



Dissertação de Mestrado em Geomática

***Simulação da Disseminação de Favelas em Ambientes Urbanos
Através de Agentes Reativos***

Autor: Jamil Coimbra Diuana

Orientador: Prof. D.Sc. Oscar Luiz Monteiro de Farias

Co-orientador: Prof. D.Sc. Orlando Bernardo Filho

Programa de Pós – Graduação em Engenharia da Computação
Área de concentração: Geomática

2007



Faculdade de Engenharia

***Simulação da Disseminação de Favelas em Ambientes Urbanos
Através de Agentes Reativos***

Jamil Coimbra Diuana

Dissertação submetida ao corpo docente da Faculdade de Engenharia da Universidade do Estado do Rio de Janeiro – UERJ, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia da Computação.

Orientador: Prof. D.Sc. Oscar Luiz Monteiro de Farias

Co-orientador: Prof. D.Sc. Orlando Bernardo Filho

Programa de Pós – Graduação em Engenharia da Computação
Área de concentração Geomática

Rio de Janeiro
2007

DIUANA, JAMIL COIMBRA

Simulação da Disseminação de Favelas
em Ambientes Urbanos Através de Agentes
Reativos [Rio de Janeiro] 2006.

xii, 74 p. 29,7cm (FEN/UERJ, M.Sc.
Programa de Pós-Graduação em Engenharia
da Computação – Área de Concentração
Geomática, 2007)

Dissertação – Universidade do Estado do
Rio de Janeiro – UERJ

1. Dinâmica ocupacional
 2. Simulação
 3. Agentes reativos
 4. Disseminação de favelas
 5. Sistemas de informação geográfica.
- I.FEN/UERJ II. Título (série)

FOLHA DE JULGAMENTO

Título: *Simulação da Disseminação de Favelas em Ambientes Urbanos
Através de Agentes Reativos*

Candidato: Jamil Coimbra Diuana

Programa de Pós-Graduação em Engenharia da Computação

Área de Concentração: Geomática

Data da Defesa: 27 de fevereiro de 2007

Aprovada por:

Orientador: Prof. Oscar Luiz Monteiro de Farias – D.Sc./UERJ

Co-orientador: Prof. Orlando Bernardo Filho – D.Sc./UERJ

Neide dos Santos – D.Sc./UERJ

Teresa Cristina Veiga – D.Sc./UFRJ

Dedicatórias

Dedico este trabalho aos meus pais, Jamil Elias Diuana e Maria Lyette Coimbra Diuana, à minha esposa Denise Maria, aos meus filhos Fabio, Felipe e Renata Maria e à pequenina Rosane Maria.

Agradecimentos

Para ingressar nesta jornada de pesquisas e estudos e concluí-la com sucesso, certas pessoas foram fundamentais. A elas, meus sinceros agradecimentos.

A meus pais, por minha formação, a minha esposa, Denise Maria e aos meus filhos Fabio, Felipe e Renata Maria. A Vilma, minha irmã.

Aos orientadores, Prof. Oscar Luiz Monteiro de Farias e Prof. Orlando Bernardo Filho.

Aos colegas do Departamento de Cartografia do IBGE, especialmente Lucy Mary Soares e Jayme Augusto Paiva, responsáveis pelo meu ingresso na pós-graduação, ao me estimularem a realizar o concurso ao Programa de Pós-Graduação em Engenharia da Computação Área de concentração: Geomática.

Aos colegas Alberto Delou, Mônica Fontes e Genésio Vasconcellos, por seu companheirismo neste período de estudo.

A Roberto Pereira, pelo estímulo a meu trabalho e aos companheiros de setor: Nelson de Barros Alves, Osmar Pereira Cardoso Filho, Antonio Carlos Ferreira Pascoal, Filomenia Ferreira de Oliveira, Ari Ferreira de Araújo, Luiz Cláudio Sarmento, Camilo Thomaz, Hamilton Ramos, Sonia Nascimento e Carlos Alberto Martins.

Aos amigos: Eduardo Porto Abrahão, Elzio Ganin, Daniela Engel Aduan Janoski, arquiteta da equipe Arqui-Traço, a Luiz Carlos Santos, engenheiro do Instituto Pereira Passos, a Miguel Ângelo Coimbra Thomé, também engenheiro, e a Claudia Oliveira, matemática, por suas contribuições com informações necessárias à realização da dissertação.

Aos professores pelo empenho e dedicação dados neste período.

A DEUS que me guia nos momentos mais difíceis da vida.

Resumo da Dissertação apresentada à FEN/UERJ como parte dos requisitos necessários para obtenção do grau de Mestre (M.E.) em Engenharia de Computação - Área de Concentração Geomática

Simulação da Disseminação de Favelas em Ambientes Urbanos

Através de Agentes Reativos

Autor: Jamil Coimbra Diuana

2007

Orientador: Prof. D.Sc. Oscar Luiz Monteiro de Farias

Co-orientador: Prof. D.Sc. Orlando Bernardo Filho

Programa de Pós – Graduação em Engenharia da Computação
área de concentração Geomática

Na maioria das grandes cidades do terceiro mundo e de países em desenvolvimento o crescimento acelerado das favelas está pressionando os serviços públicos (água, energia, transporte, saneamento básico e saúde), degradando por completo o padrão de vida da população e criando um ambiente favorável ao agravamento da violência, do tráfico de drogas etc. Construir cenários relacionados ao crescimento das favelas já existentes, ao nascimento de novas e a sua evolução é útil para o planejamento urbano e elaboração de políticas sociais que minimizem o impacto negativo causado pelo crescimento descontrolado das favelas. Esta dissertação busca construir um modelo de simulação baseado em agentes reativos, que permita analisar a ocupação do espaço urbano. O ambiente em que os agentes se encontram imersos e que descreve o espaço geográfico urbano é representado com o apoio de imagens do fotográficas tipo “raster”. A análise leva em conta diferentes situações e envolve diversos aspectos, tais como: econômicos, migratórios e ambientais. Na simulação empregou-se o *software* NetLogo 3.02, um sistema particularmente direcionado para o desenvolvimento de modelos e simulações de fenômenos naturais, sociais e biológicos baseados em agentes reativos.

Palavras-Chaves: Simulação, Agentes autônomos, Agentes reativos, Ocupação do solo urbano, Crescimento de favelas.

Abstract of Dissertation presented to FEN/UERJ as a partial fulfillment of the requirements for
the degree of Master of Engineering (M.E.)

Simulation of the Dissemination of Slums in Cities through Reactive Agents

Author: Jamil Coimbra Diuana

2007

Advisors: Prof. D.Sc Oscar Luiz Monteiro de Farias
Prof. D.Sc Orlando Bernardo Filho

Post-Graduation Program in Computer Engineering, Field of Geomatics

In the majority of third world great cities and developing countries the growing rate of slums in great cities is pressuring the public services (water supply, energy, transport, sanitation, etc.), and bringing violence, traffic of drugs, etc., degrading people's entire standard of life. Building scenarios related to the growing of existing slums, the birth of new ones, and their evolution will be very useful for urban planning and for the elaboration of social policies aiming to minimize the impact caused by the uncontrolled growing of slums. This dissertation looks for to evaluate and to set up a study based in multiagent systems that allows to analyze the occupation of the slums via reactive agents supported by a raster cartographic image. The analysis will taken into account many different aspects as: Economics, Migration and Environment. The simulation was done using the NetLogo 3.02 software that is a programmable modeling environment for simulating natural, biological and social phenomena.

Keywords: Simulation, Autonomous agents, Reactive agents, Urban land occupation and Slums spreading.

ÍNDICE

CAPÍTULO 1 – INTRODUÇÃO.....	1
1.1 Objetivo.....	1
1.2 Justificativa.....	1
1.3 Motivação.....	1
CAPÍTULO 2 – O NASCIMENTO E DESENVOLVIMENTO DAS FAVELAS NO BRASIL E NO MUNDO	5
2.1 A Presença das Favelas no Mundo.....	5
2.2 História	7
2.3 Crescimento das Favelas.....	8
2.4 O Impacto das Favelas nas Cidades.....	13
2.5 Diagnóstico da Área Escolhida.....	13
CAPÍTULO 3 – ALGUNS ASPECTOS DA TEORIA DE SISTEMAS MULTIAGENTES	15
3.1 Introdução.....	15
Um sistema multiagentes é um conjunto de agentes interagindo em um ambiente comum, onde um agente é uma entidade vivendo neste ambiente e capaz de modificar o ambiente (decisão, ação), e a si próprio (percepção, raciocínio, aprendizado)e de comunicar-se com outros agentes [9].	15
3.2 Agentes Inteligentes.....	15
Um agente é um sistema computacional situado em um meio-ambiente e que é capaz de ações autônomas neste meio-ambiente para poder alcançar os objetivos desejados [28].....	16
3.3 Tipos de Agentes	20
Agente reativo simples é o tipo mais simples de agente. Estes agentes selecionam ações com base na percepção atual, ignorando o restante do histórico das percepções. Um agente que se baseie apenas em sua posição e consiga reconhecer o que existe em sua posição atual e efetua uma ação, é um exemplo de agente reativo simples. Formalmente, o comportamento do agente reativo pode ser representado pela função:.....	20
Nosso agente de termostato é um exemplo de um agente reativo [28]. Assume, sem perda de generalidade, que o ambiente do termostato pode estarem um dos dois estados - muito frio ou temperatura OK. Então a função de ação do termostato simplesmente é:.....	20
se e = temperatura OK, senão	21
Agente cognitivo é um tipo de agente mais sofisticado que pode ser dividido em quatro componentes: i) o elemento aprendizado,responsável pela execução de aperfeiçoamentos; o ii) o elemento desempenho,responsável pela seleção das ações externas; iii) o crítico informa ao elemento aprendizado como o agente está se comportando em relação a um padrão desejado de desempenho; iv) o gerador de novas idéias, responsável por sugerir ações que levarão às experiências novas e informativas. [29].....	21
Em uma teoria que explique como um agente inteligente deve se comportar, a idéia de agentes como provadores de teoremas é sedutora. Para ver como esta idéia poderia funcionar desenvolve-se um modelo simples de lógica baseado em agentes que nós chamaremos de agentes deliberativos. Por exemplo, o banco de dados do agente poderia conter fórmulas como:	21
Abra (válvula221).....	21
Temperatura (reator4567, 321).....	22
Pressão (tanque776, 28).....	22
Não é difícil ver como fórmulas como estas podem ser usadas para representar as propriedades de um ambiente. O banco de dados é a informação que o agente tem sobre seu ambiente, é constituído por fórmulas de lógica de 1ª ordem e cálculo de predicados. Veja o programa na figura 3.2 abaixo;.....	22
	22

Assim, na primeira parte do programa (linhas 3–7), o agente executa cada de suas possíveis ações, e tenta provar a fórmula. Se o agente tem sucesso em provar Do(a), então a retorna como uma ação a ser executada.....	22
Deste modo, o comportamento do agente é determinado pela regras de dedução do agente (seu “programa”) e seu banco de dados atual (representando a informação que o agente tem sobre seu ambiente).....	22
“Means-end reasoning” é o processo de decidir como alcançar um fim (i.e. uma intenção que você tem) usando os meios disponíveis (i.e. as ações que você quer executar). É conhecido na comunidade de inteligência artificial como planejamento. Planejamento é essencialmente programação automática. As entradas do sistema são:.....	22
1)A meta, intenção ou uma tarefa. É alguma coisa que o agente deseja alcançar.....	22
2)O estado atual do ambiente.....	23
3)As ações disponíveis para o agente.....	23
Como output um algoritmo de planejamento gera um plano. Espera-se que, ao ser executado o plano, a partir do estado em que o mundo é descrito, o objetivo, intenção ou tarefa será alcançado.....	23
Há duas características definindo a arquitetura de “subsumption”. A primeira é que a decisão de criação de um agente é compreendida através de um conjunto de tarefas comportamentais, cada comportamento pode ser como uma função de ação individual. Cada um desses comportamentos pretende realizar uma tarefa em particular. A implementação de Rodney Brooks assume não incluir representações simbólicas complexas, ou quase nenhuma.....	23
Em muitas implementações, esses comportamentos são implementados como regras da forma:.....	24
Situação → Ação.....	24
A segunda característica é que muitos comportamentos podem ser disparados simultaneamente. A simplicidade computacional é a grande força desta arquitetura [28].....	24
3.4 Simulação Envolvendo Agentes de Software.....	24
A teoria dos sistemas complexos é um novo campo da ciência que estuda até que ponto as partes de um sistema se separem do comportamento coletivo, e como o sistema interage com seu ambiente. Sistemas complexos são difíceis de entender porque as causas e os efeitos não estão relacionados de forma direta, ou seja, existem correlações entre eventos distantes no tempo e no espaço.....	24
A idéia é que é possível simular sistemas complexos tendo como ponto de partida elementos simples (agentes reativos) que atuam em um meio-ambiente. Estes agentes podem interagir entre si e com o meio-ambiente de modo simples. Como resultado da interação entre os agentes e o meio-ambiente o comportamento do sistema complexo (do macrocosmo) pode ser explorado e analisado.	24
Na simulação de sistemas complexos a partir de agentes reativos simples podem-se obter resultados formais sobre a dinâmica global do sistema [9].....	24
Pela simulação baseada em agentes é possível modelar situações nas quais os indivíduos têm comportamentos diferentes e complexos, podendo levar em conta tanto as propriedades quantitativas (parâmetros numéricos) quanto qualitativas (comportamento individual) do sistema representado no modelo.....	24
A simulação no computador é, então, usada para descobrir propriedades do modelo e assim ganhar entendimento dentro de um processo dinâmico, o que seria muito difícil de modelar com técnicas matemáticas padrões. [33].....	25
Sistemas de simulação multiagentes provêm de uma plataforma computacional onde a dinâmica dos sistemas espaços-temporais podem ser estudadas. Para tanto pode se usar agentes reativos (mais simples) ou cognitivos (mais elaborados). De acordo com Ferrand Nils é possível simular sistemas complexos, que têm como ponto de partida somente os agentes reativos elementares inseridos em algum ambiente. Quando individualmente considerados, estes agentes reativos têm comportamento extremamente simples, e só interagem com outros agentes situados entre vizinhança próxima. Pela interação com o ambiente e entre eles mesmos, como resultado emerge o comportamento do sistema complexo.....	25

As hipóteses formuladas no modelo são validadas através da comparação da estrutura espacial simulada com os dados observados. O objetivo é reproduzir estruturas plausíveis e dinâmicas ao invés de tentar obter uma reprodução exata da realidade específica.....	25
Um modelo é sempre usado para eliminar as estruturas mais gerais de uma organização espacial com o objetivo de encontrar suas especificidades. Os resíduos do modelo são usados para medir essas especificidades.....	25
Um modelo quando bem calibrado, pode ser usado tanto para fazer previsões como montar cenários para examinar as alterações no sistema espacial [10].....	25
3.5 Escolha do Tipo de Agente	25
Através do exemplo das formigas, podem-se destacar as principais características dos agentes em sistemas multiagentes reativos:	26
CAPÍTULO 4 - SOFTWARE LOGO.....	26
4.1 História do LOGO	26
O LOGO foi criado em 1966 em uma firma de pesquisa de Cambridge a BBN (Bolt Beranek & Newman Inc.), por Wally Feurzeig e Seymour Papert. Suas raízes intelectuais estão em inteligência artificial, lógica matemática e psicologia. A tartaruga constituía, então uma inovação grande e recente. Os primeiros quatro anos de pesquisa do Logo, desenvolvimento e trabalho pedagógico foram terminados na BBN. A primeira implementação foi escrita em LISP (BALBUCIO) em um PDP-1. Seu nome era “Ghost”.	26
A preocupação desses pesquisadores concentrava-se principalmente em desenvolver hardware e software para implementar o interpretador Logo e em demonstrar o que se podia fazer com ele, tendo em vista principalmente sua relação com a matemática.	
Trabalhava-se com ele em computadores de médio e grande porte, o que também foi um dos fatores contribuintes para que seu uso de início, ficasse restrito à universidades e centros de pesquisa.....	27
O Logo não é agora muito diferente dos conceitos anteriores básicos da primeira tartaruga. A primeira tartaruga foi um rádio controlado (telegrafia sem fios) de chão nomeado "Irving". Irving teve sensor de toque e podia ir para frente, atrás, direita, esquerda (rotações), e ding (Irving teve um sino) emitia um som [34].....	27
4.2 Implementação do LOGO.....	27
4.3 Programação do LOGO.....	28
4.4 StarLogo.....	29
O StarLogo foi criado no MIT com uma idéia de atingir metas diferenciadas. O objetivo direto não era particularmente com desempenho ou velocidade. O interesse é em prover modos novos para usuários modelarem, controlarem, e pensarem em ações que de fato acontecem em paralelo. Muitas coisas no mundo realmente agem em paralelo; o modo mais natural para modelar tais situações é com uma linguagem de programação paralela. Nestes casos, paralelismo não é um “truque” para melhorar desempenho; é o modo mais natural de expressar o comportamento desejado.....	29
4.5 NetLogo.....	30
A parte "LOGO" é porque NetLogo é um dialeto do idioma de Logo. LOGO é uma linguagem de programação extremamente poderosa, freqüentemente mal-interpretada como se fosse somente para as crianças. A variante NetLOGO hoje é uma linguagem moderna. Com NetLOGO é possível simular ambientes com agentes interagindo autônomos. Estes sistemas simulados podem ser usados para modelar fenômenos como comportamento de eleitor, sistemas ecológicos, reações químicas, e fenômenos físicos. A força do ambiente que se usa é de acesso fácil à visualização dinâmica, e fácil de aprender.....	30
4.6 Definição do Software.....	33
CAPÍTULO 5 - ESTUDO DA IMAGEM “RASTER”.....	35
5.1 Imagem “raster”.....	35
5.2 Aerofoto.....	35
5.3 Análise da Imagem utilizada na Dissertação.....	43
CAPÍTULO 6 – DESENVOLVIMENTO DO SISTEMA COM O NETLOGO USANDO UMA IMAGEM DA ÁREA ESCOLHIDA.....	45

6.1 Fundamentos do Modelo e sua Implementação no NetLogo.....	45
Em primeiro lugar precisa-se representar o ambiente onde os agentes vivem e agem. Este ambiente deve estar em correspondência direta com o espaço geográfico, onde as cidades ficam situadas. Uma imagem (“raster”) é ideal para esses propósitos. Ela representa uma região de espaço ocupada por florestas, áreas públicas, bairros tradicionais, favelas, áreas ao longo dos rios, etc. Cada “pixel” ou célula representa uma parte da área em estudo, cujo tamanho real é fornecido pela escala associada à imagem. Os agentes seriam, em princípio, as contrapartes de software dos seres humanos, que são, por fim, os principais responsáveis pela ocupação da terra nas cidades. Cada tipo de uso da terra é mapeado em cores específicas. Assim, têm-se as áreas mapeadas (subconjunto de células) em cores, de acordo com o uso da terra, resultando então, em um mapa temático do uso da terra, por exemplo: i) áreas públicas como viadutos, praças, parques; ii) áreas de proteção ambiental - florestas urbanas, zonas ao redor de rios; iii) zonas residenciais ocupadas pela classe média e classe pobre; iv) distritos industriais etc.; v) edificações e terrenos (públicos ou particulares) abandonados etc.....	45
6.2 Programando o Netlogo.....	47
6.3 O Modelo.....	50
6.4 NetLogo Usando uma Imagem “raster”.....	53
O segundo passo foi montar um código no software NetLogo para poder ler o arquivo PPM. Com a montagem desse código foi possível viabilizar a modelagem prevista.	54
O NetLogo já lia o arquivo PPM, porém a geração da imagem no computador estava muito lenta inviabilizando o modelo.....	54
CAPÍTULO 7 – ANÁLISE DOS RESULTADOS.....	55
Com a comparação da imagem obtida na simulação do modelo NetLogo (figura 7.6) com a aerofoto de 2000 (figura 7.1) e os croquis (figura 7.2), da equipe Arqui-Traço responsável pelo diagnóstico da favela do Borel e Indiana no programa “Favela Bairro”, pode-se validar o modelo e atestar o êxito da simulação.....	59
CAPÍTULO 8 – CONCLUSÕES.....	61
REFERÊNCIAS.....	63
APÊNDICE A – CÓDIGO FONTE.....	66

ÍNDICE DAS FIGURAS

Figura 2.1 – População de favelas nos países em desenvolvimento [12].....	6
Figura 3.1 – Um agente interagindo com o meio-ambiente [28].....	16
Figura 3.2 – Exemplo do programa.....	22
Figura 3.3 – Esquema genérico de uma arquitetura reativa [32].....	23
Figura 4.1 – Obtenção do quadrado.....	29
Figura 5.1 – Geometria básica de uma fotografia aérea.....	40
Figura 5.2 – Vôo fotogramétrico.....	41
Figura 5.3 – Recobrimento lateral.....	42
Figura 5.4 – Perspectiva de 04 faixas de vôo.....	42
Figura 5.5 – Recobrimento com a ocorrência de deriva e desvio.....	43
Figura 5.6 – Efeitos da deriva e desvio [43].....	43
Figura 7.1 – Aerofoto da favela do Borel em 2000.....	55
Figura 7.2 – Evolução da favela do Borel representada na cor amarela.....	56
Figura 7.3 – Imagem classificada.....	57
Figura 7.4 – Evolução da ocupação da favela após a simulação das décadas de 20-40.....	58
Figura 7.5 – Evolução da ocupação da favela após a simulação das décadas de 50-70.....	59
.....	60
Figura 7.6 – Evolução da ocupação final das favelas e do supermercado Carrefour.....	60

ÍNDICE DOS QUADROS

Quadro 2.1 – População mundial vivendo em favelas [6].....	6
Quadro 2.2 – População mundial (mil) 1950-2030 [14].....	6
Quadro 2.3 – Número de favelas por cidade.....	10
Quadro 2.4 – População pobre - Brasil.....	11
Quadro 2.5 – População favelada do município do Rio de Janeiro.....	12
Quadro 2.6 – Comparativo população favelada Brasil versus município RJ.....	12
Quadro 2.7 – População do Borel.....	13

CAPÍTULO 1 – INTRODUÇÃO

1.1 Objetivo

Esta dissertação tem como objetivo elaborar um modelo para simular a dinâmica ocupacional de sistemas complexos, como a disseminação de favelas em espaços urbanos, fundamentado no uso de agentes reativos relativamente simples.

Para testar a viabilidade do modelo proposto nesta dissertação foi feito um estudo de caso onde foram cotejados os dados obtidos na simulação com os dados reais de crescimento da favela do Borel, localizada na Tijuca, no município do Rio de Janeiro, no período compreendido entre 1920 e 2000.

