



# **Politechnika Śląska**

## **OiRPOS**

**Gra Space Invaders z elementami sztucznej inteligencji**

Autor: Szymon Gretka

Gliwice, Grudzień 2020

## 1. Wstęp

Ten open source'owy projekt składa się z gry stworzonej na urządzenia mobilne z Androidem oraz z prostej aplikacji webowej, która ma za zadanie obsługiwać (CRUDowe operacje) dane użytkownika: np. najlepsze wyniki. Gra ta jest oparta na uniwersum Space Invaders zapoczątkowanej w latach 70' ubiegłego wieku. W grze zostały dodane elementy sztucznej inteligencji (skończona maszyna stanów - eng. Finite State Machine). Linki do zdanego repozytorium dla:

- gry: <https://github.com/szymongretka/android-project>
- aplikacji webowej: <https://github.com/szymongretka/spring-thesis>

## 2. Opis działania

W dużym skrócie, celem gry jest przejście wszystkich poziomów podczas których pojawiają się kolejne "fale" przeciwników i nie dać się zestrzelić. W trakcie rozgrywki można zdobywać punkty, za które następnie można zakupić lepszy statek kosmiczny. Jest również możliwość sprawdzenia wyników 10 najlepszych graczy. Aby poziomy nie był monotoniczny, postanowiłem, aby dodać np. tzw. "bossów" którzy poza wzmocnionymi statystykami, operują na wspomnianej wcześniej sztucznej inteligencji.

Z dostępnych możliwości implementacji sztucznej inteligencji w grze można wyróżnić:

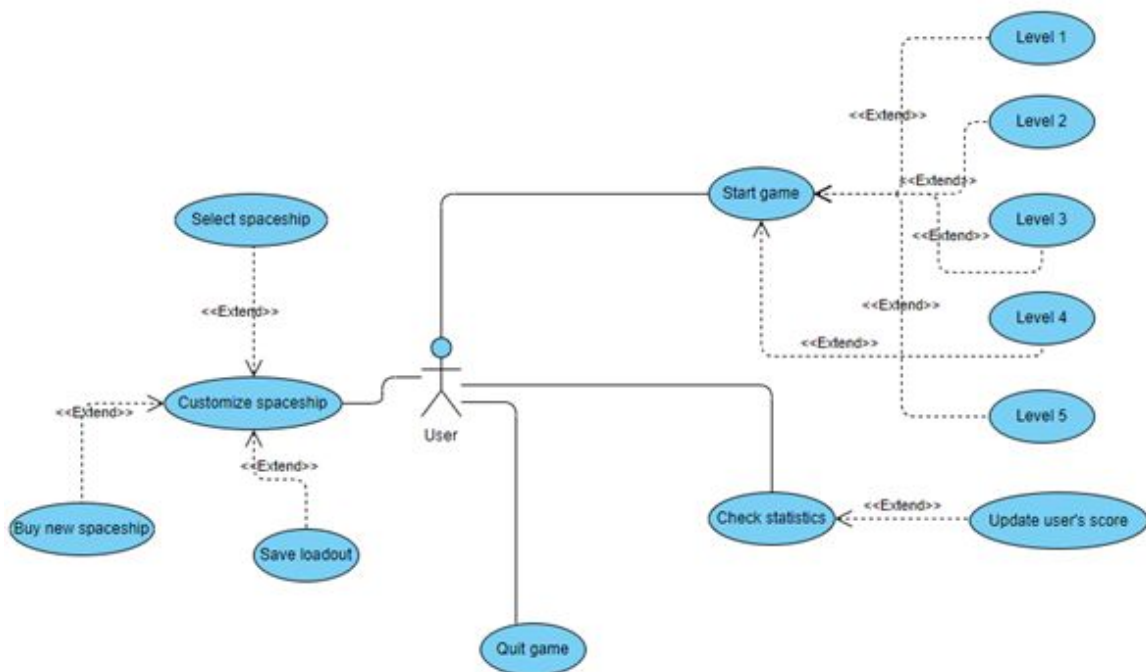
- sieci neuronowe - są one zwykle używane do symulacji zachowania gracza nad NPC (non-player character). Wynika to z faktu, że sieci neuronowe trzeba najpierw „wytrenować”, stąd raczej nie są wykorzystywane w branży gier.
- pathfinding - to coś pomiędzy podejmowaniem decyzji a ruchem AI. Dla każdego NPC, AI musi obliczyć zoptymalizowaną trasę przez świat gry, aby dotrzeć do punktu docelowego.
- movement AI - niektóre gry dotyczą głównie algorytmów ruchu, więc nie wymagają skomplikowanej techniki AI (np. gry wyścigowe). Ruchowa sztuczna inteligencja polega na rozsądnym przemieszczaniu postaci po świecie gry.
- decision making - Można wskazać dwie odrębne podsekcje: maszyny stanowe i drzewa zachowań. Krótko mówiąc, podejmowanie decyzji polega na tym, że postać przetwarza zestaw informacji, co skutkuje wykonaniem akcji, która zostanie następnie wykonana.

