

pydash

Damian Misiura **2015**

Projekt końcowy kursu Programowanie w języku Python

Zawartość:

1. Cel projektu
2. Funkcjonalność projektu
3. 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 działu 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).

Grę supaplex można znaleźć tutaj:

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

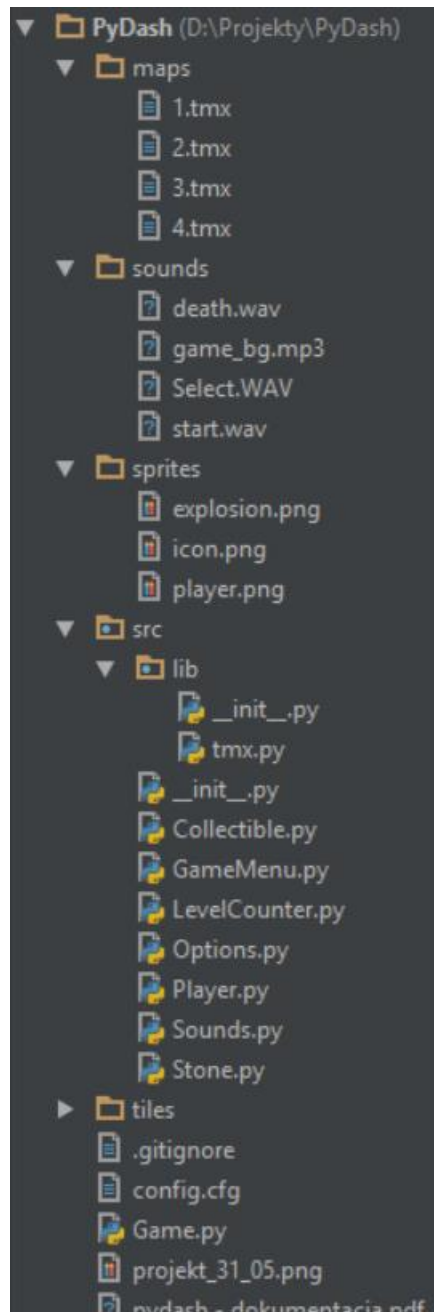
2. Funkcjonalność projektu

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)
- zmiana opcji dźwięku oraz opcji dźwiękowych
- możliwość tworzenia łamigłówek korzystając z fizyki obiektów (collectible oraz stone)

3. Architektura rozwiązania

W chwili obecnej rozwiązanie prezentuje się następująco:



Rysunek 1 - Projekt w stanie obecnym (7 czerwca)

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

Sounds – katalog zawierający dźwięki użyte w grze

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/Collectible.py – plik zawierający całą logikę związaną z obiektem który postać ma za zadanie zbierać

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/Options.py – zawiera logikę odpowiedzialną za okno opcji oraz ich wczytywanie/zapisywanie jako preferencje użytkownika

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

Src/Sounds.py – plik odpowiedzialny za odtwarzanie dźwięków w grze

Src/Stone.py – plik odpowiedzialny za całą logikę związaną z obiektem kamienia (przeszkody w grze)

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 1.9.2, tmx (dostępna w projekcie, biblioteka zewnętrzna), **Tk**

Jak uruchomić grę:

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

Sterowanie w grze:

Poruszanie się postacią – strzałki

Powrót do menu – Esc

Nawigacja w menu:

Klawisze strzałki oraz enter/spacja – nawigowanie w menu i wybór elementu menu

Mysz – opcjonalne sterowanie w menu, jedyne w menu opcji

Cel Gry:

Gracz ma za zadanie zebrać wszystkie obiekty które można kolekcjonować (tak jak w klasycznym supaplexie). Jeśli to nastąpi, gracz może się udać do wyjścia, w przeciwnym wypadku etap nie skończy się dopóki gracz nie zbierze ich wszystkich.



Rysunek 2 - przedmiot który gracz musi zebrać

Zasady gry:

- etap kończy się gdy gracz zbierze wszystkie przedmioty (rysunek 2) oraz dojdzie do wyjścia (nie wszystkie są prawdziwe)
- gracz ginie jeśli zostanie uderzony głazem lub przedmiotem kolekcjonowalnym podczas gdy nie jest zwrócony w górę
- gracz ma nielimitowaną ilość żyć, ale wraca na początek etapu po śmierci
- gra ma 4 etapy, jeśli gracz zakończy 4ty etap, wraca do 1szego etapu