



Fundação Getúlio Vargas
Escola de Matemática Aplicada

Relatorio A2 - Linguagem de Progamação

Everton Reis
Artur Teles
Vinicius dos Santos

Sumário

1 O objetivo do jogo

2 informações técnicas

- 2.1 estrutura do projeto
- 2.2 dependências
- 2.3 como executar o jogo
- 2.4 Como executar os testes unitários

3 Principais informações do jogo

- 3.1 Menu
- 3.2 Game Level
 - 3.2.1 Mapa
 - 3.2.2 Player
 - 3.2.3 Barras de vida
 - 3.2.4 Inimigos
 - 3.2.5 Chave
- 3.3 Game Over
- 3.4 Win
- 3.5 Configurações do jogo

4 Informações secundarias

- 4.1 assets
- 4.2 Músicas

5 Divisão na organização

6 Referências

- 6.1 Font
- 6.2 Sprite
- 6.3 Música

1 O objetivo do jogo

Nomeado **Key to the forgotten**, o jogo ocorre em um cenário de plataforma, em que o objetivo principal é eliminar os inimigos ao longo do caminho, adquirir experiência e subir de nível para obter buffs que irão ajudar a derrotar os três bosses(chefes) e recolher as chaves.

2 informações técnicas

Certifique-se de rodar o jogo em um ambiente virtual para evitar a instalação de pacotes que, no futuro, possam se tornar inutilizáveis ou causar conflitos com outros projetos.

Com o Python instalado (preferencialmente versões 3.9 ou superiores), rode o seguinte comando no terminal para inicializar seu ambiente virtual:

```
python -m venv nome_do_ambiente
```

Substitua `nome_do_ambiente` pelo nome desejado para o seu ambiente virtual. Por exemplo, `venv`.

Para ativar o ambiente virtual, execute o seguinte comando:

No Prompt de Comando:

```
nome_do_ambiente\Scripts\activate
```

No PowerShell:

```
nome_do_ambiente\Scripts\Activate.ps1
```

A depender do sistema operacional, a instalação pode ser diferente. Verifique para sua máquina; a opção apresentada é condizente com Windows.

Após a ativação, você verá o nome do ambiente virtual entre parênteses no início da linha de comando, como `(nome_do_ambiente)`.

2.1 estrutura do projeto

O projeto, disponível em **`**link**`**, está organizado da seguinte maneira:

- Key-to-the-forgotten
 - docs
 - src
 - assets

- test
- requirements.txt

2.2 dependências

As dependências estão no diretório **Key-to-the-forgotten**, arquivo requirements.txt. Segue a lista abaixo :

Pacote	Versão
pillow	11.0.0
24.3.1	pip
pygame	2.6.1
PyTMX	3.32

2.3 como executar o jogo

Após o download do projeto, certifique-se de está no mesmo nível do repositório **Key-to-the-forgotten**. Para isso, basta verificar se, no terminal de comando, a nomenclatura da máquina termina em:

```
Key-to-the-forgotten
```

Após isso, verifique-se se o ambiente virtual está instalado corretamente e, no terminal, utilize o comando abaixo para baixar os pacotes requeridos:

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

Para rodar o jogo, acesse o diretório **src**, e execute o seguinte comando:

```
python main.py
```

2.4 Como executar os testes unitários

Os testes unitários estão no arquivo **game_tests.py**, que pode ser acessado pelo seguinte caminho:

```
Key-to-the-forgotten/tests
```

Para rodalos, acesse o diretório **tests** e execute o seguinte comando:

```
python game_tests.py
```

3 Principais informações do jogo

3.1 Menu

A tela de **Menu** é composta por dois botões:

- **Start**: Inicia o jogo.
- **Exit**: Fecha o jogo.

Ao clicar em **Start**, o jogo é iniciado.

3.2 Game Level

3.2.1 Mapa

O mapa é feito utilizando a biblioteca PyTMX ...

3.2.2 Player

O player tem suas principais funções descritas no arquivo python **player.py**, que pode ser acessado pelo seguinte caminho

`Key-to-the-forgotten\src`

Vale destacar as seguintes funções do projeto:

- **collide**: Responsável por verificar toda a mecânica de colisão, utilizando um sistema de sobreposição (overlap). Especificações detalhadas sobre seu funcionamento estão descritas diretamente no código.
- **die**: Responsável por verificar a condição de morte do jogador. Essa função é chamada no arquivo **game.py** e sua resposta é repassada ao .
- **dash**: Responsável por executar o dash ao clicar no botão direito do mouse, desde que a habilidade esteja habilitada com um buff.
- **calc_next_level** e **calc_exp**: Responsáveis por calcular a experiência (EXP) do jogador e determinar quando ele deve subir de nível (upar).
- **on_event**: Responsável por verificar os eventos do jogo, incluindo cliques do mouse. Essa função é crucial, pois a mecânica de ataque do jogador depende dela. Ao clicar com o botão esquerdo, o jogador dispara balas (pequenos **rects** criados no arquivo **bullets.py**), considerando a direção do clique em relação à posição atual do jogador. Além disso, ao clicar com o botão direito, ativa o **dash**, se estiver disponível.

