Specyfikacja wymagań gry *„Spadochroniarz”*

Autor dokumentu: **Dawid Lubczyński**

Data utworzenia dokumentu: **09.11.16 r.**

Data ostatniej modyfikacji dokumentu: **23.01.17 r.**

Wersja dokumentu: **1.08**

Spis treści

**1. Streszczenie**3

**2. Ogólny opis**.4

**3. Specyficzne wymagania**5

**4. Harmonogram prac nad projektem** ………………………………………………………………………………….........……6

1. **Streszczenie.**

Projekt powstał w celu stworzenia własnej gry. Ma to na celu poszerzyć zakres mojej wiedzy na temat programowania, poznanie dodatkowych bibliotek języka C++, poprzez praktykę. Celem gry będzie unikanie przeszkód oraz zbieranie monet. Postać po zejściu z platformy startowej będzie spadać w dół. Wówczas rozpocznie się rozgrywka. Inspiracją do jej powstania są wspomnienia z dzieciństwa. Jako uczeń szkoły podstawowej wielokrotnie uczestnicząc w zajęciach komputerowych odkrywaliśmy coraz to bardziej zaawansowane i różnorodne gry. Od dzieciństwa postanowienie było jedno – w przyszłości „zrobię” własną grę.

1. **Ogólny opis.**

*2.1. Relacje do bieżących projektów:*

Nie dotyczy.

*2.2. Funkcje i cele*

Celem gry jest zainteresowanie osoby grającej.

*2.3. Ustalenia dotyczące środowiska*

Gra zostanie napisana w języku C++, korzystając z bibliotek zewnętrznych, posługując się programami: Code::Blocks oraz Visual Studio 2015. Zostanie ona przetestowana na komputerze przenośnym z systemem Windows 7. Wykorzystana zostanie klawiatura wbudowana, touchpad, oraz myszka.

*2.4. Relacje do innych systemów*

Gra będzie współpracować z systemem Windows 7.

*2.5. Ogólne ograniczenia*

Ograniczenie wielkości generowanej przestrzeni spadku.

*2.6. Opis modelu*

Autor tworzy oprogramowanie zgodnie z modelem ewolucyjnym oprogramowania. Wprowadza zmiany w funkcjonalności i architekturze programu (po uprzedniej konsultacji   
z osobą prowadzącą przedmiot).

3. Specyficzne wymagania

*3.1. Wymagania dotyczące funkcji systemu*

Prawidłowe działanie menu, w przypadku przegranej – wyświetlenie napisu „Koniec Gry”.

*3.2. Wymagania dotyczące wydajności systemu*

Płynne działanie gry. Brak błędów.

*3.3. Wymagania dotyczące zewnętrznych interfejsów*

Wymagana klawiatura (opcjonalnie – myszka).

*3.4. Wymagania dotyczące wykonywanych operacji*

Ruch postaci w lewo oraz w prawo.

*3.5. Wymagania dotyczące wymaganych zasobów*

Nie dotyczy.

*3.6. Wymagania dotyczące sposobów testowania*

Aby przetestować grę potrzebujemy dowolny komputer, z systemem Windows, który będzie miał podłączoną działającą klawiaturę.

*3.7. Wymagania dotyczące dokumentacji*

Nie dotyczy.

*3.8. Wymagania dotyczące ochrony*

Nie dotyczy

*3.9. Wymagania dotyczące przenośności*

Nie dotyczy.

*3.10. Wymagania dotyczące jakości*

Nie dotyczy.

*3.11. Wymagania dotyczące niezawodności*

Program powinien wyświetlać stosowne komunikaty, jeśli wystąpi nieoczekiwany błąd w jego działaniu.

*3.12. Wymagania dotyczące pielęgnacyjności*

Nie dotyczy.

*3.13. Wymagania dotyczące bezpieczeństwa*

Nie dotyczy.

***Harmonogram prac nad projektem:***

*22.11.16* – Wprowadzenie podstawowej funkcjonalności ( poruszająca się postać gracza, podstawowe elementy postaci ).

*29.11.16* – Poprawa istniejących błędów, dopracowywanie kodu ( wprowadzenie losowo generowanych przeszkód poruszających się w górę, ograniczenie możliwości ruchu postaci do szerokości okna gry).

*6.12.16* – Edycja istniejącego kodu, dopracowywanie szczegółów, dodanie kolizji oraz punktacji ( kolizja postaci z przeszkodami, wyświetlanie przeszkód tylko na szerokości okna gry, punktacja (ominięta przeszkoda = 1pkt), jeżeli gracz uderza w przeszkodę, traci on jedno życie).

*13.12.16* – Poprawki błędów, dodanie funkcji zbierania monet podczas lotu, dodanie grafiki   
( do gry dodana została funkcja zbierania monet oraz żyć, jedna moneta daje +10 pkt, jedno serce (życie) daje +1 życia, dodana została część grafiki oraz końcowe statystyki ).

*20.12.16* – Dodanie efektów wizualnych postaci, drobne poprawki. Stworzenie poziomów trudności gry. Praca nad menu ( dodana została animacja postaci – ruch w lewo, w prawo, oraz bezruch, gra dostała „poziomy trudności” - gdy gracz osiąga odpowiednio 100, 200, 300, 400, 500 pkt postać spada coraz szybciej, dodane zostało podstawowe menu oraz możliwość ponownego rozpoczęcia gry bez wychodzenia z aplikacji ).

*10.01.17*– Szukanie oraz poprawa istniejących błędów, uporządkowanie kodu ( drobne poprawki kodu, usunięcie drobnych błędów, dodanie chmur będących tłem gry ).

*17.01.17*– Sprawdzenie płynności ruchu oraz sprawdzenie działania gry. Dopracowanie tła oraz szczegółów gry ( drobne poprawki polepszające grafikę, poprawienie działania tła ).

*24.01.17* – Końcowe poprawki graficzne, usprawnienia gry, korekta kolizji, dodanie opisów.