

PLANETA X – DESENVOLVIMENTO DE UM JOGO VIRTUAL UTILIZANDO A TECNOLOGIA VRML

Fabrcio Viero de Araujo
Universidade Luterana do Brasil - ULBRA
Brasil
rufos@terra.com.br

Andre Zanki Cordenonsi
Centro Universitrio Franciscano
Brasil
andrezc@unifra.br

RESUMO

Este projeto teve como objetivo demonstrar a interatividade dos ambientes virtuais utilizando a VRML, demonstrando as suas capacidades para aplicaes na rea de entretenimento e a utilizao da mesma como ferramenta interativa voltada para aplicaes na internet. Para tanto, foi desenvolvido um jogo no estilo *Role Playing Game*. O jogo composto por duas fases, onde o usurio ter que passar pelas mesmas para chegar ao fim do jogo. A primeira fase se passa em uma nave, onde o usurio dever descobrir como fazer para que a mesma funcione. Depois de ligar a nave, ela levar o usurio at o planeta que a segunda fase. Nesta fase, o jogador dever encontrar pistas at chegar o final do jogo. Para uma melhor interatividade do jogo com o usurio, o mesmo foi embutido em pginas HTML, o que auxiliou a estruturaao do mesmo.

PALAVRAS-CHAVE: VRML, Interatividade, Realidade Virtual, Animaao, Interface

ABSTRACT

This project had as objective to present to use techniques of Virtual Reality as it has supported environments of interactive imersive, trying to bring the most real possible real world for the virtual world. The developed game brought important aspects for the implementation and the use of language VRML in the Web environment. The result of this study brought a great learning for the work of interfaces and imersive in used virtual worlds in the Internet.

KEY WORDS: VRML, Iterative World, Virtual Reality, Animation, Interface

INTRODUÇÃO

Os jogos de computador, nos dias de hoje, são um grande meio de divertimento e educação para as pessoas. Grande parte do mercado de *software* mundial é constituído por jogos de computador e de vídeo games. Tal fato foi um dos principais motivadores ao desenvolvimento deste projeto. Vários tipos de jogos se encontram associados a esta categoria e neles estão também incluídos jogos interativos voltados a educação.

A linguagem VRML (*Virtual Reality Modelling Language*) surgiu no mundo da realidade virtual como um grande marco para a criação e interação do usuário com ambientes virtuais, aproximando estes dos mundos reais (Garey, 2000). Com o advento da *internet*, a necessidade de uma linguagem para criar ambientes tridimensionais, onde o usuário pudesse interagir com o mesmo, tornou-se muito importante. Assim, surgiu a linguagem VRML que, em sua versão 1.0, possibilitava a criação de ambientes tridimensionais estáticos.

Conforme as necessidades foram aumentando, uma nova versão da linguagem VRML surgiu (VRML 2.0), trazendo como novidade a possibilidade da criação de animações e sensores de choques, o que tornava a construção de mundos virtuais muito mais realistas (Pollo, 1997).

Um dos principais objetivos deste projeto foi demonstrar os recursos da linguagem VRML que possibilitam a criação de mundos virtuais e em que tipo de aplicações pode-se usar tais mundos. Como exemplo prático destas aplicações, foi desenvolvido um jogo do tipo RPG (*Role-Playing-Game*).

Jogos do tipo RPG permitem ao usuário passear por um mundo virtual e, através de objetos, irá juntar um quebra-cabeça para descobrir os enigmas. O jogo desenvolvido é mais interativo e racional do que um jogo de ação, possibilitando assim o uso deste raciocínio como ferramenta de auxílio no aprendizado. Neste tipo de jogo, o jogador terá que descobrir enigmas para desvendar o final do mesmo. A idéia principal foi desenvolver um jogo para a *web* onde qualquer pessoa possa jogá-lo de qualquer parte do mundo. Para tanto, a linguagem VRML é a escolha mais interessante, pois trata-se de uma linguagem multiplataforma e que roda tranquilamente na *web*.

LINGUAGEM VRML

História

A *Virtual Reality Modeling Language* (VRML), é uma linguagem que possibilita descrever objetos e mundos 3D interativos através da *World Wide Web*. A idéia da VRML foi concebida em junho de 1994, durante a primeira conferência anual da WWW, na Suíça. A linguagem surgiu, originalmente, baseada na necessidade de se ter originalmente uma ferramenta de especificação e visualização gráfica em 3D na rede (Pollo, 1997). A primeira versão foi a VRML 1.0, que possibilitava o programador fazer mundos virtuais e objetos em 3D estáticos, sem muitos recursos. Já com a chegada da segunda versão, a VRML 2.0, foi possível construir, além de objetos e mundos virtuais, animações em 3D (Garey, 2000). A primeira versão foi criada pela *Silicon Graphics Inc.* e a segunda versão também foi desenvolvida pela *Silicon Graphics*, com contribuições da SONY e da Mitra. Os objetivos da versão 2.0 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. Assim como Java está sendo usada para produzir páginas HTML¹ mais dinâmicas e interativas, a VRML foi estendida para permitir que o ambiente mude e "evolua", tornando-se sensível às ações ou à simples presença de seus visitantes virtuais.

