



Projekt Gry Strategicznej Statki

Dokumentacja Inżynierii Wymagań

Jan Ściga
Mateusz Zdyński
Adrian Chmielowiec

AKADEMIA GÓRNICZO-HUTNICZA,
WYDZIAŁ INFORMATYKI, ELEKTRONIKI I TELEKOMUNIKACJI

4 Listopad 2021

Spis treści

1	Wprowadzenie	1
1.1	Przedmowa	1
1.2	Zasady gry	1
1.3	Cel dokumentu	1
1.4	Macierz kompetencji zespołu	1
2	Słownik pojęć technicznych	2
3	Definicja wymagań użytkownika	2
4	Architektura systemu (high-level)	4
4.1	Gracz vs Gracz	4
4.2	Gracz vs Komputer	4
4.3	Komputer vs Komputer	4
5	Specyfikacja wymagań systemu (Technical)	5
6	Model Systemu	6
6.1	Model systemu w postaci tabeli	6
6.2	Model systemu w postaci diagramu sekwencyjnego	8
6.3	Model systemu w postaci diagramu przypadków użycia	8
7	Podsumowanie	9
7.1	Sugerowany język implementacji	9

1 Wprowadzenie

1.1 Przedmowa

Okręty (*ang. Battleship*) to gra strategiczna przeznaczona dla dwóch osób. Została wynaleziona przez Clifforda Von Vicklera na początku XX wieku, a opatentowana przez Milton Bradley Company w 1943 roku. Gra jest znana w wielu krajach pod nazwą: gra w Statki lub Salvo.

1.2 Zasady gry

Gra toczy się na specjalnej planszy, każdy z graczy posiada po dwie plansze o wielkości 10 x 10 pól. Pola oznaczane są przez specjalne współrzędne, dzięki którym lokalizowane są statki na planszy gry. Okręty są różnej wielkości. Trafienie okrętu przeciwnika polega na odgadnięciu położenia jakiegoś statku. Strzały oddawane są naprzemiennie, poprzez podanie współrzędnych pola. W przypadku strzału trafionego, gracz kontynuuje strzelanie (czyli swój ruch) aż do momentu chybienia. Wygrywa ten, kto pierwszy zatopi wszystkie statki przeciwnika.

1.3 Cel dokumentu

Podstawową intencją powstania dokumentu jest umożliwienie spełnienia wymagań klienta w zakresie dostarczenia produktu - gry strategicznej „Statki”. W tym celu zespół tworzący niniejszy dokument wyróżnia następujące cele dokumentu adresowane do wszystkich osób związanych z realizacją tego projektu:

- Przedstawienie wymagań klienta
- Przedstawienie wymagań do planowania procesu tworzenia oprogramowania
- Oszacowanie dostępnych zasobów
- Zrozumienia co projektowany system powinien robić
- Przedstawienie punktu wyjściowego do walidacji oprogramowania
- Przedstawienie zależności pomiędzy poszczególnymi komponentami systemu

1.4 Macierz kompetencji zespołu

Zespół projektujący dokumentację inżynierii wymagań przedstawia macierz kompetencji członków zespołu.

Kompetencje	Jan Ściga	Mateusz Zdyrski	Adrian Chmielowiec
Znajomość UML	Posiada (podstawy)	Nie posiada	Nie posiada
Znajomość Python’a	Posiada (podstawy)	Posiada	Nie posiada
Testowanie oprogramowania	Nie posiada	Nie posiada	Nie posiada
Znajomość Javy	Posiada	Posiada	Posiada
Znajomość C++	Posiada (podstawy)	Posiada(podstawy)	Posiada(podstawy)

