



**Dokumentacja Projektu III – widok**

**Temat: Gra “Snake”**

**Komunikacja Człowiek-Komputer**

Wykonujący projekt:

**Gabriel Czajkowski**

Studia dzienne

Kierunek: Informatyka

Semestr: V

Grupa zajęciowa: PS2

Prowadzący:

**mgr inż. Aleksander Sawicki**

## 1. OPIS PROJEKTU

Głównym celem oraz założeniem projektu było zaimplementowanie gry wzorowanej na kultowym Snake'u działającej w trybie graficznym. Pod względem logiki miała być to dokładnie ta sama aplikacja, co w przypadku pierwszego projektu (wersji konsolowej), tylko ze zmienioną warstwą prezentacji z tekstowej na graficzną. Ponadto miała być ona przyjazna i atrakcyjna użytkownikowi końcowemu poprzez dodanie animacji oraz wygodnego i intuicyjnego w obsłudze (za pomocą myszki i klawiatury) interfejsu. Do jej implementacji wykorzystano silnik graficzny i API – WPF (Windows Presentation Foundation). Napisana została w językach XAML oraz C#, w środowisku Microsoft Visual Studio.

Snake jest to gra jednoosobowa, w której gracz kontroluje węża, poruszającego się po obramowanej planszy oraz zjadającego owoce. Pożywienie pojawia się w losowym miejscu. W grze są dwa jego rodzaje: podstawowe oraz specjalne. Za zjedzenie pierwszego z nich gracz otrzymuje ilość punktów zależną od poziomu trudności, długość węża zwiększa się o jeden człon oraz wzrasta jego prędkość. Specjalne owoce pojawiają się niezależnie od podstawowych. Dają one znacznie więcej punktów, jednak w zamian za to, długość węża zwiększa się o dwa członki oraz znacznie wzrasta jego prędkość. Gra kończy się, gdy wąż uderzy głową w ścianę lub część własnego ciała. Gracz kontroluje kierunek ruchu gada za pomocą strzałek (góra, dół, lewo, prawo) lub klawiszy WSAD (W – góra, S – dół, A – lewo, D – prawo).

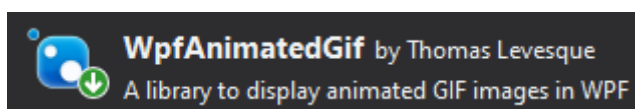
## 2. OPIS FUNKCJONALNOŚCI

- Po uruchomieniu aplikacji słyszalna będzie melodia uprzyjemniająca wrażenia z gry.
- Piosenkę można wyłączyć lub ponownie włączyć w każdym z widoków. Służy do tego specjalny przycisk znajdujący się w prawym górnym rogu, z ikoną dwóch ósemek połączonych belką (∞).
- Możliwość zmiany utworu muzycznego na inny w ustawieniach gry.
- Podczas wyboru piosenki zastosowano animacje oraz wyłączenie przycisku z aktualnie grającą melodią w celu zwiększenia intuicyjności aplikacji.
- Ikona odpowiadająca za odtwarzanie muzyki zmienia się w zależności od tego, czy utwór jest włączony czy nie.
- Podkreślenie najechania kursorem na przycisk poprzez zmianę kursora ze strzałki na rękę, zmianę jego wyglądu i/lub pojawienie się po bokach obrazków i/lub zastosowaniu mechanizmu podpowiadającego, co może się stać po jego naciśnięciu.
- Możliwość wyboru jednego z trzech poziomów trudności:
  - Łatwy,
  - Średni,
  - Trudny.
- W zależności od wybranego poziomu trudności, różnić się będzie prędkość początkowa węża, ilość zdobywanych punktów oraz zwiększenie jego prędkości za każdy zjedzony owoc.
- Możliwość wyboru własnych ustawień dotyczących:
  - Prędkości węża:
    - Wolno,
    - Średnio,
    - Szybko,
    - Bardzo szybko.
  - Częstotliwości pojawiania się specjalnych owoców:
    - Rzadko,
    - Normalnie,
    - Często,
    - Bardzo często.
- Po skorzystaniu z jakiegokolwiek z dwóch powyższych opcji personalizacji ustawień, poziom trudności zostanie ustawiony na "CUSTOMIZED", a wąż nie będzie przyspieszał po zjedzeniu jakiegokolwiek z owoców.