Funcionamento

A linguagem foi desenvolvida com a função de criar arquivos leves que pudessem ser carregados em uma página *web*. A linguagem é composta de objetos bases ou *Nodes*². Na VRML, estão definidas uma série de nós que provêm alguns recursos especiais, como ligações com outros arquivos e a inclusão de objetos definidos em outros arquivos (Garey, 2000). Por exemplo, podem ser criados objetos em um

¹ Linguagem para desenvolvimento de páginas para a Web (*Hypertext Markup Language*)

² Construção fundamental de um arquivo VRML

arquivo e compartilha-los em vários outros. Isso facilita a criação e manutenção de mundos virtuais complexos. Cada nó contém campos que mantêm os dados que o caracterizam como um elemento único dentro do arquivo.

Podemos especificar as seguintes características para um nó:

- tipo de objeto que ele representa (*Cube*, *Sphere*, *Transform*, etc.);
- os parâmetros (valores dos campos) que distinguem este nó de outros do mesmo tipo. Um nó pode ter zero ou mais campos;
- um nome que o identifique. Isto possibilita a reutilização de nós de maneira fácil e rápida, pois basta fazer referência ao nome de um nó já definido para instanciar um novo objeto com as mesmas características. Não é obrigatório nomear um nó e os nomes não precisam ser únicos;
- nós filhos. A hierarquia de objetos, em VRML, é implementada de maneira a permitir que alguns tipos de nós contenham outros nós. Estes nós, que mantêm outros agrupados, são chamados de *grouping nodes* (nós de agrupamento), e podem ter zero ou mais filhos;
- um conjunto de eventos que o nó pode receber (chamados de *eventIn*) e enviar (chamados de *eventOut*). Todos os recursos de animação e interação de VRML 2.0 são baseados na adição de comportamentos aos objetos da cena. Os eventos são a maneira de efetuar as mudanças de estado nos objetos, alterando os valores de seus campos quando for necessário.

Para se visualizar um arquivo VRML é necessário um *Plug'in*³, para que o mesmo interprete o código VRML e possa apresentar os resultados. O *Plug'in* também configura a renderização dos objetos, a qualidade gráfica e a velocidade dos movimentos do usuário. Um dos *Plug'in* para visualização de VRML é o *Cosmo Palyer*, desenvolvido pela *Cosmo Software*, sendo o *Plug'in* mais utilizado pelos programadores (Kris,1999). Ele, como os outros *Plug'ins*, é carregado no *browser*⁴.

Principais Nodes

Os principais *nodes* da linguagem VRML para a modelagem de objetos são os comandos *Sphere*, *Box*, *Cylinder*, e *Cone*, que serão definidos a seguir. Com estes objetos base, é possível criar outros objetos a partir da união dos mesmos (Kris,1999). Existem também outros *nodes* que definem propriedades para cada objeto como cor, luz e posição.

Objeto Esfera

O objeto esfera define uma esfera de raio definido pelo usuário. Sua sintaxe é a seguinte:

Sphere { radius tamanho do raio da esfera }

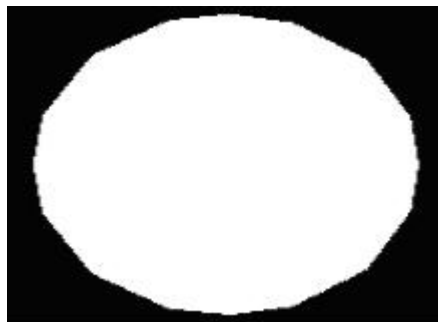


Figura 1 – Objeto esfera

A figura 1 define um objeto *sphere* simples, sem nenhum tipo de propriedade definido, como cor e iluminação.

³ Programa que traduz o código de uma determinada linguagem e apresenta os respectivos resultados

⁴ Programa que possibilita a visualização de arquivos VRML e páginas na *internet*

Objeto Cilindro

O objeto cilindro possibilita a modelagem de um objeto na forma de um cilindro, onde o usuário define o raio do objeto e a altura do objeto. Sua sintaxe é a seguinte:

Cylinder { height altura do cilindro
radius raio do cilindro }



Figura 2 – Objeto cilindro

Objeto Cone

O objeto cone possibilita a modelagem de um cone, onde o usuário define o raio da base e altura do cone. Sua sintaxe é a seguinte:

Cone { bottomRadius raio da base do cone
height altura do cone }



Figura 3 – Objeto Cone

Objeto Paralelepípedo

O objeto paralelepípedo possibilita a modelagem de um objeto paralelepípedo, onde o usuário define a altura, o comprimento e a profundidade do paralelepípedo. Sua sintaxe é a seguinte:

Box { size comprimento altura profundidade }



Figura 4 – Objeto paralelepípedo

Definindo uma Translação

Translation é o *node* que é utilizado para definir a posição de um objeto no seu espaço virtual. O usuário define as coordenadas X, Y e Z e o objeto se posiciona nesta posição no espaço virtual. A sintaxe é a seguinte:

translation valor_em_x valor_em_y valor_em_z

Para definir uma translação é preciso especificar, além da posição dos objetos, quais são os nodos, ou objetos, que devem sofrer estas modificações. No exemplo baixo, é definido um código mostrando a definição de um *node* de translação para o objeto *Box*.

