

# Desafiando o Jogador: Aplicação de Algoritmos Genéticos à *Non-Player Characters*

Matheus Silva, Saulo Popov Zambiasi

Universidade do Sul de Santa Catarina (UNISUL)  
Tubarão – SC – Brazil

matheus.silva8001@gmail.com, saulopz@gmail.com

**Abstract.** *In electronic games, characters that are not controlled by the players have to give an impression of intelligence to increase the players' immersion and enjoyment. An alternative is the use of genetic algorithms, based on Darwin's theories in which individuals more adapted to the environment are more likely to pass on their genes for the perpetuation of the species. Thus, the present work aims to apply the genetic algorithm in the evolution of an army of NPC's, with the intention that the AI of the game itself, develop ways to make combat with the player more dynamic, presenting more challenges to the player.*

**Resumo.** *Nos jogos eletrônicos, os personagens que não são controlados pelos jogadores tem que passar uma impressão de inteligência para aumentar a imersão e diversão dos jogadores. Uma alternativa é a utilização de algoritmos genéticos, baseando-se nas teorias de Darwin em que os indivíduos mais adaptados ao ambiente têm mais chances de passar seus genes adiante para a perpetuação da espécie. Desta forma o presente trabalho, busca aplicar o algoritmo genético na evolução de um exército de NPCs, com o intuito de que a própria IA do jogo, desenvolva formas de deixar o combate com o jogador mais dinâmico apresentando mais desafios ao jogador.*

## 1. Introdução

A revolução da inteligência artificial nos jogos iniciou-se por volta dos anos 80 com a criação e disseminação dos *arcades*<sup>1</sup>, com jogos tais como *Pacman* da Namco<sup>2</sup> e *Space Invaders* da Taito<sup>3</sup>. Esses se utilizam regras simples de estímulo-resposta que se baseiam no estado em que o personagem se encontra e sua resposta a esse estado. Dessa forma, eles acabaram criando uma incipiente inteligência nos *Non-Player Characters* (NPCs), com a finalidade, nesse caso, de simular a sensação de perseguição para o jogador. Com a popularização dos jogos e a evolução dos hardwares de processamento, os jogadores passaram a ser cada vez mais exigentes e, segundo Benin (2007), “a inteligência artificial tornou-se fator crítico para o sucesso em jogos de computador”.

Levando em consideração essa exigência dos jogadores, um dos algoritmos que podem ser utilizados com essa finalidade é o Algoritmo Genético, cujo objetivo é inspirar-se na natureza fazendo analogia a seleção natural e a genética evolutiva. Esse algoritmo busca criar estruturas semelhantes a cromossomos para aplicar o processo

---

<sup>1</sup> No Brasil conhecido popularmente por “Fliperama”

<sup>2</sup> Conhecida como Bandai Namco, e uma publicadora japonesa de jogos eletrônicos

<sup>3</sup> Taito é uma produtora e distribuidora de jogos eletrônicos do Japão pertencente à Square Enix.

para a busca: seleção, cruzamento e mutação (PALIOTO e COLANZI, 2020). Esses cromossomos carregam genes e, por meio deles, são selecionadas as características para a próxima geração, com o intuito de superar um desafio estabelecido.

Sob esse contexto, o presente trabalho tem como objetivo testar e avaliar a evolução de NPCs em jogos digitais, por meio do uso de algoritmos evolucionários, implementando assim um protótipo para testes, avaliação e demonstração da eficácia e dos benefícios que o algoritmo genético pode trazer aos jogos.

O presente trabalho está dividido nas seguintes seções: inicialmente é apresentada a problemática e proposta; na seção 2 são apresentados alguns embasamentos com base na proposta; na seção 3 são apresentados alguns pontos da implementação e discussões dos resultados; por fim, têm-se as considerações finais.

## **2. Revisão Bibliográfica**

Nesta seção é apresentada uma introdução aos assuntos abordados, incluindo um breve resumo sobre o conceito de Inteligência Artificial, algoritmos genéticos e uma simples explicação de sua implementação. Depois é explicado o que são jogos e em que situação o objetivo do projeto está inserido.

### **2.1. Inteligência Artificial**

A inteligência artificial(IA) é uma tentativa de criação de algoritmos que se aproximem do comportamento animal e humano a partir do processamento computacional. É possível tomar exemplos ficcionais sobre IAs, como nas obras de Isaac Asimov, um dos pais da ficção científica, que em sua bibliografia discorre sobre as possibilidades produtivas de inteligências artificiais. Entretanto, possui um entendimento filosófico sobre o tema e seus resultados, a exemplo dentro da reunião de contos *Eu Robô* (ASIMOV, 2015), no qual aborda a evolução da IA desde simples processos mecânicos, até administração governamental.