W tej grze wykorzystałem ww. skończoną maszynę stanów (FSM). Powody, dla których został wybrany spośród innych to:

- wiele podmiotów zmienia się wykonując akcje w oparciu o pewne stany wewnętrzne (poruszają się w granicach ekranu).
- stan można rozbić na inną stosunkowo niewielką liczbę odrębnych opcji
- agenci gry muszą radzić sobie z seriami wejść lub wydarzeń w określonym czasie (np. punkty zdrowia gracza).

Komponenty z jakiej składa się ta aplikacja to: menu główne, sklep, tablica wyników oraz kolejne poziomy gry.

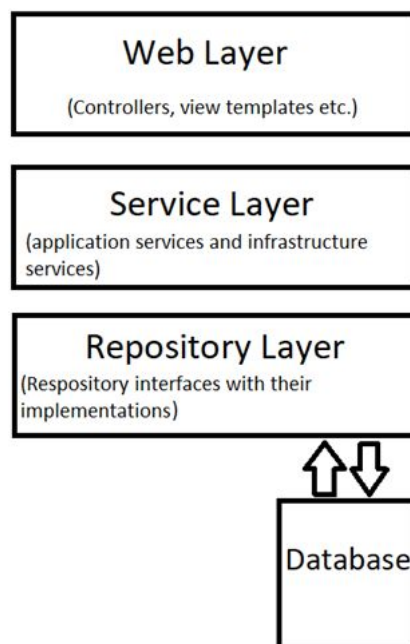
Poniżej znajduje się diagram przypadków użycia (Rys. 1).



