



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS CRATEÚS
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FRANCISCO UÁLISON RODRIGUES DE LIMA

EVOLUINDO REDES NEURAS ARTIFICIAIS PARA A GERAÇÃO DE NPCS
VEROSSÍMEIS UTILIZANDO VISÃO ARTIFICIAL

CRATEÚS

2018

FRANCISCO UÁLISON RODRIGUES DE LIMA

EVOLUINDO REDES NEURAIAS ARTIFICIAIS PARA A GERAÇÃO DE NPCS
VEROSSÍMEIS UTILIZANDO VISÃO ARTIFICIAL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Campus Crateús da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da Computação.

Orientador: Prof. Msc. Arnaldo Barreto
Vila Nova

Coorientadora: Profa. Msc. Lílian de
Oliveira Carneiro

CRATEÚS

2018

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- L698e Lima, Francisco Uálison Rodrigues de.
Evoluindo Redes Neurais Artificiais para geração de NPCs verossímeis utilizando Visão Artificial / Francisco Uálison Rodrigues de Lima. – 2018.
61 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Crateús, Curso de Ciência da Computação, Crateús, 2018.
Orientação: Prof. Me. Arnaldo Barreto Vila Nova.
Coorientação: Profa. Ma. Lilian de Oliveira Carneiro.
1. No Player Character. 2. Comportamentos Autônomos. 3. Córtex Visual Artificial. 4. Rede Neural Artificial. 5. Algoritmo Genético. I. Título.

CDD 004

FRANCISCO UÁLISON RODRIGUES DE LIMA

EVOLUINDO REDES NEURAIAS ARTIFICIAIS PARA A GERAÇÃO DE NPCS
VEROSSÍMEIS UTILIZANDO VISÃO ARTIFICIAL

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus Crateús da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Msc. Arnaldo Barreto Vila Nova (Orientador)
Universidade Federal do Ceará (UFC)

Profa. Msc. Lílian de Oliveira
Carneiro (Coorientadora)
Universidade Federal do Ceará (UFC)

Prof. Msc. Lívio Antônio Melo Freire
Universidade Federal do Ceará (UFC)

Prof. Msc. Ítalo Mendes da Silva Ribeiro
Universidade Federal do Ceará (UFC)

À minha família, por sua capacidade de acreditar em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir.

AGRADECIMENTOS

Primeiramente a Deus por ter me dado saúde e força para superar as dificuldades durante esse 4 anos de graduação.

À minha mãe Vilma Rodrigues, heroína que me deu apoio e incentivo nas horas difíceis de desânimo e cansaço.

Ao meu pai Manoel Alves que diante das dificuldades me serviu de exemplo de persistência e determinação.

Aos meus irmãos Willian Rodrigues, Frank Rodrigues, Vívian Lima e amados sobrinhos Álec e Pedro que sempre me serviram de incentivo e inspiração para continuar e não desistir.

Ao amigo da família Sgt. Lima (*In Memoriam*) que no meu primeiro dia de aula, ainda sem ter onde morar em Crateús, me acolheu para que pudesse dar início ao sonho de cursar Ciência da Computação. Onde estiver, eu deixo meu eterno obrigado.

Aos meus amigos Bruno Sousa, Wislla Nuânska e Mardônio Vieira, companheiros de trabalhos e irmãos na amizade que fizeram parte da minha formação e que vão continuar presentes em minha vida com certeza.

Ao meu orientador Prof. Arnaldo Barreto, pelo empenho dedicado à elaboração deste trabalho.

À minha querida coorientadora Profa. Lílian Carneiro pela sua valiosa contribuição para que este trabalho ganhasse vida.

A Universidade Federal do Ceará, pela oportunidade de fazer o curso que sempre sonhei e pelo suporte durante esses anos, sem o qual não teria chegado até aqui.

Aos professores, não somente por terem me ensinado, mas por terem me feito aprender. A palavra mestre nunca fará justiça aos professores dedicados aos quais sem nominar terão os meus eternos agradecimentos.

À servidora Ana Célia, por sempre estar presente no nosso dia-a-dia, conversando, dando apoio e conselhos frente as dificuldades enfrentadas nesses 4 anos.

À todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

“Ser capaz de perseguir nossos próprios sonhos
é o que nos faz fortes.”

(Chester Bennington)

RESUMO

Este trabalho aborda a geração de comportamentos autônomos em No Player Characters utilizando visão artificial. Para isso, são propostas variações com diferentes resoluções de um Córtex Visual Artificial, representado por uma Rede Neural Artificial que recebe as informações visuais (*pixels*) e as transforma em sinais para os motores de movimentação do NPC. Durante o treinamento, um processo evolutivo utilizando Algoritmo Genético é aplicado a Rede Neural com a finalidade ensiná-la a reconhecer os padrões de *pixels* vindos de uma câmera virtual. À medida que novas gerações de treino vão sendo criadas, espera-se que o NPC consiga identificar o *Player* no ambiente e siga até ele, sem qualquer descrição sensorial prévia, utilizando apenas suas percepções visuais. Os resultados obtidos mostram que, em determinados cenários, a Rede Neural conseguiu evoluir utilizando apenas o sensoramento visual do ambiente, gerando comportamentos satisfatórios na tarefa de guiar o NPC até o *Player*.

Palavras-chave: No Player Character. Comportamentos Autônomos. Córtex Visual Artificial. Rede Neural Artificial. Algoritmo Genético.

ABSTRACT

This work approaches the generation of autonomous behaviors in No Player Characters using artificial vision. For this, variations with different resolutions of an Artificial Visual Cortex are proposed, represented by an Artificial Neural Network that receives the visual information (pixels) and turns them into signals for the NPC movement engines. During training, an evolutionary process using Genetic Algorithm is applied to Neural Network in order to teach it to recognize the patterns of pixels coming from a virtual camera. As new training generations are being created, the NPC is expected to be able to identify the *Player* in environment and follow it without any previous sensory description using only their visual perceptions. The results show that, in experiment scenarios, the Neural Network was able to evolve using only the visual sensing of the environment, generating the satisfactory behavior in the task of finding the *Player*.

Keywords: No Player Character. Autonomous Behaviors. Artificial Visual Cortex. Artificial Neural Network. Genetic Algorithm.

LISTA DE FIGURAS

Figura 1 – Modelo matemático simples de um neurônio. A ativação de saída da unidade é a_j , em que a_i é a ativação de saída da unidade i e $w_{i,j}$ é o peso sobre a ligação da unidade i com essa unidade.	20
Figura 2 – Representação de um cromossomo.	21
Figura 3 – Representação básica de um Algoritmo Genético	21
Figura 4 – Visão geral do modelo proposto.	24
Figura 5 – Locomoção de um agente guiado por sua visão. Na parte superior esquerda de cada quadro da imagem é exibido o campo visual do agente selecionado pela <i>bounding box</i>	26
Figura 6 – Comportamento de Navegação. Os casos A e B são compostos por quatro quadros que mostram a trajetória do agente com cérebro em evolução. (marcado com o ponto vermelho)	27
Figura 7 – O robô. (a) Robô no ambiente. A caixa preta é, ao mesmo tempo, seu olho e sua boca. (b) Os três sensores distribuídos no olho e os respectivos campos de sensibilidade, com ângulo de 20° cada.	28
Figura 8 – Comportamento do robô equipado com um CVA. As linhas pontilhadas mostram o posicionamento de frutas e obstáculos pelo ambiente que são visualizados pelo sensor visual. No quadro 32, a linha pontilhada indica o caminho percorrido pelo robô.	29
Figura 9 – Exemplo de um genótipo e seu fenótipo (rede neural).	30
Figura 10 – Exemplo de mutação estrutural com a adição de uma nova conexão e um novo neurônio	31
Figura 11 – Operador de cruzamento (<i>crossover</i>)	32
Figura 12 – Exemplo de uma topologia mínima de entrada	34
Figura 13 – O NPC possuíra uma câmera acoplada na parte correspondente a visão do modelo utilizado (<i>Ethan</i>), a câmera fornecerá informações visuais sobre o cenário para uma RNA, que será responsável por ativar os motores de movimentação.	35
Figura 14 – CVAs propostos	36
Figura 15 – Cenário visto de cima	38
Figura 16 – Cubo na cor laranja representando os obstáculos no cenário.	38

Figura 17 – Modelo <i>Ethan</i> - O modelo A representa o NPC na cor vermelha (em RGB: 255, 0, 0) e o modelo B representa o <i>Player</i> (Objetivo) na cor azul (em RGB: 0, 0, 255).	38
Figura 18 – Cenário 1 (20x20) - A movimentação em cada quadro se refere aos comportamentos gerados pelo NPC após o treinamento para o CVA de 20x20. O traço lilás indica o caminho percorrido por ele.	40
Figura 19 – Detalhamento do caminho feito pelo NPC no quadro 3 da Figura 18, com as visões em 5 instantes diferentes de simulação	41
Figura 20 – Detalhamento do caminho feito pelo NPC no quadro 1 da Figura 21, com as visões em 5 instantes diferentes da simulação.	42
Figura 21 – Cenário 1 (50x50) - A movimentação em cada quadro se refere aos comportamentos gerados pelo NPC após o treinamento para o CVA de 50x50. O traço lilás indica o caminho percorrido por ele até o <i>Player</i>	43
Figura 22 – Gráfico que compara a quantidade de NPCs que conseguiram chegar ao <i>Player</i> em cada geração para cada tipo de CVA.	44
Figura 23 – Gráfico que compara a média do <i>fitness</i> obtido por cada geração em diferentes CVAs.	45
Figura 24 – Cenário 2 (20x20) - A movimentação em cada quadro se refere aos comportamentos gerados do NPC após o treinamento para o CVA de 20x20. O traço lilás indica o caminho percorrido por ele.	47
Figura 25 – Detalhamento do caminho feito pelo NPC no quadro 3 da Figura 24, com as visões em 5 instantes diferentes da simulação.	48
Figura 26 – Cenário 2 (50x50) - A movimentação em cada quadro se refere aos comportamentos gerados do NPC após o treinamento para o CVA de 50x50. O traço lilás indica o caminho percorrido por ele.	49
Figura 27 – Detalhamento do caminho feito pelo NPC no quadro 3 da Figura 26, com as visões em 5 instantes diferentes da simulação.	50
Figura 28 – Gráfico que compara a quantidade de NPCs que conseguiram chegar ao <i>Player</i> em cada geração para cada tipo de CVA.	51
Figura 29 – Gráfico que compara a média do <i>fitness</i> obtido por cada geração em diferentes CVAs.	51

Figura 30 – Cenário 3 (20x20) - A movimentação em cada quadro se refere a aos comportamentos gerados do NPC após o treinamento para o CVA de 20x20. O traço lilás indica o caminho percorrido por ele até o <i>Player</i>	53
Figura 31 – Detalhamento do caminho feito pelo NPC no quadro 3 da Figura 30, com as visões em 5 instantes diferentes da simulação.	54
Figura 32 – Cenário 3 (50x50) - A movimentação em cada quadro se refere aos comportamentos gerados do NPC após o treinamento para o CVA de 50x50. O traço lilás indica o caminho percorrido por ele.	55
Figura 33 – Gráfico que compara a quantidade de NPCs que conseguiram chegar ao <i>Player</i> em cada geração para cada tipo de CVA.	56
Figura 34 – Gráfico que compara a média do <i>fitness</i> obtido por cada geração em diferentes CVAs.	57

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmo Genético
CVA	Córtex Visual Artificial
IA	Inteligência Artificial
NEAT	<i>NeuroEvolution of Augmenting Topologies</i>
NPC	<i>No Player Character</i>
RGB	<i>Red Green Blue</i>
RL	<i>Reinforcement Learning</i>
RNA	Rede Neural Artificial

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Contextualização	15
1.2	Justificativa	16
1.3	Objetivos Gerais	17
1.4	Objetivos Específicos	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Trabalhos Correlatos	23
2.1.1	<i>Geração de comportamentos emergentes de navegação em agentes autônomos utilizando visão artificial</i>	23
2.1.2	<i>Integração mente e ambiente para geração de comportamentos emergentes em personagens virtuais autônomos através da evolução de redes neurais artificiais</i>	25
2.2	NEAT - NeuroEvolution of Augmenting Topologies	30
2.2.1	<i>Codificação genética</i>	30
2.2.2	<i>Especiação ou Proteção de Inovação</i>	33
2.2.3	<i>Seleção</i>	33
3	PROPOSTA	35
3.1	O Córtex Visual Artificial	35
3.2	O Controlador	36
3.2.1	<i>A RNA e o Processo Evolutivo</i>	36
3.2.2	<i>Função de Avaliação</i>	36
3.3	<i>A Game Engine</i>	37
3.4	Personagens, Obstáculos e Cenário	37
4	RESULTADOS	39
4.1	Resultados do Cenário 1	39
4.1.1	<i>CVA com 20x20 pixels - 1.200 entradas para a RNA</i>	39
4.1.2	<i>CVA com 50x50 pixels - 7.500 entradas para a RNA</i>	42
4.1.3	<i>Evolução em cada tipo de Córtex Visual Artificial (CVA)</i>	44
4.2	Resultados do Cenário 2	46
4.2.1	<i>CVA com 20x20 pixels - 1.200 entradas para a RNA</i>	46

4.2.2	<i>CVA com 50x50 pixels - 7.500 entradas para a RNA</i>	48
4.2.3	<i>Evolução em cada tipo de CVA</i>	50
4.3	Resultados do Cenário 3	52
4.3.1	<i>CVA com 20x20 pixels - 1.200 entradas para a RNA</i>	52
4.3.2	<i>CVA com 50x50 pixels - 7.500 entradas para a RNA</i>	54
4.3.3	<i>Evolução em cada tipo de CVA</i>	56
5	CONCLUSÕES E TRABALHOS FUTUROS	58
5.1	Conclusões	58
5.2	Trabalhos Futuros	59
	REFERÊNCIAS	60

1 INTRODUÇÃO

1.1 Contextualização

Jogos eletrônicos estão cada vez mais presentes na vida das pessoas, sendo utilizados para fins de entretenimento ou criação de ambiente de simulação. A partir de grandes orçamentos, enormes mundos ou cenários de jogos podem ser criados com gráficos realistas e narrativas com roteiros cada vez mais complexos. Para dar vida a esses mundos é preciso que neles existam vários mecanismos inteligentes, visando simular para o jogador os aspectos do mundo real. Geralmente, tais mundos são povoados por *No Player Character* (NPC), termo designado para personagens controlados pelo computador.

Existem duas abordagens de técnicas de Inteligência Artificial (IA) para NPCs: as que são de natureza determinística, onde os comportamentos são previsíveis; e as de natureza não determinística, que possuem um alto grau de incerteza. As técnicas determinísticas mantêm o foco em diminuir a imprevisibilidade, fazendo com que os programadores tratem todas as possíveis cenários de interação do NPC com seu ambiente e com o jogador. De forma oposta, temos as abordagens de natureza não determinística, as quais dependem das técnicas de IA empregadas para gerar os comportamentos, sem uma especificação prévia. Alguns exemplos de técnicas mais comuns nesta abordagem são as de Rede Neural Artificial (RNA), Técnicas Bayesianas e Algoritmo Genético (AG) (BOURG; SEEMANN, 2004).

Os NPCs possuem um papel fundamental quando se avalia as formas de narrar uma história em um jogo, formando uma parte do contexto da narrativa no qual o seu comportamento *in game* definirá uma interação mais realista e imersiva com o jogador. Nesse contexto, a busca por mais realismo e imersão deve ser um fator a se considerar quando desejamos que NPCs possuam um comportamento mais natural e condizente com a realidade proposta pela narrativa. Assim, é necessário utilizar técnicas de IA que possibilitem aos NPCs possuir o comportamento desejado.

Na IA um agente é visto como tudo que pode perceber seu ambiente através de sensores e atuar sobre esse ambiente a partir de seus atuadores (RUSSELL; NORVIG, 2004); em jogos, um agente pode ser visto como um NPC. Na literatura tem-se o conceito de agente verossímil (do inglês, *Believable Agents*), que consiste em um agente inteligente, que dotado de personalidade e autonomia apresenta comportamentos com as propriedades dos personagens das artes (LOYALL, 1997). Os Agentes verossímeis podem ser vistos como agentes com comportamentos

autônomos, desempenhando um papel fundamental na narrativa, no qual seu comportamento está constantemente sendo avaliado pelo espectador, que busca nos comportamentos uma a noção de personagem crível, cessando a descrença de que o personagem não seja real (LOYALL, 1997).

Com isso, a discussão é guiada para dar mais autonomia ao comportamento do NPC, baseando seu aprendizado a partir de percepções do ambiente, visando criar um comportamento mais humano. Pode-se até considerar a aplicação de modelos que incorporam melhorias emocionais a eles, como raiva, tristeza, ansiedade e medo, buscando um comportamento mais humano no gerenciamento de crises (LOIZOU *et al.*, 2012). É preciso buscar formas de dar ao NPC a autonomia necessária para seu aprendizado. Tendo em vista isso, uma característica básica para se obter autonomia em agentes é o estudo de comportamentos emergentes. Na literatura, técnicas que utilizam RNAs e AGs são utilizadas para geração de tais comportamentos em agentes autônomos (CARNEIRO *et al.*, 2014) (NOGUEIRA *et al.*, 2013). Para (NOGUEIRA, 2014) a definição de comportamento emergente é a seguinte:

o comportamento de um personagem virtual é chamado emergente quando não está explicitamente descrito em qualquer dos componentes do sistema, surgindo como resultado das interações dinâmicas dos componentes, bem como de suas propriedades individuais específicas. [...]

Dessa forma, buscar comportamentos emergentes pode ser visto como essencial para a criação de NPCs autônomos. Na prática, o agente é programado apenas com sua habilidade de aprender. Assim, quando possuir experiência suficiente sobre o ambiente, poderá considerar seu comportamento independente de ações pré-programadas (RUSSELL; NORVIG, 2004). O sensoriamento do ambiente não onisciente é a chave para fugir de comportamentos inadequados apresentado por técnicas determinísticas na IA para jogos.

Na indústria de jogos ainda há uma forte resistência em aplicar aprendizado em tempo real para NPCs, visto que há uma certa imprevisibilidade sobre o que aprenderão e elevado custo computacional relacionados aos treinamentos. Por outro lado, NPCs que possam aprender e se desenvolver a partir de suas interações com o ambiente, é uma área de grande crescimento e estudo na comunidade acadêmica (YANNAKAKIS; TOGELIUS, 2017).

1.2 Justificativa

Ao se pensar em técnicas de IA para jogos, deve-se considerar o comportamento de NPCs para que se alcance uma interação mais realista com os jogadores (QUANDT; KRÖGER, 2013). Neste contexto, a utilização de personagens autônomos em jogos proporciona uma interação mais realista, influenciando na implementação de uma estrutura dramática (GOMES

et al., 2008). Para isso, deve-se aplicar técnicas para que NPCs apresentem um maior grau de autonomia, exibindo um comportamento baseado nas suas percepções do ambiente.