O modelo da simulação foi desenvolvido em NetLogo 3.02, uma plataforma de *software* particularmente adequada para a simulação de fenômenos naturais, sociais e biológicos, baseada em agentes reativos.

1.2 Justificativa

Este estudo, ao demonstrar a possibilidade de criar modelos por meio da aplicação da teoria de sistemas multiagentes, sinaliza também para a contribuição que a evolução da programação orientada a agentes pode proporcionar às pesquisas nos campos onde a geomática está presente.

Estas simulações de processos complexos, por meio da aplicação da teoria de sistemas multiagentes, usadas em conjunto com dados do sistema de informação geográfica, validados por comparações com dados reais, tornam possível representar a evolução de determinados fenômenos e criar cenários diversificados para analisar as possíveis consequências.-

1.3 Motivação

O recente relatório da Organização das Nações Unidas (ONU), intitulado Desafio das Favelas, afirma que no ano de 2003 pelo menos um bilhão de pessoas habitavam em favelas, com a porcentagem maior ocorrendo na Ásia, África e América Latina, e que, nas próximas três décadas esses números duplicarão, alcançando 32% da população mundial total [1].

O fenômeno da urbanização superou as predições. Em 1950 havia 86 cidades no mundo com uma população de mais de um milhão de habitantes; atualmente, em 2006, elas já somam 400, e antes de 2015, serão pelo menos 550. Nesse período, as cidades absorveram quase dois terços da explosão global da população e continuam crescendo, tanto pela natalidade como pela migração, chegando esse crescimento, atualmente em uma taxa de um milhão de pessoas por semana. A população urbana atual (3,2 bilhões) é maior que a população total do mundo em

1960. O segmento rural global enquanto isso alcançou sua população máxima (3,2 bilhões) e somente começará a decrescer após 2020. Como resultado dessa urbanização acelerada, as cidades terão um crescimento populacional mundial que deverá atingir aproximadamente 10 bilhões em 2050 [2].

No Brasil, dois em cada três brasileiros (65%) moram nas capitais, onde o número dos habitantes em favelas cresceu 39,3% na última década do século XX, passando de 3 milhões para 4,2 milhões. Esse crescimento ajudou a elevar para 6,5 milhões o número de indivíduos que moram nesse tipo de habitação em todo o país. Observa-se que mais da metade da população dos "aglomerados subnormais" está nos Estados de São Paulo (2,07 milhões) e do Rio (1,38 milhão) [3].

Em nove anos, surgiram 69 novas favelas no município do Rio de Janeiro. Esses números, calculados entre 1991 e 2000, são mais preocupantes quando se consideram as estatísticas populacionais. Nesse período, a população das cidades cresceu 6,77% (371 146 pessoas), já a população das favelas, que era constituída por 876 398 pessoas em 1991, subiu para 1 092 958 em 2000 — um crescimento de quase 25% [4].

Segundo a definição do IBGE (governo) para favelas, “Aglomerado Subnormal” (favelas e similares) é um conjunto constituído de, no mínimo, 51 unidades habitacionais (barracos, casas etc.), ocupando ou tendo ocupado até um período recente, terreno de propriedade alheia (pública ou particular,) dispostas, em geral, de forma desordenada e densa, bem como carentes, em sua maioria, de serviços públicos essenciais [5].

Segundo as Nações Unidas, a favela é um assentamento contíguo onde os habitantes são caracterizados como tendo habitação e serviços básicos inadequados. A favela nem sempre tem o seu reconhecimento pelas autoridades como uma parte integral da cidade [6].

Favela definida por Silva como área predominantemente habitacional, com ocupação por população de baixa renda, precariedade da infra-estrutura e de serviços públicos, vias estreitas e de alinhamento irregular, lotes de forma e tamanho irregular e construções não licenciadas em desconformidade com os padrões legais [7].

Utilizar as diferenças de concepção que se pode observar nas definições acima implica em diferentes modos de compreender e abordar não só as favelas como também seus habitantes, com conseqüências diferenciadas nas soluções e encaminhamentos propostos para os problemas que se colocam.

No Rio de Janeiro, as favelas representam 35% da área total da cidade. Em Caracas e Santiago do Chile, este número chega a 40%; e em Lima (Peru), a 50% [8].

A população inicial das favelas é originária da zona rural onde, a competição desigual entre as propriedades rurais tradicionais e a agroindústria, está transformando esta sociedade que não consegue mais reter as pessoas no campo. Tal fenômeno vem ocasionando um fluxo

migratório para os grandes centros urbanos de uma população sem a necessária qualificação profissional para trabalhar nos grandes centros e sem moradia, e assim as favelas surgem [2]. Os autores do Desafio das Favelas [1] concluem que em vez de ser um foco para crescimento e prosperidade, as cidades se tornaram um local que absorve um excesso de população sem qualificação profissional, que trabalha desprotegida, vinculada ao trabalho informal de baixos-salários e ao comércio.

Há várias razões para a existência e crescimento de favelas. As mais importantes são: i) o avanço rápido da migração rural para as cidades; ii) a taxa de crescimento econômico insuficiente para absorver a mão-de-obra que deseja entrar no mercado de trabalho; iii) falta de legislação que disciplina o uso da terra em áreas urbanas; iv) falta de execução de legislação existente sobre as terras em áreas urbanas; v) taxa de natalidade de habitantes da favela mais elevada que a taxa de natalidade de população tradicional em áreas urbanas; vi) falta de planejamento urbano.

Assim, pode-se concluir que, sob estes vários aspectos, a ocupação das favelas constitui um sistema espaço-temporal muito complexo que funciona e sofre impacto significativo das ações humanas. Se um modelo computacional for construído de modo similar ao mundo que conhecemos, pelo menos nos aspectos relevantes para a tomada de decisões, e se comportar mais ou menos como no mundo real, e se formos capaz de usá-lo para experimentações, então estará criado um laboratório virtual para tomada de decisões [9].

Atualmente as técnicas disponíveis para modelagem da dinâmica espacial compreendem: i) equações diferenciais; ii) autômato celular; iii) sistemas de multi-agentes [10].

Para esta dissertação, foi usado um sistema com múltiplos agentes reativos para simular o crescimento de favelas nas cidades, com a hipótese de que é possível simular sistemas complexos tendo como ponto de partida agentes reativos que podem interagir entre si e com o meio-ambiente de modo simples. Como resultado dessa interação entre eles, o comportamento do sistema complexo em nível macroscópico fica claramente evidenciado e pode então ser estudado e aplicado em análises, gerando resultados significativos. Para testar o sistema proposto nesta dissertação selecionou-se para estudo uma área do bairro da Tijuca, denominada Usina que reúne características pertinentes ao crescimento de favelas.

No capítulo dois será discutido o nascimento e desenvolvimento das favelas no Brasil e no mundo; no capítulo três desta dissertação será apresentado um estudo sobre a teoria de sistema multiagentes particularmente os agentes reativos; no capítulo quatro é abordada a conceituação do software Logo; no capítulo cinco será feita uma explanação sobre o que é imagem “raster”, utilizada nesta dissertação; no capítulo seis é apresentado o uso do Netlogo na montagem do sistema desta dissertação usando uma imagem da área escolhida; no capítulo sete

é exposta a análise dos resultados e, por fim, no capítulo oito as tem-se as conclusões e sugestões propostas para estudos futuros.

CAPÍTULO 2 – O NASCIMENTO E DESENVOLVIMENTO DAS FAVELAS NO BRASIL E NO MUNDO

2.1 A Presença das Favelas no Mundo

Um terço da população urbana do mundo (um bilhão de pessoas) vive em favelas, de acordo com o relatório da ONU sobre habitação, denominado Desafio das Favelas [1]. As favelas respondem por 38% do crescimento urbano mundial e segundo as projeções mais otimistas o número de seus habitantes deverá se elevar para 1,4 bilhão em 2020 — o equivalente à população da China. Para as Nações Unidas, a favelização é uma realidade que os governos não podem ignorar, pois caso a tendência atual se mantenha, as perspectivas de qualidade de vida para a população como um todo serão catastróficas.

A pobreza nas favelas é igual às das áreas rurais e em alguns casos maior. Nas favelas o acesso às condições de moradia e saneamento adequadas, à água potável, à educação e aos serviços de saúde é sempre precário. O índice de morte de crianças por diarreia, por exemplo, é maior em favelas do que em áreas rurais. Além disso, os pobres das favelas tendem a morrer mais cedo do que os das áreas rurais.

Kofi Annan, afirma: “A pobreza rural tem sido a face mais comum da destituição no mundo, mas a pobreza urbana pode ser tão intensa, desumanizadora e ameaçadora para a vida quanto” [1].

Segundo Anna Tibaijuka: “Quando uma massa crítica de pessoas está num lugar, se você não lhes dá poder, elas vão dar poder a si próprias por meio de revolução. Se quisermos evitar o caos, temos que dar poder aos pobres” [11].

O crescimento das favelas é acelerado e sem precedentes. A pior situação é a da África subsaariana, onde 72% da população urbana vive em áreas carentes que crescem a um índice de quase 100% em alguns países. Na América Latina, registra-se uma melhoria das condições, já que o percentual da população urbana que vive em favelas diminuiu de 35%, em 1990, para 31%, em 2005. Nesse passo, Argentina, Brasil e México são os países da região que terão a maior influência na redução das taxas de população em áreas carentes até 2020.

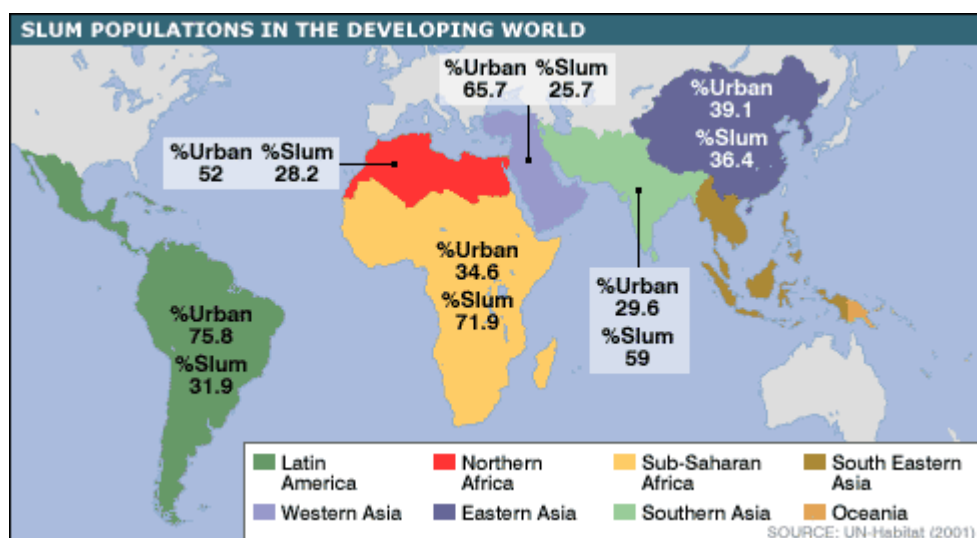


Figura 2.1 – População de favelas nos países em desenvolvimento [12]

Quadro 2.1 – População mundial vivendo em favelas [6]

Regiões do Mundo	População em Milhões
Regiões Desenvolvidas	54
América Latina e Caribe	128
África	188
Ásia	570
Outros	25
Mundo	965

Outra projeção impressionante é a de que a população urbana duplicará em 20 anos. Esse fenômeno seria consequência do crescimento nas áreas pobres que teriam uma população majoritariamente jovem (entre 15 e 49 anos) e, em menor escala, da migração do campo para as cidades.

Em 2030, a população urbana da África (o continente menos urbanizado do mundo) será maior que a da Europa, diz o relatório. Já a Ásia concentraria mais da metade da população urbana mundial (2,6 bilhões de um total de cinco bilhões). Quase quatro bilhões de pessoas estariam morando em cidades de países em desenvolvimento.

A paz e a estabilidade das cidades podem ser questionadas com a maioria da população vivendo em favelas. A única solução para melhorar as condições de vida dessa população, é convencer governos a melhorar a infra-estrutura das favelas e impedir que outras surjam [12,13].

Quadro 2.2 – População mundial (mil) 1950-2030 [14]

Ano	População	Urbana	Rural
1950	2 519 470	732 974	1 786 496
1955	2 757 399	852 556	1 904 844
1960	3 023 812	993 521	2 030 290
1965	3 337 974	1 159 063	2 178 910
1970	3 696 588	1 331 023	2 365 566
1975	4 073 740	1 518 425	2 555 314
1980	4 442 295	1 739 826	2 702 469
1985	4 843 947	1 989 845	2 854 102
1990	5 279 519	2 280 119	2 999 400
1995	5 692 353	2 564 960	3 127 392
2000	6 085 572	2 863 982	3 221 590
2005	6 464 750	3 177 456	3 287 293
2010	6 842 923	3 511 834	3 331 089
2015	7 219 431	3 867 755	3 351 677
2020	7 577 889	4 236 446	3 341 443
2025	7 905 239	4 610 560	3 294 679
2030	8 199 104	4 986 617	3 212 487

2.2 História

No Rio de Janeiro, durante muito tempo a história das favelas resumiu-se à narrativa do surgimento da favela da Providência, sobre o morro do mesmo nome, situado entre o centro e o porto da cidade e à origem do termo “favela”.

Esta história remete a 1897, quando o governo brasileiro determinou a destruição do grupo de seguidores do líder religioso Antonio Conselheiro, estabelecido no Arraial de Canudos, no sertão nordestino, por considerá-los fanáticos monarquistas e uma ameaça à segurança da recém instituída República. Com essa intenção, vários ataques foram realizados ao reduto de maltrapilhos até que, na quarta tentativa, um contingente de oito mil homens o destruiu inteiramente, massacrando todos os habitantes do arraial. Este episódio foi relatado por Euclides da Cunha, no clássico *Os Sertões*, de 1901, que, como correspondente no local, descreveu não apenas a guerra, mas o sertão, o vilarejo, o reduto rebelde e o morro que contornava Canudos, conhecido como o Morro da Favela, em alusão à planta que o recobria, denominada “favela”.

Após a destruição de Canudos, o grupamento de soldados enviados pelo governo retornou ao Rio de Janeiro, então capital do país, onde permaneceu acampado em praça pública, reivindicando sua re-incorporação ao exército. Para contornar a situação, as autoridades militares permitiram a ocupação do Morro da Providência, situado atrás do quartel general, e vários barracos de madeira foram construídos para abrigar esses soldados. Os novos moradores passaram a chamar o morro de “Morro da Favela” em alusão àquele de Canudos.

Por volta de 1920 a palavra favela passa do estatuto de nome próprio ao de substantivo nos jornais locais, passando a designar, a partir de então todos os “conjuntos de habitações

populares toscamente construídas (por via de regra nos morros)” que se espalhavam pelo Rio de Janeiro e depois pelo país inteiro.

Apenas recentemente, pesquisas de documentação de época mostraram que a “favela original” não foi aquela que difundiu o seu nome (Morro da Favela), mas a do morro de Santo Antônio, situada também no entorno imediato do núcleo histórico da cidade. Apesar de ser pouco conhecida, a história da favela de Santo Antônio (já desaparecida) tem origem semelhante à da favela da Providência. Há notícias da presença, em 1897, de 41 barracos no local; de 150, em 1901, e de 1314 em 1910.

Essa favela conheceu várias remoções (e incêndios criminosos) até sua total eliminação, quando o morro foi arrasado, na década de 50. Esta seria uma explicação para o contínuo interesse suscitado pelo Morro da Providência (Morro da Favela) e pela favela de mesmo nome, que ainda existe no local e já comemorou os seus 100 anos de existência [15].

2.3 Crescimento das Favelas

As primeiras manifestações da crise da moradia no Rio de Janeiro remetem ao período que compreende a segunda metade do século XIX e as três primeiras décadas do século XX. É o período de urbanização/industrialização, de mudanças de ordem econômica, social, política, cultural e espacial. Entre elas destacam-se: a abolição da escravidão, que culmina com a substituição do trabalho escravo pelo assalariado, a formação de mercados e a mercantilização de bens, a decadência da cafeicultura na província fluminense, o desenvolvimento dos setores secundário e terciário da economia, as grandes migrações, a definição de novas elites no poder, com a queda do Império e a proclamação da República. O crescimento da cidade do Rio de Janeiro foi intenso (235 000 habitantes em 1870, 522 000 habitantes em 1890) e foram criadas modernas infra-estruturas e serviços públicos.

No centro da cidade multiplicaram-se as moradias possíveis para esta população: os cortiços. Eram moradias de aluguel diminutas e precárias, geralmente superlotadas, com iluminação e ventilação insuficientes, que ocupavam os quintais dos antigos casarões térreos e sobrados. As más condições deste ambiente foram associadas à deterioração da saúde e à decadência moral das classes trabalhadoras.

Deslocados do centro da cidade e inviabilizados economicamente de se instalar em áreas distantes, os mais pobres foram levados a ocupar as áreas vazias nos morros vazios próximos ao centro da cidade (Santo Antônio, Providência e outros). Estes morros, que já apresentavam vestígios de moradias rústicas desde meados do século XIX, foram aos poucos ocupados, delineando-se as primeiras favelas. A partir daí, se iniciava a expansão das favelas.

Em 1907, já se encontravam grandes grupos de barracos nos morros de Copacabana, na zona sul, e, por volta de 1909/1910, nos morros do Salgueiro e da Mangueira, na zona norte da

cidade. Esta nova forma de moradia se expande rapidamente nos terrenos desocupados, de propriedade incerta e sem interesse para a construção imobiliária. Apesar do crescimento das favelas ter se tornado vertiginoso somente após os anos 1930, a estatística predial de 1933 já registrava 46.192 casebres (agrupados em favelas ou não), correspondendo a 20,58% do total de prédios da cidade.

A primeira contagem da população do Brasil foi realizada em 1872, ainda durante o Império; mas foi a partir de 1890, já sob a República, que o Censo Demográfico se tornou decenal. O Brasil mantém um excelente retrospecto de levantamentos regulares e inovadores do Censo Demográfico, tendo sido, por exemplo, o primeiro país a incluir questões sobre fecundidade e um dos poucos da América Latina a pesquisar rendimento.

Os anos 40 foi o período de mais forte proliferação de favelas no Rio de Janeiro. Foi nessa época que o primeiro censo oficial foi realizado, após as eleições de 1947, quando um grande apoio aos candidatos comunistas veio das favelas, preocupando o Governo. Apesar dos números deste censo terem sido controvertidos, ele se tornou o marco do reconhecimento oficial pelo Estado da existência das favelas, que já faziam parte da paisagem da cidade.

O censo nacional de 1950 mostrava novos números contraditórios: 59 favelas e 169 305 habitantes. Isto apontava para a falta de parâmetros comuns para fundamentar os critérios para recenseamento de favelas sendo necessário estabelecer uma clara definição, delimitação e classificação de favela. Uma definição apontava as favelas como “agrupamentos urbanos”, possuindo total ou parcialmente algumas dessas características: aglomeração mínima de 50 unidades, tipo de habitação precário, construção irregular, falta de infra-estrutura pública adequada e falta de urbanização.

Segundo fontes oficiais [16], no Rio de Janeiro, entre 1964 e 1974, 80 favelas, com 26 193 barracos foram removidas, deslocando uma população de 139.218 pessoas. Seus moradores foram colocados em conjuntos habitacionais construídos na periferia, muito distantes do mercado de trabalho. A população por diversos motivos, não se adaptou, era preciso pagar um aluguel ou financiamento; o preço do transporte era muito dispendioso e o tempo de viagem muito longo; os apartamentos eram muito rígidos, impessoais e sem área aberta. Assim, muitos voltaram a viver nas favelas que não foram afetadas pela ação do governo ou então criaram novas [15].

No princípio dos anos 80, com o processo de abertura política e o fim da ditadura militar, as políticas oficiais de habitação passaram a ser menos autoritárias. Com a redemocratização do país as propostas de urbanização que os favelados reivindicavam, desde muito foram incorporadas às políticas federais. Seus moradores passaram a ser considerados trabalhadores, e não só marginais, e a favela começou a ser percebida como o lamentável

resultado de um desenvolvimento industrial e um crescimento urbano rápido e sem planejamento. Neste contexto surgiram as políticas de “legalização”.

A disputa pelos raros espaços disponíveis faz surgirem barracos em terrenos insólitos. Assim, vêm sendo ocupadas faixas estreitas de terra não edificantes situadas ao longo de vias públicas, rios e canais, e sob viadutos e pistas elevadas. A imagem de longas seqüências de casebres embaixo de viadutos se tornou comum na cidade: estimava-se em 1990, que existiam cerca de 30 destas ocupações, que poderiam ser denominadas de favelas lineares [15].

Os números vêm confirmar o que é já percebido a olho nu. Nos últimos dez anos, pelo menos uma nova favela com mais de 50 casas surgia a cada mês no município do Rio. O censo de 2000, realizado pelo Instituto Brasileiro de Geografia e Estatística (IBGE), revela que a cidade ganhou 119 favelas a partir de 1991 e listou 513 comunidades faveladas na capital no ano de 2000 - um crescimento de 30,2% em relação ao censo anterior, feito em 1991, e de 12,3% levando-se em consideração a recontagem de 1996. No *ranking* nacional, o Rio ficou em segundo lugar, só perdendo para a cidade de São Paulo em quantidade de áreas carentes. Embora menor do que na capital, o número de favelas no Estado do Rio passou de 661, em 1991, para 811, em 2000 alcançando um aumento percentual de 22,7%.

Esses percentuais são altos, se pensarmos que o espaço urbano não cresceu - diz a professora da Universidade do Estado do Rio de Janeiro (UERJ) Rosângela de Azevedo Gomes, que faz pesquisas com favelas [17].

Com o plano diretor da cidade do Rio de Janeiro de 1991, as favelas passaram a ser zonas caracterizadas como áreas especiais de interesse social (AEIS), sendo destinadas a programas específicos de urbanização e regularização fundiária.

CONCENTRAÇÃO: De acordo com o Censo 2000, Jacarepaguá é a região que concentra o maior número de favelas (68), seguida de Bangu (21) e Realengo (14). Depois vêm Itanhangá, Anchieta, Recreio dos Bandeirantes e Complexo do Alemão, com 11 favelas cada um. Rio Comprido e Cordovil têm dez comunidades carentes.

BRASIL: No *ranking* nacional, ano 2001, a cidade de São Paulo ocupa o primeiro lugar em concentração de favelas.

Quadro 2.3 – Número de favelas por cidade

Cidades	Número de Favelas
São Paulo	612
Rio de Janeiro	513
Fortaleza	157
Guarulhos	136
Curitiba	122
Campinas	117

Belo Horizonte	101
Osasco	101
Salvador	99
Belém	93

População Brasileira e Favelada

O número de brasileiros que vivem em grandes favelas das capitais do país cresceu 39,3% na década de 1990, o que ajudou a elevar o número de indivíduos que moram neste tipo de habitação, 6,5 milhões em todo o país. Os números são de pesquisa do IBGE, divulgada no dia 12 de fevereiro de 2006 pelo jornal *O Estado de São Paulo*.