John McCarthy (apud, RUSSEL e NORVIG, 2014) fala que a inteligência Artificial, "É a ciência e a engenharia de fabricar máquinas inteligentes, especialmente programas de computador inteligentes. Ela está relacionada à tarefa semelhante de usar computadores para entender a inteligência humana, mas a IA não precisa se limitar aos métodos biologicamente observáveis". Dentro disso, em 1947 Alan Turing, por meio de palestras, propôs ideias sobre IAs na Sociedade Matemática de Londres, a qual foram, em sua época, as falas mais influentes. Após isso, em 1950 ele publicou o artigo "Computing Machinery and Intelligence", que apresenta ao mundo o teste de Turing, aprendizagem de máquina, aprendizagem por reforço e algoritmos genéticos. (RUSSEL e NORVIG, 2014).

O aprendizado de máquina é feito a partir de uma base de dados e um gabarito, com uma entrada de dados a IA é questionada e sua resposta é comparada ao gabarito. Logo, seu aprendizado é feito a partir de comparações para que a resposta da entrada seguinte seja cada vez mais precisa e correta.

Alternativamente, o aprendizado por reforço é feito sem a presença de um gabarito, sua única supervisão será uma recompensa, uma nota para sua resposta, tal

qual uma prova. Dessa forma, o objetivo da IA é maximizar essa recompensa, e esse é o seu único retorno para um aprendizado.

## **2.2. Algoritmo Genético**

Algoritmos genéticos, mais especificamente, buscam resolver problemas utilizando estruturas semelhantes a cromossomos, os quais são influenciados pela recombinação genética para assegurar a integridade de informações valiosas. (PALIOTO; COLANZI, 2020).

Segundo Darwin, aqueles organismos que melhor se adaptam ao ambiente em que vivem possuem maiores chances de sobreviver e deixar descendentes. Para Lima e Cabral (2018), “o cruzamento entre organismos bem adaptados é a melhor oportunidade de passar para as próximas gerações os atributos desejáveis, enquanto o cruzamento entre os organismos menos adaptados garante a diversidade populacional.”

Algoritmos genéticos são altamente estimados para solucionar problemas que necessitam de uma boa eficiência na busca pelos melhores resultados. Isso porque eles possuem uma característica notável, que os destaca dos demais, os algoritmos evolutivos resultam em uma população de soluções viáveis, enquanto os métodos alternativos obtêm apenas uma solução final. (IYODA e MASATO, 2000). Dentro dos algoritmos genéticos todos os candidatos a solução são denominados organismos, e um conjunto de organismos é denominado população. Todos os organismos são formados por cromossomos, esses que consistem em um conjunto de características, o qual cada uma é considerada como um gene. (LIMA e CABRAL, 2018).

As implementações mais comuns de AG (algoritmos genéticos) iniciam com uma população aleatória, essa que possui um tamanho determinado. Ainda em alguns casos existe uma taxa de mutação, geralmente sendo de 1%, que permite uma maior variedade de resultados. Ainda em alguns casos existe uma taxa de mutação, geralmente sendo de 1%, que permite uma maior variedade de resultados. (LIMA e CABRAL, 2018). Esse processo se repete, criando uma geração nova a partir dos melhores organismos da população anterior, assim permitindo um tipo de seleção natural de genes, o qual encerra-se apenas quando o critério de parada é atingido, podendo-se obter o resultado desejado ou atingir o número máximo de gerações estipulado. (STELTTER *et al.*, 2016).

## **2.3. Jogos e Computação Evolutiva**

As pesquisas na área de IA para jogos vêm crescendo muito desde o grande marco da Inteligência Artificial, quando o Deep Blue, um software que utiliza IA, derrotou o campeão mundial de xadrez Garry Kasparov, o match em questão foi jogado em 1997 com 6 partidas, o qual resultou na pontuação de 3,5 a 2,5 para Deep Blue. (OLIVEIRA, 2018). Dessa forma, os jogos conseguem atrair a atenção de outras pessoas para a área de estudos sobre IA.

Com o passar do tempo, a criação de jogos aprimorou-se consideravelmente, conforme os gráficos desenvolveram-se adjuntamente a capacidade de processamento dos dispositivos, conseqüentemente surgiu a necessidade de imersão com histórias

acontecendo em tempo real para o jogador (SOUZA e VAHLICK, A, 2013). Dentro disso, um dos pioneiros na utilização de IA em jogos é o Civilization, de 1991. Visto que, ele utiliza a inteligência artificial para simular decisões governamentais e sociais, com intuito de imergir o jogador a acreditar que está controlando uma civilização no processo desenvolvimentista ao longo da história. Para isso é necessário que a IA tome decisões críveis e alinhadas às doutrinas ideológicas de cada civilização não controlada pelo jogador, assim alimentando o teatro virtual do jogo, esse que tem como seu objetivo último entreter o jogador.

