|  |
| --- |
| Wydział Fizyki i Informatyki Stosowanej |
| **PAC-MEN** |
| Dokumentacja analityczna systemu |
|  |
| Mateusz Ciechan – Product owner  Kamil Oleszek – Scrum master  Łukasz Ujma - Architekt  Aleksander Giera – Architekt  Alicja Frankowicz – Programista/Grafik  Grzegorz Ryniak – Programista/Grafik  Karol Cwynar - Programista  Elżbieta Dziedzic - Tester  Michał Niemczyk - Tester |
| 14.10.2016 |

|  |
| --- |
| [Wpisz tutaj streszczenie dokumentu. Streszczenie jest zazwyczaj krótkim podsumowaniem treści dokumentu. Wpisz tutaj streszczenie dokumentu. Streszczenie jest zazwyczaj krótkim podsumowaniem treści dokumentu.] |

[1 Wprowadzenie 3](#_Toc259470600)

[1.1 Opis systemu 3](#_Toc259470601)

[1.2 Słownik 3](#_Toc259470603)

[2 Wymagania funkcjonalne - przypadki użycia 3](#_Toc259470604)

[2.1 PAC-MEN 3](#_Toc259470605)

[3 Wymagania niefunkcjonalne 3](#_Toc259470606)

[3.1 Wymagania względem projektu 3](#_Toc259470607)

[3.2 Reguły biznesowe 3](#_Toc259470608)

[3.3 Ograniczenia implementacji 3](#_Toc259470609)

[3.4 Dostępność, wydajność, niezawodność 3](#_Toc259470610)

[3.5 Dokumentacja 3](#_Toc259470612)

[3.6 Utrzymanie systemu 3](#_Toc259470613)

[4 Model dziedziny 3](#_Toc259470614)

[5 Modele stanów obiektów dziedziny 3](#_Toc259470615)

[6 Problemy i ryzyka 3](#_Toc259470616)

# Wprowadzenie

## Opis systemu

Celem projektu jest stworzenie multiplayerowej gry opartej na popularnej grze Pac-Man. Zasady rozgrywki w dużym stopniu są takie same jak w oryginalnej grze Pac-Man, dodatkowym elementem jest możliwość lokalnej gry dla dwóch graczy. Celem gracza jest uzyskanie większej ilości punktów od przeciwnika przed zakończeniem gry sterując Pac-Manem poprzez zbieranie punktów umieszczonych na mapie, bądź zjadanie duchów po zebraniu boosta. Zakończenie gry następuje gdy wszystkie punkty na mapie zostaną zebrane, któryś z graczy zginie od dotknięcia ducha lub gracze się ze sobą zderzą.

## Słownik

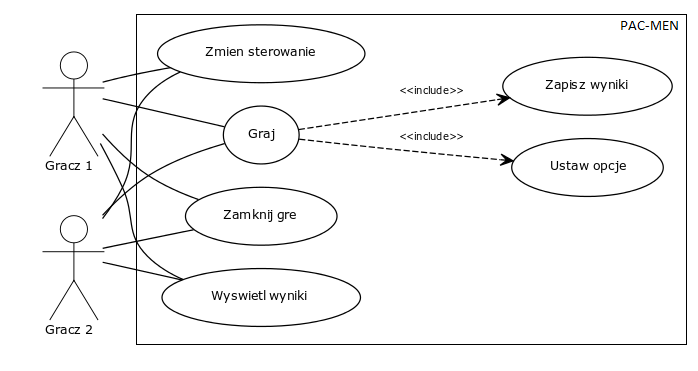
Pac-Man – postać którą steruje gracz

Multiplayer – opcja gry w kilka osób

Boost – specjalny punkt na mapie po którego zdobyciu przez krótki czas Pac-Man gracza ma możliwość zjedzenia ducha bez stracenia życia.

# Wymagania funkcjonalne - przypadki użycia

## PAC-MEN



### Opis przypadku użycia

Główna logika użycia aplikacji. Dowolny gracz ma możliwość zmiany sterowania Pac-Manami, wyświetlenia poprzednich wyników, wyjścia, bądź rozpoczęcia rozgrywki.

### Aktorzy

Gracz 1 – użytkownik odpowiedzialny za sterowanie Pac-Manem nr 1.

Gracz 2 – użytkownik odpowiedzialny za sterowanie Pac-Manem nr 2.

### Poziom

Użytkownika

### Cele użytkowników

Rozpoczęcie i zakończenie rozgrywki w PAC-MENa dla dwóch graczy.

### Wyzwalacz

Przypadek użycia jest uruchamiany na żądanie użytkownika.

### Warunki końcowe

Rozgrywka zakończyła się i został wyłoniony zwycięzca lub wybrano opcję „Zamknij grę”.

### Scenariusz główny

1. Gracz 1 (2) wybiera opcję uruchomienia gry
2. Gracz 1 (2) wybiera ustawienia rozgrywki
3. Obaj gracze rozpoczynają rozgrywkę

### Scenariusze alternatywne

Przed krokiem 1) użytkownicy mają możliwość wybrania opcji „Wyświetl wyniki” lub „Zmień sterowanie”.

### Decyzje

Będąc w kroku 2) użytkownik musi mieć możliwość cofnięcia decyzji uruchamiania gry.

# Wymagania niefunkcjonalne

## Wymagania względem projektu

### Czas realizacji

Końcowym czasem realizacji projektu jest termin ostatnich zajęć zimowego semestru AGH w roku 2016/17, czyli 27.01.2017r.

## Reguły biznesowe

### Odpłatność

System jest w pełni darmowy i dozwolona jest jego replikacja jak i rozprzestrzenianie.

## Ograniczenia implementacji

### Język

System jest napisany z wykorzystaniem języka programowania Java i wbudowanych w nim bibliotek.

## Dostępność, wydajność, niezawodność

### Łatwość użycia

Możliwość wykorzystania wszystkich funkcjonalności systemu bez potrzeby dodatkowego szkolenia.

### Niezawodność

Brak jakichkolwiek błędów występujących w trakcie działania systemu.

### Szybkość działania

Brak zauważalnych opóźnień w działaniu systemu. Czas reakcji <100ms.

## Dokumentacja

Jakie są wytyczne dla dokumentacji użytkownika, dokumentacji projektowej oraz kodu?

???

## Utrzymanie systemu

### Utrzymanie po finalizacji projektu

Po zakończeniu budowy i oddaniu systemu do użytku nie będzie on dalej utrzymywany i rozwijany.

# Model dziedziny

[Diagram]

[Opis odpowiedzialności klas obiektów]

# Modele stanów obiektów dziedziny

Opcjonalne. Dla systemów, których zachowanie zależy od stanu obiektów.

# Problemy i ryzyka

Ryzyka są to sytuacje wpływające na projekt, które mogą zdarzyć się z prawdopodobieństwem > 0 i < 1.