Uma das formas de simular este comportamento é a partir do uso de um Córtex Visual Artificial (CVA) para personagens autônomos. Basicamente, um CVA é formado por um cérebro (RNAs e AGs) que simulam os comportamentos e um sensoriamento visual (Câmera Virtual) que fornece as informações do ambiente para o cérebro (CARNEIRO *et al.*, 2014). Neste trabalho, esta abordagem será utilizada para criação de NPCs autônomos, que consigam aprender a partir de interações feitas com o ambiente, visando gerar comportamentos mais realistas e imersivos na narrativa proposta pelo jogo, bem como discutir a aplicabilidade de Visão Artificial para criação de NPCs.

1.3 Objetivos Gerais

O principal objetivo deste trabalho é criar NPCs autônomos guiados por percepções visuais para alcançar um determinado objeto, que representará o jogador em cena. Com isso, espera-se comportamentos mais naturais que possibilitem uma maior imersão e realismo das interações dos NPCs com o ambiente. No contexto de desenvolvimento, espera-se uma diminuição do tempo de criação dos NPCs, já que não será necessária a programação de comportamentos determinísticos para eles.

1.4 Objetivos Específicos

São destacados como objetivos específicos deste trabalho:

- (a) Utilizar Redes Neurais Artificiais para a criação de um Córtex Visual Artificial para o aprendizado de NPCs;
- (b) Treinar o NPC para que encontre um jogador estático apenas utilizando suas percepções visuais e siga até ele;
- (c) Avaliar a utilização de NPCs autônomos que consigam aprender a partir do ambiente;
- (d) Comparar as diferenças de desempenho e comportamentos de Redes Neurais Artificiais treinadas com grandes quantidades de entrada;

2 FUNDAMENTAÇÃO TEÓRICA

Muito antes da IA ser considerada um campo de pesquisa em Ciência da Computação, alguns pesquisadores já se questionavam se máquinas poderiam pensar como humanos. Em 1950, Alan Turing, no artigo intitulado *Computing Machinery and Intelligence* (TURING, 2009) propôs um teste chamado de jogo da imitação, sendo conhecido mais tarde como teste de Turing. A interpretação padrão do teste consiste em: dado um jogador C, o interrogador, é encarregado de descobrir, dentre os jogadores A e B, qual é um humano e qual é um computador, usando perguntas escritas feitas para A e B (SAYGIN *et al.*, 2000). O teste de Turing procurava demonstrar se computadores poderiam comportar-se como humanos. Porém, muitos pesquisadores deram pouca atenção às tentativas de passar no teste, pois criar simulações semelhantes ao comportamento humano não fazia parte dos objetivos das pesquisas em IA na época. Contudo, um computador que consiga simular o comportamento humano pode ser interessante em trabalhos envolvendo artes, jogos digitais e simulações em Realidade Virtual.

Na época do teste de Turing, jogos digitais eram praticamente inexistentes. Logo, era difícil imaginar uma IA aplicada para jogos, pois a própria IA era um campo de estudo ainda prematuro. Com o passar dos anos, em 1961, surge o jogo *Space War*, totalmente programado em *Assembly*, criado por estudantes do *Massachusetts Institute of Technology* para a apresentação do computador DEC PDP-1 (MIT, 2011). Este pode ser considerado o primeiro jogo interativo de computador. Mais tarde, em 1972, viria a surgir o primeiro console de videogame da história, o Atari, junto com o primeiro jogo comercial chamado *Pong*. Por causa das limitações de *hardware* e *software* da época, tanto o *Space War* quanto o *Pong* dispunham de uma IA bem básica, muitas vezes quase inexistente. As técnicas aplicadas nestes dois jogos eram bastante limitadas e totalmente determinísticas.

Na década de 90, à medida que novos consoles surgiam e os computadores se modernizavam, o campo de pesquisa em IA também crescia. Em 1995 surgiu o conceito de agentes inteligentes, motivado pela busca de soluções para subproblemas da IA (RUSSELL; NORVIG, 2004). Agentes podem ser utilizados para resolver diversos problemas envolvendo jogos, em que os NPCs possuem características de agentes no ambiente de jogo, necessitando de alguma “inteligência” para resolver determinados objetivos. Na literatura existem trabalhos que abordam e discutem sobre os chamados agentes verossímeis, ou agentes críveis, como na tese de Loyall (1997), em que é apresentado as seguintes características básicas para que um agente possa ser considerado verossímil:

- (a) Personalidade;
- (b) Emoção;
- (c) Motivação Própria;
- (d) Relacionamentos Sociais;
- (e) Consistência;
- (f) Ilusão de Vida.

Loyall (1997) também apresenta uma linguagem que buscava expressar os comportamentos acima, de forma que os personagens virtuais apresentem comportamentos tão detalhados quanto aos dos personagens de artes tradicionais, como cinema e teatro. Os agentes verossímeis possuem uma alta aplicabilidade em jogos digitais, pois possuem as características necessárias para a criação de narrativas mais imersivas e convincentes, a partir de aprendizado e comportamentos autônomos (GOMES *et al.*, 2008).

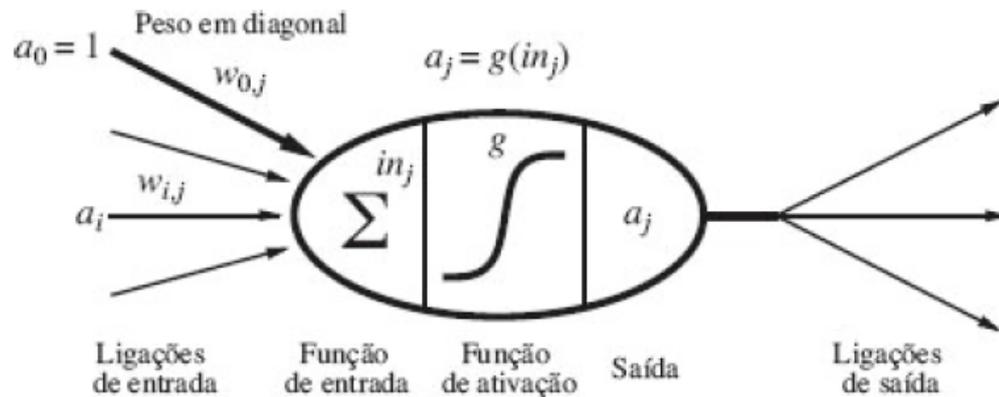
Na criação de cenários que expliquem a narrativa proposta pelo jogo de forma mais realista e convincente, a IA possui um fator de peso (BARTLE, 2004). Inicialmente, para ser considerado um agente inteligente, o NPC precisa exibir características básicas como autonomia e comportamentos orientados para um objetivo (YILDIRIM; STENE, 2008). Um NPC que aprende pode ser visto como um agente autônomo que busca adquirir um comportamento satisfatório a partir de sua interação com o ambiente (FENG; TAN, 2016). O teste de Turing tentava responder se computadores poderiam pensar como humanos. Uma analogia semelhante pode ser feita para que NPCs consigam exibir tais comportamentos, assim, garantindo uma experiência de jogo que seja mais condizente com a realidade (CHANG *et al.*, 2011).

Na narrativa, a busca por uma jogabilidade mais atraente está relacionada com uma maior autonomia e inteligência dos NPCs. Segundo (GOMES *et al.*, 2008) a narrativa nos jogos pode ser vista como: “[...] uma ‘simulação imersiva’ a ser experimentada pelo *iterator* através de seu percurso pelo espaço virtual do *game* no papel de um personagem e de sua interação com personagens autônomos[...]”. Assim, deve-se aplicar em jogos técnicas de IA relacionadas a agentes, com o objetivo de criar os comportamentos desejados (DIGNUM *et al.*, 2009). Para os NPCs serem considerados autônomos, as técnicas utilizadas precisam permitir o aprendizado do ambiente sem instruções do programador, dando ao NPC apenas a habilidade de aprender.

Na aplicação de técnicas para gerar comportamentos autônomos em NPCs, as RNAs podem ser utilizadas, pois tais redes são modeladas de forma similar ao cérebro humano, sendo compostas por seus neurônios artificiais (COPPIN, 2013). Cada neurônio é formado por um nú-

mero de unidades de processamento simples que tendem a armazenar conhecimento, permitindo aprendizagem, generalização e processos interativos com o ambiente externo (REZENDE, 2003) (HAYKIN, 2007). A Figura 1 mostra a representação básica de um neurônio artificial.

Figura 1 – Modelo matemático simples de um neurônio. A ativação de saída da unidade é a_j , em que a_i é a ativação de saída da unidade i e $w_{i,j}$ é o peso sobre a ligação da unidade i com essa unidade.



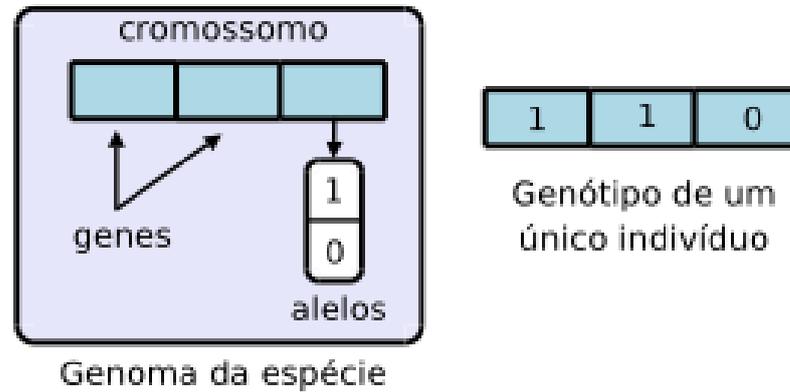
Fonte: (NORVIG; RUSSELL, 2014)

Uma RNA é composta por milhares de neurônios artificiais, que conectados entre si, formam a topologia da rede, geralmente construída de acordo com os requisitos necessários para o problema no qual a rede será utilizada. É importante que redes neurais se aprimorem à medida que vão recebendo novas informações. Uma das formas de se fazer isso é com a utilização de algoritmos evolucionários, como AGs, que evoluem a rede através de cruzamentos e mutações, conceitos que fazem parte de uma área de estudos chamada Neuroevolução. Basicamente, a neuroevolução consiste na aplicação de AGs para otimizar os pesos nas RNAs, proporcionando um melhoramento dos resultados (SILVA, 2015).

Um AG basicamente consiste em um conjunto de parâmetros chamados de genomas ou cromossomos. Os cromossomos são compostos por genes que são responsáveis por guardar as características de representação de um organismo, chamadas de fenótipos. Cada gene pode assumir um entre diversos estados (alelos) e uma configuração particular de cromossomos é chamada de genótipo (MIGUEL, 2009). A Figura 2 mostra a representação comum de um cromossomo.

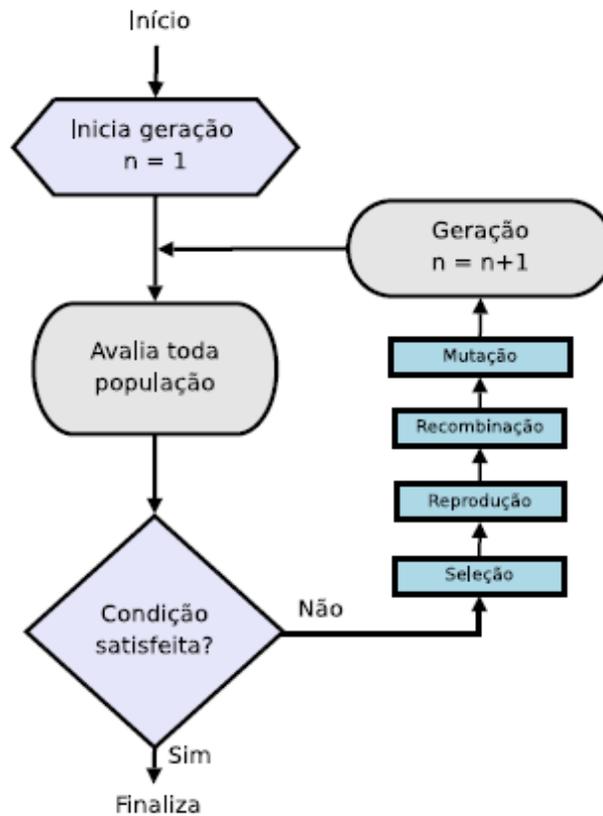
A Figura 3 mostra a estrutura básica de um AG. Existe um laço que percorre toda a geração avaliando todos os indivíduos. Após isso ocorre uma seleção de alguns para reproduzir e gerar a próxima população da geração seguinte. Neste processo também ocorrem outras fases, como seleção, recombinação e mutação.

Figura 2 – Representação de um cromossomo.



Fonte: (MIGUEL, 2009)

Figura 3 – Representação básica de um Algoritmo Genético



Fonte: (MIGUEL, 2009)

Uma das principais questões envolvendo AGs está em como tratar cada um dos indivíduos, pois deve-se avaliar quais são os critérios mais adequados para selecioná-los, procurando meios efetivos de medir a qualidade de cada um através de uma pontuação adaptativa (do inglês: *fitness*).

A utilização de RNAs para criar NPCs autônomos podem ser vistas no processo de

decisão em *Rolling Playing Games* (RPG), em que o NPC pode decidir em atacar o oponente ou não (MADSEN; ADAMATTI, 2013). As RNAs precisam ser alimentadas com informações que possibilitem aos NPCs maneiras de identificar elementos no ambiente, servindo como aprendizado para ações que os direcionam para os objetivos no jogo. Um objetivo poderia ser a identificação do jogador a partir das percepções visuais do ambiente. Este caso pode ser visto no trabalho de (CARNEIRO *et al.*, 2014) que apresenta um CVA utilizando RNA e AG, em que um agente consegue navegar por um ambiente, chegar até seu objetivo sem qualquer informação prévia, apenas utilizando suas percepções visuais; e em (NOGUEIRA, 2014), em que um CVA fornece informações do ambiente para uma RNA, possibilitando um melhoramento nos controles de movimentação de um personagem virtual. Essas abordagens mostram que é possível criar uma simulação de comportamentos emergentes de agentes autônomos em realidade virtual, utilizando-se de percepções visuais do ambiente, podendo ser aplicada no desenvolvimento de NPCs autônomos em jogos digitais. Segundo (BOURG; SEEMANN, 2004):

“[...] em jogos, as redes neurais oferecem algumas vantagens importantes sobre as técnicas mais tradicionais de IA. Primeiro, a utilização de uma rede neural permite que os desenvolvedores de jogos simplifiquem a codificação de máquinas de estado complexas ou baseadas em regras sistêmicas, direcionando os principais processos de tomada de decisão a uma ou mais redes neurais treinadas. Segundo, redes neurais oferecem o potencial para a IA do jogo se adaptar à medida que o jogo é jogado[...]”.

Nos dias atuais, técnicas avançadas de *Machine Learning*¹ são utilizadas para treinamento de NPCs, tendo como base diversas variações de RNAs. Um exemplo de estudo na área é o grupo de desenvolvimento da *Game Engine*² *Unity 3D*, que atualmente desenvolve uma *framework* focada no aprendizado e treinamento de NPCs em jogos chamada de *ML-Agents* (UNITY, 2018). Outra abordagem estudada pela *Unity* é o *Reinforcement Learning* (RL), ou Aprendizado por Reforço, que consiste em buscar formas para que agentes possam realizar ações em um ambiente, de modo a maximizar uma ideia de recompensa acumulativa. (UNITY, 2017).

As técnicas de RL estão em alta, principalmente quando aplicadas na criação de *bots*³ ou NPCs utilizando as mesmas informações e interações disponíveis para os jogadores no aprendizado. A empresa britânica *DeepMind* já vem aplicando RL com informações visuais em

¹ *Machine Learning*: área de estudo em Inteligência Artificial que aplica técnicas para que sistemas consigam aprender com dados, identificar padrões e tomar decisões com o mínimo de intervenção humana.

² *Game Engine*: programa ou conjunto de bibliotecas utilizado para simplificar e abstrair o desenvolvimento de jogos.

³ *bots*: jogador simulado por software, que dotado de uma Inteligência Artificial consegue jogar de forma similar ou até melhor que um jogador humano.

diversos experimentos, como na geração de *bots* que conseguem jogar jogos do antigo console Atari (MIND, 2013). O RL também pode ser aplicada em ambientes 3D, como no caso da plataforma *VizDoom*, baseada no clássico jogo de tiro em primeira pessoa *Doom*, que permite a criação de *bots* que conseguem jogar apenas com informações visuais. (KEMPKA *et al.*, 2016).

Existem outras *frameworks* que utilizam RL aplicado à jogos, como a *Gym*, que tem a finalidade de desenvolver e comparar algoritmos de RL, apresentando agentes baseados em aprendizado que conseguem jogar jogos como *Pong* e *Pinball* (OPENAI, 2016a). Baseado no *Gym*, existe também o *Universe*, uma *framework* que utiliza RL para criação de *bots* inteligentes que conseguem jogar jogos de Atari apenas com as informações dos *pixels* da tela (OPENAI, 2016b).

Como visto, a aplicação de RL em RNAs para geração de agentes ou *bots* autônomos em jogos é uma tendência na literatura. Apesar disto, a técnica proposta por este trabalho não utilizará RL. A busca por comportamentos autônomos é realizada utilizando a evolução de RNAs com AGs, baseadas nas abordagens apresentadas em (CARNEIRO *et al.*, 2014) e (NOGUEIRA, 2014).

