**Roguelike**

# Opis gry:

W Roguelike gracz wciela się w rolę poszukiwacza przygód uwięzionego w podziemiach pełnych wrogo nastawionych stworzeń. Na domiar złego, postać cierpi na nieuleczalną chorobę – każdy ruch powoduje stopniową utratę zdrowia. Jedynym lekarstwem są występujące w podziemiach źródła cudownej wody, której łyk nadaje chwilową ulgę. Zasoby źródeł nie są jednak nieograniczone. Jedynym ratunkiem jest zejście w dół podziemi w celu przeżycia i nadzieja, że gdzieś tam znajduje się wyjście.

Program został napisany w języku Python (wersja 3.6.5) przy pomocy biblioteki graficznej TKinter.

# Lista klas:

* **Location** – Prosta klasa przechowująca koordynaty x, y, z.
* **Tile** – Przechowuje informacje o polu, takie jak jego lokacja, czy jest interaktywne, czy zezwala na pojawianie się na nim potworów, itp.
* **Floor**
* **Wall**
* **Upstairs**
* **Downstairs**
* **Fountain**
* **Level** – Zawiera dwuwymiarową listę pól oraz metody służące do manipulacji nimi.
* **Entity** – Klasa będąca bazą dla różnych istot. Określa ich stan zdrowia, siłę ataku, itp.
* **Player**
* **Skeleton** - Groźny potwór na niskich poziomach z uwagi na umiejętność wysysania życia.
* **HUD** – Służy do wyświetlania na ekranie takich informacji o graczu jak jego zdrowie i doświadczenie.
* **Game** – Zarządza wyżej wymienionymi klasami.

# Diagram UML:

# Wskazówki dla użytkowników:

Grę włącza się uruchamiając plik game.py.

Zarówno pola jak i postacie reprezentowane są za pomocą kodu ASCII. Większość obrazu gry zajmuje etap podziemia na którym znajduje się gracz. Dolna część należy do HUD, gdzie wypisywane są informacje o obecnym stanie gracza, takie jak jego zdrowie, poziom oraz doświadczenie.

Użytkownik steruje postacią przy pomocy klawiatury. Poniżej lista dostępnych poleceń:

w – ruch w górę

s – ruch w dół

a – ruch w lewo

d – ruch w prawo

e – użycie przedmiotu / potwora / pola znajdującego się pod graczem

x – atakowanie

z – pominięcie tury

ESCAPE – wyjście z gry

Niektóre z ustawień gry, takie jak rozdzielczość, widoczność gracza, zabieranie co turę punktów życia i jego liczba, i tym podobne, można dostosować zmieniając wartości odpowiednich pól znajdujących się w pliku settings.py.

# Wskazówki dla programistów:

## Rozwój:

Grę można rozwinąć o nowe pola dziedziczące po *Tile* o własnych, unikalnych metodach interakcji z graczem. Inicjację nowych pól należy umieścić w odpowiednich metodach klasy *Level*, które zostaną wywołane w jej konstruktorze.

Oprócz pól można również dodać nowe potwory, które będą dziedziczyć po klasie *Entity.* Po ich dodaniu radzę zdefiniować funkcje pomocnicze w klasie *Game*, które będą określać liczbę wystąpień danego rodzaju potworów w zależności od poziomu / rodzaju podziemia.

Innym możliwym dodatkiem jest rozszerzenie rodzajów generowanych podziemi. Na tę chwilę jest tylko jeden, dość sztuczny, typ. Nic nie stoi na przeszkodzie aby dodać bardziej naturalne jaskinie lub zaawansowane struktury typu obóz, miasto.

Oczywiście, można również dodać przedmioty, czary, czy rozszerzyć system walki.

## Reusing:

Klasa *Location* może zostać wykorzystana nie tylko przy tworzeniu podobnych gier, ale również jako składowa wektorów, część figury, itp.

Klasy *Entity* oraz *Tile*, choć znaczniej mniej ogólne niż *Location*, mogą zostać użyte przy tworzeniu dwuwymiarowych gier o zbliżonej tematyce.

Kondraciuk sebastian 298451