

UNIVERSIDADE FEDERAL DO PAMPA

Victor Queiroz Ribeiro

**Explorando o Uso de Algoritmo Genético  
em Jogos Digitais**

Alegrete  
2019



Victor Queiroz Ribeiro

# Explorando o Uso de Algoritmo Genético em Jogos Digitais

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Engenharia de Software.

Orientador: Profa. Dra. Aline Vieira de Mello

Alegrete  
2019

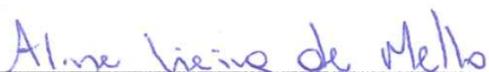


Victor Queiroz Ribeiro

## Explorando o Uso de Algoritmo Genético em Jogos Digitais

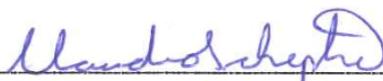
Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Engenharia de Software.

Trabalho de Conclusão de Curso defendido e aprovado em 25 de novembro de 2019.  
Banca examinadora:



---

Profa. Dra. Aline Vieira de Mello  
Orientadora  
UNIPAMPA



---

Prof. Dr. Claudio Schepke  
UNIPAMPA



---

Prof. Dr. Fábio Paulo Basso  
UNIPAMPA



## **AGRADECIMENTOS**

Primeiramente gostaria de agradecer a Deus, que junto com todas as coisas criou o café, sem o qual este trabalho não seria possível. Em segundo lugar agradeço a Dr<sup>a</sup>. Aline Viera de Mello que assim como eu, perdeu vários finais de semana para que esse trabalho pudesse ser concluído. Gostaria também de agradecer aos meus pais pelas palavras de encorajamento toda vez que eu liguei para casa ameaçando saltar de uma ponte, durante o período da minha graduação. Também agradeço aos colegas Tobias Tirola e Guilherme Bolfe pela ajuda na condução do experimento proposto neste trabalho. Agradeço também a Linus Torvalds, Richard Stallman e a toda comunidade Software Livre, pessoas que indiretamente mas essencialmente, contribuíram com este trabalho.



## RESUMO

Jogos digitais movimentam bilhões de dólares anualmente em todo o mundo. A maior parte deste dinheiro é gasto em conteúdo. O conteúdo do jogo é que mantém o jogador engajado e o jogo viciante. Este trabalho tem como objetivo explorar o uso de algoritmos genéticos juntamente com geração de conteúdo procedural como uma alternativa a criação manual de conteúdo, auxiliando desenvolvedores independentes e pequenos times na criação de jogos digitais. O método científico utilizado neste trabalho é um método comparativo, que procede pela investigação de indivíduos, classes, fenômenos ou fatos, com vistas a ressaltar as diferenças e as similaridades entre eles. Assim, neste trabalho foi desenvolvido um *framework* de algoritmo genético que foi utilizado em dois experimentos. O experimento de número 1 trata de uma comparação entre o clássico jogo *Space Invaders* e um jogo similar, denominado InvaderZ, que utiliza algoritmo genético. O experimento de número 2 trata de um sistema de reprodução, que faz parte de um jogo. Esse experimento permitiu a comparação de indivíduos de diferentes gerações de uma população e também a comparação entre populações de personagens no mundo do jogo. Durante a execução do experimento 1, que foi validado junto ao usuário, o algoritmo genético foi capaz de aprimorar a mecânica proposta pelo jogo *Space Invaders*. Já no experimento 2, o algoritmo genético foi capaz de introduzir biodiversidade em um jogo em que muitos personagens são necessários para popular o mundo. Concluiu-se que o algoritmo genético pode contribuir para o desenvolvimento de jogos digitais, auxiliando o desenvolvedor na criação de conteúdo procedural, que pode tornar o jogo muito mais interessante.

**Palavras-chave:** Algoritmo Genético. Geração de Conteúdo Procedural. Jogos Digitais.



## ABSTRACT

Digital games moves billions of dollars every year. The largest amount of this money is spent on content. The content keeps the player engaged and the game addictive. This paper introduces the use of genetic algorithm alongside with procedural content generation as an alternative to manual content creation. That way, helping independent developers and small teams on creating digital games. The scientific method utilized in this paper is a comparative method, that investigates individuals, classes, phenomenon or facts, in order to highlight the similarities and differences between them. So, in this paper a genetic algorithm framework was developed and later used on two experiments. The experiment number 1 is a comparison between the classic game Space Invaders and a generic space invaders like game, named InvaderZ, that uses genetic algorithm. The experiment number 2 is a reproduction system that is part of a game. This experiment allowed for a comparison between individuals of different generations of a population as well a comparison between different populations of characters on the game world. During the execution of the experiment 1, validated with the user, the genetic algorithm was capable of enhancing the game mechanics proposed by the game Space Invaders. On the experiment 2, the genetic algorithm was capable of introducing biodiversity in a game where many characters were needed to populate the world. It was then concluded that the genetic algorithm can contribute to the development of digital games, helping the developer on the creation of procedural content, that can make the game a lot more interesting.

**Key-words:** Genetic Algorithm. Procedural Content Generation. Digital Games.



## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1 – Total gasto em jogos . . . . .  | 17 |
| Figura 2 – Total gasto em conteúdo de jogos . . . . .  | 18 |
| Figura 3 – Diagrama de fluxo do paradigma da programação genética . . . . .  | 23 |
| Figura 4 – Processo de filtragem dos artigos encontrados . . . . .   | 28 |
| Figura 5 – Processo de desenvolvimento . . . . .   | 33 |
| Figura 6 – Diagrama de classe de um algoritmo genético . . . . .   | 38 |
| Figura 7 – Captura de tela do jogo Space Invaders . . . . .  | 39 |
| Figura 8 – Captura de tela do experimento InvaderZ . . . . .   | 40 |
| Figura 9 – Mapa gerado proceduralmente . . . . .   | 42 |
| Figura 10 – Gráficos modulares . . . . .   | 43 |
| Figura 11 – P1. Você já havia jogado o jogo Space Invaders? . . . . .  | 47 |
| Figura 12 – P2. Quantos inimigos únicos você pode notar durante o jogo Space<br>Invaders? . . . . .                    | 48 |
| Figura 13 – P3. Você conseguiu identificar padrões na forma como os inimigos se<br>moviam em Space Invaders? . . . . . | 48 |
| Figura 14 – P4. Quantos inimigos únicos você pode notar durante o jogo InvaderZ? . . . . .                             | 49 |
| Figura 15 – P5. Você conseguiu identificar padrões na forma como os inimigos se<br>moviam em InvaderZ? . . . . .       | 49 |
| Figura 16 – P6. Você concorda que InvaderZ proporciona uma experiência nova a<br>cada execução do jogo? . . . . .      | 50 |
| Figura 17 – Árvore genealógica da população <i>A</i> . . . . .   | 51 |
| Figura 18 – Árvore genealógica da população <i>B</i> . . . . .   | 52 |
| Figura 19 – Árvore genealógica da população <i>C</i> . . . . .   | 52 |
| Figura 20 – Alguns indivíduos (NPC's) espalhados por um mapa de pequena escala. . . . .                                | 53 |



## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1 – Número de trabalhos encontrados em cada base de dados . . . . . | 28 |
| Tabela 2 – Tabela de qualidade dos trabalhos . . . . .                     | 29 |
| Tabela 3 – Trabalhos relacionados agrupados por objetivo . . . . .         | 30 |
| Tabela 4 – Caso de Teste TC01 . . . . .                                    | 44 |
| Tabela 5 – Perguntas apresentadas no formulário. . . . .                   | 45 |
| Tabela 6 – Resumo dos resultados dos participantes . . . . .               | 50 |
| Tabela 7 – Caso de Teste TC02 . . . . .                                    | 63 |
| Tabela 8 – Caso de Teste TC03 . . . . .                                    | 63 |
| Tabela 9 – Caso de Teste TC04 . . . . .                                    | 64 |
| Tabela 10 – Caso de Teste TC05 . . . . .                                   | 64 |
| Tabela 11 – Caso de Teste TC06 . . . . .                                   | 64 |



## SUMÁRIO

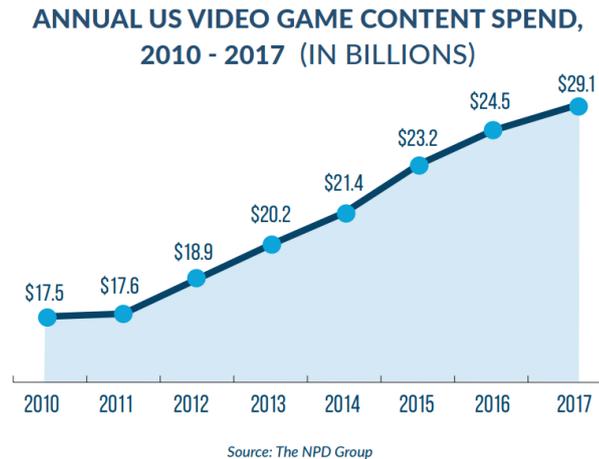
|       |   |    |
|-------|---|----|
| 1     | INTRODUÇÃO . . . . .                            | 17 |
| 1.1   | Objetivo . . . . .                              | 18 |
| 1.2   | Organização do documento . . . . .              | 19 |
| 2     | FUNDAMENTAÇÃO TEÓRICA . . . . .                 | 21 |
| 2.1   | Jogos digitais . . . . .                        | 21 |
| 2.2   | Gamificação . . . . .                           | 21 |
| 2.3   | Geração de Conteúdo Procedural . . . . .        | 21 |
| 2.4   | Framework . . . . .                             | 21 |
| 2.5   | Algoritmo genético . . . . .                    | 22 |
| 2.5.1 | Função de <i>fitness</i> . . . . .              | 22 |
| 2.5.2 | Indivíduo . . . . .                             | 23 |
| 2.5.3 | Seleção . . . . .                               | 24 |
| 2.5.4 | Reprodução . . . . .                            | 24 |
| 2.5.5 | Elitismo . . . . .                              | 24 |
| 2.5.6 | Mutação . . . . .                               | 25 |
| 3     | TRABALHOS RELACIONADOS . . . . .                | 27 |
| 3.1   | Revisão Sistemática de Literatura . . . . .     | 27 |
| 3.1.1 | Planejamento . . . . .                          | 27 |
| 3.1.2 | Condução . . . . .                              | 28 |
| 3.1.3 | Análise . . . . .                               | 29 |
| 3.1.4 | Considerações Finais . . . . .                  | 31 |
| 4     | METODOLOGIA . . . . .                           | 33 |
| 4.1   | Elicitação dos requisitos . . . . .             | 33 |
| 4.2   | Proposição de um framework . . . . .            | 34 |
| 4.3   | Desenvolvimento de Experimentos . . . . .       | 34 |
| 4.4   | Validação . . . . .                             | 34 |
| 4.5   | Análise dos Resultados . . . . .                | 35 |
| 5     | DESENVOLVIMENTO . . . . .                       | 37 |
| 5.1   | Elicitação dos Requisitos . . . . .             | 37 |
| 5.2   | Proposição de um Framework . . . . .            | 37 |
| 5.3   | Desenvolvimento de Experimentos . . . . .       | 38 |
| 5.3.1 | Experimento 1 - InvaderZ . . . . .              | 38 |
| 5.3.2 | Experimento 2 - Sistema de reprodução . . . . . | 41 |
| 5.4   | Testes funcionais e de sistema . . . . .        | 43 |
| 5.5   | Validação . . . . .                             | 44 |

|       |  |    |
|-------|--|----|
| 5.5.1 | Teste com usuário - Experimento 1 . . . . .                | 44 |
| 5.5.2 | Validação - Experimento 2 . . . . .                        | 45 |
| 6     | ANÁLISE DOS RESULTADOS . . . . .                           | 47 |
| 6.1   | Comparação entre jogos sem e com uso de algoritmo genético | 47 |
| 6.2   | Comparação entre indivíduos da mesma população . . . . .   | 51 |
| 6.3   | Comparação entre populações . . . . .                      | 53 |
| 6.4   | Resumo . . . . .   | 54 |
| 7     | CONSIDERAÇÕES FINAIS . . . . .                             | 57 |
|       | REFERÊNCIAS . . . . .                                      | 59 |
|       | APÊNDICE A – CASOS DE TESTES . . . . .                     | 63 |
|       | APÊNDICE B – TERMO DE CONFIDENCIALIDADE . . . . .          | 65 |
|       | APÊNDICE C – TERMO DE CONSENTIMENTO . . . . .              | 67 |
|       | APÊNDICE D – QUESTIONÁRIO - INVADERZ . . . . .             | 69 |

## 1 INTRODUÇÃO

Nos últimos quarenta anos, a indústria dos jogos digitais cresceu em um ritmo constante, como podemos ver na Figura 1. Essa indústria aumentou sua audiência em todo o mundo, ampliando sua demografia e adicionando novos dispositivos de acesso (videogames, computadores pessoais, consoles portáteis, dispositivos móveis, *tablets*, etc.).

Figura 1 – Total gasto em jogos



Fonte: (ESA, 2018).

Certamente a oferta e demanda mudou sob pressão de uma variedade de fatores, tais como o desenvolvimento de tecnologias em interfaces, dispositivos, redes, a emergência da computação social e comunidades e a produção de simples e curtos jogos (jogos casuais), capturando uma demanda até então insatisfeita em diversas categorias como de idade, classe socio-econômica ou gênero (PRATO; GONZALEZ; SIMON, 2014).

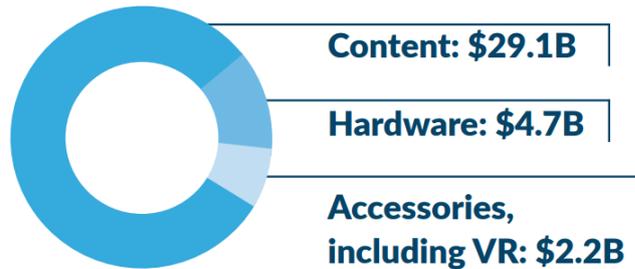
Para um jogo se tornar viciante, a distribuição de pontos de interesse ao redor do mundo deve ser feita de maneira inteligente. Por exemplo, se demora muito tempo para o jogador alcançar alguns pontos de interesse, ele pode sentir uma falta de recompensa pelo tempo que gastou, se sentir entediado e abandonar o jogo. A mesma coisa pode acontecer se ele achar todos os pontos de interesse requeridos sem a necessidade de explorar o mundo. Ao mesmo tempo, ele deve sentir que cada novo lugar que ele descobre é de alguma maneira único. Se não, seu desejo por exploração vai diminuir. Por isso, conteúdo é um fator importante para manter o jogador engajado no mundo do jogo.

