

Specyfikacja Wymagań: Projekt PJPII

Jerzy Jurkowski, Informatyka Stosowana sem. 3, nr albumu: 154869

11 listopada 2016

Spis treści

1	Wstęp.	3
1.1	Cel	3
1.2	Zakres	3
1.3	Krótki przegląd	3
2	Ogólny opis.	4
2.1	Ogólne możliwości i ograniczenia projektowanego systemu . .	4
2.2	Charakterystyka użytkowników	4
2.3	Środowisko operacyjne	5
3	Specyficzne wymagania.	6
3.1	Wymagania funkcjonalne	6
3.1.1	Obsługa danych wejściowych	6
3.1.2	Interfejs Graficzny Użytkownika(GUI)	6
3.1.3	Logika Gry	7
3.1.4	Grafika	7
3.2	Wymagania niefunkcjonalne	8
4	Harmonogram (wg. tygodni)	9

1 Wstęp.

1.1 Cel

Celem powstania niniejszej specyfikacji wymagań jest sprecyzowanie dokładnie określonych w następnych sekcjach założeń projektowych. Na ich podstawie powstanie prototyp **gry**, który po wprowadzonych usprawnieniach będzie stanowił wersję ostateczną projektu.

1.2 Zakres

Poziom złożoności i konkretności przedstawionych wymagań musi być wystarczający do stworzenia działającego prototypu przygotowanego do stopniowej rozbudowy. Po rozbudowie specyfikacja zostanie uaktualniona.

1.3 Krótki przegląd

Projekt polega na wytworzeniu **gry** o nazwie roboczej „Polaris” w której głównym zadaniem gracza jest poruszanie sterowanej postaci po ekranie, unikając i niszcząc pojawiających się przeciwników.

2 Ogólny opis.

2.1 Ogólne możliwości i ograniczenia projektowanego systemu

Projektowana w ogólności **gra** musi być w stanie:

- Obsłużyć dane wejściowe wprowadzane za pomocą klawiatury i myszy.
- Przetworzyć dane i obsłużyć logikę **gry**.
- Wyświetlić interfejs graficzny(GUI) i grafikę na ekran.

Podczas projektowania i wdrażania projektu ograniczeniami są:

- Możliwości używanego języka i biblioteki.
- Możliwości sprzętowe komputera używanego do stworzenia **gry** jak i wymagania techniczne postawione wobec użytkownika.
- Umiejętności programowania wykonawcy projektu.
- Czas wykonania projektu wynoszący 8 tygodni.

2.2 Charakterystyka użytkowników

Gra kierowana jest do użytkowników zainteresowanych rozrywką w postaci gier komputerowych. Od graczy wymaga się interakcji z **grą** oraz znajomości podstawowych mechanik gier komputerowych. Ponadto od użytkownika wymaga się sprostania poniższym wymaganiom technicznym.

- Komputer z systemem Windows 8.1 64bit.
- Posiadanie następujących urządzeń peryferyjnych: mysz komputerowa, klawiatura, monitor, urządzenie do odtwarzania dźwięku wraz z kartą dźwiękową.
- Obsługa rozdzielczości ekranu w stosunku 16:9 (co najmniej 1366x768).

2.3 Środowisko operacyjne

Powstawanie projektu odbędzie się przy użyciu następujących narzędzi:

- System operacyjny Windows 8.1 64bit obsługiwany przez czterordzeniowy procesor 2.4GHz.
- Środowisko programistyczne Microsoft Visual Studio 2015.
- Biblioteka przeznaczona do tworzenia gier Allegro 5.
- Język programowania C.
- Program do obróbki audio Reaper.
- Program do obróbki grafiki GIMP 2.

Program nie będzie korzystał z bazy danych, wszystkie zasoby graficzne i audio w postaci plików: .png,.jpg,.wav.

3 Specyficzne wymagania.

3.1 Wymagania funkcjonalne

3.1.1 Obsługa danych wejściowych

Program musi zbierać informacje z klawiatury obsługując dane w następujący sposób:

- Klawisze W,S - Poruszanie **postaci** w kierunku pionowym ekranu.
- Klawisze A,D - Poruszanie **postaci** w kierunku poziomym ekranu.
- Klawisze A,D użyte w szybkiej sukcesji(do jednej sekundy) - **Dash**, czyli szybki ruch **postaci** w kierunku poziomym.
- Lewy przycisk myszy - Podczas rozgrywki: wciśnięcie zapoczątkowuje **cięcie** ustawiając jego punkt początkowy na obecne położenie kursora, wypuszczenie ustawia punkt końcowy **cięcia** na obecne położenie kursora.

