

# **PK4 Projekt**

## **Temat: Lunar Lander**

Sprawozdanie

Jakub Krzywoń

Prowadzący: dr inż. Krzysztof Taborek

5.07.2023

## 1.Temat: Gra Lunar Lander

Lunar Lander to dwuwymiarowa, zręcznościowa gra symulacyjna. Gracz kontroluje lądownik, który jest przyciągany przez siłę grawitacji, a jego zadaniem jest odpowiednie zarządzanie paliwem i użycie silnika, by bezpieczne wylądować we wskazanym miejscu. Za każde udane lądowanie zdobywa punkty, a za nieudane traci paliwo. Gracz gra aż do skończenia się paliwa.

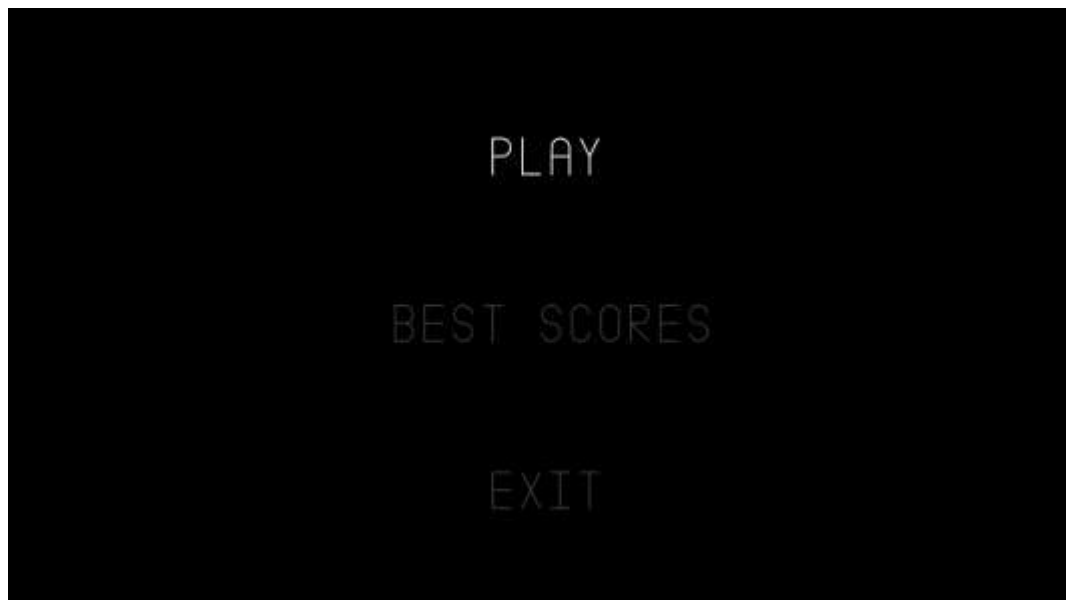
## 2.Specyfikacja zewnętrzna

### Instrukcja

Poruszanie się po menu odbywa się za pomocą strzałek oraz przycisku enter.

Po wybraniu opcji PLAY zaczyna się rozgrywka, aby obrócić lądownik należy użyć strzałek w lewo lub prawo, do uruchomienia silnika służy strzałka w górę. Z rozgrywki zawsze można wyjść naciskając esc. Po zakończeniu gry należy wpisać poprawny nick i zatwierdzić go klikając enter albo nacisnąć esc (wtedy wynik się nie zapisze). Opcja BEST SCORES wyświetla 10 najlepszych wyników, aby z niej wyjść trzeba nacisnąć enter lub esc. Opcja EXIT kończy działanie aplikacji.