Como podemos ver na Figura 2, se gasta mais dinheiro com conteúdo do que com *hardware* e acessórios somados. De fato, a criação de conteúdo procedural para jogo tem evoluído rapidamente, guiado pelo aumento da demanda de estúdios desenvolvedores de jogos. Reduzir o trabalho do *game designer* usando métodos de criação de conteúdo procedural permite que o jogo seja produzido de forma mais rápida e mais barata, sem abrir mão da qualidade. Eles também podem reduzir drasticamente o tamanho de arquivo do jogo, permitindo que esses jogos alcancem maior audiência, incluindo dispositivos

móveis, geralmente menos poderosos que computadores de mesa. Pode se dizer, sem reservas, que a introdução de métodos de criação de conteúdo procedural é um ponto forte em jogos para dispositivos móveis (ALVES; COELHO; NOGUEIRA, 2017).

Figura 2 – Total gasto em conteúdo de jogos

The **total** consumer spend on the video game industry was  
**\$36 BILLION IN 2017.**



Source: The NPD Group

Fonte: ESA (2018).

A criação de conteúdo procedural é a criação de conteúdo por meio de algoritmos, sem a necessidade da ajuda de um humano. O conteúdo é gerado a partir de um conjunto bem definido de regras (algoritmos). Algoritmos genéticos são uma classe particular de algoritmos que usam técnicas inspiradas pela biologia evolutiva para encontrar soluções aproximadas em problemas de otimização e busca. No contexto de jogos digitais, esse problema poderia ser encontrar inimigos capazes de oferecer desafios ao jogador. Nesse caso, o jogador teria o papel principal porque seu modo de jogar afetaria diretamente em quais inimigos são mais aptos ou mais desafiadores. Assim, o algoritmo genético poderia ser usado em jogos digitais para definir regras que guiarão a geração de conteúdo procedural de inimigos, mas também de mapas, cenários, regras de jogo, entre outros.

## 1.1 Objetivo

Este trabalho tem como objetivo explorar o uso de algoritmos genéticos no desenvolvimento de jogos digitais. Para atingir tal objetivo, os seguintes objetivos específicos foram definidos:

- Pesquisar o uso de algoritmos genéticos no desenvolvimento de jogos digitais;
- Propor um *framework* para desenvolvimento de algoritmos genéticos;
- Utilizar o *framework* proposto no desenvolvimento de jogos digitais;
- Avaliar a contribuição do algoritmo genético nos jogos digitais;

- Apresentar vantagens e desvantagens do uso de algoritmo genético no desenvolvimento de jogos.

Espera-se que esse trabalho contribua com os desenvolvedores de jogos digitais, em especial os desenvolvedores solos e pequenas equipes. Para isso, boas práticas de engenharia de software são recomendadas e observadas durante todo o trabalho, visando a auxiliar o processo de desenvolvimento e assim reduzindo os custos e aumentando a qualidade do produto.

## 1.2 Organização do documento

O restante deste documento está organizado nos seguintes Capítulos:

- No Capítulo 2 - Fundamentação teórica - são apresentados os conceitos relacionados a jogos digitais, geração de conteúdo procedural e algoritmo genético.
- No Capítulo 3 - Trabalhos relacionados - é descrito o processo de busca realizada, identificando onde e como algoritmos genéticos são utilizados em jogos digitais.
- No Capítulo 4 - Metodologia - são apresentados os métodos científicos utilizados para o desenvolvimento deste trabalho.
- No Capítulo 5 - Desenvolvimento - é apresentado o desenvolvimento deste trabalho dividido por etapas.
- No Capítulo 6 - Análise dos resultados - é apresentada a análise dos dados colhidos durante a execução deste trabalho.
- No Capítulo 7 - Considerações finais - o autor faz suas considerações finais e apresenta propostas para trabalhos futuros.



## 2 FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo é abordado a fundamentação teórica deste trabalho, apresentando os conceitos relacionados a jogos digitais, geração de conteúdo procedural e algoritmo genético.

### 2.1 Jogos digitais

Jogos são atividades estruturadas, com regras definidas, praticadas com fins recreativos. Geralmente envolvem estimulação mental ou física e muitas vezes ambos. Muitos deles ajudam a desenvolver habilidades práticas, servem como uma forma de exercício ou realizam um papel educativo. Jogos são uma parte fundamental da existência humana (CRAWFORD, 1982).

Um jogo digital (ou videogame ou jogo eletrônico) é o termo genérico que se refere a jogos eletrônicos criados para serem jogados num computador, console ou outro dispositivo tecnológico (PIVEC; KEARNEY, 2007), como celulares. Pode ser definido como um jogo onde existe interação entre humano e computador, recorrendo ao uso de tecnologia (GEE, 2003).

### 2.2 Gamificação

Gamificação é originária do termo em inglês “*gamification*”, e trata-se do uso de técnicas de jogos, majoritariamente virtuais, para cativar pessoas por intermédio de desafios constantes e bonificações (DETERDING, 2012).

### 2.3 Geração de Conteúdo Procedural

Os termos “procedural” e “geração” implicam que estamos lidando com procedimentos computadorizados, ou algoritmos, que criam algo. Um método de geração de conteúdo procedural pode ser executado por um computador (talvez com a ajuda de um humano) e vai produzir algo. O termo chave aqui é “conteúdo”. Neste trabalho, conteúdo é o que está contido no jogo: níveis, mapas, regras de jogo, texturas, histórias, itens, missões, armas, veículos, personagens, etc (SHAKER; TOGELIUS; NELSON, 2016).

### 2.4 Framework

Em desenvolvimento de software, um *framework* é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. Um *framework* pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação. Ao contrário das bibliotecas, é o *framework* quem dita o fluxo de controle da aplicação. Um *framework* é uma aplicação reusável, “semi-completa”,

que pode ser especializada para produzir aplicações customizadas (FAYAD; SCHMIDT, 1997).

## 2.5 Algoritmo genético

Um algoritmo genético é uma técnica de busca utilizada na ciência da computação para achar soluções aproximadas em problemas de otimização e busca, fundamentado principalmente pelo americano John Henry Holland. Algoritmos genéticos são uma classe particular de algoritmos evolutivos que usam técnicas inspiradas pela biologia evolutiva como hereditariedade, mutação, seleção natural e recombinação (ou *crossing over*) (LINDEN, 2006).

Algoritmos genéticos são implementados como uma simulação de computador em que uma população de representações abstratas de solução é selecionada em busca de soluções melhores. A evolução geralmente se inicia a partir de um conjunto de soluções criado aleatoriamente e é realizada por meio de gerações. A cada geração, a adaptação de cada solução na população é avaliada, alguns indivíduos são selecionados para a criação da próxima geração através de reprodução e ou mutação para formar uma nova população. A nova população então é utilizada como entrada para a próxima iteração do algoritmo.

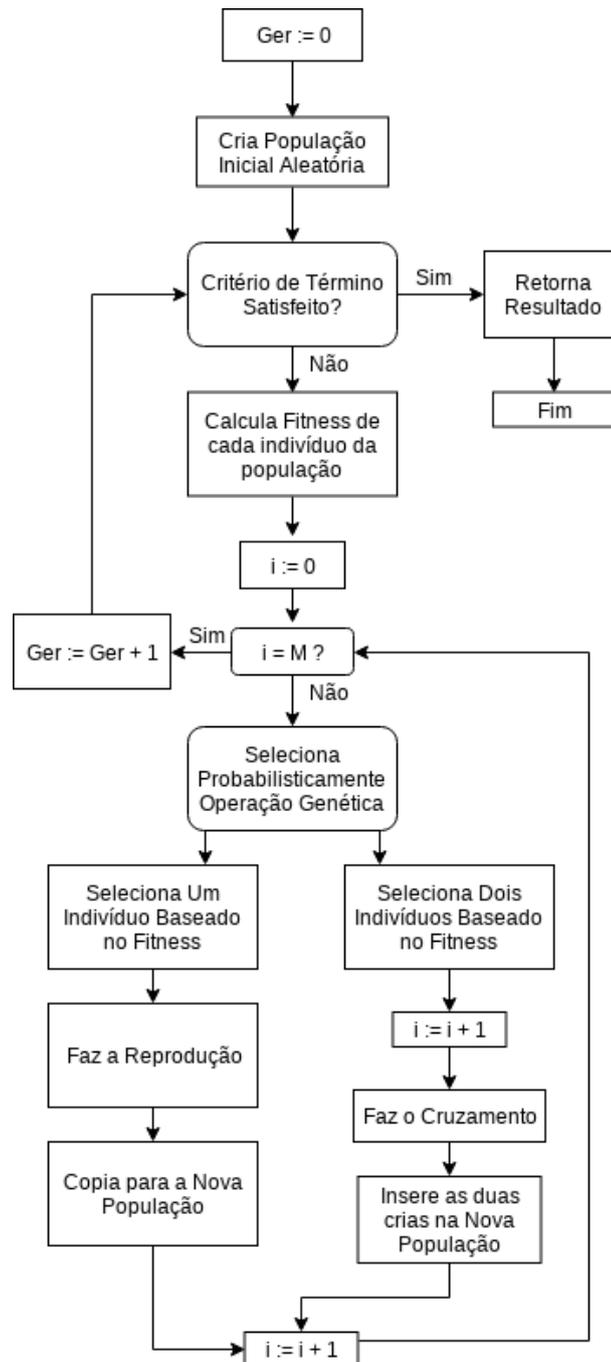
A Figura 3 apresenta um diagrama de fluxo da programação genética. O índice  $i$  faz referência a um indivíduo em uma população de tamanho  $M$ . A variável  $Ger$  é o número da atual população.

Segundo Koza e Koza (1992), o algoritmo genético é um algoritmo capaz de transformar uma população de indivíduos em uma nova população, usando operações genéticas padronizadas pelo princípio de sobrevivência do mais apto, proposto por Charles Darwin. Essas operações genéticas são citadas e explicadas a seguir.

### 2.5.1 Função de *fitness*

Segundo Goldberg (1989), a função-objetivo ou *Fitness Score* é o objeto de uma otimização. É essa função que avalia o quão “apto” é o indivíduo. Se tomarmos como exemplo uma corrida com  $N$  participantes, poderíamos escrever uma função-objetivo que conseguisse determinar qual dos indivíduos é o mais rápido. Caso a velocidade não fosse importante e sim a distância percorrida, definiríamos nossa função-objetivo para avaliar cada um dos indivíduos por distância percorrida, dando uma maior nota para os que fossem mais longe na corrida. Pela função-objetivo ser o objeto de otimização, precisamos definir o que estamos buscando otimizar. Velocidade? Distância percorrida? Quem sabe um misto dos dois: qual dos indivíduos alcançou maior distância com maior velocidade? A função-objetivo é que determina a aptidão de cada indivíduo na população.

Figura 3 – Diagrama de fluxo do paradigma da programação genética



Fonte: Adaptado de Goldberg (1989).

### 2.5.2 Indivíduo

O indivíduo é um portador do seu código genético. O código genético é uma representação do espaço de busca do problema a ser resolvido. O código genético deve ser uma representação capaz de especificar todo o conjunto dos valores no espaço de busca, e precisa ter tamanho finito (GOLDBERG, 1989).

Usando novamente o exemplo da corrida, o indivíduo seria o maratonista. Seu

código genético determinaria sua velocidade, resistência, peso, altura e quaisquer outros parâmetros que pudessem alterar o curso da corrida. Esse código genético, que define o indivíduo, é o código que vai ser passado adiante no momento da reprodução.

Um indivíduo também possui uma pontuação de *fitness*. Essa pontuação é o que o classifica na população. No exemplo da corrida, corredores mais rápidos teriam maior pontuação *fitness*, ou maior aptidão, enquanto corredores mais lentos teriam uma pontuação menor.

### 2.5.3 Seleção

A seleção é uma parte chave do algoritmo. Nesta etapa são selecionados os pais que passarão pelo processo de reprodução. Em geral, usa-se o algoritmo de seleção por “roleta”, em que os indivíduos são ordenados de acordo com a sua aptidão e lhes são atribuídas probabilidades decrescentes. Probabilidades essas proporcionais à razão entre a aptidão do indivíduo (seu *fitness*) e a soma das aptidões de todos os indivíduos da população. A escolha é feita então aleatoriamente guiada por essas probabilidades (LINDEN, 2006).

Dessa forma, os pais mais aptos têm maior chance de serem escolhidos no momento da seleção, sem deixar de lado a diversidade dos menos aptos. Outras formas de seleção podem ser aplicadas dependendo do problema a ser tratado. Como exemplo, pode-se citar a seleção por “torneio”, em que são selecionados diversos pequenos subconjuntos da população, sendo selecionado o indivíduo de maior adequação de cada um desses grupos.

### 2.5.4 Reprodução

Segundo Goldberg (1989), a reprodução geralmente é dividida em três etapas: acasalamento, recombinação e mutação. O acasalamento é a escolha de dois indivíduos para se reproduzirem, geralmente gerando dois descendentes para manter o tamanho populacional. A reprodução, ou *crossing-over*, é um processo que imita o processo biológico na reprodução sexuada: os descendentes recebem em seu código genético parte do código genético de um de seus pais e parte do código do outro. Essa reprodução garante que os melhores indivíduos sejam capazes de trocar entre si as informações que os levam a ser mais aptos a sobreviver, e assim gerar descendentes ainda mais aptos.

### 2.5.5 Elitismo

Uma variação prática do processo de se construir uma nova população é permitir que o melhor indivíduo de uma geração siga para a próxima geração, inalterado. Esta estratégia é conhecida como seleção elitista e garante que a melhor solução encontrada pelo algoritmo genético não se perca de geração para geração (KHALAFALLAH; ABDELRAHEEM, 2010).

### 2.5.6 Mutação

Por último acontecem as mutações, que geralmente acontecem com uma baixa probabilidade, e tem como objetivo permitir maior variabilidade genética na população, impedindo que a busca fique estagnada em um mínimo local. A mutação permite que o novo indivíduo não seja uma cópia exata de seus pais, introduzindo diversidade na nova população.