2 Słownik pojęć technicznych

- **AI** - algorytm decyzyjny komputera podczas gry
- **Poziomy trudności** - różne stopnie zaawansowania AI
- **Hot seat** - tryb gry wieloosobowej w turowych grach komputerowych, pozwalający grać kilku osobom na jednym komputerze. W trybie hot-seat gra polega na tym, że gracze kolejno wykonują określone czynności w grze i po skończonej turze przychodzi kolej na następnego gracza.
- **Algorytm rozgrywki** - odpowiednio wykonany algorytm umożliwiający rozgrywkę
- **Autoustawienie** - opcja automatycznego, losowego ustawienia statków
- **Replay** - możliwość odtworzenia rozgrywki
- **GUI** - Graficzny interfejs użytkownika
- **Avatar** - niewielka grafika reprezentująca uczestnika świata wirtualnego
- **Wielopatformowość** - cecha aplikacji polegająca na pracy na różnych platformach sprzętowych

3 Definicja wymagań użytkownika

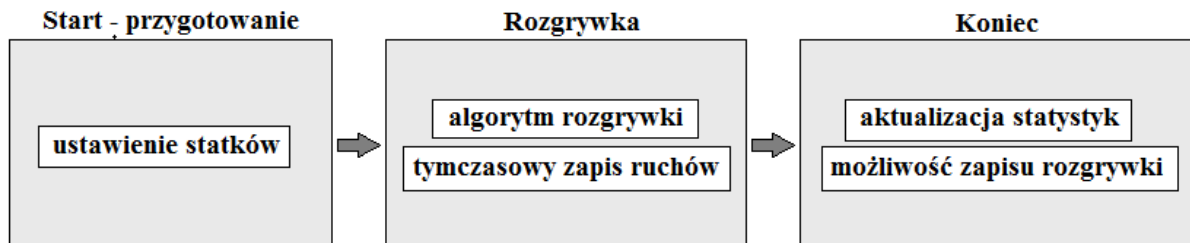
Pytanie	Odpowiedź	Uwagi
Czy gra powinna być typu Desktop czy WEB?	Dowolna forma.	
Czy wersja Desktopowa powinna działać na Linux czy Windowsie?	Dowolnie na obu lub jednym wybranym.	
Czy Webowa wersja gry powinna działać na każdej przeglądarce?	Powinna działać na przynajmniej 2 z 3 wymienionych przeglądarek: <ul style="list-style-type: none">• Chrome,• Firefox,• Safari.	
Jaką technologię powinniśmy wykorzystać przy tworzeniu tej gry?	Proszę wybrać dowolnie.	
Jakie rodzaje rozgrywek ma umożliwić ta gra?	Rozgrywkę typu Gracz vs Gracz, Komputer vs Gracz, Komputer vs Komputer.	
Czy gra powinna robić statystyki?	Tak, dla każdego poziomu oraz dla danego zawodnika.	Dla zawodnika odpowiednio: <ul style="list-style-type: none">• ile razy grał,• ile razy wygrał,• celność (wygrane/wszystkie).
Czy gra powinna posiadać menu?	Tak powinna posiadać menu z wyborem typów rozgrywek oraz statystyki.	

Pytanie	Odpowiedź	Uwagi
Czy gra powinna być w języku polskim czy angielskim?	Dowolnie.	
Czy gra powinna posiadać GUI?	Tak oraz powinna potrafić obsługiwać myszkę.	
Czy dany gracz powinien mieć możliwości logowania?	Tak, powinien mieć możliwości dodania avatara, zmiany nazwy oraz hasła.	Powinna się pojawić baza wszystkich graczy.
Czy gra powinna posiadać poziomy trudności?	Tak, 3 poziomy: <ul style="list-style-type: none"> • łatwy, • średni, • trudny. 	
Czy plansza powinna mieć określony rozmiar?	Tak 10 X 10.	
Czy statki można definiować, czy są z góry zdefiniowane?	Statki są zdefiniowane, można ustalać tylko ich pozycje.	Rodzaje statków: <ul style="list-style-type: none"> • 1 x Statek 4 masztowy, • 2 x Statek 3 masztowy, • 3 x Statek 2 masztowy, • 4 x Statek 1 masztowy.
Czy gra powinna posiadać opcję ustawiania statków?	Tak, ale tylko opcjonalnie.	
Czy w przypadku rozgrywki Komputer vs Komputer też wybieramy poziomy trudności?	Tak.	
Czy w przypadku rozgrywki Komputer vs Komputer mamy możliwość replay?	Tak, każdy ruch powinien trwać jedną sekundę i powinna zostać stworzona animacja tej rozgrywki, którą można dowolnie odtwarzać.	
Czy pozostałe typy rozgrywek też powinny mieć opcje odtwarzania rozgrywki (replay)?	Tak.	

