

**Specyfikacja wymagań
dla:**

Fallen Dream

Autor: Agata Wicikowska

Spis treści:

1. Wprowadzenie

- 1.1. Cel
- 1.2. Krótki opis aplikacji
- 1.3. Słownik pojęć
- 1.4. Literatura

2. Ogólny opis

- 2.1. Walory użytkowe i przydatność projektowanego systemu
- 2.2. Ogólne możliwości projektowanego systemu
- 2.3. Ogólne ograniczenia
- 2.4. Charakterystyka użytkowników
- 2.5. Środowisko operacyjne
- 2.6. Założenia i zależności

3. Wymagania

- 3.1. Wymagania funkcjonalne
- 3.2. Wymagania niefunkcjonalne

1. Wprowadzenie

- 1.1. Cel** - Aplikacja ma na celu zapewnienie rozrywki oraz ma pomóc w walce z koszmarami sennymi.
- 1.2. Krótki opis aplikacji** - Gra zręcznościowa polegająca na zestrzeliwaniu obiektów w postaci koszmarów sennych. Ma zapewnić komfort i rozrywkę. Gdy użytkownik obudzi się w nocy po koszmarnym śnie, gra ma na celu uspokoić użytkownika i pomóc mu ponownie zasnąć.
- 1.3. Słownik pojęć** - Fallen Dream zwane dalej grą i/lub aplikacją. Postać zwana dalej graczem. Obiekty do pokonania zwane dalej przeciwnikami.
- 1.4. Literatura** - Mapa aplikacji, wygląd okien

2. Ogólny opis

- 2.1. Walory użytkowe i przydatność projektowanej aplikacji** - Aplikacja ma zapewnić użytkownikowi uczucie spokoju po koszmarnym śnie, aby ponownie mógł zasnąć oraz ma zapewnić rozrywkę i wiele radości.
- 2.2. Ogólne możliwości projektowanego systemu** - Gra zręcznościowa 2D mająca na celu zestrzeliwanie oraz unikanie koszmarów. Użytkownik będzie poruszał postacią w prawo i w lewo za pomocą klawiszy strzałek, aby unikać koszmarów lecących na niego oraz zbierania przedmiotów pomocniczych. Gracz ma również możliwość zestrzeliwania przeciwników. Postać będzie automatycznie strzelać w przód. Jeżeli gracz wpadnie w kolizję z przeciwnikiem gra kończy się i wyświetla ekran podsumowania z wynikiem, który pozwala na ponowne zagraenie lub wyjście z gry. Aplikacja zawiera ekran menu, w którym można zamknąć grę, rozpocząć rozgrywkę oraz przejść do opcji, w których użytkownik może wyłączyć lub włączyć dźwięk i wyjść z ekranu opcji. Poruszanie się w ekranie menu, opcji oraz podsumowania umożliwiają klawisze strzałek. Zatwierdzenie wyboru w wymienionych ekranach następuje przez wciśnięcie klawisza "Enter".
- 2.3. Ogólne ograniczenia** - Brak możliwości zapauzowania gry, brak zapisu gry, brak odczytu gry, brak opcji multiplayer

- 2.4. Charakterystyka użytkowników** - Aplikacja przeznaczona dla użytkownika powyżej 3 roku życia. Polecana użytkownikom mającym problemy z zaśnięciem po strasznym koszmarze.
- 2.5. Środowisko operacyjne** - System: Windows 10, Procesor: 4-rdzeniowy 2,5GHz, RAM: 4Gb, Karta graficzna: niewymagana, Karta dźwiękowa: zintegrowana
- 2.6. Założenia i zależności** - Aplikacja ma być napisana w języku C i/lub C++ oraz wykorzystywać bibliotekę zewnętrzną Allegro5 i/lub SFML.

3. Wymagania

3.1. Wymagania funkcjonalne

Nazwa:	Sterowanie w aplikacji
Priorytet:	Normalny
Opis:	Aplikacja będzie posiadać sterowanie w menu głównym oraz opcji za pomocą klawiszy strzałek i możliwość zatwierdzenia wyboru za pomocą klawisza "Enter".

Nazwa:	Sterowanie postacią
Priorytet:	Normalny
Opis:	Aplikacja będzie posiadać możliwość sterowania postacią za pomocą klawiszy strzałek w prawo i w lewo.

