

Wydział Elektrotechniki, Automatyki i Informatyki Politechniki  
Świętokrzyskiej

# Space Invaders

Grafika Komputerowa

Kinga Nowakowska  
Maciej Umański

## **I. Temat projektu.**

Tematem projektu jest gra 2D Space Invaders. Gra polega na niszczeniu kolejnych fal kosmitów przy pomocy strzelania i zdobywaniu jak największej liczby punktów. Do stworzenia gry wykorzystano bibliotekę Raylib i język programowania c++.

## **II. Wykorzystane narzędzia.**

Język programowania: C++

Środowisko programistyczne IDE: CLion

System operacyjny: Windows 10 Home, MacOS

Biblioteka graficzna: Raylib

Biblioteka do generowania dokumentacji: Doxygen

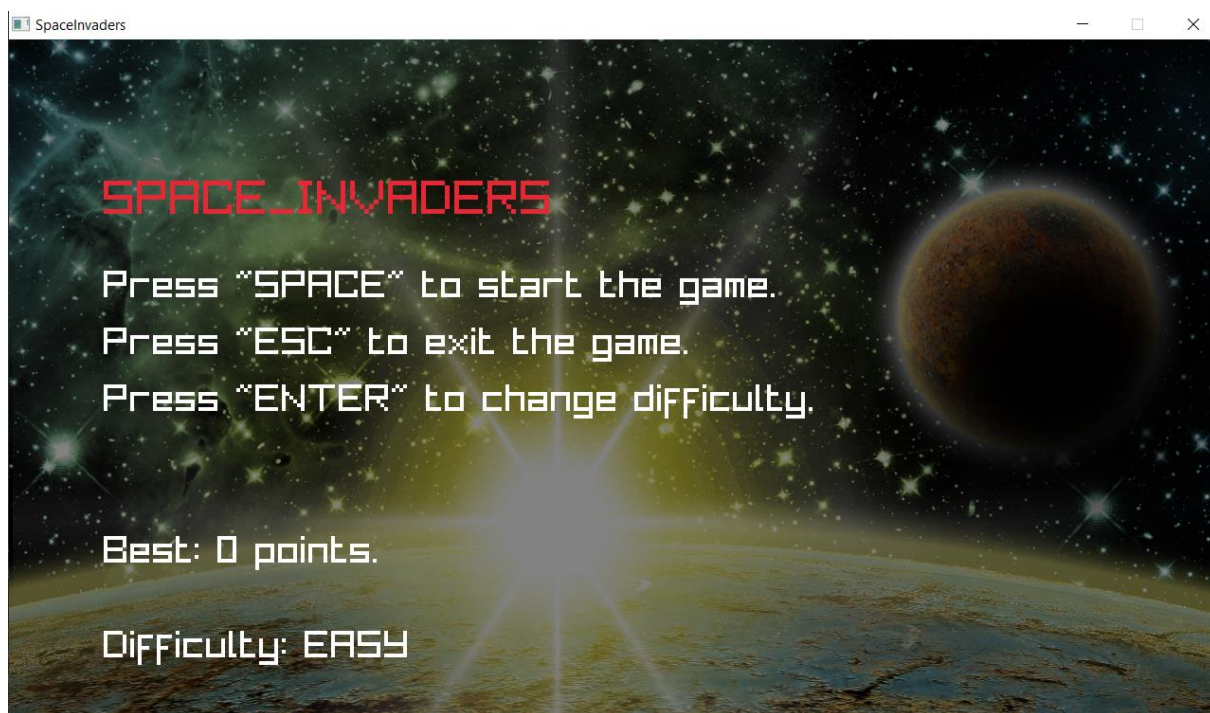
## **III. Instrukcja kompilacji:**

W celu skompilowania projektu należy otworzyć go w środowisku programistycznym intelliJ i uruchomić poprzez zakładkę „Run” bądź poprzez narzędzie cmake wykonując komendę „cmake”.