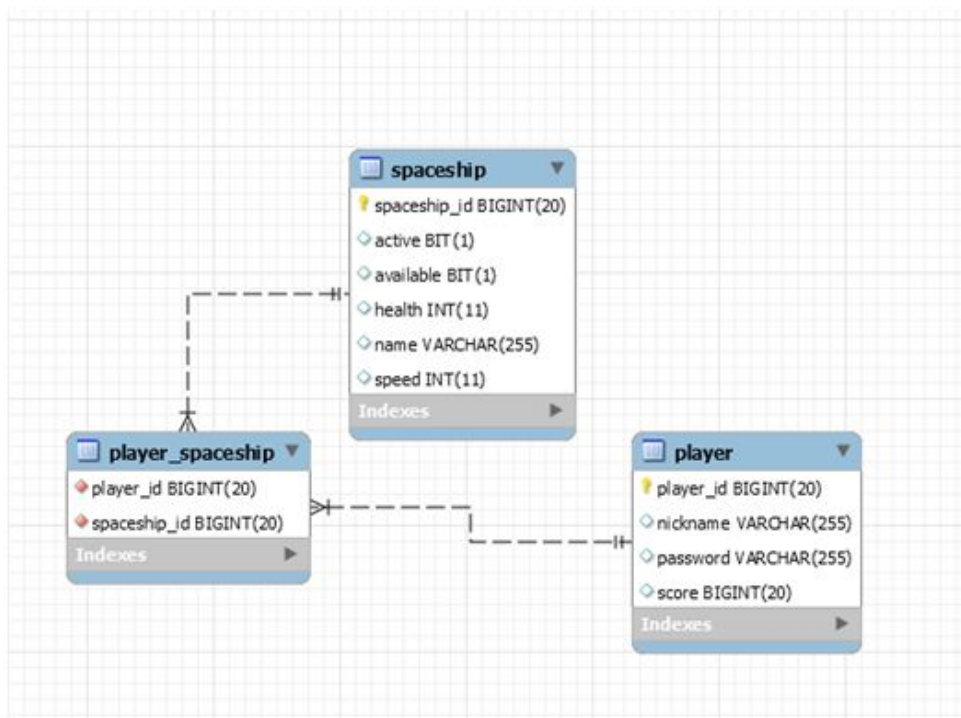
Rys. 1 Diagram use case

### 3. Szczegóły implementacji oraz użyte technologie

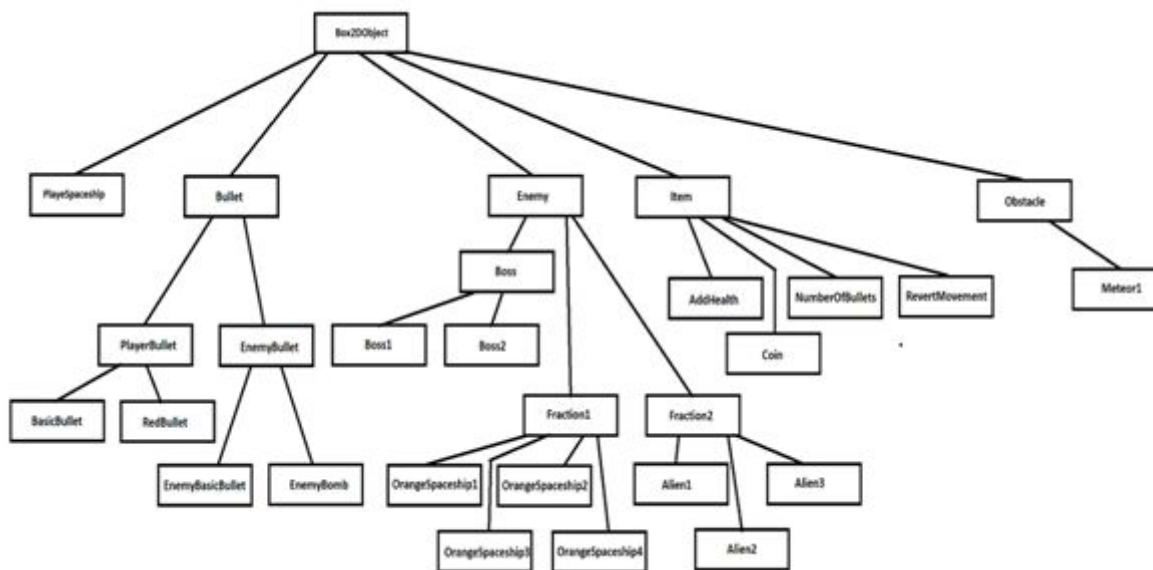
Aplikacja webowa (a ściślej mówiąc REST API) jak i gra zostały w całości napisane w języku JAVA. Jeśli chodzi o tę pierwszą to został wykorzystany framework Spring. Za integrację z bazą danych (MySQL) odpowiada Hibernate. Architektura aplikacji składa się z następujących warstw:



Jeśli chodzi o strukturę bazy danych to diagram ERD wygląda w następujący sposób:



Przechodząc do gry, to wykorzystana została biblioteka **LibGDX** (wyświetlanie grafiki, odtwarzanie muzyki, ładowanie assetów gry itd.), natomiast do symulacji obiektów (przemieszczanie wykrywanie kolizji itd.) **Box2D**. Całość została opracowana w IDE Android Studio. Poniżej widnieje schemat wszystkich obiektów "jednostek" wykorzystanych w grze.



Z ciekawostek, warto dodać jakie techniki zostały wykorzystane w celu optymalizacji aplikacji:

- asset manager - ładowanie tekstur w sposób asynchroniczny
- "pooling" obiektów - w celu ograniczenia "kolekcjonowania śmieci" (eng. garbage collecting), obiekty są otrzymywane z specjalnie przygotowanej struktury danych która je przechowuje. Normalnie, gdy obiekt byłby klasyfikowany do usunięcia, jego atrybuty są resetowane i jest on gotowy do ponownego użycia. W przypadku, gdy nie

będzie żadnych obiektów gotowych do ponownego użycia, objętość ww. struktury danych zostaje powiększona o pożądaną wartość.

