

CAPÍTULO IV

4 ARQUITETURA DO SISTEMA

4.1 Introdução

A partir do estudo realizado nos capítulos 2 e 3, foi possível perceber a necessidade deste trabalho. Antes de mostrar detalhadamente a arquitetura do sistema, será apresentada a tecnologia de apoio necessária ao funcionamento do mesmo.

4.2 Tecnologia de apoio

Novas tecnologias têm sido criadas recentemente para o suporte ao desenvolvimento de aplicações que utilizam a Realidade Virtual, com isso possibilitam a utilização de computadores pessoais (PCs) melhores capacidades gráficas.

Neste trabalho, a principal tecnologia utilizada foi o OpenGL, pois todas as outras funcionalidades foram criadas por meio de algoritmos traduzidos em linguagem de programação. Cabe ressaltar q apenas o objeto geométrico do sistema (cidade) foi modelada (desenhada) em uma ferramenta modeladora 3D (*3D Studio Max*), porem sua utilização foi necessária apenas como ferramenta auxiliar.

4.2.1 OPENGL

OpenGL não é uma linguagem de programação, mas é uma poderosa e sofisticada API (*Application Programming Interface*) para criação de aplicações gráficas 2D e 3D, ou seja, uma biblioteca de rotinas gráficas para trabalhar em duas e três dimensões. Normalmente, se diz que um programa é baseado em OpenGL ou é uma aplicação OpenGL, o que significa que ele é escrito em alguma linguagem de programação que faz chamada a uma ou mais bibliotecas OpenGL (PINHO, 2004).

OpenGL é definido como um programa de interface para *hardware* gráfico. Na verdade, OpenGL é uma biblioteca de rotinas gráficas e de modelagem, bi (2D) e tridimensional (3D), extremamente portátil e rápida. Usando OpenGL, é possível criar gráficos 3D com uma alta qualidade visual. Entretanto, a maior vantagem na sua utilização é a rapidez, uma vez que usa algoritmos cuidadosamente desenvolvidos e otimizados pela *Silicon Graphics, Inc.*, líder mundial em Computação Gráfica e Animação (WOO, 1999).

As aplicações OpenGL variam de ferramentas CAD a programas de modelagem usados para criar personagens para o cinema. Além do desenho de primitivas gráficas, tais como linhas e polígonos, OpenGL dá suporte a iluminação, colorização, mapeamento de textura, transparência, animação, entre muitos outros efeitos especiais. Atualmente, OpenGL é reconhecida e aceita como um padrão API para desenvolvimento de aplicações gráficas 3D em tempo real. (RIBEIRO, 2006)

Em vez de descrever a cena e como ela deve parecer, quando se está utilizando OpenGL é preciso apenas determinar os passos necessários para alcançar a aparência ou efeito desejado.

Por ser portátil, OpenGL não possui funções para gerenciamento de janelas, interação com o usuário ou arquivos de entrada/saída. Cada ambiente, como por exemplo o *Microsoft Windows*, possui suas próprias funções para esses propósitos. Não existe um formato de arquivo OpenGL para modelos ou ambientes virtuais. OpenGL fornece um pequeno conjunto de primitivas gráficas para construção de modelos, tais como pontos, linhas e polígonos. A biblioteca GLU (que faz parte da implementação OpenGL) é que fornece várias funções para modelagem, tais como superfícies quádricas, e curvas e superfícies NURBS (*Non Uniform Rational B-Splines*) (WOO, 1999; WRIGHT, 2000).

A palavra *pipeline* é usada para descrever um processo, que pode ter dois ou mais passos distintos. A Figura 4.1 mostra uma versão simplificada do *pipeline* OpenGL. Como uma aplicação faz chamadas às funções API OpenGL, os comandos são colocados em um *buffer* de comandos. Este *buffer* é preenchido com comandos, vértices, dados de textura etc. Quando este *buffer* é "esvaziado", os comandos e dados são passados para o próximo estágio.



Figura 4.1. Versão simplificada do pipeline OpenGL (WOO, 1999).

Após a etapa de aplicação das transformações geométricas e da iluminação, é feita a rasterização, ou seja, é gerada a imagem a partir dos dados geométricos, de cor e textura. A imagem final, então, é colocado no *frame buffer*, que é a memória do dispositivo gráfico. Isto significa que a imagem é exibida no monitor (WRIGHT, 2000).

4.3 Módulos da Arquitetura do Sistema

Este trabalho tem o propósito de utilizar os AG para a geração de uma rota dentro de uma cidade, e posteriormente à visualização desta mesma rota em um Ambiente Virtual. Permitido ao usuário um passeio pela cidade seguindo o trajeto da rota criada pelo algoritmo.

O sistema pode ser resumido como sendo a integração dos Algoritmos Genéticos (interface 2D) com RV (interface 3D modelada e programada em OpenGL). A Figura 4.2 ilustra essa arquitetura que reúne as duas áreas.

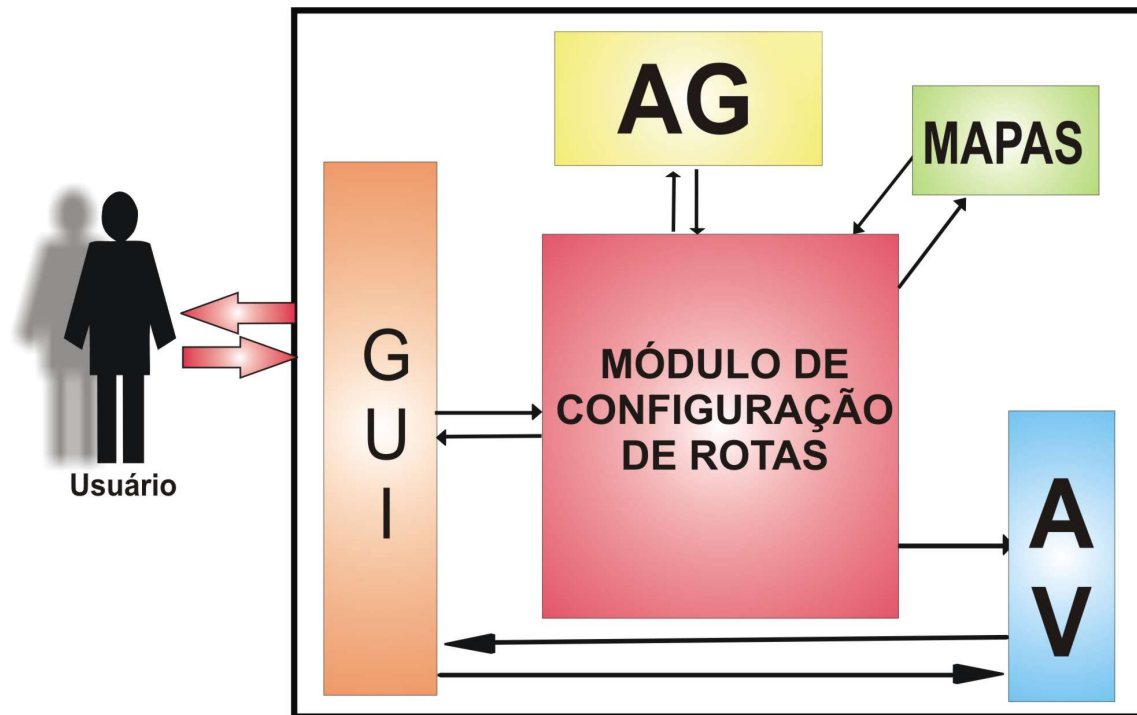


Figura 4.2. Arquitetura do Sistema - Integração entre AG e RV.

A arquitetura do sistema está dividida entre a interface do usuário, o módulo de AG, Mapas e AV.

4.3.1 Módulo Mapas

Os modelos geométricos dos mapas tanto 2D como 3D, foram construídos a partir de um modelo real extraído da cidade de Uberlândia – MG. Esses modelos foram desenhados com o uso do software de modelagem 3D Studio Max. O módulo Mapa é responsável pela leitura e conversão destes modelos para posterior uso no módulo de geração de rotas (Módulo AG) e visualização (Módulo AV).

Módulo AG

O módulo AG é responsável pela geração de todas as rotas e tem como base o uso de coordenadas de todos os cruzamentos presentes no módulo mapas. Este módulo oferece ao módulo AV a sequência de pontos necessários para a geração da rota virtual.

4.3.3 Módulo AV

O módulo AV é a parte da arquitetura responsável pela visualização e navegação do Ambiente virtual que representa a cidade. Este módulo tem como

principal tecnologia a biblioteca OpenGL responsável pela renderização dos objetos geométricos do ambiente.

4.4 Engenharia de Software/Engenharia de Requisitos

O processo de construção do sistema utilizou-se de mecanismos e instrumentos de Engenharia de Software para especificação e criação do projeto do software. Para tanto foram utilizados diagramas UML (*Unified Modeling Language*) que permitisse a exposição do projeto de forma eficaz e planejada.

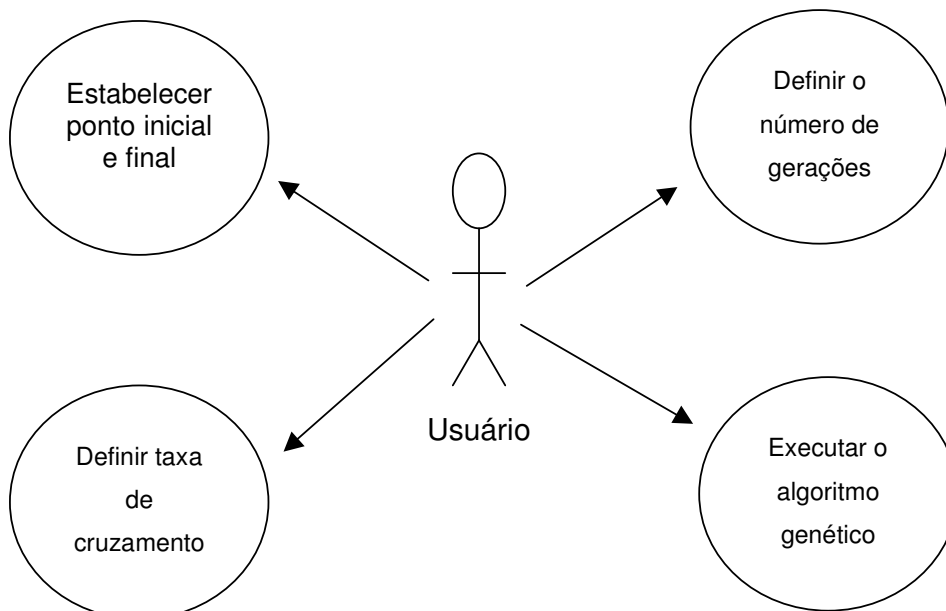
4.4.1 Diagrama de Caso de Uso (Use Case)

Casos de Uso ou Use Cases são descrições narrativas dos processos do sistema, envolvendo atores que estimulam ou recebem alguma coisa do use case. Podem ser definidos também como histórias ou casos de uso sobre o sistema e possuem dependência dos requisitos do sistema (Larman 1998). Os use cases são identificados através da leitura dos documentos de requisitos e de reuniões com os usuários envolvidos (brainstormings) (Larman 1998) e, em geral, são categorizados em (Larman 1998) primário, secundário e opcional.

