

# Dokumentacja projektu - The Mighty Spaceship

Karolina Chrząszcz – Bonus, Player

Szymon Górnioczek – Bullet, Enemies, Spaceship

Tomasz Kryg – Bullet, Bullets, My\_Window, Spaceship

Michał Wolszleger – Bullet, Enemies, Enemy, Player

## Część I

### Opis programu

#### 1. Ogólne informacje:

Program "The Mighty Spaceship" jest grą. Polega na poruszaniu się statkiem kosmicznym po planszy oraz na zestrzeliwaniu wrogów. Co pewien czas pojawiają się bonusy a także specjali wrogowie, których należy pokonać w celu awansowania na wyższy poziom. Im wyższy jest nasz poziom, tym trudniejsza staje się gra.

#### 2. Możliwości i ograniczenia

Gra przeznaczona jest dla osób w dowolnym przedziale wiekowym.

#### 3. Wymagania programu

Program wymaga systemu operacyjnego Windows w wersji: XP, Vista, 7

lub 8. Wymagane jest również zainstalowanie pakietu Visual C++ Redistributable Packages for Visual Studio 2013, który można za darmo pobrać ze strony:

<http://www.microsoft.com/en-us/download/details.aspx?id=40784>

oraz biblioteki FLTK, którą można za darmo pobrać ze strony:

<http://www.fltk.org/software.php>

## **Instrukcja obsługi**

### **1. Jak uruchomić program?**

Po zainstalowaniu powyższego pakietu, wystarczy włączyć dwukrotnym kliknięciem lewego przycisku myszy ikonę programu o plik 'TheMightySpaceship', który otwierany jest w Microsoft Visual Studio. Należy go skompilować.

### **2. Co dalej?**

Po włączeniu programu, pojawi się menu gry. Odpowiednie napisy na przyciskach informują nas o tym, gdzie trafimy po ich naciśnięciu. Naciskając przycisk z etykietką "New Game" rozpoczniemy nową grę. Aby załadować zapisaną wcześniej grę należy wybrać przycisk "Load Game". Wybierając przycisk "Instructions" otrzymamy wskazówki dotyczące gry. Przycisk "Au-

thors" spowoduje wyświetlenie autorów tworzących tę grę, natomiast klikając przycisk "Quit" zakończymy działanie programu.

## Część II

### Specyfikacja techniczna

Projekt gry "The Mighty Spaceship" podzielony został na 12 plików nagłówkowych (z rozszerzeniem \*.h) oraz 12 głównych plików (z rozszerzeniem \*.cpp).

#### 1. Pliki nagłówkowe:

- `Background.h`

Zawiera prototypy klasy `Background` służącej do narysowania tła.

- `Bonus.h`

Zawiera prototypy klasy `Bonus` pozwalającej na generowanie się bonusów podczas gry.

- `Bullet.h`

Zawiera prototypy klasy `Bullet` opisującej pociski, którymi można strzelać.

- `Bullets.h`

Zawiera prototypy klasy `Bullets` pozwalającej na generowanie się w odpowiednich momentach nowego obiektu z klasy `Bullet`.

- `config.h`

Zawiera jedynie zdefiniowane zmienne takie jak: wielkość okna, wielkość planszy, początek planszy, długość efektu poruszania się statku w prze-

strzeni oraz wielkości obrazków.

- `Enemies.h`

Zawiera prototypy klasy `Enemies`, która pozwala na całkowicie losowe generowanie się obiektów klasy `Enemy` (wrogów).

- `Enemy.h`

Zawiera prototypy klasy `Enemy` opisującej wrogów.

- `Game_Board.h`

Zawiera prototypy klasy `Game_Board` służącej do narysowania tła plan-szy.

- `My_Button.h`

Zawiera prototypy klasy `My_Button`, która jest odpowiednio określonym obiektem typu `Fl_Button`.

- `My_Window.h`

Zawiera prototypy klasy `My_Window`, która odpowiada za odpowiednie skonstruowanie i połączenie wszystkich klas w jednym oknie.

- `Player.h`

Zawiera prototypy klasy `Player`, dzięki której można zapisywać oraz ła-dować grę.

- `Spaceship.h`

Zawiera prototypy klasy `Spaceship` opisującej statek kosmiczny, którym

gracz się porusza podczas gry.

