

Przedmiotem projektu jest implementacja gry Sokoban. Gra jest rozgrywana na planszy z kwadratów, gdzie każdy kwadrat jest podłogą lub ścianą. Niektóre kwadraty na podłodze zawierają beczki, a niektóre są oznaczone jako miejsca do przechowywania. Gracz może przesunąć beczkę, podchodząc do niej i przesuwając ją na sąsiedni kwadrat. Beczek nie można ciągnąć, nie można ich też pchać na kwadraty ze ścianami lub innymi beczkami. Poziom zostaje ukończony, gdy wszystkie beczki znajdą się w miejscach przechowywania.

Projekt wyposażony jest w GUI oparte na bibliotece PyGame. Gra uruchamiana jest poprzez wykonanie pliku głównego – „sokoban.py”. Do gry zaimplementowane jest kilka rodzajów graficznego menu, do których obsługi wymagane jest użycie kursora. Sterowanie postacią oparte jest o wykorzystanie klawiszy WSAD. W przypadku chęci restartu aktualnego poziomu możliwe jest przywołanie menu restartu w dowolnym momencie rozgrywki przy użyciu klawisza ESC.

W skład projektu wchodzi następujące klasy:

Player – klasa, której obiektem jest główna postać, implementuje mechanikę animacji oraz pobierania danych wejściowych od gracza.

Map – obiektami klasy są poziomy gry Sokoban, wykorzystuje ona wszystkie pozostałe klasy oraz tekstową reprezentację poziomu do ustawienia planszy gry. Zawarte są w niej metody inicjalizujące mechanikę ruchu oraz interakcje między poszczególnymi obiektami poziomu.

Klasy reprezentujące planszę gry:

Barrel – obiektami są skrzynie/beczki.

Tile – obiektami są bloki ścian, które ograniczają możliwość wyjścia głównej postaci poza obszar poziomu.

Spot – obiektami są miejsca docelowego umieszczenia beczek/skrzyń w danym poziomie.

Klasy reprezentujące menu:

Menu – główna klasa menu, dziedziczą po niej pozostałe klasy

InitialMenu – klasa odpowiadająca za generowanie i wyświetlanie menu rozpoczynającego rozgrywkę.

RestartMenu – klasa odpowiadająca za generowanie i wyświetlanie menu restartu. Menu jest inicjowane po wciśnięciu klawisza ESC w trakcie rozgrywki.

MenuNextLevel – klasa odpowiadająca za generowanie i wyświetlanie menu przejścia do kolejnego poziomu. Menu jest inicjowane po ukończeniu każdego levelu.

Projekt zorganizowany jest w sześciu plikach:

sokoban.py – plik zawierający główną pętlę gry.

elements.py – plik przechowuje klasy: Barrel, Tile, Spot.

player.py – plik przechowuje klasę Player.

menu.py – plik przechowuję klasy związane z wyświetlaniem menu w grze.

map.py – plik przechowuję klasę Map oraz mechanikę wczytywanie tekstowych reprezentacji poziomów z pliku o formacie „.json”.

levels.json – plik przechowuje tekstowe reprezentacje poziomów gry.

Funkcjonalnością, której nie udało mi się zaimplementować, jest system wyszukiwanie deadlock’ów, czyli sytuacji, kiedy poziom nie jest możliwy do ukończenia. Jako substytut dodałem możliwość ręcznego restartu poziomu w dowolnym momencie rozgrywki. Sposób przechowywania poziomów w projekcie umożliwia ich bardzo łatwą edycję oraz rozwijanie gry o dodawanie kolejnych. Elastyczna budowa klas i ich modułów pozwala na szybkie wprowadzanie zmian na poziomie zarówno layoutu, jak i mechanik rozgrywki.