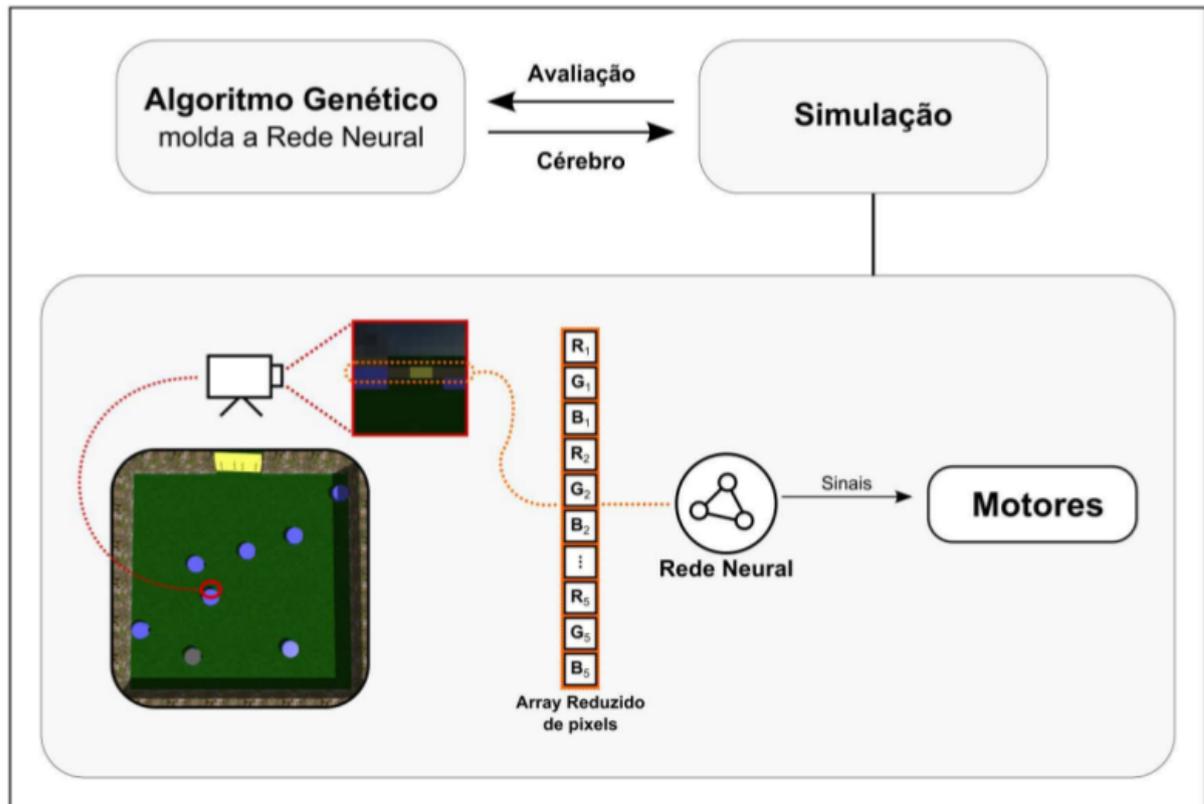
2.1 Trabalhos Correlatos

2.1.1 Geração de comportamentos emergentes de navegação em agentes autônomos utilizando visão artificial

Carneiro *et al.* (2014) apresenta uma técnica para que agentes possam se guiar pelo ambiente a partir de suas percepções visuais. Para isso, utiliza-se um CVA com uma câmera virtual acoplada ao agente, com um tamanho de quadro de 10x10 *pixels*. Cada *pixel* é representado no sistema de cores *Red Green Blue* (RGB), em que apenas a média aritmética dos 10 *pixels* da quinta linha são utilizados como visão, assim reduzidos a 5 *pixels*, que são repassados como entrada de uma RNA que emite sinais para o motores de movimentação do agente. A Figura 4 mostra a visão geral da proposta de Carneiro.

A média dos *pixels* foi utilizada para que o processamento da RNA não fosse tão pesado, visto que, com 5 *pixels*, já tem-se 15 entradas. Para criar o controlador do agente, Carneiro utiliza a mesma RNA e AG utilizado por (NOGUEIRA *et al.*, 2013 apud CARNEIRO *et al.*, 2014), porém com uma nova descrição sensorial, visando evoluir a rede e gerar comportamentos naturais que possam emergir a partir de um controlador dinâmico que baseia-se nas interações

Figura 4 – Visão geral do modelo proposto.



Fonte: (CARNEIRO *et al.*, 2014)

do corpo do agente com o ambiente.

O controlador do agente deve possuir a capacidade de se adaptar a partir de sucessivas interações com o ambiente. Para conseguir este objetivo, utiliza-se de uma codificação genética para evoluir uma Rede Neural Recorrente de Tempo Contínuo (BEER, 1995 apud CARNEIRO *et al.*, 2014), que possui dois tipos de neurônios: neurônios padrões, que formam as unidades de processamento; e neurônios moduladores, responsáveis pelas mudanças dos pesos das sinapses. Esta rede é responsável por gerar sinais que servirão para controlar os movimentos do agente baseando-se nas suas informações sensoriais.

A codificação genética é feita por uma versão simplificada de um *Analog Genetic Encog* (MATTIUSI; FLOREANO, 2007 apud CARNEIRO *et al.*, 2014), em que cada indivíduo na população possui dois cromossomos: um que guarda os parâmetros globais da rede e outro para manter a própria rede. Assim, a leitura de cada cromossomo fornece uma lista de neurônios com seus parâmetros e terminais que irão moldar a rede. Um AG necessita de uma função de aptidão ou avaliação que define o comportamento do agente, mas não diz como este deve ser, ou seja, não diz como deve ser seus movimentos. Os movimentos são obtidos a partir das interações do personagem com o ambiente, apenas ficando a cargo do programador fornecer a capacidade

do agente de aprender, ou seja, determinando a função de avaliação para esta finalidade.

A função de avaliação f_{av} proposta para o processo de adaptação e geração do comportamento é a seguinte:

$$f_{av} = \frac{1}{t} + \frac{1}{d}, \quad (2.1)$$

em que t é o tempo que um agente leva para chegar até a porta e d é a distância entre o agente e a porta. Assim, o agente com melhor avaliação será aquele que mais se aproximou da porta no menor tempo possível.

Nos testes, os cenários utilizados para simulação consistem em uma área quadrada delimitada por paredes com uma única porta. A porta possui uma cor de destaque para se diferenciar das paredes e dos outros agentes obstáculos. A posição da porta é definida aleatoriamente para evitar que o agente memorize a sua posição. Se a porta fosse fixa, os indivíduos poderiam se comportar movendo-se apenas para uma direção. A rede neural apresentada conseguiu evoluir e gerar os comportamentos de navegação satisfatórios pelos diferentes cenários utilizados, selecionando os indivíduos com melhor avaliação após 65 gerações. As Figuras 5 e 6 mostram que os resultados da abordagem apresentada para navegação do agente foi satisfatória.

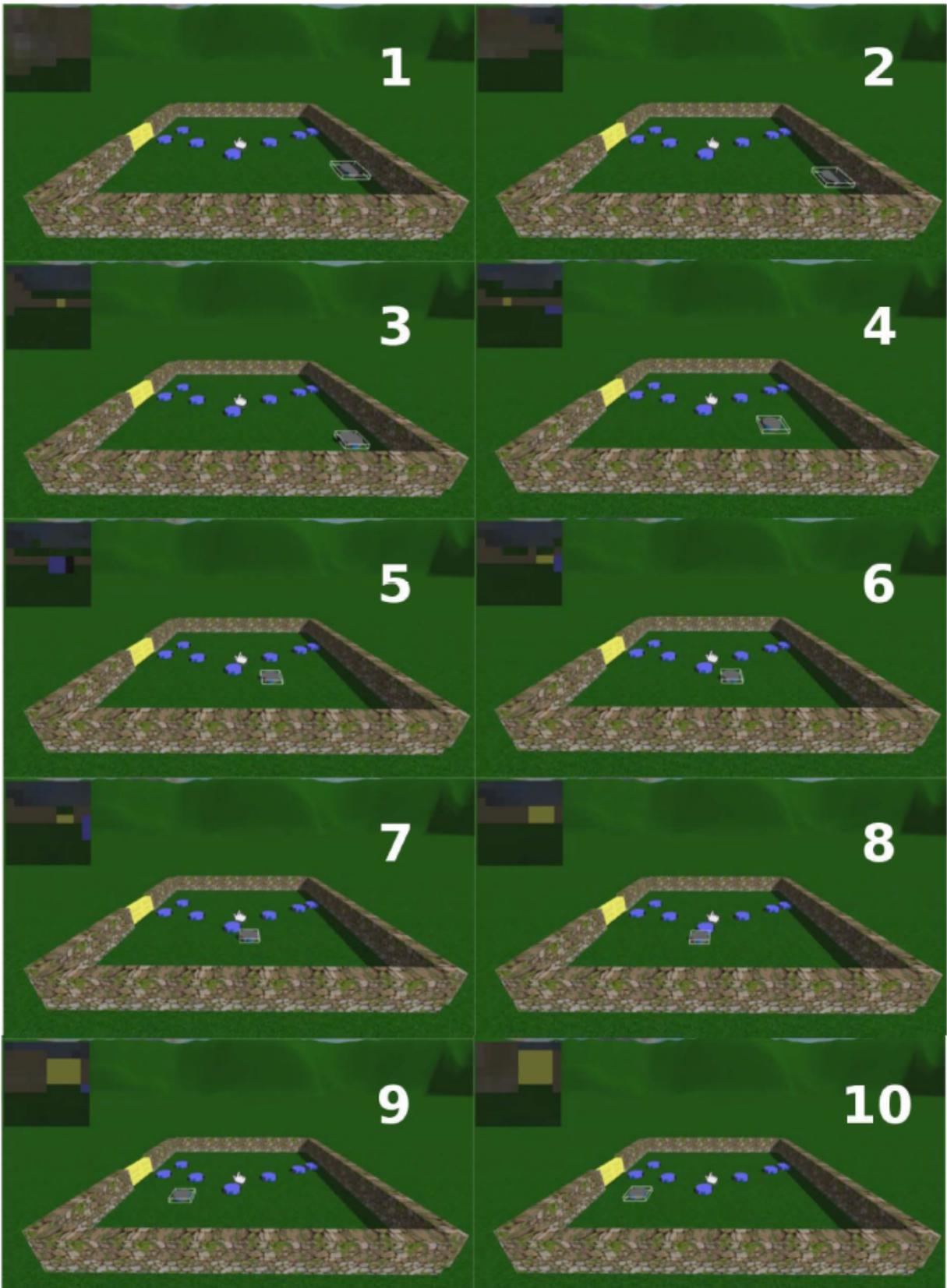
2.1.2 Integração mente e ambiente para geração de comportamentos emergentes em personagens virtuais autônomos através da evolução de redes neurais artificiais

Em outro trabalho aplicando RNAs e AG para evoluir agentes autônomos, Nogueira (2014) discute o senso de imersão de usuários em ambientes virtuais. Para alcançar tal imersão, deve-se obter comportamentos adequados de personagens virtuais, ou seja, comportamentos que condizem com os eventos que ocorrem no ambiente no qual está inserido. Nesse contexto, uma pergunta norteadora foi feita: “Como obter comportamentos autônomos naturais e realistas de personagens virtuais?”.

Para isso, Nogueira (2014) utiliza uma Rede Neural Neuromodulada em conjunto com um AG, visando obter um controlador de personagem com fortes possibilidades plásticas. Isso significa que mudanças na topologia da rede podem ocorrer, possibilitando uma dinâmica definida em escalas temporais.

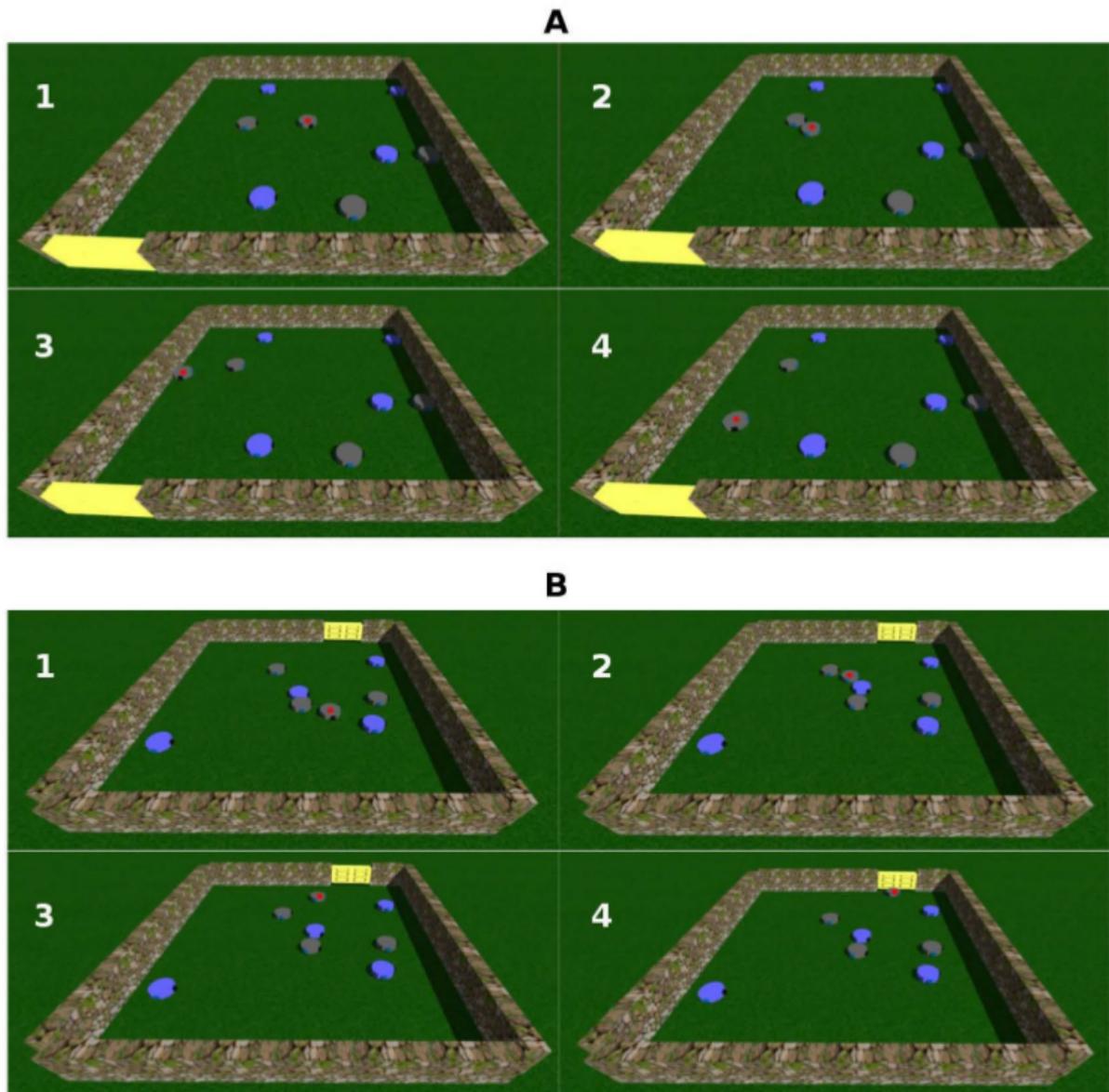
Na RNA proposta por Nogueira (2014) os neurônios de entrada são os valores de um sensor, e os neurônios de saída guardam a média de estados internos que estão conectados a eles. Isso foi feito visando tornar possível o controle dos personagens virtuais que vivem em

Figura 5 – Locomoção de um agente guiado por sua visão. Na parte superior esquerda de cada quadro da imagem é exibido o campo visual do agente selecionado pela *bounding box*.



Fonte: (CARNEIRO *et al.*, 2014)

Figura 6 – Comportamento de Navegação. Os casos A e B são compostos por quatro quadros que mostram a trajetória do agente com cérebro em evolução. (marcado com o ponto vermelho)



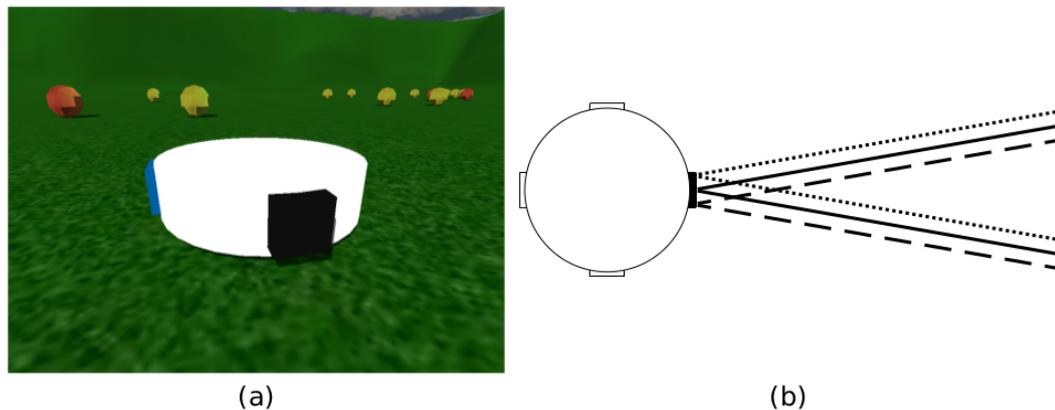
Fonte: (CARNEIRO *et al.*, 2014)

um ambiente dinâmico, permitindo o acoplamento de habilidades sensoriais aos personagens da simulação, representados por robôs. Uma codificação genética é aplicada utilizando um *Analog Genetic Encog* (MATTIUSI; FLOREANO, 2007 apud NOGUEIRA, 2014), em que os neurônios da rede são descritos no cromossomo, e as sinapses são formadas pela interação entre os genes.

Na simulação para obtenção dos comportamentos são utilizados robôs estilo *Khepera* (MONDADA *et al.*, 1999 apud NOGUEIRA, 2014). Os robôs possuem sensores que captam os dados enviados para a rede, representados pelas distâncias entre eles e as frutas e venenos

espalhados pelo ambiente. A coordenação motora de cada robô constitui em: mover-se para frente e para trás e rotação para esquerda ou direita. Um modelo de robô *Khepera* utilizado na abordagem é mostrado na Figura 7.

Figura 7 – O robô. (a) Robô no ambiente. A caixa preta é, ao mesmo tempo, seu olho e sua boca. (b) Os três sensores distribuídos no olho e os respectivos campos de sensibilidade, com ângulo de 20° cada.



Fonte: (NOGUEIRA, 2014)

Os robôs possuem uma quantidade de energia que é gasta à medida que a força e a frequência dos sinais são produzidos pela rede. Com isso, a função de avaliação para a emergência do comportamento utilizada foi a quantidade de comida coletada. Em seus experimentos, é mostrado que um robô consegue aprender a partir da utilização de seus sensores, com o objetivo de coletar frutas e evitar venenos. No primeiro caso de teste, é analisado o comportamento dos robôs na tarefa de forrageamento utilizando sensores de proximidade, os resultados mostraram que o AG foi capaz de evoluir a RNA, fazendo com que o robô apenas colhesse frutas, evitando os venenos que estivessem pelo caminho.

No segundo caso de teste, uma câmera de tamanho 10×10 pixels é acoplada ao robô, enviando para a rede apenas a média aritmética dos 10 pixels centrais no formato RGB. Mudanças foram feitas, com a adição de pedras que serviram como obstáculos. As cores foram redefinidas para uma melhor diferenciação das pedras, frutas e venenos. Foi notado nas simulações que equipando os robôs com um CVA, o comportamento apresentou melhores resultados, em que reduziu-se a tendência do robô rodar, por exemplo, comportamento que foi observado no caso de teste 1. Como a visão baseada na câmera está sempre enviando dados para rede neural, o personagem consegue ver o seu ambiente, possibilitando um melhor controle de seus movimentos, principalmente em trajetórias mais precisas na busca pelos alvos. Como pode ser visto na Figura 8.

