

FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

LAURA MATTOS FLACH

MODELAGEM PROCEDURAL DE GRUPOS EM AMBIENTES INTERNOS

Porto Alegre

2013

PÓS-GRADUAÇÃO - *STRICTO SENSU*



Pontifícia Universidade Católica
do Rio Grande do Sul

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**MODELAGEM PROCEDURAL DE GRUPOS
EM AMBIENTES INTERNOS**

LAURA MATTOS FLACH

Dissertação de Mestrado apresentada como requisito para obtenção do título de Mestre em Ciência da Computação pelo Programa de Pós-graduação da Faculdade de Informática. Área de concentração: Ciência da Computação.

Orientadora: Profa. Dra. Soraia Raupp Musse

Porto Alegre, Brasil
2013

MODELAGEM PROCEDURAL DE GRUPOS EM AMBIENTES INTERNOS

Resumo

Este trabalho apresenta a utilização de modelagem procedural para povoar ambientes com agentes virtuais. São utilizados ambientes previamente gerados também de forma procedural, juntamente com um simulador de multidões para realizar a locomoção dos agentes. O trabalho, além de criar grupos que se baseiam em informações semânticas do ambiente a ser povoado e em dados retirados do IBGE em relação à população de Porto Alegre, gera comportamentos para cada membro do grupo. Estes comportamentos gerados são coerentes com o ambiente, às características do grupo e do próprio membro, e ao tempo. Todas estas características e especificações são transformadas em *seeds*, o que permite a persistência dos dados (possibilita que o contexto seja gerado novamente a qualquer momento sem a utilização de estruturas de dados) e facilita a execução (o modelo é totalmente automático).

Palavras-chave: Modelagem Procedural, Animação Comportamental, Grupos.

PROCEDURAL MODELING OF GROUPS IN INTERNAL ENVIRONMENTS

Abstract

This work presents the utilization of procedural modelling to populate environments with virtual agents. It uses previously procedurally generated environments together with a crowd simulator to generate the agents locomotion. This work creates groups based on semantic informations of the environment to be populated and on data retrieved from IBGE for the population of Porto Alegre. It also generates behaviour for each member of the group created. These behaviours are coherent to the environment, to the group's and member's characteristics and also to time. All of these characteristics and specifications are transformed into seeds, wich makes the persistence of the data possible (the possibility of the regeneration of the context at any moment without the use of data structures) and the execution more simple (the model is fully automatic).

Keywords: Procedural Modeling, Behaviour Animation, Groups.

Lista de Figuras

2.1	Exemplos de <i>boids</i> [33].	18
2.2	Multidão contínua formando linhas e vortex [39].	19
2.3	Representação de cinco agentes e seus espaços pessoais. Agentes representados pela mesma cor de seus respectivos marcadores [6].	20
2.4	Os marcadores não caminháveis são usados no terreno para definir um lago. Em (A) é apresentado o ambiente de simulação, onde é possível observar os marcadores e marcadores especiais; e em (B) a mesma situação renderizada durante a visualização [6].	21
2.5	Evolução da trajetória de um grupo de agentes levando-se em conta os marcadores especiais, partindo de S com o destino T [6].	22
2.6	Grafo utilizado para o A*. A região 1 representa marcadores semi-caminháveis e a 2 representa uma região não caminhável, portanto sem marcadores [6].	22
2.7	Exemplo de trajetória utilizando o A* [6].	23
2.8	Exemplo de grafo de movimento retirado de [38].	25
2.9	Em (a) a divisão da área total nas três áreas principais; em (b) a divisão completa dos cômodos [28].	26
3.1	Exemplo de uma <i>seed</i> criada a partir de um ambiente virtual gerado.	28
3.2	Esquema representando o processo de criação de um ambiente virtual, baseado em [28].	28
3.3	Esquema representando o modelo de geração de comportamento.	29
3.4	Exemplo de uma <i>seed</i> da família.	30
3.5	Exemplos de <i>seeds</i> geradas para dois membros de uma família.	34
3.6	Gráfico representando o variação padrão do status da fome de um agente de acordo com o passar do tempo.	35
3.7	Gráfico representando o variação padrão do status da energia de um agente de acordo com o passar do tempo.	36
3.8	Gráfico representando o variação padrão do status do humor de um agente de acordo com o passar do tempo.	36
3.9	Gráfico representando o variação padrão do status da higiene de um agente de acordo com o passar do tempo.	37
3.10	Gráfico representando o variação padrão do status da necessidade fisiológica de um agente de acordo com o passar do tempo.	37

3.11	Atributo energia do agente ID=1 da família ID=1 em função do tempo, quando iniciada em tempo=19:00, quando o processamento é interrompido devido ao jogador não estar mais visualizando a família em questão (tempo=24:00), e quando a família é re-instanciada às 10:00 continuando até às 18:00.	41
4.1	<i>Seeds</i> dos seis ambientes utilizados como exemplo.	43
4.2	<i>Seeds</i> das seis famílias utilizadas como exemplo.	43
4.3	<i>Seed</i> do agente que é membro da família de ID=1.	44
4.4	Ambiente utilizado pela família de ID=1, com 1 dormitório. Em A, o ambiente é visualizado no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do ambiente em 3D (letras maiúsculas representam os cômodos).	44
4.5	Imagem do agente membro de ID=0 da família de ID=1 no ambiente.	45
4.6	Imagem do agente membro de ID=0 da família de ID=1 na sala.	45
4.7	Imagem do agente membro de ID=0 da família de ID=1 no banheiro.	46
4.8	<i>Seeds</i> dos agentes da família de ID=2.	46
4.9	Ambiente utilizado pela família de ID=2, com 2 dormitórios. Em A, o ambiente é visualizado no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do ambiente em 3D (letras maiúsculas representam os cômodos).	47
4.10	Imagem dos agentes membros da família de ID=2 se locomovendo pela casa.	48
4.11	Imagem dos agentes, membros da família de ID=2 para demonstrar um agente esperando o banheiro desocupar como explicado na Seção 3.4.	48
4.12	Imagem do agente de ID=2 membro da família de ID=2 em seu dormitório com a câmera próxima.	49
4.13	<i>Seeds</i> dos agentes da família de ID=3.	49
4.14	Ambiente utilizado pela família de ID=3, com três dormitórios. Em A, o ambiente é visualizado no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do ambiente em 3D (letras maiúsculas representam os cômodos).	50
4.15	Imagem dos agentes membros da família de ID=3 se locomovendo pela casa.	51
4.16	Imagem dos agentes membros da família de ID=3 em diferentes cômodos da casa.	51
4.17	Imagem dos agentes membros da família de ID=3 em diferentes cômodos da casa.	52
4.18	<i>Seeds</i> dos agentes da família de ID=4.	52

4.19 Ambiente utilizado pela família de ID=4, com 1 dormitório. Em A, o ambiente é visualizado no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do ambiente em 3D (letras maiúsculas representam os cômodos).	53
4.20 Imagem dos agentes membros da família de ID=4 em diferentes cômodos da casa.	54
4.21 Imagem dos agentes membros da família de ID=4 em seu dormitório com a câmera de perto.	54
4.22 Imagem do agente de ID=0 membro da família de ID=4 no escritório.	55
4.23 <i>Seeds</i> dos agentes da família de ID=5.	55
4.24 Ambiente utilizado pela família de ID=5, com dois dormitórios. Em A, o ambiente é visualizado no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do ambiente em 3D (letras maiúsculas representam os cômodos).	56
4.25 Imagem dos agentes membros da família de ID=5 se locomovendo pela casa.	57
4.26 Imagem dos agentes membros da família de ID=5 em seus respectivos dormitórios.	57
4.27 Imagem do agente de ID=1 membro da família de ID=5 em seu dormitório.	58
4.28 <i>Seeds</i> dos agentes da família de ID=6.	58
4.29 Ambiente utilizado pela família de ID=6, com três dormitórios. Em A, o ambiente é visualizado no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do ambiente em 3D (letras maiúsculas representam os cômodos).	59
4.30 Imagem com dois dos agentes membros da família de ID=6.	60
4.31 Imagem dos agentes membros da família de ID=6 em diferentes cômodos da casa.	60
4.32 Imagem dos agentes membros da família de ID=6 em diferentes cômodos da casa.	61
4.33 Imagem de dois dos agentes membros da família de ID=6 entrando na cozinha com a câmera próxima.	61
4.34 Tabela com os valores de cada atributo do agente de ID=0, membro da família de ID=1, juntamente com as ações realizadas pelo mesmo ao decorrer de um dia (início às 7 horas).	62
4.35 Tabela com os valores de cada atributo do agente de ID=1, membro da família de ID=2, juntamente com as ações realizadas pelo mesmo ao decorrer de um dia (início às 7 horas).	62

4.36 Tabela com os valores de cada atributo do agente de ID=2, membro da família de ID=3, juntamente com as ações realizadas pelo mesmo ao decorrer de um dia (início às 7 horas). 63

4.37 Tabela com os valores de cada atributo do agente de ID=0 membro da família de ID=4 juntamente com as ações realizadas pelo mesmo ao decorrer de um dia (início às 7 horas). 63

4.38 Tabela com os valores de cada atributo do agente de ID=1 membro da família de ID=5 juntamente com as ações realizadas pelo mesmo ao decorrer de um dia (início às 7 horas). 64

4.39 Tabela com os valores de cada atributo do agente de ID=2 membro da família de ID=6 juntamente com as ações realizadas pelo mesmo ao decorrer de um dia (início às 7 horas). 64

Lista de Tabelas

3.1	Pesos para a seleção do número de membros de acordo com o Censo 2010 para Porto Alegre [21], e pesos do caso especial (caso de 1 dormitório). . . .	30
3.2	Porcentagem do gênero dos membros de acordo com o Censo 2010 para Porto Alegre [21].	31
3.3	Papéis na família e seus respectivos pesos segundo o Censo 2010 para Porto Alegre [21].	32
3.4	Porcentagens que definem se um par de responsáveis é formado ou não por cônjuges de acordo com o Censo 2010 para Porto Alegre [21].	33
3.5	Porcentagens que definem as faixas etárias de cada membro de acordo com seu gênero de acordo com o Censo 2010 para Porto Alegre [21].	33
3.6	Tempo mínimo e máximo de cada ação juntamente com o valor que o atributo de <i>status</i> \vec{s} irá receber após a conclusão da mesma.	40

Sumário

1. Introdução	15
1.1 Estrutura da Dissertação	16
2. Trabalhos Relacionados	17
2.1 Simulação de Multidões	17
2.1.1 <i>Biocrowds</i>	20
2.2 Modelagem Procedural	23
2.2.1 Modelo Procedural para Geração de Ambientes	25
3. Modelo Procedural de Comportamento	27
3.1 Geração de Famílias	29
3.2 Geração dos Membros	30
3.3 Criação do Comportamento	34
3.4 Seleção de Ações	38
4. Resultados	43
4.1 Avaliação	65
5. Considerações Finais	67
Bibliografia	69

1. Introdução

No nosso dia-a-dia se pode constantemente observar a formação de multidões, aglomerações de pessoas e pequenos grupos. Isto pode ocorrer em lugares amplos e com uma grande densidade de pessoas, como em um espetáculo ou no centro da cidade, ou dentro de ambientes como em uma loja, em uma casa ou em um shopping. Dependendo de onde essas pessoas se encontram e também das diferentes densidades populacionais, o comportamento de cada indivíduo varia.

Estudos sobre a formação desses grupos e multidões são feitos por diversas áreas e com diferentes objetivos. Por exemplo, há estudos de multidões em grandes estádios para ajudar no planejamento, evitando situações como aglomeração exagerada na saída de jogos ou até mesmo acidentes mais graves no caso de brigas ou evacuações; nas situações de emergência em geral que podem ocorrer em grandes eventos como shows, festivais e também dentro de prédios quando uma evacuação é necessária; dentre outros. Estas e outras situações de risco podem ser identificadas e evitadas com a ajuda de modelos de simulação de multidões. Outro objetivo é estudar a formação de grupos e multidões para criar modelos que simulem da forma mais real possível o fenômeno de aglomeração. Simulações desse tipo podem ser encontradas em filmes, como "Avatar" [10] e "O Senhor dos Anéis" [41], em cenas de grandes batalhas onde animar milhões de personagens individualmente seria inviável. Também se pode observar o uso de simulações de grupos ou multidões em jogos que utilizam cidades ou lugares com muitas pessoas como cenário, que é o caso do "GTA IV" [34], "Assassin's Creed" [40] e "World of Warcraft" [4], por exemplo.

Nos casos de jogos com grandes cenários, muitas vezes pode-se perceber um problema: além dos ambientes que os personagens principais do jogo interagem, existem ambientes só de fachada, que são feitos apenas para prover realismo visual, não permitindo nenhuma interação e nem conteúdo. Exemplos desse tipo de ambiente podem ser: casas, prédios, lojas, restaurantes e etc. Muitas vezes, o usuário do jogo não consegue diferenciar os ambientes que permitem interação dos ambientes que são somente de fachada, o que pode gerar situações incômodas como esbarrar em uma porta que não abre. Porém, criar manualmente cada ambiente, além de adicionar conteúdo e humanos virtuais que realizem ações coerentes com o ambiente em que se encontram é, na maioria das vezes, inviável devido ao trabalho de modelagem exaustivo que isto acarretaria, de acordo com a proporção dos cenários.

Com foco no problema apresentado acima, uma possível solução seria a geração automática desses ambientes e pessoas. Esta solução é a área de interesse deste estudo, que foca na geração procedural de grupos e na animação comportamental dos mesmos em ambientes virtuais. São utilizados como base ambientes gerados previamente utilizando um

modelo procedural proposto por Marson [28], que será mais detalhado na Subseção 2.2.1. Esses ambientes virtuais contêm informações semânticas em cada cômodo. Por exemplo, dentro de uma casa, o agente tem como saber em qual cômodo se encontra (quarto, sala, cozinha, banheiro e etc.), além de obter as dimensões desse cômodo. O objetivo deste trabalho é gerar um modelo procedural que gere grupos de agentes e comportamentos para os mesmos de maneira procedural e automática. O modelo deverá gerar comportamentos que sejam coerentes com o ambiente em que os agentes se encontram sem haver necessidade de intervenção humana e, além da coerência com o ambiente, os comportamentos devem ter coerência com o tempo, permitindo que um grupo seja observado em diferentes momentos sem que ocorra perda de contexto. Estes fatores serão aprofundados no Capítulo 3.

Como definido em [8], uma técnica procedural utiliza um algoritmo ou equação matemática para definir alguma característica em um modelo computacional. Por exemplo, uma textura procedural para definir uma superfície de mármore não utiliza uma imagem real para definir os valores de cor, e sim um algoritmo com equações matemáticas. Vantagens de utilizar uma técnica procedural seriam a economia de espaço em disco e o processamento, já que as animações e as estruturas são geradas apenas no momento de serem utilizadas.

1.1 Estrutura da Dissertação

Esta dissertação está organizada da seguinte maneira: inicialmente, apresenta-se no Capítulo 2 os trabalhos científicos relacionados com o tema deste trabalho, os quais em grande parte serviram para embasar o desenvolvimento do modelo apresentado no Capítulo 3. Nesse capítulo é apresentado o modelo desenvolvido. Na sequência, resultados são apresentados no Capítulo 4. Por fim, considerações finais são feitas no Capítulo 5.

2. Trabalhos Relacionados

Neste capítulo são apresentados trabalhos relacionados à simulação de multidões densas e não densas, juntamente com trabalhos voltados para o comportamento de agentes ou grupos (Seção 2.1) e modelagem procedural (Seção 2.2). Esses trabalhos servem de base para o estudo e o desenvolvimento do modelo. Além disso, são apresentados de forma detalhada dois trabalhos que são os pontos de partida para o desenvolvimento do modelo atual. Na Seção 2.1.1 o *Biocrowds* [3] e na Seção 2.2.1 o já mencionado Modelo Procedural para Geração de Ambientes [28].

2.1 Simulação de Multidões

A área de simulação de multidões contém diversas formas de abordagem para gerar tais simulações, porém sempre com o mesmo objetivo final de obter um resultado o mais realista possível. Alguns trabalhos simulam multidões densas, com um número alto de agentes, enquanto outros focam em simular pequenos grupos. Nessas simulações, as multidões podem ser tratadas como uma entidade global ou cada indivíduo pode ser tratado separadamente.

No trabalho de Reynolds [33], que é conhecido como o pioneiro que definiu a área de animação comportamental, é apresentado um modelo para simulação de bandos de pássaros, cardumes de peixes e rebanhos de animais. Neste trabalho, cada personagem é tratado separadamente, como uma partícula chamada de *boïd*. Alguns exemplos de *boïds* encontram-se na Figura 2.1. Cada *boïd* conhece seus vizinhos e o ambiente e segue algumas regras para se movimentar. A primeira regra é evitar colisão. Para isso, cada um deve manter uma distância mínima de seus vizinhos e obstáculos existentes no ambiente. Outras regras fazem com que cada *boïd* controle sua velocidade de acordo com o grupo, juntamente com sua orientação, que é baseada nos centróides dos vizinhos, a fim de manter-se no bando. Essas regras podem gerar comportamentos emergentes em cada simulação.

No trabalho de Musse *et. al* [31] é apresentado um modelo (ViCrowd) que pode ser controlado por *script* ou por um controlador externo. O modelo utiliza humanos virtuais para representar cada agente e permite que comportamentos sejam definidos individualmente ou para grupos na multidão a ser simulada. Esses comportamentos permitem a classificação das multidões em três diferentes categorias. Multidões guiadas são multidões que podem ser controladas externamente através do *mouse* ou outras interfaces. Possuem pouca autonomia e inteligência, tendo uma baixa complexidade de comportamento. Multidões programadas têm um nível de autonomia médio, onde a complexidade dos seus comportamentos e nível de interação pode variar. Já em um multidão autônoma, o nível



Figura 2.1: Exemplos de *boids* [33].

de autonomia, inteligência e complexidade do comportamento são mais altos, tendo uma interação também variável.

No trabalho de Treuille et al. [39] é apresentado um modelo que é um desenvolvimento do modelo apresentado por Hughes [18] e [19], tratando a multidão como uma entidade contínua. Uma ilustração do trabalho está na Figura 2.2. Neste, a multidão é tratada como uma junção de grandes grupos ao invés de agente por agente. O movimento é visto como uma minimização de energia por partículas, campos de atração e repulsão que são criados de acordo com o conhecimento global das características da multidão. Conhecimentos como densidade populacional, objetivo geográfico e velocidade influenciam a movimentação dos agentes. Também é especificada uma mínima distância por agente, criada para evitar colisão entre os mesmos. Este trabalho unifica o planejamento global e tratamento de colisão, sendo mais indicado para simular grandes multidões.