### 3 TRABALHOS RELACIONADOS

Este Capítulo descreve a revisão sistemática que foi realizada buscando identificar onde e como algoritmos genéticos estão sendo utilizados em jogos digitais, assim como analisa os resultados obtidos.

#### 3.1 Revisão Sistemática de Literatura

Para a execução dessa revisão sistemática, o processo proposto por Kitchenham e Charters (2007) foi utilizado. Esse processo é composto pelas etapas de planejamento, condução e análise, as quais são detalhadas a seguir.

##### 3.1.1 Planejamento

Durante a etapa de planejamento foram definidas as seguintes perguntas para guiar o processo de busca:

- Como algoritmos genéticos estão sendo utilizados em jogos digitais?
- Algoritmos genéticos são utilizados para aumentar a diversidade de objetos (personagens, vegetação, itens...) em jogos digitais?
- Como são validados os algoritmos genéticos?

Com base nessas perguntas, foi criada a seguinte *string* de busca: (“genetic algorithm”) AND (“games”). Variações dessa *string* foram utilizadas nas diferentes bases. Adicionalmente, os critérios de inclusão e exclusão dos trabalhos foram especificados:

- **Critérios de inclusão**

CI1 Trabalhos que trouxessem os termos “genetic algorithm” e “games”.

CI2 Trabalhos da área da computação.

CI3 Trabalhos disponíveis na íntegra.

- **Critérios de exclusão**

CE1 Trabalhos duplicados.

CE2 Trabalhos que não usassem o termo “games” no contexto de "jogos digitais”.

CE3 Trabalhos com menos de 4 páginas.

### 3.1.2 Condução

As bases de dados escolhidas foram ACM, IEEE, Scopus e Springer. Sobre cada uma das bases foi rodada a *string* de busca, resultando em 348 trabalhos, conforme apresentado na Tabela 1.

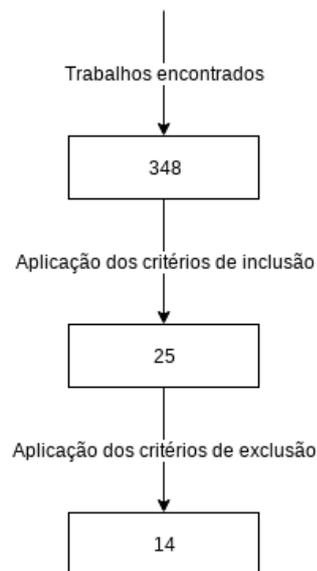
A Figura 4 mostra o processo de filtragem realizado, com o intuito de que apenas os trabalhos relevantes permanecessem para a etapa de análise. Observa-se que dos 348 trabalhos encontrados, 25 foram selecionados usando os critérios de inclusão. Após esses 25 trabalhos passaram então pelos critérios de exclusão, resultando em 14 trabalhos selecionados.

Tabela 1 – Número de trabalhos encontrados em cada base de dados

| Base de dados | Número de artigos |
|---------------|-------------------|
| ACM           | 129               |
| IEEE          | 52                |
| SCOPUS        | 89                |
| SPRINGER      | 78                |
| Total         | 348               |

Fonte: Próprio autor (2019).

Figura 4 – Processo de filtragem dos artigos encontrados



Fonte: Próprio autor (2019).

Os trabalhos selecionados então passaram por um processo de classificação que utilizou as seguintes questões de qualidade:

- QA1 - O trabalho menciona algoritmo genético em jogos digitais?
- QA2 - O trabalho usa algoritmo genético para criação de conteúdo?

Para a questão QA1 foi atribuído um valor de 1 ponto, com opções de resposta sim e não. Caso a resposta fosse sim, 1 ponto foi atribuído ao trabalho e, em caso contrário, nenhum ponto foi atribuído. Para a pergunta QA2 também foi atribuído 1 ponto. Contudo, para essa questão as opções eram sim, não e parcial. Caso a resposta fosse sim era atribuído 1 ponto ao trabalho, caso fosse parcial era atribuído 0,5 ponto e nenhum ponto caso fosse não. A Tabela 2 apresenta cada trabalho relacionado, as repostas para as questões de qualidade e sua respectiva pontuação.

Tabela 2 – Tabela de qualidade dos trabalhos

| Trabalho                             | QA1 | QA2 | Pontuação |
|--------------------------------------|-----|-----|-----------|
| Charoenkwan, Fang e Wong (2010)      | sim | não | 1         |
| Huo et al. (2009)                    | sim | não | 1         |
| Carvalho et al. (2010)               | sim | sim | 2         |
| Johnson e Wiles (2001)               | sim | não | 1         |
| Alves, Coelho e Nogueira (2017)      | sim | sim | 2         |
| Francisco e Reis (2008)              | sim | não | 1         |
| Ray, Gordon e Vaucher (2014)         | sim | não | 1         |
| Sun et al. (1994)                    | sim | não | 1         |
| Ishibuchi, Nakari e Nakashima (1999) | sim | não | 1         |
| Ali, Nakao e Chen (2000)             | sim | não | 1         |
| Louis e Miles (2005)                 | sim | não | 1         |
| Hendrikx et al. (2013)               | sim | não | 1         |
| Cole, Louis e Miles (2004)           | sim | não | 1         |
| Salge et al. (2008)                  | sim | não | 1         |

Fonte: Próprio autor (2019).

### 3.1.3 Análise

Dos trabalhos resultantes, apresentados na Tabela 3 constatou-se que muitos trabalhos utilizaram algoritmo genético substituindo o jogador, com intuito de verificar se o algoritmo conseguiria melhores resultados do que um jogador humano. Por jogador humano entende-se um humano operando um computador ou um pedaço de código escrito por um humano a fim controlar um “agente” no jogo.

O trabalho de Ray, Gordon e Vaucher (2014) descreve como o algoritmo genético foi utilizado para jogar *QWOP* (2008), jogo famoso pela sua dificuldade. O autor concluiu que por fatores aleatórios (devido ao motor de física do jogo) o algoritmo genético não foi capaz de sobressair a um humano. Já em um jogo de pedra-papel-tesoura, Ali, Nakao e Chen (2000) observaram que o algoritmo genético se saiu melhor do que um jogador humano, devido ao fato de padrões de jogo serem melhor detectados por computadores, aumentando assim sua capacidade de vencer. Ainda, o trabalho de Francisco e Reis (2008) mostrou que o algoritmo genético é capaz de criar várias estratégias de combate,

Tabela 3 – Trabalhos relacionados agrupados por objetivo

| Trabalho                             | Objetivo                     | Validação                  |
|--------------------------------------|------------------------------|----------------------------|
| Sun et al. (1994)                    | Substituir o jogador         | Desempenho no jogo         |
| Ishibuchi, Nakari e Nakashima (1999) | Substituir o jogador         | Desempenho no jogo         |
| Ali, Nakao e Chen (2000)             | Substituir o jogador         | Desempenho no jogo         |
| Louis e Miles (2005)                 | Substituir o jogador         | Desempenho no jogo         |
| Francisco e Reis (2008)              | Substituir o jogador         | Desempenho no jogo         |
| Charoenkwan, Fang e Wong (2010)      | Substituir o jogador         | Desempenho no jogo         |
| Ray, Gordon e Vaucher (2014)         | Substituir o jogador         | Desempenho no jogo         |
| Cole, Louis e Miles (2004)           | Melhorar experiência de jogo | Diversidade de estratégias |
| Salge et al. (2008)                  | Melhorar experiência de jogo | Diversidade de estratégias |
| Huo et al. (2009)                    | Melhorar experiência de jogo | Diversidade de estratégias |
| Johnson e Wiles (2001)               | Aproximar teoria à prática   | Não se aplica              |
| Hendrikx et al. (2013)               | Pesquisa                     | Não se aplica              |
| Carvalho et al. (2010)               | Criação de conteúdo          | Diversidade de conteúdo    |
| Alves, Coelho e Nogueira (2017)      | Criação de conteúdo          | Diversidade de conteúdo    |

Fonte: Próprio autor (2019).

controlando uma nave em um jogo de batalha espacial e que, em alguns casos, essas estratégias são melhores do que as estratégias programadas por um humano (*hard-coded*).

Os algoritmos genéticos também são utilizados para melhorar a experiência do usuário em jogos (*gameplay*). No trabalho de Salge et al. (2008), um algoritmo genético foi utilizado para testar mecânicas do jogo e procurar por falhas críticas, mantendo um jogo bem balanceado com maior chances de aceitação pelo público. O trabalho de Huo et al. (2009) visou testar o desempenho do algoritmo genético em um jogo de *tower defense*, para determinar a viabilidade do seu uso em jogos de estratégia em tempo real. Como um algoritmo genético, assim como a própria evolução, leva tempo até convergir a solução “adequada”, foi descartado seu uso nesse cenário, já que jogos em tempo real necessitam de um tempo de resposta mínimo.

O trabalho de Johnson e Wiles (2001) visa facilitar a colaboração entre pesquisadores de inteligência artificial e desenvolvedores de jogos. O trabalho menciona o fato do algoritmo genético não ser amplamente utilizado em jogos digitais, ainda que ele ofereça benefícios de um sistema de inteligência que evolui com o passar do tempo. O trabalho propõe colaboração nas áreas de evolução e redes neurais, que são conhecidas por deman-

darem muito processamento. Soluções não tão gananciosas em relação ao consumo de processamento podem ser utilizadas em jogos, como por exemplo um pré-processamento dos dados que não necessitam ser calculados em tempo de execução.

O trabalho de Hendriks et al. (2013) investigou o uso de criação de conteúdo procedural para jogos, categoria em que o algoritmo genético se enquadra. Nesse trabalho, os jogos foram divididos em objetos, espaço, sistemas e cenários, e através de uma pesquisa buscou-se identificar em quais dessas “etapas” a criação de conteúdo procedural foi utilizada em jogos comerciais de sucesso. Dos 23 jogos encontrados, todos usaram a criação de conteúdo procedural. Dentre eles, destacou-se o jogo *Dwarf Fortress* (2006) que usou criação de conteúdo procedural em quase todos os campos do jogo. No entanto, o trabalho de Hendriks et al. (2013) não especifica quais jogos utilizaram algoritmo genético.

No trabalho de Carvalho et al. (2010), um jogo de RPG foi utilizado para demonstrar as vantagens de se usar algoritmos genéticos na criação de cenários “vivos”, em que eventos acontecem com frequência. O resultado relatado foi a criação de um mapa com suas próprias características geográficas, como um fator extra a ser considerado pela comunidade dos jogadores.

Por fim, o trabalho de Alves, Coelho e Nogueira (2017) apresenta um algoritmo de criação procedural de conteúdo com algoritmo genético para jogos com massivo número de jogadores *online* (do inglês *Massive Multiplayer Online* - MMO). Segundo os autores, a geração incremental de conteúdo permite um progresso eficaz no jogo com pequeno impacto no desempenho, inclusive com a possibilidade de lançamento do jogo em dispositivos de baixo desempenho.

#### 3.1.4 Considerações Finais

Dos trabalhos analisados, apenas o trabalho de Cole, Louis e Miles (2004) utiliza o algoritmo genético em tempo de execução para adaptar a estratégia dos *bots* (personagens controlados por computador) contra o jogador, mas não há criação de conteúdo em tempo de execução (o *bot* se adapta ao jogador, mas é sempre o mesmo *bot*).

O presente trabalho sugere o uso de algoritmos genéticos em tempo de execução para a criação de novos indivíduos, fazendo com que a evolução seja parte da própria experiência do jogo. No exemplo de jogos separados por fases, por exemplo, a cada nova fase uma nova onda de inimigos mais evoluídos poderia ser gerada de uma população da fase anterior. Essa nova geração de inimigos teria um melhor desempenho que a geração anterior, aumentando o nível de dificuldade do jogo e oferecendo ao jogador novos desafios.

Ainda é possível usar o algoritmo genético como uma referência a própria realidade, introduzindo conceitos de biologia e geografia no jogo. Em um mundo dividido por regiões, a população de cada uma dessas regiões apresentaria seus próprios traços genéticos (com características visuais e comportamentais). O indivíduo de uma população específica poderia dar início a uma nova população de indivíduos, ao se reproduzir

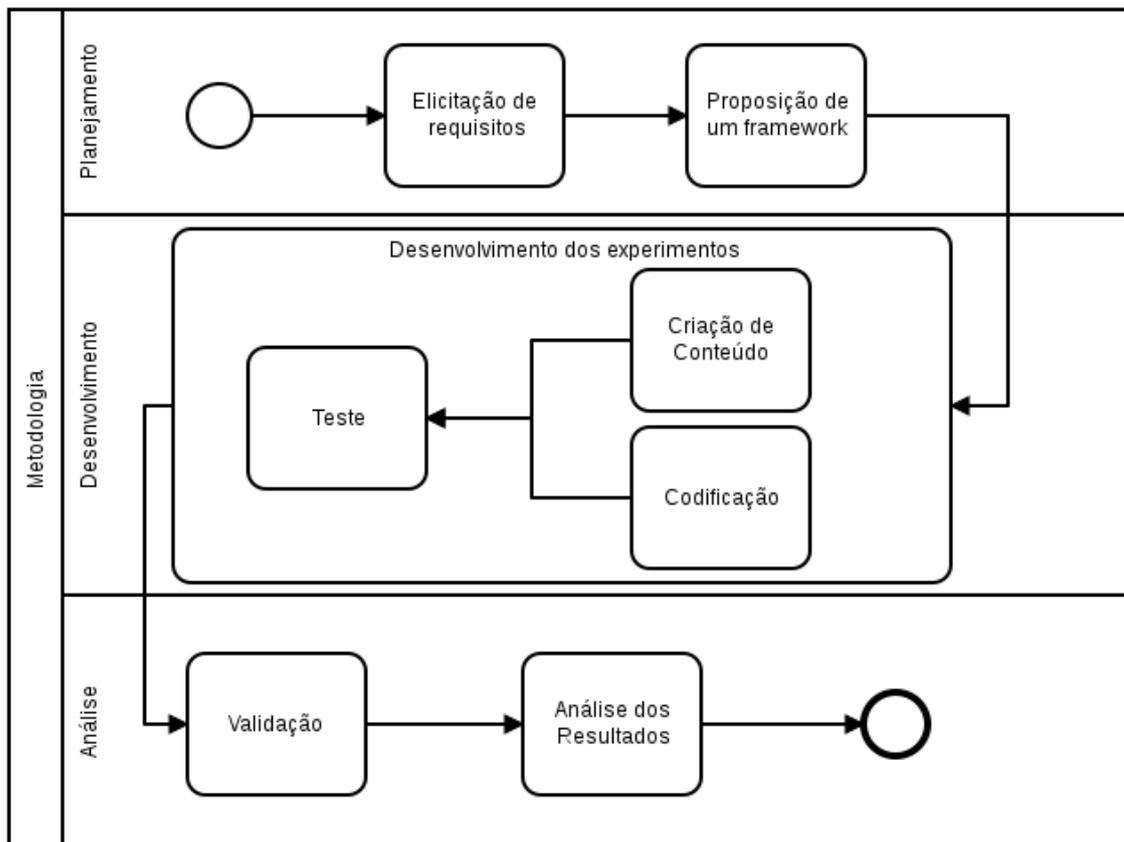
com um outro indivíduo de uma região diferente da sua. Esse tipo de sistema traria uma experiência única ao jogo ao imitar a realidade. Indivíduos dos polos do planeta têm características visuais e comportamentais diferentes de indivíduos dos trópicos, por exemplo. Ao desbravar o mundo, o jogador encontraria sinais dessa biodiversidade regional, despertando seu interesse em permanecer no jogo e desbravar mais e mais do mundo.

