

REV	DATA	ZMIANY
0.1	08.02.2021	<i>Adamek Mateusz (madamek@student.agh.edu.pl)</i>
0.2		

SIMPLE GAME

Autor: Adamek Mateusz
Akademia Górniczo-Hutnicza

Spis treści

1. WSTĘP.....	4
1.1 WYMAGANIA SYSTEMOWE.....	ERROR! BOOKMARK NOT DEFINED.
2. FUNKCJONALNOŚĆ (FUNCTIONALITY).....	5
3. PROJEKT TECHNICZNY (TECHNICAL DESIGN)	7
3.1 HIERARCHIA KLAS.....	ERROR! BOOKMARK NOT DEFINED.
4. OPIS REALIZACJI (IMPLEMENTATION REPORT)	9
4.1 MENU	ERROR! BOOKMARK NOT DEFINED.
4.2 BALL.....	ERROR! BOOKMARK NOT DEFINED.
4.3 PAWN	ERROR! BOOKMARK NOT DEFINED.
4.4 META.....	ERROR! BOOKMARK NOT DEFINED.
4.5 END_MENU.....	ERROR! BOOKMARK NOT DEFINED.
4.6 OPTIONS	ERROR! BOOKMARK NOT DEFINED.
4.7 LEVEL_MENU	ERROR! BOOKMARK NOT DEFINED.
4.8 MUSIC_CHOICE_MENU	ERROR! BOOKMARK NOT DEFINED.
4.9 COLOR_MENU	ERROR! BOOKMARK NOT DEFINED.
4.10 ACTION	ERROR! BOOKMARK NOT DEFINED.
5. PODRĘCZNIK UŻYTKOWNIKA (USER'S MANUAL).....	12
5.1 TEST APLIKACJI NA SYSTEMIE WINDOWS.....	12
5.2 SPOSOBY INTERAKCJI MIĘDZY UŻYTKOWNIKIEM A GRĄ	ERROR! BOOKMARK NOT DEFINED.
5.3 AKCJE ZWIĄZANE Z ROZGRYWKĄ.....	ERROR! BOOKMARK NOT DEFINED.
5.4 WYBÓR FUNKCJONALNOŚCI.....	ERROR! BOOKMARK NOT DEFINED.
BIBLIOGRAFIA.....	18
PLIKI PROJEKTOWE.....	18

Lista oznaczeń

SG	Simple Game
SFML	Simple and Fast Multimedia Library
GUI	Graphical user interface

1. Wstęp

Głównym celem projektu było stworzenie prostej gry z graficznym interfejsem użytkownika. Interfejs graficzny został wykonany przy użyciu SFML. Głównym celem gry jest zdobycie nagrody znajdującej się na planszy. Przeszkody poruszają się w różnych kierunkach przez co utrudniają osiągnięcie celu. Zderzenie się z przeszkodą lub osiągnięcie celu jest sygnalizowane specjalnym komunikatem. Po wyświetleniu się komunikatu można rozpocząć grę od nowa przez powrót do menu głównego lub zakończyć grę wybierając opcję "exit".

1.1 Wymagania Systemowe

Podstawowe założenia projektu:

1. Przygotowanie syntetycznego opisu
2. Opracowanie architektury systemu przetwarzania sygnałów cyfrowych oraz dalsze dookreślenie potrzeb i założeń takich jak m.in. interfejsy modułów
3. Opracowanie modułu ...

2. Funkcjonalność (*functionality*)

Gra zwana później aplikacją posiada szereg funkcjonalności, najważniejsze z funkcjonalności tej gry wymiennie poniżej.

- DIFFICULTY LEVEL(zmiana poziomu trudności)
 - EASY
 - MENIDUM
 - HARD
- MUSIC(wybór muzyki grającej w tle)
 - podklad
 - My_Dorbel
 - Hardest_Button
- PLAYER COLOR(wybór koloru(tekstury) gracza)
 - BLUE
 - YELLOW
 - GREEN

Poniższe funkcjonalności prezentują się następująco.

