Universidade Federal de Alagoas
Instituto de Computação

Dissertação de Mestrado

Aplicação do modelo de Algoritmo Genético Baseado em Tipos Abstratos de Dados (GAADT) na adaptação de cenários bidimensionais de MMORPGs

Leonardo Filipe Batista Silva de Carvalho
lfilipebsc@gmail.com

Maceió, Março de 2011

DOCUMENTO RECEBIDO

Em: 05 / 03 / 12

Assinatura do Funcionário

LEONARDO FILIPE BATISTA SILVA DE CARVALHO

Aplicação do modelo de Algoritmo Genético Baseado em Tipos Abstratos de Dados (GAADT) na adaptação de cenários bidimensionais de MMORPGs

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Curso de Mestrado em Modelagem Computacional de Conhecimento do Instituto de Computação da Universidade Federal de Alagoas.

Orientador:

Prof.ª Dr.ª Roberta Vilhena Vieira Lopes

Co-orientador:

Prof. Dr. Fábio Paraguaçu Duarte da Costa

Maceió, Março de 2011

Catalogação na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária Responsável: Maria Auxiliadora G. da Cunha

C331a Carvalho, Leonardo Filipe Batista Silva de.
Aplicação do modelo de algoritmo genético baseado em tipos abstratos de dados (GAADT) na adaptação de cenários bidimensionais de MMORPGs / Leonardo Filipe Batista Silva de Carvalho. – 2011.
120 f. : il.

Orientadora: Roberta Vilhena Vieira Lopes.
Co-Orientador: Fábio Paraguaçu Duarte da Costa.
Dissertação (mestrado em Modelagem Computacional de Conhecimento) – Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2011.

Bibliografia: f. 103-110.
Apêndices: f. 111-120

1. Inteligência artificial. 2. GAADT. 3. Jogos interativos. 4. MMORPG. 5. RPG. I. Título.

CDU: 004.4

UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa Multidisciplinar de Pós-Graduação em Modelagem Computacional de Conhecimento
Avenida Lourival Melo Mota, Km 14, Bloco 09, Cidade Universitária
CEP 57.072-900 – Maceió – AL – Brasil
Telefone: (082) 3214-1364

Membros da Comissão Julgadora da Dissertação de Mestrado de Leonardo Filipe Batista Silva de Carvalho, intitulada: "Aplicação do modelo de Algoritmo Genético baseado em Tipos Abstratos de Dados (GAADT) na adaptação de cenários bidimensionais de MMORPGs", apresentada ao Programa de Pós-Graduação em Modelagem Computacional de Conhecimento da Universidade Federal de Alagoas em 28 de março de 2011, às 14h30min, na sala de aula do Mestrado em Modelagem Computacional de Conhecimento.

COMISSÃO JULGADORA

Roberta Vilhena Vieira Lopes

Profa. Dra. Roberta Vilhena Vieira Lopes
UFAL – Instituto de Computação
Orientador

Fab Paraguaçu Duart da Co

Prof. Dr. Fábio Paraguaçu Duarte da Costa
UFAL – Instituto de Computação
Co-orientador

Prof. Dr. Manoel Agamemnon Lopes
UFAL – Centro de Ciências Agrárias
Examinador

Geber L Ramalho

Prof. Dr. Geber Lisboa Ramalho
UFPE – Departamento de Sistemas de Computação
Examinador

Maceió, março de 2011.

Agradecimentos

Agradeço a Deus, meus pais e ao restante de minha família, pelo incentivo e apoio que sempre me deram e que me fizeram chegar até aqui.

Gostaria de agradecer a meus orientadores a Prof.ª Dr.ª Roberta Vilhena Vieria Lopes e o Prof. Dr. Fábio Paraguaçu Duarte da Costa pelos conhecimentos, conversas e brincadeiras divididas ao longo da orientação, e no caso da professora Roberta, agradeço também por ter me convencido a tentar ingressar no mestrado. Gostaria também de lembrá-la que desta vez não mudei meu tema de última hora, ainda que só por diversão tenha tentando fazê-la pensar que sim, durante o período do último Natal =) (mesma época em que a procurei para mudar meu tema durante a graduação).

A meus amigos de laboratório, em particular a Helio meu companheiro de trabalho, das viagens pelo país e fora dele, e das idas para a UFAL (nas quais ele prefere ficar deitado co-chilando no banco de trás enrolado no cinto de segurança enquanto eu dirijo) e a Douglas, também companheiro das idas para a UFAL (nas quais prefere ficar acordado conversando assuntos importantes, como videogames e mangás), e das horas de lazer, com direito a cinema, pizza e Guitar Hero. Agradeço também a todos os meus outros amigos, que contribuíram para conclusão desta dissertação, de forma direta ou indireta.

Por fim, agradeço a todos aqueles que me ajudaram a ter os conhecimentos que hoje possuo e a ser a pessoa que sou.

Resumo

A importância do uso de recursos de Inteligência Artificial em jogos eletrônicos cresce em resposta a necessidade de jogos que apresentem comportamentos e elementos mais próximos da realidade. Uma necessidade que aliada ao alto nível de interação com o usuário e uma grande variedade de jogos, que oferecem a possibilidade de simular diferentes situações e comportamentos em ambientes complexos controlados, caracteriza os jogos eletrônicos como um importante nicho a ser explorado para a aplicação de diferentes técnicas de inteligência artificial.

Este foi o conceito inicial do qual partiu a idéia de utilizar um jogo de MMORPG como pano de fundo para demonstrar a aplicação de técnicas de inteligência artificial, um gênero de jogo que preza por um ambiente rico, interativo e com eventos significativos ocorrendo de forma simultânea, similarmente ao que ocorre no mundo real.

Tomando proveito do contexto do MMORPG, esta dissertação irá demonstrar a aplicação do algoritmo genético baseado em tipos abstratos de dados (GAADT) para proporcionar a alteração das características de um mapa de jogo em razão da passagem do tempo, de forma similar ao que ocorreria no mundo real. Um conceito ainda pouco explorado em jogos eletrônicos, sobretudo em MMORPGs.

Espera-se ainda que a aplicação criada aqui para a validação do modelo do GAADT a este problema possa ser refinada, de modo a proporcionar aos educadores uma ferramenta que lhes permitam utilizar o cenário de jogo de um MMORPG como um meio para traçar paralelos didáticos entre a ambientação do jogo e o conteúdo de sala de aula.

Palavras-chave: Inteligência Artificial. GAADT. Jogos. RPG. MMORPG.

Abstract

The importance of using Artificial Intelligence in video games is growing in response to the need that videogames have of showing behaviors and other game elements in a closer regard to what is witnessed at the real world. This need combined with a high level of user interaction and a large variety of games offering different situations and behaviors in complex controlled environments, places the videogames as an invaluable exploring niche for the application of different artificial intelligence techniques.

This was the initial concept that led to the idea of using a MMORPG as background for demonstrating the application of artificial intelligence techniques, a game genre which values a rich environment, full of interactivity and with significant events happening simultaneously, in a similar manner to what is witnessed at the real world.

Taking advantage of the MMORPG context, this essay will demonstrate the application of the genetic algorithm based on abstract data type (GAADT) to provide the modification of a game map features due to the course of time, closely resembling what would happen at the real world. A concept that is thus far, little explored in videogames, particularly in MMORPGs.

In addition, it is expected that the application created here for validating the GAADT's algorithmic model for the problem presented here can be refined, thus, providing educators with a tool that allows them to use a MMORPG game scenario as a way to draw didactic parallels between the game environment and the content seen at the classroom.

Keywords: Artificial Intelligence. GAADT. Games. RPG. MMORPG.

Lista de Figuras

2.1 Página de abertura original do jogo *Colossal Cave Adventure* de Will Crowther (1976) utilizando de *interface* textual 25
2.2 Tela do jogo *NetHack* de *The NetHack DevTeam* (1987) utilizando de *interface* baseada em caracteres ASCII 26
2.3 Tela do jogo *dnd* de Gary Whisenhunt e Ray Wood (1975) baseado em D&D e utilizando de *interface* gráfica rudimentar com elementos textuais 26
2.4 Telas do jogo *Final Fantasy* lançado em 1987 pela *Squaresoft* 27
2.5 Tela do jogo *Diablo* lançado pela *Blizzard Entertainment* em 1997 28
2.6 Tela do jogo *Ragnarok Online* lançado pela *Gravity Corp.* em 2002 29
2.7 Tela do jogo *World of Warcraft* lançado pela *Blizzard Entertainment* em 2004 . 30
2.8 Uma árvore binária de triângulos retângulos isósceles de quatro níveis 35

3.1 Ilustração dos conceitos dos mapas adotados 38
3.2 Representação de imagens do cenário em razão quantidade de *tiles* e suas dimensões 40
3.3 Exemplo do operador de Sobel para a estimação do cálculo da gradiente dos eixos x e y para uma máscara de convolução de 3×3 45
3.4 Exemplo do alinhamento de *pixels* em razão de um *pixel* central Λ 46
3.5 Semicírculo utilizado na aproximação da orientação de arestas em razão das quatro direções principais 46

4.1 Exemplos de genes para o problema 63
4.2 Exemplo de um cromossomo com seus genes distríbuidos ao longo do eixo z . . 66
4.3 Outros dois exemplos de cromossomos 66
4.4 Exemplos dos pontos gerados pela função $pontos^{+}$ (máximo de oito adjacências para cada ponto da imagem) 78

5.1 *Interface* gráfica da aplicação após as alterações 93
5.2 Modulo de inserção de descrições exibindo uma das descrição de *tiles* cadastradas 93
5.3 Exemplo dos resultados da detecção de bordas sem a redução da cromaticidade das imagens de entrada 96
5.4 Exemplo dos resultados da detecção de bordas com a redução da cromaticidade das imagens de entrada 96
5.5 Modelo exemplificando o funcionamento do sistema por meio de uma máquina de estados finitos 99
5.6 Comparação de um mapa de entrada e o mapa de saída gerado a partir dele pelo GAADT 110
5.7 Comparação entre a região de saída do GAADT antes e depois do processo evolutivo 110

5.8 Comparação entre as regiões de entrada e saída para diferentes cromossomos do GAADT 111

A.1 Modulo de inserção de descrições exibindo uma das descrição de *tiles* cadastradas para a primeira implementação do trabalho desta dissertação (tal como consta em Carvalho et al. (2010*a*)) 124
A.2 Comparação de um mapa de entrada e o mapa de saída gerado a partir dele para a primeira implementação do GAADT desta dissertação (equivalente ao da implementação de Carvalho et al. (2010*a*)) 126
A.3 Gráfico para a adaptação dos cromossomos para três execuções distintas do GAADT 130
A.4 Comparação de um mapa de entrada e o mapa de saída gerado a partir dele para a primeira implementação do GAADT desta dissertação (tal como consta em Carvalho et al. (2010*c*)) 131

Lista de Tabelas

3.1 Valores compondo o conjunto *Pred* 49
3.2 Valores compondo o conjunto *Tipo* 52

5.1 Inicialização do GAADT 109

A.1 Tabela de dados de cada geração para uma execução do GAADT de Carvalho et al. (2010*a*) 125

Lista de Códigos

5.1 Pseudocódigo para o corte do mapa: criação de genes em razão de conjuntos de bases (células do mapa) 101
5.2 Pseudocódigo para o corte do mapa: seleção de células para o conjunto de bases de um gene 102
5.3 Pseudocódigo para o 1º laço da criação da população inicial de cromossomos . 107
5.4 Pseudocódigo para o 2º laço da criação da população inicial de cromossomos . 107
5.5 Pseudocódigo para o 3º laço da criação da população inicial de cromossomos . 108

Sumário

1 Introdução **12**

1.1 Cenários dinâmicos em RPGs 13
1.2 Descrição do problema 15
1.3 Motivação 16
1.4 Objetivos 18
1.4.1 Objetivos gerais 18
1.4.2 Objetivos específicos 18
1.5 Organização do trabalho 19

2 Uma perspectiva dos procedimentos existentes **21**

2.1 Os jogos de RPG 21
2.1.1 Os RPGs eletrônicos 25
2.2 Algoritmos para manipulação topológica de cenários de jogos eletrônicos . . . 30
2.2.1 Múltipla-resolução 31
2.2.2 Visualização de terreno dinâmico em tempo real 33
2.2.3 Malha de adaptação ideal em tempo real (*Real-Time Optimally Adapting Mesh* - ROAM) 35

3 Abordagens adotadas **37**

3.1 Conceitos do tipo dos mapas adotados 37
3.1.1 Mapa 38
3.1.2 Camadas 38
3.1.3 Tiles 39
3.1.4 Tilesets 40
3.2 Os algoritmos para detecção de bordas 41
3.2.1 O algoritmo de Canny para a detecção de bordas 43
3.3 Definição formal do problema 47

4 Algoritmos genéticos **56**

4.1 O uso dos algoritmos genéticos nos jogos eletrônicos 56
4.2 O algoritmo genético baseado em tipos abstratos de dados (GAADT) 57
4.3 Aplicação do GAADT ao problema 59
4.3.1 Tipos básicos 60
4.3.2 Operadores genéticos 67
4.3.3 O ambiente 83
4.3.4 O algoritmo 84

5 O Sistema **88**
5.1 Levantamento de ferramentas e tecnologias 89
5.1.1 RPG Maker 89
5.1.2 XNA 90
5.1.3 Projeto The Mana World e ferramentas relacionadas 90
5.2 O editor de mapas Tiled: Conceitos herdados e módulos adicionados 91
5.2.1 Criação de um módulo para armazenamento das descrições dos tiles . . 92
5.2.2 Criação de um módulo para geração de uma imagem de bordas 94
5.2.3 Criação de um módulo para a evolução de um mapa 96
5.3 O funcionamento do sistema 97
5.4 O corte do mapa e a formação dos genes 100
5.5 O agrupamento de genes e a criação dos cromossomos da população inicial . . 103
5.6 Resultados 108

6 Conclusão **112**
6.1 Considerações finais 112
6.2 Trabalhos futuros 113

Referências Bibliográficas **114**

A Considerações sobre as implementações iniciais do GAADT para o problema **122**
A.1 A primeira implementação 122
A.2 Implementações posteriores 127

Capítulo 1

Introdução

A Inteligência Artificial, IA, começou a entrar na área de jogos junto ao surgimento dos primeiros jogos eletrônicos, caracterizados pela simplicidade e falta de recursos. É conhecido o fato de que até pouco tempo atrás, a aplicação de recursos de Inteligência Artificial em jogos eletrônicos não acompanhou a mesma evolução vivenciada pelo segmento de computação gráfica, que veio a se tornar o grande destaque do ramo de jogos eletrônicos. Em conseqüência disto, pouco se fez para mudar a imagem preconcebida do uso de recursos de Inteligência Artificial em jogos passada por títulos já há muito ultrapassados, que empregam de forma notória o uso de personagem e outros elementos de jogo de comportamentos visivelmente padronizados (Carvalho, 2008).

Entretanto, o patamar notável que a aplicação dos recursos de computação gráfica atingiu ao curso de sua evolução, foi justamente o que resultou em uma busca por se criar novos diferenciais que pudessem ser agregados a experiência de jogo. Com isto, maior atenção passou a ser dedicada às demais características da criação de jogos eletrônicos, incluindo, a aplicação de recursos de Inteligência Artificial que passou então a constituir mais um fator crítico com relação à qualidade e sucesso de um jogo (Carvalho, 2008).

Encontrar problemas ou situações que possam fazer uso de recursos de Inteligência Artificial não é uma tarefa difícil quando a intenção é aplicar tais recursos dentro do contexto oferecido pelos jogos eletrônicos. A grande variedade de gêneros de jogo, para a qual, cada jogo apresenta ainda uma ampla e distinta série de problemas e possíveis soluções em conjunto com um alto grau de interação com o usuário, justificam o uso de jogos eletrônicos em um contexto acadêmico para o propósito de teste e validação de recursos de Inteligência Artificial. Uma justificativa que se estende ainda às diversas características e comportamentos exibidos por estes diferentes tipos de jogos, que acabam por oferecer uma grande gama de cenários à aplicação de técnicas de Inteligência Artificial na forma ideal de plataformas complexas e controladas (Carvalho, 2008).

Dito isto, o trabalho que será apresentado ao longo desta dissertação irá então se propor a empregar um modelo algorítmico pertencente à linha de Inteligência Artificial para tratar o

problema particular de trabalhar determinadas características de cenários de jogos eletrônicos.

Neste sentido, este capítulo passa agora a uma breve explanação sobre o gênero de jogo que será utilizado nesta dissertação e sua relação com os cenários que definem sua ambientação, estabelecendo assim as bases para a definição informal do problema sendo trabalhado nesta dissertação e as motivações que levaram a tratá-lo. Estes tópicos serão discutidos respectivamente pelas seções 1.1, 1.2 e 1.3, e seguidos imediatamente pelos objetivos que são abordados por esta dissertação e pela descrição da organização deste trabalho.

1.1 Cenários dinâmicos em RPGs

Dentre os diversos gêneros de jogos existentes, um gênero em particular se destaca aqui como o foco desta dissertação, o RPG (*Role Playing Game*), mais especificamente a variedade de RPG conhecida como MMORPG (*Massively Multiplayer Online Role Playing Game*).

O RPG é formado por jogos caracterizados pelo envolvimento predominante colaborativo dos jogadores, no sentido de que, de forma conjunta constroem histórias em que cada um deles assume o papel de um personagem (Adams, 2004). O MMORPG difere-se do RPG, sobretudo, pela quantidade massiva de jogadores que coexistem simultaneamente no ambiente de jogo no qual os jogadores se conectam, daí o significado do termo.

A característica principal do RPG reside no fato de cada jogador assumir o papel de um personagem e que através deste personagem, pode atuar de maneira dinâmica no mundo de jogo. Por esta razão, era de se esperar que as sensações de realismo e imersão proporcionadas pelos cenários de um RPG (e por conseqüência, os de um MMORPG) possuíssem importância fundamental para a experiência de jogo, por terem o papel de reforçar a vivência que um jogador experimenta como parte do ambiente transmitido pelo jogo (Go et al., 2007). Em outras palavras, os cenários possuem a tarefa de reforçar a idéia que os jogadores têm de ser uma parte dinâmica do ambiente que vêem e percebem.

Ainda que possuam um papel tão importante para a experiência de jogo, não é comum que os cenários de MMORPG sejam suscetíveis a mudanças autônomas em sua geografia, o que proporcionaria uma sensação única de dinâmica ao cenário. Esta sensação normalmente é simulada por meio de características pré-definidas, que são acionadas sob determinadas circunstâncias. Um exemplo disto é a prática comum em jogos eletrônicos com cenários tridimensionais (3D) de se inserir uma imagem que simule a cratera criada por uma explosão ao invés de alterar a topologia do cenário e efetivamente inserir uma cratera real no local da explosão, por meio da elevação e do rebaixamento topológico da área afetada (Cai et al., 2006). Grandes alterações na topologia de cenários normalmente são o produto de grandes atualizações, não raro, na forma de expansões[1] ou novas versões do *software* do jogo.

[1] Uma expansão, ou de forma mais correta um pacote de expansão, é uma adição a um jogo de RPG, tabuleiro ou eletrônico já existente. Tais adições têm como propósito adicionar novas áreas, elementos de jogo, ou histórias a um jogo completo e já lançado.

O uso dos artifícios mencionados no parágrafo anterior é ainda reforçado pela pouca variedade de algoritmos que se prestem a tais tarefas, sendo que os algoritmos disponíveis podem ainda ser modelos cujos procedimentos para visualização de superfícies suscetíveis a mudanças arbitrárias de uma topologia e sua conectividade (Shamir et al., 2000) não sejam adequadas a necessidade de atualizações *online*, ou, como o modelo oferecido por He et al. (2002) destinem-se a uma situação em particular. Uma apresentação mais detalhada sobre as particularidades destes dois modelos será apresentada ao longo do capítulo 2, que também irá discutir sobre outros algoritmos de propósitos similares.

Foi procurando de alguma forma suprir a esta carência de algoritmos que se buscou trabalhar novas abordagens, capazes de satisfazer a simulação de um ambiente como o do mundo real que é suscetível a mudanças em razão da simples passagem do tempo ou da ocorrência de eventos distintos de caráter não programado. Uma abordagem que se beneficiaria ainda do caráter crítico dos jogadores, devido à complexa relação que compartilham com os cenários de um MMORPG.

A perspectiva da construção de uma aplicação para a validação dos princípios de um modelo que se preste a solucionar tal problema, oferece ainda a oportunidade de contextualizar o cenário de jogo, concedendo-lhe determinadas propriedades didáticas e, de uma forma aparente ou não, utilizar o MMORPG como uma ferramenta para atrair o interesse de alunos, o que permitiria ensinar conteúdos de disciplinas tais como história ou geografia de maneira que não se perceba que se está aprendendo (Fortuna, 2003).

Este tipo de abordagem torna possível capturar o conteúdo didático de sala de aula e de forma efetiva, transformar a aula em jogo, facilitando o envolvimento do aluno com o tema, e tornando a aula mais agradável, produtiva e divertida (Marcatto, 1996). É necessário apenas tomar o cuidado de que o jogo não passe a ser unicamente um produto didático, o que poderia privá-lo de seu caráter prazeroso e espontâneo (Fortuna, 2003).

A dinâmica do RPG, na posição de um jogo de interpretação de papéis, lhe concede ainda a capacidade de transformar o paradigma de comunicação unidirecional em interativo (Silva, 2001) e assim possibilitar a vivência e experimentação do conteúdo de sala de aula de uma maneira que garanta o interesse do aluno (Bittencourt & Giraffa, 2003).

Incluir neste contexto a aplicação de um algoritmo que permita ao cenário passar por mudanças periódicas de caráter não previsível, reforça a necessidade dos jogadores em empregar conceitos de transferência de conhecimento por meio da socialização e da autonomia que são proporcionadas pela atividade lúdica do RPG, que neste caso, assume a postura de uma prática de ensino-aprendizagem mais libertadora (Martins, 2000) que estende o conceito de imersão para além do mero significado de envolver o jogador no ambiente de jogo por meio dos cenários (Battaiola et al., 2002), agregando a isto, a sensação de engajamento de seus participantes (Jenson & Castel, 2002).

1.2 Descrição do problema

O problema de oferecer certa dinâmica a um cenário por meio da alteração aleatória de suas características geográficas possui um grande nível de abstração, sendo, portanto, necessário definir o que se quer estabelecer por "alteração das características do cenário".

Para isto, é necessário primeiro se ter o conhecimento de que, o problema a ser tratado por esta dissertação, irá trabalhar a partir de um dado mapa de jogo preexistente que será fornecido como entrada para o problema, e que este mapa, é composto por uma malha com um comprimento e uma largura fixa, que possui ainda diferentes níveis de profundidade.

As células da malha do mapa podem abrigar a variados tipos de objetos, sendo que aqui serão pertinentes apenas os que caracterizam imagens que representam os diferentes elementos geográficos que compõem o cenário do mapa (chamados aqui de *tiles*). É interessante afirmar que, apesar da malha do mapa se assemelhar a um cubo, uma vez que suas células estão distribuídas ao longo de um espaço cartesiano de três coordenadas, estes mapas são ditos aqui como bidimensionais (2D) em razão de que, durante o ato de sua exploração pelos personagens do jogo, os mesmos podem deslocar-se pelo mapa em apenas duas dimensões por vez.

Em relação a esta entrada, o que se espera obter aqui como resultado ao final do problema é um conjunto de células pertencentes ao domínio cartesiano do mapa de entrada, cujos elementos geográficos tenham sido substituídos por outros, caracterizando assim a alteração da geografia do mapa.

Finalmente, cabe então estabelecer o que esta dissertação quer definir com o termo "alteração". O uso deste termo define que a intenção que esta dissertação busca para o problema sendo trabalhado não implica apenas em afirmar que um mapa de jogo não será criado do zero, o que pode ser evidenciado pela entrada pretendida para o problema, como também, que os elementos geográficos apresentados pelo conjunto de células do resultado do problema deverão apresentar variações em relação a suas características geográficas que estejam diretamente ligadas a chegada de uma das quatro estações climáticas do ano (primavera, verão, outono ou inverno), e também, às características específicas originalmente exibidas pela região do mapa de entrada compreendida por estas células e representadas por seus elementos geográficos distintos. Uma afirmação que sugere ainda que, não é exigido que todos os mapas de jogo, ou mesmo que nem todas as regiões de um mapa, possuam as quatro estações climáticas bem definidas.

Às características da intenção do problema expressas no parágrafo anterior, é agregado ainda o fato de que, a transição de uma dada estação para outra não se repete, uma afirmação que se aplica ainda à duração destas estações, seus períodos intermediários e mesmo a seus índices pluviométricos, todos os quais, são definidos de forma aleatória para cada mapa que se está trabalhando.

Para alcançar esta intenção é necessário guiar a solução do problema a fim de aproximar-se o tanto quanto possível do resultado desejado. Isto é feito, definindo-se critérios de admissão que validam os resultados do problema como soluções desejáveis. Neste sentido, o primeiro critério

de admissão definido para o problema desta dissertação estipula que as alterações topológicas presentes na geografia dos resultados devem ser consistentes com tipos de elementos geográficos que foram anteriormente definidos como interessantes e/ou desejáveis de se encontrar no resultado final. Estes elementos são definidos aqui como uma seqüência de tipos de interesse, construída a partir do tipo de terreno predominante do mapa de entrada e do tipo de mudança geográfica pretendida para os resultados.

Esta seqüência terá ainda o papel de instituir a predominância de determinados tipos geográficos sobre outros, a qual se reflete ainda, na ordem a ser seguida pelo surgimento das mudanças topológicas do cenário enquanto tenta se aproximar do tipo geográfico de maior prioridade. Por esta razão, os elementos desta seqüência estarão organizados do elemento de maior prioridade, na primeira posição, para o de menor prioridade, na última posição da seqüência.

O último critério de admissão adicionado ao problema afirma que, da configuração presente no cenário do mapa fornecido como entrada para o problema seja preservada. Um critério que conseqüentemente, estabelece um percentual máximo do mapa de jogo que se encontra passível de ser submetido aos processos responsáveis pela alteração de sua geografia.

Todos os conceitos apresentados nesta seção que caracterizam o problema sendo trabalhado nesta dissertação serão revisitados ao longo da seção 3.3 que irá apresentar a definição formal que lhes é atribuída por esta dissertação, bem como demais conceitos que caracterizam a definição do problema e que foram omitidos neste momento.

1.3 Motivação

A motivação deste trabalho partiu então do desejo de oferecer uma solução para o problema de se oferecer uma alteração dinâmica das características geográficas de um cenário de jogo, definido nos moldes dos contextos apresentados na seção anterior, e que pudesse ainda ser submetido a um posterior refinamento que lhe permita abranger aos conceitos educacionais mencionados nos quatro últimos parágrafos da seção 1.1.

Para este propósito, optou-se então por uma abordagem baseada em algoritmos evolucionários, tendo sido feita a opção particular pelo uso de algoritmos genéticos (AG). O uso de algoritmos genéticos como abordagem computacional para solucionar o problema apresentado na seção 1.2 pode ser entendido a partir de sua comparação com o ambiente do mundo real, que evidencia que ambos compartilham de uma característica de aleatoriedade que lhe é naturalmente intrínseca. Uma similaridade que fez com que o modelo oferecido pelos algoritmos genéticos fosse visto como uma alternativa mais interessante de ser empregada para o problema sendo trabalhado aqui.

Dentre a variedade de algoritmos genéticos existentes, escolheu-se trabalhar nesta dissertação com o modelo oferecido pelo algoritmo genético baseado em tipos abstratos de dados (*genetic algorithm based on abstract data type* - GAADT), apresentado pela primeira vez em

Lopes (2003). A escolha por se utilizar o GAADT em lugar do tradicional modelo de algoritmo genético concebido por Holland está ligada a forma com que Holland escolheu expressar a metáfora que percebeu entre a evolução das espécies e os sistemas adaptativos complexos.

A forma com que Holland expressou a metáfora sobre a qual seu algoritmo opera, e mesmo aqueles algoritmos concebidos com o propósito de modificar o algoritmo original de Holland, reflete-se no comprometimento do algoritmo com uma representação pobre dos conceitos de indivíduo e população. Não o bastante, o algoritmo caracteriza-se ainda por apresentar um comportamento concreto para os operadores genéticos, que não permite sua adaptação para problemas de natureza diferente (Lopes, 2003).

De fato, é possível mesmo perceber que os problemas experimentos por usuários que utilizam o algoritmo original de Holland, ou mesmo usuários de um algoritmo oriundo de uma das modificações posteriormente propostas a ele, não estão ligados a metáfora utilizada por Holland e sim à forma com que esta metáfora está sendo expressa, quer utilize de indivíduos formados por cadeias de caracteres binários ou alfanuméricos com tamanhos fixos ou não (Lopes, 2003).

Como poderá ser visto ao longo do capítulo 4 o GAADT se difere destes modelos de algoritmo genético e o tornam um modelo mais interessante para trabalhar o problema sendo abordado por esta dissertação uma vez que se mostra como uma alternativa para as limitações herdades destes algoritmos, utilizando assim os conceitos de tipos abstratos de dados para construir de um modelo de algoritmo genético que expressa de maneira abstrata a metáfora de Holland e lhe permite uma instanciação para diferentes tipos de problema (Lopes, 2003).

A adoção do GAADT como modelo de solução para o problema, ainda beneficia o projeto desta dissertação pela perspectiva de desenvolvimento do projeto. Neste sentido, ao se levar em conta que o que é visado alcançar aqui constitui o que seria, ao menos, um modulo do projeto de um MMORPG completo, a implementação do GAADT por meio dos conceitos de tipo abstrato de dados permite aos elementos deste algoritmo um tratamento orientado a objetos[2], que facilita a vinculação e a troca de informação do modelo com diferentes módulos do projeto.

Em razão disto, é possível afirmar ainda que, em adição ao foi passado por meio do primeiro parágrafo desta seção, a motivação para este trabalho parte também de um interesse do autor em se aprofundar em assuntos referentes à criação de jogos eletrônicos, bem como de tomar proveito da oportunidade de utilizá-los para o teste e validação de novas metodologias e algoritmos de Inteligência Artificial. Uma perspectiva que é vista como interessante uma vez que os jogos eletrônicos em geral constituem plataformas complexas e controladas de alto nível de interação com o usuário (principalmente em jogos da variedade MMORPG), constituindo o que seria uma situação ideal para testar e validar a aplicação de tais algoritmos.

A perspectiva do objetivo apresentado por este trabalho despertou ainda o interesse em, uma vez concluída a aplicação desenvolvida com o propósito de validar a solução do problema apresentado na seção anterior, a mesma pudesse ser utilizada dentro de um contexto educacional, no

[2]Paradigma de programação que utiliza estruturas de dados contendo dados, métodos a as formas de interações entre os mesmos, com forte predominância do uso de tipos abstratos de dados.

sentido de gerar cenários de jogo que não apenas se adéqüem ao enredo proposto para o jogo, mas que possuam equivalentes no mundo real com os quais os professores possam traçar paralelos, e assim, apresentar aos alunos o conteúdo de sala de aula de uma maneira diferenciada.

1.4 Objetivos

1.4.1 Objetivos gerais

A proposta inicial deste trabalho consistiu em utilizar o ambiente e as características de jogo de um RPG para trabalhar uma abordagem de algoritmos evolucionários que se preste a manipular as características topológicas de um cenário deste tipo de jogo.

Neste âmbito, o que se busca aqui é utilizar o modelo do GAADT para alterar as características geográficas de um mapa de jogo dado como entrada de maneira a gerar um mapa de saída que possa atender de maneira satisfatória a sociedade de jogadores conectados a ele em razão da coesão apresentada entre os elementos de cenário adicionados e o contexto que foi definido para o mapa de saída, bem como, a forma como se dá a disposição de tais elementos no cenário de jogo. Tal como foi apresentado pela descrição do problema na seção 1.2.

O contexto apresentado no parágrafo anterior, permite ainda sua interpretação a partir da perspectiva educacional inicialmente discutida ao final da seção 1.1, concedendo-lhe o benefício adicional de estimular o compartilhamento de experiências e conhecimentos entre os indivíduos da sociedade de jogadores por meio de suas interações. Este objetivo, bem como o primeiro objetivo apresentado nesta seção, foram refinados até se chegar ao formato que será agora apresentado na seção 1.4.2.

1.4.2 Objetivos específicos

A partir do problema computacional apresentado na seção 1.2 e do escopo geral dos objetivos pretendidos por este trabalho, tal como transmitido pela seção 1.4.1, os objetivos deste trabalho no que concerne um ponto de vista computacional podem ser definidos como:

- Realizar um mapeamento do cenário representado por um mapa de jogo definido como entrada para o sistema nos tipos básicos utilizados pelo GAADT na construção de seu processo evolutivo;
- Identificar quais das regiões apresentadas no mapa de jogo dado como entrada são passíveis de serem comparadas entre si;
- Identificar quais das características do mapa de entrada são relevantes na simulação da mudança de estação pretendida;

- Definir uma função heurística que determine o quão boa é uma dada característica presente no mapa sendo trabalhado, em razão da mudança de estação pretendida;
- Estabelecer o peso com o qual contribuem as regiões de alterações climáticas que se encontram nas adjacências da região sendo trabalhada;
- Definir uma função de validação para as alterações climáticas realizadas em uma região do mapa trabalhado com base em suas propriedades de coesão e na suavidade de suas transições;
- Estabelecer um limite que determine o percentual máximo do quanto o mapa de entrada do sistema pode diferir do mapa gerado como saída, que por conseqüência, implica automaticamente na definição de um percentual mínimo que estabeleça o quanto do mapa de entrada do sistema deve ser equivalente ao mapa gerado como saída.

Uma vez que a aplicação que valide a solução do problema que está sendo apresentado por esta dissertação tenha sido concluída, o que se espera aqui é que seja possível tomar proveito do cenário dinâmico e das ambientações de jogo que se esperam conseguir ao final deste trabalho, de maneira que professores e desenvolvedores possam de forma conjunta refinar os resultados deste sistema de uma forma que permita utilizar jogos de MMORPG (que empreguem do tipo de cenário do jogo adotado por esta dissertação) como ferramentas de sala de aula que possibilitem a alunos traçar paralelos entre as experiências vivenciadas no ambiente de jogo e o conteúdo didático que é visto em sala de aula.

A perspectiva de alcançar tal objetivo torna-se interessante uma vez que se perceba que sua contemplação não apenas atenderia a sociedade de jogadores com cenários de jogos que além de mutáveis apresentam coesão entre seus elementos, como também, permitiria que o compartilhamento de conhecimento e experiências advindos de suas interações se expandisse para fora do ambiente de jogo, agregando aos jogadores, neste caso alunos, conhecimentos que seriam de relevância para as vidas que levam uma vez que o momento de jogo termina.

1.5 Organização do trabalho

Uma vez que foi definida a proposta que será tratada ao longo deste trabalho, esta dissertação irá agora prosseguir com a apresentação de uma perspectiva sobre as abordagens já existentes e que possuam uma proposta similar a do problema sendo abordado aqui. A exposição destas abordagens ocorrerá no capítulo 2, que também abordará os conceitos dos jogos de RPG e os diferentes formatos em que podem se apresentar, buscando assim definir o contexto da modalidade de jogo sendo adotado por esta dissertação e das abordagens comumente aplicadas ao tipo de problema trabalhado aqui. Em seguida o capítulo 3 irá apresentar as metodologias que

foram empregadas neste trabalho, o que inclui os conceitos pertinentes aos mapas de jogo utilizados, algoritmos aplicados (a exceção do GAADT) e a própria definição formal do problema abordado.

O capítulo 4 fará uma breve exposição dos tipos de situação em que os algoritmos genéticos são utilizados no contexto dos jogos eletrônicos, seguindo então à apresentação dos conceitos do algoritmo genético que será aplicado aqui, o GAADT, e a como estes conceitos foram transpostos para atender o problema proposto. O capítulo 5 irá então apresentar o sistema como um todo, incluindo o processo de escolha do ambiente de desenvolvimento utilizado, seus conceitos, as modificações que lhe foram impostas e por fim os resultados que foram obtidos a partir de sua execução. O capítulo 6 apresentará as conclusões tiradas deste trabalho e as propostas que podem ser feitas a partir dos resultados obtidos. É possível ainda fazer um comparativo entre os resultados obtidos pelo sistema apresentado aqui com aqueles das suas implementações anteriores, os quais, estão disponíveis no apêndice A.

Capítulo 2

Uma perspectiva dos procedimentos existentes

Este capítulo irá tratar de realizar uma explanação mais aprofundada sobre os jogos pertencentes ao gênero RPG, no qual se inclui o MMORPG, de forma a explorar em maiores detalhes a importância que o sucesso da aplicação de um modelo de algoritmo, que se preste a alterar as características geofísicas de um mapa de jogo, exerceria sobre a comunidade de jogadores que se conectam ao ambiente de jogo.

A segunda parte deste capítulo, irá então abordar algumas das técnicas existentes que se prestam a tratar de ao menos um dos aspectos da manipulação da topologia de um cenário de jogo, no caso particular desta dissertação, um cenário de jogo de RPG.

É importante destacar que, não foram encontrados estudos que se prestassem a trabalhar a reformulação da topologia de um cenário de jogo correspondente ao tipo utilizado aqui – o qual está mais próximo dos cenários de jogos que antecedem os jogos tridimensionais, os quais fazem uso de aglomerados de polígonos para formar o ambiente de jogo e personagens – que possui um formato bidimensional para a exploração do cenário, uma perspectiva de visualização localizada acima do cenário de jogo em um dado ângulo de inclinação, e que utiliza de conjuntos de imagens para construir os cenários e representar as demais entidades que habitam o mundo de jogo.

Além disso, enquanto a maioria das técnicas existentes que se prestam a manipular a topologia de um cenário de jogo atua a partir de um ponto de vista intimamente ligado ao ramo da computação gráfica, o modelo abordado neste trabalho atua sob a perspectiva computacional da inteligência artificial, diferenciando-se dos demais modelos que serão apresentados aqui.

2.1 Os jogos de RPG

O RPG, sigla para *Role-Playing Game* e que pode ser traduzido como jogo de interpretação de papeis é descrito por Zuchi (2000) como um jogo em que cada participante assume o papel

de um personagem dentro de uma história imaginária, de forma que os cenários e personagens enfrentados no decorrer da história são definidos e interpretados pelo jogador no papel de mestre do jogo, ao qual, também cabe o papel de narrador da história que, em muitos casos, agrega ainda os papeis de autor e roteirista (Pereira et al., 1992). Neste contexto o mestre de jogo torna-se, portanto, o responsável por descrever as situações vivenciadas pelos demais jogadores, além de tomar o papel de uma espécie de árbitro que utiliza das regras específicas do jogo para tomar suas próprias decisões e autorizar aquelas declaradas pelos demais jogadores, lançando de dados para conseguir um resultado aleatório para a execução ou o efeito das mesmas (Bittencourt & Giraffa, 2003).

Este modelo de jogo corresponde basicamente ao modelo do jogo *Dungeons & Dragons*, normalmente referenciado apenas como D&D, criado em 1974 por Gary Gygax e Dave Arneson e considerado como o jogo responsável por dar início aos RPGs modernos (Crigger, 2008). O título emergiu a partir dos chamados "jogos de guerra" (*war games*) comuns nos anos 50 e cuja origem pode ainda ser traçada a um período anterior. Nestes jogos, os jogadores lidavam com operações militares de tipos variados, sendo comum que assumissem o controle de unidades militares inteiras (Barton, 2007).

Ainda que o título tenha tomado emprestados diversos elementos da mecânica destes jogos (com ênfase na realização de cálculos), foi ele o responsável pela inclusão de algumas inovações que divergiam radicalmente da temática original. Um exemplo disto foi o uso de um mundo fantástico para o desenvolvimento da história, inspirado nos livros da trilogia O Senhor dos Anéis de J.R.R. Tolkien, ao invés de uma recriação precisa de batalhas históricas (Barton, 2007).

Neste sentido, cabe ainda afirmar que o enredo do título auxiliou sua ascensão, uma vez que foi possível tomar proveito do interesse existente na época pelas obras de J. R. R. Tolkien, atraindo fãs das obras do autor em razão de sua ambientação, rica em elfos, dragões e itens mágicos (Crigger, 2008). Uma atração que pode ser compreendida ao se aceitar a idéia de que ler sobre os personagens da obra e ter a oportunidade de representar personagens dentro de um contexto semelhante são dois tipos de experiências diferentes (Barton, 2007).

Outra inovação foi a forte aposta do título na introdução do conceito da representação de papeis, um dos principais responsáveis por diferenciar o título, e o gênero que viria a iniciar, de seus predecessores por encorajar a temática de que o jogador controle um único personagem ao contrário de um batalhão, desenvolvendo para o mesmo uma história própria e utilizando de estatísticas para descrever as qualidades e fraquezas de seu personagem (Crigger, 2008).

O título também introduziu o uso do sistema de pontos de experiência como mediador para elevar o nível dos personagens dos jogadores e seus atributos, além de outras mecânicas que hoje são comuns a quase totalidade dos RPGs (Crigger, 2008).

A partir destas características é possível perceber que o jogo criado por Gygax e Arneson não se limitou simplesmente a criar ou preencher folhas de personagens e alterar os atributos associados aos mesmos, mas que, acima de tudo, priorizava a criação e narração de histórias, a criatividade e a imaginação, elementos que hoje constituem o maior legado do título e que

fundamentam o conceito da representação de papeis, que dá nome ao gênero, dentro da temática do jogo e que permite a qualquer pessoa, tornar-se um mago, bárbaro, paladino, ou qualquer outro herói que deseje, pelo simples uso de mapas desenhados a mão e o rolar de dados.

Este contexto é até hoje o espírito que não apenas os títulos de D&D, mas qualquer outro título atual de RPG busca capturar. Cabe ainda afirmar que os conceitos definidos por D&D possuem ainda presença tão marcante nos demais RPGs existentes que é provável que tais títulos tenham sempre a sombra do primeiro pairando sobre si. Tal afirmação é reforçada pela afirmação pública feita em 2008 por Richard Garriott, criador da série *Ultima*, após a morte de Gygax naquele mesmo ano dizendo que "*Milhões e Milhões de jogadores ao redor do mundo vivem e jogam em mundos imaginários construídos com base no que foi inicialmente concebido por Gary*" (Crigger, 2008).

