3D BUILDER - EDITOR GRÁFICO PARA GERAÇÃO INTERATIVA DE AMBIENTES VIRTUAIS EM VRML[1]

3D BUILDER - A GRAPHICAL PUBLISHER FOR INTERACTIVE VIRTUAL ENVIRONMENT GENERATION IN VRML

Simone Ceolin[2]
Andre Zanki Cordenonsi[3]

RESUMO

Este trabalho apresenta o desenvolvimento do Editor Gráfico 3D Builder que disponibiliza para o usuário a construção de ambientes virtuais a partir de uma interface gráfica amigável. O editor possibilita a escolha de formas geométricas (cone, esfera, cilindro, cubo, disco, ponto, reta e caixas), que serão utilizadas para construir o ambiente que o usuário deseje. Cada objeto apresenta propriedades que podem ser modificadas de acordo com as necessidades do usuário. Após a construção do ambiente, este poderá ser gravado no formato .wrl, para a visualização em *browsers* que trabalham com a linguagem VRML. A construção do Editor foi realizada usando a linguagem de programação *Delphi* e componentes de *OpenGl* .

Palavras-chave: VRML, realidade virtual, *openGl*.

RESUMO

This paper present the Graphical Publisher 3D Builder development, a friendly graphical interface to generate virtual environments. The publisher can work with a lot of geometric forms (cone, sphere, cylinder, cube, disc, point, straight line and boxes), which can be used to build the environment by the user. Each object have some properties that can be modified by the user. The final virtual environment can be saved in *wrl* format. This format can be visualized in any browser which supports the VRML language. The publisher interface was build using the Delphi language and OpenGL components.

Key words: VRML, virtual reality, openGl.

[1] Trabalho Final de Graduação.
[2] Curso de Sistemas de Informação. UNIFRA.
[3] Orientador

INTRODUÇÃO

A Realidade Virtual (RV) pode ser considerada, de uma maneira simplificada, como a forma mais avançada de interface usuário/computador até agora disponível. Ela é capaz de dar ao ser humano condições de vivenciar uma realidade que não existe. Com aplicação na maioria das áreas do conhecimento e com um grande investimento das indústrias de equipamentos, sistemas e dispositivos de entrada e saída especiais, a realidade virtual vem experimentando um desenvolvimento acelerado nos últimos anos, indicando perspectivas bastante promissoras para os diversos segmentos vinculados com a área. A interface com realidade virtual envolve um controle tridimensional altamente interativo de processos computacionais. O usuário entra no espaço virtual das aplicações e visualiza, manipula e explora os dados da aplicação em tempo real, usando todos seus sentidos, particularmente os movimentos naturais tridimensionais do corpo.

A grande vantagem desse tipo de interface é que o conhecimento intuitivo do usuário a respeito do mundo físico pode ser transferido para manipular o mundo virtual. Para suportar esse tipo de interação, o usuário pode utilizar dispositivos não convencionais, como capacetes de visualização e controle, além das luvas. Estes dispositivos dão ao usuário a impressão de que a aplicação funciona no ambiente tridimensional real, permitindo a exploração do ambiente e a manipulação natural dos objetos com o uso das mãos, por exemplo, para apontar, pegar e realizar outras ações.

Segundo (ZAGO, 1998), nos últimos anos, surgiu a idéia de levar a Realidade Virtual para a Internet. Dessa idéia surgiu a VRML, que é a abreviação de *Virtual Reality Modeling Language*, ou Linguagem para Modelagem em Realidade Virtual. A VRML é uma linguagem independente de plataforma que permite a criação de ambientes virtuais, nos quais se pode navegar e visualizar objetos por ângulos diferentes e até interagir com eles. Atualmente, graças ao crescente investimento de grandes empresas, a VRML é o padrão para desenvolvimento de aplicações de realidade virtual multiusuário na Internet. O objetivo da linguagem é levar a realidade virtual para o usuário comum, sem a necessidade de equipamentos especiais, por meio da www. Com o rápido avanço da tecnologia, os computadores pessoais estão cada vez mais rápidos e poderosos, o que permite a utilização da RV por usuários comuns.

A linguagem VRML possui toda a estrutura necessária para o desenvolvimento de aplicações de realidade virtual. A primeira versão (VRML 1.0) não possibilitava muita interação do usuário com o mundo virtual. Versões mais recentes acrescentaram características como animação, som, movimentos de objetos e interação entre usuários. A última versão disponível é

denominada *Moving Worlds* VRML 2.0. Por ser independente de plataforma, uma aplicação VRML roda em qualquer máquina que tenha um navegador VRML, ou seja, é uma aplicação que pode rodar em um Computador Pessoal, numa *Silicon Graphics*, num *Macintosh*, sem nenhuma alteração.

