

pydash

Damian Misiura **2015**

Projekt końcowy kursu Programowanie w języku Python

Zawartość:

1. Cel projektu
2. Funkcjonalność projektu
3. Planowane funkcjonalności
4. Architektura rozwiązania

1. Cel projektu

Projekt ma na celu stworzenie gry typu boulder dash wykorzystując język Python oraz biblioteki utworzone z użyciem tego też języka. Jest to temat nr 5 działy GRY:

Boulder dash [1 osoba] - gra znana jako boulder dash lub supaplex.

Rezultat pracy ma w dużym stopniu przypominać kultową grę Supaplex (z wyjątkiem tekstur, przeciwników i paru power-upów i prawdopodobnie dźwięków).

Grę supaplex można znaleźć tutaj:

(<http://www.dosgamesarchive.com/download/supaplex/>)

2. Funkcjonalność projektu

Projekt ma umożliwiać następujące funkcje:

- gra na czterech dostępnych mapach
- menu główne
- zjedzenie całej podłogi na dostępnych mapach (zamienia się na czarne tło)
- łatwość zmiany głównej postaci (plik z nową musi mieć taki sam format jak postać obecnie, również dokładnie taki sam rozmiar sprite'ów)

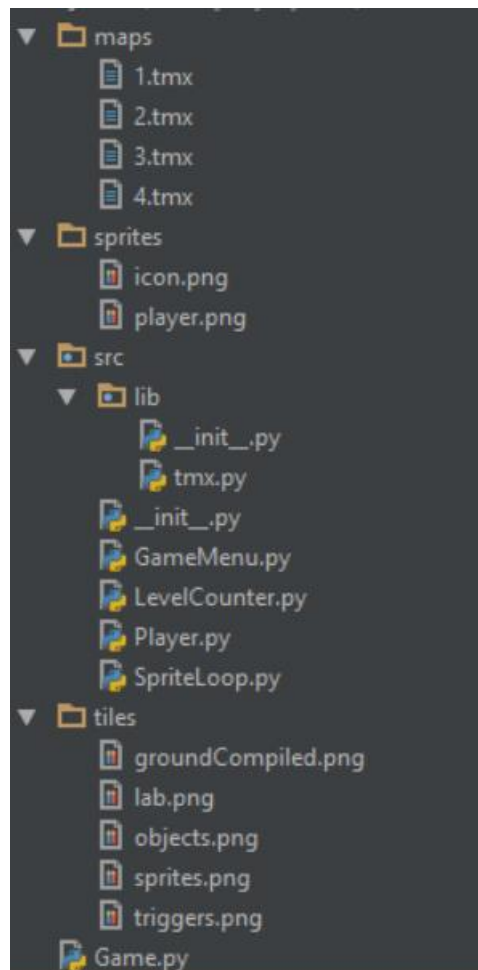
3. Planowane funkcjonalności

- głązy działające z prostymi zasadami fizyki
- przedmioty, których kolekcjonowanie pozwala na zakończenie etapu
- animacje (wraz z tekstem) po zakończonym etapie

- dźwięki w grze
- zmiana opcji gry (rozdzielczość ekranu, on/off dźwięku i muzyki)
- dwa poziomy trudności
- możliwe do modyfikacji wszystkie sprite'y dostępne w grze

4. Architektura rozwiązania

W chwili obecnej rozwiązanie prezentuje się następująco:



Rysunek 1 - Projekt w stanie obecnym (31 maja)

Maps – katalog zawiera wszystkie mapy użyte w projekcie, każda mapa ma format .tmx (format używany przez program Tiled)

Sprites – użyte sprite'y w grze, które posiadają animacje, w obecnej wersji jest to ikona gry oraz postać

Src/lib – użyte biblioteki w grze, obecnie używana jest biblioteka tmx

Tmx.py – konwertuje plik .tmx na obiekt języka Python, używany do manipulacji mapą

Src/GameMenu.py – plik odpowiedzialny za obsługę menu w grze, jego logikę oraz wygląd

Src/LevelCounter.py – zawiera logikę liczącą obecny poziom w grze

Src/Player.py – plik odpowiedzialny za całą logikę, wygląd i zachowanie gracza

Src/SpriteLoop.py – plik odpowiedzialny za animację sprite'ów (niewykorzystany teraz)

Tiles – wszystkie „kratki” wykorzystane w grze (oprócz sprites.png, zostaną dodane w późniejszych wersjach)

Game.py – zawiera logikę gry, zachowanie ekranu podczas wydarzeń w grze

Wykorzystane biblioteki:

Pygame 1.9.0, tmx (dostępna w projekcie)

Jak uruchomić grę:

Uruchomić plik Game.py, on zawiera logikę uruchamiania gry. Zmiana opcji wyświetlania zostanie dokończona przy finalnym projekcie