3.2.3 Barras de vida

No arquivo python **bars.py** é encontrado as funções que definem as barras de vida do player e boss e a barra de experiência do player. Podem ser acessadas pelo seguinte caminho

Key-to-the-forgotten\src

3.2.4 Inimigos

Os inimigos possuem uma classe base em comum, nomeada **BaseEnemy**, que pode ser encontrada no arquivo Python **enemy.py**, seguindo o caminho:

Key-to-the-forgotten\src\enemy.py

Essa classe contém características comuns a todos os inimigos, como vida, velocidade, dano, entre outras.

Mais especificamente, para os tipos de inimigos, cada um com atributos definidos em sua respectiva classe (em **gamesettings.py**), temos:

1. **WeakMovingEnemy**: Inimigos do tipo movimentação cuja fonte de dano é a colisão com o jogador. São mais fracos que os **StrongMovingEnemy**, têm dano inicial de contato de 20 e vida de 100.
2. **StrongMovingEnemy**: Inimigos do tipo movimentação cuja fonte de dano é a colisão com o jogador. São mais fortes que os **WeakMovingEnemy**, têm dano inicial de contato de 30 e vida de 200.
3. **ShootingEnemy**: Inimigos do tipo atirar cuja fonte de dano é por meio de balas (**rects**), criadas na classe **bullets**. Possuem vida baixa em comparação com os inimigos que colidem (apenas 150), mas têm alto dano (30) e o dobro do raio de ataque dos inimigos do tipo movimentação.
4. **BossShootingEnemy**: Inimigos do tipo atirar cuja fonte de dano é por meio de balas (**rects**) criadas na classe **bullets**. Surgem quando o **Boss** utiliza o terceiro tipo de ataque. Esses inimigos possuem uma boa vida (200) em comparação com os inimigos que colidem, além de terem alto dano (30) e o dobro do raio de ataque. Ao contrário dos inimigos padrões, esses se movimentam seguindo o **Boss**.
5. **Boss**: O chefe do jogo, um inimigo do tipo atirar que surge a cada 6 andares. É o inimigo mais forte do jogo. Ao contrário dos inimigos padrões, movimenta-se automaticamente para cima da tela. O **Boss** possui três tipos de ataque, que ocorrem com probabilidades predefinidas, calculadas com o auxílio da função **random**, do módulo **random**:
 - **attack1**: Um ataque com uma bala única, direcionada ao personagem com base na posição do **Boss**.

- **attack2**: Um ataque que dispara balas em 20 direções ao redor do **Boss**.
- **attack3**: Invoca os **BossShootingEnemy** ao redor do **Boss**.

3.2.5 Chave

A lógica da chave é implementada na classe **Key**, localizada no arquivo **enemy.py**. Essa classe é responsável por verificar se o jogador coletou a chave ou não (isto é, se a colisão ocorreu).

A contabilização das chaves coletadas é realizada pela variável **self.keys_collected**, presente na classe **GameManager**, localizada no arquivo **game.py**.

3.3 Game Over

A tela de **Game Over** é composta por dois botões:

- **Reset**: Retorna o jogador ao menu principal.
- **Exit**: Fecha o jogo.

3.4 Win

A tela de **Win** (vitória) aparece na seguinte condição: a cada 6 andares, o **Boss** surge, e ao derrotá-lo, uma chave é dropada (disponibilizada) para o jogador coletá-la. Após coletar as 3 chaves (derrotando o **Boss** 3 vezes), o jogador é considerado vencedor, e a tela de vitória (**Win**) é exibida.

Nessa tela, são apresentados dois botões:

- **Menu**: Retorna o jogador ao menu principal.
- **Exit**: Fecha o jogo.

3.5 Configurações do jogo

As configurações do jogo e os parametros podem ser encontrados em **gamesettings.py**, que pode ser acessada pelo seguinte caminho:

```
Key-to-the-forgotten\src
```

4 Informações secundarias

4.1 assets

As sprites do jogo estão no directorio **assets**, que podem ser acessadas pelo seguinte caminho

`Key-to-the-forgotten\assets\sprites`

Estão organizadas da seguinte maneira:

- **assets**
 - boss
 - bossenemy
 - buffs
 - menu
 - shootenemy
 - strongenemy
 - weakenemy
 - **key**

As referencias para os assets estão no final desse documento.

4.2 Músicas

As músicas do jogo estão no diretorio **sfx**, que pode ser acessado pelo seguinte caminho

`Key-to-the-forgotten\assets\sfx`

Está organizado da seguinte maneira:

- **sfx**
 - player
 - strenemy
 - shootenemy
 - weakenemy
 - boss
 - key
 - background

5 Divisão na organização

Artur : criou algumas das classes do jogo, como Player, BaseEnemy, Bullet, Weapon etc e seus principais métodos. Também implementou o sistema de buffs e level up, o algoritmo de geração de mapa e adicionou os sprites do player e dos inimigos e os efeitos sonoros.

Everton: responsável pela criação inicial do mapa e a base do sistema de colisão, barras de vida, interação entre estados do jogo, divisão e organização do projeto, relatorio e testes unitários.

Vinicius: Sprites mapa e criação do mapa no Pytmx.

6 Referências

6.1 Font

[1] Press+Start+2P

6.2 Sprite

[2] pixel-fantasy-caves

[3] free-characters-with-melee-attack-pixel-art

6.3 Música

[4] Música Menu