

Programowanie Komputerów 4

Projekt semestralny

05.06.2020

Prowadzący:

mgr in. Krzysztof Pasterak

Autorzy:

Kamil Machulik

Patryk Lipka

Zawartość sprawozdania:

0. Strona tytułowa (autor, tytuł, data, prowadzący)
1. Temat
2. Analiza tematu (doprecyzowanie tematu, uzasadnienie wyboru klas, algorytmów, bibliotek, itp.)
3. Specyfikacja zewnętrzna (instrukcja dla użytkownika, przykłady działania, zrzuty ekranu)
4. Specyfikacja wewnętrzna (klasy [znaczenie obiektu, powiązania z innymi klasami, istotne pola i metody], diagram hierarchii klas, istotne struktury danych i algorytmy, wykorzystane techniki obiektowe, ogólny schemat działania programu)
5. Testowanie i uruchamianie (obowiązkowo, wykrycie i usunięcie dużej liczby błędów nie obniża oceny)
6. Uwagi i wnioski (dowolne, do tematu, programu, prowadzącego, zajęć itp., punkt nieobowiązkowy)

1. Temat:

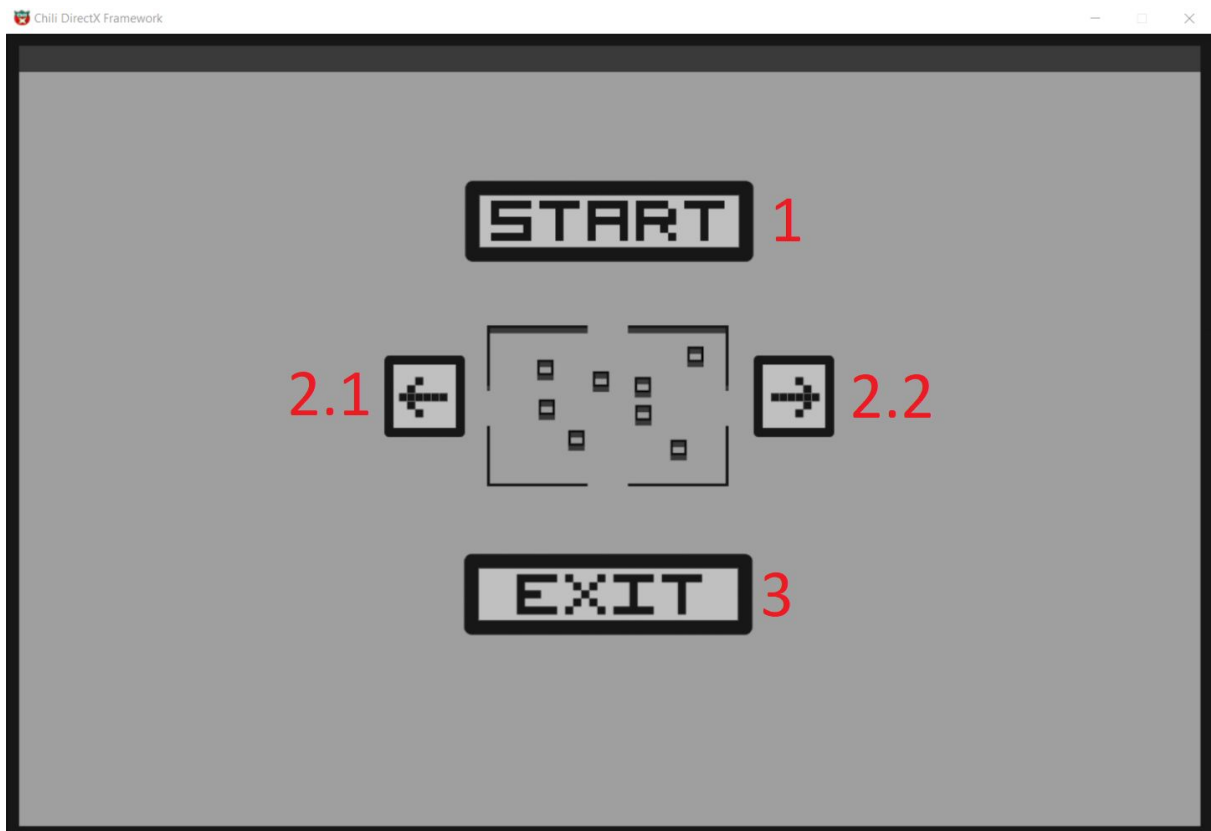
Jako projekt semestralny zaprojektowaliśmy i zaimplementowaliśmy grę polegającą na przetrwaniu jak największej liczby rund wśród wrogich postaci.

