

# Dokumentacja

## Opis ogólny:

Program został napisany w języku Python przy użyciu biblioteki Pygame. Jest to symulacja w stylu prostej gry komputerowej 2D. Symulacja składa się z określonej ilości rund. Na każdym poziomie gracz symuluje walkę z przeciwnikiem. Punkty zdrowia przeciwnika zwiększają się adekwatnie do poziomu gry. Punkty zdrowia gracza zwiększają się w następujący sposób: na początku każdej rundy gracz losowo podchodzi do jednego z dwóch osadników, gdy podejdziesz do pierwszego jego punkty zdrowia wzrosną o losową liczbę z zakresu od 200 do 250, lecz gdy podejdziesz do drugiego punkty zdrowia zwiększają się o losową liczbę z zakresu od 250 do 300. Symulacja kończy się, gdy gracz straci wszystkie punkty zdrowia.

## Klasy:

### 1. Entity

Klasa `Entity` reprezentuje ogólną encję w grze, która może być rysowana na ekranie. Obejmuje postacie graczy i potworów.

Metody:

- `\_\_init\_\_(image\_path: str, pos: Tuple[int, int], hp: int)` : Inicjalizuje obiekt Entity.
  - `image\_path` : Ścieżka do pliku obrazu encji.
  - `pos` : Pozycja początkowa encji na ekranie.
  - `hp` : Początkowa ilość punktów zdrowia encji.
- `draw\_hp(screen: pygame.Surface) -> None` : Rysuje pasek stanu zdrowia encji na ekranie.
  - `screen` : Powierzchnia pygame, na której rysowany jest pasek stanu zdrowia.
- `move\_towards(target\_rect: pygame.Rect, step: int = 3) -> None` : Przesuwa encję w kierunku podanego prostokąta.
  - `target\_rect` : Prostokąt, do którego encja ma się przesunąć.
  - `step` : Szybkość przesunięcia encji.

## 2. Player

Klasa `Player` reprezentuje gracza w grze.

Metody:

- `\_\_init\_\_(name: str, image\_path: str, pos: Tuple[int, int])`: Inicjalizuje obiekt Player.
  - `name`: Nazwa gracza.
  - `image\_path`: Ścieżka do pliku obrazu gracza.
  - `pos`: Pozycja początkowa gracza na ekranie.
- `draw\_hp(screen: pygame.Surface) -> None`: Rysuje pasek stanu zdrowia gracza na ekranie.
  - `screen`: Powierzchnia pygame, na której rysowany jest pasek stanu zdrowia.

## 3. Monster

Klasa `Monster` reprezentuje potwora w grze.

Metody:

- `\_\_init\_\_(image\_path: str, pos: Tuple[int, int])`: Inicjalizuje obiekt Monster.
  - `image\_path`: Ścieżka do pliku obrazu potwora.
  - `pos`: Pozycja początkowa potwora na ekranie.
- `move\_towards(target\_rect: pygame.Rect, step: int = 3) -> None`: Przesuwa potwora w kierunku podanego prostokąta.
  - `target\_rect`: Prostokąt, do którego potwór ma się przesunąć.
  - `step`: Szybkość przesunięcia potwora.

## 4. Players

Klasa `Players` zarządza grupą graczy w grze.

Metody:

- `\_\_init\_\_():` Inicjalizuje obiekt Players i dodaje obiekty graczy do niej.
- `choose\_target() -> Optional[Player]`: Wybiera gracza docelowego dla potwora do ataku. Zwraca None, jeśli nie ma dostępnych celów.

## 5. LevelDrawer

Klasa `LevelDrawer` zarządza rysowaniem informacji o poziomie na ekranie.

Metody:

- `\_\_init\_\_(screen: pygame.Surface, font\_size: int = 36, color: Tuple[int, int, int] = (255, 0, 0))`: Inicjalizuje obiekt LevelDrawer.
  - `screen`: Powierzchnia pygame, na której rysowana jest informacja o poziomie.
  - `font\_size`: Rozmiar czcionki tekstu informacji o poziomie.
  - `color`: Kolor tekstu informacji o poziomie.
- `draw(level: int) -> None`: Rysuje informacje o poziomie na ekranie.
  - `level`: Numer aktualnego poziomu.

## 6. GameStateLogger

Klasa `GameStateLogger` zarządza logowaniem stanu gry.

Metody:

- `\_\_init\_\_():` Inicjalizuje obiekt GameStateLogger.
- `log\_initial\_state(level: int, player2\_hp: int, monster\_hp: int) -> None`: Loguje początkowy stan gry.
  - `level`: Numer aktualnego poziomu.
  - `player2\_hp`: Początkowa ilość punktów zdrowia gracza 2.
  - `monster\_hp`: Początkowa ilość punktów zdrowia potwora.
- `log\_round(level: int, player\_hp\_start: int, monster\_hp\_start: int, player2\_hp\_end: int, player2\_bonus\_hp: int, monster\_hp\_end: int, damage\_dealt: int) -> None`: Loguje rundę gry.
  - `level`: Numer aktualnego poziomu.
  - `player\_hp\_start`: Początkowa ilość punktów zdrowia gracza.
  - `monster\_hp\_start`: Początkowa ilość punktów zdrowia potwora.
  - `player2\_hp\_end`: Końcowa ilość punktów zdrowia gracza 2.
  - `player2\_bonus\_hp`: Dodatkowe punkty