Politechnika Śląska

Wydział Matematyk Stosowanej

Kierunek Informatyka

Gliwice, 29.01.2022

Programowanie I

**projekt zaliczeniowy**

**"*Uniki*"**

**Paweł Dyśko gr. lab. 2/3**

**1. Opis projektu.**

,,Uniki” to jest gra polegająca na omijaniu przeszkód. Gracz może poruszać się w pionie(góra/dół) aby uniknąć zderzenia się z wygenerowanymi przeszkodami.

**2. Wymagania**

* Program tworzy menu, w którym do wyboru gracza są następujące opcje: 1.Graj 2.Opis gry i instrukcja 3.Wybór poziomu trudności 4.Wyjdź.
* Gracz może dowolnie zmienić domyślny poziom trudności.
* Program generuje planszę, na której ustawiony zostaje ‘pionek’ gracza.
* Gracz za pomocą przycisków ‘W’/’S’/strzałki do góry/strzałki do dołu steruje swoim pionkiem w pionie.
* Program generuje i losowo ustawia przeszkody, które następnie poruszają się w stronę gracza.
* W momencie, gdy gracz uderzy w przeszkodę cała plansza zmienia kolor na czerwony i jeśli gracz ma ochotę, może zagrać ponownie bez wychodzenia do menu.

**3. Przebieg realizacji**

Pliki tools (.h i .cpp) służyły, żeby umieścić w nich funkcje do losowania liczb pseudolosowych. Nie zajmują dużo miejsca ale wolałem mieć do nich łatwy i szybki dostęp.

W plikach app (.h i .cpp) umieściłem funkcje odpowiedzialne za działanie okna gry oraz funkcje, które upraszczały mi przypisywanie działań do poszczególnych klawiszy.

W pliku config.h umieściłem zmienne odpowiedzialne za wielkość okna oraz trudność gry, ponieważ chciałem je mieć w łatwodostępnym miejscu, żeby w razie potrzeby ich zmiany nie musieć ich szukać w kodzie.

Plik uniki.cpp jest moim plikiem głównym, w którym tworzyłem klasy i funkcje odpowiedzialne za stworzenie działanie menu, planszy gry i samej gry. W tym pliku oczywiście znajduje się również funkcja main.

Do stworzenia projektu użyłem bibilioteki zewnętrznej SFML. Jest to biblioteka graficzna do C++, która pozwala tworzyć i wyświetlać w oknie gry lub grafiki 2D.