Menu wyboru funkcjonalności:

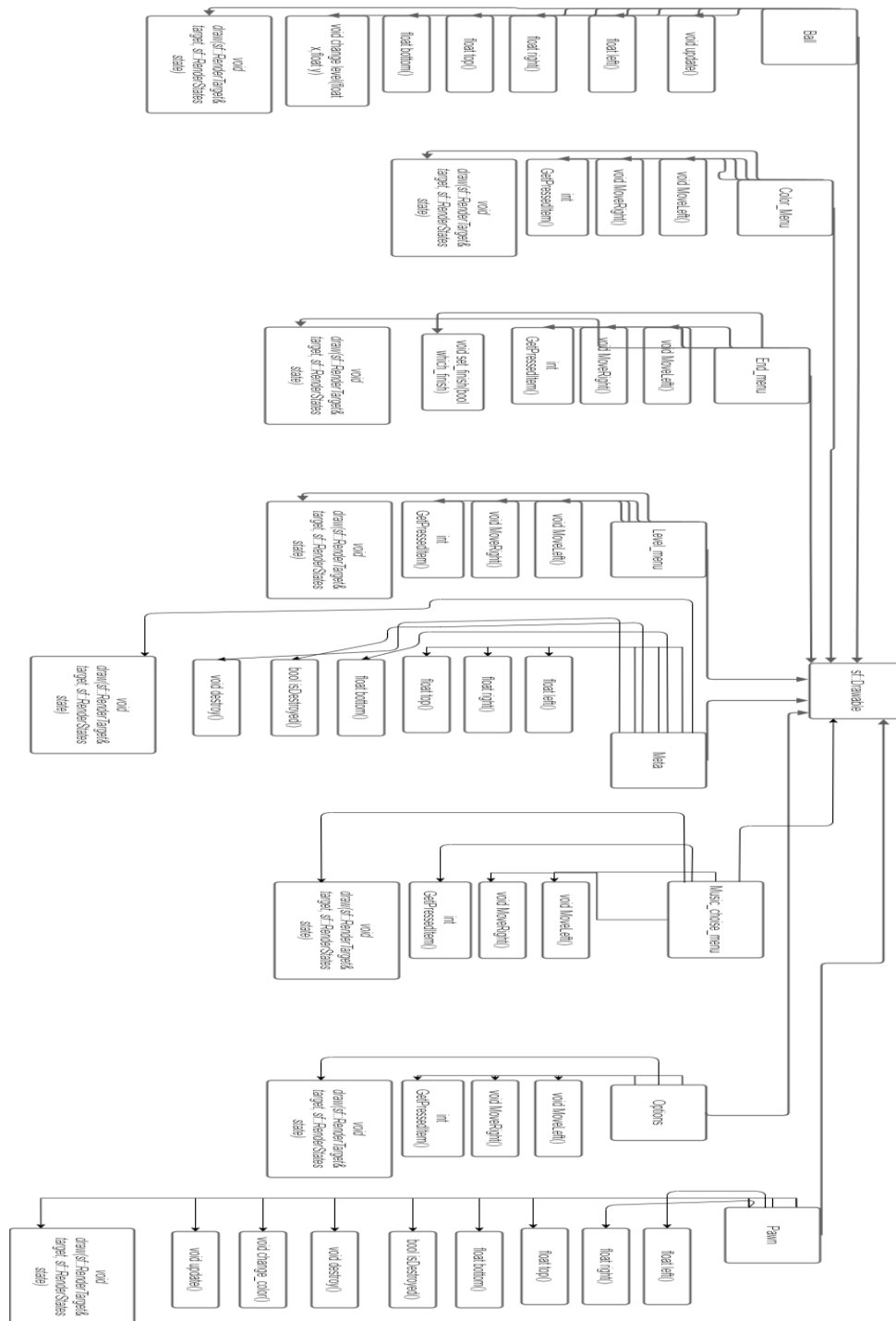


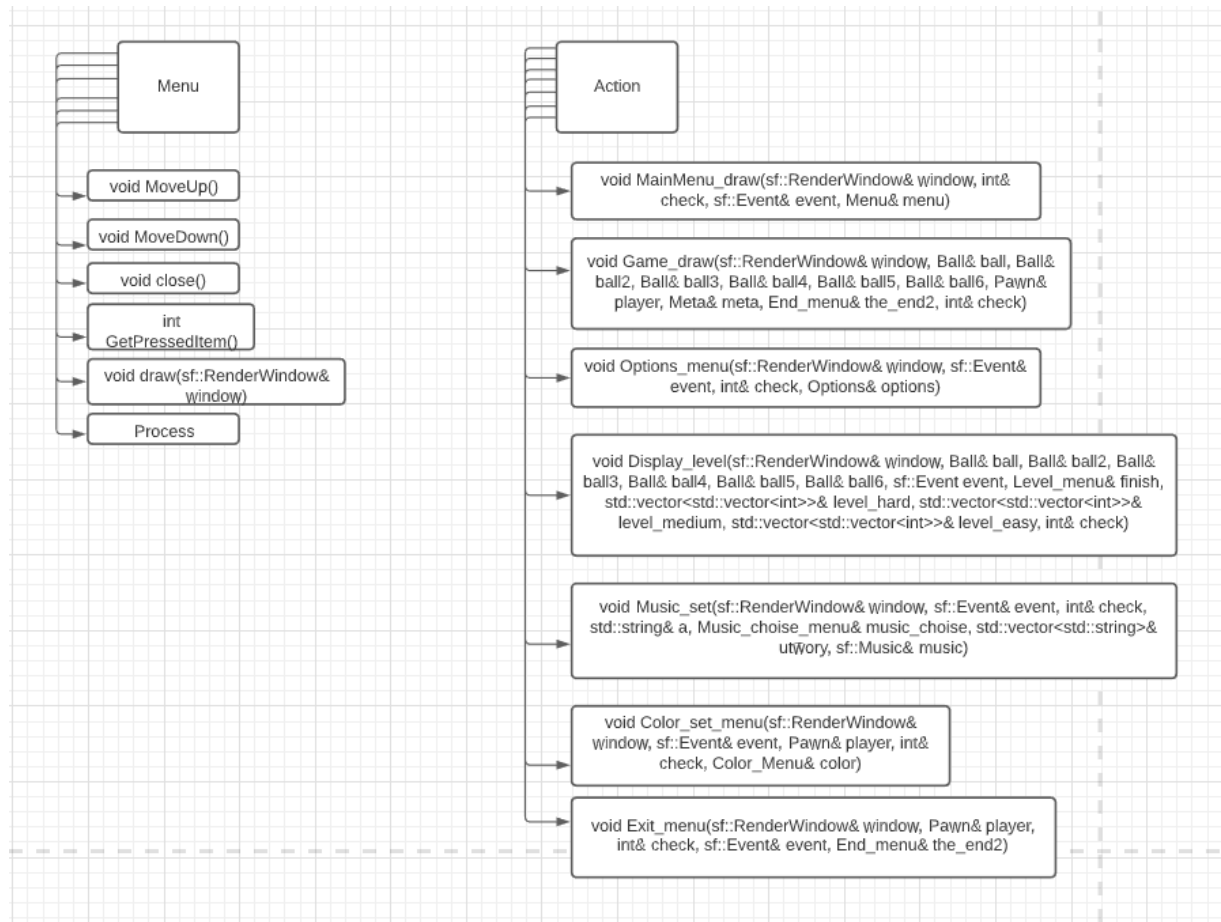
Przykładowe submenu wyboru docelowej funkcjonalności.



3. Projekt techniczny (*technical design*)

3.1 Hierarchia klas





4. Opis realizacji (*implementation report*)

Simple Game jest podzielona na kilka klas i plik główny, w pliku głównym o nazwie JPO2.cpp, plik główny JPO2.cpp ma za zadanie połączyć funkcjonalności pozostałych klas. W tym pliku są wywoływane są konstruktory danych klas. Oraz elementy pozwalające wywoływać metody poszczególnych klas.

W naszej aplikacji możemy wyróżnić następujące klasy

- Action
- Ball
- Color_Menu
- End_menu
- Level_menu
- Menu
- Meta
- Music_choise_menu
- Options
- Pawn

4.1 Menu

Klasa Menu jest odpowiedzialna za stronę tytułową naszej aplikacji, po uruchomieniu naszej aplikacji to ona wyświetla nam się jako pierwsza. W konstruktorze klasy Menu są przyjmowane parametry szerokość i wysokość które definiują rozmiar naszego okna które będziemy chcieli wyświetlić. W tej klasie znajduje się także pięć funkcji publicznych oraz 3 zmienne prywatne. Poniżej po krótko opisze działanie powyższych metod.

