

Autorzy projektu: Krzysztof Sobieski, Maciej Surawski, Piotr Ołtarzewski,
Sebastian Majewski, Edyta Zakrzewska

1. Wymagania funkcjonalne

```

graph TD
    Gracz((Gracz)) --- Wyjdz((Wyjdz))
    Gracz --- Wyświetlenie_menu((Wyświetlenie menu))
    Wyjdz --- Wyświetlenie_menu
    Wyświetlenie_menu --- Opcje((Opcje))
    Opcje -.->|«include»| Ustawienia_dźwięku((Ustawienia dźwięku))
    Opcje -.->|«include»| Ustawienia_grafiki((Ustawienia grafiki))
    Ustawienia_dźwięku -.->|«extend»| Zmiana_wartości1((Zmiana wartości))
    Ustawienia_grafiki -.->|«extend»| Zmiana_wartości2((Zmiana wartości))
    Opcje -.->|«extend»| Powrót_do_menu((Powrót do menu))
    Powrót_do_menu -.->|«extend»| Ustawienia_grafiki
    Wyświetlenie_menu --- Gra((Gra))
    Gra --- Wyświetlenie_miasta((Wyświetlenie miasta))
    Wyświetlenie_miasta --- Czarny_rynek((Czarny rynek))
    Czarny_rynek -.->|«include»| Kup((Kup))
    Kup -.->|«extend»| Brak_miejsca((Brak miejsca))
    Kup -.->|«extend»| Brak_srodkow((Brak środków))
    Wyświetlenie_miasta --- Sklep((Sklep))
    Sklep -.->|«include»| Sprzedaj((Sprzedaj))
    Sprzedaj -.->|«include»| Dolicz_walute((Dolicz walutę))
    Wyświetlenie_miasta --- Loch((Loch))
    Loch -.->|«include»| Wyświetlenie_pokoju((Wyświetlenie pokoju))
    Wyświetlenie_pokoju -.->|«include»| Walka_z_przeciwnikiem((Walka z przeciwnikiem))
    Walka_z_przeciwnikiem -.->|«include»| Walka_z_bosem((Walka z bosem))
    Walka_z_bosem -.->|«include»| Przejscie_do_nastepnego_pokoju((Przejscie do następnego pokoju))
    Przejscie_do_nastepnego_pokoju -.->|«include»| Walka_z_bosem
    Przejscie_do_nastepnego_pokoju -.->|«include»| Przejscie_lochu((Przejscie lochu))
    Walka_z_przeciwnikiem -.->|«include»| Smierc((Śmierć))
    Smierc -.->|«include»| Komunikat_konca_gry((Komunikat końca gry))
    Smierc -.->|«include»| Przejscie_lochu
    Wyświetlenie_miasta --- Postać((Postać))
    Postać -.->|«extend»| Zarządzanie((Zarządzanie))
    Postać -.->|«extend»| Ulepszenie((Ulepszenie))
    Postać -.->|«extend»| Ekwipunek((Ekwipunek))
    Ekwipunek -.->|«extend»| Walka_z_przeciwnikiem
    Wyświetlenie_miasta --- Spłacenie_długu((Spłacenie długu))
    Spłacenie_długu -.->|«extend»| Zakonczenie_gry((Zakończenie gry))
    Zakonczenie_gry -.->|«extend»| Napisy_koncowe((Napisy końcowe))
    Napisy_koncowe -.->|«extend»| Pokonanie((Pokonanie))
    Pokonanie -.->|«include»| Walka_z_przeciwnikiem
    Pokonanie -.->|«include»| Przejscie_do_nastepnego_pokoju
    Pokonanie -.->|«include»| Walka_z_bosem
    Pokonanie -.->|«include»| Przejscie_lochu
    Nagrody((Nagrody)) -.->|«include»| Pokonanie
  
```

- **Wymagania funkcjonalne gry przedstawione w postaci przypadków użycia.**

Tabela1. Przypadki użycia dla Gracza.

