



**Politechnika Wrocławska**

---

**Laboratorium : Platformy prog. .Net i Java Y02-15b**

**Grupa : Czwartek 18:55 - grupa 10**

Zadanie 2:

Wielowątkowa aplikacja okienkowa w technologii Java  
Swing - Snake

Wykonali:

Michał Szkudlarek - nr indeksu: 259248

Konrad Gawerda - nr indeksu: 259333

# 1 Cel projektu

Naszym projektem jest aplikacja napisana w języku Java z wykorzystaniem biblioteki Swing. Gra Snake jest wielowątkowa, z czego w aplikacji tworzone są 4 wątki odpowiadające kolejno za:

- mysz
- generowanie przeszkód
- węża sterowanego przez gracza
- węża sterowanego przez SI

Aplikacja została wyposażona w interfejs graficzny(GUI), w tym celu wykorzystano bibliotekę Abstract Window Toolkit(AWT). W celu stworzenia aplikacji wykorzystane zostało środowisko programistyczne IntelliJ IDEA, gdzie został wygenerowany diagram UML oraz dokumentacja przy pomocy Javadoc.

## 2 Funkcjonalność

Aplikacja posiada poniższe funkcjonalności:

- generowanie losowych przeszkód
- system kolizji
- wąż sterowany przez gracza
- wąż sterowany przez SI
- system zliczania punktów
- restart gry
- wyświetlenie najlepszych wyników
- obsługa zapytań z klawiatury i myszy

### 3 System kolizji

Wąż podczas zderzenia z pożywieniem (jabłko lub mysz) powiększa swój rozmiar oraz zdobywa punkt. Nie jest on jednak odporny na zderzenie z:

- własnym ogonem
- wężem sterowanym przez SI
- przeszkodami pionowymi, jak i poziomymi - które generowane są losowo na początku każdej rozgrywki
- granicami planszy



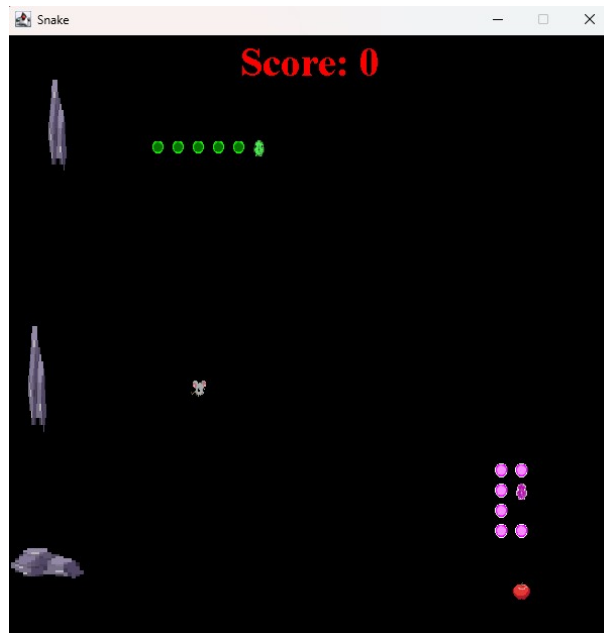
Zrzut ekranu 1: Przykład losowo wygenerowanych przeszkód



Zrzut ekranu 2: Przykład losowo wygenerowanych przeszkód

## 4 Wąż sterowany przez gracza oraz wąż sterowany przez SI

W grze użytkownik nie jest jednym graczem, steruje on podstawową wersją węża (kolor zielony), jednak jego rywalem jest wąż sterowany przez SI (nie został zaimplementowany żaden gotowy algorytm, skorzystaliśmy z własnej logiki, dzięki której wąż sukcesywnie sprawdza czy istnieje możliwość kolizji i w przypadku wystąpienia takiej sytuacji realizuje wszystkie możliwe kroki, aby nie dopuścić do kolizji, wąż SI jest koloru fioletowego).