4.2 Ball

Klasa Ball jest odpowiedzialna za tworzenie przeszkód w postaci piłek, Klasa w swoim konstruktorze przyjmuje takie parametry jak pozycja piłki X, Y oraz prędkość piłki w płaszczyźnie X i Y . W klasie są zdefiniowane funkcje odpowiedzialne za zachowanie piłki podczas spotkania się z krawędzią naszego okna(współzrędnymi które określają rozmiar naszego okna), Mianowicie jest obliczane czy współzrędnne piłki pokryły się z krawędzią. W funkcjach top(), bottom(), lef(), right() są obliczane nasze krawędzie naszej piłki, ponieważ w konstruktorze punkt odniesienia naszej piłki ustawiamy na jej środek. Funkcja change_level jest odpowiedzialna za zmianę szybkości poruszania się piłki, funkcja update za odświeżanie piłki na ekranie.

4.3 Pawn

Klasa Pawn jest to klasa w której definiujemy naszego pionka(player'a) to nim będziemy się poruszać w naszej grze, dlatego tutaj też tak samo jak w klasie Ball będziemy kalkulować krawędzie naszego pionka , po to aby sprawdzać czy nasz pionek nie znajduje się na styku z naszym oknem. Tutaj w przeciwieństwie do klasy Ball nie będziemy odbijać pionka ale będziemy mu zabraniali wyjść poza zadany obszar naszej planszy(rozmiaru naszego głównego okna). Pionek również będzie posiadał taką funkcję jak ustawianie prędkości pionka ale także zmianę i możliwość ustawiania koloru pionka z poziomu aplikacji(funkcja publiczna) . W funkcji update będziemy sterować naszym pionkiem i aktualizować jego pozycję na naszej planszy, sterowania będzie wykonywane za pomocą Key Event z biblioteki SFML. Znajdziemy tutaj dodatkowe funkcje które będą sprawdzać egzystencję naszego Pionka. Jeżeli nasz pionek nie będzie egzystował to będzie. Wywoływana odpowiednia funkcja z Klasy End Menu.

4.4 Meta

Klasa Meta jest podobnie skonstruowana jak klasa Ball czy Pawn, w porównaniu do klasy Pawn jest statycznym obiektem usytuowanym w jednej ściśle określonej pozycji. Tak samo jak klasa Pawn posiada funkcje sprawdzające egzystencję naszej mety. Brak jej egzystencji oznacza zderzenie player'a (obektu klasy Pawn) z obiektem typu Meta. Brak istnienia mety wywołuje pośrednio działanie klasy End Menu. Funkcje takie jak left, right, top, bottom, określają krawędzie naszego obiektu Meta.

4.5 End_Menu

Klasa podobna w budowie do klasy Menu zawiera zestaw etykiet możliwych do wyświetlenia i do wyboru. Do wyboru etykiet służą takie funkcje jak MoveLeft czy MoveRight które pozwalają się przesuwać bo interfejsie submenu za pomocą strzałek lewo prawo. End_menu ma etykiety ułożone horyzontalnie dlatego sposób poruszania jest inny niż w Menu. W odróżnieniu do Menu tutaj są dwie możliwe opcje wyświetlenia etykiet(Menu końcowego) takowego wyświetlenie jest zależne jaki rezultat uzyska nasz player (pozytywny czy negatywny)

4.6 Options

Klasa Options jest to klasa w której pojawiają się label(etykiety) do wyboru funkcjonalności naszej aplikacji. Klasa Options pełni coś w rodzaju submenu. Zdefiniowane funkcje w tej klasie takie jak MoveUp czy MoveDown pozwalają poruszać się po oknie tej klasy za pomocą strzałek góra dół i wybierać jedną z dostępnych etykiet(opcji, funkcjonalności).

4.7 Level_menu

Klasa praktycznie o identycznej budowie jak klasa End_menu, posiada tylko jedno określone zakończenie nie zależne od stanu rozgrywki. Klasa ma za zadanie umożliwić użytkownikowi wybór i zmianę poziomu trudności rozgrywki. Wzmiana label'i jest za pomocą strzałek lewo prawo oraz zatwierdzanie klawiszem enter.

4.8 Music_choise_menu

Klasa nie różniąca się w budowie od klasy Options czy klasy Menu, służy i jest odpowiedzialna za wybór podkładu, ścieżki dźwiękowej.

