Deve ser levado em conta também se a comunicação entre os grupos é permitida ou não.

Uma maneira de formar grupos de trabalho é obtida através do uso de restrição de comunicação entre os participantes. Nos casos onde o professor é quem escolhe os componentes do grupo, ele poderá restringir a comunicação de um aluno somente aos outros participantes do grupo. No outro caso, onde os alunos formam os grupos, algo parecido acontece, só que agora cada aluno escolhe com quem se comunicar.

CAPÍTULO 4

DESENVOLVIMENTO DO AMBIENTE VIRTUAL 3D

Neste capítulo serão descritas todas as fases para o desenvolvimento do ambiente virtual em 3D (sala de aula) utilizando o 3D *Studio Game*.

4.1. COMANDO BÁSICOS ÚTEIS DO 3D STUDIO GAME

Nesta seção será feita uma descrição dos comandos básicos do 3D *Studio Game* utilizados para o desenvolvimento do ambiente virtual. Este procedimento visa facilitar o entendimento das várias etapas de criação do cenário.

4.1.1. Configurações do sistema

Para facilitar o desenvolvimento do cenário, é necessário levar em considerações algumas configurações do sistema, antes de realizar a sua primeira compilação.

Após a criação de um projeto, é importante acessar o *menu File/ Preferences*. Na aba *Advanced*, deve-se ativar a opção *Reload of external modified files*. Esta opção faz com que arquivos externos, como modelos ou imagens de *Sprites*, sejam automaticamente atualizados no WED caso sejam modificados.

Isto pode ser feito acessando o *menu File/ Map Properties*. O próximo passo é clicar sobre o botão *New Script* () e selecionar a opção *_A6_Template_Project*. Em seguida, clicar em ok.

Por padrão, o cenário está sempre noite, tornando visível apenas os locais próximos a luzes no cenário. Para tornar o ambiente mais claro, deve-se acessar o *menu File/*

Map Properties e alterar a opção *Ambient* para as respectivas cores RGB (*Red, Blue, Green*)= 25, 25, 25.

As sombras dos objetos são relativas à sua exposição à luz do sol, que por padrão será escuro. Para alterar isto, deve-se acessar o *menu File/ Map Properties*, e mudar o valor da opção *Sun* para as respectivas cores RGB (*Red, Blue, Green*)= 100, 100, 100, como mostra a Figura 24.

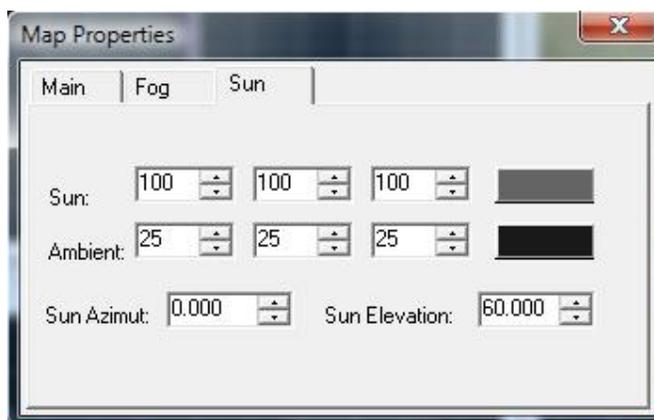


Figura 24 - Propriedades do mapa.

Também é possível alterar o ângulo horizontal do sol, que afeta também diretamente na sombra dos objetos. Isto pode ser feito alterando o valor *Sun Elevation* para um valor entre 0 e 360. Por padrão este valor é 60. Esta opção muda apenas o ângulo de elevação do sol, sendo a posição padrão horizontal 0.0 (norte).

4.1.2. Adicionado objeto

Uma das etapas mais importantes é adicionar objetos na criação do ambiente virtual. Serão descritos os passos para executar este processo.

Para adicionar um bloco é necessário acessar o *menu Object/ Add Cube/ Small, Medium ou Large*, como mostra a Figura 25. Em seguida é necessário redimensionar o seu tamanho de forma proporcional a uma sala de aula convencional.

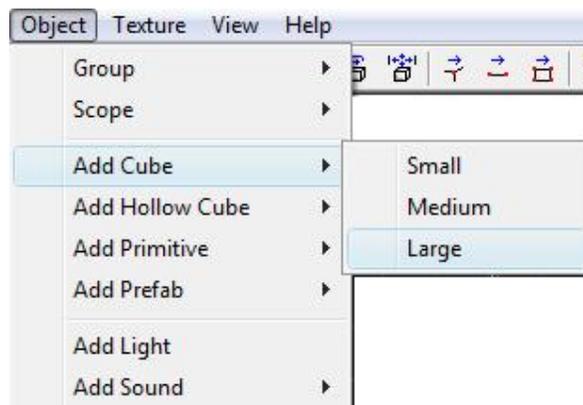


Figura 25 - Adicionando bloco.

4.1.3. Bloco oco

Havendo um bloco adicionado no cenário, este bloco deve ser selecionado, para que se possa aplicar sobre ele o comando *Hollow Block*, conforme mostra a Figura 26. Este comando faz com que o quadrado se torne oco, tendo cada um dos seus lados com a espessura de uma parede.

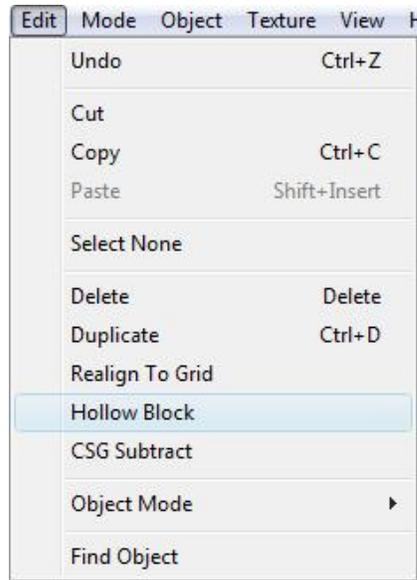


Figura 26 - Bloco oco.

4.1.4. Texturas

Para adicionar texturas em cada parte de um bloco, uma vez selecionado o bloco, deve ser ativado o comando *Scope Down* (📏↓). Este comando faz com que o bloco selecionado se divida em cada parte interna, podendo assim selecionar individualmente cada parte do bloco. Assim será possível selecionar e aplicar as texturas sobre cada componente do cenário. De acordo com a Figura 27, selecionando a aba *Textures*, pode-se alterar a textura de um objeto com um duplo-clique sobre a textura escolhida



Figura 27 – Texturas.

4.1.5. Subtração de objetos

Através deste comando é possível criar portas e janelas, além de construções mais complexas. Para se criar uma porta, o processo é criar um objeto do tipo bloco, e redimensioná-lo para que fique do tamanho de uma porta. Feito isto, o mesmo deve ser posicionado no local onde se deseja criar uma porta. O próximo comando faz um processo de subtração sobre os objetos que estiverem em interseção com o objeto selecionado. Ou seja, como o bloco criado está passando através da parede, a parte da parede que toca o bloco será subtraída, deixando uma fenda do mesmo tamanho do bloco, após retirá-lo. Este comando deve ser acessado no *menu Edit/ CSG Subtract*, conforme mostra a Figura 28.

Este mesmo procedimento pode ser executado para a criação das janelas.