De acordo com o estudo, mais da metade da população dos “aglomerados subnormais” está nos Estados de São Paulo (2,07 milhões) e do Rio (1,38 milhão). Abaixo temos o quadro da população pobre brasileira.

Quadro 2.4 – População pobre - Brasil¹

Regiões	Número de Pobres	
	2003	2004
Norte	4.147.801	3.855.427
Nordeste	25.220.497	24.000.690
Sudeste	22.254.249	21.630.193
Sul	4.205.445	3.906.226
Centro-Oeste	4.535.114	4.305.659
Brasil	60.363.106	57.698.195
Metropolitano	21.128.931	20.950.811
Urbano	29.462.147	27.910.810
Rural	9.772.028	8.836.574

Os dados censitários de 1950 a 1991 mostram que a taxa de crescimento anual da população favelada do Rio de Janeiro começou a decrescer na década de 60, sofrendo uma queda brusca nos anos 70. Tal queda se deve a três fatores (**Quadro 2.5**). Em primeiro lugar, a diminuição do ritmo de crescimento populacional não foi relativa apenas aos residentes em favelas, mas a toda a população carioca. No período 1950-1960 a população cresceu cerca de 3% ao ano, e os favelados, 7% ao ano. Na década de 70 esses percentuais caíram para 1,8% e 2,5%, respectivamente. Nesse período o movimento migratório em direção à metrópole do Rio de Janeiro começava a perder o ímpeto verificado nos anos 40 e 50, e a então capital, principal área de atração desses fluxos, sofreu os impactos dessa mudança. Nota-se, entretanto, que a proporção de favelados em relação ao total da população continuou aumentando, até mesmo no período 1970-1980, quando a taxa de crescimento dos primeiros alcançou seu menor valor (**Quadro 2.5**). Em outras palavras, o ritmo de crescimento da população favelada se manteve bem acima da progressão dos demais moradores [18].

¹ Fonte: IBGE / PNAD, 2003 e 2004 - Tabulações Especiais

Quadro 2.5 – População favelada do município do Rio de Janeiro²

ANOS	POPULAÇÃO DO MUNICÍPIO DO RJ	POPULAÇÃO NORMAL	POPULAÇÃO FAVELADA
1950	2.375.280	2.205.975	169.305
1960	3.300.431	2.965.368	335.063
1970	4.251.918	3.686.783	565.135
1980	5.090.723	4.368.299	722.424
1990	5.480.768	4.598.285	882.483
2000	5.851.914	4.759.131	1.092.783

Classicamente, atribuiu-se à migração, particularmente à do Nordeste, a causa do crescimento demográfico nas favelas. Os dados censitários, no entanto, indicam que atualmente a migração explica cada vez menos o acelerado processo de favelização em curso na cidade. Na zona oeste, por exemplo, região que apresentou, entre 1991 e 1996, o maior incremento absoluto de população favelada (em torno de 22 mil pessoas), apenas 2 mil e 600 pessoas residentes em favelas eram migrantes da década de 90. Podemos inferir que o surgimento e a expansão de novas favelas (localizadas predominantemente na zona oeste) têm ocorrido por meio da mobilidade espacial no interior do próprio município, seja do bairro para a favela, seja de favelas consolidadas para favelas recentes. A fuga do aluguel, tanto no mercado formal quanto informal, e a redução da oferta de habitação ou lote popular explicam essa mobilidade em direção às favelas periféricas. Vale lembrar, que o valor do aluguel dos imóveis nas favelas da zona sul e zona norte equivalem, atualmente, ao de um apartamento em Copacabana ou no Centro.

A localização dessas favelas em áreas com grande demanda por trabalho de baixa qualificação nos setores de serviços e construção civil se mantêm como fator central para essa inserção do migrante.

Resumindo, dois em cada três desses brasileiros (65%) moram nas capitais, onde o número de favelados cresceu 39,3% em uma década, passando de 3 milhões para 4,2 milhões.

Total população brasileira em 2006 **186.770.562**, total população favelada **6.520.000**.

Percentual da população brasileira favelada **3,5 %**. Total população Carioca **5.851.914**, total população favelada RJ **1.092.783** . Percentual da população carioca favelada **18,7%** [17,19,20,21,22,2,23,18].

Quadro 2.6 – Comparativo população favelada Brasil versus município RJ

Local	População	População Favelada	Percentual
Brasil	186.770.562	6.520.000	3,5 %
Rio de Janeiro	5.851.914	1.092.783	18,7%

2.4 O Impacto das Favelas nas Cidades

“As favelas não constituem puramente impiedoso crime contra a estética, elas são particularmente uma grave e permanente ameaça à tranquilidade e à salubridade pública. Erigidas contra todos os preceitos da higiene: sem água, sem esgotos, sem a mínima parcela de limpeza, sem remoção de lixo; são como largas sentinas cobertas de dejetos e dos demais resíduos da existência humana, amontoados de imundícies e podridões repastando nuvens de moscas, infiltrando nos quarteirões da cidade toda a sorte de moléstia e de impurezas. Desprovidas de qualquer espécie de policiamento, construídas livremente de latas e frangalhos em terrenos do Patrimônio Nacional, libertadas de todos os impostos, alheias a toda ação fiscal: são excelente estímulo à indolência, atraente chamariz de vagabundos, reduto de capoeiras, vallacoitos de larápios que levam a insegurança e a intranquilidade aos quatro cantos da cidade pela multiplicação dos assaltos e dos furtos” (Pimenta, 1926) [18].

As palavras acima mostram o impacto das favelas nas cidades desde início do século XX. Os problemas gerados pelas favelas são grandes e precisam ser combatidos, a meta da ONU de retirar 100 milhões da faixa de pobreza, quando temos 1 bilhão de pobres em todo mundo é tímida, mas precisa ser realizada.

As favelas já fazem parte da vida das cidades. O grande desafio a ser enfrentado é a redução do número de pessoas vivendo neste tipo de habitação e a não proliferação de outras. Ações do tipo “Favela-Bairro” melhoram a qualidade de vida da população das favelas mas não são suficientes. Os custos envolvidos na recuperação dessas áreas são necessários, mas ainda muito altos. Podendo-se prever, através de modelos, onde essas disseminações podem ocorrer, a tomada de decisão quanto às políticas de contenção desse crescimento e de direcionamento de recursos poderá ser mais equilibrada.

2.5 Diagnóstico da Área Escolhida

O estudo envolve na região da Usina as favelas: Borel, Indiana e o supermercado Carrefour. O morro do Borel começou a ser ocupado em 1922. Em mais de setenta anos de ocupação, a favela se transformou em uma das maiores da cidade.

A ocupação encontra-se consolidada, existindo uma divisão entre Chácara do Céu e o Borel. A área estudada situa-se no Maciço da Tijuca, vale do Rio Maracanã, próxima à floresta da Tijuca, na Zona Norte do município do Rio de Janeiro. A variação da sua população é mostrada na quadro abaixo.

Quadro 2.7 – População do Borel³

ANOS	POPULAÇÃO	DOMICÍLIO
------	-----------	-----------

³ Fonte: IBGE

1980	8551	1861
1991	7121	1764
1996	5658	1508
2000	6831	1844

A Tijuca atualmente é um bairro autônomo e de grande absorção de mão de obra, devido ao comércio, às indústrias e a sua população de classe média. O Bairro pode ser considerado como um subcentro econômico da cidade.

A favela do Borel é delimitada pela Rua São Miguel entre os números 400 e 572 e limitada pelo Rio Maracanã a jusante. A área se apresenta como um divisor de águas da bacia com declividades entre 30 e 70 graus atingindo a cota de 250 metros.

Existe uma outra comunidade chamada Indiana situada às margens do rio Maracanã cuja ocupação começou a partir de 1960, no local onde havia algumas ruínas de uma antiga fábrica de papel pertencente à Seda Moderna.

Com a ocupação total da antiga fábrica, consolidou-se uma nova comunidade às margens do rio Maracanã denominada Indiana II, por ser próxima a Indiana. Hoje a comunidade se encontra bastante adensada em toda sua extensão, não havendo possibilidades de acesso de veículos por causa dos becos estreitos. A população é de 1108 habitantes com 277 domicílios. [24]

Uma grande área, próxima, atualmente sujeita à ocupação é o supermercado Carrefour, que se encontra desativado e apresenta as características necessárias ao surgimento de uma nova favela, tais como: i) o espaço vazio debaixo de viadutos, às margens de rios, terrenos e edificações (públicos ou particulares) abandonados; ii) a existência de oportunidades de emprego próximas; iii) a proximidade de transporte público, ruas, ou estradas; iv) facilidades para fazer compras, e acesso a outros serviços (escolas, bancos etc.); v) atrativos do local escolhido; vi) vizinhança de uma favela já existente.

O modelo desenvolvido simula uma ocupação, como se a área estivesse abandonada, mostrando que ela precisa de uma ação das autoridades para que isso não ocorra.

CAPÍTULO 3 – ALGUNS ASPECTOS DA TEORIA DE SISTEMAS MULTIAGENTES

3.1 Introdução

A área de Sistemas multiagentes (SMA) se interessa pelo estudo de agentes *autônomos* em um universo Multiagentes [25]. O termo *autônomo* designa aqui o fato de que os agentes têm uma existência própria, independente da existência de outros agentes. O objetivo da área é estudar modelos genéricos a partir dos quais se possa conceber agentes, organizações e interações, de modo a instanciar tais conceitos, quando se deseja resolver um problema particular. Dito de um outro modo, o objetivo é conceber os meios, a partir dos quais se possa assegurar que agentes cooperem efetivamente com o intuito de resolver um problema específico quando este for apresentado ao sistema. [26].

Agentes são sistemas computacionais com duas importantes capacidades; a primeira é que são capazes de ações autônomas de decidir por si próprio o que precisa ser feito para satisfazer o objetivo a ser atingido. A segunda é a interação entre agentes com trocas de dados e ações sociais inerentes ao dia-a-dia.

Um agente pode determinar por si próprio o que precisa ser feito para satisfazer os objetivos planejados, não sendo necessário instruí-lo, passo a passo, durante todo o tempo.

Para poder interagir com sucesso, os agentes irão requerer habilidade de cooperação, coordenação e negociação um com os outros.

Pode-se notar que o campo de sistemas multiagentes é altamente interdisciplinar e abrange diversas áreas como economia, filosofia, ecologia, geomática e ciências sociais. Por isso admite múltiplas visões.

Um sistema multiagentes é um conjunto de agentes interagindo em um ambiente comum, onde um agente é uma entidade vivendo neste ambiente e capaz de modificar o ambiente (decisão, ação), e a si próprio (percepção, raciocínio, aprendizado) e de comunicar-se com outros agentes [9].

Uma suposição tácita é que agentes têm um função ativa e que afetam o seu ambiente mais do que seu ambiente os afeta. Dois termos usados para descrever características dos agentes são autonomia e racionalidade. Autonomia se refere ao fato de que o agente age sem intervenção direta do ser humano e racionalidade significa que o agente, atuando de forma racional, isto é, nunca contraria seus próprios interesses e consegue maximizar seu desempenho [27].

3.2 Agentes Inteligentes

Não existe um consenso quanto ao termo agente. Vários autores diferem em suas conceituações, cada um dando um grau de importância diferente a determinados aspectos. Essencialmente, existe um consenso de que autonomia é um ponto central, quanto ao mais, existe pouca concordância.

Um agente é um sistema computacional situado em um meio-ambiente e que é capaz de ações autônomas neste meio-ambiente para poder alcançar os objetivos desejados [28].

Dado um determinado sistema, denomina-se agente cada uma de suas entidades ativas. O conjunto de agentes forma uma sociedade. As entidades passivas são designadas pelo termo ambiente. Um agente raciocina sobre o ambiente, sobre os outros agentes e decide racionalmente quais objetivos deve perseguir, quais ações devem tomar, etc. Um agente é uma entidade real ou virtual; está em um ambiente; percebe o ambiente; é capaz de agir no ambiente; pode se comunicar com outros agentes; e tem um comportamento autônomo [26].

Normalmente os agentes possuem capacidades perceptivas que permitem sentir o seu ambiente antes de executar uma ação ou mais ações selecionadas de um repertório de ações disponíveis, e que são escolhidas em função desta sua percepção do meio. Este conjunto de ações representa a capacidade efetiva dos agentes: sua habilidade para modificar seu ambiente [28].

Um agente é tudo que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores [29] (ver figura 3.1). Um agente humano tem olhos, ouvidos e outros órgãos como sensores, e tem mãos, perna, boca e outras partes que servem como atuadores. Um agente robótico poderia ter câmeras e detectores de faixa infravermelho funcionando como sensores e vários motores como atuadores [29].

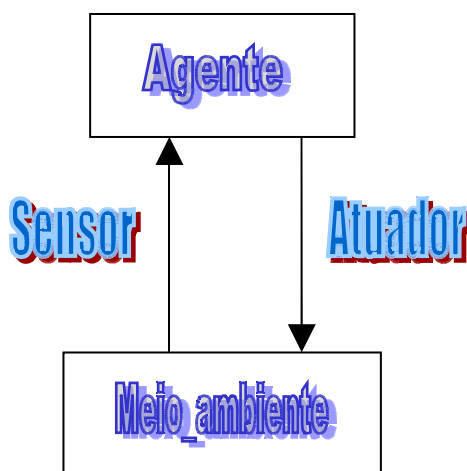


Figura 3.1 – Um agente interagindo com o meio-ambiente [28].

Um sistema de controle simples, envolvendo um termostato pode ser visto como um agente. O termostato é colocado em uma sala (meio-ambiente), produz uma saída, avisando que determinada temperatura foi atingida e com isso induz a ação de se o ligar o ar-condicionado.

Todo agente deve estar contido em um ambiente. Os números de ambientes são vastos. Entretanto, pode-se identificar um número bastante reduzido de dimensões. Em grande parte, essas dimensões determinam o projeto apropriado de agentes e a aplicabilidade de cada uma das principais famílias de técnicas de implementação. O ambiente de simulação é o espaço para movimentação e interação dos agentes, este é designado como o ambiente de tarefa.

As propriedades dos ambientes de tarefa [29] são:

Completamente observável versus parcialmente observável

Quando os sensores de um agente permitem acesso ao estado completo do ambiente em cada instante. Diz-se que o ambiente de tarefa é completamente observável. Um ambiente é completamente observável se os sensores detectam todos os aspectos do meio-ambiente, caso contrário este é parcialmente observável.

Determinístico versus estocástico

Se o próximo estado do ambiente é completamente determinado pelo estado atual e pela ação executada pelo agente, diz-se que o ambiente é determinístico; caso contrário, ele é estocástico.

Episódico versus seqüencial

Em um ambiente de tarefas episódicas, a experiência do agente é dividida em episódios atômicos. Cada episódio consiste na percepção do agente, e depois na execução de uma única ação. Repare que a próxima ação é independente da ação anterior.

Estático versus dinâmico

Se o ambiente puder se alterar enquanto um agente está deliberando, diz-se que o ambiente é dinâmico para esse agente; caso contrário, ele é estático. Ambientes estáticos são fáceis de manipular, porque o agente não precisa continuar a observar o mundo enquanto está decidindo sobre a realização do que fazer.

Discreto versus contínuo

A distinção entre discreto e contínuo pode se aplicar ao estado do ambiente, ao modo como o tempo é tratado, e ainda às percepções e ações dos agentes.

Agente único versus multiagentes

Ao simularmos o mundo de um aspirador de pó atuando na limpeza de uma sala tem-se claramente um ambiente envolvendo um único agente, enquanto o controle de sobrevivência entre lobos e ovelhas em uma fazenda é claramente um ambiente compreendendo multiagentes.

As capacidades que um agente deve possuir, a fim de que seja considerado um agente inteligente incluem [27]:

Reatividade. Agentes inteligentes são capazes de perceber o seu meio-ambiente e responder em tempo hábil às mudanças observadas, a fim de alcançar os objetivos para os quais foram projetados.

Pró-Ativos. Agentes inteligentes são capazes de tomar iniciativas, de modo a exibir comportamento direcionado aos objetivos a fim de satisfazer os objetivos para os quais foram projetados.

Habilidade Social. Agentes inteligentes são capazes de interagir com outros agentes para poder satisfazer os objetivos para os quais foram projetados.

Agentes são construídos para a realização de uma tarefa específica. Para que o agente possa executar esta tarefa é necessário que informemos ao agente, de algum modo, qual a tarefa a ser realizada. Uma forma de especificar a tarefa é codificar um programa, que será, então, executado pelo agente. A vantagem óbvia desta abordagem é que não haverá incerteza alguma sobre o que o agente irá fazer. Em contrapartida, deveremos dizer exatamente passo a passo o que o agente deverá fazer. Todavia na ocorrência de uma circunstância imprevista, o agente será incapaz de executar a tarefa. Assim, o mais comum é dizer ao agente o que fazer, sem explicar o como fazer [27].

Uma arquitetura de *software* constitui a principal parte do projeto de uma aplicação. Representa como as partes que compõem o sistema interagem, onde ocorrem essas interações, e quais são as principais propriedades destas partes. Através deste mecanismo obtém-se uma visão do funcionamento do sistema, incluindo aspectos como a performance, confiabilidade, portabilidade, escalabilidade e interoperabilidade.

Arquitetura Abstrata para Agentes Inteligentes [28]

É fácil formalizar a visão abstrata de agentes. Assuma que o ambiente pode estar em um conjunto finito E de estados discretos:

$$E = \{e, e, \dots\}.$$

Note que não importa se o ambiente é discreto ou não, ambientes contínuos podem ser modelados para um ambiente discreto para qualquer grau de precisão.

Presume-se que agentes possuam um repertório de possíveis ações disponíveis para eles, as quais transformam o estado do ambiente.

$$Ac = \{\alpha, \alpha', \dots\}$$

O modelo básico de agentes interagindo com seus ambientes é o seguinte: O ambiente começa em algum estado, e o agente então escolhe uma ação para executar, baseado naquele estado. Como resultado dessa ação, o ambiente pode responder com um número finito de estados. Entretanto, somente um estado realmente resultará e o agente não conhece antecipadamente qual será. Baseado no segundo estado, o agente escolhe de novo uma ação a executar, agora baseada no novo estado. Como consequência o ambiente responde com um dos possíveis estados; o agente então escolhe outra ação e assim sucessivamente.

A execução de uma agente em um ambiente é uma seqüência intercalada entre os estados do ambiente e ações:

$$r: e_0 \xrightarrow{\alpha_0} e_1 \xrightarrow{\alpha_1} e_2 \xrightarrow{\alpha_2} e_3 \xrightarrow{\alpha_3} \dots \xrightarrow{\alpha_{u-1}} e_u$$

R é o grupo de todas as possíveis seqüências finitas (sobre E e Ac);

R^{Ac} é um subgrupo de R que termina com uma ação; e

R^E é um subgrupo de R que termina com um estado do ambiente.

r, r', \dots são membros de R

Para poder representar os efeitos das ações dos agentes sobre um ambiente. Introdúz-se a função transformadora de estado [30];

$$\gamma: R^{Ac} \rightarrow \rho(E)$$

Assim, a função transformadora mapeia a execução de um agente para um grupo de possíveis estados do ambiente.

Dois pontos importantes a se notar. O primeiro, o ambiente é dependente do histórico, isto é, o próximo estado do ambiente não é somente determinado pela ação realizada pelo agente e o estado corrente do ambiente; as ações realizadas anteriormente também contribuem para determinar o estado corrente. Segundo, essa definição se aplica em um ambiente não-determinístico. Existe, assim, a incerteza sobre o resultado de uma ação em algum estado.

Se $\gamma(r) = \emptyset$ (onde, r é assumido terminar com uma ação), logo, não há possíveis estados sucessores para r . Neste caso, diz-se que o sistema terminou sua execução.

Um ambiente Env pode ser representado pela tripla $Env = (E, e_0, \gamma)$, onde E é um conjunto de estados do ambiente, e_0 pertence a E , e é um estado inicial e γ é uma função transformadora.

Precisa-se introduzir um modelo de agente que habita sistemas. Nós modelamos agentes como funções na qual mapeiam execuções em ações [31]

$$Ag: R^E \rightarrow Ac$$

Assim, uma agente decide qual ação será realizada, baseado no histórico do sistema [28].

3.3 Tipos de Agentes

Agente reativo simples é o tipo mais simples de agente. Estes agentes selecionam ações com base na percepção atual, ignorando o restante do histórico das percepções. Um agente que se baseie apenas em sua posição e consiga reconhecer o que existe em sua posição atual e efetua uma ação, é um exemplo de agente reativo simples. Formalmente, o comportamento do agente reativo pode ser representado pela função:

$$Ag: E \rightarrow Ac$$

Nosso agente de termostato é um exemplo de um agente reativo [28]. Assume, sem perda de generalidade, que o ambiente do termostato pode estar em um dos dois estados - muito frio ou temperatura OK. Então a função de ação do termostato simplesmente é:

Ag(e) = Aquecedor desligado
Aquecedor ligado

se e = temperatura OK,
senão

Agente cognitivo é um tipo de agente mais sofisticado que pode ser dividido em quatro componentes: i) **o elemento aprendido**, responsável pela execução de aperfeiçoamentos; o ii) **o elemento desempenho**, responsável pela seleção das ações externas; iii) **o crítico** informa ao elemento aprendido como o agente está se comportando em relação a um padrão desejado de desempenho; iv) o **gerador de novas idéias**, responsável por sugerir ações que levarão às experiências novas e informativas. [29]

Agentes cognitivos mantêm uma representação *explicita* de seu ambiente e dos outros agentes da sociedade, podendo manter um *histórico* das interações e ações passadas, isto é, têm memória do passado. A comunicação com outros agentes é feita de modo *direto*, através do envio e recebimento de mensagens. Seu mecanismo de controle é *deliberativo*, ou seja, tais agentes raciocinam e decidem sobre quais objetivos devem alcançar, que planos seguir e quais ações devem ser executadas num determinado momento. Seu modelo de organização é baseado em modelos *sociológicos*, como o das organizações humanas.

Outros Tipos de Agentes

- Móveis, são capazes de movimentar-se entre diferentes computadores em uma rede interna (intranet) ou externa (internet);
- Estacionários, são fixos em um mesmo ambiente. Eles são o oposto dos Móveis;
- Competitivos, que competem entre si a fim de realizar a sua tarefa;
- Colaborativos, que colaboram com outros agentes, um complementando a tarefa do outro.

