|  |
| --- |
| 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 |

[2 Wprowadzenie 3](#_Toc472028134)

[2.1 Opis systemu 3](#_Toc472028135)

[2.2 Słownik 3](#_Toc472028136)

[3 Wymagania funkcjonalne - przypadki użycia 3](#_Toc472028137)

[3.1 PAC-MEN 3](#_Toc472028138)

[3.1.1 Opis przypadku użycia 3](#_Toc472028139)

[3.1.2 Aktorzy 3](#_Toc472028140)

[3.1.3 Poziom 3](#_Toc472028141)

[3.1.4 Cele użytkowników 3](#_Toc472028142)

[3.1.5 Wyzwalacz 4](#_Toc472028143)

[3.1.6 Warunki końcowe 4](#_Toc472028144)

[3.1.7 Scenariusze przypadków użycia 4](#_Toc472028145)

[3.1.8 Decyzje 5](#_Toc472028146)

[4 Wymagania niefunkcjonalne 6](#_Toc472028147)

[4.1 Wymagania względem projektu 6](#_Toc472028148)

[4.1.1 Czas realizacji 6](#_Toc472028149)

[4.2 Reguły biznesowe 6](#_Toc472028150)

[4.2.1 Odpłatność 6](#_Toc472028151)

[4.3 Ograniczenia implementacji 6](#_Toc472028152)

[4.3.1 Język 6](#_Toc472028153)

[4.4 Dostępność, wydajność, niezawodność 6](#_Toc472028154)

[4.4.1 Łatwość użycia 6](#_Toc472028155)

[4.4.2 Niezawodność 6](#_Toc472028156)

[4.4.3 Szybkość działania 6](#_Toc472028157)

[4.5 Dokumentacja 6](#_Toc472028158)

[4.6 Utrzymanie systemu 6](#_Toc472028159)

[4.6.1 Utrzymanie po finalizacji projektu 6](#_Toc472028160)

[5 Model dziedziny 6](#_Toc472028161)

[6 Modele stanów obiektów dziedziny 7](#_Toc472028162)

[7 Problemy i ryzyka 7](#_Toc472028163)

# 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

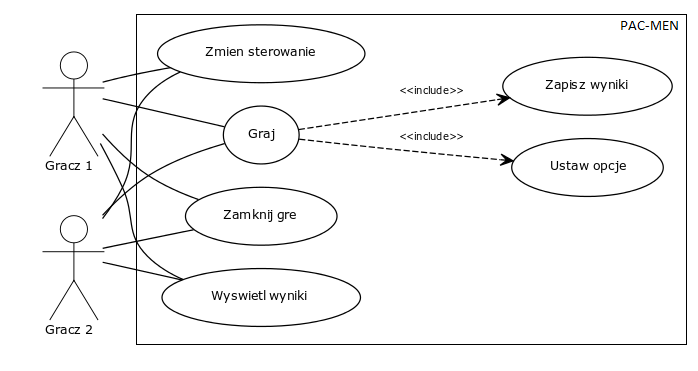
Pac-Man – postać którą steruje gracz

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”.

### Scenariusze przypadków użycia

**S1. Rozpoczęcie gry**

1.1 Opis:

Rozpoczęcie nowej gry poprzez wybranie odpowiedniej opcji z menu głównego.

1.2. Warunki początkowe:

Na ekranie wyświetla się menu główne.

1.3. Warunki końcowe:

Na ekranie zostaje wyświetlona plansza i rozpoczyna się przebieg gry.

1.4. Przebieg główny:

1. Użytkownik uruchamia grę.
2. System wyświetla menu główne.
3. Użytkownik wybiera opcję „Nowa gra”.
4. Rozpoczęcie rozgrywki. Gracze sterują Pac-menem za pomocą klawiszy:

|  |  |  |
| --- | --- | --- |
| **Kierunek** | **Gracz 1** | **Gracz 2** |
| Góra | W | I |
| Dół | S | K |
| Lewo | A | J |
| Prawo | D | L |

1.5. Przebieg alternatywny:

1.5.1 Zmiana ustawień klawiszy przed rozpoczęciem rozgrywki:

1. Użytkownik w menu głównym wybiera przycisk „Opcje”.

2. System wyświetla opcje gry.

3. Użytkownik zmienia ustawienia klawiszy sterowania gracza 1 lub gracza 2.

4. Użytkownik zapisuje wprowadzone zmiany.

5. Użytkownik wybiera opcję „Nowa gra”.

6. Rozpoczęcie rozgrywki.

**S2. Zakończenie gry - zwycięstwo**

2.1 Opis:

Zakończenie rozgrywki z powodu „zjedzenia” wszystkich punktów na planszy.

2.2 Warunki początkowe:

Trwa przebieg gry.

2.3 Warunki końcowe:

Wyświetlenie menu głównego.

2.4 Przebieg główny:

1. W czasie trwania rozgrywki zostaje zebrany ostatni punkt na planszy.

2. Zostaje wyświetlony komunikat z informacją o zwycięstwie gracza który zdobył więcej punktów.

3. Użytkownik zamyka komunikat.

4. Zostaje wyświetlone menu główne.

**S3. Zakończenie gry - przegrana**

3.1 Opis:

Zakończenie rozgrywki z powodu „złapania” przez ducha lub dotknięcia drugiego Pac-Mana.

3.2 Warunki początkowe:

Trwa przebieg gry.

3.3 Warunki końcowe:

Wyświetlenie menu głównego.

3.4 Przebieg główny:

1. W czasie trwania rozgrywki następuje kolizja Pac-Mana z duchem lub z drugim Pac-Manem.

2. Zostaje wyświetlony komunika z informacją o przegranej.

3. Użytkownik zamyka komunikat.

4. Zostaje wyświetlone menu główne.

**S4. Przeglądanie wyników**

4.1 Opis:

Umożliwienie przeglądania zapisanych wyników rozgrywek.

4.2 Warunki początkowe i końcowe:

Wyświetlenie menu głównego.

4.3 Przebieg główny:

1. Użytkownik wybiera z menu opcję „Wyniki”.

2. Wyświetlone zostają liczby zdobytych punktów z zapisanych rozgrywek.

3. Użytkownik powraca do menu.

# 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.