Figura 8 – Comportamento do robô equipado com um CVA. As linhas pontilhadas mostram o posicionamento de frutas e obstáculos pelo ambiente que são visualizados pelo sensor visual. No quadro 32, a linha pontilhada indica o caminho percorrido pelo robô.



Fonte: (NOGUEIRA, 2014)

Baseado nos trabalhos discutidos nesta seção, a proposta deste trabalho irá utilizar uma Rede Neural mais robusta, chamada de *NeuroEvolution of Augmenting Topologies* (NEAT),

que será discutida na próxima seção.

2.2 NEAT - *NeuroEvolution of Augmenting Topologies*

Esta seção irá discutir o método chamado NEAT (STANLEY; MIIKKULAINEN, 2002). Este método já foi utilizado em trabalhos sobre Neuroevolução aplicada à NPCs e em discussões sobre Vida Artificial (SILVA, 2015) (MIGUEL, 2009).

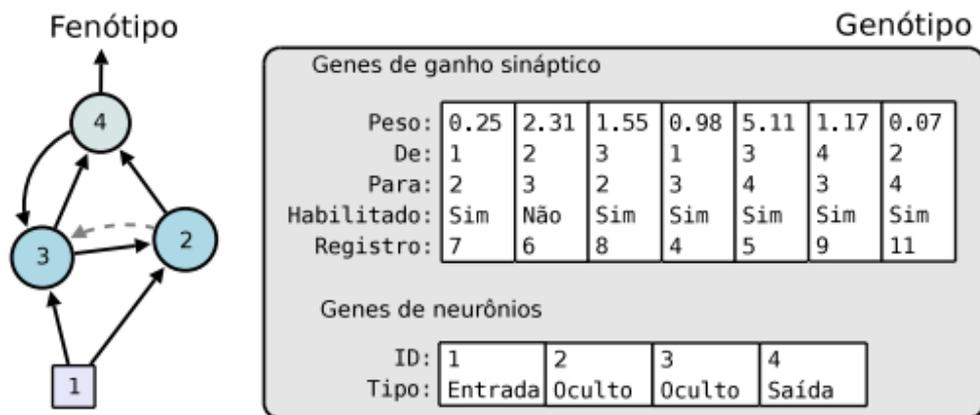
Os AGs comuns para evoluir RNAs geralmente evoluem os pesos das redes. A NEAT pode evoluir tanto pesos quanto topologias (estruturas da rede), não existindo nenhuma restrição sobre as modificações que possam ser feitas em sua arquitetura.

As informações presentes nesta seção foram baseadas no trabalho original da NEAT (STANLEY; MIIKKULAINEN, 2002) e nos trabalhos de (SILVA, 2015) e (MIGUEL, 2009).

2.2.1 Codificação genética

Cada indivíduo é formado por duas características básicas: o fenótipo de um indivíduo, que consiste em uma rede neural; e o genótipo (genoma), que guarda informações genéticas sobre as conexões de um fenótipo. A Figura 9 mostra a estrutura de fenótipo e genótipo na NEAT.

Figura 9 – Exemplo de um genótipo e seu fenótipo (rede neural).



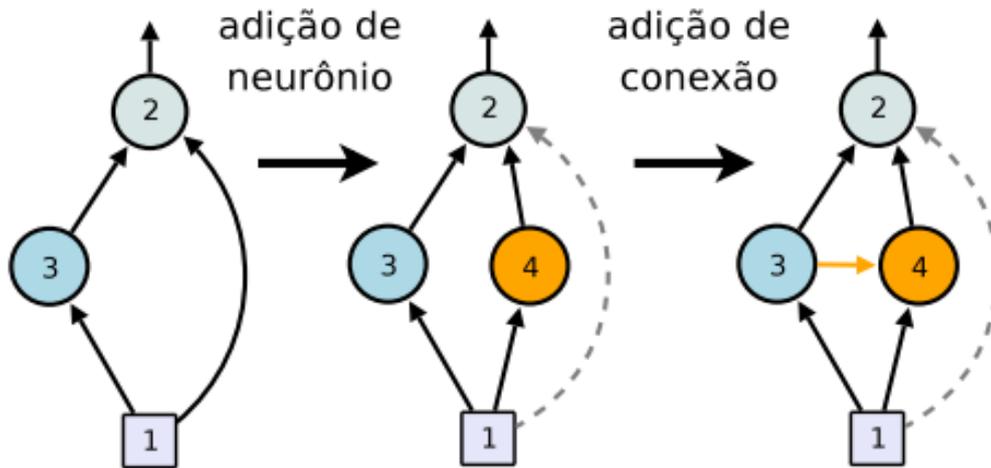
Fonte: (MIGUEL, 2009)

O genótipo é dividido em dois tipos de genes: os de conexões (genes de ganhos sináptico) que guardam informações sobre as conexões no fenótipo, que inclui o número de inovação (ou registro histórico) do gene, concedido a cada novo gene que surge a partir de uma

mutação; e os de neurônio que indicam se os nós presentes na topologia são de entrada, saída ou ocultos.

Como em outros tipos de AGs, o da NEAT possui as fases de mutação e cruzamento. A mutação é dividida em dois tipos: paramétrica e estrutural. A paramétrica é responsável por modificar os pesos dos genes, representando os ganhos sinápticos da rede. A estrutural é responsável pela adição de novos neurônios e conexões entre os neurônios existentes. A Figura 10 mostra um exemplo de uma mutação estrutural.

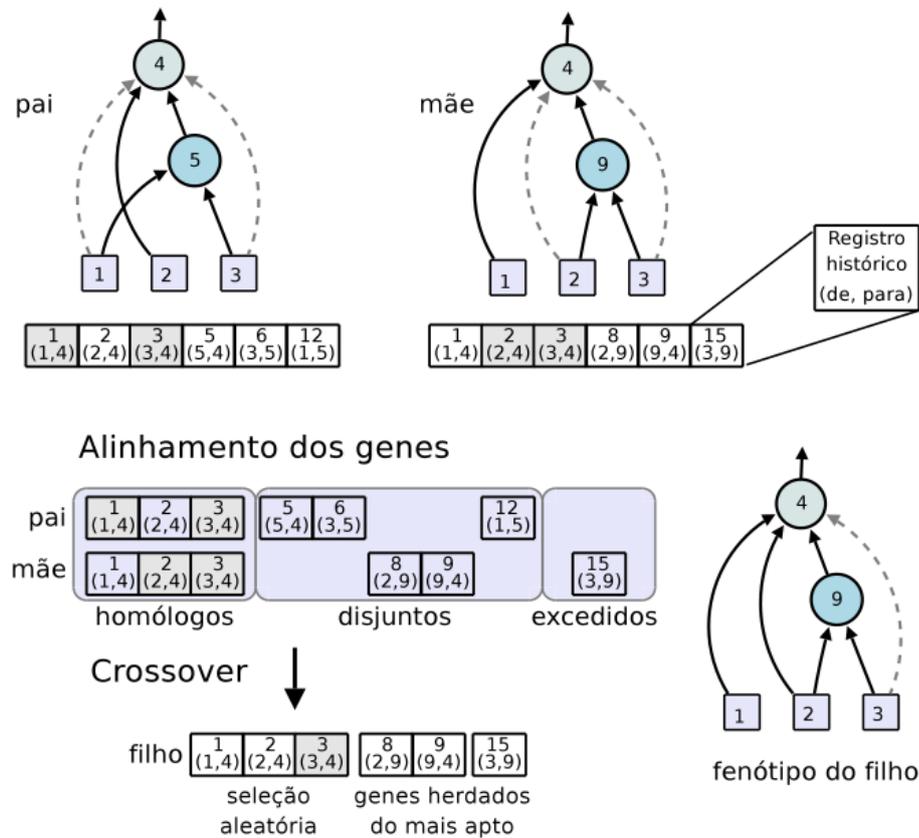
Figura 10 – Exemplo de mutação estrutural com a adição de uma nova conexão e um novo neurônio



Fonte: (MIGUEL, 2009)

Uma nova conexão entre dois neurônios é feita adicionando um gene na lista dos genes de conexão. Desta forma, liga-se dois genes não conectados. Para adicionar um novo neurônio cria-se mais dois genes que representam uma conexão, conectando o novo neurônio aos outros dois que estavam conectados e, após isso, desabilita-se a conexão antiga.

O cruzamento leva em consideração a topologia da rede para gerar um novo indivíduo. O processo ocorre alinhando-se os genes de um pai e uma mãe em grupos distintos, para isso, utiliza-se seus registros históricos. Os grupos são divididos em: homólogos, que são os genes de mesmo registro; os disjuntos são os genes que são diferentes entre si; e os excedidos são os genes do indivíduo que possui o maior registro histórico dentre todos os genes. Os genes do novo indivíduo (filho) serão formados pela seleção aleatória dentre os genes homólogos, pela escolha dos genes do indivíduo mais apto (maior pontuação adaptativa) nos genes disjuntos e pela adição simples dos genes excedidos. A Figura 11 mostra como ocorre o cruzamento na NEAT.

Figura 11 – Operador de cruzamento (*crossover*)

Fonte: (MIGUEL, 2009)

Um problema que pode ocorrer é que genótipos de tamanhos variados possuem uma maior dificuldade em fazer a permutação dos genes. Esse problema é minimizado utilizando o registro histórico de cada gene para separá-los por similaridade topológica. Este processo é chamado de especiação.

2.2.2 *Especiação ou Proteção de Inovação*

Quando sofre uma mutação, uma estrutura topológica perde desempenho e pode ser eliminada da população por causa da perda de informações sinápticas. Assim, para estes casos o que a NEAT faz é especiar genótipos, para que as inovações (evoluções) topológicas sejam feitas com indivíduos parecidos, isso permite que novas estruturas se desenvolvam antes de competir com o restante da população. Portanto, o objetivo deste processo é proteger novas estruturas, dando-lhes mais tempo para se adaptarem. Para medir a similaridade S (distância) entre dois genótipos é utilizado a seguinte fórmula (2.2).

$$S = \frac{c_1}{N}E + \frac{c_2}{N}D + c_3W, \quad (2.2)$$

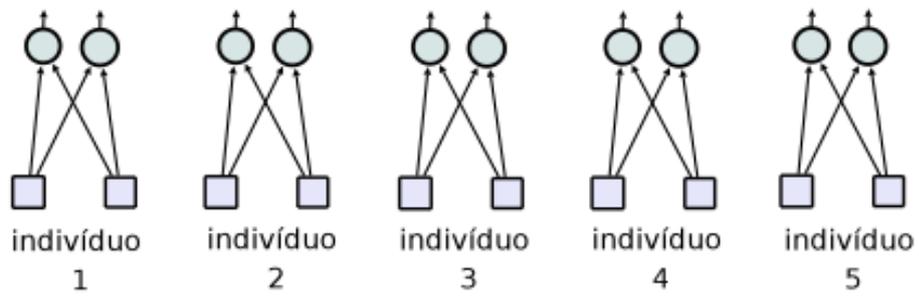
S é calculada utilizando os registros históricos dos genes de cada indivíduo: a quantidade de genes excedidos (E); os genes disjuntos D ; a diferença de ganhos conectivos entre genes semelhantes W ; número de genes do maior genótipo da população N ; e os valores c_1 , c_2 e c_3 , que ponderam as diferenças entre dois genótipos e suas respectivas topologias.

2.2.3 *Seleção*

A seleção ocorre ordenando todos os indivíduos utilizando seu *fitness*. Pega-se o melhor de cada espécie (conjunto de indivíduos similares) para ser o seu representante e ser mantido na próxima geração. Após este processo, uma porcentagem é escolhida para reprodução e os demais são descartados. A reprodução ocorre escolhendo aleatoriamente dois pais entre os indivíduos restantes para gerar um filho. O processo se repete até que se chegue na quantidade de indivíduos da nova população.

A primeira geração de indivíduos carrega em seus genes a menor topologia possível, isso significa que todas as entradas estão conectadas diretamente às saídas (Figura 12), isso faz com que o espaço de busca seja minimizado.

Figura 12 – Exemplo de uma topologia mínima de entrada

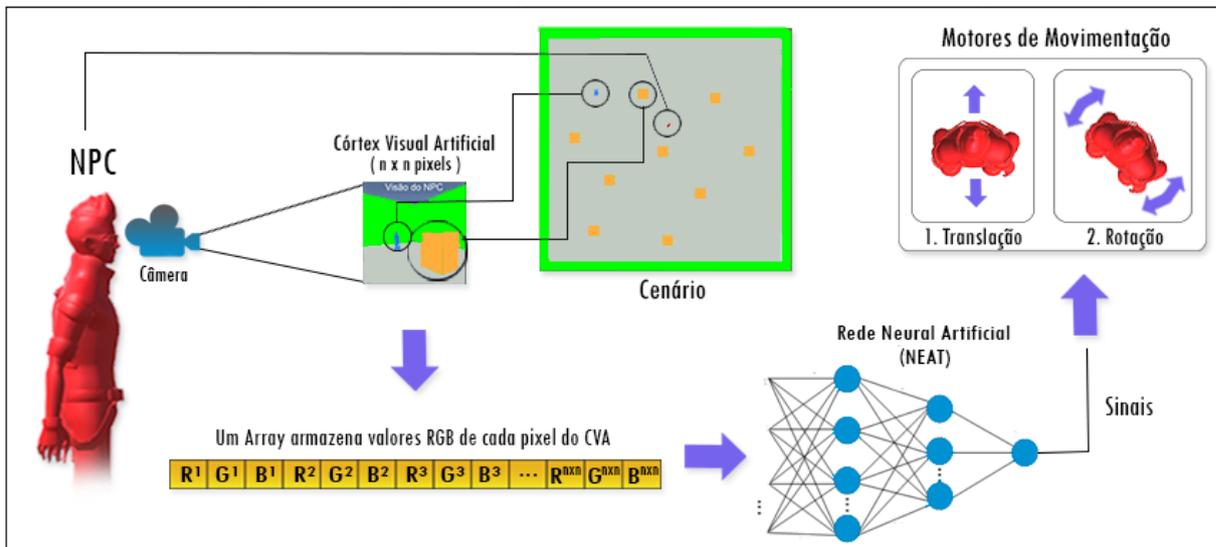


Fonte: (MIGUEL, 2009)

3 PROPOSTA

Como visto no Capítulo 2, as técnicas de Carneiro *et al.* (2014) e Nogueira (2014) utilizam um CVA para simulação de comportamentos autônomos em personagens virtuais. Com a finalidade de gerar NPCs autônomos, capazes de aprender a partir de suas interações com o ambiente, este trabalho utilizará como base o modelo de CVA proposto por esses dois trabalhos. Como estes CVAs irão dispor de mais informações visuais do ambiente, espera-se que o processo de evolução da rede e aprendizado do NPC leve a comportamentos mais detalhados e precisos. A Figura 13 detalha a proposta deste trabalho.

Figura 13 – O NPC possuíra uma câmera acoplada na parte correspondente a visão do modelo utilizado (*Ethan*), a câmera fornecerá informações visuais sobre o cenário para uma RNA, que será responsável por ativar os motores de movimentação.

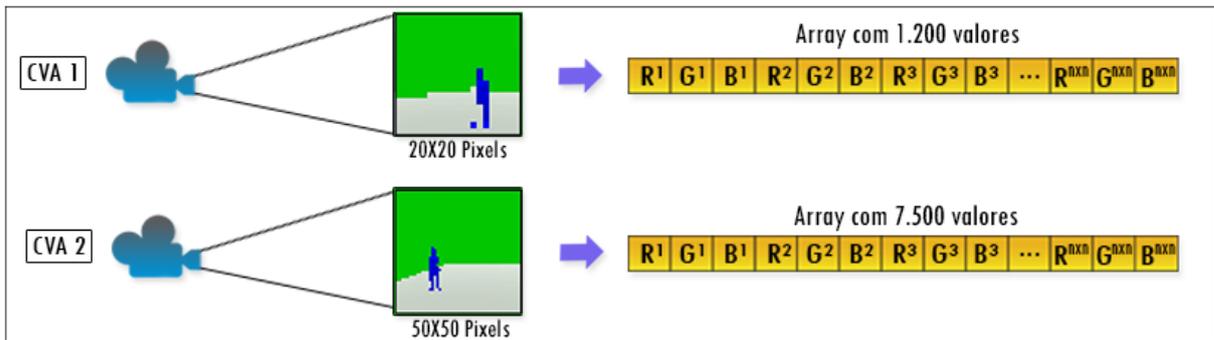


Fonte: Próprio Autor

3.1 O Córtex Visual Artificial

Cada CVA basicamente é formado por uma câmera virtual que fornece os *pixels* capturados pela resolução de seu quadro. Serão 2 tipos de resolução para cada CVA: o primeiro com 20x20 *pixels* resultará em 1.200 entradas (cada *pixel* é representado por 3 valores no sistema de cores RGB) para a rede; e o segundo com 50x50 *pixels*, resultando em 7.500 entradas. A Figura 14 mostra o funcionamento dos 2 CVAs propostos por este trabalho.

Figura 14 – CVAs propostos



Fonte: Próprio Autor

3.2 O Controlador

Os movimentos do NPC são determinados pelos sinais de saída da RNA, cujas entradas são as informações visuais do NPC. O processo evolutivo da rede, baseado em Algoritmo Genético, levará a um melhoramento gradativo do comportamento do NPC.

3.2.1 A RNA e o Processo Evolutivo

Neste trabalho é utilizada a *framework* UnityNEAT. Esta é uma implementação do método NEAT para a *Game Engine* Unity, criada para utilização no jogo *EvoCommander*, que tem o objetivo de demonstrar o potencial da utilização de Neuroevolução em jogos eletrônicos (JALLOV *et al.*, 2017).

O processo evolutivo utilizado neste trabalho é o mesmo do método NEAT, especificado na Seção 2.2. Apenas alguns parâmetros precisam ser configurados: o tamanho da população, o número de espécies, a quantidade de entradas e saídas e a função de avaliação

3.2.2 Função de Avaliação

Durante a fase de treinamento, para avaliar quais os indivíduos que terão o melhor desempenho, é utilizada a mesma função de avaliação f_{av} apresentada por Carneiro *et al.* (2014)

$$f_{av} = \frac{1}{t} + \frac{1}{d}. \quad (3.1)$$

Nesta função, d representa a distância do NPC até o *Player* e t o tempo que o NPC levou para chegar até o *Player*. Os melhores NPCs avaliados serão aqueles que conseguirem obter a menor distância até o *Player* no menor tempo possível. O tempo máximo estabelecido é

de 25 segundos de simulação para cada indivíduo.

3.3 A Game Engine

Por se tratar de uma proposta voltada para jogos, toda a simulação é feita na *Game Engine Unity 3D*, escolhida por oferecer as condições necessárias para criar uma experiência que poderá ser aplicada no desenvolvimento de jogos.