Nazwa:	Włącznik/Wyłącznik dźwięku w ekranie opcji
Priorytet:	Normalny
Opis:	Aplikacja będzie posiadać w ekranie opcji możliwość wyłączenia i włączenia dźwięku. Przełączanie za pomocą kliknięcia klawisza "Enter", gdy aktywne jest pole "Dźwięk".

Nazwa:	Rozpoczęcie gry
Priorytet:	Normalny
Opis:	Aplikacja będzie posiadać możliwość rozpoczęcia gry z menu. Rozpocząć grę można, gdy przejdzie się klawiszami strzałek do pola "Graj" i zatwierdzi wybór przyciskiem "Enter". Gra po wykonaniu tej czynności powinna przejść do okna rozgrywki i rozpocząć grę.

Nazwa:	Wyjście z gry
Priorytet:	Normalny
Opis:	Aplikacja będzie posiadać możliwość wyjścia z gry z menu oraz z ekranu podsumowania. Wyjść z gry można, gdy przejdzie się klawiszami strzałek do pola "Zamknij grę" i zatwierdzi wybór przyciskiem "Enter". Gra po wykonaniu tej czynności powinna się zamknąć.

Nazwa:	Ponowne rozpoczęcie rozgrywki
Priorytet:	Normalny
Opis:	Aplikacja będzie posiadać możliwość rozpoczęcia ponownie rozgrywki z ekranu podsumowania. Rozpocząć ponownie grę można, gdy przejdzie się klawiszami strzałek do pola "Spróbuj ponownie" i zatwierdzi wybór przyciskiem "Enter".

Nazwa:	Wyjście z ekranu opcji
Priorytet:	Normalny
Opis:	Aplikacja będzie posiadać możliwość wyjścia z ekranu opcji do ekranu menu. Wyjść z ekranu opcji do ekranu menu można, gdy przejdzie się klawiszami strzałek do pola "Wróć do menu" i zatwierdzi wybór przyciskiem "Enter".

Nazwa:	Zestrzeliwanie przeciwników
Priorytet:	Normalny
Opis:	Aplikacja będzie posiadać automatyczne tworzenie pocisków przez postać, które podczas kolizji z przeciwnikami powodują zniknięcie (zabicie) przeciwnika.

Nazwa:	Zderzenie gracza z przeciwnikiem
Priorytet:	Normalny
Opis:	Aplikacja będzie posiadać możliwość przejścia do ekranu podsumowania, gdy gracz zderzy się z przeciwnikiem (gracz zostaje zabity).

Nazwa:	Licznik wyniku
Priorytet:	Normalny
Opis:	Aplikacja będzie posiadać możliwość zliczania wyniku. Będzie on wyświetlany podczas rozgrywki w lewym górnym rogu okna oraz w ekranie podsumowania. Licznik wyniku zlicza punkty na podstawie ilości zabitych przeciwników.

Nazwa:	Przedmioty pomocnicze
Priorytet:	Niski
Opis:	Aplikacja będzie posiadać możliwość generowania przedmiotów pomocniczych, które po zebraniu przez gracza będą dawać określone bonusy takie jak: nieśmiertelność, podwójny strzał, podwójne punkty.

3.2. Wymagania niefunkcjonalne

Nazwa:	Wciśnięcie innych klawiszy niż dozwolone
Priorytet:	Normalny
Opis:	Aplikacja będzie posiadać możliwość blokady działania na innych klawisze niż dozwolone. Gra nie będzie reagować na inne klawisze niż dozwolone.

Nazwa:	Animacje 2D
Priorytet:	Normalny
Opis:	Aplikacja będzie posiadać grafikę 2D, która ze względu na swoje wady posiada prostsze, ograniczone animacje w przeciwieństwie do grafiki 3D.

Nazwa:	Perspektywa 2D
Priorytet:	Normalny
Opis:	Aplikacja będzie posiadać grafikę 2D, która ze względu na swoje wady posiada widok tylko z jednej perspektywy.

Harmonogram:

1. Stworzenie GUI
2. Przygotowanie koncepcji postaci oraz przeciwników
3. Implementacja ruchu
4. Utworzenie funkcji odpowiedzialnej za generowanie przeciwników
5. Implementacja strzelania oraz kolizji
6. Dalsza implementacja kolizji oraz testy
7. Dodanie licznika wyniku oraz implementacja opcji i możliwości wyjścia z programu
8. Dodanie przedmiotów pomocniczych oraz testowanie aplikacji