## 2. Główne pliki:

- `Background.cpp`

Tutaj zdefiniowany jest konstruktor i destruktor klasy `Background` oraz funkcja `draw()`, która rysuje w odpowiednim miejscu zadane tło.

- `Bonus.cpp`

W tym miejscu znajdują się definicje konstruktora, destruktor oraz funkcji działających w klasie `Bonus` takich jak funkcja `draw()` oraz inne funkcje odpowiadające za generowanie, widzialność oraz kolizje.

- `Bullet.cpp`

Tutaj znajduje się konstruktor oraz destruktor klasy `Bullet` wraz z funkcją `draw()` pozwalającą na rysowanie pocisku a także funkcjami pozwalającymi na poruszanie się pocisku i sprawdzanie czy pocisk opuścił dopuszczalne pole gry.

- `Bullets.cpp`

Tutaj zdefiniowane zostały funkcje pozwalające na pojawianie się obiektów klasy `Bullet` w odpowiednich miejscach, usuwanie pocisków, które opuściły planszę oraz na rysowanie wszystkich istniejących pocisków.

- `Enemies.cpp`

W tym miejscu zdefiniowane zostały funkcje działające na klasie `Enemy`

i pozwalające na losowe generowanie oraz poruszanie się obiektów klasy Enemy i usuwanie wrogów, którzy opuścili pole gry. Są tu również funkcje pozwalające sprawdzić czy jakikolwiek wróg został trafiony pociskiem lub czy zderzył się z naszym statkiem. Znajduje się tutaj także funkcja generująca raz na jakiś czas mocniejszego wroga, który odbija się po planszy i znika dopiero po kilkukrotnym zestrzeleniu.

- **Enemy.cpp**

Tutaj zdefiniowany jest konstruktor oraz destruktory do klasy Enemy. Konstruktor określa losowe miejsce pojawienia się wroga. Znajduje się tutaj również definicja funkcji `draw()`, a także definicje funkcji pozwalających na poruszanie się wrogów oraz sprawdzanie czy i z której strony opuścili planszę.

- **Game\_Board.cpp**

Tutaj zdefiniowany został konstruktor oraz destruktory klasy `Game_Board` wraz z funkcją `draw()` pozwalającą na narysowanie pola gry.

- **main.cpp**

Ten plik odpowiada za połączenie wszystkiego w całość i włączenie się programu. Są tutaj zdefiniowane dwie funkcje pozwalające na konwertowanie liczb na znaki oraz dwie funkcje callback z użyciem funkcji `Fl::repeat_timeout`. Pierwsza z nich odpowiada przede wszystkim za od-

świeżanie się okna z prędkością 60 fps a także za przesuwanie się odpowiednich obiektów oraz bezustanne sprawdzanie kolizji, które mogły nastąpić. Druga funkcja (callbackBonus) pozwala na pojawianie się bonusów. W funkcji main generowane jest jedynie obiekt klasy `My_Window` oraz dwie funkcje `Fl::add_timeout`.

- `My_Button.cpp`

W tym miejscu zdefiniowane zostały tylko konstruktor oraz destruktor klasy `My_Button`. Konstruktor jest konstruktorem klasy `Fl_Button` z odpowiednimi początkowymi właściwościami przycisku.

- `My_Window.cpp`

Tutaj zdefiniowany został konstruktor oraz destruktor klasy `My_Window` oraz funkcje callback, które są przypisane do odpowiednich przycisków. Konstruktor tworzy wszystko co widać i czego nie widać podczas gry i łączy to w jednej klasie.

- `Player.cpp`

W tym miejscu zdefiniowany został konstruktor klasy `Player` oraz funkcje odpowiadające za zakończenie gry, zapis obecnego stanu gry do pliku, reset gry oraz załadowanie zapisanego stanu gry.

- `Spaceship.cpp`

Tutaj zdefiniowane zostały konstruktor oraz destruktor klasy `Spaceship`.



Jest tu zdefiniowana także funkcja `draw()` pozwalająca na rysowanie statku kosmicznego. Jest tu także definicja funkcji `handle`, która pozwala na sprawdzenie jaki klawisz został naciśnięty i przydzielenie go do odpowiedniej operacji. Znajduje się tu również funkcja odpowiadająca za poruszanie się statku.