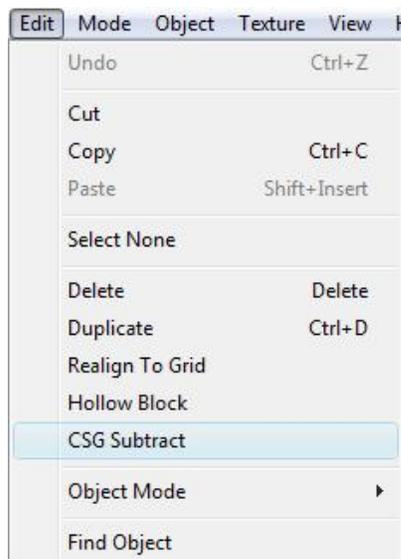


Figura 28 - Subtraindo Objeto.

4.1.6. Iluminação

Para um maior nível de realismo sobre o cenário, é necessário o posicionamento de algumas luzes. Luzes podem ser adicionadas ao cenário através do comando *Object/ Add Light*. Clicando sobre uma luz com o botão direito do mouse, e acessando suas propriedades, é possível alterar a cor da luz, além do seu raio, criando assim uma luz forte ou fraca, de acordo com a necessidade, como mostra a Figura 29.

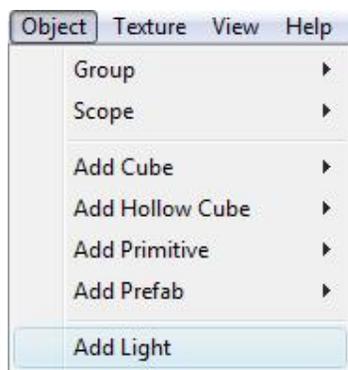


Figura 29 - Adicionando luzes.

4.1.7. Sprites

Sprites são imagens que podem ser carregadas diretamente no cenário, como placas, árvores, cartazes nas paredes, etc. É possível criar *sprites* fixos (caso das cortinas) e *sprites* móveis, como por exemplo, uma árvore. Ao se criar um *sprite*, se não for alterado a sua rotação, a imagem do *sprite* torna-se móvel durante a execução, rotacionando-se automaticamente para que se mantenha sempre de frente para o usuário. Este efeito é usado para criar árvores no cenário, pois deixa a execução muito mais leve do que usar um modelo de árvore.

Para adicionar um *sprite* no cenário, deve-se em primeiro lugar ter a imagem no formato *.PCX ou *.BMP. Tendo isso, é possível adicionar um *sprite* ao cenário acessando o comando *Object / Add Sprite*. A imagem desejada deve estar na pasta do projeto, então esta estará disponível nesta opção, podendo ser adicionada, posicionada e redimensionada no cenário, como mostra a Figura 30.

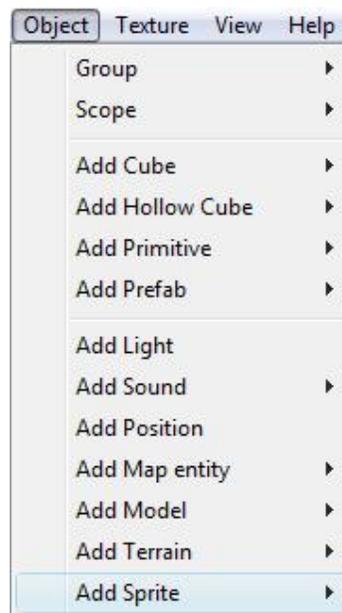


Figura 30 - Adicionando Sprite.

4.1.8. Modelos

Modelos são objetos capazes de receber habilidades e animações. Os modelos podem ser criados no MED, ou em outros editores de modelos compatíveis com o 3DSG. Para adicionar um modelo no cenário, este deve estar na mesma pasta que o projeto. A partir daí, basta acessar o *menu Object/ Add Model*, como mostra a Figura 31. Serão listados todos os objetos que estiverem na pasta do projeto. Escolhendo um objeto, este será adicionado no cenário, podendo ser redimensionado e posicionado no local desejado.

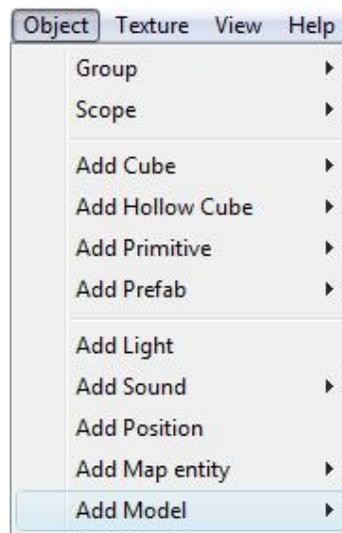


Figura 31 - Adicionando modelos.

4.1.9. Prefabs

Outra importante particularidade o *3D Studio Game*, é a quantidade de objetos pré-fabricados. O sistema conta com inúmeros objetos, tais como móveis, pontes, escadas, cercas, eletros-domésticos, etc. Funciona da mesma maneira que para adicionar um modelo, mas ao invés disso, o atalho é *Object/ Add Prefab*. Surgirá

uma lista contendo inúmeros itens gráficos que podem ser adicionados ao cenário. Estes itens já vêm texturizados, bastando serem redimensionados e posicionados, como mostra a Figura 32.

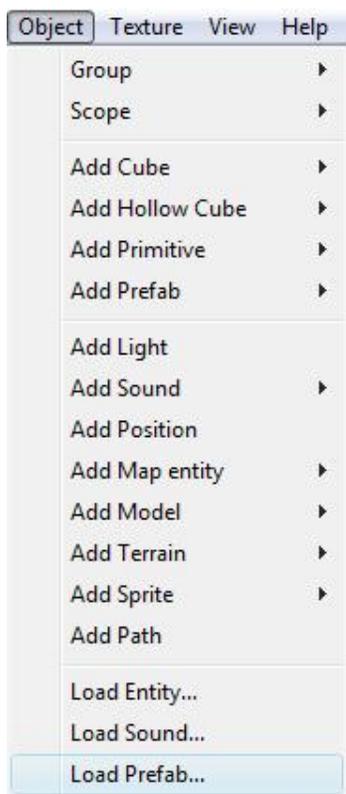


Figura 32 - Adicionando prefabs.

4.1.10. Sky-cube

Para que o ambiente tenha um fundo com imagem, como um céu com horizonte, é necessário criar um bloco oco, que deve abranger todo o cenário. Após isso, deve ser relacionada a este bloco a imagem do Sky Cube. Trata-se de uma imagem específica para isto, que deve se encontrar no diretório do projeto. Este arquivo deve estar nomeado como "NomeDoArquivo+6.tga". O +6 é reconhecido pelo compilador do 3DSG, que passa a reconhecer esta imagem como um cubo desdobrado de seis

lados (*Sky-cube*). Este arquivo tem algumas peculiaridades. Sua largura deve ser 6 vezes maior do que a altura. Cada quadro deve ter o mesmo tamanho em altura e largura. A disposição das imagens do cubo deve ser posicionada na seguinte sequência: atrás, à direita, à frente, à esquerda, para baixo, e para cima, da esquerda para a direita. A Figura 33 mostra um exemplo da imagem de um *Sky-cube*.



Figura 33 - Imagem de um *Sky-cube*.