4 Architektura systemu (high-level)

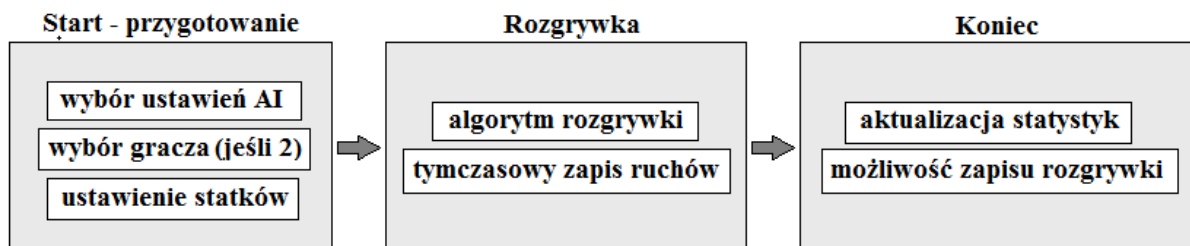
Projekt obejmuje 3 tryby rozgrywki:

4.1 Gracz vs Gracz



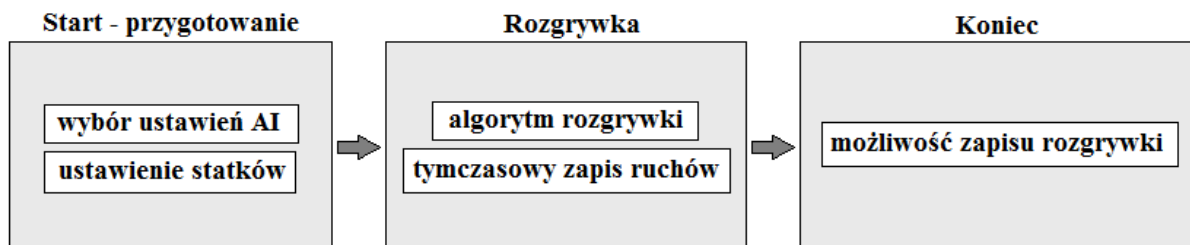
Aby uruchomić ten tryb wymagane jest dwóch zalogowanych użytkowników. Rozgrywka dwuosobowa w postaci Hot Seat. Naprzemiennie są wyświetlane mapy dla odpowiedniego gracza. Na początku należy ustawić własne statki na mapie. W trakcie rozgrywki ruchy graczy są tymczasowo zapisywane. Po wygranej następuje podliczenie odpowiednich wartości i na ich podstawie aktualizacja statystyk. Wymagana opcja Replay do obejrzenia rozgrywki na podstawie tymczasowo zapisanych ruchów oraz opcja zapisu rozgrywki w Zapisanych rozgrywkach.

4.2 Gracz vs Komputer



W fazie przygotowawczej należy wybrać odpowiedni poziom trudności AI, gracza(jeśli jest dwóch zalogowanych) oraz ustawić statki(losowe dla AI). Dalej przebieg jest taki jak w trybie Gracz vs Gracz.

4.3 Komputer vs Komputer



W tym trybie użytkownik jest obserwatorem rozgrywki. W fazie przygotowawczej należy wybrać odpowiedni poziom trudności AI, oraz ustawić statki(losowe dla AI). W trakcie rozgrywki ruchy są tymczasowo zapisywane, odstępy pomiędzy ruchami powinny wynosić 1 sekundę. Wymagana opcja Replay do obejrzenia rozgrywki na podstawie tymczasowo zapisanych ruchów oraz opcja zapisu rozgrywki w Zapisanych rozgrywkach.