No entanto, a criação de ambientes VRML é prejudicada pelo fato de tratar-se de uma linguagem de programação. Logo, a construção de um mundo virtual envolve a edição de um arquivo texto com os comandos necessários para que o *browser* possa interpretar e construir o ambiente. O código VRML é um texto que descreve o ambiente e os eventos que podem estar associados a este como, por exemplo, quando o usuário toca um interruptor e uma luz acende. Não é necessário usar nenhum compilador, pois o *browser* se encarrega de interpretar o código e gerar o ambiente descrito por ele, conforme (JAMSA, 1999)

O *browser* carrega o arquivo texto contendo a descrição do ambiente, monta-o e carrega suas texturas e passa o comando para o usuário. De acordo com a movimentação do mouse pelo usuário, o *browser* move o ambiente. Logo, o usuário pode navegar pelo ambiente de forma livre, olhar por qualquer ângulo e posição. Como o arquivo apenas descreve o ambiente, o *browser* fica encarregado de gerar as imagens em tempo-real durante a navegação.

Percebe-se, então, que a maior dificuldade na construção de imagens virtuais em VRML advém do fato dos elementos gráficos serem definidos por meio de comandos. É necessário o aprendizado de uma linguagem nova para a geração dos ambientes virtuais. Portanto, torna-se interessante a construção de um sistema que permita a criação e manipulação de ambientes virtuais de forma gráfica, ocultando do usuário a complexidade da sintaxe da Linguagem VRML.

REVISÃO BIBLIOGRÁFICA

VIRTUAL REALITY MODELING LANGUAGE

Segundo (CASTIER, 1997), em 1992, foi disponibilizada a biblioteca gráfica Inventor da *Silicon Graphics*, que permitia aos programadores desenvolverem, com rapidez, programas gráficos 3D interativos, baseados nos conceitos de cena e na descrição de objetos. Em maio de 1994, em Genebra, na primeira Conferência da *World Wide Web*, o grupo de discussão de realidade virtual decidiu desenvolver uma linguagem de descrição de cena que pudesse ser usada na *Web*. A história da VRML inicia muito antes da apresentação da proposta de sua primeira especificação. A linguagem representa o ápice do desenvolvimento da computação gráfica em 3D, cujo objetivo principal sempre foi o de conseguir apresentar cenários que realmente parecessem reais numa tela de computador.

A Virtual Reality Modeling Language (VRML), ou Linguagem para Modelagem em Realidade Virtual, é uma linguagem independente de plataforma que permite a criação de ambientes virtuais em que se pode navegar, visualizar objetos por ângulos diferentes e até interagir com eles. O objetivo da linguagem é levar a realidade virtual para o usuário comum, por meio da internet.

Para (DUARTE, 1998), a Linguagem para a Modelagem em Realidade Virtual tem obtido crescente aceitação como tecnologia padrão da Web para exibição de conteúdo gráfico 3D e tem se tornado um meio rico de expressão de idéias na *Web*, pois o mundo VRML é interativo e pode conter animações e sons. É o resultado de um processo de discussão e cooperação aberto, sintetizando o conhecimento e experiência de milhares de pessoas. A VRML é simples e acessível e representa o primeiro passo a caminho da *Web 3D* e interativa. Ela pode ser aplicada na área da ciência (Medicina, Geociência, Engenharia, Arquitetura), entretenimento (jogos, animação), negócios (publicidade, manuais de produtos), artes e ensino.

Um dos principais problemas da VRML foi a falta de hardware barato que oferecesse desempenho razoável para explorar os mundos virtuais. No entanto, fabricantes de computadores têm sentido o potencial de aplicações 3D e vêm oferecendo hardware gráfico a preços menores. Com o rápido avanço da tecnologia, os computadores pessoais estão cada vez mais rápidos e poderosos e isto faz com que a realidade virtual deixe de ser objeto de estudo dos grandes centros de pesquisa e possa ser utilizada por usuários comuns.

A linguagem possui toda a estrutura necessária para o desenvolvimento de aplicações de realidade virtual, ou seja, é possível construir websites completos, com espaço e profundidade infinitos, em que os objetos tridimensionais proporcionam a interação necessária e o acesso às informações. Esses objetos podem ser ligados a arquivos de texto, áudio ou vídeo, arquivos HTML[1], ou mesmo a outros sites HTML e mundos VRML.

A primeira versão (VRML 1.0), com exceção de algumas extensões, não possibilitava muita interação do usuário com o mundo virtual, isto é, possuía comportamento interativo muito limitado. Os arquivos descreviam ambientes constituídos de objetos que podiam conter ligações (*hyperlinks*) com outros mundos, documentos HTML ou outros tipos MIME (*Multipurpose Internet Mail Extensions*) válidos, mas que não respondiam à ações do usuário em tempo real, como era de se esperar em um sistema realmente interativo.

