

Politechnika Śląska
Wydział Matematyk Stosowanej
Kierunek Informatyka

Gliwice, 20.01.2022

Programowanie I
projekt zaliczeniowy
"Breakout"

Mateusz Sawosz gr.3 lab. 6

1. Opis projektu.

Remake kultowej gry Breakout, wyprodukowanej i wydanej przez Atari. Gra polega na odbijaniu piłki w taki sposób by nie dotknęła ona dolnej krawędzi okna, jednocześnie niszcząc jak największą ilość cegiełek.

2. Wymagania

1. Mechanika gry

- Poruszanie platformą*
- Fizyka odbijania się piłki*
- Generowanie cegiełek oraz niszczenie/ uszkodzanie ich w momencie kolizji z piłką*
- Przerywanie gry w momencie zetknięcia się z dolną krawędzią*
- Przerwanie gry z rezultatem wygranej przy zniszczeniu wszystkich cegiełek*

2. Funkcjonalności aplikacji

- Menu rozpoczęcia gry*
- Menu pozwalające uruchomić grę ponownie po przegranej*
- Zliczanie punktów w trakcie gry za zniszczenie cegiełki*
- Sygnał dźwiękowy w momencie zetknięcia się piłki z krawędziami oraz cegiełkami*
- Menu pozwalające regulować głośność sygnałów dźwiękowych, prędkości i wielkości piłki oraz platformy za pomocą "suwaków"*
- Ikona pliku .exe, oraz ikona w lewym górnym rogu aplikacji*

- 3 stany kursora:

- Fioletowy - bez interakcji

- Pomarańczowy - po najechnięciu na przyciski

- Zielony - po kliknięciu przycisku

- Ograniczenie możliwości rozszerzania okna, wymuszone brakiem skalowalności przygotowanych grafik

3. Przebieg realizacji

Wykonawca opisuje wykonane przez siebie zadania. Należy zamieścić opis plików z których składa się projekt, opis algorytmu, gdy program jest związany z algorytmiką. W przypadku korzystania z zewnętrznych bibliotek należy je tu krótko opisać (do czego służą, z jakich funkcji się korzystało)

Bricks.cpp - funkcja generująca cegiełki w 4 różnych kolorach

Score.cpp - funkcja zliczająca aktualną ilość punktów zdobywanych poprzez niszczenie cegiełek

MaxScore.cpp - funkcja obliczająca maksymalną ilość punktów możliwych do zdobycia w trakcie gry

Main.cpp - główna funkcja programu, realizująca wyświetlanie okna, zawierająca 5 stanów gry (menu, game, gameover, won, options)

Main.h - plik z deklaracjami funkcji oraz z dołączonymi bibliotekami takimi jak:

- SFML/Graphics.hpp - odpowiadająca za obiekty graficzne w aplikacji

- SFML/Window.hpp - odpowiadająca za wyświetlanie okna aplikacji

- SFML/System.hpp - odpowiadająca za wiele przydatnych funkcji podczas pisania aplikacji

- SFML/Audio.hpp - odpowiadająca za odtwarzanie plików dźwiękowych

Przez cały czas pracy nad projektem korzystałem z bardzo wielu funkcji i metod, stąd pozwolę wypisać sobie tylko niektóre z nich, oczywiście na temat każdej funkcji mam potrzebną wiedzę w jaki sposób działa(w odniesieniu do jej wykorzystania w moim projekcie).

-loadFromFile - wczytywanie z pliku obrazka jako texture

-.setTexture o .setTextureRect, pierwsza służy do ustawienia obrazka jako texture, druga zaś do "wycięcia" z pliku posiadającej więcej textur, tej konkretnej której potrzebujemy - za pomocą pozycji X, Y od której chcemy zacząć wycinać, oraz jaką długość i szerokość ma mieć wycięta textura

-.setSize i .setPosition - ustawia wielkość obiektu, kolejna zaś jego pozycję na ekranie

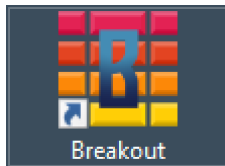
-to_string() wykorzystana do zamiany int na string(w celu wyświetlania ilości punktów na ekranie)

