

# Relatório do trabalho de Linguagens da Programação

Davi de França Vasconcelos Nunes <sup>1</sup>

Henrique Gabriel Gasparelo <sup>2</sup>

Isaías Gouvêa Gonçalves <sup>3</sup>

José Thevez Gomes Guedes <sup>4</sup>

5 de dezembro de 2024

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Apresentação do Jogo</b>	<b>2</b>
<b>3</b>	<b>Escolhas de Design</b>	<b>2</b>
<b>4</b>	<b>Processo de desenvolvimento</b>	<b>3</b>
<b>5</b>	<b>Dificuldades encontradas</b>	<b>3</b>
<b>6</b>	<b>Divisão dos trabalhos</b>	<b>4</b>
<b>7</b>	<b>Considerações Finais</b>	<b>4</b>

---

<sup>1</sup>FGV - Rio de Janeiro/RJ - Ciência de Dados e Inteligência Artificial - Matrícula: 241708053

<sup>2</sup>FGV - Rio de Janeiro/RJ - Ciência de Dados e Inteligência Artificial - Matrícula: 241708055

<sup>3</sup>FGV - Rio de Janeiro/RJ - Ciência de Dados e Inteligência Artificial - Matrícula: 241708012

<sup>4</sup>FGV - Rio de Janeiro/RJ - Ciência de Dados e Inteligência Artificial - Matrícula: 241708005

# 1 Introdução

Neste trabalho foi elaborado um jogo de aventura, ação e RPG, utilizando a linguagem *Python* e a biblioteca *Pygame*. Nomeado "Atwo Survivors", foi inspirado no conceito de *Vampire Survivors*, a partir do qual recebeu suas modificações na mecânica de jogo e funcionalidades. O presente relatório visa apresentar como o jogo foi elaborado, com a divisão das tarefas do grupo, principais características do processo de desenvolvimento e dificuldades observadas.

## 2 Apresentação do Jogo

O Atwo Survivors se inspirou nas mecânicas principais do jogo *Vampire Survivors*, consistindo em um jogo de visão periférica (semelhante aos jogos de RPG tradicionais) em que um personagem principal deve combater diversos inimigos que aparecem de forma aleatória no mapa e sobreviver, até que, após um certo intervalo de tempo, o jogador deve enfrentar um *boss*. Na versão apresentada do jogo, o personagem principal deve vencer dois *bosses* para vencer. Ao longo do jogo, ao combater inimigos, o jogador coleta pontos de experiência, que evoluem seu nível e o permitem aprimorar suas habilidades, adquirir novas armas e evoluí-las.

A diferença central entre o Atwo Survivors e o jogo no qual se inspirou é a mecânica de combate: originalmente o ataque é automático, onde o jogador controla somente os movimentos do personagem e gerencia suas habilidades. No Atwo Survivors, o jogador também é responsável pelos ataques a serem utilizados, selecionando a arma a ser utilizada e em que momento e direção serão feitos os ataques. Tal recurso aumenta consideravelmente a dificuldade do jogo.

## 3 Escolhas de Design

A princípio o jogo possui uma temática medieval, uma vez que o personagem principal utiliza uma armadura. Da mesma forma, as primeiras criaturas e os cenários reforçam essa primeira impressão. No entanto, ao decorrer do jogo, a presença de alguns itens como uma escopeta e o segundo *boss* revelam a escolha de design inusitada que foi adotada para o Atwo Survivors, que mistura elementos medievais com outros de diferentes temáticas. O jogador poderá lidar com ataques mágicos, armas de fogo ou uma simples espada.

Os *sprites* principais<sup>1</sup> do jogo foram obtidos na plataforma *Itch.io* e produzidos pelo artista Foozle<sup>2</sup>.

A sonoplastia do jogo foi elaborada a partir de efeitos sonoros gratuitos de uso livre, e a trilha sonora foi criada com inteligência artificial.

---

<sup>1</sup>Os demais *sprites* podem ser encontrados em <https://claymor3.itch.io/basic-gun-asset-pack> e <https://pixelcreations.itch.io/rpg-items-16x16>

<sup>2</sup>Os arquivos podem ser encontrados no pacote "Lucifer" disponível em: <https://foozlecc.itch.io/>

## 4 Processo de desenvolvimento

O jogo foi desenvolvido sob a lógica de orientação a objetos, em que o código fonte esteve subdividido em arquivos que contêm as classes e seus respectivos métodos, implementando as funcionalidades do jogo. A pasta `/src` está organizada nos seguintes módulos:

- `config.py`: Configurações do jogo (ex.: tamanho da janela, parâmetros do jogo).
- `drop_item.py`: Código relacionado à lógica de queda de itens.
- `enemies.py`: Definições e propriedades das classes de inimigos.
- `enemy_ai.py`: Lógica para a inteligência artificial para a movimentação dos inimigos.
- `game.py`: Lógica principal do jogo e código de inicialização.
- `items_abilities.py`: Lógica para gerenciamento de itens e habilidades do jogo.
- `main.py`: Ponto de entrada do jogo, iniciando o loop principal.
- `main_character.py`: Código para o personagem do jogador (ex.: movimento, interações).
- `map.py`: Lógica para gerenciamento de mapas e tiles.
- `props.py`: Código para lidar com objetos interativos no ambiente do jogo.
- `repositorio_sons.py`: Módulo para gerenciamento de recursos de áudio e reprodução de sons.
- `repositorio_sprites.py`: Módulo para carregar e gerenciar recursos de sprite.
- `sprites.py`: Lógica geral de manipulação e renderização de sprites.
- `ui.py`: Componentes da interface do usuário e sua renderização.

## 5 Dificuldades encontradas

Apesar de ser uma biblioteca por si só repleta de funcionalidades, o *Pygame* ainda assim é significativamente mais difícil de se trabalhar do que uma *game engine* mais popular, como *Unity*, *Unreal* ou *RPG Maker*, uma vez que as principais funcionalidades do jogo, como movimento, animação, dano, dentre outros, devem ser feitas de forma "crua" no código, sem a utilização de *frameworks* prontas e adaptadas à necessidade do desenvolvedor. Desta forma, diversas dificuldades foram encontradas no processo de criação do jogo.

**Movimentação dos inimigos:** Os inimigos do jogo devem atacar o personagem principal, e para isso deve existir um sistema de movimentação que leve o inimigo até o personagem em uma velocidade consistente. Não existe nenhum recurso nativo de Pygame que permite essa movimentação.

A solução adotada, portanto, utiliza um pouco de álgebra linear: é traçado um vetor entre o inimigo e o jogador, esse vetor é então normalizado (tem sua distância encurtada para um, para conter apenas a informação da direção do deslocamento) e depois é inserida na movimentação do inimigo a partir da velocidade padrão. A mesma estratégia foi usada para os *bosses*, com pequenas alterações.

**Spawn de itens e inimigos:** Para que haja uma dinâmica de jogo, ao logo do mapa devem ser gerados itens coletáveis e inimigos a serem combatidos. Uma mecânica que poderia em primeira instância parecer fácil gerou diversas dificuldades ao longo do desenvolvimento, especialmente por conta de diversos *bugs* que foram corrigidos das mais diferentes formas de acordo com a necessidade.

**Ataque à distância:** Assim como a perseguição de inimigos, o ataque à distância foi mais um grande desafio utilizando vetores. No entanto, esse em particular apresenta uma dificuldade a mais: manter a renderização de um ataque em curso desconsiderando possíveis movimentações da câmera do jogo. A solução foi feita por meio de adaptações nas coordenadas atualizadas em tempo real.

## 6 Divisão dos trabalhos

Naturalmente, em um trabalho de dimensões como esse, as tarefas tiveram de ser divididas entre os membros do grupo para que o projeto saísse conforme esperado. As divisões não foram seguidas perfeitamente à risca pois diferentes demandas surgiram ao longo do desenvolvimento, além de que em certos momentos do projeto, nem todos os membros puderam se dedicar integralmente. A divisão fundamental das tarefas ficou da seguinte forma:

- **Henrique Gaspardo:** Interface de usuário, Arquivo principal do jogo, correção de *bugs*, arquivo README, sonoplastia, design.
- **José Thevez:** Personagem principal, movimentação, mecânicas de combate, inimigos, mecânicas dos *bosses*, sonoplastia, mecânicas de animação.
- **Davi França:** Cenário do jogo, *spawn* de itens coletáveis no mapa.
- **Isaías Gonçalves:** Inimigos, animações, sistema de inteligência artificial de perseguição, relatório.

## 7 Considerações Finais

Neste trabalho foi desenvolvido o jogo Atwo Survivors, construído a partir de um código em branco em que foram explorados os mais diversos recursos da biblioteca Pygame.

A experiência de desenvolvimento foi desafiadora, e oportunidade para diversos aprendizados sobre a rotina de desenvolvimento de um projeto mais aprofundado. O resultado final supriu as expectativas de um jogo com todas as principais mecânicas e dinâmicas de progressão, sendo portanto um projeto satisfatório.