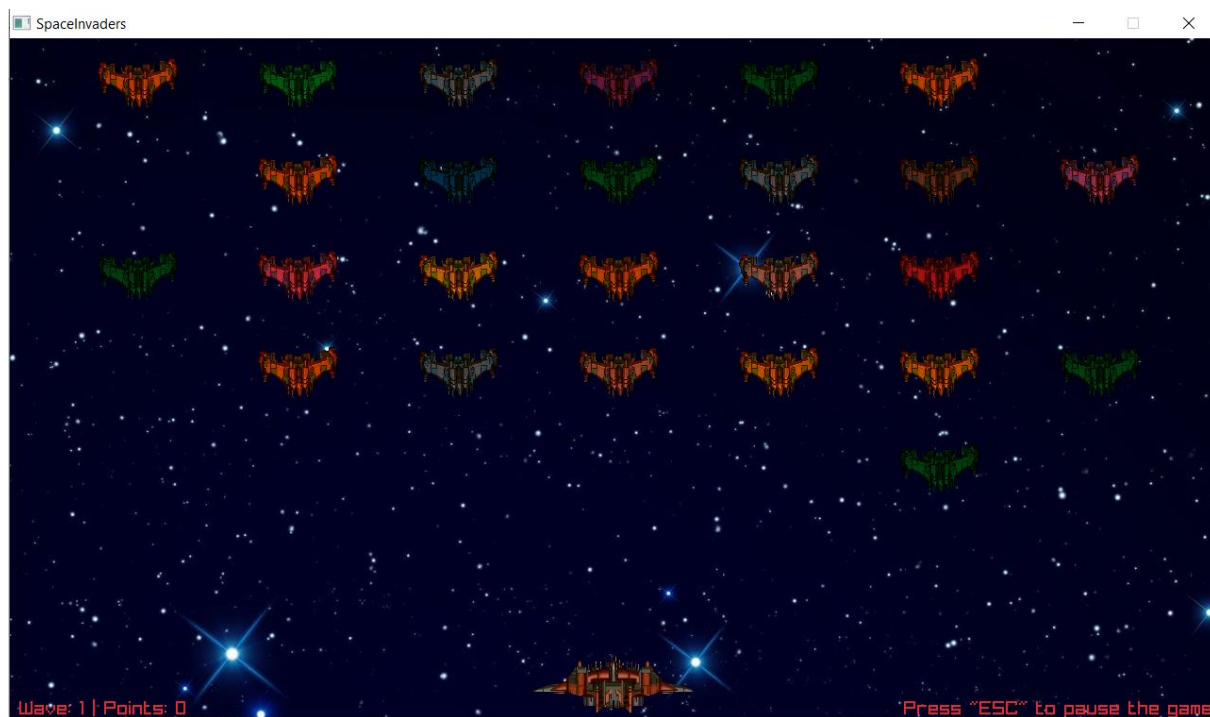
## **IV. Ogólne zasady gry.**

Po uruchomieniu gry widać proste menu, w którym możemy rozpocząć grę klawiszem SPACE, opuścić grę klawiszem ESC lub zmienić poziom trudności gry klawiszem ENTER. Dostępne są trzy poziomy trudności: łatwy, średnio trudny lub trudny. Na ekranie startowym widoczny jest również najlepszy zdobyty wynik. W grze poruszamy się w osi poziomej w kierunkach lewym i prawym z wykorzystaniem odpowiednich strzałek na klawiaturze, a strzały można oddawać z wykorzystaniem klawisza Spacji. Po zestrzeleniu wszystkich przeciwników z planszy, mamy możliwość kontynuować grę z następną falą, czyli z przeciwnikami poruszającymi się coraz szybciej i dostawać więcej punktów za każdy trafiony strzał. Zmieniając trudność gry, zwiększany jest numer fali na 6 lub 11. W przypadku gdy którykolwiek z przeciwników przejdzie całą planszę w dół, przegrywamy. Najlepszy wynik jest zapisywany.