Oraz zawiera dodatkowe opcje:

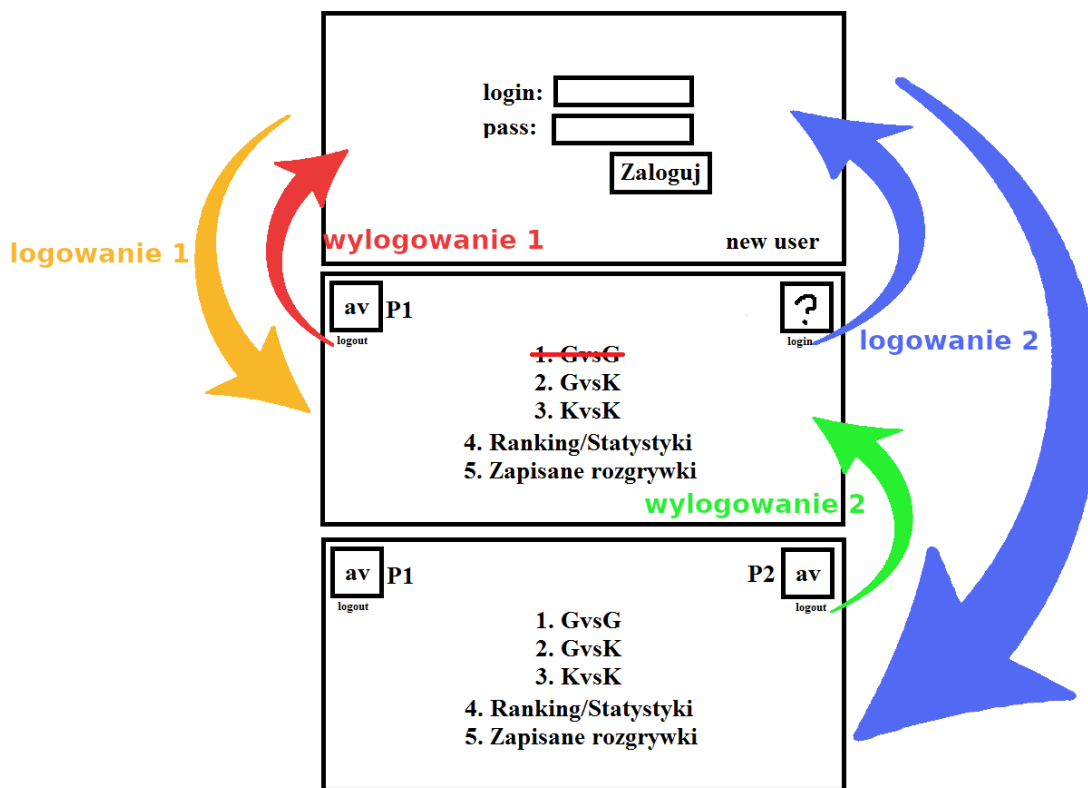
- **Rankingi/Statystyki**

Zawierają odpowiednie informacje na temat danego gracza. Kolejne zapytania do bazy w celu pobrania danych za pomocą przycisku Odśwież lub po każdym wejściu w tą opcję.

- **Zapisane rozgrywki**

Zawierają listę możliwych do odtworzenia(obejrzenia w sposób taki jak w trybie Komputer vs Komputer) rozgrywek, które zostały wcześniej zapisane. Po wybraniu odpowiedniej rozgrywki powinno nastąpić odtworzenie gry na podstawie ruchów zapisywanych podczas rozgrywek.

Przykładowe działanie logowanie i zmiany wyświetlanej zawartości.



5 Specyfikacja wymagań systemu (Technical)

- System powinien zawierać bazę danych do przechowywania danych o użytkownikach, ich statystykach oraz rozegranych rozgrywek.
- Po uruchomieniu, system powinien sprawdzać czy baza danych istnieje, jeśli nie to ją utworzyć.
- System powinien zawierać opcję logowania poprzez login i hasło.
- Gracz powinien mieć możliwość zmiany avataru oraz hasła.
- System powinien móc zalogować do dwóch użytkowników.
- Aplikacja powinna zawierać GUI oraz obsługiwać myszkę.
- Gracz powinien posiadać opcję automatycznego ustawiania statków.