---

[1] Uma página *Web* é composta de textos e comandos especiais (*tags*) de uma linguagem de programação chamada HTML, uma abreviação de *Hypertext Markup Language*. Esta linguagem é bastante simples e tem como finalidade básica formatar o texto exibido, além de criar ligações entre as páginas da *Web*, criando documentos com o conceito de hipertexto. (RAMALHO, 1999)

O primeiro browser aderente a esta especificação foi o *Cosmo Player* da SGI. Os *browsers* para VRML 1.0 não permitem a exibição de VRML 2.0. A maioria dos *browsers* para VRML 2.0 também exibem VRML 1.0, e, na maioria dos casos, são capazes também de exibir VRML 97.

O nó descreve o tipo do objeto, que pode ser uma esfera, um cilindro, uma transformação, uma definição de luz ou textura, etc. Também define as características de cada um, como tamanho de um cubo, diâmetro de uma esfera, intensidade da luz ambiente e cor. Existem dois tipos de nós: Nós de Agrupamento (Quadro 1) e Nós-Filhos (Quadro 2). Segundo (POLLO,1997), o nó é a construção fundamental de um arquivo VRML.

Quadro 1 - Nós de Agrupamento

Nós de agrupamento				
Anchor	Billboard	Collision	Group	Transform

Quadro 2 - Nós-filhos

Nós-filhos		
Anchor	LOD	Sound
Background	NavigationInfo	SpotLight
Billboard	NormalInterpolator	SphereSensor
Collision	OrientationInterpolator	Switch
ColorInterpolator	PlaneSensor	TimeSensor
CoordinateInterpolator	PointLight	TouchSensor
CylinderSensor	PositionInterpolator	Transform
DirectionalLight	ProximitySensor	Viewpoint
Fog	ScalarInterpolator	VisibilitySensor
Group	Script	WorldInfo
Inline	Shape	filhos de nós prototipados

Para (POLLO, 1997), o MIME é um protocolo padrão que define o tipo de arquivo que trafega pela Internet. Os browsers identificam este tipo de arquivo e os executam automaticamente.A especificação de um tipo MIME divide-se em duas partes (Quadro 3), que são separadas por uma barra. A primeira parte indica o tipo geral, como texto, áudio ou vídeo. A segunda parte indica o subtipo que, por sua vez, indica o formato exato do tipo geral.

As padronizações dos tipos MIME são feitas pela comunidade Internet. A definição para o padrão VRML é definido por *model/vrml*. Os tipos que

são temporários, ou são muito novos, têm a nomenclatura iniciada por *x-*. Antes da padronização do VRML, a definição para o tipo MIME era *x-world/x-vrml*. Por este motivo, os navegadores reconhecem tanto o padrão *model/vrml*, como o *x-world/x-vrml*, como sendo um MIME do tipo VRML.

Um dos grandes problemas da versão 1.0 é que as cenas geradas eram estáticas, ou seja, elas não mudam com o passar do tempo. A proposta era, propositalmente, simples, pois deveria servir apenas como um impulso original para o processo de aprimoramento da linguagem e para chamar a atenção da comunidade Internet para a nova tecnologia.

Quadro 3 - Exemplos de tipos MIME

Conteúdo do Tipo MIME	Descrição
text/plain	Texto sem formatação, como uma mensagem de e-mail.
text/html	Texto HTML, como os textos formatados que são apresentados nas páginas da Web.
Image/gif	Imagem no formato GIF, normalmente usada como figura nas páginas da Web.
video/mpeg	Vídeo codificado no formato MPEG.

Os objetivos da nova versão já são bem mais ambiciosos. A nova especificação traz inovações que prometem revolucionar a maneira como as pessoas acessam as informações na *Web*. Dentre elas, a adição de comportamento ao cenário 3D é, com certeza, a mais importante. A Linguagem VRML foi extendida para permitir que o ambiente mude e "evolua", tornando-se sensível às ações ou à simples presença de seus visitantes virtuais.

Na VRML 2.0, os objetos podem se mover pelo mundo virtual e responder a eventos temporais ou disparados pelo usuário. Além disso, é possível adicionar som espacial e filmes de vídeo à cena, tornando a experiência do observador muito mais realista. Os novos recursos de interação elevam o padrão a um nível um pouco mais próximo do Ciberespaço ideal. O que há de comum entre as duas versões é que, em essência, um arquivo VRML é simplesmente uma coleção de objetos arranjados (organizados) em uma certa ordem. Como na vida real, um objeto geralmente tem uma forma, uma superfície com certas propriedades (cor, brilho, suavidade, etc.) e uma posição no espaço tridimensional. Outros objetos VRML incluem sons, luzes e pontos de vista, que também têm uma posição no espaço 3D. Os objetos em VRML são chamados de nodes (ou nós).