Após criar o bloco oco, deve-se acessar as propriedades deste objeto, clicando sobre ele com o botão direito do mouse, em seguida selecionando a opção *Properties*. Na janela que surgirá, é necessário selecionar a opção *Sky* que se encontra na aba *<blocks>*, conforme mostra a Figura 34.

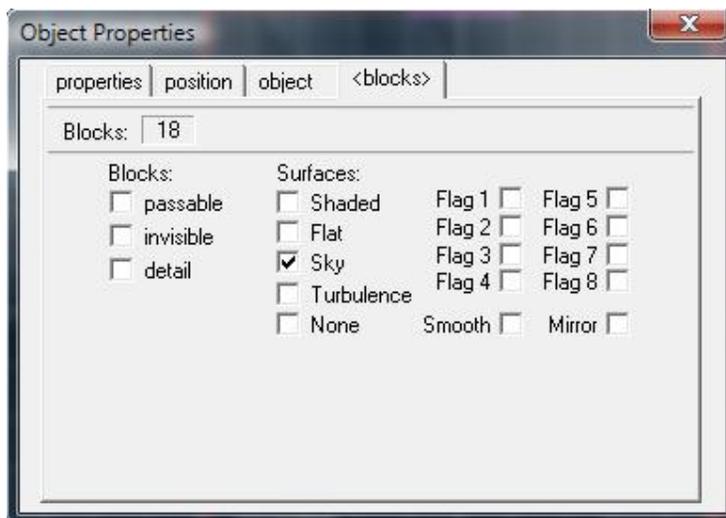


Figura 34 - Propriedades do Objeto.

Com isto, o cubo já estará apto a receber a imagem de fundo do *Sky Cube*. Resta agora apenas configurar o *script* do cenário, adicionando uma função para que isto aconteça.

Para abrir o *script*, deve ser acessada no painel ao lado esquerdo da tela do WED a aba *Resources*. Neste local constam todos os arquivos de *scripts* pertencentes ao projeto. O primeiro item da lista é o *script* principal, nomeado como NomeDoProjeto.wdl. É nele que consta o método *main()*, além dos métodos de inicialização na execução do sistema. Para abri-lo no SED (Script Editor), basta acessá-lo com um duplo-clique, como mostra a Figura 35.

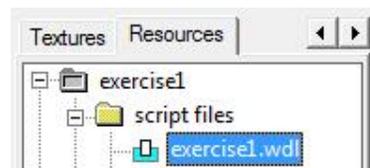


Figura 35 - Acessando Script pelo WED.

Na janela que surgirá estará constando o principal código fonte para o funcionamento do cenário. O código para reconhecimento do *Sky Cube* deve ser adicionado antes da função *main()*.

```

Sky cubesky {
    Type=<NomeDoArquivo+6.tga>;
    Layer = 1;
    Material = mat_sky;
    Flags = cube, visible;
}
  
```

4.2. DESENVOLVIMENTO DO AMBIENTE VIRTUAL 3D

Toda a parte de implementação do sistema foi desenvolvida usando o software 3D Studio Game. Devido à quantidade de trabalho a ser desenvolvida dentro do curto

prazo de tempo, foi extraído o máximo dos recursos já existentes, para não perder tempo tendo que refazê-los. Isto se refere principalmente aos modelos. Estes não foram desenvolvidos, foram usados modelos já existentes e disponíveis gratuitamente para desenvolvedores e estudantes.

Dentre estes, o MED foi o menos utilizado, visto que foram utilizados apenas modelos já fabricados para este trabalho.

4.2.1. Ambiente Virtual (cenário)

Serão utilizados os conceitos básicos definidos na seção 4.1 para o desenvolvimento do ambiente virtual 3D. Toda a criação deste cenário, mesmo sendo criado e visualizado graficamente, está relacionada com *scripts*. Para isso, é necessário antes de qualquer coisa, criar um arquivo de *script* para o projeto, em seguida mudar as configurações de luminosidade do ambientes sombras.

O objeto criado inicialmente foi um bloco oco, onde levou se em consideração o redimensionamento do tamanho da sala para atender os procedimentos básicos.

Após isso, foi adicionado texturas em cada parte deste bloco, o que dá a aparência realista ao cenário. A Figura 36 mostra este bloco oco texturizado, além da planta baixa do projeto, com as dimensões do bloco. Contêm ainda informação sobre as vistas das laterais do bloco.

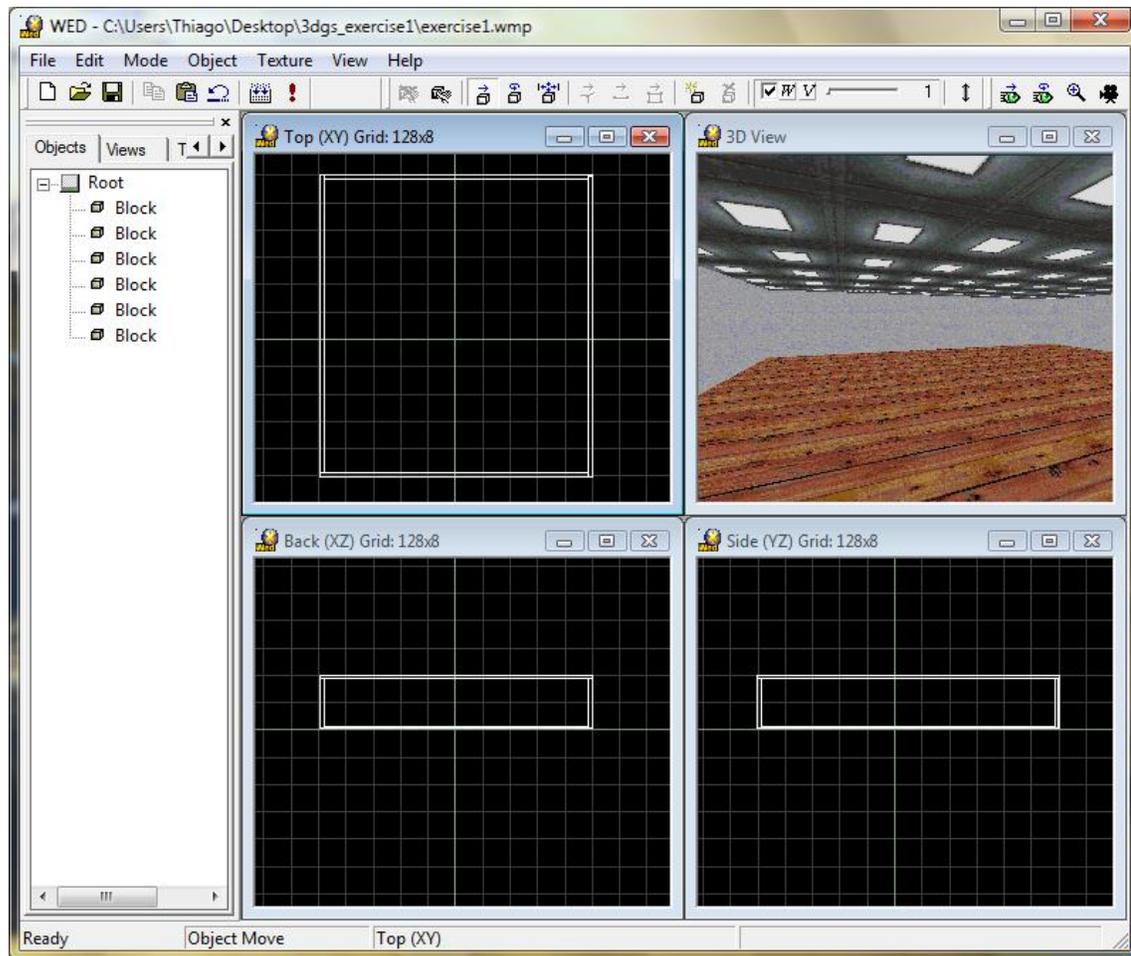


Figura 36 - Cenário com texturas.