- W trybach, w których występuje AI powinna być opcja wyboru z trzech poziomów trudności.
- W trybie AI vs AI ruch odbywa się co sekundę.
- W trakcie trwania rozgrywek, ruchy powinny być tymczasowo zapisywane.
- Po rozgrywce, system powinien zliczyć odpowiednie dane i na ich podstawie zaktualizować statystyki.
- Statystyki prowadzone osobno dla każdego trybu gry: Gracz vs Gracz, oraz 3 Gracz vs AI(osobno dla różnych poziomów trudności).
- Po rozgrywce, system powinna być opcja 'Replay' do obejrzenia rozgrywki na podstawie tymczasowo zapisanych ruchów oraz opcja 'Zapisz rozgrywkę', która zapisze te ruchy do bazy danych i przypisze do zalogowanych kont(lub do wybranego - opcja wyboru).
- System powinien zawierać opcję 'Zapisane rozgrywki', w której będą się znajdować zapisane wcześniej rozgrywki, które po wybraniu można obejrzeć.

6 Model Systemu

6.1 Model systemu w postaci tabeli

Poniższa tabela przedstawia zbiór elementów wraz z opisami oraz wymaganiami, które ich dotyczą. Elementy te stanowią części składowe modelowanego systemu gry w Statki.

Tablica 3: Modelowanie systemu w postaci tabeli

Modelowany element	Opis
Aktorzy	Komputer, Gracze
Opis	Celem działania modelowanego systemu jest umożliwienie rozegrania gry w statki zarówno w trybie Gracz vs Gracz, Komputer vs Komputer jak i Gracz vs Komputer, przechowywania rankingu oraz podstawowych statystyk dotyczących rozegranych gier.
Dane	Dane wprowadzane są do komputera za pomocą myszki (dot. Trybu Gracz vs Gracz oraz Gracz vs Komputer). Komputer wprowadza swoje dane za pomocą ustalonych algorytmów.
Statystyki	Statystyki dotyczą rozegranych w przeszłości gier danego gracza i obejmują następujące wielkości: <ul style="list-style-type: none"> • Ilość rozegranych gier • Ilość wygranych gier • Celność (rozumiana jako stosunek ilości trafień do ilości strzałów ogólnie) • Powyższe statystyki powinny być widoczne z rozbiciem na trzy poziomy trudności. • Powyższe statystyki powinny być widoczne również dla gier Gracz vs Gracz

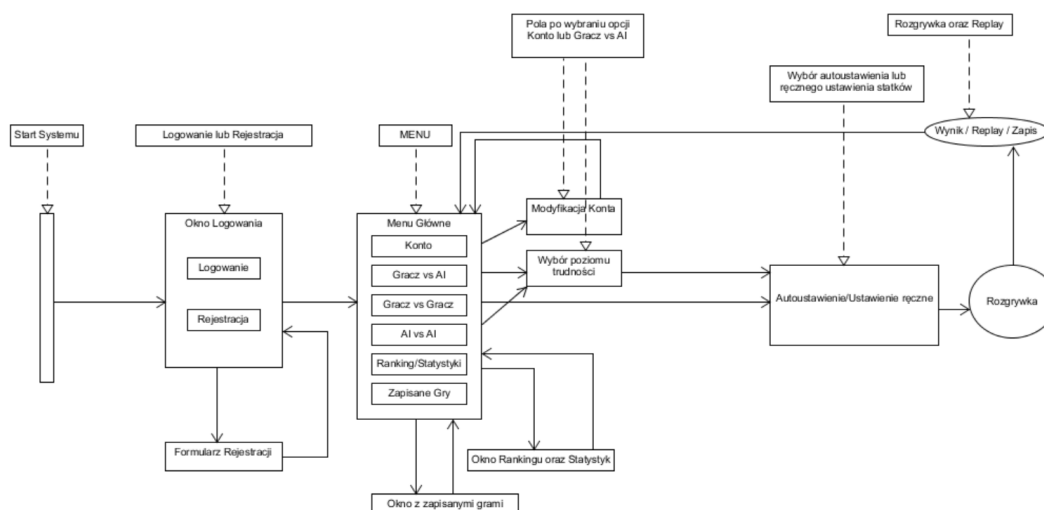
Tablica 3: Modelowanie systemu w postaci tabeli