Para criar um ambiente VRML, utiliza-se um editor de texto comum. O código é um texto que descreve o ambiente e os eventos que podem estar associados a este ambiente. Não é necessário usar nenhum compilador. O *browser* se encarrega de interpretar o código e gerar o ambiente descrito por ele, por meio da carga do arquivo texto que contém a descrição do ambiente, montando o ambiente e carregando as texturas. Deste passo em diante, o comando fica com o usuário. Para onde o usuário mover o *mouse*, o *browser* move o ambiente. O usuário pode, então, navegar pelo ambiente de forma livre e olhá-lo por qualquer ângulo e posição. Como o arquivo apenas descreve o ambiente, o *browser* fica encarregado de gerar as imagens em tempo-real durante a navegação.

EXEMPLO DE CÓDIGO EM VRML, (CASSAL, 1999)

Abaixo, é mostrado um exemplo de código em VRML que é interpretado pelo browser. A imagem resultante deste código pode ser observada na figura 2.

```
#VRML V2.0 utf8
# Relógio
Group{
  children [
    # Corpo do Relógio
    Shape {
      appearance Appearance {
        material Material {
        diffuseColor 1.0 0.0 0.0
# Definição de cor R=100% - G=0% - =0%
        }
      }
      geometry Box {
        size 0.10 0.10 0.03
# Tamanho X=0.10 - Y=0.10 - Z=0.03
      }
    }
    # Ponteiro Minutos
    Transform {
      translation -0.015 0.0 0.018
# Translação nos eixos X e Z
      children [
        Shape {
          appearance DEF Branco Appearance {
```

```
                material Material {
                  diffuseColor 1.0 1.0 1.0
# Definição de cor R=100% - G=100% - B=100%
                }
              }
              geometry Box {
                size 0.045 0.005 0.005
              }
            }
            # Ponteiro Hora
            Transform {
              translation 0.015 0.0075 0.005
              children [
                Shape {
                  appearance USE Branco
                  geometry Box {
                    size 0.005 0.030 0.005
                  }
                }
              ]
            }
          ]
        }
      ]
}
```

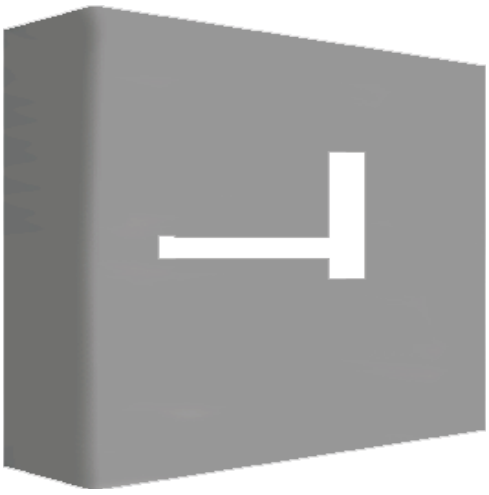

Figura 1 - Relógio

PADRÃO *OPENGL*

O padrão *OpenGl* é uma interface em *softwar*e para acessar o *hardware* gráfico. Ela consiste de, aproximadamente cento e cinqüenta funções usadas para definir objetos e operações para gerar aplicações em 3D. O *OpenGL* (*Open Graphics Library*) ou Biblioteca Gráfica Aberta foi construída para se manter independente da plataforma e do sistema operacional no qual a aplicação é rodada e, por isso, não há comandos para o gerenciamento de janelas e entrada de dados.

O *OpenGL* é uma API (*Aplication Programming Interface*) e, como tal, não provê comandos de alto nível para descrever objetos. Ao invés disso, o usuário deve construir o seu modelo a partir de primitivas geométricas (pontos, linhas e polígonos).

O usuário tem que construir os objetos a partir de entidades primitivas. Para resolver isso, existe a *OpenGL Utility Library* (GLU) ou Biblioteca de Utilidades da *OpenGL*, que provê várias funções como superfícies quadráticas, entre outras. A GLU existe em todas as implementações da *OpenGL*.

As operações que a *OpenGL* pode executar na geração de imagens são: construir formas complexas a partir de formas primitivas como pontos, linhas, polígonos, imagens e *bitmaps*; arranjar objetos no espaço tri-dimensional e permitir a escolha de um ponto do observação destes objetos; calcular a cor dos objetos a partir de definições do programa, condições das luzes, aplicações de texturas ou combinações destes três fatores; e converter a descrição matemática dos objetos e suas cores em pontos na tela. Esse último processo denomina-se de rasterização. Durante estas operações, o *OpenGL* pode executar outras tarefas como a eliminação de partes de objetos que são ocultadas por outros objetos.