-.draw() - wyświetla na ekranie obiekty

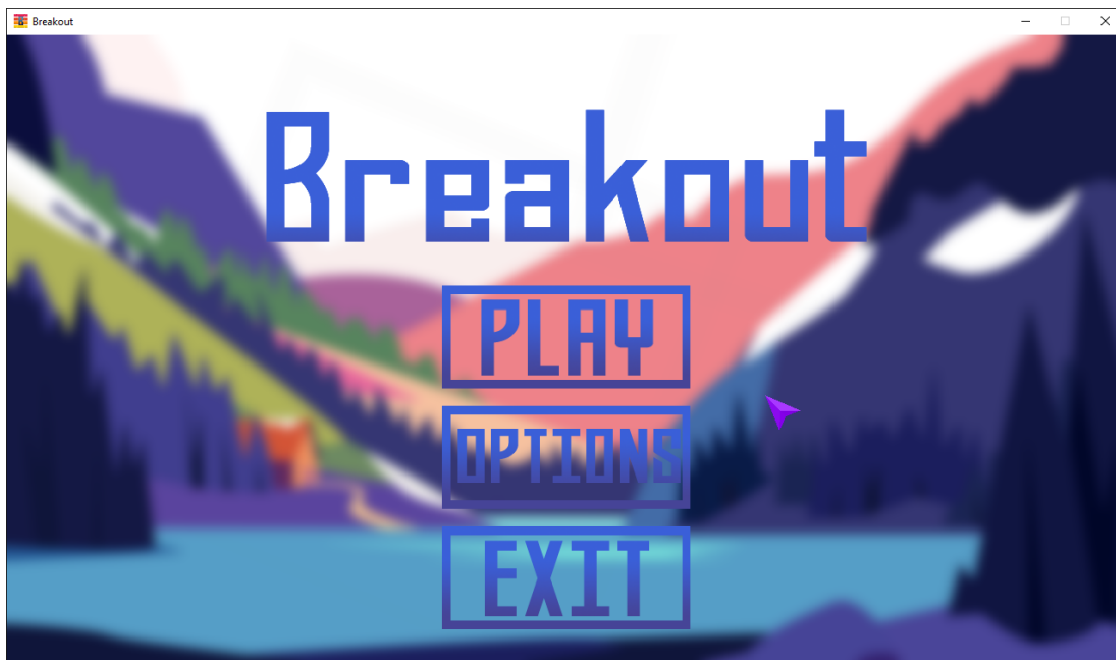
-.play() - odpowiada za odtwarzanie pliku dźwiękowego w trakcie działania aplikacji

-.clear() - czyszczenie ekranu

4. Instrukcja użytkownika



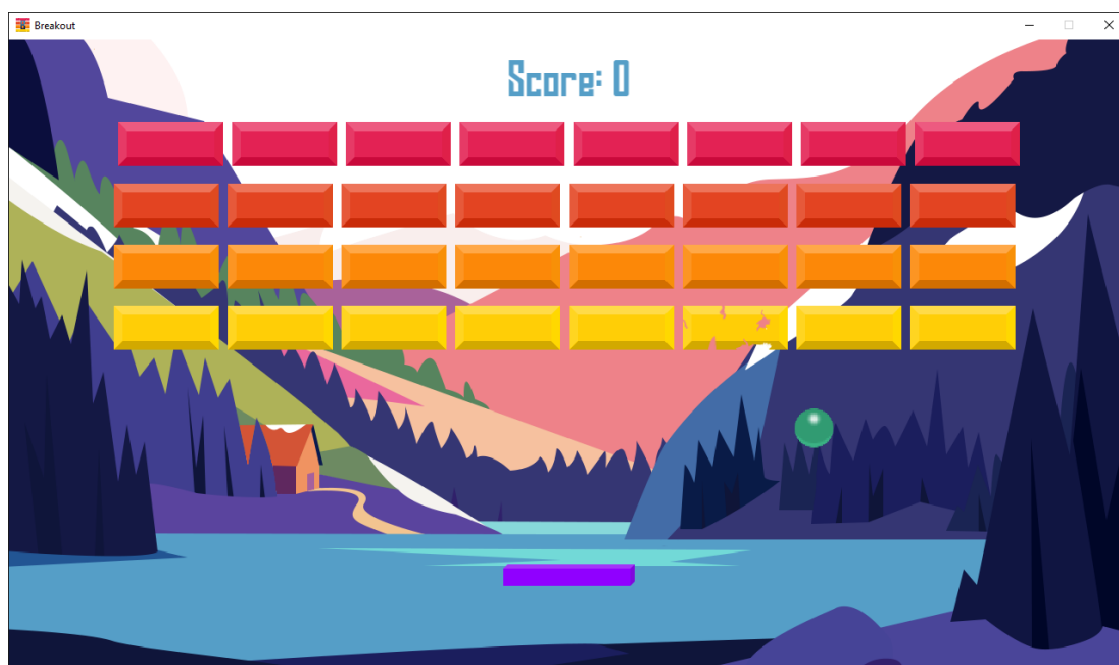
Ekran startowy po uruchomieniu aplikacji, pozwala rozpocząć grę, przejść do ustawień, bądź wyłączyć aplikację.