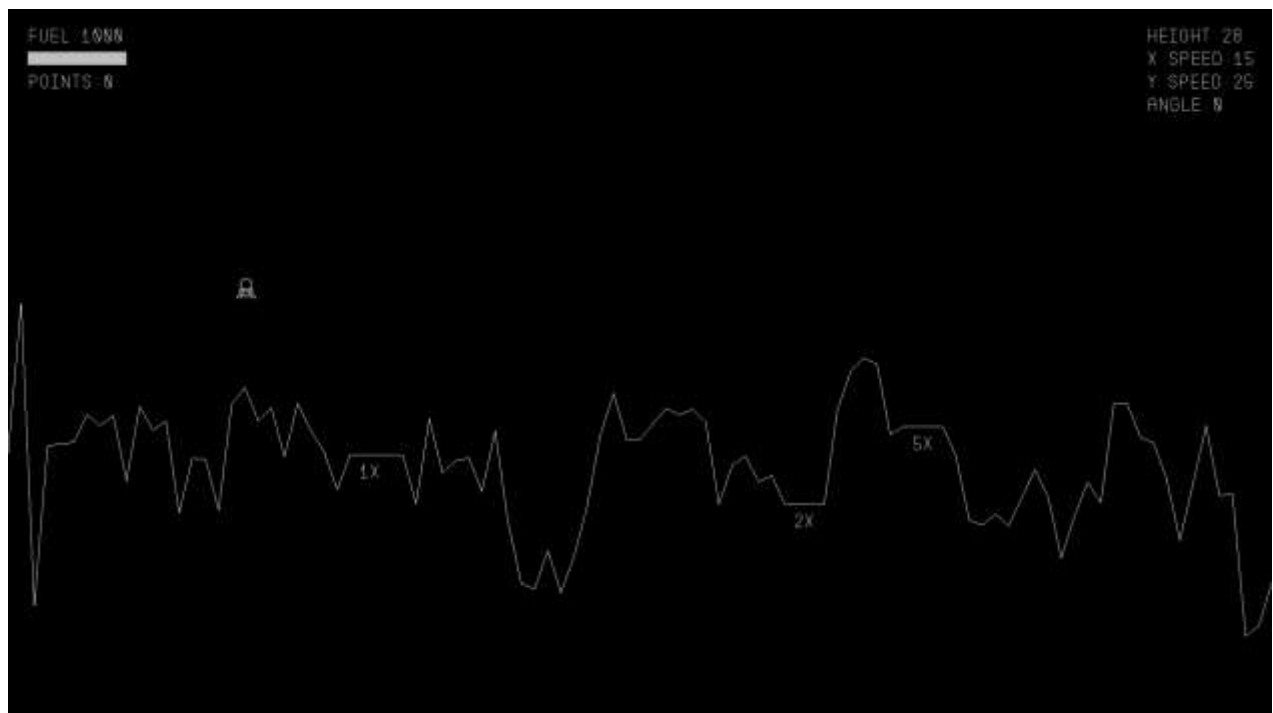
### Menu:

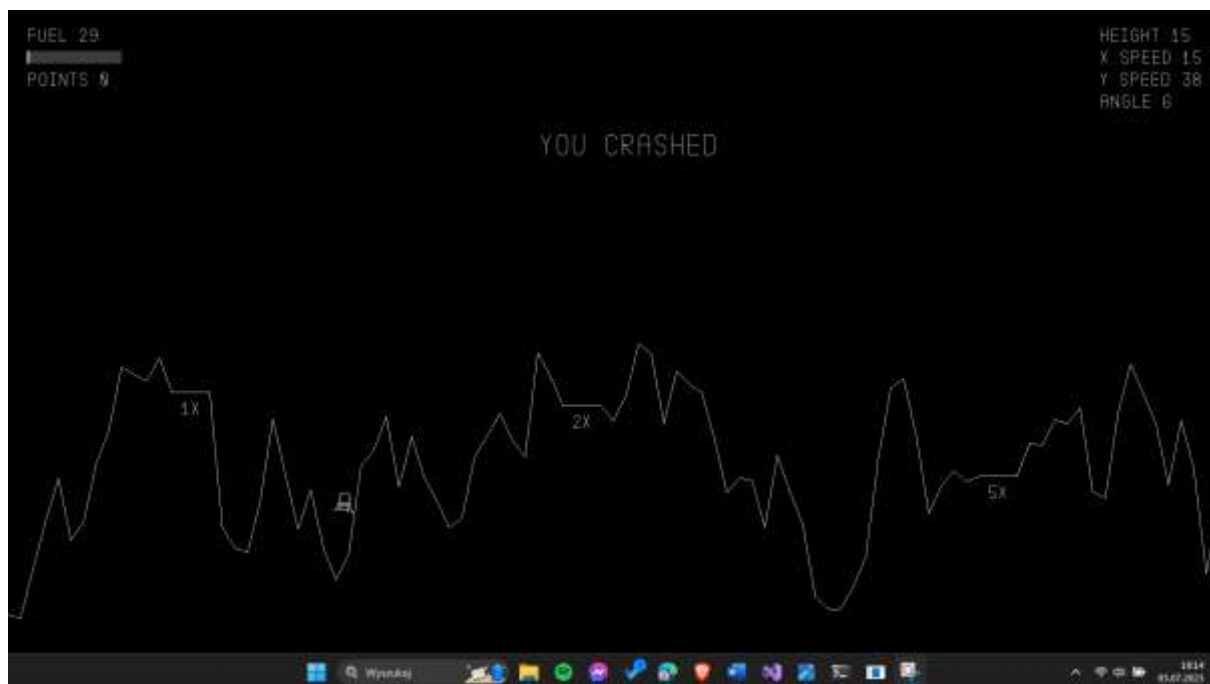


Best Scores:

1	JAKUB	215%
2	TESTOWE10	95%
3	JAKUB	60%
4	ASD	60%
5	JAKUB	50%
6	TEST2	40%
7	TEST	35%
8	JKE	30%
9	SKEDUDE	1%
10	ZERO	0

Rozgrywka:





Ekran końcowy:



### 3. Specyfikacja wewnętrzna

Schemat blokowy działania gry:

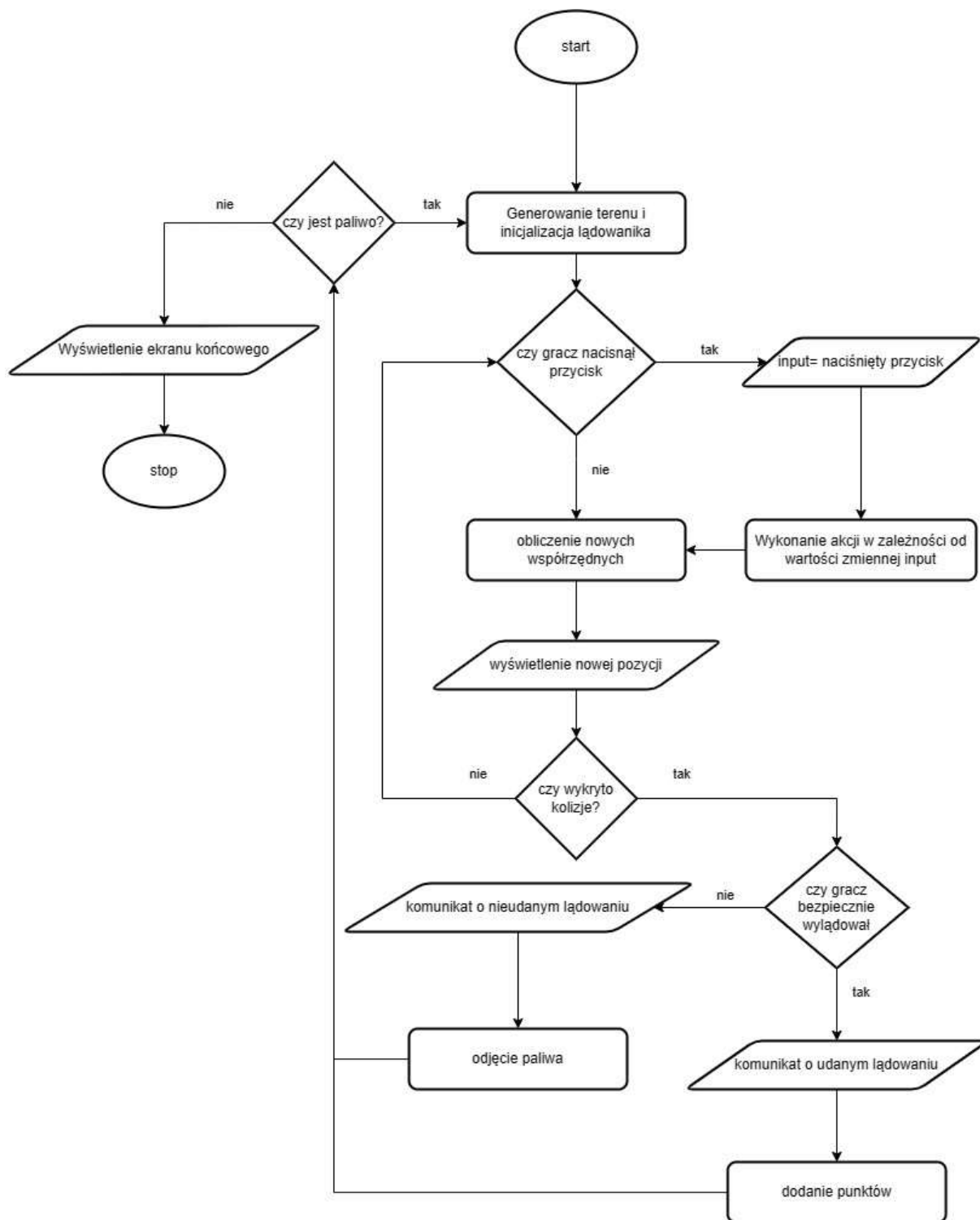
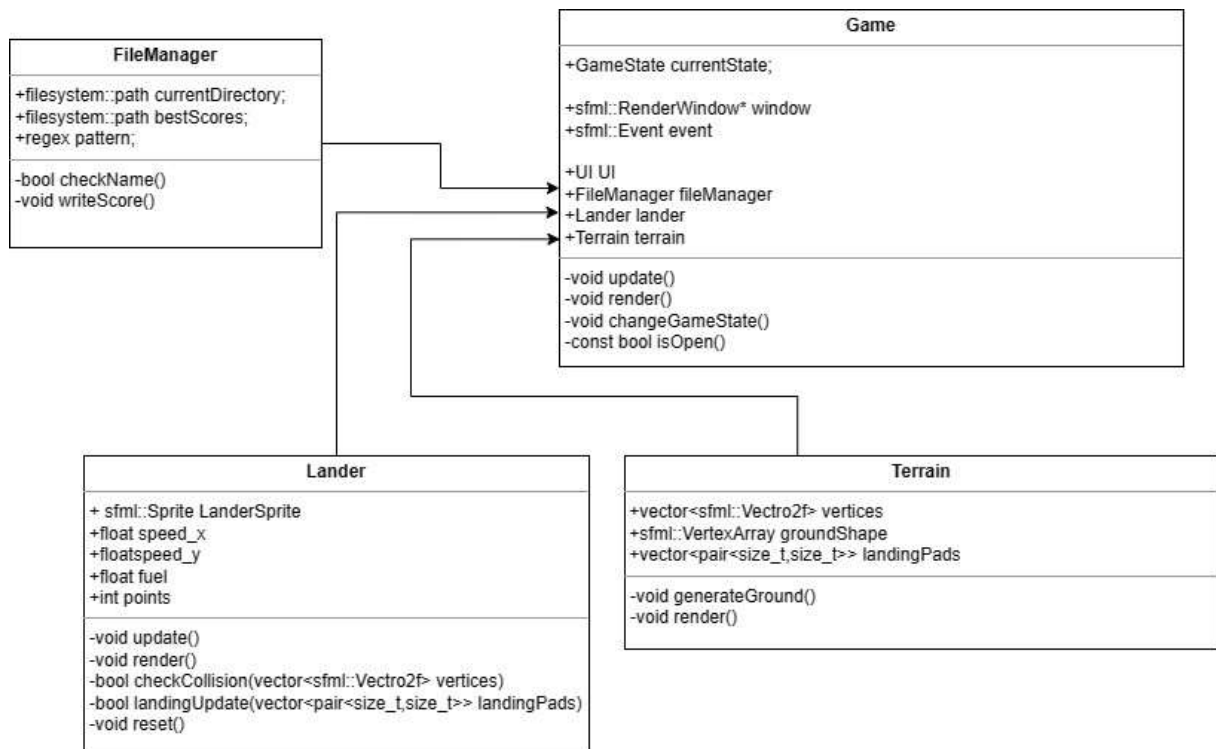