A criação das portas e janelas no ambiente virtual foi feita utilizando o comando de subtração de objetos. Após a execução deste comando obtêm se a adição de portas e janelas no cenário, como mostra a Figura 37.

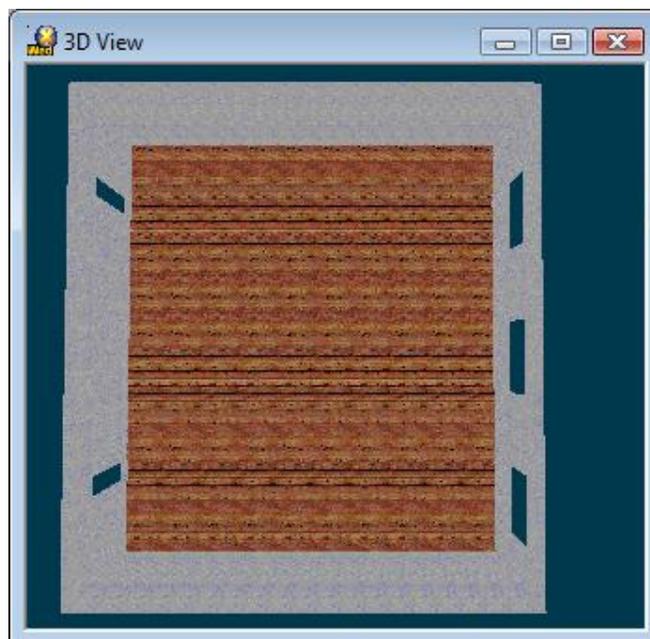


Figura 37 - Adicionando portas e janelas.

O cenário foi iluminado nos principais locais, como dentro da sala e ao seu redor. A luz do sol neste ambiente foi configurada para um clima de noite, para que as luzes do cenário fiquem mais visíveis. As configurações de iluminação do ambiente e a criação de luzes foram feitas de acordo com a descrição da seção 4.1.

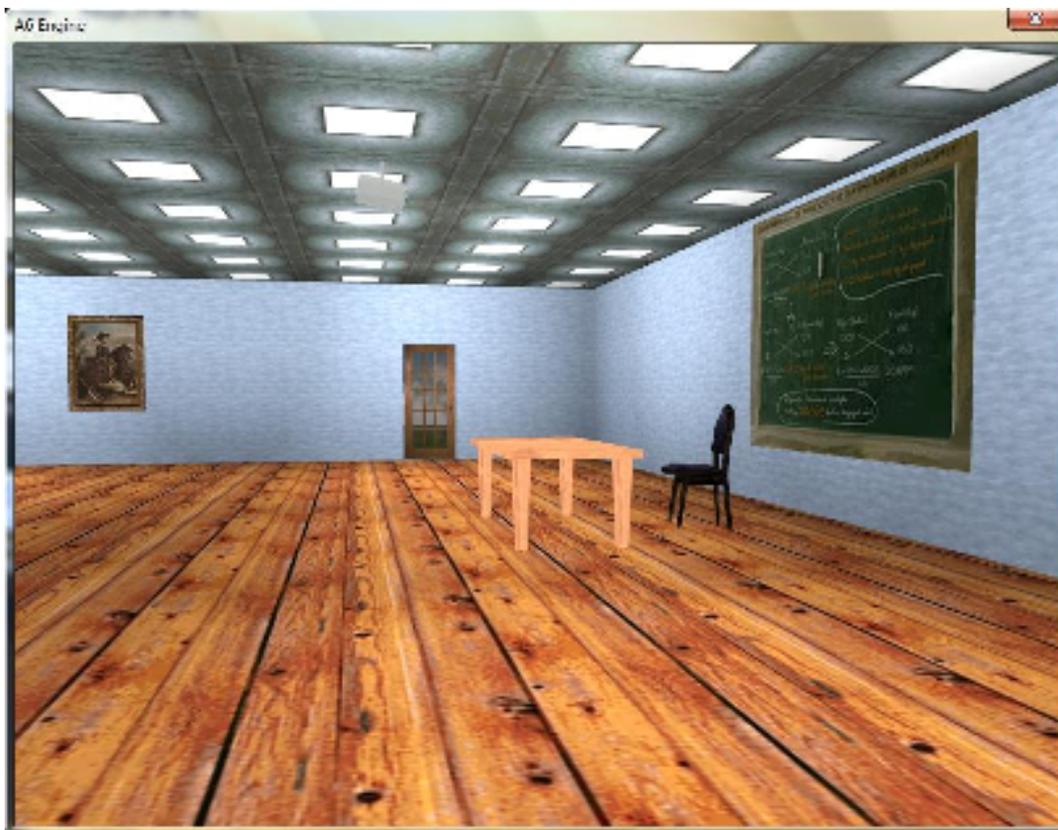


Figura 38 - Ambiente Virtual interno.

A Figura 38 mostra o ambiente virtual externo com os modelos e *sprites* adicionados.

Os *Sprites* são imagens em 2D e também foram adicionadas no cenário. Os *Sprites* adicionados foram as cortinas, lousa, quadros nas paredes e árvores. Estas imagens são simples e estão em formato *.BMP ou *.PCX.

Os objetos mesa e cadeira que se encontram frente à lousa são um dos modelos adicionados no ambiente. Conforme já foi comentado, neste trabalho não foi desperdiçado tempo com a modelagem de objetos, visto que estes já existem e encontram-se gratuitamente em sites relacionados ao assunto.



Figura 39 - Ambiente Virtual externo.

Os objetos como mesa, cadeira, bancos e lata de lixo são modelos prontos e foram importados para o cenário. As explicações de como adicionar modelos encontram-se na seção 4.1.

Um dos principais modelos deste trabalho é o *avatar* (personagem controlado pelo usuário). A Figura 40 mostra o *avatar* visto de uma câmera externa.



Figura 40 - Ambiente Virtual com avatar.

Para que o cenário tenha um horizonte e um céu foi adicionado um *Sky-cube*. A Figura 41 representa o *Sky-cube* utilizado neste cenário.



Figura 41 - Sky-cube utilizado.

A Figura 42 mostra o *Sky-cube* aplicado no ambiente virtual.