Alguns jogos consistem em controlar vários grupos de personagens, montar táticas de defesa e ataque, então tanto o jogador quanto a máquina, precisavam ter essa capacidade. O tipo de jogo que mais demonstrou os métodos de simular uma vida para os NPCs, foram os simuladores de gestão. Um exemplo é o RimWorld, criado pela Ludeon Studios em 2018, o qual simula uma colônia em um planeta todo controlado por uma IA. Ademais, essa inteligência artificial é uma contadora de histórias, segundo a própria criadora Ludeon Studios, pois também gerencia os efeitos climáticos, modificações geográficas, encontros aleatórios, outras comunidades e suas relações com o jogador, invasões periódicas e uma economia dinâmica dentro do planeta. Esse acúmulo de funções em uma unidade de inteligência acaba demonstrando sua evolução do original civilization de 91 até o que é possível alcançar nos dias atuais.

O principal benefício da IA nos jogos, é aumentar a diversão, pois como colocado por Kishimoto (2004):

Os personagens de um jogo devem simular inteligência e erros humanos e ter personalidades, devem ser capazes de fornecer diferentes níveis de dificuldade ao jogador, para que o mesmo se sinta desafiado. Além disso, os jogadores cada vez mais demandam melhores oponentes em jogos mais complexos (KISHIMOTO, 2004).

Além disso, a utilização da IA também contribui no desenvolvimento de jogos trazendo vários benefícios, principalmente aumentando a imersão em situações como: auxiliar na navegação da câmera e permitir aos personagens do jogo aprenderem um comportamento mais semelhante à vida real. Ainda assim existem alguns problemas, como o fato de que para a utilização da melhor forma de uma IA, é necessário um maior tempo de desenvolvimento, ou o fato de também dificultar os testes, pois os resultados não são previsíveis. (KISHIMOTO, 2004).

Um trabalho que se utilizou de algoritmos genéticos, buscando a solução de algum problema envolvendo a IA em jogos eletrônicos, foi o de Benin (2007). O qual um sistema é implementado com o intuito de melhorar o comportamento dos NPCs, para isso é simulada uma comunidade de Ciclopes que se mantém viva e aprimorada com base na utilização de um algoritmo genético.

O que mais se aproxima do presente trabalho, é o feito por Júnior e Lima (2010), onde é utilizado o algoritmos genéticos juntamente com computação evolutiva, com a finalidade de criar uma arena que NPCs possam ser desafiadores ao jogador. Essa arena contava com inimigos previamente definidos, que aprendiam a vencer a tática utilizada pelo jogador na última batalha.

### 3. Desenvolvimento

Este trabalho utiliza como base um jogo no estilo de jogos de sobrevivência, no qual o jogador controla três personagens que podem ser customizados quanto aos seus equipamentos e são atacados constantemente por hordas de inimigos controlados pela inteligência artificial. Ao final de cada batalha a IA deve se adaptar às estratégias empregadas pelo jogador na rodada, assim formulando novas táticas e modificando os equipamentos dos inimigos. Dessa forma, aumentando a dificuldade do desafio que o jogador deve enfrentar para que seus personagens sobrevivam.

Para iniciar a criação do protótipo é necessário definir os genes dos indivíduos. Como o objetivo do protótipo é demonstrar a adaptabilidade dos inimigos utilizando IA, a vida dos indivíduos não faz parte do gene, permanecendo fixa em 50 caso seja um inimigo, se for um jogador fica definida como três vezes maior. Os genes presentes são opções de armas e de armaduras e cada um contando com três opções, sendo essas, três opções de armaduras e mais três de armas; dentre elas a armadura de couro, completa e cota de malha; dentre as armas martelo, lança e espada longa. Conforme visto na Figura 1, é possível ver alguns tipos de operações que podem ser efetuadas pela classe do indivíduo.

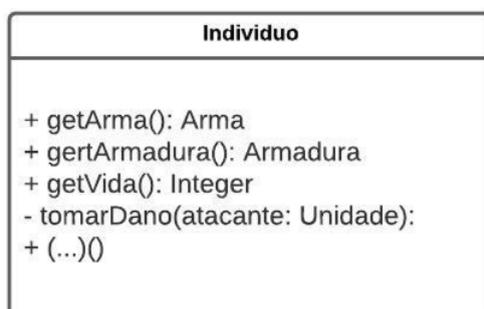
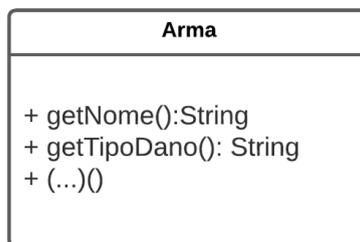


Figura 1: Representação da classe do Indivíduo.