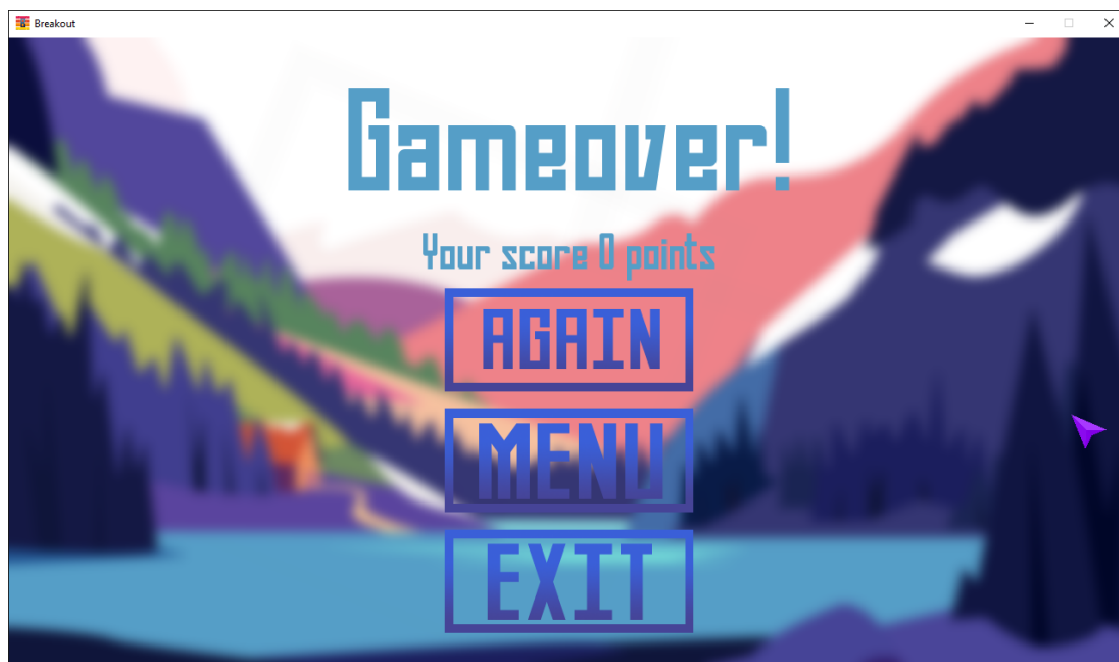
Options - opcje - pozwalają ustawić głośność, oraz modyfikować ustawienia rozgrywki takie jak wielkość piłki, jej prędkość oraz prędkość platformy