Os use cases primários representam processos comuns do sistema (freqüentemente utilizados); os secundários representam processos pequenos e/ou raros (raramente utilizados); e os opcionais representam processos que podem não ser desenvolvidos ou utilizados.

4.4.2 Casos de Uso

A Figura 4.4 mostra o diagrama de Use-Case do sistema desenvolvido.

Figura 4.4. Diagrama de Use-Case.**Tabela 4.1. Relação dos Casos de Uso.**

Func.	Nome da Funcionalidade	Relação dos Casos de Uso
1	Estabelecer ponto inicial e final	Estabelecer ponto inicial e final no mapa
2	Definir taxa de cruzamento	Definir taxa de cruzamento para o algoritmo
3	Definir número de gerações	Definir número de gerações para o algoritmo
4	Executar o AG	Executar o algoritmo em busca da melhor solução
5	Gerar Rota Virtual	Gerar o Modelo 3D com base na melhor solução

4.4.2.1 Caso de Uso: Estabelecer ponto inicial e final no mapa

→ Diagrama:

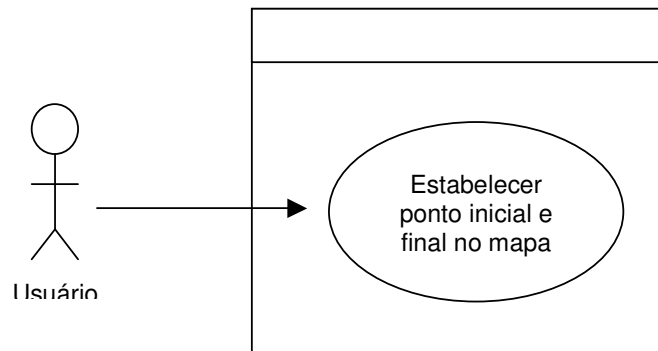


Figura 4.5. Use-Case “Estabelecer ponto inicial e final no mapa”.

→ Descrição:

Tabela 4.2. Descrição do Caso de Uso ‘Estabelecer ponto inicial e final no mapa’.

Nome do Caso de Uso: Estabelecer ponto inicial e final no mapa	Funcionalidade: Definição dos pontos de origem e de destino para o percurso no mapa.
Ator envolvido: Usuário (pessoa que irá fazer uso do sistema fornecendo as informações que estão disponíveis para que o mesmo possa gerar um resultado satisfatório).	
PRÉ-CONDIÇÃO 1. O mapa deve estar desenhado e com os pontos e cruzamentos pré-definidos.	
DESCRIÇÃO DO CASO 1. Indicar o ponto inicial (de origem) da rota. 2. Indicar o ponto final (de destino) da rota.	
PÓS-CONDIÇÃO 1. Os pontos (inicial e final) serão cadastrados no mapa e no algoritmo.	

4.4.2.2 Caso de Uso: Definir taxa de cruzamento para o algoritmo

→ Diagrama:

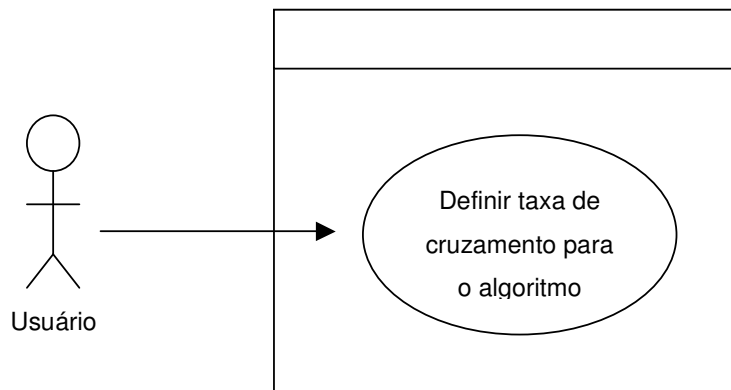


Figura 4.6. Use-Case “Definir taxa de cruzamento para o algoritmo”.

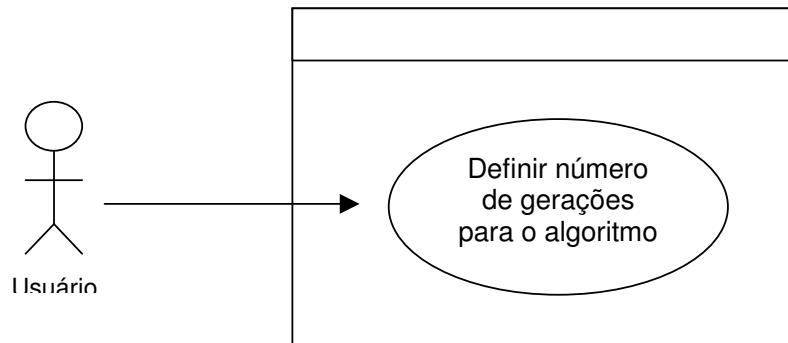
→ Descrição:

Tabela 4.3. Descrição do Caso de Uso ‘Definir taxa de cruzamento para o algoritmo’.

Nome do Caso de Uso: Definir taxa de cruzamento para o algoritmo	Funcionalidade: Definição da taxa de cruzamento do AG
Ator envolvido: Usuário (pessoa que irá fazer uso do sistema fornecendo as informações que estão disponíveis para que o mesmo possa gerar um resultado satisfatório).	
PRÉ-CONDIÇÃO 1. Os pontos em que se deseja realizar o percurso devem estar estabelecidos.	
DESCRIÇÃO DO CASO 1. Indicar a taxa de cruzamento desejada para o algoritmo.	
PÓS-CONDIÇÃO 1. O algoritmo será executado e a população evoluirá de acordo com a taxa de cruzamento estabelecida.	

4.4.2.3 Caso de Uso: Definir número de gerações para o algoritmo

→ Diagrama:

**Figura 4.7. Use-Case “Definir número de gerações para o algoritmo”.**

→ Descrição:

Tabela 4.4. Descrição do Caso de Uso ‘Definir número de gerações para o algoritmo’.

Nome do Caso de Uso: Definir número de gerações para o algoritmo	Funcionalidade: Definição do número de gerações do AG
Ator envolvido: Usuário (pessoa que irá fazer uso do sistema fornecendo as informações que estão disponíveis para que o mesmo possa gerar um resultado satisfatório).	
PRÉ-CONDIÇÃO 1. Os pontos em que se deseja realizar o percurso devem estar estabelecidos.	
DESCRIÇÃO DO CASO 1. Indicar o número de gerações desejadas para o algoritmo.	
PÓS-CONDIÇÃO 1. O algoritmo será executado e a população evoluirá de acordo com a quantidade de gerações estabelecida.	

4.4.2.4 Caso de Uso: Executar o algoritmo em busca da melhor solução

→ Diagrama:

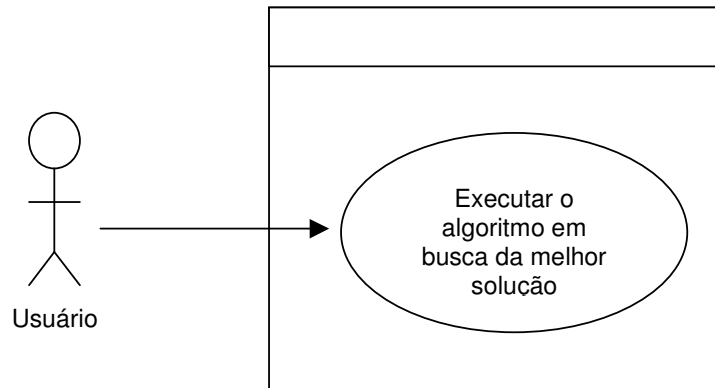


Figura 4.8. Use-Case “Executar o algoritmo em busca da melhor solução”.

→ Descrição:

Tabela 4.5. Descrição do Caso de Uso ‘Executar o algoritmo em busca da melhor solução’.

Nome do Caso de Uso: Executar o algoritmo em busca da melhor solução	Funcionalidade: Execução e evolução do AG em busca da melhor solução
Ator envolvido: Usuário (pessoa que irá fazer uso do sistema fornecendo as informações que estão disponíveis para que o mesmo possa gerar um resultado satisfatório).	
PRÉ-CONDIÇÃO 1. Os pontos em que se deseja realizar o percurso devem estar estabelecidos. 2. A taxa de cruzamento tem que estar definida. 3. A quantidade de gerações tem que estar definida.	
DESCRIÇÃO DO CASO 1. Clicar no botão de evolução.	
PÓS-CONDIÇÃO 1. O algoritmo será executado e a população evoluirá até encontrar a melhor solução, ou seja, a melhor rota ou percurso.	

4.5 Considerações Finais

A arquitetura do sistema segue a modelagem conceitual do AG Clássico, com um fluxo básico de execução composto pela codificação do cromossomo, criação da população, definição da aptidão, seleção dos indivíduos e cruzamentos dos indivíduos na população. O melhor indivíduo é base para construção da rota 3D ou rota virtual.

Além disso, o diagrama de Use Case representa os principais processos que podem ser utilizados pelo usuário na execução do algoritmo, com o objetivo de se chegar à uma melhor rota no mapa.

CAPÍTULO V

5 DETALHES DA IMPLEMENTAÇÃO

5.1 Introdução

Este capítulo apresenta a implementação do sistema, abordando todas as fases necessárias à execução do AG e da conversão para Ambiente Virtual. A implementação do modelo 3D começa com a modelagem dos objetos que serão utilizados no ambiente virtual. A linguagem utilizada foi o *Object Pascal*, utilizando-se a versão 6.0 do ambiente de programação Borland Delphi TM.

5.2 Estrutura do Cromossomo

A rota possui diversos pontos que podem ser visitados durante o percurso, cada um desses pontos possui suas coordenadas cartesianas e o conjunto desses pontos representa um indivíduo.

A Figura 5.1 ilustra a estrutura de um individuo que irá compor a população.

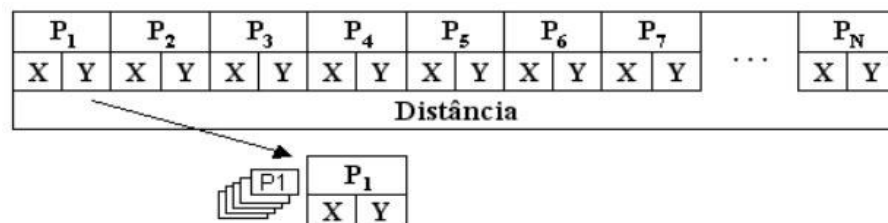


Figura 5.1. Estrutura do Cromossomo (Indivíduo).

CAPITULO V – DETALHES DA IMPLEMENTAÇÃO

A Figura 5.2 mostra o código que gera todos os indivíduos da população. Cada um desses é formado por um vetor de coordenadas que representam cada um dos cruzamentos (pontos) do mapa. Cada um desses pontos possui suas coordenadas X e Y , que representam a sua posição em determinado local do mapa. Para cada ponto P , existe outro vetor de possibilidades de tráfego. Um indivíduo válido é aquele que possui uma seqüência válida entre o ponto inicial e o ponto final e, o melhor indivíduo é aquele que possui a menor distância entre esses pontos.

```
Para i=1 até TamanhodaPopulacao Faça
Inicio
  pop[i].ind[1] := ptini.ind;
  for j:=2 to 500 do
  begin
    qtc := 0;
    for k:=1 to 8 do
    begin
      if (Populacao.Individuo[pop[i].ind[j-1]].conexoes[k] <> 0) then
        Inc(qtc);
    end;
    tent := 0;
    repeat
      Inc(tent);
      cruz := RandomRange(1,qtc+1);
      pop[i].ind[j] := Populacao.Individuo[pop[i].ind[j-1]].
conexoes[cruz];
    until ((pop[i].ind[j] <> pop[i].ind[j-2]) or (tent > 7));
    if (pop[i].ind[j] = ptfin.ind) then
    begin
      Inc(achou);
      break;
    end;
  end;
end;
Fim;
```

Figura 5.2. Pseudocódigo da geração da população.