4.9 Color_Menu

Klasa o identycznej budowie jak klasa Level_Menu czy podobnej budowie co klasa End_Menu. Służy do wyboru koloru naszego player'a (obiektu klasy Pawn).

4.10 Action

Klasa Action odpowiedzialna za budowę całej aplikacji i możliwość przełączania między submenu oraz obsługa wszystkich możliwych okien. Nadzoruje pracę i wywołuje daną klasę w zależności od ustawionej danej flagi. Klasa Action dzieli na poszczególne funkcje działanie danej klasy i to funkcje klasy Action są wywoływane w pliku głównym(JPO2.cpp). Zastosowanie klasy Action kilkukrotnie zmniejsza plik główny.

5. Podręcznik użytkownika (*user's manual*)

5.1 Test Aplikacji na systemie Windows

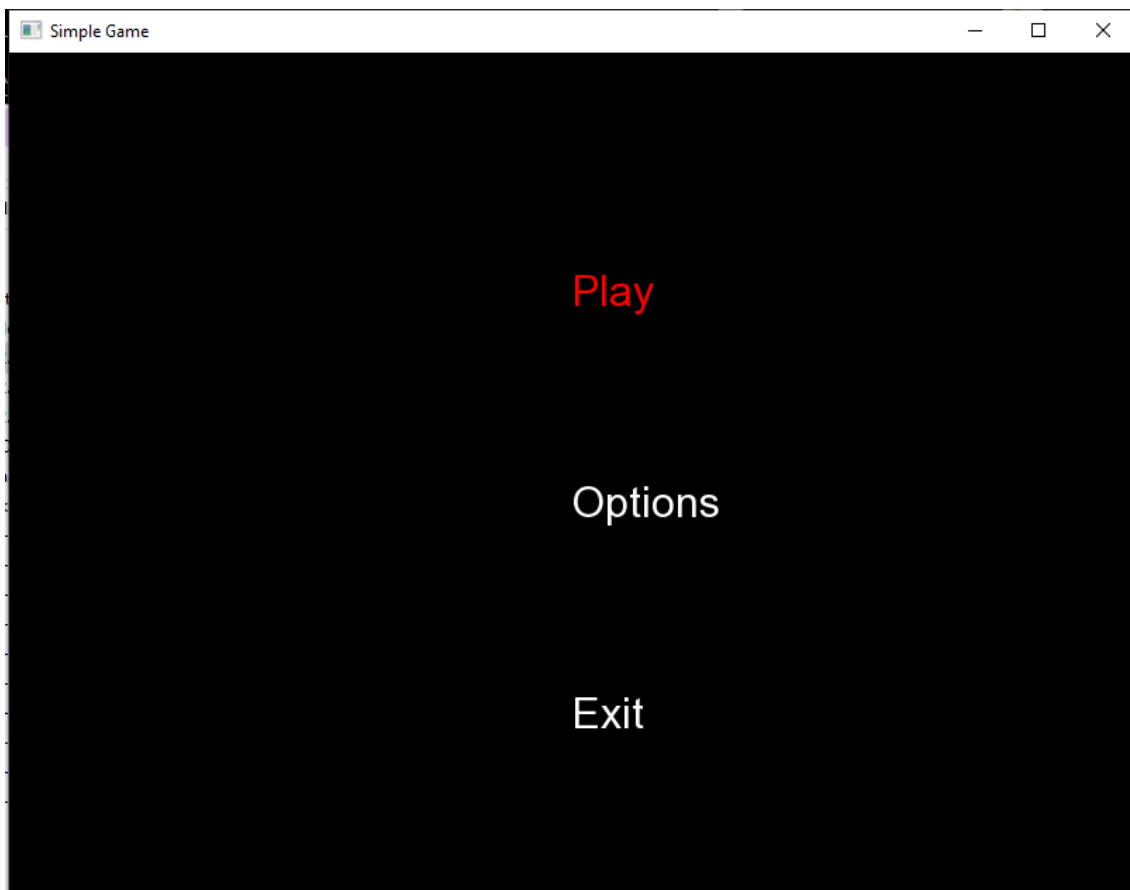
W celu uruchomienia aplikacji SG przechodzimy do pliku z katalogiem gdzie znajduje się nasza aplikacja i klikamy na nią podwójnie lewym klawiszem myszki.