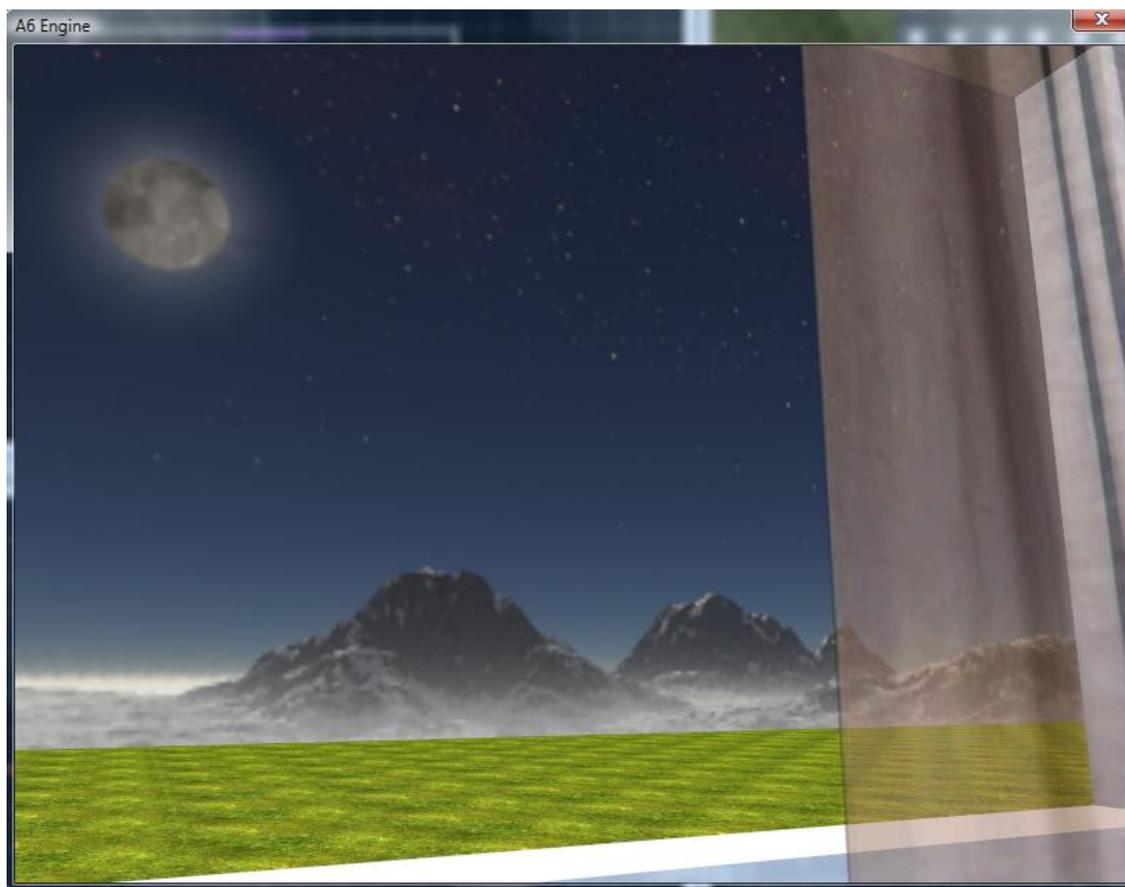


Figura 42 - Sky-cube no ambiente virtual.

4.3. SCRIPTS

Será feita uma breve explicação a respeito de uma das maiores dificuldades encontrada neste trabalho. Com o cenário praticamente pronto, o próximo passo é adicionar um personagem ao cenário. Como já foi visto anteriormente, não há segredos em relação a isto. A maior dificuldade é assimilar os movimentos deste personagem.

A quantidade de *scripts* que devem ser escritos para que o personagem se movimente para frente, trás, direita e esquerda a partir das teclas direcionais do teclado é muito grande. Isto também vale para o trajeto da câmera seguindo o

personagem por onde ele andar e os modos de visualização em primeira e terceira pessoa. Outra coisa importante a ser considerada é a detecção de colisão com paredes e outros objetos. Tudo isso requer uma quantidade muito grande de *scripts*.

Quando for pressionada a tecla cima, o personagem deve caminhar para frente. Para que este efeito seja o mais próximo do real, devem-se levar em consideração fatores da física, tais como Aceleração, Inércia e Fricção.

Se o jogador fosse caminhar para frente a uma velocidade constante, a fórmula seria $2.ds = v*dt$.

- v=Velocidade em quants por tick
- ds = Distância em quants
- dt = Tempo em ticks

Uma observação importante que deve ser considerada é que a unidade de medida neste mundo virtual é em *quants*, que vale por volta de uma polegada, e o tempo é medido em *ticks*, que é por volta de 1/16 segundos.

No entanto, este corpo não pode parar exatamente no momento em que o jogador soltar a tecla cima, devido à força da inércia. Quanto maior a massa do corpo, menor será a desaceleração (supondo que seja uma força constante), o que alteraria a fórmula para $dv = a*dt$, sendo $a = F/m$.

- dv = Mudança de velocidade
- a = Mudança de velocidade por aceleração
- F = Força
- m = Massa

No cenário devem ser considerados três tipos de forças.

- **Força de impulso:** a força aplicada quando o usuário pressiona uma tecla para que o personagem se movimente. ($F = p * m$).

F = força

p = impulso

m = massa

- **Força do vento:** pode ser uma corrente que leva o personagem em uma determinada direção, ou a gravidade puxando-o para baixo. Esta força é proporcional a massa do personagem. ($D = d*m$).

D = vento

d = distância

m = massa

- **Força de fricção:** esta força tenta reduzir a velocidade continuamente, de acordo com a massa do jogador sobre o tipo de chão. ($R = -f*m*v$).

R = Força de fricção

f = Fator de fricção (superfície ou chão)

v = velocidade

m = massa

Todos estes três tipos de força (F , D e R) acrescentam-se para mudar a velocidade do personagem.

$$dv = a * dt = (F + D + R) / m * dt = (p + d - f * v) * dt$$

Assim, dv deve ser adicionado à velocidade a cada frame. p , d e f são fatores de impulso, vento e fricção, e dt é o tempo pelo qual a aceleração mudou de velocidade.

Esta última fórmula pode ser traduzida para uma função correspondente:

```

var force[3];
var dist[3];
action move_me
{
while (1) {
    force.PAN = -10 * KEY_FORCE.X; // calcula força de giro
    MY.SKILL14 = TIME*force.PAN + max(1-TIME*0.7,0)*MY.SKILL14;
    // velocidade de giro
    MY.PAN += TIME * MY.SKILL14; // gira o jogador
    force.X = 10 * KEY_FORCE.Y; // calcula a força de translação
    MY.SKILL11 = TIME*force.X + max(1-TIME*0.7,0)*MY.SKILL11;
    // calcula velocidade
    dist.X = TIME * MY.SKILL11; // distância para cobrir
    dist.Y = 0;
    dist.Z = 0;
    MOVE(MY,dist,nullvector); // move o jogador
    move_view(); // move a câmera
    wait(1);
    }
}

```

Este é o *script* responsável pelo movimento do personagem, ainda desconsiderando a força do vento.

Entra aqui uma das maiores vantagens em se usar um *engine* como o 3D *Studio Game*. Existem duas maneiras de fazer este movimento: escrevendo manualmente a função descrita acima, ou simplesmente associando ao personagem funções pré-definidas. É isso mesmo, funções como esta já vem prontas junto com a instalação do *engine*, onde todo o embasamento físico e matemático já está implementado, bastando para o desenvolvedor apenas o trabalho de adicioná-las ao projeto e associar a ação a um objeto do mapa, no caso o personagem.

4.3.1. Movimentos e colisões

Para adicionar o *script* que trata o gerenciamento das colisões entre os objetos e a movimentação do personagem, deve-se acessar o *menu File/ Add Script*, conforme a Figura 43.