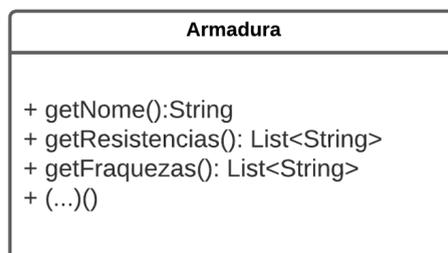
Cada gene do indivíduo possui características que alteram drasticamente o combate possibilitando várias adaptações por parte da IA e do jogador, que tenham como objetivo vencer a peleja. Cada arma possui apenas um tipo de dano entre: cortante, esmagamento ou perfurante, que são características cruciais na escolha do equipamento e na formação da tática. Isso porque os tipos de dano são aplicados para as resistências e fraquezas das armaduras, que caso sejam fraquezas o personagem recebe o dobro do dano base e, alternativamente, em caso de resistência recebem metade do dano base.

A Figura 2 apresenta a representação das operações da classe das Armas utilizadas.



*Figura 2: Representação da classe Arma.*

A Figura 3 mostra a representação das possíveis operações da classe da armadura dos personagens.



*Figura 3: Representação da classe Armadura.*

No início é preciso definir os indivíduos que farão parte da equipe do jogador, para isso o jogador deve selecionar uma armadura e uma arma para cada indivíduo (Figura 4). Após o jogador montar seus personagens, são solicitadas as informações de quantidade de inimigos por rodada, e quantidade de gerações (Figura 8), o qual serão utilizadas para o controle e sistema de parada do algoritmo genético. Após as tomadas de decisão e a composição estratégica do jogador, são geradas as unidades iniciais da máquina para combater os personagens criados do jogador. Durante o combate, os inimigos buscarão neutralizar o primeiro personagem até derrotá-lo, depois o segundo e assim sucessivamente. Os personagens do jogador seguirão esse padrão de ação para a luta.

Como o protótipo simula um jogo de sobrevivência em hordas, os personagens do jogador começam atacando, visto que já estavam esperando o ataque inimigo. Quando uma unidade efetua um ataque contra outra são verificados os genes, os quais tipificam o dano da arma e os tipos de resistências e fraquezas da armadura. Nas regras dessa peleja virtual, o dano base das armas é 2, aplicado quando a armadura não possui resistência nem fraqueza para o tipo de dano da arma equipada, caso a armadura possua resistência, o atacante causa apenas metade do dano, entretanto se a armadura apresentar fraqueza contra o tipo de dano da arma, o dano causado é o dobro do dano base. Essas tipificações de danos, com suas resistências e fraquezas, aumentam a variabilidade de táticas que o jogador e a inteligência artificial podem adotar para sobrepujar-se um ao outro.

```

ESCOLHA A ARMADURA DO INDIVIDUO 01

|-----|-----|-----|
| Digite 1 para selecionar | Digite 2 para selecionar | Digite 3 para selecionar |
|-----|-----|-----|
| Armadura: Couro | Armadura: Cota de Malha | Armadura: Completa |
| Resistencias: [ESMAGAMENTO] | Resistencias: [CORTANTE] | Resistencias: [CORTANTE, PERFURANTE] |
| Fraquesas: [PERFURANTE, CORTANTE] | Fraquesas: [PERFURANTE] | Fraquesas: [ESMAGAMENTO] |
|-----|-----|-----|

2
ESCOLHA A ARMA DO INDIVIDUO 01

|-----|-----|-----|
| Digite 1 para selecionar | Digite 2 para selecionar | Digite 3 para selecionar |
|-----|-----|-----|
| Arma: Espada Longa | Arma: Martelo | Arma: Lança |
| Tipo Dano: CORTANTE | Tipo Dano: ESMAGAMENTO | Tipo Dano: PERFURANTE |
| Distância de Acerto: 1 | Distância de Acerto: 1 | Distância de Acerto: 2 |
|-----|-----|-----|

2

```

Figura 4: Exemplo da criação de um indivíduo.

A Figura 5 apresenta um exemplo de inicialização das definições para iniciar o processamento do algoritmo genético.

```

Insira a quantidade de inimigos por geração:
8
Insira a quantidade de gerações:
500

```

Figura 5: Exemplo das definições para o algoritmo genético.