Z punktu widzenia linii komend można to wykonać w następujący sposób:

```
C:\Users\Adamek Mateusz\Desktop\JP0\po_action\rtf\New folder\Simple_Game_Adamek_Mateusz>SimpleGame.exe
```

Czyli przejdźcie do folderu z grą oraz uruchomienie w powyższy sposób pliku o rozszerzeniu .exe

Gra musi być uruchamiana wraz z pozostałymi plikami zawartymi w katalogu, uruchomienie gry bez tych plików nie będzie możliwe. Aby uruchomić grę nie będąc w tym katalogu należy stworzyć "shortcut" do pliku .exe i umieścić go w dogodnej lokalizacji.



5.2 Sposoby interakcji między użytkownikiem a grą

- **Poruszanie się po menu**

Do poruszania się w grze wykorzystujemy strzałki znajdujące się na klawiaturze. Poruszanie się w aplikacji jest zależne od sposobu rozmieszczenia elementów (obiektów). Rozmieszczenie horyzontalne obiektów jest równoznaczne z używaniem strzałek lewo-prawo w celu poruszania się po aplikacji. Ułożenie wertykalne charakteryzuje się tym że będziemy używać strzałek góra-dół w celu przemieszczania się po aplikacji.

- **Interakcja w samej grzej**

W grze w celu omijania obiektów używamy wyłącznie strzałek znajdujących się na klawiaturze.

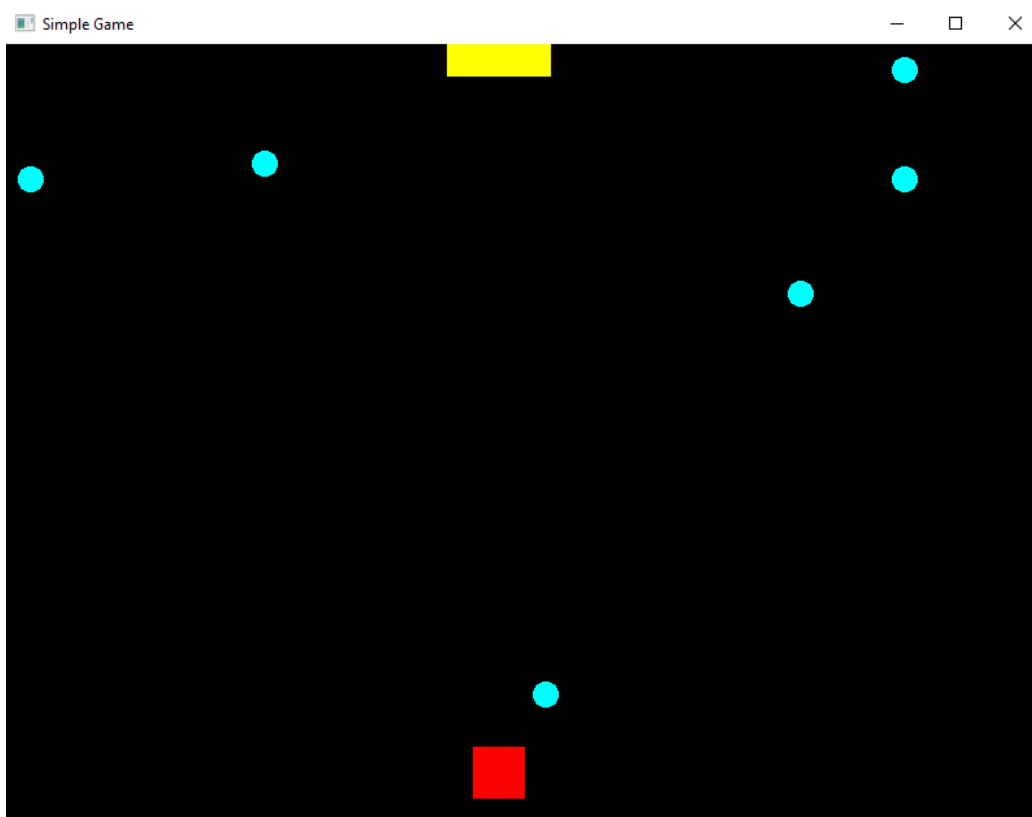
- **Klawisz ESC – służy do całkowitego wyjścia i zamknięcia gry.**
- **Klawisz Enter służy do zatwierdzania.**