5.3 Geração do Melhor Indivíduo

O melhor indivíduo encontrado pela execução do algoritmo genético é um conjunto seqüencial de coordenadas que sugerem o melhor caminho a ser percorrido pelo usuário do sistema tendo como referências o ponto inicial e o ponto final. A Figura 5.3 ilustra a composição desse indivíduo.

Pi	P1	P2	P3	P4	P5	P6	P7	P8	P9	Pf
X_i, Y_i	X_1, Y_1	X_2, Y_2	X_3, Y_3	X_4, Y_4	X_5, Y_5	X_6, Y_6	X_7, Y_7	X_8, Y_8	X_9, Y_9	X_f, Y_f

Figura 5.3. Composição do indivíduo final.

O indivíduo final é a melhor solução encontrada pelo algoritmo, ou seja, a menor distância válida entre o ponto inicial e o ponto final. Essa composição é conjunto de coordenadas que serão os apontadores para objetos (cruzamentos) que estão no modelo em três dimensões.

5.4 Definição da Aptidão (Fitness)

A aptidão da população é avaliada de acordo com a menor distância da rota encontrada. A distância da rota é obtida, somando-se as distâncias entre o ponto de origem e de destino escolhidos, e analisando-se as distâncias entre os possíveis pontos intermediários que serão percorridos. E a cada nova iteração, essa aptidão é reavaliada.

Quanto menor for a distância percorrida, mais evoluída a população está e mais apta estará para resolver a problemática da execução do algoritmo. Assim, chegará um determinado momento em que a aptidão da população não mais se modificará, diz-se então, que a população chegou em seu estágio máximo de evolução, o que caracteriza a parada da execução do algoritmo.

Primeiramente é necessário definir que cada indivíduo será formado por um conjunto de ponto, onde cada ponto representa uma definição de mudança de direção na rota, especificamente esquinas ou cruzamentos de uma cidade ou organização similar, e é composto por uma coordenada (x, y). A Figura 5.4 representa a equação de aptidão, em que *dist* representa a distância da rota.

$$f(a) = \frac{1}{dist}$$

Figura 5.4. Equação para cálculo da aptidão.

A Figura 5.5 representa o pseudocódigo que para geração da aptidão de cada indivíduo.

```
Para i=1 até TamanhodaPopulacao Faça
  Inicio
  for j:=1 to 500 do if pop[i].ind[j] = 0 then break;
  pontos := j-1;
  Pop[i].distancia := 0.0;
  for j:=1 to pontos-1 do
    begin
      x      :=      Populacao.individuo[Pop[i].ind[j]].cruz.x      -
Populacao.individuo[Pop[i].ind[j+1]].cruz.x;
      y      :=      Populacao.individuo[Pop[i].ind[j]].cruz.y      -
Populacao.individuo[Pop[i].ind[j+1]].cruz.y;
      try
        Pop[i].distancia := Pop[i].distancia +sqrt(sqr(x)+sqr(y));
      except
        ShowMessage('Erro      no      cálculo      de      distância:
'+FormatFloat('0.00',x)+'-' +FormatFloat('0.00',y));
      end;
    end;
  Pop[i].aptidao := (1 / Pop[i].distancia) * 10000;
Fim;
```

Figura 5.5. Composição do indivíduo final.

O pseudocódigo da Figura 5.5 apresenta um método para calculo da distância de um ponto ao outro por meio da soma da distancia dos pontos intermediários. Por se tratar de valores dos eixos X e Y, foi usada uma equação que propiciasse o calculo da distancia entre um ponto $P1(x1, y1)$ e outro ponto $P2(x2, y2)$: raiz quadrada $((x2-x1)^2 + (y2-y1)^2)$.

5.5 Codificação do Cromossomo

O cromossomo (indivíduo) foi definido como uma estrutura de dados do tipo “array” (vetor) dinâmico com tamanho calculado em função da quantidade de cruzamentos selecionados.

O cromossomo é composto por um conjunto de pontos (representando as coordenadas x e y) e pelo seu índice ou posição. Além desses, ainda engloba a sua aptidão (representando a sua distância no percurso) e as suas conexões (pontos por onde pode passar).

5.6 Criação da População

A população é formada por um conjunto de indivíduos, sendo que cada um representa uma rota. Por meio dos operadores genéticos (seleção e cruzamento), a população irá evoluir e encontrar ou não, a melhor rota.

CAPITULO V – DETALHES DA IMPLEMENTAÇÃO

A Figura 5.6 mostra a representação da população, que é composta por um determinado conjunto de indivíduos I , que irão ser selecionados e cruzados entre si a fim de encontrar a melhor solução, ou seja, o melhor caminho a ser percorrido.

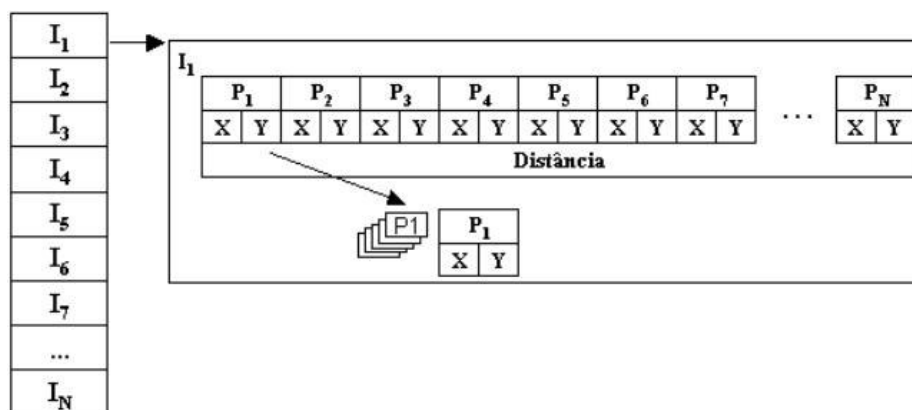


Figura 5.6. Estrutura da População.

A população é um conjunto de indivíduos, gerados aleatoriamente e tendo como princípio o ponto de saída ou de origem (índice 1) e o ponto de chegada ou de destino (índice n). A seqüência de pontos compreendidos entre esse intervalo será dependente do número de pontos compreendidos entre os índices inicial e final. Esses pontos presentes na configuração inicial do sistema referem-se aos locais por onde uma pessoa tem que passar, para que possa ir de um lugar a outro.

A população foi definida como uma estrutura de dados do tipo “array” (vetor) dinâmico que aponta para o primeiro vetor criado para armazenar todos os pontos possíveis do cenário escolhido.

5.7 Método de Seleção

O método empregado para a seleção dos indivíduos para o cruzamento é o método de seleção por roleta (descrito no item 2.2.6), uma vez que é o método mais utilizado na maioria das implementações de AG's e mostrou-se eficaz na solução problema específico de rotas.

A roleta representa em sua totalidade a somatória das aptidões dos cromossomos da população e cada cromossomo adquire percentualmente uma “fatia” dessa roleta, conforme sua aptidão individual. Assim, aqueles indivíduos que possuem uma aptidão maior, são os que caracterizam uma distância menor do

CAPITULO V – DETALHES DA IMPLEMENTAÇÃO

percurso, e, portanto, tem uma fatia maior da roleta e maior chance de serem selecionados.

Pode-se representar a seleção pelo algoritmo genético em questão pelos seguintes passos:

- ✓ Somar todas as aptidões das populações.
- ✓ Calcular o percentual individual de cada indivíduo em relação à população.
- ✓ Selecionar um valor percentual aleatoriamente.
- ✓ Percorrer o vetor representando a roleta, selecionando um indivíduo.
- ✓ Selecionar um valor percentual aleatoriamente.
- ✓ Percorrer o vetor representando a roleta, selecionando um indivíduo.
- ✓ Comparar os indivíduos que foram selecionados.
- ✓ Se estes forem iguais, retornar ao passo 5, caso contrário, realizar o cruzamento.

A Figura 5.7 apresenta o pseudocódigo da seleção por roleta.

```
// soma todas as distancias
SomaAptidao:=0;
for i := 1 to MaxPop do
begin
    SomaAptidao := SomaAptidao + Pop[i].aptidao;
end;

// calcula a probabilidade (porcentagem) de cada individuo na roleta
for i := 1 to MaxPop do
begin
    Pi[i] := Pop[i].aptidao / SomaAptidao;
end;
Ci[1] := Pi[1];
for i := 2 to MaxPop do
begin
    Ci[i] := Ci[i-1] + Pi[i];
end;
//METODO ROLETA - Escolhe aleatoriamente o elemento para fazer o
cruzamento
E1 := SeleccionaIndividuo((RandomRange(0,100)/100));
repeat
    E2 := SeleccionaIndividuo((RandomRange(0,100)/100));
until E1 <> E2;
```

Figura 5.7. Pseudocódigo da seleção por roleta.

De acordo com o código da Figura 5.7 o primeiro passo foi a totalização de todas as aptidões dos indivíduos (SomaAptidão). Em seguida é calculada a porcentagem de aptidão de cada individuo (P_i) usando o total da aptidão. Para finalizar é calculada a porção que cada individuo terá na roleta simulada (C_i). Na escolha do individuo um número é sorteado aleatoriamente numa faixa de 0 a 100, mesmo conjunto de valores calculados para a roleta, portanto o individuo escolhido é aquele que possui na sua porção o número sorteado (SeleccionaIndividuo).

5.8 Método de Cruzamento

O método de cruzamento PMX (descrito no item 2.2.7), é o operador que permite uma maior variação na troca dos genes entre os cromossomos envolvidos no cruzamento, sem propiciar a repetição de um gene referente ao mesmo ponto, e, também, é o mais utilizado na maioria das implementações de AG's. Assim, foi utilizado um operador de cruzamento baseado no método do PMX, caracterizando um híbrido com determinadas variações necessárias de acordo com o problema específico. A Figura 5.8 apresenta o pseudocódigo do cruzamento, que faz uso do elitismo para preservar os melhores indivíduos.

```
Randomize();
for i:=1 to 500 do if pop[E1].ind[i] = 0 then break;
    pontos1 := i-1;
for i:=1 to 500 do if pop[E2].ind[i] = 0 then break;
    pontos2 := i-1;
tents :=0;
compativel := false;
repeat
    inc(tents);
    Corte1 := RandomRange(3,Pontos1-2);
    for i:=pontos2 downto 1 do
        if (pop[E2].ind[i] = pop[E1].ind[Corte1]) then
            begin
                compativel := true;
                Corte2 := i;
            end;
        if tents > 10 then break;
until compativel;
if (Corte1 + (Pontos2-Corte2)) > 500 then exit;
if (Corte2 + (Pontos1-Corte1)) > 500 then exit;
for i:=1 to 500 do
begin
    aux[i] := pop[E1].ind[i];
end;
filho1 := E1; filho2 := E2;
if (E1 = 1) then begin
    for i:=1 to 500 do pop[maxpop].ind[i] := pop[E1].ind[i];
    filho1 := MaxPop;
end;
if (E2 = 1) then begin
    for i:=1 to 500 do pop[maxpop].ind[i] := pop[E2].ind[i];
    filho2 := MaxPop;
end;
j := Corte2;
for i:=Corte1 to 500 do
begin
    Pop[filho1].ind[i] := Pop[E2].ind[j];
    Inc(j);
    if j > 500 then break;
end;
j := Corte1;
for i:=Corte2 to 500 do
begin
    Pop[filho2].ind[i] := Aux[j];
    Inc(j);
    if j > 500 then break;
end;
```

Figura 5.8. Pseudocódigo do Cruzamento.