## 4 METODOLOGIA

O método científico utilizado neste trabalho é um método comparativo, que, segundo Prodanov e Freitas (2013), procede pela investigação de indivíduos, classes, fenômenos ou fatos, com vistas a ressaltar as diferenças e as similaridades entre eles.

A Figura 5 apresenta o processo de desenvolvimento deste trabalho, o qual é composto por três fases: planejamento, desenvolvimento e análise. Na fase de planejamento são elicitados os requisitos e um *framework* para a implementação do algoritmo genético é proposto. Na fase de desenvolvimento acontece a criação de conteúdo, a codificação e o teste dos experimentos. Na fase de análise, os experimentos são validados e os resultados analisados. Todo o processo possui sete etapas, as quais são detalhas a seguir.

Figura 5 – Processo de desenvolvimento



Fonte: Próprio autor (2019).

### 4.1 Elicitação dos requisitos

Nesta etapa foram elicitados os requisitos que os experimentos deveriam atender. A elicitação dos requisitos é uma parte fundamental da engenharia de software, pois garante que o software certo seja desenvolvido de maneira correta. Ao final desta etapa foi gerado o documento de requisitos, que delimita o escopo do conjunto de funcionalidades que o sistema deve prover.

## 4.2 Proposição de um framework

Durante esta etapa, um *framework* para implementação de algoritmos genéticos foi modelado e desenvolvido. Esse *framework* auxiliou no desenvolvimento dos experimentos, que não precisaram implementar o algoritmo genético do zero, reduzindo o custo e facilitando o reuso. Ao final desta etapa foi gerado um diagrama de classe do *framework* bem como seu código fonte.

## 4.3 Desenvolvimento de Experimentos

Nesta etapa foram desenvolvidos dois experimentos gamificados. Esses experimentos atenderam os requisitos elicitados e também utilizaram o *framework* proposto. No desenvolvimento de experimentos gamificados, é necessário realizar a criação de conteúdo audiovisual, a sua codificação e testes. As etapas de criação de conteúdo e codificação foram realizadas em paralelo, visto que elementos criados usando algoritmo genético necessitam de gráficos modulares.

Nesta etapa também foram realizados testes funcionais e de usabilidade, a fim de garantir uma boa interação entre os usuários e o experimento. Estes testes são importantes, pois existe uma grande diversidade de dispositivos usados para jogar. Esses dispositivos variam em seu tamanho de tela, sistema operacional, interface (alguns possuem teclado e mouse, outros tela sensível ao toque), sendo necessário testar como os experimentos se comportam nesses variados dispositivos.

Pelos experimentos se tratarem de aplicações web, as próprias ferramentas do desenvolvedor, fornecidas pela *Google* no navegador *Chrome*, foram usadas para testes funcionais e de responsividade.

## 4.4 Validação

A validação do experimento 1 foi conduzida através de testes com os usuários e a extração automática de dados. Nos testes com os usuários, o experimento 1 foi executado por um conjunto de usuários, que também executaram um jogo similar que não usa algoritmos genéticos. Durante a execução do experimento 1 que usa algoritmo genético, dados foram extraídos automaticamente e salvos em banco de dados para análise.

Adicionalmente, os usuários responderam a um questionário em que avaliaram os experimentos com e sem o uso de algoritmos genéticos, usando como critérios: número total de inimigos por jogo e padrão como os inimigos atacam o jogador. O fato dos usuários executarem os dois tipos de experimentos permitiu a comparação entre eles.

O experimento 2, que se trata de um projeto em andamento, utiliza algoritmo genético como um de seus sistemas para introduzir biodiversidade nos personagens não jogáveis (NPC's). Ao invés de personagens criados previamente e distribuídos aleatoria-

mente, foram criados alguns poucos personagens e distribuídos em locais específicos do mapa. Esses personagens, por sua vez, têm a capacidade de se reproduzirem, gerando novos personagens que carregam os traços genéticos de seus progenitores. Como método de validação, o sistema de reprodução genética foi executado e alguns desses personagens e imagens foram colhidas para acompanhar o desenvolvimento das novas populações geradas.

#### **4.5 Análise dos Resultados**

Os dados obtidos no processo de validação permitiram três tipos de comparação:

1. Comparação entre um experimento sem e com uso de algoritmo genético, permitindo verificar as vantagens e desvantagens do seu uso;
2. Comparação entre gerações da mesma população, permitindo verificar se indivíduos de uma mesma população compartilham características genéticas;
3. Comparação entre populações, permitindo verificar se indivíduos de populações distintas possuem características genéticas diferentes.

Os resultados dessas comparações são apresentados na forma de gráficos e tabelas. Também, a fim de comparação, imagens colhidas durante o experimento 2 são usadas para ilustrar o resultado do uso do algoritmo genético.



## 5 DESENVOLVIMENTO

Este Capítulo apresenta o desenvolvimento deste trabalho dividido em etapas, como descrito no Capítulo 4. Na Seção 5.1, são apresentados os requisitos não funcionais para a criação dos experimentos. O *framework* proposto é descrito na Seção 5.2. A Seção 5.3 detalha os dois experimentos utilizados para explorar o uso dos algoritmos genéticos. Os testes funcionais e de sistemas são apresentados na Seção 5.4. Já a validação realizada para os dois experimentos é descrita na Seção 5.5.

### 5.1 Elicitação dos Requisitos

Os requisitos não funcionais elicitados para a criação dos experimentos estão especificados a seguir.

- RNF01 - O experimento deve ser capaz de rodar em todas as plataformas.
- RNF02 - O experimento deve ser capaz de rodar em diversos tamanhos de tela.
- RNF03 - O experimento deve rodar sem a necessidade de instalação no dispositivo.
- RNF04 - O experimento deve ser interativo.
- RNF05 - O experimento deve apresentar *feedback* para todas ações executadas.
- RNF06 - O experimento deve implementar algoritmo genético.
- RNF07 - O experimento deve ser *gameficado*.

### 5.2 Proposição de um Framework

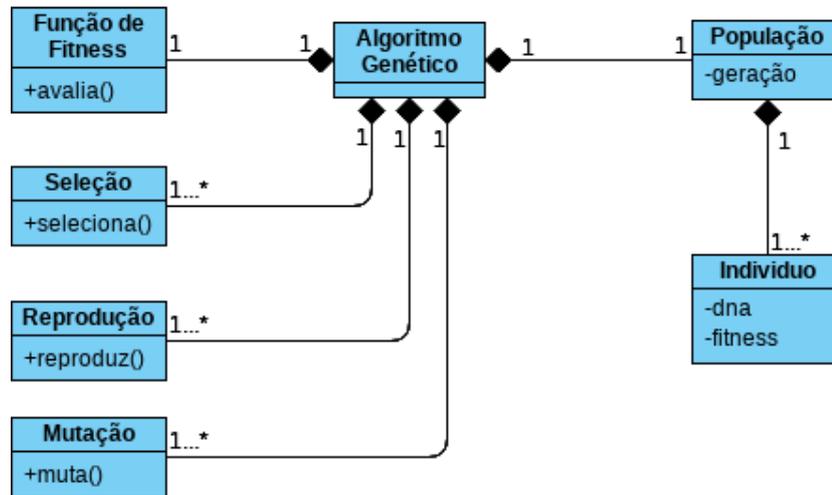
Como vimos na Sessão 2.4 do Capítulo 2, um *framework* auxilia no desenvolvimento de novas aplicações pelo reuso, uma vez que o mesmo código fonte não tem de ser escrito do zero a cada nova aplicação. Nesta etapa temos um protótipo de um *framework* que implementa um algoritmo genético. Este *framework* é genérico e pode ser usado para qualquer problema de otimização. Entretanto, neste trabalho, ele é usado no desenvolvimento de experimentos gameficados.

A Figura 6 apresenta o diagrama de classes de um algoritmo genético. Como pode ser observado, um indivíduo deve ter os atributos *dna* e *fitness*, sendo que *dna* define suas características visuais a comportamentais e *fitness* define sua aptidão. O indivíduo faz parte de uma população.

Uma população tem o atributo *geração* que indica qual a iteração da população atual (por exemplo geração 1, geração 2... geração N). A população faz parte do algoritmo genético. Para que o algoritmo genético funcione, ele precisa de:

- Uma função *fitness*, que é a função capaz de avaliar a aptidão de cada indivíduo;

Figura 6 – Diagrama de classe de um algoritmo genético



Fonte: Próprio autor (2019).

- Um método de seleção, que é como os indivíduos que vão se reproduzir são selecionados;
- Um ou mais métodos de reprodução, que definem como será a combinação de *dna* do novo indivíduo gerado;
- Um método de mutação, que é o método responsável por inserir diversidade no indivíduo.

Todos experimentos desenvolvidos neste trabalho implementaram esse modelo de classes para que haja evolução nos indivíduos.

### 5.3 Desenvolvimento de Experimentos

Dois experimentos gamificados foram idealizados para explorar o uso de algoritmos genéticos em jogos. Ambos experimentos implementam o *framework* proposto neste trabalho, bem como usam algoritmo genético em tempo de execução. Estes experimentos são detalhados a seguir.

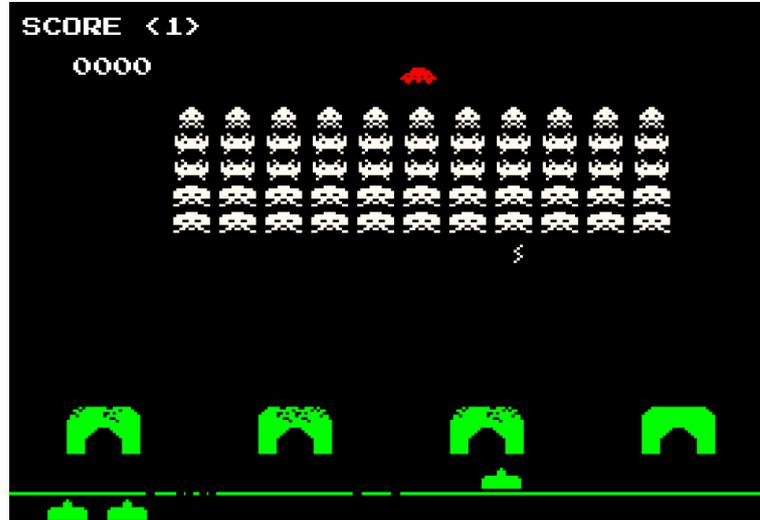
#### 5.3.1 Experimento 1 - InvaderZ

O experimento 1 é um jogo do tipo *Space Invaders* (NISHIKADO, 1978). Neste famoso jogo, alienígenas tentam destruir a terra e o jogador tenta detê-los. O comportamento dos alienígenas, bem como sua aparência são previamente definidos.

Como podemos ver na Figura 7, o jogo *Space Invaders* possui um total de quatro inimigos. Esses inimigos se movem sempre da mesma maneira: os três inimigos na cor branca vão de um lado ao outro da tela e descem em direção ao jogador toda vez que

atingem as extremidades laterais da tela; o quarto inimigo, na cor vermelha, cruza a tela esporadicamente, surgindo do canto direito e indo para o canto esquerdo.

Figura 7 – Captura de tela do jogo Space Invaders



Fonte: (NISHIKADO, 1978)

No experimento 1, intitulado InvaderZ, os inimigos evoluem seu comportamento e sua aparência usando algoritmo genético, tornando o jogo mais desafiador para o jogador. O objetivo de cada invasor inimigo é percorrer toda a distância vertical do mapa, sem ser atingido pelos disparos do jogador, como vemos na Figura 8. Os inimigos ganham pontos referentes à distância percorrida. A função de *fitness* usa a seguinte fórmula matemática (Equação 5.1) para determinar a aptidão de cada invasor:

$$f(d) = \frac{d}{T} \quad (5.1)$$

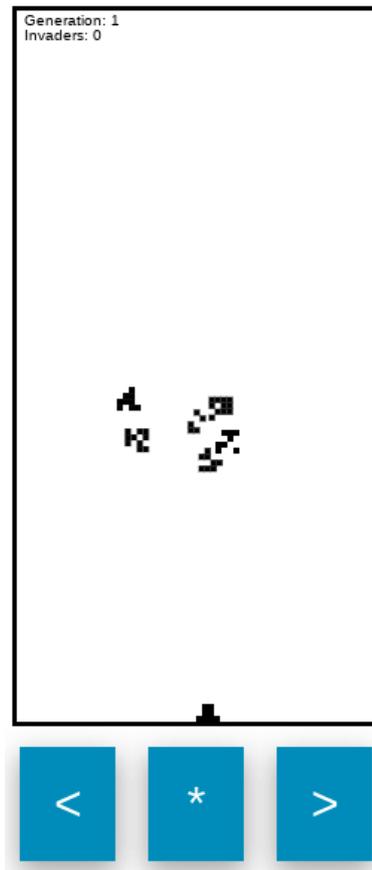
Assim, a aptidão de um indivíduo é igual a distância que ele percorreu ( $d$ ) sobre a distância total do mapa ( $T$ ) (sua distância normalizada).

O trecho do código responsável pela evolução dos inimigos em InvaderZ é apresentado na Listagem 5.1 e explicado a seguir.