Após cada combate, a IA efetua simulações de cenários de combate, mantendo assim a metade sobreviventes em caso de derrota, essa metade se reproduz aleatoriamente na população. Dentro disso, o sistema de reprodução cada pai simula uma batalha de 1 contra 1 contra cada unidade controlada pelo jogador, aquele que tiver mais vitórias transfere o gene arma para o filho, e o gene armadura é aleatoriamente de um dos pais. No processo de reprodução dos sobreviventes, os NPCs possuem uma chance de mutação de 1% em seus genes, que caso se concretize a arma e armadura do indivíduo são substituídas aleatoriamente, aumentando, assim, a diversidade da população. Uma representação do andamento do algoritmo pode ser visto na figura 6, da qual apresenta um exemplo do sistema de produção dos filhos do algoritmo genético durante a sua execução.

```

Individuo pai01 = individuosMaquina.get(random1);
Individuo pai02 = individuosMaquina.get(random2);
List<Individuo> filhos = this.cruzamento(pai01,pai02, getListaIndividuos(individuosJogador), individuosMaquina.get(individuosMaquina.size()-1).getId());
individuosMaquina.addAll(filhos);

```

Figura 6: Exemplo do sistema de reprodução.

Na figura 7 é apresentado um trecho do código em linguagem de programação Java referente ao algoritmo de mutação do programa.

```
public List<Individuo> efetuarMutacao(List<Individuo> individuos) {
    int chanceMutacao = 1;
    for (Individuo individuo : individuos) {
        if (ThreadLocalRandom.current().nextInt( bound: 100) == chanceMutacao) {
            individuo.setArma(armaMutacao());
            individuo.setArmadura(armaduraMutacao());
        }
    }
    return individuos;
}
```

*Figura 7: Exemplo do sistema de mutação.*

Por fim, após a quantidade de gerações estipulada, um novo combate é realizado entre as novas unidades da IA contra as unidades do jogador. A finalidade desse combate não é a supremacia da IA sobre o jogador depois de uma rodada de evolução, mas sim que ocorra uma escalada de dificuldade entre as lutas. Por conseguinte, será exigido que o jogador adapte suas estratégias contra a máquina, assim as táticas de combate mais eficientes sempre serão modificadas entre as jogatinas. Visto que essa adaptabilidade do algoritmo gerará uma mudança no paradigma de eficiência dos genes no combate, fato que, em última análise, exigirá do jogador um fator de microgerenciamento e previsão sobre suas próprias estratégias. Dessa forma, a complexidade das táticas será elevada pela presença da inteligência artificial moldando-se de forma equilibrada e dinâmica e não devido a um aumento de atributos.

Na figura 8 está demonstrado os dados de controle do Algoritmo Genético, que serão utilizados para o combate, consequentemente o aprendizado da IA.

```
Insira a quantidade de inimigos por geração:
7
Insira a quantidade de gerações:
1000
```

*Figura 8: Exemplo do sistema de controle do Algoritmo Genético.*

A figura 9 apresenta os personagens customizados pelo jogador, prontos para um combate. Nesse caso, o personagem 1 possui um martelo que causa dano do tipo “esmagamento” e uma armadura que possui resistências e fraquezas, essas que são resistência a “cortante” e “perfurante”, ao mesmo tempo que possui fraqueza ao tipo “esmagamento”. Esse indivíduo exemplifica o que são os genes no algoritmo, características que, nessa situação, são eletivas do jogador.

```

| Unidade: 1 |
| Vida: 150 |
| Arma: Martelo ( Tipo do dano ESMAGAMENTO ) |
| Armadura: Completa ( Resistencia aos tipos: [CORTANTE, PERFURANTE] ) ( Fraquezas aos tipos: [ESMAGAMENTO] ) |

| Unidade: 2 |
| Vida: 150 |
| Arma: Lança ( Tipo do dano PERFURANTE ) |
| Armadura: Couro ( Resistencia aos tipos: [ESMAGAMENTO] ) ( Fraquezas aos tipos: [PERFURANTE, CORTANTE] ) |

| Unidade: 3 |
| Vida: 150 |
| Arma: Lança ( Tipo do dano PERFURANTE ) |
| Armadura: Cota de Malha ( Resistencia aos tipos: [CORTANTE] ) ( Fraquezas aos tipos: [PERFURANTE] ) |