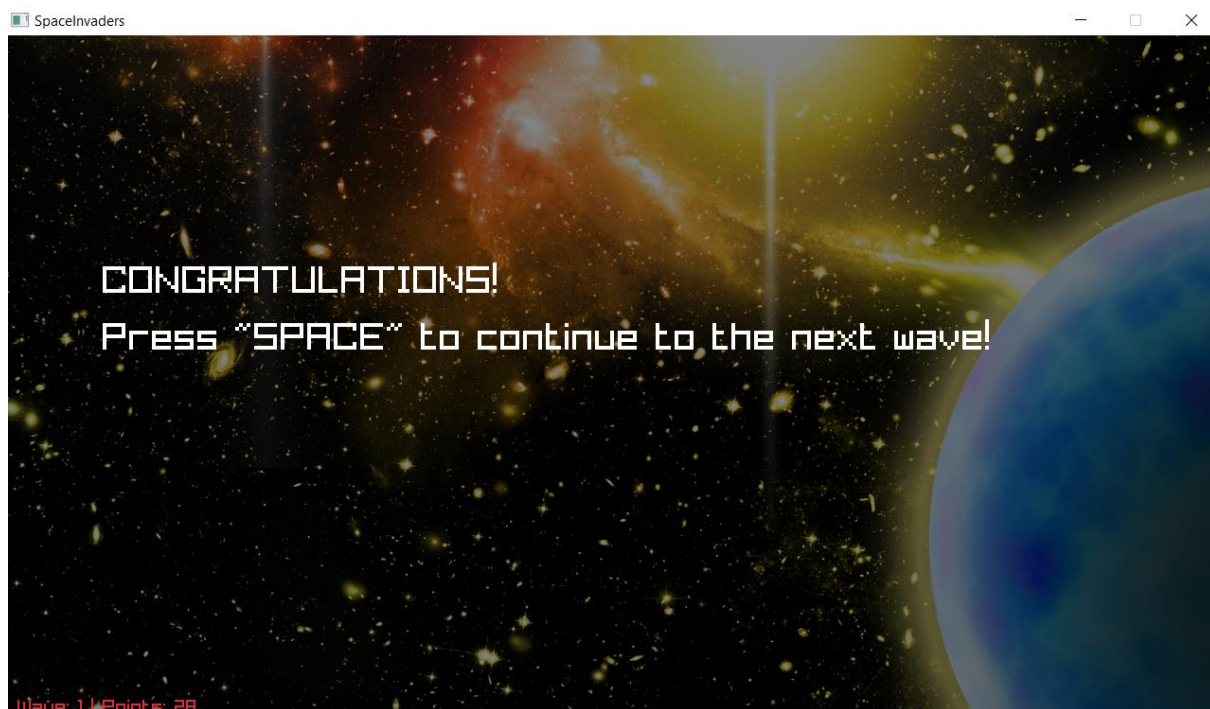
## V. Przykładowe zrzuty ekranu z gry.