5.3 Akcje związane z rozgrywką

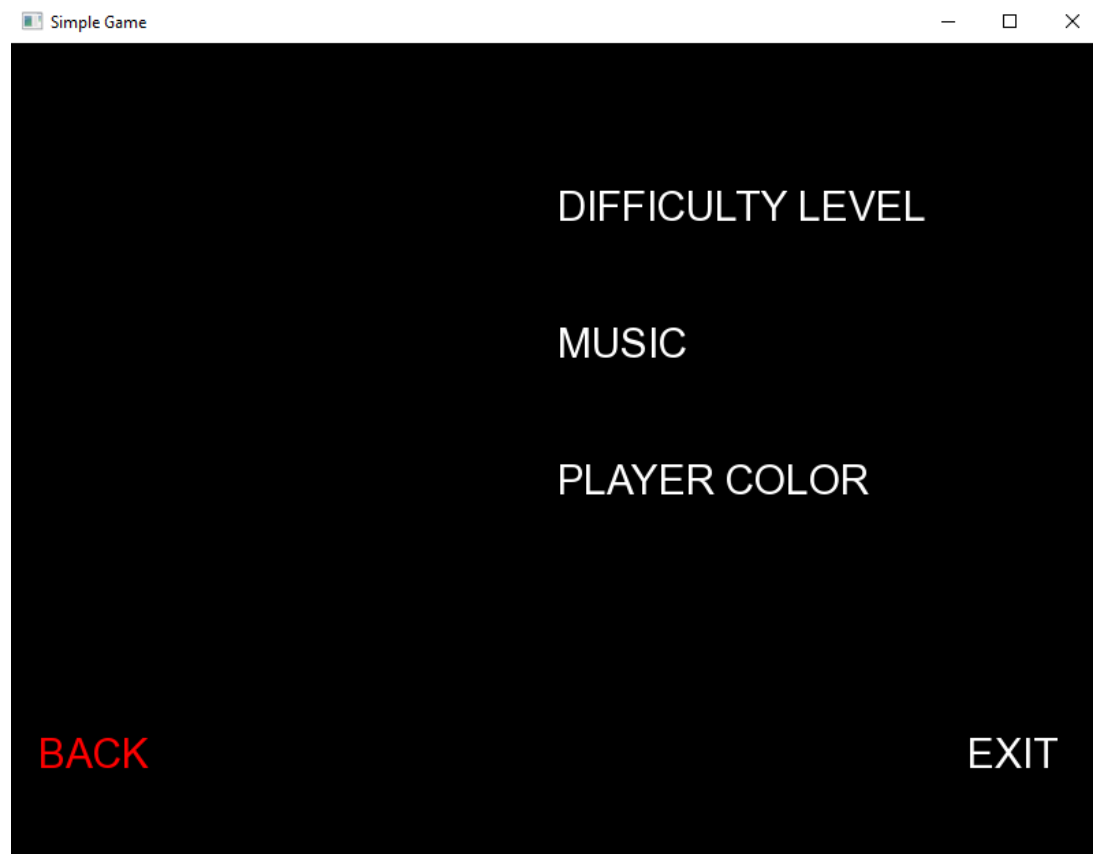
W celu rozpoczęcia gry, należy tak przesunąć strzałkami aby przycisk 'Play' zapalił się na kolor czerwony, w taki sposób dokonujemy wyboru. Taki wybór należy jeszcze zatwierdzić klawiszem enter.

