

Projekt - Programowanie w języku Java

Temat projektu:

Gra Space Invaders

Członkowie zespołu:

Adam Rzepka i Bartłomiej Tokar

Grupa 2ID13A, Studia stacjonarne 1 stopnia, 2 rok

1. Ogólny opis projektu

Tematem naszego projektu było stworzenie gry Space Invaders przy użyciu języka programowania Java. Oryginalna gra Space Invaders pochodzi z Japonii, pierwszy raz została wydana w roku 1978 przez Taito na automaty do gier. Oryginalna wersja gry polegała na zniszczeniu wszystkich statków kosmitów, które powoli zjeżdżały w dół ekranu. W przypadku naszej wersji gry zdecydowaliśmy się na drobną modyfikację rozgrywki - gracz ma większą swobodę ruchów, a kosmici lecą z losowych miejsc na ekranie, dodatkowo w naszej grze dodaliśmy tzw. bossów - mocniejsze statki, które nie są tak łatwe do strącenia jak zwykli kosmici. Do realizacji projektu użyliśmy biblioteki LibGDX - specjalnie zaprojektowanej do tworzenia gier na różne platformy bazującej na OpenGL.

2. Informacje na temat funkcjonalności

Nasza gra jest zbudowana w prosty i czytelny dla użytkownika sposób. Każdy ekran ma swoją własną muzykę, a klikanie guzików w menu wydaje dźwięki. W grze są w sumie 3 ekrany: menu gry, ekran gry oraz menu końca gry.

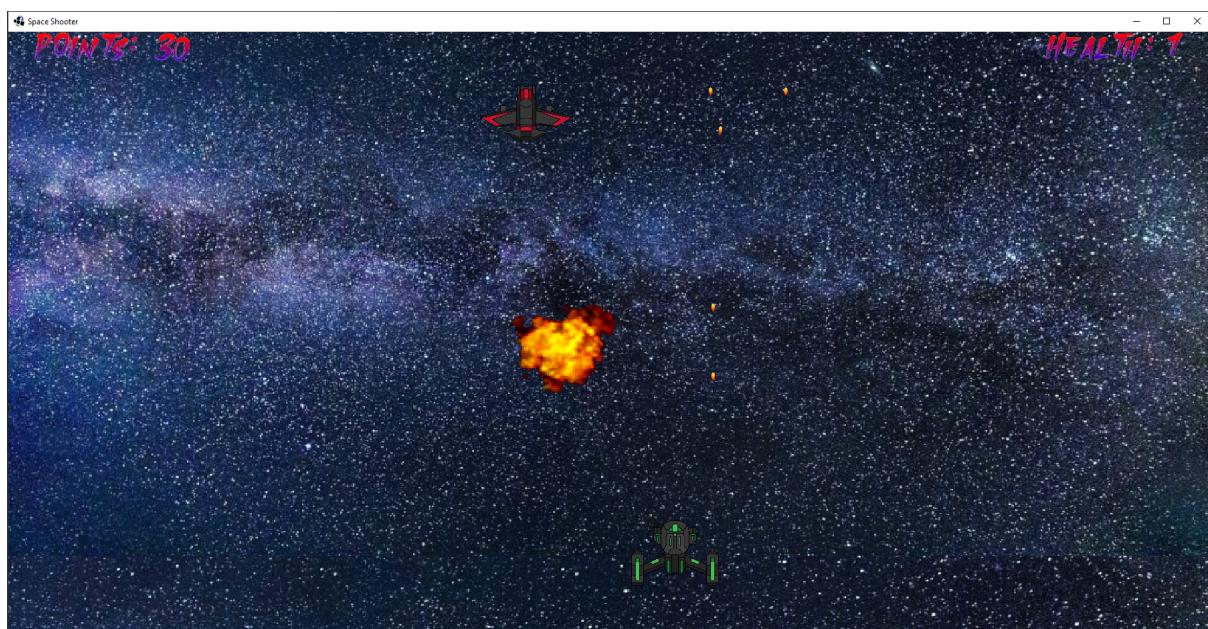
Menu gry posiada 3 przyciski:

- przycisk Play - powoduje rozpoczęcie rozgrywki,
- przycisk Info - powoduje wyświetlenie informacji na temat twórców gry,
- przycisk Exit - powoduje wyjście z gry. Z gry można wyjść również za pomocą przycisku 'x'.



zdjęcie podglądowe menu gry

Po kliknięciu guzika Play przechodzimy bezpośrednio do rozgrywki. W tym momencie możemy już przejąć kontrolę nad statkiem gracza. Pojawia się również HUD (Head-up display) wyświetlający ilość życia gracza, oraz ilość zdobytych punktów. Punkty są przydzielane za niszczenie statków przeciwnika.



ekran rozgrywki, w lewym górnym rogu widać licznik punktów, w prawym górnym rogu licznik zdrowia gracza. Statek gracza jest z zielonymi akcentami, natomiast statki przeciwnika posiadają akcent czerwony.

Nasza gra, tak samo jak podobne tego typu gry "shoot'em up" nie ma "końca". To znaczy, że dopóki graczu nie skończy się życie, może grać cały czas, pobijając swoje poprzednie rekordy zdobytych punktów. Gracz traci punkty życia w dwóch przypadkach:

- gdy statek przeciwnika przekracza dolną granicę ekranu,
- gdy statek gracza zostanie trafiony przez bossa.

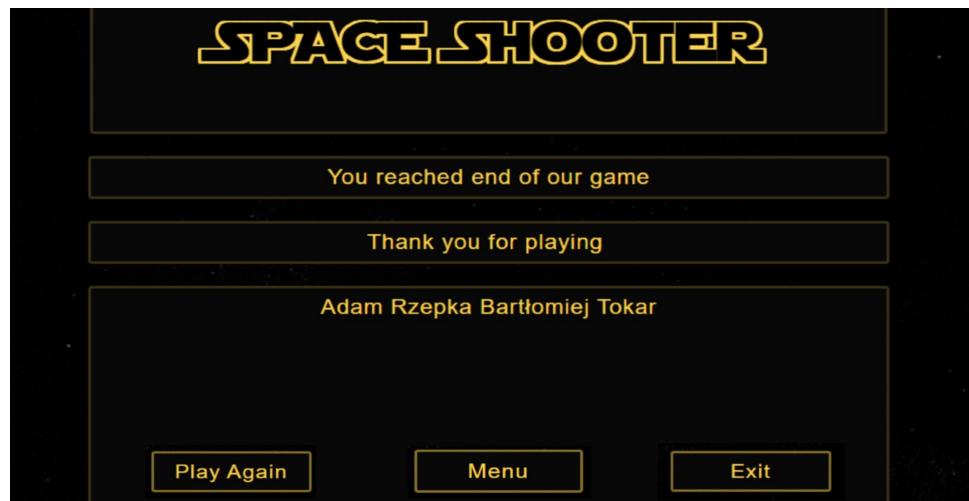
Po zdobyciu określonej liczby punktów, na ekranie pojawi się boss - czyli mocniejszy kosmita. Charakteryzuje się on między innymi:

- o wiele większym rozmiarem, niż zwykły statek przeciwnika,
- bardziej nieprzewidywalnymi ruchami - statek bossa porusza się również na boki,
- w porównaniu do zwykłych przeciwników, statek bossa będzie również strzelał do gracza, oraz nie będzie próbował przekroczyć dolnej granicy ekranu.

Zniszczenie statku bossa daje większą nagrodę punktową, oraz zwiększa ilość punktów potrzebną na pojawienie się kolejnego bossa.



pojawienie się statku bossa



Ekran po zakończeniu rozgrywki

Nasza gra jest również w pełni udźwiękowiona - każdy ekran gry ma swoją ścieżkę dźwiękową, wystrzały i wybuchy także mają osobne efekty dźwiękowe.