OPENGL NO MERCADO

Atualmente, o *OpenGL,* cada vez mais se impondo como a interface padrão para o desenvolvimento de aplicações gráficas em 2D ou 3D. Como exemplos de aplicações criadas com o *OpenGL* temos poderosos pacotes de modelagem como *SoftImage* e *3DStudio*.

O *OpenGL* possui poderosas funções de descrição, mapeamento de textura, efeitos especiais e visualização de imagens. Muitas vezes chamado de "*linguagem assembly*" da computação gráfica, as aplicações criadas com o *OpenGL* garantem portabilidade, flexibilidade, funcionalidade e interatividade. Graças a isso, desde 1992, quando foi criado pela *Silicon Graphics*, tem sido um dos APIs mais usados pela indústria de *software*.

Suas aplicações incluem o mercado nas áreas de *CAD*, *Content Creation*, Energia, Desenvolvimento de Jogos, Interfacemento de Processos Industriais, Medicina, VRML e outros. A figura 3 apresenta um exemplo utilizando o padrão *OpenGl* e implementado na linguagem *Delphi* (JACOBS, 1998).

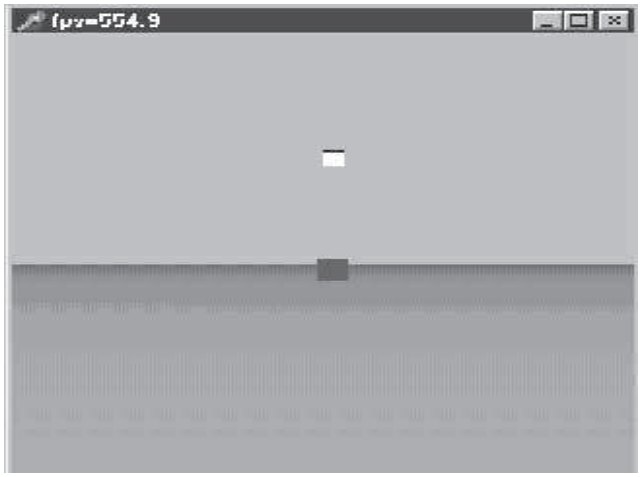

Figura 2 - Cubos feitos em *OPENGL* utilizando *Delphi*

METODOLOGIA

Inicialmente, foi realizada uma análise comparativa entre dois editores gráficos, encontrados na literatura, que tentam realizar a ponte entre a síntese de imagens e a Linguagem VRML, sem a necessidade de programação.

EDITOR GRÁFICO *RENDERSOFT*

O Editor *RenderSoft* (RENDERSOFT, 1999), cuja interface é apresentada na Figura 2, disponibiliza para o usuário a criação de figuras a partir dos objetos que ele apresenta (pirâmide, cubo, esfera, etc.). O usuário pode agrupar as figuras selecionadas, além de permitir a criação de animações. Depois de criar a figura, o editor permite salvar com extensão .wrl, em que o usuário pode visualizar a mesma no *browser* do VRML.

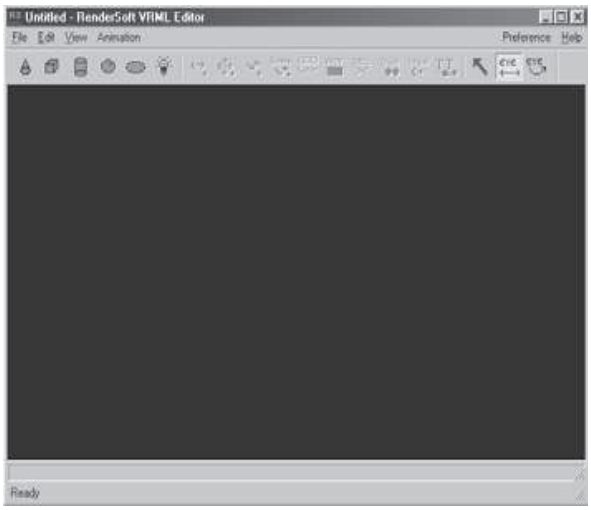

Figura 3 - *RenderSoft* VRML

EDITOR GRÁFICO ATSWORLDS

Segundo (SILVA, 1999), o *ATSWorlds* é um programa que visa à criação de objetos VRML a partir da técnica de *Sweep*. Ele foi desenvolvido como parte do trabalho de conclusão de André Tavares da Silva, sob orientação do professor Fernando Santos Osório, pela Unisinos - Universidade do Vale do Rio dos Sinos. Este trabalho tem por objetivo integrar técnicas de modelagem com a VRML. A técnica escolhida para realização desta integração foi a modelagem por *sweep*, por ser uma maneira natural e intuitiva para construir uma grande variedade de objetos, sendo muito utilizada em sistemas de modelagem de objetos tridimensionais.