- Zastosowanie strzałek powrotu do poprzedniego widoku pomagających w poruszaniu się po aplikacji.
- Możliwość zobaczenia autora gry.
- Możliwość wyświetlenia 10 najlepszych graczy wraz z osiągniętymi przez nich miejscami, punktami, poziomem trudności na których grali oraz datą rozgrywki.
- Możliwość sortowania tabeli z najlepszymi wynikami ze względu na miejsce, wynik, nazwę gracza, poziom trudności lub datę rozegrania gry.
- Zastosowanie animacji specjalnego owocu (obracający się arbuz), uprzyjemniającej wrażenia z gry.
- Każdej zmianie ustawień, towarzyszyć będzie obrazek, ułatwiający zrozumienie dokonywanego wyboru.
- Przed rozpoczęciem rozgrywki konieczne jest wprowadzenie nazwy gracza.
- Zastosowanie mechanizmu walidacji podczas wprowadzania nicku (musi składać się z minimum 3 i maksymalnie 12 znaków).
- Odliczanie do rozpoczęcia rozgrywki.
- Podczas gry wyświetlanie:
  - Nazwy gracza.
  - Ilości zdobytych przez niego punktów.
  - Poziomu trudności, na którym gra.
- Po przegranej wyświetlenie ilości zdobytych punktów oraz dodatkowej informacji w przypadku osiągnięcia nowego rekordu.
- Zapis i odczyt wyników do/z pliku txt.
- Możliwość łatwego rozpoczęcia kolejnej gry po porażce lub powrót do menu głównego.
- Sterowanie wężem za pomocą strzałek.
- Podczas próby wyjścia z gry zapytanie, czy jest się pewnym podjętej decyzji.

### 3. SZCZEGÓLNIIE INTERESUJĄCE ZAGADNIENIA PROJEKTOWE

Pierwszym interesującym zagadnieniem jest zastosowanie gifów. Zaimplementowano je przy pomocy biblioteki "WpfAnimatedGif", której autorem jest Thomas Levesque. Początkowo konieczna była instalacja wybranego pakietu. W następnym kroku niezbędne okazało się podpięcie biblioteki do przestrzeni nazw. Sposób wykorzystania biblioteki różni się od implementacji obrazka jedynie podaniem ścieżki do wybranego zasobu.



Obraz 1. Wybrany pakiet.

```
xmlns:gif="http://wpfanimatedgif.codeplex.com"
```

Obraz 2. Podpięcie biblioteki do przestrzeni nazw.

```
<Image
  Name="RightSnakeSong_Gif"
  Grid.Column="2"
  Width="60"
  Height="40"
  gif:ImageBehavior.AnimatedSource="/Images/Settings/Music/MusicGif.gif"
  Visibility="Hidden" />
```

Obraz 3. Sposób wykorzystania biblioteki

Kolejnym ciekawym rozwiązaniem okazał się sposób wyświetlania najlepszych 10 wyników wraz z możliwością ich sortowania. W tym celu wykorzystano kontrolkę "DataGrid" oraz mechanizm wiązania danych – *Data Binding*.

```
<DataGrid
    Name="Results_DataGrid"
    Grid.Row="1"
    Margin="0,5,0,0"
    HorizontalAlignment="Center"
    VerticalAlignment="Top"
    AutoGenerateColumns="False"
    Background="■"#E2E2E2"
    CanUserReorderColumns="False"
    CanUserResizeColumns="False"
    CanUserResizeRows="False"
    IsReadOnly="True">
    <DataGrid.Columns>
        <DataGridTemplateColumn Header="MEDAL">
            <DataGridTemplateColumn.CellTemplate>
                <DataTemplate>
                    <Image
                        Width="25"
                        Height="25"
                        Source="{Binding Place, Converter={StaticResource ResourceKey=ImageConverter}}" />
                </DataTemplate>
            </DataGridTemplateColumn.CellTemplate>
        </DataGridTemplateColumn>
        <DataGridTextColumn Binding="{Binding Place}" Header="PLACE">
```

Obraz 4. Część kodu kontrolki DataGrid wraz z mechanizmem Data Bindingu.

```
private List<GameResult> GetResults()
{
    List<GameResult> resultsList = new List<GameResult>();
    StreamReader streamReader = new StreamReader("Results.txt");
    string line;
    while ((line = streamReader.ReadLine()) != null)
    {
        string[] row = line.Split(' ');
        resultsList.Add(new GameResult(ImagePath: null, Place: "", Score: row[0],
            Nickname: row[1], DifficultyLevel: row[2], Date: row[3]));
    }
    streamReader.Close();
    resultsList.Sort((GameResult x, GameResult y) => y.Score.CompareTo(x.Score));
    //BEST 10 SCORES
    resultsList = resultsList.Take(10).ToList();
    int counter = 1;
    foreach (GameResult person in resultsList)
    {
        if (counter < 10) person.Place = "0" + counter.ToString();
        else person.Place = counter.ToString();
        counter++;
    }
    return resultsList;
}

1 reference
private void ShowResults(List<GameResult> resultsList) { Results_DataGrid.ItemsSource = resultsList; }
```

Obraz 5. Metody GetResults() oraz ShowResults() odpowiedzialne za prawidłowe wyświetlenie 10 najlepszych wyników.