E a partir destes conceitos que se pode definir o RPG como um jogo de representação de papeis que permite a interação de personagens dentro de uma trama, desenvolvida dentro do cenário de jogo e conduzida por um mestre de jogo, o qual é o responsável por descrever em detalhes a ambientação e o sistema de regras adotado. Dentro desta mesma perspectiva é possível ainda identificar os personagens como sendo construídos a partir de valores próprios para determinadas habilidades manuais e físicas, bem como, suas deficiências e perfis psicológicos (Bolzan, 2003).

Estes princípios permitem contextualizar o RPG em razão de três componentes principais que correspondem à ambientação, à história e ao sistema de regras do jogo (Bittencourt & Giraffa, 2003). De acordo com esta metodologia, a ambientação corresponde ao contexto em que a história se desenvolve, sendo alguns dos temas mais recorrentes as ambientações medieval, futurista e a ficção científica. A história corresponde ao roteiro do jogo, o qual é normalmente composto de forma não linear, ficando aberta ao jogador a seleção das ações que irá desempenhar. O terceiro elemento, o sistema de regras, é, tal como o nome indica, o conjunto de regras que deve ser utilizado na resolução das ações descritas pelo jogador.

Dentro desta mecânica de jogo, é possível contextualizar os elementos responsáveis pela construção de um RPG e por viabilizar a comunicação entre os jogadores. Segue então de acordo com o trabalho de Andrade et al. (2004) que:

- **Jogador**: assume o papel de um personagem habitando o ambiente de jogo, sendo o responsável por controlar as ações do mesmo;
- **Mestre de jogo**: controla os personagens que interagem com o jogador ao longo do desenvolvimento da trama. Além disso, o próprio desenvolvimento da trama é sua responsabilidade;
- **Sistema de regras**: conjunto de regras que impõe as restrições responsáveis por limitar os tipos de capacidade de um personagem, definindo os níveis destas capacidades e seus critérios de evolução, i.e. o momento em que avançam ao nível seguinte;

- **Cenário**: o cenário corresponde ao mundo de jogo, sendo composto por um conjunto de mapas em que o jogador pode ser inserido;

- **Personagem**: corresponde ao *avatar* utilizado pelo jogador para explorar e interagir com o mundo de jogo e com quaisquer de seus habitantes. Os jogadores podem construir seus próprios personagens antes do início da partida, ou mesmo, utilizar de personagens predefinidos que sejam oferecidos pelo jogo;

- **Trama**: o enredo que compõe a história do jogo, incluindo também as decisões e ações realizadas durante o desenvolvimento da história;

- **NPC**: acrônimo para *Non-Player Character*. São os personagens não controlados pelo jogador, os quais são manipulados pelo mestre de jogo ou pelo sistema de regras.

Cabe também afirmar que o gênero, de um ponto de vista educacional, explora as múltiplas inteligências de um indivíduo de acordo com a teoria de Gardner (1994). O ato da criação e narração de histórias, e ainda, todo o processo de comunicação verbal relaciona-se com a inteligência lingüística. O gênero compete também à inteligência musical, uma vez que pode utilizar, ou mesmo requerer a criação, de músicas para auxiliar no clima da trama do jogo. A inteligência lógica/matemática é utilizada no momento da criação das fichas de personagens em razão da necessidade de se trabalhar com números e operações aritméticas, abrangendo também os raciocínios referentes à aplicação das regras, as operações matemáticas pertinentes após o lance dos dados e todo o processo de planificação e avaliação efetuados pelos jogadores e mestre de jogo (Bittencourt & Giraffa, 2003).

Em diversos momentos do jogo também é feito o uso de mapas, os quais podem mesmo ser criados pelos jogadores para facilitar a exploração dos labirintos descritos no jogo, além é claro das ilustrações de monstros, personagens e itens, todos referentes ao uso da inteligência espacial. Todo o processo de cooperação entre os jogadores durante uma partida inclui-se na inteligência interpessoal, lidando também com questões intrapessoais uma vez que o jogo pode simular técnicas de psicodrama. Mesmo a habilidade cinestésica pouco estimulada pelo gênero pode ser compensada ao se adotar a modalidade em *Live Action*, na qual os jogadores interpretam seus papeis. Contudo, esta modalidade deprecia o uso das inteligências lógica/matemática, em espacial, por não utilizar de dados, desenhos ou outros esquemas, além de contar com um sistema de regras bastante simplificado (Bittencourt & Giraffa, 2003).

Toda a temática apresentada até agora se refere aos tradicionais RPGs de "mesa" jogados com papel, caneta, dados e mesmo um tabuleiro, tal como a concepção inicial originada em D&D. Ao se transpor estes conceitos para o computador, criando o que alguns denominam ainda de CRPG (*Computer Role-Playing Game*) a metodologia sofre algumas alterações que lhe permitem ser representada neste novo ambiente e a tomar proveito dos recursos oferecidos por ele (Tobaldini & Brancher, 2006). Esta temática será tratada a seguir.

2.1.1 Os RPGs eletrônicos

No que diz respeito aos RPGs como jogos digitais, sua popularização começou juntamente com a expansão dos computadores pessoais, tomando proveito do controle complexo oferecido pelo teclado do computador. No contexto de jogos digitais, cabe ainda ao RPG uma quebra de paradigma, por popularizar um gênero de jogo que exige grande investimento de tempo e esforço para a conclusão (Schwab, 2004).

Os primeiros títulos eram comumente baseados em texto e viriam posteriormente a inspirar a criação das *interfaces* MUD (*Multi-User Dungeon*), também comumente empregadas neste período. Podiam também utilizar de uma arte conceitual baseada em caracteres ASCII (Schwab, 2004)(Bolzan, 2003). Exemplos de jogos utilizando estes tipos de *interface* são ilustrados nas Figuras 2.1, 2.2, e 2.3, sendo a última uma imagem do jogo *dnd* de 1975, uma adaptação do D&D tradicional cuja *interface* mescla gráficos rudimentares com elementos de outras das técnicas mencionadas.

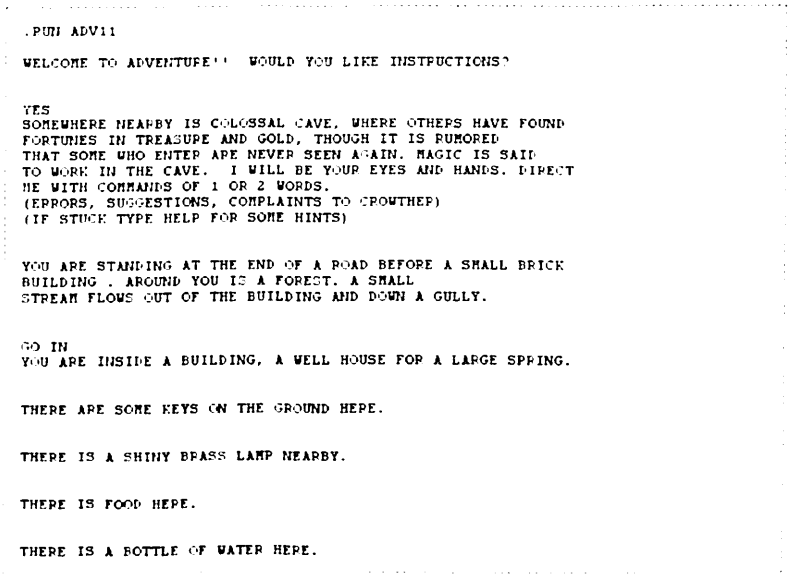


Figura 2.1: *Página de abertura original do jogo Colossal Cave Adventure de Will Crowther (1976) utilizando de interface textual*

Tal como suas contrapartes jogadas em mesa e usando de papel e caneta, estes primeiros jogos, quase que inteiramente verbais, se apoiavam no conceito de que os jogadores deveriam tomar o lugar de um personagem e explorar os mundos descritos em tela, tendo o computador no papel do mestre de jogo que era programado para responder com palavras as ações do jogador (Bolzan, 2003).

Ainda que este sistema possa parecer rudimentar, ele fornecia complexidade aos jogos em razão da infinidade de combinações possíveis de serem criadas por meio das opções e ações oferecidas pelo jogo, que tinham seus bancos de dados constantemente ampliados, resultando assim em uma grande liberdade de escolha para o jogador (Tobaldini & Brancher, 2006).

Foi a partir destes jogos que os RPG eletrônicos evoluíram até os atuais RPGs gráficos,

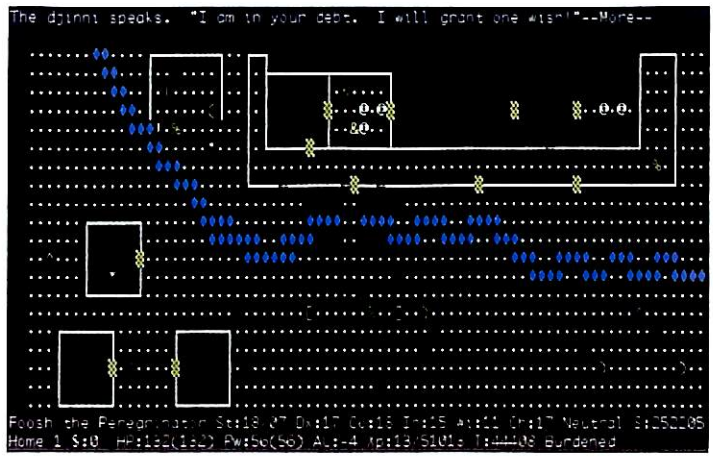


Figura 2.2: *Tela do jogo NetHack de The NetHack DevTeam (1987) utilizando de interface baseada em caracteres ASCII*

mantendo sempre o conceito original da representação de papeis, buscando alcançar um nível de imersão suficiente para que os jogadores invistam nos títulos lançados as muitas horas de jogo, normalmente, necessárias para concluí-los (Schwab, 2004)(Tobaldini & Brancher, 2006).

Contudo, deve ser entendido que esta evolução teve o efeito colateral de afastar os jogos computadorizados que se auto-intitulam RPGs dos conceitos tradicionais deste gênero. De fato, grande número dos títulos disponíveis atualmente são, na realidade, jogos de aventura com elementos característicos de jogos RPG (Tobaldini & Brancher, 2006).

Figura 2.3: *Tela do jogo dnd de Gary Whisenhunt e Ray Wood (1975) baseado em D&D e utilizando de interface gráfica rudimentar com elementos textuais*

A transposição dos RPGs para o meio digital, em razão do uso do computador ou mesmo de consoles (videogames), gerou inúmeras modalidades para o gênero, cada uma possuindo

características próprias. A seguir serão detalhadas as modalidades de RPG considerados interessantes para o contexto deste trabalho, utilizando para isto do que é passado pelo trabalho de Bittencourt & Giraffa (2003).

2.1.1.1 Os jogos clássicos

A primeira geração de RPGs computadorizados teve início logo no lançamento de D&D nos anos 70. Os primeiros modelos tratavam ainda, de lidar com a freqüente necessidade de se verificar tabelas em razão dos resultados de dados e do grande número de jogadores necessários para compor o jogo, permitindo que o mesmo fosse apreciado de modo individual (Crawford, 1984). Um dos primeiros jogos de RPG desenvolvido e distribuído por uma empresa seguindo o estilo D&D foi *Temple of Apshai* em 1979 da *Automated Simulations*, que teve duas seqüências lançadas posteriormente (Hunter, 2006).

Dentre os outros títulos lançados nesta época encontra-se a série *Ultima* iniciada em 1980 (e que permanece, até o atual momento, em atividade), com treze títulos lançados, incluindo o que posteriormente, viria a ser considerado como o primeiro jogo virtual online com mundo persistente (Bittencourt & Giraffa, 2003). Algum tempo depois, seriam lançados em 1987 os jogos *Phantasy Star* e *Final Fantasy* (Figura 2.4), ambos da *Squaresoft*, sendo o último uma das mais cultuadas séries de jogos já criadas.

(a) *Exemplo de cenário sendo explorado* **(b)** *Tela de combate por turnos*

Figura 2.4: *Telas do jogo Final Fantasy lançado em 1987 pela Squaresoft*

As características dos jogos compondo o que se poderia chamar de primeira geração de RPGs computadorizados incluem a mecânica de um único jogador; uma exploração limitada do mundo de jogo; uma trama semelhante à de jogos de aventura que foca a coleta de objetos que serão posteriormente necessários ao progresso na trama; a mecânica de fichas de personagens e sua evolução em razão do acumulo de pontos de experiência, tal como adotada nos RPGs de mesa; e uma impossibilidade, para a grande maioria dos títulos, de se personalizar elementos do jogo ou mesmo criar um personagem principal.

Além disto, estes jogos possuíam ainda uma simplicidade característica, devido às restrições tecnológicas dos anos 80 e início dos 90, o que conseqüentemente os tornavam jogos limitados quando comparados aos RPGs de mesa (Bittencourt & Giraffa, 2003).

2.1.1.2 Os RPGs de múltiplos jogadores

Em 1997 a *Blizzard* lançava o jogo *Diablo* (Figura 2.5) e com isso mudou o conceito do gênero RPG ao incluir a modalidade multiplayer, que em outras palavras, significa dizer que tornou possível a coexistência de vários jogadores em uma mesma partida, quer fosse através de uma rede local ou da Internet. As críticas aplicadas ao jogo o acusam de descaracterizar o RPG, por gerar muitas batalhas e não permitir a interpretação de papeis, tornando-o mais semelhante a um jogo de ação.

Figura 2.5: *Tela do jogo Diablo lançado pela Blizzard Entertainment em 1997*

Tais críticas não foram aplicadas a *Baldur's Gate* lançado em 1998 pela *Bioware* possibilitando o jogo *multiplayer* em rede e apresentando um roteiro composto por várias sub-tramas, conceitos que lhe permitiram evitar que o jogador deixasse de agir pela falta de itens que viabilizassem seu progresso na trama principal. Uma temática que permitia que grupos de jogadores construíssem seus próprios personagens e que incluíssem o elemento de interpretação.

Outra inovação foi trazida em 2002 por *Neverwinter Nights*, também da *Bioware*, que permitia que jogadores criassem suas próprias histórias, armas e objetos de jogo.

Segundo Bittencourt & Giraffa (2003) as características dos RPGs desta geração incluem a possibilidade de múltiplos jogadores; ampla exploração de um mundo virtual com inúmeras localidades; histórias com várias sub-tramas e, em alguns casos, a possibilidade de se criar as próprias aventuras; maior fidelidade aos conceitos do RPG, incluindo a criação de personagens, evolução por acumulo de pontos de experiência, uso de sistemas de regras consagrados que permitem a cooperação e a interpretação; e a possibilidade de personalizar o próprio personagem.

2.1.1.3 Os RPGs com mundos virtuais persistentes

Também em 1997 é lançado o jogo *Ultima Online* pela *Eletronic Arts* e desenvolvido pela *Origins*. O jogo exige que o jogador crie um cidadão virtual que irá "viver" em um mundo ambientado em uma fantasia medieval, no qual terá uma profissão e poderá tornar-se um guerreiro ou um mago através de treinamento em localidades específicas, ou, através da experiência obtida por batalhas contra criaturas fantásticas.

Independente de um determinado jogador estar ou não conectado, existem pessoas interagindo e mudando o mundo, sendo precisamente a isto que se refere à terminologia de mundo virtual persistente. Também são comumente chamados de MMOG (*Massively Multiplayer Online Game*), ou como anteriormente mencionado neste trabalho, MMORPG.

O MMORPG também possui a característica dos RPGs *online* de permitir que os jogadores criem objetos, mas sua principal característica é a constante intervenção de uma equipe de mestres de jogo que criam tramas e desafios para os personagens. Além disto, as tramas possuem estrutura não-linear, sem começo, meio, ou fim, transformando o mundo virtual em uma história em aberto a ser explorada.

Outros títulos que pertencem a esta modalidade incluem o jogo *Ragnarok Online* da *Gravity Corp.* (Figura 2.6) e os jogos *StarCraft* e *World of Warcraft* (Figura 2.7), o último sendo o MMORPG com o maior número de assinantes no mundo, contando com 12 milhões no período de outubro de 2010 (Blizzard Entertainment, 2010).

Figura 2.6: *Tela do jogo Ragnarok Online lançado pela Gravity Corp. em 2002*

As características que definem estes jogos são a imensa quantidade de jogadores conectados e interagindo em um mesmo mundo virtual; a exploração extremamente ampla do mundo; o uso de tramas não lineares; a grande possibilidade de interpretação e cooperação que os títulos proporcionam; a possibilidade de customização do personagem e a criação de objetos que farão parte do mundo virtual e se tornarão acessíveis a outras pessoas.

Figura 2.7: *Tela do jogo World of Warcraft lançado pela Blizzard Entertainment em 2004*

Ainda segundo os relatórios da IGDA (*International Game Developers Association*) liberado em 2003, os MMOG como um todo são reconhecidos como necessitando de intensa interação social por parte dos jogadores, dos quais, muitos tentam ainda alcançar posições de *status* dentro da comunidade de jogadores. Estes jogos possibilitam ainda a socialização através do comércio e o aperfeiçoamento de habilidades por meio de treino e de atividades cooperativas. Destaca-se também a possibilidade de que os jogadores passem um grande período na comunidade em razão das amizades virtuais que são criadas (Bittencourt & Giraffa, 2003).

2.2 Algoritmos para manipulação topológica de cenários de jogos eletrônicos

Durante a realização do levantamento bibliográfico desta dissertação, percebeu-se que algoritmos voltados à tarefa focada por este trabalho, de realizar alterações geográficas na topologia de um cenário representando um mapa de jogo, costumam pertencer à área da computação que diz respeito à computação gráfica. Uma ocorrência que está possivelmente ligada à própria natureza do problema abordado.

A alteração da topologia de um cenário de jogo em razão de circunstâncias que possuiriam tal capacidade em um ambiente real não era possível no passado. Em outras palavras, isto significa que a ocorrência de uma explosão não teria o efeito esperado de alterar a topologia do terreno e criar uma cratera, ou de forma similar, pneus de veículos também não possuiam formas de deixar marcas em solos macios como areia e outros (Cai et al., 2006).

O comum era que este tipo de situação se desse apenas em ocasiões específicas, nas quais possuía importância explicita para o desenvolvimento da história do jogo e, portanto, era devidamente pré-programada para ocorrer em um ponto particular e em razão de uma circunstância

ou ação em especial.

Ainda que o uso de artifícios para simular alterações na topologia de um terreno de jogo continue sendo comuns, como por exemplo, no jogo *Counter-Strike* que simplesmente posiciona a imagem de uma cratera na área de uma explosão, sem realizar qualquer real alteração da topologia do terreno afetado (Cai et al., 2006), já existem algoritmos que se prestem a simular a alteração da topologia de um terreno em razão de determinados acontecimentos ou de outras circunstâncias.

Para o tipo de problema abordado aqui, quando aplicado a ambientes que constituem cenários verdadeiramente tridimensionais formados pela agregação de polígonos, as técnicas disponíveis à aplicação, normalmente, envolvem alguma forma do método de múltipla-resolução em razão da perspectiva. Uma técnica que tem mostrado significativos avanços em relação as que utilizam o modelo discreto do nível de detalhamento (*Level of Detail* - LOD) baseado em distância (He et al., 2002), para minimizar o número de polígonos a serem renderizados[1].

Esta e outras técnicas comumente empregadas para a construção de modelos mais complexos, com o propósito de representar e visualizar alterações em terrenos serão abordadas ao longo desta seção.

2.2.1 Múltipla-resolução

A malha de múltipla-resolução para representar um objeto geométrico M é uma representação que incorpora um conjunto de malhas $\{M^1, M^2, \ldots\}$, cada uma como uma representação diferente do objeto original M em respeito a uma dada tolerância (Shamir et al., 2000).

Tal contextualização, permite caracterizar o método de múltipla-resolução (*multiresolution*), também conhecido como múltiplo nível de detalhamento (LOD), como uma seqüência de aproximações para uma dada superfície de entrada, com cada elemento da seqüência equivalendo a um nível de detalhamento diferente dos demais (He et al., 2002).

Para isto, as aproximações são armazenadas em uma estrutura hierárquica, que freqüentemente adota modelos de grafos diretos acíclicos (*Directed Acyclic Graph* - DAG), ou alguma outra estrutura em forma de árvore. Em qualquer dos casos, os nós da estrutura selecionada passam a representar diferentes partes de aproximação para variados detalhes da superfície de entrada, assumindo que os nós próximos ao topo da estrutura representam as aproximações de menor detalhamento. Este tipo de modelo assume ainda que às arestas seja reservado o papel de relacionar os trechos correspondendo a diferentes aproximações (He et al., 2002).

O método de múltipla-resolução também considera o cálculo do desvio, ou erro, das malhas de aproximação contidas na estrutura hierárquica em relação à superfície. É neste ponto que os algoritmos que adotam a idéia de se trabalhar este modelo em razão de uma perspectiva, utilizam de um espaço de erro em relação à tela, de forma a medir o tamanho do erro da aproximação

[1] Renderização é o processo pelo qual se pode obter o produto final de um processamento digital qualquer, sendo normalmente aplicado em programas de modelagem 2D e 3D, áudio, ou vídeo.

geométrica projetada na tela (He et al., 2002).

Neste contexto enquadra-se ainda um espaço de erro global, do qual é originalmente derivado o erro da tela. O espaço de erro global é visto de maneira independente da perspectiva, de forma que, para cada iteração a hierarquia é "podada" em vista de se alcançar uma única aproximação de superfície que satisfaça o critério de erro (He et al., 2002).

Este tipo de modelo caracteriza-se ainda por possuir dois tipos de classificação que possuem como propósito, distinguir o modelo em virtude de sua estrutura hierárquica, a qual pode ser estabelecida como uma estrutura de hierarquias de vértices, ou, como uma estrutura de hierarquia de faces.

Os modelos de múltipla-resolução por hierarquia de faces (*Face Hierarchy Models* - FHMs) focam a construção do modelo na indicação da relação entre as faces de aproximação de diferentes níveis de detalhamento. Exemplos de algoritmos que exploram este tipo de hierarquia podem ser encontrados em Lindstrom & Pascucci (2001), Lindstrom et al. (1996), Scarlatos & Pavlidis (1992), de Berg & Dobrindt (1995), Gross et al. (1995), De Floriani et al. (1998), Duchaineau et al. (1997), sendo ainda possível encontrar um modelo de hierarquia de faces de propósito geral no trabalho apresentado em De Floriani (1989).

Já os modelos de múltipla-resolução por hierarquia de vértices (*Vertex Hierarchy Models* - VHMs) fundamentam a base de sua construção em explorar a relação entre os vértices de aproximação de diferentes níveis de detalhamento. Este tipo de modelo assume que cada nó em sua hierarquia corresponde a um conjunto de vértices. Por sua vez, o conjunto de vértices designado por seus nós filhos são utilizados para substituir o vértice atual em um subseqüente refinamento da aproximação da superfície de entrada. Além disso, a medida de erro mencionada em parágrafos anteriores pode ser associada a cada um dos nós da hierarquia.

Por fim, o modelo afirma que informação adicional sobre a distribuição dos polígonos da imagem se faz necessária e que esta informação pode ser associada a cada nó da hierarquia. Uma característica devida ao fato de que, a informação obtida exclusivamente através do vértice é incapaz de determinar a aproximação em relação à superfície. Exemplos de algoritmos que adotam um modelo de hierarquia de vértices podem ser encontrados em Rossignac & Borrel (1993), Luebke & Erikson (1997), Hoppe (1996) (1997) (1998), Xia & Varshney (1996), e Garland & Heckbert (1997).

É necessário atentar ao fato de que este tipo de modelo costuma trabalhar a partir da suposição de que a entrada do sistema é dada por uma superfície de geometria estática, ou então, de que as mudanças ocorridas em razão da passagem de tempo, são determinadas através de características e parâmetros pré-definidos.

Em ambos os casos, todas as alterações são conhecidas antes mesmo que a renderização do cenário seja feita, permitindo que assim a aplicação de uma etapa de pré-processamento que vise gerar uma múltipla-resolução da estrutura, a partir da qual é possível realizar uma renderização eficiente em tempo de execução. Uma suposição que não condiz com o contexto de uma aplicação de terreno dinâmico em tempo real (He et al., 2002).

2.2.2 Visualização de terreno dinâmico em tempo real

Os métodos de visualização de terrenos dinâmicos em tempo real são normalmente baseados em algoritmos de múltipla-resolução com alterações estruturais. As alterações aplicadas possuem o propósito de viabilizar o suporte a situações que modifiquem o terreno apresentado, dentro do contexto de tempo real.

A natureza destas alterações consiste em que a estrutura hierárquica seja inicialmente construída a partir da superfície de entrada original. Uma vez que isto tenha sido feito, é possível modificar esta estrutura de maneira que reflita mudanças que venham a ocorrer na superfície de entrada (He et al., 2002).

Alguns algoritmos, como os encontrados em Schroeder et al. (1992) e Hoppe (1996), empregam ainda como metodologia, a construção da seqüência de aproximação para estrutura de entrada por meio de processos de otimização, desta forma, garantindo que cada aproximação representada pela estrutura simplifique a aproximação dos objetos da malha anterior, ao mesmo tempo em que aumenta o valor do parâmetro do erro de aproximação o mínimo possível.

Há também o algoritmo descrito em Sumner et al. (1999), que oferece um modelo para a simulação de superfícies de terreno suscetíveis a deformações causadas pelo impacto dos corpos rígidos de personagens animados contra elas.

Neste modelo, foi incluída a exibição de pegadas criadas por um corredor em terrenos compostos por areia, lama e neve. O modelo também incluía a exibição de marcas criadas pelos pneus de uma bicicleta, bem como a colisão da bicicleta contra a superfície do terreno e as deformações resultantes da queda do corredor sobre a superfície.

Dentre as desvantagens deste modelo, podem ser citados os fatos de que, ainda que as pegadas criadas pelo corredor exibissem nítidas diferenças entre os três tipos de terrenos, o modelo da simulação das superfícies de terreno era de pequena escala. Além disso, o método empregado não podia ser aplicado como valor de treinamento de sistemas voltados para a visualização da simulação de terrenos.

Outro dos modelos oferecidos para o propósito da visualização de terrenos dinâmicos em tempo real é o modelo apresentado por Shamir et al. (2000) que consiste em uma abordagem de visualização de superfícies baseada em uma múltipla-resolução dinâmica, que usa de um grafo direto acíclico para formar a estrutura hierárquica do modelo, e assim, permitir a modificação incremental do grafo ao mesmo tempo em que ocorre a deformação da superfície.

O resultado deste modelo foi à criação de uma super-hierarquia, batizada pelos autores de T-DAG, que trabalha em razão da combinação dos grafos de todos os passos de tempo da execução.

A abordagem deste modelo obteve sucesso para a visualização dinâmica de superfície sujeitas a alterações arbitrárias, inclusive no que diz respeito à relação entre a topologia e sua conectividade. A desvantagem deste modelo reside no fato de não constituir um padrão adequado para atualizações *online* em virtude do custo relativamente alto da modificação dos grafos

do T-DAG.

Outra abordagem com relação ao tipo de metodologia abordada por este tópico é apresentada por He et al. (2002), que proporcionou um modelo para representação e exibição de terrenos dinâmicos tomando como base a aplicação do conceito de múltipla-resolução em razão de uma perspectiva.

Para construir este modelo, foi utilizada uma extensão do modelo de malha de adaptação ideal em tempo real (*real-time optimally adapting mesh* - ROAM) (Duchaineau et al., 1997). Isto resultou em uma abordagem dotada de uma eficiente hierarquia para a atualização do terreno ao mesmo tempo em que sua deformação ocorre.

Além disso, a abordagem utiliza do modelo de extensão dinâmica da resolução (*Dynamic EXTEnsion of Resolution* - DEXTER) para garantir uma eficiente extensão da memória de processamento, que é solicitada apenas quando se faz necessária.

A desvantagem deste modelo reside no fato de se adequar apenas à simulação da direção de automóveis em tempo real, com o detalhe de que tal simulação destinava-se apenas para ambientes de tipo *off-road*.

Outro fator negativo relaciona-se as propriedades de deformação do terreno consideradas pelo modelo. As propriedades adotadas não são realistas e não condizem com aquelas estabelecidas pelos modelos baseados na física da deformação de terrenos do mundo real, trabalhando ainda com terrenos de pequena escala e desprezando a continuidade entre as regiões trabalhadas.

Um modelo similar é apresentado em Cai et al. (2006) , que oferece um modelo de múltipla-resolução do nível de detalhamento (*multiresolution* LOD) para visualização dinâmica de terreno, que emprega de uma extensão da malha de adaptação ideal em tempo real (ROAM), um modelo que será apresentado em maiores detalhes na próxima seção.

O modelo de Cai et al. (2006) trabalha utilizando-se dos valores de erro dos elementos da malha de adaptação (tanto os erros em razão da perspectiva quanto aqueles independentes da perspectiva) das filas de triângulos formando diamantes (tanto aquelas destinadas à união como as destinadas à quebra de seus elementos) e desta forma, consegue aumentar a taxa de atualização dos quadros da imagem.

O modelo também aplica a geração procedural de texturas de maneira a gerar uma textura que coincida com a deformação de terreno, utilizando de uma varredura de cada *pixel* do mapa de texturas a fim de determinar os valores RGB de cada um. Entretanto, este tipo de metodologia requer que o tipo de resultado esperado seja previamente conhecido.

No estudo de caso do modelo, o qual é voltado para a criação de crateras em campos de batalha, pressupõem-se que as partes da cratera mais próximas ao centro são as mais queimadas, e portanto, formam as regiões mais escuras que devem clarear para um tom de solo a medida que se afastam do centro para, ao fim, retornar a cor original do terreno.

Apesar de ter apresentado bons resultados para a representação e exibição de terrenos dinâmicos, por meio de eficientes atualizações da hierarquia do terreno à medida que passa por deformações. Este modelo possui a desvantagem de ser destinado exclusivamente ao tratamento

e criação do objeto de seu estudo de caso, crateras em campos de batalha, não englobando qualquer outro tipo de deformação do terreno.

2.2.3 Malha de adaptação ideal em tempo real (*Real-Time Optimally Adapting Mesh* - ROAM)

O modelo de malha de adaptação ideal em tempo real, abreviado pelo acrônimo ROAM derivado do nome do modelo em sua língua nativa, foi apresentado pela primeira vez em Duchaineau et al. (1997). É um tipo de algoritmo destinado a visualização de terrenos, que gera adaptativamente malhas de triângulos retângulos isósceles para renderizar a camada que define a base regular da grade do terreno.

A representação da estrutura hierárquica deste modelo, estabelecida durante uma etapa de pré-processamento, corresponde a de uma árvore binária construída seguindo padrões de triângulos retângulos isósceles. Dentro desta arquitetura, cada triângulo que não corresponda a um nó folha possui dois filhos, que são obtidos, através da operação de separação do triângulo por meio de uma aresta que ligue seu vértice superior ao ponto médio de sua aresta base, como pode ser abstraído a partir da Figura 2.8.

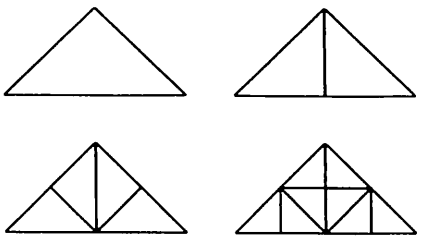


Figura 2.8: *Uma árvore binária de triângulos retângulos isósceles de quatro níveis*

As visualizações proporcionadas por este modelo são livres de eventuais "quebras" em sua continuidade. Esta continuidade das malhas é alcançada através do controle mantido sobre a construção da vizinhança de cada triângulo da árvore, que realiza sucessivas operações de separação e união dos triângulos para as áreas em que foram identificados problemas, um procedimento que se repete até que todos os triângulos da região tenham sido convertidos em triângulos retângulos isósceles (Duchaineau et al., 1997).

O modelo é capaz ainda de realizar a representação de uma área quadrada. Para se obter esta configuração, um par de árvores binárias compostas por triângulos retângulos isósceles é necessário. E necessário ainda que, os triângulos do nó raiz das duas árvores, denominados de diamantes na terminologia do algoritmo, possuam o mesmo tamanho e compartilhem a mesma aresta para a base do triângulo. Uma área quadrada pode ser inferida a partir da área central do terceiro triângulo da Figura 2.8.

E a partir de fileiras de triângulos-diamantes encaminhadas as operações de quebra e união que o algoritmo realiza progressivamente as atualizações em sua malha. Em ambas as filas,

os triângulos-diamantes são ordenados de acordo com sua prioridade, que pode ser calculada a partir de uma dada métrica de erro. O algoritmo garante ainda que a prioridade seja única de cada elemento.

Similarmente a outros modelos hierárquicos, uma "poda" da estrutura de árvore é necessária para que se possa obter uma representação do terreno em forma de malha. O critério utilizado para esta operação é o de uma métrica de erro em razão da perspectiva, de forma que a cada quadro, a "poda" resultante do quadro anterior é ajustada movendo-se a linha de corte para baixo em direção a algum dos nós folha, e para cima na direção da raiz da árvore.

Esta operação considera ainda que seja intuitivo que alguns triângulos da malha anterior sejam substituídos por aqueles nós que descendem deles na árvore, e também, que alguns dos triângulos sejam substituídos por seus nós ancestrais.

Os ajustes realizados por este processo correspondem precisamente às operações de separação e união anteriormente mencionadas nesta seção. Neste contexto, a operação de separação substitui um triângulo por seus dois nós filhos na estrutura de árvore, de forma similar, a operação de união substitui dois nós que descendam de um mesmo nó pai por este mesmo nó pai. Tal semântica deixa evidente que a operação de separação é o inverso da operação de união, e vice-versa.

Ao se transpor este modelo para o contexto de uma aplicação dinâmica, uma vez que o modelo faz a adoção, por meio das arvores binárias de sua estrutura, de uma abordagem de hierarquia que não está vinculada de qualquer forma aos dados que estão armazenados, não há necessidade em se alterar a estrutura básica do modelo ao se modificar o terreno.

Contudo, é necessário atualizar o valor dos espaços de erros globais dos elementos da malha, ou seja, os triângulos, a cada interação. A operação de se computar os espaços de erros globais está implementada de acordo com um modelo arquitetural de baixo para cima (*bottom-up*). Logo, as atualizações no modelo dinâmico também devem seguir este tipo de arquitetura.

Indicadores de atualização são adicionados a estrutura de dados dos vértices, permitindo que os triângulos afetados pela atualização possam ser facilmente identificados através da verificação de ocorrências de modificações em seus vértices.

A atualização dos erros assume que os nós folhas possuem valor zero, a exceção dos que sofreram modificações. Os nós folhas alterados, informam este fato a seus nós pais, que solicitam que seus próprios erros sejam recalculados. Esta informação é passada de nó antecessor a nó antecessor, até alcançar aos triângulos localizados nas raízes das árvores, fazendo com que o erro de cada nó de triângulo notificado seja recalculado (He, 2000). O processo de atualização deve estender-se também a malha de triângulos de maneira a refletir, por exemplo, mudanças na elevação do terreno. Maiores detalhes deste processo podem ser consultados em He et al. (2002).

Capítulo 3

Abordagens adotadas

A abordagem utilizada nesta dissertação realizou o emprego do GAADT como solução para o problema da alteração das características topológicas de um mapa de jogo em função de um tipo de alteração desejado para o cenário representado por este mapa, tal como foi apresentado na seção 1.2.

Todavia, a definição dada na referida seção carece de uma representação formal. Um requisito que necessita primeiro que determinados conceitos sobre a constituição dos tipos de mapa de jogo sendo utilizados aqui sejam mostrados em maiores detalhes.

Este capítulo pretende então preencher esta lacuna, apresentando em detalhes os conceitos de mapa de jogo e dos algoritmos de detecção de bordas, conceitos que serão essências não apenas à definição formal do problema como também para que, ao final deste capítulo, seja possível compreender como se procedeu para aplicar o GAADT ao problema pretendido neste trabalho.

Agora, o restante deste capítulo tratará de apresentar como se deu a aplicação do modelo do GAADT ao problema apresentado por esta dissertação. Este capítulo irá então iniciar com a definição dos conceitos referentes à composição dos mapas de jogo sendo empregados nesta dissertação ao longo de toda a seção 3.1.

Em seguida, a seção 3.2 irá se encarregar de apresentar os conceitos dos algoritmos de detecção de bordas, os quais, serão utilizados tanto durante a definição formal do problema sendo tratado nesta dissertação, que será fornecida na seção 3.3, como também, pela definição da função grau do GAADT, que será apresentada no próximo capítulo deste trabalho, juntamente com os demais conceitos do GAADT.

3.1 Conceitos do tipo dos mapas adotados

Esta seção discutirá os conceitos que caracterizam os mapas de jogo sendo utilizados neste trabalho e que são pertinentes à compreensão das discussões que serão realizadas deste ponto em diante. Tais conceitos estão ilustrados na Figura 3.1, que também retrata a aparência original

da aplicação utilizada por este trabalho e que será apresentada mais adiante, ao longo do capítulo 5. Os próximos itens desta seção tratarão de descrever os elementos representados nesta figura.

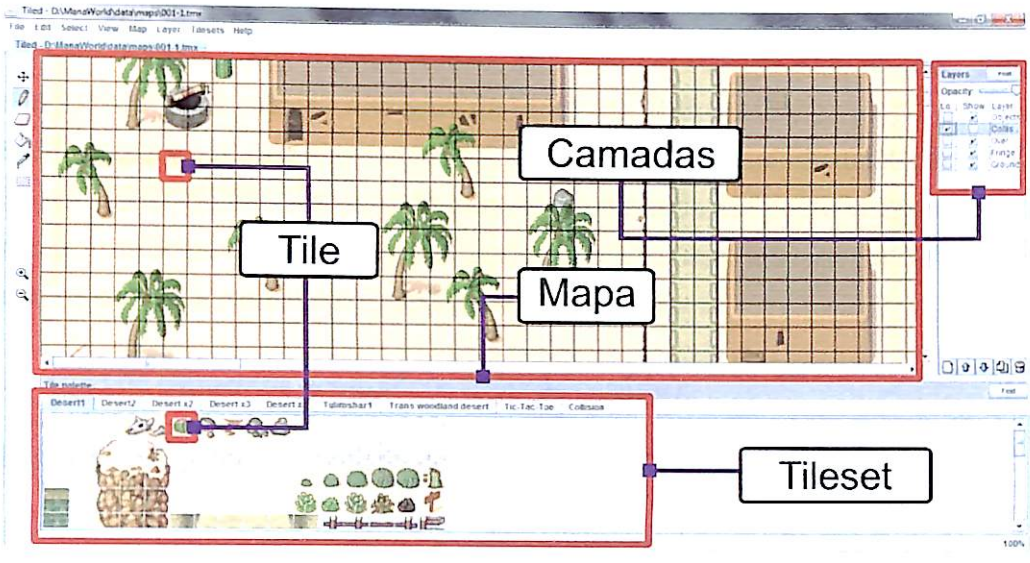


Figura 3.1: *Ilustração dos conceitos dos mapas adotados*

3.1.1 Mapa

Um mapa corresponde a um cenário de jogo que compõe o mundo habitado pelos jogadores e pelas demais entidades que nele atuam. No processo de se construir um mapa de jogo utilizando a aplicação herdada, o mapa é representado como uma malha tridimensional de comprimento w_{mapa} por largura h_{mapa}, com $\{w_{mapa}, h_{mapa} \in \mathbb{N} | (w_{mapa} > 0) \wedge (h_{mapa} > 0)\}$ e ainda com $w_{mapa} \leq h_{mapa}$, ou $h_{mapa} \leq w_{mapa}$. Além disto, um mapa é composto ainda de uma profundidade de l_{mapa} camadas, com l_{mapa} sendo tal que $\{l_{mapa} \in \mathbb{N} |\ l_{mapa} \geq 1\}$.

Sendo assim, é intuitivo imaginar que o número máximo de células de um mapa, e por conseqüência, o número máximo de elementos que podem ser alocados a ele durante sua construção, é definido por um valor arbitrariamente chamado de a, o qual, uma vez que este trabalho faz uso apenas de mapas retangulares, é calculado como $a = (w_{mapa} \times h_{mapa} \times l_{mapa})$. Por sua vez, cada célula do mapa possui uma profundidade de valor um, com um comprimento w_{cel} e uma largura h_{cel} que são definidas pelo usuário, de forma que $\{w_{cel}, h_{cel} \in \mathbb{N} | (0 < w_{cel} \leq w_{mapa}) \wedge (0 < h_{cel} \leq h_{mapa})\}$.

3.1.2 Camadas

As camadas são os níveis em que os elementos do mapa ficam distribuídos em função do comprimento e largura do mapa. Logo, a construção do mapa se dá através da distribuição de seus elementos em diferentes profundidades de sua malha, sobrepondo-os na forma característica de uma pilha de objetos. Afirma-se ainda que, um mapa deve possuir uma profundidade l_{mapa}, de forma que $\{l_{mapa} \in \mathbb{N} |\ l_{mapa} \geq 1\}$.

Com esta afirmação percebe-se que, ainda que o cenário de um mapa seja primariamente explorado de forma bidimensional, sua construção leva em conta três dimensões, uma vez que os elementos são dispostos no cenário em razão de coordenadas indicando a distância (tomada em relação aos eixos x e y) e a profundidade (tomada em relação ao eixo z) na qual qualquer elemento se encontra em relação à origem do mapa em $[0,0,0]$.

Dito isto, percebe-se que a construção do mapa baseia-se efetivamente na atribuição de elementos as células de sua malha, cada uma, representando um ponto em coordenadas $[x,y,z]$ de um sistema tradicional de coordenadas espaciais, neste caso, um sistema tridimensional. Por fim, afirma-se que uma camada do mapa pode ser destinada a:

- Alocação de elementos geográficos (*tiles* de cenário);
- Marcação dos pontos de colisão do cenário (*tiles* de colisão);
- Alocação de objetos que, neste caso, são compostos de uma variedade de tipos que inclui, por exemplo, os NPCs[1], monstros e portais para outras áreas do jogo.