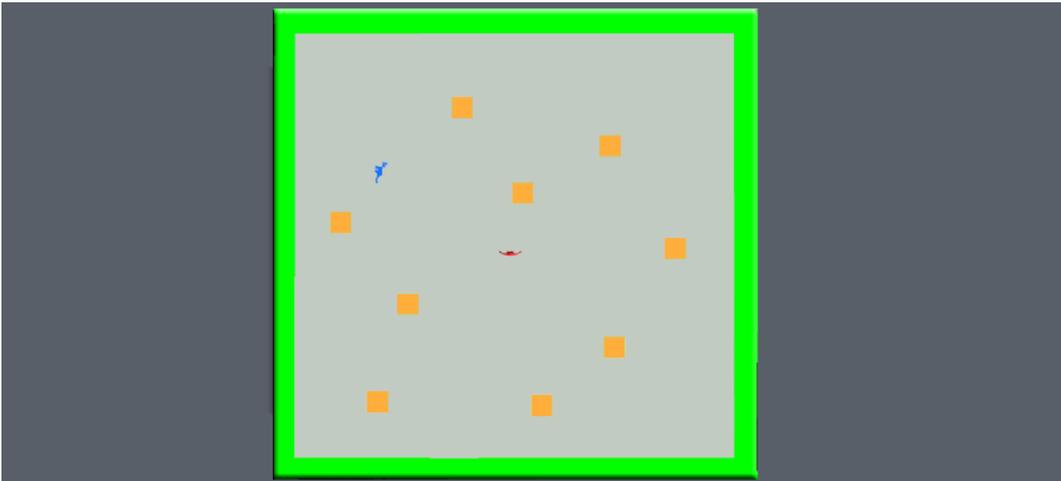
3.4 Personagens, Obstáculos e Cenário

O cenário é composto por uma área quadrada com 4 paredes, 9 obstáculos, 1 *Player* e 1 NPC (Figura 15). O NPC é sempre posicionado no centro da sala com o ângulo de rotação definido de forma aleatória. O *Player* a depender do cenário de treino, poderá estar em uma posição fixa na sala ou ser posicionado aleatoriamente. Os obstáculos são cubos com cor laranja (Figura 16), que representam caixotes com variações de alturas que precisam ser desviados pelo NPC no caminho até o *Player*. O modelo padrão fornecido pela *Unity Technologies*, chamado *Ethan*, é utilizado para representar o *Player*, na cor azul, e o NPC na cor vermelha (Figura 17). Este modelo foi escolhido por ser uma representação comum de personagem humanóide utilizado em jogos. As paredes possuem a cor verde (Figura 15), visando facilitar a diferenciação das cores do *Player* e obstáculos pelo CVA.

Para os treinamentos foram definidos 3 cenários:

- (a) Cenário 1: O Objetivo (*Player*) estará em um ponto fixo localizado em um canto da sala.
- (b) Cenário 2: O posicionamento do objetivo (*Player*) na sala será definido de forma aleatória, respeitando uma distância com raio 15 até o NPC.
- (c) Cenário 3: O objetivo (*Player*) estará em um ponto fixo localizado em um canto da sala, porém existem obstáculos (Figura 16) espalhado pela sala que eventualmente poderão cobrir a visualização do *Player*.

Figura 15 – Cenário visto de cima

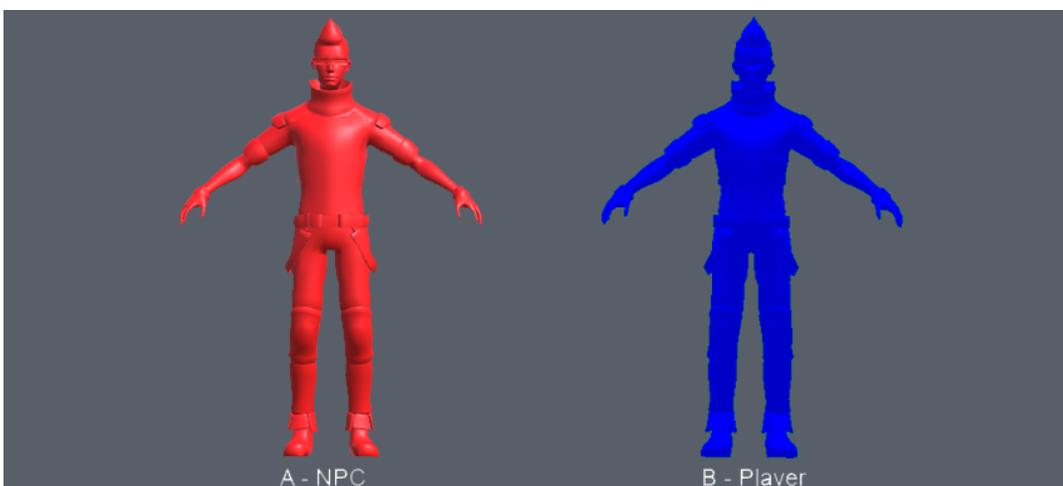


Fonte: Próprio Autor

Figura 16 – Cubo na cor laranja representando os obstáculos no cenário.



Fonte: Próprio Autor

Figura 17 – Modelo *Ethan* - O modelo A representa o NPC na cor vermelha (em RGB: 255, 0, 0) e o modelo B representa o *Player* (Objetivo) na cor azul (em RGB: 0, 0, 255).

Fonte: Próprio Autor

4 RESULTADOS

No início de cada treinamento, o indivíduo (NPC) da população estará sempre posicionado no centro da sala, apenas variando o seu ângulo de rotação. Será avaliado os impactos na aprendizagem dos indivíduos de uma geração quando se trabalha com CVAs de dois tipos de tamanhos: um de 20x20 *pixels* e outro de 50x50 *pixels*. Conseqüentemente a quantidade de entradas para a rede irá variar. Para cada cenário de treinamento, a quantidade dos indivíduos da população é 50 e o número de especiação (proteção de novas espécies) é 10.

A configuração do computador utilizado nas simulações deste trabalho são: processador Core i3 dual-core de 1.75Ghz; memória RAM de 4 *GigaBytes*; e disco rígido de 500 *GigaBytes*.

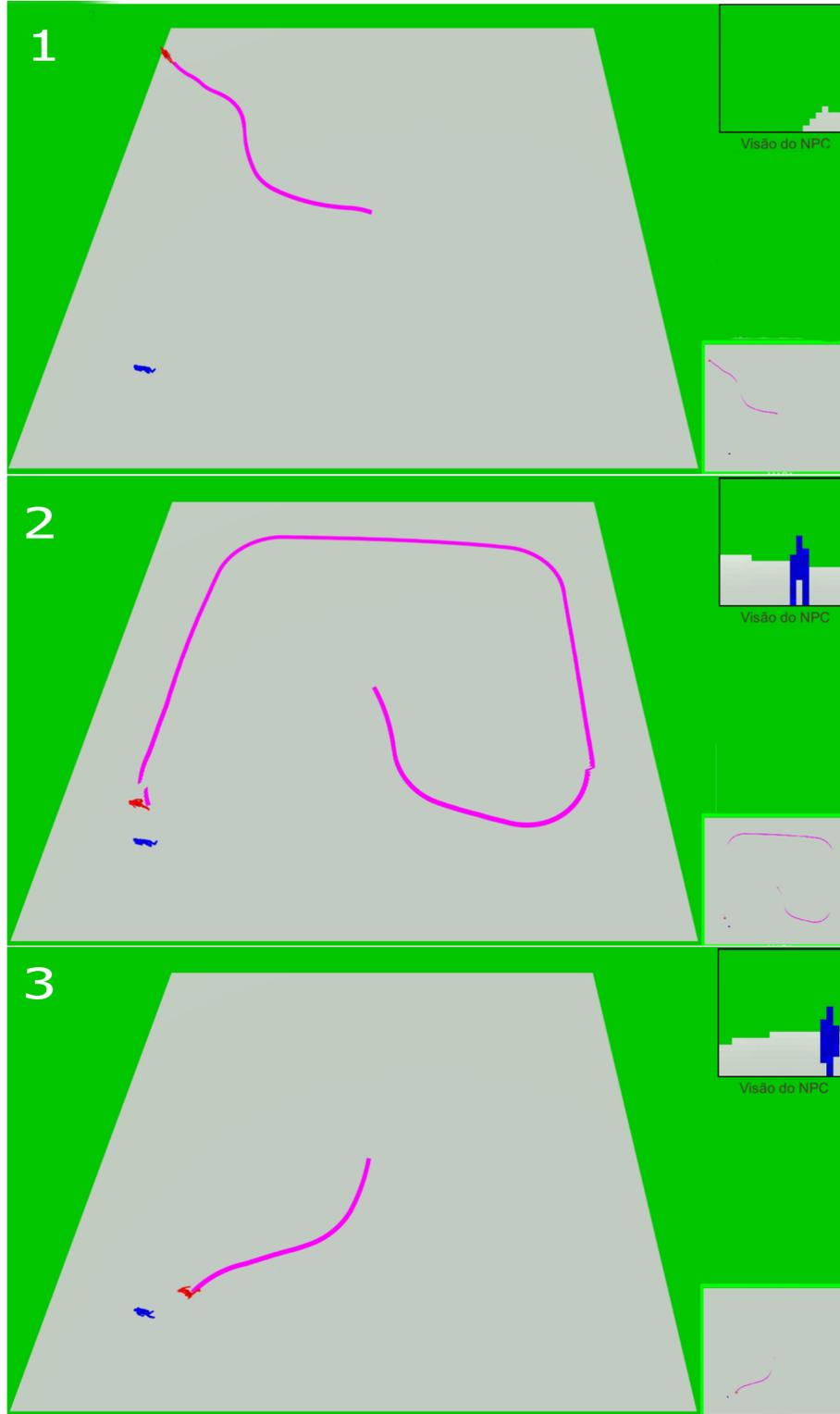
4.1 Resultados do Cenário 1

Neste cenário, o *Player* (objetivo) está em um ponto fixo localizado em um canto da sala e o NPC está posicionado no centro, mudando apenas seu ângulo de rotação inicial que é definido de forma aleatória, variando de 0 até 360 graus a cada início de treino.

4.1.1 CVA com 20x20 pixels - 1.200 entradas para a RNA

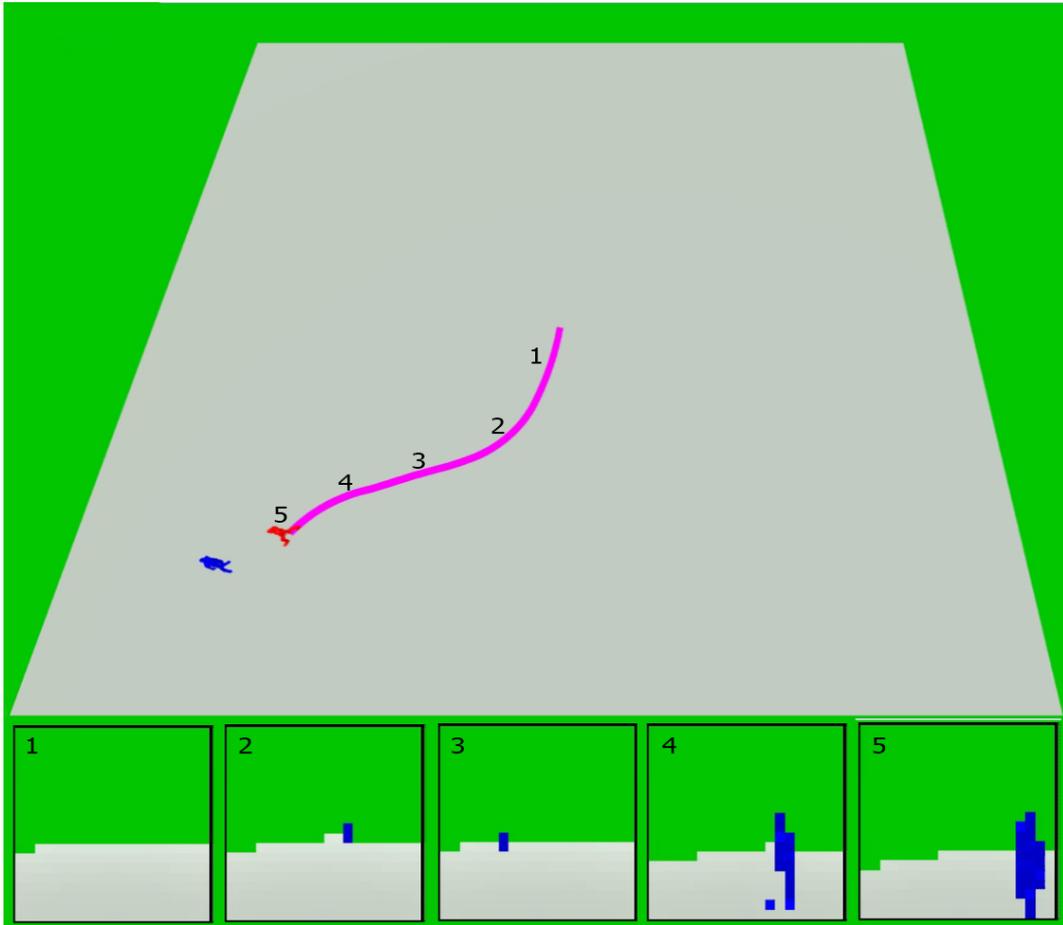
Durante o treinamento, foi observado que o NPC teve certa dificuldade em identificar o *Player*, principalmente quando o NPC se posiciona no canto oposto ao *Player*, enquanto o procura. Esta dificuldade pode ser atribuída à baixa resolução da visão, fazendo com que o NPC enxergue o *Player* apenas a uma curta distância.. Essa questão gerou um impacto na aprendizagem durante o treinamento, fazendo com que o comportamento do NPC não fosse satisfatório em situações onde o *Player* não fosse avistado rapidamente na simulação, como mostrado no quadro 1 da Figura 18. Os quadros 2 e 3 mostram que a rede conseguiu traçar caminhos até o *Player*. No quadro 3, como o *Player* foi visto rapidamente, o NPC foi capaz de atingir o objetivo de forma direta, enquanto que no quadro 2, o NPC explorou o ambiente à procura do *Player*. A Figura 19 mostra com mais detalhes o caminho feito pelo NPC no quadro 3 da Figura 18.

Figura 18 – Cenário 1 (20x20) - A movimentação em cada quadro se refere aos comportamentos gerados pelo NPC após o treinamento para o CVA de 20x20. O traço lilás indica o caminho percorrido por ele.



Fonte: Próprio Autor

Figura 19 – Detalhamento do caminho feito pelo NPC no quadro 3 da Figura 18, com as visões em 5 instantes diferentes de simulação



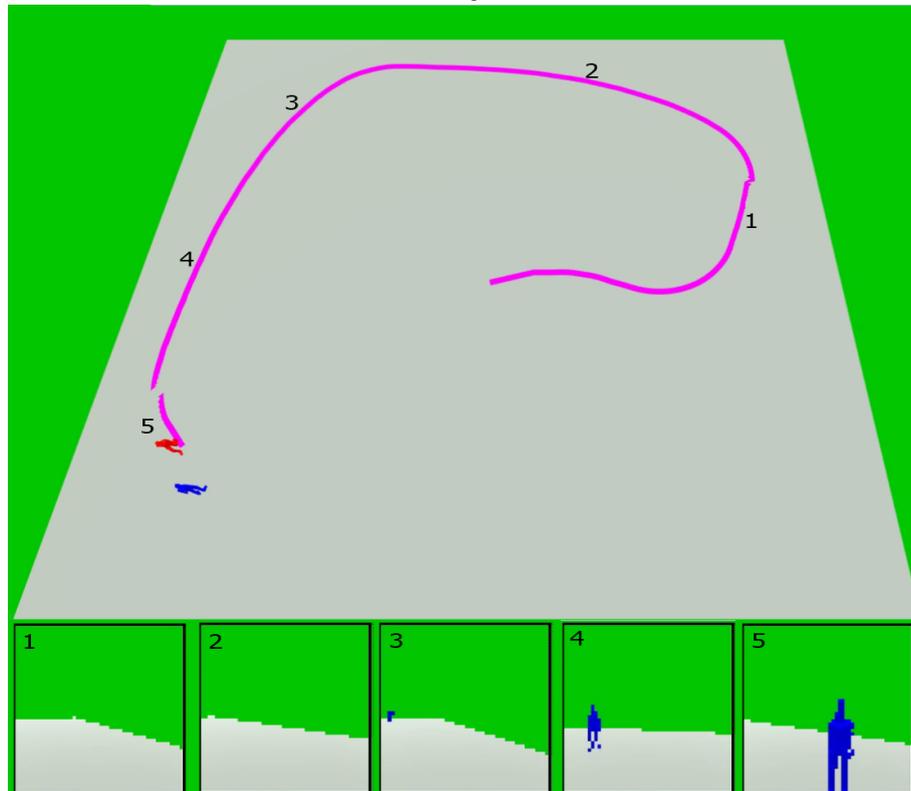
Fonte: Próprio Autor

4.1.2 CVA com 50x50 pixels - 7.500 entradas para a RNA

A Figura 20 mostra o caminho com pontos de visualização indicando a visão do NPC. Observa-se que a partir do momento que ele viu os *pixels* azuis, ele se deslocou até o *Player*.