Modelowany element	Opis
Plansza	<p>W momencie rozgrywki każdy z Graczy powinien widzieć dwie plansze:</p> <ul style="list-style-type: none"> • Plansza gdzie zaznaczane będą strzały oddane przez gracza • Plansza gdzie widoczne będą statki gracza wraz z ewentualnymi trafieniami zadanyymi przez przeciwnika <p>Plansza powinna mieć standardowe wymiary (10x10)</p>
Widoki	W przypadku rozgrywki Gracz vs Gracz, każdy z nich powinien dysponować osobnym oknem gdzie znajdowałyby się jego plansze.
Statki	<p>System powinien umożliwiać zarówno samodzielne jak i automatyczne ustawianie statków na planszy. Rodzaje dostępnych statków:</p> <ul style="list-style-type: none"> • 1 x Statek 4 masztowy • 2 x Statek 3 masztowy • 3 x Statek 2 masztowy • 4 x Statek 1 masztowy
Wieloplatformowość	<p>W przypadku realizacji aplikacji Webowej system powinien działać na przynajmniej dwóch z trzech wymienionych przeglądarek.</p> <ul style="list-style-type: none"> • CHROME • SAFARI • FIREFOX <p>W przypadku aplikacji desktopowej system powinien działać na systemach Windows oraz Linux.</p>
Replay	System powinien umożliwiać odtworzenie dowolnej z zapisanych i rozegranych gier.
Hasło	<p>System powinien umożliwiać logowanie gracza za pomocą nazwy oraz hasła z następującymi warunkami</p> <ul style="list-style-type: none"> • Możliwość aktualizacji hasła • Hasło powinno zawierać co najmniej 8 znaków w tym co najmniej jedną cyfrę, znak specjalny oraz dużą literę
Konto	Każdy gracz powinien posiadać nazwę oraz avatar, który będzie mógł zaktualizować. System powinien umożliwiać założenie nowego konta.
Gra Komputera	Każdy ruch komputera powinien odbywać się co jedną sekundę
Poziomy trudności	W przypadku rozgrywki z komputerem, gracz powinien mieć możliwość wyboru jednego z trzech poziomu trudności (ŁATWY, ŚREDNI, TRUDNY)
Sygnalizacja trafienia	Sygnalizacja trafienia powinna być przekazywana przez krótkie wyświetlenie komunikatu np. czerwonego napisu „TRAFIENIE !”
Wynik	Wynik (tzn. nazwa zwycięzcy powinien być widoczny po zatopieniu statków przeciwnika wraz z możliwością powrotu do menu głównego, odtworzenia zakończonej gry i zapisania gry)

6.2 Model systemu w postaci diagramu sekwencyjnego

Poniższy diagram w sposób ogólny przedstawia zarys modelowanego systemu, dostępne okna dla użytkownika oraz sekwencje działań które może wykonać w systemie.

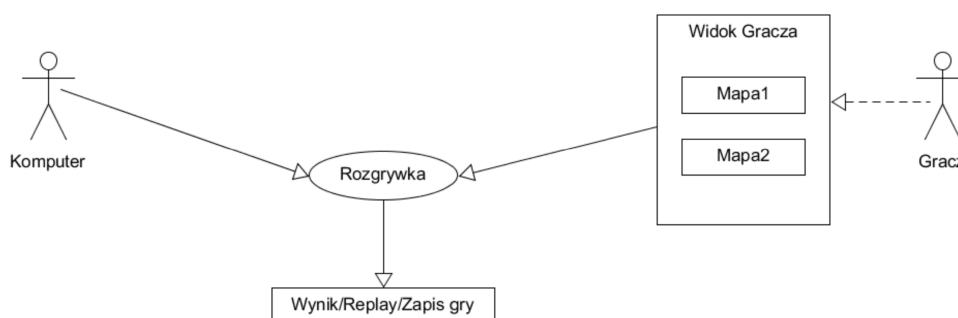
Rysunek 1: Diagram sekwencyjny modelowanego systemu