```

*Figura 9: Exemplo de indivíduos montados pelo usuário.*

Na figura 10 estão dispostos os NPCs criados pela IA, os quais são inicialmente gerados aleatoriamente, utilizando-se dos genes disponíveis, esses que representam as características. Dessa forma, os equipamentos “arma” e “armadura” são os genes que compõem um indivíduo da geração, conseqüentemente os equipamentos mais bem sucedidos, ou melhor adaptados à estratégia do jogador, serão mantidos e herdados na próxima geração.

```

| Unidade: 4 |
| Vida: 50 |
| Arma: Espada Longa ( Tipo do dano CORTANTE ) |
| Armadura: Couro ( Resistencia aos tipos: [ESMAGAMENTO] ) ( Fraquezas aos tipos: [PERFURANTE, CORTANTE] ) |

| Unidade: 5 |
| Vida: 50 |
| Arma: Espada Longa ( Tipo do dano CORTANTE ) |
| Armadura: Completa ( Resistencia aos tipos: [CORTANTE, PERFURANTE] ) ( Fraquezas aos tipos: [ESMAGAMENTO] ) |

| Unidade: 6 |
| Vida: 50 |
| Arma: Lança ( Tipo do dano PERFURANTE ) |
| Armadura: Cota de Malha ( Resistencia aos tipos: [CORTANTE] ) ( Fraquezas aos tipos: [PERFURANTE] ) |

| Unidade: 7 |
| Vida: 50 |
| Arma: Martelo ( Tipo do dano ESMAGAMENTO ) |
| Armadura: Cota de Malha ( Resistencia aos tipos: [CORTANTE] ) ( Fraquezas aos tipos: [PERFURANTE] ) |

```

*Figura 10: Exemplo de indivíduos gerados randomicamente para a IA.*

Já na figura 11 é retratado o resultado do primeiro combate entre as unidades do jogador e da IA. Esse combate acabou gerando o resultado de vitória do jogador, que manteve duas unidades vivas. Após essa derrota, o algoritmo irá adaptar-se aos genes do jogador, de forma a reproduzir-se pelo número de gerações determinadas na figura 8, com o intuito de superar a estratégia pregressa do jogador que a derrotou.

```

----- BATALHA INICIAL -----
O JOGADOR GANHOU!!!
Restaram vivos: 2 unidades

| Unidade: 2 |
| Vida: 65 |
| Arma: Lança ( Tipo do dano PERFURANTE ) |
| Armadura: Couro ( Resistencia aos tipos: [ESMAGAMENTO] ) ( Fraquezas aos tipos: [PERFURANTE, CORTANTE] ) |

| Unidade: 3 |
| Vida: 150 |
| Arma: Lança ( Tipo do dano PERFURANTE ) |
| Armadura: Cota de Malha ( Resistencia aos tipos: [CORTANTE] ) ( Fraquezas aos tipos: [PERFURANTE] ) |

```

*Figura 11: Resultado da batalha entre os indivíduos da IA e do jogador.*

Na figura 12 está disposto o novo exército da IA, após toda a evolução. Essa que ocorreu com a simulação do combate anterior a cada nova geração de indivíduos, a fim de obter o melhor resultado possível. Gerações de seleção genética com uma única finalidade, derrotar o jogador e sobrepor a estratégia vitoriosa. Após a derrota, a inteligência começa a atuar, simulando essa sociedade que possui um objetivo em mente, vitória sobre o jogador.

```

| Unidade: 3004 |
| Vida: 50 |
| Arma: Martelo ( Tipo do dano ESMAGAMENTO ) |
| Armadura: Completa ( Resistencia aos tipos: [CORTANTE, PERFURANTE] ) ( Fraquezas aos tipos: [ESMAGAMENTO] ) |

| Unidade: 3005 |
| Vida: 50 |
| Arma: Martelo ( Tipo do dano ESMAGAMENTO ) |
| Armadura: Completa ( Resistencia aos tipos: [CORTANTE, PERFURANTE] ) ( Fraquezas aos tipos: [ESMAGAMENTO] ) |

| Unidade: 3006 |
| Vida: 50 |
| Arma: Martelo ( Tipo do dano ESMAGAMENTO ) |
| Armadura: Completa ( Resistencia aos tipos: [CORTANTE, PERFURANTE] ) ( Fraquezas aos tipos: [ESMAGAMENTO] ) |

| Unidade: 3007 |
| Vida: 50 |
| Arma: Martelo ( Tipo do dano ESMAGAMENTO ) |
| Armadura: Completa ( Resistencia aos tipos: [CORTANTE, PERFURANTE] ) ( Fraquezas aos tipos: [ESMAGAMENTO] ) |