O pseudocódigo da Figura 5.8 apresenta primeiramente a verificação da compatibilidade de cruzamento dos dois indivíduos selecionados (E1 e E2). Após a verificação um ponto de corte é escolhido aleatoriamente (Corte1) no primeiro indivíduo, e em seguida o índice que representa o primeiro ponto de corte é localizado no segundo indivíduo (Corte2). As partes posteriores aos pontos de corte são trocadas finalizando o processo de cruzamento.

CAPITULO V – DETALHES DA IMPLEMENTAÇÃO

A Figura 5.9 ilustra o método de cruzamento do híbrido do cruzamento binário, que ocorre entre dois indivíduos selecionados, denominados *pais*, recombinaando seus genes para originar outros indivíduos, denominados *filhos*, assim como no sistema biológico natural. O processo de cruzamento obedece a uma restrição eliminatória. Um ponto de corte aleatório, aponta para um cruzamento (ponto ou coordenada x e y). Para haver o cruzamento é necessário que o outro indivíduo escolhido também possua esse mesmo ponto, pois o indivíduo é constituído de uma seqüência válida de coordenadas. Dessa forma é possível melhorar o indivíduo sempre que uma parte considerada ruim (maior distância) seja trocada por uma outra parte considerada boa (menor distância).

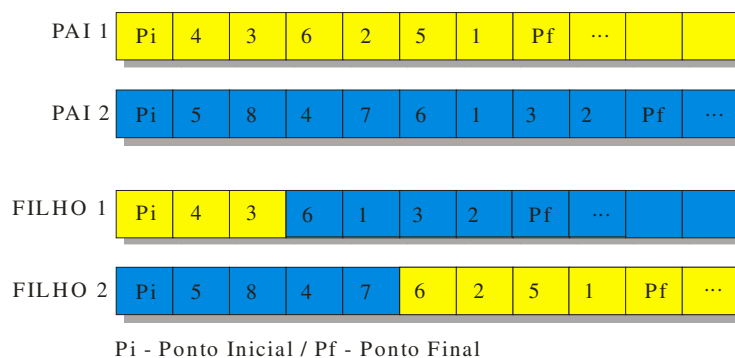


Figura 5.9. Exemplificação do cruzamento do híbrido do PMX.

O método de cruzamento implementado pode ser descrito pelos seguintes passos:

- ✓ Definir o primeiro ponto de corte.
- ✓ Verificar compatibilidade entre os indivíduos.
- ✓ Repetir: alterar os genes do indivíduo A com os genes do indivíduo B, entre os pontos de corte, analisando as possíveis conexões entre os pontos.

5.9 Criação dos objetos

No protótipo são encontrados objetos que foram modelados por meio de um modelador 3D (*3D Studio Max*). Este *software* permite a modelagem de objetos bem como a renderização de imagens e animação de cenas. Na Figura 5.10 é demonstrado o ambiente do *software* com os objetos modelados, alguns destes

CAPITULO V – DETALHES DA IMPLEMENTAÇÃO

objetos utilizam formas básicas para a sua modelagem como esferas, cones e cubos.

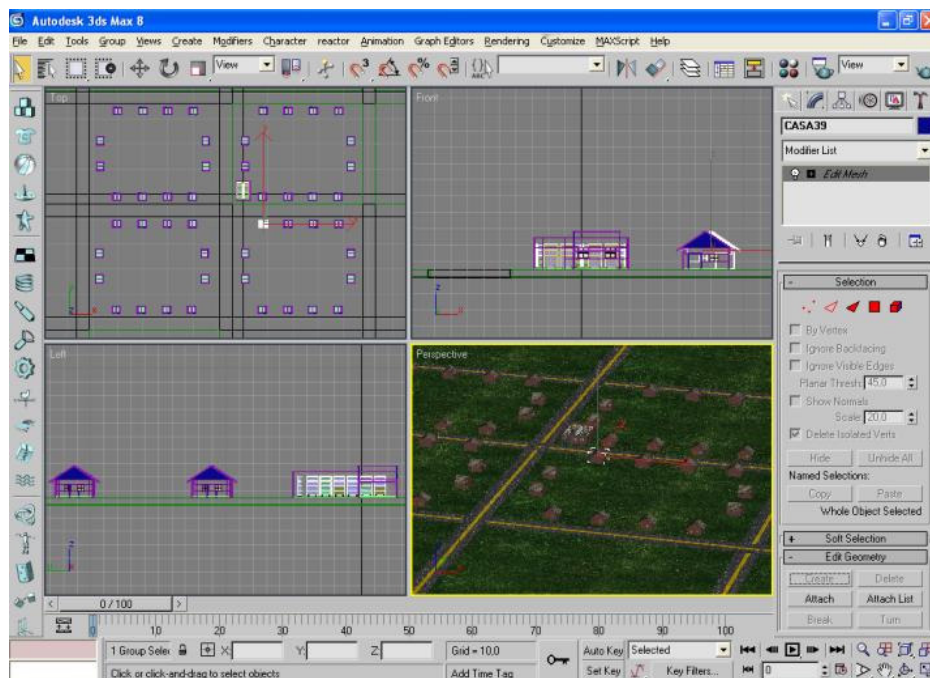


Figura 5.10. Modelador 3D (3D Studio Max).

5.10 Modelagem Geométrica

A modelagem geométrica teve início com o desenho de todos os componentes do bairro usado neste protótipo. A base para o desenho foi o modelo 2D, portanto o modelo 3D segue exatamente as proporções e dimensões do modelo anterior. Alguns objetos foram modelados seguindo com exatidão o objeto real (um prédio público, uma referência histórica, e outros que merecem destaque), outros objetos foram modelados seguindo uma generalização (casas e outros prédios). Cada cruzamento do trecho urbano é um objeto e pode ser selecionado pelo usuário para permitir uma prévia configuração do sistema (base de dados de cruzamentos, definições de direções do cruzamento em relação a outros). Outros objetos da cena são os espaços prediais, também selecionáveis e passíveis a associações a uma biblioteca de outros objetos (casas, prédios, praças, e outros considerados turísticos: aeroporto, zoológico, lagos, e etc.). Esses objetos são selecionáveis mediante o comando OpenGL *GLPicking*. O resultado dessa modelagem construída com o uso

do *software 3D Studio MAX* foi transformada para um arquivo com extensão *.3DS*, o qual possui detalhes geométricos divulgados pelo fabricante permitindo o uso do mesmo.

5.11 Conversão de Formatos

A conversão do formato 3DS conforme a Figura 5.11, foi construída usando o ambiente de programação Borland Delphi TM e baseia na geração de um objeto contendo como propriedades matrizes de índices, vértices, normais, materiais e texturas do modelo geométrico transformando em triângulos. O principal método desse objeto é a função *draw*, que por meio da diretiva *GL_TRIANGLES* desenha todo o objeto na cena. A Figura 5.11 apresenta o código que realiza a leitura do arquivo 3DS.

```
CreateGLContext(Handle)
Model:=T3DModel.Create
Model.LoadFromFile('cidade.3ds')
glEnable(GL_LIGHT0)
glEnable(GL_LIGHTING)
glPointSize(4)
EnviromentMap
```

Figura 5.11. Código para importação do arquivo 3DS.

5.12 Navegação e Visualização

O processo de navegação baseia-se na seqüência de pontos, gerada pelo algoritmo genético, e, se não houver interferência do usuário, o resultado final será um passeio pela cidade virtual na rota estabelecida. O usuário poderá acelerar o passeio, diminuir a velocidade, parar e dar um giro completo sobre o seu eixo.

Para tanto o usuário dispõe de teclas configuradas para essa operação, essas teclas são: **SETA PARA CIMA, SETA PARA BAIXO, SETA PARA ESQUERDA E SETA PARA DIREITA**. A sensação de navegação é propiciada por transformações geométricas sofridas pelo ambiente, ou seja, o ponto de visão é sempre o mesmo, o que altera é o posicionamento do ambiente. A Figura 5.12 apresenta o código responsável por essa navegação e visualização do ambiente.

```
if Msg.wParam = VK_Down then begin
    TX := TX - sin(ROT*pi/180);
    TZ := TZ + cos(ROT*pi/180); end;
if Msg.wParam = VK_Up then begin
    TX := TX + sin(ROT*pi/180);
    TZ := TZ - cos(ROT*pi/180);
if Msg.wParam = VK_Left then begin
    ROT := ROT - 1;
    IF ROT < 0 THEN ROT := ROT + 360;
if Msg.wParam = VK_Right then begin
    ROT := ROT + 1;
    IF ROT > 360 THEN ROT := ROT - 360;
glRotate(ROT, 0, 1, 0);
glTranslatef(TX, TY, TZ);
```

Figura 5.12 Código que implementa a navegação e visualização do ambiente.

De acordo com a Figura 5.12 é possível perceber existe uma transformação geométrica de rotação (`glRotate`) e uma outra de translação (`glTranslate`) calculadas de acordo com o posicionamento inicial do ambiente e possíveis com o uso das funções seno e co-seno do ângulo atual em que o objeto se encontra. Para a navegação sem interferência do usuário, ou seja, a sugerida pelo Algoritmo Genético, foi necessário calcular o ângulo para cada cruzamento da seqüência, e o deslocamento na direção do cruzamento para calcular o novo ângulo para o seguinte até encerrar o trajeto. A Figura 5.13 ilustra o código implementado para essa situação.

```
PXI := PontoXInicial; PXF := PontoXFinal;
PYI := PontoYInicial; PYF := PontoYFinal;
DeltaX := (PXI - PXF); DeltaY := (PYI - PYF);
ângulo := RadtoDeg(ArcSin(abs(DeltaX / Hypot(DeltaX, DeltaY))));
ROT := RotacaoOriginal + Angulo;
TX := TX + 0.1 * sin(ROT*pi/180);
TZ := TZ - 0.1 * cos(ROT*pi/180);
If TX = PontoXFinal and TY = PontoYFinal
    RecalculaAngulo;
```

Figura 5.13. Pseudo-código que implementa a navegação automática.

A visualização da rota virtual inicia com o posicionamento do ponto de visão sobre o objeto que representa o ponto inicial. E, para ter a sensação de estar por exemplo dentro de um carro com um posicionamento acima do solo e tendo como referência um ponto de visão baseado no horizonte foi usada uma diretiva OpenGL de visualização (`gluLookAt(0, 8, 0, 0, 5, 120, 0, 1, 0)`). Outras interações possíveis no ambiente (barra de navegação – Figura 5.15) foram possíveis por meio de outros métodos do objeto criado. Apesar de não ser uma tendência na área de Realidade

Virtual, o uso de uma barra tradicional foi usada devido aos altos custos de implementação de um sistema de Realidade Virtual Imersiva.