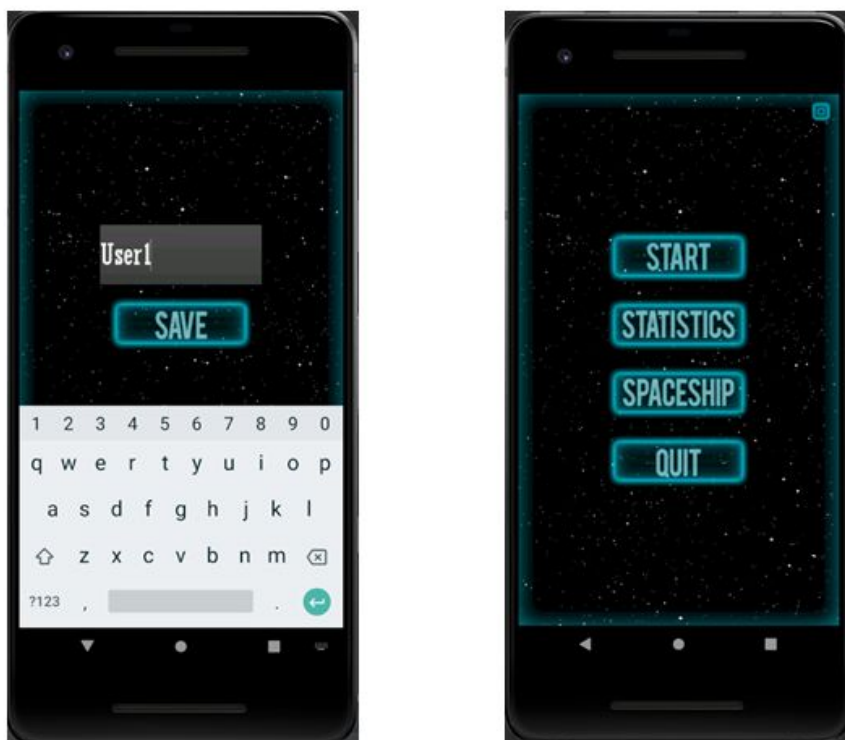
- "Packer" tekstur - aby zapobiec bindowaniu (relatywnie kosztownej operacji) każdej teksturze z osobna, którą trzeba narysować, lepiej przechowywać dużo mniejszych tekstur na większym obrazie, "zszyć" (zbindować) obraz raz, a następnie wielokrotnie odwoływać się do tych poszczególnych obszarów.

#### 4. Jak uruchomić?

Aby uruchomić aplikację, należy ją najpierw pobrać korzystając z tego linku: <https://github.com/szymongretka/android-project/releases>. Następnie, aby móc zainstalować tę grę, konieczne jest zaznaczenie opcji „instaluj aplikacje z nieznanego źródła”. Umożliwia to instalowanie programów spoza rynku Google. Tę funkcję można znaleźć w ustawieniach w części dotyczącej bezpieczeństwa. W tej chwili tę grę można uruchomić tylko na urządzeniach z systemem operacyjnym Android (w przyszłości planowane jest również udostępnienie na innych platformach - jak iOS czy Windows).

#### 5. Przykłady działania

Główną funkcjonalnością tego systemu jest umożliwienie użytkownikowi zagrania w grę opartą na uniwersum Space Invaders. Ponadto zastosowane jest sprawdzanie najlepszych wyników i dostosowywanie zdolności statku kosmicznego. Generalnie, po zainstalowaniu i uruchomieniu gry po raz pierwszy, użytkownik jest proszony o ustawienie swojego pseudonimu (Rys. 2), później na jego podstawie dokona zmian w swoim wyposażeniu i może załadować swój najlepszy wynik. Niestety, nie ma systemu logowania do konkretnego konta użytkownika, więc mogą pojawić się pewne niedogodności, jeśli dwóch lub więcej użytkowników ustawi ten sam pseudonim.