Listing 5.1 – Código fonte do InvaderZ responsável pela evolução

```
class Genetics {
    constructor() {
        this.population = new Population(6, 16);
        this.bestOfGeneration;
        this.mutation = new Mutation();
        this.fitness = new Fitness();
        this.selection = new Selection();
        this.reproduction = new Reproduction();
    }
}
```

Figura 8 – Captura de tela do experimento InvaderZ



Fonte: Próprio autor (2019).

```

evolve(){
  this.fitness.evaluate(this.population);
  let newPopulation = new Population(6,16);
  for(let x = 0; x < this.population.size; x++){
    let populationTmp = this.population.copy();
    let a = this.selection.select(populationTmp);
    let b = this.selection.select(populationTmp);
    let child = this.reproduction.reproduct(a,b);
    if( Math.random() < 0.1 ){
      child = this.mutation.mutate(child);
    }
    newPopulation.individuals[x] = child;
  }
  this.population = newPopulation;
}

elitism(){
  this.population.createPopulation();
  let rand = Math.floor(Math.random() * this.population.size);

```

```
let invader = new Invader(w/4/2, Math.random()*-20,  
  this.bestOfGeneration.dna.slice());  
this.population.individuals[rand] = invader;  
}  
  
}
```

Nesse código (Listagem 5.1) pode-se ver as classes Population (população), Mutation (mutação), Fitness (aptidão), Selection (seleção) e Reproduction (reprodução), que são apresentadas na Figura 6. Uma população é composta por indivíduos, que nesse caso são os invasores. Esses invasores tentam cruzar toda a tela passando pelo jogador e evitando seus disparos. Para cada nova geração, que nesse caso é uma nova fase do jogo, é gerada uma nova população.

A evolução da geração é descrita no método *evolve*. Nesse método, a população atual é avaliada de acordo com a função *fitness*. A população atual é então copiada para uma população temporária, de onde dois pais são selecionados. A probabilidade de um indivíduo ser selecionado está diretamente relacionada a sua aptidão. Esses dois pais selecionados então passam por um processo de reprodução em que vão gerar um novo indivíduo, que imediatamente é inserido na nova população. Ao final do processo de evolução, a nova população se torna a população atual do jogo.

Neste experimento também é usado o conceito de elitismo, em que o melhor indivíduo de cada geração é passado para a geração posterior. Especificamente neste experimento, se a cada sete gerações os invasores não conseguirem vencer o jogador, uma nova população aleatória é gerada e apenas o melhor indivíduo das gerações passadas é inserido nessa nova geração.

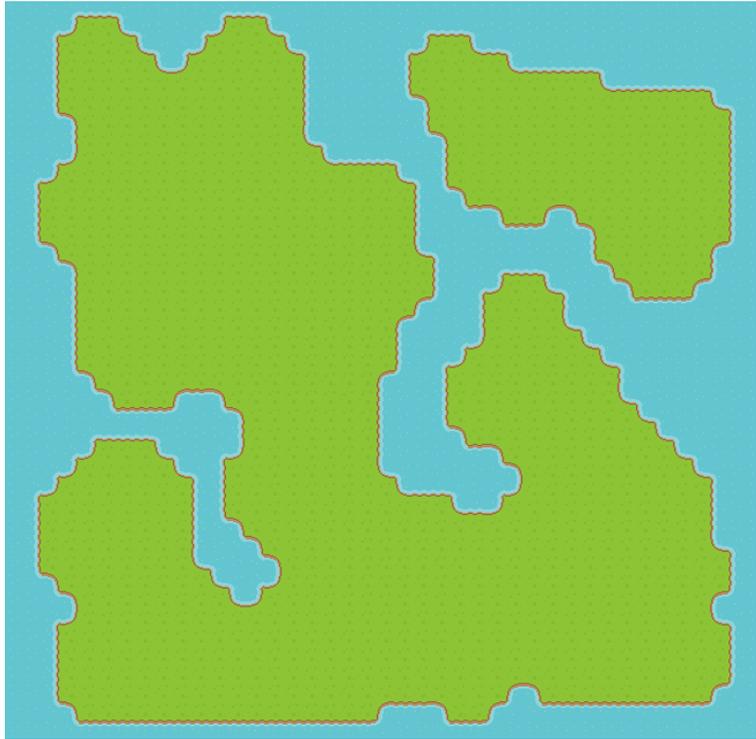
### 5.3.2 Experimento 2 - Sistema de reprodução

O experimento 2 é um jogo em desenvolvimento. A ideia é que este jogo simule um ecossistema real. O mapa é gerado proceduralmente a cada novo jogo. Os personagens não jogáveis (NPC) desse jogo são distribuídos em regiões do mapa, gerando tribos. Essas tribos compartilham características genéticas entre si. Por exemplo, tribos ou populações mais ao Sul do mapa apresentam características diferente das tribos mais ao Norte.

Jogos geralmente são divididos em sistemas. Estes sistemas são responsáveis apenas pelas funções que lhes são atribuídas. Por exemplo, um sistema de inventário tem como objetivo apenas manter os itens que o jogador coleciona ao longo do jogo. O experimento 2 é um sistema de reprodução. Nesse sistema, personagens ou objetos orgânicos (vegetação) podem se reproduzir, gerando novas espécies. Esse sistema tem como objetivo introduzir biodiversidade no jogo, bem como fazer com que novas espécies herdem características visuais e comportamentais de seus progenitores.

A Figura 9 mostra um mapa gerado proceduralmente. Esse mapa é gerado cada vez que jogo é iniciado, fazendo com que a cada nova execução do jogo seja uma nova

Figura 9 – Mapa gerado proceduralmente



Fonte: Próprio autor (2019).

experiência. De igual forma, os habitantes nesse mapa são distribuídos por regiões. A ideia é que cada população dessas regiões tenha características únicas. Fazendo assim com que o jogador queira explorar novas regiões e descobrir novas populações.

Geralmente em um jogo digital tradicional, um personagem possui uma imagem que o representa. No caso deste experimento, a imagem do personagem é criada a partir de seus traços genéticos. Para que isso aconteça, é necessário aspectos visuais do personagem (tipo de corpo, tipo de cabelo, cor da pele, entre outros) tenham sua própria representação gráfica. Na Figura 10 é mostrado o exemplo de gráficos modulares para personagens. Se um personagem tiver a tonalidade de pele 1 e o estilo 1, a imagem desse personagem vai ser a combinação dessas características.

No trecho de código apresentado na Listagem 5.2, pode-se ver que as características genéticas de um personagem são definidas por um *Array*, em que o primeiro elemento representa a tonalidade de pele do personagem e o segundo elemento seu estilo.

Listing 5.2 – Código fonte da classe Personagem

```
class Personagem {  
  
    constructor(dna){  
        this.dna = dna;  
    }  
}
```

```

}

const personagem01 = new Personagem ([1, 2]);

const personagem02 = new Personagem ([3, 1]);

```

Figura 10 – Gráficos modulares



Fonte: Próprio autor.

Separando os gráficos por módulos, é possível criar várias combinações. Como os traços genéticos de cada personagem estão codificados no seu *dna*, quando um personagem cruzar com outro, eles gerarão um novo personagem que terá traços genéticos resultantes da combinação do *dna* de seus pais. Por exemplo, se um personagem de tonalidade 1 e estilo 2 cruzar com um personagem de tonalidade 3 e estilo 1, um possível resultado seria um personagem de tonalidade 1 e estilo 1. Ou seja, ele herda a tonalidade de um de seus pais e o estilo do outro.

#### 5.4 Testes funcionais e de sistema

Os testes funcionais tiveram por objetivo validar os experimentos em relação aos requisitos obtidos através da elicitaco de requisitos. J os testes de sistemas visaram executar o sistema sob ponto de vista de seu usurio final, varrendo as funcionalidades em busca de falhas em relao aos objetivos originais. Os testes foram executados em condioes similares, em ambientes Linux, Windows, iOS e Android utilizando os navegadores Chrome, Firefox e Safari.

Casos de testes foram escritos e rodados nos sistemas e navegadores previamente descritos, com o intuito de garantir a compatibilidade do experimento com o maior nmero possvel de dispositivos. No Apndice A so apresentados os casos de testes criados para o experimento 1. Cada uma das aoes descritas nos casos de testes, como pode ser visto na Tabela 4, foram executadas nos dispositivos e observados os resultados. Se o resultado obtido era o mesmo que o esperado, o teste foi considerado um sucesso. Por exemplo, ao clicar no boto disparar tiro era previsto que um projtil fosse disparado pelo jogador,

Tabela 4 – Caso de Teste TC01

| ID: TC01   |                                     |   |
|--|-------------------------------------|---|
| Descrição: Testar controles da nave - Smartphone |                                     |   |
| Resultado: PASSOU                                |                                     |   |
| Passo  | Ação                                | Resultado Esperado                      |
| 01   | Clicar na seta para o lado esquerdo | Nave se movimentar para o lado esquerdo |
| 02   | Clicar na seta para o lado direito  | Nave se movimentar para o lado direito  |
| 03   | Clicar no botão do meio             | Nave dispara um tiro                    |

se isso acontecesse, o teste era considerado bem sucedido. Esses testes garantiram que o experimento pudesse ser executado nos principais dispositivos do mercado, aumentando o número de possíveis usuários.

## 5.5 Validação

Nesta sessão é apresentada como se deu a validação dos dois experimentos propostos. Para o experimento 1, intitulado *InvaderZ*, foi feito um teste com usuário e também aplicado um questionário. Já o experimento 2, que ainda não possui nome por se tratar de um projeto em andamento, o sistema de reprodução genética foi testado isoladamente, com recolhimento de amostras gráficas geradas durante a execução.

### 5.5.1 Teste com usuário - Experimento 1

A validação do experimento 1 foi realizada nos dias 16 e 25 de outubro de 2019, na sala 102 (Lab 5) da Universidade Federal do Pampa (UNIPAMPA) sob a supervisão da Dr<sup>a</sup>. Aline Vieira de Mello e dos graduandos Guilherme Bolfe e Tobias Tirola, ambos alunos do curso de Engenharia de Software.

Os participantes voluntários do experimento confirmaram sua presença assinando nome, curso e matrícula em um documento. Os mesmos ainda assinaram o termo de consentimento (Apêndice C) e receberam as devidas orientações de como proceder durante o experimento.

O roteiro do experimento foi automatizado. Ao acessar o link em que o experimento estava disponível, o jogo *Space Invaders* era apresentado e o voluntário podia jogar por 10 minutos. Passado esse tempo, o voluntário era redirecionado para o jogo *InvaderZ*, onde deveria jogar por mais 10 minutos. Logo após, o voluntário era redirecionado para o formulário de avaliação, que continha perguntas sobre os dois jogos.

Os dados foram coletados de duas formas. Uma automática, em que dados são coletados e enviados para um banco de dados cada vez que o participante interage com o experimento. Esses dados continham informações sobre até que geração (fase) o participante alcançou, quantos inimigos únicos foram gerados, quantos inimigos o participante

Tabela 5 – Perguntas apresentadas no formulário.

| ID | Tipo    | Pergunta  |
|----|---------|---|
| P1 | Fechada | Você já havia jogado o jogo Space Invaders?   |
| P2 | Fechada | Quantos inimigos únicos você pode notar durante o jogo Space Invaders?                    |
| P3 | Fechada | Você conseguiu identificar padrões na forma como os inimigos se moviam em Space Invaders? |
| P4 | Aberta  | Quantos inimigos únicos você pode notar durante o jogo InvaderZ?                          |
| P5 | Fechada | Você conseguiu identificar padrões na forma como os inimigos se moviam em InvaderZ?       |
| P6 | Fechada | Você concorda que InvaderZ proporciona uma experiência nova a cada execução do jogo?      |

atingiu (sua pontuação) e também informações sobre o navegador e sistema operacional que ele usou para execução do experimento. A outra coleta de dados se deu manualmente, por meio do formulário, em que cada participante pode relatar sua experiência durante a execução do teste com o usuário.

A Tabela 5 apresenta as seis perguntas que compõem o formulário. Apenas a pergunta 4 (P4) é do tipo aberta, as demais são do tipo fechada. As respostas para a P4 foram normalizadas para a geração de gráficos, conforme detalhado no Capítulo 6.

### 5.5.2 Validação - Experimento 2

No experimento 2, a validação se deu por método de observação. Os personagens não jogáveis (NPCs) foram criado com características (visuais e comportamentais) individuais. Esse personagens então foram introduzidos no mundo com a possibilidade de se reproduzirem com outros personagens não jogáveis. Assim, a criação de novos personagens não se trata de algo aleatório, mas uma combinação de características dos seus progenitores. Portanto, em uma determinada região onde existe um tipo de personagem com uma característica distinguível (cor da pele, tipo de cabelo...), é esperado que essa característica seja comum entre a maioria dos habitantes desta região.

O sistema de reprodução (que implementa o algoritmo genético) foi executado isoladamente e foram capturadas telas do jogo a cada nova população criada e a cada novo indivíduo gerado dentro dessa população. As telas capturadas foram então observadas, a fim de verificar se indivíduos da mesma população compartilham características e se indivíduos de populações diferentes possuem características distintas.

No Capítulo 6 é apresentado a análise dos resultados.



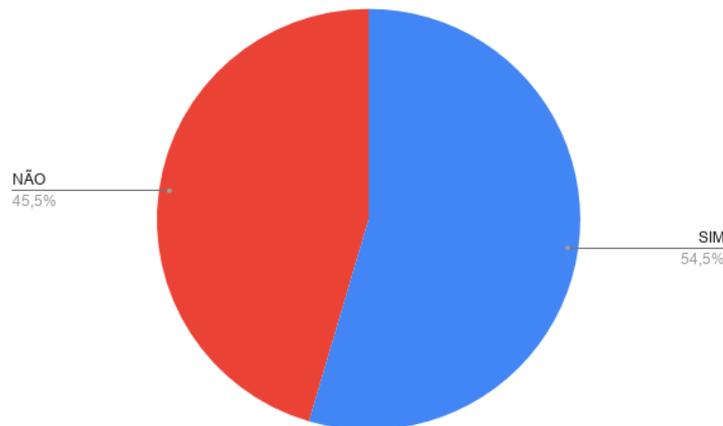
## 6 ANÁLISE DOS RESULTADOS

Este Capítulo apresenta o resultado da análise dos dados resultantes dos dois experimentos realizados. A Seção 6.1 apresenta a comparação entre o jogo Space Invaders (sem o uso de algoritmo genético) e o InvaderZ (com o uso de algoritmo genético). A comparação entre indivíduos da mesma população é apresentada na Seção 6.2, enquanto a Seção 6.3 traz a comparação entre indivíduos de populações diferentes. A análise dos resultados obtidos com os dois experimentos, apresentada na Seção 6.4, possibilita avaliar as vantagens e desvantagens do uso de algoritmos genéticos em jogos.

