Andrzej Pultyn, nr albumu 325213

Wściekłe Ptaki – projekt PIPR zima 2022/2023

- Opis projektu

Celem projektu było stworzenie gry przypominającej „Angry Birds”. Jej celem jest zestrzelenie wszystkich przeciwników (zielonych świnek) na mapie, zanim wyczerpią się podejścia.

W projekcie jest 7 poziomów. Po wyborze jednego z nich w konsoli, uruchamia się okno gry. Gracz ustawia kąt strzału i jego siłę. Następnie następuje animacja lotu. W przypadku, gdy przeciwnik zostanie trafiony, to znika z gry, a gracz zachowuje szansę. Jeśli zaś ptak trafi w krawędź okna, podłoże albo drewnianą zaporę, to gracz traci jedno podejście.

Po skończeniu poziomu (nie ważne z jakim skutkiem) można wybrać inny. Gra zapamiętuje, które poziomy zostały już zrobione, są one oznaczone na liście. Ponadto na górze tekstu w konsoli pokazany jest procent pokonanej gry, czyli ile etapów na 7 zostało już ukończonych.

Gra składa się z plików:

- main.py - uruchamia cały projekt, pozwala uruchomić dany poziom, oraz zawiera kod do pokazywania odpowiednich danych w menu w konsoli
- game_engine.py – zawiera cały aspekt graficzny gry, oraz fizykę postaci (lot, kolizje itp.)
- side_programms.py – plik zawierający oddzielne funkcje i klasę GameObject. Do funkcji należą:
  - change_angle – obliczająca prawidłowy kąt po naciśnięciu odpowiedniego klawisza
  - get_list_objects – zwracająca listy obiektów w danym poziomie
- test_side_programms.py – zawierający testy do funkcji change_angle

Ostatnim elementem projektu jest katalog graphics, z którego game_engine pobiera odpowiednie grafiki.

- Opis rozwiązania

Sama gra opiera się na bibliotece pygame. Biblioteka umożliwia tworzenie gier w czasie rzeczywistym. Program importuje grafiki z folderu Graphics, oraz przydziela im prostokąt w oparciu o jego wymiary. Kolizje między obiektami, albo z krawędzią mapy opierają się właśnie na takich prostokątach, na ich wymiarach oraz położeniu. Umieszcza się dowolny jego punkt w dowolnym miejscu na ekranie, a następnie na niego nakłada się grafikę.

Poziomy tworzy się poprzez stworzenie listy przeciwników oraz przeszkód w funkcji get_list_objects z pliku side_programms.py. Każdy taki element jest obiektem klasy GameObject. Każdy z nich zawiera własny prostokąt, oraz pozycję. W przypadku trafienia przeciwnika, zostaje on usunięty z listy przeciwników w grze. Przy użyciu takiego rozwiązania łatwo jest przedstawić wszystko graficznie – wystarczy przeiterować po obu listach, dla każdego obiektu ustawić ich prostokąt na odpowiednią pozycję, oraz nadać odpowiednią grafikę.

Kąt oraz siłę strzału gracz ustawia przy użyciu klawiatury. Kąt zmienia się klawiszami W (w górę), S (w dół), A (w lewo), D (w prawo), na zasadzie, że kąt zmienia się o 5 stopni w kierunku

wyznaczonym przez klawisz. Gdy kąt strzału jest równy docelowemu, lub przesuniętemu o 180 stopni, to nic się nie zmieni. Obliczaniem odpowiedniego kąta po przyciśnięciu przycisku zajmuje się funkcja change_angle, również z pliku side_programms.py. Następnie po wciśnięciu spacji następuje strzał.

Symulacja lotu opiera się na rzucie ukośnym znanym z fizyki. Przez ustawienie kąta i siły strzału, ustawiane są prędkości początkowe w dwóch płaszczyznach. Wartości prędkości są obliczane z funkcji trygonometrycznych z biblioteki math. Następnie z każdą kolejną klatką, prędkość w osi pionowej maleje. Z uwagi na to, że pygame przy tworzeniu okna, jako punkt (0, 0) traktuje lewy góry róg, a nie dolny, to pozycja gracza zmienia się o wartość prędkości w danej chwili z przeciwnym znakiem.

Wybór poziomów następuje w konsoli, gdzie wypisywane jest polecenie wyboru poziomu, lista poziomów z zaznaczonymi już ukończonymi, oraz procent ukończenia gry. Wyboru dokonuje się poprzez wprowadzenie numeru odpowiedniego poziomu, lub 0, jeśli chce się zakończyć działanie programu.

- Instrukcja użytkowania
    - Grę uruchamiamy poprzez plik main.py
    - Poziom, który chcemy zagrać, wprowadzamy z klawiatury
    - Po wprowadzeniu 0, program zakończy działanie
    - Kąt strzału zmieniamy klawiszami:
        - W – w górę
        - S – w dół
        - A – w lewo
        - D – w prawo
    - Siłę strzału zmieniamy klawiszami:
        - Strzałka do góry – więcej siły
        - Strzałka do doły – mniej siły
    - Potwierdzenie kąta i siły oraz oddanie strzału następuje po wciśnięciu spacji
- Refleksje dotyczące projektu

Biblioteka pygame okazała się bardzo intuicyjnym i prostym w obsłudze narzędziem do programowania gier. Przede wszystkim zaskoczyło mnie proste identyfikowanie kolizji. Zmotywowało mnie to do zaimplementowania drewnianych zapór. Jednak z drugiej strony nie miałem pomysłu na testy jednostkowe samej gry. Jak przetestować fizykę, czy kolizje. Starałem się wydzielić części kodu jako oddzielną funkcję i ją przetestować. Udało mi się to z change_angle, ale wiem, że tych testów powinno być dużo więcej.