Nazwa	Opis	Odpowiedź systemu
Wyświetlanie menu	Użytkownik widzi menu z którego może wybrać następujące opcje : graj , opcje , wyjdź	Wyświetlenie odpowiedniej sceny oraz reakcja na wybrane opcje. W zależności od wybranej opcji następuje przejście do niej.
Przycisk Graj	Użytkownik klika graj dzięki czemu rozpoczyna nową grę	Przejście z menu do „lobby” czyli miasta . Załadowanie dodatkowych obiektów do interakcji.
Przycisk Wyjdź	Użytkownik klika wyjdź przez co zamyka grę.	Zakończenie działania gry.
Przycisk Opcje	Wyświetlenie dodatkowych opcji służących do zmiany dźwięku oraz grafiki	Przejście do nowej sceny . Załadowanie opcji które umożliwią zmianę głośności oraz grafiki
Loch	Użytkownik klika na loch. Wyświetlony zostaje pokój gdzie będzie walczyć z przeciwnikiem. Po pokonaniu przeciwnika gracz przenosi się do kolejnego. Po ukończeniu wszystkich gracz przenosi się do miasta	Przejście do nowej sceny. Załadowanie i wyświetlenie przeciwników oraz innych obiektów. Odblokowanie kolejnych interakcji. Powtarzanie wyświetlania kolejnych pokoi aż do ostatniego. Po przejściu ostatniego powrót do lobby.
Walka z przeciwnikiem	Użytkownik walczy z przeciwnikiem. Oboje mają swoje statystyki . Mogą zadawać sobie obrażenia . Gracz używa kart do ranienia przeciwnika. Wygrania umożliwia dalsze zwiedzanie lochu , śmierć kończy grę.	Liczenie zadanych obrażeń oraz monitorowanie zdrowia. Sprawdzanie warunku zwycięstwa lub porażki. Wyświetlenie odpowiedniego komunikatu na koniec walki.
Nagrody	Gracz podnosi walutę , przedmioty , karty.	Przedmioty zostają przypisane do ekwipunku gracza.

	Przeniesienie ich do ekwipunku.	Wyświetlenie odpowiednich ikon.
Sklep	Gracz odwiedza sklep gdzie może sprzedać lub kupić przedmioty , karty. W przypadku sprzedaży doliczona zostaje waluta. Przedmiot znika z ekwipunku. W przypadku kupna zostaje odliczona waluta. Przedmiot pojawia się w ekwipunku.	Przejdźcie do nowej sceny . W przypadku kupna przypisanie nowych przedmiotów kart do ekwipunku gracza. Odjęcie waluty. Możliwość wyświetlenie komunikatu o braku miejsca. W przypadku sprzedaży usunięcie informacji o przedmiotach , kartach z ekwipunku gracza. Dodanie waluty.
Czarny rynek	Gracz odwiedza czarny gdzie może kupić przedmioty , karty. W przypadku kupna zostaje odliczona waluta. Przedmiot pojawia się w ekwipunku.	Przejdźcie do nowej sceny . W przypadku kupna przypisanie nowych przedmiotów kart do ekwipunku gracza. Odjęcie waluty. Możliwość wyświetlenie komunikatu o braku miejsca.
Postać	Wyświetlone zostaną szczegóły postaci. Możliwość wybrania ulepszenia postaci oraz ekwipunku. Możliwość ulepszania kart potrzebnych do walki.	Pobranie, zmiana i wyświetlenie statystyk. Wyświetlenie przedmiotów w ekwipunku oraz zarządzanie nimi.
Splącanie długu	Gracz spłaca dług w ratach lub za jednym razem. Dług rośnie codziennie. Splącenie długu umożliwia graczowi zakończenie gry.	Zmiana wartości długu w zależności od podjętych działań użytkownika. Osiągnięcie konkretnej wartości umożliwia zakończenie gry. Przejdźcie do kolejnej sceny. Wyświetlenie napisów końcowych.

- **Historia użytkownika**

Poniżej opisane zostały wymagania funkcjonalne w formie user stories.

Gra

1.Jako gracz chcę zmienić głośność gry żeby dostosować głośność gry do swoich potrzeb.

Opis: Przejście do odpowiedniej sceny. Zmiana wartości głośności.

2.Jako gracz chcę zmienić grafikę oraz rozdzielczość gry żeby dostosować je do swojego monitora oraz komputera

Opis: Przejście do odpowiedniej sceny. Zmiana wartości grafiki.

3.Jako gracz chcę rozpocząć nową grę aby zagrać ponieważ się nudzę.

Opis: Przejście do odpowiedniej sceny. Wyświetlenie lobby wraz z jego zawartością. Umożliwienie użytkownikowi dokonywania nowych interakcji.

4.Jako gracz chcę udać się do lochu aby zdobyć walutę oraz przedmioty ponieważ jest to potrzebne do przejścia gry.

Opis: Przejście do odpowiedniej sceny. Wyświetlenie przeciwników oraz obiektów z którymi może dojść do interakcji bądź kolizji.

5.Jako gracz chcę pokonać przeciwnika żeby ukończyć loch i przeżyć.

Opis: Walka z przeciwnikiem. Umożliwienie użycia kart. Monitorowanie zdrowia gracza i potwora . Monitorowanie pozostałych statystyk oraz wartości obrażeń.

6.Jako gracz chcę odwiedzić sklep żeby sprzedać przedmioty.