### 6.1 Comparação entre jogos sem e com uso de algoritmo genético

Vinte e um (21) estudantes da UNIPAMPA participaram dos testes com usuário e responderam o formulário. Como pode ser visto no gráfico da Figura 11, mais da metade dos participantes conhecia o jogo original Space Invaders.

Figura 11 – P1. Você já havia jogado o jogo Space Invaders?

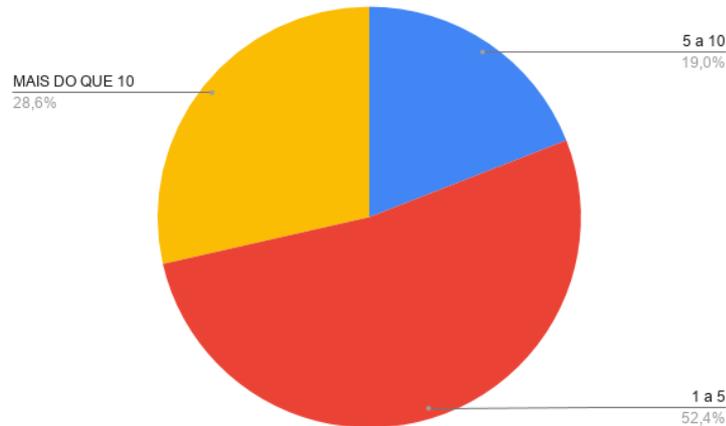


Fonte: Próprio autor.

O gráfico da Figura 12 apresenta o número de inimigos que os participantes indicaram ter percebido durante o jogo Space Invaders. Nota-se que mais da metade dos participantes perceberam que existem menos do que 5 inimigos ao todo no jogo. Um dos prováveis motivos pelo qual alguns participantes responderam que existem mais do que 5, é o fato dos inimigos possuírem uma animação quando se movem, o que pode ter confundido o participante fazendo-o pensar que eram mais inimigos do que realmente eram.

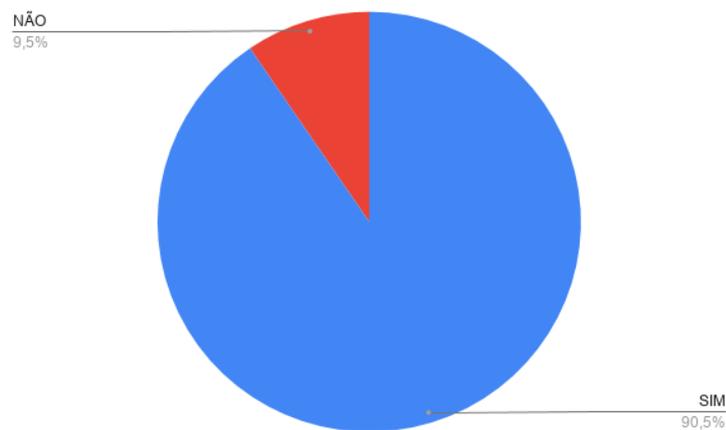
De acordo com o gráfico mostrado na Figura 13, quase todos os participantes puderam identificar o padrão que os inimigos se moviam. Uma vez que o participante consegue identificar o padrão de movimento, ele pode antecipar sua ação e vencê-lo facilmente. Além disso, o padrão, que é sempre o mesmo, pode tornar a experiência de jogo cansativa e massante.

Figura 12 – P2. Quantos inimigos únicos você pode notar durante o jogo Space Invaders?



Fonte: Próprio autor.

Figura 13 – P3. Você conseguiu identificar padrões na forma como os inimigos se moviam em Space Invaders?

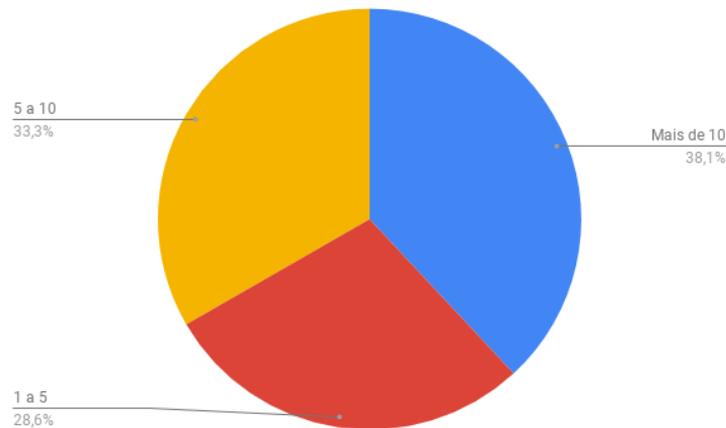


Fonte: Próprio autor.

A pergunta 4 foi apresentada de forma aberta no formulário, em que cada participante poderia responder livremente quantos inimigos únicos ele conseguiu identificar. Para a análise dos resultados, essas respostas foram então normalizadas nos mesmos intervalos apresentados pela pergunta 2. Como se pode ver no gráfico da Figura 14, quase 70% dos participantes puderam notar mais do que cinco inimigos únicos. Acredita-se que alguns dos participantes responderam por geração e não ao todo, ou seja, identificaram 5 ou menos inimigos únicos a cada fase do jogo.

O gráfico da Figura 15 mostra que mais da metade dos participantes não conseguiram identificar padrões na forma como os inimigos se moviam pela tela. Na realidade, não existe padrão definido para os inimigos, mas se um deles “aprende” que indo pelo canto da tela ele tem menor possibilidade de ser abatido, esse comportamento vai ser reprodu-

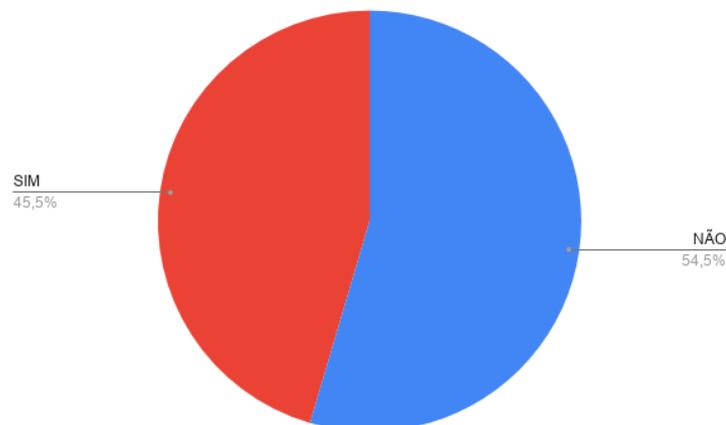
Figura 14 – P4. Quantos inimigos únicos você pode notar durante o jogo InvaderZ?



Fonte: Próprio autor.

zido nas próximas gerações, pois inimigos que vivem mais tem uma maior pontuação de *fitness*.

Figura 15 – P5. Você conseguiu identificar padrões na forma como os inimigos se moviam em InvaderZ?

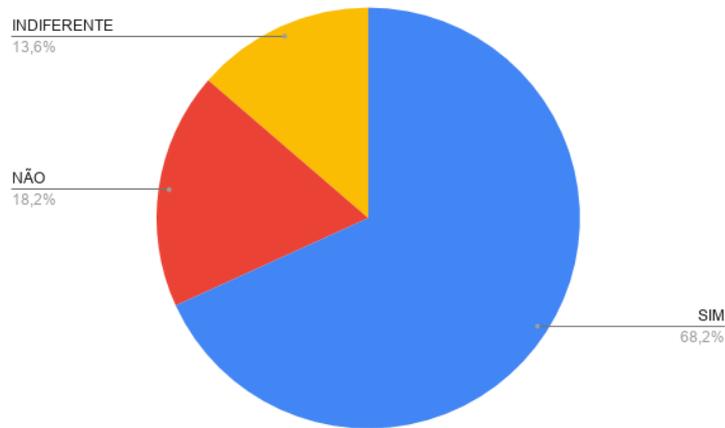


Fonte: Próprio autor.

A Figura 16 mostra que quase 70% dos participantes concorda que InvaderZ proporciona uma experiência nova a cada execução do jogo, uma vez que os inimigos são gerados proceduralmente e a forma como eles avançam em direção ao jogador vai estar diretamente relacionada a forma como o jogador joga. Se o jogador elimina os inimigos mais lentos e que se movem pouco, isso afetara diretamente a próxima geração de inimigos, que serão mais velozes e se movimentarão mais de um lado para o outro, esquivando dos tiros disparados pelo jogador.

A Tabela 6 apresenta os valores máximo (Max), mínimo (Min) e médio (Med) para a pontuação, número de gerações e número de inimigos únicos obtidos pelos participantes

Figura 16 – P6. Você concorda que InvaderZ proporciona uma experiência nova a cada execução do jogo?



Fonte: Próprio autor.

enquanto jogavam o InvaderZ. Esses dados foram coletados de forma automática.

Tabela 6 – Resumo dos resultados dos participantes

|                 | Max | Min | Med |
|-----------------|-----|-----|-----|
| Pontuação       | 297 | 1   | 105 |
| Geração (fases) | 50  | 2   | 19  |
| Inimigos únicos | 192 | 12  | 73  |

Fonte: Próprio autor.

Pode ser observado na Tabela 6 que em média 73 inimigos únicos foram gerados no InvaderZ, com um máximo de 192 e um mínimo de 12 inimigos únicos. Os valores de mínimo e máximo se alinham perfeitamente com a pontuação máxima e mínima colhida. A pontuação mínima igual a 1 e o número mínimo de inimigos únicos criados igual a 12, indicam que um dos participantes jogou apenas até a segunda fase do jogo InvaderZ, visto que esse jogo gera 6 inimigos por fase. Em média, os participantes chegaram até a fase 19.

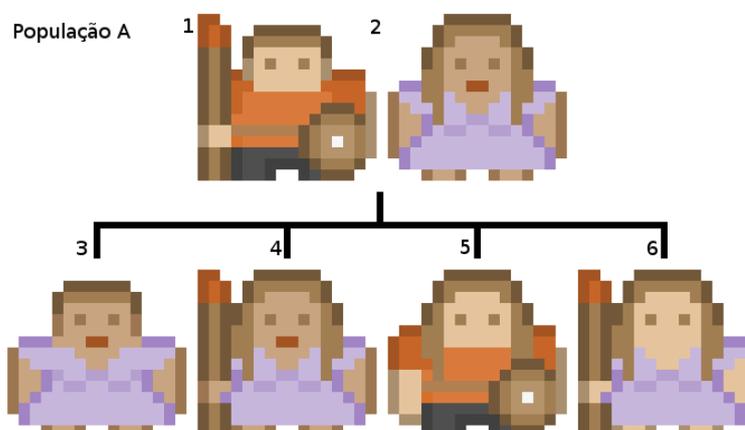
Tanto os dados colhidos automaticamente quanto as respostas ao formulário, apontam que o experimento *InvaderZ* gerou mais inimigos únicos, e que o padrão com que os inimigos se moviam não podia ser identificado. Esses dados confirmam que, enquanto o jogo original *Space Invaders* oferece sempre a mesma experiência de jogo, *InvaderZ* oferece uma experiência única a cada vez que é executado. Assim, a variedade no número de inimigos e a imprevisibilidade de seus movimentos fazem de *InvaderZ* um jogo mais interessante do que a versão original *Space Invaders*.

## 6.2 Comparação entre indivíduos da mesma população

A comparação entre indivíduos da mesma população é realizada sobre o experimento 2. Quatro indivíduos (personagens não jogáveis) foram criados. Dentre as ações que esses personagens podem executar durante o jogo está a reprodução. Para que haja a reprodução, dois indivíduos devem estar perto um do outro. Imagens foram capturadas, mostrando os indivíduos e populações geradas à partir desses quatro indivíduos.

Como pode ser visto na Figura 17, a população *A* começou com dois indivíduos distintos. Um indivíduo com um tom de pele claro, cabelo curto, camisa laranja, calça preta, uma lança na mão esquerda e um escudo na mão direita. O outro indivíduo tem o tom da pele mais escuro, cabelos longos, usa um vestido rosa e não carrega nenhum tipo de armas. Esses dois indivíduos, por meio de reprodução, geraram 4 novos indivíduos. Ao analisarmos cada um deles, podemos ver que são uma combinação de elementos presentes em seus progenitores. O indivíduo *A3* tem o tom de pele e os vestido igual ao progenitor *A2* e o cabelo igual ao do *A1*. O indivíduo *A4* possui todas as características do progenitor *A2*, mas carrega uma lança igual ao progenitor *A1*. O indivíduo *A5* tem o tom de pele, a vestimenta e o escudo do progenitor *A1*, contudo tem o cabelo do progenitor *A2*. Por último, o indivíduo *A6* tem os cabelos e a vestimenta igual o progenitor *A2*, e o tom de pele e a lança do progenitor *A1*.

Figura 17 – Árvore genealógica da população *A*

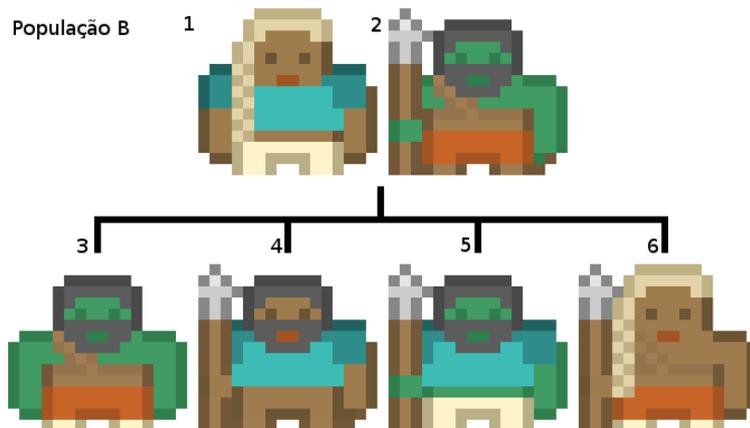


Fonte: Próprio autor.