Este programa pode ser executado tanto como aplicativo quanto *applet*. Como aplicativo, tem a vantagem de poder gravar os objetos gerados em disco, bem como ler e salvar os pontos usados para gerar os objetos. O aplicativo gera o arquivo .wrl, mas não o mostra. Para visualizar a figura, é necessário chamar o *browser*.

No entanto, como applet, o usuário pode visualizar os objetos gerados na mesma tela, sem a necessidade de ficar mudando de aplicativo cada vez que é criado um novo objeto ou realizada alguma alteração. Em contrapartida, o trabalho realizado não pode ser salvo em disco.

Para executar o *ATSWorlds* como aplicativo, basta ter o *Java* versão 1.1.x ou superior. Ao utilizar o *ATSWorlds* como *applet*, serão necessários alguns recursos adicionais. Ele utiliza uma classe *Java* que está disponível a partir da versão 1.2.2 para comunicação com o seu navegador.

DESCRIÇÃO DA FERRAMENTA PROPOSTA

O Sistema *3D Builder* é um editor gráfico que gera, por meio de recursos acionados por ícones e menus, arquivos VRML. Ele pode ser definido como uma ferramenta de construção de ambientes em Linguagem VRML para modelagem em Realidade Virtual, no qual o usuário interage com uma interface gráfica e obtém um arquivo final, sem a necessidade de programação. Desta forma, o editor possibilita uma forma mais agradável para o usuário que precisar construir e modificar ambientes virtuais, por meio da manipulação direta dos objetos tridimensionais em uma interface gráfica amigável.

O usuário interage com uma interface gráfica e obtém um arquivo VRML, sem a necessidade de programação. No sistema proposto, espera-se obter uma forma mais agradável para o desenvolvedor que necessita construir e modificar ambientes virtuais, pela manipulação direta dos objetos tridimensionais em uma interface gráfica amigável.

O editor permite a escolha de formas geométricas básicas, que serão utilizadas para construir o ambiente que o usuário deseje. Para tanto, são fornecidas diversas formas geométricas (cone, esfera, cilindro, cubo, disco, ponto, reta e box) que fornecem ao usuário ferramentas para que ele possa realizar o seu trabalho de forma satisfatória.

Para a definição das formas geométricas que fariam parte do editor, foi necessário realizar uma comparação entre os componentes do *OPENGL* e a VRML, pois o Editor somente pode disponibilizar objetos que a VRML aceita, para que seja feita a conversão a contento.

O usuário do Editor *3D Builder* (Figura 5) pode adicionar diversos objetos, modelando suas características por meio da seleção do objeto desejado. Para adicionar mais de um objeto no Editor, foi necessário criar vetores dinâmicos. Com isto, o usuário pode inserir quantos objetos quiser, porque o vetor não tem o tamanho prédeterminado, armazenando todos os objetos criados.

A partir da seleção, é mostrada ao usuário uma janela de propriedades, contendo todas as características primitivas do objeto (Figura 6). A partir delas, o usuário terá noção do objeto, o que torna mais fácil a troca das propriedades do mesmo. Cada forma geométrica possui suas propriedades, tais como: cor, textura, rotação, translação e escala.

Algumas propriedades das formas geométricas tiveram que ser modificadas, quando da conversão do padrão *OpenGl* para a linguagem VRML, tais como:

· a propriedade cor, no Editor, é definida pelo padrão do ambiente *Delphi*. Este padrão representa a cor como um único número em hexadecimal.

Na linguagem VRML, o padrão é o RGB (*red green blue*). Foi necessário converter os padrões de cores para a gravação do arquivo VRML;

· as texturas que podem ser utilizadas no padrão *OpenGl* são arquivos do tipo bitmap (bmp). Logo, o editor aceita apenas arquivos deste tipo. No entanto, a linguagem VRML trabalha com as extensões padrão da internet, ou seja, .jpg, .gif e .jnp. Para resolver este problema, foi realizado a transformação dos arquivos bitmap para jpeg. Depois da transformação, o novo arquivo com a textura em jpeg é criado automaticamente no diretório onde é gravado o arquivo VRML resultante do editor.

· O ângulo de visão das figuras foram invertidas em 90°, pois a câmera de visualização padrão no VRML é invertida em relação ao padrão do *OpenGl*.

Ao inserir uma forma geométrica, esta será colocada na origem (0,0,0) para melhor visualização pelo usuário. Além disso, o usuário pode mover-se pelo ambiente, utilizando o mouse. Em qualquer momento, ele pode pressionar o botão origem, que o levará automaticamente para a posição de origem, na qual os objetos são sempre criados.