Em uma teoria que explique como um agente inteligente deve se comportar, a idéia de agentes como provadores de teoremas é sedutora. Para ver como esta idéia poderia funcionar, desenvolve-se um modelo simples de lógica baseado em agentes que nós chamaremos de agentes deliberativos. Por exemplo, o banco de dados do agente poderia conter fórmulas como:

Abra (valvula221)

Temperatura (reator4567, 321)

Pressão (tanque776, 28)

Não é difícil ver como fórmulas como estas podem ser usadas para representar as propriedades de um ambiente. O banco de dados é a informação que o agente tem sobre seu ambiente, é constituído por fórmulas de lógica de 1ª ordem e cálculo de predicados. Veja o programa na figura 3.2 abaixo;

```
1.  function action( $\Delta : D$ ) : A
2.  begin
3.      for each  $a \in A$  do
4.          if  $\Delta \vdash_P Do(a)$  then
5.              return  $a$ 
6.          end-if
7.      end-for
8.      for each  $a \in A$  do
9.          if  $\Delta \not\vdash_P \neg Do(a)$  then
10.              return  $a$ 
11.          end-if
12.      end-for
13.      return null
14. end function action
```

Figura 3.2 – Exemplo do programa

Assim, na primeira parte do programa (linhas 3–7), o agente executa cada de suas possíveis ações, e tenta provar a fórmula. Se o agente tem sucesso em provar $Do(a)$, então a retorna como uma ação a ser executada.

Deste modo, o comportamento do agente é determinado pela regras de dedução P do agente (seu “programa”) e seu banco de dados atual Δ (representando a informação que o agente tem sobre seu ambiente).

“Means-end reasoning” é o processo de decidir como alcançar um fim (i.e. uma intenção que você tem) usando os meios disponíveis (i.e. as ações que você quer executar). É conhecido na comunidade de inteligência artificial como planejamento. Planejamento é essencialmente programação automática. As entradas do sistema são:

- 1) A meta, intenção ou uma tarefa. É alguma coisa que o agente deseja alcançar.

- 2) O estado atual do ambiente
- 3) As ações disponíveis para o agente

Como *output* um algoritmo de planejamento gera um plano. Espera-se que, ao ser executado o plano, a partir do estado em que o mundo é descrito, o objetivo, intenção ou tarefa será alcançado.

As arquiteturas reativas (vide Figura 3.3) não incluem nenhum tipo de modelo central do mundo e não utilizam raciocínio simbólico. Elas baseiam-se na formulação de que um agente pode desenvolver inteligência a partir de interações com seu ambiente, não necessitando de um modelo predefinido [32].

Os agentes reativos têm a admirável propriedade de serem simples, mas se caracterizam por ter inteligência muito limitada. Entretanto [29], Brooks argumenta que comportamentos inteligentes podem ser gerados sem representações explícitas ou raciocínio abstrato, mas como uma propriedade emergente de certos sistemas complexos.

O comportamento do agente é descrito pela função de agente (descrição matemática abstrata), que mapeia qualquer seqüência de percepções para uma ação, e implementado pelo programa de agente, que está intimamente relacionado com a arquitetura do agente [29]. Na prática o programa de agente toma a percepção atual como entrada e a função de agente recebe o histórico de percepções completo. Deve-se construir previamente uma Tabela, nomeada regras de condição, que contenha a ação apropriada para todas as seqüências e percepções possíveis.

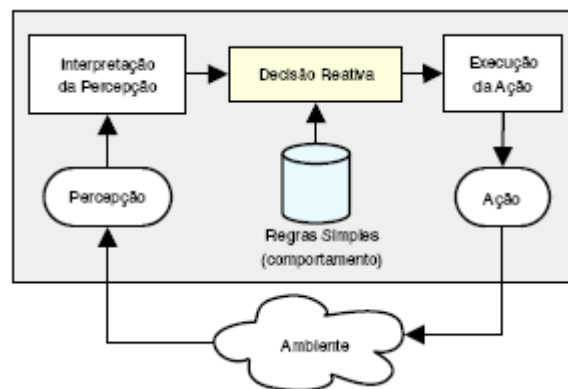


Figura 3.3 – Esquema genérico de uma arquitetura reativa [32]

Há duas características definindo a arquitetura de “*subsumption*”. A primeira é que a decisão de criação de um agente é compreendida através de um conjunto de tarefas comportamentais, cada comportamento pode ser como uma função de ação individual. Cada um desses comportamentos

pretende realizar uma tarefa em particular. A implementação de Rodney Brooks assume não incluir representações simbólicas complexas, ou quase nenhuma.

Em muitas implementações, esses comportamentos são implementados como regras da forma:

Situação → Ação

A segunda característica é que muitos comportamentos podem ser disparados simultaneamente. A simplicidade computacional é a grande força desta arquitetura [28].

3.4 Simulação Envolvendo Agentes de Software

A teoria dos sistemas complexos é um novo campo da ciência que estuda até que ponto as partes de um sistema se separem do comportamento coletivo, e como o sistema interage com seu ambiente. Sistemas complexos são difíceis de entender porque as causas e os efeitos não estão relacionados de forma direta, ou seja, existem correlações entre eventos distantes no tempo e no espaço.

A idéia é que é possível simular sistemas complexos tendo como ponto de partida elementos simples (agentes reativos) que atuam em um meio-ambiente. Estes agentes podem interagir entre si e com o meio-ambiente de modo simples. Como resultado da interação entre os agentes e o meio-ambiente o comportamento do sistema complexo (do macrocosmo) pode ser explorado e analisado.

Na simulação de sistemas complexos a partir de agentes reativos simples podem-se obter resultados formais sobre a dinâmica global do sistema [9].

Pela simulação baseada em agentes é possível modelar situações nas quais os indivíduos têm comportamentos diferentes e complexos, podendo levar em conta tanto as propriedades quantitativas (parâmetros numéricos) quanto qualitativas (comportamento individual) do sistema representado no modelo.

A simulação no computador é, então, usada para descobrir propriedades do modelo e assim ganhar entendimento dentro de um processo dinâmico, o que seria muito difícil de modelar com técnicas matemáticas padrões. [33]

Sistemas de simulação multiagentes provêm de uma plataforma computacional onde a dinâmica dos sistemas espaços-temporais podem ser estudadas. Para tanto pode se usar agentes reativos (mais simples) ou cognitivos (mais elaborados). De acordo com Ferrand Nils é possível simular sistemas complexos, que têm como ponto de partida somente os agentes reativos elementares inseridos em algum ambiente. Quando individualmente considerados, estes agentes reativos têm comportamento extremamente simples, e só interagem com outros agentes situados entre vizinhança próxima. Pela interação com o ambiente e entre eles mesmos, como resultado emerge o comportamento do sistema complexo.

As hipóteses formuladas no modelo são validadas através da comparação da estrutura espacial simulada com os dados observados. O objetivo é reproduzir estruturas plausíveis e dinâmicas ao invés de tentar obter uma reprodução exata da realidade específica.

Um modelo é sempre usado para eliminar as estruturas mais gerais de uma organização espacial com o objetivo de encontrar suas especificidades. Os resíduos do modelo são usados para medir essas especificidades.

Um modelo quando bem calibrado, pode ser usado tanto para fazer previsões como montar cenários para examinar as alterações no sistema espacial [10].

3.5 Escolha do Tipo de Agente

Há necessidade de agentes complexos para realizar uma tarefa complexa?

Os defensores dos sistemas multiagentes reativos dizem que não. O exemplo clássico é uma colônia de formigas. Cada formiga isoladamente é uma entidade bem simples; não se atribui inteligência a uma formiga. Entretanto, o trabalho realizado por uma colônia de formigas é bem complexo: procura de alimento, transporte do alimento até o formigueiro, cuidado com os ovos e larvas, defesa da colônia, etc. E elas têm tido êxito, adaptando-se às mudanças de condições, como os períodos glaciais, e sobrevivendo há muitos milhões de anos.

Através do exemplo das formigas, podem-se destacar as principais características dos agentes em sistemas multiagentes reativos:

- não há representação explícita de conhecimento - o conhecimento dos agentes é implícito e se manifesta através do seu comportamento;
- não há representação do ambiente - o seu comportamento se baseia no que é percebido a cada instante do ambiente, mas sem uma representação explícita deste;
- não há memória das ações - os agentes reativos não mantêm um histórico de suas ações, de forma que o resultado de uma ação passada não exerce nenhuma influência sobre as suas ações futuras;
- a forma de organização dos agentes reativos é similar a dos animais, em oposição à organização social dos sistemas cognitivos;
- grande número de membros - os sistemas multiagentes reativos têm, em geral, um grande número de agentes, da ordem de dezenas, centenas ou mesmo milhões de agentes.

Portanto, os agentes reativos são muito simples e não possuem representação de seu ambiente. Desta forma, todas as informações relativas ao seu comportamento estão no ambiente e suas reações dependem unicamente de sua percepção deste ambiente. Eles não possuem registro de suas ações passadas nem podem antecipar (planejar) o futuro [26].

Neste trabalho é utilizado um sistema de agentes reativos, composto de entidades muito simples para simular o processo espaço-temporal do surgimento das favelas e sua disseminação.

Os agentes reativos só têm a capacidade para captar o estado dos “*patches*” (área do terreno) representados na imagem. Baseados no estado dos “*patches*” (elevação, abandono, proteção ambiental) e algumas outras condições macro como (taxa de migração, taxa de crescimento econômico), é possível efetuar uma ação apropriada como, por exemplo, (expansão ou contração da favela) mudando as características da área em questão.

Devido ao fato que a simulação pretendida é baseada em agentes reativos muito simples, e que este ambiente é representado em uma imagem “*raster*”, então é possível usar ferramentas de simulação tais como: Starlogo ou NetLogo. No próximo capítulo tem-se o estudo da linguagem Logo, que subsidia a escolha da ferramenta a ser utilizada nesta dissertação.

CAPÍTULO 4 - *SOFTWARE* LOGO

4.1 História do LOGO

O LOGO foi criado em 1966 em uma firma de pesquisa de Cambridge a BBN (Bolt Beranek & Newman Inc.), por Wally Feurzeig e Seymour Papert. Suas raízes intelectuais estão

em inteligência artificial, lógica matemática e psicologia. A tartaruga constituía, então uma inovação grande e recente. Os primeiros quatro anos de pesquisa do Logo, desenvolvimento e trabalho pedagógico foram terminados na BBN. A primeira implementação foi escrita em LISP (BALBUCIO) em um PDP-1. Seu nome era “Ghost”.

A preocupação desses pesquisadores concentrava-se principalmente em desenvolver hardware e *software* para implementar o interpretador Logo e em demonstrar o que se podia fazer com ele, tendo em vista principalmente sua relação com a matemática. Trabalhava-se com ele em computadores de médio e grande porte, o que também foi um dos fatores contribuintes para que seu uso de início, ficasse restrito à universidades e centros de pesquisa.

O Logo não é agora muito diferente dos conceitos anteriores básicos da primeira tartaruga. A primeira tartaruga foi um rádio controlado (telegrafia sem fios) de chão nomeado "Irving". Irving teve sensor de toque e podia ir para frente, atrás, direita, esquerda (rotações), e ding (Irving teve um sino) emitia um som [34].

4.2 Implementação do LOGO

Há mais de 130 implementações de LOGO, cada uma das quais tendo suas próprias vantagens e desvantagens. Uma implementação de multiplataforma popular é UCBLLogo. MSWLogo, seu “*freeware*” derivado de Windows, é geralmente usado nas escolas no Reino Unido. LOGO de Comenius está disponível em holandês, alemão, o tcheco etc. LOGO de SOLI era uma versão francesa extensamente usada nas escolas primárias nos anos oitenta. Em 1984, ExpertLogo para o Macintosh levou as tartarugas para 3D e na superfície de esferas. Também foi compilado em vez de interpretado e conseqüentemente era muito mais rápido. ARLOGO vem do UCBLLogo, considerada a primeira linguagem de programação de fonte aberta em árabe.

Um derivado moderno de LOGO é uma variação que permite a milhares de “tartarugas”, moverem-se independentemente. Há duas implementações populares: StarLogo do MIT e o NetLogo da **Universidade Northwestern**. Estas duas linguagens permitem a exploração de fenômenos emergentes e acumulam experiências em estudos sociais, biologia, física, e muitas outras ciências. Embora o foco esteja na interação de um número grande de agentes independentes, estas variações ainda capturam o sabor original de LOGO.

Existem outros ambientes de desenvolvimento de sistemas multiagentes reativos encontrados na literatura. O sistema SIEME – “*Simulateur Evènementiel Multi-Entités*” [35], é um simulador para sistemas multiagentes, desenvolvido por Laurent Magnin, do laboratório Laforia da Universidade de Paris. O objetivo deste sistema é facilitar a descrição de um modelo

multiagente ou mais precisamente o ambiente e as propriedades físicas dos agentes para simulação em computador.

O sistema SIEME é totalmente integrado ao sistema de programação Smalltalk 80, de modo que uma aplicação SIEME pode ser executada sem modificação em várias plataformas (Macintosh, Sun/Unix, PC). O sistema SWARM é uma plataforma de *software* multiagente para a simulação de sistemas adaptivos. Fornece uma biblioteca orientada a objetos de componentes reutilizáveis para a construção de modelos e para analisar experimentos sobre o mesmo. O ambiente SIMULA, brasileiro da CPGCC da UFRGS 1997, para o desenvolvimento de sistemas multiagentes reativos, destina-se a atender usuários que tenham conhecimento da tecnologia do uso de agentes para a construção de sistemas reais, científicos e simulações [36].

4.3 Programação do LOGO

A idéia é que uma “tartaruga” com uma caneta a ela associada pode ser ensinada a fazer coisas simples do tipo, mover 100 espaços à frente ou se virar. Destes blocos de construção básicos podem-se construir formas mais complexas como quadrados, triângulos, círculos. Com estas formas é possível desenhar estruturas complexas como casas ou barcos à vela.

A tartaruga se move com comandos que são relativos à sua própria posição, “ESQUERDA 90” significa gira 90 graus à esquerda. Um estudante poderia entender o movimento da tartaruga imaginando o que elas fariam se eles fossem a tartaruga. Papert a isto denominou ‘corpo sintônico’.

A idéia de gráficos de tartaruga também é útil, por exemplo, em sistemas de Lindenmayer para gerar fractais.

Os comandos, aparentemente muito simples, podem ser dados em grupos de instruções, essencialmente para criar bibliotecas de comandos mais complexos. Na prática, formas curtas são usadas. Por exemplo, "ESQUERDA 90" é escrita para "LT 90". Palavras-chaves normalmente são escritas em maiúscula por novatos, mas textos mais avançados usam minúscula.

Exemplo: Montar quadrado

FORWARD 100

LEFT 90

FORWARD 100

LEFT 90

FORWARD 100

LEFT 90

FORWARD 100

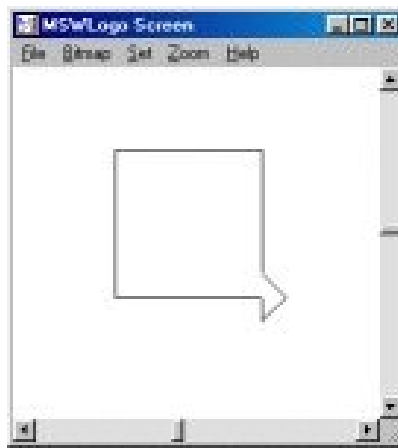


Figura 4.1 – Obtenção do quadrado

4.4 StarLogo

O StarLogo foi criado no MIT com uma idéia de atingir metas diferenciadas. O objetivo direto não era particularmente com desempenho ou velocidade. O interesse é em prover modos novos para usuários modelarem, controlarem, e pensarem em ações que de fato acontecem em paralelo. Muitas coisas no mundo realmente agem em paralelo; o modo mais natural para modelar tais situações é com uma linguagem de programação paralela. Nestes casos, paralelismo não é um “truque” para melhorar desempenho; é o modo mais natural de expressar o comportamento desejado.

O StarLogo é um programa modelador de meio-ambiente para explorar os funcionamentos de sistemas descentralizados, sistemas que são organizados sem um organizador, coordenados sem um coordenador. Com StarLogo, você pode modelar muitos fenômenos da vida real, como bandos de pássaro, engarrafamentos, colônias de formiga, e economias de mercado.

Em sistemas descentralizados, padrões podem surgir sem controle centralizado.

Cada vez mais, pesquisadores estão escolhendo modelos descentralizados para as organizações que eles constroem no mundo e a cada dia surgem novas teorias e tecnologias. A maioria das pessoas continua resistindo a estas idéias, assumindo centralizado onde nenhum existe, por exemplo, assumindo incorretamente que aqueles bandos de pássaro têm os líderes. StarLogo é projetado para ajudar os estudantes, como também os pesquisadores a desenvolver novos modos de pensar e entender sistemas descentralizados. [37]

StarLogo é uma versão especializada da linguagem de programação Logo. Com versões tradicionais de Logo, você pode criar desenhos e animações dando comandos as tartarugas gráficas na tela de computador. StarLogo estende esta idéia permitindo-lhe controlar milhares de tartarugas gráficas em paralelo. Além disso, StarLogo faz o mundo computacional das tartarugas ativo: você pode escrever programas para milhares de diferentes meio-ambientes que

compõem o ambiente das tartarugas. Tartarugas e ambientes podem interagir uns com os outros; por exemplo, você pode programar as tartarugas para "cheirar" ao redor do mundo, e mudar os seus comportamentos baseado no que eles sentem no terreno abaixo de si. StarLogo é particularmente apropriado para uso em projetos de vida artificiais.

O “*software*” Starlogo foi testado por cerca um mês e para poder concretizar o objetivo deste estudo algumas dificuldades surgiram para realizar determinadas tarefas, como por exemplo, utilizar uma aerofoto que é uma imagem “*raster*”. O Starlogo tem limitações com importação de imagens.

4.5 NetLogo

O StarLogo original foi desenvolvido no MIT Mídia Laboratório em 1989/1990 e foi executado em um supercomputador maciçamente paralelo chamado a Máquina de Conexão.

Alguns anos depois (1994), a versão paralela simulada foi desenvolvida para o computador da Apple o Macintosh. Aquela versão se tornou MacStarLogo.

StarLogoT (1997), desenvolvido pelo Centro para Aprendizagem Conectada (CCL) da Universidade NorthWestern, é essencialmente uma versão estendida de MacStarLogo com muitos recursos e capacidades adicionais.

Desde então duas multiplataformas Multiagentes Logo foram desenvolvidas baseadas em Java: NetLogo (da CCL NorthWestern) e uma versão StarLogo (do MIT).

A linguagem do NetLogo difere em muitos aspectos de MIT StarLogo. Ambas as linguagens estavam inspiradas pelo StarLogo original, mas foram redesenhadas de modos diferenciados.

O projeto do NetLogo foi dirigido pela necessidade de revisar e ampliar a linguagem, assim tornou-se mais fácil de usar e mais poderoso, e pela necessidade de apoiar a arquitetura de HubNet.

NetLogo incorpora quase todas as funcionalidade estendidas do StarLogoT, como também um grande número de características mais novas.

NetLogo é um programa modelador ambiente para simular fenômenos naturais e sociais. Foi idealizado particularmente para modelar sistemas complexos que se desenvolvem com o passar do tempo. Pesquisadores podem dar instruções para centenas ou milhares de "agentes" independentes todos operando simultaneamente. Com Isto torna-se possível explorar a conexão entre o comportamento de indivíduos a nível micro e macro. Níveis padrão emergem da interação de muitos indivíduos.

A parte "LOGO" é porque NetLogo é um dialeto do idioma de Logo. LOGO é uma linguagem de programação extremamente poderosa, freqüentemente mal-interpretada como se fosse somente para as crianças. A variante NetLOGO hoje é uma linguagem moderna. Com

NetLogo é possível simular ambientes com agentes interagindo autônomos. Estes sistemas simulados podem ser usados para modelar fenômenos como comportamento de eleitor, sistemas ecológicos, reações químicas, e fenômenos físicos. A força do ambiente que se usa é de acesso fácil à visualização dinâmica, e fácil de aprender.

A parte "Net" significa evocar a descentralização, natureza interconectada de fenômenos que você pode modelar com o NetLogo. Também se refere ao ambiente HubNet, o ambiente de simulação de participação em rede que está incluído no NetLogo.

NetLogo é escrito completamente em Java (versão 1.4.1) sendo interpretado. Ele permite aos estudantes realizarem simulações abertas com eles, enquanto explora o seu comportamento sob várias condições. Também é um ambiente que permite aos estudantes, professores e desenvolvedores a criar os seus próprios modelos. NetLogo é simples bastante para que os estudantes e professores possam executar simulações facilmente ou até mesmo possam construir suas próprias. E, está avançado bastante para servir como uma ferramenta poderosa para os pesquisadores em vários outros campos.

NetLogo também pode dar poder a uma ferramenta de simulação participatória de sala de aula chamada HubNet. Pelo uso de computadores em rede ou dispositivos de mão como as calculadoras Texas (TI 83+), cada estudante pode controlar um agente em uma simulação.

NetLogo é a próxima geração das séries de programas de modelagem multiagentes que começaram com StarLogo. Tem a funcionalidade de produto StarLogoT e inclui significantes recursos novos e uma linguagem e interface do usuário redesenhada. NetLogo é escrito em Java assim pode ser executado em todas as principais plataformas (Mac, Windows, Linux etc) e é executado como uma aplicação independente. Modelos individuais podem ser executados como *applets* de Java dentro de um *web browser*.

Recursos

Multiplataforma : MACOS, WINDOWS, LINUX et al

• Linguagem:

- ◆ Totalmente programável
- ◆ Estrutura de linguagem simples
- ◆ Linguagem é Logo dialética estendida para suportar agentes e concorrência
- ◆ Número ilimitado de agentes e variáveis
- ◆ Muitos comandos interno
- ◆ Inteiro, precisão dupla e ponto flutuante
- ◆ Roda em multiplataforma

• Ambiente:

- ♦ Exibe modelo em 2D e 3D
- ♦ Vetores escaláveis e rotacionais
- ♦ “Tartaruga” e “terreno” (*turtle* and *patch*)
- ♦ Construtor de Interface com *buttons, sliders, switches, choosers, monitors, text boxes*
- ♦ Controla velocidade dos agentes
- ♦ Sistema de plotagem poderoso e flexível
- ♦ Área de informação para anotações do seu modelo
- ♦ HubNet: simulações participatória em rede
- ♦ Monitor de agentes para inspeção e controle de agentes
- ♦ Exporta e importa funções (exporta dados, salva e restaura estado do modelo)
- ♦ Ferramenta *BehaviorSpace* usada para coletar dados de múltiplos modelos executados
- ♦ Modelador de sistemas dinâmicos

A distribuição do “*software*” Netlogo é livre e tem um grupo de discussão bastante ativo no “*site yahoo*”, com 2776 membros em 13 de março de 2006, tendo sido fundado em 23 de janeiro de 2002 e a comunicação é na língua inglesa [38].

