Politechnika Śląska

Wydział Matematyk Stosowanej

Kierunek Informatyka

Gliwice, 25.01.2023

Programowanie I

**projekt zaliczeniowy**

**"*Icy RMS*"**

**Oskar Chruściński gr. lab. 3/5**

**1. Opis projektu.**

*Projekt to prosty klon popularnej gry Icy Tower, z ograniczoną funkcjonalnością w porównaniu do oryginału. Gracz za pomocą klawiatury porusza się po nieskończonej wieży skacząc po losowo generowanych platformach. Wraz z postępem w grze zwiększa się poziom trudności poprzez zwiększenie prędkości gracza. Rozgrywka kończy się wraz z spadnięciem gracza poza mapę. Gra posiada dodatkowo tryb dla dwóch graczy i zapisywanie do pliku 10 najlepszych wyników.*

**2. Wymagania**

* *Główne menu z możliwościami odpalenia gry dla jednego gracza, dwóch graczy, wyświetlenia tablicy wyników oraz wyjścia z gry*
* *Graficznie zrobiona gra, polegajaca na skakaniu w górę wieży po losowo generowanych platformach*
* *Zwiększający się wraz z postępem w grze wynik oraz stopień trudności*
* *Zakończenie rozgrywki kiedy gracz spadnie poza ekran gry*
* *Tryby gry dla jednego gracza i dla dwóch graczy*
* *Po zakończonej grze gracz może wpisać swój nick, który, jeżeli gracz osiągnął odpowiednio wysoki wynik, zostanie wpisany do pliku, skąd będzie wyświetlony na ekranie podczas przeglądania tablicy wyników*

**3. Przebieg realizacji**

*Projekt składa się z trzech głównych plików programowych:*

* *main.cpp - główny szkielet gry*
* *functions.h - spis funkcji użytych w programie*
* *functions.cpp - ciała funkcji z pliku .h*

*Dodatkowo pliki graficzne, czcionka oraz plik scores.txt z top10 wyników.*

*Została zastosowana biblioteka SFML. Główne funkcje użyte w programie:*

* *Obsługa tekstur i sprite'ów*
* *Obsługa tekstu i czcionki*
* *Obsługa klawiatury i myszki*
* *Generowanie okna graficznego, rysowanie na nim sprite'ów, pisanie tekstu*

*Projekt może znajdować się w trzech głównych stanach - Główne menu, gra oraz tablica wyników. Do sterowania tym służy enum gamestate.*

*Wszystkie przyciski na głównym menu oraz na tablicy wyników opierają się na wykryciu kliknięcia myszy oraz porównania koordynatów kursora z umiejscowaniem przycisków. Wciśnięcie odpowiedniego przycisku zmienia stan gry.*

*Ekran tablicy wyników pobiera z pliku wiersze jeden po drugim oraz odpowiednio je formatuje i wypisuje na ekran. Jeżeli do tego ekranu trafiliśmy poprzez zakończenie rozgrywki to dodatkowo użytkownik musi podać nickname, który zostanie wpisany wraz z wynikiem do pliku na odpowiednie miejsce (o ile wynik jest większy niż najniższy wynik w pliku).*

*Tryb gry rysuje zamiast menu ekran rozgrywki, składający się z tła, dwóch ścian ograniczających planszę po bokach, platform oraz gracza (lub graczy w wypadku trybu dla dwóch graczy). Program wychwytuje zdarzenia z klawiatury i odpowiednio zmienia pozycje x i y konkretnego gracza. Kiedy gracz znajdzie się powyżej połowy planszy ekran zacznie się "obniżać", a raczej tworzyć iluzje obniżania się poprzez obniżenie platform oraz samego gracza. Z każdym obniżeniem planszy zwiększa się wynik gracza. Kiedy platforma znajdzie się poza planszą od dołu, to na jej miejsce zostanie wygenerowana kolejna na górze oraz zostanie nieco zwiększony poziom trudności. Gra kończy się w momencie spadnięcia gracza poza planszę na dole.*

*Na uwagę zasługuje tutaj mechanika skakania. Opiera się ona na flagach jumping oraz falling i dodatkowych zmiennych maxHeight, jumpStart oraz jumpMultiply. Domyślnie gracz ma obie flagi ustawione na false. Kiedy w takim stanie wciśnie W (lub strzałkę w górę dla drugiego gracza) to flaga jumping zostanie ustawiona na true. Zmienne jumpStart i maxHeight pozwalają sprawdzają, czy gracz nie doskoczył już na maksymalną wysokość i jeżeli doskoczy to ustawia flagę falling na true. Tak samo dzieje się po puszczeniu W lub strzałki. Flaga falling sprawdza możliwość wylądowania na jakiejś platformie, co gdy nastąpi resetuje wszystkie flagi i wartości związane ze skokiem i ustawia gracza na platformie (z lekkim marginesem błędu, co jest dość powszechnym zjawiskiem w platformówkach z tego co mi wiadomo), w przeciwnym wypadku kontynujqc spadanie (zmiana koordynatu y).*

**4. Instrukcja użytkownika**

*Główne Menu: klikanie lewym przyciskiem myszy na przyciski.*

*Gra (gracz 1/gracz 2): ruch w lewo: a/strzałka w lewo; ruch w prawo: d/strzałka w prawo; skok: w/strzałka w górę*

*Ekran wyników: kliknięcie na przycisk w prawym dolnym rogu powraca do menu; po zakończeniu gry poprzez klawiaturę wpisywany zostaje nick użytkownika, backspace kasuje ostatni znak, enter zatwierdza*

**5. Podsumowanie i wnioski.**

*Największe wyzwanie stanowiła mechanika skoku. Udało mi się stworzyć w miarę płynne poruszanie się postaci w rozsądnym tempie, z trybem dla dwóch graczy I względnie celem do którego można w grze dążyć (osiągnięcie jak najwyższego wyniku). Nie zrealizowałem znanego z oryginału "odbijania się od ścian" skutkującego nabieraniem prędkości. Niewielkim problemem było też zrealizowanie wpisywania nicku użytkownika do tablicy wyników i na ekran, ale udało się.*

*W dalszych krokach kod moim zdaniem zasługuje przede wszystkim na poważny rewrite z uwzględnieniem bardziej poprawnego szkieletu gry (funkcje Update(), większe rozdrobnienie na funkcje, podejście obiektowe). Z mechanik można dodać odbijanie się od ścian, zmianę scenerii wraz z postępem, animacje. Pomyśleć można również nad upgrade'm graficznym.*