#VRML V2.0 utf8

Transform

```
{
  translation 3 6 9
  children
  [
    Shape {
      geometry Box {size 6 6 6}
    }
  ]
}
```

Nó Indexedfaceset

Este nó foi utilizado para criar o terreno do planeta. Com ele você determina os pontos no espaço a partir do nó *coord Coordinate*, os pontos são as coordenadas X,Y e Z. A partir desses dados o *IndexedFaceSet* indexa (liga) os pontos previamente definidos no nó *coord Coordinate*..

```
geometry IndexedFaceSet {
  coord Coordinate {
    point [ 0.00 0.00 12.00
            52.19 -9.21 12.00
            88.54 16.89 20.00
            102.38 53.67 5.00
            73.50 90.74 1.00
            34.48 85.12 12.00
            - 0.89 42.47 12.00
            0 0 12.00 ]
  }
}
```

```
]
}
coordIndex [
0
1
2
3
4
5
6
7
]
}
```

Definindo uma Rotação

Uma rotação consiste em alterar a posição de um objeto em torno de um eixo no espaço. Sendo assim, deve-se estabelecer um eixo (X, Y, Z) para a rotação e um quarto parâmetro para o ângulo de rotação.

rotation x y z angulo

No exemplo a seguir, um cone será rotacionado em 90° no eixo Y, enquanto o outro cone manterá sua posição normal, trasladado de 5 unidades no eixo X.

```
#VRML V2.0 utf8
Transform
{
  rotation 0 1 0 1.57 #eixo y ângulo 1.57 radianos
  children [
    Shape {
      geometry Cone{ bottomRadius 2 height 5 }
    }
  ]
}
```

Definindo a Aparência dos Objetos

As possíveis manipulações com relação à aparência dos objetos são feitas com as diretivas: **Appearance**, **ImageTexture** e **Material**.

Utilizando texturas

O formato do arquivo contendo uma textura pode ser um GIF, JPG ou PNG. Deve-se definir qual imagem será aplicada como textura, através de sua localização *url*⁵.

```
#VRML V2.0 utf8
Transform
{
  children
  [
    Shape
    {
      appearance Appearance
      {
        texture ImageTexture
        {url "outros\logo.gif"} # imagem
        geometry Box{size 2 2 2}
      }
    }
  ]
}
```

⁵ Endereço de uma página na *internet* ou no computador local

```
}
]
}
```

Utilizando Cores

O formato das cores é o *RGB* (*red green blue*), sendo que cada componente pode assumir valores entre 0.0 e 1.0.

#VRML V2.0 utf8

Transform

```
{
  children
  [
    Shape
    {
      appearance Appearance
      {
        material Material{diffuseColor 0 0 1}
        #cor azul
      }
      geometry Sphere {radius 2}
    }
  ]
}
```

Todas as possibilidades de cores e elementos relacionados à aparência do objeto estão sendo utilizadas no exemplo abaixo:

#VRML V2.0 utf8

Transform

```
{
  children
  [
    Shape
    { #a palavra appearance se refere ao campo do
      # comando Shape, já a palavra Appearance se
      #refere ao nodo, que por sua vez
      #possui um campo chamado material, que
      #deve conter um nodo do tipo Material
      appearance Appearance # aparência do cone
      {
        material Material
        { diffuseColor 0 0 1, #cor difusa
          ambientIntensity 0.6,
            #intensidade do ambiente
          emissiveColor 0 0 1, #cor emissiva
          shininess 0.3 #brilho
          specularColor 0 0 0.7 #cor especular
          transparency 0 #transparencia do objeto
        }
      }
      geometry Cone {bottomRadius 2 height 5}
        # cone de altura 5 e raio da base 2
    }
  ]
}
```

O objeto representado pelo código anterior pode ser visualizado na figura 5.

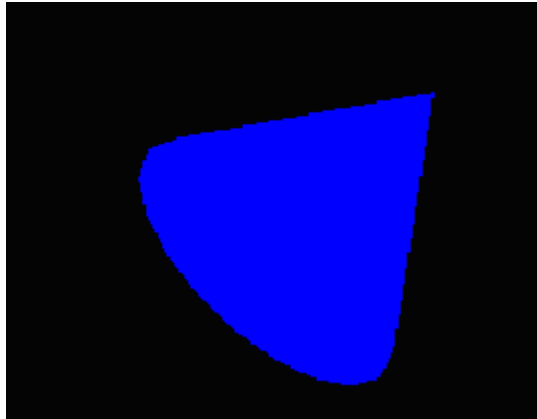


Figura 5 – Objeto cone

JOGOS DE COMPUTADOR VIA INTERNET

Com o surgimento da *internet*, surgiu uma nova categoria dentro dos jogos de computador, os jogos *multiplayer*⁶ via *internet*. Estes jogos são desenvolvidos com o propósito de tornar os mesmos mais realistas com o mundo em que vivemos.

