

FUNDAÇÃO GETÚLIO VARGAS MATEMÁTICA APLICADA

JEAN GABRIEL DOMINGUETI
RODRIGO DA CRUZ RIBEIRO
ELIANE DA SILVA MOREIRA

BATALHAS MATEMÁTICAS

Rio de Janeiro, Dezembro de 2024

Sumário

1	Introdução	3
2	Processo de Desenvolvimento do Jogo	3
2.1	Design	3
2.2	Implementações Principais	3
2.3	Controles dos Jogadores	3
2.4	Divisão de Tarefas	4
2.5	Dificuldades Enfrentadas	4
3	Como Executar o Projeto	5
4	Conclusão	5
5	Links	5

1 Introdução

Este documento descreve o desenvolvimento do jogo **Batalhas Matemáticas**, inspirado em *Space Invaders*: um clássico de tiro onde o jogador controla um veículo e deve destruir inimigos enquanto desvia de seus ataques.

Conforme o desenvolvimento de **Batalhas Matemáticas** foi progredindo, o jogo tornou-se algo bem distanciado da inspiração inicial por conter dois jogadores que lutam um contra o outro, diferentes paisagens, inimigos e veículos, diversos obstáculos, e muitos outros detalhes que não estão presentes em *Space Invaders*. Este relatório aborda os principais aspectos do desenvolvimento, incluindo lógica de jogo, design e implementação.

2 Processo de Desenvolvimento do Jogo

O desenvolvimento do jogo foi feito em **Python** utilizando a biblioteca Pygame, aplicando conceitos de orientação a objetos. O projeto contou com a abordagem de diferentes elementos, funções e implementações. Segue um breve resumo de cada tópico trabalhado.

2.1 Design

Inicialmente, três cenários do jogo foram desenhados: deserto, oceano e espaço. Cada cenário conta com sprites próprias, desde veículos bem equipados até inimigos que geram *power-ups* para os veículos. Ao todo, há 4 *power-ups*: vida, velocidade, dano e tiro. Tente coletá-los antes que expirem! Os sons do Tubarão que aparece de vez em quando na tela do oceano é um trecho da trilha do filme Tubarão, o som do Verme da Areia que apreço no deserto foi retirado do jogo Terraria e as músicas de fundo foram retiradas do jogo Stardew Valley. Os links para outros sons utilizados serão citados no final do relatório.

2.2 Implementações Principais

Foram implementadas as funcionalidades dos veículos, inimigos, powerups e tiros em arquivos separados assim como telas, botões, unittests e a Main do projeto. Resumidamente, as funções de destaque são:

- Veículos: podem ganhar melhorias com os *power-ups* gerados após a destruição de inimigos;
- Inimigos: possuem uma IA simples para perseguir os veículos;
- Tela: apresenta obstáculos que aparecem ao longo do jogo;
- Unittest: verifica se os principais módulos do projeto estão funcionando corretamente.

2.3 Controles dos Jogadores

Os dois jogadores podem se mover livremente, rotacionar e disparar uns contra os outros e contra inimigos (cuidado, os inimigos também podem atacar você!). As teclas de controle são:

- **Jogador 1:**

- a - Esquerda
- d - Direita
- w - Cima
- s - Baixo
- c - Rotação anti-horária
- v - Rotação horária
- b - Disparo

- **Jogador 2:**

- LEFT - Esquerda
- RIGHT - Direita
- UP - Cima
- DOWN - Baixo
- COMMA - Rotação anti-horária
- PERIOD - Rotação horária
- SEMICOLON - Disparo

2.4 Divisão de Tarefas

O desenvolvimento do jogo foi dividido entre os membros da equipe da seguinte maneira:

- **Eliane Moreira:** Funcionalidades básicas dos veículos, inimigos e *power-ups*.
- **Rodrigo Ribeiro:** Rotação, disparo e melhorias das funcionalidades básicas.
- **Jean Domingueti:** Design, obstáculos, colisão e estruturação principal do jogo.

2.5 Dificuldades Enfrentadas

Principais dificuldades enfrentadas e soluções implementadas:

- **Implementação dos limites:** Os veículos e inimigos estavam saindo dos limites da tela e sobrepondo-se uns aos outros. Para mantê-los dentro dos limites e não se sobreponem foram feitos ajustes nas colisões adequando-as aos problemas encontrados.
- **Implementação da rotação e dos tiros:** Realizar os cálculos matemáticos para garantir que o ângulo em que a nave está apontando é correto, disparo do tiro a partir desta direção se certificando que o módulo do seu vetor velocidade se mantém constante e fazer com que ao um inimigo morrer os seus tiros já disparados continuem na tela. Para resolver os desafios foram utilizadas as funções *cos* e *sin* da biblioteca **math** e a criação de uma lista geral que guarda todos os tiros dos inimigos.

3 Como Executar o Projeto

Clone o repositório <https://github.com/JDomingueti/batalhas-matematicas> , navegue até o diretório do jogo e execute os passos a seguir.

Crie um ambiente virtual utilizando o comando:

```
python -m venv venv
```

Ative o ambiente virtual (Windows):

```
venv\Scripts\activate
```

Ative o ambiente virtual (Unix ou MacOS):

```
source venv/bin/activate
```

Para instalar os pacotes necessários execute:

```
pip install -r requirements.txt
```

Para verificar os unittests execute:

```
cd src/tests.py
```

Para iniciar o jogo execute:

```
cd src/main.py
```

Divirta-se!

4 Conclusão

O desenvolvimento do jogo Batalhas Matemáticas foi desafiador e enriquecedor colaborando para aumentar o conhecimento na área de planejamento e criação de jogos.

5 Links

Som explosão - (0:21): [YouTube - Som Explosão](#)

Crab rave - (1:10): [YouTube - Crab Rave](#)

Águas vivas: [Pixabay - Águas Vivas](#)

Bola de vento: [Pixabay - Bola de Vento](#)

Gafanhotos: [Pixabay - Gafanhotos](#)

Space: [YouTube - Space](#)

Cometa: [Pixabay - Cometa](#)

Laser inimigos: [Pixabay - Laser Inimigos](#)

Laser player: [Pixabay - Laser Player](#)