Tal afirmativa determina ainda que, apenas é possível alocar nas células de uma camada elementos que sejam de seu mesmo tipo. Isto significa que, não é possível alocar elementos geográficos em uma camada de objetos, ou de forma inversa, alocar objetos em uma camada de elementos geográficos. Esta regra não se aplica as camadas de colisão e de elementos geográficos, já que ambas se tratam na verdade de camadas de *tiles* que se diferem apenas em razão de sua especialização. Contudo, cabe aqui a ressalva de que, sempre que daqui em diante este trabalho se referir a *tiles* ou a camada de *tiles*, a menos que esteja explicito no texto, não se está levando em consideração aqueles elementos que tratam da colisão de objetos.

3.1.3 Tiles

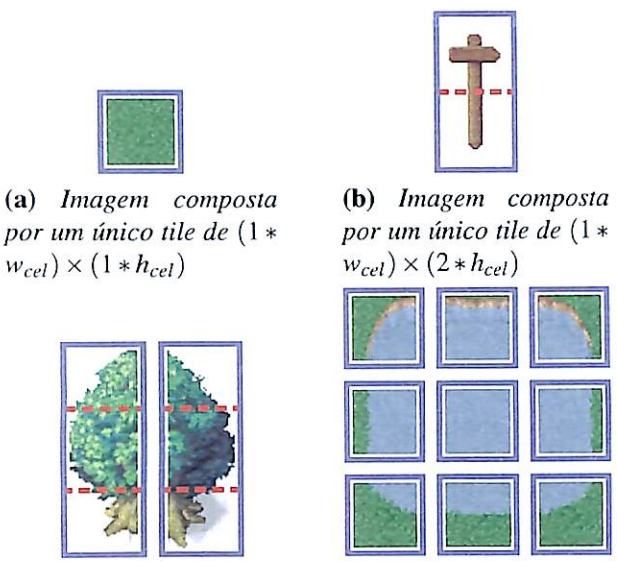
O *tile* (em português, tijolo) é a entidade que ocupa as células da malha do mapa, no papel de entidade geográfica. É basicamente um objeto contendo uma imagem de cenário que pode ser associada a uma ou mais células da malha do mapa.

Um *tile* possui dimensões $(i * w_{cel}) \times (j * h_{cel})$ com w_{cel} e h_{cel} respectivamente o comprimento e a largura de uma célula da malha do mapa, e com i e j constantes tais que $\{i,j \in \mathbb{N} | (1 \leq i < w_{mapa}) \wedge (1 \leq j < h_{mapa})\}$. Para este trabalho, entretanto, se assume que as constantes i e j são tais que $\{i,j \in \mathbb{N} | (i = 1) \wedge (1 \leq j < h_{mapa})\}$. A Figura 3.2 corresponde a exemplos de *tiles* de um mapa de jogo para diferentes dimensões e imagens de um cenário de jogo. No que diz respeito ao espaço ocupado em relação às camadas do mapa, afirma-se que um *tile* ocupa uma única camada do mapa, no caso, a da célula em que está inserido.

[1]Sigla para *Non-Player Characters* - personagens não jogáveis. São quaisquer personagens que não estejam sendo controlados por algum jogador, i.e., os personagens que são controlados pelo computador.

Desta forma um *tile* está sempre posicionado em um ponto $[x,y,z]$ da malha do mapa. Mesmo que um *tile* extrapole os limites de sua célula de origem e invada células vizinhas, considera-se que o *tile* ocupa apenas a célula na qual foi inicialmente alocado.

De fato, quando as dimensões de largura e/ou comprimento de um *tile* excedem os limites definidos para os de uma céula da malha do mapa, *tiles* na mesma camada que ele e que estejam posicionados em células invadidas, na grande maioria das vezes, ainda serão visíveis através do *tile* que as invadiu, dando a impressão de que estão sendo visto por cima ou por baixo do *tile* invasor, de forma similar à impressão causada pela sobreposição de *tiles* em células localizadas em diferentes camadas do mapa e que possuem coordenadas x,y idênticas. Este efeito pode ser mais bem observado na imagem 4.1c da página 63 do capítulo 4.



(a) *Imagem composta por um único tile de* $(1 * w_{cel}) \times (1 * h_{cel})$

(b) *Imagem composta por um único tile de* $(1 * w_{cel}) \times (2 * h_{cel})$

(c) *Imagem composta por um conjunto de tiles de* $(1 * w_{cel}) \times (3 * h_{cel})$

(d) *Imagem composta por um conjunto de tiles de* $(1 * w_{cel}) \times (1 * h_{cel})$

Figura 3.2: *Representação de imagens do cenário em razão quantidade de tiles e suas dimensões*

3.1.4 Tilesets

Um *tileset* é uma imagem maior na qual está contido um conjunto de imagens representando pedaços de diferentes elementos geográficos de um cenário.

O editor de mapas realiza o mapeamento das imagens contidas em um *tileset*, recortando-as de acordo com os valores w_{cel} e h_{cel} definidos pelo usuário, criando assim um conjunto de *tiles* a partir deste *tileset*. Normalmente um mapa possui vários *tilesets*, cada um com conjunto próprio de *tiles* vinculado a ele.

3.2 Os algoritmos para detecção de bordas

A técnica de detecção de bordas é um processo pelo qual se tenta capturar as propriedades significantes dos objetos em uma imagem. Estas propriedades incluem descontinuidade nos atributos fotométricos, geométricos e físicos das características do objeto representado (Ziou & Tabbone, 1998).

Estas informações geram variações em uma imagem retratada em escala de cinza, sendo o mais comum, o uso das variações em descontinuidades (arestas de passo, *step edges*), extremidades locais (linhas de aresta, *line edges*) e estruturas bidimensionais formadas no entroncamento de ao menos duas arestas (junção, *junction*) (Ziou & Tabbone, 1998).

A detecção de bordas tem como propósito a localização e a identificação dos fenômenos físicos que causam estas variações. As arestas físicas de uma imagem oferecem importante informação visual para viabilizar este processo, uma vez que correspondem a descontinuidades do objeto retratado no meio físico, bem em sua geometria e fotometria[2].

Ainda de acordo com o trabalho de Ziou & Tabbone (1998), o passo é o tipo mais comum de aresta encontrado, sendo o resultado de variados fenômenos, dentre os quais estão à ocorrência de um objeto parcialmente escondido por outro, ou mesmo a projeção de uma sombra sobre uma superfície. Na prática, podem ser vistos como a combinação de diversos pontos de inflexão. É na aresta de passo que se dão as descontinuidades nos níveis de cinza de uma imagem. Este tipo de aresta é encontrado por meio do cálculo do operador de gradiente[3] de uma imagem, resultando em uma derivada de primeira ordem para a imagem, na qual as arestas de passo são representadas por máximos positivos ou mínimos negativos da derivada. Também é possível encontrá-las através da segunda derivada, resultante da aplicação do operador de Laplace[4] sobre a imagem, na qual as arestas de passo correspondem aos pontos em que o valor da segunda derivada é zero.

Já as arestas de linhas são o resultado da iluminação mutua entre objetos em contato direto ou entre objetos finos colocados contra um plano de fundo. Os zeros correspondem aos extremos locais de uma imagem e localizam-se nos pontos de valor zero para a primeira derivada da imagem, ou nos máximos locais dos resultados da segunda derivada da imagem, dada pelo operador de Laplace. Outros tipos de arestas podem ser encontrados em Ponce & Brady (1985).

O último destes elementos, as junções, são formadas em resposta as junções físicas da imagem, que resultam do encontro de pelo menos duas arestas físicas da cena representada. Há ainda outras circunstâncias que justificam a formação de junções, como por exemplo, efeitos de iluminação ou oclusão. Existem diversos tipos de junção que podem ser formados, tomando

[2] A fotometria é a ciência de medição da luz em termos de como é percebida pelo olho humano. Para os casos descritos neste trabalho a fotometria é comumente medida com base na luminância dos *pixels* da imagem.

[3] O gradiente é uma função escalar $f(x_1, x_2, \ldots, x_n)$ dada por (∇f ou $\vec{\nabla} f$) com delta sendo o operador diferencial da função e que aponta na direção que apresenta a maior ampliação do campo escalar, sendo que o maior fator de ampliação corresponde a sua magnitude.

[4] O operador de Laplace é um operador diferencial de segunda ordem em um espaço Euclidiano de n dimensões e definido como a divergência (∇) do gradiente (∇f).

as formas geométricas das letras T, L, Y, X e ainda de outras tantas. Junções podem ser identificadas de várias formas, como por exemplo, pontos com grandes curvaturas, ou uma grande variação na direção do gradiente (Ziou & Tabbone, 1998).

Um algoritmo para detecção de bordas, aceita imagens discretas e digitalizadas como entrada e produz como resultado um mapa de arestas. Em alguns algoritmos, este mapa pode conter ainda informação explicita quanto a posição e a robustez das linhas das arestas, e ainda, sua orientação e escala. De um ponto de vista computacional estes algoritmos podem ser divididos em dois tipos de modelo.

O primeiro destes corresponde aos algoritmos que não fazem uso de conhecimento prévio da cena retratada e das arestas a serem detectadas, sendo classificados então como detectores "autônomos" que não sofrem influência do sistema de captura da imagem ou do contexto da informação recebida.

Desta forma, são algoritmos flexíveis no sentido de que não se limitam a imagens específicas. Contudo, são baseados em um processamento local que considera apenas a identificação das arestas com base em seus *pixels* vizinhos.

O segundo tipo é dito como "contextual" no sentido de que são guiados pelos resultados de outros componentes do sistema, ou, por um conhecimento prévio das arestas ou da estrutura da cena. Portanto, são normalmente empregados para propósitos específicos nos quais as imagens a serem processadas sempre incluem os mesmos objetos.

De uma perspectiva conceitual, os esquemas para detecção de bordas normalmente propostos para estes dois tipos de modelos incluem as operações de diferenciação, suavização e rotulação.

O processo de diferenciação consiste da avaliação das derivadas desejáveis da imagem, o que pode ocorrer através da computação das derivadas necessárias para se localizar as arestas de uma imagem, causadas pela variação da escala de cinza de uma imagem em razão dos fenômenos físicos que produzem suas arestas. O operador de diferenciação é caracterizado por sua ordem, dada em respeito à ordem de suas derivadas parciais; sua invariância, dada em razão da invariância de seu operador $\Omega_{x,y}$ em relação a sua matriz de rotação; e por sua linearidade, dada também em razão de seu operador $\Omega_{x,y}$, de forma que, se para quaisquer escalares positivos α e β e para quaisquer funções $f(x,y)$ e $g(x,y)$ for verdade afirmar que $\Omega_{x,y}(\alpha f(x,y) + \beta g(x,y)) = \alpha\Omega_{x,y}(f(x,y)) + \beta\Omega_{x,y}(g(x,y))$ então o processo é linear. Perceba que o operador $\Omega_{x,y}$ é o próprio processo de diferenciação, sendo ainda correto afirmar que de forma a satisfazer estes critérios, os operadores mais utilizados para a operação de diferenciação são o operador de gradiente, o operador de Laplace, e as derivadas direcionais de segunda ordem. Maiores detalhes sobre este processo podem ser encontrados em Torre & Poggio (1986), Lyvers & Mitchell (1988), Wu & Xie (1990), Freeman & Adelson (1991), Ziou & Tabbone (1998).

O processo de suavização consiste em reduzir o ruído da imagem e regularizar a diferenciação numérica. Entretanto, a aplicação do recurso de suavização deve ser cuidadosa, uma vez que

a redução do ruído da imagem também resulta em perda de informação. Por esta razão, o uso do recurso de suavização deve ser guiado para alcançar uma filtragem com um meio termo ótimo entre redução de ruído e conservação de arestas. Este último contexto corresponde basicamente ao que se encontra por traz da teoria da regularização, que formaliza como um problema de otimização a busca por filtro ótimo (Poggio et al., 1988). Maiores detalhes sobre suavização de imagem podem ser encontrados em Brady & Horn (1983), Poggio & Torre (1984), Torre & Poggio (1986), Pitas & Venetsanopoulos (1986), Poggio et al. (1988), Danielsson & Seger (1990), Ziou & Fabre (1994), Ziou & Wang (1996), Ziou & Tabbone (1998).

A rotulação consiste em localizar as bordas da imagem e aumentar a relação sinal/ruído das arestas da imagem pela supressão de falsas arestas. A supressão de falsas arestas aumenta a proporção sinal/ruído das operações de diferenciação e suavização. A escolha do operador de rotulação, e conseqüentemente, de como ocorre seu processo de localização de arestas, depende do operador de diferenciação sendo utilizado, podendo citar-se como exemplo, os modelos iniciais do operador de rotulação que, para diferenciações utilizando do operador de gradiente, usavam o valor do limiar do modulo do gradiente para identificar as arestas da imagem. Ainda que possa ser verdade que o comportamento desta operação depende do desempenho dos operadores de diferenciação e suavização, outros fenômenos colaboram para a criação de falsas arestas sendo que comumente é utilizada uma regra em razão da plausibilidade de uma aresta para um dado limiar (Canny, 1986) (Ziou & Tabbone, 1998). Discussões mais aprofundadas sobre o processo de rotulação podem ser encontradas em Hueckel (1971), Marr & Hildreth (1980), Canny (1983), Canny (1986), Huertas & Medioni (1986), Clark (1988), Clark (1989), Dellepiane et al. (1989), Tabbone & Ziou (1991), Shen & Castan (1992), Ziou & Tabbone (1993), Tabbone & Ziou (1993), Tabbone (1994), Ziou & Tabbone (1995), Ziou & Tabbone (1998).

3.2.1 O algoritmo de Canny para a detecção de bordas

O algoritmo desenvolvido por Canny para a detecção de bordas, apresentado pela primeira vez em Canny (1986), e conhecido por muitos como o detector de bordas ótimo, cuja intenção era a de aumentar a eficiência dos diversos algoritmos de detecção de bordas disponíveis no tempo em que começou seu trabalho.

O algoritmo segue uma lista de critérios de forma a aprimorar os então correntes métodos de detecção de borda. Estes critérios incluem uma baixa taxa de falsas arestas; a identificação dos pontos das arestas, que é o mesmo que dizer que a distância entre os *pixels* detectados da aresta estão a uma distância mínima da aresta; e a possibilidade de existir apenas uma resposta do detector para uma mesma aresta, um fator ausente em algoritmos anteriores, para os quais era possível se obter múltiplas respostas para uma mesma aresta, contribuindo para a imprecisão dos resultados.

Com base nestes critérios, o algoritmo inicializa seus procedimentos realizando a suavização da imagem de maneira a eliminar os ruídos. Posteriormente, é calculado o gradiente da imagem

que é então utilizada para destacar as regiões de elevadas derivadas espaciais. Estas regiões são então varridas de maneira a remover quaisquer *pixels* alocados que não correspondam a um valor de máximo, em um processo conhecido como supressão de não-máximos.

Em seguida, a cadeia com os valores de gradiente é reduzida por meio de um processo de histerese[5] utilizando dois valores de limiares (um máximo e um mínimo) de tal forma que, caso a magnitude do gradiente se encontre abaixo do valor mínimo, seu valor é definido como zero, indicando assim que não há uma aresta para o ponto daquele *pixel*. Caso a magnitude do gradiente esteja acima do limiar máximo, então o algoritmo interpreta que existe uma aresta naquele ponto e o valor do gradiente é definido como o valor máximo.

Na última situação possível, aquela caracterizada por uma magnitude de gradiente entre os valores de limiar máximo e mínimo, o valor da magnitude do gradiente é definida como zero, a menos que, exista um caminho que leve do *pixel* sendo avaliado até um *pixel* de magnitude acima do limiar máximo. Neste último caso, o valor da magnitude é então definido como um valor máximo. Este processo pode ser descrito com maiores detalhes, tal como dado por Green (2002), da seguinte forma:

- Na primeira etapa, o algoritmo realiza uma filtragem da imagem, que tem como propósito eliminar da mesma quaisquer ruídos antes que seja encaminhada a localização e apropriada detecção de suas arestas. Para este processo é utilizado um filtro Gaussiano de máscara simples, que pode ser calculada ou previamente fornecida para o problema. Esta máscara irá delimitar a área da imagem sendo avaliada, e realizará a suavização por meio de métodos padrões de convolução. Uma vez que a área delimitada pela máscara é normalmente muito menor que a área total da imagem é possível imaginar que, a máscara é deslizada sobre a imagem de forma que a área que delimita corresponde a uma região sendo manipulada, portanto realizando a suavização da imagem uma região de cada vez. Deve-se atentar que a largura determinada para máscara Gaussiana (Ito, 2000) é inversamente proporcional a sensibilidade do algoritmo ao ruído. Isto é o mesmo que dizer que quanto maior a máscara, menor a sensibilidade ao ruído. Além disto, o valor do erro na localização das arestas também aumenta de forma sutil à medida que a largura da máscara é aumentada.

- Após suavizar a imagem eliminando os ruídos, o algoritmo prossegue com a identificação da robustez das arestas com base no gradiente da imagem. O operador de Sobel[6] é utilizado para computar uma medida bidimensional do gradiente espacial de uma imagem.

[5]Histerese é a tendência de sistemas de conservar suas propriedades na ausência do estímulo que as gerou. Podem exibir dependência do caminho (*path dependence*) ou "memória independente da taxa". Sistemas com histerese têm saídas que dependem ou do histórico da entrada do sistema (determinando o caminho percorrido pela entrada até alcançar seu valor atual), ou, de uma inspeção do estado interno do sistema.

[6]O operador de Sobel calcula a gradiente de intensidade para cada ponto de uma imagem, identificando a direção da variação de tons claros para escuros e a quantidade da variação, com isso, estimando a variância de luminosidade para cada ponto da imagem.

A robustez da aresta é então dada para cada ponto da imagem, em relação à aproximação absoluta da magnitude do gradiente. O operador de Sobel utiliza de um par de máscaras de convolução de 3×3, uma destinada ao cálculo da estimação do gradiente no eixo x (colunas) e a outra para o eixo y (linhas). Esta situação é ilustrada pela Figura 3.3. O cálculo da magnitude, ou robustez da aresta, é dado então como na equação 3.1.

$$|G| = |G_x| + |G_y| \tag{3.1}$$

-1	0	+1
-2	0	+2
-1	0	+1

(a) G_x

+1	+2	+1
0	0	0
-1	-2	-1

(b) G_y

Figura 3.3: *Exemplo do operador de Sobel para a estimação do cálculo da gradiente dos eixos x e y para uma máscara de convolução de* 3×3

- A direção da aresta é computada utilizando o valor do gradiente nas direções x e y. Contudo, um erro será gerado sempre que o somatório das componentes do eixo x for igual a zero. O algoritmo deve, portanto, possuir uma restrição que atente a este problema. Sempre que o valor da gradiente na direção x for igual a zero, a direção da aresta deve ser igual a noventa graus, 90^o, ou zero grau, 0^o, dependendo do valor assumido pela gradiente na direção y. Caso G_y tenha valor zero, a direção da gradiente x será de zero grau. Do contrário, será de noventa graus. A fórmula apresentada na equação 3.2 é utilizada para localizar a direção da aresta.

$$\theta = tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{3.2}$$

- Uma vez que a direção da aresta é conhecida, o próximo passo é relacionar esta direção com uma direção que possa ser identificada na imagem. Por exemplo, se os *pixels* de uma imagem de 5×5 são alinhados tal como exibido na Figura 3.4, é possível assumir que partindo do *pixel* Λ marcado em azul, existem apenas quatro direções possíveis no que diz respeito aos *pixels* circundando Λ, os quais são zero grau (na direção horizontal), 45 graus (junto à diagonal positiva), 90 graus (na direção vertical), ou, 135 graus (junto à diagonal negativa). O algoritmo então considera que a orientação da aresta deve ser definida em respeito a uma destas quatro direções, optando-se por aquela de que estiver mais próxima. Em outras palavras, se, por exemplo, a orientação de um ângulo for equivalente a dez graus então deve ser considerada como zero grau. Isto pode ser feito tomando-se a

perspectiva de um semicírculo dividido em cinco partes tal como na Figura 3.5. A partir da imagem é possível intuir que a qualquer direção correspondendo à região delimitada em amarelo (a qual vai de zero a 22,5 graus e de 157,5 a 180 graus, tal que, [0°; 22,5°) e [157,5°; 180°)) é atribuída o valor de zero grau. De forma similar, direções cujos valores corresponderem à região delimitada em verde (definida entre [22,5°; 67,5°)), azul (definida entre [67,5°; 112,5°)), ou vermelho (definida entre [112,5°; 157,5°)), terão seus valores estabelecidos respectivamente como 45, 90 ou 135 graus.

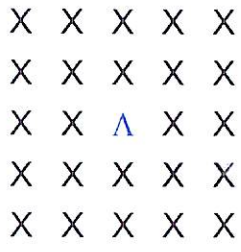


Figura 3.4: *Exemplo do alinhamento de pixels em razão de um pixel central* Λ

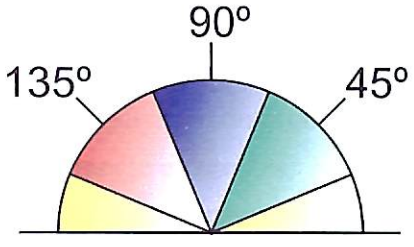


Figura 3.5: *Semicírculo utilizado na aproximação da orientação de arestas em razão das quatro direções principais*

- Conhecidas as direções das arestas, o algoritmo prossegue para a supressão de não-máximos. Este processo é utilizado de forma a varrer uma aresta em sua direção apropriada e remover qualquer *pixel* cujo valor não se enquadra como pertencente a uma aresta. O ato de remoção em si equivale a atribuir a tais *pixels* um valor igual a zero.

- Por último, o processo de histerese é aplicado com o propósito de remover estrias da imagem, formadas devido à quebra do contorno de uma aresta em razão da flutuação do operador de saída acima e abaixo do limiar dado para a imagem. Isto ocorre caso um único valor de limiar seja aplicado a uma imagem e a robustez média da aresta equivaler justamente a este valor. Em tais situações, haverá casos em que, em razão do ruído, a aresta irá oscilar abaixo deste valor. Da mesma forma o valor também oscilará acima deste valor ocasionando que a aresta apareça como uma linha tracejada. É justamente para evitar estes acontecimentos que o algoritmo utiliza dois limiares, um limiar de valor máximo e um limiar de valor mínimo. Qualquer *pixel* com um valor maior que o limiar máximo é tomado como um *pixel* de aresta e marcado como tal. Em seguida, qualquer *pixel* conectado ao primeiro e que tenha um valor maior que o limiar mínimo também

é selecionado como um *pixel* de aresta. Logo, o algoritmo inicia uma aresta para um gradiente maior que o valor do limiar máximo e pára apenas ao atingir um valor abaixo do limiar mínimo.

3.3 Definição formal do problema

Esta seção irá agora apresentar uma definição formal para o problema sendo tratado por esta dissertação, tendo como base os conceitos do problema que foram descritos ao longo da seção 1.2.

Segundo (Lopes, 1981) um problema pode ser visto como uma tupla $\Gamma = \langle D, R, d, i, A \rangle$, na qual, D representa o conjunto dos dados de entrada para o problema; R corresponde ao conjunto dos possíveis resultados; d é um elemento de D que satisfaz a uma série de restrições dadas na forma de predicados; i é a intenção do problema, que também pode se afirmar ser o problema em si; e por fim A que caracteriza as condições de admissibilidade para os resultados possíveis, definindo assim quais deles são de fato saídas aceitas para o problema sendo tratado.

A definição formal de problema que está expressa no parágrafo anterior, em conjunto com as afirmações feitas ao longo da seção 1.2 acerca das características do problema sendo tratado por esta dissertação, possibilitam a formalização do problema de se gerar alterações topológicas na geografia de mapas de jogo previamente existentes.

Tal possibilidade deve-se ainda, ao conhecimento da constituição dos mapas de jogo sobre os quais a intenção do problema definido aqui irá atuar. Informação esta, que foi apresentada ao longo da seção 3.1 e que permite definir o formato da entrada D do problema. Uma definição que está expressa na equação 3.3.

$$D = \{x, y, z, obj, pred | (x \in X_{mapa}) \wedge (y \in Y_{mapa}) \wedge (z \in \mathbb{N}) \wedge (obj \in Objetos) \wedge (pred \in Pred)\} \tag{3.3}$$

A equação 3.3 apresenta assim uma definição formal para a entrada D do problema, i.e. um mapa de jogo, que o caracteriza como um conjunto de pontos definidos por três coordenadas espaciais (x, y e z) que alocam um único objeto (obj). Uma formalização que corresponde à analogia de que um mapa de jogo é um conjunto de células que permitem, cada uma, a alocação de um único objeto.

Os conjuntos que limitam os domínios dos elementos x e y na equação 3.3 estão expressos nas equações 3.4 e 3.5, enquanto que o elemento equivalente a coordenada espacial z é limitado pelo próprio conjunto dos números naturais.

$$X_{mapa} = \{x | (x \in \mathbb{N}) \wedge (0 \leq x < w_{mapa})\} \tag{3.4}$$

$$Y_{mapa} = \{y | (y \in \mathbb{N}) \wedge (0 \leq y < h_{mapa})\} \tag{3.5}$$

Os valores w_{mapa} utilizado em 3.4 e h_{mapa} utilizado em 3.5 correspondem respectivamente ao comprimento e a largura total do mapa de entrada. A definição do elemento obj apresentado na equação 3.3 e do conjunto $Objetos$ que define seu domínio requer primeiro que se compreenda que, o elemento geográfico *tile* apresentado na seção 3.1 não é o único tipo de objeto que pode ser alocado as células de um mapa de jogo.

Além dele, outros objetos possíveis de se alocar nas células do mapa incluem portais de acesso a outros mapas, NPCs, criaturas como monstros (que mesmo possuindo a capacidade de se deslocar pelo cenário do mapa são considerados como que vinculados a apenas uma de suas células), ou mesmo, para o contexto deste trabalho, camadas para a marcação dos pontos de colisão.

Oferecer uma definição formal para os objetos do mapa de jogo que se estenda para além da definição dos elementos geográficos foge ao escopo deste trabalho. Por está razão, esta dissertação irá unicamente afirmar que tais objetos pertencem a um conjunto chamada aqui de $NaoTile$ e que a intersecção deste conjunto com o conjunto dos elementos geográficos do mapa, chamado aqui de $Tile$, resulta no conjunto vazio. Esta linha de raciocínio está expressa nas equações 3.6 e 3.7, enquanto que a equação 3.8 apresenta a formalização que esta dissertação adota para o conjunto $Tile$.

$$Objetos = Tile \cup NaoTile \tag{3.6}$$

$$Tile \cap NaoTile = \varnothing \tag{3.7}$$

$$Tile = \{(w_{cel}, h_{tl}, gId, Img) | (Img \in Imagem) \wedge ((w_{cel}, h_{tl}, gId) \in \mathbb{N})\} \tag{3.8}$$

O valor gId na equação 3.8 equivale a um índice global. Um valor único que diferencia cada *tile* do mapa de jogo dos demais. A equação também estabelece que além deste índice, cada *tile* possui uma imagem associada a ele, a qual, irá sempre possuir uma largura igual a das células do mapa de jogo (w_{cel}) e uma altura dada por h_{tl}. A altura h_{tl} da imagem é estabelecida como sendo maior ou igual à altura das células do mapa, h_{cel}, de forma que $\{h_{tl} = i \times h_{cel} | \ i \in \mathbb{N}^*\}$.

O último elemento da equação 3.3 a ser definido, $pred$, possui como tarefa de distinguir qual o tipo de terreno que predomina no mapa de entrada do problema. Esta informação irá permitir

que se determine o rumo a ser seguido pelas alterações geográficas que serão executadas ao longo dos procedimentos a serem apresentados no próximo capítulo. O valor de *pred* é definido aqui como uma *string* proveniente do conjunto *Pred* que contém todas as *strings* que definem os possíveis valores *pred* e que caracterizam assim os possíveis tipos em que este trabalho permite que se classifique a aparência predominante de um mapa de jogo. Os valores que este trabalho utiliza para construir o conjunto *Pred* estão listados na tabela 3.1.

Valores do conjunto *Pred*
Floresta
Deserto
Caverna
Neve

Tabela 3.1: *Valores compondo o conjunto Pred*

Conhecida a entrada para o problema é necessário estabelecer o tipo de resultado que se espera obter a partir dela. Neste sentido, tendo em mente que ao longo desta dissertação, foi afirmado que o trabalho sendo desenvolvido aqui procura realizar a alteração de regiões do mapa de entrada é possível subentender que, o produto final buscado aqui não equivale a um mapa de jogo completo, mas sim a um conjunto de células pertencentes ao domínio dos pontos espaciais do mapa de entrada e cujo conteúdo difira daquele das células originalmente alocadas nestes pontos.

A adoção desta prática objetiva que o conteúdo do conjunto de células obtido por meio do resultado do problema, venha a substituir o conteúdo das células do mapa de entrada localizadas nas mesmas coordenadas espaciais x, y e z que as células de saída. A lógica para suportar esta idéia esta expressa na maneira como o conjunto de resultados R é definido pela equação 3.9.

$$R = \{x, y, z, tl | (x \in X_{mapa}) \wedge (y \in Y_{mapa}) \wedge (z \in \mathbb{N}) \wedge (tl \in Tile)\} \tag{3.9}$$

Agora que a entrada do problema foi definida em função de seu conjunto de dados de entrada D apresentado na equação 3.3, e que sua saída foi definida em razão de seu conjunto de resultados R, tal como transmitido através da equação 3.9, é então possível definir formalmente a intenção do problema, anteriormente descrita por esta dissertação no decorrer da seção 1.2, em função dos conjuntos D e R tal como expresso pela equação 3.10.

$$i \subseteq D \times R \tag{3.10}$$

Deve ser entendido que mesmo que a intenção do problema esteja dentro do domínio expresso pela equação 3.10, nem todo elemento pertencente a este domínio contempla uma resposta admissível para o que se deseja obter como resultado ao final da solução do problema.

Por este motivo, é necessário que tanto a entrada como a saída do problema sejam submetidas a restrições que "filtrem" estes dados a fim de caracterizar o objetivo do problema, o que inclui simplificar os procedimentos a serem aplicados e definir quais aspectos dos dados devem ser atendidos de maneira que seja possível se trabalhar sobre o problema.

No que diz respeito aos dados de entrada para o problema, a primeira das restrições impostas está relacionada à como se dá a estruturação do mapa de entrada. Em vista de simplificar os procedimentos a serem aplicados, evitando assim eventuais quebras na continuidade dos dados de entrada a serem trabalhados, foi decidido que, um mapa que venha a ser fornecido como entrada deve possuir todas as suas camadas de elementos geográficos antes daquelas responsáveis por alocar demais elementos do cenário. Isto é o mesmo que afirmar que camadas que possuam objetos do conjunto *NaoTile* podem ser vinculadas ao esquema do mapa de jogo apenas após a ocorrência de todas as camadas que alocam objetos do conjunto *Tile*.

Para contemplar esta restrição, a presente dissertação partiu do pressuposto de que uma camada que se destine a alocação de objetos do conjunto *Tile* é incapaz de alocar objetos que não pertençam a este conjunto. Este princípio foi utilizado para estabelecer as equações 3.11, 3.12, 3.13 e 3.14, que utilizam do parâmetro υ para indicar uma célula do mapa de jogo de entrada, tal que $d = \{\upsilon_1, \ldots, \upsilon_n\} \in D$.

$$\begin{aligned}\Upsilon_1 = \{\ & \forall d = \{\upsilon_1, \ldots, \upsilon_n\} \in D\ \forall i \in \{1,2,\ldots,n\} \mid z_i = 0 \\ & \exists \upsilon_i = (x_i, y_i, z_i, obj_i, pred_i) \in \{\upsilon_1, \ldots, \upsilon_n\}\ \}\end{aligned} \tag{3.11}$$

$$\begin{aligned}\Upsilon_2 = \{\ & \forall d = \{\upsilon_1, \ldots, \upsilon_n\} \in D\ \forall i \in \{1,2,\ldots,n\} \\ & \forall \upsilon_i = (x_i, y_i, z_i, obj_i, pred_i) \in \{\upsilon_1, \ldots, \upsilon_n\} \mid (z_i = 0) \rightarrow obj_i \in Tile\ \}\end{aligned} \tag{3.12}$$

$$\begin{aligned}\Upsilon_3 = \{\ & \forall d = \{\upsilon_1, \ldots, \upsilon_n\} \in D\ \forall i \in \{1,2,\ldots,n\}\ \forall \upsilon_i = (x_i, y_i, z_i, obj_i, pred_i) \in \{\upsilon_1, \ldots, \upsilon_n\} \mid \\ & ((z_i > 0) \wedge (obj_i \in Tile)) \rightarrow (\forall j \in \{1,2,\ldots,n\}\,((z_j \leq z_i) \rightarrow obj_j \in Tile))\ \}\end{aligned} \tag{3.13}$$

$$\begin{aligned}\Upsilon_4 = \{\ & \forall d = \{\upsilon_1, \ldots, \upsilon_n\} \in D\ \forall i \in \{1,2,\ldots,n\}\ \forall \upsilon_i = (x_i, y_i, z_i, obj_i, pred_i) \in \{\upsilon_1, \ldots, \upsilon_n\} \mid \\ & ((z_i > 0) \wedge (obj_i \in NaoTile)) \rightarrow (\forall j \in \{1,2,\ldots,n\}\,((z_j \geq z_i) \rightarrow obj_j \in NaoTile))\ \}\end{aligned} \tag{3.14}$$

Uma vez que foram estabelecidos os critérios iniciais para a estrutura da entrada do problema, o que se percebeu em seguida foi que de maneira a lidar apropriadamente com os objetos de tipo *tile*, era necessário que os mesmos possuíssem informação suficiente para permitir distingui-los em razão do tipo de elemento geográfico representado pela imagem vinculada a

cada um deles, visto que em seu formato atual não permitia sequer diferenciar uma árvore de um lago.

A alternativa encontrada foi alimentar os mapas de entrada com mais dados, que vinculariam então um *tile* a um conjunto de atributos que lhe daria informações sobre as características que representa. A forma como estes dados foram alimentados será apresentada mais adiante, apenas no capítulo 5, reservando-se para este momento apenas a descrição das informações que foram vinculadas aos *tiles*, as quais são:

- O tipo de cenário que o *tile* representa (uma *String*);
- O índice do tipo de cenário deste *tile* (um número natural);
- Uma *flag* que indica se o *tile* requer, ou não, a ocorrência de um ponto de colisão (um valor booleano).

Estes novos dados permitiram a criação do conjunto de registros que serão apresentados pela equação 3.15, na qual correspondem respectivamente aos parâmetros tp, in e fl. A informação definida por estes elementos se mostrou tão importante para o trabalho desempenhado por esta dissertação, que resultou na criação da restrição da equação 3.16 de maneira a garantir que todos os tiles utilizados pelo mapa de entrada do problema estejam devidamente associados a estes parâmetros.

$$DscTile = \{(tl, tp, in, fl) | (tl \in Tile) \wedge (tp \in Tipo) \wedge (in \in \mathbb{N}) \wedge (fl \in Boolean)\} \quad (3.15)$$

$$\Upsilon_5 = \{\ \forall d = \{\upsilon_1, \ldots, \upsilon_n\} \in D\ \forall i \in \{1,2,\ldots,n\}\ \forall \upsilon_i = (x_i, y_i, z_i, obj_i, pred_i) \in \{\upsilon_1, \ldots, \upsilon_n\} \mid (obj_i \in Tile) \rightarrow (\exists_1 dsc = (tl, tp, in, fl) \in DscTile\ (obj_i = tl))\ \} \quad (3.16)$$

O conjunto *Tipo* apresentado na equação 3.15 é o responsável por estabelecer a variedade de tipos de elementos geográficos que podem ser atribuídos aos mapas de jogo sendo utilizados nesta dissertação. Uma vez associado os diferentes registros do conjunto *DscTiles* os valores do conjunto *Tipo* juntamente com o valor do parâmetro *in* permitem identificar diferentes variedades de um mesmo tipo de terreno, um conceito que pode ser entendido ao compreender que dois diferentes registros possuindo, por exemplo, $tp = grama$ e $in = 5$ pertencem a uma mesma variedade de grama (neste caso, a 5ª variedade de grama cadastrada), a qual, difere de um terceiro registro definido por $tp = grama$ e $in = 4$, ou de um quarto, dado por $tp = areia$ e $in = 5$. Uma relação completa dos valores encontrados no conjunto *Tipo* está disponível na tabela 3.2.

Valores do conjunto *Tipo*			
Adorno	Água	Areia	Construção
Grama	Montanha	Neve	Obstáculo
Pedra	Residência	Terreno	Vazio
Vegetação			

Tabela 3.2: *Valores compondo o conjunto Tipo*

Por fim, uma vez que o conhecimento do tipo predominante dos elementos geográficos do mapa é necessário para que sejam identificados os tipos de *tiles* que serão trabalhados ao longo do procedimento, foram impostas aos dados de entrada as restrições de que deveria haver um tipo predominante associado ao mapa e que, este tipo predominante deve ser o mesmo para todos os elementos de seu conjunto de células. Estas restrições estão expressas nas equações 3.17 e 3.18.

$$\begin{aligned}\Upsilon_6 = \{\ & \forall d = \{\upsilon_1,\ldots\upsilon_n\} \in D\ \forall i \in \{1.2,\ldots,n\} \\ & \exists \upsilon_i = (x_i.y_i,z_i.obj_i,pred_i) \in \{\upsilon_1,\ldots,\upsilon_n\}\,|(pred_i \in Pred) \wedge (pred_i \neq \varnothing)\ \}\end{aligned} \tag{3.17}$$

$$\begin{aligned}\Upsilon_7 = \{\ & \forall d = \{\upsilon_1,\ldots,\upsilon_n\} \in D\ \forall i,j \in \{1,2,\ldots,n\} \\ & \forall(\upsilon_i = (x_i,y_i,z_i,obj_i,pred_i),\upsilon_j = (x_j,y_j,z_j,obj_j,pred_j)) \in \{\upsilon_1,\ldots,\upsilon_n\}\,| \\ & pred_i = pred_j\ \}\end{aligned} \tag{3.18}$$

Em conjunto, estas restrições definem o formato desejado para a entrada do problema sendo tratado. Logo, para o problema desta dissertação, uma entrada d é aceita como pertencente ao conjunto de entradas D se e somente se satisfaz a todas as restrições que foram estabelecidas para este conjunto. Uma característica que está expressa na equação 3.19.

$$d \subseteq D \leftrightarrow (\Upsilon^1 \wedge \Upsilon^2 \wedge \Upsilon^3 \wedge \Upsilon^4 \wedge \Upsilon^5 \wedge \Upsilon^6 \wedge \Upsilon^7) \tag{3.19}$$

Agora que as restrições do problema estão definidas é necessário estabelecer critérios de admissão que guiem as características que se desejam obter no resultado que será retornado como saída pelo problema. Neste sentido, a definição do primeiro destes critérios se inicia então pela identificação do tipo de estação climática a que será submetido o mapa de entrada. A escolha desta estação é realizada de maneira aleatória pelo sistema e armazenada em um parâmetro denominado *clima*. A equação 3.20 ilustra esta operação para uma definição genérica do conjunto *Clima*.

$$random_1(Clima = \{cl_1, \ldots, cl_n\}) = \{\forall cl_i \in \{cl_1, \ldots, cl_n\}\ \exists_1 i \in \{1, \ldots, n\}\ (clima = cl_i)\} \tag{3.20}$$

Definir a estação climática pela qual o cenário de jogo está passando constitui a essência da alteração geográfica a ser evidenciada no mapa de jogo. Em razão dela, é construída uma seqüência de tipos geográficos que são considerados interessantes de serem trabalhados no mapa de jogo dado como entrada. Esta seqüência é definida por meio de uma função *tipoInt* que mapeia o tipo predominante *pred* obtido a partir dos dados do mapa de entrada *d* e a estação climática *clima* definida pela equação 3.20 na seqüência de tipos de interesse para o problema, chamada aqui de *INT*. A função *tipoInt* está definida na equação 3.21.

$$tipoInt(pred, clima) = \{\ INT = \langle tp_1, \ldots, tp_n \rangle \mid (pred \in Pred) \wedge (clima \in Clima) \wedge (\forall i \in \{1, 2, \ldots, n\}\ \forall tp_i \in \langle tp_1, \ldots, tp_n \rangle\ (tp_i \in Tipo))\} \tag{3.21}$$

A seqüência de tipos de interesse *INT* criada através da equação 3.21 servirá como base para guiar quais tipos de elementos geográficos serão avaliados no mapa de entrada, bem como, identificar elementos que ainda não existam no mapa de entrada e que sejam passiveis de surgir ao longo do processo a ser executado, e assim, configurar nos resultados do problema.

Para um melhor entendimento do contexto que o parágrafo anterior deseja transmitir, considere como exemplo um mapa de entrada caracterizado predominantemente como uma floresta e que, em razão de um verão prolongado, passa então por um período de secas e começa a adquirir características desérticas. Em tal situação, é sensato admitir que a saída gerada pela equação 3.21 como elementos interessantes de serem avaliados para os parâmetros de entrada $pred = floresta$ e $clima = verão$ seja dada como $INT = \langle areia, grama, água \rangle$.

Perceba ainda que, caso o cenário do mapa de origem representasse uma região de clima frio cujo elemento predominante fosse neve e ainda assim fosse sujeita a um processo de desertificação em razão caracterizado no parágrafo anterior, o mapeamento da equação 3.21 para os parâmetros de entrada dados por $pred = neve$ e $clima = verão$ seria representado por $INT = \langle areia, grama, neve, água, gelo \rangle$.