Nestes jogos, normalmente, não existem agentes inteligentes ou adversários controlados pelo computador. Os próprios adversários são os jogadores onde cada um deve cumprir seus próprios objetivos. Por exemplo, em um simulador de voo *singleplayer*⁷, o usuário irá combater outros aviões que são controlados pelo computador. Já em uma versão *multiplayer*, vários usuários se conectam em uma missão e cada um irá controlar um avião, acabando com o papel da máquina como objetivo final.

Em um jogo *multiplayer*, deve-se levar em conta alguns problemas que não ocorre nos jogos *singleplayer*, como: (1) velocidade de transmissão na rede: para que o jogo funcione corretamente e com velocidade satisfatória, a rede deve suportar um tráfego alto de informações; (2) suporte a TCP/IP⁸: na *internet* o protocolo de transmissão e endereçamento é o TCP/IP, e todo o jogo baseado para funcionar na *internet* deve possuir como base de transferência de dados este protocolo de rede para que o mesmo funcione satisfatoriamente. (3) servidor dedicado: alguns jogos complexos, como *Quake*, deverão possuir um servidor dedicado onde os usuários deverão se conectar, pois muitos jogadores se conectam ao mesmo tempo e o servidor faz o papel de unir os mesmos.

Mas com a chegada de novas tecnologias e o aumento da capacidade de tráfego pela *internet*, os jogos *multiplayer* deverão ganhar espaço no mercado e isto irá trazer a criação de novas tecnologias para o desenvolvimento dos mesmos.

DESENVOLVIMENTO DO JOGO EM VRML

Enredo

No ano de 2098, os seres humanos iniciaram uma jornada de colonização no sistema solar. Marte foi o primeiro passo, no ano de 2025, pois era o planeta mais adequado a tal feito, por causa da distância, do tipo de solo e de possuir gelo que podia ser transformada em água potável. Mas no ano de 2030, um novo planeta inóspito, mas com grandes chances de possuir recursos naturais, foi encontrado. Os cientistas ficaram estarecidos com a descoberta, pois o planeta teria surgido do nada. Ou será que este sempre esteve ali e nunca teria sido descoberto antes? O planeta foi batizado de **PLANETA X**, por haver

⁶ Jogos onde participam mais de um jogador ao mesmo tempo

⁷ Jogos onde participam um único jogador

⁸ Transmission Control Protocol/Internet Protocol

tantas incógnitas em sua descoberta. Em 2033 a sonda Venusian I foi enviada para vasculhar o planeta. Os dados mostraram que o planeta era como os cientistas haviam previsto, uma terra incógnita pronta para ser decifrada. O **PLANETA X** não ficava muito longe da Terra, pois a distância equivalia a ir e vir à lua 4 vezes. Em consequência disto, o planeta foi o escolhido para ser o primeiro a ter uma colônia terrestre de grandes proporções, utilizando as técnicas que foram usadas na pequena colônia em Marte.

No dia 25 de dezembro de 2035, cinco anos após a sua descoberta, a primeira missão tripulada partiu com destino ao **PLANETA X**, para ser feita uma exploração mais detalhada do mesmo. A nave Europa, nomeada assim por causa da lua de Júpiter de mesmo nome, foi preparada para levar 8 astronautas. Chegando ao planeta, foi constatado que o mesmo possuía água e, devido às características físico-químicas, havia sinais de que o planeta poderia ter sido habitado no passado por seres extraterrestres. Os astronautas permaneceram dois meses sob sua atmosfera rica em gás carbônico, coletando amostras e dados para serem levados à terra.

No dia primeiro de março de 2036, a Europa se acoplava à Estação Espacial Internacional, que fora adaptada e melhorada no decorrer dos anos. A nave de desembarque para a terra levou todos os dados e os astronautas responsáveis pela missão. Na NASA, uma equipe de cientistas de todo o mundo estudou durante um ano todos os materiais coletados: amostras de solo, água e minerais que foram encontrada no planeta. Foi constatado que o mesmo era o ideal para o início de uma colonização e a posterior criação de uma Terra artificial. Foi definido um grupo de quinhentos colonizadores, dentre os quais quatrocentos e cinquenta cientistas; as outras cinquenta pessoas eram 25 casais que tentariam o início de uma adaptação fora da Terra. No decorrer de cinquenta anos, mais trezentas e oitentas missões tinham sido executadas para o planeta, preparando-o para a sua colonização. No dia 25 de dezembro de 2098, as quinhentas pessoas escolhidas para o *Big Step*, assim batizado, foram levadas para a grande Estação Espacial Internacional para entrarem a bordo da nave Titus, que era a antiga nave Europa remodelada e ampliada, para o feito. No dia primeiro de janeiro de 2099, a nave Titus seguiu sua trajetória rumo ao **PLANETA X**, a terra incógnita.