Recentemente, o trabalho de Guy et al. [13] se foca no problema de gerar comportamentos de multidão heterogêneos, modificando parâmetros de simulação para emular traços de personalidade de um indivíduo na multidão e avaliar o impacto dessas personalidades individuais no coletivo. Alguns trabalhos focados no controle de movimento de pequenos grupos são propostos por Lai et al. [25], que apresenta um método para a construção de grafos de movimento para grupos de agentes discretos, incluindo uma nova estrutura de grafo que garante o movimento sintetizado e técnicas para computar probabilidades de transição neste grafo, aprofundadas em [25]. O modelo proposto por Moussaid et al. [29] mostra que os pequenos grupos de pedestre, normalmente compostos por até três membros, tendem a seguir algumas formações ao se deslocar, formações estas que dependem da densidade

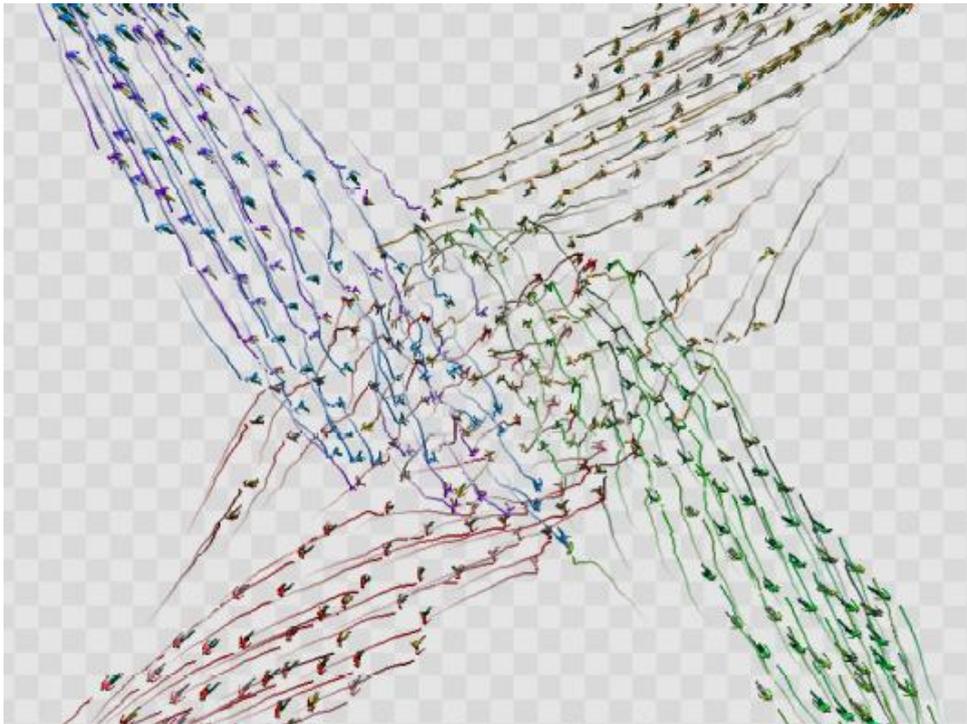


Figura 2.2: Multidão contínua formando linhas e vortex [39].

populacional. A uma baixa densidade, o grupo tende a formar uma linha perpendicular ao vetor objetivo, já a uma densidade um pouco mais alta, a formação fica semelhante à forma de uma letra V e a uma densidade muito alta os agentes tendem a formar uma fila. O trabalho descreve essas formações através de um modelo baseado na comunicação social entre os membros do grupo. Outro trabalho é o de Karamouzas et al. [23], que se baseia neste trabalho de Moussaid para realizar formações de grupos, simulando diferentes manobras para evitar colisão dependentes da velocidade e espaço, se esforçando sempre para manter uma formação que facilite a interação social do grupo.

Toni Conde e Daniel Thalmann [7] apresentam uma contribuição original para a área de animação comportamental, apresentando um método que permite que agentes virtuais autônomos aprendam automaticamente um modelo comportamental. Esse método é baseado na exploração do ambiente por cada agente. O agente explora o ambiente virtual no qual se encontra e constrói estruturas na forma de modelos ou mapas cognitivos, podendo passar esse conhecimento para outros agentes. Valores são atribuídos para cada ação, e uma busca A* [15] é feita nas estruturas sempre buscando o maior valor possível. O processo de aprendizado se baseia na observação de ações de um agente denominado *expert*. Ao observar um número de trajetórias do *expert* o agente pode evoluir através da técnica de aprendizado por reforço inverso [1].

2.1.1 *Biocrowds*

O *Biocrowds* é um modelo de simulação de multidões desenvolvido por Bicho [3] na sua tese de doutorado, e é uma das principais base para este trabalho. O modelo do Bicho é baseado no modelo geométrico para geração de padrões de nervuras em folhas vegetais proposto por Runions [35]. Este modelo é baseado na hipótese de canalização apresentada por Sachs em [36] e [37], que demonstra que o crescimento das nervuras da planta é estimulado por hormônios chamados auxinas que ocupam o espaço livre atraindo as nervuras em sua direção. No modelo do *Biocrowds* essas auxinas são substituídas por marcadores, que representam o espaço livre por onde os agentes podem se locomover.

Além disso, o conceito de proxêmica [14] também é utilizado. Isto define que cada agente tem seu espaço pessoal dentro de uma multidão. No *Biocrowds* cada agente tem seu espaço definido pelos marcadores mais perto dele do que de qualquer outro. Esses marcadores são atribuídos ao agente, como ilustra a Figura 2.3, e só podem ser utilizados por ele naquele momento. Esta divisão do conjunto de marcadores representa a decomposição do espaço de acordo com a distância a determinados pontos que, no caso, são as posições dos agentes. Essa decomposição se assemelha ao Diagrama de Voronoi [32].

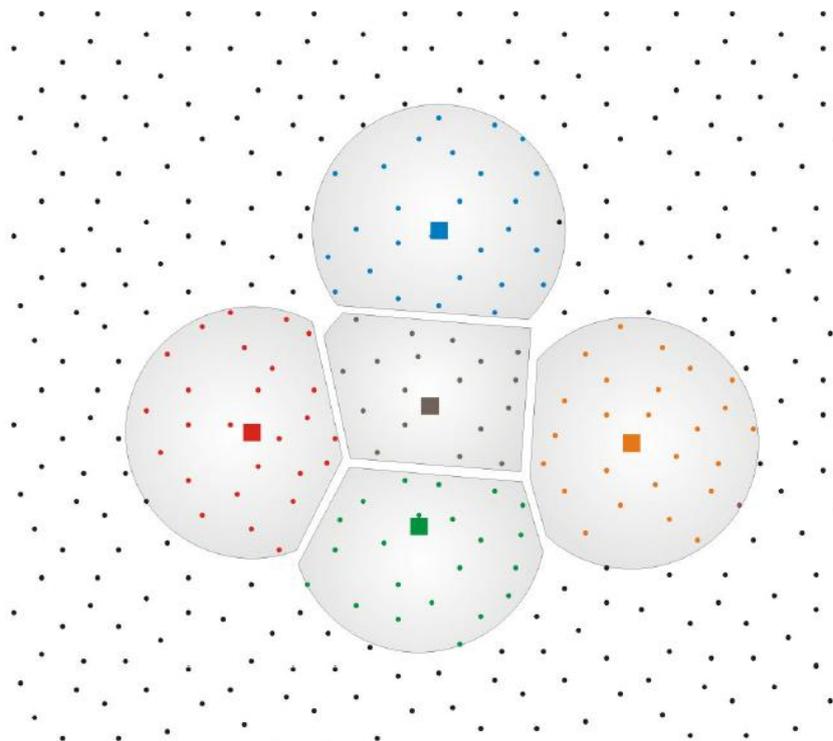


Figura 2.3: Representação de cinco agentes e seus espaços pessoais. Agentes representados pela mesma cor de seus respectivos marcadores [6].

Adaptando o modelo de Runions para simulação de multidões, Bicho modificou alguns fatores. Primeiramente, os agentes só podem ser influenciados pelos marcadores que estão

atribuídos a ele, ou seja, pelos marcadores que estão dentro da sua região de percepção e que estão mais próximos a ele do que a qualquer outro agente, fator que faz com que o modelo seja livre de colisão. Os marcadores no modelo de Bicho são mantidos até o fim da simulação, representando os caminhos livres que podem ser percorridos, diferentemente das auxinas que são removidas e renovadas a cada iteração no modelo de Runions. O agente, no modelo de Bicho, se desloca influenciado pelos marcadores atribuídos a ele juntamente com o seu objetivo final, ou seja, a cada iteração é efetuado um cálculo para verificar a próxima posição do agente, cálculo esse baseado nesses dois fatores. Após o deslocamento, os marcadores são liberados e um novo cálculo é feito, atribuindo novamente ao agente os marcadores mais próximos à sua posição atual e calculando sua próxima posição. A velocidade dos agentes no modelo *Biocrowds* também varia de acordo com o espaço disponível e suas velocidades máximas.

No trabalho de Cassol et al. [6], o modelo do *Biocrowds* [3] é aperfeiçoado para suportar a simulação de comportamentos de grupos de personagens baseados em raciocínio de terrenos. O trabalho desenvolveu uma ferramenta de geração de terreno denominada *Terrain-Tool* que gera mapas de altura, geometria de terrenos e define a semântica dos mesmos. O conceito de diferentes tipos de marcadores foi adicionado, sendo que os marcadores originais do *Biocrowds* agora podem ser definidos como caminháveis, semi-caminháveis e não caminháveis. Esta definição permite que áreas do terreno sejam definidas como não caminháveis pelos agentes como, por exemplo, paredes, e outras áreas como semi-caminháveis, ou seja, os agentes podem caminhar através delas, porém em situações mais específicas já que estas geram desconforto. Exemplos seriam lagos ou grama. Nas Figuras 2.4 e 2.5 são ilustrados exemplos dessas áreas com características especiais.

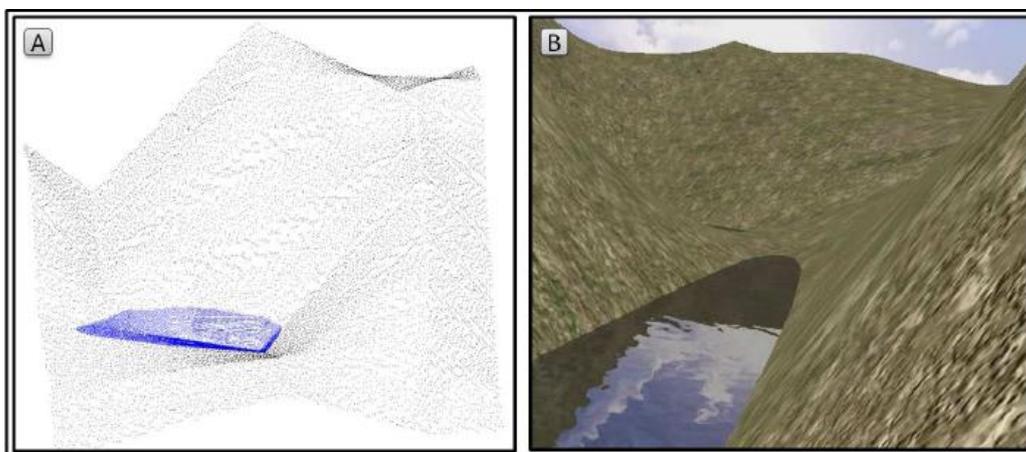


Figura 2.4: Os marcadores não caminháveis são usados no terreno para definir um lago. Em (A) é apresentado o ambiente de simulação, onde é possível observar os marcadores e marcadores especiais; e em (B) a mesma situação renderizada durante a visualização [6].

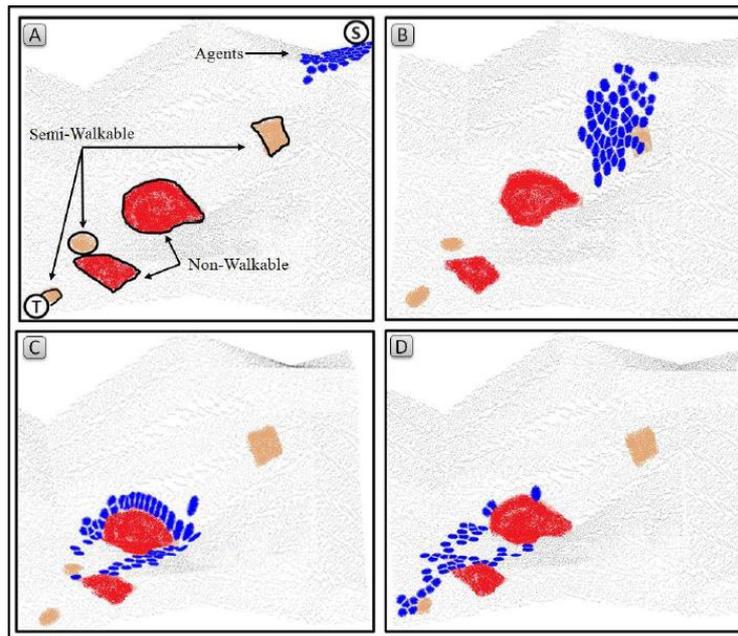


Figura 2.5: Evolução da trajetória de um grupo de agentes levando-se em conta os marcadores especiais, partindo de S com o destino T [6].

O conceito de planejamento de caminhos também foi introduzido neste trabalho. Os terrenos são estruturados em *grids* ou em um conjunto de nodos, o que permite o cálculo de melhor caminho entre a posição inicial e o objetivo, que também leva em consideração as áreas não caminháveis. Foi introduzido o algoritmo A* [15] para realizar esse cálculo e a utilização do mesmo é opcional pelo usuário. Também é possível adicionar um fator custo em relação às bordas e às depressões do terreno, fazendo com que o caminho a ser calculado pelo algoritmo A* evite essas regiões. A Figura 2.6 ilustra como o terreno é dividido e a Figura 2.7 um exemplo de caminho calculado pelo algoritmo usando A* e levando em conta o peso das bordas, o que faz o caminho desviar do vale presente nas montanhas.

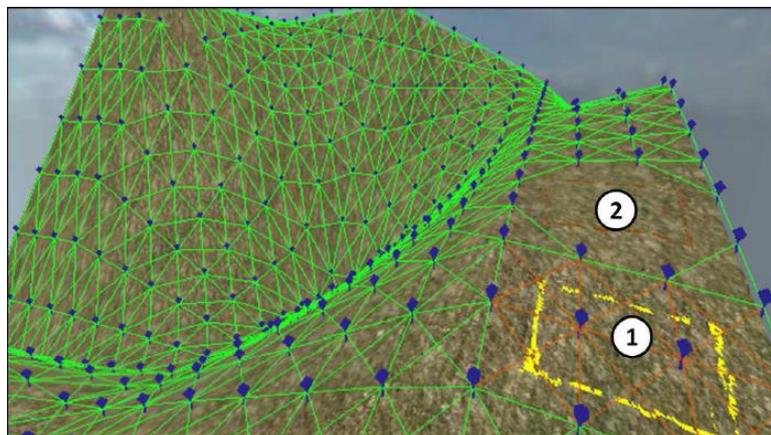


Figura 2.6: Grafo utilizado para o A*. A região 1 representa marcadores semi-caminháveis e a 2 representa uma região não caminhável, portanto sem marcadores [6].

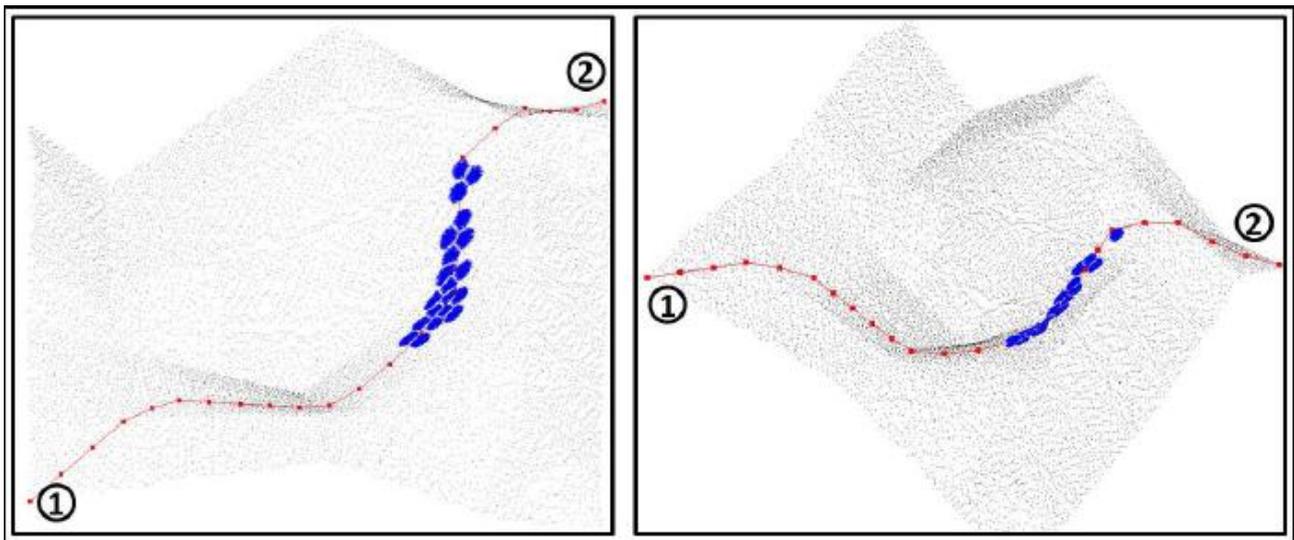


Figura 2.7: Exemplo de trajetória utilizando o A* [6].

2.2 Modelagem Procedural

Na área de modelagem procedural, não foram encontrados trabalhos que abordem exatamente a linha de estudo desejada, que no caso seria animação comportamental. Devido a isto, nesta seção serão apresentados brevemente alguns trabalhos sobre diferentes áreas da modelagem procedural, mostrando onde normalmente ela é aplicada [9]. Por fim, dois trabalhos que são próximos do tema deste estudo serão descritos.

David Fletcher et al. [9] mostram em seu trabalho que atualmente existem pesquisas sobre a criação de prédios, cidades virtuais, oceano, terras e etc. A modelagem procedural está cada vez mais se tornando uma técnica chave nessa criação. Devido ao grande número de detalhes necessários ao se criar um mundo virtual, esta tarefa se torna muito complexa, porém com técnicas procedurais essa complexidade pode ser reduzida. O nível de interação do usuário, o tamanho e a qualidade/realismo dos modelos produzidos são os três atributos mais importantes quando se mede a efetividade de uma técnica procedural. No entanto, criar uma gramática ou regra que um computador consiga seguir e ao mesmo tempo, produzir estruturas totalmente diferentes é complexo e requer elementos randômicos [9].