Rys. 2 Tworzenie nowego użytkownika oraz menu główne

Jeśli gracz ustawi swój pseudonim, może przejść do menu głównego (Rys. 2). Dostępnych jest kilka opcji: Rozpocznij grę, sprawdź najlepsze wyniki, dostosuj statek kosmiczny lub zakończ grę. Począwszy od końca - po naciśnięciu przycisku Zakończ gra po prostu zamyka się i usuwa wszystkie załadowane zasoby (assets). Jeśli użytkownik chce dostosować swój statek kosmiczny, może to osiągnąć naciskając odpowiedni przycisk w menu głównym. Po otwarciu ekranu personalizacji (Rys. 3) może wybrać, który statek kosmiczny chce kupić (jeśli jeszcze nie został przez niego kupiony), wybrać i ustawić aktywny (tylko te, które zostały już zakupione), a poza tym wprowadzić zmiany lokalnie użytkownik ma możliwość zapisywania zmian w chmurze (wymagane połączenie internetowe). Na Rys. 4 widoczny jest widok odpowiedzialny za wyświetlanie wyników najlepszych graczy.



Rys. 3



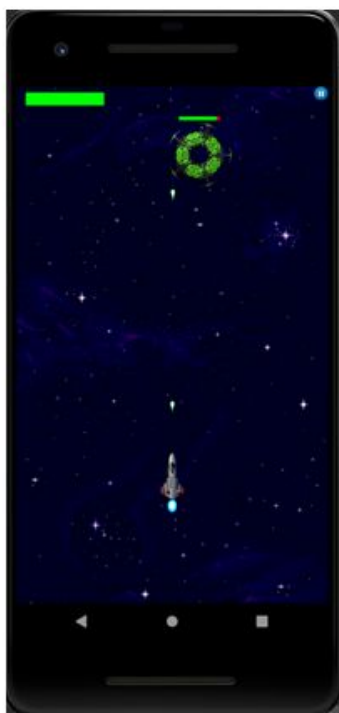
Rys. 4

Po naciśnięciu przycisku Start pojawia się ekran z innym poziomem (Rys. 5), a użytkownik może wybrać dowolny etap, który chce, a następnie jest generowany świat gry.



Rys. 5

Rzeczywista rozgrywka składa się z różnych scenariuszy. Na przykład pierwszy poziom jest z nich najłatwiejszy i pomaga zrozumieć mechanikę rządzącą światem gry. Przechodząc dalej, drugi etap (Rys. 6 i 7) wprowadza pierwszego „inteligentnego” przeciwnika (bossa). Ponieważ jest pierwszym tego typu przeciwnikiem, jego zdolności ograniczają się do poruszania się w granicach ekranu, sporadycznego strzelania, a jeśli statkowi kosmicznemu gracza kończą się punkty zdrowia (HP), może go staranować.



Rys. 6



Rys. 7



Z kolei w trzecim (Rys. 8 i Rys. 9) kluczem jest omijanie przeszkód (meteorytów) i pozostanie przy życiu do ostatniej fali.



Rys. 8



Rys. 9

Czwarty przywraca bossów, ale tym razem, przeciwnik (Rys. 10 i 11) ma kilka nowych dodatkowych umiejętności - takich jak strzelanie bomby, większa szybkostrzelność i porusza się w inny sposób.



Rys. 10



Rys. 11



## 6. Wymagania sprzętowe

Przechodząc do wymagań i ograniczeń sprzętowych to ta gra nie wymaga dużych zasobów, jest tylko jeden warunek do spełnienia - urządzenie z systemem Android w wersji 7.0 lub nowszej. Dzieje się tak z powodu ograniczenia minSdkVersion. Aby obsługiwać wszystkie funkcje związane z Javą 8, poziom minSdkVersion API musi być większy lub równy 24. Z uwagi na fakt, że Android 7.0 był pierwszą wersją obsługującą ten konkretny API, jest to jedyne wymaganie do prawidłowego działania gry.

## 7. Ograniczenia

Ponieważ aplikacja internetowa nie została uruchomiona na żadnym serwerze zewnętrznym, skutkuje to brakiem możliwości komunikacji z tą aplikacją w innej sieci niż lokalnej. Kolejnym ograniczeniem jest brak możliwości sprawdzenia wyników od więcej niż dziesięciu najlepszych graczy (w grze).