Com a chegada de novos habitantes ao antigo planeta, e a colônia já pré-construída, todos começaram a iniciar o processo de criação de uma nova terra. Os cientistas desenvolveram um reciclador na Terra, uma usina que transformava o monóxido de carbono existente na atmosfera do planeta em oxigênio. Aos poucos, começaram a ser cultivadas plantas desenvolvidas em laboratórios, e previamente testadas em Marte, no **PLANETA X**. Ao longo do tempo, o planeta começou a ganhar vida e os colonizadores se adaptaram rapidamente ao planeta. Com o passar dos anos, a colônia começou a desenvolver e crescer, os casais começaram a gerar filhos e uma verdadeira nova Terra a nascer. Novas pessoas começavam a chegar no planeta, juntamente com mais cientistas, para estudá-lo. Com o tempo, foi encontrada água subterrânea salgada e pequenos rios de água fervente, mas potável. Devido à situação favorável, as chances de criação de uma nova Terra fortaleceram-se.

Mas no ano de 2199, a beira de completar 100 anos, o contato com a colônia se perdeu. Uma onda de pânico tomou os corpos de cada ser humano na terra. O que haveria acontecido com os colonizadores do espaço?

Urgentemente, uma nave de busca e patrulha militar foi enviada para o planeta com trinta agentes espaciais, treinados para este tipo de situação. A nave militar Supernova partiu rumo ao **PLANETA X** no dia doze de outubro de 2199.

Chegando à colônia, foi constatado que todos os habitantes haviam desaparecido, sem rastros e nem pistas. Agora, somente o ar de morte e mistério habitava a colônia que outrora estava cheia de vida e trazia a esperança para todos os habitantes da Terra.

E é aí que nosso jogo começa.

Filosofia do Jogo

O jogo exige raciocínio e conhecimento, onde o usuário é um agente espacial com a função de descobrir, através de pistas, o que ocorreu com os colonizadores do **PLANETA X**. A tática do jogo é reunir o maior número de pistas e juntá-las em um quebra cabeças, levando-o a desvendar o mistério do desaparecimento dos colonizadores.

As pistas podem ser objetos que o usuário irá achar. Ele irá checar o objeto e verificar se pode existir alguma pista que o leve a algum lugar ou a alguma conclusão. Outra opção é algum texto que pode ser encontrado em algum local do jogo, o usuário irá ler o texto e terá que associar com outras pistas para ir montando o quebra-cabeças e chegar a alguma conclusão.

Ambiente

No jogo, os ambientes são de extrema importância, pois é neles que o usuário irá explorar e tentar achar as pistas para chegar ao fim do jogo. Os ambientes devem ser muito interativos e de fácil entendimento. Não se pode criar um objeto que deseja-se passar uma informação para o usuário, se o mesmo não entender o que é este objeto. Se isto ocorrer, e se a função do objeto é de fornecer uma pista de extrema importância, o usuário não conseguirá concluir o enigma. Isto seria prejudicial para um bom jogo.

Funcionamento do Jogo

A idéia principal do jogo é fazer com que o usuário entenda a sua finalidade. O ponto forte do jogo é a interatividade que o usuário terá com os objetos do jogo e os seus ambientes. O jogo será embutido em um conjunto de páginas HTML, que irão auxiliar o usuário no acesso ao mesmo. Com o decorrer do jogo o usuário irá descobrir enigmas onde possibilitará o mesmo a passar de fase .

Além dos ambientes onde o usuário irá interagir, o jogo também possui animações desenvolvidas em VRML, cujo objetivo é tentar passar para o usuário o que está ocorrendo durante a sua participação no jogo.

Existem quatro animações: uma introdutória, duas que servem para passar ao usuário a passagem de uma fase para outra, e uma animação final, que passa ao usuário a conclusão do jogo. Também existem outras pequenas animações embutidas dentro dos ambientes para deixar o jogo mais perto da realidade.

Fases

O jogo se constitui de duas fases. A primeira inicia com usuário dentro de uma nave. Ele deverá descobrir como fazer a nave funcionar para que ela possa chegar até o planeta. Nesta fase, foram desenvolvidos dois ambientes distintos. O primeiro é a cabine da nave (cúpula), onde se encontra o painel de controle da nave, conforme pode ser visualizado na figura 6. O segundo é a fuselagem da nave, onde se encontra um corredor que dá acesso ao sistema principal de energia da nave, conforme mostra a figura 7. O usuário terá que descobrir como ativá-lo para poder ligar a nave.

Ao conseguir ativar o sistema de energia da nave, uma animação será executada, passando ao usuário a idéia de que a nave está sendo ativada. A mesma será levada para o planeta, quando é iniciado a segunda fase do jogo.