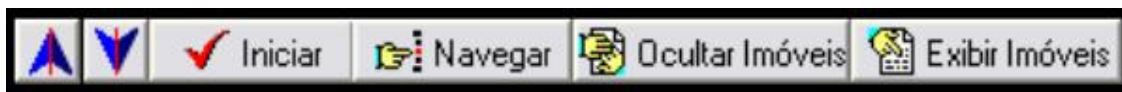


Figura 5.14. Barra de navegação do ambiente.

5.13 Desenvolvimento do Ambiente Virtual

O Ambiente Virtual, como já foi dito anteriormente, foi criado por meio do uso da biblioteca gráfica OpenGL, sendo necessário o uso do ambiente de programação Borland Delphi TM 6.0 para compilação do código e implementação da interação do usuário com o ambiente e para as animações necessárias. A estrutura construída do ambiente virtual foi desenvolvida para permitir que qualquer cidade modelada possa ser manipulada na geração de rotas.

5.14 Considerações Finais

A implementação do sistema foi realizada utilizando-se uma linguagem de programação orientada a objetos e a eventos, disponibilizando recursos para a criação de uma estrutura de dados que modelasse de forma satisfatória a problemática do algoritmo, o que facilitou bastante a implementação de uma interface amigável ao usuário, bem com a programação de funções e estruturas específicas da modelagem conceitual do Algoritmo Genético e para geração de um Ambiente Virtual.

CAPÍTULO VI

6 FUNCIONAMENTO DO SISTEMA

6.1 Introdução

Este capítulo descreve a forma de funcionamento do sistema, os parâmetros de entrada necessários para a execução do algoritmo e os passos para que o processamento do sistema seja desempenhado de forma correta. Também apresenta as funcionalidades do ambiente virtual que simula o bairro de uma cidade.

6.2 Parâmetros Genéticos

Para o funcionamento do sistema são necessários alguns parâmetros de entrada, os quais irão guiar o processamento do AG em busca da melhor solução, ou seja, da melhor rota a ser seguida, como mostra a Figura 6.1, esses parâmetros podem ser configurados dentro do próprio sistema.

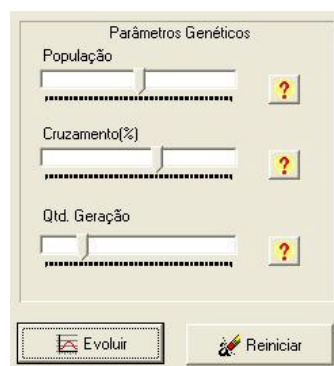


Figura 6.1. Parâmetros Genéticos.

Tais parâmetros são:

- ✓ **Tamanho da população:** este parâmetro define o tamanho da população, isto é, a quantidade de indivíduos a serem gerados na população.
- ✓ **Taxa de cruzamento:** este parâmetro define a probabilidade com que um cruzamento ocorrerá dentro da população, ou seja, define a porcentagem de indivíduos que irão participar do cruzamento. O valor pré-definido é de 60%, pois esta é a taxa adotada na maioria das implementações de Algoritmos Genéticos, no entanto, o usuário pode alterar esse valor conforme sua necessidade.
- ✓ **Quantidade de gerações:** este parâmetro define o número de gerações que deverão ocorrer no AG em busca da melhor solução. Este valor é determinado pelo usuário de acordo com a problemática específica.

6.3 Processamento e Resultado

Inicialmente, o sistema apresentará além dos parâmetros genéticos, parâmetros relacionados aos cruzamentos existentes que serão usados na rota, os quais deverão ser definidos pelo usuário ou carregados diretamente, conforme mostra a Figura 6.2, onde existe um menu para a configuração das conexões.



Figura 6.2. Menu Conexões.

A partir deste menu o usuário poderá:

- ✓ **Iniciar Conexão:** Inicia o processo de conexão que consiste em definir diretamente no mapa quais cruzamentos serão usados.
- ✓ **Definir Conexões:** Definir a direção de cada conexão em relação à outra (caso de ruas de mão única, esse passo é importante).

- ✓ **Ponto Inicial:** Essa é uma das principais opções da interface, é responsável por definir qual será o ponto de partida.
- ✓ **Carregar Cruzamentos:** Opção que possibilita o carregamento automático de todos os pontos previstos como cruzamento no mapa.
- ✓ **Limpar Cruzamentos:** Essa opção é inversa a opção anterior, limpando todos os cruzamentos cadastrados.

Após a configuração de cada conexão entre os pontos, o sistema apresentará a interface principal, conforme mostra a Figura 6.3.



Figura 6.3. Interface inicial do sistema.

A Figura 6.4 ilustra também a interface com os cruzamentos já escolhidos e o algoritmo disparado mostrando o gráfico da relação da distância para geração.



Figura 6.4. Ponto Inicial e Ponto Final Definidos.

A Figura 6.5 ilustra também a interface com os cruzamentos já escolhidos e o algoritmo disparado mostrando o gráfico da relação da distância para geração.

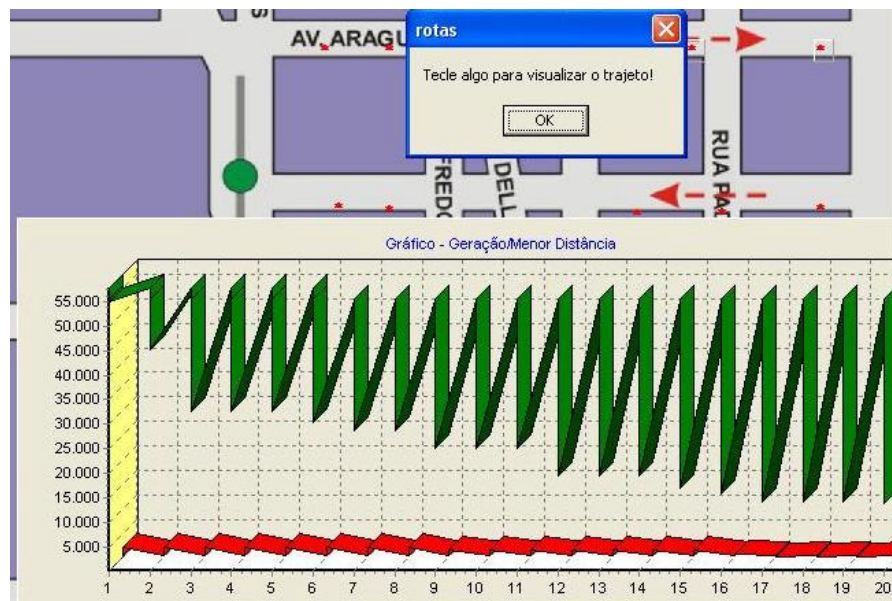


Figura 6.5. Gráfico Menor Distância.

O sistema após o processamento demonstra a melhor rota encontrada, que poderá ou não ser realmente a melhor possibilidade, e irá plotar a mesma sobre o mapa, conforme demonstrado na Figura 6.6.



Figura 6.6 Interface após processamento.

Se o usuário deseja escolher outros cruzamentos a serem participantes da trajetória, ou buscar uma nova rota baseadas nos mesmos pontos, deverá clicar no botão “REINICIAR”. Para reiniciar o processamento do algoritmo, após a redefinição ou não dos parâmetros. Logo após deverá clicar novamente no botão “EVOLUIR” e o sistema irá plotar no mapa uma nova rota. Conforme mostra a Figura 6.7:

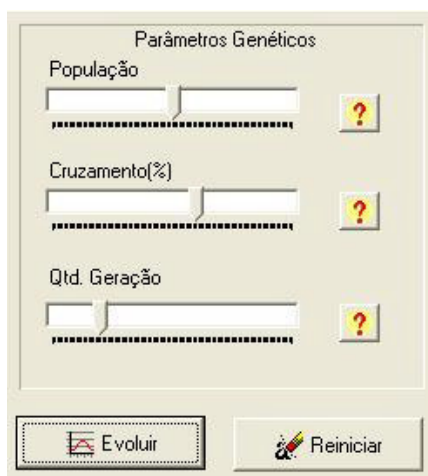


Figura 6.7 Botões Evoluir e Reiniciar.

CAPITULO VI – FUNCIONAMENTO DO SISTEMA

A interface 3D além de oferecer ao usuário uma navegação pela rota gerada pelo algoritmo, que pode ser visualizada através do menu Modelo de Visão, conforme a Figura 6.8.

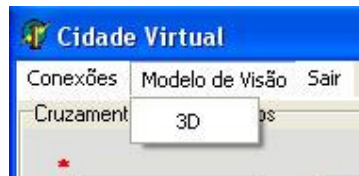


Figura 6.8 Menu Modelo de Visão.

A seleção do modelo de visão 3D leva a interface 3D que melhor visualizada pela Figura 6.9



Figura 6.9 Visão superior da Cidade Virtual.

A barra de navegação existente no sistema, apresenta funções para o auxílio na navegação pelo ambiente, conforme apresentamos na Figura 6.10.

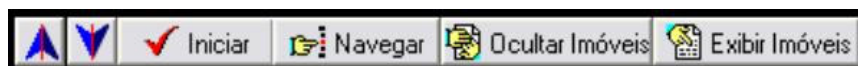
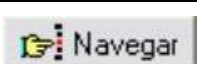
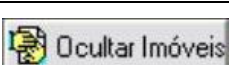
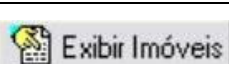


Figura 6.10. Barra de Navegação do Sistema 3D.

A tabela 6.1 mostra as funções de cada botão da barra de navegação do sistema 3D.

Tabela 6.1 Funções da Barra de Navegação.

ICONE	NOME	FUNÇÕES
	Navegação	Navegação (translação) no eixo Z do ambiente.
	Iniciar	Posição inicial da cidade com visão do topo.
	Navegar	Inicia a navegação por uma rota estabelecida ou permite ao usuário navegar livremente.
	Ocultar imóveis	Ocultar todos os imóveis da cidade mostrando apenas as ruas e cruzamentos.
	Exibir imóveis	Exibe todos os imóveis da cidade.

Ao iniciar o processo de Navegação do sistema, este vai direto ao ponto inicial determinado pelo usuário e posiciona para percorrer o caminho, e inicia o movimento. A cada “esquina” encontrada, é feita a parada e o reposicionamento para a direção correta do percurso, conforme a Figura 6.11.

O que torna o ambiente mais interativo é de que não há necessidade de seguir apenas o caminho definido pelo sistema, podendo o usuário interferir e parar a qualquer momento.