Exigências do Sistema

O NetLogo é projetado para rodar em quase todos os tipos de computadores, mas alguns sistemas mais antigos ou menos poderosos não são suportados. As exigências exatas são resumidas abaixo.

Aplicação

Em todos os sistemas, é exigido aproximadamente 25MB de espaço livre no disco rígido:

Windows

- Windows NT, 98, ME, 2000, or XP
- 64 MB RAM

É necessária Java Virtual Machine para executar o NetLogo.

Mac OS X

- OS X versão 10.2.6 ou posterior (10.3 ou posterior é recomendada)

- 128 MB RAM (256 MB RAM é altamente recomendado)

Outras plataformas

NetLogo só irá funcionar em plataformas nas quais Java Virtual Machine, versão 1.4.1 ou posterior estiver disponível.

Applets Salvos

Modelos NetLogo salvos como Java “*applets*” devem funcionar em qualquer *web browser* e plataforma na qual a versão “*Java Virtual Machine*” 1.4.1 ou posterior estiver disponível. A visão 3D não está disponível nos “*applets*”.

Visão 3D

A visão 3D no NetLogo é um recurso novo. Abaixo a relação onde o teste ocorreu:

Sistemas Operacionais

- Linux 2.6.8 (Debian i386)
- Linux 2.6.8 (Debian amd64)
- Mac OS X 10.3.8, 10.4.0 e 10.4.1, sendo exigido o Java 1.4.2.
- Windows 2000
- Windows XP

Placas gráficas

ATI Radeon 7500; ATI Radeon 9200; ATI Radeon 9600; ATI Radeon 9800 XT; ATI Radeon IGP 345; ATI Radeon Mobility; ATI FireGL V3100; Intel 82830M; nVidia GeForce MX; nVidia GeForce FX 5200; nVidia GeForce FX Go5650 ; nVidia Quadro NVS. [39]

4.6 Definição do *Software*

O ambiente de desenvolvimento de sistemas multiagentes pode ocorrer em dois tipos de plataformas. A plataforma de uma linguagem de programação que exige o conhecimento e o uso de uma linguagem de programação tradicional (C, Java, Delphi), que seria realizada passo a passo, em linhas de comando. Os trabalhos “Modelagem Comportamental para Agentes Autônomos em Ambientes Reais” [40] e “Simulação e Controle de Áreas Marítimas” [41] foram desenvolvidos em linguagens tradicionais. Outra plataforma é usar ferramentas de simulação tais como: Starlogo ou NetLogo , com uma estrutura que facilita a criação de agentes e a sua programação.

A escolha de uma ferramenta recaiu sobre uma que já fosse criada e previamente feita sob a visão de linguagem orientada a agentes e que permitisse uma programação mais rápida e ao mesmo tempo oferecesse o recurso de usar uma programação tradicional (JAVA), se a programação existente ainda não for suficiente para as suas necessidades. Uma alternativa é o uso do programa de interface aplicada, o qual permite escrever um código customizado JAVA para o NetLogo.

Devido ao fato que a simulação planejada está baseada em agentes reativos muito simples, e que o ambiente é representado com uma imagem “*raster*”, é possível usar algumas ferramentas de simulação de multiagentes como visto acima.

Após o estudo inicial a escolha direcionou-se para o Netlogo, por todos os fatores mencionados neste capítulo. Para superar a dificuldade de poder importar uma imagem “*raster*” que era extremamente necessária à dissertação, foi necessário recorrer à ajuda do grupo Netlogo [38], que é altamente prestativo, disponibilizando os códigos necessários e trocando informações para serem estudadas e com isso facilitando o aprendizado do NetLogo, permitindo alcançar o objetivo almejado.

Desta forma a definição recaiu sobre o NetLogo, um *software* em contínua evolução e acompanhamento pelo Centro para Aprendizagem Conectada (CCL) da Universidade NorthWestern, disseminando-se rapidamente por todo mundo e no Brasil. Na internet já se encontra alguns usuários no Brasil como o Prof. Crediné Silva de Menezes (Departamento de Informática da Universidade Federal do Espírito Santo) [33].

CAPÍTULO 5 - ESTUDO DA IMAGEM “RASTER”

5.1 Imagem “*raster*”

Imagens “*raster*”, ou “bitmap” (mapa de bits, em português) obtida por “escaneamento” ou por “sensoriamento remoto”, é uma representação digital constituída por uma série de pontos, ou “*pixels*”, agrupados em linhas e colunas compondo a representação visual de um documento. Cada *pixel* pode ser descrito como BINÁRIO (branco ou preto, “*on*” ou “*off*”) ou representar 256 níveis de cinza (0 a 255).

O tratamento de imagens deste tipo requer ferramentas especializadas que envolvem cálculos muito complexos, como interpolação, álgebra matricial, etc. Um exemplo de programa capaz de manipular imagens “*raster*” é o **Photoshop** (Adobe).

Resolução de uma imagem *raster* é o número de pontos capturados em cada polegada (25,4 mm) do documento. Em geral são utilizadas resoluções de 200 DPI (200 pontos por polegada), 400 DPI e 500 DPI. Quanto maior a resolução, maiores serão os níveis de detalhes capturados do original e mais perfeita será a imagem. Para uma resolução de 400 DPI, em cada polegada existem 400 pontos em ambas as direções ou 160.000 pontos (400 x 400) por polegada quadrada; para uma resolução de 200 DPI, somente 40.000 pontos (200 x 200) para a mesma polegada quadrada.

Softwares tais como o Autodesk “*raster*” Design e o ProVec possibilitam, editar, redimensionar, cortar, imprimir, “limpar” um arquivo “*raster*”, criar detalhes e até mesmo produzir um arquivo híbrido CAD-RASTER de um arquivo “*raster*” [42]. Uma fotografia digital é em essência uma imagem “*raster*”.

5.2 Aerofoto

Para viabilizar o estudo foi utilizada uma aerofoto ou fotografia aérea da área referente ao bairro da Usina produzida pela empresa Esteio.

Mapeamento

Entende-se por mapeamento a aplicação do processo cartográfico sobre uma coleção de dados ou informações, com vistas à obtenção de uma representação gráfica da realidade perceptível, comunicada a partir da associação de símbolos e outros recursos gráficos que caracterizam a linguagem cartográfica [43].

Aerofotogrametria

A fotogrametria é a ciência que permite executar medições precisas utilizando fotografias métricas. Embora apresente uma série de aplicações nos mais diferentes campos e ramos da ciência, como na topografia, astronomia, medicina, meteorologia e tantos outros, tem sua maior aplicação no mapeamento topográfico [43].

A fotogrametria tem por finalidade determinar a forma, dimensões e posição dos objetos contidos numa fotografia, através de medidas efetuadas sobre a mesma.

Inicialmente a fotogrametria buscava apenas determinar a posição dos objetos, pelo método das interseções, sem observar ou medir o relevo, muito embora desde 1732 se conhecessem os princípios da estereoscopia, empregada para observar (sem medir), o relevo do solo contido em pares fotográficos analisados estereoscopicamente.

Em 1901, o alemão Pulfrich, apoiando-se em princípios estabelecidos por Stolze, introduziu na Fotogrametria o chamado índice móvel ou marca estereoscópica. Então, não só foi possível observar o relevo, como medir as variações de nível do terreno.

Pulfrich construiu um primeiro aparelho que denominou "estereocomparador", e com ele iniciou os trabalhos dos primeiros levantamentos com base na observação estereoscópica de pares de fotografias utilizados em fotogrametria terrestre.

A partir de então uma série de outros aparelhos foram construídos e novos princípios foram estabelecidos, porém, para tomada de fotografias era necessário que os pontos da estação que referenciavam o terreno continuassem no solo, com todos os seus inconvenientes.

Durante a guerra de 1914 - 1918 tornou-se imperioso um maior aproveitamento da fotogrametria, usando-se, para tomada de fotografias, pontos de estação sempre mais altos, sendo utilizados, para tal, balões, balões cativos e até "papagaios".

Com o advento da aviação desenvolveu câmaras especiais para a fotografia aérea, substituindo quase que inteiramente a fotogrametria terrestre pela aerofotogrametria.

Assim, aerofotogrametria é definida como a ciência da elaboração de cartas mediante fotografias aéreas tomadas com câmara aero-transportadas (eixo ótico posicionado na vertical), utilizando-se aparelhos e métodos estereoscópicos.

Vôo Fotogramétrico

É realizado após um completo planejamento da operação, resultante de um estudo detalhado com todas as especificações sobre o tipo de cobertura a ser executado.

A tomada das fotografias aéreas obedece a um planejamento meticuloso e uma série de medidas é adotada para que se possa realizar um voo de boa qualidade. É necessário consultar o mapa climatológico para conhecimento do mês e dias favoráveis à realização do voo fotogramétrico.

Um projeto de recobrimento é um estudo detalhado, com todas as especificações sobre o tipo de cobertura, por exemplo:

Condições naturais da região

- Local a ser fotografado
- Área a fotografar
- Dimensões da área
- Relevo
- Regime de ventos
- Altitude média do terreno
- Variação de altura do terreno
- Mês para execução do voo
- Número de dias favoráveis ao voo

Apoio logístico

- Transporte
- Hospitais
- Alimentação

Condições técnicas (base e aeronave)

- Base de operação
- Alternativa de pouso
- Recursos na base
- Modelo da aeronave
- Autonomia
- Teto de serviço operacional
- Velocidade média de cruzeiro
- Tripulação

Condições técnicas (plano de voo)

- Altura de voo
- Altitude de voo
- Escala das fotografias
- Superposição longitudinal
- Superposição lateral
- Câmara aérea
- Tipo e quantidade de filme empregado
- Rumo das faixas
- Número de faixas e número de fotos
- Velocidade máxima (arrastamento)
- Tempo de exposição ideal
- Intervalo de exposição
- Distância entre faixas
- Base das fotos

As fotografias aéreas devem ser tomadas sempre com elevação do sol superior a 30°, em dias claros, nos quais as condições climáticas sejam tais que permitam a obtenção de negativos fotográficos claros e bem definidos, isto é, bem contrastados.

Fotograma

É a fotografia obtida através de câmaras especiais, cujas características óticas e geométricas permitem a retratação acurada dos dados do terreno, de forma que os pormenores topográficos e planimétricos possam ser identificados e projetados na carta, bem como forneçam elementos para a medição das relações entre as imagens e suas posições reais, tais como existiam no momento da exposição. O termo é empregado genericamente, tanto para os negativos originais, como para as cópias e diapositivos. Por extensão pode também ser aplicado à tradução fotográfica dos dados obtidos por outros sensores remotos que não a câmara fotográfica. O formato mais usual é o de 23 x 23 cm.

Uma fotografia aérea é um retrato da superfície da terra, em que os acidentes e detalhes geográficos aparecem como são vistos da aeronave.

Fotografia aérea ou imagem vertical: São assim denominadas aquelas cujo eixo principal é perpendicular ao solo. Na prática tal condição não é rigorosamente atingida em

consequência das inclinações da aeronave durante o voo. Esta não deve exceder a 3%, limite geralmente aceito para classificar-se uma fotografia como vertical.

Fotointerpretação

É a técnica de analisar imagens fotográficas com a finalidade de identificar e classificar os elementos naturais e artificiais e determinar o seu significado.

Câmaras Fotogramétricas Aéreas

A câmara aérea desloca-se durante a exposição, necessitando de objetivas adequadas, obturadores de alta velocidade e filmes de emulsão ultra-rápida, reduzindo a um mínimo o tempo de exposição, sem prejudicar a qualidade da imagem.

As câmaras aéreas classificam-se de acordo com o ângulo que abrange a diagonal do formato, ângulo esse que define a cobertura proporcionada pela câmara:

- Ângulo normal: até 75° - Para abranger uma área a uma determinada altura de voo.
- Grande angular: de 75° até 100° - A altura de voo será menor, com menor distância focal (f).
- Supergrande angular: maior que 100° - A altura de voo e a distância focal serão ainda menores.

Também são classificadas pela distância focal da objetiva:

- Curta: até 150 mm
- Normal: de 150 a 300 mm
- Longa: acima de 300 mm

Escala Fotográfica

A escala fotográfica é definida como sendo a relação entre um comprimento de uma linha na fotografia e a sua correspondente no terreno.

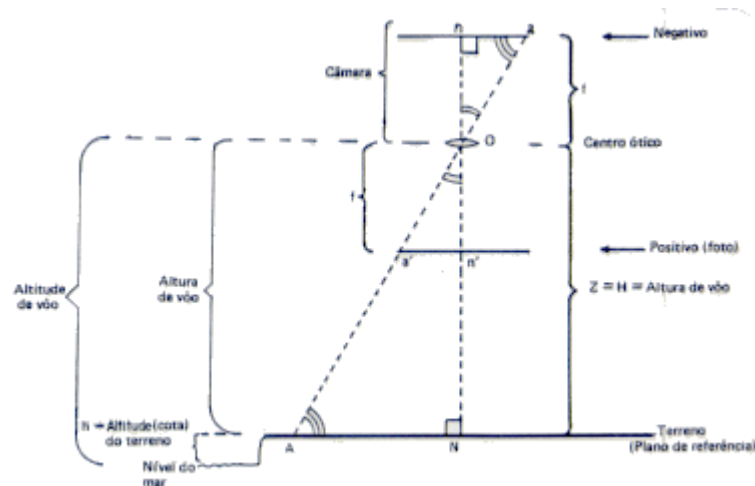


Figura 5.1 – Geometria básica de uma fotografia aérea

Considerando a Figura 5.1, nota-se que os raios de luz refletidos do terreno passam pelo eixo ótico da lente. O eixo ótico e o plano do negativo são perpendiculares, assim como o eixo ótico e o plano do terreno. Desta forma, o ponto principal da fotografia e o ponto Nadir representam o mesmo ponto.

Pode-se afirmar que os triângulos **NOA** e **noa** são semelhantes, assim, pode-se calcular a escala da fotografia usando essa semelhança de triângulos. Existem três elementos: a medida na foto, a medida no terreno e a escala conhecida ou a determinar.

A escala mantém a seguinte relação com os triângulos semelhantes:

$$E = \frac{na}{NA} = \frac{oa}{OA} = \frac{no}{NO}$$

Onde:

AN = distância real

an = distância na fotografia

NO = altura de vôo = H

no = distância focal = f

Assim, a escala da fotografia pode ser determinada conhecendo-se a distância focal e a altura de vôo.

$$E = \frac{no}{NO} = \frac{f}{H}$$

Ou ainda através de uma distância na fotografia entre dois pontos a e b quaisquer e a sua respectiva medida no terreno.

$$E = \frac{ab}{AB}$$

Exemplo: Em um recobrimento aéreo, a uma altura de vôo igual a 6.000 m, utilizando-se uma câmara com distância focal de 100 mm, a escala da fotografia será:

$$E = \frac{f}{H} = \frac{100mm}{6000000mm} = \frac{1}{60000}$$

Cobertura Fotográfica

É a representação do terreno através de fotografias aéreas, as quais são expostas sucessivamente, ao longo de uma direção de vôo. Essa sucessão é feita em intervalo de tempo tal que, entre duas fotografias haja uma superposição longitudinal de cerca de 60%, formando uma faixa. Nas faixas expostas, paralelamente, para compor a cobertura de uma área é mantida uma distância entre os eixos de vôo de forma que haja uma superposição lateral de 30% entre as faixas adjacentes. Alguns pontos do terreno, dentro da zona de recobrimento, são fotografados várias vezes em ambas as faixas (figura 5.2).

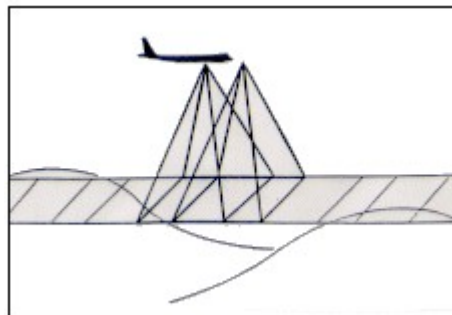


Figura 5.2 – Vôo fotogramétrico

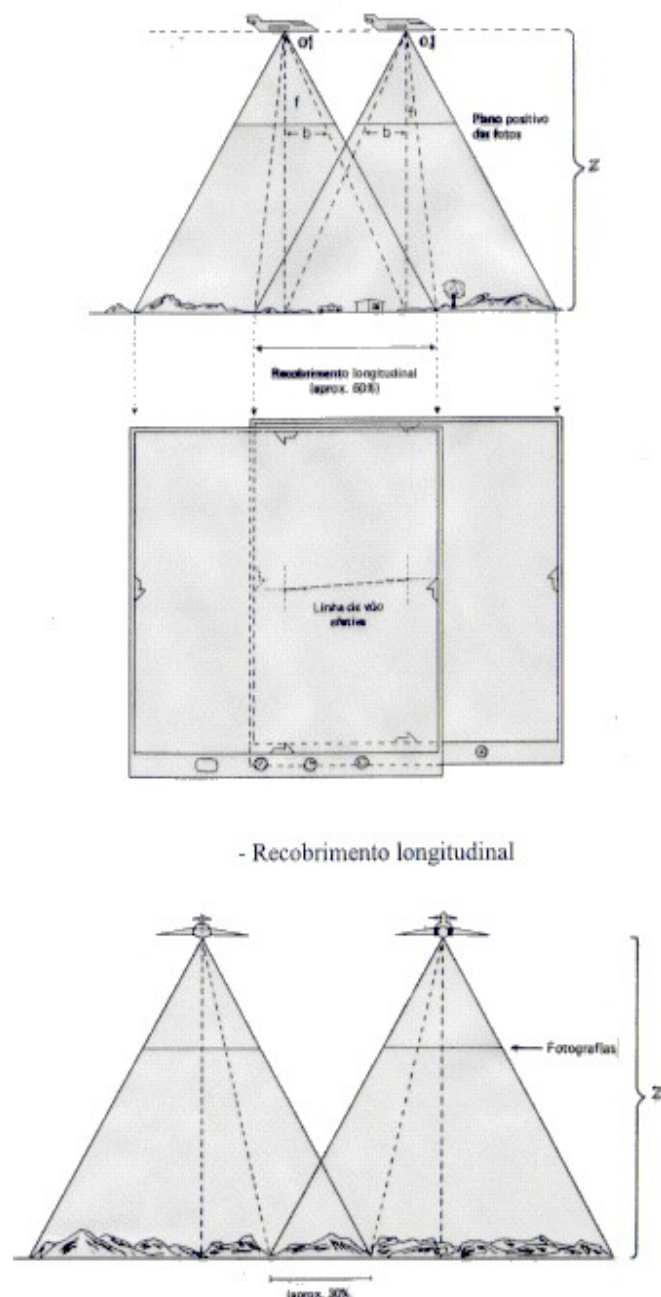


Figura 5.3 – Recobrimento lateral

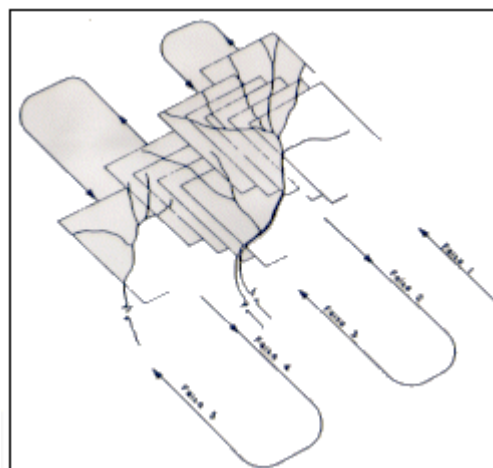


Figura 5.4 – Perspectiva de 04 faixas de vôo

O recobrimento de 60% tem como objetivo evitar a ocorrência de "buracos" (área sem fotografar) na cobertura. Estes podem ocorrer principalmente devido às oscilações da altura de vôo e da ação do vento (figuras 5.5. e 5.6).

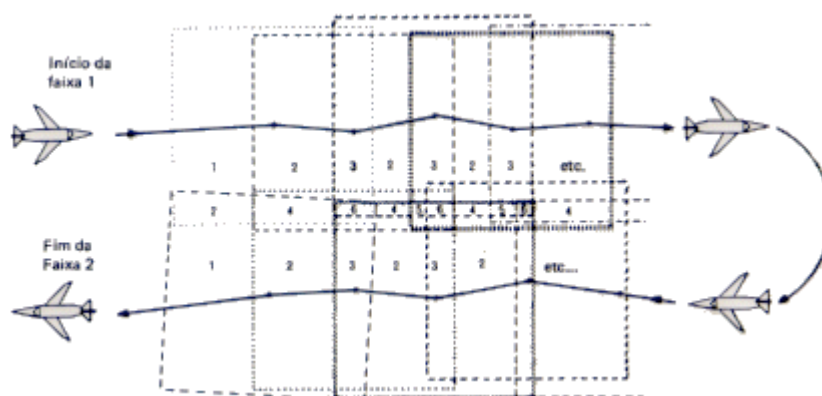


Figura 5.5 – Recobrimento com a ocorrência de deriva e desvio

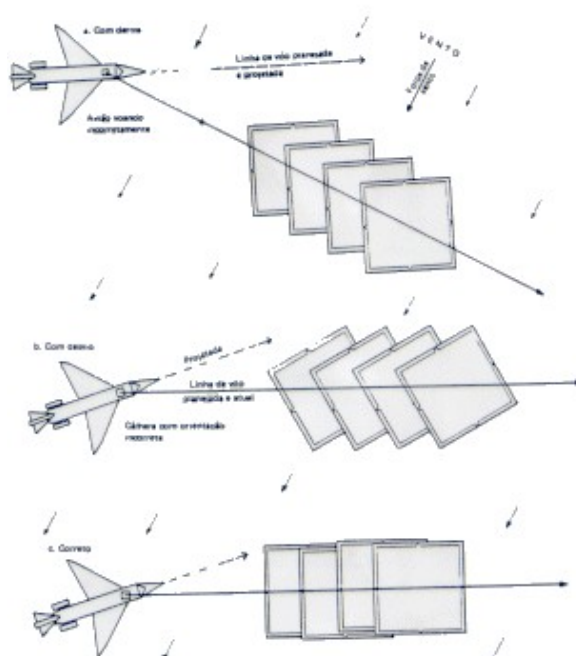


Figura 5.6 – Efeitos da deriva e desvio [43]

5.3 Análise da Imagem utilizada na Dissertação

A imagem “*raster*”, do ano de 2000, da área de estudo, foi digitalizada com 300 *pixels* por polegada a partir de uma aerofoto de dimensões 23cmx 23 cm, produzida pela empresa Esteio numa escala de 1:8000, fornecida pelo Instituto Pereira Passos. Outras duas imagens, do ano de 1975 e 2004, são utilizadas como referências para uma melhor compreensão da área.