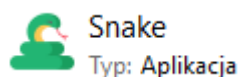
Ostatnim ciekawym rozwiązaniem jest podpis autora aplikacji. Wykonany został poprzez wybranie odpowiedniej czcionki imitującej pismo odręczne oraz przerobieniu jej na grafikę wektorową.

```
<Canvas
  Name="Gabriel"
  Width="380"
  Height="130">
  <Canvas.RenderTransform>
    <TranslateTransform X="-63" Y="-90" />
  </Canvas.RenderTransform>
  <Canvas.Resources />
  <Canvas>
    <Canvas>
      <Canvas.RenderTransform>
        <TranslateTransform X="120.81938171386719" Y="106.35999870300293" />
      </Canvas.RenderTransform>
      <Path
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Name="path3093"
        Fill="□" #FF000000">
```

Obraz 6. Część kodu odpowiedzialnego za wektorowy podpis autora.

#### 4. INSTRUKCJA INSTALACJI

W celu skompilowania kodu gry oraz uruchomienia aplikacji niezbędne jest posiadanie zainstalowanego środowiska programistycznego Microsoft Visual Studio. Następnie należy zainicjować plik wykonywalny z rozszerzeniem .exe o nazwie Snake znajdujący się w folderze Debug. Ścieżka: Snake/bin/Debug/Snake.exe



Obraz 7. Ikona do uruchomienia aplikacji.

#### 5. INSTRUKCJA KONFIGURACJI

Aplikacja nie ma specjalnych wymagań dotyczących konfiguracji.

## 6. INSTRUKCJA UŻYTKOWANIA

Po całej aplikacji możemy przemieszczać się za pomocą myszki bądź klawiatury – program jest bardzo intuicyjny. W celu wybrania interesującej nas opcji wystarczy najechać na nią kursorem i kliknąć lewy przycisk myszki.

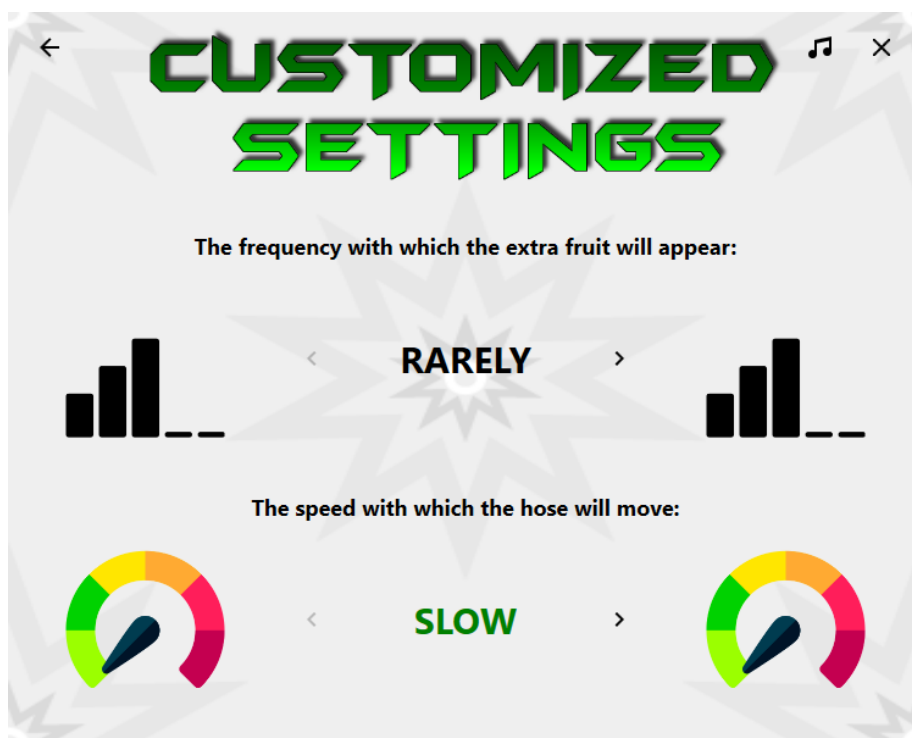


Obraz 8. Wygląd menu po najechaniu kursorem na przycisk “NEW GAME”.