Uma vez que é a seqüência *INT* deve guiar a manipulação dos elementos geográficos do mapa de entrada, espera-se que, ao final do processo, o conjunto de elementos devolvido como resultado para o problema seja constituído por tipos geográficos definidos em *INT*. Este primeiro critério de admissibilidade para os resultados, está expresso na equação 3.22 que utiliza $\overline{\upsilon}$ para indicar a representação de uma célula para os resultados do problema para o mapa de jogo de entrada, e que assume ainda um possível resultado $r \in R$ tal que $r = \{\overline{\upsilon}_1, \ldots, \overline{\upsilon}_n\}$, e A como um subconjunto de R que compreende os elementos de R que são resultados admissíveis para o problema.

$$\overline{\Upsilon}_1 = \{\ \forall r = \{\overline{\upsilon}_1,\ldots,\overline{\upsilon}_n\} \in A\ \forall i \in \{1,2,\ldots,n\}\ \forall \overline{\upsilon}_i = \{x_i,y_i,z_i,tl_i\} \in \{\overline{\upsilon}_1,\ldots,\overline{\upsilon}_n\} \\ \exists_1 dsc = (tl,tp,in,fl) \in DscTile |\ (tl_i = tl) \wedge (tp \in INT)\} \tag{3.22}$$

O último critério de admissão definido para o problema está ligado a afirmação feita na seção 1.2 que diz que ao menos 75% da área total do mapa de entrada deve ser preservada no ato da conclusão do processo. Apesar de tal critério poder ser visto como de entendimento aparentemente direto, em razão de como se dá a constituição dos mapas sendo trabalhados nesta dissertação, alguns cuidados devem ser tomados acerca dessa afirmação.

Neste capítulo, foi afirmado na seção 3.1.1 que este trabalho trata apenas de mapas de formato retangular de comprimento w_{mapa} e largura h_{mapa} e de profundidade l_{mapa}, cuja área é definida como $a = (w_{mapa} \times h_{mapa} \times l_{mapa})$. Um valor que, intuitivamente, corresponde ao total de células do mapa de jogo.

Contudo, apenas os elementos geográficos, apresentados neste trabalho na qualidade de *tiles*, são pertinentes de serem alterados pelos processos que serão descritos no próximo capítulo deste trabalho. Logo, uma vez que a pode ser visto como o somatório da área de cada camada que constitui o mapa de jogo, e que, nem todas as camadas do mapa são destinadas a alocação de *tiles*, trabalhar em cima do valor oferecido por a não se mostra como uma alternativa interessante.

Não o bastante, uma breve observação visual acerca da configuração das camadas do mapa de jogo é o suficiente para que se evidencie que a camada definida para a profundidade zero é, para a grande maioria dos casos, a única camada que possui todas as suas células ocupadas. Desta forma, é possível perceber que, para muitos casos, é bem provável que nem mesmo o mapa originalmente fornecido como entrada para o problema ocupe 75% do total de células disponíveis.

Por esta razão, uma alternativa vista aqui como mais interessante foi identificar o total de células do mapa de entrada que possuem tiles vinculados a elas e utilizar este valor como referencia para o cálculo a ser realizado pelo critério de admissão. A lógica utilizada para esta abordagem é definida pelas equações 3.23 e 3.24.

$$area(d = \{\upsilon_1,\ldots,\upsilon_n\}) = \left\{ area |\ (area \in \mathbb{N}) \wedge (area = \sum_{i=1}^{n} umTile(\upsilon_i)) \right\} \tag{3.23}$$

$$umTile(\upsilon = (x,y,z,obj,pred)) = \begin{cases} 1 & \text{se } obj \in Tile \\ 0 & \text{caso contrário} \end{cases} \tag{3.24}$$

A preocupação demonstrada pelos parágrafos anteriores com a correta medição da área real

ocupada pelo mapa de entrada não é necessária com relação à área ocupada pelo conjunto de células gerado como saída do problema, já que o próprio domínio do conjunto R definido pela equação 3.9 estabelece que os objetos do conjunto *Tile* são o único tipo de objeto capaz ser vinculado a estas células. Por esta razão, é possível utilizar a própria cardinalidade de um elemento deste conjunto como referência para os cálculos da área que ocupa. Logo, o critério de admissão que mantém a área original do mapa de jogo dentro de seu limite mínimo de preservação corresponde aquele apresentado pela equação 3.25.

$$\overline{\Upsilon}_2 = \{\forall r = \{\overline{\upsilon}_1, \ldots, \overline{\upsilon}_n\} \in A |\ (n \in \mathbb{N}) \wedge ((n/area) \leq 0{,}25)\} \tag{3.25}$$

Agora que todas estas informações foram estabelecidas, é possível afirmar que o conjunto de admissibilidade A que define os resultados cabíveis do problema sendo trabalhado, está contido no domínio definido por D^R se e somente se um dado elemento $r = \{\overline{\upsilon}_1, \ldots, \overline{\upsilon}_n\} \in A$ satisfaz as restrições impostas nas equações 3.22 e 3.25. Esta condição está expressa na equação 3.26.

$$A \subseteq D^R \leftrightarrow (\overline{\Upsilon}_1 \wedge \overline{\Upsilon}_2) \tag{3.26}$$

Por fim, a equação 3.26 permite definir que um elemento $r = \{\overline{\upsilon}_1, \ldots, \overline{\upsilon}_n\}$ é um resultado que satisfaz o problema apresentado, se para uma função f aplicável a um elemento d do conjunto de entradas D e a um elemento r pertencente ao subconjunto de resultados A é satisfeita a equação 3.27.

$$\forall r \in D^R((r \Vdash \Gamma) \leftrightarrow (\forall f \in d((f, r(f)) \in i) \wedge (r \in A))) \tag{3.27}$$

Capítulo 4

Algoritmos genéticos

Este capítulo irá apresentar como se deu a aplicação do modelo de algoritmo genético baseado em tipos abstratos de dados, ou simplesmente GAADT, na solução do problema que foi descrito na seção 1.2 e formalmente definido pela seção 3.3.

Neste sentido, este capítulo se inicia realizando uma breve discussão sobre o contexto geral do uso de algoritmos genéticos em jogos eletrônicos, procurando assim descrever os tipos de situação em que tais algoritmos costumam ser empregados.

Uma vez que a seção 4.1 tenha concluído essa discussão, a seção 4.2 apresentará os conceitos específicos do GAADT, procurando destacar em que se diferencia do algoritmo genético tradicional que foi inicialmente proposto por Holland. Por último, a seção 4.3 irá fazer a correlação entre o modelo de algoritmo do GAADT e o problema abordado por esta dissertação.

4.1 O uso dos algoritmos genéticos nos jogos eletrônicos

A técnica de algoritmo genético procura utilizar o princípio evolucionário encontrado na natureza para encontrar soluções para problemas algorítmicos, tomando como base para isto, o modelo natural da sobrevivência do mais apto. Por esta razão, são categorizados dentro do contexto da inteligência artificial como algoritmos de busca heurística que se utilizam ainda de uma abordagem de metáforas entre os conceitos da biologia e dos algoritmos de busca (Carvalho, 2008).

Sua aplicação em um contexto de jogos eletrônicos é justificada pelo fato de que, durante o processo de criação de um jogo, pode-se esbarrar em problemas à implementação da inteligência artificial que vão contra o senso comum, quer seja por sua dificuldade computacional, ou mesmo a simples falta de tempo para alcançar uma solução da forma convencional, a qual seria o processo de tentativa e erro. Estes problemas costumam apresentar um grande número de resultados possíveis, podendo também haver um grande número de variáveis de entrada que devem ser consideradas durante a identificação de um resultado (Carvalho, 2008).

Um exemplo deste caso seria o jogo *Gran Turismo 4*, com mais de 500 veículos além de

dúzias de dispositivos referentes a diferentes peças do sistema de direção de cada veículo sendo simulado. Por este exemplo, fica clara a tarefa dispendiosa que seria realizar iterativamente um ajuste manual dos parâmetros para a simulação da física envolvida no desempenho de cada um destes veículos, tanto de um ponto de vista humano (no que diz respeito à realização desta atividade), como da perspectiva de um cronograma de produção (Schwab, 2004).

O exemplo acima permite ainda intuir o grande espaço amostral que existe para se buscar valores adequados de se atribuir aos diferentes dispositivos de um veículo sendo simulado, e que estes valores, não são necessariamente os mesmos para todos os dispositivos de um veículo, ou mesmo, para os mesmos dispositivos em um veículo diferente.

É dentro deste tipo de perspectiva que se encontra o problema sendo abordado por esta dissertação (que foi apresentado pelas seções 1.2 e 3.3), uma vez que, tal como o exemplo mostrado, caracteriza-se por um grande espaço amostral que oferece ainda diferentes respostas possíveis. Um conceito que deve ser mantido em mente pelo restante deste capítulo, que irá agora apresentar o modelo de algoritmo genético que foi utilizado como solução para o problema abordado.

4.2 O algoritmo genético baseado em tipos abstratos de dados (GAADT)

Como mencionado no capítulo 1 desta dissertação, a escolha do modelo GAADT em lugar do algoritmo genético tradicional criado por Holland pode ser explicada por meio de um estudo do algoritmo que foi proposto por Holland e das modificações posteriormente sugeridas a ele. O mesmo capítulo também afirma que o GAADT difere do modelo tradicional, pela forma como expressa as metáforas percebidas por Holland entre a evolução das espécies e os sistemas adaptativos complexos (Lopes, 2003).

A maneira como o GAADT aborda tais metáforas pode ser definida em maiores detalhes por meio dos seguintes princípios:

- Qualquer problema tem uma função de codificação que mapeia todos os seus possíveis resultados na forma da representação adotada por seus cromossomos;
- Todo cromossomo possui uma função de adaptação que indica o quão adaptado ele é em relação a seu ambiente;
- Os cromossomos mais adaptados a seu ambiente geram descendentes em resposta a ação dos operadores genéticos. Tais cromossomos são passíveis de herdar de seus ancestrais aquelas características que se julgue mais adequadas ao ambiente;
- Os cromossomos menos adaptados possuem uma tendência a desaparecer.

O primeiro dos princípios listados acima remonta a um debate dentro da comunidade de algoritmos evolucionários sobre a melhor forma de representar o cromossomo, se como um vetor de tamanho fixo ou variável, e também, se como um alfabeto binário ou um não-binário (Goldberg et al., 1989) (Antonisse, 1989). Em respeito a esta disputa, o que a experiência tem demonstrado até então é que, para cada tipo de problema abordado, o mais provável é que um tipo de representação se sobressaia sobre o outro como uma melhor escolha (Sastry & Goldberg, 2002) (Schaffer, 1985).

Por esta razão, ao contrário de propor um novo modelo de algoritmo genético que esteja amarrado a uma representação específica, preferiu-se adotar o conceito de tipos abstratos de dados, adotando também uma representação em três níveis de percepção, a qual divide o cromossomo em três elementos fundamentais, as bases, os genes e, finalmente, os cromossomos. Esta concepção permite ao GAADT modelar a representação de um cromossomo de maneira que, tal representação, não esteja presa a qualquer problema em particular e nem a solução para este problema. O conceito estabelecido por esta característica permite ao GAADT reproduzir qualquer representação de cromossomo existente, dependendo unicamente da interpretação dada aos tipos cromossomo, gene e base (Lopes, 2003).

O segundo princípio estabelece que a função de adaptação de um cromossomo deve ser compatível com seu ambiente. Este contexto sugere que se o ambiente mudar, a função de adaptação deve também mudar e redirecionar o processo de busca de maneira a procurar pelo cromossomo que melhor satisfaça o ambiente em sua forma atual.

Para atender a este princípio, o GAADT possui a característica de ser um modelo de algoritmo evolucionário trabalhando sobre um ambiente que é definido como uma estrutura que possui a população como um de seus componentes. Não o bastante, o algoritmo interpreta a ocorrência de mudanças ambientais como marcos que determinam o início de um novo período evolucionário, de maneira que a cada novo período, a população de cromossomos é sujeita a ação dos operadores genéticos. Esta abordagem garante que cada nova população construída será composta exclusivamente por cromossomos que satisfaçam os atuais requerimentos de seu ambiente.

Uma vez que o período evolutivo chegue ao fim, ocorre o período de estagnação. Por sua vez, o algoritmo interpreta que um período de estagnação chega a um fim sempre que se detecta uma nova mudança ambiental, o que em troca, garante o início de um novo período evolutivo. Desta forma, é possível caracterizar o algoritmo como operando em respeito a um ciclo evolução-estagnação.

O resultado que o algoritmo gera para o problema será dado sempre em respeito à forma atual do ambiente e ocorrerá na forma do cromossomo mais adaptado da população de estagnação que foi alcançada. Para isto, é necessário que esta população atenda a um dado critério de parada que é estabelecido durante a instanciação do GAADT para o problema sendo trabalhado. Tal critério se faz necessário em razão de que, o GAADT, como modelo de algoritmo, não possui a intenção de imitar a natureza (uma vez que a mesma possui toda a eternidade para

evoluir e progredir), mas sim, a intenção de fornecer soluções para problemas bem formulados (Lopes, 2003).

O terceiro princípio está ligado à transmissão de características a uma nova geração. Em especial, o contexto que se quer transmitir através deste princípio é o de que, os cromossomos selecionados como pais que darão origem a novos cromossomos, transmitirão a seus descendentes, dentre as características que possuem, aquelas que são consideradas adaptadas à atual configuração do ambiente. A importância da definição deste princípio se deve ao fato de que os operadores de cruzamento encontrados na literatura, costumam ocupar-se apenas determinar se as características presentes nos cromossomos filhos encontram-se ou não presentes em seus genitores, ignorando por completo o grau de relevância com que estas características contribuem para a adaptação de seus cromossomos ao ambiente em que se encontram, quer sejam *cromossomos-pais* ou *cromossomos-filhos*.

Em contraste, o operador de cruzamento do GAADT procura utilizar na construção de novos indivíduos apenas os chamados *genes-dominantes*, isto é, genes que contribuem para a adaptação de um cromossomo ao ambiente em que atualmente se encontra. Para tanto, faz-se uso de uma função que determina o grau de adaptação de um gene. Em outras palavras, isto significa que tal função se ocupa em definir um valor quantitativo para a contribuição de um determinado gene na adaptação de seu cromossomo em relação ao atual estado do ambiente em que ele está inserido (Lopes, 2003).

O último destes princípios está relacionado ao ato de impedir que cromossomos que não estejam adaptados a seu atual ambiente possam evoluir. Em função disto, tais cromossomos são removidos da população, privando-lhes assim de qualquer real possibilidade de evolução (Lopes, 2003). Para o problema sendo trabalhado aqui, os cromossomos com uma adaptação considerada insuficiente ao ambiente em que se encontram são submetidos à ação de um operador de mutação antes de serem eliminados. Desta forma, assegura-se a preservação de características que possam contribuir para a adaptação de um cromossomo a configuração em que o ambiente se encontre em uma geração futura.

Agora que as diferenças entre o modelo de algoritmo proposto por Holland e o modelo do GAADT foram devidamente apresentadas, é possível detalhar como se deu a aplicação do modelo do GAADT ao problema abordado por esta dissertação. Este será o foco da próxima seção deste capítulo.

4.3 Aplicação do GAADT ao problema

Nesta seção será feita a apresentação do procedimento adotado para a aplicação do modelo do GAADT ao problema da geração de mapas para um jogo MMORPG, utilizando os conceitos que foram descritos ao longo da seção 1.2 e formalmente definidos na seção 3.3.

Inicialmente, será feita aqui uma explanação dos conceitos e formalismos do GAADT, in-

cluindo seus tipos básicos e operadores genéticos. Estes elementos serão exemplificados por meio de sua correlação com o problema abordado por esta dissertação. Posteriormente, será feita a explicação do algoritmo como um todo.

4.3.1 Tipos básicos

Neste trabalho, os indivíduos de uma população são representados na forma de cromossomos. Por sua vez, cada um destes cromossomos é representado através do material genético que os constitui. Este material genético possui nas bases suas unidades elementares de formação.

Definição 4.3.1. Base: Uma base é a unidade genética elementar passível de compor o material genético dos cromossomos da população. Aqui, a base é um elemento $b = (x, y, z, tl, tp, in, fl)$ que equivale a uma célula localizada em qualquer posição x, y, z da malha do mapa de jogo e ocupada por um elemento geográfico, i.e. um *tile*, denotado por tl, que esteja devidamente identificado por um tipo de descrição de *tile*, tp, e por um índice de descrição in, possuindo ainda uma flag fl que indica se este elemento geográfico requer, ou não, a ocorrência de um ponto de colisão para a célula em que se encontra.

Perceba que os atributos x, y, z e tl da definição acima são elementos que pertencem a entrada D do problema desta dissertação, tal como foi definida pela equação 3.3, unidos dos atributos tp, in e fl do conjunto $DscTile$ apresentado na equação 3.15, que determinam a caracterização do elemento geográfico representado pelo *tile* indicado por tl (que também pertence ao conjunto $DscTile$).

Para o GAADT, todas as bases b utilizadas pelo algoritmo devem ter origem em um único conjunto de bases denominado B. Tendo em vista os parâmetros do elemento b apresentado na definição 4.3.1, os quais, são provenientes de dois conjuntos já definidos nesta dissertação, e pressupondo que exista uma função que mapeia os elementos de D em $DscTile$ (uma suposição suportada pela restrição que a equação 3.16 impõe sobre D e que estabelece uma relação entre os conjuntos D e $DscTile$) é possível definir um conjunto DC como o da equação 4.1 e utilizá-lo na composição do conjunto de bases B desta dissertação, representado na equação 4.2.

$$DC = \{(x, y, z, tl, tp, in, fl) | \quad (x \in X_{mapa}) \wedge (y \in Y_{mapa}) \wedge (z \in Z_{mapa}) \wedge (tl \in Tile) \wedge (tp \in Tipo) \wedge (in \in \mathbb{N}) \wedge (fl \in Boolean)\} \tag{4.1}$$

$$B = DC \cup B_{\lambda} \tag{4.2}$$

Dentre os conjuntos apresentados na equação 4.1 como delimitantes do domínio dos atributos do conjunto DC, apenas o conjunto Z_{mapa} não foi anteriormente definido e está expresso na equação 4.3. Esta equação utiliza de um valor $\tilde{l}_{mapa}$ para representar a diferença entre o total de

camadas do mapa de entrada (definido pelo valor l_{mapa} apresentado na seção 3.1.1 e novamente na seção 3.1.2) e o número de camadas do mapa de entrada que não se destinam a alocação de *tiles* (um valor que será chamado aqui de $\ddot{l}_{mapa}$). Ou seja, $\tilde{l}_{mapa} = l_{mapa} - \ddot{l}_{mapa}$.

$$Z_{mapa} = \{z | (z \in \mathbb{N}) \wedge (0 \leq z < \tilde{l}_{mapa})\} \tag{4.3}$$

A convenção que a equação 4.3 adota para estabelecer o limite para o valor das coordenadas z das bases do GAADT está de acordo com as restrições que as equações 3.13 e 3.14 impõem sobre os dados de entrada, e que exigem que as camadas do mapa que não se ocupem da representação de elementos geográficos estejam dispostas em níveis mais elevados, reservando-se os níveis de menor profundidade exclusivamente para a alocação dos elementos geográficos.

Definido o conjunto *DC* e os demais conjuntos que o compõem, resta ainda definir o conjunto B_λ. O GAADT trabalha com a suposição de que existe um único elemento b_λ, uma *base-inócua* cuja informação não interfere na identidade do gene no qual está inserido. Entretanto, para esta dissertação, se aceita a existência de mais de uma *base-inócua* em razão dos diferentes valores que podem ser atribuídos as coordenadas x, y e z de uma base.

Logo, uma *base-inócua* é caracterizada neste trabalho como qualquer base $b = (x, y, z, tl, tp, in, fl)$ com um tipo de descrição de *tile* (tp) que a caracterize como uma célula não ocupada do mapa. Na prática, isto pode tanto significar que a célula não possui um *tile* associado a ela, ou que, a imagem representando o elemento geográfico deste *tile* é desprovida de opacidade e, portanto, não apresenta nenhuma influência para a construção do cenário. A fim de simplificar as considerações a serem feitas ao longo desta seção, irá se considerar que o último destes dois casos é o único que ocorre para qualquer mapa de jogo dado como entrada para o algoritmo. Este tipo particular de descrição, chamado aqui de tp_λ, equivale ao valor "Vazio" da tabela 3.2. A representação formal do conjunto de *bases-inócuas* B_λ é apresentada na equação 4.4.

$$B_\lambda = \{b = (x, y, z, tl, tp, in, fl) \,|\, (tp = tp_\lambda)\} \tag{4.4}$$

A definição que esta dissertação adota para seus elementos de base termina por estipular que cada célula da malha do mapa de jogo é uma base em potencial. Por sua vez, os elementos de base serão agrupados em seqüências, para assim, formarem elementos mais complexos. Neste sentido, estas seqüências de base irão representar características do cenário como rios, casas e diversos outros elementos. Estas seqüências de bases são chamadas de genes.

Em razão de um gene representar uma determinada característica, as bases que compõem sua seqüência devem possuir certo nível de correlação, de maneira a atender a um propósito. Neste contexto, o que ocorre é que no ato da formação de um gene deve existir um conjunto de regras que defina critérios para sua construção e que valide a seqüência de bases que se pretende

utilizar.

Tal como a definição do conjunto base, a definição deste conjunto de regras também deve levar em consideração o tipo de solução pretendida, uma vez que as regras deste conjunto atuarão como *leis de formação* para as características expressas pelos genes. O GAADT representa este conjunto na forma de seus axiomas de formação de genes, abreviado como *AFG*.

O modelo do GAADT especifica que o conjunto de AFGs deve ser definido em razão da semântica adotada para o gene, juntamente, com o tipo de solução que se deseja alcançar para o problema em análise. Por esta razão, antes de se definir critérios para validar a composição de um gene, deve-se primeiro estabelecer um consenso em relação a esta composição.

Definição 4.3.2. Gene: um gene g é uma "microrregião" do mapa completamente contida em uma única camada e definido como uma seqüência de bases do tipo $g = \langle b_1, \ldots, b_n \rangle$ na qual todas as bases $b = (x, y, z, tl, tp, in, fl)$ da seqüência possuem os mesmos valores de tipo (tp) e índice (in) de descrição.

Esta definição de gene por si só já estabelece três critérios que deverão ser contemplados pelo AFG do problema. O primeiro deles, diz respeito à afirmação de que um gene está contido em uma única camada do mapa, o que é o mesmo que dizer que, de forma a ser uma seqüência válida, todas as bases b da seqüência devem possuir um mesmo valor para a componente de coordenada z. A definição também explicita que todas as bases da seqüência devem possuir os mesmos valores para os elementos tp e in. Estes critérios estão expressos nos axiomas das equações 4.5 e 4.6.

$$\begin{aligned} AFG_1 = \{\ & \forall g = \langle b_1, \ldots, b_n \rangle \in G \\ & \forall \left(b_\alpha = (x_\alpha, y_\alpha, z_\alpha, tl_\alpha, tp_\alpha, in_\alpha, fl_\alpha), b_\beta = (x_\beta, y_\beta, z_\beta, tl_\beta, tp_\beta, in_\beta, fl_\beta)\right) \\ & \in \{b_1, \ldots, b_n\} \mid \\ & z_\alpha = z_\beta \} \end{aligned} \tag{4.5}$$

$$\begin{aligned} AFG_2 = \{\ & \forall g = \langle b_1, \ldots, b_n \rangle \in G \\ & \forall \left(b_\alpha = (x_\alpha, y_\alpha, z_\alpha, tl_\alpha, tp_\alpha, in_\alpha, fl_\alpha), b_\beta = (x_\beta, y_\beta, z_\beta, tl_\beta, tp_\beta, in_\beta, fl_\beta)\right) \\ & \in \{b_1, \ldots, b_n\} \mid \\ & \left(tp_\alpha = tp_\beta\right) \wedge \left(in_\alpha = in_\beta\right) \} \end{aligned} \tag{4.6}$$

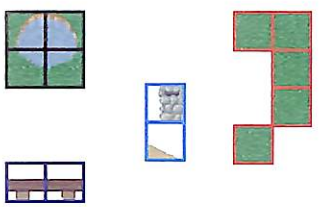
Além destes critérios, também se estipulou que a distância euclidiana entre a menor coordenada x e maior coordenada x da seqüência de bases, deve ser inferior a dois terços do comprimento total do mapa (w_{mapa}). De forma análoga, também se estipula que a distância euclidiana entre a menor e a maior coordenada y da seqüência de bases deve ser inferior a dois terços da largura do mapa (h_{mapa}). Estes critérios são expressos nos axiomas das equações 4.7 e 4.8.

$$
\begin{aligned}
AFG_3 = \{\ & \forall g = \langle b_1, \ldots, b_n \rangle \in G \\
& \exists_1 \left(b_{x^+} = (x_{x^+}, y_{x^+}, z_{x^+}, tl_{x^+}, tp_{x^+}, in_{x^+}, fl_{x^+}),\right. \\
& \quad \left. b_{x^-} = (x_{x^-}, y_{x^-}, z_{x^-}, tl_{x^-}, tp_{x^-}, in_{x^-}, fl_{x^-})\right) \in \{b_1, \ldots, b_n\} \\
& \forall b = (x, y, z, tl, tp, in, fl) \in \{b_1, \ldots, b_n\} \mid \\
& ((x \leq x_{x^+}) \wedge (x_{x^-} \leq x)) \wedge (0 < (x_{x^+} - x_{x^-} + 1) < (2 \times w_{mapa}/3))\}
\end{aligned}
\tag{4.7}
$$

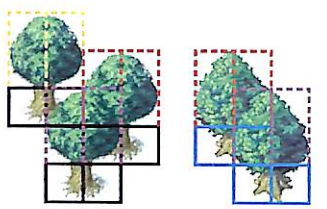
$$
\begin{aligned}
AFG_4 = \{\ & \forall g = \langle b_1, \ldots, b_n \rangle \in G \\
& \exists_1 \left(b_{y^+} = (x_{y^+}, y_{y^+}, z_{y^+}, tl_{y^+}, tp_{y^+}, in_{y^+}, fl_{y^+}),\right. \\
& \quad \left. b_{y^-} = (x_{y^-}, y_{y^-}, z_{y^-}, tl_{y^-}, tp_{y^-}, in_{y^-}, fl_{y^-})\right) \in \{b_1, \ldots, b_n\} \\
& \forall b = (x, y, z, tl, tp, in, fl) \in \{b_1, \ldots, b_n\} \mid \\
& \left((y \leq y_{y^+}) \wedge (y_{y^-} \leq y)\right) \wedge \left(0 < (y_{y^+} - y_{y^-} + 1) < (2 \times h_{mapa}/3)\right)\}
\end{aligned}
\tag{4.8}
$$

Logo, para a instância do modelo do GAADT empregada nesta dissertação, um gene é qualquer seqüência de bases $b = (x, y, z, tl, tp, in, fl)$ que satisfaça os critérios definidos por estes AFGs. Em razão disto, é correto afirmar que o conjunto de genes do problema equivale ao conjunto AFG. A Figura 4.1 ilustra alguns exemplos de genes criados a partir do conjunto de AFGs apresentado nesta dissertação, para um dado mapa fornecido como entrada para o algoritmo.

(a) *Genes compostos por um único elemento de base*



(b) *Genes compostos por mais de um elemento de base*



(c) *Genes com elementos que se estendem para fora dos pontos de alocação x e y*

Figura 4.1: *Exemplos de genes para o problema*

Os genes por sua vez, são agrupados em conjuntos de maneira a formar os cromossomos da população, o último dos três níveis de percepção do modelo do GAADT que foram anteriormente mencionados seção 4.2.

Partindo da lógica de que uma base é uma unidade elementar e um gene é uma característica composta por estas unidades, um cromossomo no papel de uma seqüência de genes é um agregado de características, que constitui um elemento que é único dentro de sua população ou de qualquer população futura.

Neste sentido, o que ocorre é que o conjunto de genes associado a um cromossomo possui a função de distingui-lo, concedendo-lhe uma identidade única que é utilizada de maneira a coibir

a existência de mais de uma cópia de um mesmo cromossomo dentro de uma população atual, ou o seu ressurgimento em uma população futura (Lopes, 2003).

As características que os genes associados a um cromossomo lhe concedem também permitem sua classificação em grupos taxonômicos (espécie e família) em razão do nível de similaridade de suas características com a de outros cromossomos. Deve-se contudo ser cuidadoso no ato da classificação, uma vez que no passado pesquisadores foram conduzidos a conclusão errônea de que quaisquer cromossomos com alto grau de similaridade genética pertencem a um mesmo grupo taxonômico (Gould, 1999).

O que se constatou de fato foi a existência de um "relógio biológico" que determina o início e o fim da formação de características distintas (quer sejam genéticas ou morfológicas), que ocorrem em estágios embrionários específicos para cada espécie (King & Wilson, 1975). Este relógio também é responsável por garantir que o cruzamento de dois cromossomos de uma mesma espécie resulte em um cromossomo pertencente a esta espécie (King & Wilson, 1975).

Este conceito é representado pelo GAADT através de seu conjunto de axiomas de formação de cromossomos, o *AFC*, que é o responsável por reger a validação das seqüências de genes com as quais se intenciona formar cromossomos. Esta contextualização concede ao AFC um papel similar ao do AFG uma vez que irá atuar como um conjunto de "leis de formação" para a composição dos cromossomos.

O conjunto de regras definido pelo AFC deve adequar-se a situação na qual está sendo empregado, levando em consideração, o tipo de problema abordado, a solução pretendida e a semântica que está sendo adotada para o cromossomo. Logo, antes de definir as regras que irão reger a formação dos cromossomos, é preciso definir qual a sua composição e a semântica dos critérios que serão impostos a eles.

Definição 4.3.3. Cromossomo: Um cromossomo é uma "macrorregião" do mapa distribuída ao longo de uma ou mais camadas do mapa e definido como um conjunto de genes $c = \{g_1, \ldots, g_n\}$ que obedece as regras estabelecidas pelo conjunto AFC.

Uma vez que esta definição determina que, para o problema, o cromossomo é um conjunto de genes compondo uma macrorregião do mapa, o algoritmo assume que um cromossomo não pode possuir dois elementos que correspondam a uma mesma microrregião ou mesmo a uma mesma célula do mapa de jogo. Este critério está definido no AFC da equação 4.9.

$$
\begin{aligned}
AFC_1 = \{\ & \forall\, c = \{g_1, g_2, \ldots, g_m\} \in C \\
& \forall \left(g_\alpha = \left\langle b_1^\alpha, \ldots, b_{n_1}^\alpha \right\rangle, g_\beta = \left\langle b_1^\beta, \ldots, b_{n_2}^\beta \right\rangle \right) \in \{g_1, g_2, \ldots, g_m\} \\
& \forall b_i = (x_i, y_i, z_i, tl_i, tp_i, in_i, fl_i) \in \left\{ b_1^\alpha, \ldots, b_{n_1}^\alpha \right\} \\
& \forall b_j = (x_j, y_j, z_j, tl_j, tp_j, in_j, fl_j) \in \left\{ b_1^\beta, \ldots, b_{n_2}^\beta \right\} | \\
& (x_i \neq x_j) \vee (y_i \neq y_j) \vee (z_i \neq z_j)\ \}
\end{aligned}
\tag{4.9}
$$

Além deste critério, foi também definido que qualquer cromossomo deve possuir ao menos um elemento na camada base do mapa. Em outras palavras, isto significa dizer que qualquer cromossomo deve possuir em sua composição ao menos um elemento cujas coordenadas espaciais para a componente z possuam valor igual zero. Este critério está definido no AFC da equação 4.10.

$$\begin{aligned} AFC_2 = \{\ & \forall\, c = \{g_1, g_2, \ldots, g_m\} \in C \\ & \forall (g = \langle b_1, \ldots, b_n \rangle) \in \{g_1, g_2, \ldots, g_m\} \\ & \exists b = (x, y, z, tl, tp, in, fl) \in \{b_1, \ldots, b_n\} \mid z = 0\ \} \end{aligned} \tag{4.10}$$

O critério da equação 4.10 visa assim determinar a existência de uma camada de origem, que delimita o início de um cromossomo com o propósito de garantir sua continuidade ao longo das camadas do mapa para as quais se estenda. Em razão disto, é necessário ainda estipular que para qualquer cromossomo possuindo genes que estejam localizados em uma camada de coordenada z maior que à da origem em zero, deve haver ao menos um gene que se localize em uma camada de coordenada $z-1$. Este último critério para a formação de um cromossomo está definido na equação 4.11.

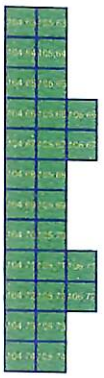
$$\begin{aligned} AFC_3 = \{\ & \forall\, c = \{g_1, g_2, \ldots, g_m\} \in C \\ & \forall \left(g_\alpha = \left\langle b_1^\alpha, \ldots, b_{n_1}^\alpha \right\rangle, g_\beta = \left\langle b_1^\beta, \ldots, b_{n_2}^\beta \right\rangle\right) \in \{g_1, g_2, \ldots, g_m\} \\ & \forall b_i = (x_i, y_i, z_i, tl_i, tp_i, in_i, fl_i) \in \left\{b_1^\alpha, \ldots, b_{n_1}^\alpha\right\} \\ & \forall b_j = (x_j, y_j, z_j, tl_j, tp_j, in_j, fl_j) \in \left\{b_1^\beta, \ldots, b_{n_2}^\beta\right\} \mid \\ & (z_i \geq z_j) \wedge \\ & (\forall z \in \{0, 1, \ldots, z_i\}\ \exists \left(g_\gamma = \langle b_1^\gamma, \ldots, b_{n_\gamma}^\gamma \rangle\right) \in \left(\{g_1, g_2, \ldots, g_m\} - \{g_\beta\}\right)) \wedge \\ & (\forall b = (x_b, y_b, z_b, tl_b, tp_b, in_b, fl_b) \in \left\{b_1^\gamma, \ldots, b_{n_\gamma}^\gamma\right\} (z_b = z))\ \} \end{aligned} \tag{4.11}$$

A instância do GAADT empregada nesta dissertação assume um cromossomo como qualquer conjunto de genes $c = \{g_1, \ldots, g_n\}$ que satisfaça os critérios definidos por estes AFCs. De forma análoga com o que ocorre com o conjunto AFG, é correto afirmar que o conjunto de cromossomos do problema equivale ao conjunto AFC.

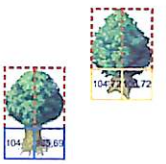
Cada cromossomo gerado pelo algoritmo representa um resultado em potencial para o problema abordado, a partir da ótica da solução pretendida. As Figuras 4.2 e 4.3 ilustram exemplos de cromossomos, utilizando do valor da coordenada z para indicar em qual camada do mapa cada gene destes cromossomos se encontra.

Ao se afirmar que um cromossomo representa um resultado potencial para o problema abordado, é possível intuir que a existência de diferentes cromossomos, cada um, constituindo um diferente resultado para o problema. Um conjunto de possíveis resultados representados na forma de cromossomos é o que se chama de uma população.

(a) *Cromossomo*

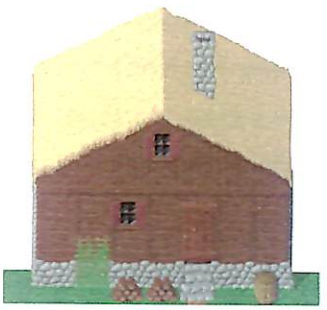


(b) *Gene de (a) em $z = 0$ e seus pontos x e y*

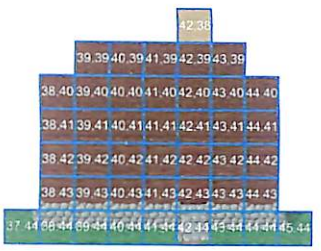


(c) *Genes de (a) em $z = 1$ e seus pontos x e y*

Figura 4.2: *Exemplo de um cromossomo com seus genes distríbuidos ao longo do eixo z*



(a) *Cromossomo*



(b) *Gene de (a) em $z = 0$ e seus pontos x e y*

(c) *Genes de (a) em $z = 1$ e seus pontos x e y*

(d) *Gene de (a) em $z = 2$ e seus pontos x e y*

(e) *Gene de (a) em $z = 3$ e seus pontos x e y*

(f) *Um cromossomo formado por um único gene (em $z = 0$) de um único elemento do tipo base*

Figura 4.3: *Outros dois exemplos de cromossomos*

Conceder a uma população uma representação de seus indivíduos na forma dos cromossomos que os compõem permite que uma identidade única seja atribuída a cada indivíduo, bem como, avaliá-los de maneira imparcial uma vez que, para o GAADT, o cromossomo de um indivíduo é um elemento que é único, que não se repete em qualquer população atual, ou tampouco, reaparece em uma população futura.

Definição 4.3.4. População: uma população é um conjunto P, formado exclusivamente por cromossomos que satisfaçam o conjunto de AFCs definido para o problema.

O conceito de população dado por esta definição permite intuir que a população, por ser composta por elementos do tipo cromossomo, possui uma complexidade maior que a do cro-

mossomo. Além disto, é possível também intuir que, dado o contexto que esta dissertação adota para os cromossomos do algoritmo, uma população pode ser caracterizada como um conjunto de macrorregiões do mapa de jogo, cada uma das quais, constitui um possível resultado para o problema abordado.

Seguindo o raciocínio do parágrafo anterior, a população inicial para o algoritmo é então vista como compreendendo o conjunto de todas as regiões que compõem o mapa de entrada original fornecido para o sistema.

Esta contextualização para o tipo população estabelece ainda que as restrições impostas pelo GAADT sobre suas populações impedem que, em qualquer momento do processo evolucionário simulado pelo algoritmo, sejam geradas duas regiões de mapa idênticas. Todavia, é permitida a geração de dois cromossomos que delimitem uma mesma região do mapa, desde que estes cromossomos possuam conjuntos de características diferentes.

Por fim, o GAADT entende ainda que a geração de uma população vazia se deve a uma suposição incorreta para a solução do problema, uma vez que isto seria o mesmo que afirmar que o problema proposto não possui solução.

4.3.2 Operadores genéticos

Os operadores genéticos são os responsáveis pela geração dos novos indivíduos de uma população. O modelo do GAADT estabelece dois tipos de operadores genéticos, o operador de reprodução e o operador de mutação. A prova dos procedimentos definidos por estes operadores não será apresentada por esta dissertação, podendo ser encontrada em Lopes (2003).

Estes operadores trabalham de maneira distinta. Neste sentido, o operador de reprodução é caracterizado por combinar os genes de dois cromossomos diferentes, os quais recebem então a denominação de *cromossomos-pais*, e gerar novos cromossomos para a população, os quais recebem o nome de *cromossomos-filhos*. Em contraste, a atuação do operador genético de mutação tem como característica a alteração da identidade de um cromossomo por meio da modificação de seus genes, criando assim um novo cromossomo chamado de *cromossomo-mutante*.

O GAADT assume que os genes provenientes dos *cromossomos-pais* utilizados na composição dos *cromossomos-filhos* são aqueles que melhor satisfazem o ambiente, isto é, são aqueles que, em razão das característica que expressam, melhor satisfazem as restrições que lhes são impostas pelo problema sendo abordado. Estes genes são conhecidos como *genes-dominantes*.

Deve ser frisado, que estes *genes-dominantes* não correspondem aqueles da biologia tradicional (Basile & Magalhaes, 1975) devido a grande diferença entre se afirmar que um gene g_1 melhor satisfaz as restrições de um problema do que um gene g_2, e entre se afirmar que o fator hereditário expresso por um gene g_1 supera o do gene g_2.

Desta forma, o que o modelo do GAADT realmente interpreta ao afirmar que um gene g_1 é dominante sobre um gene g_2 é que, dado que estes dois genes expressem uma mesma

característica de maneiras diferentes (i.e. com diferentes fenótipos), considera-se que o gene g_1 melhor satisfaz as restrições impostas pelo problema, se possuir um grau de adaptação maior ou igual ao do gene g_2.

O grau de adaptação do gene mencionado no parágrafo anterior corresponde a uma função definida pelo GAADT como *grau*, que possui a tarefa de avaliar o quão bem um determinado gene está adaptado a atual configuração de seu ambiente em vista das restrições que este ambiente lhe impõe.

A função *grau* deve ainda considerar a existência de um gene g_λ, que não expressa qualquer característica e que possui um grau de adaptação menor que o de qualquer outro gene. Por esta razão, a presença ou a ausência deste gene não possui influência sobre a identidade de um cromossomo. Este gene é conhecido como *gene-inócuo*. Nesta dissertação é aceita a existência de diferentes *genes-inócuos*, os quais pertencem ao conjunto expresso na equação 4.12, um conjunto que está incluso no conjunto de genes G do sistema.

$$G_\lambda = \left\{ g_\lambda = \left\langle b_1^{g_\lambda}, \ldots, b_n^{g_\lambda} \right\rangle \mid \forall\, i \in \{1,2,\ldots,n\} \left(b_i^{g_\lambda} \in B_\lambda \right) \right\} \tag{4.12}$$

Definição 4.3.5. Grau: o grau de adaptação de um gene é uma função *grau* de tipo $grau : G \rightarrow K$ que mapeia um gene a um valor k pertencente a um corpo ordenado[1] K, portanto, $k \in K$. Em conseqüência disto, k é então chamado de $grau(g)$ e reflete uma estratificação comparativa entre a adaptação dos genes, definida de acordo com a perspectiva da solução adotada ao problema proposto.

A função *grau* deve assumir que um *gene-inócuo* terá um grau de adaptação equivalente ao do elemento neutro do corpo ordenado K, cujo valor é tal que $\{\forall (g, g_\lambda) \in G | (g \neq g_\lambda) \wedge (grau(g) > grau(g_\lambda))\}$.

Nesta dissertação, o corpo ordenado K utilizado é convencionado como o conjunto dos números racionais. Conseqüentemente, uma vez que se assuma $K = \mathbb{Q}$ o tipo da função *grau* expresso na definição 4.3.5 pode ser escrito como $grau : G \rightarrow \mathbb{Q}$, o que implica em dizer que $k \in \mathbb{Q}$.

A função *grau* criada para esta dissertação utiliza de uma série de artifícios para o propósito de expressar um gene na forma de um valor coerente, uma metodologia que foi adotada em razão da complexidade de se avaliar conceitos tão abstratos como o quão bem uma seção do mapa de jogo se adéqua as seções a sua volta.

Para se falar do primeiro artifício utilizado aqui, é preciso relembrar que, ao longo da seção 3.3 foi apresentado pela equação 3.22 um critério de admissão imposto aos resultados do problema que, exige que os tipos de elementos geográficos apresentados pelos resultados sejam

[1] Corpo ordenado é uma estrutura algébrica, com duas operações, sem divisores próprios de zero e munido de uma ordem. Um exemplo típico é o conjunto $\langle \mathbb{Q}, \leq, +, \times, 0, 1 \rangle$. Para maiores informações consulte Winter (1974) e Roman (1995).

compatíveis com aqueles contidos em uma seqüência de tipos de interesse denominada *INT*, fornecida por meio da equação 3.21.