Escala é a relação entre a medida de um objeto ou lugar representado no papel e sua medida real, a escala numérica indica a relação entre os comprimentos de uma linha na carta e o correspondente comprimento no terreno, em forma de fração, com a unidade para numerador. A relação d/D significa que as dimensões no desenho são menores que as naturais ou do modelo

As escalas mais comuns têm para numerador a unidade e para denominador, um múltiplo de 10. Isto significa que 1cm na carta cuja escala é 1:25.000 corresponde a 25.000 cm ou 250 m no terreno. Uma escala é tanto maior quanto menor for o denominador como, por exemplo, 1:50.000 é maior que 1:100.000.

Precisão Gráfica é a menor grandeza medida no terreno, capaz de ser representada em desenho na mencionada escala.

A experiência demonstrou que o menor comprimento gráfico que se pode representar em um desenho é de 1/5 de milímetro ou 0,2 mm, sendo este o erro mínimo admissível. Fixado esse limite prático, pode-se determinar o erro tolerável nas medições cujo desenho deve ser feito em determinada escala. O erro de medição permitido será calculado da seguinte forma:

$$E = 1/M e = 0,0002m \times M$$

Sendo *e* igual erro tolerável em metros.

O erro tolerável, portanto, varia na razão direta do denominador da escala e inversa da escala, ou seja, quanto menor for a escala, maior será o erro admissível. Os acidentes cujas dimensões forem menores que os valores dos erros de tolerância, não serão representados graficamente. Em muitos casos é necessário utilizar-se convenções cartográficas, cujos símbolos irão ocupar no desenho, dimensões independentes da escala.

Na escala 1:50.000, o erro prático (0,2 mm) corresponde a 10m no terreno. Verifica-se então que multiplicando 10 x 5.000 encontrar-se-á 50.000, ou seja, o denominador da escala mínima para que os acidentes com 10m de extensão possam ser representados. [43]

Alguns cálculos matemáticos podem ser feitos em relação à aerofoto da área escolhida cuja escala é 1:8000. Sendo o tamanho da foto de 23cmx23cm, 23cm multiplicados por 8000 resultam em 184000 cm. A área real trabalhada é de (1840m x 1840m) = 3385600 m², ou seja, 3,385 km².

Análise de tamanho em km² do “*patch*” ou *pixel* usado no NetLogo a imagem “*raster*” utilizada, tendo como a área total 3,385 km² e a imagem no NetLogo possui 303.201 “*patches*”, logo um “*patch*” da dissertação corresponde à 3,385 km² / 303201 “*patches*”, ou seja, a área real por “*patch*” é de 11,166 m². Lembramos que os agentes atuam sobre um “*patch*”, assim podemos ter a ordem de grandeza envolvida na dissertação.

Estes cálculos têm por objetivo permitir avaliar as dimensões da área que está sendo estudada, são valores estimados, porém permitem ter uma noção bem próxima do real e de como se deu a ocupação do espaço físico da favela do Borel.

CAPÍTULO 6 – DESENVOLVIMENTO DO SISTEMA COM O NETLOGO USANDO UMA IMAGEM DA ÁREA ESCOLHIDA

6.1 Fundamentos do Modelo e sua Implementação no NetLogo

O objetivo desta pesquisa é estudar alguns fenômenos antrópicos a partir de uma representação precisa e detalhada do ambiente em que eles ocorrem. Para servir como base para o trabalho aqui apresentado foi utilizada uma fotografia aérea da Usina da Tijuca,

O ambiente de simulação é o espaço para movimentação e interação dos agentes. Neste trabalho os agentes correspondem a pessoas imersas em uma determinada região geográfica. Além disso, deve-se especificar uma representação para estes objetos. O objetivo da definição do ambiente é criar um modelo do espaço geográfico onde o fenômeno de interesse ocorre. A seleção dos objetos da região a serem representados no modelo está relacionada com os objetivos da simulação, porque os agentes também interagem com eles.

Em primeiro lugar precisa-se representar o ambiente onde os agentes vivem e agem. Este ambiente deve estar em correspondência direta com o espaço geográfico, onde as cidades ficam situadas. Uma imagem (“*raster*”) é ideal para esses propósitos. Ela representa uma região de espaço ocupada por florestas, áreas públicas, bairros tradicionais, favelas, áreas ao longo dos rios, etc. Cada “*pixel*” ou célula representa uma parte da área em estudo, cujo tamanho real é fornecido pela escala associada à imagem. Os agentes seriam, em princípio, as contrapartes de *software* dos seres humanos, que são, por fim, os principais responsáveis pela ocupação da terra nas cidades. Cada tipo de uso da terra é mapeado em cores específicas. Assim, têm-se as áreas mapeadas (subconjunto de células) em cores, de acordo com o uso da terra, resultando então, em um mapa temático do uso da terra, por exemplo: i) áreas públicas como viadutos, praças, parques; ii) áreas de proteção ambiental - florestas urbanas, zonas ao redor de rios; iii) zonas residenciais ocupadas pela classe rica; classe média e classe pobre; iv) distritos industriais etc.; v) edificações e terrenos (públicos ou particulares) abandonados etc.

Normalmente uma imagem de satélite ou imagem aerofotogramétrica é usada para produzir o mapa temático de uso da terra, empregando a técnica de supervisão classificada. Neste caso precisa-se de um profissional com profundo conhecimento sócio-espacial da cidade em questão. Este profissional classificará a imagem, a partir de algumas categorias predefinidas como mostrado no parágrafo anterior.

A representação da imagem classificada resultante é o ambiente onde os agentes vivem. Os agentes vão agir sobre as células já coloridas em função do uso da terra. Fazendo assim, eles percebem o ambiente, reconhecendo áreas disponíveis para serem ocupadas, e reagem a

isso imitando o comportamento humano no que concerne a evolução de cidade ao longo do tempo e, particularmente, no crescimento das favelas.

No mundo real o nascimento e crescimento de favelas são determinados pelas seguintes condições: a disponibilidade de áreas que possam ser facilmente ocupadas preferencialmente sem resistência. A cidade de Rio de Janeiro, por exemplo, tem um perfil topográfico muito irregular, que é pontilhado por morros e montanhas diversos e que abriga uma das mais conhecidas florestas urbanas do mundo. Estas áreas estão entre as mais receptivas e/ou favoráveis para o crescimento das favelas já existentes ou surgimento de novas. A estas áreas pode-se agregar: i) o espaço vazio debaixo de viadutos, as margens de rios, terrenos e edificações (públicos ou particulares) abandonados; ii) a existência de oportunidades de emprego nas proximidades; iii) a proximidade de transporte público, ruas, ou estradas; iv) facilidades para fazer compras, e outros serviços (escolas, bancos etc.); v) atrativos do local escolhido; vi) A vizinhança de uma favela já existente.

As pessoas que vivem perto de favelas têm problemas como: desvalorização da propriedade, proximidade de violência (balas perdidas, tráfico de drogas), e poluição visual. Assim, as áreas na cercania de favelas já existentes acabam sendo candidatas fortes a se tornarem também uma favela.

No modelo de simulação levam-se em conta estas condições determinantes, como também os fatores sociais e macroeconômicos listados a seguir e que influenciam ou regulam a velocidade com que se dá o nascimento e disseminação das favelas: i) O taxa da migração rural; ii) a taxa de crescimento econômico em certos casos é insuficiente para absorver a mão-de-obra que deseja entrar no mercado de trabalho, o que implica em um crescimento em potencial da população favelada ; iii) falta de legislação que disciplina o uso da terra em áreas urbanas; iv) falta de execução da legislação existente sobre as terras em áreas urbanas; v) a taxa de natalidade de habitantes de favela mais rápida que a taxa de natalidade da população tradicional em áreas urbanas; vi) falta de planejamento urbano.

A área selecionada no bairro da Usina tem todas as principais características necessárias para o crescimento de favelas e surgimento de novas, servindo de base para realizar a simulação em questão.

A ocupação do bairro da Tijuca, que se intensificou a partir do desenvolvimento dos bondes no final do século XIX, inicialmente ao longo da rua Conde de Bonfim e depois se estendendo por toda a área entre os morros da região, já está consolidada. É uma das áreas mais populosas da cidade do Rio de Janeiro, com poucos terrenos vazios, entre os quais estão o remanescente do metrô. Os morros que cercam o bairro (Andaraí, Borel, Formiga, Salgueiro etc.), integrantes do maciço da Tijuca, são marcados pela ocupação de favelas.

Trata-se de uma zona residencial, de habitações basicamente multifamiliares, verificando-se uma diminuição populacional nas proximidades do trecho final da rua Conde de Bonfim, área da Usina. Existe um centro comercial e de serviços na Tijuca bastante desenvolvido e diversificado. E, ainda nesta região se realiza uma grande circulação de transportes coletivos, individuais e de pedestres.

A área de estudo também se caracteriza por ter fatores que influenciam a velocidade com que se dá a ocupação das áreas. Grande parte da população vem do Nordeste e de Minas Gerais, procurando os morros do bairro por serem próximos ao centro e pela boa oferta de trabalho que o bairro oferece. A Tijuca, atualmente é um bairro autônomo e de grande absorção de mão-de-obra, devido ao comércio, às indústrias e sua população majoritariamente de classe média. Existe também uma legislação que teoricamente protege as regiões dos morros acima da curva de nível de 100 metros, consideras Zonas Especiais (ZE1) [24].

6.2 Programando o Netlogo

O NetLogo é um ambiente de modelagem programável para explorar o comportamento de sistemas descentralizados e sistemas complexos. É particularmente bem situado para modelagem de sistemas complexos desenvolvidos no decorrer do tempo. O NetLogo fornece um modo fácil de começar a modelagem baseada em agentes, mesmo para aqueles que não têm avançada habilidade em programação e matemática. O ambiente NetLogo é composto de três tipos de agentes: *turtles*, *patches*. O modelador pode dar, simultaneamente, instruções a centenas ou milhares de “agentes” independentes que trabalham paralelamente, tornando possível explorar as conexões entre o comportamento de micro-níveis e os de modelos de macro-níveis que emergem das interações de muitos indivíduos. NetLogo usa uma linguagem de modelagem própria, derivada da linguagem de programação Logo do Media Lab – MIT [44].

A programação em NetLogo consiste em atribuir comportamentos a três grupos de agentes: o observador (“*observer*”), os agentes (“*turtles*”) e o ambiente (“*patches*”). O observador é um “criador” que especifica as condições de funcionamento e dá controle aos demais elementos. O observador não tem uma localização, ele nos possibilita uma visão global do mundo dos agentes e “*patches*”. Os agentes são seres que se movem pelo mundo. O mundo é bidimensional e é dividido em uma grade de “*patches*”. Cada “*patch*” é um quadrado de “chão” em cima do quais os agentes se movem, os “*patches*” são como células de uma matriz.

Quando o NetLogo inicia, não há ainda nenhum agente. O observador pode fazer novos agentes. “*Patches*” podem fazer novos agentes também. (“*Patches*” não podem se mover, mas de qualquer modo eles estão “vivos” como os agentes e o observador).

“*Patches*” têm coordenadas. No “*patch*” as coordenadas (0, 0) indicam a origem e as coordenadas dos outros “*patches*” são as distâncias horizontais e verticais em relação à origem.

Pxcor e Pycor são as coordenadas dos "*patches*". Como nas coordenadas matemáticas padrão de vôo, pxcor aumenta quando se move para direita e pycor aumenta quando se move para cima.

O número total de "*patches*" é determinado pelo min-pxcor, max-pxcor, min-pycor e max-pycor da configuração. Neste trabalho min-pxcor, max-pxcor, min-pycor e max-pycor são respectivamente -265, 265, -285 e 285. Isto significa para calcular número de "*patches*" a conta a ser feita é $(265 * 2 + 1) \times (285 * 2 + 1)$ num total de 303201 "*patches*". (Este número de "*patches*" está relacionado à aerofoto da área da Usina usada nesta dissertação.)

Os agentes também têm coordenadas (xcor e ycor). As coordenadas de um "*patch*" sempre são inteiras, mas as coordenadas de um agente podem ter decimais. Isto significa que um agente pode estar posicionado em qualquer ponto dentro de seu "*patch*"; não precisa estar no centro do "*patch*". Um agente quando realiza uma ação sobre o "*patch*", ele age somente em um único "*patch*".

O mundo dos "*patches*" pode ter formas diferentes. Por default o mundo do "*patches*" é um *torus* isto significa que não existem fronteiras. Assim quando um agente se move além da extremidade do mundo, desaparece e reaparece na extremidade oposta, e todo "*patch*" tem o mesmo número de "*patches*" "vizinhos", se você for um "*patch*" na extremidade do mundo, alguns de seus "vizinhos" estão na extremidade oposta. Porém, você pode mudar a forma do ambiente ajustando as configurações de envoltura editando a visão, definindo que o mundo acaba no seu limite sem a forma de um *torus*. Os "*patches*" ao longo daquele limite têm menos que oito vizinhos conforme o padrão do NetLogo e os agentes não se movem além da extremidade do mundo.

O Ambiente

O ambiente onde os agentes (*turtles*) vivem é um tabuleiro em forma de quadrilátero, com dimensões especificadas pelo programador. Sobre este tabuleiro os agentes podem se deslocar. Quando as bordas são atingidas os agentes simplesmente continuam o seu percurso no lado oposto ao que se encontrava no tabuleiro. O tabuleiro é formado por uma matriz de pequenas células (*patches*), também tratados como agentes.

Os Agentes

O ambiente pode ser povoado por agentes, aos quais podemos atribuir comportamentos, denominados de "*procedures*". Por exemplo, um grupo de agentes se movimenta aleatoriamente

pelo ambiente reconhecendo as variáveis do ambiente como as suas próprias, modificando o ambiente com uma outra cor qualquer.

O Observador

Para comandar o processo de simulação temos um tipo de agente especial, o “observador”. O observador é o responsável por povoar o ambiente com seus agentes.

O ambiente Gráfico

O ambiente NetLogo apresenta na sua interface principal três janelas denominadas de “*Interface*”, “*Information*” e “*Procedures*”.

Janela *Interface*

A janela de *Interface* é onde se visualiza a execução do modelo. Quando o NetLogo é aberto a janela de interface está vazia, inclusive a janela gráfica onde os agentes e os *patches* aparecem.

A barra de ferramentas da janela de interface contém botões que permitem editar, excluir e criar itens na tabela de interface (*Button*, *Slider*, *Switch*, *Chooser*, *Monitor*, *Plot*, *Output* e *Text*), mais detalhes do uso desses recursos serão apresentados a seguir. A janela gráfica inicialmente aparece sem nenhum agente atuando sobre ela. A criação dos agentes se dá, pela atuação do observador, através de procedimentos ou pela janela de comando.

Janela *Procedures*

A janela *Procedures* é o local onde se escreve e armazena o código dos modelos. Essa janela é útil para escrever modelos mais longos ou com vários procedimentos. É através da escrita do código que se define o comportamento dos agentes.

Janela *Information*

A Janela *Information* é utilizada para descrever uma introdução do modelo e explicar seu funcionamento e modo de usar, além do que pode ser explorado, possíveis extensões e

características sobre o modelo e o ambiente.

Ambiente de Execução

O ambiente Netlogo apresenta uma série de funcionalidades para a programação. A execução dos modelos é visualizada através da janela de interface. Nele observamos e controlamos o comportamento dos agentes através dos botões de controle, sendo que cada um deles está associado a um código que é preenchido na janela *Procedures*. Os comportamentos dos agentes também podem ser controlados através da janela do *Command Center* (centro de comando), que aceita receber os comandos do NetLogo.

Nesta janela é possível executar comandos do observador, das *turtles* e dos *patches*. Os botões (*Button*), de modo geral, servem para disparar um conjunto de procedimentos, eles podem ser programados para executar apenas uma vez ou continuamente o procedimento.

Os *Sliders* acumulam variáveis globais que são acessíveis a todos os agentes. Eles são usados como um modo rápido de alterar os valores das variáveis nos modelos sem necessidade de alterar o programa e gravar o procedimento. Janelas do tipo *Plot* mostram gráficos gerados por dados do modelo em tempo real.

Os *Switches* são representações visuais de verdadeiro ou falso para as variáveis. O usuário tem a opção de fazer essa escolha na execução do modelo. Através dos *Monitors* é possível mostrar qualquer expressão, podendo essa ser uma variável, ou uma expressão mais complexa, por exemplo, um cálculo. Eles podem também apresentar um contador.

Text abre uma caixa de diálogo para informações e/ou explicações sobre elementos do modelo. *Janela gráfica* permite observar o comportamento dos agentes no momento da execução do modelo. A janela gráfica inicialmente aparece como um quadrilátero preto sobre a tabela de interface. Esse é o mundo gráfico das tartarugas e *patches* do NetLogo. Há nessa janela um controlador de velocidade dos agentes que pode ser utilizado em qualquer momento da execução do modelo. Em alguns modelos é possível interagir com as tartarugas e os *patches* clicando sobre eles com o mouse e arrastando-os. Caso queira inspecionar algum dos agentes em particular, basta selecioná-lo com o botão direito do *mouse* e optar por “*inspect turtles*” ou “*inspect patch*” que um menu *pop-up* apresenta (ou ficará apresentando) as características do agente selecionado (caso o sistema esteja em execução).

6.3 O Modelo

O ambiente da simulação é uma imagem temática, representada como uma estrutura de ordem bidimensional. Esta imagem temática representa usos diferentes dos espaços físicos no bairro da Usina. Cada “*pixel*” bidimensional é um “*patch*”. Agentes representam as pessoas que

definem o uso desses espaços e podem alterar o estado da cor dos “*patches*”, guiados pelo valor de algumas variáveis globais que simbolizam fatores macroeconômicos e sociais que influenciam o crescimento das favelas. Os “*patches*” que mudam o seu estado são os mais receptivos ou favoráveis a expansão de favela. O observador é responsável por criar os agentes e atualizar e mostrar o estado de ambiente global, descrevendo a evolução das favelas.

O desenvolvimento do estudo foi realizado no NetLogo, utilizando-se uma aerofoto da área da Usina onde se encontram as favelas do Borel e Indiana, referente ao ano 2000 produzida pela empresa Esteio e cedida pelo Instituto Pereira Passos.

Esta aerofoto foi escaneada no formato JPG, colorida usando o Adobe Photoshop ainda no formato JPG e finalmente levada para o formato PPM com o *software* Irfanview. Foi então lida pelo NetLogo e transformada em arquivo formato DAT. No NetLogo a imagem classificada passa a ser tratada como um ambiente matricial – coleção de “*patches*” que são as células da matriz.

A programação do modelo precisava trabalhar os critérios importantes para selecionar as áreas a serem ocupadas, bem como os fatores que influenciam na velocidade com que se dá a ocupação de áreas abandonadas, de forma a permitir alcançar os objetivos desejados.

A programação foi realizada definindo os atributos e variáveis aos “*patches*” e aos agentes, de forma a poder atingir o objetivo desejado. Para os “*patches*” das áreas classificadas, foram criados atributos como elevação, abandono, proteção ambiental etc. Em relação à população se considerou taxa de natalidade e mortalidade.

Os agentes reativos, estando situados sobre o “*patches*” têm capacidade de perceber os atributos dos respectivos “*patches*”, como também analisar as variáveis definidas no modelo, com isto eles conseguem saber se podem naquela área construir o seu barraco ou não.

No modelo também foi feita uma simulação de favelização do supermercado Carrefour localizado naquela área, que se encontra atualmente desativado.

Uma explicação do funcionamento do modelo quanto ao uso da janela gráfica, botões, *sliders*, monitor, etc é mostrada a seguir.

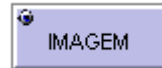
A janela gráfica permite observar o comportamento dos agentes no momento de execução do modelo na área da Usina.



O botão *Setup* ao ser acionado inicia os agentes e as variáveis. O botão *Go* inicia a execução do modelo.



Esses botões ao serem acionados mostram as imagens das aerofotos e dos croquis.



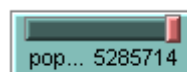
Esse botão quando acionado mostra a imagem classificada.



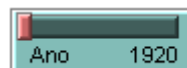
Esse botão quando acionado elimina os agentes.



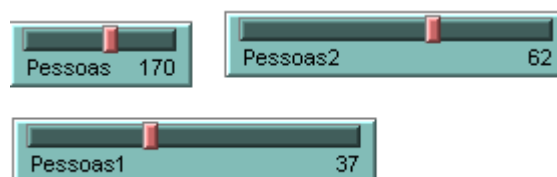
Esses botões quando acionados, um salva a imagem corrente o outro retorna a última imagem salva.



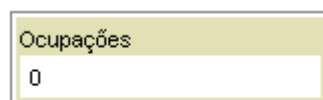
Esse *slider* acumula a população do município do Rio de Janeiro, pode-se alterar os valores para diferentes análises.



Esse *slider* acumula os anos, o modelo reage conforme o ano escolhido.



Esses *sliders* acumulam os agentes que atuam no Borel e Indiana (pessoas), Chácara do Céu (pessoas1) e Carrefour (pessoas2), o número de agentes altera a velocidade de ocupação.



Esse botão monitora o número de ocupações realizadas.

Ao executarmos o modelo, o procedimento interno *Startup* é acionado, então definimos as variáveis, acionamos o botão *Setup* e depois o botão *Go*. Realiza-se a simulação para todas as décadas até concluirmos a simulação e alcançarmos as metas desejadas.

6.4 NetLogo Usando uma Imagem “*raster*”

Para realizar a dissertação era preciso usar uma imagem “*raster*” para representar a área a ser trabalhada. Como o NetLogo ainda não tem capacidade de importar diretamente uma imagem “*raster*”, foi preciso usar alguns recursos de programação do próprio NetLogo a fim de conseguir êxito para importar a imagem “*raster*”.

Sabendo que o NetLogo tem capacidade de importar arquivos PPM, foi preciso primeiramente transformar a imagem “*raster*” no formato JPG em uma imagem PPM (*Portable Pixmap*). Entre vários *softwares* especialistas em tratar imagens, fez-se a escolha do programa IRFANVIEW para a conversão do formato JPG para PPM.

O arquivo PPM

O arquivo PPM consiste de duas partes: um cabeçalho e os dados da imagem. O cabeçalho é estruturado em no mínimo três partes normalmente finalizadas por um “*carriage returns*” e/ou “*linefeeds*”, mas as especificações PPM só exigem espaço em branco. A primeira parte é um número mágico, o identificador PPM, pode ser "P3" ou "P6" (não incluem aspas). A segunda parte são os dois próximos números que definem a largura e altura da imagem. A última parte do cabeçalho dá o número máximo dos componentes de cores dos “*pixels*” (0..255) a serem usados . Um comentário pode ser incluído com o caractere "#". A seguir um cabeçalho PPM válido.