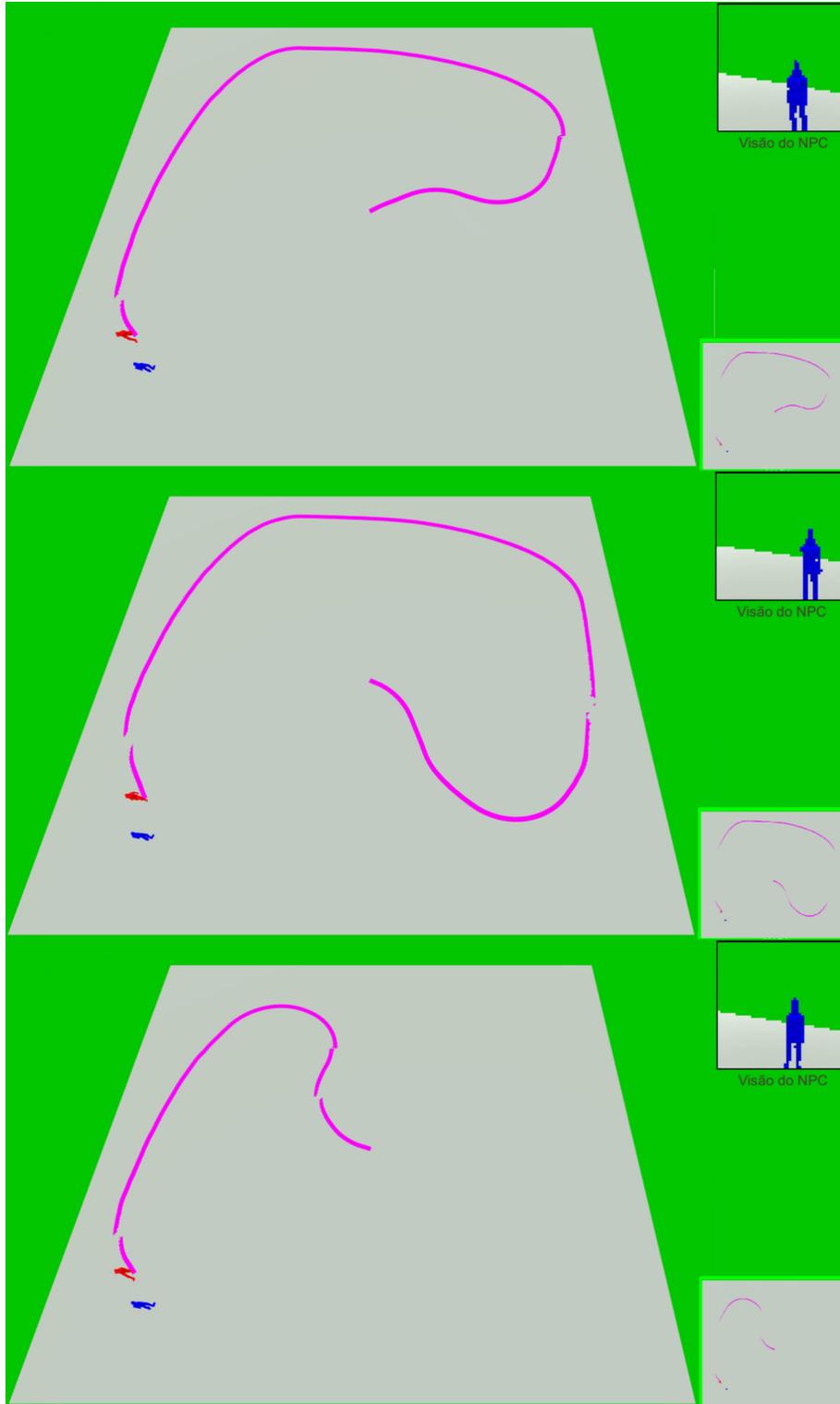
Utilizando um CVA com maior resolução, os problemas relacionados a visualização do *Player* discutidos na subseção anterior não ocorreram, pois, com mais informações visuais o NPC conseguiu visualizar o *Player* mesmo estando em qualquer parte do cenário. Este detalhe foi decisivo para o treinamento, pois em qualquer posicionamento inicial o NPC conseguiu traçar um caminho até o *Player*, como pode ser observado na Figura 21.

Figura 20 – Detalhamento do caminho feito pelo NPC no quadro 1 da Figura 21, com as visões em 5 instantes diferentes da simulação.



Fonte: Próprio Autor

Figura 21 – Cenário 1 (50x50) - A movimentação em cada quadro se refere aos comportamentos gerados pelo NPC após o treinamento para o CVA de 50x50. O traço lilás indica o caminho percorrido por ele até o *Player*.



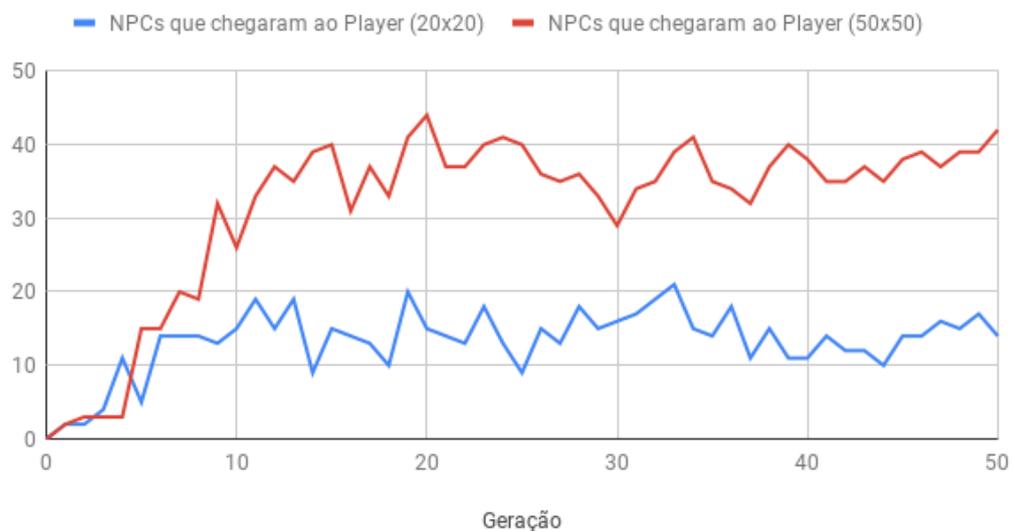
Fonte: Próprio Autor

4.1.3 Evolução em cada tipo de CVA

O gráfico da Figura 22 mostra a quantidade de NPCs que chegaram até o *Player* em cada tipo de CVA. Observa-se que os resultados para o CVA de 50x50 *pixels* foram bem melhores do que o de 20x20. Isso demonstra que para este cenário, um maior sensoriamento visual foi relevante na tarefa de encontrar o *Player*. Porém, deve-se levar em conta que para alguns casos isolados, nem sempre chegar até o *Player* significa que o caminho percorrido foi o ideal (caminho com realizado com menos tempo), pois, pode acontecer do NPC fazer vários giros pela sala antes de encontrar o *Player*. Isso gera um impacto na pontuação do indivíduo, visto que o tempo gasto para chegar ao *Player* é levado em consideração pela função de avaliação.

Figura 22 – Gráfico que compara a quantidade de NPCs que conseguiram chegar ao *Player* em cada geração para cada tipo de CVA.

NPCs que conseguiram chegar ao Player (Cenário 1)

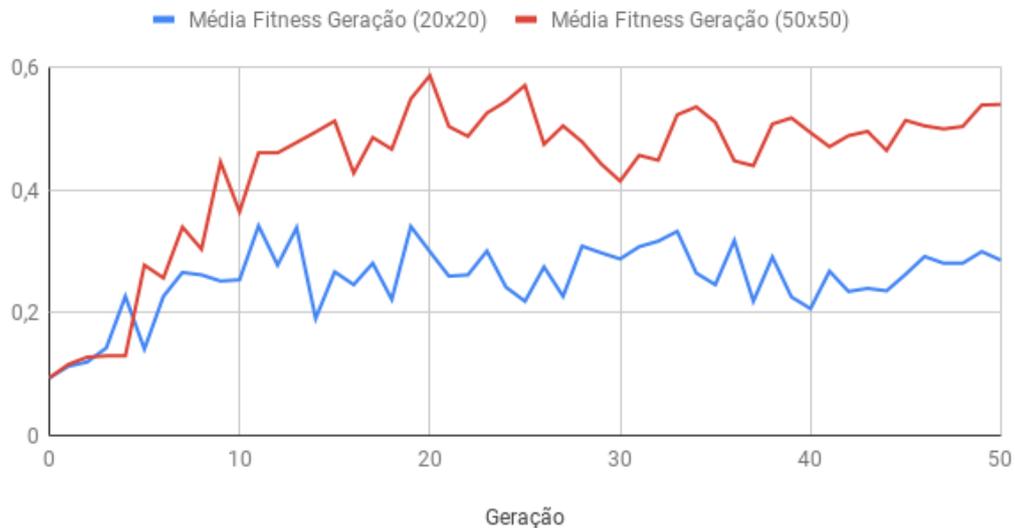


Fonte: Próprio Autor

No gráfico da Figura 23 é mostrada a média da função de avaliação dos indivíduos em cada geração. Nota-se que os NPCs com visão de maior resolução conseguiram melhores avaliações no geral, reforçando a hipótese de que quanto maior a resolução, melhor o comportamento alcançado. O fato da não visualização do *Player* pelo CVA de 20x20 teve um impacto com relação ao processo evolutivo da NEAT, visto que a evolução das topologias (fenótipos) de cada indivíduo não possuíam informações genéticas de aprendizado presentes em seu genótipo.

Figura 23 – Gráfico que compara a média do *fitness* obtido por cada geração em diferentes CVAs.

Média Fitness por Geração (Cenário 1)



Fonte: Próprio Autor

4.2 Resultados do Cenário 2

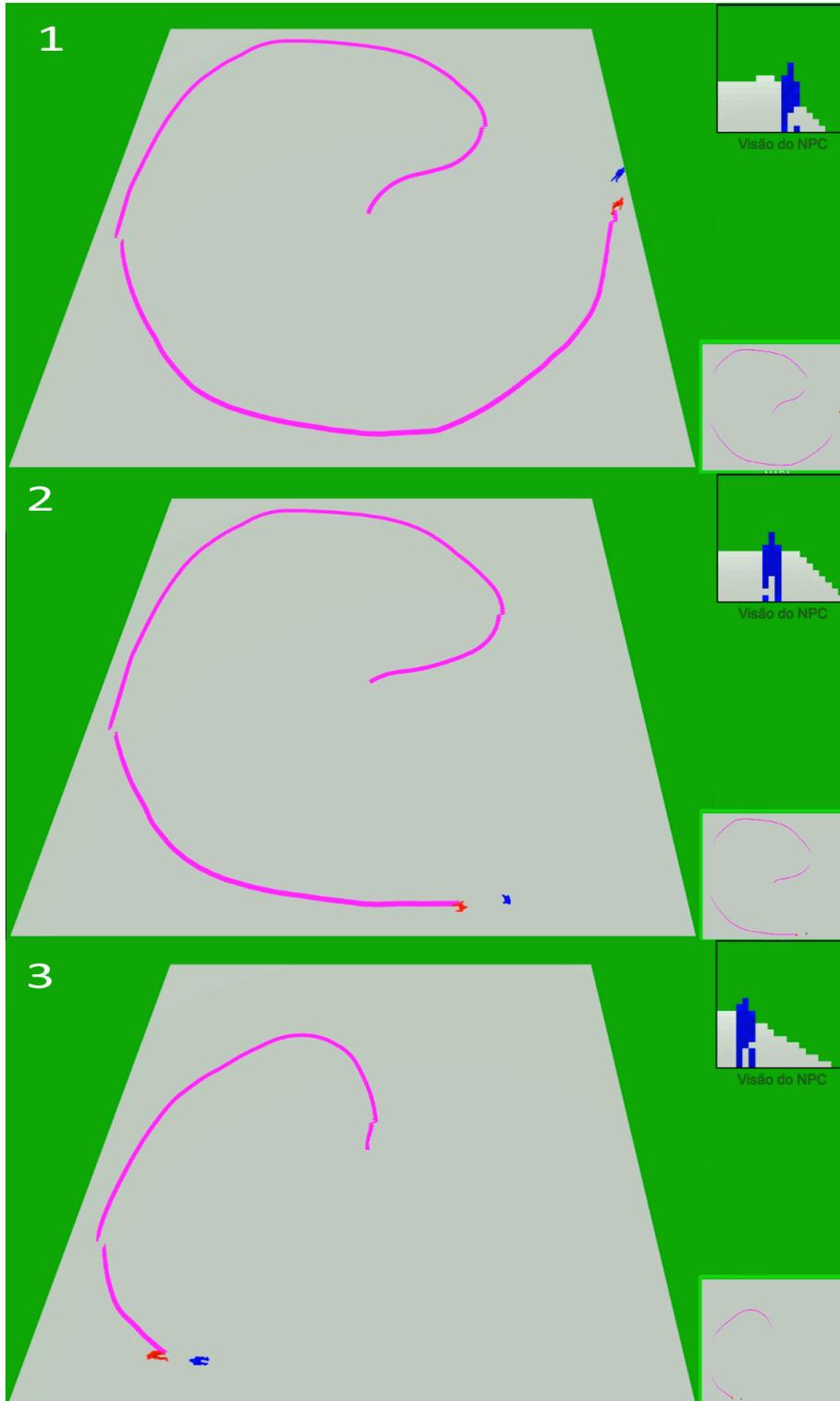
Neste cenário, o posicionamento do objetivo (*Player*) na sala é definido de forma aleatória durante o treinamento, respeitando uma distância de 15 unidades métricas até o NPC.

4.2.1 CVA com 20x20 pixels - 1.200 entradas para a RNA

Para este CVA o NPC conseguiu aprender a encontrar o *Player* seguindo uma tendência de caminho circular. A rede conseguiu aprender que seguindo este caminho próximo às laterais em algum momento o *Player* seria encontrado. Ressalta-se que os problemas de visualização do *Player* não aconteceram com frequência neste caso, porque o posicionamento aleatório deixa menos frequente as situações onde o *Player* muito distante do NPC. A Figura 24 mostra três diferentes caminhos feitos pelo NPC já treinado, nos testes não houveram casos em que ele não conseguia traçar um caminho até o *Player*.

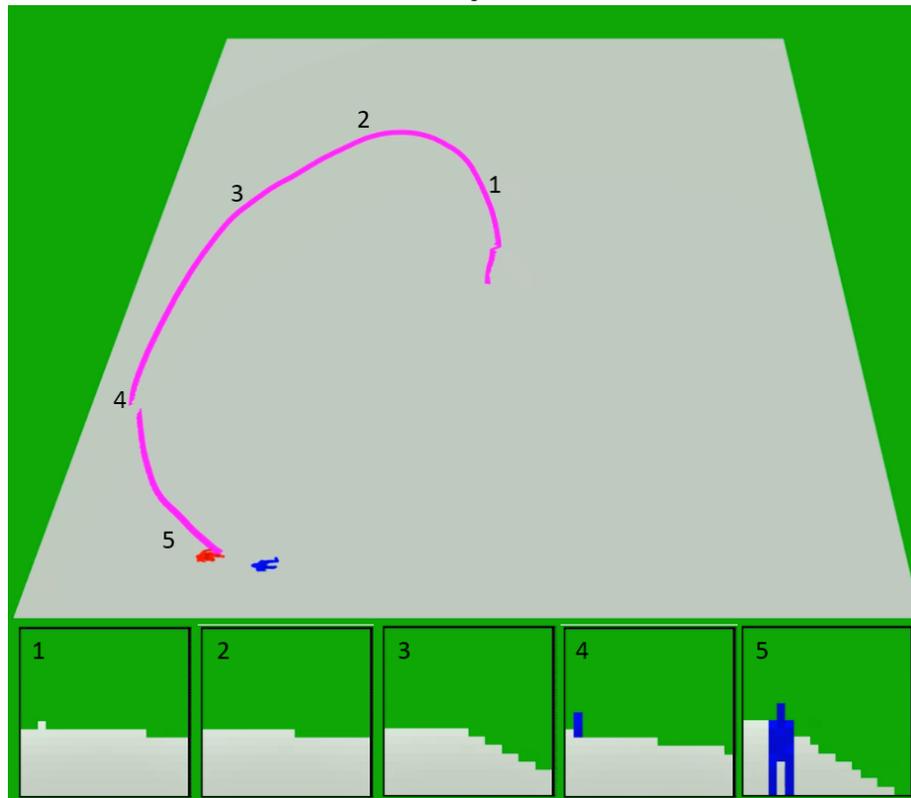
A Figura 25 mostra diferentes pontos de visão do NPC em direção ao *Player* no caminho do quadro 3 da Figura 24. Neste caso, o NPC iniciou na sala de costas para o *Player*, mas ele aprendeu que bastava seguir um caminho circular pela sala para encontrá-lo.

Figura 24 – Cenário 2 (20x20) - A movimentação em cada quadro se refere aos comportamentos gerados do NPC após o treinamento para o CVA de 20x20. O traço lilás indica o caminho percorrido por ele.



Fonte: Próprio Autor

Figura 25 – Detalhamento do caminho feito pelo NPC no quadro 3 da Figura 24, com as visões em 5 instantes diferentes da simulação.



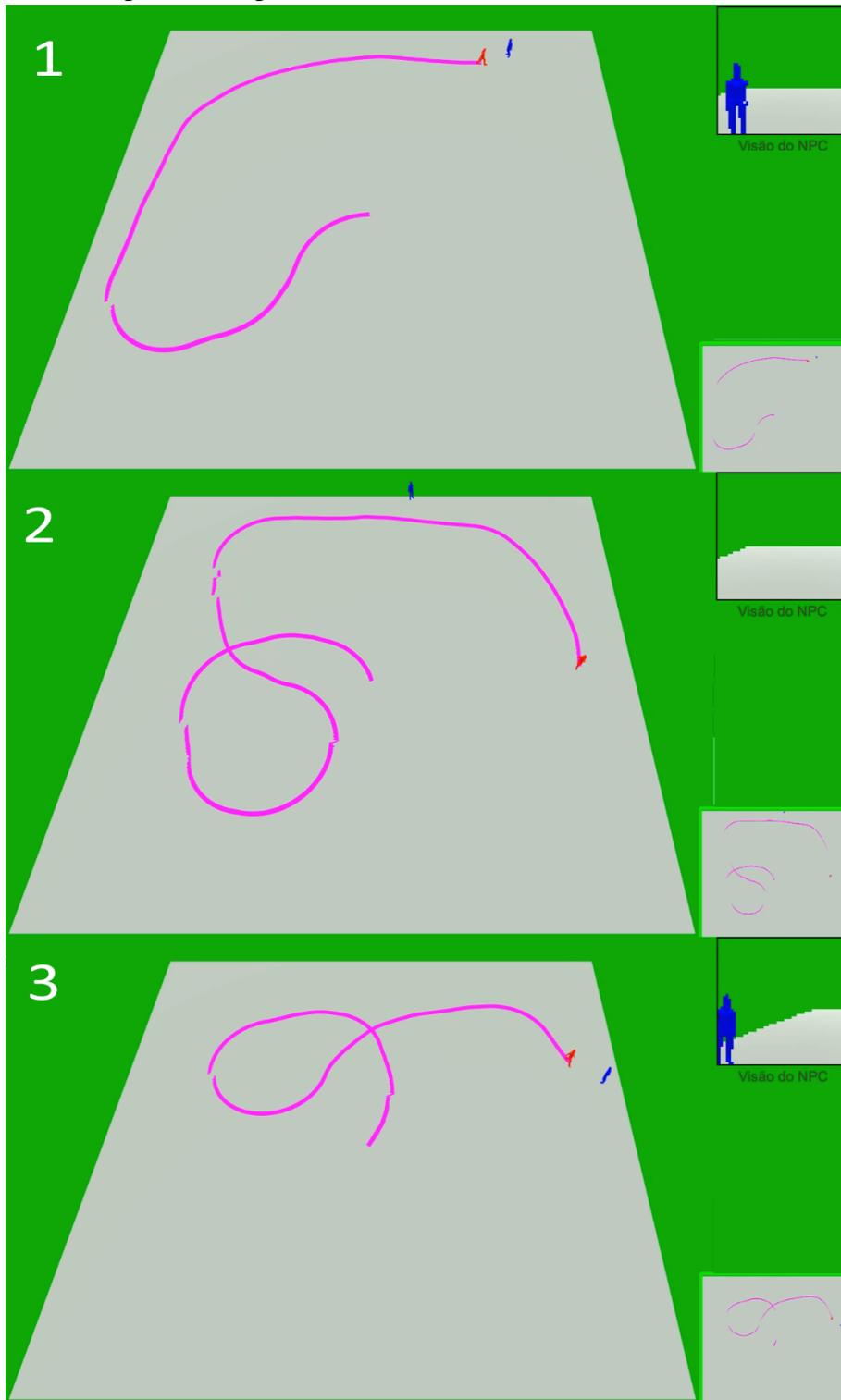
Fonte: Próprio Autor

4.2.2 CVA com 50x50 pixels - 7.500 entradas para a RNA

Utilizando uma resolução de 50x50, o NPC continuou apresentando, em geral, bons comportamentos, assim como na resolução de 20x20. A Figura 26 mostra três diferentes caminhos. Nos quadros 1 e 3 o NPC conseguiu encontrar o *Player*. No quadro 2, o NPC chega a visualizar o *Player*, mas não vai em sua direção. Este comportamento pode ser devido à proximidade do *Player* com a parede, pois a rede deve ter identificado que encostar na parede não seria uma boa decisão. Talvez, com um maior número de gerações, a evolução tratasse melhor estes casos isolados.