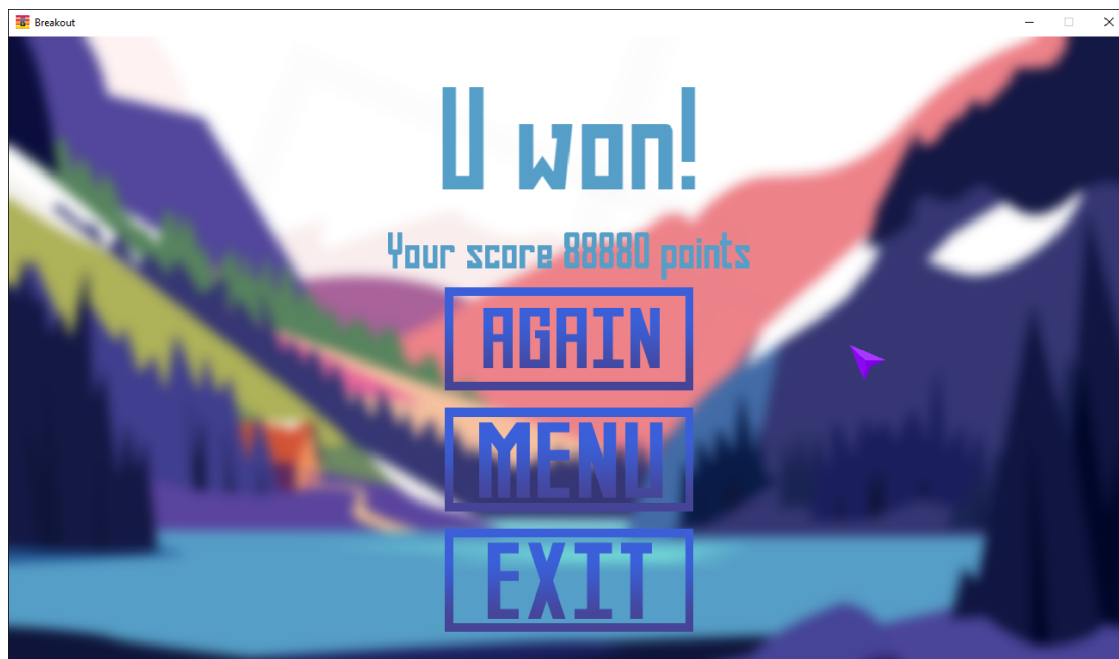
Play - uruchomienie gry - platformą służącą do odbijania piłki można poruszać za pomocą A i D, bądź strzałek Lewa i Prawa



Game over - wyświetlane w przypadku znalezienia się piłki w pozycji niemożliwej do odbicia za pomocą platformy



Won - ekran informujący o wygranej



5. Podsumowanie i wnioski.

W związku z tym, że była to moja pierwsza aplikacja okienkowa początkowo stało przede mną wiele wyzwań i problemów wynikających z brakiem wystarczającej wiedzy na temat SFML, lecz po spędzeniu wymaganego czasu czytając dokumentację, zapoznając się z poradnikami dostępnymi na stronie <https://www.sfml-dev.org/tutorials> oraz wieloma filmami na Youtube szczerkawo zdobywana wiedza zaczęła składać się w jedną całość.

Jednym z bardziej zapamiętanych "problemów", było generowanie cegiełek, początkowo każdą cegiełkę generowałem indywidualnie, lecz było to niezbyt optymalne. W tym momencie pomyślałem, że `sf::RectangleShape` jest niczym innym jak zmienną, taką jak `int` czy `float`, więc dopisałem kwadratowe nawiasy, licząc że utworzę w ten sposób całą tablicę prostokątów, które później staną się cegiełkami. Rzecz na ogół oczywista, ale na wstępnym etapie bardzo mnie to ucieszyło, że wpadłem na coś takiego bez żadnej pomocy.

Kolejnym irytującym problem była orientacja całej aplikacji, tj. punkt (0,0) znajduje się w lewym górnym rogu i idąc "w dół" wzdłuż osi OY argumenty się zwiększają, a nie zmniejszają jak występuje to w układzie kartezjańskim. Przypuszczam że jest tak z powodu, iż monitory wyświetlają poszczególne klatki od góry do dołu. Błąd był irytujący z powodu wielu błahych błędów - spowodowanych przyzwyczajeniem do tradycyjnego układu kartezjańskiego przy kompilacji, przez co niejednokrotnie musiał spędzić miło czas z debbugerem szukając dlaczego piłka przelatuje przez cegiełkę, skoro przecież "dobrze" ustawiłem hitboxy.

Jeśli chodzi o dalszy rozwój, to raczej takowego nie będzie, początkowo chciałem zamieścić jeszcze tablicę wyników wyświetlającą 10 najlepszych wyników, polegającą na tym że użytkownik ma możliwość wprowadzenia swojej nazwy, a następnie po grze w pliku tekstowym zapisywana jest jego nazwa wraz z ilością zdobytych punktów. Jednak dopadło mnie już znudzenie tym projektem, sama implementacja tablicy wyników, nie byłaby niczym wymagającym, więc postanowiłem zakończyć ten projekt i zacząć pisać coś innego, bardziej

rozbudowanego w wolnym czasie operując już obiektami (Gra platformowa typu Mario, z umiejętnościami niczym z MMO RPG`a).