Na área de planejamento procedural de ruas, o trabalho de Lechner et al. [27] discute o uso de terrenos e provê um sistema que retorna um mapa codificado por cores que representam áreas residenciais, comerciais, industriais, parques recreativos e ruas. Glass et al. [11] examinam o problema de assentamentos informais de padrões de estradas, e identificam quatro técnicas para ajudar neste problema: Diagramas de Voronoi, Subdivisão, Ruído e L-Systems, explicadas em [9]. Lechner et al. [27] criam uma textura, que é gerada utilizando o LSystem em tabelas criadas pelo usuário com atributos de tamanho, tipo e

saúde, para guardar informações sobre uso de terreno, população e estradas. As estradas são geradas a partir dessa textura.

Na área de geração procedural de prédios, além do trabalho de Marson [28] que será detalhado na Subseção 2.2.1, o trabalho de Müller et al. [30] discute uma técnica que usa um número finito de formas primitivas que são reusadas onde for necessário para construir uma estrutura específica. O prédio inteiro pode ser armazenado como uma gramática de formas, uma técnica muito utilizada para a criação de prédios.

Na área de efeitos procedurais de água em tempo real, o trabalho de Jeschke et al. [22] se concentra em criar uma cena de ondas interativa e com um oceano de tamanho infinito. A criação das ondas é dividida em duas partes: a modelagem da espuma e geração procedural de formas de onda. Para gerar proceduralmente a forma da onda, Jeschke primeiro analisa o conhecimento humano sobre o ciclo de vida de uma onda e, após, sugere a utilização de uma mistura de funções escala e rotação para a geração. Já o trabalho de Larsson et al. [26] apresenta um sistema procedural que é capaz de modelar águas abertas com navios e áreas de praia.

Mais próximo à área de pesquisa desse estudo, por tratar de animação de movimentos em humanos virtuais, o trabalho de Sung et al. [38] apresenta um algoritmo para sintetizar movimento de grandes grupos de agentes. O trabalho foca no problema de navegação, apresentando um algoritmo livre de colisão capaz de criar movimentos que satisfazem restrições de duração, posição, orientação e pose, por exemplo, fazendo com que diversos agentes se encontrem em uma posição e tempo específicos. A fim de não gerar muito custo computacional, o trabalho representa o espaço de possíveis movimentos com um grafo de movimentos gerado através do método de Gleicher et al. [12]. Para criar o movimento, o trabalho segue duas etapas: primeiro utilizam um planejador de caminhos baseado em *probabilistic roadmaps* (PRM) [24] para navegar em ambientes complicados. Após, o resultado é refinado através de um algoritmo de busca randomizada que retorna um movimento que satisfaça as restrições. Um exemplo do grafo de movimento encontra-se na Figura 2.8.

Outro trabalho que se foca em animar movimentos é o trabalho de Antonissen e Riff [2] que apresenta um algoritmo baseado em formigas chamado ACOV (*ant colony optimization with covariance*) que utiliza heurísticas para achar rapidamente uma sequência de movimentos que satisfazem restrições e atingem o objetivo principal da animação. O algoritmo utiliza uma combinação de métodos de restrição de tempo e espaço [42] juntamente com redirecionamento de movimento [16], usando animações armazenadas para achar novas soluções.

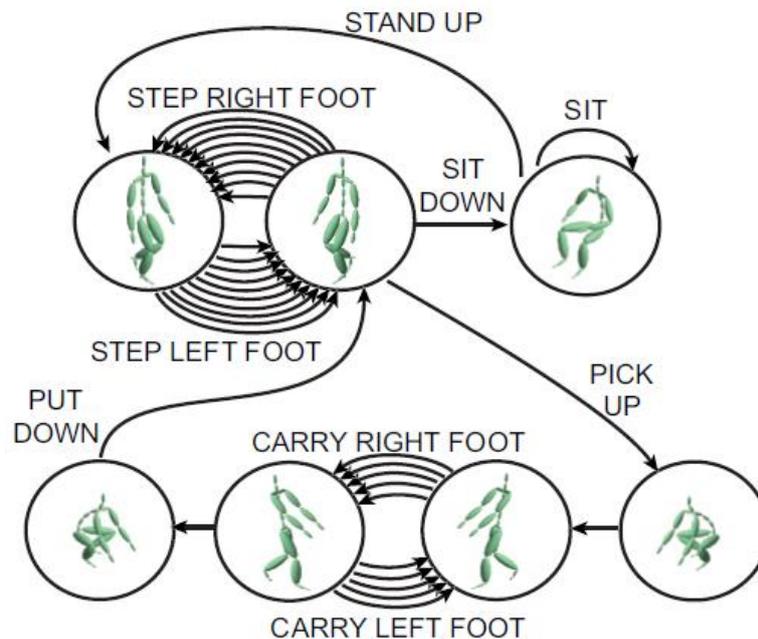


Figura 2.8: Exemplo de grafo de movimento retirado de [38].

2.2.1 Modelo Procedural para Geração de Ambientes

No trabalho de Marson [28] é apresentada uma nova abordagem para a geração de plantas de casas com informações semânticas agregadas de forma automática, plantas estas que são utilizadas como parâmetro de entrada no modelo a ser apresentado neste trabalho (Capítulo 3). Essa abordagem se baseia no método *Squarified Treemaps* proposto por Bruls et al. [5] que utiliza estruturas do tipo árvore e o valor das áreas a serem utilizadas para subdividir o espaço total do ambiente da melhor maneira possível, subdivisões estas que futuramente serão os cômodos.

O modelo funciona da seguinte maneira: primeiramente alguns parâmetros são definidos, como altura, largura e profundidade do ambiente, além de uma lista com as dimensões de cada cômodo juntamente com suas funcionalidades (que definem como o cômodo será utilizado). Existem três opções de funcionalidade: área social, que inclui a sala de estar, sala de jantar e o banheiro; área de serviço, que inclui a cozinha, lavanderia e a dispensa; área privativa, que inclui o quarto principal, quarto, banheiro íntimo e possivelmente um ambiente secundário que pode ser usado como escritório, livraria e etc. Essa lista não é fixa e pode ser alterada pelo usuário. Tendo esses parâmetros definidos, a divisão da área da residência ocorre em dois passos. Primeiro são computadas as áreas de cada uma das três partes (social, serviço e privativa) e, depois de calculadas, estas áreas são utilizadas como entrada para o método *Squarified Treemaps* que então define onde e como os cômodos serão gerados. Esse método gera um layout inicial com três retângulos. Tendo essas três regiões definidas, o método *Squarified Treemaps* é novamente executado para cada

uma das regiões, gerando finalmente o layout com a divisão dos cômodos, como mostra a Figura 2.9.

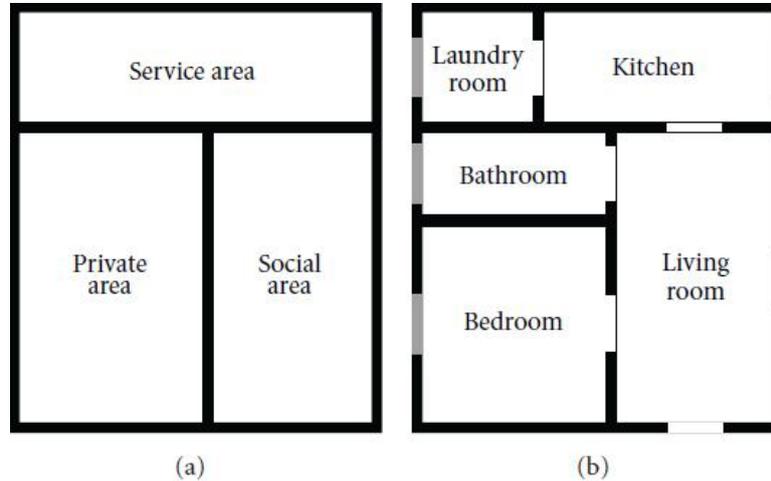


Figura 2.9: Em (a) a divisão da área total nas três áreas principais; em (b) a divisão completa dos cômodos [28].

Tendo este resultado das subdivisões dos cômodos, mais duas etapas são realizadas. Primeiramente as conexões entre os cômodos devem ser criadas. Tendo isto feito, em alguns casos alguns dos cômodos ainda são inacessíveis e, por isso, devem ser tratados. Para isso corredores são criados, conectando todos os cômodos inacessíveis com partes acessíveis. Maiores detalhes de como os corredores são criados pode ser consultado em [28].

3. Modelo Procedural de Comportamento

Como dito anteriormente, são usados como base além do *Biocrowds* [3] os ambientes virtuais gerados pelo trabalho de Marson [28], abordados na Subseção 2.2.1. Também foi acrescentada ao trabalho de Marson a funcionalidade que permite obter as coordenadas de posição de cada móvel presente no ambiente. Com estas informações é possível evitar a geração de marcadores nestas áreas, o que impede que os agentes colidam com os móveis, melhorando a visualização do modelo.

O objetivo do modelo procedural de comportamento é utilizar esses ambientes virtuais gerados e as informações fornecidas por eles para gerar a animação comportamental de pequenos grupos de agentes (que também são gerados de forma automática). Esses pequenos grupos são animados dentro do ambiente, e suas ações devem ter coerência com o mesmo. Por exemplo, a ação comer é executada na cozinha. Além da coerência com o ambiente, o modelo também utiliza o tempo como forma de garantir a coerência dos comportamentos.

A coerência das ações com o tempo se dá no sentido de verificar que tipo de ação é compatível com certa hora do dia. Para tanto, cada agente possui um horário fixo para sair e voltar para casa, representando alguma atividade fixa como trabalho ou estudo. Além desta informação, cada agente possui um horário de preferência, ou seja, se o agente é uma pessoa matutina ou vespertina. Estas informações ajudam a definir sua rotina de sono e comportamento em geral, que são definidos por atributos de *status* (fome, energia, humor, necessidades fisiológicas e higiene). Esses atributos, juntamente com as definições de horário e características dos agentes, serão detalhados nas Seções 3.2 e 3.3

O modelo também tem como objetivo a persistência de todos os dados em relação ao tempo. Isto significa que a simulação pode ser interrompida e retomada a qualquer momento, mantendo a mesma família e agentes, juntamente com a coerência das ações. Esta persistência é possibilitada com a utilização de *seeds* que controlam a aleatoriedade na geração das famílias e comportamentos. Para melhor exemplificar essa persistência desejada, utiliza-se uma situação hipotética: durante um jogo em terceira pessoa ao passar por uma casa qualquer, o jogador olha pela janela e visualiza uma família de humanos virtuais na cozinha. Como esta casa especificamente não faz parte da trama principal do jogo, o jogador desvia sua atenção da casa. Se após algum tempo o jogador passar por esta mesma casa e olhar pela janela novamente, ele deve encontrar a mesma família habitando a casa, provavelmente não mais na cozinha. Ou seja, mesmo que o processamento do comportamento dos agentes não continue executando enquanto a casa não é visualizada, o modelo consegue manter a coerência que será detalhada na Seção 3.3.

O modelo funciona da seguinte forma: A partir de um ambiente virtual gerado, é criada

uma *seed* que representa este ambiente. Como demonstrado na Figura 3.1, esta *seed* é formada com os seguintes dados: número total de cômodos, número total de quartos, área total da planta e os valores das menores e maiores coordenadas (x,y,z) que representam o *bounding box* do ambiente. A *seed* é útil para manter a persistência do ambiente (permitir que o mesmo seja gerado novamente se necessário). Ela também serve de entrada para os próximos passos do modelo que serão abordados nas próximas subseções. Estes passos são:

- Geração das Famílias
- Geração dos Membros
- Criação do Comportamento
- Seleção de Ações

Nas Figuras 3.2 e 3.3 podemos visualizar, respectivamente, um esquema representando a origem das *seeds* do ambiente, e um esquema mostrando uma visão geral dos passos do modelo proposto.

Número De Cômodos	Número De Quartos	Área Total	Mínima Coordenada (x,y,z)	Máxima Coordenada (x,y,z)
5	1	60	10,0,0	20,6,0

Figura 3.1: Exemplo de uma *seed* criada a partir de um ambiente virtual gerado.

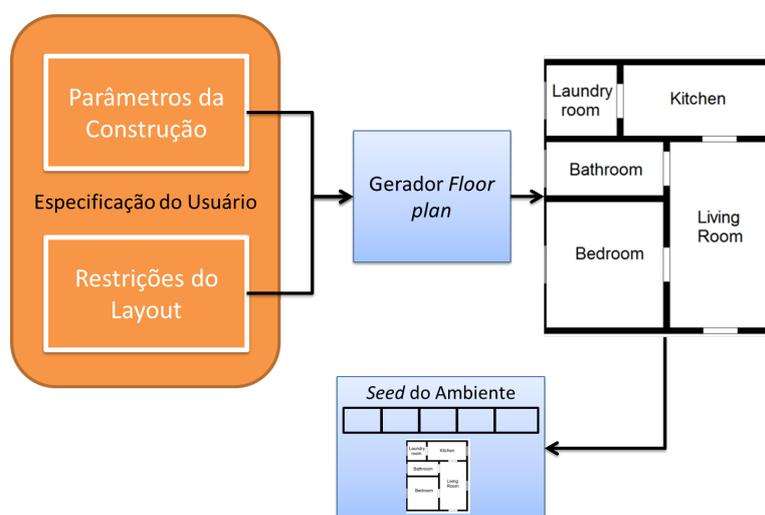


Figura 3.2: Esquema representando o processo de criação de um ambiente virtual, baseado em [28].

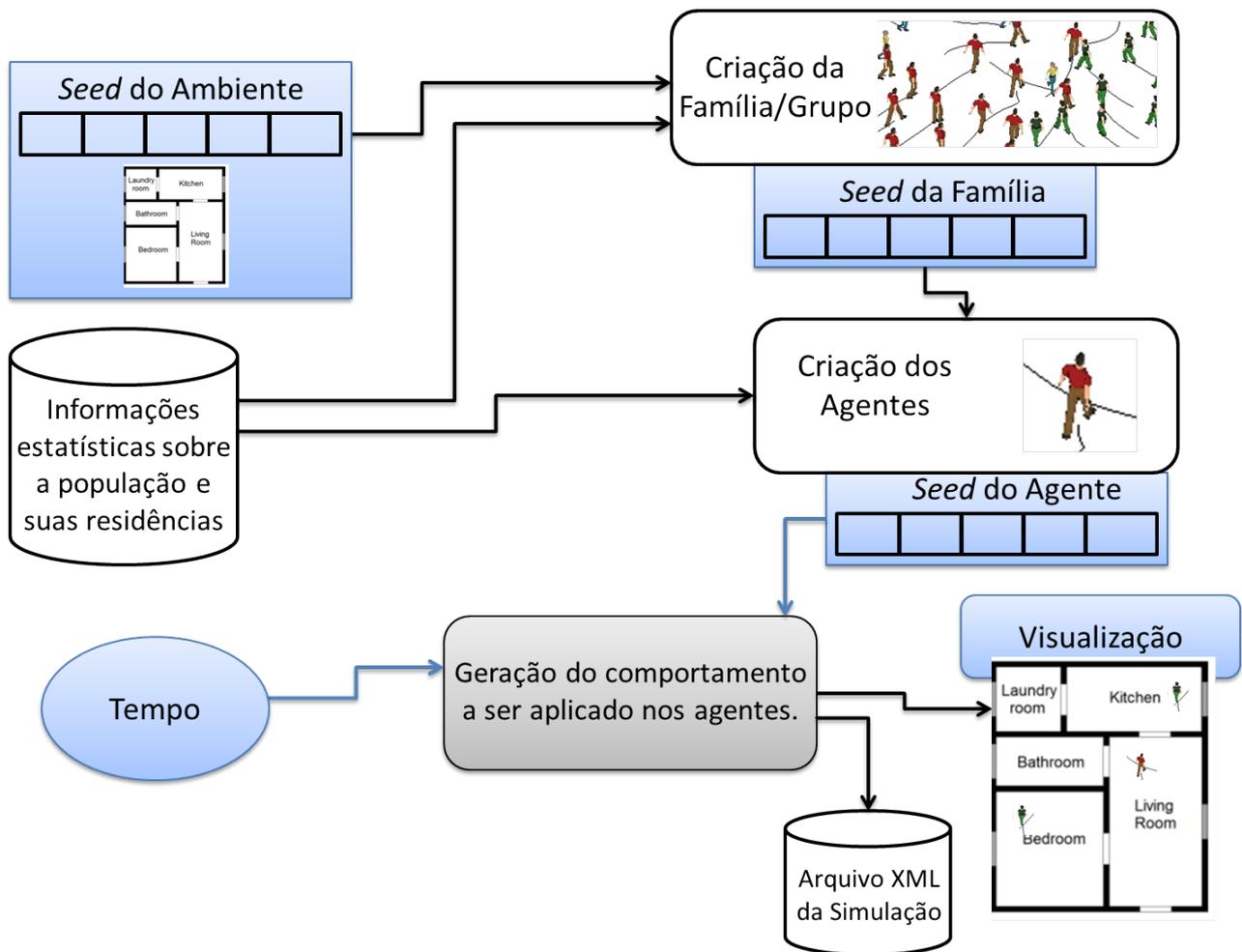


Figura 3.3: Esquema representando o modelo de geração de comportamento.

3.1 Geração de Famílias

Como mostra o esquema da Figura 3.3, o modelo se baseia em um conjunto de regras que irão gerar uma família ou grupo para popular um ambiente específico. A *seed* do ambiente (Figura 3.1) é usada para gerar a aleatoriedade de seleção dessas regras para a geração dos atributos da família. Estes atributos são:

- **ID:** Número identificador que fará parte da *seed* da família (Figura 3.4).
- **Número de membros:** define a quantidade de membros (agentes) pertencentes à família.

As regras para a geração das famílias e dos membros se baseiam em dados do Censo 2010 [20] para Porto Alegre [21] da seguinte maneira: para cada atributo a ser definido, tanto de uma família quanto de um membro, existe um arquivo texto que lista os valores possíveis desse atributo juntamente com seu respectivo peso (porcentagem de presença do valor

desse atributo nos dados utilizados, ou seja, a probabilidade desse valor ser selecionado). O modelo então utiliza uma função aleatória de seleção aplicando os pesos de cada valor a ser selecionado. No caso do atributo número de membros da família, foi especificado para otimizar o modelo que o máximo de membros em uma família é 5 devido às baixas ocorrências de números maiores nos dados estatísticos.