Com isto em mente, o que se afirma aqui é que, o primeiro destes artifícios está vinculado justamente à necessidade que o sistema possui de fornecer resultados compatíveis com esta restrição e que, portanto, o GAADT emprega em suas operações o uso da seqüência de tipos de interesse *INT* resultante da equação 3.21 para determinar quais elementos do mapa de jogo são interessantes de serem trabalhado pelo algoritmo.

Neste sentido, a ordem em que os elementos de *INT* se manifestam dentro da seqüência influi no cálculo do grau do gene, uma vez que, para esta dissertação, se entende que tal ordem denota a prioridade que estes elementos possuem entre si. Um conceito que pode ser mais bem entendido retomando-se o exemplo do mapa que, a partir de sua configuração original de floresta, passa a adquirir características desérticas. Um exemplo que utiliza de uma seqüência de tipos de interesse dada por $INT = \langle areia, grama, água \rangle$ e que foi apresentado pela primeira vez na página 53 da seção 3.3.

Para este caso, é possível imaginar que na tentativa de emular as transições que ocorreriam no mundo real em virtude de um aumento na temperatura desta região, o algoritmo inicia seus procedimentos com a diminuição das áreas representando água em favor daquelas representando grama. Seguindo este raciocínio, a grama viria eventualmente a morrer em razão do calor e da falta de água dando então lugar a areia.

A analogia do parágrafo anterior ilustra como o algoritmo procura realizar a substituição dos elementos geográficos do cenário, partindo do elemento que considera de menor prioridade (o último elemento da seqüência), para aquele que considera como o de maior prioridade (o primeiro elemento da seqüência).

Este contexto é utilizado durante a avaliação de um gene, que tem o valor de seu grau determinado em razão da característica da seqüência *INT* que está atualmente sendo considerada para o cálculo. Com este propósito, grupos de genes são separados e encaminhados um a um para o cálculo de seu grau, em razão de uma mesma característica sendo avaliada e que foi aleatoriamente escolhida em tempo de execução. Esta escolha é indicada atribuindo-se o valor da posição que esta característica ocupa em *INT* a um atributo denominado *passo*, definido por meio da função de tipo $INT \rightarrow \mathbb{N}$ que está expressa na equação 4.13.

$$random_2(INT) = \{passo | (passo \in \mathbb{N}) \wedge (0 < passo \leq \#INT)\} \tag{4.13}$$

Além de se conhecer a característica sendo avaliada é necessário também avaliar como esta característica se manifesta através de seu gene. Isto significa que o fenótipo da característica expressa por um gene também influi no cálculo de seu grau.

A avaliação do fenótipo de uma característica está vinculada a como ela é representada no mapa de jogo. Por esta razão é necessário "montar" a representação da característica. Esta

tarefa é realizada concatenando as diferentes imagens representadas nos *tiles* associados a cada elemento de base de um gene em uma única imagem.

A imagem resultante desta concatenação terá sua origem nas menores coordenadas x e y contidas no conjunto de bases do gene e irá se estender até as coordenadas x e y de maior valor contidas neste mesmo conjunto de bases. Aqui, não é necessário que se leve em consideração o valor da coordenada z, devido ao fato de todas as coordenadas z em um gene possuírem o mesmo valor, definindo o gene como contido em um plano (i.e. a camada). Além disto, a imagem sendo gerada é uma estrutura bidimensional que não comportaria uma terceira componente espacial.

Após a concatenação de todas as imagens contidas nas bases do gene em uma única imagem, a mesma é encaminhada a um módulo do sistema responsável por aplicar sobre ela um processo de detecção de bordas, que possui o propósito de delimitar os contornos da imagem de entrada e com isto, gerar uma imagem de saída composta unicamente por *pixels* brancos (bordas da imagem) e pretos (demais áreas). Este módulo que utiliza do algoritmo apresentado na seção 3.2.1 será discutido de forma mais detalhada na seção 5.2.2.

A partir da imagem de bordas gerada, o algoritmo procura identificar a correta disposição dos elementos ao longo da região delimitada por um gene, partindo do pressuposto que uma maior freqüência de *pixels* brancos está associada a uma maior irregularidade na disposição dos elementos representados na imagem. Como resultado, a função *grau* irá assumir que quanto menor a freqüência de *pixels* brancos, melhor o fenótipo de uma característica. A função responsável por calcular o total de *pixels* brancos para um gene está expressa na equação 4.14 por meio da função *totalBrancoGene*, uma função de tipo $G \rightarrow \mathbb{N}$.

$$totalBrancoGene(g = \langle b_1, b_2, \ldots, b_n \rangle) = \sum_{i=1}^{n} totalBrancoBase(b_i) \tag{4.14}$$

A equação 4.14 emprega em seus cálculos uma mecânica que determina o total de *pixels* brancos na imagem de bordas representando um gene, através do somatório do total de *pixels* brancos que existem na representação de cada base deste gene. Para tanto, a equação 4.14 utiliza ainda da função *totalBrancoBase*, que possui a tarefa de varrer os *pixels* da imagem de entrada que se encontram dentro da região delimitada pelas coordenadas x e y da célula do mapa ocupada por um dada base. A equação *totalBrancoBase* é definida como de tipo $B \rightarrow \mathbb{N}$ e está expressa na equação 4.15.

$$totalBrancoBase(b = \langle x, y, z, tl, tp, in, fl \rangle) = \sum_{i=x}^{x+w_{cel}} \sum_{j=y}^{y+in_{cel}} branco(i, j) \tag{4.15}$$

Por fim, a equação 4.15 utiliza ainda de uma função *branco* de tipo $\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ que retorna o valor um para qualquer valor x, y passado como parâmetro que equivalha a posição de um *pixel*

branco na imagem de bordas sendo trabalhada, ou um valor zero em caso contrário. Esta função está expressa na equação 4.16 que define ainda o uso de uma função $edges(x,y)$, responsável por mapear a coordenada x,y de entrada na imagem de bordas e retornar o *pixel* alocado para esta posição.

$$branco(x,y) = \begin{cases} 1 & \text{se } edges(x,y) = pixel \text{ branco} \\ 0 & \text{caso contrário} \end{cases} \tag{4.16}$$

Após a contagem do total de *pixels* brancos da imagem representando o gene, é possível avaliar o quão adequada é a representação da característica que este gene expressa por meio de seu fenótipo. Neste sentido, não basta apenas se ter o total de *pixels* brancos proporcionado pela equação 4.14. É necessário ainda avaliar a dimensão da área do mapa ocupada pelo fenótipo deste gene. Este procedimento é imprescindível em virtude da suposição feita pelo algoritmo de que, a representação de qualquer característica contida na seqüência de tipos de interesse INT, que não a primeira, deve ter sua manifestação reduzida em favor da característica que a antecede.

Este e outros aspectos do cálculo do grau do gene serão devidamente cobertos pela função *grau* uma vez que a fluidez dos elementos representados tenha sido avaliada. Este último aspecto de avaliação é coberto pela função $distri(g)$ de tipo $G \rightarrow \mathbb{Q}$ e definida pela equação 4.17, responsável por calcular um valor racional para o gene de entrada com base na fluidez de sua imagem de bordas (dada pela equação 4.14 como seu total de *pixels* brancos) em conjunto com a área total que este gene ocupa em número de *pixels* (dada pelo produto da cardinalidade do gene, $\#g$, com o comprimento e a largura padrão das células do mapa, w_{cel} e h_{cel}).

$$distri(g) = \begin{cases} 1 - \left(\frac{1}{\#g \times w_{cel} \times h_{cel}}\right) & \text{se } totalBrancoGene = 0 \\ 1 - \left(\frac{totalBrancoGene(g)}{\#g \times w_{cel} \times h_{cel}}\right) & \text{caso contrário} \end{cases} \tag{4.17}$$

Uma vez que a função $distri(g)$ se encarrega de avaliar o critério da fluidez da imagem representada por uma característica, aumentando seu valor para menores quocientes entre o total de *pixels* brancos e a área que o gene ocupa em *pixels*, cabe a função *grau* desta dissertação, estabelecer uma correlação entre a área ocupada pelo gene (dada por sua cardinalidade) e o que se deseja para a característica sendo analisada. Logo, a função de grau do gene deve atentar a diferentes operações em razão do valor do atributo *passo* sendo considerado para a avaliação, responsável por definir a característica em relação a qual o gene de entrada está sendo avaliado.

A função grau deve então atribuir a qualquer gene de entrada que possua a característica avaliada valores proporcionalmente maiores a área que o gene ocupa, se tal característica corresponde ao primeiro elemento da seqüência de tipos de interesse INT definida para o problema ($passo = 1$), ou em caso contrário, proporcionalmente menores a área que ocupam ($passo > 1$).

A função deve ainda atentar a não existência de elementos de *INT* no gene e ao caso particular dos *genes-inócuos*. Uma formalização para a função de grau do gene atendendo a estes critérios é definida pela equação 4.18 de tipo $G \rightarrow \mathbb{Q}$.

$$grau(g = \langle b_1, \ldots, b_n \rangle) = \begin{cases} \frac{distri(g)}{\#g} & \text{se } (passo > 1) \wedge \\ & (\exists_1 tp_{passo} \in INT \\ & \forall b = \langle x, y, z, tl, tp, in, fl \rangle \in \{b_1, \ldots, b_n\} \mid \\ & tp = tp_{passo}) \\ 1 - \frac{distri(g)}{\#g} & \text{se } (passo = 1) \wedge \\ & (\exists_1 tp_{passo} \in INT \\ & \forall b = \langle x, y, z, tl, tp, in, fl \rangle \in \{b_1, \ldots, b_n\} \mid \\ & tp = tp_{passo}) \\ 0 & \text{se } g \in G_\lambda \\ 1 \times 10^{-20} & \text{caso contrário} \end{cases} \tag{4.18}$$

A definição da função *grau* é essencial para se calcular a adaptação dos cromossomos da população, uma característica que se deve ao fato do cromossomo ser na verdade um conjunto de genes. Esta perspectiva permite o cálculo da adaptação de um cromossomo por meio da soma do *grau* de seus genes.

Definição 4.3.6. Adaptação: a adaptação de um cromossomo é dada por uma função *adapt* de tipo $adapt : C \rightarrow K$ que corresponde a soma do grau de seus genes. A função *adapt* é definida pela equação 4.19.

$$adapt(c = \{g_1, \ldots, g_n\}) = \sum_{i=1}^{n} (\Theta(c, g_i) \times grau(g_i)) \tag{4.19}$$

O somatório representado na equação 4.19 varia em função de valores i e n pertencentes ao conjunto dos números naturais, assumindo $n = \#g$ e com o valor de i sendo incrementado a cada iteração tal que, $0 < i \leq n$. A equação também utiliza de uma função Θ_{c,g_i} que determina o peso com que um gene i do cromossomo contribui para sua adaptação.

O modelo do GAADT afirma que não é possível realizar qualquer asserção acerca do parâmetro peso antes que se conheça o problema a que o modelo está sendo aplicado. Por esta razão, até que alguma consideração tenha sido feito acerca do problema, o GAADT atribui a função *adapt* uma representação genérica que a define como o mapeamento de um gene e um cromossomo na forma de um valor real racional não negativo. Portanto, $\Theta : C \times G \rightarrow \mathbb{Q}_+$.

Para o problema desta dissertação, a função peso é definida em razão de seu gene de entrada, utilizando como critério o fato deste gene possuir, ou não, um dos tipos de descrição de interesse

contidos na seqüência *INT* anteriormente definida pela equação 3.21. Desta forma, a função mapeia o gene de um cromossomo como um valor natural, implicando assim na redefinição do tipo da função de maneira que $\Theta : C \times G \rightarrow \mathbb{N}$. A função que determina o peso de um gene para o cromossomo é definida pela equação 4.20.

$$\Theta(c, g = \langle b_1, \ldots, b_n \rangle) = \begin{cases} 1 & \text{se } \forall b = \langle x, y, z, tl, tp, in, fl \rangle \in \{b_1, \ldots, b_n\} \mid tp \in INT \\ 0 & \text{caso contrário} \end{cases} \tag{4.20}$$

Note que a definição para a função peso concedida pela equação 4.20 permite intuir que os genes de um cromossomo, que não possuam associados a suas bases tipos de descrições de *tiles* que pertençam a seqüência de tipos de interesse, dada por *INT*, não terão influência direta no cálculo da adaptação do cromossomo, uma vez que qualquer valor associado ao grau de tais genes será anulado no ato da multiplicação por um peso zero.

A definição da função de adaptação de um cromossomo pela equação 4.19 permite ainda trabalhar o conceito de adaptação média de uma população. Um conceito que define que é possível calcular um valor médio para a adaptação de todos os cromossomos contidos em qualquer dada população *P*.

Definição 4.3.7. Adaptação média: a adaptação média de uma população, dada por $adapt_m$, é expressa como uma função de tipo $adapt_m : \wp(C) \rightarrow K$, cujo cálculo é demonstrado pela equação 4.21.

$$adapt_m(P = \{c_1, \ldots, c_n\}) = \frac{\sum_{i=1}^{n} adapt(c_i)}{\#P} \tag{4.21}$$

Em conjunto, os valores qualitativos definidos pelas funções *grau*, *adapt* e $adapt_m$ permitem auxiliar a criação e qualificação de cromossomos em uma população, de forma a decidir quais dentre estes cromossomos devem ser mantidos em uma próxima população.

Todavia, a real geração de novos cromossomos é um processo pertinente aos operadores genéticos de mutação e reprodução, sendo que o GAADT trabalha a reprodução dos cromossomos de suas populações unicamente por meio de um operador de cruzamento. Os procedimentos dos operadores de mutação e cruzamento são auxiliados ainda por um outro procedimento, que possui a tarefa de identificar os genes dominantes de um cromossomo por meio de uma função denominada *domi*.

A função *domi* recebe um par de genes de entrada, cada gene proveniente de um cromossomo diferente, e retorna como resultado o gene de maior grau de adaptação. Esta situação ocorre para os casos em que os dois genes fornecidos com entrada expressam uma mesma característica. Caso contrário, esta função retorna um *gene-inócuo* g_λ. Para o GAADT, dois genes expressam uma mesma característica se existe para o problema um atributo considerado como

relevante que é satisfeito por ambos os genes.

A função do atributo relevante é estabelecida pelo GAADT como uma função de tipo $atribRel : \wp(G)$, e é definida para o problema desta dissertação por meio da equação 4.22.

$$\begin{aligned} atribRel = \{(G_1, tp_i) | \quad & (G_1 \subseteq G) \wedge (tp_i \in INT) \wedge \\ & (\forall g = \langle b_1, \ldots, b_n \rangle \in G_1 \\ & \forall b = \langle x, y, z, tl, tp, in, fl \rangle \in \{b_1, \ldots, b_n\} \, (tp = tp_i)) \, \} \end{aligned} \tag{4.22}$$

O estabelecimento da relação do gene dominante depende ainda de uma relação *mesma* que determina se um gene g_1 apresenta o mesmo tipo de característica identificada em um gene g_2. Esta relação de tipo $mesma : \wp(G \times G)$ está representada na equação 4.23.

$$mesma = \{(G_1, G_2) \mid \forall (G_1, G_2) \subseteq G \, \forall ((G_1, tp_1), (G_2, tp_2)) \in atribRel \, (tp_1 = tp_2)\} \tag{4.23}$$

De posse desta equação é possível estabelecer uma função que defina formalmente a operação do gene dominante.

Definição 4.3.8. Dominante: A função do gene dominante é uma função *domi* de tipo $domi : G \times G \rightarrow G$ e especificada pela equação 4.24.

$$domi(g_1, g_2) = \begin{cases} g_\lambda & se \quad (g_1, g_2) \notin mesma \\ g_1 & se \quad (g_1, g_2) \in mesma \wedge (grau(g_1) \geq grau(g_2)) \\ g_2 & se \quad (g_1, g_2) \in mesma \wedge (grau(g_1) < grau(g_2)) \end{cases} \tag{4.24}$$

A definição da função do gene dominante é importante para os procedimentos que serão realizados pelo operador de cruzamento, o qual, irá receber dois *cromossomos-pais* previamente identificados pelo algoritmo como aptos a cruzarem e devolverá como resultado, uma população de cromossomos composta exclusivamente pelos genes dominantes destes *cromossomos-pais*.

Este raciocínio estabelece a necessidade de uma operação que selecione cromossomos em uma população atual, que sejam aptos a cruzar entre si. É possível obter este tipo de população criando uma sub-população de cromossomos, a partir de uma população atual, para a qual todos os cromossomos atendam a um determinado predicado *r*.

Definição 4.3.9. Seleção: a seleção é uma operação que seleciona em uma população atual cromossomos que satisfaçam a um determinado predicado *r* pertencente ao conjunto de requisitos *Rq* do ambiente, $r \in Rq$. Esta operação é definida como $sel(P_1, r) = P_1 \cap r$.

Esta dissertação realiza a seleção dos *cromossomos-pais* adotando uma metodologia em que, dada uma população de cromossomos de entrada para a operação de seleção, a população P_2

a ser submetida ao operador de cruzamento será composta unicamente por cromossomos que atendam o requisito r estabelecido pela equação 4.25.

$$\begin{aligned} r = \{P_2 |\ & (P_2 \subseteq C) \wedge (c = \{g_1, \ldots, g_n\} \in P_2) \leftrightarrow \\ & (\forall b = (x, y, z, tl, tp, in, fl) \in \{g_1, \ldots, g_n\}\ \exists tp^r \in INT_r (tp = tp^r)) \} \end{aligned} \tag{4.25}$$

A definição do requisito r apresentada na equação 4.25 utiliza ainda de uma subseqüência de INT denominada INT_r. Esta subseqüência é definida em tempo de execução de acordo com o que é expresso pela equação 4.26.

$$\begin{aligned} INT_r = \{\langle tp_1^r, \ldots, tp_n^r \rangle |\ & \forall i \in \{1,2,\ldots,n\}\ \forall j \in \{1,2,\ldots,m\}\ \exists tp_j \in INT \\ & (i \leq j) \wedge (j = i + (m-n)) \wedge (tp_i^r = tp_j) \} \end{aligned} \tag{4.26}$$

Uma vez que o operador de seleção tenha isolado da população atual aqueles cromossomos que julgue aptos a geraram novos cromossomos, tal geração irá ocorrer por meio de uma função de fecundação definida pelo GAADT como fec. Esta função criará novos indivíduos unicamente com os genes dominantes de seus *cromossomos-pais*.

Definição 4.3.10. Fecundação: A fecundação é uma função *fec* do tipo $fec : C \times C \rightarrow \wp(G)$, cujo conjunto de genes resultante é composto unicamente pelos genes dominantes dos cromossomos de entrada.

Para esta dissertação, a função de fecundação é direta de sua definição para um dado par de cromossomos da população e está expressa na equação 4.27.

$$\begin{aligned} fec(c^M, c^F) = \{\{g_1, \ldots, g_m\} |\ & (c^M = \{g_1^M, \ldots, g_l^M\} \in MACHO) \wedge \\ & (c^F = \{g_1^F, \ldots, g_n^F\} \in FEMEA) \wedge \\ & (g \subseteq \{g_1^M, \ldots, g_l^M\} \cup \{g_1^F, \ldots, g_n^F\}) \leftrightarrow \\ & (\forall g^M \in \{g_1^M, \ldots, g_l^M\}\ \forall g^F \in \{g_1^F, \ldots, g_n^F\} \\ & (domi(g^M, g^F) = g)) \} \end{aligned} \tag{4.27}$$

A equação 4.27 ilustra que, para esta dissertação, o conjunto de genes resultante da operação será composto por elementos pertencentes a união dos conjuntos de genes dos cromossomos de entrada, que satisfaçam a condição de gene dominante. A equação determina ainda que para cada par de cromossomos recebidos, um deve fazer parte da população $MACHO$, e o outro, parte da população $FEMEA$.

As populações $MACHO$ e $FEMEA$ são construídas a partir da população P_2 resultante da aplicação do predicado r da equação 4.25 sobre a população atual. Para isto, novos predicados do conjunto de requisitos Rq do ambiente são aplicados sobre P_2. Logo, $MACHO = sel(P_2; M)$

e $FEMEA = sel(P_2;F)$, com $M \in Rq$ e $F \in Rq$.

O modelo do GAADT estabelece ainda que, de acordo com a especificação adotada para a solução do problema, novos cromossomos podem ser criados por processos de reprodução sexuada ($M \cap F = \varnothing$), assexuada ($M = F$) ou mista ($M \cap F \neq \varnothing$). Esta dissertação utiliza apenas de reproduções por processos sexuados.

Para o problema desta dissertação, a definição dos predicados M e F parte da definição do predicado r, expresso aqui pela equação 4.25, que define a população P_2 sobre a qual estes dois novos predicados serão aplicados. Além disso, também será feito uso da seqüência INT_r definida em 4.26, de maneira que os cromossomos da população $MACHO$ serão todos os cromossomos de P_2 que possuam ao menos um gene com o tipo de descrição tp definido pelo primeiro elemento de INT_r.

De forma similar, os cromossomos do conjunto $FEMEA$ serão todos os cromossomos de P_2 que contenham ao menos um gene cujos elementos de base possuam um tipo de descrição que pertença a seqüência INT_r e que já não pertença ao conjunto de cromossomos da população $MACHO$. Os predicados M e F são expressos respectivamente pelas equações 4.28 e 4.29.

$$\begin{aligned} M = \{MACHO| \quad & (MACHO \subseteq P_2) \wedge (c = \{g_1,\ldots,g_n\} \in MACHO) \leftrightarrow \\ & (\exists g = \langle b_1,\ldots,b_m\rangle \in \{g_1,\ldots,g_n\}) \wedge \\ & (\forall b = (x,y,z,tl,tp,in,fl) \in \{b_1,\ldots,b_m\} \\ & \exists tp \in (INT_r = \langle tp_1^r,\ldots,tp_l^r\rangle)) \wedge (tp = tp_1^r)\ \} \end{aligned} \tag{4.28}$$

$$\begin{aligned} F = \{FEMEA| \quad & (FEMEA \subseteq P_2) \wedge (c = \{g_1,\ldots,g_l\} \in FEMEA) \leftrightarrow \\ & (c \notin MACHO) \wedge (\exists g^F = \langle b_1,\ldots,b_m\rangle \in \{g_1,\ldots,g_l\}) \wedge \\ & (\forall b = (x,y,z,tl,tp,in,fl) \in \{b_1,\ldots,b_m\}\ \exists tp \in INT_r) \wedge \\ & (\forall c^M = \{g_1,\ldots,g_n\} \in MACHO \\ & \exists g^M \in \{g_1,\ldots,g_n\}\,(vizinhos(g^M,g^F) = \text{verdadeiro}))\ \} \end{aligned} \tag{4.29}$$

A equação 4.29 utiliza ainda de uma restrição imposta por uma função chamada *vizinhos*. Esta função de tipo $G \times G \rightarrow Booleano$ recebe um par de genes como entrada e procura identificar se existe ao menos um ponto de contanto comum a estes dois genes, utilizando para isto das coordenadas x e y contidas em suas bases.

Aqui, o que se está chamando de ponto de contato é a ocorrência de uma aresta ou um vértice que seja compartilhado por dois genes que delimitam diferentes regiões do mapa de jogo.

Também se caracteriza como a existência de um ponto de contato, a possibilidade de se traçar uma perpendicular que conecte dois genes diferentes. Esta última situação garante que dois genes representando regiões sobrepostas do mapa, em razão de diferentes coordenadas z, sejam considerados como genes com ao menos um ponto de contato.

A função *vizinhos*, responsável por avaliar a ocorrência de tais circunstâncias, está definida

na equação 4.30.

$$vizinhos(g_1, g_2) \begin{cases} \text{Verdadeiro} & \text{se } (pontos(g_2) \cap pontos^+(g_1)) \neq \varnothing \\ \text{Falso} & \text{caso contrário} \end{cases} \tag{4.30}$$

A equação 4.30 faz uso de duas funções denominadas *pontos* e $pontos^+$. A primeira destas funções trabalha simplesmente por coletar as coordenadas x e y contidas nas bases de um gene e criar um único conjunto de pontos a partir delas. A segunda função estende o conceito da primeira, gerando um conjunto de pontos que inclui não apenas os pontos das bases do gene de entrada, como também os pontos criados pelas operações de soma e subtração de valores a cada par de coordenadas.

Para a última destas duas equações, o valor acrescido a coordenada x da base estará sempre entre menos um e mais um, enquanto que o valor acrescido a coordenada y irá oscilar entre o valor negativo do quociente da altura do *tile* contido na base do gene pela altura padrão de um *tile* do mapa (altura das células) e o valor mais um.

Esta mecânica para a função $pontos^+$ visa uma varredura de toda área do mapa que possa ser afetada por um *tile* cuja imagem se propague para além dos limites da célula a que está vinculado. Tal verificação não se faz necessária para o comprimento da imagem que será sempre igual ao comprimento das células do mapa. As funções *pontos* e $pontos^+$ estão definidas respectivamente nas equações 4.31 e 4.32, sendo ambas do tipo $G \rightarrow \mathbb{N}$. Além disso, a Figura 4.4 oferece um exemplo ilustrativo das coordenadas geradas pela função $pontos^+$ em razão das adjacências de um gene de entrada.

$$\begin{aligned} pontos(g = \langle b_1, \ldots, b_n \rangle) = \{\{(x,y)\} \mid\ & \forall (x,y) \in \mathbb{N} \\ & \exists b = (x_b, y_b, z_b, tl_b, tp_b, in_b, fl_b) \in \{b_1, \ldots, b_n\} \\ & ((x_b = x) \wedge (y_b = y))\} \end{aligned} \tag{4.31}$$

$$\begin{aligned} pontos^+(g = \langle b_1, \ldots, b_n \rangle) = \{\{x,y\} \mid\ & ((x,y) \in \mathbb{N}) \wedge \\ & (\forall b = (x_b, y_b, z_b, (tl_b = (w_{cel}, h_{tl}, gid, img)), \\ & \quad tp_b, in_b, fl_b) \in \{b_1, \ldots, b_n\}\ \forall i, j \in \mathbb{N} \\ & (-(h_{tl}/h_{cel}) \leq i \leq 1) \wedge (-1 \leq j \leq 1) \wedge \\ & (((x_b + j, y_b + j) \in \mathbb{N}) \rightarrow ((x_b + j, y_b + j) \in \{(x,y)\}))))\} \end{aligned} \tag{4.32}$$

Definida a operação de fecundação por meio da equação 4.27 e das demais equações utilizadas por ela, o operador de cruzamento pode ser definido para o problema sendo trabalhado.

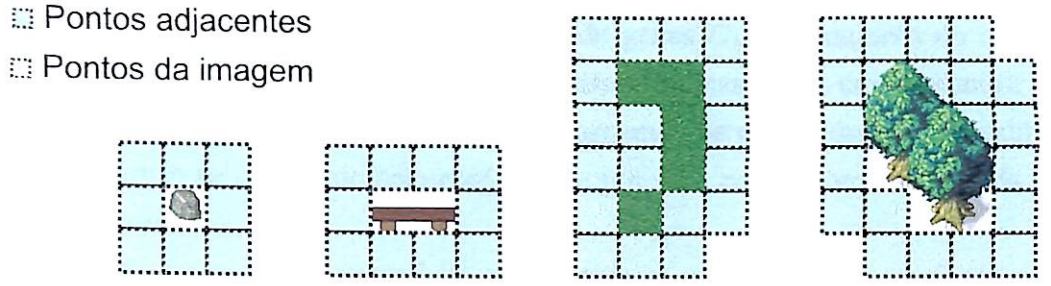


Figura 4.4: *Exemplos dos pontos gerados pela função pontos$^+$ (máximo de oito adjacências para cada ponto da imagem)*

Definição 4.3.11. Cruzamento: o cruzamento é uma função *cruz* de tipo $cruz : MACHO \times FEMEA \rightarrow P$ e definida pela equação 4.33

$$cruz(c_1, c_2) = \{c = \{g_1, \ldots, g_n\} \mid c \subseteq fec(c_1, c_2)\} \tag{4.33}$$

O operador de cruzamento utilizado para o problema apresentado nesta dissertação advém da imediata aplicação da equação 4.33, que emprega ainda a função de fecundação definida pela equação 4.27, para quaisquer dois cromossomos c_1 e c_2 da população P_1 resultante da operação de seleção.

A aplicação do operador genético de cruzamento permite a criação de novos cromossomos a partir dos cromossomos de uma população atual, para com isto, dar origem a uma próxima população. Contudo, há ainda outro operador genético disposto a criação de novos cromossomos.

Este operador, conhecido como operador de mutação, manipula os genes dos cromossomos de uma população, realizando sobre eles procedimentos que visam gerar novos cromossomos por meio da alteração de sua identidade genética.

O GAADT define os procedimentos empregados pelo operador de mutação como operações de inserção, supressão e troca, que atuam sobre os genes do cromossomo dado como entrada para o operador. Como os nomes sugerem, estas operações inserem, removem ou substituem um ou mais genes de um cromossomo por outros, mantendo intacto o restante de seus genes. É interessante que cada uma destas operações que compõem o operador de mutação seja definida antes de se apresentar uma definição apropriada para o operador de mutação.

Definição 4.3.12. Inserção: a operação de inserção, referenciada por *ins* e de tipo $ins : C \times \wp(G) \rightarrow C$, é uma função responsável por adicionar um conjunto de genes ao cromossomo de origem. É definida pela equação 4.34

$$ins(c, G_1) = \begin{cases} c \cup G_1 & \text{se } c \cup G_1 \in AFC \\ c & \text{caso contrário} \end{cases} \tag{4.34}$$

A operação de inserção assume um conjunto de genes G_1, subconjunto do conjunto de genes G do problema, que se deseja inserir no conjunto de genes de um cromossomo c passado como entrada para a operação. O GAADT então assume que o cromossomo c' resultante da operação de inserção será formado pelo conjunto de genes do cromossomo de entrada c, unido do conjunto de genes G_1.

Esta afirmação irá se provar verdadeira apenas para os casos em que o conjunto resultante da união destes dois conjuntos de genes satisfaça as regras estabelecidas pelo conjunto de AFCs adotado para o problema. Do contrário, a operação de inserção retornará o próprio cromossomo de entrada c.

Definição 4.3.13. Supressão: a operação de supressão é uma função *del* de tipo $del : C \times \wp(G) \rightarrow C$ e responsável por remover um determinado conjunto de genes dos genes do cromossomo dado como entrada para a operação. A função definida por este operador está expressa na equação 4.35.

$$del(c, G_1) = \begin{cases} c - G_1 & \text{se } c - G_1 \in AFC \\ c & \text{caso contrário} \end{cases} \tag{4.35}$$

De forma similar a operação de inserção, a operação de supressão atua por meio da alteração dos genes de um cromossomo c fornecido como entrada para o operador. Neste caso, um conjunto de genes G_1 assumido como um subconjunto do conjunto de genes G do problema é subtraído do conjunto de genes do cromossomo c.

Caso o conjunto de genes resultante desta subtração seja um conjunto compatível com as regras definidas pelo conjunto de AFCs do problema, estes genes irão então compor o cromossomo resultante da operação. Caso o conjunto de genes resultante da operação não seja capaz de satisfazer as regras do conjunto AFC, o resultado do operador será o próprio cromossomo de entrada c.

Definição 4.3.14. Troca: o operador de troca denominado *troc* e de tipo $C \times \wp(G) \times \wp(G) \rightarrow C$, é uma função responsável por remover um conjunto G_2 de genes do conjunto de genes do cromossomo c dado como entrada para o operador, ao mesmo tempo em que, adiciona um conjunto de genes G_1 ao conjunto de genes do cromossomo c. A função que define o operador de troca está expressa na equação 4.36.

$$troc(c, G_1, G_2) = \begin{cases} (c \cup G_1) - G_2 & \text{se } (c \cup G_1 \in AFC) \wedge (c - G_2 \in AFC) \\ c \cup G_1 & \text{se } (c \cup G_1 \in AFC) \wedge (c - G_2 \notin AFC) \\ c - G_2 & \text{se } (c \cup G_1 \notin AFC) \wedge (c - G_2 \in AFC) \\ c & \text{se } (c \cup G_1 \notin AFC) \wedge (c - G_2 \notin AFC) \end{cases} \tag{4.36}$$

Perceba que ambos os conjuntos de genes G_1 e G_2 são subconjuntos do conjunto de genes G definidos para o problema. Além disto, uma análise das operações realizadas pela função *troc* em 4.36 torna aparente o fato de que tais operações correspondem àquelas definidas pelos operadores de inserção e de supressão que foram respectivamente definidas pelas equações 4.34 e 4.35.

Tal perspectiva permite compreender as operações de inserção e de supressão como casos particulares da operação de troca. Esta afirmação permite ainda que a função que define o operador de troca seja expressa em razão destas duas outras operações, de maneira que $troc(c,G_1,G_2) = del(ins(c,G_1),G_2)$.

Nesta dissertação, a aplicação da operação de troca se dá imediatamente de sua definição, sendo necessário apenas definir critérios para a composição dos conjuntos de genes de inserção e supressão, que são passados como parâmetros ao operador juntamente com o cromossomo c. Uma definição para a composição destes conjuntos será apresentada mais a frente nesta mesma seção.

Uma vez que as operações de inserção, supressão e troca tenham sido definidas, é possível esboçar uma definição para o operador de mutação em razão destes operadores.

Definição 4.3.15. Mutação: a operação de mutação é definida por um predicado $mut \subseteq \wp(P)$ e expressa por uma função denominada *mut* que corresponde a equação 4.37.

$$mut(c) = \{c' | \ (\exists G_1, G_2 \subset G) \rightarrow (c' = troc(c,G_1,G_2)) \wedge (c' \in cut(P))\} \tag{4.37}$$

A função *cut* utilizada pela equação 4.37 corresponde a um critério de aceitação pertencente ao conjunto Rq de requisitos do ambiente, imposto a qualquer cromossomo submetido ao operador de mutação. Além deste critério, o GAADT estabelece a existência de algumas proposições que devem ser satisfeitos pelo operador de mutação. Estas proposições são apresentadas na lista de tópicos a seguir.

- $\forall c \in P \, \forall G_1 \subset G | \ (G_1 \not\subset c) \rightarrow (ins(c,G_1) = G_c \cup G_1)$;
- $\forall c \in P \, \forall G_1 \subset G | \ ((G_1 \subset c) \wedge (G_1 \neq G_\lambda)) \rightarrow (del(c,G_1) = G_c - G_1)$;
- $\forall c \in P \, \forall G_1, G_2 \subset G | \ ((G_2 \subset c) \wedge (G_1 \not\subset c)) \rightarrow (troc(c,G_1,G_2) = (c - G_2) \cup G_1)$.

Com relação aos conjuntos de genes encaminhados a operação de troca, esta dissertação determina que o conjunto G_2 de genes a serem suprimidos é composto sempre por um único gene g_2 que possua em seus elementos de base um tipo de descrição de *tiles*, dada por tp, pertencente a seqüência INT de tipos de interesse. Esta afirmação é formalmente definida como $\{\forall (g = \langle b_1, \ldots, b_n \rangle) \in G_2 \, \forall (b = (x,y,z,tl,tp,in,fl)) \in \{b_1, \ldots, b_n\} (tp \in INT)\}$.

Além disto, o conjunto de genes a serem inseridos no cromossomo dado como entrada da operação de troca será sempre composto por, ao máximo, dois genes. Destes genes, um será um gene g_β de mesmo tipo e índice de descrição de *tiles* que o gene g do conjunto G_2, enquanto o outro gene, g_α, será um gene do tipo que antecede o tipo do gene g_β na seqüência de tipos de interesse INT.

Destaca-se ainda que as coordenadas x, y, z dos elementos de base do gene do conjunto G_2 serão seqüencialmente divididas entre os genes g_α e g_β em razão de um índice natural aleatório. Esta mecânica tem como objetivo evitar a ocorrência de descontinuidades nos cromossomos sendo gerados. Além disto, a criação de dois genes a partir de um, visa minimizar o surgimento de mudanças abruptas na continuidade das características geográficas representadas pelos cromossomos, uma vez que procura gerar uma "microrregião" de um novo tipo por meio de um gene g_α, e uma reamostragem de uma "microrregião" localizada em suas imediações, g_β.

Devido a adoção deste conceito, a equação para o operador de mutação foi definida assumindo-se o formalismo expresso na equação 4.38.

$$\begin{aligned} mut(c) = \{c' \mid\ & (c = \{g_1, \ldots, g_n\} \in C) \wedge \\ & ((\forall g = \langle b_1, \ldots, b_m \rangle \in \{g_1, \ldots, g_n\}\ \exists G_1, G_2 \subset G) \leftrightarrow \\ & (\forall b = (x, y, z, tl, tp, in, fl) \in \{b_1, \ldots, b_m\} \\ & (tp \in INT \wedge G_1 = quebra(g) \wedge G_2 = \{g\}))) \rightarrow \\ & (c' = troc(c, G_1, G_2)) \wedge (c' \in cut(P))\} \end{aligned} \tag{4.38}$$

Seguindo o contexto anteriormente apresentado, a equação 4.38 define a formação do conjunto G_2 por um único gene g que possui em suas bases um tipo de descrição tp pertencente a seqüência de tipos de interesse INT adotada para o problema. Ao mesmo tempo, o conjunto G_1 corresponde ao conjunto de genes resultante da atuação de uma função chamada *quebra* sobre o gene g de G_2. Esta definição para a função *mut* mantém ainda as demais características da função apresentada pela cquação 4.37.

A função *quebra* é uma função desenvolvida com o propósito de "dividir" um gene de entrada g em dois (sempre que possível), atribuindo as coordenadas x, y e z contidas nas bases do gene de entrada até uma dada posição i a um novo conjunto de bases, que será atribuído ao primeiro gene, enquanto que, atribui as coordenadas x, y e z contidas nas bases do gene de entrada a partir de uma dada posição $i+1$ a um novo conjunto de bases destinado ao segundo gene, que compartilha ainda com o gene de entrada, os mesmos valores para o tipo de descrição tp e índice da descrição in. A função quebra está definida na equação 4.39.

$$\begin{aligned} quebra(g) = \{\{g_\alpha, g_\beta\} \mid\ & \forall (i \leq \#g) \in \mathbb{N}\ \exists (g_\alpha, g_\beta) \in G \\ & (g_\alpha = novo(g, i) \wedge g_\beta = reamostra(g, i))\} \end{aligned} \tag{4.39}$$

As operações descritas para a equação *quebra* no parágrafo anterior são atribuídas as funções *novo* e *reamostra*. Como apresentado no referido parágrafo, a primeira destas funções irá alocar no novo gene sendo criado as coordenadas x, y e z contidas no gene de entrada até a posição i definida pela equação 4.39. A função irá ainda garantir que os *tiles* alocados nas bases do novo gene sejam do tipo de descrição que antecede aquele do gene de entrada na seqüência *INT* sempre que isto for possível, isto é, sempre este tipo não seja o primeiro elemento da seqüência.

A segunda função irá gerar um novo gene a partir do gene de entrada, que reflita reamostragens da região delimitada por este gene, a partir das coordenadas de suas bases para uma posição $i+1$, de acordo com a posição i fornecida como parâmetro pela função *quebra*. Para qualquer uma destas duas funções sendo definidas aqui, caso não seja possível gerar um novo gene que satisfaça as restrições desejadas, será retornado um *gene-inócuo* qualquer. As funções *novo* e *reamostra* estão definidas respectivamente nas equações 4.40 e 4.41.

$$novo(g,i) = \begin{cases} g' = \langle b'_1,\ldots,b'_i\rangle & \text{se } (g = \langle b_1,\ldots,b_i,b_{i+1},\ldots,b_n\rangle) \wedge (i \in \{1,2,\ldots,n\}) \wedge \\ & (\forall b' = (x',y',z',tl',tp',in',fl') \in \{b'_1,\ldots,b'_i\} \\ & \exists b = (x,y,z,tl,tp,in,fl) \in \{b_1,\ldots,b_i\} \\ & \exists tp_j \in INT(tp = tp_j) \wedge \\ & (((1 < j \leq l) \rightarrow (tp' = tp_{j-1})) \vee \\ & ((j = 1) \rightarrow (tp' = tp))) \wedge \\ & (x' = x) \wedge (y' = y) \wedge (z' = z)) \\ g_\lambda & \text{caso contrário} \end{cases} \tag{4.40}$$

$$reamostra(g,i) = \begin{cases} g' = \langle b'_1,\ldots,b'_l\rangle & \text{se } (g = \langle b_1,\ldots,b_i,b_{i+1},\ldots,b_n\rangle) \wedge \\ & (i \in \{0,1,\ldots,n-1\}) \wedge (l = n-i) \wedge \\ & (\exists b' = (x',y',z',tl',tp',in',fl') \in \{b'_1,\ldots,b'_l\} \\ & \forall b = (x,y,z,tl,tp,in,fl) \in \{b_{i+1},\ldots,b_n\} \\ & (x' = x) \wedge (y' = y) \wedge (z' = z) \wedge \\ & (tp' = tp) \wedge (in' = in)) \\ g_\lambda & \text{caso contrário} \end{cases} \tag{4.41}$$

A definição da operação de mutação requer ainda que se estabeleça um critério de corte, definido aqui como *cut* e que atua sobre os cromossomos da população atual P do cromossomo de entrada c, tal como pode ser observado pelas equações 4.37 e 4.38 anteriormente apresentadas nesta seção. Este critério constitui um requisito pertencente ao conjunto Rq de requisitos do ambiente e estabelece a construção de uma sub-população do conjunto de cromossomos C, em

vista da aplicação de restrições que tomam como base comparativa a população de entrada P.

O critério de corte adotado por esta dissertação é expresso pela função *cut* definida na equação 4.42 que estabelece que, a sub-população C_Δ criada a partir de P será composta unicamente por cromossomos que possuam uma adaptação maior ou igual a adaptação média da população atual P.

$$cut(P) = \{C_\Delta | \forall c \in C((c \in C_\Delta) \leftrightarrow (adapt(c) \geq adapt_m(P)))\} \tag{4.42}$$

4.3.3 O ambiente

Por se tratar de um algoritmo genético, o GAADT atua sobre populações de cromossomos que satisfazem as características impostas por um ambiente A no qual estão contidos.