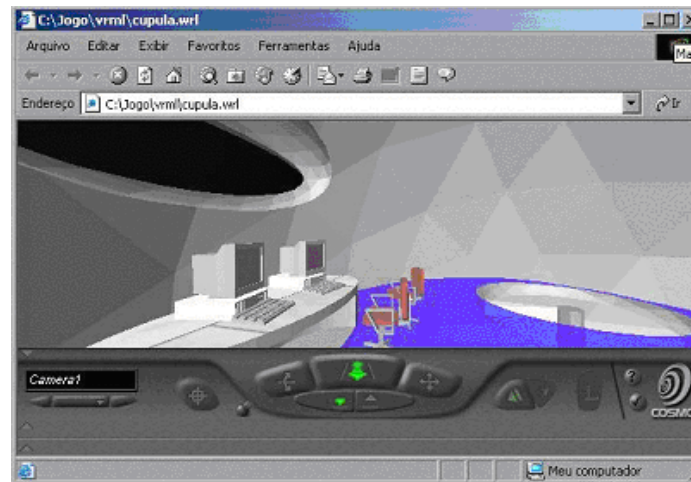


Figura 6 – Cúpula da nave

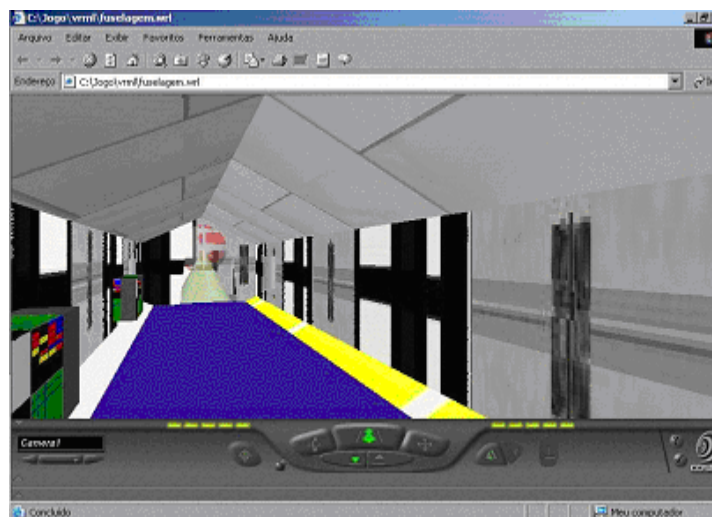


Figura 7 – Fuselagem da nave

Ao passar para a segunda fase do jogo, o usuário irá mudar de ambiente. Este novo ambiente é composto por outros pequenos ambientes, onde o usuário poderá interagir mais diretamente. O ambiente base é o terreno do planeta, conforme mostra a figura 8, que é constituído por pontos interligados entre si.

Dentro deste terreno vários outros ambientes estão inseridos. Existem objetos secundários que servem apenas para aproximar o ambiente da realidade, tais como imagens para simular o céu do planeta, entre outras estruturas. Outros objetos foram criados para que o usuário possa interagir, explorando interiormente os mesmos para poder descobrir os enigmas.

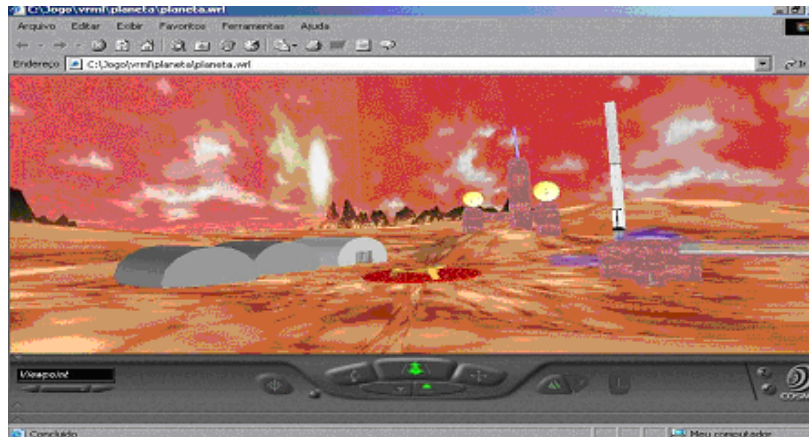


Figura 8 – O planeta e seus objetos

Basicamente, o planeta é constituído de:

- duas torres de comunicação que tem função de aproximar o ambiente da realidade;
- uma torre de informações, onde o usuário poderá entrar e explorar;
- um hangar que contém uma nave para o usuário explorar;
- uma torre alta, para o usuário subir e ter uma visão de todo o planeta de cima da mesma;
- uma caverna, que o usuário deverá achá-la, pois é nesta que ficará uma pista muito importante sobre como chegar ao fim do jogo.

Todos estes ambientes servem para o usuário explorar. Em alguns destes existirão pistas para chegar ao fim do jogo.