```

*Figura 12: Exemplo de indivíduos após a ação do Algoritmo genético.*

Na Figura 13 é demonstrado o resultado do combate após as seleções genéticas ocorridas ao longo das gerações. Esse que ocorre com a vitória da IA devido a mudança de características para adequar-se à estratégia humana do jogador. Adaptação essa que é perceptível nas escolhas da inteligência artificial, visto que ela escolhe o armamento que mais causará danos à primeira unidade, e, ao mesmo tempo, elenca a armadura que sofrerá menos danos dos demais sobreviventes.

```

A MAQUINA GANHOU!!!
Restaram vivos: 3 unidades

| Unidade: 3008 |
| Vida: 18 |
| Arma: Martelo ( Tipo do dano ESMAGAMENTO ) |
| Armadura: Completa ( Resistencia aos tipos: [CORTANTE, PERFURANTE] ) ( Fraquezas aos tipos: [ESMAGAMENTO] ) |

| Unidade: 3009 |
| Vida: 50 |
| Arma: Martelo ( Tipo do dano ESMAGAMENTO ) |
| Armadura: Completa ( Resistencia aos tipos: [CORTANTE, PERFURANTE] ) ( Fraquezas aos tipos: [ESMAGAMENTO] ) |

| Unidade: 3010 |
| Vida: 50 |
| Arma: Martelo ( Tipo do dano ESMAGAMENTO ) |
| Armadura: Completa ( Resistencia aos tipos: [CORTANTE, PERFURANTE] ) ( Fraquezas aos tipos: [ESMAGAMENTO] ) |

```

Figura 13: Resultado da batalha dos indivíduos do jogador contra a nova geração dos da IA.

Ademais, os genes selecionados basearam-se na sua população, que é maior que a do jogador, identificando que a fraqueza é mais benéfica do que a resistência, visto que a base de dano é 2 e se multiplica quando encontra a fraqueza, e divide-se quando encontra resistência, ocasionando 1 de dano na resistência e 4 na fraqueza. Consequentemente, por possuírem um número maior de indivíduos e todos combaterem primeiramente o personagem 1 do jogador, focaram seus esforços nos genes que neutralizam mais rapidamente a primeira ameaça e abusam de sua vantagem numérica. Dessa forma, o algoritmo traçou com sucesso uma nova tática e adaptou-se às decisões humanas, para assim vencer o combate nas futuras gerações.

#### 4. Considerações Finais

O presente trabalho circunda a problemática da melhoria nos desafios em jogos digitais de sobrevivência, melhorando os *non-player characters* adversários como forma de manter um jogo mais atraente.

Dessa forma, foi apresentada uma breve abordagem sobre Inteligência Artificial, e mais especialmente sobre Algoritmos Genéticos em jogos. Trazendo a questão de que essas tecnologias de implementação podem tornar os games menos repetitivos para os jogadores. Isso porque, aumentam a complexidade das estratégias que o jogador pode e deve utilizar para se sobrepor ao desafio ante a ele. Além de uma simples escalada de dificuldade justa, essas IAs podem aumentar a imersão de jogos, como visto no em Civilization, e também na apresentação de histórias em RimWorld.

Sob esse contexto, foi desenvolvido um protótipo que demonstra de maneira simplificada o funcionamento de um algoritmo genético para sua implementação em jogos eletrônicos. Isso porque, caso aplicada, essa tecnologia pode abrir um novo portal para uma maior imersão e um salto de qualidade com base em uma inteligência adaptativa. A implementação foi testada e brevemente avaliada.

## Referências

Asimov, Isaac. Eu, robô. Aleph, 2015.

Benin, Max Ricardo. EVOLUÇÃO DE NPCs E ADVERSÁRIOS EM JOGOS DE COMPUTADOR USANDO ALGORITMOS GENÉTICOS. 2022. 114 f. Monografia (Especialização) - Curso de Ciência da Computação, Faculdades Barddal, Florianópolis, 2007. Disponível em:

[https://www.arisa.com.br/~saulo/aulas/Publicacoes/TCC\\_MaxRicardoBenin\\_VersaoFinal.pdf](https://www.arisa.com.br/~saulo/aulas/Publicacoes/TCC_MaxRicardoBenin_VersaoFinal.pdf). Acesso em: 31 maio 2022.

Boulic, R. and Renault, O. (1991) “3D Hierarchies for Animation”, In: *New Trends in Animation and Visualization*, Edited by Nadia Magnenat-Thalmann and Daniel Thalmann, John Wiley & Sons Ltd., England.

Dyer, S., Martin, J. and Zulauf, J. (1995) “Motion Capture White Paper”, [http://reality.sgi.com/employees/jam\\_sb/mocap/MoCapWP\\_v2.0.html](http://reality.sgi.com/employees/jam_sb/mocap/MoCapWP_v2.0.html), December.

