

# Dokumentacja projektu: „Text Adventure Game”

## Używane biblioteki:

**json**

**random**

## Cel projektu:

Napisanie gry tekstowej, przyjmującej polecenia od użytkownika w postaci poleceń. Gra powinna być wczytywana z pliku i powinna mieć możliwość zapisu.

## Założenia projektu:

- Zarówno mapa jak i gracz są wczytywane z pliku o formacie json.
- Mapa może być dowolnie duża, ale powinna być kwadratowa.
- Gracz wygrywa po pokonaniu wszystkich przeciwników na mapie.
- Gracz nie może wyjść poza granice mapy.

Po uruchomieniu pliku 'game.py' w terminalu pojawi się linijka z zapytaniem o wczytanie gry. Po wpisaniu 'no' gra zainicjuje się z pliku 'init.json', po wpisaniu 'yes' oraz nazwy pliku gra zainicjuje się z danego pliku. Gra wyświetli listę komend możliwych do użycia. Przy używaniu komend 'move' oraz 'use gem' należy podać kierunek ('east', 'west', 'north', 'south'). Przy używaniu komendy 'fight' należy podać nazwę przeciwnika. Reszta komend nie wymaga podawania dodatkowych informacji.

Plik 'classes.py' zawiera klasy **Player**, **Enemy**, **Gem**, **Location** i **Game**, oraz funkcje do odczytywania danych z pliku i do zapisywania danych.

Klasa **Player** zawiera atrybuty: name, current\_location, locations, power, health, equipment. Atrybut name to imię gracza. Atrybut current\_location jest zapisany w pliku inicjującym jako krotka zawierająca obecne położenie gracza. Atrybut locations jest zapisany w pliku inicjującym jako lista zawierająca obiekty klasy **Location**. Atrybut power jest numerem (int) i reprezentuje siłę gracza. Atrybut health jest numerem (int) i reprezentuje życie gracza. Atrybut equipment jest zapisany w pliku inicjującym jako lista zawierająca obiekty klasy **Gem**.

Klasa **Enemy** zawiera atrybuty: name, health, power. Atrybut name reprezentuje nazwę przeciwnika. Atrybut health jest numerem (int) i reprezentuje życie przeciwnika. Atrybut power jest numerem (int) i reprezentuje siłę przeciwnika.

Klasa **Gem** zawiera atrybuty: name, color. Atrybut name reprezentuje nazwę kamienia. Atrybut color reprezentuje kolor kamienia.

Klasa **Location** zawiera atrybuty: name, description, barrier, barrier\_color, enemies, items, x, y. Name reprezentuje nazwę lokacji. Description reprezentuje opis lokacji. Barrier

reprezentuje obecny stan bariery, 1 oznacza, że bariera istnieje, 0 oznacza, że bariery nie ma. Enemies jest listą obiektów klasy **Enemy** i reprezentuje przeciwników w danej lokacji. Items jest listą obiektów klasy **Gem** i reprezentuje kamienie w danej lokacji. X, y są numerami (int) reprezentującymi, w którym miejscu na mapie leży dana lokacja.

Klasa **Game** odpowiada za rozgrywkę. Metoda `init_player` odpowiada za wczytywanie pliku, z którego chcemy wczytać grę. Metoda `play` odpowiada za rozgrywanie gry.

Klasa **FileHandler** odpowiada za obsługę plików. Metoda `read_from_json` odpowiada za wczytywanie danych z pliku. Metoda `save_to_json` odpowiada za zapis danych z gry do pliku.

**Problemy:**

- Mapa musi być kwadratowa, nie może być w innym kształcie