Diagram klas:



Klasy:

- Game – główna klasa organizująca działanie gry
- Lander – reprezentuje lądownik, sprawdza kolizję, przechowuje punkty gracza
- Terrain- reprezentuje teren, przechowuje położenie miejsc do lądowania, wykorzystuje bibliotekę random do generowania losowego terenu
- FileManager- wykorzystuje biblioteki regex i filesystem, przechowuje ścieżkę do pliku z najlepszymi wynikami oraz posiada metodę sprawdzającą poprawność nicku, korzysta z biblioteki algorithm aby sortować wyniki

Wykorzystane biblioteki z laboratorium:

Filesystem – ustalenie ścieżki do pliku z najlepszymi wynikami oraz przechowywanie jej

Regex – sprawdzenie poprawności nicku wprowadzonego przez gracza

## 4. Testowanie i uruchamianie

Przy testowaniu wykrywania kolizji wbudowana w sfml metoda wykrywania kolizji intersects która miała sprawdzać czy dwa bounding boxy obiektów się nie przecinają nie działała prawidłowo, wykrywała kolizję przed faktycznym zderzeniem. Próbowałem wykryć kolizję między obracanym spritem a punktami reprezentującymi teren poprzez aproksymacje linii terenu pod lądownikiem i sprawdzaniem jej przecięcia z bounding boxem sprite'a, jednak ta metoda nie spełniała moich oczekiwań w przypadku dużej różnicy poziomu między dwoma sąsiednimi punktami. Ostatecznie zastosowałem metodę korzystającą z teorii SAT, poprzez rzutowanie środka sprite'a na odcinek oraz obliczenie nakładania się na osi x i y.

Wykrycie błędu polegającego na braku czyszczenia wektora z stanowiskami lądowania przy każdym generowaniu terenu, co powodowało niepoprawne naliczanie punktów.

## 5. Uwagi i wnioski

Aby rozbudować program można do niego dodać:

- dodatkowe pola lądowania ( np. z paliwem)
- komunikat o niepoprawnym nicku
- animacje silnika
- dźwięk
- bardziej wydajną i dokładną detekcję kolizji