



WYDZIAŁ ELEKTRONIKI  
I TECHNIK INFORMACYJNYCH

# **SPRAWOZDANIE**

**PROJEKTU Z PRZEDMIOTU:**

**PROI – PROGRAMOWANIE OBIEKTOWE**

**„GRA RUN’N’GUN”**

**OPRACOWALI – CZUBAK ADAM, MICHALSKI SEBASTIAN**

**CZERWIEC 2022**

## Cel i opis projektu

Projekt pt. „Gra Run’n’gun” ma na celu napisanie programu w języku C++, który umożliwi funkcjonalną rozgrywkę. Inspiracją jest gra z serii „Contra”. Gra powinna też dostarczać możliwości zapisu i wczytania stanu gry, rozegrania kilku rozgrywek pod rząd, wyświetlania wyników i statystyk itp.

Do użycia w projekcie zostały zaproponowane takie technologie jak:

- biblioteka Qt - graficzny interfejs okienkowy,
- biblioteka SFML - gry komputerowe,
- biblioteka Box2D - symulacja fizyki w przestrzeni dwuwymiarowej (bardzo dobra biblioteka na potrzeby prostych gier komputerowych)

## Opis programu

Program został podzielony na kilka plików i folderów:

- folder Fonts,
- folder Textures,
- run’n’gun.cpp,
- Game.h; Game.cpp,
- Player.h; Player.cpp,
- Enemy.h; Enemy.cpp,
- Bullet.h; Bullet.cpp,
- Block.h; Block.cpp.

Folder Fonts zawiera wymagane czcionki do wyświetlania statystyk.

Folder Textures zawiera tekstury gracza, przeciwników, pocisków i mapy.

Plik run’n’gun.cpp zawiera funkcję main(), która tworzy obiekt klasy Game.

Pliki Game zawierają klasę Game, która odpowiada za liczne funkcje. Przede wszystkim tworzy okno z grą, wczytuje tekstury z plików, tworzy obiekt klasy Player, a także obiekty klasy, Bullet, Enemy, Block. Dokonuje wyświetlenia wszystkich elementów w oknie.

Pliki Player zawierają klasę Player, która odpowiada za funkcje związane z graczem. Jego życia, naboje, animacje, poruszanie się i zadawanie ataku.

Pliki Enemy zawierają klasę Enemy, która odpowiada za przeciwników w grze. Ich pozycję na mapie i zadawanie obrażeń, a także przyjmowanie obrażeń.

Pliki Bullet zawierają klasę Bullet, która odpowiada za pociski, zarówno te wystrzelone przez gracza jak i przeciwników. Ich prędkość oraz poruszanie się.

Pliki Block zawierają klasę Block. Obiekty tej klasy są to bloczki które zostały naniesione na tło mapy aby umożliwić kolizje z graczem.

## **Zastosowane narzędzia**

Głównym narzędziem jakie zostało zastosowane jest biblioteka SFML. Jest to na tyle rozbudowana biblioteka, że po większym zapoznaniu się z jej funkcjami odrzuciliśmy potrzebę stosowania innych bibliotek, ponieważ ta pozwoliła nam spełnić wszystkie nasze wymagania. Umożliwia ona wyświetlanie grafiki 2D, a co więcej poruszanie się obiektem View po kolejnych etapach mapy. Dzięki niej są również możliwe kolizje z obiektami różnych klas. Poruszanie się obiektów, wyświetlanie statystyk i „grawitacja” również dzięki niej zostały zaimplementowane.

Biblioteka SFML jest całkiem prosta w obsłudze i zrozumieniu z wyjątkiem niektórych funkcji, które wymagają przeczytania uważnie dokumentacji SFML. Na stronie SFML jest również kilka przydatnych instrukcji i przykładów wraz ze zdjęciami użycia tej biblioteki do wyświetlania grafiki.

## **Instrukcja użytkownika**

Program można uruchomić dzięki plikowi exe, który został skompilowany przez kompilator. Do jego uruchomienia są potrzebne również foldery Textures oraz Fonts, a także pliki biblioteki SFML, takiej jak:

- sfml-graphics-2.dll,
- sfml-system-2.dll,
- sfml-window -2.dll.

Ten zestaw plików pozwala na bezbłędne uruchomienie się gry.

Sterowanie w grze wygląda następująco:

- klawisze A, D – odpowiednio ruch w lewo i prawo,
- klawisz S – zejście blocek niżej,
- klawisz B – strzał pociskiem,
- klawisz Spacja – podskok,
- klawisz R – restart gry,
- klawisz P – zapis gry,
- klawisz O – wczytanie ostatniego zapisu gry.

Trzymanie klawisza B powoduje cykliczne wystrzały pocisków.

Zadaniem w grze jest zastrzelenie przeciwników i unikanie pocisków od nich np. skacząc nad pociskami, pokonując całą mapę.