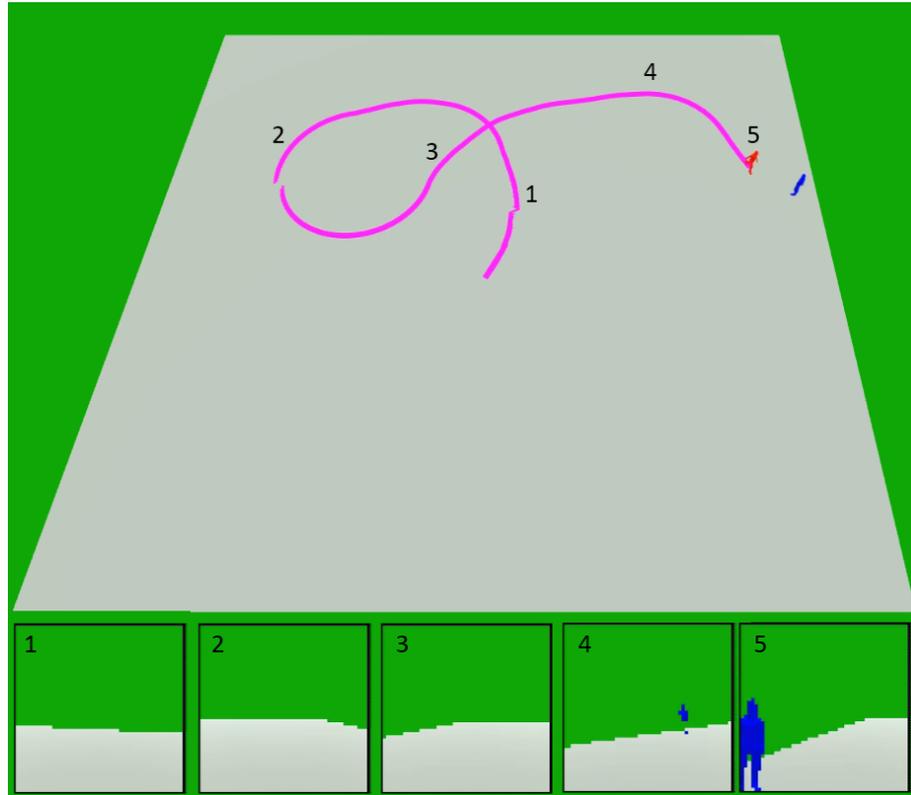
A Figura 27 mostra diferentes pontos de visão do NPC em direção ao *Player* no caminho do quadro 3 da Figura 26. Neste caso, o NPC deu uma volta à procura do *Player* encontrando-o a partir do ponto de visão 4.

Figura 26 – Cenário 2 (50x50) - A movimentação em cada quadro se refere aos comportamentos gerados do NPC após o treinamento para o CVA de 50x50. O traço lilás indica o caminho percorrido por ele.



Fonte: Próprio Autor

Figura 27 – Detalhamento do caminho feito pelo NPC no quadro 3 da Figura 26, com as visões em 5 instantes diferentes da simulação.



Fonte: Próprio Autor

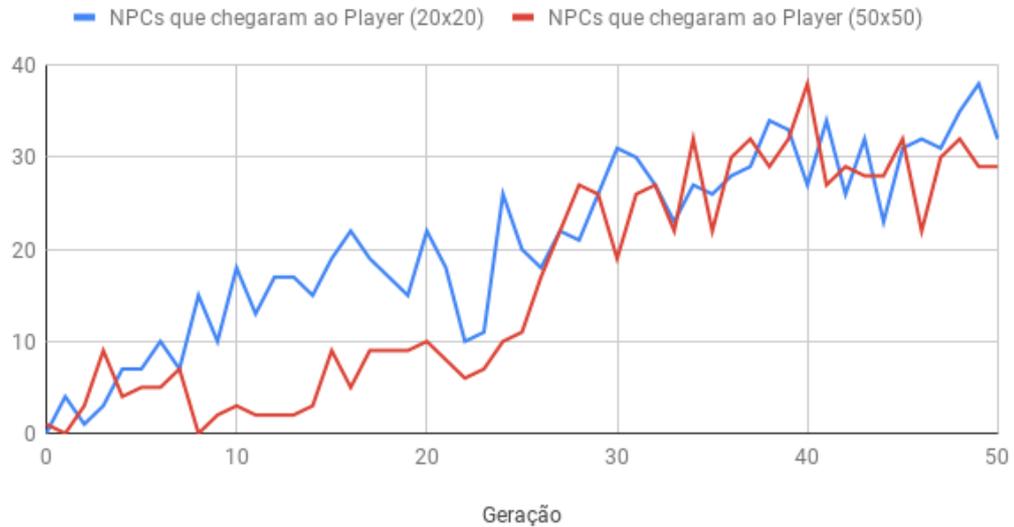
4.2.3 Evolução em cada tipo de CVA

Este é um cenário importante porque em jogos a aleatoriedade da movimentação e do posicionamento do *Player* é algo comum e NPCs autônomos precisam aprender a lidar com isso.

Ao contrário do que se esperava, dentro de 50 gerações, o maior sensoriamento visual neste cenário não teve uma influência muito positiva nos testes realizados. Os gráficos da Figura 28 (quantidade de NPCs que alcançaram o *Player* por geração) e da Figura 29 (média da função da avaliação por geração) comparam as duas resoluções de visão, mostrando que as curvas se aproximam a partir da geração 25. A não regularidade das posições do *Player* em um mesmo local, dificultou o aprendizado pelo CVA de 50x50, visto que quanto mais entradas, maiores serão as topologias (fenótipos) dos indivíduos. Isso impacta diretamente nas mutações e *crossover* para geração de novas topologias. Observa-se, em ambos os gráficos, uma certa dificuldade inicial para evolução das gerações no CVA de 50x50 *pixels*. Talvez, a rede precise de um maior quantidade de gerações para o NPC se adaptar melhor à mudança constante no posicionamento do *Player*, sendo um novo caso a ser estudado em trabalhos futuros.

Figura 28 – Gráfico que compara a quantidade de NPCs que conseguiram chegar ao *Player* em cada geração para cada tipo de CVA.

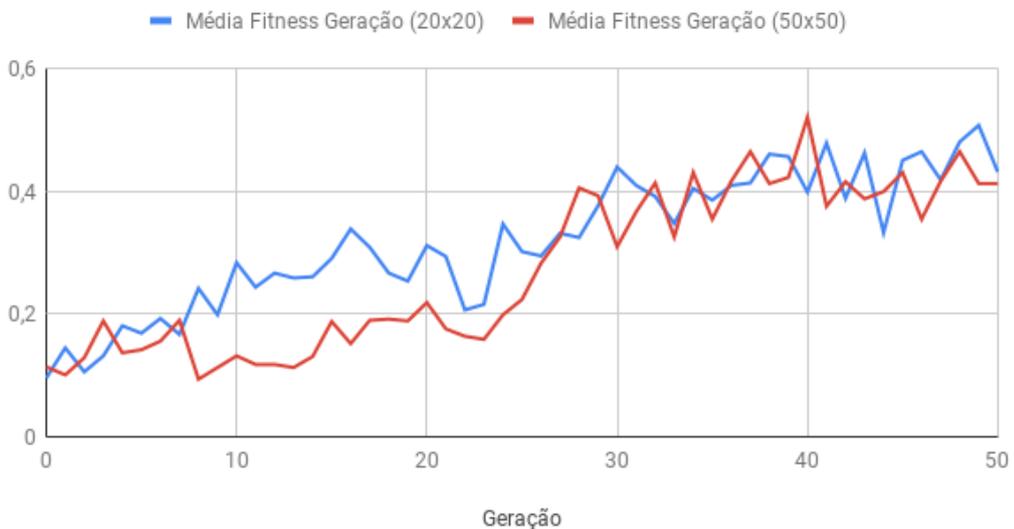
NPCs que conseguiram chegar ao Player (Cenário 2)



Fonte: Próprio Autor

Figura 29 – Gráfico que compara a média do *fitness* obtido por cada geração em diferentes CVAs.

Média Fitness por Geração (Cenário 2)



Fonte: Próprio Autor

4.3 Resultados do Cenário 3

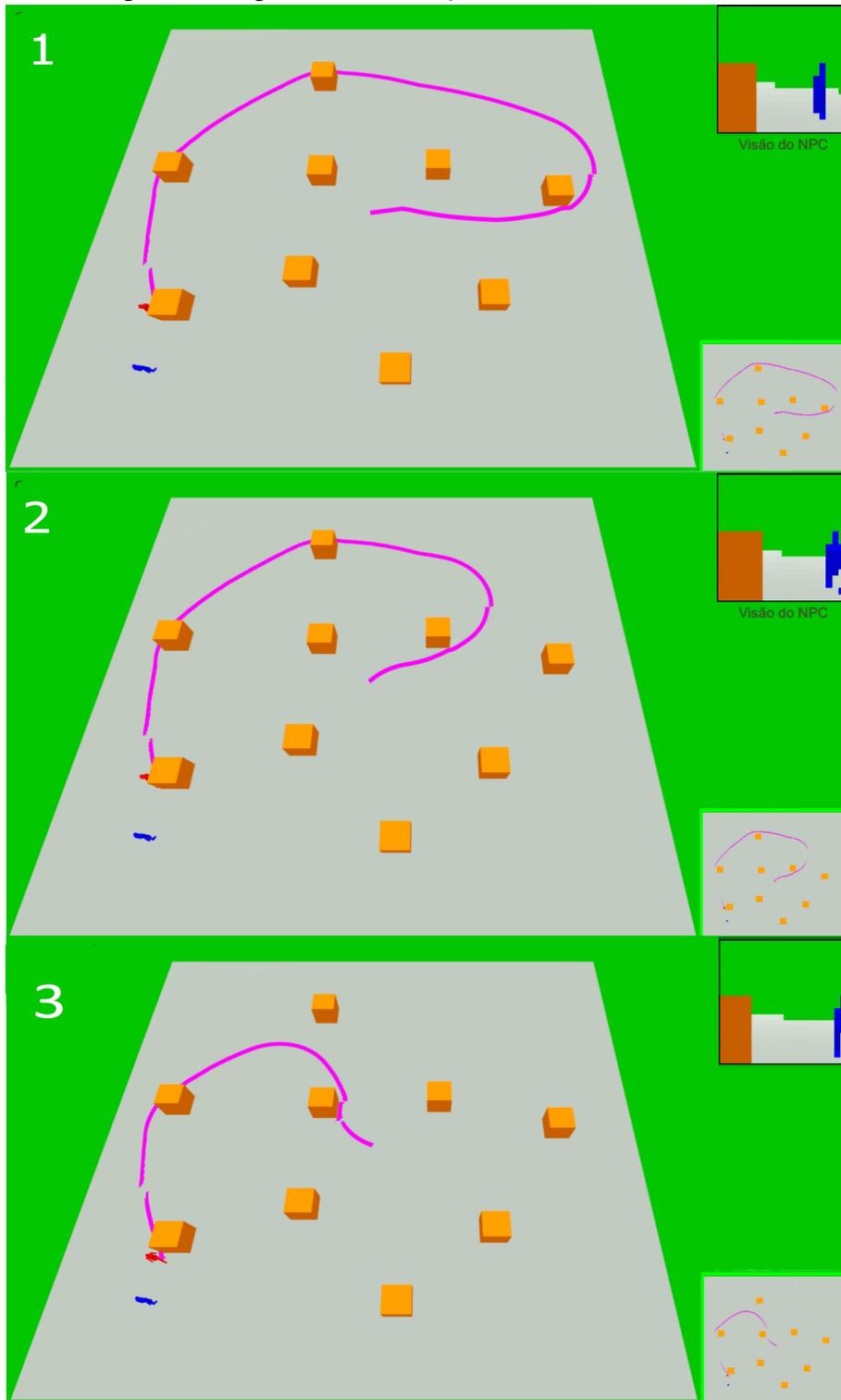
Neste cenário, o objetivo (*Player*) está fixo em um determinado canto da sala e nove obstáculos estão espalhados pela sala. Eventualmente, os obstáculos poderão encobrir a visualização do *Player* pelos CVAs. O objetivo é que o NPC procure pelo *Player* mesmo encoberto, adquirindo uma espécie de memória para guiá-lo até o objetivo. O objetivo é que NPC busque o *Player* sem necessariamente estar conseguindo visualizá-lo, adquirindo uma espécie de memória para guiá-lo até o objetivo.

4.3.1 CVA com 20x20 pixels - 1.200 entradas para a RNA

Os treinamentos utilizando este CVA mostraram comportamentos satisfatórios, como pode ser analisado na Figura 30. O NPC conseguiu se guiar pelo cenário, desviando de obstáculos e alcançando o *Player*. Cada quadro da Figura 30 mostra um caminho diferente percorrido pelo NPC treinado buscando o objetivo. Em todas as situações o NPC conseguiu aprender a contornar os obstáculos para alcançar o *Player*.

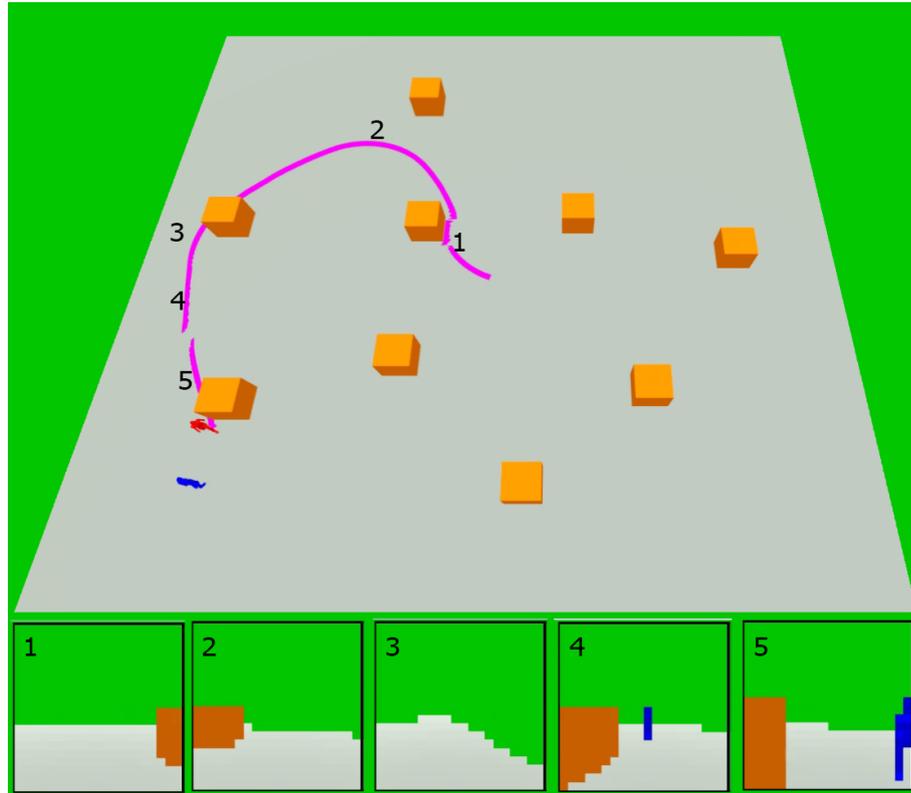
A Figura 31 mostra diferentes pontos de visão do NPC em direção ao *Player* durante uma das simulações realizadas.

Figura 30 – Cenário 3 (20x20) - A movimentação em cada quadro se refere a aos comportamentos gerados do NPC após o treinamento para o CVA de 20x20. O traço lilás indica o caminho percorrido por ele até o *Player*.



Fonte: Próprio Autor

Figura 31 – Detalhamento do caminho feito pelo NPC no quadro 3 da Figura 30, com as visões em 5 instantes diferentes da simulação.