Não bastando satisfazer às características do ambiente A, a própria evolução dos cromossomos ocorre em virtude das características apresentadas por este ambiente, ao longo de seus respectivos tempos de vida. Em outras palavras, mudanças nas características do ambiente são refletidas na evolução apresentada por seus cromossomos em populações seguintes àquela na qual se deu a mudança ambiental.

O modelo do GAADT define seu ambiente A como uma 8-tupla expressa de forma que $A = \langle P, \wp(P), Rq, AFG, AFC, Tx, \sum, P_0 \rangle$. O significado de cada elemento contido na tupla do ambiente está expresso na lista de tópicos a seguir.

- P é a população atual do ambiente, i.e. o conjunto de cromossomos atualmente encontrados no ambiente;
- $\wp(P)$ é o conjunto potência de P, o que é o mesmo que dizer que este conjunto contém todas as possíveis combinações dos elementos de P, incluindo o conjunto vazio;
- Rq é o conjunto de requisitos do ambiente A. O conjunto de requisitos possui influência direta na genealogia dos cromossomos da população P e é expresso na forma de características que impõem restrições para ao ambiente por meio de fórmulas lógicas de primeira ordem;
- AFG é o conjunto de axiomas de formação de genes. Possui a tarefa de determinar quais conjuntos de bases estão aptos a comporem genes para os cromossomos da população P;
- AFC é o conjunto de axiomas de formação de cromossomos. Com tarefa similar a do AFG, possui a função de determinar quais conjuntos de genes estão aptos a comporem os cromossomos da população P;
- Tx define a classificação taxonômica dos cromossomos de P, classificando-os em razão dos critérios de espécie e família. Neste sentido, Tx corresponde ao conjunto de pares

de cromossomos (c',c) que assume o cromossomo c' como sendo construído a partir do cromossomo c, por meio da atuação dos operadores genéticos de cruzamento e mutação. Esta postura permite que se mantenha um registro da genealogia dos cromossomos pertencentes a qualquer população P que o GAADT crie ao longo de sua execução.

- $\sum$ é o conjunto dos operadores genealógicos que agem sobre a população P. Este conjunto de operadores é responsável por moldar a população ao longo das gerações construídas pelo algoritmo, refletindo nela as características do ambiente no qual está inserida;
- P_0 é uma sub-população particular do conjunto potência da população, dado por $\wp(P)$. Esta população é chamada de população inicial e deve conter ao menos um cromossomo.

Uma vez definidos o ambiente e os cromossomos associados a ele, o GAADT opera a partir do princípio de evolução darwinista, o qual determina que qualquer espécie existente em um momento atual é o resultado da transmissão hereditária das características de outras espécies. Em contrapartida, estas características são o resultado de sucessivas variações, i.e. mutações, sofridas por seus cromossomos de origem ao longo das gerações decorridas até que o momento evolucionário atual fosse alcançado.

Este princípio assume ainda que dentre as espécies geradas, sobrevivem apenas aquelas que melhor se adéquam a seu ambiente atual (Darwin, 2002). Neste contexto, incluem-se ainda quaisquer alterações ambientais produzidas pela natureza que venham a ocorrer durante o tempo de vida de uma espécie.

Nesta dissertação, a adoção de uma ótica darwinista para solucionar o problema por meio de um modelo algorítmico, relega o papel desempenhado pela natureza dentro do processo evolucionário ao próprio GAADT. Logo, cabe ao GAADT submeter os cromossomos de uma população à ação dos requisitos Rq definidos para um ambiente A, resultando assim em um processo que gera novos cromossomos para uma próxima população do ambiente, a partir daqueles cromossomos existentes em sua população atual.

4.3.4 O algoritmo

O modelo do algoritmo genético baseado em tipos abstratos de dados é definido como uma função *GAADT* que recebe uma população inicial P_0 como entrada e a submete a simulação de um processo evolutivo, decorrente da aplicação sucessiva dos operadores genéticos definidos pelo conjunto $\sum$.

Este comportamento resulta em uma população de saída ao final do processo, denominada de P_t, composta unicamente por cromossomos que, ou são oriundos de qualquer população anterior a ela (as quais são as populações $P_0, P_1, \ldots, P_{t-1}$) e que continuam a satisfazer os requisitos Rq do ambiente A, ou então, são novos cromossomos obtidos por meio da ação genealógica dos operadores de cruzamento e mutação sobre cromossomos da população P_{t-1} e que satisfazem o conjunto de requisitos Rq do ambiente A.

A mecânica do procedimento apresentado no parágrafo anterior justifica a afirmação de que, uma população P_t resulta da evolução de uma população P_0. Este contexto assume ainda que os cromossomos pertencentes ao conjunto de populações composto por $\{P_0, P_1, \ldots, P_{t-1}\}$, julgados inadequados em relação aos requisitos do ambiente, não irão figurar como parte da população P_t.

Este último conceito permite interpretar estes cromossomos inadequados como parte de uma população de cromossomos "mortos", um argumento que justifica a ausência destes cromossomos na população resultante P_t.

Um cromossomo que passe a integrar esta população de cromossomos "mortos" não volta a se manifestar em quaisquer das gerações futuras construídas pelo GAADT. Entretanto, tais cromossomos continuam a ser representados pelo modelo como forma de manter seu registro, e assim, evitar sua reaparição em qualquer iteração da função *GAADT*.

O ato de restringir o ressurgimento deste conjunto de cromossomos em populações futuras é uma reflexão do princípio de evolução darwinista adotado pelo modelo do algoritmo, que não contempla a possibilidade de que uma espécie já extinta retorne a vida em qualquer momento posterior a seu desaparecimento.

Conseqüentemente, faz-se necessário definir este critério e explicitar seu propósito como o de preservar os cromossomos de uma dada população atual P, orientando o corte de cromossomos de uma população que não devam figurar em qualquer população que se siga à atual.

O modelo do GAADT assume este critério como um ponto de corte representado através de um predicado unário chamado p_{corte}. Este predicado pertence ao conjunto Rq de requisitos do ambiente e atua sobre os cromossomos de qualquer população atual P. A definição formal que o GAADT atribui a este critério está expressa na equação 4.43 que possui o parâmetro φ como um valor a ser definido.

$$p_{corte}(P = \{c_1, \ldots, c_n\}) = \{c |\ (c \in P) \wedge (adapt(c) \leq \varphi)\} \tag{4.43}$$

Para esta dissertação, o critério utilizado como ponto de corte irá possuir o mesmo princípio apresentado pela função *cut* na equação 4.42. A diferença neste caso está apenas no fato de que os cromossomos a figurar no conjunto sendo criado por p_{corte} são provenientes unicamente da população atual P dada como entrada, em contraste com a função *cut* na qual todo o conjunto de cromossomos do sistema era considerado como fonte para os cromossomos do novo conjunto sendo formado.

Neste sentido, o que se quer definir é que para a instância do GAADT utilizada nesta dissertação, o parâmetro φ equivale ao valor da adaptação média da população de entrada P. Logo, aqui cabe apenas o devido estabelecimento desta afirmação por meio de uma reescrita da equação 4.43 na forma da equação 4.44.

$$p_{corte}(P = \{c_1, \ldots, c_n\}) = \{c |\ (c \in P) \wedge (adapt(c) \leq adapt_m(P))\} \tag{4.44}$$

Finalmente, no que diz respeito aos critérios de parada adotados pela função *GAADT*, são definidos dois critérios de parada passíveis de interromper a execução do algoritmo. O primeiro destes critérios vincula ao algoritmo um número máximo de iterações possível de se executar antes que um resultado seja retornado. Este critério é definido em razão de um valor T responsável por determinar o número máximo de execuções.

O segundo critério é definido em função das características do que se considera como uma população ótima. Este critério é alcançado uma vez que o algoritmo, durante sua simulação de processo evolutivo, gere uma população para a qual todos os cromossomos possuam adaptação maior ou igual a um valor k pré-determinado.

Tal como outros critérios definidos para o modelo, estes dois critérios de parada também constituem requisitos pertencentes ao conjunto Rq de requisitos do ambiente A para o qual o algoritmo atua.

Definição 4.3.16. GAADT: o GAADT é um algoritmo genético baseado em tipos abstratos de dados definido por uma função denominada *GAADT* de tipo $GAADT : A \rightarrow A$ e expressa pela equação 4.45.

$$GAADT(P_t) = \begin{cases} P_{otm} & \text{se } P_{otm} = \{c |\ \forall\, c : P_t(adapt(c) \geq k)\} \neq \varnothing; \\ P_{t+1} & \text{se } t+2 = T; \\ GAADT(P_{t+1}) & \text{caso contrário} \end{cases} \tag{4.45}$$

A equação 4.45 assume uma população de entrada P_t e uma próxima população dada por P_{t+1} que, de acordo com o que foi estabelecido ao longo desta seção, é definida em razão dos operadores genéticos e dos critérios aplicados sobre o ambiente A, o que resulta em, $P_{t+1} = cruz(a,b) \cup mut(c) \cup p_{corte}(P_t)$, tomando os valores a, b e c como cromossomos de P_t, de maneira que $a, b, c \in P_t$.

A equação 4.45 assume ainda um valor k do corpo ordenado K adotado para o problema, imposto ao ambiente A como critério de aceitação de cromossomos na população P_t que satisfazem ao problema. É adotado também um número máximo de iterações definido por T, de forma que $T \in \mathbb{N}$ e, conseqüentemente, qualquer iteração t é tal que $t \in T$.

Os dois critérios expressos no parágrafo anterior correspondem aos dois critérios de parada definidos para o GAADT, sendo a execução do último destes critérios garantida para todos os casos em que o algoritmo não pare em virtude do primeiro critério. O último elemento da equação 4.45 corresponde a população definida por $t = 0$, denotando assim a população P_0, assumida como a população inicial do problema.

O algoritmo definido pela função *GAADT* é então dito como um procedimento correto. Uma afirmação que se deve ao fato de que o processo realizado por esta função, retorna uma saída bem determinada para qualquer população de entrada Pt.

Capítulo 5

O Sistema

Ao longo desta dissertação, foi afirmado em capítulos anteriores que, em razão do problema apresentado aqui, cujos detalhes constam na seção 3.3, decidiu-se por trabalhar uma solução a partir da perspectiva oferecida pelo modelo de algoritmo genético baseado em tipos abstratos de dados, o GAADT.

Com este propósito, apesar de se aceitar que a forma ideal de transcorrer com um trabalho que trata de jogos seja elaborar o jogo em questão do início, o que se entende é que a complexidade de um projeto deste porte não é viável para o período de tempo disponibilizado por um mestrado. Por esta razão, foram procuradas maneiras de agilizar o processo de desenvolvimento, a fim de que fosse possível se focar na tarefa de aplicar os modelo de algoritmo apresentado no capítulo 4 o tão logo quanto possível.

Logo, neste trabalho optou-se por realizar um levantamento inicial de tecnologias e aplicações que pudessem acelerar o processo de desenvolvimento, ou ainda, que permitissem a utilização de seu código fonte, realizando-se no mesmo apenas as alterações necessárias ao escopo do projeto.

Dito isto, este capitulo será, inicialmente, dividido em duas seções. A primeira delas irá focar o levantamento mencionado no parágrafo anterior, frisando as qualidades e dificuldades de cada ferramenta analisada e o porquê de terem sido selecionadas, ou não, na elaboração do projeto desta dissertação. Enquanto que a segunda seção irá discutir as características da ferramenta selecionada e dos módulos adicionados a ela em razão do trabalho sendo realizado aqui, sendo o próprio GAADT um destes módulos.

Tomando o contexto desta segunda seção como base, o restante deste capítulo irá então apresentar como todos estes conceitos foram amarrados de forma a dar suporte à atual configuração do sistema que, além de suas operações originais, possui a capacidade de reconfigurar um dado mapa de entrada, mediante a identificação de seus componentes geográficos, por meio da aplicação do algoritmo desenvolvido para esta dissertação que, utiliza do modelo do GAADT para gerar uma saída que resulta em uma nova configuração para o mapa de entrada, em razão da modificação de suas características geográficas.

O contexto de operação do sistema será apresentado em maiores detalhes a partir da seção 5.3, e irá incluir também quaisquer pontos considerados relevantes para sua compreensão que, por qualquer razão, tenham sido omitidos até este momento. Ao final, este capítulo apresentará ainda os resultados que foram obtidos pelo uso do GAADT como solução algorítmica para o problema e as discussões que podem ser feitas a partir destes resultados.

5.1 Levantamento de ferramentas e tecnologias

Dentre as ferramentas e metodologias analisadas para a realização deste trabalho, aquelas consideradas como relevantes para a compreensão da escolha feita neste trabalho, encontram-se listadas pelas subseções 5.1.1 à 5.1.3.

5.1.1 RPG Maker

O RPG Maker (Enterbrain, 1992-2011*a*), como o próprio nome sugere, é uma ferramenta voltada especificamente para a criação de RPGs. Possui uma *interface* com o usuário bastante intuitiva e para a grande maioria dos projetos descarta completamente a necessidade do usuário possuir qualquer conhecimento prévio de programação, uma vez que todas as ações do jogo podem ser definidas por meio de eventos através dos menus da *interface*. Desta maneira, o RPG Maker constitui uma ferramenta que agiliza a criação de um RPG. Para desenvolvedores mais experientes a ferramenta oferece ainda a criação de programações específicas por meio da linguagem RGSS (Enterbrain, 2004-2011*b*) adotada pela ferramenta, uma variação da linguagem Ruby (Matsumoto, 1995-2011).

Contudo, a ferramenta é voltada, sobretudo, para jogos que comportam apenas um jogador, tendo apresentado algumas limitações quando aplicada ao contexto de um MMORPG que inviabilizaram seu uso nesta dissertação, dentre as quais estão:

- Falta de suporte nativo a rede, com os pacotes oferecidos para esta finalidade tendo se mostrado pouco confiáveis;
- Dificuldade, ou mesmo inviabilidade, de se manipular os elementos do jogo a um nível mais baixo, como por exemplo, identificar o ponto em que uma imagem se localiza no mapa de jogo;
- Também não foi possível localizar informações referentes à como comunicar o banco de dados nativo da aplicação com um banco de dados externo, ainda que se saiba que isto é possível.

Estas características criaram o receio de que uma vez que toda a ambientação do jogo estivesse concluída, o que teria sido um trabalho relativamente rápido e prático, fosse impraticável

a implementação dos requisitos do projeto, pondo a perder todo o esforço aplicado. Tais dúvidas acabaram por inviabilizar o uso da ferramenta.

5.1.2 XNA

O XNA (Reed, 2008) é uma ferramenta da Microsoft voltada especificamente para a criação de jogos e que faz uso da linguagem C# (Petzold, 2001). Ainda que a ferramenta possua uma comunidade de usuários ativa, além de diversos pacotes que poderiam ser usados mesmo como base para o desenvolvimento de jogos completos, para todos os efeitos, o consenso é que optar pelo uso do XNA acarretaria em iniciar a criação de um jogo do zero, o que não é compatível com a realidade de tempo disponível para este trabalho. Por esta razão, o uso do XNA neste trabalho também se tornou inviável. Ainda assim, a ferramenta pode vir a ser uma excelente escolha em trabalhos futuros.

5.1.3 Projeto The Mana World e ferramentas relacionadas

O projeto The Mana World (The Mana World Dev Team, 2004-2011) é formado por uma comunidade de desenvolvedores que trabalha na criação de um jogo de RPG homônimo, utilizando-se de um cliente em C++ (Dawson, 2010) e de um conjunto de ferramentas disponibilizados por outros grupos de desenvolvimento, dentre os quais merece menção neste trabalho, o grupo responsável pela criação e manutenção do editor de mapas adotado pelo projeto, o Tiled (Lindeijer & Turk, 2006-2011).

O projeto e as ferramentas adotados por ele são de código livre. Isto em conjunto com uma comunidade de desenvolvimento ativa que interage por meio de fóruns possibilita o diálogo e o esclarecimento de dúvidas de implementação. Por ter sido concebido como um MMORPG o jogo apresenta suporte nativo a rede não tendo sido detectadas quaisquer complicações nos testes realizados.

O jogo em si possui cenários e jogabilidade similares aos do MMORPG Ragnarok (Gravity Co. Ltd, 2002-2011), oferecendo familiaridade aos jogadores. Também foi constatado que a estrutura de jogo em si está suficientemente funcional para atender as necessidades deste trabalho. Finalmente, existe certa facilidade de implementação para os requisitos objetivados por esta dissertação, uma vez que as ferramentas utilizadas pelo cliente, servidor e pelo editor de mapas são códigos livres que se encontram acessíveis ao público.

As dificuldades em se trabalhar com as ferramentas deste projeto encontram-se apenas em suas complexidades. Em particular, os códigos fontes do cliente e editor de mapas que são extensos e não triviais. Somado a isto, existe também a variedade de linguagens de programação adotadas, com o cliente encontrando-se em C++ (Dawson, 2010), o editor de mapas em Java (Oracle, 1995-2011) – atualmente sendo distribuído em Qt (Blanchette & Summerfield, 2008)–, e por fim o servidor, escrito em C (Kernighan & Ritchie, 1988).

Mesmo com as dificuldades de compreensão e adaptação do código que são inerentes as ferramentas deste projeto, a opção por se trabalhar com este conjunto de ferramentas oferece a certeza de ser possível implementar os requisitos que são necessários para alcançar os objetivos desta dissertação.

Em razão do foco deste trabalho estar relacionado à geração de cenários de jogo, uma maior atenção foi dedicada ao editor de mapas, utilizado aqui, como intermediário entre os mapas de jogo originais e a aplicação do modelo do GAADT, para evoluir estes mapas concedendo-lhes assim uma nova configuração. Todavia, existe a intenção de que as demais ferramentas sejam retomadas de uma forma mais aprofundada em trabalhos futuros.

5.2 O editor de mapas Tiled: Conceitos herdados e módulos adicionados

A aplicação desenvolvida, tem como escopo atender os objetivos especificados na seção 1.4.2, ou seja, oferecer aos jogadores conectados ao cliente de jogo e a agentes que nele habitam um ambiente mutável similar ao do mundo real, no sentido de que este ambiente, também passa por alterações geofísicas em suas características em decorrência da passagem do tempo. Para atender a esta necessidade, uma vez que se decidiu trabalhar com as ferramentas relacionadas ao projeto The Mana World (The Mana World Dev Team, 2004-2011), foi utilizado o aplicativo Tiled (Lindeijer & Turk, 2006-2011), um editor de mapas de propósito geral de código livre e escrito em Java capaz de atender a vários gêneros de jogos eletrônicos, no qual, foi possível adicionar os módulos necessários a este projeto. Em conjunto com este editor de mapas, foram utilizados os mapas de jogo, e sua respectivas imagens de cenário, fornecidos pelo cliente e fontes do projeto The Mana World.

A opção de se utilizar o editor de mapas para realizar as alterações nos elementos geográficos do cenário ao invés de fazê-las no cliente de jogo, ou de uma forma mais apropriada, diretamente em seu servidor, deu-se em virtude de que a implementação no cliente de jogo resultaria em cada cliente gerar, para cada mapa de jogo, sua própria variação daquele cenário. Já a implementação no servidor requereria grande quantidade de retrabalho, incluindo a recriação de estruturas que são próprias do editor de mapas. Decidiu-se então que os mapas gerados pelo editor seriam posteriormente substituídos no servidor do jogo e atualizados no lado do cliente, mesmo porque, cliente e servidor já possuem uma estrutura funcional de atualização de arquivos.

Para satisfazer os requisitos propostos por este projeto, o editor de mapas teve módulos adicionados a ele que lhe permitiram identificar os elementos que compõem o cenário de um mapa, bem como, uma vez que tais elementos tenham sido identificados, alterar de maneira autônoma as características geofísicas de um mapa por meio de computação evolucionária, realizando assim uma adição gradual de novas características ao referido mapa.

Estes conceitos se refletem respectivamente na criação e alimentação do conjunto *DscTile* da equação 3.15, e da própria aplicação do modelo do GAADT ao problema, que procura assim, replicar um comportamento que se constata no mundo real, onde é mesmo possível que unicamente em função da decorrência do tempo, um ambiente caracterizado como floresta possa mudar até que se torne um deserto ou mesmo uma área completamente alagada.

Em relação a este último contexto, o que se entende aqui é que neste trabalho é feita uma exploração inicial com relação à abordagem adotada, por se acreditar que, quaisquer critérios utilizados aqui precisaram ser continuamente refinados para atender a constante demanda de por inovação (Schwab, 2004).

Os conceitos originais que foram herdados deste editor de mapas e que são pertinentes para este trabalho foram anteriormente apresentados na seção 3.1, uma vez que constituíam conceitos e terminologias necessários a própria definição do problema sendo tratado por esta dissertação. Informações adicionais sobre o editor de mapas (cuja interface original está ilustrada na Figura 3.1 da página 38) podem ser encontradas em Lindeijer & Turk (2006-2011).

Agora, este trabalho terá continuidade com a apresentação das mudanças que foram realizadas na aplicação original para atender de forma satisfatória aos requisitos necessários ao escopo deste projeto.

5.2.1 Criação de um módulo para armazenamento das descrições dos tiles

Ao longo da definição formal para o problema desta dissertação apresentado na seção 3.3 foi afirmado que se constatou a necessidade de identificar os elementos geográficos do mapa de jogo de maneira a permitir que o algoritmo fosse capaz de distingui-los em função da característica de cenário que representam e assim poder trabalhar adequadamente sobre o mapa de jogo de forma autônoma. Este conceito, refletido pela equação 3.15 na página 51, não foi contemplado pelos desenvolvedores originais do editor de mapas, que o elaboraram tendo em mente que teria sempre um usuário humano atuando sobre ele.

De forma similar, também não havia uma maneira para se definir qual o tipo predominante do cenário do mapa, um valor identificado pelo atributo *pred* do conjunto D dos dados de entrada apresentado na equação 3.3 da página 47, que é utilizado pela equação 3.21 da página 53 para mapear quais os tipos geográficos de elementos do mapa que serão trabalhados pelo processo evolucionário apresentado no capítulo 4.

Logo, o passo inicial deste módulo tratou de prover ao usuário a funcionalidade que lhe permitisse identificar o tipo original do mapa. Um conceito que está ilustrado na Figura 5.1 e cuja informação é salva no próprio mapa de jogo, um tipo específico de XML[1] (Hunter et al., 2007) utilizado pelo editor de mapas (TMX), que teve sua estrutura alterada para acomodar este novo dado.

O maior trabalho deste módulo, entretanto, foi criar a funcionalidade que permite ao usuário

[1] Sigla para Extensible Markup Language.

Figura 5.1: *Interface gráfica da aplicação após as alterações*

identificar os diferentes papeis que podem ser associados a um *tile* em função do componente geográfico que representa. Esta funcionalidade, que constitui o próprio módulo de que trata esta seção, está ilustrada na Figura 5.2.

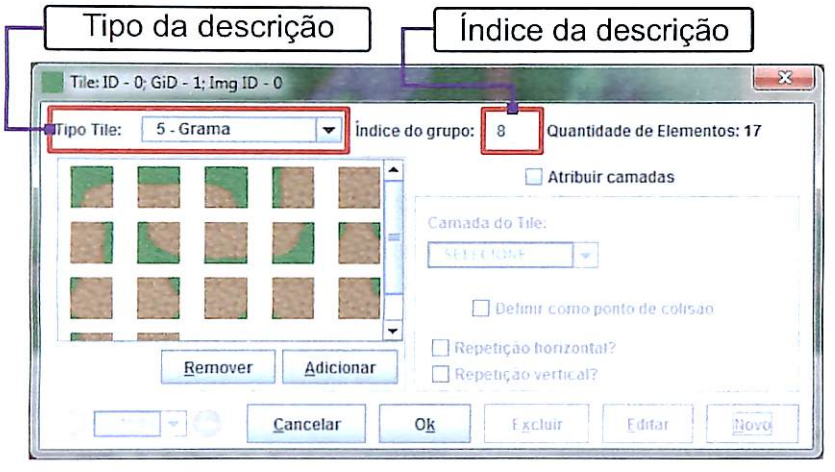


Figura 5.2: *Modulo de inserção de descrições exibindo uma das descrição de tiles cadastradas*

Abstraindo-se as características e parâmetros definidos pelo módulo, o que ele realiza de fato é a criação de registros de informações que individualizam os *tiles* em razão do tipo de cenário que representam. Neste sentido, para cada registro existe um único *tile* vinculado a uma descrição de tipo de cenário, um índice para esta descrição, e uma *flag* que identifica o *tile* como um ponto de colisão.

Estes parâmetros correspondem respectivamente aos atributos tl, tp, in e fl do conjunto $DscTile$ da equação 3.15 e estão ilustrados na tela apresentada na Figura 5.2 que agrupa determinados registros em um mesmo grupo em função de valores idênticos para o tipo de descrição e para o índice da descrição. Uma abordagem que está de acordo com o conceito apresentado na seção 3.3 referente a identificação de diferentes tipos de um mesmo terreno e que pode ainda

ser entendida ao se imaginar que por exemplo, não é possível compor um riacho a partir de um único *tile* representando água e que, ao mesmo tempo, nem todo *tile* representando água pertence ao desenho deste riacho.

Uma analogia similar pode ser aplicada à identificação dos *tiles* que exigem a ocorrência de pontos de colisão. Neste sentido, é possível imaginar que pedras ou paredes dispostas no mapa de jogo podem requerer a ocorrência de eventos de colisão, ao mesmo tempo em que, *tiles* representando uma ponte ou um terreno não devem caracterizar a ocorrência destes eventos.

Os dados cadastrados por este módulo são armazenados no próprio mapa de jogo, que teve a estrutura de seu arquivo modificada para este propósito. Cada registro da informação corresponde então a um elemento do conjunto *DscTile* da equação 3.15 e pode ser acessado pelo GAADT, que os utiliza então para distinguir os diferentes elementos geográficos do mapa de jogo, por meio do mapeamento da informação contida nestes registros em seus elementos genéticos básicos, tal como mostrado na equação 4.2 da página 60.

5.2.2 Criação de um módulo para geração de uma imagem de bordas

O módulo que será discutido nesta seção é o responsável por gerar uma imagem de bordas utilizando do algoritmo de Canny (1986) para a detecção de bordas (discutido na seção 3.2.1) sobre uma imagem gerada por meio da agregação de diferentes *tiles* do mapa de jogo.

Tendo em mente os conceitos dos algoritmos de detecção de borda que foram apresentados na seção 3.2, bem como do algoritmo de Canny que foi apresentado ao longo da mesma seção, esta seção fará agora a apresentação dos procedimentos utilizados para a criação do módulo sendo tratado aqui e dos demais artifícios utilizados para este propósito.

5.2.2.1 Aplicação do algoritmo na construção do módulo

O algoritmo de Canny para a detecção de bordas é utilizado como parte da solução do modelo do GAADT, compondo o núcleo do mecanismo implementado para a avaliação das regiões elaboradas pelo algoritmo, guiando a busca do modelo por melhores resultados. O mecanismo responsável por selecionar no mapa de entrada agrupamentos de genes (regiões menores) na formação de cromossomos (regiões maiores) também utiliza o algoritmo de Canny no processo de seleção.

Em cada uma destas situações, o módulo assume valores próprios para os parâmetros dos limiares máximo, mínimo e para a largura do filtro Gaussiano em razão das tarefas e particularidades específicas que cada situação destas exige que sejam atentadas. Em ambos os casos, o módulo ainda é encarregado da tarefa secundária de realizar a contagem dos *pixels* brancos dentro da área da imagem delimitada por um conjunto de pontos solicitados.

O módulo engloba ainda uma série de artifícios responsáveis por reunir as diferentes partes de imagem em suas diferentes células e camadas do mapa e gerar uma única imagem bidimensional composta pelo agrupamento e sobreposição de todos estes elementos. Contudo, há um

artifício utilizado aqui que se destaca dos demais.

Antes de se falar neste artifício, entretanto, é necessário compreender que ainda que o algoritmo utilizado efetue de maneira adequada a detecção das bordas que delimitam os elementos físicos representados na imagem de entrada, para os propósitos que este algoritmo foi empregado neste trabalho, é possível afirmar que os resultados possuem excesso de detalhamento, superando mesmo o que se consideraria o ideal para o escopo desejado.

Esta afirmação é feita no sentido de que, mais do que um completo detalhamento das arestas originadas pelos elementos compondo uma imagem, o que este trabalho necessita alcançar com este módulo é um apropriado detalhamento do contorno externo que delimita cada elemento representado em uma imagem de entrada. Uma necessidade decorrente do objetivo que este módulo busca que é permitir uma contagem dos *pixels* brancos de uma imagem, que possibilite detectar a correta disposição dos elementos que a compõe. Um objetivo que pode ser prejudicado devido a alta taxa de *pixels* brancos gerada pela representação de elementos complexos.

Por esta razão foi adotado um artifício para reduzir a complexidade cromática dos elementos constituintes de uma imagem em razão de suas três principais componentes de cores de acordo com o padrão RGB (*Red*, *Green*, *Blue* - Vermelho, Verde, Azul).

O que se fez foi, tomando-se o conjunto de descrições alimentadas no sistema por meio do módulo apresentado na seção 5.2.1, varrer os *pixels* das imagens que representam cada um dos *tiles* pertencentes a um mesmo grupo, i.e. conjuntos de descrições com valores idênticos para os parâmetros de tipo de descrição (tp) e índice de descrição (in), e identificar as três cromaticidades mais freqüentes de cada grupo.

A identificação destes valores foi realizada considerando cada *pixel* das imagens de um grupo e discriminando-as em razão de sua componente RGB de maior valor. Estes *pixels* foram então contados e aqueles de maior freqüência para cada uma das máximas componentes de RGB foram atribuídos as imagens dos *tiles* contidos nos registros deste grupo.

Posteriormente, cada imagem sendo avaliada pelo módulo passa por um procedimento que antecede sua agregação em uma única imagem e seu encaminhamento ao algoritmo de detecção de bordas. Neste processo, são recuperados os *pixels* equivalendo as cromaticidades mais freqüentes dos grupos a que pertencem cada uma destas imagens menores, as quais têm então seus *pixels* originais substituídos por aquele correspondendo ao *pixel* reservado à componente RGB de maior valor do *pixel* original que está sendo avaliado.

O emprego deste artifício resultou em imagens de bordas formadas quase que exclusivamente pelas arestas externas dos elementos representados nas imagens, constituindo assim uma representação mais adequada ao propósito buscado nesta implementação. As figuras 5.3 e 5.4 apresentam exemplos dos resultados obtidos utilizando o algoritmo de Canny para detecção de bordas respectivamente com e sem o emprego do artifício apresentado, considerando valores para os limiares máximo, mínimo e a largura da máscara Gaussiana respectivamente de $1,2$; $0,8$ e $3,0$ para a Figura 5.3 e de $1,0$; $0,5$ e $3,0$ para a Figura 5.4.

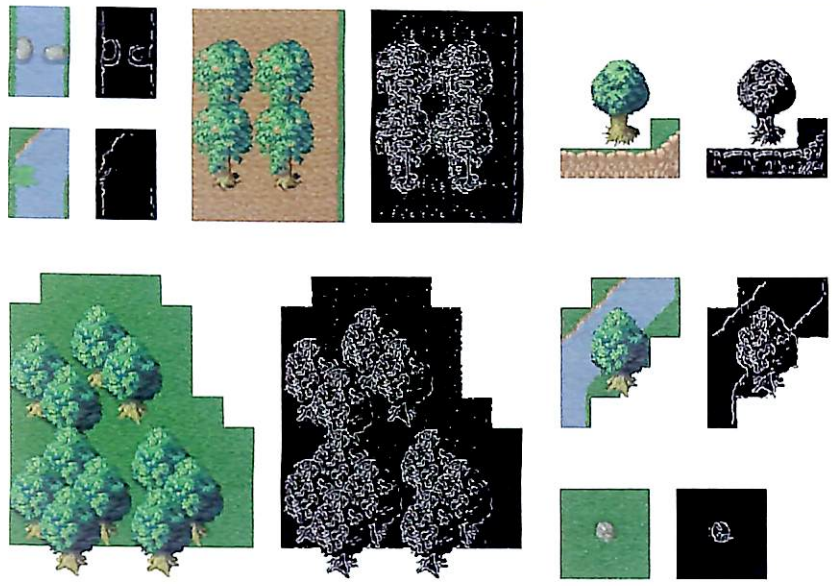


Figura 5.3: *Exemplo dos resultados da detecção de bordas sem a redução da cromaticidade das imagens de entrada*

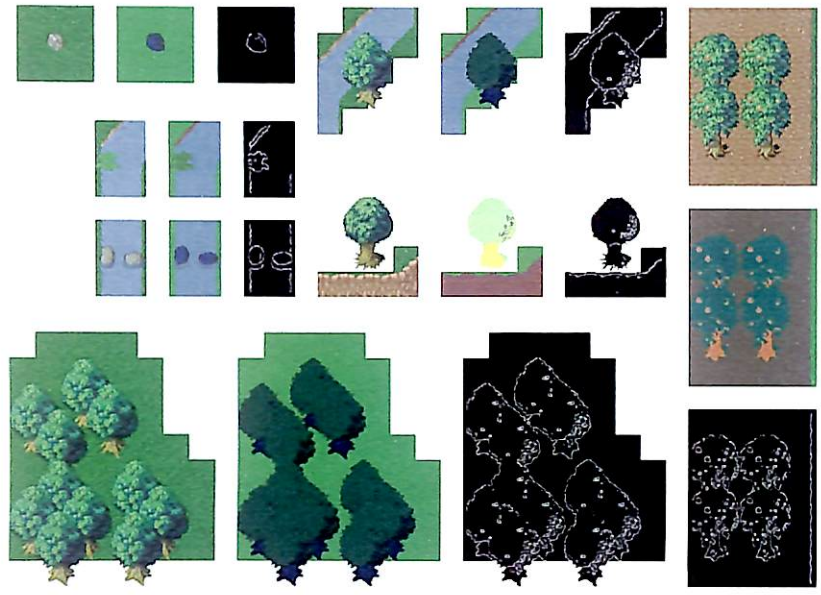


Figura 5.4: *Exemplo dos resultados da detecção de bordas com a redução da cromaticidade das imagens de entrada*

5.2.3 Criação de um módulo para a evolução de um mapa

Uma vez que os dados referentes à identificação dos elementos geográficos do mapa tenham sido alimentados é possível iniciar a execução do GAADT de forma apropriada.

Externamente a implementação deste módulo não exigiu nada além da inclusão de um botão na *interface* gráfica da aplicação, como pode ser visto na Figura 5.1, que possui unicamente o propósito de iniciar a execução dos procedimentos implementados por este módulo.

Internamente, entretanto, o modulo é complexo e corresponde precisamente a aplicação do modelo do GAADT na forma que foi apresentado no capítulo 4 em vista das características e

restrições estabelecidas pela definição formal do problema mostrada ao longo da seção 3.3.

5.3 O funcionamento do sistema

A simulação de processo evolutivo realizada pelo GAADT pode ser inicializada tão logo o sistema tenha sido alimentado com as informações de cada um dos elementos geográficos do cenário que se pretendem utilizar durante a execução do GAADT, incluindo tanto os elementos existentes no mapa de entrada como elementos a serem, potencialmente, inclusos.

Para este propósito, o sistema também requer que o tipo de cenário tenha sido informado, o que caracteriza a correta indicação do tipo que predomina na geografia do cenário. Esta informação é essencial para o processo evolutivo que o sistema desenvolve, uma vez que será utilizada em conjunto com o tipo de cenário que se quer obter para determinar os tipos de elementos geográficos que o algoritmo considerará como interessantes de se avaliar.

Neste sentido, é possível perceber que estas duas informações serão utilizadas pelo sistema para definir a composição da seqüência de tipos de interesses *INT* apresentada pela equação 3.21 do capítulo 3, a qual, de acordo com o que foi apresentado no capítulo anterior, é responsável não apenas por informar quais características do mapa de jogo merecem atenção, como também, por identificar o quão prioritária uma característica é sobre a outra.

Cabe também lembrar que, conforme foi afirmado nas seções 3.3 e 5.2.1, ainda que o tipo predominante do mapa de jogo seja informado pelo usuário, a estação climática a que o cenário do mapa é submetido é aleatoriamente escolhida pelo sistema, tal como apresentado pela equação 3.20. Um procedimento que visa evitar a geração mecânica de determinados tipos de cenários finais em razão de seus respectivos tipos de cenário inicial.

Uma vez que estes critérios tenham sido definidos, o sistema tem como tarefa inicial cortar o mapa de entrada em seções menores. Neste sentido, o termo corte significa aqui o ato de agrupar as células do mapa de jogo em conjuntos de elementos que compartilham de uma característica em comum. Um procedimento que intenciona criar conjuntos de células que possam ser atribuídas ao GAADT na qualidade de genes e que parte do princípio de que, para o GAADT, cada célula da malha do mapa é por si só uma base em potencial que requer apenas que exista um *tile* vinculado a ela.

Por esta razão, o procedimento de corte é individualmente realizado para cada camada do mapa de jogo, garantindo assim a geração de conjuntos de células que estejam vinculadas a uma única coordenada espacial z, uma condição que é definida como um dos requisitos do AFG adotado por este trabalho e que pode ser constatada no capítulo 4.

Uma vez que o que se intenciona com esta seção é apresentar a estrutura geral de funcionamento do sistema, explanações mais detalhadas sobre como ocorre o procedimento de corte do mapa serão adiadas até a próxima seção deste capítulo. Logo, cabe aqui apenas afirmar que os conjuntos de células provenientes do corte são elaborados de maneira a satisfazer tanto aos cri-

térios definidos pelo conjunto de bases B do GAADT como àqueles definidos por seu conjunto de axiomas de formação de genes (AFG).

Concluída esta etapa, todos os genes responsáveis por compor o mapa de jogo de entrada terão sido identificados. Portanto, a etapa seguinte terá então a tarefa de agrupar estes genes em diferentes subconjuntos para assim criar os cromossomos que irão compor a população inicial do GAADT.

Com este propósito em mente, os genes criados são então agrupados em conjuntos que são submetidos às operações definidas pelos AFCs apresentados no capítulo anterior. Este processo se repete até que todos estes genes tenham sido utilizados na composição de cromossomos para a população inicial. As operações responsáveis por esta tarefa serão mais bem explorados na seção 5.5.

Cabe aqui esclarecer que, ainda que o corte do mapa seja realizado de maneira a compor um conjunto de genes apropriados para o GAADT, com o qual são criados os cromossomos para a população inicial, nenhum destes procedimentos compete às tarefas do GAADT, que apenas solicita a realização do corte do mapa e utiliza sua saída para alimentar a chamada responsável por requerer a criação de uma população, a qual passa então a ser utilizada como população inicial e que equivale a representação integral do mapa fornecido como entrada.

A maneira como o mapa originalmente dado como entrada para o sistema é cortado em regiões, que por sua vez, são posteriormente agrupadas em diferentes conjuntos, é o que justifica as afirmações feitas no capítulo 4 de que os genes são "microrregiões" do mapa, enquanto que os cromossomos, são "macrorregiões" formadas a partir dos genes.

Uma vez que estas operações tenham fornecido os cromossomos para a população inicial do GAADT, o mesmo retornará como resultado um único cromossomo que irá corresponder a uma nova configuração dos elementos geográficos de uma das regiões existentes na população inicial, ou mesmo, um cromossomo delimitando uma nova região e que não consta nas regiões delimitadas pelos cromossomos da população inicial.

Em ambos os casos, as características do cromossomo gerado ao final do processo evolucionário do GAADT será o produto da aplicação dos operadores genéticos sobre os cromossomos da população inicial e das populações subseqüentemente geradas a partir dela, caracterizando a saída do processo como resultante de pequenas alterações realizadas sobre os cromossomos do GAADT ao longo de sua execução em todas as populações anteriores à população final.

Cada população do algoritmo teve ainda seus cromossomos individualmente avaliados em função de seus genes. Um procedimento realizado por meio da função *adapt* apresentada na equação 4.19 e que toma ainda como base de cálculo o valor de cada gene do cromossomo sendo avaliado, determinado por meio da função *grau* do gene que é apresentada na equação 4.18 e que atribui ao gene um valor real não negativo com base na prioridade das características geográficas que representa e na forma como esta característica se manifesta no cenário.

Um modelo exemplificando as etapas realizadas pelo sistema por meio de um diagrama de máquina de estados finitos, inicializada para um dado mapa de entrada com seus elementos

geográficos já bem definidos, está ilustrado na Figura 5.5. A ilustração demonstra ainda que uma vez que o mapa de entrada original tenha sido modificado de maneira a incorporar as características resultantes do cromossomo de saída, não existem empecilhos em realimentá-lo no sistema como mapa de entrada para uma execução seguinte do algoritmo, sendo possível assim aplicar sucessivos processos evolutivos sobre um mesmo mapa.