Exemplo cabeçalho 1

P6 1024 788 255

Exemplo cabeçalho 2

P6

1024 788

A comentário

255

Exemplo cabeçalho 3

P3

1024 # largura da imagem

788 # altura da imagem

A comentário

1023

O formato próprio dos dados de imagem depende do identificador mágico PPM. Se for o "P3" então a imagem é determinada como um texto ASCII, o valor numérico de cada “*pixel*” varia de 0 ao valor máximo dado no cabeçalho. As linhas não devem ser mais longas que 70 caracteres.

Exemplo cabeçalho 43

P3

#

4 4

15

0 0 0 0 0 0 0 0 0 15 0 15

0 0 0 0 15 7 0 0 0 0 0 0

0 0 0 0 0 0 0 15 7 0 0 0

15 0 15 0 0 0 0 0 0 0 0 0

Se o identificador PPM for o "P6" então os dados da imagem são armazenados no formato byte, tendo um byte por componente de cor (r, g, b). Comentários só podem acontecer antes do último campo do cabeçalho e só um byte pode aparecer depois do último campo de cabeçalho, normalmente “carriage returns” ou “linefeeds”. Arquivos de imagem "P6" são obviamente menores que "P3" e muito mais rápidos de ler. Note que arquivos PPM com identificador "P6" só podem ser usados para cores de um único byte. Cada “*pixel*” é armazenado como um byte, valor 0 (preto), valor 255 (branco). Os componentes são armazenados na ordem "habitual", vermelho, verde e azul (r,g,b). [45]

O segundo passo foi montar um código no software NetLogo para poder ler o arquivo PPM. Com a montagem desse código foi possível viabilizar a modelagem prevista.

O NetLogo já lia o arquivo PPM, porém a geração da imagem no computador estava muito lenta inviabilizando o modelo.

O próximo passo foi montar outro código para transformar o arquivo PPM em um arquivo DAT dentro da programação NetLogo. Assim o modelo ficou sensivelmente mais rápido e pronto para poder realmente desenvolver o modelo desta dissertação.

CAPÍTULO 7 – ANÁLISE DOS RESULTADOS

O estudo da dissertação se fundamentou na evolução da Favela do Borel e da área ao seu redor. Localizadas na zona Norte da cidade do Rio de Janeiro, no bairro da Usina na Rua São Miguel, junto a uma área de proteção ambiental. É uma das mais antigas favelas da cidade do Rio de Janeiro, tendo surgido por volta de 1920.

A área da Usina, onde já existem várias favelas estabelecidas, é um local com prédios da classe média que vem tendo uma grande desvalorização imobiliária ao longo dos anos.

O supermercado Carrefour, de grande porte, foi desativado em parte por estes motivos. Em suma, é uma área que resume as características necessárias à simulação da disseminação de favelas em ambiente urbano, conforme descrito neste trabalho.

A validação do modelo foi feita a partir da comparação dos resultados finais da simulação com os dados adquiridos junto à equipe Arqui-Traço [24] e com as aerofotos disponíveis para os anos 1975, 2000 e 2004.

No ano 2000 a população de Borel era de 6831 pessoas distribuídas em 1844 domicílios. Na figura 7.1, temos uma fotografia aérea de favela de Borel tomada no ano 2000, pela empresa Esteio e fornecida pelo Instituto Pereira Passos, mostrando sua localização.



Figura 7.1 – Aerofoto da favela do Borel em 2000

Para validar o modelo reconstruiu-se o nascimento e crescimento da favela do Borel, tendo como ponto de partida, os croquis levantados pela equipe Arqui-Traço, das décadas de

1920-1940, 1950-1970, 1980-1990 e uma fotografia aérea do ano 2000. No período 1920-1940, a favela cresceu basicamente ao longo da Rua São Miguel (cor marrom). Este assentamento inicial, destacado em cor amarela, na imagem superior esquerda da Figura 7.2. Depois da abertura da estrada da Independência (1950-1970), uma estrada serpentina que une a Rua São Miguel ao alto do morro do Borel, a favela estendeu-se as partes mais altas, acompanhando a estrada (imagem superior direita da Figura 7.2, cor amarela). Finalmente, no período 1980-1990 a favela cresceu ao redor das áreas já estabelecidas, até que achou obstáculos físicos e estabeleceu-se num formato muito próximo ao de hoje mostrado na imagem inferior esquerda da figura 7.2 cor amarela [24].

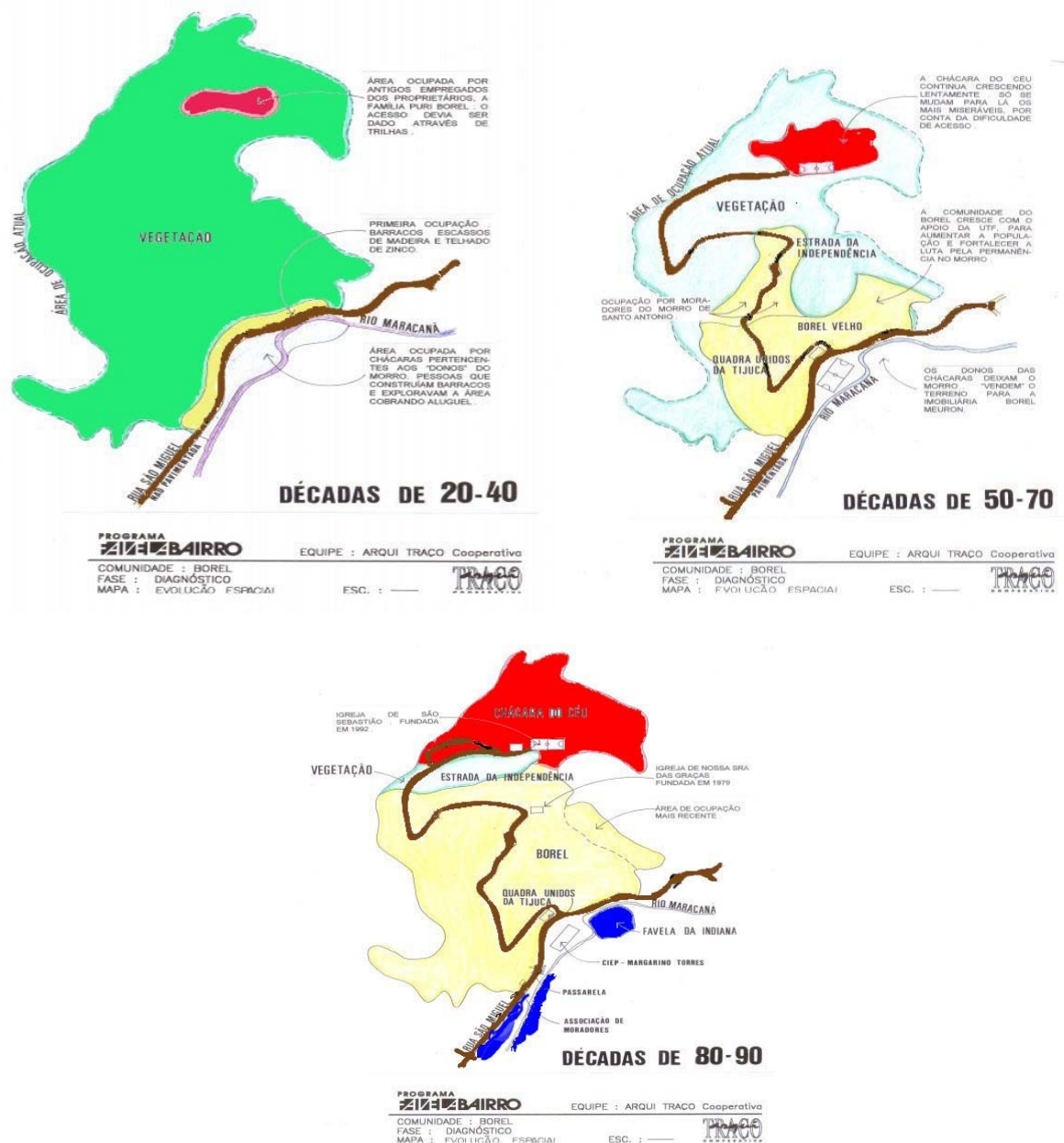


Figura 7.2 – Evolução da favela do Borel representada na cor amarela
Na simulação, realizada sobre a imagem “*raster*” da área da Usina, os agentes reativos vão interagir com o terreno e colorir, segundo critérios previamente estabelecidos a imagem produzida nesta dissertação, com cores diferenciadas, de forma a poder validar a ocupação.

A parte da imagem com cor amarela representa o formato próximo do atual da favela do Borel e a parte de cor vermelha, a Chácara do Céu uma continuação do Borel embora os moradores não se sintam parte da comunidade do Borel

Os agentes vão colorir as áreas com cores diferentes, representando a evolução da ocupação ao longo dos anos, e evidenciando assim as diferenças. Cinza para década de 20-40, preto para década de 50-70, turquesa para década 80-00.

As favelas Indiana I e Indiana II que se localizam em frente à favela do Borel no outro lado da Rua São Miguel, também foram priorizadas neste trabalho. A Indiana I foi colorida da cor amarela e a Indiana II que situa ao longo do Rio Maracanã foi colorida de azul. Neste modelo, os agentes vão simular a ocupação colorindo a Indiana I de preto e a Indiana II de rosa.

O supermercado Carrefour, por estar desativado, passa a ser uma edificação de alto risco de invasão, estando a população à espera de uma oportunidade para vir ocupá-la. Os agentes percebendo que a área se encontra abandonada, colorem de cyan a área do Carrefour simulando uma possível ocupação futura. A figura 7.3 mostrada abaixo, é a imagem classificada a ser utilizada neste trabalho.

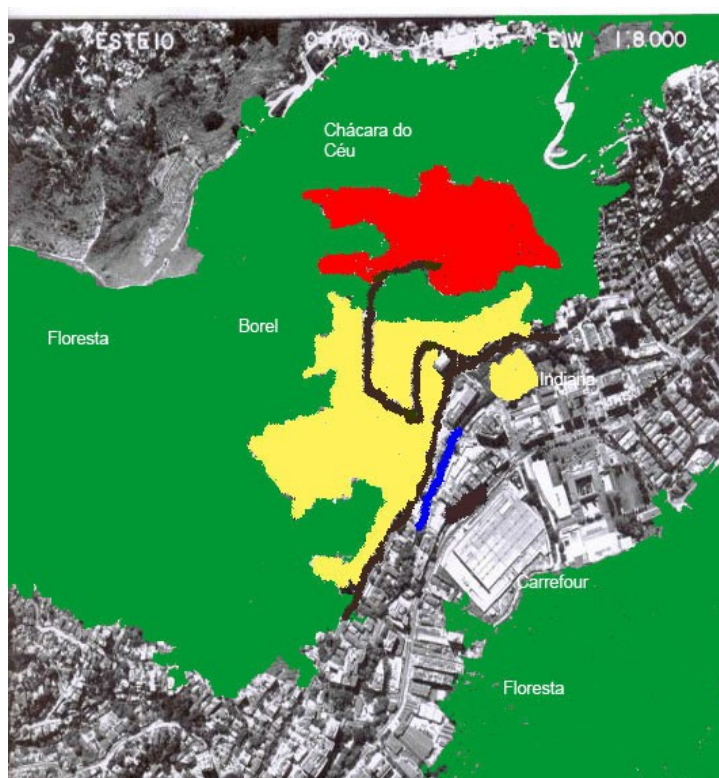


Figura 7.3 – Imagem classificada

A figura 7.4 é uma cópia da interface gráfica gerada no modelo NetLogo após a execução da simulação do ambiente referente às décadas de 20-40 mostrando a imagem obtida. A imagem resultante após a execução do modelo, mostra a evolução da ocupação do espaço físico da área das favelas, das décadas de 20-40.

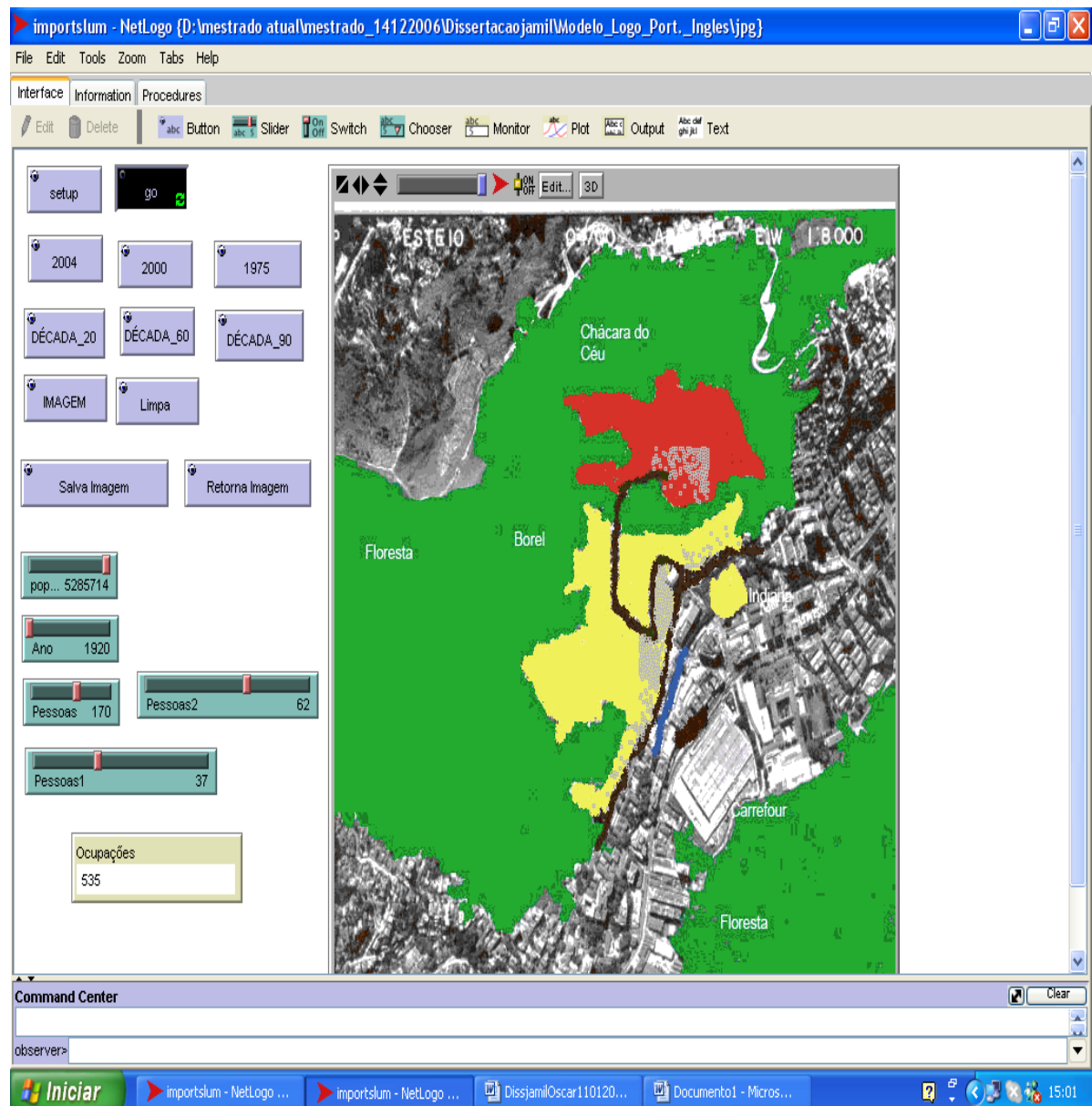


Figura 7.4 – Evolução da ocupação da favela após a simulação das décadas de 20-40.

A figura 7.5 é uma cópia da interface gráfica gerada no modelo NetLogo após a execução da simulação do ambiente referente à década de 50-70 mostrando a imagem obtida. A imagem resultante após a execução do modelo mostra a evolução da ocupação do espaço físico da área das favelas, das décadas de 50-70.

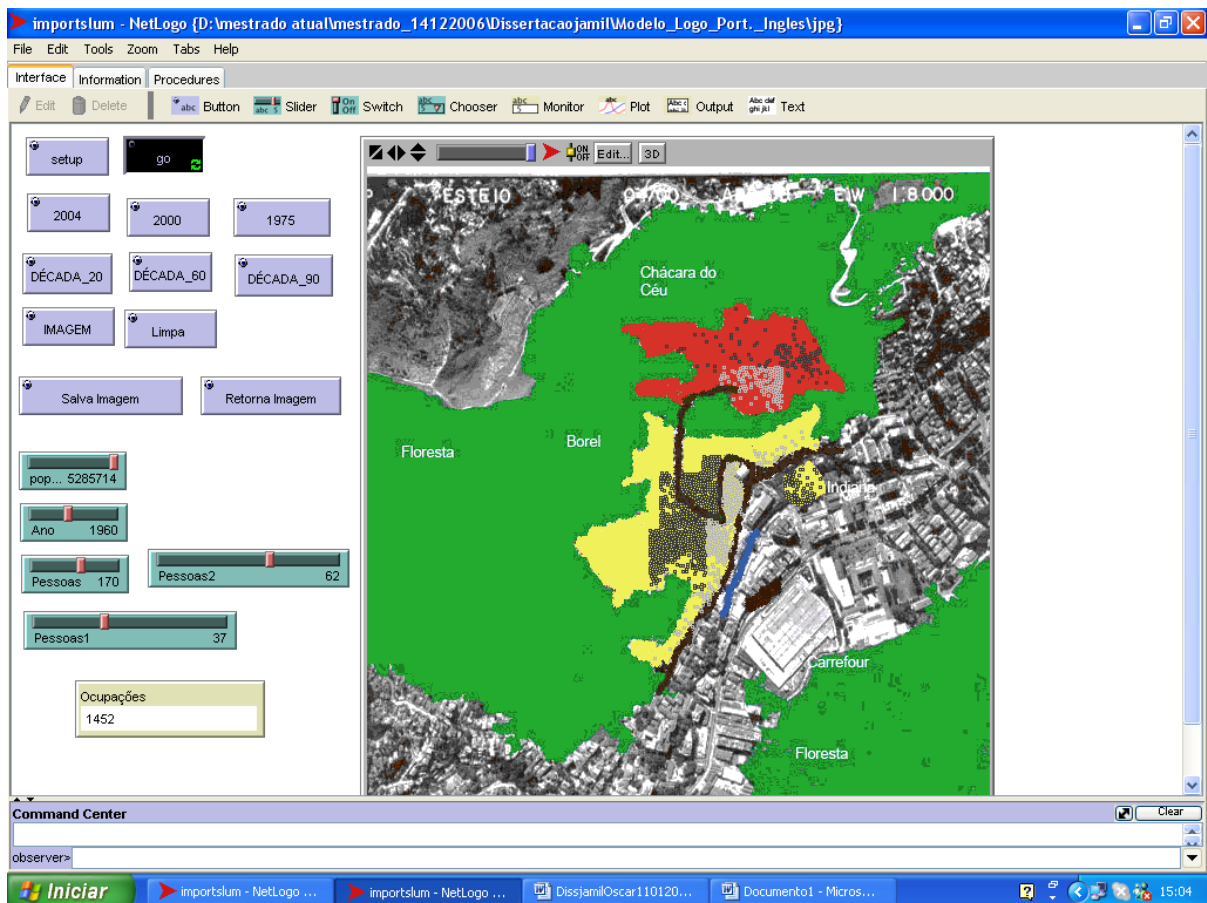


Figura 7.5 – Evolução da ocupação da favela após a simulação das décadas de 50-70

A figura 7.6 é uma cópia da interface gráfica gerada no modelo NetLogo após a execução final da simulação do ambiente mostrando a imagem obtida. A imagem final resultante após a execução do modelo mostra a evolução da ocupação do espaço físico da área das favelas, desde 1920 até os dias atuais.

Depois de executar a simulação e alcançada a distribuição de espaço físico da favela como mostrado na imagem reproduzida na interface gráfica da figura 7.6, chegou-se muito próximo à forma atual de favela do Borel e das áreas adjacentes.

Com a comparação da imagem obtida na simulação do modelo NetLogo (figura 7.6) com a aerofoto de 2000 (figura 7.1) e os croquis (figura 7.2), da equipe Arqui-Traço responsável pelo diagnóstico da favela do Borel e Indiana no programa “Favela Bairro”, pode-se validar o modelo e atestar o êxito da simulação.

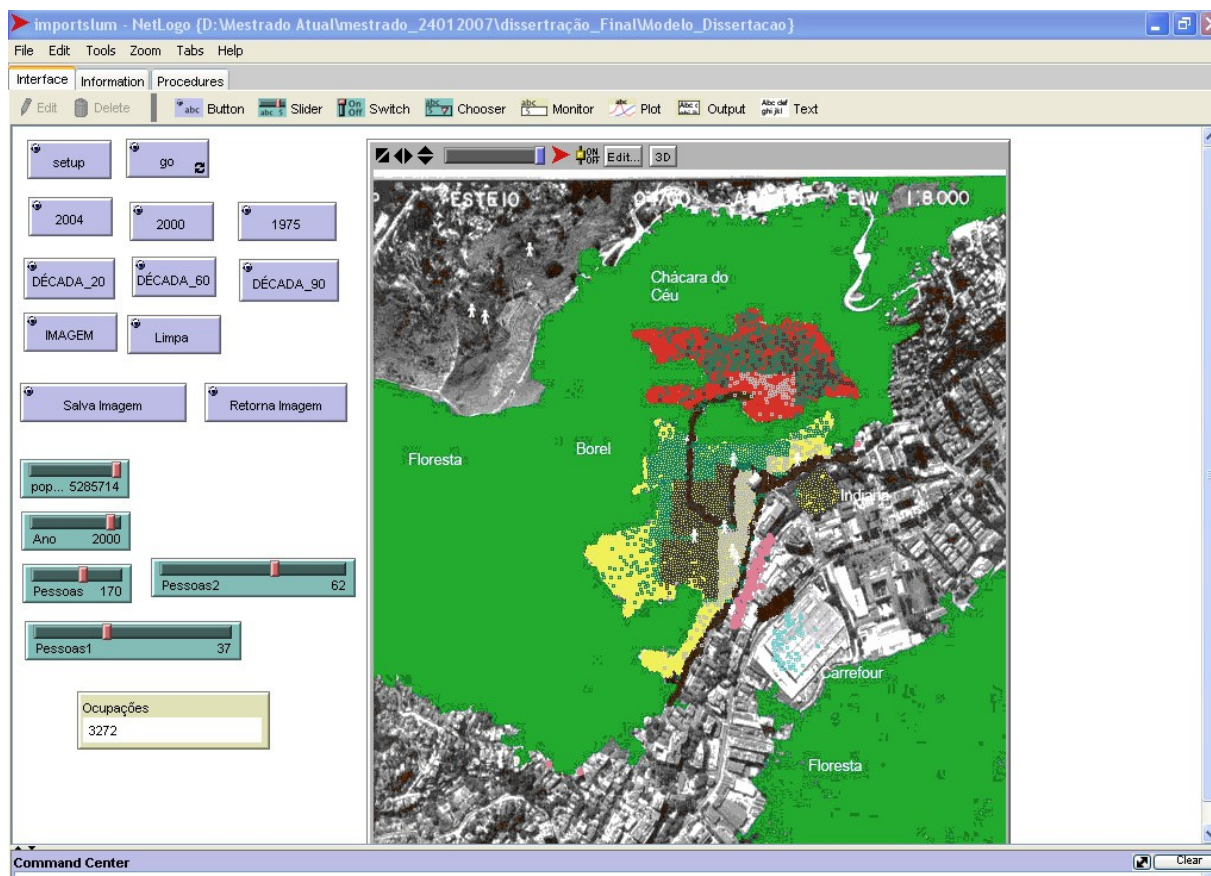


Figura 7.6 – Evolução da ocupação final das favelas e do supermercado Carrefour

CAPÍTULO 8 – CONCLUSÕES

O uso do Sistema Multiagentes aplicado no meio-ambiente, neste trabalho, representado pela imagem “raster” da área da favela abre perspectivas enriquecedoras às pesquisas ambientais. A diversidade de conhecimento necessária ao desenvolvimento deste trabalho, como programação orientada a agentes, dados sociais, uso da geomática, etc mostra que estas novas perspectivas abrem várias oportunidades para ampliação dos estudos ambientais.