Podczas gry wężem poruszać można za pomocą przycisków:

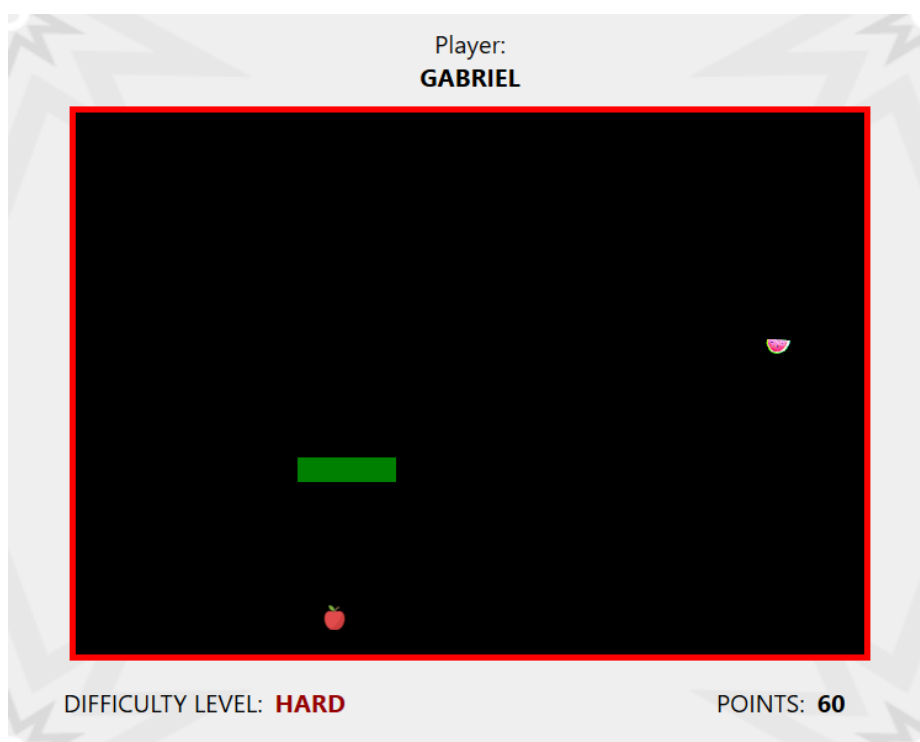
- strzałka w górę (↑) – zmiana kierunku ruchu na północny,
- strzałka w dół (↓) – zmiana kierunku ruchu na południowy,
- strzałka w lewo (←) – zmiana kierunku ruchu na zachodni,
- strzałka w prawo (→) – zmiana kierunku ruchu na wschodni,
- przycisk W – zmiana kierunku ruchu na północny,
- przycisk S – zmiana kierunku ruchu na południowy,
- przycisk A – zmiana kierunku ruchu na zachodni,
- przycisk D – zmiana kierunku ruchu na wschodni.

Domyślnym poziomem trudności jest EASY, jednak można go zmieniać w ustawieniach na MEDIUM/HARD lub na własne ustawienia.



Obraz 9. Widok personalizacji ustawień.

W lewym dolnym rogu, podczas rozgrywki znajduje się poziom trudności, w prawym dolnym obecna ilość zebranych punktów, a na samej górze po środku nazwa gracza. Standardowe owoce to jabłka koloru czerwonego, natomiast specjalne to obracające się wokół własnej osi arbuzy.



Obraz 10. Widok rozgrywki

Gra kończy się, gdy wąż uderzy głową w ścianę lub część własnego ciała.

W celu sprawdzenia najlepszych wyników należy wybrać z menu opcję "BEST RESULTS".

Im większy poziom trudności rozgrywki, tym szybszy staje się wąż oraz większa jest ilość punktów zdobywanych za podstawowe pożywienie.

## **7. WNIOSKI**

Projekt spełnia wszystkie wymagania postawione przed jego stworzeniem. Zawiera przyjemny, prosty w obsłudze oraz przejrzysty interfejs graficzny. Dzięki wykorzystaniu gifów, muzyki grającej w tle oraz opcji personalizacji ustawień aplikacja jest atrakcyjna dla użytkownika końcowego. Odświeżane są tylko niezbędne elementy, a nie całe okno co znacznie wpływa na szybkość działania programu. Jest on dynamiczny, nie zacina się i nie ma żadnych błędów.

## **8. SAMOOCENA**

Myślę, że napisana przeze mnie aplikacja działająca w trybie graficznym spełnia wszystkie postawione wymagania. Jest podobna do oryginału oraz posiada identyczną logikę jak poprzedni projekt w wersji tekstowej dzięki poświęceniu dużej ilości czasu i wysiłku na jej napisanie i dopracowanie. Nie posiada żadnych błędów oraz działa bardzo szybko. Jestem zadowolony z efektu końcowego mojej pracy.