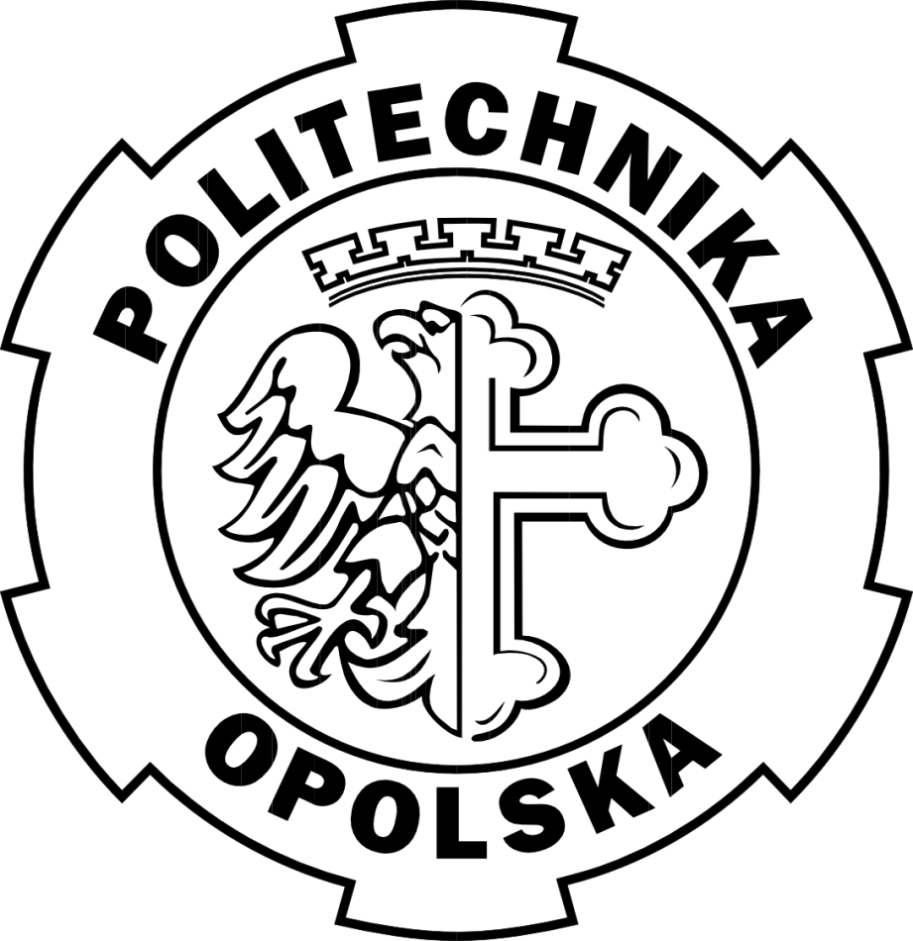
**POLITECHNIKA OPOLSKA**

**Wydział Elektrotechniki, Automatyki i Informatyki**

**Kierunek: Informatyka**



**Przedmiot:**

**Zarządzanie projektami informatycznymi**

**Opracowali:**

**Grupa 4**

Piotr Słupecki

Łukasz Grześkowiak

Sebastian Blaik

Adrian Baran

DOKUMENTACJA

Projekt gry Bomberman

1. Narzędzia i programy

Gra zostanie napisana w języku programowania C#. Korzystamy z programu Microsoft Visual Studio oraz rozszerzenia *GitHub Extension for Visual Studio* którego używamy jako system kontroli wersji. Pozwala on na sklonowanie istniejącego już repozytorium, dodanie nowych commitów, pobranie najnowszych zmian z aktywnego brancha oraz wyświetlenie listy zmienionych plików oraz samych zmian. Możemy też skorzystać z programu git, a poniżej jego przykładowe polecenia:

git clone <https://github.com/<twój-login>/test.git> - klonowanie instniejacego juz repozytorium