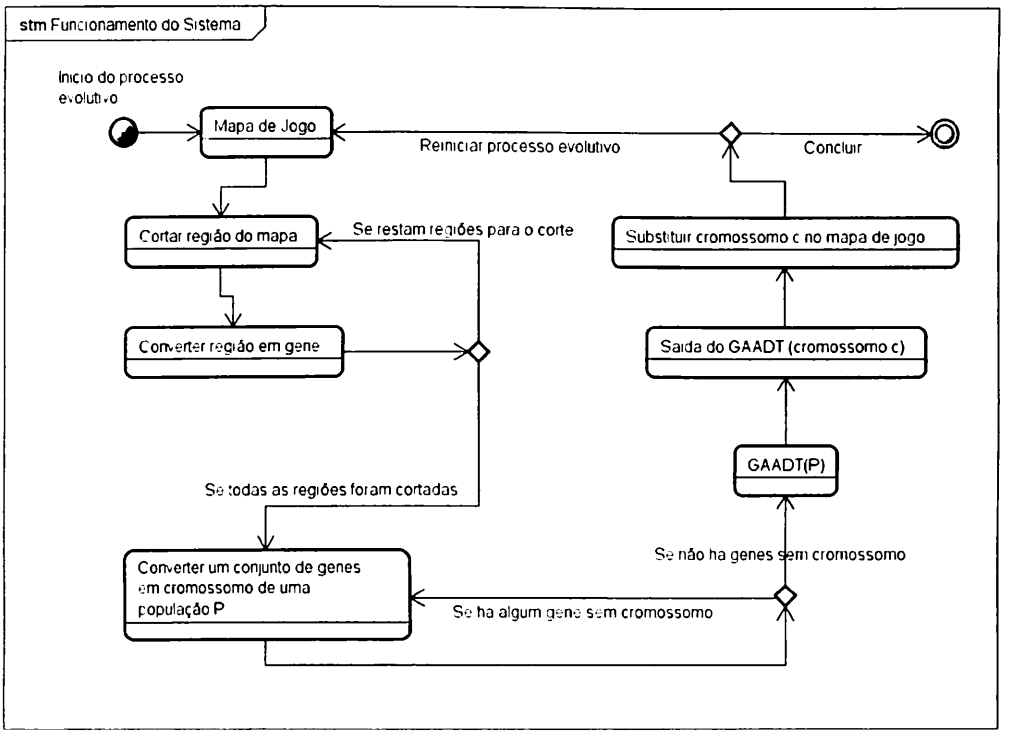


Figura 5.5: *Modelo exemplificando o funcionamento do sistema por meio de uma máquina de estados finitos*

Ressalta-se ainda que, no ato de incorporar no mapa de entrada as características do cromossomo gerado como saída do processo evolutivo do GAADT, uma correta asserção dos pontos de colisão existentes na região representada por este novo cromossomo é realizada utilizando-se do valor do atributo fl, uma vez que o mesmo constitui uma *flag* para eventos de colisão e está contido em todos os elementos de base do GAADT, tal como especificado ao longo da seção 4.3.1. Uma abordagem que está de acordo com o contexto apresentado aqui, que compara cada elemento de base a uma célula do mapa de jogo.

Logo, não há empecilhos para se recuperar esta informação ou realizar a devida marcação das áreas de colisão do mapa, cabendo apenas ao usuário a responsabilidade de informá-las corretamente no ato do cadastramento das descrições de *tiles*, por meio do módulo apresentado na seção 5.2.1. Desta forma, o sistema irá então considerar que qualquer posição x, y do mapa de jogo deve caracterizar um ponto de colisão se houver, ao menos uma camada de coordenada z, para a qual, o *tile* na célula na coordenada x, y, z exija a ocorrência de uma colisão.

5.4 O corte do mapa e a formação dos genes

Durante o desenvolvimento da instância do GAADT para o problema apresentado por esta dissertação, foi percebida a necessidade de se criar um procedimento capaz de cortar o mapa de entrada do sistema em blocos menores, facilitando assim o ato de se trabalhar as características geográficas do mesmo.

Uma vez que se identificou esta necessidade, foi decidido que as seções de mapa a serem criadas deveriam corresponder a genes, para que, a partir deles, pudessem ser formados cromossomos que alimentassem a população inicial do GAADT. Por se adotar esta convenção, cada bloco de mapa recortado pelo procedimento é submetido ao conjunto de AFGs descrito ao longo deste trabalho, garantindo assim que cada região recortada satisfaz a condição de gene.

Para atender a este contexto são criados elementos de base a partir de cada célula do mapa de entrada. Estes elementos são então agrupados em conjuntos que são validados, ou não, em razão dos AFGs do algoritmo. O processo de corte leva ainda em consideração diferentes parâmetros definidos pelo mapa de entrada, incluindo suas dimensões de comprimento e largura.

O procedimento então assume que cada camada dará origem a seu próprio conjunto de genes. Esta suposição está relacionada a restrição imposta pelo AFG a seus genes, que exige que todas as bases de um gene pertençam a uma mesma coordenada z. Logo, uma vez que dado que o índice de uma camada equivale ao índice de coordenada z de uma base, é natural que as bases do algoritmo formadas a partir das células do mapa sejam agrupadas em conjuntos de elementos que dizem respeito, unicamente, a camada em que estas células se encontram.

Os conjuntos de genes formados pelas bases delimitam diferentes regiões do mapa de jogo. Cada uma delas pode ser vista como um bloco de informações disposto lado a lado com os demais blocos da camada em que se encontra. Esta analogia permite ainda uma melhor compreensão do cálculo das dimensões máximas permitidas a um gene no que diz respeito a seu comprimento e sua largura, uma vez que o GAADT desta dissertação estipula por meio de seus AFGs um valor máximo para estas dimensões, as quais, correspondem precisamente à diferença entre os maiores e menores valores para coordenadas x e y contidas em um gene.

Todos estes critérios são levados em consideração pelo processo de corte, que toma como ponto de partida as coordenadas espaciais de uma dada célula do mapa de jogo, atribui à base originada a partir desta célula a um conjunto de bases, e adiciona a ele as bases com origem nas células de coordenadas espaciais vizinhas as da primeira base do conjunto e que compartilham de sua mesma característica. Este processo se repete até que não haja mais células vizinhas às das bases do conjunto que compartilhem com ele de uma mesma característica, ou, até que a região "recortada" atinja o comprimento ou a largura máxima permitidos.

Este trabalho assume que duas células do mapa correspondem a elementos de base de uma mesma característica, se os *tiles* associados a elas possuem os mesmos valores para os parâmetros do tipo de descrição de *tiles* (tp) e do o índice desta descrição (in). Ambos definidos na seção 4.3.1 como atributos do conjunto base.

Neste sentido, o processo que agrega bases de células do mapa em um único conjunto com o propósito de, a partir dele, criar um gene, pode ser visto como um laço que, a cada execução, tenta incluir neste conjunto, bases provenientes de células localizadas em uma das adjacências da região que delimita.

Por sua vez, as adjacências de qualquer célula que defina uma base em um ponto $[x,y,z]$ do mapa, são células que compartilham de ao menos um vértice, ou uma aresta, com esta célula. Ou seja, são células localizadas imediatamente a esquerda $[x-1,y,z]$, direita $[x+1,y,z]$, superior esquerda $[x-1,y-1,z]$, superior direita $[x+1,y-1,z]$, inferior esquerda $[x-1,y+1,z]$ e inferior direita $[x+1,y+1,z]$. Bem como, diretamente "acima" $[x,y-1,z]$ e "abaixo" $[x,y+1,z]$ da célula sendo avaliada. Este procedimento opera de forma equivalente aquele definido para função $pontos^{+}$ por meio da equação 4.32.

O laço ao qual o procedimento apresentado no parágrafo anterior pertence constitui um laço mais interno do processo de agregação, de forma que, cada conclusão de sua execução, quer seja pela impossibilidade de alocar mais elementos ao conjunto ou por quaisquer dos critérios de parada que validam a formação de um gene, resulta em um único conjunto de bases cuja composição é validade pelo AFG da instância do GAADT desta dissertação. Este processo é um terceiro laço que o algoritmo utiliza para compor um gene e se localiza dentro de um segundo, responsável por "eliminar" do mapa de entrada as células que correspondem às bases já alocadas no conjunto e que é executado até que não haja mais células disponíveis para se utilizar.

Já o laço mais externo (o primeiro laço) possui como tarefa recuperar das camadas do mapa de entrada, todas aquelas que são destinadas a alocação de elementos geográficos. Uma afirmação que permite intuir que o processo de corte trabalha unicamente em respeito às camadas de *tiles* e desconsidera completamente as camadas de colisão e as demais camadas de objetos contidas no mapa de jogo.

A estrutura em três laços apresentadas ao longo desta seção corresponde à estrutura essencial do método empregado aqui para o corte do mapa. Um processo que é apresentado em forma de pseudocódigo no Código 5.1 e que resulta em um vetor de genes como retorno da operação.

Já o pseudocódigo apresentado pelo Código 5.2 diz respeito a formação do conjunto de pontos que indicam as coordenadas espaciais x, y e z das células que fornecerão informações para os elementos de base que serão elaborados por meio do Código 5.1, para que então seja possível construir os genes. Cabe apenas ressaltar que, o Código 5.2 refere-se às células adjacentes aos pontos de coordenadas mencionados aqui, como pertencentes a sua "vizinhança".

Código 5.1: *Pseudocódigo para o corte do mapa: criação de genes em razão de conjuntos de bases (células do mapa)*

```
1 for cada camada de tiles
2     while a camada possuir ao menos um tile
3         for cada linha e coluna da camada
```

```
4               selecione um único ponto desta camada;
5               if a célula possui um tile
6                   encaminhe a célula para o agrupamento;
7                   converta as células agrupadas em um conjunto
8                   de bases;
9                   construa um gene a partir deste conjunto de
10                  bases;
11                  adicione o gene a um vetor de retorno;
12                  remova da camada todas as células convertidas
13                  em bases e alocadas no novo gene formado
14                  (defina como null os tiles das células);
```

Código 5.2: *Pseudocódigo para o corte do mapa: seleção de células para o conjunto de bases de um gene*

```
1  instancie um vetor de pontos já alocados (points);
2  instancie um vetor auxiliar de pontos (aux);
3  instancie a dimensão euclidiana deste conjunto;
4  adicione o ponto da célula de entrada ao vetor
5  de pontos alocados;
6  do
7      recalcule a dimensão euclidiana;
8      if aux é vazio
9          construa a vizinhança para o ponto de entrada e
10         atribua a aux;
11     if aux não é vazio
12         instancie um vetor auxiliar aux2;
13         for cada ponto armazenado em aux
14             construa a vizinhança e atribua a aux2;
15     for cada ponto de aux2
16         if aux e points não possuem o ponto
17             adicione o ponto a aux;
18     for cada ponto de aux
19         recupere o tile naquele ponto;
20         if tile == null ou
21         tp_dsc_tile != tp_dsc_tile_ponto_entrada ou
22         in_dsc_tile != in_dsc_tile_ponto_entrada
23             remova este ponto de aux;
24         if points já contém todos os pontos de aux, ou,
25         a última dimensão euclidiana calculada se iguala
```

```
26          a altura ou a largura máxima definida
27              esvazie aux;
28      for cada ponto de aux
29          se o vetor de pontos alocados (points) não possui o
30          ponto, adicione-o;
31  while aux diferente de vazio
```

A construção de genes mencionada no Código 5.1 equivale a, submeter o conjunto de bases formado ao conjunto de axiomas de formação de genes que foi apresentado na seção 4.3.1. Neste contexto, o algoritmo utilizado aqui entende que há apenas dois resultados possíveis para este tipo de operação. Ou o conjunto de bases é validado pelo AFG, que neste caso, fornece o novo gene a ser alocado no vetor de retorno, ou então, o conjunto de bases é considerado inválido e o AFG retorna um conjunto vazio.

A metodologia empregada aqui para a seleção dos conjuntos de bases, tomada a respeito das restrições impostas pelo AFG, impossibilita que sejam formados conjuntos passiveis de serem rejeitados pelas regras do conjunto AFG. Todavia, caso tal situação fosse possível, seria necessário aplicar um tratamento especial sobre os elementos que se mostrassem problemáticos, garantido suas validações em um momento posterior e assim, evitando que a população inicial de cromossomos apresentasse descontinuidades na representação do mapa de entrada original.

Para este propósito são empregados de artifícios no momento da construção de cromossomos a partir dos genes resultantes do processo demonstrado por esta seção, os quais, são parte do conteúdo que será apresentado a seguir.

5.5 O agrupamento de genes e a criação dos cromossomos da população inicial

Uma vez que os procedimentos apresentados na seção anterior tenham sido concluídos, terá sido criado o conjunto de genes que representa todas as diferentes regiões do mapa de entrada original. Agora, é necessário agrupar estes genes em cromossomos distintos para assim, gerar de forma efetiva a população inicial do GAADT.

Tal como ocorre com a formação dos genes, a formação dos cromossomos através dos procedimentos que são executados por este módulo deve ser capaz de, gerar conjuntos de genes que satisfaçam os critérios definidos pelas regras do conjunto de AFC adotado para problema sendo estudado nesta dissertação.

De forma igualmente similar ao que ocorre no módulo da seção anterior, o algoritmo utilizado aqui pode ser expresso em função de três laços que são reconhecidos como sendo os seus laços principais, e que se encontram aninhados de maneira que um laço mais externo é sempre o responsável pela execução de um laço mais interno.

O primeiro destes laços possui a tarefa de recuperar o último elemento do vetor de genes e adicioná-lo ao vetor que corresponde a um único conjunto de genes que será submetido às regras do AFC. A este vetor, são agregados genes localizados em células adjacentes as do seu primeiro gene, identificando-os através do mesmo princípio apresentado na seção anterior. Esta região é também definida como a "vizinhança", deste gene, ou de forma alternativa, sua "área de atuação", e é caracterizada na forma de pontos do mapa de entrada que constituem células que compartilham de ao menos uma aresta, ou vértice, com os elementos do, até o momento, único gene deste vetor.

Diferente de um gene, um cromossomo não está limitado a uma única camada do mapa. Por este motivo, ao se discutir suas adjacências, deve-se ter em mente que a região que se define como sua "vizinhança" não irá abranger apenas uma camada em particular que, neste momento, diz respeito a camada na qual se encontra o único gene do conjunto. Para o cromossomo, o perímetro definido por sua "vizinhança" se estende verticalmente, alcançando todas as camadas do mapa que se ocupem de elementos geográficos do cenário.

Em razão desta contextualização, dentro do primeiro laço há um segundo laço que possui a tarefa de identificar a coordenada espacial z em que se estão buscando genes que possam ser alocados no conjunto sendo alimentado. Este processo decrementa o valor de z a cada inicialização do laço, exceto para os casos em que o valor zero já está associado a z. Para estes casos, é atribuído a z o maior valor que consta no conjunto de bases B para o seu eixo de coordenadas.

Logo, uma vez que a vizinhança do conjunto de genes tenha sido delimitada, a busca por novos elementos para este conjunto se inicia pela "vizinhança" localizada diretamente abaixo da camada do último gene que lhe foi inserido. Este procedimento utiliza uma variável de controle como critério de parada que recebe o valor zero sempre que uma iteração do laço resultar na adição de novos genes ao conjunto, sendo incrementada em caso contrário. A parada em si, ocorre sempre que a variável de controle atingir o valor do total de camadas de *tiles* que existem no mapa de entrada, ou seja, o maior valor de z de acordo com o conjunto de bases B.

Desta forma, é possível perceber que o segundo laço tem a função determinar a camada na qual se estão buscando genes para compor um conjunto que pode, ou não, ser validado como um cromossomo, e que agrega ainda, a tarefa de identificar o momento em que esta busca deve ser interrompida. Dito isto, é possível então imaginar o segundo laço como uma estrutura que controla a execução de um terceiro laço, este último, sendo o responsável por varrer, na camada indicada, as posições delimitadas pela "vizinhança" do conjunto de genes e assim, identificar genes que se encontram dentro desta área e que possam ser incluídos neste conjunto.

A cada execução que o terceiro laço realiza, o mesmo procura alocar novos genes no conjunto de genes que está sendo formado, o qual, tem como referência para um ponto de partida o primeiro gene que lhe foi dado como entrada. Cada execução que resulte na alocação bem sucedida de genes ao conjunto ocasiona a redefinição de sua "vizinhança", que é recalculada de forma a abranger a nova área delimitada por todos os elementos do conjunto.

Para este procedimento, o que o algoritmo assume é que dada uma região definida por um conjunto de genes já selecionado (o qual contém ao menos um gene) e uma "microrregião" que corresponde a um gene que se deseja incluir neste conjunto, entende-se que tal gene pode ser incluso neste conjunto desde que satisfaça seqüencialmente aos critérios listados abaixo.

- A intersecção da área delimitada pela representação deste conjunto com aquela delimitada por toda a "vizinhança" do gene que se deseja incluir no conjunto é diferente do conjunto vazio;

- A imagem de bordas construída por meio do modulo apresentado em 5.2.2.1 para a região composta pelo conjunto de genes, e aquela composta pela união desta região com a região definida pelo gene que se deseja incluir no conjunto, deve possuir um total de pixels brancos que não ultrapasse em vinte unidades o valor total de pixels brancos do conjunto original. Valor este, definido como uma margem de erro arbitrária em razão dos bons resultados apresentados nas primeiras etapas de testes.

- Possuir ao menos 30% de toda região definida pelo gene que se deseja incluir no conjunto dentro da área de atuação delimitada pelo mesmo, ou então; que o gene que se deseja incluir no conjunto, possua o mesmo tipo e índice de descrição de *tiles* (tp e in) que os de genes já alocados dentro do conjunto para, ao menos 50%, de seus pontos que intersectem pontos do conjunto de genes, independente da camada onde se localizam.

O segundo critério que foi apresentado no último dos três itens acima desempenha a função de um último recurso do processo de avaliação, através do qual, se procura identificar se há um mínimo de afinidade entre um gene que se deseja adicionar ao conjunto de genes e aqueles genes já alocados neste conjunto, para com isto, determinar se é ou não interessante adicionar este gene ao conjunto.

Esta operação é a última realizada por este terceiro laço do algoritmo e se encerra tão logo não seja mais detectada a existência de quaisquer genes na camada sendo avaliada que possam ser incluídos no conjunto de genes sendo construído.

De volta ao segundo laço, o valor da coordenada z é alterado e uma nova chamada ao terceiro laço é realizada, para com isto, avaliar o atual conjunto de genes em razão do novo valor para a profundidade z, procurando assim identificar outros genes que possam ser adicionados ao conjunto. O processo do segundo laço se repete indefinidamente, até que, de acordo com o que foi anteriormente expresso nesta seção, seja constatado o valor máximo permitido para a variável de controle.

Ao que se chama aqui de segundo laço, inclui-se ainda um procedimento que se localiza fora da estrutura de repetição propriamente dita e que constitui um último esforço para alocar no conjunto de genes aqueles, dentre os genes restantes, plausíveis de serem aceitos como pertencentes ao conjunto de genes sendo formado. Tal precaução é um artifício que procura

evitar que qualquer gene que "de fato" não possa ser alocado em qualquer conjunto de genes sendo construído seja deixado de fora da população inicial do GAADT.

Os critérios que este artifício emprega para determinar a alocação ou não de um gene ao conjunto sendo formado são os de que, ou o gene possua ao menos 50% de seu conteúdo diretamente acima da região delimitada pelos genes do conjunto, ou então, que satisfaça o critério de afinidade que é dado como o segundo critério de aceitação do último dos três tópicos apresentados por esta seção para a verificação do terceiro laço deste módulo.

Cabe aqui deixar claro que, os critérios apresentados no parágrafo anterior são os únicos a que são submetidos quaisquer genes que possuam ao menos uma de suas dimensões com um valor igual ao máximo permitido, seja o comprimento ou a largura.

Isto se deve ao fato de que tais genes são sumariamente descartados dos procedimentos da estrutura de repetição do segundo e terceiro laços, uma vez que se percebeu que, em caso contrário, sua adição a um conjunto de genes submetido a tais procedimentos faz com que seu tamanho exagerado consuma em excesso os genes disponíveis para a composição da população inicial do GAADT, deixando pouco espaço para a criação de uma população inicial diversificada.

A aplicação unicamente deste artifício a genes de dimensões exageradas não prejudica sua composição uma vez que, devido à forma como um mapa de jogo é desenhado na ferramenta adotada, tais genes podem ocorrer apenas na forma de terrenos que estejam presentes na camada de índice zero do mapa ($z = 0$).

Além disto, uma vez que o segundo e terceiro laços descartam tais genes, e que, sua localização na camada base do mapa, i.e. a camada de índice zero, implica em que estes genes estarão localizados nas posições iniciais do vetor, é possível garantir que o primeiro laço só encaminhará estes genes como elementos de entrada para os procedimentos dos laços seguintes, nos instantes finais de sua execução, em um momento no qual, a maioria dos genes disponíveis já terá sido alocada a algum cromossomo, restando, sobretudo, genes que não puderam ser anteriormente classificados.

O que se quer passar aqui por genes não classificados tem relação com o comportamento desempenhado pelo primeiro laço. Neste sentido, ao receber um conjunto de genes como saída dos procedimentos realizados pelo segundo laço, o primeiro laço procura transformar este conjunto de genes em um cromossomo, razão pela qual, o submete as regras implementadas aqui para o conjunto de axiomas de formação de cromossomos e, caso esta ação resulte em um cromossomo, o mesmo é armazenado em uma conjunto de cromossomos que será posteriormente fornecido como a população inicial do GAADT.

Caso esta chamada ao conjunto de AFCs não resulte em um cromossomo, o gene que foi originalmente fornecido como ponto de partida para a formação deste conjunto de genes, e ao redor do qual os demais genes do conjunto foram agrupados, é removido da posição final do vetor de genes disponíveis e colocado na primeira posição deste vetor, garantindo com isto que será avaliado uma segundo vez apenas quando for concluída a avaliação de todos os outros

genes contidos no vetor.

Os procedimentos apresentados nesta seção estão expressos como pseudocódigos nos Códigos 5.3, 5.4 e 5.5 como forma alternativa de ilustrar seus principais conceitos.

Código 5.3: *Pseudocódigo para o 1º laço da criação da população inicial de cromossomos*

```
1  inicialize com zero o número de execuções (turns);
2  recupere o número de genes no vetor (size);
3  while turns menor que size
4      recupere o último gene no vetor (last);
5      selecione elementos no vetor de genes para serem alocados
6      com last em um mesmo conjunto de genes; //(segundo laço)
7      submeta o conjunto de genes ao AFC
8      if o AFC retorna um cromossomo
9          adicione o cromossomo a população de retorno;
10     else
11         realoque este gene no vetor, da última para
12         a primeira posição;
13     if o total de genes atualmente no vetor não é igual a size
14         atualize o valor de size com o total de genes no vetor
15         e atribua o valor zero a turns;
16     else
17         incremente turns
```

Código 5.4: *Pseudocódigo para o 2º laço da criação da população inicial de cromossomos*

```
1  inicialize o número de execuções com zero (turns);
2  inicialize o número máximo de execuções com o número de camadas
3  válidas (maxTurns);
4  inicialize a coordenada z em análise com o valor
5  da coordenada z associada ao gene;
6  inicialize o número de elementos no conjunt com zero(area);
7  if o vetor possui z == 0 e dimensões que ultrapassam o
8  comprimento ou a largura máxima permitidos.
9      atribua maxTurns a turns;
10 while turns menor que maxTurns
11     if o último valor armazenado em area é igual ao total de
12     elementos no conjunto
13         incremente turns;
14     else
15         atribua zero a turns;
```

```
16      if z é maior que zero
17          decremente z;
18      else
19          atribua a z o maior índice permitido;
20      agregue genes a este conjunto; //(terceiro laço)
21  verifique se ainda existem genes que possam
22  ser alocados neste conjunto;
23  retorne o conjunto de genes
```

Código 5.5: *Pseudocódigo para o 3º laço da criação da população inicial de cromossomos*

```
1   for cada gene no vetor
2       if z == gene.z
3           if a ''área de atuação'' abrange o gene e o gene ainda
4           não pertence ao conjunto
5               boolean add = avalie se deve adicionar o gene;
6               if add == true e z == 0
7                   se o gene não possui dimensões euclidianas
8                   inferiores as máximas permitidas, add = false;
9               if add == true
10                  adicione o gene ao conjunto;
11                  redimensione a área de atuação;
```

5.6 Resultados

Um exemplo do resultado obtido pela aplicação do GAADT para o problema desta dissertação está ilustrado na Figura 5.6 que exibe o mapa de entrada original fornecido ao algoritmo e o resultado retornado em razão do cromossomo mais adaptado que foi gerado, ao longo das sucessivas gerações decorridas até se alcançar uma população assumida como de estagnação.

Para o exemplo desta figura, foi submetido ao GAADT um mapa representando uma floresta, para o qual o algoritmo gerou como saída um mapa com características de uma região desértica, uma transição que poderia ser considerada como decorrente de um período de estiagem no enredo do jogo. A Tabela 5.1 apresenta os valores assumidos pelos cromossomos mais e menos adaptados ao longo das gerações de populações que o sistema criou durante a simulação de seu processo evolucionário, bem como a média de adaptação destas populações.

A tabela permite identificar a existência de dois momentos nos quais a população parou de evoluir, correspondendo as gerações que vão de 3 a 5 e de 6 a 7. Nestes momentos a adaptação dos indivíduos da população não sofreu alterações, uma característica que em ambos os casos foi mantida até a ocorrência de um novo salto evolucionário. A última população, para a qual

não foi detectado um novo salto evolucionário por 20 gerações, foi então aceita como uma população de estagnação, sendo selecionado então o cromossomo mais adaptado desta população como o resultado do problema.

Resultados			
geração	melhor adaptação	pior adaptação	adaptação média
1	0,924541016	0,00302477	0,572786602
2	0,946289063	0,00302477	0,611779318
3	0,9921875	0,00302477	0,63175816
4	0,9921875	0,00302477	0,63175816
5	0,9921875	0,00302477	0,63175816
6	0,993339978	0,00302477	0,672413871
7	0,993339978	0,00302477	0,672413871
8	0,996235983	0,01467041	0,713797671
9	0,996235983	0,01467041	0,713797671
10	0,996235983	0,01467041	0,713797671
11	0,996235983	0,01467041	0,713797671
12	0,996235983	0,01467041	0,713797671
13	0,996235983	0,01467041	0,713797671
14	0,996235983	0,01467041	0,713797671
15	0,996235983	0,01467041	0,713797671
16	0,996235983	0,01467041	0,713797671
17	0,996235983	0,01467041	0,713797671
18	0,996235983	0,01467041	0,713797671
19	0,996235983	0,01467041	0,713797671
20	0,996235983	0,01467041	0,713797671
21	0,996235983	0,01467041	0,713797671
22	0,996235983	0,01467041	0,713797671
23	0,996235983	0,01467041	0,713797671
24	0,996235983	0,01467041	0,713797671
25	0,996235983	0,01467041	0,713797671
26	0,996235983	0,01467041	0,713797671
27	0,996235983	0,01467041	0,713797671

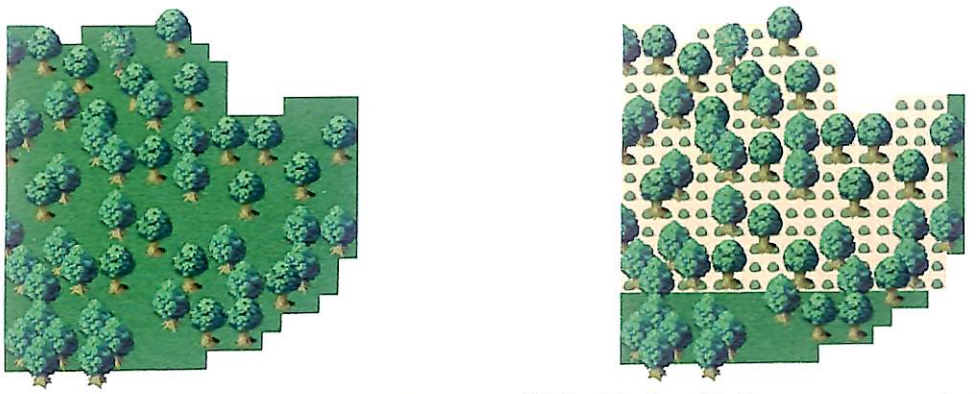
Tabela 5.1: *Inicialização do GAADT*

A Tabela 5.1 também permite perceber os momentos em que o algoritmo começa a convergir, para os quais, os valores de média de adaptação de suas populações vão se encaminhando

(a) *Mapa de entrada (floresta)* **(b)** *Mapa de saída (deserto)*

Figura 5.6: *Comparação de um mapa de entrada e o mapa de saída gerado a partir dele pelo GAADT*

para valores cada vez mais próximos, sinalizando que o algoritmo se aproxima de uma população de estagnação, e portanto, de um resultado. Além disto, a tabela demonstra como o algoritmo não fica necessariamente preso a um primeiro resultado, sendo possível para o algoritmo que, em algum momento, seu processo de busca aleatória o leve a encontrar um resultado ainda melhor, como ocorre no caso demonstrado aqui.

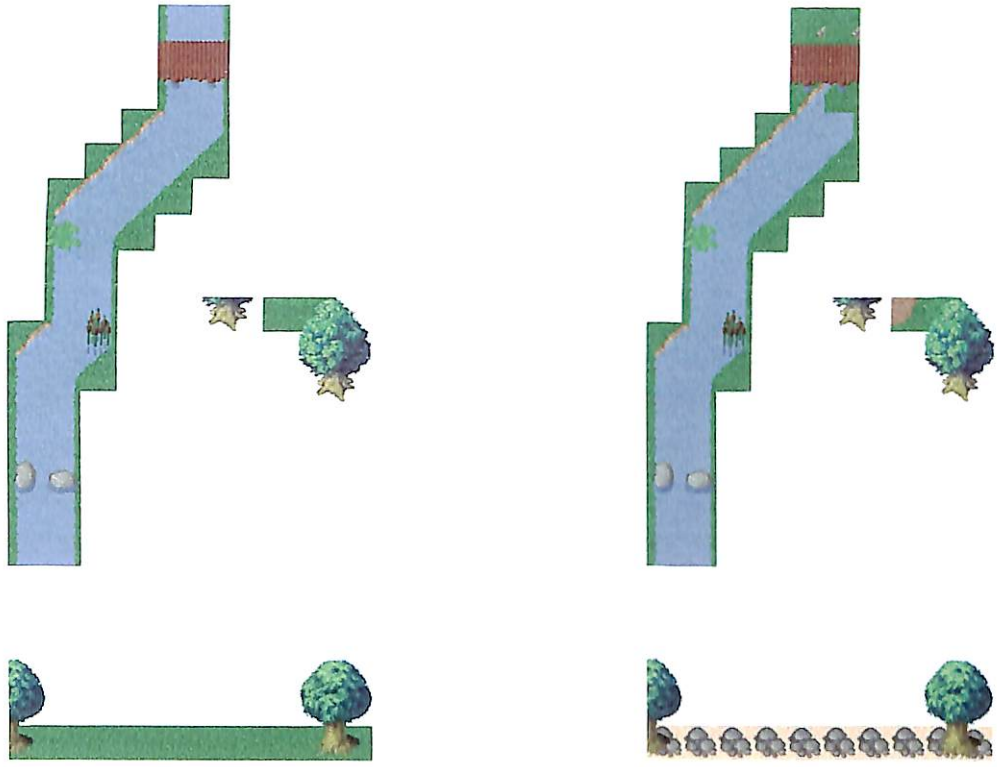


(a) *Região de entrada do cromossomo que viria a ser o mais adaptado* **(b)** *Região de saída do cromossomo mais adaptado*

Figura 5.7: *Comparação entre a região de saída do GAADT antes e depois do processo evolutivo*

Por fim, a Figura 5.7 exibe a região original do mapa de entrada que acabou por ser alterada na região de saída do sistema, como pode ser inicialmente visualizado na Figura 5.6. Como dado comparativo, a Figura 5.8 exibe algumas regiões do mapa que foram alteradas e que mesmo assim foram descartadas como resultados finais por possuírem um valor de adaptação do cromossomo inferior ao da região da Figura 5.7. Note que nesta figura é possível ver

que o algoritmo não apenas substitui a grama da floresta por regiões arenosas, como também gerou cromossomos que representavam a diminuição das áreas alagadas para dar lugar a áreas de terreno, no que seria uma etapa intermediária do processo de desertificação.



(a) *Região de entrada de diferentres cromossomos* **(b)** *Região de saída de diferentes cromossomos*

Figura 5.8: *Comparação entre as regiões de entrada e saída para diferentes cromossomos do GAADT*

Capítulo 6

Conclusão

A metodologia empregada ao longo esta dissertação utiliza técnicas de inteligência artificial, mais especificamente, o algoritmo genético baseado em tipos abstratos de dados, com o propósito de alterar as características geográficas dos mapas bidimensionais de um MMORPG por meio de um processo evolucionário. Uma metodologia que diverge daquela tradicionalmente aplicada para realizar a manipulação topológica de cenários de jogos, que normalmente, é constituída por algoritmos que possuem suas bases nos conceitos da computação gráfica.

Uma discussão das primeiras versões desta implementação e seus respectivos resultados podem ser encontrados em Carvalho et al. (2010*a*), Carvalho et al. (2010*b*) e Carvalho et al. (2010*c*), destacando assim, que o trabalho apresentado por esta dissertação constitui o resultado do refinamento dos conceitos empregados por estes dois trabalhos ao modelo do GAADT.

Os mapas ilustrados como resultados no capítulo 5 permitem argüir que, para sua aceitação apropriada por jogadores humanos, ainda se faz necessário, um maior refinamento da construção do cenário. Contudo, é possível afirmar que, de um ponto de vista funcional, os mapas são considerados adequados para o uso dentro do contexto de cenário dos jogos de MMORPG.

6.1 Considerações finais

A afirmação feita no parágrafo anterior, não procura mascarar a necessidade de se melhorar os resultados. O que na verdade se deseja com tal afirmação é frisar que a aplicação do GAADT como modelo para a solução do problema, foi capaz de gerar mapas adequados ao uso em um real ambiente de jogo.

Com relação à qualidade dos mapas gerados como resultados e a sua aceitação por jogadores humanos, o que se entende é que se faz necessário um refinamento constante dos critérios aqui empregados para a construção dos mapas, de forma não apenas a oferecer cenários mais concisos, como também, de integrar novas interpretações, proporcionando assim um continuo aperfeiçoamento do que constitui uma característica de jogabilidade que adiciona uma perspectiva extra ao jogo.

Uma questão de caráter preocupante que se identificou durante a avaliação do algoritmo diz respeito ao tempo necessário para alcançar um resultado, uma espera que está diretamente ligada a um espaço exponencial de busca, em conjunto, com o tempo necessário para se criar as imagens que representam as regiões de mapa delimitadas por cada cromossomo em razão de seu conjunto de genes, este último, um procedimento que inclui ainda a submissão destas imagens ao modulo de detecção de bordas e o tempo necessário para concluir a criação da imagem de bordas, bem como, a avaliação dos resultados deste procedimento e o cálculo da adaptação do cromossomo com base nestes resultados.

6.2 Trabalhos futuros

O tempo que o processo está consumindo em sua forma atual faz com que se estude a hipótese de que, versões futuras do trabalho apresentado nesta dissertação sejam implementadas diretamente em hardware, a fim de lidar com este problema de uma maneira que cause um impacto significativo na redução de seu tempo de processamento, em virtude do processamento acelerado que pode ser obtido por meio de uma implementação em hardware.

Com relação aos aspectos didáticos apresentados logo no início deste trabalho, o que se percebe deste trabalho é que após um refinamento apropriado, abre-se um espaço para a aplicação de assuntos relacionados a disciplinas de história e geografia, em uma abordagem similar à trabalhada em Tobaldini & Brancher (2006) e Tobaldini et al. (2006) no sentido de que, é possível tomar proveito da manipulação de cenário sendo trabalhada por esta dissertação e refiná-la para simular características biogeográficas coerentes com determinadas áreas do mundo real, bem como, aproveitar-se da ambientação do jogo para traçar um paralelo entre características arquitetônicas e culturais de determinadas raças de personagens, com as civilizações do mundo real nas quais foram inspiradas.

Esta perspectiva permitiria se alcançar não uma real camuflagem do conteúdo educativo (um comportamento desencorajado por Fortuna (2003) por caracterizar uma forma alienante de ensino que poderia passar ao aluno a sensação de estar sendo enganado), mas sim, a possibilidade de apresentá-lo através do contexto e ambientação do enredo do jogo, sendo possível ainda aproveitar-se do conhecimento prévio que os alunos possam ter dos temas abordados para facilitar a transmissão dos assuntos discutidos (Clua et al., 2002).

Com relação ao refinamento do sistema apresentado por esta dissertação para seu emprego como uma ferramenta de auxílio didático, fica apenas a ressalva final de que é importante que seja sempre lembrado que o uso do RPG como ferramenta de ensino, ou de qualquer outra didática diferenciada, não deve ser empregado de maneira excessiva, o que acabaria por lhe adicionar o risco de ser privado de seu caráter lúdico e prazeroso (Fortuna, 2003), tornando-o tão cansativo e alienante quanto tradicionais aulas utilizando apenas de giz e quadro-negro (Bittencourt & Giraffa, 2003).

Referências Bibliográficas

Adams, J. (2004), *Programming role playing games: with DirectX Game Development*, The Premier Press game development series, 2ª ed., Course Technology PTR.

Andrade, R. F. F., Barcia, R. M. & Pezzi, S. (2004), O Aprendizado na Internet Utilizando Estratégias de Roleplaying Game *em* 'Congresso Nacional de Ambientes Hipermídia para Aprendizagem - CONAHPA'.

Antonisse, J. H. (1989), A new interpretation of schema notation that overturns the binary encoding constraint *em* 'Proceedings of the International Conference on Genetic Algorithms'.

Barton, M. (2007), 'Gamasutra - The History of Computer Role-Playing Games', Disponível em: `http://www.gamasutra.com/view/feature/3623/the_history_of_computer_.php`. Último acesso em 27 de janeiro de 2011.

Basile, R. & Magalhaes, L. E. (1975), *Citologia e Genética*, Cultrix.

Battaiola, A. L., Elias, N. C., Domingues, R. d. G., Assaf, R. & Ramalho, G. L. (2002), Desenvolvimento de um Software Educacional com base em Conceitos de Jogos de Computador *em* 'XIII Simpósio Brasileiro de Informática na Educação', SBC, pp. 282–290.

Bittencourt, J. R. & Giraffa, L. M. M. (2003), A Utilização dos Role-Playing Games Digitais no Processo de Ensino-Aprendizagem, Relatório Técnico, Programa de Pós Graduação em Ciência da Computação, Pontifícia Universidade Católica do Rio Grande do Sul (PPGCC/ PUCRS).

Blanchette, J. & Summerfield, M. (2008), *C++ GUI Programming with Qt 4*, 2ª ed., Open Source Software Development Series, Prentice Hall.

Blizzard Entertainment (2010), 'World of warcraft® subscriber base reaches 12 million worldwide', Disponível em: `http://us.blizzard.com/en-us/company/press/pressreleases.html?101007`. Último acesso em 27 de janeiro de 2011.

Bolzan, R. F. F. A. (2003), O Aprendizado na Internet Utilizando Estratégias de Roleplaying Game (RPG), Tese de Doutorado.

Brady, M. & Horn, B. K. P. (1983), 'Rotationally symmetric operators for surface interpolation', *Computer Vision, Graphics, and Image Processing* **22**(1), 70–94.

Cai, X., Li, F., Sun, H. & Zhan, S. (2006), Research of Dynamic Terrain in Complex Battlefield Environments, *em* 'Technologies for E-Learning and Digital Entertainment', Vol. 3942 de *Lecture Notes in Computer Science*.

Canny, J. (1983), Finding edges and lines in images, Technical report, Cambridge, MA, USA.

Canny, J. (1986), 'A Computational Approach to Edge Detection', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-8**(6), 679–698.

Carvalho, L. F. B. S. (2008), Um Jogo de Damas Evolutivo, Trabalho de conclusão de curso, Instituto de Computação, Universidade Federal de Alagoas.

Carvalho, L. F. B. S., Neto, H. C. S., Lopes, R. V. V. & Paraguaçu, F. (2010*a*), Application of a genetic algorithm based on abstract data type in electronic games *em* 'proceedings of the 9th mexican international conference on artificial intelligence - special session - (MICAI 2010)', IEEE Computer Society Press, pp. 28–33.

Carvalho, L. F. B. S., Neto, H. C. S., Lopes, R. V. V. & Paraguaçu, F. (2010*b*), An application of genetic algorithm based on abstract data type for the problem of generation of scenarios for electronic games *em* 'proceedings of the 2nd international conference on intelligent computing and intelligent systems (ICIS 2010)', IEEE Computer Society Press.

Carvalho, L. F. B. S., Neto, H. C. S., Lopes, R. V. V. & Paraguaçu, F. (2010*c*), Applying genetic algorithms based on abstract data type (gaadt) for adaptive evolution of electronic games' scenarios *em* 'anais do v simpósio brasileiro de jogos e entretenimento digital (sbgames 2010)'.

Clark, J. J. (1988), 'Singularity theory and phantom edges in scale space', *IEEE Trans. Pattern Anal. Mach. Intell.* **10**, 720–727.

Clark, J. J. (1989), 'Authenticating edges produced by zero-crossing algorithms', *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, 43–57.

Clua, E. W. G., Junior, C. L. d. L. & Nabais, R. J. d. M. (2002), Importância e Impacto dos Jogos Educativos na Sociedade, *em* 'I Workshop Brasileiro de Jogos e Entretenimento Digital (I WJogos 2002)'.

Crawford, C. (1984), *The Art of Computer Game Design*, Osborne/McGraw-Hill. Disponível em: `http://library.vancouver.wsu.edu/sites/library.vancouver.wsu.edu/files/ACGD.pdf`. Último acesso em 07 de março de 2011.

Crigger, L. (2008), '1UP.com - Chasing D&D: A History of RPGs', Disponível em: `http://www.1up.com/do/feature?cId=3168091`. Último acesso em 26 de janeiro de 2011.

Danielsson, P.-E. & Seger, O. (1990), 'Rotation invariance in gradient and higher order derivative detectors', *Comput. Vision Graph. Image Process.* **49**, 198–221.

Darwin, C. (2002), *A origem das espécies e a seleção natural*, 5ª ed., Hemus. Tradução do orignal de 1859, "On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life".

Dawson, M. (2010), *Beginning C++ Through Game Programming*, 3ª ed., Course Technology PTR.

de Berg, M. & Dobrindt, K. (1995), On levels of detail in terrains *em* 'Proceedings of the eleventh annual symposium on Computational geometry', ACM, pp. 426–427.

De Floriani, L. (1989), 'A pyramidal data structure for triangle-based surface description', *IEEE Comput. Graph. Appl.* **9**, 67–78.

De Floriani, L., Magillo, P. & Puppo, E. (1998), Efficient implementation of multi-triangulations *em* 'proceedings of the conference on visualization '98', IEEE Computer Society Press, pp. 43–50.

Dellepiane, S., D., G. D. & G., V. (1989), Automatic parameter computation for edge detection by the zero-crossing method *em* 'Proceedings of the 12th Conference on Signal Processing and Images (GRETSI)', pp. 617–620. France.