3. Informacje na temat obsługi

Projekt pobieramy z githuba i uruchomiamy za dzięki klasie DesktopLauncher

Sterowanie w grze jest podobne do innych tego typu gier. Za pomocą strzałek zmieniamy pozycję statku gracza, za pomocą spacji strzelamy. Warto zaznaczyć, że pojedyncze wcisnięcie spacji wyśle tylko jedną salwę pocisków.



DesktopLauncher - Klasa uruchamiająca grę. Posiada podstawowe informacje o konfiguracji gry takie jak nazwa czy rozdzielczość.

MyGdxGame - Główna klasa gry. Zmieniamy za jej pomocą sceny.
SetScreenTo...() - Metody odpowiedzialne za zmienienie sceny.

Player – Klasa w której tworzymy gracza
Draw() - Funkcja umieszcza grafike na ekranie
IntersectsPlayer() - Metoda tworząca prostokąt na podstawie którego tworzymy “hitbox” gracza

Menu - Klasa zawierająca menu gry. Na backgroundzie są tworzone przyciski. Wybieramy daną opcję klikając w nią myszką.

KoniecGry - Jest to menu pomocnicze wyświetlające się w momencie porażki. Znajdują się na nim słowa od autorów dla gracza oraz przyciski dzięki którym możemy wrócić do rozgrywki, do menu lub wyjść z gry.

Info - Jest to jedna z opcji wyświetlającej się w menu. Po wybraniu jej wyświetlają się autorzy gry wraz z opisem nad czym każdy z autorów pracował przy projekcie.

GameScreen – Klasa odpowiadająca za gameplay czyli serce całego projektu. To tutaj wyświetlamy gracza oraz przeciwników.
KeyInput() - Odpowiada za obsługę klawiszy w grze. Strzałką przypisujemy poruszanie się a spacjami strzały.

Napisy() - Odpowiada za wyświetlenie napisów w górnych rogach ekranu. Dzięki niej wiemy ile mamy punktów oraz ile punktów życia nam zostało.

SpawnPrzeciwnikaCoXsec(float SprawnTime) - Powoduje utworzenie statku przeciwnika co podaną ilość czasu. Czas zmienia się wraz z ilością zabitych bosów.

RenderExplosions() - Odpowiada za wyświetleniu eksplozji na ekranie.

Explosion - Klasa tworząca eskpozję. Efekt uzyskujemy dzięki wczytywaniu kilku obrazów do tablicy.

EnemyBullet - Klasa tworząca pociski przeciwnika. Przeciwnik wystrzela pociski raz na określony z góry czas. Posiada metody odpowiedzialne za utworzenie hitboxu oraz poruszanie się.

Enemy - Klasa tworząca statek przeciwnika. Posiada metody tworzące hitbox oraz poruszanie się.

Bullet - Klasa tworząca pociski gracza. Posiada metody odpowiedzialne za hitbox oraz poruszanie się

Boss – Klasa tworząca bosa. Bos porusza się tylko na zmiennej x. Jest uzbrojony w 4 działała którymi strzela w stronę gracza. Gdy boss wyświetla się na ekranie to spawn innych przeciwników jest zatrzymany. Posiada metodę odpowiedzialną za hitbox.

`BossMovement()` - Metoda odpowiedzialna za poruszanie się bosa. Porusza się tylko po zmiennej x a gdy doleci do końca ekranu to zmienia swój ruch w drugą stronę.

Włożona praca przez autorów

Adam Rzepka:

- Podstawowe struktury gry
- Ruch gracza
- Kolizje
- Spawn Przeciwników
- Poruszający się background

Bartłomiej Tokar

- Grafika i muzyka
- Efekty dźwiękowe
- Boss
- Eksplozje animacja
- Poruszanie pocisku