Os pesos para um ambiente ter uma família de um, dois, três, quatro ou cinco membros são exatamente a porcentagem da existência de cada tamanho de família em Porto Alegre. Porém, como os valores do IBGE para este atributo não são relacionados com o número de dormitórios do domicílio, foi decidido a criação de uma segunda lista de pesos para tratar ambientes com apenas um dormitório, diminuindo as chances de famílias com mais de 2 membros serem criadas nesses ambientes (situação considerada mais incomum). A Tabela 3.1 informa os valores dos pesos para cada quantidade de membros em porcentagem.

Número de Membros	Pesos Gerais	Pesos Caso Especial
1	22,65%	40%
2	30,18%	45%
3	24,45%	12%
4	16,07%	2%
5	6,65%	1%

Tabela 3.1: Pesos para a seleção do número de membros de acordo com o Censo 2010 para Porto Alegre [21], e pesos do caso especial (caso de 1 dormitório).

Tendo essas informações definidas, a *seed* da família é criada. Esta *seed* contém o ID da família, o número de membros da mesma juntamente com a *seed* do ambiente utilizada para a geração da mesma. Um exemplo de *seed* da família é demonstrado na Figura 3.4.

<i>ID da Família</i>	<i>Número de Membros</i>	<i>Seed do Ambiente</i>
1	2	516010002060

Figura 3.4: Exemplo de uma *seed* da família.

3.2 Geração dos Membros

Após a quantidade de membros da família ser definida, o próximo passo a ser realizado é geração de cada membro e a definição de suas características (atributos). Para isso, a

seed da família (Figura 3.4) é utilizada para gerar a aleatoriedade de seleção dos valores de cada atributo. Um membro tem os seguintes atributos:

- **Gênero**
- **Papel na família:** Define a condição do membro na família, ou seja, que papel ele exerce
- **Idade**
- **Quarto:** Contém o ID do quarto no qual o agente dorme
- **Horário:** É uma estrutura de dados que contém os seguintes parâmetros:
 - Tipo** - Define o período do dia que o agente tem uma ocupação fora de casa (trabalho ou estudo). Por exemplo, o horário de tipo **integral-manhã** define que o horário é integral (10 horas) e inicia pela manhã
 - Hora de Preferência** - Define se a pessoa é mais matutina ou vesperina. Sua utilidade será detalhada na Seção 3.3
 - Hora de Saída** - Define a hora que o agente sai de casa
 - Hora de Chegada** - Define a hora que o agente retorna para casa
- **Atributos de Status:** Estes atributos são utilizados para realizar a seleção de ações que os agentes realizam e serão detalhados na Seção 3.3

Primeiramente é definido o gênero. Os valores possíveis para este atributo são:

- $\vec{a}_{genero} = \{1, 2\}$ (masculino ou feminino)

São utilizados dados com a probabilidade de um membro ser de gênero masculino ou feminino de acordo com o número total de membros de uma família. Estas probabilidades são demonstradas em forma de porcentagem na Tabela 3.2.

Número de Membros	Porcentagem do Gênero Masculino	Porcentagem do Gênero Feminino
1	39.39%	60.61%
2	48.00%	52.00%
3	55.12%	44.88%
4	59.75%	40.25%
5	55.91%	44.09%

Tabela 3.2: Porcentagem do gênero dos membros de acordo com o Censo 2010 para Porto Alegre [21].

Com os gêneros selecionados, o modelo define o papel de cada membro da família. Os possíveis valores deste atributo são:

- $\vec{a}_{papel} = \{1 - 18\}$ (responsável, filho(a), enteado(a), genro(nora), pai/padrasto(mãe/madastra), sogro(a), neto(a), bisneto(a), irmão(ã), avô(avó), outro(a) parente, agregado(a), convivente, pensionista, empregado(a), parente de empregado(a), individual em domicílio coletivo, cônjuge)

Estes papéis, tanto quanto a probabilidade de cada papel ser designado, também são aplicações diretas do Censo 2010 para Porto Alegre [21], com exceção do papel 18, cônjuge. Este papel é utilizado quando um membro responsável é selecionado como parte de um casal. Quando isto acontece, o papel deste membro é modificado de 1 para 18. Vale notar que as probabilidades variam de acordo com o gênero, e isso também é transmitido para o modelo. Os valores das probabilidades para o atributo papel na família estão demonstradas em forma de porcentagem na Tabela 3.3. Fora estas probabilidades, o modelo aplica uma regra para garantir que cada família tenha pelo menos um membro responsável.

Descrição do Papel	Peso do Gênero Masculino	Peso do Gênero Feminino
Responsável	51,98%	58,22%
Filho(a)	33,87%	27,55%
Enteado(a)	1,5%	11,19%
Genro (Nora)	1,32%	0,96%
Pai/Padrasto (Mãe ou Madastra)	1,04%	2,67%
Sogro(a)	0,14%	0,51%
Neto(a)	4,02%	3,18%
Bisneto(a)	0,15%	0,12%
Irmão(ã)	2,02%	1,9%
Avô(ó)	0,04%	0,14%
Outro(a) Parente	1,88%	1,59%
Agregado(a)	0,26%	0,26%
Convivente	0,82%	0,71%
Pensionista	0,05%	0,04%
Empregado(a)	0,02%	0,3%
Parente de Empregado(a)	0,01%	0,02%
Individual em Domicílio Coletivo	0,88%	0,6%

Tabela 3.3: Papéis na família e seus respectivos pesos segundo o Censo 2010 para Porto Alegre [21].

Para continuar a definir as características de cada membro, nas famílias que possuem um número par de membros responsáveis, é definido se os membros de cada par são ou não cônjuges, sendo o gênero dos integrantes opostos ou iguais. As porcentagens para um par de membros serem ou não cônjuges se encontram na Tabela 3.4.

Gêneros do Casal	Cônjuges	Não Cônjuges
Opostos	36,57%	63,43%
Iguais	0,28%	99,72%

Tabela 3.4: Porcentagens que definem se um par de responsáveis é formado ou não por cônjuges de acordo com o Censo 2010 para Porto Alegre [21].

O próximo atributo a ser definido é a idade do membro. A idade é um atributo que define a faixa etária do agente. Essa definição é feita também através de um sorteio com pesos que variam de acordo com o gênero. O modelo adiciona algumas restrições de acordo com o papel que foi designado ao agente. Estas restrições não permitem que um responsável, um sogro(a), um avô(ó), um pensionista e um empregado(a) sejam menores de idade por exemplo. Os valores possíveis para idade são:

- $\vec{a}_{idade} = \{1, 2, 3\}$ (menor, adulto, idoso)

Vale notar que a faixa etária menor representa idades de 0 a 17 anos, adulto de 18 a 59 anos e idoso de 60 a 100 ou mais anos. As porcentagens que definem a probabilidade de um membro pertencer a cada faixa etária estão demonstradas na Tabela 3.5.

Faixa etária	Gênero Masculino	Gênero Feminino
Menor	25,23%	21,21%
Adulto	62,54%	61,33%
Idoso	12,23%	17,46%

Tabela 3.5: Porcentagens que definem as faixas etárias de cada membro de acordo com seu gênero de acordo com o Censo 2010 para Porto Alegre [21].

Como muitos ambientes tem mais de um dormitório, houve a necessidade da definição do quarto em que cada agente irá dormir. O modelo faz a distribuição dos agentes por estes quartos de acordo com as seguintes regras: Se existe apenas um quarto, todos os agentes dormem neste mesmo quarto; o ambiente tem mais de um quarto e não existem casais na família, os agentes são distribuídos uniformemente pelos quartos; o ambiente tem mais de um quarto e existe um ou mais casais na família, primeiramente cada casal é colocado em um quarto e o restante dos agentes são colocados em um outro quarto desocupado (se ele existir); Finalmente, se ocorrer um caso em que todos os quartos estiverem ocupados por casais e houver ainda um membro restante, este membro irá dividir um quarto com um casal. A seleção de que quarto é de um agente ou casal é feita de forma aleatória.

Os horários são definidos de acordo com a idade dos agentes, ou seja, cada horário contém pesos diferentes para menores, adultos e idosos. Tanto os parâmetros que definem cada horário quanto as porcentagens de seleção de cada um são pré-definidos pelo usuário e podem ser alterados. Os horários possíveis atualmente são os seguintes:

- $\vec{a}_{\text{horario.tipo}} = \{1, 2, 3, 4, 5\}$ (parcial-manha, parcial-tarde, parcial-noite, integral-manha, integral-tarde)
- $\vec{a}_{\text{horario.horaPreferencia}} = \{1, 2\}$ (manhã ou noite, atribuídos em cada tipo de horário)
- $\vec{a}_{\text{horario.saida}} = \{8, 12, 14, 17\}$ (atribuídos em cada tipo de horário)
- $\vec{a}_{\text{horario.chegada}} = \{13, 18, 19, 22\}$ (atribuídos em cada tipo de horário)

Todas os valores de \vec{a} de cada agente formam a sua respectiva *seed* (membro da família) que também servirá como parâmetro para a definição do comportamento (explicado na Seção 3.3). Utilizando o exemplo da família de dois membros que gerou a *seed* representada na Figura 3.4 e também considerando que os membros foram selecionados como um casal, a Figura 3.5 representa exemplos de *seeds* geradas para cada um dos membros desta família.

ID do Agente										Seed da Família										
Gênero	Papel	Idade	Quarto	Horário																
0	1	18	2	4	4	18	19	1	2	5	1	6	0	1	0	0	2	0	6	0
ID do Agente										Seed da Família										
Gênero	Papel	Idade	Quarto	Horário																
1	2	18	2	4	4	18	19	1	2	5	1	6	0	1	0	0	2	0	6	0

Figura 3.5: Exemplos de *seeds* geradas para dois membros de uma família.

3.3 Criação do Comportamento

A criação do comportamento consiste em fazer com que cada agente realize ações no ambiente ao qual este pertence de forma automática. Para tanto, também controla os atributos de *status* de cada agente juntamente com a presença deste agente no ambiente ou não. Como dito na Seção 3.2, cada agente possui atributos de *status* \vec{s} , que controlam diversas propriedades do agente. São eles:

- **Fome(F)**: determina o nível da necessidade de comer.

- **Energia(E)**: determina o nível de energia.
- **Humor(H)**: determina o nível de humor.
- **Higiene(Hi)**: determina o nível de higiene.
- **Necessidade Fisiológica(N)**: determina o nível da necessidade fisiológica.

Os valores destes atributos são os fatores determinantes no critério de seleção das ações. Eles são variados de acordo com uma função do tempo que se baseia em curvas de variação padrão (*default*) de cada atributo (curvas estas representadas por \vec{f}). Os valores destas variações padrão são pré-definidos e, considerando que pessoas podem variar na sua rotina horária (algumas acordam e dormem cedo e outras acordam e dormem tarde) foram criados dois diferentes conjuntos de valores: um para pessoas com hábitos matinais e outro para pessoas com hábitos noturnos. Para selecionar qual destes dois conjuntos de valores serão utilizados é que se utiliza a informação **Hora de Preferência** ($\vec{a}_{horario.horaPreferencia}$) de cada membro, explicada na Seção 3.2.

Nas Figuras 3.6, 3.7, 3.8, 3.9, 3.10 são apresentadas, respectivamente, as curvas de variação padrão \vec{f} para o atributo \vec{s} (F , E , H , Hi e N respectivamente).

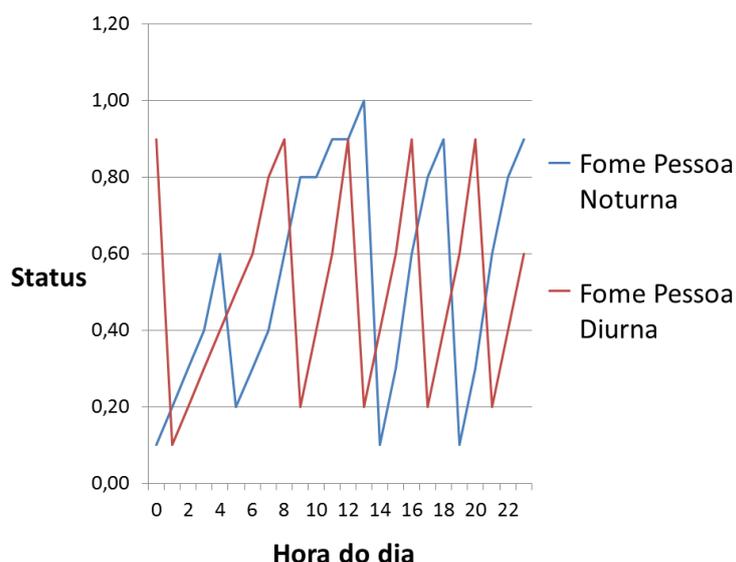


Figura 3.6: Gráfico representando o variação padrão do status da fome de um agente de acordo com o passar do tempo.

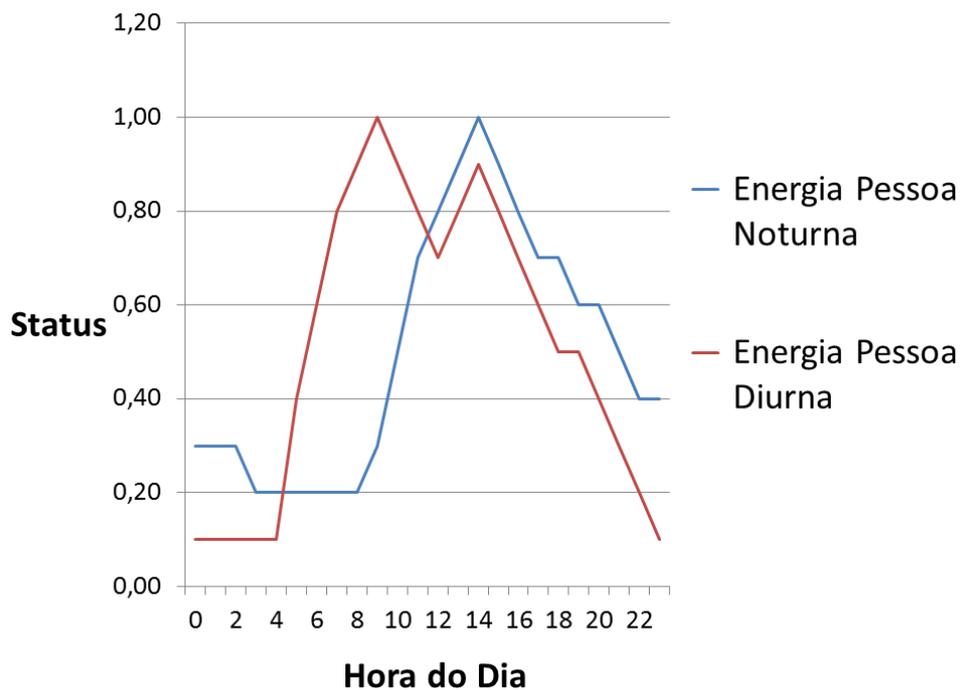


Figura 3.7: Gráfico representando o variação padrão do status da energia de um agente de acordo com o passar do tempo.

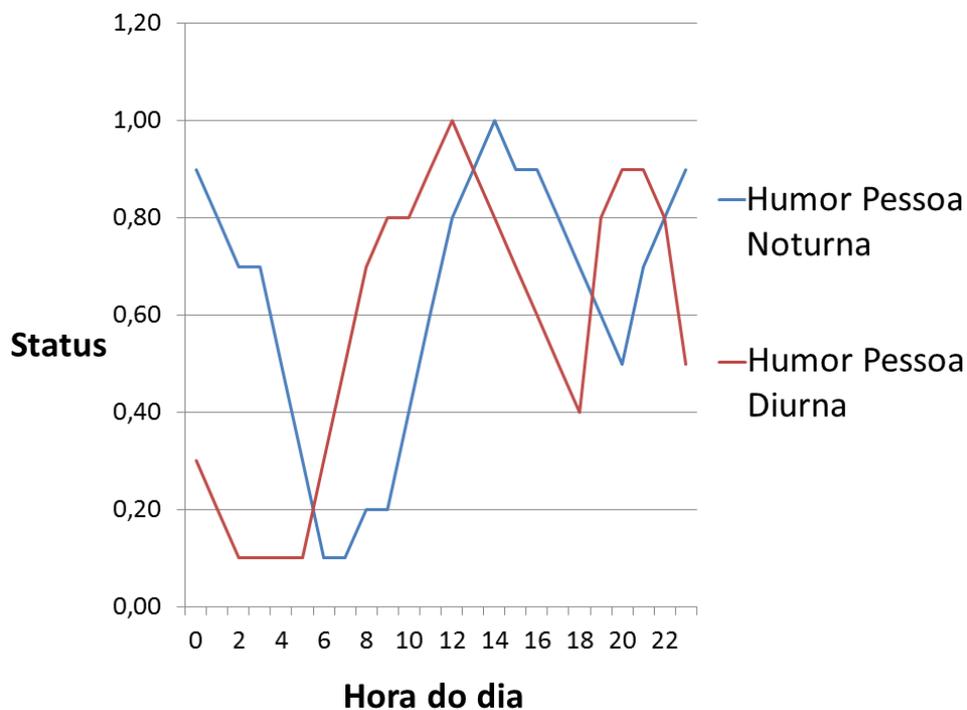


Figura 3.8: Gráfico representando o variação padrão do status do humor de um agente de acordo com o passar do tempo.

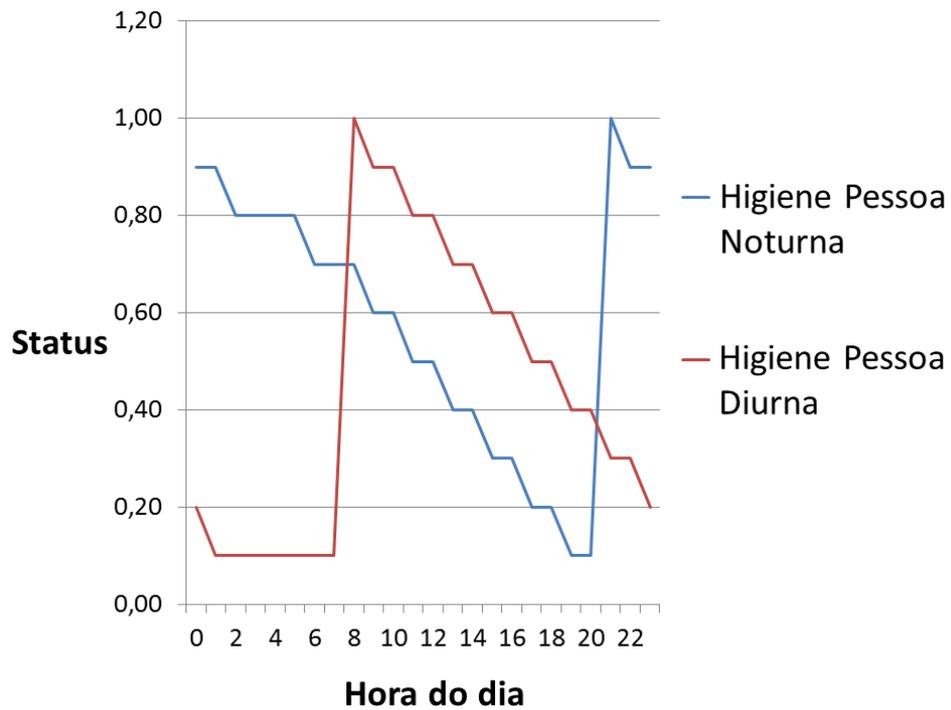


Figura 3.9: Gráfico representando o variação padrão do status da higiene de um agente de acordo com o passar do tempo.