Na janela que surgirá, deve ser escolhido o arquivo de *script* plBiped01.wdl, clicando em seguida no botão ok. Feito isso, este *script* será adicionado ao projeto juntamente com os *scripts* requisitados por ele. Todos os *scripts* adicionados podem ser visualizados na aba *Resources*, ao lado esquerdo da janela do WED, como já foi comentado.

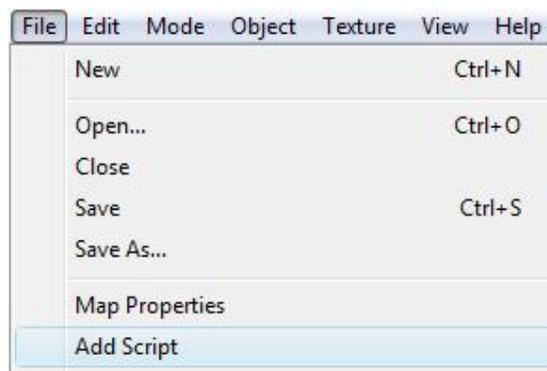


Figura 43 - Adicionando script.

4.3.2. Câmera em primeira pessoa

Da mesma maneira, deve-se adicionar o arquivo de *script* camera1stPerson01.wdl, *script* esta responsável pela câmera em primeira pessoa (este tipo de câmera simula a visão do personagem, não podendo assim visualizar o próprio corpo).

Uma vez adicionados estes *scripts* no projeto, é necessário adicionar um modelo de personagem no cenário, e associar isto a este modelo. Para fazer isso, é necessário selecionar o personagem, e então clicar sobre este com o botão direito do mouse, em seguida deve-se acessar a opção *Behavior*. Na janela seguinte, deve ser escolhida a opção plBiped01, conforme a Figura 44, em seguida clicar em ok para fechar esta janela.

Com isso, têm-se um personagem com as funcionalidades básicas de movimentação. Este personagem está configurado com os atributos do *script* camera1stPerson01.wdl, por isso sua visualização funcionará apenas como primeira pessoa. É possível agora adicionar uma visualização em terceira pessoa, onde o personagem é visto de fora do seu corpo.

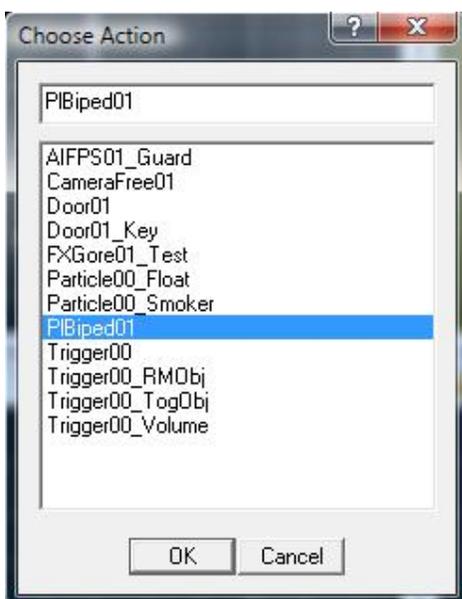


Figura 44 - Associando script ao personagem.

4.3.3. Câmera em terceira pessoa

Para adicionar o *script* que trata o gerenciamento de uma câmera em 3ª pessoa ao projeto, deve-se acessar o *menu File/ Add Script*. Na janela que surgirá, deve ser escolhido o arquivo de *script* camera3rd01.wdl.

Agora, após compilar e executar o projeto pode-se visualizar o personagem tanto em 1ª quanto em 3ª pessoa, utilizando a tecla F7 para alterar o modo da câmera.

4.3.4. Câmera livre

É possível agora adicionar uma câmera livre alternativa. Para isto é necessário adicionar o *script* cameraFree01.wdl. Esta câmera não poderá ser associada ao personagem, uma vez que ele já está associado ao *script* plBiped01.wdl, ele já não é mais um corpo livre no espaço, pois já está sendo afetado pelo seu peso, massa e as leis da física. Para isto, deve-se criar um objeto qualquer para associar a este novo *script*. Pode-se adicionar ao cenário o modelo Ball.mdl, que se encontra no diretório "Gstudio6\template_6\models". O passo seguinte é associar a este objeto "Bola" o *script* responsável pela câmera livre, selecionando o objeto e clicando sobre ele com o botão direito do mouse, acessando assim a opção *Behavior*, e selecionando a opção cameraFree01, conforme a Figura 45. Com isto, ao compilar e executar o programa, pressionando a tecla F7 será possível alternar entre os modos de câmeras (1ª pessoa, 3ª pessoa e câmera livre). A câmera livre é nada mais do que a visão da bola, podendo esta ser movimentada para qualquer direção, visto que ela não está submetida às leis da física, exceto as colisões. Este modo de câmera pode ser movimentado através das teclas I, J, K, L, U e O.

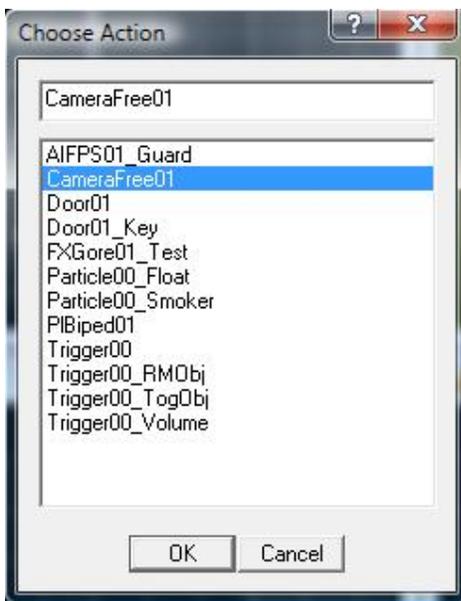


Figura 45 - Adicionando script de Câmera Livre.

4.3.5. Abrir porta

Ao se aproximar de uma porta com o *avatar*, a porta deve se abrir quando o usuário pressionar a tecla *enter*. Para a realização deste procedimento foi desenvolvida uma ação específica para este propósito.

Para isto, deve ser aberto o SED, e adicionado em um novo arquivo o *script* a seguir:

```
var indicator = 0;
```

```
var my_pos[3];
```

```
var my_angle[3];
```

```
function scan_me()
```

```
{
```

```

my_pos.X = CAMERA.X;
my_pos.Y = CAMERA.Y;
my_pos.Z = CAMERA.Z;
my_angle.PAN = CAMERA.PAN;
my_angle.TILT = CAMERA.TILT;
temp.PAN = 120;
temp.TILT = 180;
temp.Z = 200; // alcance do scan – 200 quants
indicator = 1; // isto é para a abertura
SCAN(my_pos,my_angle,temp);
}

DEFINE _COUNTER SKILL25; // usa um nome significativa para SKILL25
    SOUND Abre <door_op.wav>;
    SOUND Fecha <door_cl.wav>;

function door_event()
{
    if (indicator != 1) { return; } // deve ser o tipo certo de scan
    if (MY._COUNTER <= 0)
    { // se a porta estiver fechada, abre ela
        PLAY_SOUND, Abre, 100; // the number sets volume
        wait(1);
        while (MY._COUNTER < 90)
        {
            MY.PAN -= 6*TIME; // gira à direita
            MY._COUNTER += 6*TIME;
            wait(1);
        }
        MY.PAN += MY._COUNTER-90; // corrige o excedente
        MY._COUNTER = 90;
    } else { // caso contrário feche ela
        PLAY_SOUND, Fecha, 100; // the number sets volume
        wait(1);
    }
}

```

```

while (MY._COUNTER > 0) {
    MY.PAN += 6*TIME; // gire à esquerda
    MY._COUNTER -= 6*TIME;
    wait(1);
}
MY.PAN += MY._COUNTER; // corrije o excedente
MY._COUNTER = 0;
}
}

action porta
{
ON_ENTER = scan_me;
    MY.event = door_event;
    MY.enable_scan = ON; // faz porta sensível para o scaneamento
    MY._COUNTER = 0; // contador de grau, independente do ângulo inicial
}

```

Este código está comentado em suas partes mais importantes para que uma pessoa com conhecimentos de algoritmo seja capaz de compreender seu funcionamento.