2. Analiza tematu:

Gra polega na obronie przed atakiem zombie, zdobywaniu punktów oraz przechodzeniu do kolejnych rund, w których odblokowywane są przez nas kolejne bronie i ulepszenia. Do stworzenia gry wykorzystaliśmy framework Chili Framework DirectX, która zapewnia takie funkcje jak obsługa okna, myszy, klawiatury, dźwięku i obrazu oraz zewnętrzną klasę loadpng do dekodowania grafik w formacie png. Głównymi klasami w naszej grze są klasy Player, Board, Weapon, Enemy.

3. Specyfikacja zewnętrzna:

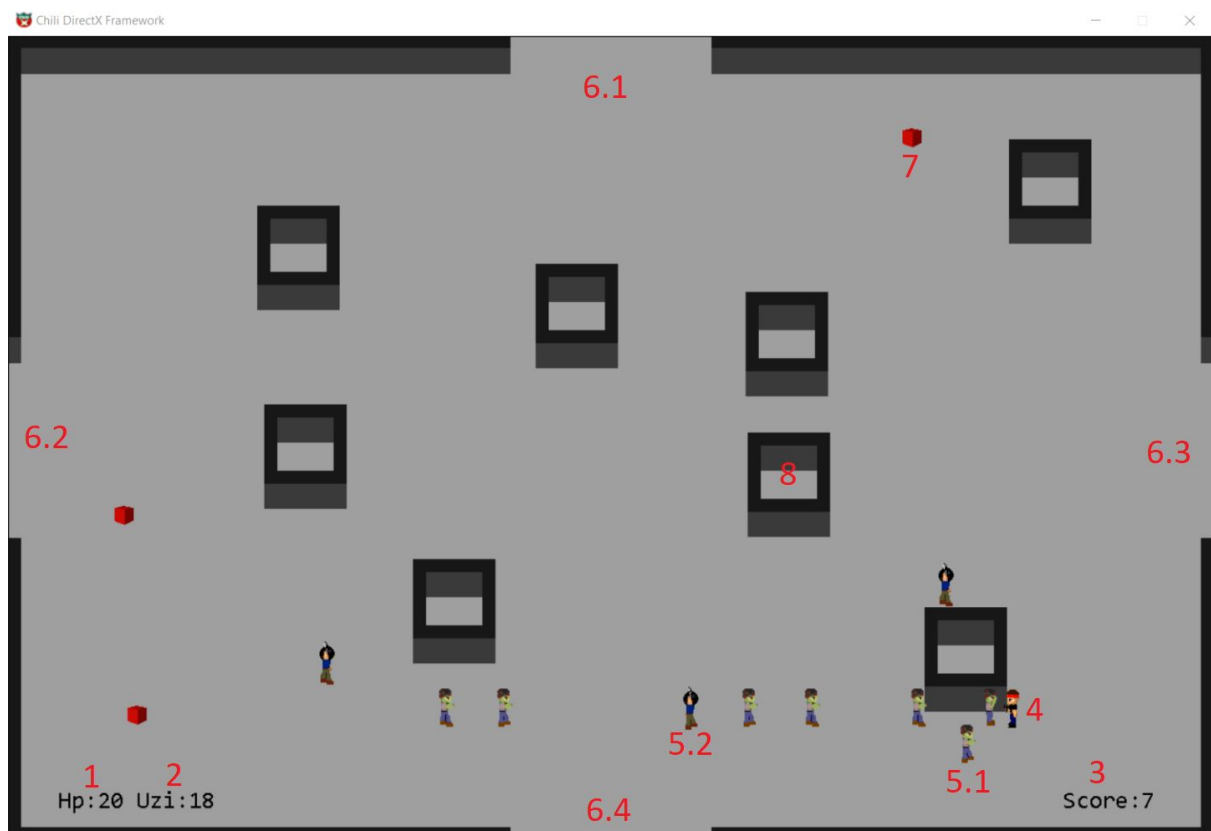
Po włączeniu gry ukazuje nam się menu obsługiwane myszką:



Opis dostępnych przycisków:

1. Uruchamia grę na aktualnie wybranej mapie
2. Przyciski do zmian mapy wraz z miniaturką aktualnie wybranej mapy
 - 2.1 Zmienia mapę na poprzednią
 - 2.2 Zmienia mapę na następną
3. Wyjście z gry

Ekran gry:



Opis poszczególnych elementów:

1. Zdrowie gracza
2. Aktualnie wybrana broń wraz z jej liczbą amunicji
3. Wynik uzyskany w aktualnej grze
4. Gracz
5. Wrogowie:
 - 5.1 Zombie
 - 5.2 Bomber
6. Miejsca z których przeciwnicy wchodzą na mapę
7. Skrzynka, którą Gracz może zebrać
8. Przeszkoda

Sterowanie:

Chodzenie- klawisze strzałek

Strzelanie- spacja

Zmiana broni- klawisz z zmienia broń na poprzednią, a x na następną (bronie należy odblokować)

Pauza- klawisz escape

Grę wyłączamy klikając na przycisk zamknięcia okna.