git add <plik> - dodawanie istniejącego pliku do repozytorium

git rm <plik> - usuwanie istniejącego pliku z repozytorium

git status - sprawdzanie istniejącej wersji

git commit -a - stworzenie nowej kopi plików po zmianach

git push - wysyłanie nowych plików

git diff - sprawdzenie zmian przed wysłaniem

git revert - przywracanie starszej wersji pliku

Repozytorium zostało stworzone na GitHubie, poniżej link:

https://github.com/iNQKING/Bomberman

2. Dokumentacja kodu

Główną klasą programu jest ConsoleApplication która wyświetla menu główne gdzie mamy opcje wyboru:

„Graj”, „Jak grać”, „O grze”, „Wyjście z programu” oraz wyświetla którą opcję wybraliśmy.

Console.Write("\t\t1. Graj\n");

Console.Write("\t\t2. Jak grać?\n");

Console.Write("\t\t3. O grze\n");

Console.Write("\t\t0. Wyjście z programu\n");

Console.Write("\n\n\t\tTwój wybór: ");

choiceString = Console.ReadLine();

choice = int.Parse(choiceString);

Menu główne jest oparte na caseach, poniżej przyklad pierwszego case („Graj”):

case 1:

Console.Clear();

while (true)

{

console.ClearMap(map);

console.DrawMap(map);

console.DrawBomberman(bomberman, map);

interaction.Move(bomberman);

if (check.CrashWall(bomberman) || check.CrashBody(bomberman, map))

{

Console.WriteLine("\n\nKONIEC!");

//Console.Beep(1000, 1000);

break;

}

}

break;

Następną klasą jest Mapa, która definiuje rozmiar planszy. Poniższa pętla jest odpowiedzialna za wstawianie pustego znaku wewnątrz mapy:

class Map

{

const int height = 20; //Rozmiar

const int width =70;

private char[,] arrayMap;

public Map()

{

arrayMap = new char[height, width];

for (int i = 0; i < height; i++)

for (int j = 0; j < width; j++)

arrayMap[i, j] = ' '; //Pusty znak

}

public char[,] ArrayMap

{

get { return this.arrayMap; }

}

}

Klasa ConsoleOperation odpowiada za rysowanie obramowania mapy oraz postaci (bomberman):

public void DrawMap(Map map)

{

WriteAt("╔", 0, 0);

WriteAt("╗", 71, 0);

WriteAt("╚", 0, 21);

WriteAt("╝", 71, 21);

WriteAt("══════════════════════════════════════════════════════════════════════", 1, 21);

WriteAt("══════════════════════════════════════════════════════════════════════", 1, 0);

for (int i = 0; i < height; i++)

{

WriteAt("║", 0, 1 + i);

for (int j = 0; j < width; j++)

{

Console.Write(map.ArrayMap[i, j]);

}

WriteAt("║", 71, 1 + i);

}

}

A niżej kod odpowiadający za wyświetlania postaci:

public void DrawBomberman(Bomberman bomberman, Map map)

{

foreach (Point point in bomberman.Body)

map.ArrayMap[point.X, point.Y] = 'o'; }

InteracionUser jest klasa odpowiedzialną za ruch postaci (poruszanie się i deklaracja klawiszy):

enum Direction { Left = 'a', Right = 'd', Up = 'w', Down = 's' }

Point pozycja;

Direction currentDirection = Direction.Left;

ConsoleOperation console = new ConsoleOperation();

Kod pokazujący pętle if w klasie InteractionUser:

if (Console.KeyAvailable)

{

ConsoleKeyInfo key = Console.ReadKey(true);

if (key.KeyChar == (char)Direction.Up) // jeśli wcisnęliśmy klawisz 'W'

{

if (currentDirection != Direction.Down) // jeśli aktualny kierunek węża jest różny od kierunku w dół

{

pozycja = new Point(pozycja.X - 1, pozycja.Y);