A ação “porta” é a que deve ser atribuída ao objeto modelo porta no cenário, conforme a Figura 46. No escopo desta ação, o comando “ON_ENTER = scan_me” faz com que, ao pressionar a tecla *enter*, seja acessada a função “scan_me()” definida no começo do *script*. Na função “scan_me()”, o comando “temp.Z = 200” é o que delimita a distância mínima entre o *avatar* e a porta, para que esta se abra ao pressionar *enter*. Caso este requisito seja satisfeito, a variável “indicator” recebe o valor 1.

Com o comando “MY.event = door_event” na ação “porta”, é chamada a função “door_event”, onde será verificado se a variável “indicator” é igual a 1. Se for, o restante do código fará com que a porta se abra ou feche, alterando o valor “MY.PAN”.

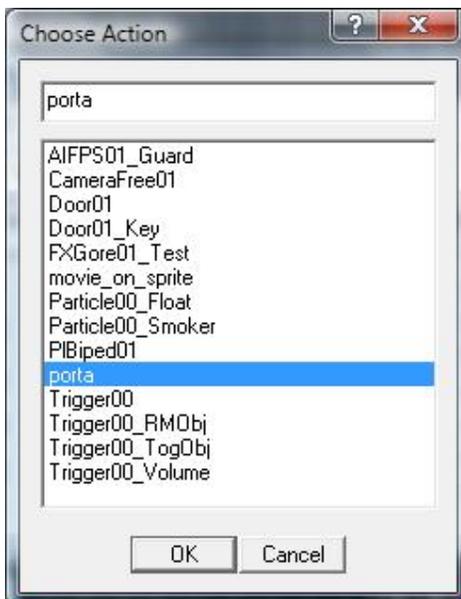


Figura 46 - Ação porta.

4.4. CRIAÇÃO DE UMA LOUSA VIRTUAL

A *engine* possibilita ainda a execução de vídeos de diversas formas. É possível adicionar vídeo em um modelo, em um *sprite*, além de um filme entre as fases de um jogo em tela cheia (*cinematics*).

Uma observação importante sobre *action* é que ela é como uma função, mas no seu código, ao invés de *function* é utilizada a palavra *action*. A principal diferença entre *action* e *function* é que a primeira pode ser atribuída a objetos modelos ou *sprites*.

Para a criação da lousa virtual, foi utilizada uma *action* contendo os códigos responsáveis pela execução do vídeo, e depois esta *action* foi atribuída ao objeto lousa.

A Figura 47 mostra uma tele-aula sendo exibida, utilizando o *sprite* “lousa” como um telão.

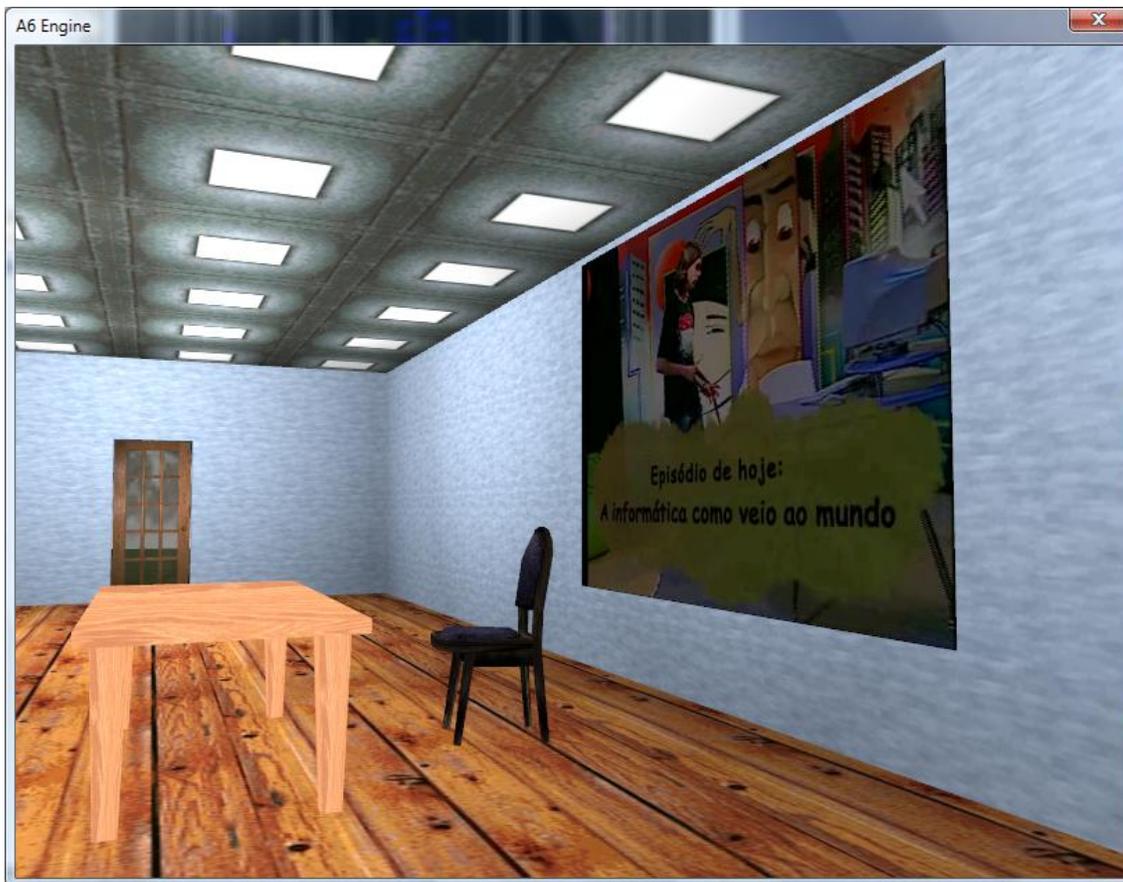


Figura 47 - Criação de uma lousa virtual.

Para criar esta *action* é necessário acessar o SED (Script Editor) e criar um novo. Feito isto, deve ser digitado o código a seguir:

```
action movie_on_sprite
{
    var x=0;
    while (key_i == 0) {wait (1);} // pressione "I" para tocar o video
    media_play ("filme.wmv",bmap_for_entity (my, 0), 100);
    while (key_o == 0) {wait (1);} // pressione "O" para parar o video
    x = media_handle;
```

```

media_pause (x);
while (key_p == 0) {wait (1);} // pressione "P" para parar o video
media_start(x); // continua a execução do vídeo
x=0;
}

```

Finalmente, esta *action* deve ser salva no mesmo diretório que o projeto, com o nome `movie_on_sprite.wdl`.