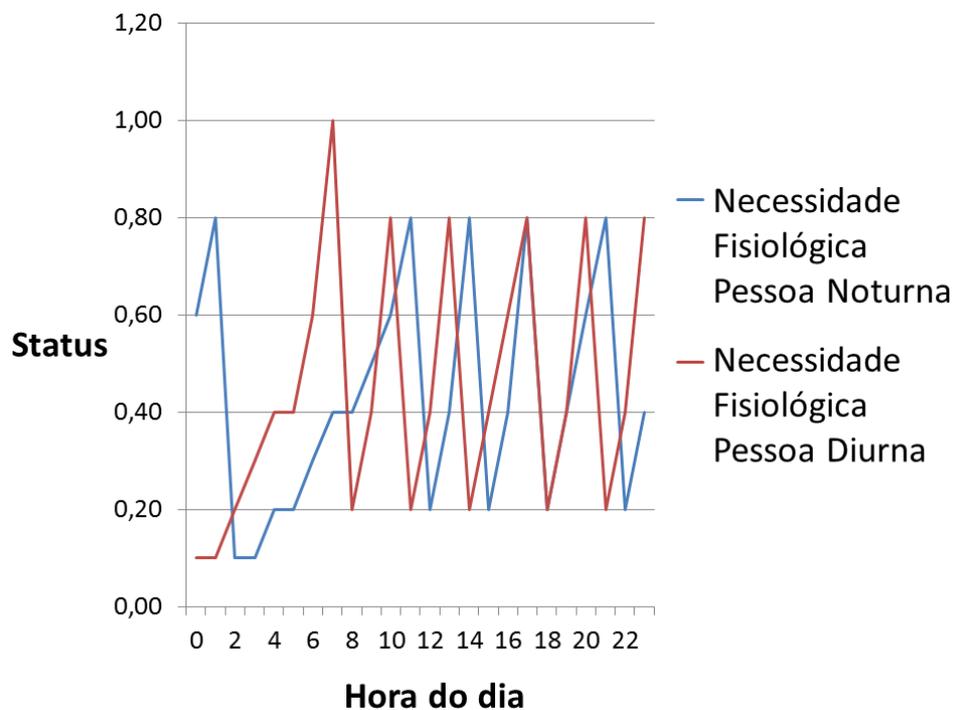


Figura 3.10: Gráfico representando o variação padrão do status da necessidade fisiológica de um agente de acordo com o passar do tempo.

Estes valores fixos \vec{f} são utilizados para inicializar os valores \vec{s} e também para variá-los em função do tempo. A inicialização ocorre da seguinte forma: Considerando o tempo de início da simulação como t , cada agente deve ter seus atributos \vec{s} inicializados de acordo com o valor de seu respectivo \vec{f} em t , como mostra a Equação 3.1:

$$\vec{s}_t = \vec{f}_t. \quad (3.1)$$

A variação dos valores de \vec{s} em função do tempo ocorre de hora em hora e é realizada de forma proporcional aos valores de \vec{f} como define a Equação 3.2:

$$\vec{s}_{t+1} = \vec{f}_{t+1} + (\vec{f}_t - \vec{s}_t). \quad (3.2)$$

Vale lembrar que os valores devem sempre permanecer no intervalo de 0 a 1. A outra forma de variar os valores de \vec{s} é executando uma ação que o afeta. Neste caso, se nota que as ações influenciam os atributos e os atributos determinam a escolha das ações.

3.4 Seleção de Ações

Os atributos de *status* \vec{s} definem as ações que os agentes selecionam uma vez que estes se encontram em casa. Neste caso, é utilizada uma máquina de estados finita determinística ou Autômato Finito Determinístico (AFD) [17] definido por uma quintupla $(\Sigma, J, j_0, \delta, F)$, onde:

- Σ é o alfabeto de entrada (C, Q, S, B, E, Co) . O alfabeto é definido pela primeira letra do cômodo da casa onde as ações podem ocorrer, dependendo do *status* (Cozinha, Quarto, Sala de Estar, Banheiro, Escritório e Corredor).
- J é um conjunto de estados finito e não vazio.
- j_0 é um estado inicial, um elemento de J . Neste caso, os agentes começam em qualquer lugar da casa.
- δ é uma função de transição de estado, $\delta : J \times \Sigma \rightarrow J$. Esta função é dependente dos valores dos atributos $\vec{s} (F, E, H, Hi, N)$.
- F_i é o conjunto de estados finais, igual a J .

A simulação é iniciada colocando os agentes em casa e inicializando o AFD com um primeiro estado J (um dos lugares C, Q, S, B, Co). Após, a função de transição δ pode começar a ser aplicada, detectando níveis de *status* que são menores ou maiores (dependendo do atributo) que uma *threshold* pré-definida para cada atributo (exceto humor H) \vec{s} que são as seguintes:

- $F: F_t \geq 0.7$
- $E: E_t \leq 0.3$
- $Hi: Hi_t \leq 0.3$
- $N: N_t \geq 0.8$

Quando os valores destes atributos de *status* atingem ou ultrapassam a *threshold* definida, o estado modifica. Por exemplo, se $Hi_t = 0.1$ então $J = B$. Obviamente podem ocorrer conflitos, como quando dois atributos estão dentro do limite de suas *thresholds* simultaneamente. Para resolver este problema, foi determinada uma lista de prioridades de seleção, determinando uma ordem de verificação dos valores dos atributos e suas *thresholds* que é a seguinte: (N, F, H, E) . Se nenhum dos valores está dentro dos limites de suas *thresholds*, então uma função randômica utilizando a *seed* do agente é executada para selecionar uma das ações que são relacionadas com o humor (H) do agente, ou seja, atividades de lazer.

Vale notar que o modelo, especificamente para ações executadas no banheiro (B), suporta o caso em que todos os banheiros da casa estão ocupados por agentes. Quando este caso ocorre, o agente que deseja ir à B se desloca para o corredor Co e aguarda até um dos banheiros ser desocupado.

Outro fator importante de cada ação é o tempo de duração da mesma, já que a simulação é apresentada visualmente ao usuário. Para isso, cada ação tem como pré definição um tempo mínimo e máximo para sua execução. Utilizando estas informações, é determinado através de uma função randômica (utilizando a *seed* do agente) o tempo total em que a ação irá ser executada. Quando uma ação é terminada, o atributo de *status* que é impactado pela mesma é alterado para seu valor máximo ou mínimo (dependendo do atributo). Por exemplo, quando um agente termina de comer, seu *status* de fome é alterado para 0. Os tempos (mínimo e máximo) de cada ação, além do valor que cada atributo de *status* recebe após esta ser executada se encontram na Tabela 3.6

Status \vec{s}	Estado no AFD	Duração mínima (mins)	Duração máxima (min)	Novo valor do Status \vec{s}
Necessidade Fisiológica	Banheiro	5	30	0
Higiene	Banheiro	15	15	1
Fome	Cozinha	30	60	0
Energia	Quarto	300	480	1
Humor	Sala de Estar, Quarto ou Escritório	10	90	1

Tabela 3.6: Tempo mínimo e máximo de cada ação juntamente com o valor que o atributo de *status* \vec{s} irá receber após a conclusão da mesma.

O tempo é um importante aspecto neste tipo de simulação, já que o usuário pode ter vontade de avançar o mesmo quando, por exemplo, os agentes estiverem dormindo. Neste caso, o modelo tem uma classe própria para tratar o tempo da simulação, possibilitando que o usuário altere a velocidade do tempo para uma de 3 possíveis escalas: 1 *frame* pode ser $= \frac{1\text{segundo}}{30}$, $= \frac{1\text{minuto}}{30}$ ou $= \frac{1\text{hora}}{30}$.

Também, como este modelo é principalmente usado para personagens de plano de fundo, o usuário pode estar interagindo em outro ambiente de uma cidade virtual e não estar visualizando o ambiente em questão. Consequentemente, o modelo desliga a visualização e o processamento dos agentes dentro do ambiente. Porém, o usuário pode, a qualquer momento, voltar novamente a visualizar a casa. O desafio então é poder re-instanciar as entidades da simulação de maneira coerente também considerando o que estava sendo executado no momento em que a simulação foi desligada.

Primeiramente, no momento em que o desligamento é acionado, todas as *seeds* (ambiente, família e agente) são salvas. Além destas, o modelo salva o estado atual dos atributos de *status* \vec{s}_{ts} juntamente com o tempo em que este salvamento está ocorrendo (aqui indicado como tempo de salvamento ts). Se o ambiente for re-instanciado em um tempo específico t , o modelo primeiramente gera o ambiente, depois a família utilizando a *seed* do ambiente (mesma família é gerada) e finalmente as características fixas de cada agente (\vec{a}) (utilizando a *seed* da família). Para definir os atributos de *status* \vec{s}_t para cada agente no tempo t , o modelo considera o último valor salvo de cada atributo do agente em junção com as curvas de variação padrão \vec{f} de cada um, de acordo com a Equação 3.2, porém substituindo t por ts .

A Figura 3.11 ilustra o processo de re-instanciação (às 10:00) do atributo energia de um agente específico, em comparação ao comportamento padrão deste mesmo atributo.

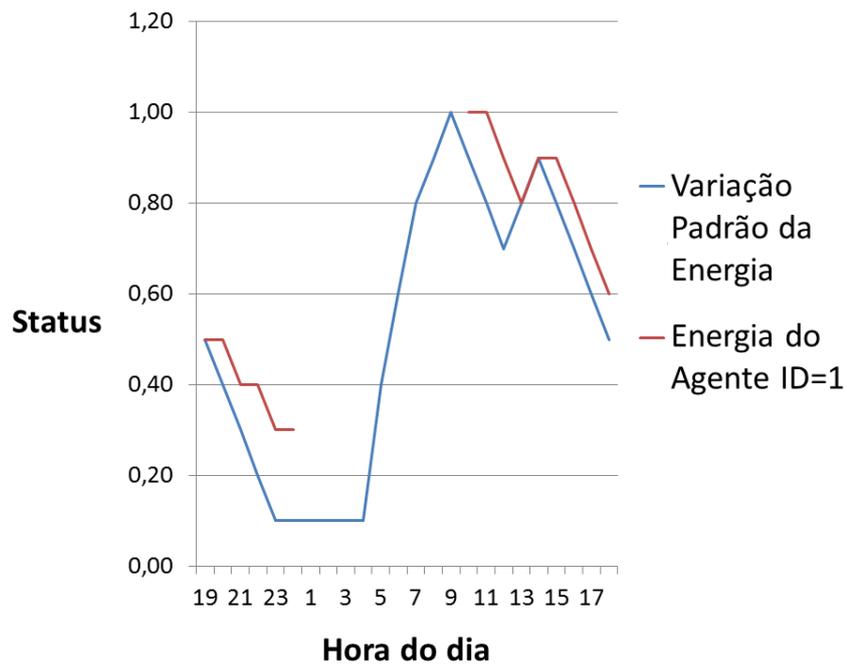


Figura 3.11: Atributo energia do agente ID=1 da família ID=1 em função do tempo, quando iniciada em tempo=19:00, quando o processamento é interrompido devido ao jogador não estar mais visualizando a família em questão (tempo=24:00), e quando a família é reinstanciada às 10:00 continuando até às 18:00.

4. Resultados

Nesta seção são demonstrados alguns resultados gerados pelo modelo proposto. Para tanto, três diferentes casas foram geradas utilizando o modelo de Marson [28] e mais uma cópia de cada uma dessas casas foram utilizadas, porém utilizando posições e móveis diferentes, o que gerou seis famílias distintas que serão detalhadas a seguir.

As Figuras 4.1 e 4.2 ilustram respectivamente as *seeds* dos ambientes e as *seeds* das famílias geradas.

<i>Número de Cômodos</i>	<i>Número de Quartos</i>	<i>Área Total</i>	<i>Mínima Coordenada (x,y,z)</i>	<i>Máxima Coordenada (x,y,z)</i>
5	1	43	9,5,0	15,12,0
7	2	63	0,5,0	7,14,0
8	3	67	17,5,0	24,14,0
5	1	43	26,5,0	32,12,0
7	2	63	34,5,0	41,14,0
8	3	67	42,5,0	49,14,0

Figura 4.1: *Seeds* dos seis ambientes utilizados como exemplo.

<i>ID da Família</i>	<i>Número de Membros</i>	<i>Seed do Ambiente</i>
1	1	514395015120
2	3	72630507140
3	5	836717502414
4	2	5143265032120
5	2	7263345041140
6	4	8367425049140

Figura 4.2: *Seeds* das seis famílias utilizadas como exemplo.

Agora cada família será detalhada individualmente. Na família de ID=1, apenas um membro é gerado. Sua *seed* está representada na Figura 4.3

ID do Agente	Gênero	Papel	Idade	Quarto	Horário	Seed da Família
0	1	1	2	3	511223	11514395015120

Figura 4.3: *Seed* do agente que é membro da família de ID=1.

A Figura 4.4 ilustra em A o ambiente utilizado pela família de ID=1 no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do mesmo ambiente em 3D. Pode-se notar que em A não existem marcadores nas áreas onde os móveis estão presentes (móveis podem ser visualizados em B).

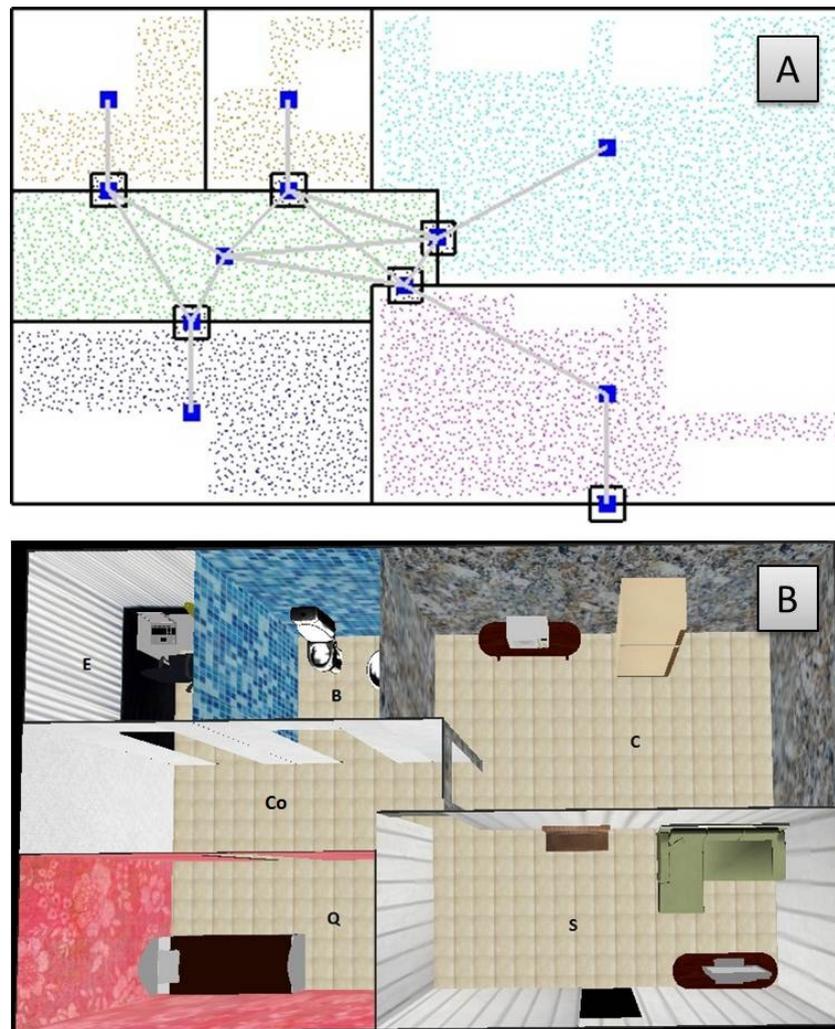


Figura 4.4: Ambiente utilizado pela família de ID=1, com 1 dormitório. Em A, o ambiente é visualizado no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do ambiente em 3D (letras maiúsculas representam os cômodos).

A Figura 4.5 mostra uma imagem do agente no banheiro da casa, e as Figuras 4.6 e 4.7 mostram imagens do agente em diferentes cômodos da casa com a câmera mais próxima.

