

Gliwice, 19.03.2018 r.

# **Projekt z Grafiki Komputerowej**

Dokument projektowy

Tytuł projektu:  
Platformowa gra 2D

Prowadzący:  
dr Michał Staniszewski

Skład sekcji:  
Marek Skrzypecki  
Wojciech Wrona

## 1. Treść zadania.

Gra platformowa 2D zrealizowana z wykorzystaniem języka C++ oraz bibliotek graficznych umożliwiających tworzenie gier komputerowych w tym języku. Gracz porusza się postacią i omijając bądź niszcząc przeciwników oraz przeszkody na swojej drodze, pokonuje kolejne poziomy gry.

## 2. Analiza zadania.

1) Podstawy teoretyczne problemu

2) Wykorzystywane zagadnienia grafiki komputerowej

- Przekształcenia afiniczne i reprezentacja obiektów.
- Animacja komputerowa
- Wykrywanie kolizji

3) Wykorzystywane biblioteki i narzędzia programistyczne

- Visual Studio jako rozbudowane narzędzie programistyczne zostanie wykorzystane w celu implementacji projektu z wykorzystaniem języka C++.
- Biblioteka SFML - biblioteka graficzna do języka C++ ułatwiająca korzystanie z mechanizmów do tworzenia grafiki komputerowej, w wykorzystywanym środowisku programistycznym. SFML wprowadza podejście obiektowe do programowania efektów graficznych co pozwala na lepszą integrację warstwy graficznej programu, z resztą programu napisaną z wykorzystaniem podejścia programowania zorientowanego obiektowo. Dodatkowo biblioteka SFML zapewnia multiplatformowość tworzonych z jej wykorzystaniem programów. Jest ona zorientowana w celu optymalizacji grafiki dwuwymiarowej, w związku z czym nie wspiera ona mechanizmów do tworzenia grafiki trójwymiarowej. Jest to ograniczenie, które nie wpływa na specyfikację projektu ze względu na wybór implementacji grafiki dwuwymiarowej. Biblioteka SFML nie jest przystosowana do prostego tworzenia graficznego interfejsu użytkownika w przeciwieństwie do, na przykład, biblioteki QT, jednak jej możliwości pozwalają na tworzenie intuicyjnego interfejsu gry komputerowej.

4) Struktury danych

- Klasa opisująca mapę zrealizowana wewnętrznie w postaci tablicy jednowymiarowej.
- Klasa przechowująca konfigurację klas graficznych, w tym zbiór tekstur wykorzystywanych przez grę w tablicy.
- Klasa przechowująca informacje o znajdujących się na mapie przeciwnikach implementowana z wykorzystaniem tablicy jednowymiarowej.

5) Algorytmy

- Algorytm odpowiadający za obsługę fizyki w grze
- Algorytm poruszania się przeciwników

6) Ograniczenia specyfikacji

- Ograniczenie grafiki do grafiki dwuwymiarowej
- Gra przeznaczona jest dla jednego użytkownika, bez rozróżnienia użytkowników lokalnych

### 3. Plan pracy.

- 1) Stworzenie dokumentacji. - 1 dzień data ukończenia: 20/03/2018
- 2) Przygotowanie środowiska programistycznego oraz repozytorium. - 1 dzień data ukończenia: 23/03/2018
- 3) Stworzenie projektu odpowiedzialnego za grafikę oraz działającej pustej aplikacji graficznej. - 1 dzień data ukończenia: 24/03/2018
- 4) Stworzenie klasy przechowującej globalne ustawienia grafik, między innymi ścieżki tekstur. - 1 dzień data ukończenia: 25/03/2018
- 5) Stworzenie klasy odpowiedzialnej za działanie MENU – 1 dzień data ukończenia: 26/03/2018
- 6) Stworzenie zestawu klas do obsługi mapy gry - kamień milowy, data ukończenia: 03/04/2018 :
  - I. Stworzenie klasy implementującej mapę gry w formie tablicy. - 1 dzień
  - II. Stworzenie klasy rysującej mapę na podstawie tablicy. - 1 dzień
  - III. Integracja klas wymienionych w podpunktach I i II. - 1 dzień
  - IV. Przetestowanie działania mapy. -1 dzień
- 7) Stworzenie zestawu klas do obsługi poruszania się gracza po poziomie – kamień milowy, data ukończenia: 10/04/2018:
  - I. Stworzenie klasy odpowiedzialnej za animację postaci przy poruszaniu – 1 dzień
  - II. Stworzenie klasy odpowiedzialnej za obsługę fizyki w grze – 2 dni
  - III. Stworzenie klasy odpowiedzialnej za obsługę kolizji w grze – 1 dzień
  - IV. Integracja powyższych klas i testowanie ich działania – 1 dzień
- 8) Stworzenie zestawu klas odpowiedzialnych za obsługę przeciwników oraz interakcje gracza z nimi - kamień milowy, data ukończenia: 26/04/2018 :
  - I. Implementacja klasy odpowiedzialnej za tworzenie przeciwników na mapie - 1 dzień
  - II. Implementacja klasy odpowiedzialnej za poruszanie/animację przeciwników - 1 dzień
  - III. Implementacja klasy odpowiedzialnej za interakcję gracza z przeciwnikami - 2 dni
  - IV. Implementacja usuwania pokonanych przeciwników z mapy - 1 dzień
  - V. Przetestowanie działania stworzonych funkcjonalności oraz ich integracji z wcześniejszymi etapami projektu - 2 dni
- 9) Stworzenie zestawu klas zarządzających pojedynczą rozgrywką (rozpoznanie końca gry oraz warunków początkowych) - 2 dni data ukończenia: 08/05/2018
- 10) Obsługa statystyk gracza. - 2 dni data ukończenia: 13/05/2018
- 11) Przetestowanie działania programu + ewentualne poprawki - 3 dni data ukończenia: 16/05/2018 – finalna wersja beta

### 4. Wstępny podział pracy.

Skrzypecki Marek	Wrona Wojciech
<ul style="list-style-type: none"><li>• 1)Dokumentacja</li><li>• 3)Pusta Aplikacja Graf</li><li>• 4)Globalne ust. grafik</li><li>• 5)Menu</li><li>• 6)Mapa</li><li>• 8)Przeciwnicy</li><li>• 11)Testowanie</li></ul>	<ul style="list-style-type: none"><li>• 1)Dokumentacja</li><li>• 2)Środowisko + Repo</li><li>• 3)Pusta Aplikacja Graf</li><li>• 7)Poruszanie, animacja</li><li>• 9)Zarządzanie rozgrywką</li><li>• 10)Obsługa Statystyk</li><li>• 11)Testowanie</li></ul>

## **5. Link do repozytorium**

<https://github.com/CerberPup/Arcane-Inscriber>