Opis: Przejście do nowej sceny. Umożliwienie sprzedaży. Doliczenie waluty za każdy sprzedany przedmiot. Usunięcie przedmiotu lub karty z ekwipunku.

7.Jako gracz chcę odwiedzić sklep żeby kupić przedmioty.

Opis: Przejście do nowej sceny. Umożliwienie kupna. Odjęcie waluty za każdy kupiony przedmiotu. Dodanie go do ekwipunku. Jeśli jest on pełny opcja kupna zablokowana.

8.Jako gracz chcę odwiedzić czarny rynek żeby kupić wyjątkowe przedmioty.

Opis: Przejście do nowej sceny. Umożliwienie kupna. Odjęcie waluty za każdy kupiony przedmiot lub kartę. Dodanie go do ekwipunku. Jeśli jest on pełny opcja kupna zablokowana.

9.Jako gracz chcę ulepszyć postać aby stała się silniejsza.

Opis. Zwiększanie statystyk postaci.

10. Jako gracz chcę ulepszyć karty by dawały lepsze bonusy lub miały lepsze efekty użytkowe.

Opis: Ulepszenie wymaga odpowiedniej ilości tej samej karty. Po ulepszenie dają ona lepszy efekt.

11. Jako gracz chcę zarządzać swoim ekwipunkiem aby nie panował tam chaos.

Opis: Dowolna możliwość przenoszenia przedmiotów w ekwipunku.

12. Jako gracz chcę spłacić dług aby ukończyć grę.

Opis: Należy wykonać tę czynność aby ukończyć grę. Wydanie odpowiedniej ilości waluty .

2. Wymagania niefunkcjonalne

Wymagania niefunkcjonalne zostały poddane oraz rozdzielone ba cztery podstawowe obszary wymagań.

Tabela 1.2. Wymagania niefunkcjonalne dla gry

Obszar wymagań	Nr	Opis
Użyteczność	1	Wszystkie funkcjonalności gry dostępne dla użytkownika muszą działać.
	2	Wszystkie funkcjonalności gry dostępne dla użytkownika muszą mieścić się na pojedynczym ekranie przy rozdzielczości 1280x720.
Niezawodność	3	Wszystkie błędy gry muszą być sygnalizowane czytelnym dla użytkownika komunikatem. Niedopuszczalne jest nagłe zakończenie gry z powodu błędu.
Wydajność	4	Gra powinna osiągać wartość co najmniej 30 klatek na sekundę.
	5	Gra nie powinna się uruchamiać dłużej niż 5 min.
	6	Gra nie powinna wykorzystywać w 100 % zasobów komputera.
Utrzymanie	7	Gra jest przeznaczona na komputery z systemem operacyjnym Windows 7 lub nowszy.

	8	Potrzebna jest określona pojemność dysku do przechowywania gry.

3. Metodyka

Model kaskadowy (waterfall) - Polega on na wykonywaniu podstawowych czynności jako odrębnych faz projektowych, w porządku jeden po drugim. Każda czynność to kolejny schodek (kaskada):

1. Planowanie systemu (w tym specyfikacja wymagań)
2. Analiza systemu (w tym analiza wymagań i studium wykonalności)
3. Projekt systemu (poszczególnych struktur itp.)
4. Implementacja (wytworzenie kodu)
5. Testowanie (poszczególnych elementów systemu oraz elementów połączonych w całość)
6. Wdrożenie i pielęgnacja powstałego systemu.

Jeśli któraś z faz zwróci niesatysfakcjonujący produkt cofamy się wykonując kolejne iteracje aż do momentu kiedy otrzymamy satysfakcjonujący produkt na końcu schodków.

Metodyka ta została wybrana ponieważ najbardziej pasuje do naszego projektu.

4. Technologia

Wykorzystane technologie:

- Język programowania – C#
- Framework – Microsoft .NET Framework

Wykorzystane biblioteki:

- Sytem
- UnityEngine

Środowisko:

- Silnik Unity

5. Podział pracy

- Pisanie kodu – Krzysztof Sobieski , Sebastian Majewski
- Grafika – Edyta Zakrzewska , Maciej Surawski

- Dokumentacja – Piotr Ołtarzewski
- Testowanie – Wszyscy członkowie grupy

6. Harmonogram

Deadline	Nazwa elementu projektu
9.04.2019	Zaprojektowanie sterowania mechanikami gry
16.04.2019	Stworzenie grafiki miasta i postaci
30.04.2019	Zaimplementowanie sterowania postaci
07.05.2019	Stworzenie grafiki lochów ,bossów oraz innych elementów
21.05.2019	Zaprogramowanie mechanik bossów
4.06.2019	Scalenie elementów gry
11.06.2019	Przetestowanie i naprawianie błędów , dokumentacja techniczna projektu
18.06.2019	Gotowy projekt