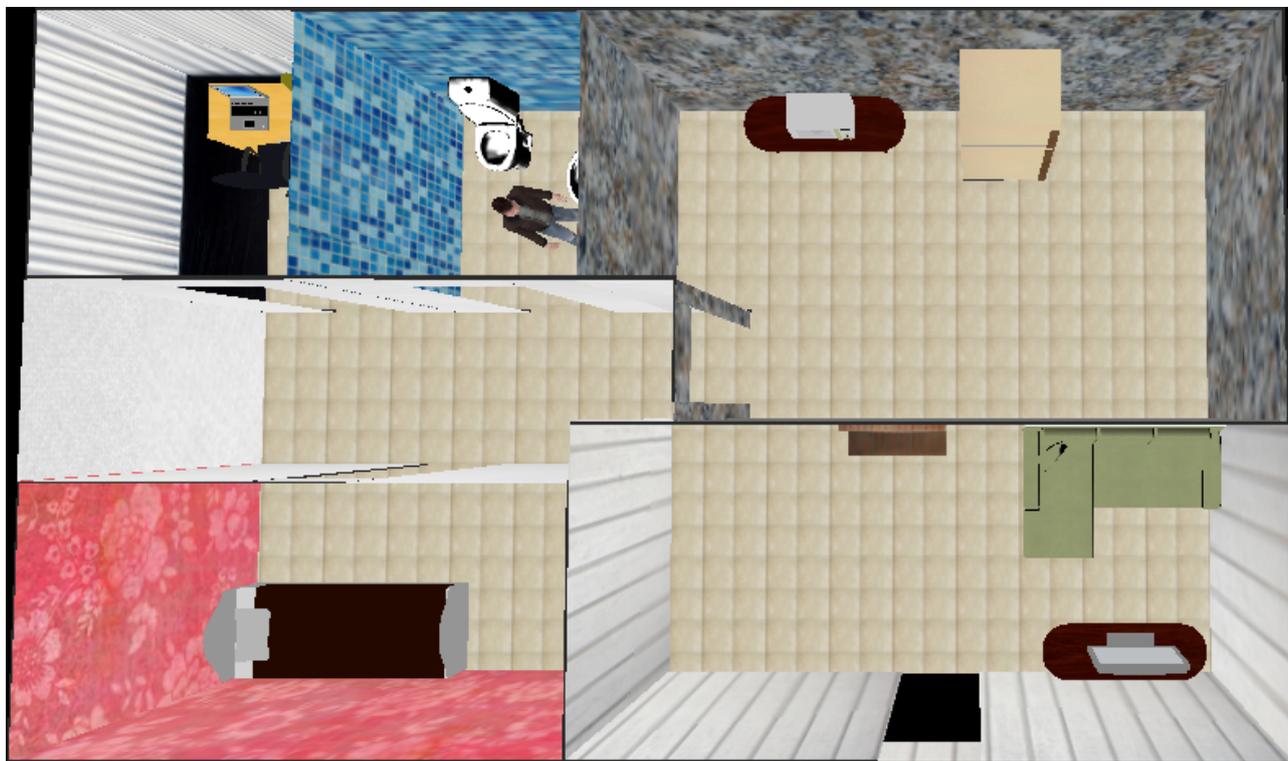


Figura 4.5: Imagem do agente membro de ID=0 da família de ID=1 no ambiente.



Figura 4.6: Imagem do agente membro de ID=0 da família de ID=1 na sala.

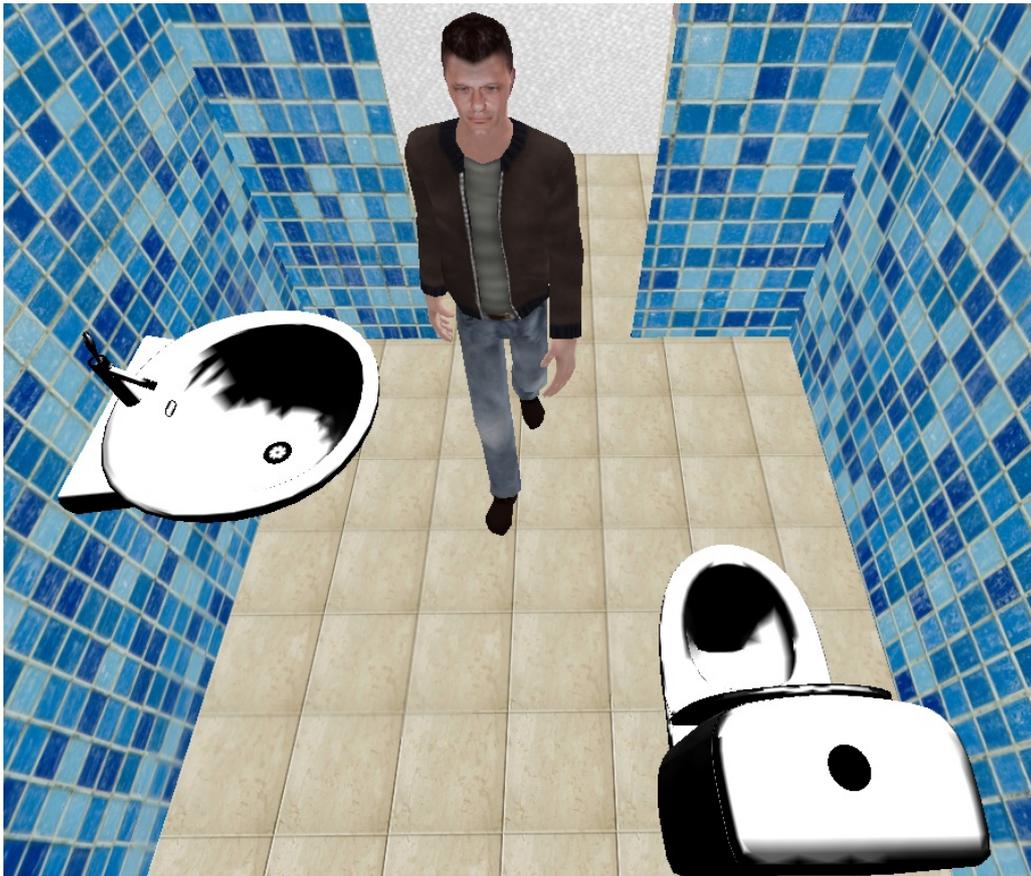


Figura 4.7: Imagem do agente membro de ID=0 da família de ID=1 no banheiro.

Agora a família de ID=2 será detalhada. Ela possui três membros, e as *seeds* deles estão demonstradas na Figura 4.8.

ID do
Agente *Gênero* *Papel* *Idade* *Quarto* *Horário* *Seed da Família*

<i>ID do</i> Agente	<i>Gênero</i>	<i>Papel</i>	<i>Idade</i>	<i>Quarto</i>	<i>Horário</i>	<i>Seed da Família</i>
0	2	18	2	4	221419	2372630507140
1	1	18	2	4	41819	2372630507140
2	2	2	1	3	221419	2372630507140

Figura 4.8: *Seeds* dos agentes da família de ID=2.

A Figura 4.9 ilustra em A o ambiente utilizado pela família de ID=2 no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do mesmo ambiente em 3D. Pode-se notar que em A não existem marcadores nas áreas

onde móveis estão presentes (móveis podem ser visualizados em B).

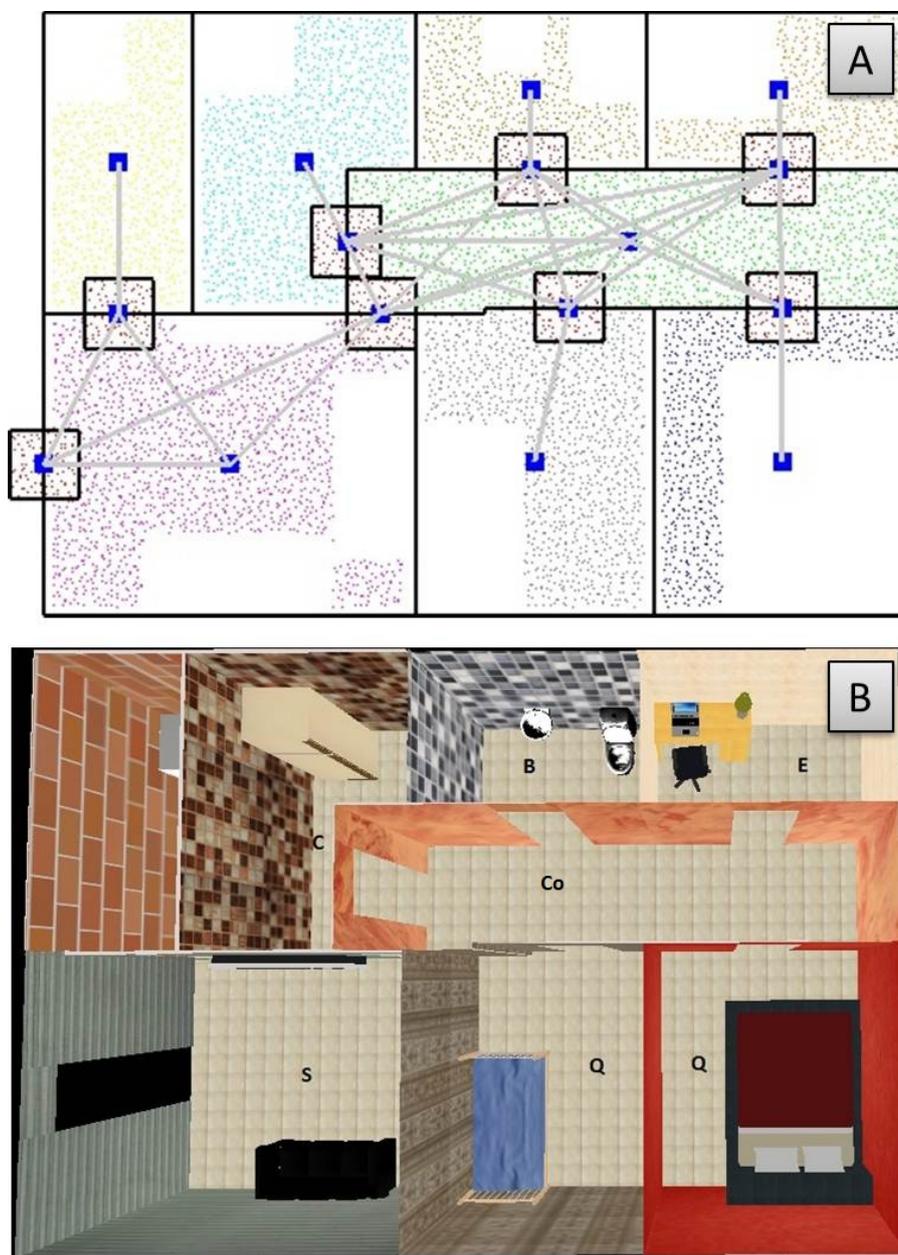


Figura 4.9: Ambiente utilizado pela família de ID=2, com 2 dormitórios. Em A, o ambiente é visualizado no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do ambiente em 3D (letras maiúsculas representam os cômodos).

A Figura 4.10 mostra uma imagem com os agentes se locomovendo pela casa em uma câmera próxima. A Figura 4.11 demonstra um agente aguardando o banheiro ser desocupado para utilizá-lo (como explicado na Seção 3.4). Já a Figura 4.12 mostra a imagem de um membro da família em seu dormitório.



Figura 4.10: Imagem dos agentes membros da família de ID=2 se locomovendo pela casa.



Figura 4.11: Imagem dos agentes, membros da família de ID=2 para demonstrar um agente esperando o banheiro desocupar como explicado na Seção 3.4.

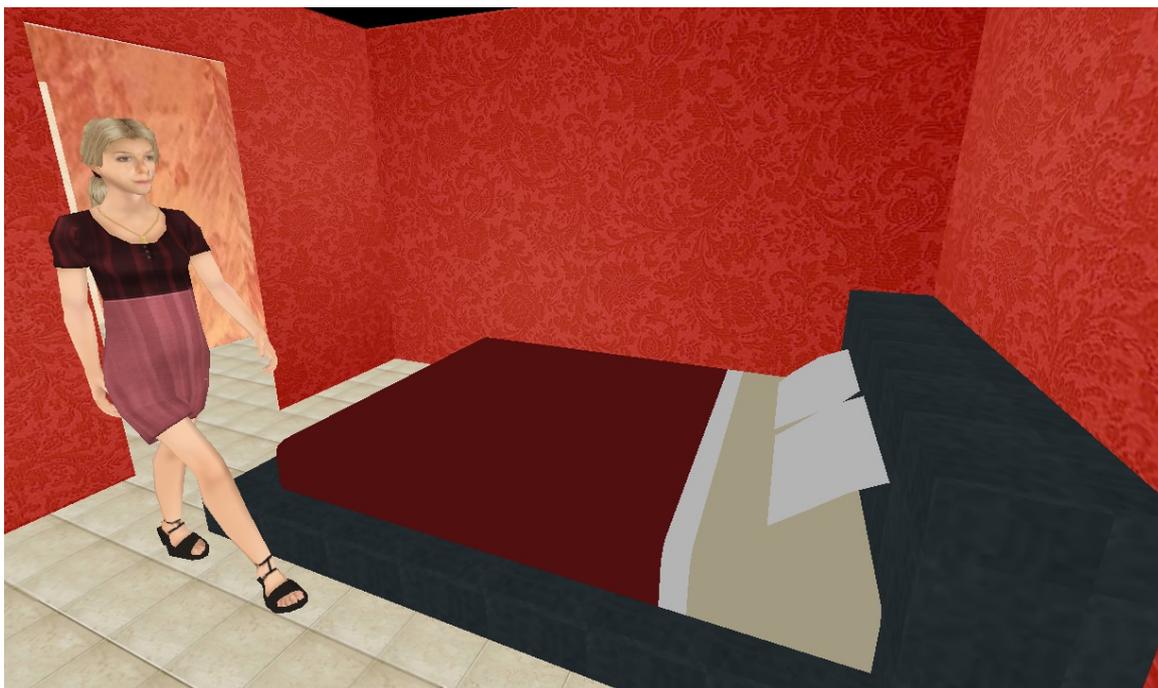


Figura 4.12: Imagem do agente de ID=2 membro da família de ID=2 em seu dormitório com a câmera próxima.

Agora a família de ID=3 será detalhada. Ela possui cinco membros, e as *seeds* deles estão demonstradas na Figura 4.13.

<i>ID do Agente</i>	<i>Gênero</i>	<i>Papel</i>	<i>Idade</i>	<i>Quarto</i>	<i>Horário</i>				<i>Seed da Família</i>
0	2	18	2	3	4	18	19	35836717502414	
1	1	18	2	3	4	18	19	35836717502414	
2	1	2	1	4	2	2	14	19	35836717502414
3	1	2	1	4	2	2	14	19	35836717502414
4	1	2	1	5	2	2	14	19	35836717502414

Figura 4.13: *Seeds* dos agentes da família de ID=3.

A Figura 4.14 ilustra em A o ambiente utilizado pela família de ID=3 no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização

do mesmo ambiente em 3D. Pode-se notar que em A não existem marcadores nas áreas onde móveis estão presentes (móveis podem ser visualizados em B).

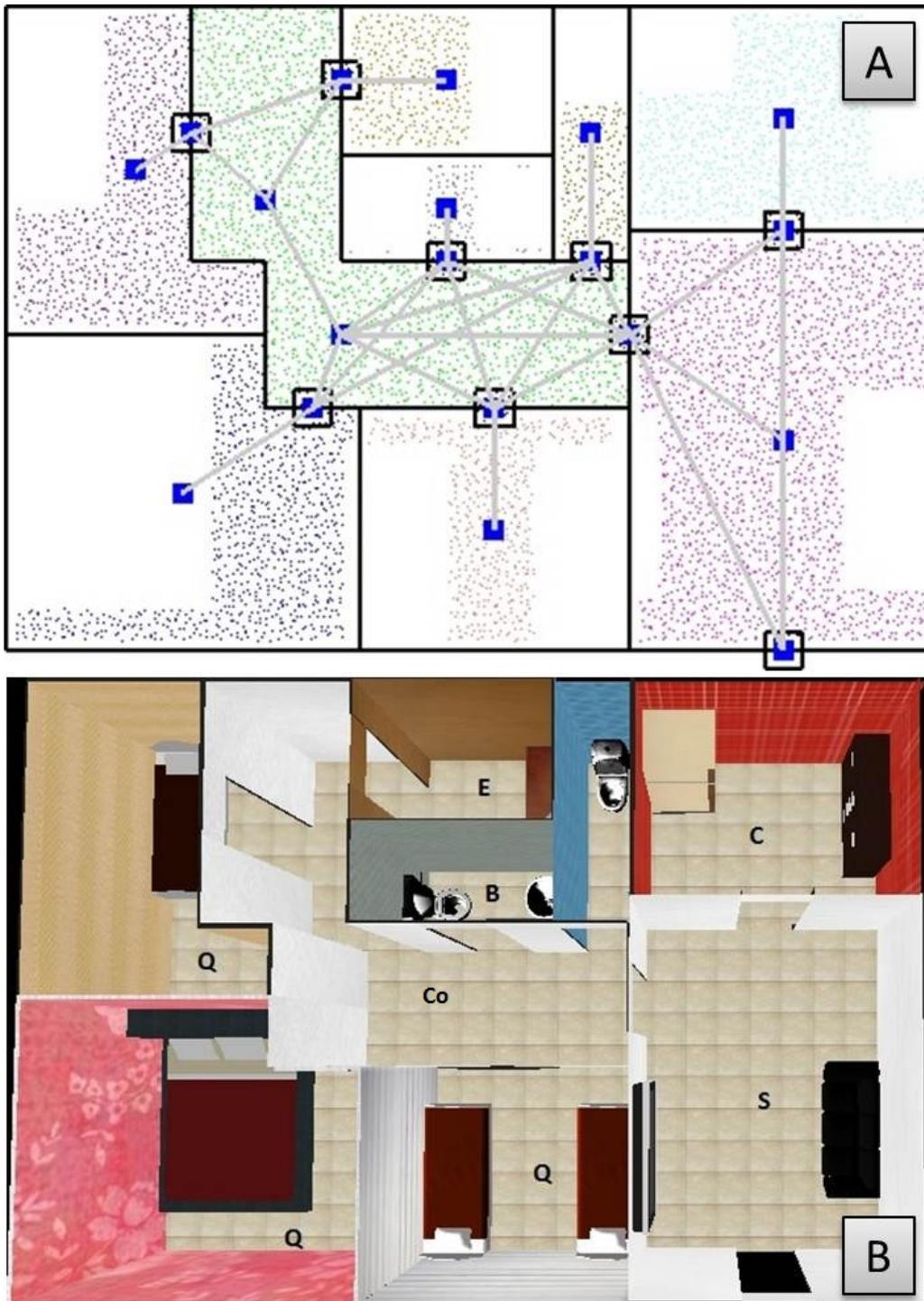


Figura 4.14: Ambiente utilizado pela família de ID=3, com três dormitórios. Em A, o ambiente é visualizado no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do ambiente em 3D (letras maiúsculas representam os cômodos).

A Figura 4.15 mostra uma imagem com os agentes se locomovendo pela casa, e as Figuras 4.16 e 4.17 mostram imagens dos agentes em diferentes cômodos da casa.



Figura 4.15: Imagem dos agentes membros da família de ID=3 se locomovendo pela casa.



Figura 4.16: Imagem dos agentes membros da família de ID=3 em diferentes cômodos da casa.



Figura 4.17: Imagem dos agentes membros da família de ID=3 em diferentes cômodos da casa.

Agora a família de ID=4 será detalhada. Ela possui dois membros, e as *seeds* deles estão demonstradas na Figura 4.18.

ID do
Agente Gênero Papel Idade Quarto Horário Seed da Família

0	2	18	2	3	4	18	19	425143265032120
1	1	18	2	3	4	18	19	425143265032120

Figura 4.18: *Seeds* dos agentes da família de ID=4.