O uso de modelos de simulação baseado em agentes reativos para simular sistemas complexos em situações onde a complexidade matemática do modelo analítico dificulte seu uso ou a simulação seja impossível de atingir, é um modo muito prático e útil para obter resultados que seriam impraticáveis de alcançar por outros métodos. O comportamento do sistema complexo emerge como o resultado da interação de múltiplas entidades simples, que agem em base de reação direta a estímulos. O paradigma de sistemas multiagentes reativos pode ser usado para analisar a dinâmica de sistemas espaciais complexos, como o estudo de caso apresentado nesta dissertação sobre a evolução da favela do Borel, no Rio de Janeiro. Este paradigma também poderá ser usado para construir modelos relacionados ao surgimento e expansão de novas favelas, constituindo-se uma ferramenta de simulação muito útil para planejamento urbano e para a elaboração de políticas sociais dirigidas a reduzir o impacto do crescimento descontrolado de áreas de favela.

O programa modelador de ambiente NetLogo mostrou ser bastante eficaz para este estudo de caso e permitiu a implementação de um sistema de multiagentes na área do Borel simulando o ambiente onde as pessoas (agentes) vivem. Foi possível avaliar a ocupação das favelas em um espaço físico real ocupado por pessoas carentes que precisam de um lugar para viver. O objetivo de simular a disseminação de favelas em ambientes urbanos através de agentes reativos foi alcançado e validado pela análise comparativa com as imagens e dados reais.

Os agentes reativos programados para o estudo de sistemas complexos deram resposta satisfatória. Outros estudos sobre este tema podem testar o uso outros tipos de agentes, como agentes cognitivos de modo a obter outra visão, acrescentando novas informações importantes a este estudo.

O estudo, desenvolvido nesta dissertação, poderá ser aplicado a outras áreas abandonadas e testados por outros mestrandos. O índice crescente de novas favelas é muito alto na cidade do Rio de Janeiro e o uso das ferramentas aqui descritas permite que as autoridades possam utilizar as informações existentes e simular cenários de possíveis ocupações, e assim tomar decisões que venham a melhorar as condições de vida da cidade.

Outros estudos de casos utilizando a metodologia aqui desenvolvida podem vir a ser desenvolvidos por outros mestrandos para as mais diversas finalidades. Por exemplo, estudos

sobre o impacto do crescimento das favelas nos sistemas de transporte, saúde, educação e saneamento. Estes trabalhos futuros seriam altamente enriquecedores e complementares ao estudo desenvolvido neste trabalho.

REFERÊNCIAS

- [1] United Nations Human Settlement Programs. *The Challenge of Slums - Global Report on Human Settlements 2003*, ISBN: 1-844407-037-9, pp. 310, 2003. acessado em junho 2006.
- [2] Davis, M. *Planet of Slums*; Urban Involution and the Informal Proletariat; April 2004.
- [3] População das favelas. Estado de São Paulo 12 de fevereiro de 2006; acessado em maio 2006.
- [4] Coelho, S. Pesquisadores quantificam crescimento espacial e populacional de favelas cariocas, <http://www.fiocruz.br/ccs/arquivosite/novidades/out04/favela_sar.htm>, acessado em 9 de julho de 2006.
- [5] Censo Demográfico 2000, Instituto Brasileiro de Geografia e Estatística. IBGE.
- [6] UN Improving the lives of 100 million Slum
- [7] Silva, J. S. Um espaço em busca de seu lugar: as favelas para além dos Estereótipos.
- [8] <http://www.comciencia.br/reportagens/cidades/cid07.htm>, acessado em 10 de julho de 2006.
- [9] Ferrand, N. "Multi-reactive agents paradigm for spatial modelling", In *Spatial Models and GIS – New Potential and New Models*. Edited by Fotheringham and Wegener; GISDATA 7 Series Editors Masser and Sagé; Taylor & Francis group, 2000. pp.167-184.
- [10] Mathian, H.e Mikula, B., and Sanders, L. Modelling the Dynamics of Spatial Systems within GIS: Problems and Perspectives; in *Spatial Models and GIS – New Potential and New Models*. . Edited by Fotheringham and Wegener; GISDATA 7 Series Editors Masser and Sagé; Taylor & Francis, 2000, pp. 203-221.
- [11] Tibaijuka, A., programa Habitat ONU "O estado das cidades do mundo 2006/2007.
- [12] <http://www.unhabitat.org>
- [13] Kirby, A. Slums Growth 'Shames the World'. 6 of October 2003
- [14] <http://www.un.org/esa/population/unpop.htm> UN Population Division.
- [15] Vaz, L. F.e Jacques P. B. Pequeno histórico das favelas no Rio de Janeiro.
- [16] *Favelas Sempre* do Jornal do Brasil de 16/11/00.
- [17] Pontes, F. e Schmidt, S. A invasão silenciosa das favelas.
- [18] Ribeiro, L. C. Q., Lago, L. C. Metrópole, Desigualdades Sócioespaciais, ippur.ufrj.br.
- [19] Pontes, F. e Schmidt, S Cercas para frear as favelas.
- [20] [http:// www.ibge.gov.br](http://www.ibge.gov.br), w3.dpe.ibge.gov.br
- [21] Pontes, F. e Schmidt, S. *Nova política habitacional da prefeitura inclui reassentamentos em casarões no Centro* para "O Globo", de 29/01/2001.
- [22] Pontes, F. e Schmidt, S *Censo do IBGE revela que surgiu uma ocupação por mês na última década*, para "O Globo", de 28/01/2001.

- [23] Os dados mais recentes sobre a população das favelas na cidade do Rio de Janeiro. Instituto Pereira Passos – 2002.
- [24] Equipe Arqui-Traço Cooperativa. *Programa Favela Bairro – Diagnóstico da Favela do Borel*. Rio de Janeiro, Fevereiro 1996.
- [25] Demazeau, Y. e Müller, J. P. Decentralized artificial intelligence. In: Demazeau, Y. e Müller, J.P. eds. *Decentralized A I.*, Amsterdam.
- [26] Álvares, L. O. e Sichman, J. S. Introdução aos Sistemas Multiagentes. Brasília: SBC, JAI, 16., 1997.
- [27] Wooldridge M. J., Jennings R. N.; *Agents theories, Architectures and Languages: A SURVEY*.
- [28] Wooldridge M. J. *An Introduction to MultiAgent Systems*. John Wiley & Sons, Ltd, April 2002.
- [29] Russel, S. e Norvig, P. Inteligência Artificial; Agentes Inteligentes, capítulo 2 – (segunda edição).
- [30] Fagin, et al., 1995, p154
- [31] Russel and Subramanian, 1995, pp. 580, 581.
- [32] Bastos, E. N. F. Uma Plataforma de Sistemas Multiagentes para o Ambiente ZOPE.
- [33] Pontarolo, E., Azevedo, L. L., Magalhães C. S. e Magdalena. S. Exploração do ambiente orientado a agente NetLogo.
- [34] Harvey, B. Computer Science Logo Style, , MIT Press (3 volumes) ISBN 0-262-58148-5, ISBN 0-262-58149-3, ISBN 0-262-58150-7.
- [35] Magnin, L.. *SIEME - Simulateur d'environnement pour systèmes multi-agents*. 1996. [<http://www-laforia.ibp.fr/magnin/these/sieme>].
- [36] Minar, N., Burkhard, R., Langton, C. and M. Askenazi. The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations, *Working paper 96-06-042*, Santa Fe Institute, Santa Fe, NM, USA, 1996. <<http://www.santafe.edu/projects/swarm/overview.ps>><http://www.swarm.org>.
- [37] Resnick, M. Manual Starlogo , desenvolvido no Media Laboratory MIT, Cambridge e Massachusetts com suporte da Fundação Nacional de Ciência e do grupo Lego.
- [38] <http://groups.yahoo.com/netlogo-users>.
- [39] Netlogo User Manual. <http://ccl.northwestern.edu/netlogo/> Center for connected Learning
- [40] Silva, C. A. Modelagem Comportamental paraAgentes Autônomos em ambientes reais.
- [41] Farias, O. L. M. e LEITE, L. T.. Simulation and Control of Maritime Area through Multi-Agent Systems and Geographical Information Systems. In: - IAWTIC'2005.
- [42] <http://www.ajs.com.br/scaneamento/scann20.htm#1>.
- [43] Noções Básicasde Cartografia IBGE Diretoria de Geociências – DGC/CCAR.

- [44] Menezes, C. S., Pontarolo, E., Azevedo, L. L., Magdalena B. C. Exploração do ambiente orientado a agente Netlogo.
- [45] Bourke, P. PPM / PGM / PBM image files July 1997.
- [46] Moreira, A. M. Favela, um bairro?
- [47] Leite, M. P. *Violência, insegurança e cidadania: reflexões a partir do Rio de Janeiro*, http://www.socialwatch.org/es/informeImpreso/pdfs/panorbrasileiroe2005_bra.pdf, 8 de julho de 2006.

APÊNDICE A – CÓDIGO FONTE

```

;=====Criar Variaveis=====
;
;
Globals [
  ticks
  num_ocup
  num_die
  mapname
  minhalista
]
patches-own [pcolor0 pcolor1 pcolor2 pcolor3 pcolor4 pcolor5 pcolor6 pcolor7 pcolor8 proteamb elevation abandono]
breeds [gente]
;;;
;=====Inicia ao chamar o programa =====
to startup ; slow, do just once
  ;if not file-exists? (borel00class.dat and borel00.dat and borel75.dat and borel04.dat and mapa20.dat and mapa50.dat and
mapa90.dat)
  ;cria_ppm
  ;end
  set num_ocup 0
  lerimagem
end

;=====Setup=====Botão
;=====
to setup
  clear-turtles clear-all-plots
  set ticks 0
  set num_die 0
  setup-agents
  topo
end

to setup-agents
  create-custom-gente pessoas [setup-agent]
  create-custom-gente pessoas1 [setup-agent1]
  create-custom-gente pessoas2 [setup-agent2]
end

to setup-agent
  set size 12
  set color white
  set shape "person"
  while [not na-rua?] [ ; para começar espalhado nas ruas
    setxy random screen-size-x random screen-size-y
  ]
end

to setup-agent1
  set size 12
  set color black
  set shape "person"
  while [not na-rua1?] [ ; para começar espalhado na CHACAR DO CÉU
    setxy random screen-size-x random screen-size-y
  ]
end

to setup-agent2
  set size 12
  set color white
  set shape "person"
  while [not na-rua2?] [ ; para começar espalhado no carrefour
    setxy random screen-size-x random screen-size-y
  ]
end

to-report na-rua?
  report ((abs (pcolor - brown) < 5 ) or (abs (pcolor - blue) < 5))
end

to-report na-rua1?
  report ((abs (pcolor - red) < 5))
end

to-report na-rua2?
  report ((abs (pcolor - orange) < 5))
end

```



```

end
;=====Definir Elevação=====
to topo
ask patches [if (abs (pcolor - lime) < 5) [set proteamb 7]]
ask patches [set elevation 0]
ask patches [ if ((pxcor > -42 and pycor > -146) and (pxcor < -16 and pycor < -110))[set elevation 1]]
ask patches [ if ((pxcor > -17 and pycor > -140) and (pxcor <= 29 and pycor <= -77))[set elevation 1]]
ask patches [ if ((pxcor > 21 and pycor > -77) and (pxcor <= 41 and pycor <= -45))[set elevation 1]]
ask patches [ if ((pxcor > 21 and pycor > -47) and (pxcor <= 49 and pycor <= -16))[set elevation 1]]
ask patches [ if ((pxcor > 37 and pycor > -18) and (pxcor <= 54 and pycor <= 22))[set elevation 1]]
ask patches [ if ((pxcor > 37 and pycor > 22) and (pxcor <= 52 and pycor <= 32))[set elevation 1]]
ask patches [ if ((pxcor > 64 and pycor > 30) and (pxcor <= 86 and pycor <= 45))[set elevation 1]]
ask patches [ if ((pxcor > 83 and pycor > 38) and (pxcor <= 119 and pycor <= 59))[set elevation 1]]
ask patches [ if ((pxcor > 32 and pycor > 70) and (pxcor <= 88 and pycor <= 114))[set elevation 1]]
ask patches [ if ((pxcor > 8 and pycor > -76) and (pxcor <= 23 and pycor <= -13))[set elevation 2]]
ask patches [ if ((pxcor > -25 and pycor > -77) and (pxcor <= 8 and pycor <= -25))[set elevation 2]]
ask patches [ if ((pxcor > 8 and pycor > -73) and (pxcor <= 23 and pycor <= -13))[set elevation 2]]
ask patches [ if ((pxcor > -16 and pycor > -23) and (pxcor <= 23 and pycor <= 24))[set elevation 2]]
ask patches [ if ((pxcor > 20 and pycor > -14) and (pxcor <= 35 and pycor <= 38))[set elevation 2]]
ask patches [ if ( abs (pcolor - yellow) < 5 and elevation = 0)[set elevation 3]]
set minhalista [ 2 3]
ask patches [ if (pcolor = red and elevation != 1)[set elevation random-one-of minhalista]]
ask patches [ if ((pxcor > 90 and pycor > -4) and (pxcor < 127 and pycor < 40))[set abandono 4]]
ask patches [ if ((pxcor > 90 and pycor > -4) and (pxcor < 127 and pycor < 40))[set elevation 0]]
ask patches [ if ((pxcor > 69 and pycor > -137) and (pxcor < 91 and pycor < -101))[set abandono 5]]
ask patches [ if ((pxcor > 80 and pycor > -140) and (pxcor < 110 and pycor < -87))[set abandono 5]]
end
;=====Iniciar Botao =====
to go
if ticks = 0 [if (count turtles) = 0 [setup]]
set ticks ticks + 1
ask turtles [
move
;grafico
]
end
to move
ask gente
[ step
constroi-barraco
]
end
to step ;
set heading 0
rt random 360
lt random 360
fd .1
if ((pcolor = 8 or (abs (pcolor - lime) < 5))) [die]
;if ano = 1920 [
;if( pycor > 56)[die]]
if ano = 1960 [
if((abs (pcolor - lime) < 5 ))[die]]
if ano = 2000 [
if((abs (pcolor - lime) < 5 ))[die]]
end
;=====Montagem Grafico=====
;to grafico
; set-current-plot "settlement"
; set-current-plot-pen "num_ocup"
; plot count patches with [pcolor = 7]
; ;set-current-plot-pen "food-in-pile2"
; ;plot count patches with [pcolor = sky]
; ;set-current-plot-pen "food-in-pile3"
; ;plot count patches with [pcolor = blue]
;end
;=====Atuação dos agentes =====
to constroi-barraco ;
IF ano = 1920[
if ( ( abs (pcolor - yellow) < 5 or abs (pcolor - red) < 5 ) and elevation = 1 and proteamb < 7 and populacao > 4500000)
; ask neighbors [set pcolor random-one-of [5 6 7] set num_ocup num_ocup + .00125 ]]
[ ask neighbors [set pcolor 7 set num_ocup num_ocup + .00125 ]]
IF ano = 1960 [
if ( ( abs (pcolor - yellow) < 5 or abs (pcolor - red) < 5) and elevation = 2 and proteamb < 7 and populacao > 4500000 )

```

```

;[ ask neighbors [set pcolor random-one-of [1 2 3] set num_ocup num_ocup + .00125]]
[ ask neighbors [set pcolor 2 set num_ocup num_ocup + .00125]]
if ( abs (pcolor - yellow) < 5 and abandono = 4 and populacao > 4500000 ); ocupar Indiana
[ ask neighbors [set pcolor 2 set num_ocup num_ocup + .00125]]
IF ano = 2000 [
  if ( ( abs (pcolor - yellow) < 5 or abs (pcolor - red) < 5) and elevation = 3 and proteamb < 7 and populacao > 4500000)
;[ask neighbors [set pcolor random-one-of [73 74] set num_ocup num_ocup + .00125]]
[ask neighbors [set pcolor 74 set num_ocup num_ocup + .00125]]
IF ano = 2000 [
  if ( abandono = 5 and populacao > 4500000 and proteamb < 7); ocupar carrefour
;[ask neighbors [set pcolor random-one-of [85 86] set num_ocup num_ocup + .00125]]
[ask neighbors [set pcolor 86 set num_ocup num_ocup + .00125]]
IF ano = 2000 [
  if ( ( abs (pcolor - blue) < 5 )) ;and elevation = 3 and proteamb < 7 and populacao > 4500000); ocupar indiana II
[ask patches in-radius-nowrap 3 [set pcolor 135 set num_ocup num_ocup + .00125]]

end

;=====Ler Imagem=====
to lerimagem
  set mapname "borel04"; set mapname "xpto"
  import-dat mapname
  ask patches [set pcolor0 pcolor]
  set mapname "borel00"; set mapname "xpto"
  import-dat mapname
  ask patches [set pcolor1 pcolor]
  set mapname "borel75"; set mapname "xpto"
  import-dat mapname
  ask patches [set pcolor2 pcolor]
  set mapname "mapa20"; set mapname "xpto"
  import-dat mapname
  ask patches [set pcolor4 pcolor]
  set mapname "mapa50"; set mapname "xpto"
  import-dat mapname
  ask patches [set pcolor5 pcolor]
  set mapname "mapa90"; set mapname "xpto"
  import-dat mapname
  ask patches [set pcolor6 pcolor]
  set mapname "borel00class"; set mapname "xpto"
  import-dat mapname
  ask patches [set pcolor7 pcolor]
end

;=====Salva Imagem atual=====
to salva
  ask patches [set pcolor8 pcolor]
end

;=====Criar PPM=====
to cria_PPM
  ;set mapname "borel04"; set mapname "xpto"
  ;create-dat mapname
  ;set mapname "mapa20"; set mapname "xpto"
  ;create-dat mapname
  ;set mapname "mapa50"; set mapname "xpto"
  ;create-dat mapname
  ;set mapname "mapa90"; set mapname "xpto"
  ;create-dat mapname
  ;set mapname "borel00class"; set mapname "xpto"
  ;create-dat mapname
  ;set mapname "borel75"; set mapname "xpto"
  ;create-dat mapname
  ;set mapname "borel00"; set mapname "xpto"
  ;create-dat mapname
end

;=====limpa agentes=====
to limpa
  ask turtles [die]
  ;set num_ocup 0
end

;=====Voltar Imagem=====
to image04
  ask patches [set pcolor pcolor0]
  ask turtles [die]
end

```

```

to image75
  ask patches [set pcolor pcolor2]
  ask turtles [die]
end
to image00
  ask patches [set pcolor pcolor1]
  ask turtles [die]
end
to image20
  ask patches [set pcolor pcolor4]
  ask turtles [die]
end
to image50
  ask patches [set pcolor pcolor5]
  ask turtles [die]
end
to image90
  ask patches [set pcolor pcolor6]
  ask turtles [die]
end
to ActualImage
  ask patches [set pcolor pcolor7]
  ask turtles [die]
  set num_ocup 0
end
to retorna
  ask patches [set pcolor pcolor8]
end
;-----
;      Não exclua; usado para montar o mapa. Não usado pelo modelo.
;-----

;=====Cria arquivo dat=====
to create-dat [mapfile]
  print "...criando patches, espere alguns momentos"
  import-ppm mapfile
  export-dat mapfile
end
;=====Importa PPM=====
to import-ppm [ppmfile]
  locals [x y scale]
  set ppmfile ppmfile + ".ppm"
  file-close-all
  file-open ppmfile
  set x 1 set y file-read-line
  while [first file-read-line = "#"] [set x x + 1]
  file-close
  file-open ppmfile
  repeat x [set x file-read-line]
  set x file-read
  set y file-read
  set scale 1 + file-read
  if x != screen-size-x and y != screen-size-y [print "Oops: necessita editar o tamanho da tela para o arquivo ppm"]
  ask paturtles print[ x y]
  ask patches [set pcolor rgb (file-read / 256 ) (file-read / 256 ) (file-read / 256 )]
  file-close
  cleanup-map
end
;=====Exporta arquivo Dat=====
to export-dat [datfile]
  set datfile datfile + ".dat"
  file-close-all
  if file-exists? datfile [file-delete datfile]
  file-open datfile
  ask patches [file-write floor pcolor if pxcor = screen-edge-x [file-print ""]; grava a cor de cada patch no arquivo
                                     ; aberto quando chega ao final do eixo x finaliza
  ask patches [ifelse pxcor = screen-edge-x [file-print floor pcolor][file-write floor pcolor]]
  file-close
end
;=====Importa arquivo Dat=====
to import-dat [datfile]
  ifelse length datfile = 0 [
    (foreach values-from patches [self] [set pcolor-of ?1 ?2])

```

```

]]
  set datfile datfile + ".dat"
  file-open datfile
  ask patches [set pcolor file-read]
  file-close
]
ask patches with [pcolor = 9] [set pcolor 9.9999] ; to save space, 9.9. converted to 9 during export.
end
;=====Limpa Imagem =====
to cleanup-map
  ask patches with [(floor pcolor) mod 10 = 9] [set pcolor 9.9]
  ask patches with [pcolor != 9.9] [set pcolor round pcolor]
  ask patches with [pcolor > 120] [set pcolor pcolor - 110]
end

```