Caso o usuário não queira mais trabalhar com a cena criada, poderá apagar todos os objetos da cena para começar de novo. O Editor possibilita apagar todo o ambiente criado. Antes de ter sua área de trabalho limpa, o usuário é alertado para se realmente deseja apagar o que está na cena.

Logo, toda a forma do objeto pode ser definida, deixando o mesmo com as características que o usuário necessitar para a construção do ambiente final (Figura 8).

A construção do Editor foi realizada usando a Linguagem de Programação *Delphi V* e componentes de *OpenGl* distribuídos pela *SignSoft*. O ambiente final resultante, construído pelo usuário, poderá ser gravado em formato .wrl e visualizado nos *Browsers* disponíveis que manipulem ambientes virtuais em VRML.

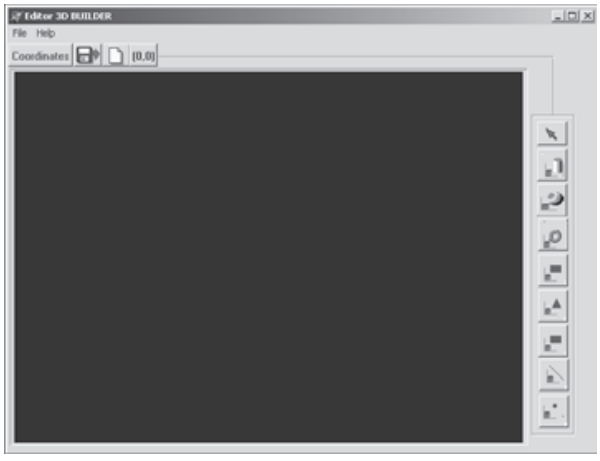

Figura 4 - Tela de Trabalho do Editor

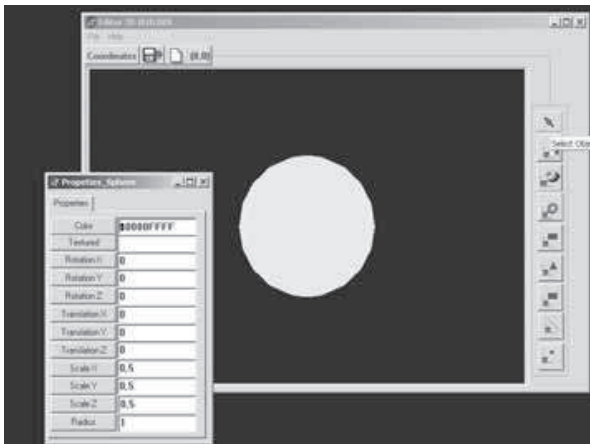

Figura 5 - Propriedades do Objeto Selecionado

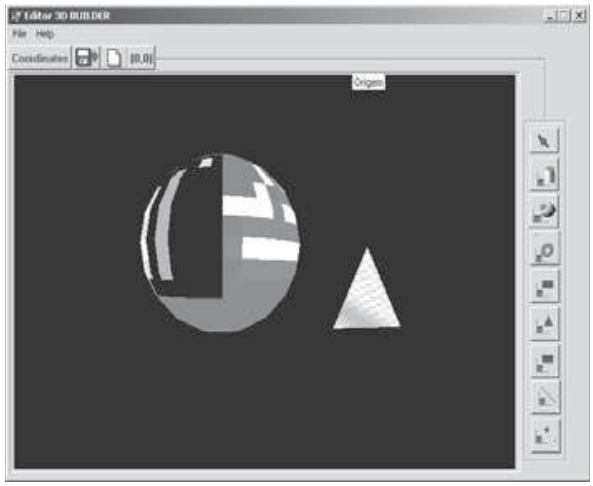

Figura 6 - Editor Gráfico com Objetos

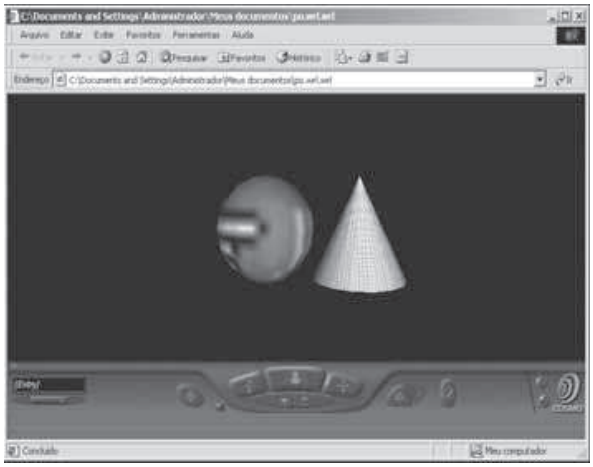

Figura 7 - Objetos no *Pluging* VRML

CONCLUSÕES

Na atualidade, não se pode falar em informática sem mencionar a Internet, a grande rede internacional de computadores que permite a comunicação e troca de informações por todo o planeta e que, atualmente, também oferece a possibilidade de se usar a Realidade Virtual. Esta última pode ser obtida por meio da linguagem VRML, que descreve seus mundos na forma textual, permite a inserção de links, que é uma das principais características da *Web* e, principalmente, não depende de nenhum ambiente operacional, somente a utilização de um browser que possua um *Plug-in* VRML instalado.

Motivados pela integração com outras linguagens, a Linguagem VRML foi escolhida, junto com o Padrão *OpenGL*, para a criação do Editor Gráfico *3D Builder*. A partir dessas duas tecnologias, o Editor se tornou de fácil acesso ao usuário, não o deixando distante de um programa normal que pode ser instalado em qualquer máquina.

Foram analisados dois editores gráficos, o *ATSWORLDS* e o *RenderSoft*, cada um com suas características. Os dois oferecem limitações e algumas dificuldades para o usuário, o que pode deixar o mesmo insatisfeito. Com a análise destas dificuldades, surgiu a idéia de criar um editor gráfico que fosse de fácil utilização para a criação de ambientes virtuais. Este editor facilitaria a criação de ambientes virtuais pelo usuário, possibilitando que o mesmo possa ter várias alternativas de criação.