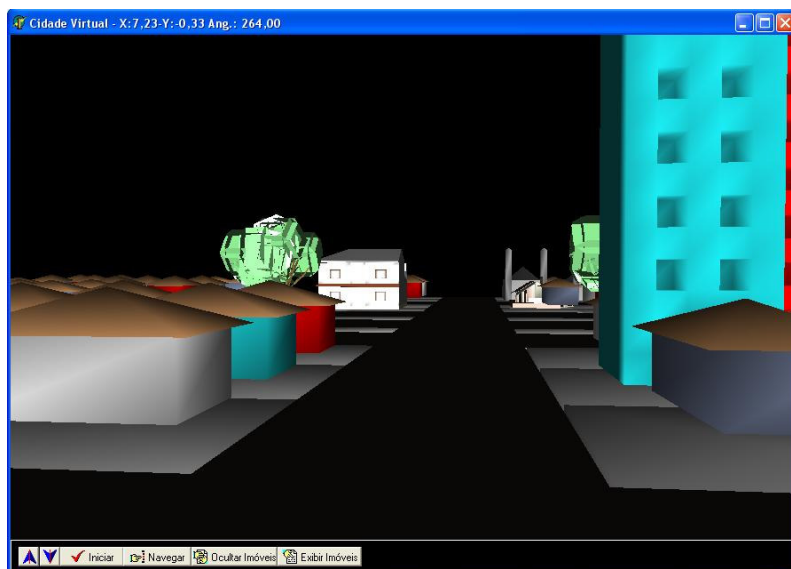


Figura 6.11 Interface 3D do percurso.

Este Modelo de Visão 3D também propicia a visão do ambiente sem os objetos (imóveis), que é acionada ocultando imóveis, conforme a Figura 6.12.



Figura 6.12 Visão do ambiente sem os objetos.

6.4 Considerações Finais

O sistema apresenta uma interface amigável para o usuário e fácil de ser utilizada, onde o mesmo não necessita de conhecimentos computacionais

específicos para utilizar o sistema, bastando apenas ter conhecimento do ambiente operacional.

É necessário somente a correta definição dos parâmetros, e caso o usuário não queira alterá-los, todos já estão pré-definidos, oferecendo suporte a um bom processamento do algoritmo. Além disso, é preciso também uma correta escolha dos cruzamentos participantes e definição das possíveis conexões.

CAPÍTULO VII

7 RESULTADOS / LIMITAÇÕES

7.1 Introdução

Neste capítulo, é apresentada uma análise do protótipo descrito nos Capítulo 5 e 6. Esta análise teve como objetivo demonstrar a viabilidade, a eficácia e eficiência do protótipo. A análise realizada abrange os seguintes aspectos:

- a) A obtenção da melhor rota por meio do Algoritmo Genético.
- b) Reprodução da rota em um Ambiente Virtual.

Estes resultados são constatados visualmente e não foi necessária a construção de uma metodologia específica para identificar o correto resultado

7.2 Ambiente Experimental

Como ambiente para o experimento, foi usado um microcomputador compatível com IBM PC (*Pentium IV 3.2 MHz; memória RAM 512 Mbytes e Disco Rígido de 80 Gbytes*). Não foi utilizada nenhuma placa de vídeo aceleradora, portanto a parte gráfica não necessita de periféricos específicos.

7.3 Análise dos Resultados Obtidos

O estudo de caso usado foi apenas um bairro da cidade de Uberlândia-MG, obedecendo às restrições de mão e contramão. Mesmo sendo um pequeno bairro, existem no mesmo, 75 cruzamentos o que provoca, dependendo da escolha do ponto inicial e do ponto final um número grande combinações. No entanto foi

possível analisar visualmente se algoritmo encontrou ou não a melhor solução. Os parâmetros genéticos (População, Taxa de Cruzamento e Quantidade de Gerações) devem ser configurados para exigir o menor esforço computacional possível. Dessa forma o sistema apresenta em sua interface inicial o resultado de testes que demonstraram que 90% dos casos os parâmetros genéticos foram suficientes:

- ✓ População: 50 indivíduos
- ✓ Taxa de Cruzamento: 50%
- ✓ Quantidade de Gerações: 20.

O processo de “paralisia” aconteceu freqüentemente nos testes demonstrando que a melhor solução poderia ser encontrada em alguns casos nas primeiras gerações. O sistema não possui opção de parada por paralisia, pois foram constatados nos testes, que algumas situações depois de um período de paralisia, havia sempre alteração na aptidão de indivíduos menos aptos, possibilitando também melhoria na aptidão do melhor indivíduo.

Para o teste do ambiente virtual, o parâmetro de avaliação adotado foi a verificação se a navegação obedecia a seqüência sugerida pelo algoritmo genético.

7.3.1 Comparação com outros trabalhos

De acordo com Tabela 7.1, o protótipo desenvolvido neste trabalho denominado de Rota Virtual atendeu todos os itens avaliados, sendo seu principal diferencial em relação aos outros trabalhos e a principal contribuição deste.

CAPITULO VII– RESULTADOS /LIMITAÇÕES

Tabela 7.1 Comparação com outros trabalhos

Softwares	Geração de Rotas	Uso de AG	Uso de RV	Integração AG + RV	Visualização Rota - 2D/3D
Aplicação da Generalização Cartográfica em Mundos Virtuais	NÃO	NÃO	SIM	NÃO	3D
Aplicação de Algoritmos Genéticos no Desenvolvimento de Rotas Turísticas	SIM	SIM	NÃO	NÃO	NÃO
Construção e Gestão da Complexidade de Cenários Urbanos 3D em Ambientes Virtuais Imersivos	NÃO	NÃO	SIM	NÃO	3D
O uso de Algoritmos Genéticos na solução de rotas em trechos urbanos	SIM	SIM	NÃO	NÃO	2D
Roteamento de Veículos Dinâmico usando Algoritmos Genéticos	SIM	SIM	NÃO	NÃO	2D
Sistema Melhor Roteiro Belém	SIM	NÃO	NÃO	NÃO	2D
Uma Abordagem para o Problema da Otimização de Rotas de Veículos baseada em Operadores Genéticos	SIM	SIM	NÃO	NÃO	NÃO
Uma Proposta de Algoritmo Genético de Duas Fases para Roteamento de Veículos	SIM	SIM	NÃO	NÃO	2D
Uso de Algoritmos Genéticos em Sistema de Apoio à Decisão para Alocação de Recursos no Campo e na Cidade	SIM	SIM	NÃO	NÃO	2D
Uso de Algoritmos Genéticos na Construção de Rotas Turísticas	SIM	SIM	NÃO	NÃO	2D
ROTA VIRTUAL	SIM	SIM	SIM	SIM	2D/3D

7.4 Avaliação do Sistema

O sistema foi apresentado a 20 (vinte) usuários, sendo 15 (quinze) pesquisadores (professores e estudantes) da área de computação, além de usuários específicos (cinco) da área de turismo de uma prefeitura.

Primeiramente, foi explicado a esses usuários o objetivo do sistema e, em seguida, o grupo testou coletivamente o sistema. Após a execução do sistema, os usuários responderam a um questionário.

Analisando as respostas nos questionários, foi possível avaliar os itens que seguem abaixo e para cada item foi gerado um gráfico comparativo.

7.4.1 Quanto à finalidade do uso da ferramenta:

No gráfico da Figura 7.1, observa-se que a grande maioria dos usuários respondeu que o sistema é **muito útil**. Algumas pessoas que responderam que o sistema é **útil** justificaram as suas respostas, considerando que o sistema é adequado apenas como recurso adicional na definição de rotas.

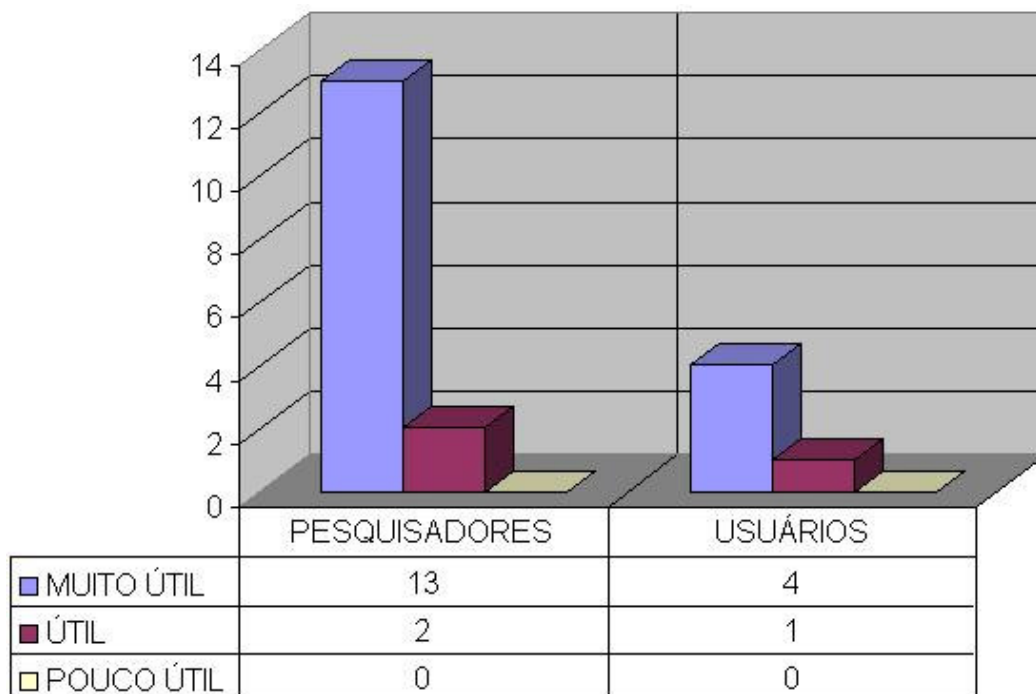


Figura 7.1. Análise quanto à finalidade.

7.4.2 Quanto à interface com o usuário:

Baseados na Figura 7.2, podem verificar que a maior parte das pessoas que avaliou o sistema como **muito intuitivo e intuitivo** e apenas três usuários o consideraram **pouco intuitivo** e não justificaram as suas respostas, a maioria das pessoas elogiou sistema mesmo o considerando **intuitivo**, e outras citaram a necessidade de um prévio conhecimento de Informática para a execução do mesmo.

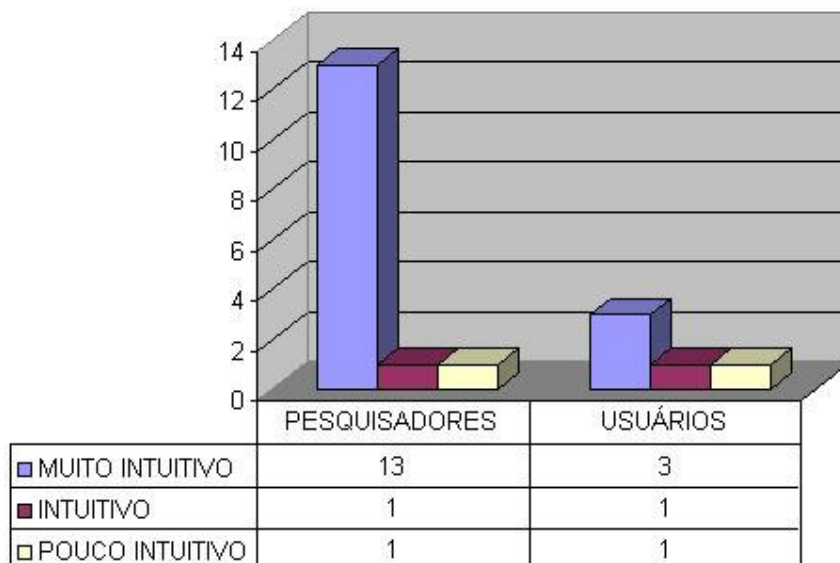


Figura 7.2. Análise quanto a Interface.