Construção do Jogo

O jogo foi construído em partes distintas, para uma melhor estruturação do mesmo. Os ambientes foram construídos em arquivos diferentes, posteriormente indexados em um único arquivo. Tal metodologia possibilita um controle melhor na fase de desenvolvimento, pois torna os ambientes menos pesados para trabalhar com os mesmos. Basicamente, os ambientes desenvolvidos e sua inter-relação podem ser observados na figura 9.

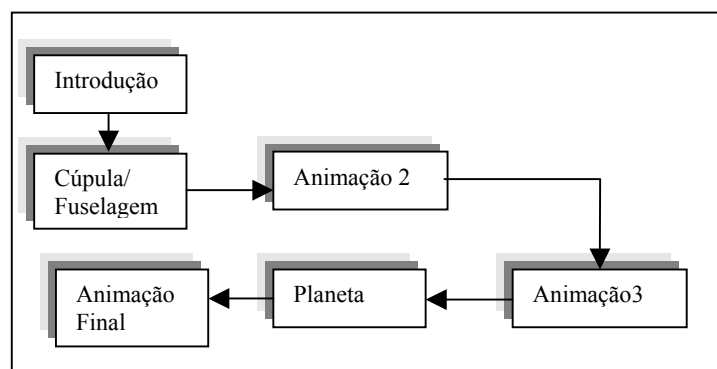


Figura 9 – Estrutura do funcionamento do jogo

Alguns objetos como, a estação espacial Internacional, foram adquiridos na página da NASA.

Sons e Texturas

No desenvolvimento do jogo, foram inseridos sons e imagens texturizadas como complemento para dar maior realismo aos ambientes. No entanto, estas técnicas requisitam de um *hardware* mais potente, principalmente na parte de vídeo (placa aceleradora e memória) e memória RAM.

Este é um ponto importante que pode influenciar no funcionamento do jogo, quando este for rodado a partir da *internet*. As texturas utilizadas estão no formato *jpg* e *gif*. Estes formatos foram escolhidos por serem suportados pelo VRML, representar um padrão na *internet* e possuir um tamanho, em *bytes*, relativamente pequeno (Garey,2000).

Os sons utilizados são do formato *wav* para pequenos efeitos, pois os arquivos *wav* se tornam relativamente grandes se a duração do som for muito longa. Para os sons de duração maior, foi utilizado arquivos no formato *mid*, pois estes conseguem comprimir melhor sons mais longos.

Animações

Para dar um maior realismo ao jogo algumas animações foram desenvolvidas para prender a atenção do usuário. As mesmas também necessitam de um *hardware* mais potente, pois além de serem animações possuem objetos complexos, sons e texturas. A figura 10 demonstra uma animação do jogo, com todas as características comentadas anteriormente.

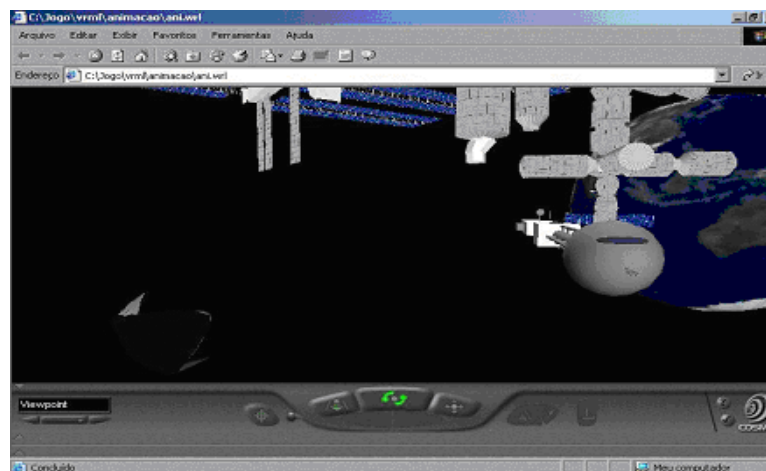


Figura 10 – Animação introdutória

Desenvolvimento das Páginas Html

As páginas HTML, onde foram embutidos os arquivos VRML, são constituídos de páginas simples onde o usuário será informado sobre o projeto do trabalho, as partes compostas do jogo e uma introdução ao mesmo. Esta página inicial dá uma visão mais abrangente de como funciona o jogo.

O conjunto de páginas é constituído de uma página inicial onde se encontra uma animação em *Flash*, inserida como uma introdução ao jogo. Esta página inicial leva para a página principal. Na página principal existem controles que podem levar o usuário para os diversos ambientes, como mostra a figura 11.