O Editor desenvolvido é uma ferramenta relativamente "leve", não necessitando de máquinas de grande porte para rodar. Portanto, ele pode ser utilizado por qualquer usuário, pois o editor é de fácil compreensão. Outra característica importante é o fato de o editor suportar um número indeterminado de objetos, o que facilita o trabalho de quem desenvolve mundos virtuais.

Além disso, não é necessário nenhum tipo de configuração especial para a geração dos arquivos VRML, pois o Editor já está programado para fazer as transformações necessárias para o usuário. Tal característica facilita a implementação dos ambientes virtuais. Também é importante salientar que o que o usuário visualiza no editor, em *OpenGl*, é o que será construído em VRML, o que gera uma melhor qualidade na criação dos ambientes. O Editor exporta para a VRML de uma maneira rápida e precisa.

Também é necessário relatar as dificuldades que existiram para deixar o Editor acessível e prático para o usuário:

· a criação do vetores dinâmicos, para a alocação de número ilimitados de figuras;

· as características que cada objeto tem no editor, que tinham que ser compatíveis entre o *OpenGl* e a VRML;

· a conversão de cores do padrão *Delphi* para RGB;

· a conversão de textura *bitmap* para jpeg.

REFERÊNCIAS BIBLIOGRÁFICAS

ZAGO, Anselmo. 1998 **VRML.** Disponível em: < http://www.zago.anselmo.com.br >

JACOB, Jon Q. 1998. **Developer's Guide to OpenGl**.

RAMALHO, José Antônio Alves. 1999. **HTML4 Prático e Rápido.** Série Ramalho. SP. Berkeley Brasil.

IPOLITO, Juliano. 1996. **Tutorial VRML 1.0.** Departamento de Computação - Universidade Federal de SP. Disponível em: < http://www.dc.ufscar.br. >

JAMSA, Kris; SCHMAUDER, Phil; YEE, Nelson. 1999. **VRML - Biblioteca do Programador.** 1° Ed. São Paulo: Makron Books

POLLO, Luiz F; Lima, J.C.D. 1997. **Software para a Geração Automática de Modelos 3D em VRML.** Trabalho de Graduação - UFSM. Disponível em: < http://www.inf.ufsm.br/~pollo/TG. >

CASSAL, Marcos Luís. 1999. **Construindo Ambientes Virtuais com VRML.** Trabalho Individual - UFRGS. Jan.

DUARTE, Mauro Lucio; ZANONI, Cícero; SILVA, Daniela Eloy. 1998. **Tutorial VRML2.0.**

CASTIER, Beatriz; PEREIRA, Luciano. 1997 **Tutorial VRML.** Departamento da Ciência e Computação. Disponível em: < http://www.dcc.ufba.br/mat056 >

RENDERSOFT. 1998. **VRML.** Editor Version 1.722. 1997-1998. RenderSoft Software and Web Publishing

SingSoft VisIt. **OpenGl. Versão 2.5**. Disponível em: < http://www.signsoft.com.visit >

SILVA, André Tavares. 2000. **Modelagem de objetos 3D em VRML:** Uma implementação multiplataforma orientada ao ensino. workshop de realidade virtual. Gramado - RS