7.4.3 Quanto à facilidade de uso:

Observando o gráfico na Figura 7.3, a maioria dos usuários considerou os comandos apresentados de fácil entendimento, os demais justificaram a necessidade de um prévio conhecimento de Informática, considerando que muitos usuários poderão ter dificuldades em executar os comandos, exigindo antes da execução a apresentação de algumas informações adicionais para operacionalização do sistema.

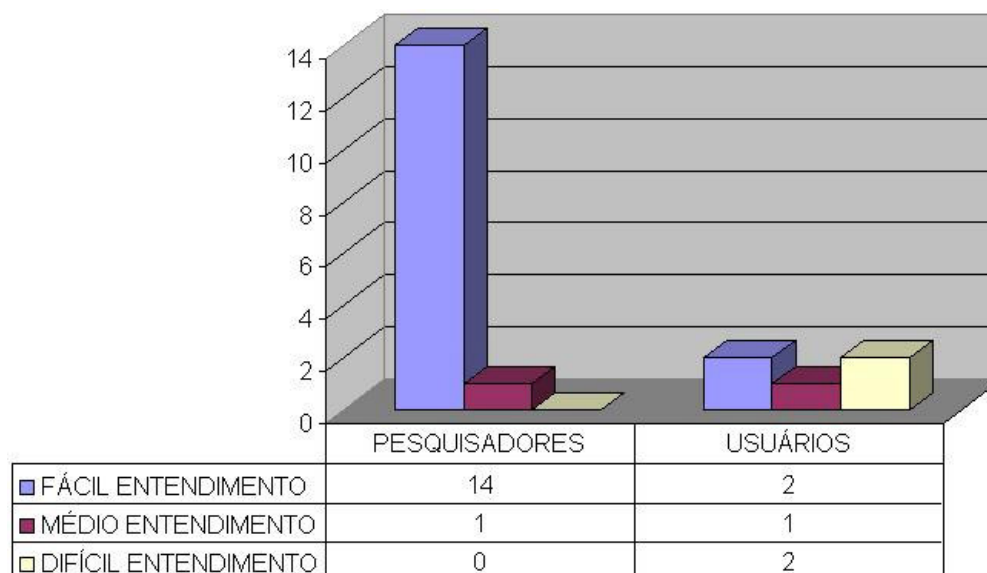


Figura 7.3. Análise quanto à facilidade de uso.

7.4.4 Quanto aos recursos do programa

O programa permitiu a definição clara de um ponto de origem e um ponto de destino, demonstrando a melhor rota?

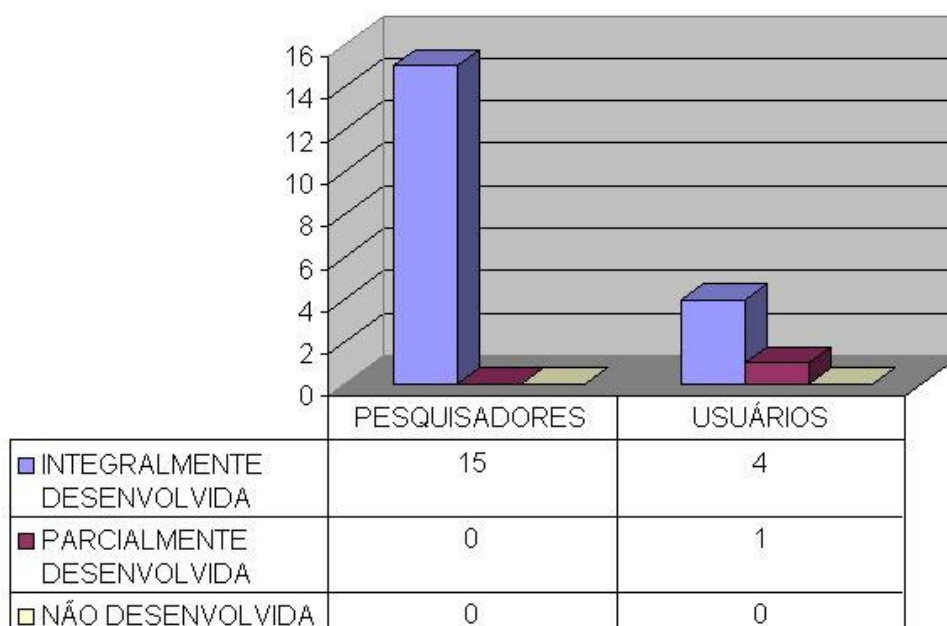


Figura 7.4. Análise quanto aos recursos do programa.

Pelo gráfico da Figura 7.4, pode-se observar que a maioria dos usuários, que respondeu ao questionário, considerou que o programa permitiu a aquisição de

informações úteis a respeito de rotas virtuais.

Em uma outra avaliação do sistema, de acordo com as normas ISO/IEC 9126 (ISO, 1991), foram obtidos os seguintes resultados como mostra a Figura 7.5.

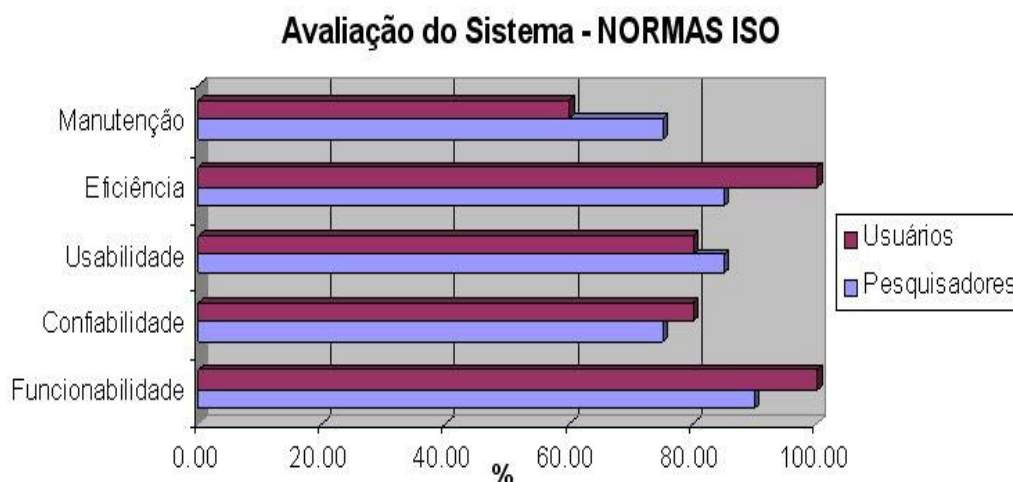


Figura 7.5. Análise dos aspectos de avaliação para *softwares*

Analisando todos os itens avaliados nos questionários, conclui-se que o sistema protótipo desenvolvido foi bem aceito pelos usuários entrevistados. Estes contribuíram com algumas sugestões, descritas a seguir:

- ✓ A interface do 2D que representa o funcionamento do algoritmo genético foi criticada por aqueles que não conhecem o conceito de computação evolutiva, principalmente pelos usuários específicos da área. A sugestão feita refere-se aos parâmetros genéticos que deveriam ser colocados como meios para melhorar a eficácia do protótipo.
- ✓ Para os pesquisadores da área a principal sugestão foi a inserção de uma cidade completa com a existência de vários bairros.
- ✓ Analisando a avaliação feita pelos usuários e as sugestões propostas pelos mesmos, constatou-se que houve motivação por parte deles na utilização do sistema, comprovando a importância da utilização de sistema de rotas virtuais.

7.5 Considerações Finais

Os testes e as avaliações realizadas com o protótipo demonstraram a sua eficácia e o potencial como produto para ser utilizado por prefeituras, hotéis e

empresas que necessitam de definição de rotas e principalmente que essas rotas sejam visualizadas de forma atrativa com mecanismos interativos que conquistem o usuário. Em automóveis já existem essas iniciativas que geram o trajeto para o motorista. Porém a iniciativa de uma rota 3D traria mais confiabilidade ao motorista ao confrontar o local onde está com que lhe é apresentado pelo protótipo.

CAPITULO VIII

8 CONCLUSÕES E TRABALHOS FUTUROS

8.1 Introdução

Este capítulo apresenta os principais tópicos discutidos nesse trabalho, relaciona os possíveis trabalhos futuros advindos dessa pesquisa e avalia a principal contribuição da mesma.

8.2 Conclusões

Durante a pesquisa, constatou-se que existem diversas iniciativas para solucionar problemas relacionados à definição de rotas urbanas e interurbanas. Também foi constatada a existência de modelos de cidades virtuais modeladas e confeccionadas de acordo com as peculiaridades de cada trabalho.

A RV tem demonstrado ser uma poderosa ferramenta de apoio a projetos cartográficos. A análise visual e a sensação de presença no mundo virtual são alguns dos maiores benefícios trazidos por esta tecnologia.

Portanto, A maior contribuição deste trabalho foi criar uma Arquitetura que uniu a área de AG e RV na construção de um AV que represente rotas virtuais seguindo a orientação dada pelo AG.

8.3 Trabalhos Futuros

Como continuação deste trabalho, considera-se importante os seguintes temas:

1 - Quanto ao algoritmo para busca de soluções ótimas:

- ✓ Testar outros modelos de algoritmos.
- ✓ Comparar o modelo atual com outros protótipos que possuem alteração nas características dos operadores genéticos.
- ✓ Realizar testes com um universo maior de possibilidades, por exemplo, usar uma cidade em sua totalidade.
- ✓ Avaliar o protótipo com métricas que calcule a complexidade do algoritmo e o esforço computacional.

2 - Quanto às técnicas de RV:

- ✓ Construir modelos menores em bytes que podem ser portáteis para a WEB.
- ✓ Testar outras metodologias de busca e conversão de mapas para aproveitar a modelagem 2D.
- ✓ Adaptação a outros modelos, como por exemplo CAD.
- ✓ Permitir a interação do usuário com os objetos existentes na cidade, por exemplo, entrar em um Museu, Zoológico, Cinema e outros.
- ✓ Construir modelos geométricos a partir de coordenadas verdadeiras usando GPS podendo estabelecer rotas virtuais de com base no posicionamento real do usuário.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, Marcelo Mazetto Aragão. **Uso de Algoritmos Genéticos como Proposta na Redução de Custos de Energia Elétrica em Armazéns Gerais**. 6th Encontro de Pesquisa e 4th Encontro de Iniciação Científica, Itumbiara-GO, Novembro, 2005.

ANDRADE, Márcio Faria. **Elaboração da Grade Horária de Trabalho para Funcionários de uma Lavanderia usando Algoritmos Genéticos**. 6th Encontro de Pesquisa e 4th Encontro de Iniciação Científica, Itumbiara-GO, Novembro, 2005.

AUKSTAKALNIS, S; BLATNER, D. **Silicon Mirage: The Art and Science of Virtual Reality**. Berkeley: Peatchpit Press, 1992. 235p.

BARCELLOS, João Carlos Holland de. **Algoritmos Genéticos Adaptativos: Um estudo comparativo**. 2000, 143 p. Dissertação (Mestrado em Engenharia) – Escola Politécnica, Universidade de São Paulo; São Paulo, 2000.