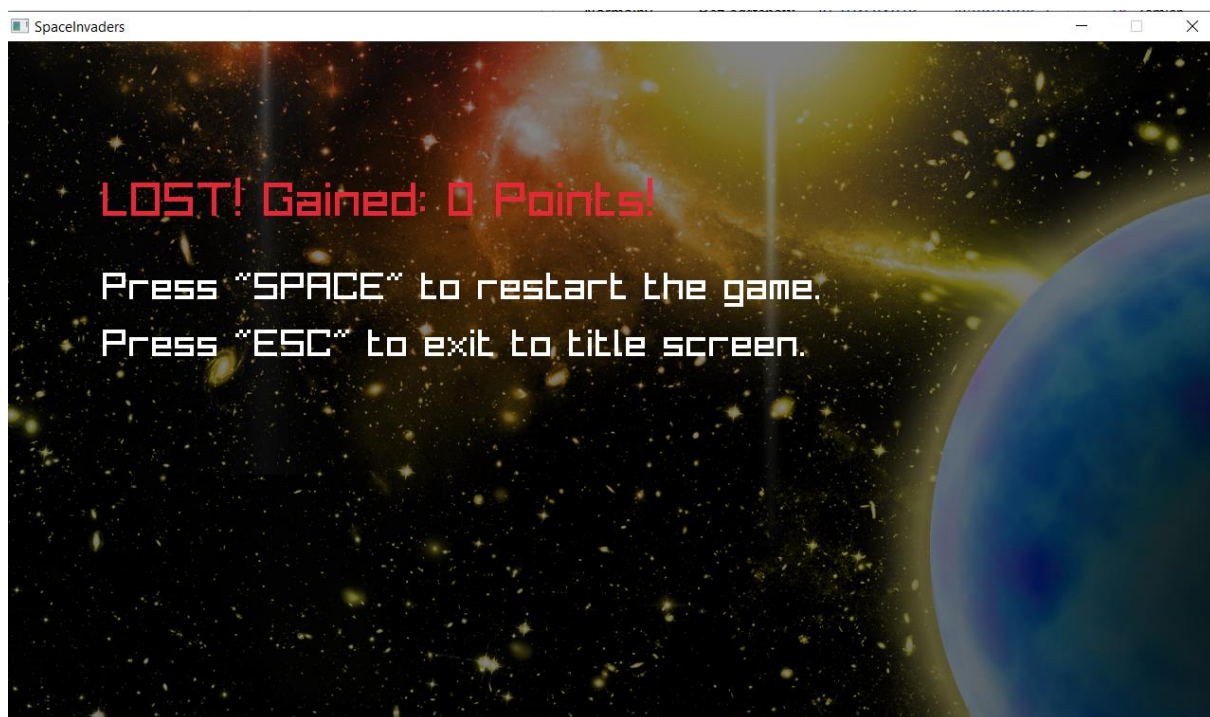
Rysunek 1 Menu startowe



Rysunek 2 Tryb gry



Rysunek 3 Widok po zakończeniu poziomu z sukcesem



Rysunek 4 Widok po przegraniu

## **VI. Struktura projektu.**

- a) source/assets/ - katalog zawiera zasoby wykorzystane do stworzenia interfejsu graficznego programu. Są to tekstury dla poszczególnych widoków w grze, tekstury dla obiektów wrogów, gracza i pocisków oraz ścieżki dźwiękowe odtwarzane w przypadku wystrzału i eksplozji oraz główna ścieżka dźwiękowa gry.
- b) source/ - katalog zawiera definicje i implementacje zdefiniowanych metod dla modeli wykorzystywanych w grze. Są to m.in. modele Gracza, Wroga czy Pocisku, ale również model Licznika Punktów i pomocnicza klasa dysponująca dodatkowymi metodami. Modele zdefiniowane są w plikach nagłówkowych (\*.h), natomiast implementacja poszczególnych metod znajduje się w plikach (\*.cpp).
- c) main.cpp – plik zawierający deklaracje zmiennych i typów globalnych, funkcje: inicjalizująca grę, aktualizująca jej stan, ładująca zasoby, rysująca tekstury na ekranie oraz zapisująca najlepszy wynik w pliku. Zawiera główną logikę gry, do zrealizowania której wykorzystuje wspomniane modele, zmienne globalne i załadowane zasoby.
- d) html/ - katalog z plikami dokumentacji. Aby wyświetlić dokumentację należy otworzyć plik index.html (znajdujący się w tym katalogu) w przeglądarce.

## **VII. Podsumowanie.**

Projekt został zrealizowany zgodnie z początkowymi założeniami. Wykorzystuje bibliotekę umożliwiającą programowanie interfejsów graficznych Raylib oraz język c++. Gra jest atrakcyjna dla potencjalnego użytkownika, wykorzystuje mechanizm gratyfikacji za wygrany poziom poprzez zwiększenie liczby punktów możliwych do zdobycia oraz posiada różne poziomy trudności, które stanowią formę wyzwania. Wykorzystane tekstury, ścieżki dźwiękowe i transformacje tekstur uatrakcyjnają rozgrywkę. Program zapewnia obsługę klawiatury do nawigowania i sterowania w grze. Kod źródłowy programu stworzono zgodnie z zasadami tworzenia czystego kodu, zachowując podział na pliki nagłówkowe i źródłowe oraz zachowując odpowiednią strukturę katalogów. Do programu została wygenerowana dokumentacja zgodnie z zasadami opisywania kodu oprogramowania Doxygen.