**4. Instrukcja użytkownika**

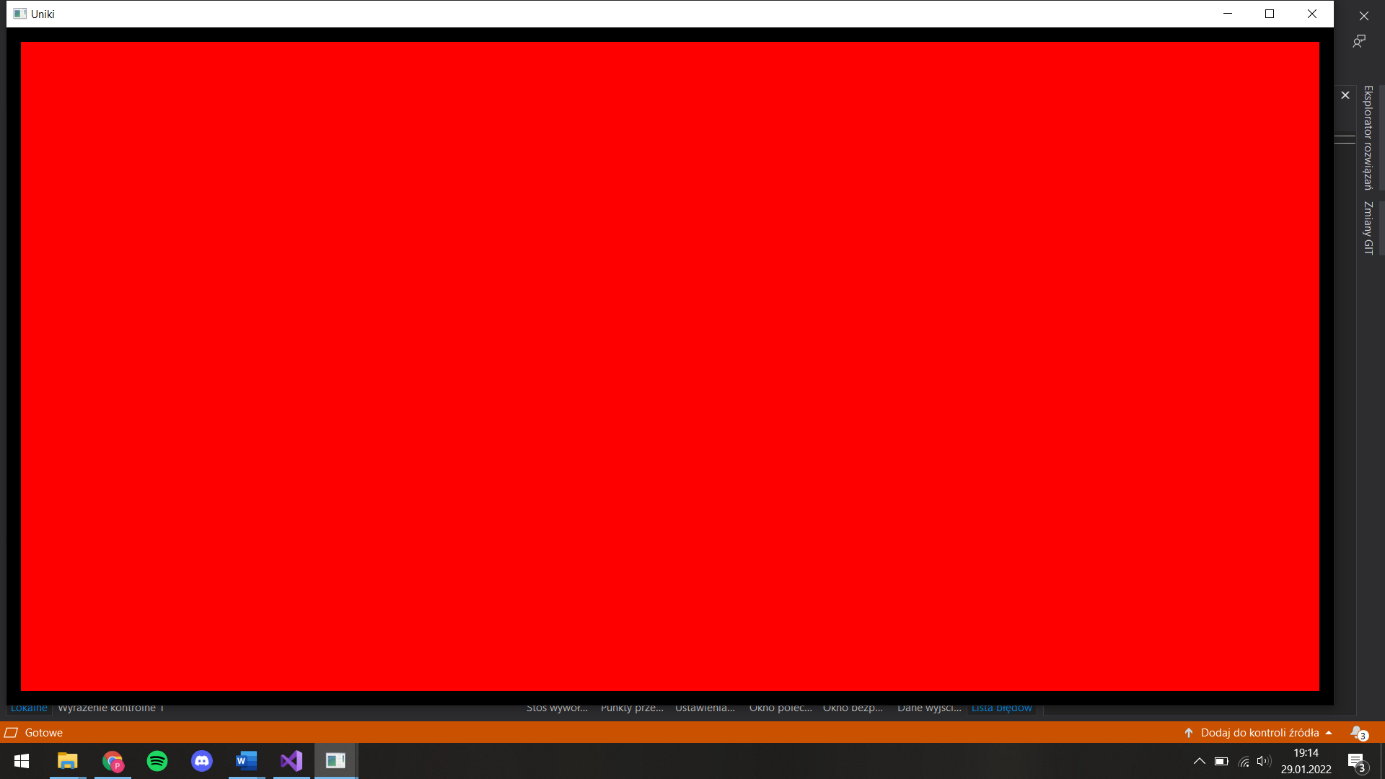
**Obraz zawierający tekst

Opis wygenerowany automatycznie**

Będąc w menu gracz podaje liczbę i klika Enter.

Obraz zawierający tekst, zegar

Opis wygenerowany automatycznie

Będąc już w grze gracz porusza się w pionie za pomocą przycisków ‘W’/’S’/strzałki do góry/strzałki do dołu aby uniknąć zderzenia się z przeszkodami. W momencie gdy gracz się zderzy z przeszkodą, cała plansza zmieni kolor na czerwony:

Aby powrócić do gry należy kliknąć Backspace. Gracz może również w dowolnej chwili ‘pokolorować’ planszę na czerwono, żeby na przykład przestać na chwilę grać bez wychodzenia do menu. Może to zrobić naciskając Backspace, a powrót do gry jest taki sam jak w przypadku porażki. Aby wyjść do menu, gracz w dowolnej chwili gry lub będąc w sytuacji czerwonej planszy, musi kliknąć Esc lub krzyżyk w prawym górnym rogu okienka. Aby zamknąć program należy będąc w menu wybrać numer 4.

**5. Podsumowanie i wnioski.**

Podczas tworzenia projektu napotkałem problemy ze swtorzeniem okienkowego menu. Na chwilę obecną zastąpiłem je menu konsolowym, ale w przyszłości będę starał się stworzyć menu okienkowe, po którym poruszałoby się nie za pomocą klawiszy lecz z użyciem myszki. Dodatkowo będę chciał spróbować zmienić sterowanie w grze ze sterowania klawiszami na sterowanie myszką (tak aby pionek gracza podążał za kursorem myszki). Z rzeczy, które udało mi się zrealizować to sposób działania gry, który sobie założyłem z początkiem pisania programu oraz funkcje które ułatwiają mi przypisywanie działania dla poszczególnych klawiszy. Szczególnie jestem zadowolony z funkcji bindKeyboard(), która nie jest wyjątkowo skomplikowana, ale pozwala mi przypisywać wszystkim klawiszom im działania w szybki i prosty sposób oraz, co bardzo ważne, w jednym miejscu.Obraz zawierający tekst

Opis wygenerowany automatycznie

**Obraz zawierający tekst, monitor, zrzut ekranu, ekran

Opis wygenerowany automatycznie**

Dzięki temu w głównej funkcji main wystarczy, że wywołam funkcję bind().