BATISTA, Rejaine Marques; RIBEIRO, Marcos Wagner de Souza. **O Uso de Algoritmos Genéticos na Solução de Rotas em Trechos Urbanos**. VII Encontro de Pesquisa, ILES/ULBRA – Instituto Luterano de Ensino Superior/ Universidade Luterana do Brasil, Itumbiara-Goiás, Novembro de 2006.

BESSA, Wender Borges. **Uso de Algoritmos Genéticos na Construção de Rotas Turísticas**. VI Encontro de Pesquisa, ILES-ULBRA – Instituto Luterano de Ensino

REFERÊNCIAS BIBLIOGRÁFICAS

Superior / Universidade Luterana do Brasil Itumbiara-Goiás, Novembro de 2005, 47 p.

BYRNE, C. M. **The Use of Virtual Reality as Educational Tool**. Washington University. Disponível em <<http://www.hitl.washington.edu/publications>>. Acesso em: 10 junho 2005.

CAMACHO, M. de L. A. S. M. **Realidade Virtual e Educação**. Lisboa, Universidade Aberta. Disponível em <<http://phoenix.sce.fct.unl.pt/simposio/30.htm>>. Acesso em: 10 junho 2006.

CARMO, Daniele Leite do. **Aplicação de Algoritmos Genéticos no Desenvolvimento de Rotas Turísticas**. 2004. 67 p. Trabalho de Final de Curso (Bacharel em Sistemas de Informação), Uniminas; Uberlândia, 2004.

CRUZ-NEIRA, C. et al. **The CAVE Audio Visual Experience Automatic Virtual Environment**. Communication of the ACM, Jun. 1992. p.64-72.

DALBONI, Fábio Linhares. **Algoritmos Evolutivos Eficientes para um Problema de Roteamento de Veículo**. 2003, 141 p. Dissertação (Mestrado em Computação Aplicada e Automação) – Curso de Pós-Graduação em Computação Aplicada e Automação, Universidade Federal Fluminense; Rio de Janeiro, 2003.

FOSSE, Juliana Moulin; VEIGA, Luis A.Koenig. **Realidade Virtual como ferramenta na Cartografia 3D**. Universidade Federal do Paraná. 2003 Disponível em:<http://geodesia.ufsc.br/geodesia-online/arquivo/geocoloq_2003/artigos/t031.pdf> Acesso em 25 de Janeiro 2007.

GOLDBERG, David Edward. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Canadá: Addison-Wesley Publishing Company Inc., 1989, 412 p.

ISO/IEC 9126.. **Software Product Evaluation** - Quality Characteristics and Guideline for their Use. International Standards Organization, 1991.

JUNIOR, Osmar de Oliveira Braz. **Otimização de horários em instituições de ensino superior através de algoritmos genéticos**. 2000, 144 p. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em

REFERÊNCIAS BIBLIOGRÁFICAS

Engenharia de Produção, Universidade Federal de Santa Catarina; Florianópolis, 2000.

KELNER, Judith et al. **Modelagem Virtual Urbana: Perspectiva Histórica e Estudo de Caso**. In: V Symposium on Virtual Reality, 2002, Fortaleza, CE. Anais V Symposium on Virtual Reality - Fortaleza: SBC, 2002, pp. 280-290.

KIRNER, Cláudio. PINHO, M. S. **Sistemas de Realidade Virtual**. Grupo de Pesquisa em Realidade Virtual, Departamento de Computação – UFSCar. 1997. Disponível em <<http://www.dc.ufscar.br/~grv/>>. Acesso em 20 de setembro 2006.

LUCAS, Diogo Correa. **Algoritmos Genéticos**: um estudo de seus conceitos fundamentais e aplicação no problema da grade horária. 2000, 66 p. Monografia (Bacharelado em Informática) – Instituto de Física e Matemática. Universidade Federal de Pelotas; Pelotas, 2000.

MACHOVER, Carl Machover. **The promise of virtual reality**. Eurographics '94. Oslo, Norway, September 1994. Tutorial Notes EG'94 TN1.

MOREIRA, Maria Isabel Giusti; AGUIAR, Dr. Marilton Sanchotene. **Um Sistema Classificador de Imagens de Sensoriamento Remoto Baseado em Algoritmos Genéticos**. In: XVIII Congresso Nacional de Matemática Aplicada e Computacional, 2005, São Paulo. Anais; São Paulo: SENAC, 2005. p. 1-6.

NARCISO, Marcelo Gonçalves; LORENA, Luiz A. Nogueira. **Uso de Algoritmos Genéticos em Sistemas de Apoio a Decisão para Alocação de Recursos no Campo e na Cidade**. In: III Congresso Brasileiro da SBI-AGRO - Sociedade Brasileira de Informática aplicada a Agropecuária e Agroindústria, 2002, Foz do Iguaçu. III Congresso da SBI-Agro; Campinas: Embrapa Informática Agropecuária, 2002.

PIMENTEL, João; BATISTA, Nuno; GOES, Luiz; DIONÍSIO, José. **Construção e Gestão da Complexidade de Cenários Urbanos 3D em Ambientes Virtuais Imersivos**. Disponível em <<http://visualis.ist.pt.>> Acesso em 20 de janeiro de 2006.

REFERÊNCIAS BIBLIOGRÁFICAS

PINHO, M. S. **Biblioteca Gráfica OpenGL.** Disponível em: <http://www.inf.pucrs.br/~pinho/CG/Aulas/OpenGL/OpenGL.html> Acesso em: 18 jul. 2006.

_____. M. S. **Manipulação Simultânea de Objetos em Ambientes Virtuais Imersivos.** 2002. 174f. Tese (Doutorado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, UFRGS, Rio Grande do Sul, 2002.

POZO, Aurora; et all. **Computação Evolutiva.** Grupo de Pesquisas em Computação Evolutiva – Departamento de Informática, Universidade Federal do Paraná; Curitiba, [2000?], 61 p. Disponível em: <<http://www.ufpr.br>>. Acesso em: 10.05.2007

REBELLO, Fabrício Rocha; HAMACHER, Silvio. **Uma Proposta de AG de Duas Fases para Roteamento de Veículos.** In: XXXII Simpósio Brasileiro de Pesquisa Operacional, 2000, Viçosa. Anais; Viçosa, 2000. p. 672-688.

RIBEIRO, Glaydston Mattos; LORENA, Luiz A Nogueira. **Roteamento de Veículos Dinâmico Usando Algoritmos Genéticos.** Disponível em: <http://www.lac.inpe.br/~lorena/glaydston/Artigo_G_L_ANPET_2.pdf>. Acesso em 20 de fevereiro de 2007

_____. Glaydston Mattos; LORENA, Luiz A. Nogueira. **Roteamento de Veículos Dinâmico usando Algoritmos Genéticos.** In: XIX ANPET - Congresso de pesquisa e ensino em transportes, 2005, Recife. Panorama Nacional da Pesquisa em Transportes. Rio de Janeiro: ANPET, 2005. v. 19. p. 1593-1603.

RODRIGUES, Marco Antonio Pereira. **Problema do Caixeiro Viajante:** Um algoritmo para a resolução de problemas de grande porta baseado em busca local dirigida. 2000, 104 p. Dissertação (Mestrado em Engenharia) – Universidade Federal de Santa Catarina; Florianópolis, 2000.

SCHUTZ, M. G. F e PIRES, F. M. **Uma abordagem para o problema da otimização de rotas de veículos baseada em operadores genéticos.** *Inv. Op.* [online]. dez. 2003, vol.23, no.2 [citado 27 Agosto 2007], p.197-209. Disponível em: <http://www.scielo.oces.mctes.pt/scielo.php?script=sci_arttext&pid=S0874-51612003000200006&lng=pt&nrm=iso>. ISSN 0874-5161. Acesso em: 10.05.2006

REFERÊNCIAS BIBLIOGRÁFICAS

_____. M. G. F.; PIRES, F. M. **Uma Abordagem para o Problema de Otimização de Rotas de Veículos Baseada em Operadores Genéticos.** Investigação Operacional, Vol.23, Nº 2, pp.197-209 (2003).

SEMENTILLE, A.C. BREGA, J.R.F., KIRNER, C., KUBO, M.M.. **Ambientes Virtuais Distribuídos usando CORBA:** Um Estudo de Caso. In: III Workshop de Realidade Virtual – WRV'2000, 3, 2000, Gramado-RS. Proceedings Gramado-RS, 2000. p.145-156.

SOUZA, Darlon Orlamunder de. **Algoritmos Genéticos aplicados ao planejamento do transporte principal de madeira.** 2004, 184 p. Dissertação (Mestrado em Ciências Florestais) – Setor de Ciências Agrárias, Universidade Federal do Paraná; Curitiba, 2004.

TEIXEIRA, Otávio Noura. **Computação Evolucionária:** dos aspectos filosóficos à implementação dos algoritmos genéticos aplicados na solução do problema do caixeiro viajante simétrico. 2004, 267 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Departamento de Informática, Universidade Federal do Pará; Belém, 2004.

TEIXEIRA, S. **Delphi 6, o Guia do desenvolvedor.** Rio de Janeiro: Campus, 2002.

WOO, M. et al. **OpenGL Programming Guide:** the official guide to learning OpenGL, 3.ed. Reading, Massachusetts: Addison Wesley, 1999. 730 p.

WRIGHT, Richard S. Jr.; SWEET, Michael. **OpenGL SuperBible.** 2nd ed. Indianapolis, Indiana: Waite Group Press, 2000. 696 p.

APÊNDICE

MODELO DO QUESTIONÁRIO DE AVALIAÇÃO DO SISTEMA

APÊNDICE

Avaliação do Sistema que Simula o Processo

Avaliador: _____

Data da Avaliação: ____/____/2007

Nível de Escolaridade:

() Ensino Médio () Ensino Superior () Pós-Graduação

Principais finalidades de utilização do computador: _____

Quanto ao Sistema proposto:

I. Quanto à Finalidade (Eficiência):

() Muito útil () útil () pouco útil

(Justificativa) _____

II. Quanto à Interface (Usabilidade):

() Fácil entendimento dos comandos

() Médio entendimento dos comandos

() Difícil entendimento dos comandos

(Justificativa) _____

III. Quanto à facilidade de uso (Funcionalidade):

() Muito intuitivo

() Intuitivo

() Pouco Intuitivo

(Justificativa) _____

IV. Quanto aos recursos do Programa, a experiência proposta (Manutenção):

() foi integralmente desenvolvida

() foi parcialmente desenvolvida

() não foi desenvolvida por completo

(Justificativa) _____

IV.b. Os objetos disponíveis permitem:

() conceber a experiência proposta

() conceber parte da experiência proposta

() não permitem conceber a experiência

APÊNDICE

(Justificativa) _____

IV.c. A relação entre os objetos disponíveis e suas propriedades:

() é simples de ser definida e permite o desenvolvimento proposto

() não é simples de ser definida, mas permite o desenvolvimento proposto

() não é simples de ser definida e não permite o desenvolvimento proposto

(Justificativa) _____

IV.d. () Sugiro inserir novos objetos no experimento, tais como: _____

IV.e. () Sugiro inserir novas relações entre objetos no experimento, tais como: _____

V. Conseguiu visualizar a rota definida por meio do Ambiente Virtual?

() sim

() não

() em parte

(Justificativa) _____

VII. Observações sobre o programa que achar relevante:

VIII. Sugestões Adicionais:

Assinatura do Avaliador: _____