要件定義

タイトル「ぴよぴよ戦士」シューティングゲーム

①ゲーム概要

- ・プレイヤーはことりを左右に操作し、弾を発射して敵を倒す。
- ・敵や弾はスプライトとして画面上に描画される。
- ・敵に弾が当たるとスコアが1点加算される。
- ・敵がプレイヤーに接触するとゲームオーバー
- ・敵がプレイヤーを通過して画面外へ出てもゲームオーバー。
- ・上位5件スコアは保存され、次回起動時に表示される。

②機能要件

【プレイ機能】

項目 詳細 プレイヤー操作 左右キーで自機を移動。スペースキーで弾を発射。 弾発射 弾は画面上方向に移動。画面外に出ると消去される。 敵は画面上部から下方向に出現・移動。 敵出現 当たり判定 弾と敵が衝突すると敵と弾を削除、スコア加算。 プレイヤー衝突判定 敵がプレイヤーに衝突するとゲームオーバー ゲームリスタート Rキーでスコアリセット・敵弾リセット・ゲーム再開。 Qキーでゲーム終了 ゲーム終了

【画面表示】

項目 詳細

スタート画面 名前入力欄とゲーム開始指示 (Enterキー) ゲーム画面 プレイヤー・敵・弾を描画、スコア表示 ゲームオーバー画面 最終スコア、上位スコア一覧表示

③非機能要件

リソース解放 終了時はpygame.quit()とsys.exit()でメモリ解放 スコア保存 スコアデータをjsonファイルに保存して次回以降も参照可能

④操作方法

操作キー 動作

ゴ:: プレイヤー**名**を入力 文字キー スタート画面からゲーム開始 Enter + -←/→‡− プレイヤーを左右に移動

Space +-弾を発射

ゲーム中にリスタート(ゲームをリセットして再開) R + -

ゲーム終了 Qキーまたは閉じるボタン

⑤スコア仕様

- ・敵を倒すごとに固定スコア(例: **+1点)**。
- ・上位スコアはトップ5件まで保持。
- ・スコアは名前と紐づけて保存。

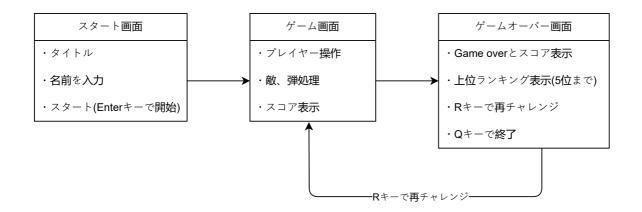
6開発環境・ライブラリ

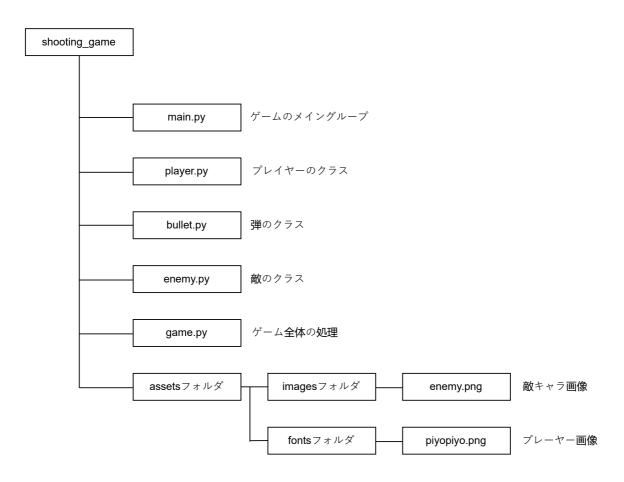
Python-3.13.3 pygame ライブラリ VS⊐ード

⑦使用方法

(1) パソコンにPython-3.13.3とVSコードインストールしてご準備いただき、VSコードの拡張機能にPythonを入れてください。(2) VSコードのターミナルでpip install pygameを実行し、pygame ライブラリをインストールしてください。

(3) VSコードでshooting_gameフォルダを開いていただき、main.pyを右クリックしターミナルでpythonファイルを実行してください。 ★のんきな見た目に仕上げました。お楽しみいただけますと幸いです。





main(ゲーム起動部)

【責務】

- ・pygameの初期化
- ・Gameクラスのインスタンス**生成**
- ・メインループの**実行**

【主な処理】メソッド(内容)

pygame.init() (pygameライブラリの初期化)

game = Game() (Gameクラスのインスタンスを作成)

while True: (無限ループでゲームを回す(イベント処理・更新・描画))

for event in pygame.event.get() (ユーザーのイベントを取得(キー入力など))

game.handle_input(event) (イベントをGameクラスに渡して入力処理)

game.update() (プレイヤー、弾、敵などすべてのオブジェクトを更新)

game.draw() (現在の状態に応じて画面を描画)

pygame.display.flip() (画面を更新して描画結果を反映)

clock.tick(FPS) (フレームレートを一定に保つ)

pygame.quit(), sys.exit() (終了時にPygameを閉じてプログラムを終了)

Gameクラス(ゲーム全体を管理)

【責務】

- ・ゲームの状態管理 (開始・進行・終了)
- ・スコア・ランキングの管理と保存
- ・画面描画・更新の制御 (ゲームループの中心)

【主な処理】メソッド(内容)

__init__ (各クラスの初期化、フォント・スコアデータの読み込み)

handle_input() (キー入力処理 (名前入力 / ゲーム操作 / リスタート))

update() (プレイヤー・弾・敵の更新、衝突判定)

draw() (現在の状態に応じて画面描画 (タイトル・ゲーム中・ゲームオーバー))

save_score() (スコアランキングを保存 (ゲームオーバー時に1回だけ呼ばれる))

restart_game() (プレイヤー・弾・敵などの初期化、ゲーム再開準備)

【他クラスとの相互作用】

- ・Player/Bullet/Enemyクラスのインスタンスを生成・管理
- ・pygameのイベント検知、描画、更新処理と連携

Playerクラス(プレイヤーの制御)

【責務】

- ・キー**入力**による**移動・射撃**の制御
- ・プレイヤーの位置・見た目の管理

【主な処理】メソッド(内容)

update(keys) (左右移動の更新、画面端での移動制限)

shoot() (Bulletインスタンス生成 (弾発射))

【他クラスとの相互作用】

- ・gameクラスから制御を**受**ける
- ・弾 (Bulletクラス)を生成

Bulletクラス(プレイヤーの弾)

【責務】

弾の**移動処理**

敵との当たり判定

【主な処理】メソッド(内容)

update() (弾を上方向に移動)

rect 当たり判定用に使用される

【他クラスとの相互作用】

- ・gameクラスのbullet_groupに追加される
- ・enemyクラスと衝突判定(groupcollideで管理Gameクラスに記述)

Enemyクラス(敵キャラ) 【責務】

- ・敵キャラの**生成と移動**
- ・プレイヤー衝突時の**ゲームオーバー判定**

【主な処理】

__init__() (ランダムな<u>位置</u>に敵を**生成)**

update() (敵を下方向に移動、画面外に到達したら削除 or ゲームオーバー)

【他クラスとの相互作用】

- ・**Gameクラス**のenemy_groupに追加される
- ・Bullet/Playerクラスと衝突時に消滅、またはゲームオーバー

