Politechnika Śląska

Wydział Matematyk Stosowanej

Kierunek Informatyka

Gliwice, 26.01.2022

Programowanie I

**projekt zaliczeniowy**

**"*Spaceship*"**

**Jan Rapacz 2/4 lab**

**1. Opis projektu.**

*Prosta gra 2D stworzona za pomocą biblioteki SFML. Celem gry jest zdobycie jak największej ilości punktów poprzez unikanie i niszczenie laserami nadlatujących meteorytów.*

**2. Wymagania**

1. Kierowanie statkiem odbywa się na pomocą klawiszy „A” i „D”

2. Zdobywamy punkty kiedy niszczymy meteoryty

3. Kontakt z meteorytem kończy grę.

4. Po „śmierci” możemy nacisnąć Enter aby spróbować jeszcze raz

5. Za pomocą Spacji możemy wystrzeliwać lasery

**3. Przebieg realizacji**

Dla znacznego ułatwienia projektu zdecydowałem się napisać go za pomocą programowania obiektowego i biblioteki SFML. Program składa się z 7 klas w tym jedna to Singleton (ResourceLoader.h) który wczytuje wszystkie potrzebne tekstury na początku działania programu w celu optymalizacji. Początkowo miałem trochę problemów z OOP w c++, głównie za sprawą dosyć dziwnej dla mnie mechaniki dołączania plików nagłówkowych. Udało mi się jednak skonsultować to ze znajomymi z roku którzy pomogli mi w rozwiązaniu kilku bugów. Sama logika gry nie była zbyt skomplikowana. Każdy ruszający się obiekt w grze dziedziczy po klasie abstrakcyjnej GameEntity.h która zawiera takie pola jak szerokość, wysokość , pozycja x i y czy prędkość. Istotnym elementem klasy jest też metoda „collideWith” która pozwala zaimplementować zachowanie obiektu gdy ten zderza się z innym. Klasa ta znacznie zwiększyła czytelność kodu a także zmniejszyła nadmiarowość kodu.

Do gry udało mi się dodać również tekstury aby była bardziej atrakcyjna dla oka.

Link to tekstur: <https://axassets.itch.io/spaceship-simple-assets>

**4. Instrukcja użytkownika**

Sterowanie za pomocą „A” i „D”

Strzelanie laserami za pomocą Spacji

**5. Podsumowanie i wnioski.**

Szczęśliwie udało mi się zrealizować wszystkie założenia projektu. Jedną z większych trudności sprawiło mi zdebugowanie błędów spowodowanych przez błędne dodawanie plików nagłówkowych - importowałem jeden plik nagłówkowy do drugiego, a ten drugi importowałem także do pierwszego co powodowało pętle nieskończonych importów. Zamiast importowania wystarczyło po prostu zadeklarować, że taka klasa istnieje bez konieczności dołączania całego pliku nagłówkowego. Poza tym składania OOP w C++ różni się nieco od tej znanej mi np. z języka Java więc musiałem nadrobić kilka zaległości.

Pisanie projektu pozwoliło mi nieco lepiej zrozumieć niektóre mechaniki języka C++, między innymi dynamiczne zarządzanie pamięcią gdyż często korzystałem ze wskaźników.

Dalszym kierunkiem rozwoju mogłoby być np. dodanie listy najlepszych wyników, dodanie lepszego menu i poziomów trudności.