Holton, M. and Alexander, S. (1995) “Soft Cellular Modeling: A Technique for the Simulation of Non-rigid Materials”, *Computer Graphics: Developments in Virtual Environments*, R. A. Earnshaw and J. A. Vince, England, Academic Press Ltd., p. 449-460.

Knuth, D. E. (1984), *The TeXbook*, Addison Wesley, 15<sup>th</sup> edition.

Smith, A. and Jones, B. (1999). On the complexity of computing. In *Advances in Computer Science*, pages 555–566. Publishing Press.

Silva, J. A. S.; Mairink, C. H. P. Inteligência artificial: aliada ou inimiga. *LIBERTAS: Rev. Ciênc. Soc. Apl.*, Belo Horizonte, v. 9, n. 2, p. 64-85, ago./dez. 2019.

Nunes, Dierle; Marques, Ana L.P.C.. Inteligência artificial e direito processual: vieses algorítmicos e os riscos de atribuição de função decisória às máquinas. In: *Revista de Processo*. 2018. p. 421-447.

Russel, S.; Norvig, P.. *Inteligência Artificial*, 3a Edição. [s.l.] Elsevier Brasil, 2014.

Palioto, Narcizo G.F.; Colanzi, Thelma. Configuração de Algoritmos Genéticos Multiobjetivos para Otimização de Projeto de Arquitetura de Linha de Produto. In: *ESCOLA REGIONAL DE ENGENHARIA DE SOFTWARE (ERES)*, 4. , 2020, Evento Online. *Anais [...]*. Porto Alegre: Sociedade Brasileira de Computação, 2020 . p. 204-213. DOI: <https://doi.org/10.5753/eres.2020.13731>.

Lima, A.; Cabral, S. Algoritmo Genético. [s.l.: s.n.]. 2018. Disponível em: <[https://www.researchgate.net/profile/Silvio-Patricio/publication/343106354\\_Generic\\_algorithm\\_for\\_MLE/links/5f16f653299bf1720d56b69c/Generic-algorithm-for-MLE.pdf](https://www.researchgate.net/profile/Silvio-Patricio/publication/343106354_Generic_algorithm_for_MLE/links/5f16f653299bf1720d56b69c/Generic-algorithm-for-MLE.pdf)>

Oliveira, Guilherme P. de.. Métodos de Inteligência Artificial aplicados em jogos baseados em turnos. 2018. 81 f. Tese (Doutorado) - Curso de Computação, Universidade Federal de Uberlândia, Uberlândia, 2018. Disponível em: <https://repositorio.ufu.br/bitstream/123456789/22184/3/MetodosInteligenciaArtificial.pdf> f. Acesso em: 08 jun. 2022.

Stelter, Ítalo L. et al. ALGORITMO GENÉTICO E O SUPER MARIO. 2016. 32 f. Tese (Doutorado) - Curso de Sistemas de Informação, Universidade Federal da Grande Dourados, Dourados, 2016. Disponível em: <https://repositorio.ufgd.edu.br/jspui/bitstream/prefix/2894/1/ItaloLimaSteltter.pdf>. Acesso em: 08 jun. 2022.

Iyoda, Eduardo M.. Inteligência computacional no projeto automático de redes neurais híbridas e redes neurofuzzy heterogêneas / Eduardo Masato Iyoda. Campinas, SP: [s.n.], 2000. Disponível em: [https://www.dca.fee.unicamp.br/~vonzuben/theses/emi\\_mest](https://www.dca.fee.unicamp.br/~vonzuben/theses/emi_mest) . Acesso em: 08 jun. 2022.

Souza, M.; Vahldick, A. Influência dos jogos no campo da inteligência artificial. Revista Brasileira de Contabilidade e Gestão, [S. l.], v. 2, n. 4, p. 157-160, 2013. Disponível em: <https://www.revistas.udesc.br/index.php/reavi/article/view/4062>. Acesso em: 13 jun. 2022.

Kishimoto, André. Inteligência Artificial em Jogos Eletrônicos. 2004. 11 f. Dissertação (Doutorado) - Curso de --, --, --, 2004. Disponível em: [http://www.karenreis.com.br/pdf/andre\\_kishimoto.pdf](http://www.karenreis.com.br/pdf/andre_kishimoto.pdf). Acesso em: 13 jun. 2022

Imótheo Júnior, Gilberto; Lima, Sérgio M.B.. Algoritmos Genéticos Aplicados a Jogos Eletrônicos. 2010. 58 f. TCC (Graduação) - Curso de Sistemas de Informação, 1 Faculdade Metodista Granbery, Juiz de Fora, 2010. Disponível em: <http://re.granbery.edu.br/artigos/MzU2.pdf>. Acesso em: 20 jun. 2022.