Duchaineau, M., Wolinsky, M., Sigeti, D., Miller, M., Aldrich, C. & Mineev-Weinstein, M. (1997), Roaming terrain: Real-time optimally adapting meshes *em* 'visualization '97., proceedings', pp. 81 –88.

Enterbrain (1992-2011*a*), 'RPG Maker', Disponível em: `http://www.rpgmakerweb.com/product.html?prod=xp`. Último acesso em 19 de dezembro de 2010.

Enterbrain (2004-2011*b*), 'Ruby Game Scripting System (RGSS)', Tutoriais disponíveis em: `http://www.rgss.com.br/rgss/index.html`. Último acesso em 19 de dezembro de 2010.

Fortuna, T. R. (2003), Sala de aula é lugar de brincar? A importância do lúdico no planejamento, *em* 'Planejamento: Análises menos convencionais', 3ª ed., Porto Alegre: Mediação, pp. 127–141.

Freeman, W. T. & Adelson, E. H. (1991), 'The design and use of steerable filters', *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 891–906.

Gardner, H. (1994), *Estruturas da Menta - A Teoria das Inteligências Múltiplas*, Porto Alegre: Artes Médicas Sul.

Garland, M. & Heckbert, P. S. (1997), Surface simplification using quadric error metrics *em* 'proceedings of the 24th annual conference on computer graphics and interactive techniques', ACM Press/Addison-Wesley Publishing Co., pp. 209–216.

Go, C., Basa, T. & Lee, W.-H. (2007), MMORPG Map Evaluation Using Pedestrian Agents, *em* 'Advances in Hybrid Information Technology', Vol. 4413 de *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 323–332.

Goldberg, D., Deb, K. & Korb, B. (1989), 'Messy genetic algorithm: Motivation, analysis, and first results', *Complex Systems* (3), 493–530.

Gould, S. J. (1999), *Darwin e os Grandes Enigmas da Vida*, Martins Fontes.

Gravity Co. Ltd (2002-2011), 'Ragnarok Online', Disponível em: `http://www.levelupgames.com.br/ragnarok/`. Último acesso em 19 de dezembro de 2010.

Green, W. E. (2002), 'Canny edge detection tutorial', Disponível em: `http://www.pages.drexel.edu/~weg22/can_tut.html`. Último acesso em 27 de janeiro de 2011.

Gross, M. H., Gatti, R. & Staadt, O. (1995), Fast multiresolution surface meshing *em* 'proceedings of the 6th conference on visualization '95', IEEE Computer Society, pp. 135–142.

He, Y. (2000), Real-time visualization of dynamic terrain for ground vehicle simulation, PhD thesis.

He, Y., Cremer, J. & Papelis, Y. (2002), Real-time Extendible-resolution Display of On-line Dynamic Terrain, *em* 'Proceedings of the 2002 Conference on Graphics Interface', pp. 151–160.

Hoppe, H. (1996), 'Progressive Meshes', *Computer Graphics* **30**(Annual Conference Series), 99–108.

Hoppe, H. (1997), View-dependent refinement of progressive meshes *em* 'proceedings of the 24th annual conference on computer graphics and interactive techniques', ACM Press/Addison-Wesley Publishing Co., pp. 189–198.

Hoppe, H. (1998), Smooth view-dependent level-of-detail control and its application to terrain rendering *em* 'proceedings of the conference on visualization '98', IEEE Computer Society Press, pp. 35–42.

Hueckel, M. H. (1971), 'An operator which locates edges in digitized pictures', *J. ACM* **18**, 113–125.

Huertas, A. & Medioni, G. (1986), 'Detection of intensity changes with subpixel accuracy using laplacian-gaussian masks', *IEEE Trans. Pattern Anal. Mach. Intell.* **8**, 651–664.

Hunter, D., Rafter, J., Fawcett, J., van der Vlist, E., Ayers, D., Duckett, J., Watt, A. & McKinnon, L. (2007), *Beginning XML (Programmer to Programmer), 4ª ed.*, Wrox.

Hunter, W. (2006), 'The Dot Eaters - Videogame History 101', Disponível em: `http://www.thedoteaters.com/`. Último acesso em 27 de janeiro de 2011.

Ito, K. (2000), Gaussian filter for nonlinear filtering problems *em* 'proceedings of the 39th ieee conference on decision and control', Vol. 2, pp. 1218–1223.

Jenson, J. & Castel, S. (2002), Serious Play: Challenges of Educational Game Design *em* 'Proceedings of the American Research Association Annual Meeting in New Orleans', AERA.

Kernighan, B. W. & Ritchie, D. M. (1988), *C Programming Language*, 2ª ed., Prentice Hall.

King, M. C. & Wilson, A. C. (1975), *Science* (4184), 107–116.

Lindeijer, T. & Turk, A. (2006-2011), 'Tiled Map Editor', Disponível em: `http://www.mapeditor.org/`. Último acesso em 19 de dezembro de 2010.

Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L., Faust, N. & Turner, G. (1996), 'Real-Time Continuous Level of Detail Rendering of Height Fields', *Proceedings of SIGGRAPH'96* pp. 109–118.

Lindstrom, P. & Pascucci, V. (2001), Visualization of large terrains made easy *em* 'Proceedings of the conference on Visualization '01', IEEE Computer Society, pp. 363–371.

Lopes, M. A. (1981), Introdução a uma Teoria Geral de Problemas, Tese de Doutorado, Pontifícia Universidade Católica do Rio de Janeiro.

Lopes, R. V. V. (2003), Um Algoritmo Genético Baseado em Tipos Abstratos de Dados e sua Especificação em Z, Tese de Doutorado, Centro de Informática, Universidade Federal de Pernambuco.

Luebke, D. & Erikson, C. (1997), View-dependent simplification of arbitrary polygonal environments *em* 'siggraph '97: Proceedings of the 24th annual conference on computer graphics and interactive techniques', ACM Press/Addison-Wesley Publishing Co., pp. 199–208.

Lyvers, E. P. & Mitchell, O. R. (1988), 'Precision edge contrast and orientation estimation', *IEEE Trans. Pattern Anal. Mach. Intell.* **10**, 927–937.

Marcatto, A. (1996), *Saindo do Quadro: Uma metodologia educacional lúdica e participativa baseada no Role Playing Game*, 2ª ed., Edição do autor.

Marr, D. & Hildreth, E. (1980), 'Theory of edge detection.', *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)* **207**(1167), 187–217.

Martins, L. A. (2000), A Porta do Encantamento: Jogos de Representação nas Perspectivas da Socialização e da Educação, Dissertação de Mestrado, Faculdade de Educação, Universidade Estadual de Campinas .

Matsumoto, Y. (1995-2011), 'Ruby', Tutoriais disponíveis em: `http://www.ruby-lang.org/pt/`. Último acesso em 19 de dezembro de 2010.

Oracle (1995-2011), 'Java SE', Disponível em: `http://www.oracle.com/technetwork/java/javase/overview/index.html`. Último acesso em 19 de dezembro de 2010.

Pereira, C. E. K., Andrade, F. M. & Freitas, L. E. R. (1992), *Desafio dos Bandeirantes - Aventuras na Terra de Santa Cruz*, GSA.

Petzold, C. (2001), *Programming Windows with C# (Core Reference).*

Pitas, I. & Venetsanopoulos, A. N. (1986), 'Edge detectors based on nonlinear filters', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(4), 538–550.

Poggio, T. & Torre, V. (1984), Ill-posed problems and regularization analysis in early vision, Technical report.

Poggio, T., Voorhees, H. & Yuille, A. (1988), 'A regularized solution to edge detection', *J. Complex.* **4**, 106–123.

Ponce, J. & Brady, M. (1985), Towards a Surface Primal Sketch, Technical Report AIM-824, Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Reed, A. (2008), *Learning XNA 3.0: XNA 3.0 Game Development for the PC, Xbox 360, and Zune*, O'Reilly media.

Roman, S. (1995), *Field Theory (Graduate Texts in Mathematics)*, Springer-Verlag.

Rossignac, J. & Borrel, P. (1993), *Multi-resolution 3D approximations for rendering complex scenes em 'Modeling in Computer Graphics: Methods and Applications'*, Springer-Verlag.

Sastry, K. & Goldberg, D. E. (2002), How well does a single-point crossover mix building blocks with tight linkage?, Relatório Técnico Nº 2002013, Illinois Genetic Algorithms Laboratory, University of Illinois.

Scarlatos, L. & Pavlidis, T. (1992), 'Hierarchical triangulation using cartographic coherence', *CVGIP: Graph. Models Image Process.* **54**, 147–161.

Schaffer, J. D. (1985), Learning multiclass pattern discrimination *em* 'Proceedings of the International Conference on Genetic Algorithms'.

Schroeder, W. J., Zarge, J. A. & Lorensen, W. E. (1992), Decimation of triangle meshes *em* 'proceedings of the 19th annual conference on computer graphics and interactive techniques', SIGGRAPH '92, ACM, pp. 65–70.

Schwab, B. (2004), *AI Game Engine Programming*, 1ª ed., Charles River Media.

Shamir, A., Pascucci, V. & Bajaj, C. L. (2000), Multiresolution Dynamic Meshes with Arbitrary Deformations, *em* 'Proceedings of IEEE Visualization 2000', pp. 423–430.

Shen, J. & Castan, S. (1992), 'An optimal linear operator for step edge detection', *CVGIP: Graph. Models Image Process.* **54**, 112–133.

Silva, M. (2001), *Sala de Aula Interativa*, Quartet.

Sumner, R. W., O'Brien, J. F. & Hodgins, J. K. (1999), 'Animating Sand, Mud, and Snow', *Computer Graphics Forum* **18**(1).

Tabbone, S. (1994), Edge detection, subpixels and junctions using multiple scales, PhD thesis, Institut National Polytechnique de Lorraine, França.

Tabbone, S. & Ziou, D. (1991), Elimination of false edges by threshold separation and propagation of thresholds *em* 'Proceedings of the 13th Conference on Signal Processing and Images (GRETSI)', pp. 1025–1028. France.

Tabbone, S. & Ziou, D. (1993), Efficient edge detection using two scales*em* 'proceedings of the 1993 conference on computer vision and pattern recognition (cvpr '93)', pp. 789 –790.

The Mana World Dev Team (2004-2011), 'The Mana World - A Free Open Source 2D MMORPG in Development', Disponível em: `http://themanaworld.org/`. Último acesso em 19 de dezembro de 2010.

Tobaldini, M. A. & Brancher, J. D. (2006), Um RPG Educacional Computadorizado e Missões Contextualizadas com seus Ambientes, *em* 'Anais do XV Seminário de Computação', pp. 85–96.

Tobaldini, M. A., Martinelli, M. & Brancher, J. D. (2006), Arquiteturas Históricas no Ambiente de um Jogo de RPG Educacional Computadorizado, *em* 'Anais do V Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGAMES 2006)', pp. 85–96.

Torre, V. & Poggio, T. (1986), 'On edge detection', *IEEE Trans. Pattern Anal. Mach. Intell.* **8**, 147–163.

Winter, D. J. (1974), *The structure of fields (Graduate Texts in Mathematics)*, Springer-Verlag.

Wu, L. & Xie, Z. (1990), 'Scaling theorems for zero-crossings', *IEEE Trans. Pattern Anal. Mach. Intell.* **12**, 46–54.

Xia, J. C. & Varshney, A. (1996), Dynamic view-dependent simplification for polygonal models *em* 'proceedings of the 7th conference on visualization '96', IEEE Computer Society Press, pp. 327–334.

Ziou, D. & Fabre, J.-P. (1994), 'Effects of edge orientation on the performance of first-order operators', *Pattern Recogn. Lett.* **15**, 1053–1063.

Ziou, D. & Tabbone, S. (1993), 'A multi-scale edge detector', *Pattern Recognition* **26**(9), 1305–1314.

Ziou, D. & Tabbone, S. (1995), Adaptive elimination of false edges for first order detectors *em* 'proceedings of the 8th international conference on image analysis and processing', Springer-Verlag, pp. 89–94.

Ziou, D. & Tabbone, S. (1998), 'Edge Detection Techniques - An Overview', *International Journal of Pattern Recognition and Image Analysis* **8**, 537–559.

Ziou, D. & Wang, S. (1996), Isotropic processing for gradient estimation *em* 'proceedings of the 1996 conference on computer vision and pattern recognition (cvpr '96)', IEEE Computer Society, pp. 660–665.

Zuchi, I. (2000), O Desenvolvimento de um Protótipo de Sistema Especialista baseado em Técnicas de RPG para o Ensino da Matemática, Dissertação de Mestrado, Universidade Federal de Santa Catarina - UFSC.

Apêndice A

Considerações sobre as implementações iniciais do GAADT para o problema

O trabalho que foi apresentado ao longo desta dissertação é o produto de um refinamento de uma abordagem que teve sua primeira versão apresentada em (Carvalho et al., 2010*a*). Deste ponto em diante a metodologia aplicada ao problema exposto na seção 1.2 foi revisada e modificada até alcançar a forma que foi apresentada pelos capítulos desta dissertação. Um fato que implicou não apenas na melhor definição dos termos do GAADT, mas que contribuiu também para a melhor formalização do problema, permitindo assim sua definição tal como apresentada na seção 3.3.

Ainda que uma leitura de Carvalho et al. (2010*a*), Carvalho et al. (2010*b*) e Carvalho et al. (2010*c*) possa esclarecer de forma mais ampla as etapas pelas quais este trabalho passou até atingir seu formato atual, serão apresentados agora os pontos principais que diferenciam cada um destes trabalhos daquele que foi apresentado aqui, e quais as constatações que foram obtidas de cada uma delas, incluindo seus resultados.

Com isto, o que se espera neste momento é capacitar o leitor a traçar um perfil comparativo entre as diferentes perspectivas que foram adotadas na abordagem do problema apresentado por este trabalho e entre os resultados gerados por cada uma delas.

A.1 A primeira implementação

As primeiras tentativas de se aplicar o GAADT para o problema descrito em 1.2 resultaram no trabalho apresentado em Carvalho et al. (2010*a*). Esta primeira implementação já possuía como característica a adoção de uma população formada por cromossomos com cadeias de tamanho variável. Ao mesmo tempo, esta versão contrasta com aquela apresentada por esta dissertação por adotar suas próprias restrições na construção de seus conjuntos de bases, genes

e cromossomos, bem como, por admitir sua própria interpretação para a composição destes elementos.

Esta implementação já permitia o discernimento entre os diferentes papeis que podem ser assumidos por cada elemento geográfico do mapa de jogo. Entretanto, esta informação não estava vinculada aos elementos do conjunto base, os quais eram definidos simplesmente como compostos por *tiles* passiveis de utilização na construção do cenário. Tal afirmação equivale a dizer que o conjunto de bases é formado por todos os *tiles* contidos nos *tilesets* associados ao mapa de jogo dado como entrada para o algoritmo, os quais, são ainda associados a coordenadas espaciais x e y que determinam seu posicionamento.

A diferença entre as coordenadas x e y adotadas nesta primeira implementação e aquela que foi apresentada na equação 4.2 é que ao passo em que a última armazena as coordenadas x e y de uma base em relação aos respectivos eixos cartesianos do mapa de jogo, aqui, o eixo foi deslocado para a origem do gene em que esta base se encontra alocada, correspondendo portanto a sua localização dentro de um gene em particular. O conjunto base desta implementação é então dado por um conjunto $B = Tile \cup \mathbb{N}$ que é composto por elementos b tais que $(b = (tl_{x,y}) \in B) \leftrightarrow ((tl \in Tile) \wedge (x, y \in \mathbb{N}))$.

A definição dos axiomas de formação de genes e de cromossomos definidos para esta primeira instanciação do algoritmo também eram menos específicas que suas versões atuais. Neste sentido, o AFG simplesmente definia um gene como uma seqüência de bases que permite a repetição de elementos, enquanto que o AFC define um cromossomo como um conjunto de genes que não exceda o número de máximo de *tiles* que podem ser alocados no mapa de jogo dado como entrada para o algoritmo (cujo valor equivaleria ao resultado da equação 3.23 apresentada nesta dissertação). A instância do GAADT utilizada neste momento também apresenta uma condição de parada mais simples que a da versão atual do algoritmo, estando vinculada apenas a um número máximo de iterações.

A equação utilizada para o cálculo do grau do gene ainda não estava bem definida, de maneira que a função de adaptação do cromossomo assumia o papel de definir o valor da adaptação de um cromossomo em relação a seu ambiente com base em seus próprios cálculos, os quais, não isolavam de forma adequada o grau de um gene. Esta carência foi o que motivou a construção da primeira versão do módulo de descrição de *tiles* apresentado na seção 5.2.1, que foi originalmente concebido com o propósito de auxiliar a realização destes cálculos.

Neste primeiro momento, os elementos que formavam uma mesma imagem eram "montados" em uma única imagem dentro do módulo de descrição de *tiles*. Durante a execução do algoritmo, a função de adaptação iria então avaliar a correta disposição dos elementos do mapa sendo modificado. Para este propósito, o algoritmo tomava como base o posicionamento destes elementos em suas respectivas imagens "montadas" dentro do módulo de descrição de *tiles*, levando ainda em consideração, a possibilidade, ou não, da repetição destes elementos em uma mesma imagem. Informação esta que também era inserida no módulo.

É possível ainda afirmar que, ainda que os axiomas para a formação de genes e de cromosso-

mos não fossem suficientemente restritivos, os módulos encarregados de fazer o corte do mapa de jogo, a montagem dos genes e a dos cromossomos já exibiam uma preocupação em elaborar estes elementos por meio da separação do mapa de jogo em regiões. Para atender a esta necessidade, estes módulos aplicavam um conceito de afinidade que é similar ao atualmente aplicado pelos procedimentos apresentados nas seções 5.4 e 5.5.

A Figura A.1 ilustra o módulo para o cadastro das descrição de *tiles* nesta etapa do desenvolvimento, cuja parte central esquerda exibe a "montagem" de uma árvore formada por diferentes *tiles*, com as opções que podem ser vinculadas a cada um destes *tiles* (destacados pelo fundo laranja) sendo exibidas na parte central direita. Já o cálculo responsável por determinar o valor de adaptação de um cromossomo é apresentado pela equação A.1.

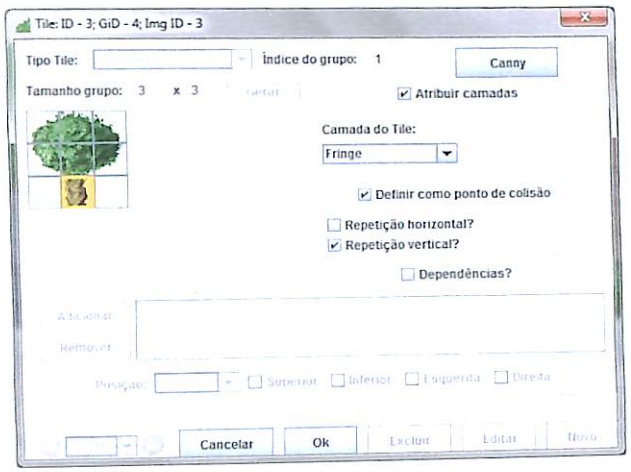


Figura A.1: *Modulo de inserção de descrições exibindo uma das descrição de tiles cadastradas para a primeira implementação do trabalho desta dissertação (tal como consta em Carvalho et al. (2010a))*

$$
\begin{aligned}
adapt(c) = \{ & \forall c = \{g_1, \ldots, g_n\} \in C \; \forall g = \langle b_1, \ldots, b_m \rangle \in \{g_1, \ldots, g_n\} \\
& \forall b \in \{b_1, \ldots, b_m\} \; \exists adapt \in \mathbb{Q}_+ | \\
& (tl \in Tile) \wedge (x, y \in \mathbb{N}) \wedge (adapt = \sqrt{\textstyle\sum_{i=1}^{n} \sum_{j=1}^{m} abs(harm(g_i, b_j, j))})\}
\end{aligned}
\tag{A.1}
$$

O cálculo da equação A.1 baseia-se no conceito de que, uma vez que seja determinada uma qualidade sendo buscada, deve-se verificar se esta qualidade está adequadamente disposta no cenário de jogo. Portanto, é necessário verificar se as características sendo inseridas se encontram em harmonia com suas circunvizinhanças, ou, se apresentam uma aparência deslocada em relação ao restante do cenário. O cálculo deve também quantificar a freqüência com que esta característica se manifesta. É com estes propósitos que a equação A.1 emprega de duas outras funções que auxiliam na determinação de seu resultado, expressas pelas funções *harm* da equação A.2 e *abs* da equação A.3.

A função *harm* da equação A.2 é utilizada para determinar se as bases de um gene estão dispostos de forma adequada. Para isto, a equação toma como base a disposição dos elementos

no módulo da Figura A.1. A partir desta disposição a equação assume um elemento $\overline{b}$ como uma base que deve ser alocada em uma posição j do gene sendo avaliado, permitindo-lhe assim identificar áreas problemáticas na construção do mapa de jogo (correspondentes a diferentes genes do GAADT) e aplicar as devidas correções quando necessário.

$$harm(g = \langle b_1, \ldots, b_m \rangle, b_j) = \begin{cases} repl(g, b_j, \overline{b}) & se \quad \forall \overline{b} \in B \\ & \quad \nexists b_j \in \{b_1, \ldots, b_m\}(\overline{b} = b_j) \\ b & se \quad \forall \overline{b} \in B \\ & \quad \exists b_j \in \{b_1, \ldots, b_m\}(\overline{b} = b_j) \end{cases} \tag{A.2}$$

Já a função *abs* apresentada na equação A.1 visa determinar se cada uma das bases que lhe é fornecida como entrada apresenta uma dada qualidade sendo buscada. Os valores de saída desta operação são então somados e a raiz do valor resultando é então devolvido como o valor final da função *adapt*, que atribui este valor como a adaptação de seu cromossomo de entrada. A equação A.3 exibe as operações realizadas pela função *abs*.

$$abs(b) \begin{cases} 1 & \text{se a base exibe a qualidade desejada} \\ 0 & \text{se a base não exibe a qualidade desejada} \end{cases} \tag{A.3}$$

Resultados de Carvalho et al. (2010*a*)			
geração	adaptação média	pior adaptação	melhor adaptação
1	0,0	0,0	0,0
2	0,6	9,64	0,0
3	1,7	10,15	0,0
4	4,15	11,36	0,0
5	9,71	13,45	7,07
6	13,02	13,45	12,57
7	13,74	15,33	13,45
8	15,74	17,72	15,17
9	18,06	19,67	17,52
10	20,02	22,07	19,65
11	21,91	25,42	21,05
12	24,59	27,39	22,8
13	26,99	27,4	26,36
14	27,48	28,77	27,39
15	28,12	28,77	27,4

Tabela A.1: *Tabela de dados de cada geração para uma execução do GAADT de Carvalho et al. (2010a)*

A Tabela A.1 apresenta os resultados de uma execução desta versão GAADT para os cromossomos mais e menos adaptados de cada geração, bem como, a média de adaptação de cada uma destas populações. Como esta versão foi um teste inicial, optou-se por ao final do processo, substituir todos os cromossomos da população final no mapa de entrada a fim de melhor visualizar as diferentes alterações geradas pelo processo evolutivo. A Figura A.2 ilustra os mapas de entrada e saída para esta instância do GAADT.

(a) *Mapa de entrada (floresta)*

(b) *Mapa de saída (deserto)*

Figura A.2: *Comparação de um mapa de entrada e o mapa de saída gerado a partir dele para a primeira implementação do GAADT desta dissertação (equivalente ao da implementação de Carvalho et al. (2010a))*

Os resultados desta primeira versão despertaram o entendimento de que havia a necessidade de uma maior vinculação entre os registros de descrições alimentados pelo sistema através do módulo de descrição de *tiles*. Os resultados obtidos também permitiram perceber que uma avaliação da correta disposição dos elementos geográficos do cenário, com base simplesmente no que o usuário deveria fornecer como "seu desenho ideal" por meio da montagem que fazia no módulo de descrição de *tiles*, não apenas geravam resultados pouco satisfatórios, como também, eram desnecessariamente complexos, em razão do elevado número de condicionais que deve-

riam ser verificadas pelos componentes da função de adaptação a fim de determinar a correta disposição destes elementos.

Estas conclusões levaram a revisão do algoritmo e ao desenvolvimento de novas instanciações, as quais procuraram dar maior enfoque a uma melhor definição dos tipos base, gene e cromossomo e a determinação da correta disposição dos elementos geográficos por meio da análise do formato de suas imagens através da avaliação de seus pixels, o que conseqüentemente levaria a adoção de algoritmos para a detecção de bordas.

A.2 Implementações posteriores

A partir do conhecimento adquirido com a primeira instanciação do GAADT ao problema proposto por esta dissertação, a qual foi apresentada na seção anterior, a instanciação do algoritmo passou um refinamento que consistiu da redefinição de seus tipos base, gene e cromossomo, bem como, de sua função de adaptação. Uma instanciação que está disponível em Carvalho et al. (2010*b*).

Neste segundo momento, o que se buscou foi promover uma maior integração entre os tipos básicos do GAADT e as informações que poderiam ser providas por meio do módulo de descrição de *tiles*. Para este propósito, o conjunto base foi então definido como sendo composto pelas coordenadas espaciais x, y e z equivalentes aos pontos cartesianos que identificam as células da malha do mapa de entrada, unidas a um conjunto *Desc*, o qual, comporta todos os registros alimentados no sistema através do módulo de descrição de *tiles* e é o responsável por identificar as características dos *tiles* do mapa de jogo. Desta forma, o conjunto de bases pode então ser expresso como $B \cup Desc \cup b_\lambda$, cujos elementos são dados por $b = (x,y,z) \in B$, $d \in Desc$ e com b_λ como a base-inócua.

De maneira similar, a composição dos genes foi alterada com os genes g do algoritmo passando a ser definidos de forma que $g = (b_1 = (x_1,y_1,z_1), b_2 = (x_2,y_2,z_2), d_{b_1})$. Esta representação assume o gene como uma seção do mapa de jogo que compreende a área retangular delimitada entre as bases b_1 e b_2, e com o elemento d_{b_1} correspondendo a descrição do objeto *tile* localizado no mapa de jogo na coordenada indicada pela base b_1. Cabe ainda afirmar que, os genes-inócuos são aqueles que possuem um tipo de descrição de *tile* particular associada a eles e que é dada por d_λ. Logo, $\{\exists d_\lambda \in Desc | \, \forall g = (b_1,b_2,d)((d = d_\lambda) \rightarrow (g = g_\lambda))\}$

Definida a estrutura desejada para os genes, procurou-se então oferecer uma maior restrição a sua composição, reforçando assim os critérios estabelecidos pelo conjunto de axiomas de formação de genes que passou então a definir que não poderia haver em um mesmo gene duas bases que expressassem coordenadas espaciais x, y e z idênticas. Desta forma, para qualquer gene $g = (b_1(x_1,y_1,z_1), b_2 = (x_2,y_2,z_2), d)$ é assumido que $x_1 < x_2$, $y_1 < y_2$ e $z_1 \leq z_2$.

Esta definição do conjunto de AFG permite inferir que, em contraste com a instanciação apresentada por este trabalho, o AFG não limita um gene a uma única camada do mapa de jogo

(uma única coordenada z) e que, sequer estabelece uma continuidade para sua proliferação entre as diferentes camadas que pode acessar, tolerando assim não apenas a sua potencial propagação por diferentes níveis do mapa de jogo como a sua potencial descontinuidade.

Além destas características, esta segunda instância do GAADT apresenta cromossomos c compostos por conjuntos de genes tais que $c = \{g_1, \ldots, g_n\}$. O conjunto AFC definido para estes cromossomos é similar ao conjunto AFG instanciado para o algoritmo, no sentido de que estabelece que não é permitido a um cromossomo possuir dois ou mais genes compartilhando de uma mesma coordenada espacial x, y e z em suas bases.

Este conjunto de axiomas define ainda que os valores atribuídos as coordenadas espaciais acima citadas devem ser maiores ou iguais a zero e menores que a dimensão do mapa de jogo de seu respectivo eixo de coordenadas. A definição destes conceitos permaneceu pertinente mesmo na instância atual do algoritmo, para a qual, de acordo com esta analogia, o primeiro destes conceitos é garantido pela própria definição do conjunto base e do seu uso implícito dos conjuntos definidos pelas equações 3.4, 3.5 e 4.3, enquanto que o último deles é garantido pelo axioma que está expresso na equação 4.9.

Foi também nesta segunda instanciação do algoritmo que a função responsável por determinar o grau de um gene começou a ganhar um formato mais próximo daquele proposto pela definição do GAADT. Contudo, diferente do que ocorre na instanciação atual, ainda não havia o emprego de uma abordagem que considera como parte de sua avaliação a verificação do contorno da imagem que o caracteriza, mantendo ao invés disso uma abordagem similar a apresentada na seção A.1 que busca quantificar a freqüência de uma determinada característica desejada.

Por esta razão, a função de grau do gene era escolhida de acordo com a característica que se deseja avaliar. Tal afirmação pode ser mais bem entendida retomando o exemplo de um mapa de jogo que passa a adquirir características desérticas. Nesta situação em particular, a função *grau* possui então a responsabilidade de quantificar a qualidade "areia" presente no gene dado como entrada para a equação. Esta função é então recursivamente executada a fim de identificar quais das bases do gene realmente apresentam a característica desejada e é definida como $grau(g) = areia(g) = areia(b_1, b_2, d_{b_1}) = areia((x_1, y_1, z_1), (x_2, y_2, z_2), d_{(x_1, y_1, z_1)})$.

A avaliação de outras características de cenário que não a da qualidade "areia" seria análoga às operações realizadas por esta função, que é apresentada aqui pela equação A.4, diferenciando-se apenas no tipo de característica que está sendo buscada.

A relação que a função *grau* compartilha com a função *areia* da equação A.4 (e com as demais funções que se disponham a quantificar uma qualidade particular que possa estar presente nos genes do algoritmo) permite instanciar a função de adaptação do cromossomo de forma semelhante a que é adotada pela instância do GAADT desta dissertação.

$$areia((x,y,z),(\bar{x},\bar{y},\bar{z}),d) = \begin{cases} 1 + areia((a,y,z),(\bar{x},\bar{y},\bar{z}),d_{(a,y,z)}) & \text{se } (a = x+1 < \bar{x}) \wedge \\ & (\text{“areia”} \in d_{(a,y,z)}) \\ 0 + areia((a,y,z),(\bar{x},\bar{y},\bar{z}),d_{(a,y,z)}) & \text{se } (a = x+1 < \bar{x}) \wedge \\ & (\text{“areia”} \notin d_{(a,y,z)}) \\ 1 + areia((x,e,z),(\bar{x},\bar{y},\bar{z}),d_{(x,e,z)}) & \text{se } (e = y+1 < \bar{y}) \wedge \\ & (\text{“areia”} \in d_{(x,e,z)}) \\ 0 + areia((x,e,z),(\bar{x},\bar{y},\bar{z}),d_{(x,e,z)}) & \text{se } (e = y+1 < \bar{y}) \wedge \\ & (\text{“areia”} \notin d_{(x,e,z)}) \\ 1 & \text{se } (x = \bar{x}) \wedge (y = \bar{y}) \wedge (\text{“areia”} \in d_{(x,y,z)}) \\ 0 & \text{se } (x = \bar{x}) \wedge (y = \bar{y}) \wedge (\text{“areia”} \notin d_{(x,y,z)}) \end{cases} \tag{A.4}$$

O cálculo da função de adaptação desta segunda instanciação do GAADT está expresso na equação A.5. A diferença entre esta função de adaptação e a que é adotada pela atual instância do algoritmo consiste unicamente no que diz respeito às operações específicas que são realizadas pelas funções associadas a cada uma delas e pelo fato de que, os resultados retornados pela equação A.5 correspondem a raiz do valor obtido pelo somatório realizado por ela, um traço que foi herdado da primeira instanciação do algoritmo e que não foi conservado em sua versão atual.

$$adapt(c = \{g_1, \ldots, g_n\}) = \sqrt{\sum_{i=1}^{n} grau(g_i)} \tag{A.5}$$

Esta segunda instanciação do GAADT apresentou resultados melhores que os da sua segunda instanciação. Entretanto, também foi constatado um elevado tempo de convergência para cada uma das iterações do algoritmo, as quais são responsáveis por criar as novas gerações de cromossomos. O gráfico que ilustra o processo de convergência desta implementação do algoritmo para três de suas execuções independentes e que é equivalente ao encontrado em Carvalho et al. (2010*b*) está replicado na Figura A.3, que identifica ainda a evolução dos cromossomos mais e menos adaptados de cada geração, bem como a evolução da média de adaptação das populações geradas.

Antes de alcançar o formato apresentado por esta dissertação, o algoritmo passou ainda por uma terceira revisão, na qual seu tipo gene foi mais uma vez redefinido. Uma característica que, conseqüentemente, repercutiu em alterações na definição de seus axiomas de formação de genes. Neste sentido, o tipo gene passou então a ser definido como $g = (b_1(x_1, y_1, z_1, d_1), b_2 = (x_2, y_2, z_2, d_2))$, uma afirmação que indica que também houve uma alteração no tipo base.

Mantendo-se as definições de variáveis estipuladas na segunda implementação do algoritmo, e assumindo as novas variáveis d_1 e d_2 tais que $d_1 \in Desc$ e $d_2 \in Desc$. Esta abordagem visou

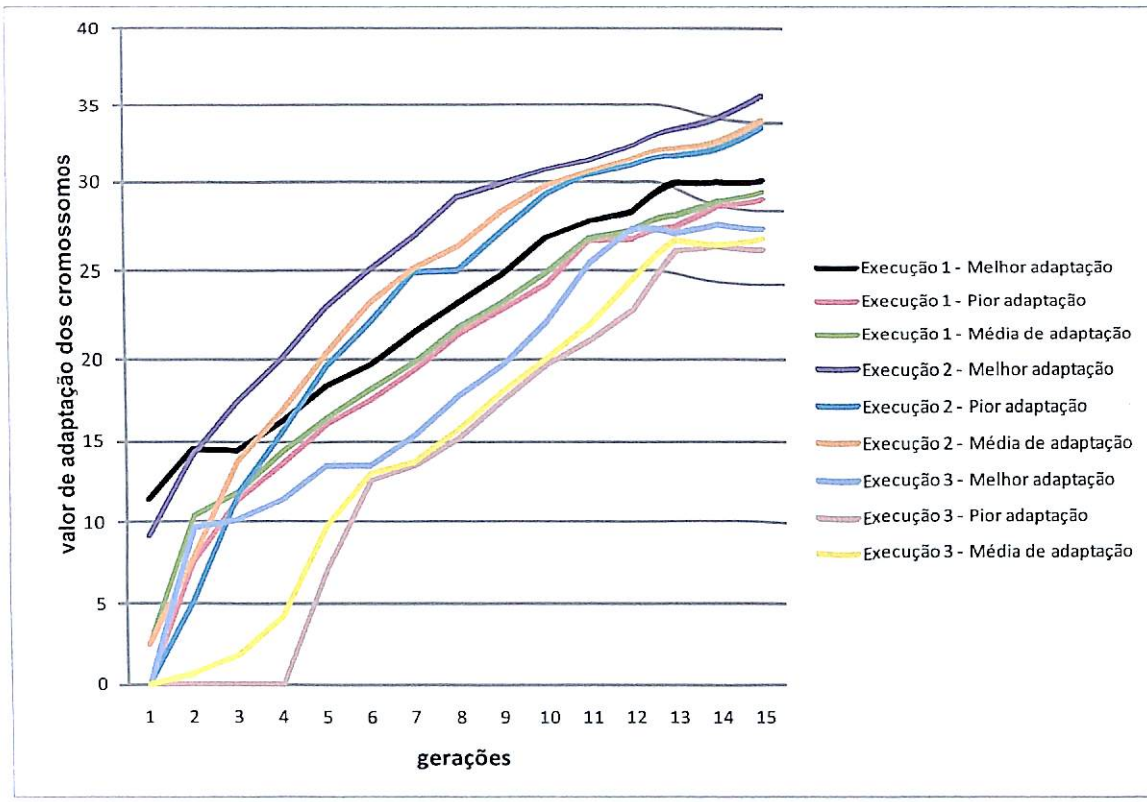


Figura A.3: *Gráfico para a adaptação dos cromossomos para três execuções distintas do GAADT*

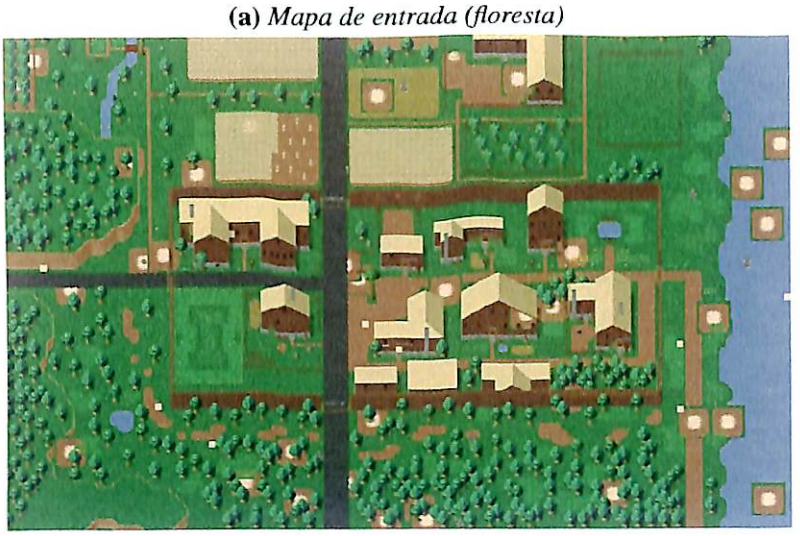
ampliar o conhecimento acerca de cada célula do mapa de jogo.

De acordo com esta nova perspectiva, um conjunto de bases (cada qual representando uma célula do algoritmo) não estaria mais vinculado a uma única descrição d. Ao contrário, cada base b passou a ser vinculada a sua própria descrição, uma medida que se fez necessária dado que, esta terceira instanciação, também não limitava a formação de um gene a uma única característica geográfica do mapa de entrada. Esta terceira instanciação está disponível em Carvalho et al. (2010*c*) e tem seus resultados ilustrados aqui pela Figura A.4 que exibe mapas de entrada e saída equivalentes a esta instância do GAADT.

Apesar da nova estrutura que foi dada as bases, genes e ao conjunto de AFGs desta implementação, os resultados obtidos após as alterações foram equivalentes aqueles oriundos da segunda instanciação do algoritmo. Esta constatação levou então a se optar por uma completa reestruturação da instância do algoritmo, que procurou manter a idéia geral da abordagem ao mesmo tempo em que modifica a forma em que se dá sua aplicação, buscando com isto aplicar o conhecimento aprendido a partir das implementações anteriores.

Tal decisão implicou na remodelação do módulo para armazenamento das descrições de *tiles*, simplificando seu formato e diminuindo a quantidade de informações a serem armazenadas, fazendo com isto que estas informações se tornassem de propósito mais geral. A decisão de generalizar estas informações foi influenciada pelo entendimento de que haveriam formas mais adequadas de identificar a correta disposição dos elementos geográficos de cenário, que

(a) *Mapa de entrada (floresta)*



(b) *Mapa de saída (deserto)*

Figura A.4: *Comparação de um mapa de entrada e o mapa de saída gerado a partir dele para a primeira implementação do GAADT desta dissertação (tal como consta em Carvalho et al. (2010c))*

são identificados pelos tipos base, gene e cromossomo. Necessidade esta que já havia sido identificada nas instanciações anteriores do algoritmo e que pode então ser suprida por meio da incorporação de um algoritmo para detecção de bordas como parte da função do grau de um gene.

UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa Multidisciplinar de Pós-Graduação em Modelagem Computacional de Conhecimento
Avenida Lourival Melo Mota, Km 14, Bloco 09, Cidade Universitária
57.072-900 Maceió AL Brasil CGC: 24.464.109/0001-48
Telefone: (082) 3214-1364

MODELAGEM COMPUTACIONAL DE CONHECIMENTO

Ata da defesa de dissertação do aluno
Leonardo Filipe Batista Silva de Carvalho

Realizou-se no dia 28 de março de dois mil e onze, a partir das 14h30min, na sala de aula do Programa de Pós-Graduação em Modelagem Computacional de Conhecimento da Universidade Federal de Alagoas, a defesa de dissertação de Mestrado em Modelagem Computacional de Conhecimento, intitulada "Aplicação do modelo de Algoritmo Genético baseado em Tipos Abstratos de Dados (GAADT) na adaplação de cenários bidimensionais de MMORPGs", apresentada por Leonardo Filipe Batista Silva de Carvalho, graduado em Ciência da Computação, como requisito parcial para a obtenção do grau de Mestre em Modelagem Computacional de Conhecimento, perante a seguinte comissão examinadora:

Professora e orientadora Roberta Vilhena Vieira Lopes
Instituto de Computação – UFAL

Professor e co-orientador Fábio Paraguaçu Duarte da Costa
Instituto de Computação – UFAL

Professor Manoel Agamemnon Lopes
Centro de Ciências Agrárias – UFAL

Professor Geber Lisboa Ramalho
Departamento de Sistemas de Computação – UFPE

PROGRAMA DE PÓS GRADUAÇÃO – MODELAGEM COMPUTACIONAL DE CONHECIMENTO – Universidade Federal de Alagoas

Confere com o original.
05/12/11

A dissertação foi considerada aprovada pela Comissão. Finalizados os trabalhos, às 16h30min, lavrou-se a presente ata, que vai assinada por mim e pelos membros da Comissão.

Maceió, 28 de março de 2011.

José Vitor de Menezes Torres
Secretário

Roberta Vilhena Vieira Lopes
Roberta Vilhena Vieira Lopes
Doutora em Ciência da Computação – Universidade Federal de Pernambuco/Brasil

Fábio Paraguaçu Duarte da Costa
Doutor em Informática – Universite d'Aix-Marseille III (Droit, Econ. et Sciences)/França

Manoel Agamemnon Lopes
Doutor em Informática – Pontifícia Universidade Católica do Rio de Janeiro/Brasil

Geber Lisboa Ramalho
Doutor em Informatique – Universite Pierre et Marie Curie (Paris VI)/França

DOCUMENTO RECEBIDO
Em: 05 / 03 / 2012
Assinatura do Funcionário