A mesma observação pode ser feita ao analisar a Figura 18. Dois progenitores de características únicas unindo-se para formar uma nova população *B* de indivíduos que carregam seus traços genéticos. Na Figura 19, podemos ver o progenitor *A1* da população *A* se reproduzindo com o indivíduo *B4* da população *B*, gerando uma nova população *C*.

Ao analisar indivíduos de uma mesma população, vemos que são semelhantes entre si. Se vestem igual, carregam os mesmos tipos de armas, tem o mesmo tom de pele. Essas características, somadas a outros sistemas do experimento, definem o comportamento do

Figura 18 – Árvore genealógica da população B

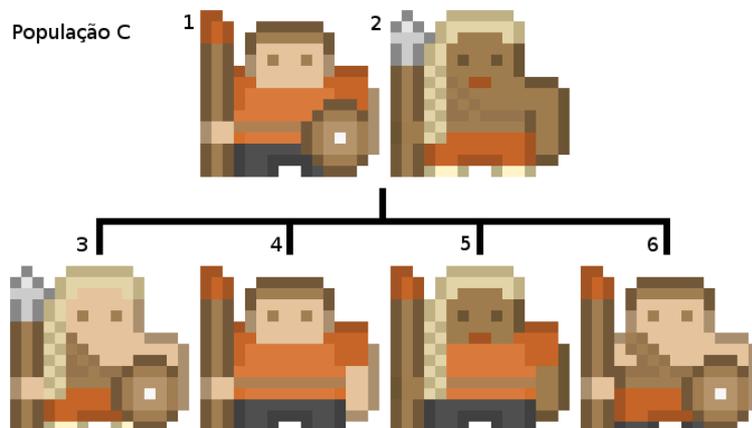


Fonte: Próprio autor.

indivíduo. Por exemplo, indivíduos carregando armas podem ser mais agressivos do que indivíduos desarmados. O algoritmo genético, no experimento 2, insere conceitos de biologia e geografia por esses motivos.

Populações (tribos) são formadas por indivíduos de um mesmo local. Esse indivíduo compartilha características físicas e comportamentais daquela região. Contudo, nada é roteirizado ou pré-definido. Se um indivíduo cruza o mapa, deixando sua região natal para uma nova região, e encontra um novo indivíduo com quem ele se reproduza, uma nova população será gerada, combinando novos traços genéticos, aumentando a diversidade de indivíduos no mundo.

Figura 19 – Árvore genealógica da população C



Fonte: Próprio autor.

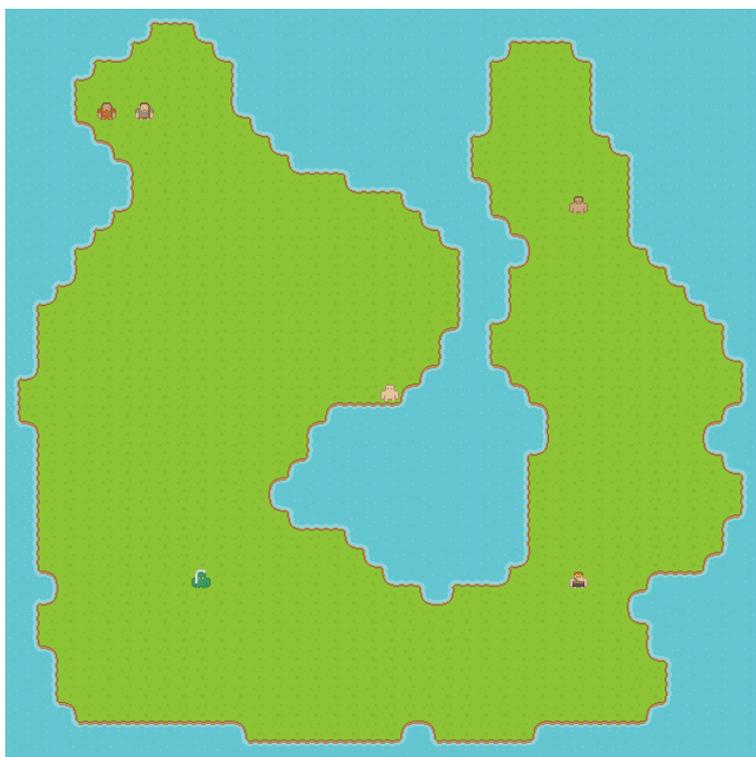
### 6.3 Comparação entre populações

Ao analisar as populações  $A$ ,  $B$  ou  $C$ , vemos que os indivíduos gerados são frutos da combinação genética de seus progenitores. Contudo, ao se comparar uma população com a outra, especialmente no caso das populações  $A$  e  $B$ , vemos que não são nada similares (tom de pele, cor do cabelo ou vestimenta), bem como acontece na vida real.

Tribos ou raças de uma mesma região tendem a se comportar da mesma forma, se vestir de maneira similar, comer as mesmas coisas. O intuito de ser implementar o algoritmo genético no experimento 2 é justamente esse: trazer elementos da vida real para o mundo virtual. Uma vez que para que ocorra a reprodução, dois indivíduos devem estar em uma mesma região, então esta região será habitada por indivíduos similares entre si.

Uma visão simplificada seria dividir o mundo em quatro partes: norte, sul, leste e oeste. Cada uma dessas regiões apresentaria indivíduos similares, que carregam o mesmo tipo de arma, se comportam da mesma maneira. Contudo, a população do norte não necessariamente seria similar a população do sul, uma vez que foram geradas por indivíduos diferentes. Salvo o caso de um indivíduo migrar do norte para o sul e se reproduzir com um indivíduo local, o que pode acontecer, uma vez que não há como prever as ações dos personagens não jogáveis, tornando cada execução do jogo uma experiência única.

Figura 20 – Alguns indivíduos (NPC's) espalhados por um mapa de pequena escala.



Fonte: Próprio autor.

Na Figura 20, que é um mapa em pequena escala, podemos ver alguns indivíduos

espalhados pelos cantos do mundo. Ao levarmos em conta que esses indivíduos precisam estar perto um do outro para que haja a reprodução, podemos começar a perceber o conceito de tribos formando-se. Tribos que compartilham características físicas e comportamentais. Entretanto existe a hipótese de um indivíduo deixar o seu local de nascimento e peregrinar pelo mapa para um novo local, mais afastado. Isso, contudo, contribui para a possibilidade da geração de uma nova tribo, por exemplo, que nasça da mistura de um indivíduo do canto superior direito do mapa com um outro indivíduo do canto inferior esquerdo.

## 6.4 Resumo

Nos dois experimentos realizados, podem ser observadas vantagens na utilização do algoritmo genético. No experimento 1, *InvaderZ*, a criação de inimigos e sua adaptação à forma como o jogador joga, fez com que o jogo oferecesse maior quantidade de conteúdo (gerado proceduralmente, sem a necessidade da intervenção de um programador ou artista gráfico) e trouxe uma experiência única a cada jogo, aumentando seu valor de re-jogabilidade. Essas qualidades fazem um jogo se destacar dos demais e tornam o processo de desenvolvimento menos árduo para pequenos times e desenvolvedores solos, como apresentando no Capítulo 1.

Já no experimento 2, o conceito de reprodução genética trouxe uma experiência única, uma vez que introduziu o conceito de tribos regionalizadas. Isso faz com que o jogador sinta desejo de explorar o mundo e descobrir novas regiões e conhecer seus habitantes. Histórias podem ser escritas acompanhando as ações desses personagens não jogáveis. Ao andar pelo sul de uma região, o jogador pode encontrar um indivíduo de uma população que é característica do norte, provocando uma série de perguntas: Como ele foi parar ali? Ele migrou de sua região natal? Ele é o único? Existem mais iguais a ele? Ele é agressivo? Essas descobertas tornam o jogo mais interessante e prendem a atenção do jogador, que sente cada vez mais vontade de conhecer o mundo em que se passa o jogo. Portanto, o presente trabalho encontrou um bom indício de que o algoritmo genético implementado aumenta a imersão dos jogadores.

Para que o algoritmo genético possa ser implementado no jogo é necessário uma preparação. As entidades que serão os indivíduos no jogo precisam estruturadas de uma forma que seus traços genéticos possam ser vistos no seu comportamento e/ou sua aparência. No caso do experimento 2, os gráficos foram pensados de forma modular, como podemos ver na Figura 10. Com um vetor, podemos representar um indivíduo. Por exemplo, o vetor: [1, 4, 5, 7, 10, 3, 22] indica no experimento que o indivíduo tem o tipo de corpo 1, tipo de cabelo 4, tipo de chapéu 5, objeto na mão esquerda 7, objeto na mão direita 10, tipo de calça 3 e tipo de sapato 22. O método que “desenha” o indivíduo na tela está preparado para exibir diversos tipos de gráficos, baseado no número que ele recebe. Se o número do tipo de corpo for 1, ele sabe que precisa desenhar o gráfico relacionado ao

tipo de corpo 1. Isso pode parecer um tanto quanto trabalhoso, mas uma vez que o código for estruturado dessa maneira, uma infinidade de indivíduos com diferentes combinações poderão ser criadas, sem a necessidade de alteração de código.



## 7 CONSIDERAÇÕES FINAIS

Neste trabalho foi investigado o uso do algoritmo genético em jogos digitais. Foram apresentados alguns jogos que adotam algoritmos genéticos, bem como foram identificadas as finalidades para as quais esses algoritmos são utilizados. Nota-se que apesar de alguns jogos de sucesso comercial usarem o algoritmo genético, muitos deles não exploram o máximo potencial que ele pode oferecer. Por exemplo, a maioria dos jogos não adota o algoritmo genético em tempo de execução para a criação de novos indivíduos, perpetuando a criação conteúdo tradicional, ou seja, de forma estática e sem evolução.

A fim de testar o algoritmo genético no desenvolvimento de jogos digitais, foi proposto um *framework* com o intuito de auxiliar o desenvolvedor. Esse *framework* conta com as classes e métodos necessários para que o desenvolvedor possa implementar o algoritmo genético em seus jogos.

O *framework* proposto foi então utilizado na criação de dois experimentos. No experimento 1, InvaderZ, o algoritmo genético foi o total responsável pela criação dos inimigos. A forma como eles se comportam no jogo e sua aparência, dependem do DNA de cada indivíduo. Indivíduos que se destacavam em seus objetivos, no caso vencer o jogador, são recombinados dando origem a novos indivíduos mais evoluídos. Após a validação do experimento com o usuário, concluiu-se que o jogo, por oferecer um novo desafio a cada execução, se tornou mais interessante do que o jogo no qual ele foi inspirado, *Space Invaders*.

Já no experimento 2, em que o algoritmo genético foi usado como um subsistema, notou-se um grande benefício na criação de personagens não jogáveis: o fato de que eles não precisariam ser criados por um artista gráfico, tarefa que demanda mão de obra e tempo. Uma vez implementado o processo de reprodução genética, novos personagens poderiam ser criados em tempo de execução do jogo. O algoritmo genético trouxe aspectos da vida real para o jogo, oferecendo biodiversidade e regionalidade, observados no mundo real.

O algoritmo genético pode contribuir para a criação de jogos em uma das principais etapas: a criação de conteúdo. Na introdução deste trabalho vimos que o gasto com conteúdo no mercado de jogos é grande. O desenvolvedor podendo ofertar variedade de conteúdo aumentará as chances de sucesso do seu jogo. Conteúdo novo adicional são oferecidos de tempos em tempos, na esperança de manter o jogador interessado no jogo.

Como vantagem, podemos ver que a criação de conteúdo procedural facilita o desenvolvimento de entidades no jogo. Com mais conteúdo, o jogo tem mais chance de se tornar mais interessante e cativar mais o jogador. Se ao executar o jogo, o jogador se deparar sempre com o mesmo tipo de adversário, que se comporta sempre da mesma maneira, uma vez que ele vença o jogo, ele se sentirá desmotivado a jogar aquele jogo novamente. Contudo, estruturando seu jogo de forma a implementar o algoritmo genético, vemos que o desenvolvedor tem opção de criar uma diversidade de conteúdo que segue a uma regra,

ou seja, algo não aleatório. Se ele quer um determinado tipo de vegetação, ele precisa criar apenas alguns exemplos que atendam seus requerimentos e cruzar essas espécies entre si, dando origem a novas espécies similares, mas com suas próprias características. O mesmo se aplica a personagens, animais, e diversas outras entidades.

Como desvantagem, vemos que para que o desenvolvedor usufrua do que o algoritmo genético pode oferecer, ele precisa estruturar seu projeto de acordo. Gráficos precisam ser modulares. Um tipo de vestimenta precisa servir em todos os personagens que possam usar essa vestimenta. O mesmo serve para armas, chapéus, tipos de cabelos, etc. Da mesma forma o *framework* proposto neste trabalho deve ser utilizado e os conceitos de indivíduo, reprodução, pontuação (*fitness*) devem estar presente.

Como trabalhos futuros, pode-se apontar o uso de redes neurais juntamente com algoritmo genético em jogos digitais. Uma das principais dificuldades de se treinar uma rede neural eficiente é a quantidade de dados necessários. Para que a rede neural possa, por exemplo, classificar uma imagem entre cão ou gato, são necessários diversas imagens de cães e gatos para serem usados como exemplo, na etapa de treinamento da rede neural. Da mesma forma, para que uma rede neural pudesse aprender como jogar um jogo, seria necessário apresentar a ela diversos dados de pessoas jogando tal jogo.

O algoritmo genético pode ser usado no processo de treinamento da redes neurais combinando as redes que apresentaram melhor resultado, gerando novas redes neurais mais evoluídas. O processo de evolução é o mesmo apresentado neste trabalho. Uma população de redes neurais deve ser criada e apresentada a um problema cuja a solução é conhecida. As redes neurais que mais se aproximarem da solução devem receber maior pontuação. Assim, as melhores redes neurais vão se reproduzir entre si até que o resultado seja uma rede neural que consegue solucionar o problema proposto.