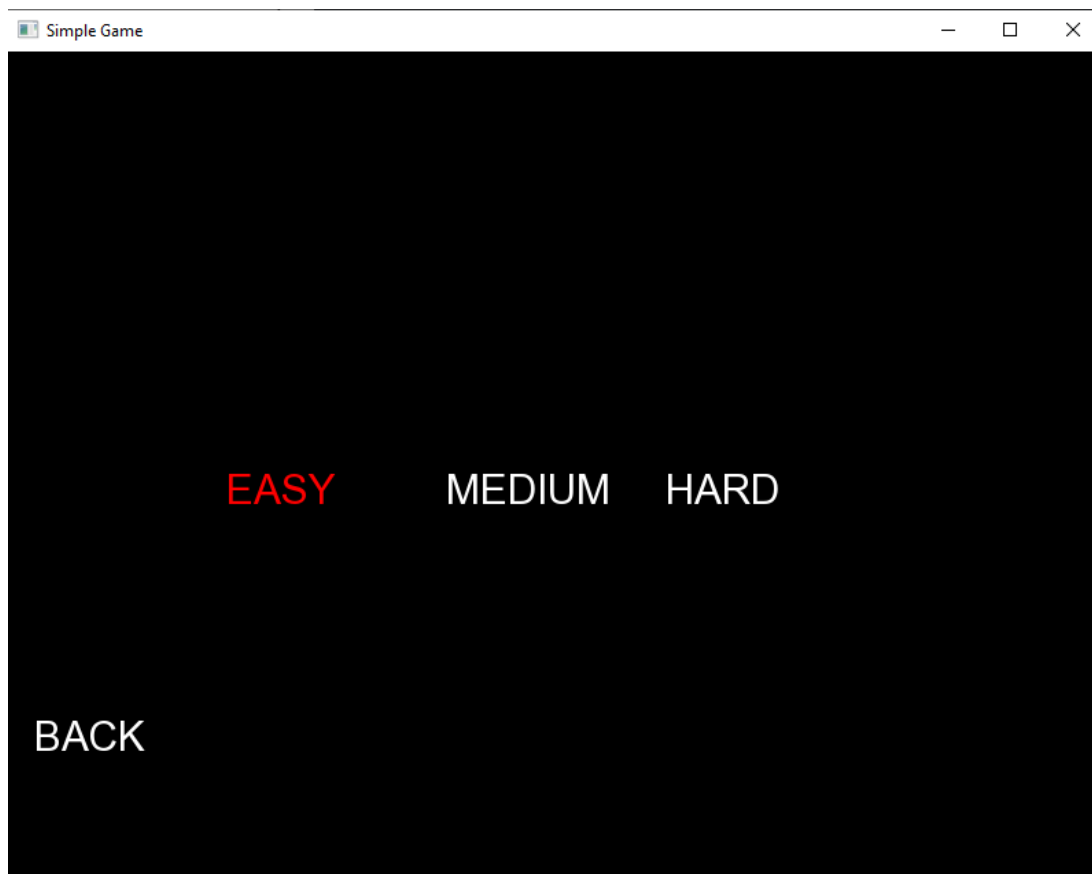


5.4 Wybór Funkcjonalności

Podstawowe Funkcjonalności

Poniżej opisze przykładowy wybór jednej z funkcjonalności jaka jest zmiana poziomu trudności. Przechodzimy strzałką na etykietę Options wybieramy ją zatwierdzając klawiszem enter, Później z sub-menu (menu wyboru naszych funkcjonalności, zmiany stanów rozgrywki) wybieramy etykietę "DIFFICULTY LEVEL" i zatwierdzamy klawiszem enter. Po zatwierdzeniu pojawi nam się kolejne sub-menu w który będziemy mieć możliwość wybrania naszego poziomu trudności pomiędzy EASY, MEDIUM, HARD. Domyślnie ustawiony poziom trudności to MEDIUM, wybór jednego z dwóch pozostałych spowoduje zmianę szybkości poruszania się piłek w grze. Nasz wybór zatwierdzamy klikając klawiszem enter.





Wyjście, zakończenie gry

Grę można zakończyć wybierając z "Main Menu" opcję "Exit" i zatwierdzając ją klawiszem enter. Istnieją też inne metody wyjścia z aplikacji. Można to wykonać wciskając klawisz ESP lub zamykając okienko klikając na X

Bibliografia

- [1] <https://www.sfml-dev.org/documentation/2.5.1/>
- [2] <https://www.youtube.com/watch?v=4Vg9d1pjL20&t=3s>
- [3] <https://www.youtube.com/watch?v=JlAd3X3PX6o&list=PLk6mhiZKpyW4KRTZc8sc0aYOLFmTSLA7r>

Pliki projektowe

<https://github.com/AdamekMateusz/JPO2>

<https://drive.google.com/file/d/15MC01hYJeiW13sGZXkhne20uiXL9R-EY/view?usp=sharing>