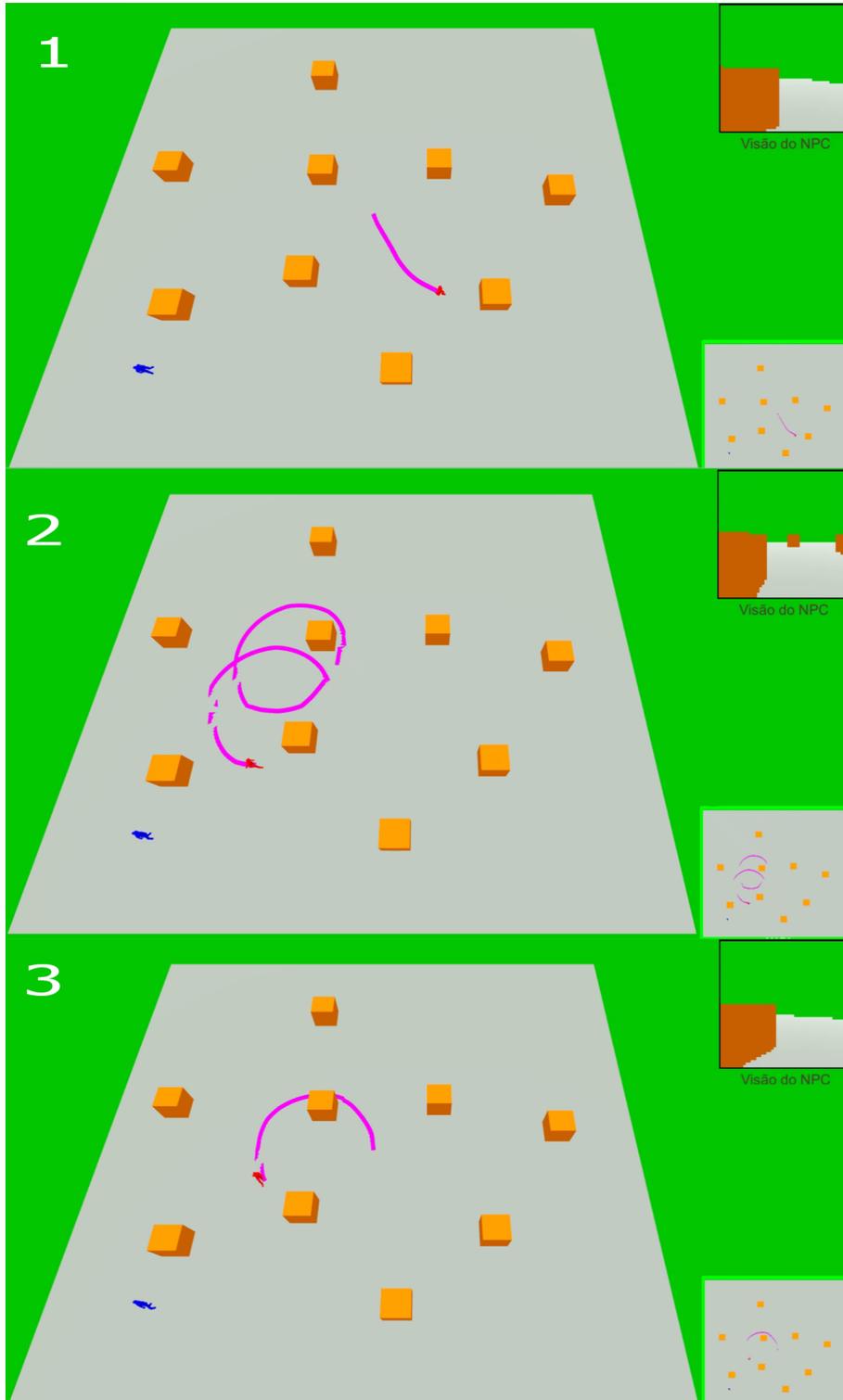


Fonte: Próprio Autor

4.3.2 CVA com 50x50 pixels - 7.500 entradas para a RNA

Para este CVA os resultados não foram satisfatórios considerando 50 gerações. Como pode ser observado na Figura 32, apesar do NPC ter encontrado o *Player* em vários momentos durante os treinamentos, ele não conseguiu aprender que os *pixels* azuis representavam o objetivo e nem aprender sobre a sua localização. O NPC treinado tinha uma tendência forte a fazer movimentos circulares. Em alguns momentos, mesmo visualizando os *pixels* azuis em alguns momentos, ele não conseguiu associá-los a ganhos de na função de avaliação. Assim como levantado no Cenário 2, suspeita-se que pelo tamanho da topologia deve ser necessário treinar mais gerações a fim de convergir a evolução genética para o resultado esperado.

Figura 32 – Cenário 3 (50x50) - A movimentação em cada quadro se refere aos comportamentos gerados do NPC após o treinamento para o CVA de 50x50. O traço lilás indica o caminho percorrido por ele.

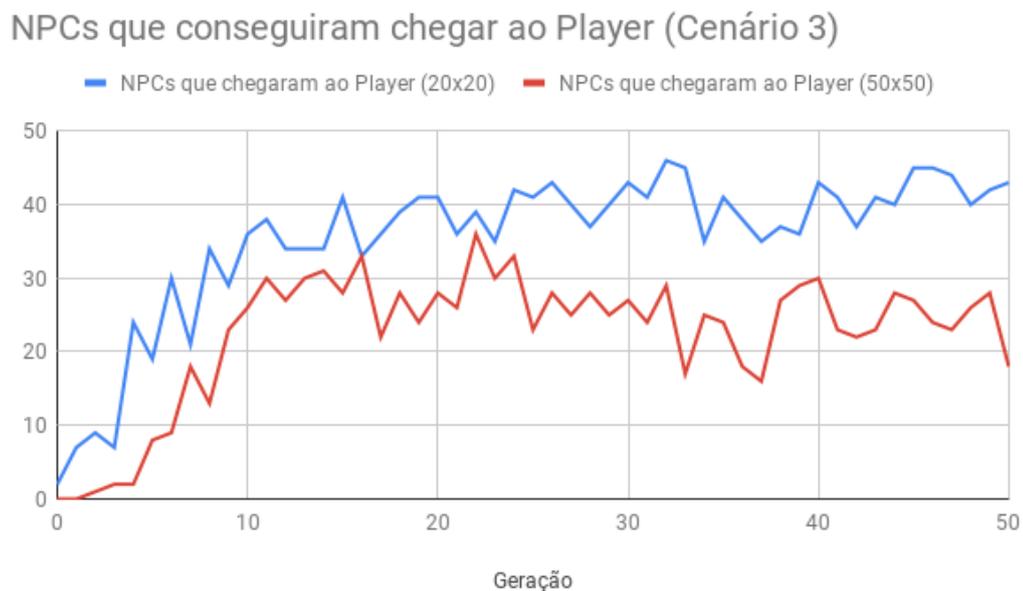


Fonte: Próprio Autor

4.3.3 Evolução em cada tipo de CVA

Os gráficos da Figura 33 e da Figura 34 ressaltam que o processo evolutivo do CVA de 20x20 pixels conseguiu bons resultados. O mesmo não ocorreu com o CVA de 50x50 pixels, provavelmente pelo fato de que com mais entradas, mais complexa fica a estrutura da rede e isso impacta diretamente nos processos de mutação e cruzamento dos indivíduos. As Figuras 33 e 34 mostram, respectivamente, os gráficos referentes aos NPCs que conseguiram chegar ao *Player* e a média do *fitness* por geração. Nota-se que para este cenário, um menor sensoramento visual foi o ideal, visto que a NEAT conseguiu adaptar-se e evoluir suas topologias com mais facilidade.

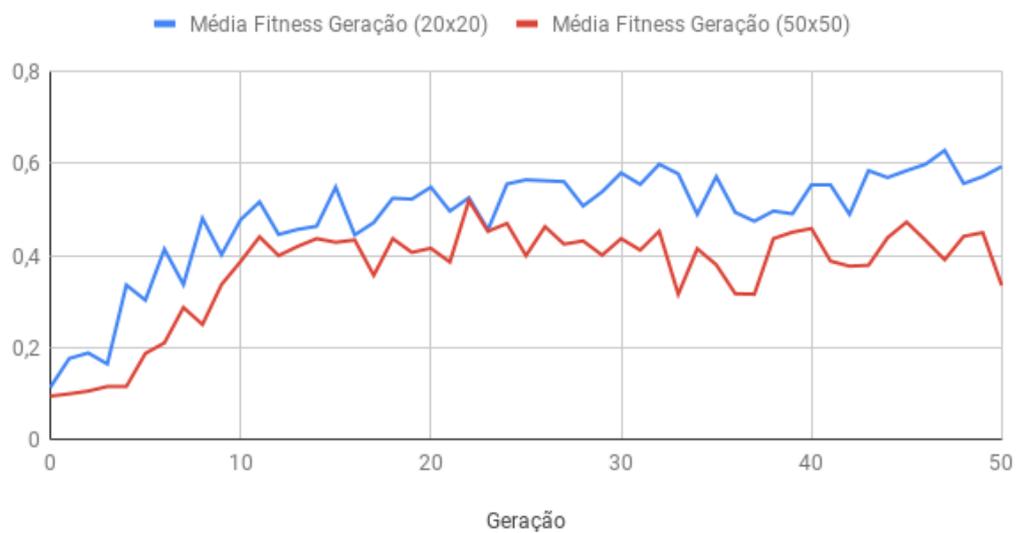
Figura 33 – Gráfico que compara a quantidade de NPCs que conseguiram chegar ao *Player* em cada geração para cada tipo de CVA.



Fonte: Próprio Autor

Figura 34 – Gráfico que compara a média do *fitness* obtido por cada geração em diferentes CVAs.

Média Fitness por Geração (Cenário 3)



Fonte: Próprio Autor

5 CONCLUSÕES E TRABALHOS FUTUROS

5.1 Conclusões

Neste trabalho, estudou-se a geração de comportamentos autônomos em NPCs utilizando CVAs de diferentes resoluções, buscando comparar as diferenças de desempenho e comportamentos. Para isso, foi utilizada toda a estrutura, codificação e evolução genética da NEAT, aplicando-a no treinamento de um NPC para que ele possa ir em direção a um *Player* (objetivo).

Os resultados obtidos em cada cenário de simulação mostraram-se, no geral, satisfatórios, conseguindo gerar NPCs capazes de se guiar até o *Player*. O CVA de 20x20 *pixels* demonstrou uma melhor adaptabilidade, conseguindo bons resultados em quase todos os cenários. Os testes mostraram que o CVA de 50x50 *pixels* não conseguiu realizar os caminhos e aprender a localização do *Player* de forma satisfatória. Nestes casos, o método evolutivo da NEAT sofreu o impacto da grande quantidade de entradas para rede, trabalhando com topologias bem mais complexas para cada indivíduo. A especiação, fixada em 10 para todos os cenários, também é uma variável a ser estudada em trabalhos futuros. Este valor significa que foi realizada a proteção de pelo menos 10 indivíduos no processo de cruzamento para que fossem mantidos na evolução representantes de cada espécie. É possível que, com uma maior especiação e um maior tempo de treinamento, o CVA de 50x50 *pixels* possa obter melhores resultados. Para as configurações e cenários propostos, a NEAT não conseguiu lidar com a grande quantidade de entradas e com a aleatoriedade envolvida nos cenários. Dessa forma, CVAs com resoluções muito maiores que as utilizadas se mostraram inviáveis.

A utilização de visão artificial como forma de sensoriamento se mostra uma abordagem eficaz para criar comportamentos autônomos em agentes e NPCs, apesar de que é preciso lidar com diversas questões envolvendo a diferenciação de elementos a partir de cores e analisar como definir uma função de avaliação que seja eficiente para este tipo de sensoriamento. Carneiro *et al.* (2014) e Nogueira (2014) utilizavam CVAs de baixa resolução para gerar comportamentos autônomos em robôs virtuais. Este trabalho mostrou que com o uso de uma rede mais robusta e mais informações visuais é possível obter melhorias nos comportamentos em determinados cenários de testes.

Para aplicar a proposta deste trabalho em um jogo real é necessário percorrer um longo caminho de amadurecimento da técnica, visto que existem muitas questões a serem

avaliadas em técnicas não determinísticas para jogos, principalmente quando o objetivo é criar NPCs com comportamentos verossímeis aos olhos do jogador. Mas, tais técnicas são promissoras, pelo fato de existir um aprendizado sobre o ambiente. Assim, as simulações vistas neste trabalho podem ser generalizadas para outros problemas relacionados a vida artificial e robótica, adaptando-se as devidas variáveis, como função de avaliação e cenários de treinamento.

5.2 Trabalhos Futuros

Pode-se destacar como propostas para trabalhos futuros:

- (a) Explorar o impacto das demais variáveis de configuração da NEAT nos cenários propostos, como o valor de especiação e a quantidade de gerações;
- (b) Utilizar RNAs aprimoradas para a utilização de grandes quantidades de valores de entrada, como a *framework* HyperNEAT (STANLEY *et al.*, 2009).
- (c) Adicionar diferentes motores de movimento e de ação, como pulo ou pegar objetos, buscando NPCs com enumerate mais complexos.

REFERÊNCIAS

- BARTLE, R. A. **Designing virtual worlds**. [S.l.]: New Riders, 2004.
- BEER, R. D. On the dynamics of small continuous-time recurrent neural networks. **Adaptive Behavior**, Sage Publications Sage CA: Thousand Oaks, CA, v. 3, n. 4, p. 469–509, 1995.
- BOURG, D. M.; SEEMANN, G. **AI for game developers**. [S.l.]: "O'Reilly Media, Inc.", 2004.
- CARNEIRO, L. d. O.; CAVAICANTE-NETOL, J. B.; VIDAL, C. A.; NOGUEIRA, Y. L.; NOVAL, A. B. V.; GENETICO, A. Geração de comportamentos emergentes de navegação em agentes autônomos utilizando visão artificial. 2014.
- CHANG, Y.-H.; MAHESWARAN, R. T.; LEVINBOIM, T.; RAJAN, V. Learning and evaluating human-like npc behaviors in dynamic games. In: **AIIDE**. [S.l.: s.n.], 2011.
- COPPIN, B. **Inteligência artificial**. [S.l.]: Grupo Gen-LTC, 2013.
- DIGNUM, F.; WESTRA, J.; DOESBURG, W. A. van; HARBERS, M. Games and agents: Designing intelligent gameplay. **International Journal of Computer Games Technology**, Hindawi, v. 2009, 2009.
- FENG, S.; TAN, A.-H. Towards autonomous behavior learning of non-player characters in games. **Expert Systems with Applications**, Elsevier, v. 56, p. 89–99, 2016.
- GOMES, R. C. L. F. *et al.* Agentes verossímeis: uma investigação sobre a construção dos personagens autônomos nos videogames. Pontifícia Universidade Católica de São Paulo, 2008.
- HAYKIN, S. **Redes neurais: princípios e prática**. [S.l.]: Bookman Editora, 2007.
- JALLOV, D.; RISI, S.; TOGELIUS, J. Evocommander: A novel game based on evolving and switching between artificial brains. **IEEE Transactions on Computational Intelligence and AI in Games**, IEEE, v. 9, n. 2, p. 181–191, 2017.
- KEMPKA, M.; WYDMUCH, M.; RUNC, G.; TOCZEK, J.; JAŚKOWSKI, W. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In: IEEE. **Computational Intelligence and Games (CIG), 2016 IEEE Conference on**. [S.l.], 2016. p. 1–8.
- LOIZOU, M.; HARTLEY, T.; SLATER, S.; NEWMAN, R.; PANNESE, L. Emotions for intelligent agents in crisis management simulations: A survey. In: IEEE. **Computer Games (CGAMES), 2012 17th International Conference on**. [S.l.], 2012. p. 213–219.
- LOYALL, A. B. **Believable Agents: Building Interactive Personalities**. [S.l.], 1997.
- MADSEN, C. A. B. C. W.; ADAMATTI, D. F. Using artificial neural networks in npc decision-making process. 2013.
- MATTIUSSI, C.; FLOREANO, D. Analog genetic encoding for the evolution of circuits and networks. **IEEE Transactions on evolutionary computation**, IEEE, v. 11, n. 5, p. 596–607, 2007.
- MIGUEL, C. G. **Evolução Estrutural e Paramétrica de Redes Neurais Dinâmicas em Vida Artificial**. Tese (Doutorado) — Dissertação de mestrado acadêmico em Engenharia Elétrica. Escola Politécnica da Universidade de São Paulo, 2009.

- MIND Deep. **Playing Atari with Deep Reinforcement Learning**. 2013. Disponível em: <<https://deepmind.com/research/publications/playing-atari-deep-reinforcement-learning/>>. Acesso em: 9 nov. 2018.
- MIT. **Spacewar! | The MIT 150 Exhibition**. 2011. Disponível em: <<http://museum.mit.edu/150/25>>. Acesso em: 18 jun. 2018.
- MONDADA, F.; FRANZI, E.; GUIGNARD, A. The development of khepera. In: **Experiments with the Mini-Robot Khepera, Proceedings of the First International Khepera Workshop**. [S.l.: s.n.], 1999. p. 7–14.
- NOGUEIRA, Y. L. B. Integração mente e ambiente para geração de comportamentos emergentes em personagens virtuais através da evolução de redes neurais artificiais. Universidade Federal do Ceará, 2014.
- NOGUEIRA, Y. L. B.; BRITO, C. E. F. de; VIDAL, C. A.; NETO, J. B. C. Evolving plastic neuromodulated networks for behavior emergence of autonomous virtual characters. In: **ECAL**. [S.l.: s.n.], 2013. p. 577–584.
- NORVIG, P.; RUSSELL, S. **Inteligência Artificial: Tradução da 3a Edição**. [S.l.]: Elsevier Brasil, 2014. v. 1.
- OPENAI. **Gym**. 2016. Disponível em: <<https://gym.openai.com>>. Acesso em: 18 jun. 2018.
- OPENAI. **Universe**. 2016. Disponível em: <<https://blog.openai.com/universe>>. Acesso em: 18 jun. 2018.
- QUANDT, T.; KRÖGER, S. **Multiplayer: The social aspects of digital gaming**. [S.l.]: Routledge, 2013. v. 3.
- REZENDE, S. O. **Sistemas inteligentes: fundamentos e aplicações**. [S.l.]: Editora Manole Ltda, 2003.
- RUSSELL, S. J.; NORVIG, P. **Inteligencia Artificial: un enfoque moderno**. [S.l.: s.n.], 2004.
- SAYGIN, A. P.; CICEKLI, I.; AKMAN, V. Turing test: 50 years later. **Minds and Machines**, Springer, v. 10, n. 4, p. 463–518, 2000.
- SILVA, L. G. S. Aplicando a neuroevolução para obter npcs de alta qualidade. 2015.
- STANLEY, K. O.; D’AMBROSIO, D. B.; GAUCI, J. A hypercube-based encoding for evolving large-scale neural networks. **Artificial life**, MIT Press, v. 15, n. 2, p. 185–212, 2009.
- STANLEY, K. O.; MIIKKULAINEN, R. Evolving neural networks through augmenting topologies. **Evolutionary computation**, MIT Press, v. 10, n. 2, p. 99–127, 2002.
- TURING, A. M. Computing machinery and intelligence. In: **Parsing the Turing Test**. [S.l.]: Springer, 2009. p. 23–65.
- UNITY. **Unity AI Reinforcement Learning with Q-Learning**. 2017. Disponível em: <https://blogs.unity3d.com/2017/08/22/unity-ai-reinforcement-learning-with-q-learning/?_ga=2.216462038.1334117804.1529341091-2127128803.1521747161>. Acesso em: 18 jun. 2018.

UNITY. **Unity-Technologies/ml-agents: Unity Machine Learning Agents Toolkit**. 2018. Disponível em: <<https://github.com/Unity-Technologies/ml-agents>>. Acesso em: 18 jun. 2018.

YANNAKAKIS, G. N.; TOGELIUS, J. **Artificial Intelligence and Games**. [S.l.]: Springer, 2017.

YILDIRIM, S.; STENE, S. B. A survey on the need and use of ai in game agents. In: SOCIETY FOR COMPUTER SIMULATION INTERNATIONAL. **Proceedings of the 2008 Spring simulation multiconference**. [S.l.], 2008. p. 124–131.