## REFERÊNCIAS

- ALI, F. F.; NAKAO, Z.; CHEN, Y.-W. Playing the rock-paper-scissors game with a genetic algorithm. In: IEEE. **Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)**. [S.l.], 2000. v. 1, p. 741–745. Citado 2 vezes nas páginas 29 e 30.
- ALVES, T.; COELHO, J.; NOGUEIRA, L. Content generation for massively multiplayer online games with genetic algorithms. In: SPRINGER. **Proceedings of the Computational Methods in Systems and Software**. [S.l.], 2017. p. 37–49. Citado 4 vezes nas páginas 18, 29, 30 e 31.
- CARVALHO, L. F. B. S. de et al. Application of a genetic algorithm based on abstract data type in electronic games' scenarios adaptive evolution. In: IEEE. **2010 Ninth Mexican International Conference on Artificial Intelligence**. [S.l.], 2010. p. 28–33. Citado 3 vezes nas páginas 29, 30 e 31.
- CHAROENKWAN, P.; FANG, S.-W.; WONG, S.-K. A study on genetic algorithm and neural network for implementing mini-games. In: IEEE. **2010 International Conference on Technologies and Applications of Artificial Intelligence**. [S.l.], 2010. p. 158–165. Citado 2 vezes nas páginas 29 e 30.
- COLE, N.; LOUIS, S. J.; MILES, C. Using a genetic algorithm to tune first-person shooter bots. In: IEEE. **Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)**. [S.l.], 2004. v. 1, p. 139–145. Citado 3 vezes nas páginas 29, 30 e 31.
- CRAWFORD, C. The art of digital game design. **Vancouver: Washington State University**, 1982. Citado na página 21.
- DETERDING, S. Gamification: designing for motivation. **interactions**, ACM, v. 19, n. 4, p. 14–17, 2012. Citado na página 21.
- ESA. **2018 Essential Facts About the Computer and Video Game Industry**. theesa.com, 2018. 17 p. Disponível em: <<http://www.theesa.com/esa-research/2018-essential-facts-about-the-computer-and-video-game-industry>>. Citado 2 vezes nas páginas 17 e 18.
- FAYAD, M.; SCHMIDT, D. C. Object-oriented application frameworks. **Communications of the ACM**, ACM, v. 40, n. 10, p. 32–38, 1997. Citado na página 22.
- FRANCISCO, T.; REIS, G. M. Jorge dos. Evolving combat algorithms to control space ships in a 2d space simulation game with co-evolution using genetic programming and decision trees. In: ACM. **Proceedings of the 10th annual conference companion on Genetic and evolutionary computation**. [S.l.], 2008. p. 1887–1892. Citado 2 vezes nas páginas 29 e 30.
- GEE, J. P. What video games have to teach us about learning and literacy. **Computers in Entertainment (CIE)**, ACM, v. 1, n. 1, p. 20–20, 2003. Citado na página 21.
- GOLDBERG, D. E. Genetic algorithms in search. **Optimization, and Machine Learning**, Addison Wesley Publishing Co. Inc., 1989. Citado 3 vezes nas páginas 22, 23 e 24.

HENDRIKX, M. et al. Procedural content generation for games: A survey. **ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)**, ACM, v. 9, n. 1, p. 1, 2013. Citado 3 vezes nas páginas 29, 30 e 31.

HUO, P. et al. Application and comparison of particle swarm optimization and genetic algorithm in strategy defense game. In: IEEE. **2009 Fifth International Conference on Natural Computation**. [S.l.], 2009. v. 5, p. 387–392. Citado 2 vezes nas páginas 29 e 30.

ISHIBUCHI, H.; NAKARI, T.; NAKASHIMA, T. Implementation of genetic algorithms for a spatial ipd game with a generalized objective function. In: IEEE. **IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028)**. [S.l.], 1999. v. 4, p. 248–253. Citado 2 vezes nas páginas 29 e 30.

JOHNSON, D.; WILES, J. Computer games with intelligence. In: IEEE. **10th IEEE International Conference on Fuzzy Systems.(Cat. No. 01CH37297)**. [S.l.], 2001. v. 3, p. 1355–1358. Citado 2 vezes nas páginas 29 e 30.

KHALAFALLAH, A.; ABDEL-RAHEEM, M. Electimize: new evolutionary algorithm for optimization with application in construction engineering. **Journal of Computing in Civil Engineering**, American Society of Civil Engineers, v. 25, n. 3, p. 192–201, 2010. Citado na página 24.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. 2007. Citado na página 27.

KOZA, J. R.; KOZA, J. R. **Genetic programming: on the programming of computers by means of natural selection**. [S.l.]: MIT press, 1992. v. 1. Citado na página 22.

LINDEN, R. **Algoritmos genéticos: uma importante ferramenta da inteligência computacional**. [S.l.]: Brasport, 2006. Citado 2 vezes nas páginas 22 e 24.

LOUIS, S. J.; MILES, C. Playing to learn: Case-injected genetic algorithms for learning to play computer games. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 9, n. 6, p. 669–681, 2005. Citado 2 vezes nas páginas 29 e 30.

NISHIKADO, T. **Space Invaders**. 1978. Disponível em: <<http://www.spaceinvaders.de/>>. Citado 2 vezes nas páginas 38 e 39.

PIVEC, M.; KEARNEY, P. Games for learning and learning from games. **Organizacija**, v. 40, n. 6, 2007. Citado na página 21.

PRATO, G. d.; GONZALEZ, C. A. F.; SIMON, J. P. Innovations in the video game industry: Changing global markets. **Communications & strategies**, IDATE, n. 94, p. 17–38, 2014. Citado na página 17.

PRODANOV, C. C.; FREITAS, E. C. de. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico-2ª Edição**. [S.l.]: Editora Feevale, 2013. Citado na página 33.

RAY, S.; GORDON, V. S.; VAUCHER, L. Evolving qwop gaitis. In: ACM. **Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation**. [S.l.], 2014. p. 823–830. Citado 2 vezes nas páginas 29 e 30.

SALGE, C. et al. Using genetically optimized artificial intelligence to improve gameplaying fun for strategical games. In: ACM. **Proceedings of the 2008 ACM SIGGRAPH symposium on Video games**. [S.l.], 2008. p. 7–14. Citado 2 vezes nas páginas 29 e 30.

SHAKER, N.; TOGELIUS, J.; NELSON, M. J. **Procedural content generation in games**. [S.l.]: Springer, 2016. Citado na página 21.

SUN, C.-T. et al. Genetic algorithm learning in game playing with multiple coaches. In: IEEE. **Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence**. [S.l.], 1994. p. 239–243. Citado 2 vezes nas páginas 29 e 30.



## APÊNDICE A – CASOS DE TESTES

Tabela 7 – Caso de Teste TC02

| ID: TC02                                      |                                 |   |
|---|---------------------------------|---|
| Descrição: Testar controles da nave - Teclado |                                 |   |
| Resultado: PASSOU                             |                                 |   |
| Passo   | Ação                            | Resultado Esperado                      |
| 01  | Apertar tecla “seta esquerda”   | Nave se movimentar para o lado esquerdo |
| 02  | Apertar tecla “seta direita”    | Nave se movimentar para o lado direito  |
| 03  | Apertar tecla “barra de espaço” | Nave dispara um tiro                    |

Tabela 8 – Caso de Teste TC03

| ID: TC03                                 |  |                             |
|--|--|-----------------------------|
| Descrição: Testar destruição de inimigos |  |                             |
| Resultado: PASSOU                        |  |                             |
| Passo                                    | Ação                                   | Resultado Esperado          |
| 01                                       | Esperar inimigo aparecer na tela       | Novo inimigo avistado       |
| 02                                       | Se movimentar até a posição do inimigo | Nave alinhada com o inimigo |
| 03                                       | Clicar no botão do meio                | Nave dispara um tiro        |
| 04                                       | Esperar tiro acertar o inimigo         | Inimigo destruído           |

Tabela 9 – Caso de Teste TC04

| ID: TC04                                    |  |                           |
|---|--|---------------------------|
| Descrição: Testar criação de novos inimigos |  |                           |
| Resultado: PASSOU                           |  |                           |
| Passo                                       | Ação   | Resultado Esperado        |
| 01  | Movimentar a nave até estar alinhada com o inimigo | Nave alinhada com inimigo |
| 02  | Disparar tiro                                      | Inimigo destruído         |
| 03  | Repetir até todos os inimigos estarem destruídos   | Nenhum inimigo na tela    |
| 04  | Esperar novos inimigos surgirem                    | Novos inimigos na tela    |

Tabela 10 – Caso de Teste TC05

| ID: TC05                           |   |                        |
|------------------------------------|---|------------------------|
| Descrição: Testar recomeço de jogo |   |                        |
| Resultado: PASSOU                  |   |                        |
| Passo                              | Ação  | Resultado Esperado     |
| 01                                 | Esperar todos os inimigos cruzarem a linha de chegada | Nenhum inimigo na tela |
| 02                                 | Disparar tiro   | Começo de um novo jogo |
| 03                                 | Esperar novo jogo começar                             | Novo jogo              |

Tabela 11 – Caso de Teste TC06

| ID: TC06                           |  |                           |
|------------------------------------|--|---------------------------|
| Descrição: Testar recomeço de jogo |  |                           |
| Resultado: PASSOU                  |  |                           |
| Passo                              | Ação   | Resultado Esperado        |
| 01                                 | Movimentar a nave até estar alinhada com o inimigo | Nave alinhada com inimigo |
| 02                                 | Disparar tiro                                      | Inimigo destruído         |
| 03                                 | Repetir até todos os inimigos estarem destruídos   | Nenhum inimigo na tela    |

**APÊNDICE B – TERMO DE CONFIDENCIALIDADE**  
**TERMO DE CONFIDENCIALIDADE**

Título do projeto: EXPLORANDO O USO DE ALGORITMO GENÉTICO EM JOGOS DIGITAIS

Pesquisador responsável: Aline Vieira de Mello

Pesquisadores participantes: Victor Queiroz Ribeiro

Campus/Curso: Alegrete / Engenharia de Software

Telefone para contato: (53) 999590923

Os pesquisadores do presente trabalho se comprometem a preservar a privacidade e o anonimato dos participantes cujos dados serão coletados (1) nas respostas dos questionários aplicados antes, durante e depois da utilização dos jogos. Concordam, igualmente, que estas informações serão utilizadas única e exclusivamente para execução da presente pesquisa. As informações somente poderão ser divulgadas preservando o anonimato dos participantes e serão mantidas em poder do responsável pela pesquisa, professora pesquisadora Aline Vieira de Mello e pelo acadêmico pesquisador Victor Queiroz Ribeiro por um período de 5 anos. Após este período, os dados serão destruídos.

Alegrete, \_\_\_ de \_\_\_\_\_ de 2019.

\_\_\_\_\_  
Aline Vireira de Mello  
SIAPE 1894083

\_\_\_\_\_  
Victor Queiroz Ribeiro  
CPF 073.694.186-00



## APÊNDICE C – TERMO DE CONSENTIMENTO

### TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Título do projeto: Explorando Uso de Algoritmo Genético em Jogos Digitais

Pesquisador responsável: Aline Vieira de Mello

Pesquisadores participantes: Victor Queiroz Ribeiro

Instituição: Universidade Federal do Pampa - Unipampa

Você está sendo convidado(a) para participar, como voluntário(a), de testes de usabilidade, diversidade e engajamento no trabalho de conclusão de curso (TCC) da Unipampa intitulado “EXPLORANDO O USO DE ALGORITMO GENÉTICO EM JOGOS DIGITAIS”. Esse trabalho de conclusão de curso tem como objetivo a criação e a avaliação de jogos digitais que fazem uso de algoritmos genéticos. Você pode a qualquer momento pedir esclarecimentos sobre o TCC, os jogos e os testes (informações coletadas, identificação dos responsáveis, etc.). Você também poderá parar de participar a qualquer momento apenas avisando o pesquisador sem sofrer qualquer tipo de penalidade ou prejuízo (você poderá continuar jogando como quiser). Após ler e tirar suas dúvidas sobre as informações a seguir, se aceitar participar da pesquisa, assine no final deste documento, que tem duas cópias. Uma delas é sua e a outra será arquivada pelo pesquisador responsável. Classificação dos jogos: Livre Para Todos Os Públicos Violência: Violência fantasiosa; presença de armas sem violência; mortes sem violência; ossadas e esqueletos sem violência. Sexo e Nudez: Nudez não erótica; ou mesmo sem a presença de nudez; sem a presença de conteúdo sexual. Drogas: Consumo moderado ou insinuado de drogas lícitas sem relevância para a obra.

O que você precisará fazer nos testes:

1. Ir ao local do encontro com seus próprios meios.
2. Jogar cada um dos jogos respeitando a ordem e o limite de tempo indicados pelo pesquisador.
3. Responder algumas perguntas depois de jogar os jogos.

Riscos que você corre ao participar da pesquisa:

1. Se irritar por ter que jogar um jogo que não gosta.
2. Se irritar por não conseguir jogar um ou mais jogos.
3. Se irritar pela demora em ser atendido ao pedir ajuda.
4. Ficar constrangido com alguma pergunta do questionário.

Benefícios da pesquisa:

1. Ajudar o mercado de jogos e desenvolvedores indies.

Participar dessa pesquisa não gera nenhum custo. Você também não receberá qualquer dinheiro ou ajuda financeira para participar da pesquisa ou chegar ao local do encontro. Seu nome e outros dados serão mantidos em sigilo, e as informações coletadas na pesquisa (respostas do questionário) serão guardadas pelos pesquisadores responsáveis.

Os resultados poderão ser divulgados no texto final do TCC, em publicações ou outras formas de divulgação respeitando sempre o sigilo

---

Nome do Participante da Pesquisa

---

Assinatura do Participante da Pesquisa

---

Nome do Pesquisador Responsável

---

Assinatura do Pesquisador Responsável

---

Local e data

**APÊNDICE D – QUESTIONÁRIO - INVADERZ**

**Pergunta 1** - Você já havia jogado o jogo *Space Invaders*?

- Sim
- Não

**Pergunta 2** - Quantos inimigos únicos você pode notar durante o jogo *Space Invaders*?

- 1 a 5
- 5 a 10
- Mais que 10

**Pergunta 3** - Você conseguiu identificar padrões na forma como os inimigos se moviam em *Space Invaders*?

- Sim
- Não

**Pergunta 4** - Quantos inimigos únicos você pode notar durante o jogo *InvaderZ*?

- 1 a 5
- 5 a 10
- Mais que 10

**Pergunta 5** - Você conseguiu identificar padrões na forma como os inimigos se moviam em *InvaderZ*?

- Sim
- Não

**Pergunta 6** - Você concorda que *InvaderZ* proporciona uma experiência nova a cada execução do jogo?

- Sim
- Não
- Indiferente