Este código deve ser inicializado com a palavra “*action*” seguida do nome da ação.

Logo depois é declarada uma variável *x*, recebendo o valor inicial zero. O primeiro *loop* fará com que o sistema fique eternamente esperando até que seja pressionada a tecla “I”. Uma vez que esta tecla for pressionada, o seguinte comando “`media_play ("filme.wmv", bmap_for_entity (my, 0), 100);`” fará com que o filme comece a ser executado. “*Media_play*” é um comando do próprio 3DSG, que recebe como parâmetro primeiramente o filme a ser reproduzidos, a entidade do mapa na qual o filme deve ser exibido, e o volume do áudio.

Observação: O comando “`wait (1);`” é o que impede que o sistema trave enquanto espera por alguma ação do usuário. Como o comando descrito acima se trata de um *loop* infinito, esta função dispara um tempo intermediário onde são processadas as demais ações, funcionando como uma *thread*.

O próximo *loop* fará com que o sistema espere até que seja pressionada a tecla “O”. Neste momento a variável *x* recebe o valor de “`media_handle`”, que é uma variável da própria *engine*, responsável por guardar a posição atual do vídeo em *frames*.

O comando “`media_pause (x);`” fará com que o vídeo seja parado na posição *x*.

Por fim, o último *loop* aguardará até que seja pressionada a tecla “P”. Após isso, o comando “`media_start(x);`” fará com que o vídeo continue a ser executado a partir da posição contida na variável *x*.

Após a criação da ação responsável por executar um vídeo em um *sprite*, basta selecionar o *sprite* desejado no WED e atribuir a ele a ação `movie_on_sprite`.

CAPÍTULO 5

CONCLUSÃO

Neste trabalho, foi apresentado um conceito de como aplicar a Realidade Virtual no ensino a distância. Foi concretizado a implementação de um ambiente em 3D, constituído de um plano (chão), céu e árvores, além das construções como sala e postes de luz. Conta também com objetos como mesa, bancos, lata de lixo, cadeira, projetor, lousa, quadros, cortinas, portas, janelas, relógio de parede, armários, etc. Todo o ambiente é adequadamente iluminado. Outra importante propriedade do cenário é o personagem, que vem a ser controlado pelo usuário, tendo em sua implementação todo um tratamento de movimentos, pulo, e colisões, além da visão em primeira ou terceira pessoa. As portas foram dotadas com a habilidade de abrir e fechar com um comando do usuário, ao chegar com seu *avatar* próximo a ela. Outra importante característica para o EAD é a lousa virtual. Através dela é possível rodar um vídeo que futuramente, poderá ser assistido simultaneamente por todos os usuários conectados a este ambiente.

Partindo de todo este princípio, muita coisa pode se agregar a este projeto. Além da comunicação entre várias máquinas, cabe desenvolver um chat, onde estes personagens possam se comunicar através de textos, com base nas regras de interação entre os personagens. Cabe também ressaltar, após esta etapa, uma possível implementação de uma forma de comunicação entre os usuários via áudio, através de microfones.

Outra aplicabilidade seria utilizar a lousa virtual não só para transmitir vídeos, mas também para que o professor possa se manifestar através dela para os alunos, via webcam. O mural de recados poderá ser um NPC (Non Player Character), ou seja, um *avatar* que não é controlado por ninguém, servindo apenas para transmitir mensagens, ou responder dúvidas mais frequentes.

Por fim, conclui-se que os problemas resolvidos neste trabalho (cenário, personagens, objetos, lousa virtual) foram de primordial importância para o

desenvolvimento tecnológico na área de ensino utilizando as técnicas de Realidade Virtual.

REFERÊNCIAS BIBLIOGRÁFICAS

BEGAULT, D.R. - **3-D Sound for Virtual Reality and Multimedia**, Academic Press, Cambridge, MA, 1994.

BUGATTI, Idelberto. **O projeto revir**. URL <<http://www.dc.ufscar.br/~grv/projetos.htm>> Acessado em 15 de junho de 2008.

CRUZ-NEIRA, C. et al. - **The CAVE Audio Visual Experience Automatic Virtual Environment**, Communication of the ACM, 35(6):64-72, June 1992.

DEDE, Chis e Loftin, Bowen; **Projeto Science Space**. URL <<http://www.virtual.gmu.edu/> acessado em 18 de junho de 2008>.

Game Studio. Disponível em <<http://www.3dgamestudio.com/>> Acesso em 05/06/2008.

GAVA, Antonio Carlos. **Educação a Distância**. 2002. 79 p. Dissertação (mestrado) – Departamento Faculdades Integradas Teresa Martin, São Paulo, SP, 2002.

GRADECKI, J. **Kit de montagem da Realidade Virtual**. São Paulo, Berkeley, 1994.

HANNA, D. E. et al. - **Practical tips for teaching online groups : essentials of web-based education**. Madison: Atwood Publishing, 2000.

HIT Lab. ...a virtual classroom for teaching space science and exploration. URL <<http://www.hitl.washington.edu/projects/starship/>> . Acessado em 17 de junho de 2008.

HODGES, Larry; Wineman , Jean. **Projeto Gorila Virtual** URL <<http://murmur.arch.gatech.edu/phd/RESEARCH/Gorilla.htm>> . Acessado em 30 de novembro de 2008.

HOSN & COSTA, Raid Nadim Aboul, Claudia Pereira Jackson Costa, **Capítulo III: Formas de Realidade Virtual Imersivas**. Disponível em <<http://ca.geocities.com/raidhosn/cap03.htm#realidade>> . Acesso em 26/06/2008.

JACOBSON, L. **Realidade Virtual em casa**. Rio de Janeiro, Berkeley, 1994.

KIRNER, C.P. - **Introdução à Realidade Virtual**. Disponível em: <<http://www.dcc.ufscar.br/~grv>>. Acesso em 27/06/2008.

KIRNER, Claudio. **Projeto Professor Virtual**. URL <<http://www.dc.ufscar.br/~grv/pvirtual.htm>> . Acessado em 15 de junho de 2008.

MORAN, José Manuel. **O que é educação à distância**. Disponível em <<http://www.eca.usp.br/prof/moran/dist.htm>> . Acesso em 25/06/2008.

MORIE, J.F. - **Inspiring the Future: Merging Mass Communication, Art, Entertainment and Virtual environments**, *Computer Graphics*, 28(2):135-138, May 1994.

PIMENTEL, K. & TEIXEIRA, K. **Virtual reality - through the new looking glass**. 2.ed. New, 1995.

ROBERTSON, G.G. et al. - **Nonimmersive Virtual Reality**, *IEEE Computer*, Feb. 1993, pp. 81-83.

SILVA, Rogers Ferreira , SILVEIRA, Jorge Guedes, BALBINOT, Ricardo. Plataforma DeskEaD para Aplicações de Educação a Distância. In: **Boletim Bimestral**. RNP – Rede Nacional de Ensino e Pesquisa, Jul 2003.

Universidade de Indiana, **Virtual Solar System**. URL <<http://vss.crlt.indiana.edu/>>. Acessado em 17 de junho de 2008.