A Figura 4.19 ilustra em A o ambiente utilizado pela família de ID=4 no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do mesmo ambiente em 3D. Pode-se notar que em A não existem marcadores nas áreas onde móveis estão presentes (móveis podem ser visualizados em B).

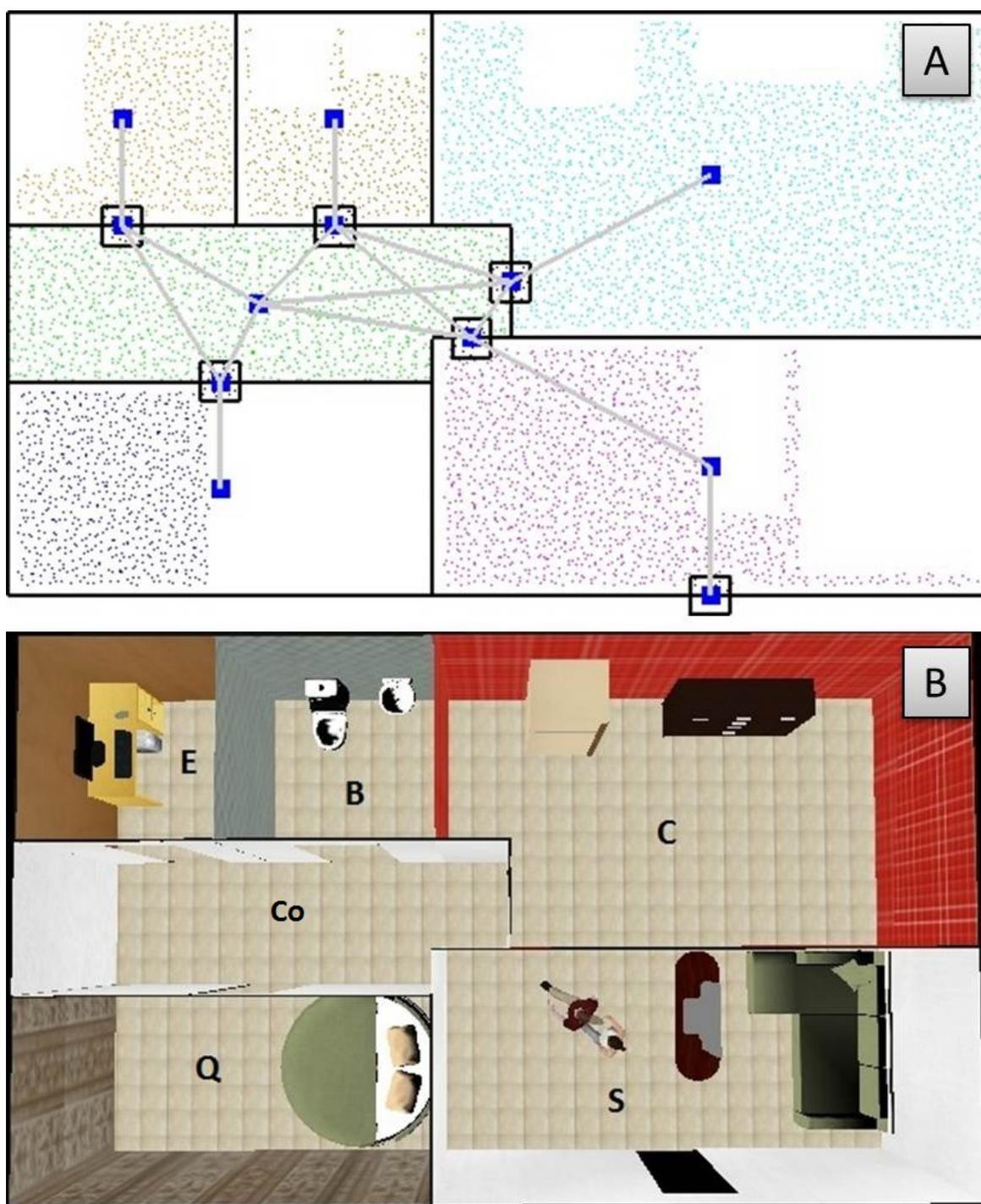


Figura 4.19: Ambiente utilizado pela família de ID=4, com 1 dormitório. Em A, o ambiente é visualizado no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do ambiente em 3D (letras maiúsculas representam os cômodos).

A Figura 4.20 mostra os agentes em diferentes cômodos da casa e as Figuras 4.21 e 4.22 mostram imagens dos agentes com a câmera aproximada.



Figura 4.20: Imagem dos agentes membros da família de ID=4 em diferentes cômodos da casa.



Figura 4.21: Imagem dos agentes membros da família de ID=4 em seu dormitório com a câmera de perto.



Figura 4.22: Imagem do agente de ID=0 membro da família de ID=4 no escritório.

Agora a família de ID=5 será detalhada. Ela possui dois membros, e as *seeds* deles estão demonstradas na Figura 4.23.

<i>ID do Agente</i>	<i>Gênero</i>	<i>Papel</i>	<i>Idade</i>	<i>Quarto</i>	<i>Horário</i>			<i>Seed da Família</i>
0	2	1	2	3	4	18	19	527263345041140
1	1	2	2	3	5	11	23	527263345041140

Figura 4.23: *Seeds* dos agentes da família de ID=5.

A Figura 4.24 ilustra em A o ambiente utilizado pela família de ID=5 no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do mesmo ambiente em 3D. Pode-se notar que em A não existem marcadores nas áreas onde móveis estão presentes (móveis podem ser visualizados em B).

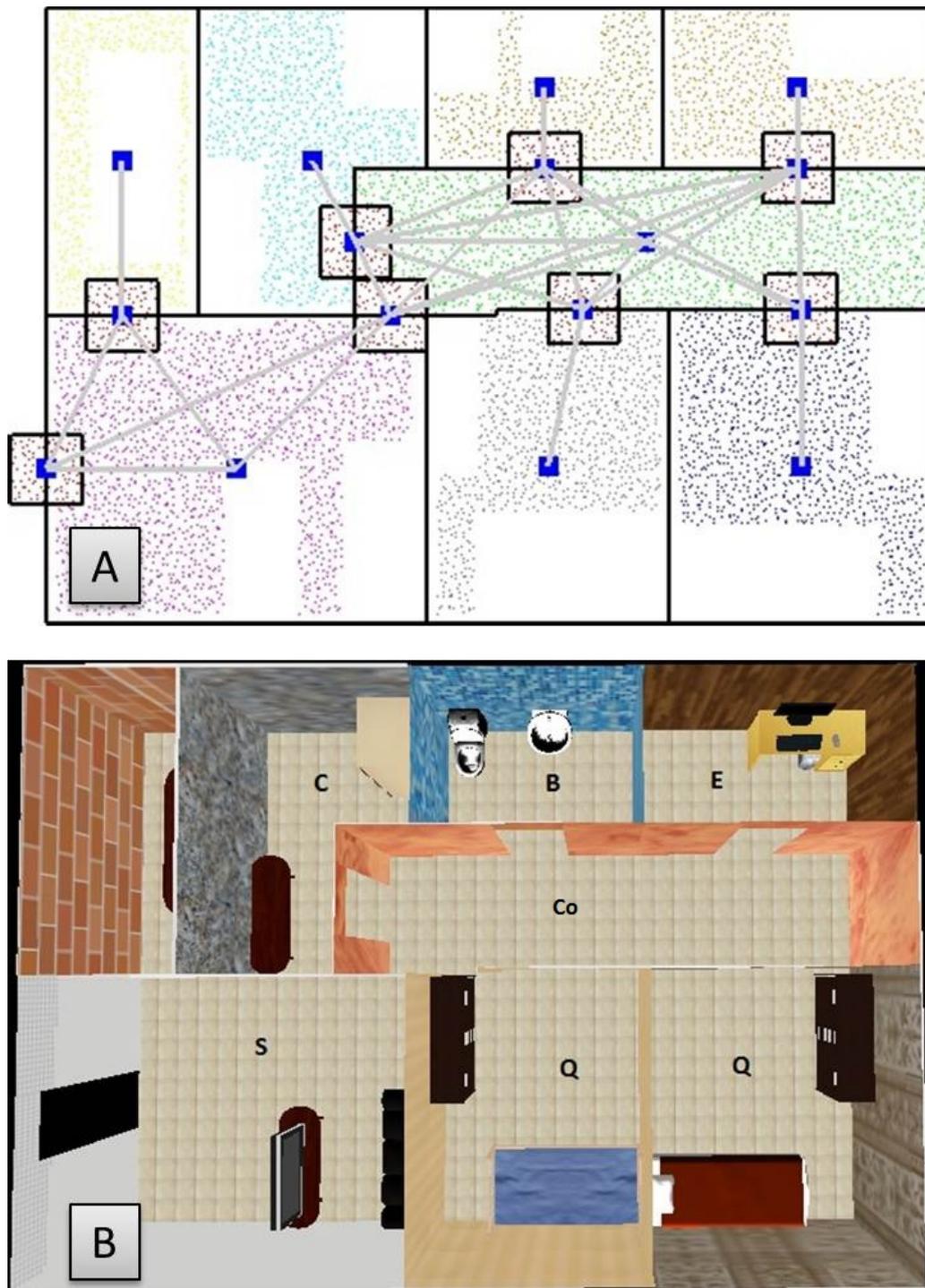


Figura 4.24: Ambiente utilizado pela família de ID=5, com dois dormitórios. Em A, o ambiente é visualizado no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do ambiente em 3D (letras maiúsculas representam os cômodos).

A Figura 4.25 mostra uma imagem com os agentes se locomovendo pela casa, e as Figuras 4.26 e 4.27 mostram imagens dos agentes em diferentes cômodos da casa.



Figura 4.25: Imagem dos agentes membros da família de ID=5 se locomovendo pela casa.



Figura 4.26: Imagem dos agentes membros da família de ID=5 em seus respectivos dormitórios.



Figura 4.27: Imagem do agente de ID=1 membro da família de ID=5 em seu dormitório.

Agora a família de ID=6 será detalhada. Ela possui quatro membros, e as *seeds* deles estão demonstradas na Figura 4.28.

<i>ID do Agente</i>	<i>Gênero</i>	<i>Papel</i>	<i>Idade</i>	<i>Quarto</i>	<i>Horário</i>				<i>Seed da Família</i>
0	2	18	2	3	1	18	13	648367425049140	
1	1	18	2	3	5	1	12	23	648367425049140
2	2	2	1	5	1	18	13	648367425049140	
3	1	2	3	4	2	2	14	19	648367425049140

Figura 4.28: *Seeds* dos agentes da família de ID=6.

A Figura 4.29 ilustra em A o ambiente utilizado pela família de ID=6 no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do mesmo ambiente em 3D. Pode-se notar que em A não existem marcadores nas áreas onde móveis estão presentes (móveis podem ser visualizados em B).

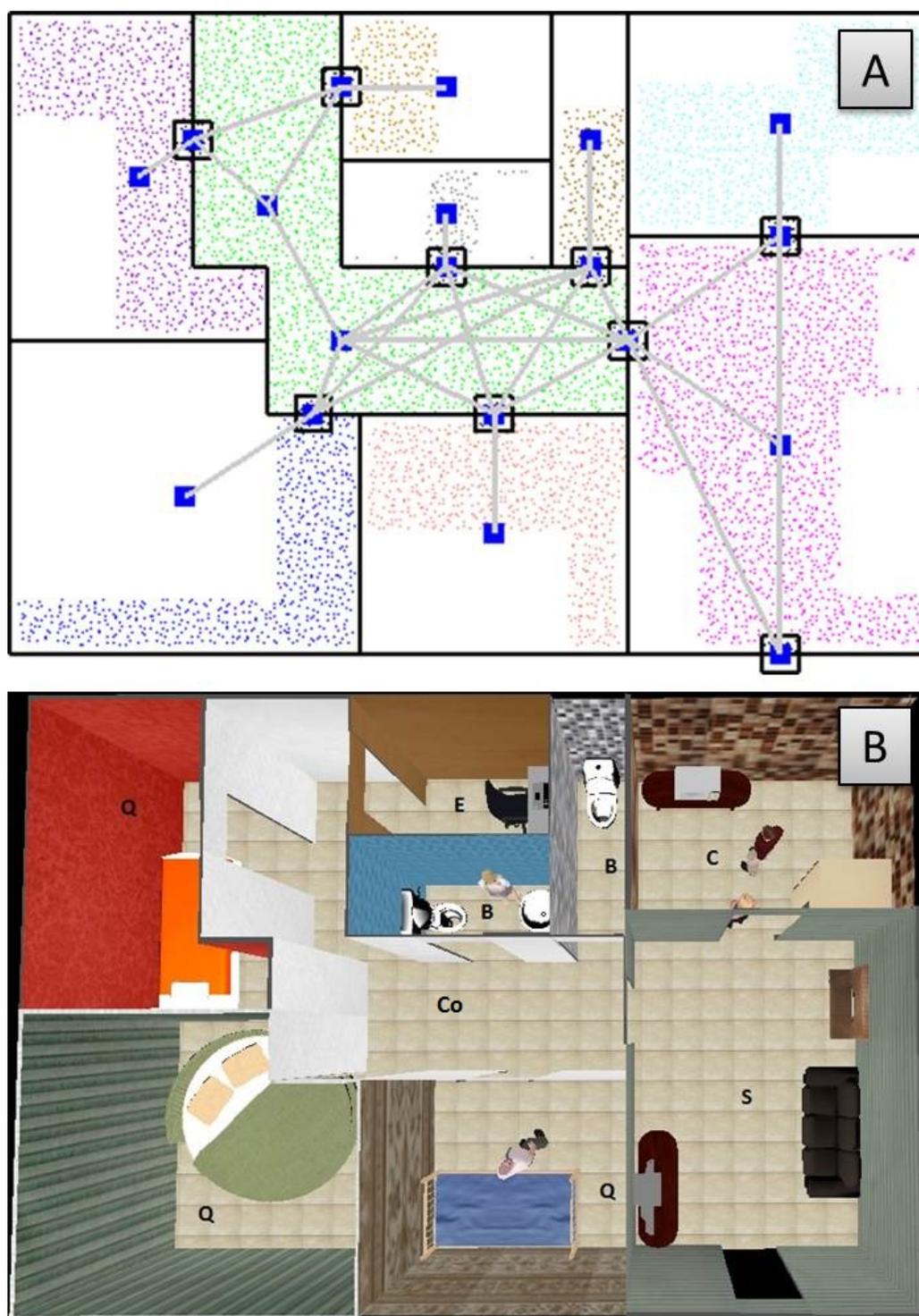


Figura 4.29: Ambiente utilizado pela família de ID=6, com três dormitórios. Em A, o ambiente é visualizado no simulador de ambientes internos, juntamente com o grafo de locomoção dos agentes. Em B, a visualização do ambiente em 3D (letras maiúsculas representam os cômodos).

A Figura 4.30 mostra o ambiente com apenas dois dos membros presentes, devido à saída dos demais. As Figuras 4.31 e 4.32 mostram imagens dos agentes em diferentes

cômodos da casa. Por fim, a Figura 4.33 mostra alguns agentes entrando na cozinha com a câmera próxima.



Figura 4.30: Imagem com dois dos agentes membros da família de ID=6.



Figura 4.31: Imagem dos agentes membros da família de ID=6 em diferentes cômodos da casa.



Figura 4.32: Imagem dos agentes membros da família de ID=6 em diferentes cômodos da casa.

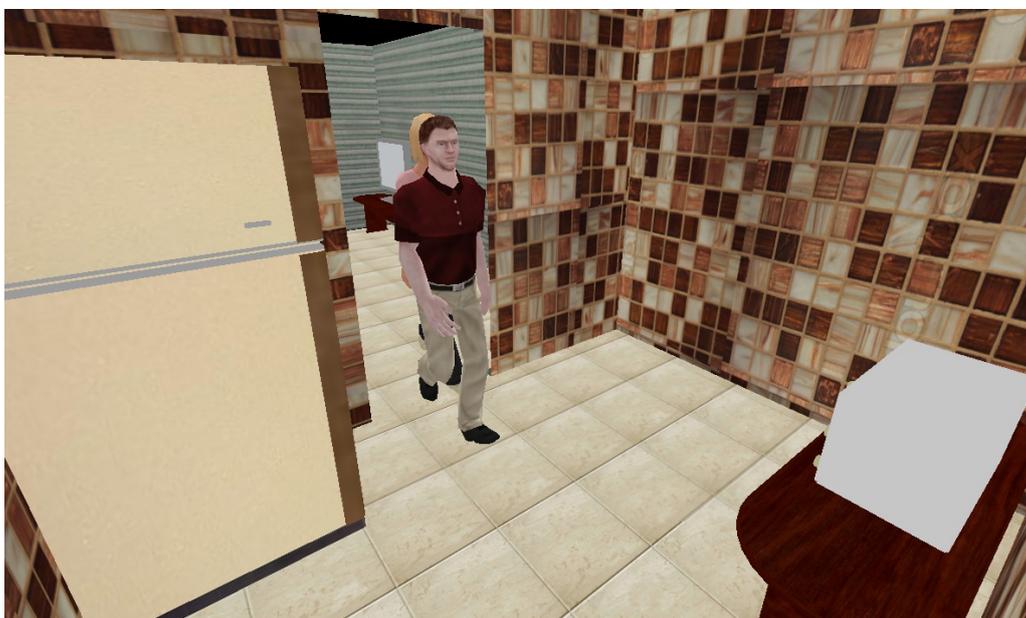


Figura 4.33: Imagem de dois dos agentes membros da família de ID=6 entrando na cozinha com a câmera próxima.

Nas Figuras 4.34, 4.35, 4.36, 4.37, 4.38 e 4.39, são demonstrados os valores dos atributos e as ações realizadas pelos agentes de ID=0 de cada família (famílias de ID=1, ID=2, ID=3, ID=4, ID=5 e ID=6, respectivamente) nas simulações de onde as imagens contidas neste capítulo foram retiradas. Todas as simulações foram iniciadas às 7h.

Hora	Atributos					Ações		
	E	F	Hi	H	N	Nome	Atributo Afetado	Local
7	0.6	0.6	0.1	0.3	0.6	Necessidades Fisiológicas	N	Banheiro
8	0.8	0.8	0.1	0.5	1	Comer	F	Cozinha
9	0.8	0.9	0.1	0.5	0.2	Continua na mesma ação		
10	0.8	0.9	0.1	0.5	0.6	Tomar Banho	Hi	Banheiro
11	0.8	0.2	0.1	0.5	0.6	Ler	H	Dormitório
12	0.7	0.5	1	0.5	0.6	Sair de Casa	-	Porta da Rua
13	Agente se encontra fora de casa.							
14								
15								
16								
17								
18								
19								
20								
21								
22								
23	0.1	0.2	0.4	0.5	0.6	Dormir	E	Dormitório

Figura 4.34: Tabela com os valores de cada atributo do agente de ID=0, membro da família de ID=1, juntamente com as ações realizadas pelo mesmo ao decorrer de um dia (início às 7 horas).