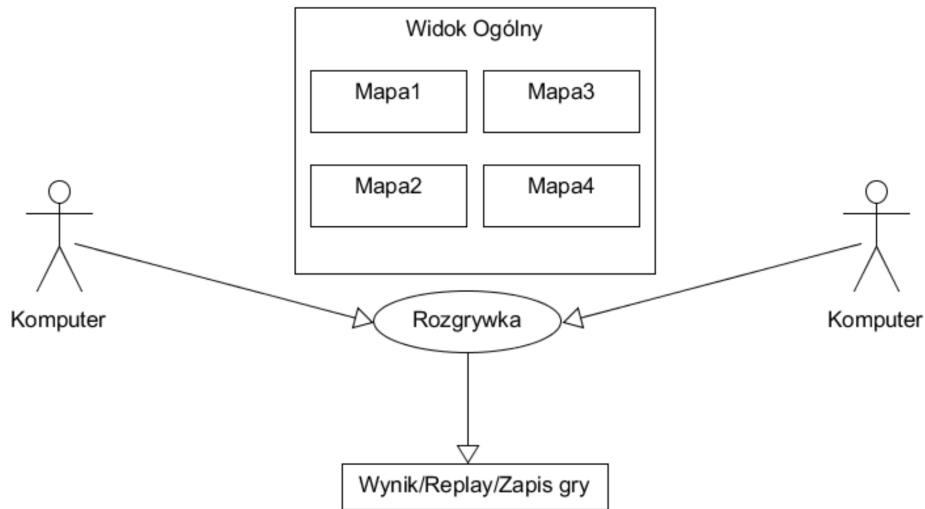
6.3 Model systemu w postaci diagramu przypadków użycia

Poniższe diagramy pozwalają przedstawić różne rodzaje gier oraz widoki, które powinny być dostępne dla grających stron.

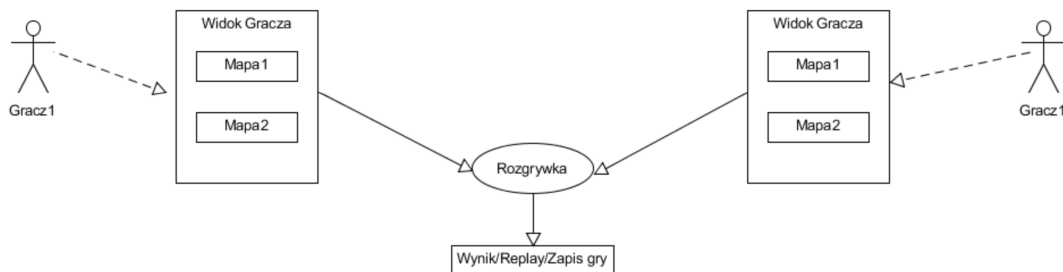
Rysunek 2: Diagram przypadku dla gry Gracz vs Komputer



Rysunek 3: Diagram przypadku dla gry Komputer vs Komputer



Rysunek 4: Diagram przypadku dla gry Gracz vs Gracz



7 Podsumowanie

7.1 Sugerowany język implementacji

Sugerowanym językiem implementacji jest Python

Python jest jednym z najprostszych a zarazem jednym z najbardziej wszechstronnych języków programowania, nie należy jednak błędnie interpretować jego pozornie łatwej składni i semantyki jako nieodpowiedniego do programowania gier. Python oferuje programistom w pełni rozwinięte techniki OOP (programowanie zorientowane obiektowo), podobnie jak C++ i Java.

Framework Pygame, oparty na tym popularnym języku, nabiera tempa każdego dnia, dzięki możliwości umożliwienia programistom szybszego prototypowania swoich gier. Python zyskuje sławę jako jeden z najlepszych języków programowania gier wideo.