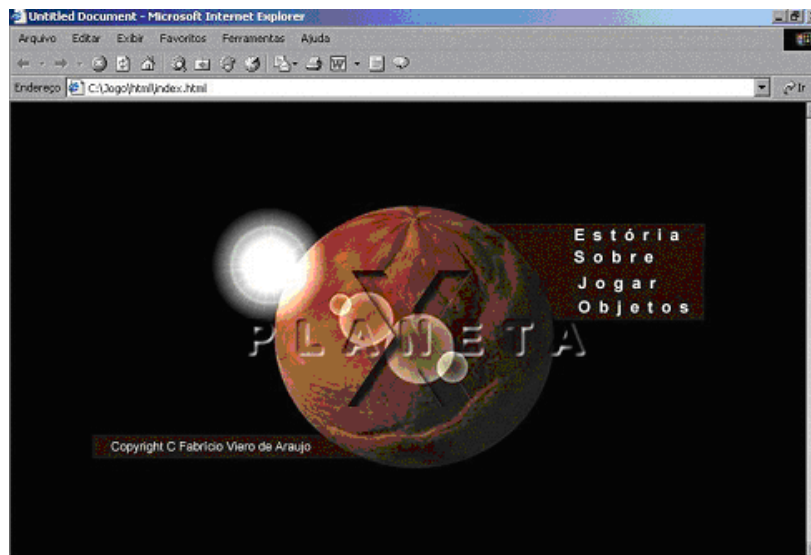


Figura 11 – Design da página inicial do jogo

CONCLUSÃO

Atualmente, não podemos fugir da nossa realidade onde a *internet* tornou-se um fenômeno, trazendo diversas funcionalidades e facilidades à nossa vida. Logo, para que este mundo se torne mais real na *internet*, a linguagem VRML pode ser utilizada como um meio para a geração da realidade virtual, construindo uma interface entre o mundo real e o quase real.

No entanto, a realidade virtual necessita de muito mais taxa de transferência na rede e de *hardwares* mais potentes, para uma boa performance, o que torna a realidade virtual na *internet* ainda um pouco distante do usuário comum.

Apesar das dificuldades técnicas em relação à velocidade, neste trabalho pode-se observar que muitos recursos da linguagem VRML podem ser aplicados, tanto no entretenimento como em aplicações comerciais e educacionais, de uma forma aceitável por parte do usuário. Sua vasta gama de recursos como animação, visualização de objetos tridimensionais e a possibilidade de interação do usuário com os ambientes traz vantagens muito significativas para os programadores. Um outro ponto forte da linguagem VRML é a interação quase que perfeita com a linguagem Java, onde a mesma possui bibliotecas de VRML pré-definidas o que facilitam a sua interação. O único ponto fraco da linguagem VRML é que a mesma precisa de um *plug-in* para a sua visualização, o que faz com que não seja aceita ainda unanimemente na *internet*.

Uma das principais dificuldades de desenvolver um jogo é a parte do enredo do mesmo, já que este deve ser muito atrativo para que o jogador goste do mesmo. Uma outra dificuldade inerente ao desenvolvimento de jogos de *RPG* foi a definição dos enigmas utilizados como permissão do usuário para passar de fase. Estes devem ser muito bem esquematizados, para que possam transmitir ao usuário o que está acontecendo no decorrer do jogo. Na parte de texturas, o maior problema encontrado foi achar as figuras corretas para todos os objetos.

Durante a construção do jogo, foram utilizado diversos editores gráficos para VRML, que facilitam a construção dos objetos e das animações do jogo. Além disso, uma outra facilidade na construção de jogos em VRML é a implementação do mesmo em páginas HTML, o que torna a instalação e execução do jogo mais fácil para o usuário final.

Trabalhos Futuros

Alguns trabalhos poderiam ser implementados a partir deste projeto. Um trabalho interessante seria a criação de um mundo virtual para ser colocado na *internet*, que possibilitasse aos jogadores a interação entre si como se fosse uma sala de bate-papo tridimensional. Desta forma, os jogadores poderiam trocar idéias sobre o jogo. Também seria interessante a criação de novas fases, ampliando o mundo virtual do jogo, gerando novas possibilidades de exploração. Finalmente, poderia ser desenvolvido um sistema de cadastro *on-line*, onde cada usuário pudesse manter um controle das fases do jogador, assim como prover facilidades tais como a gravação da localização do jogador e sua fase corrente.

REFERÊNCIAS BIBLIOGRÁFICAS

- CosmoSoftware Home page , CosmoSoftware Inc., <http://www.cosmosoftware.com>
- Estação Espacial Internacional – NASA, Home Page - <http://spaceflight.nasa.gov/>
- Garey,Rikk; Bell,Gavin. The Annotated VRML 2.0 Reference Manual. Addison-Wesley, 2000.
- Kris Jamsa, Ph.D.; Schmauder, Phil; Yee, Nelson. VRML Biblioteca do Programador, Makron Books, 1999.
- ParallelGraphics Home Page , ParallelGraphics Inc. <http://www.parallelgraphics.com>.
- Pollo, Luiz F. ; Lima, J. C. D. Software para a Geração Automática de Modelos 3D em VRML.Trabalho de Graduação - UFSM. Dez. 1997. <http://www.inf.ufsm.br/~pollo/TG>.
- Silicon Graphics Home Page, Silicon Graphics Inc, URL: <http://www.sgi.com>
- Web 3D Consortium Home Page, <http://www.web3d.org/>