W grze uzyskujemy 1 punkt za każdego zabitego przeciwnika typu Zombie oraz 5 punktów za każdego zabitego przeciwnika typu Bomber (zabijanie ich poprzez podchodzenie do nich nie daje nam punktów). Po zabiciu wszystkich przeciwników w danej rundzie przechodzimy do rundy następnej. Przechodzenie do kolejnych rund umożliwia nam używanie nowych broni, a także ulepszanie aktualnie posiadanych broni.

Bronie zaimplementowane w grze:

1. Pistolet- broń podstawowa mająca nieskończoną ilość amunicji ale oprócz tego nie wyróżnia się niczym szczególnym. Wraz z rundami ulepsza się jego szybkostrzelność oraz obrażenia.
2. Uzi- pistolet maszynowy strzelający znacznie szybciej od podstawowego pistoletu. Bardzo szybko zabija przeciwników, lecz równie szybko zużywa swój zapas amunicji. Wraz z rundami ulepsza się jego szybkostrzelność oraz obrażenia.
3. Shotgun- strzelba, która strzela powoli, lecz atakuje rozszerzającymi się atakami. Wraz z rundami ulepszają się jego obrażenia oraz liczba wiązek strzałów przy jednym pociągnięciu za spust.
4. Snajperka- karabin snajperski strzelający najwolniej w całej grze, jednakże posiada on duże obrażenia i może przebijać przez przeciwników. Wraz z rundami zwiększa się liczba przebijanych przeciwników, szybkostrzelność oraz obrażenia.

Ekran śmierci:



5. Specyfikacja wewnętrzna:

Klasa Player:

Główna klasa reprezentująca postać gracza w grze oraz służąca do przetwarzania interakcji użytkownika z programem. Ważnymi atrybutami klasy są atrybuty :

- typu surface - reprezentujące wygląd z aktualną posiadaną bronią.
- typu Weapon - klasa player posiada kontener przechowujący wskaźniki na aktualnie posiadane bronie oraz atrybut wskazujący na bieżącą broń.
- enum class Sequence - służy nam do łatwego sterowania animacjami ruchu.
- typu Object - klasa ta określa położenie gracza na mapie oraz jego przestrzeń.

Klasa Board:

Klasa ta zawiera w sobie takie funkcjonalności jak inicjalizowanie kolejnych rund , dodawanie kolejnych broni i ulepszeń do gracza, inicjalizacja mapy. Ważnymi atrybutami klasy board jest wektor wskaźników na wszystkie bronie dostępne w grze, wskaźnik na obiekt typu Map reprezentujący aktualnie wybraną mapę.

Klasa Weapon:

Główna klasa która zapewnia interfejs do wszystkich rodzajów broni. Jest to klasa po której dziedziczą wszystkie bronie.

Klasa Object:

Główna klasa agregująca w sobie klasę Vec2D. Klasa ta zawiera takie funkcjonalności jak określanie pozycji oraz możliwości poruszania się po mapie, zdefiniowanie obszaru zajmowanego w danym momencie na mapie(funkcjonalność tą zapewnia pod klasa HitBox).

Klasa Animation:

Klasa odpowiada za tworzenie animacji poprzez umieszczenie odpowiednich obrazków w wektorze odpowiednich ramek.

Klasa FrameTimer:

Klasa odpowiada za operowanie czasem podczas działania programu tzn. zapewnia odpowiednią ilość czasu na wyświetlenie animacji i innych podobnych działań , które wymagają określenia jak długo dana operacja ma być wykonywana.

Diagram powiązań klas wraz z diagramem wywołań i innymi diagramami znajduje się w dokumentacji, którą zamieściliśmy w repozytorium

Wykorzystane techniki obiektowe:

- Dziedziczenie
- Polimorfizm
- Wzorce
- RTTI
- Inteligentne wskaźniki
- Algorytmy i iteratory
- Kontenery STL

6. Testowanie i uruchamianie:

Program został przetestowany pod względem wycieków pamięci za pomocą programu Visual Leak Detector w różnych konfiguracjach uruchomieniowych, znalezione wycieki pamięci udało nam się usunąć, więc program powinien być od nich wolny. Poprawność działania programu została sprawdzona poprzez wielokrotne użytkowanie przez różne osoby (wysyłaliśmy plik wykonywalny znajomym, by pomogli nam przyspieszyć testowanie gry), co potwierdziło poprawne działanie gry.

7. Uwagi i wnioski:

Program do poprawnego działania potrzebuje w folderze Images plików graficznych, które są do niego importowane, a także w folderze Sounds plików dźwiękowych.

Tworzenie projektu w dwie osoby wymusiło rozplanowanie pracy na obie osoby tak, aby nie wchodzić sobie w drogę. Prowadzący zajęć służył nam pomocą, gdy tego potrzebowaliśmy.

Link do kodu wraz z dokumentacją: https://github.com/PatrykLipka/Projekt_PK4