

Star Fighters - raport końcowy

Projekt końcowy z programowania obiektowego

Opis gry

Gra jest przeznaczona dla dwóch osób. Gracze wcielają się w pilotów statków kosmicznych. Celem gracza jest zniszczenie statku przeciwnika. Statki wyposażone są w działa, umożliwiające atak zabójczymi kulami. Rywalizacja jest dodatkowo wzbogacona o latające asteroidy, stanowiące zagrożenie dla obu graczy. Wygodne menu daje możliwość sprawdzenia sterowania oraz przygotowania się do gry. Gracze mogą także dokonać personalizacji kształtu oraz koloru ich statków.

Użyte narzędzia i struktura projektu

Gra jest napisana w języku Python, przy użyciu biblioteki Pygame. Do napisania interfejsu użytkownika skorzystaliśmy z Pygame GUI. Biblioteka ta została napisana w sposób obiektowy, zatem dobrze integruje się z naszym projektem. W celu zachowania przejrzystej struktury projektu, został on podzielony na osobne fragmenty:

- *assets*: folder zawierający zasoby użyte w programie, takie jak obrazy i dźwięk.
- *src*: główna część projektu. Zawiera wszystkie pliki składające się na kod źródłowy gry.
- *themes/theme.json*: plik zmieniający wygląd elementów z Pygame GUI.
- *requirements.txt*: plik zawierający dane na temat użytych bibliotek.

Kod źródłowy podzieliliśmy na osobne pliki zależnie od funkcji, jaką dana część pełni w naszym programie. Są to następujące pliki:

- *main.py*: uruchamia grę.
- *Game.py*: zawiera pętlę gry, która wywołuje metody odpowiadające za przetwarzanie akcji graczy, aktualizację stanu gry oraz rysowanie jej na ekranie.
- *GameObjects.py*: w tym pliku znajduje się implementacja klas reprezentujących obiekty wykorzystywane w grze. W szczególności są to klasy reprezentujące statki, asteroidy, kule oraz wybuchy.
- *Screens.py*: klasy zawierające implementacje ekranów: gry, wyboru statku, menu. W dużej części jest to obsługa interfejsu graficznego.
- *ImagePanelButton.py*: implementacja przycisku do wyboru statku.
- *UserEvent.py*: klasa reprezentująca zdarzenia występujące w grze.

Uruchomienie

Poniższa instrukcja zakłada użycie wirtualnego środowiska, które pozwala uniknąć konfliktu zależności z innymi projektami w Pythonie. Może być konieczne zainstalowanie odpowiedniego pakietu. Na Ubuntu można zainstalować go poleceniem `apt install python3.10-venv`.

Projekt został przetestowany na nowej instalacji Ubuntu 22.04 z Pythonem w wersji 3.10.6. Gra nie działa prawidłowo jeśli zostanie uruchomiona poprzez *Windows Subsystem for Linux*.

1. Utworzenie wirtualnego środowiska `python3 -m venv venv`
2. Aktywowanie wirtualnego środowiska (konieczne za każdym razem gdy uruchamiamy projekt) `source venv/bin/activate`
3. Instalacja zależności `pip install -r requirements.txt`
4. Uruchomienie gry `python3 src/main.py`

Aby dezaktywować wirtualne środowisko należy użyć polecenia `deactivate`.

Podsumowanie

Projekt pozwolił nam się zmierzyć nie tylko z programowaniem obiektowym, ale również nauczył nas, że nie zawsze realizacja pomysłów jest tak łatwa, jak się wydaje. Największą trudnością okazała się sprawna współpraca w pisaniu kodu i zgranie naszych pomysłów w spójną całość. Zdobyte doświadczenie niewątpliwie znacznie pomoże nam w pracy nad przyszłymi projektami. Końcowa gra nieco różni się od naszego pierwotnego pomysłu. Przykładowo, zrezygnowaliśmy z implementacji klasy reprezentującej gracza, która okazała się niepotrzebna. Dodaliśmy natomiast rozbudowany interfejs użytkownika, w którym znalazło się menu główne, instrukcja dotycząca sterowania oraz ekran wyboru statków graczy. Był to element programu, który okazał się bardziej wymagający w implementacji niż wstępnie zakładaliśmy. Mimo to główne założenia projektu zostały zachowane i nie odeszliśmy znacząco od pierwotnego pomysłu. Obiektowy styl wykonania projektu pozwolił nam zaimplementować techniki poznane podczas semestru, takie jak hermetyzacja, dziedziczenie czy abstrakcja.

Źródła

Podczas tworzenia projektu korzystaliśmy z następujących źródeł:

- dokumentacja biblioteki Pygame <https://www.pygame.org/docs/>
- dokumentacja biblioteki Pygame GUI oraz przykłady jej użycia:
 - <https://pygame-gui.readthedocs.io/en/latest/index.html>
 - https://github.com/MyreMylar/pygame_gui_examples
- poradniki do biblioteki Pygame:
 - <https://pygame.readthedocs.io/en/latest/index.html>
 - https://dr0id.bitbucket.io/legacy/pygame_tutorials.html
 - https://www.youtube.com/watch?v=b_DkQrJxpck
- wprowadzenie do obiektowego projektowania gier w Pygame <http://pygametutorials.wikidot.com/tutorials-basic>
- <https://stackoverflow.com/questions/4183208/how-do-i-rotate-an-image-around-its-center-using-pygame>
- czcionka używana w menu <https://www.fontspace.com/public-pixel-font-f72305>

Dostęp do wszystkich stron 12.06.2023. Pozostałe materiały użyte w grze (modele statków, asteroidy, tła, logo, muzyka) stworzyliśmy samodzielnie.

Załączniki

- Wstępny diagram klas
- Końcowy diagram klas