W **Menu** nawigacja wewnątrz graficznego interfejsu.

- Klawisz Escape - włączenie **pauzy**, z opcją wejścia do **menu** po jego ponownym naciśnięciu.

3.1.2 Interfejs Graficzny Użytkownika(GUI)

Po włączeniu **gry** ukazuje się graczowi **menu**.

W **menu** pokazują się następujące wybory:

- Nowa gra
- Volume +/-
- Wyjście z gry

Podczas **pauzy** silnik gry zatrzymuje się i po użyciu:

- Escape - powrót do **menu**.
- Dowolny przycisk przerywa **pauzę** i wznowia rozgrywkę.

Podczas rozgrywki jedyna widoczna część interfejsu punkty gracza.

3.1.3 Logika Gry

Stan gry(pauza,menu itp.) obsługuje zmienna stanu gry.

Postać ma zdefiniowane następujące operacje podporządkowane danym wejściowym:

- Poruszać się w kierunku pionowym i poziomym ze stałą niewielką prędkością.
- **Dashować** - poruszać się w kierunku poziomym ze stałą wysoką prędkością przez krótki okres czasu. Nie można używać podczas **dashowania**.
- Wykonywać **Cięcia**. Za ich pomocą gracz może niszczyć **przeciwników**. Dzieje się to poprzez utworzenie obszaru w kształcie prostokąta na podstawie danych wejściowych w postaci dwóch punktów na płaszczyźnie. Jeśli **przeciwnik** znajdzie się w tym obszarze **ginie**.
- Przy kolizji z przeciwnikiem, **gra** wraca do **menu**.

Przeciwnicy to obiekty o kształtach okręgów o określonym promieniu i poruszające się zgodnie z określonymi dla nich typami ruchów. Kontakt **postaci i przeciwnika** kończy się utratą życia. **Przeciwnicy** są generowani z pseudolosowym położeniem z częstotliwością zależną od czasu rozgrywki.

Przewidywane typy ruchów to:

- Stała prędkość, stały kierunek.
- Przeciwnik który po zniszczeniu rozbija się na 4 przeciwników innego typu.

3.1.4 Grafika

Zmienne położenia obiektów powinny być zależne od obecnie używanej rozdzielczości ekranu.

Podczas rozgrywki tło stanowi animację. Można ten problem rozwiązać poprzez translację, rotację, skalowanie, zmianę koloru bitmapy z kanałem alfa.

Menu tworzone zostanie za pomocą generowania napisów o określonych pozycjach.

Każda ze zdefiniowanych operacji **postaci** powoduje pewną animację. Animacje postaci mogą zostać przeprowadzone w następujący sposób:

- Obsługa zmiany stanów ruchu i rysowanie różnych bitmap zależnych od niej.
- Zmiana koloru **postaci**.
- Rotacja **postaci**.

Wszyscy **przeciwnicy** obracają się stale wokół własnej osi. Przy **śmierci** zatrzymują się i robią tzw. „fade to alpha”. W danym momencie czasu rozgrywki nie może być utworzonych więcej niż 32 przeciwników.

Cięcie wyświetlane jest na ekranie poprzez narysowanie kolorowego prostokąta który po czasie staje się przezroczysty („fade to alpha”).

3.2 Wymagania нефunkcjonalne

Gra i jej elementy graficzne muszą być przejrzyste i czytelne w taki sposób, aby jak najbardziej umożliwić użytkownikowi podejmowanie szybkich decyzji podczas rozgrywki.

Wykonywanie programu na sprzęcie użytkownika musi odbywać się zapewniając stały „frame rate” (maksymalnie 60 klatek na sekundę).

Zasady **gry** powinny być zbalansowane w taki sposób aby zapewnić optymalny poziom trudności.

4 Harmonogram (wg. tygodni)

1. Pętla gry, obsługa ruchu postaci.
2. Prototypowanie modelu przeciwnika i mechaniki cięcia.
3. Pseudolosowa generacja przeciwników, wprowadzenie nowych typów przeciwników i Alternatywne rodzaje przeciwników.
4. Animacje cz.1
5. Animacje cz.2
6. Implementacja GUI oraz balansowanie gry.
7. Oprawa dźwiękowa.
8. Dopracowywanie kodu.