Hora	Atributos					Ações		
	E	F	Hi	H	N	Nome	Atributo Afetado	Local
7	0.6	0.6	0.1	0.3	0.6	Esperando desocupar banheiro	-	Corredor
8	0.8	0.8	0.1	0.5	1	Necessidades Fisiológicas	N	Banheiro
9						Sair de casa	-	Porta da Rua
10	Agente se encontra fora de casa.							
11								
12								
13								
14								
15								
16								
17								
18								
19	0.5	0.6	0.4	0.8	0	Ler	H	Dormitório
20	0.4	0.9	0.4	0.8	0.4	Comer	F	Cozinha
21	0.3	0.9	0.3	1	0.4			
22	0.2	0.9	0.3	0.9	0.4			
23	0.1	0.7	0.5	0.7	0.6	Dormir	E	Dormitório

Figura 4.35: Tabela com os valores de cada atributo do agente de ID=1, membro da família de ID=2, juntamente com as ações realizadas pelo mesmo ao decorrer de um dia (início às 7 horas).

Hora	Atributos					Ações		
	E	F	Hi	H	N	Nome	Atributos Afetados	Local
7	0.1	0.3	0.7	0.1	0.3	Sleeping	E	Dormitório
8	0.1	0.6	0.7	0.1	0.4			
9	0.1	0.8	0.6	0.1	0.5			
10	0.1	0.8	0.6	0.1	0.6			
11	0.1	0.9	0.5	0.1	0.8	Esperar desocupar banheiro	-	Corredor
12	1	0.9	0.5	0.1	0.8	Necessidades Fisiológicas	N	Banheiro
13	1	1	0.4	0.1	0.2	Comer	F	Cozinha
14	1	1	0.4	0.1	0.6	Sair de Casa	-	Porta da Rua
15	Agente se encontra fora de casa.							
16								
17								
18								
19	0.7	0	0.1	0.6	0.2	Tomar Banho	Hi	Banheiro
20	0.7	0.2	1	0.5	0.4	Computador	H	Escritório
21	0.6	0.5	1	0.5	0.6	Ler	H	Sala de Estar
22	0.5	0.8	1	1	0.6			
23	0.5	0.8	1	0.5	0.6	Comer	F	Cozinha
0	0.4	0.8	1	1	0.6			
1	0.4	0.8	1	0.9	0.6			
2	0.3	0.8	0.9	0.8	0.6	Dormir	E	Quarto

Figura 4.36: Tabela com os valores de cada atributo do agente de ID=2, membro da família de ID=3, juntamente com as ações realizadas pelo mesmo ao decorrer de um dia (início às 7 horas).

Hora	Atributos					Ações		
	E	F	Hi	H	N	Nome	Atributos Afetados	Local
7	0.6	0.6	0.1	0.3	0.6	Necessidades Fisiológicas	N	Banheiro
8	0.8	0.8	0.1	0.5	0	Sair de Casa	-	Porta da Rua
9	Agente se encontra fora de casa.							
10								
11								
12								
13								
14								
15								
16								
17								
18								
19	0.5	0	0.4	0.8	0.2	Computador	H	Escritório
20	0.4	0.3	0.4	1	0.6			
21	0.3	0.3	0.3	0.9	0.6			
22	0.2	0	0.3	0.9	0.6	Dormir	E	Dormitório

Figura 4.37: Tabela com os valores de cada atributo do agente de ID=0 membro da família de ID=4 juntamente com as ações realizadas pelo mesmo ao decorrer de um dia (início às 7 horas).

Hora	Atributos					Ações		
	E	F	Hi	H	N	Nome	Atributos Afetados	Local
7	0.6	0.6	0.1	0.3	0.6	Esperar desocupar banheiro	-	Corredor
8	0.8	0.9	0.1	0.5	1	Necessidades Fisiológicas Comer	N F	Banheiro Cozinha
9	0.8	0.9	0.1	0.5	0.2	Tomar Banho	Hi	Banheiro
10	0.8	0.2	0.1	0.5	0.6	Televisão	H	Sala de Estar
11	0.8	0.4	0.9	0.5	0.6			
12	0.7	0.7	0.9	1	0.6	Sair de Casa	-	Porta da Rua
13	Agente se encontra fora de casa.							
14								
15								
16								
17								
18								
19								
20								
21								
22								
23	0.1	0.4	0.3	0.5	0.6	Dormir	E	Dormitório

Figura 4.38: Tabela com os valores de cada atributo do agente de ID=1 membro da família de ID=5 juntamente com as ações realizadas pelo mesmo ao decorrer de um dia (início às 7 horas).

Hora	Atributos					Ações		
	E	F	Hi	H	N	Nome	Atributos Afetados	Local
7	0.6	0.6	0.1	0.3	0.6	Necessidades Fisiológicas	N	Banheiro
8	0.8	0.9	0.1	0.5	0	Sair de Casa	-	Porta da Rua
9	Agente se encontra fora de casa.							
10								
11								
12								
13	0.8	0.2	0.7	0.9	0.6	Televisão	H	Sala de Estar
14	0.8	0.4	0.7	0.8	0.6			
15	0.8	0.6	0.6	0.9	0.6	Ler	H	Dormitório
16	0.7	0.9	0.6	0.9	0.6	Comer	F	Cozinha
17	0.6	0.9	0.5	0.9	0.6	Computador	H	Escritório
18	0.5	0.2	0.5	0.8	0.6			
19	0.5	0.4	0.4	0.8	0.6			
20	0.4	0.7	0.4	0.8	0.6	Televisão	H	Sala de Estar
21	0.3	0.7	0.3	1	0.6			
22	0.2	0.7	0.3	0.9	0.6	Dormir	E	Quarto

Figura 4.39: Tabela com os valores de cada atributo do agente de ID=2 membro da família de ID=6 juntamente com as ações realizadas pelo mesmo ao decorrer de um dia (início às 7 horas).

As simulações demonstradas anteriormente trazem apenas alguns exemplos de situações possivelmente geradas pelo modelo. Estas situações ainda podem variar ao passo que a família se modifica, juntamente com os horários de cada membro. Outro fator que altera bastante a execução das ações pelos agentes, são as tabelas de variação padrão de seus atributos, detalhadas no Capítulo 3.

4.1 Avaliação

A avaliação do modelo desenvolvido neste trabalho não é trivial devido ao nível de subjetividade que existe nos temas abordados como, por exemplo: características de uma família; modo como um agente se comporta durante o dia e etc. Essas características são muito relativas e complexas, sendo influenciadas por inúmeras variáveis e passíveis de opiniões diversificadas. Devido a essa dificuldade, se resolveu utilizar três atributos para medir a efetividade do modelo procedural, retirados do trabalho de David Fletcher et al. [9]. Estes atributos são: nível de interação do usuário, o tamanho e qualidade/realismo do modelo produzido.

5. Considerações Finais

Este trabalho apresentou um modelo cujo objetivo é gerar pequenos grupos (famílias) para popular ambientes virtuais gerados de forma procedural. Além da criação destes grupos, o modelo simula comportamentos para cada membro do mesmo, sendo estes comportamentos coerentes com o ambiente, características do próprio agente e com o tempo. O modelo é totalmente procedural, portanto não necessita de nenhuma interação com usuário durante sua execução, sendo essa interação restrita apenas aos dados de entrada (que uma vez criados não necessitam ser modificados ou criados novamente, dependendo apenas da necessidade/desejo do usuário).

A utilização dados do Censo [20] em forma de arquivo para a geração das famílias permitiu que o trabalho tenha resultados coerentes com a realidade. A utilização de *seeds* e de métodos randômicos que utilizam as mesmas possibilitou que o modelo não necessite de estruturas de dados pesadas para manter suas informações, pois elas são geradas novamente quando necessário. Essa característica possibilita a persistência dos dados. A persistência e coerência dos comportamentos em relação ao tempo também é garantida com a utilização de curvas de variação padrão, como explicado no Capítulo 3, para calibrar os atributos de *status* dos agentes. Essa calibração possibilita que o usuário interrompa a visualização da simulação pelo tempo que desejar e, ao voltar a visualizá-la, as ações também voltam a ser executadas como se aquele tempo tivesse realmente passado, porém sem que o modelo tenha se mantido executando enquanto não visualizado.

O modelo também garante a coerência das ações com o ambiente/cômodos da casa, inclusive restringindo a entrada de um agente em um banheiro quando este estiver ocupado, além de fazer com que os agentes saiam do ambiente no horário em que devem realizar atividades externas. Também evita a colisão de agentes com os móveis presentes no ambiente ao não gerar marcadores onde estes estão situados. Todas estas características têm como objetivo tornar a simulação o mais real possível, além de automática.

A utilização de modelos procedurais para, além de gerar ambientes virtuais, também povoar estes ambientes de forma automática evitando uma constante necessidade de interação e modelagem pode trazer muitos ganhos para as áreas de jogos e simulação. Resultados foram gerados a fim de comprovar a viabilidade de utilização deste modelo nestas áreas e também avaliar se atingiu seus objetivos iniciais. Como ainda é uma área não explorada e com um tema muito subjetivo, como já abordado anteriormente na Seção 4.1, este trabalho tem uma avaliação muito complexa. Levando-se em consideração os atributos de avaliação citados também na Seção 4.1, o modelo não exige interação durante sua execução (sendo a interação com o usuário mínima e restringida apenas aos dados de entrada), tendo um realismo satisfatório por se basear em informações reais e, por fim, possuir um

nível de detalhe que se restringe a características pessoais de cada agente e do ambiente, tornando a visualização convincente.

Dentre as principais melhorias que podem ser consideradas como trabalhos futuros, pode-se citar a criação de quadras e até mesmo cidades dentro de uma mesma simulação (ruas com vários ambientes e famílias distintos). A inclusão de semântica nos móveis seria interessante para possibilitar que os agentes realizem ações coerentes com os mesmos também de forma automática. A fim de tornar o modelo ainda mais realista, seria interessante possibilitar a interação entre agentes, aumentando assim a complexidade do modelo.

De modo geral, o modelo conseguiu atingir os objetivos apresentados de modo satisfatório, conseguindo gerar resultados coerentes e bastante variados. É promissor por criar uma base em uma área a ser ainda explorada e com muito potencial.

Bibliografia

- [1] Abbeel, P.; Ng, A. Y. "Apprenticeship learning via inverse reinforcement learning". In: Proceedings of the 21st International Conference on Machine Learning, 2004, pp. 1-8.
- [2] Antonissen, A.; Riff, M. C. "An ant based approach for generating procedural animations". In: Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 2010, pp. 19-26.
- [3] Bicho, A. L. "Da modelagem de plantas à dinâmica de multidões: um modelo de animação comportamental bio-inspirado". Tese de Doutorado, Programa de Pós-Graduação em Engenharia Elétrica, UNICAMP, 2009, 114p.
- [4] BLIZZARD INC. "World of Warcraft Game". Capturado em: <http://us.blizzard.com/pt-br/games/wow/>, Janeiro 2013.
- [5] Bruls, M.; Huizing, K.; van Wijk, J. "Squarified treemaps". In: Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization, 2000, pp. 33-42.
- [6] Cassol, V. J.; Marson, F. P.; Vendramini, M.; Paravisi, M.; Bicho, A. L.; Jung, C. R.; Musse, S. R. "Simulation of autonomous agents using terrain reasoning". In: Proceedings of the 12th International Conference on Computer Graphics and Imaging (CGIM), 2011, pp. 101-108.
- [7] Conde, T.; Thalmann, D. "Learnable behavioural model for autonomous virtual agents: low-level learning". In: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems, 2006, pp. 89-96.
- [8] Ebert, D. S.; Musgrave, F. K.; Peachey, D.; Perlin, K.; Worley, S. "Texturing and modeling: a procedural approach". San Francisco, EUA: Morgan Kaufmann Publishers, 2002, 722p.
- [9] Fletcher, D.; Yue, Y.; Kader, M. A. "Challenges and perspectives of procedural modelling and effects". In: Proceedings of the 14th International Conference Information Visualisation, 2010, pp. 543-550.
- [10] FOX CORP. "Avatar Movie". Capturado em: <http://www.avatarmovie.com/>, Janeiro 2013.
- [11] Glass, R.; Morkel, C.; Bangay, D. "Duplicating road patterns in south african informal settlements using procedural techniques". In: Proceedings of the Afrigraph'06, 2006, pp. 161-169.

- [12] Gleicher, M.; Shin, H. J.; Kovar, L.; Jepsen, A. "Snap-together motion: assembling runtime animations". In: Proceedings of the Symposium on Interactive 3D Graphics, 2003, pp. 181-188.
- [13] Guy, S. J.; Kim, S.; Lin, M. C.; Manocha, D. "Simulating heterogeneous crowd behaviors using personality trait theory". In: Proceedings of the Symposium on Computer Animation, 2011, pp. 43-52.
- [14] Hall, E. T. "The hidden dimension". New York, EUA: Doubleday, 1966, 210p.
- [15] Hart, P. E.; Nilsson, N. J.; Raphael, B. "A formal basis for the heuristic determination of minimum cost paths", IEEE Transactions on Systems Science and Cybernetics, vol. 4, Jul 1968, pp. 100-107.
- [16] Hodgins, J. K.; Pollard, N. S. "Adapting simulated behaviors for new characters". In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'97), 1997, pp. 153-162.
- [17] Hopcroft, J. E.; Ullman, J. D. "Introduction to automata theory, languages and computation". Boston, EUA: Addison-Wesley, 1979, 418p.
- [18] Hughes, R. L. "A continuum theory for the flow of pedestrians", Transportation Research Part B Methodological, vol. 36, Jul 2002, pp. 507-535.
- [19] Hughes, R. L. "The flow of human crowds", Annual Review of Fluid Mechanics, vol. 35, Jan 2003, pp. 169-182.
- [20] IBGE. "Pesquisa Censo 2010". Capturado em: <http://www.censo2010.ibge.gov.br/>, Janeiro 2013.
- [21] IBGE. "Pesquisa Censo 2010: Resultados da Cidade de Porto Alegre". Capturado em: <http://www.ibge.gov.br/cidadesat/link.php?uf=rs>, Janeiro 2013.
- [22] Jeschke, S.; Birkholz, H.; Schumann, H. "A procedural model for interactive animation of breaking ocean waves". In: Proceedings of the 11th International Conference in Central Europe on Computer Graphics Visualization and Computer Vision (WSCG'03), 2003, pp. 3585-3630.
- [23] Karamouzas, I.; Overmars, M. "Simulating the local behaviour of small pedestrian groups". In: Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology (VRST'10), 2010, pp. 183-190.

-
- [24] Kavraki, L.; Latombe, J. C. "Randomized preprocessing of configuration space for fast path planning". In: Proceedings of the IEEE International Conference on Robotics and Automation, 1994, pp. 2138-2145.
- [25] Lai, Y. C.; Chenney, S.; Fan, S. "Group motion graphs". In: Proceedings of the ACM SIGGRAPH 2005 Papers, 2005, pp. 281-290.
- [26] Larsson, M.; Zalzal, J.; Duda, G. "A procedural ocean toolkit". In: ACM SIGGRAPH 2006 Sketches, 2006, pp. 18.
- [27] Lechner, T.; Watson, B.; Wilensky, U.; Tissue, S.; Felsen, M.; Moddrell, A. "Procedural modeling of urban land use". In: ACM SIGGRAPH 2006 Research Posters, 2006, pp. 135.
- [28] Marson, F.; Musse, S. R. "Automatic real-time generation of floor plans based on squarified treemaps algorithm", International Journal of Computer Games Technology, vol. 2010, Jan 2010, pp. 7:1-7:10.
- [29] Moussaid, M.; Perozo, N.; Garnier, S.; Helbing, D.; Theraulaz, G. "The walking behaviour of pedestrian social groups and its impact on crowd dynamics", PLOS ONE, vol. 5, Abr 2010, pp. 1-7.
- [30] Muller, P.; Wonka, P.; Haegler, S.; Ulmer, A.; van Gool, L. "Procedural modeling of buildings". In: Proceedings of the ACM SIGGRAPH 2006 Papers, 2006, pp. 614-623.
- [31] Musse, S. R.; Thalmann, D. "Hierarchical model for real time simulation of virtual human crowds", IEEE Transactions on Visualization and Computer Graphics, vol. 7, Abr-Jun 2001, pp. 152-164.
- [32] O'Rourke, J. "Computational geometry in c". Cambridge, Inglaterra: Cambridge University Press, 1998, 392p.
- [33] Reynolds, C. W. "Flocks, herds and schools: a distributed behavioral model". In: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, 1987, pp. 25-34.
- [34] ROCKSTAR INC. "Grad Theft Auto IV Game". Capturado em: <http://www.rockstargames.com/IV/>, Janeiro 2013.
- [35] Runions, A.; Fuhrer, M.; Lane, B.; Federl, P.; Rolland-Lagan, A.; Prusinkiewicz, P. "Modeling and visualization of leaf venation patterns". In: Proceedings of the ACM SIGGRAPH 2005 Papers, 2005, pp. 702-711.

- [36] Sachs, T. "Polarity and the induction of organized vascular tissues", *Annals of Botany*, vol. 33, Mar 1969, pp. 263-275.
- [37] Sachs, T. "The control of the patterned differentiation of vascular tissues", *Advances in botanical research*, vol. 9, Dez 1981, pp. 152-262.
- [38] Sung, M.; Kovar, L.; Gleicher, M. "Fast and accurate goal-directed motion synthesis for crowds". In: *Proceedings of the ACM SIGGRAPH 2005 Papers*, 2005, pp. 291-300.
- [39] Treuille, A.; Cooper, S.; Popovic, Z. "Continuum crowds". In: *Proceedings of the ACM SIGGRAPH 2006 Papers*, 2006, pp. 1160-1168.
- [40] UBISOFT INC. "Assassin's Creed Game Series". Capturado em: <http://assassinscreed.ubi.com>, Janeiro 2013.
- [41] WARNER BROS. INC. "The Lord of the Rings: The Return of the King Movie". Capturado em: <https://www.warnerbros.com/lord-rings-return-king>, Janeiro 2013.
- [42] Witkin, A.; Kass, M. "Spacetime constraints". In: *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, 1988, pp. 159-168.



Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br