

ANGRY NON-BIRDS

PROJEKT

Prosta gra podobna do Angry Birds. Została napisana w Pythonie z wykorzystaniem dwóch bibliotek: Pygame (interfejs graficzny) i Pymunk (symulacja fizyki).

Celem gry jest dobranie odpowiedniej mocy i kąta wystrzelenia pocisku tak, aby trafić cele. Składa się z trzech poziomów o narastającym poziomie trudności i skończonej liczbie prób.

PROGRAM

Program składa się z wielu klas:

Klasa Game

Odpowiada za przeprowadzenie całej rozgrywki, obsługując wydarzenia.

Klasa GameStage

Podstawowa klasa obsługująca etap rozgrywki.

Posiada dwie metody draw() i action() wywoływane z w pętli rozgrywki w klasie Game.

Klasa Intro

Klasa dziedzicząca po klasie GameStage. Obsługuje obraz początkowy gry.

Klasa Credits

Klasa dziedzicząca po klasie GameStage. Obsługuje napisy końcowe.

Klasa Level

Klasa dziedzicząca po klasie GameStage. Obsługuje poziom rozgrywki.

Tworzy przestrzeń do symulacji przy pomocy Pymunk. Obsługuje wystrzelenie pocisku, m.in. przy pomocy funkcji pomocniczych obliczających siłę i kąt nachylenia wystrzału. Obsługuje zderzenia obiektów w symulacji i ich usuwanie.

Klasy Level1, Level2, Level3

Dziedziczą po klasie Level. Odpowiadają za przygotowanie poszczególnych poziomów (granic, przeszkód, celów).

Klasa Ball

Reprezentuje pocisk wystrzeliany przez gracza.

Klasa Boundary

Reprezentuje statyczne obiekty granic w symulacji.

Klasa Structure

Reprezentuje dynamiczne obiekty przeszkód.

Klasa Target

Reprezentuje cele.

INSTRUKCJA UŻYTKOWNIKA

Do uruchomienia programu niezbędne jest pobranie bibliotek Pygame i Pymunk. Aby to uczynić należy wpisać komendy:

```
pip install pygame
```

```
pip install pymunk
```

Następnie należy sklonować repozytorium:

```
git clone https://gitlab-stud.elka.pw.edu.pl/jkedzier/angry-non-birds.git
```

Aby uruchomić program należy wywołać skrypt main.py znajdujący się w głównym katalogu repozytorium.

ROZGRYWKA

Użytkownik za pomocą kliknięcia przycisku myszki rozpoczyna procedurę dobierania mocy i kąta wystrzelenia pocisku. Czarna linia reprezentuje kierunek i siłę naciągu.

Ponowne kliknięcie wystrzeliwuje kulkę. Cele bezpośrednio trafione od razu zostają usunięte. Trafione przeszkody nie zostają usunięte, ale się przewracają. Mogą w ten sposób uderzyć w cel. Jeżeli podczas zderzenia przeszkody z celem wytracona energia kinetyczna będzie odpowiednio duża, to cel zostanie usunięty. Analogiczna zasada ma miejsce w momencie zderzenia celu ze statycznym obiektem granicy.

Kolejne kliknięcie usuwa pocisk oraz obiekty, które wyleciały poza granice symulacji.

REFLEKSJE

Obsługa symulacji fizycznej okazała się najtrudniejsza. Szczególnie zderzenia obiektów były problematyczne. Dlatego na tą część projektu musiałem poświęcić najwięcej czasu. Na szczęście wykorzystanie biblioteki Pymunk, ułatwiło wykonanie tego zadania.

Do poprawy zostało wyświetlanie bieżącego stanu symulacji, które obecnie odbywa się przy pomocy funkcji debug_draw() z modułu pymunk.pygame_util. Bardziej pożądanym byłoby wykorzystanie klas Sprite i Group z modułu pygame.sprite.