Zrzut ekranu 3: Długość węża na początku rozgrywki



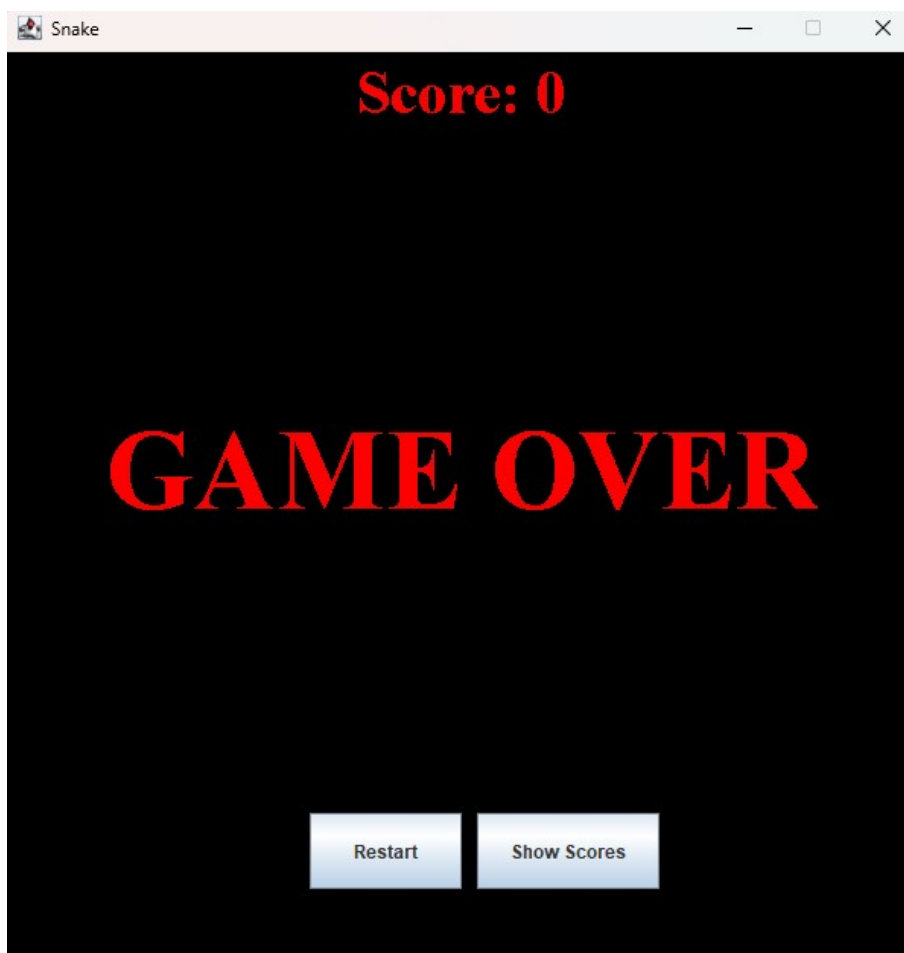
Zrzut ekranu 4: Długość węża po zdobyciu kilku punktów

## 5 Koniec gry oraz restart

Gra kończy się w momencie, kiedy wąż sterowany przez gracza zderzy się z przeszkodą, wąż SI może zderzyć się z przeszkodą, jednak nie powoduje to zakończenia rozgrywki, ale wąż SI jest wykluczony z rozgrywki.

Po zakończeniu rozgrywki pojawia się napis **"Game Over"** sygnalizujący zakończenie rozgrywki, dostępne są dwa przyciski:

- restartu gry
- wyświetlenia najlepszych wyników



Zrzut ekranu 5: Ekran wyświetlany po zakończeniu rozgrywki

## 6 Obsługa zapytań z myszy i klawiatury

Gra obsługuje zapytania zarówno z myszy, jak i klawiatury  
W przypadku klawiatury:

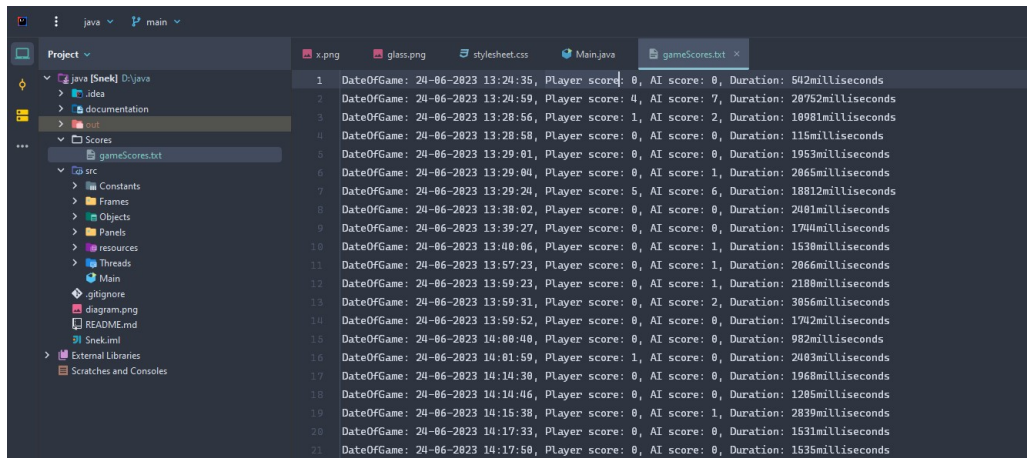
- lewa strzałka - odpowiada za skręcanie w lewo podczas rozgrywki
- prawa strzałka - odpowiada za skręcanie w praw podczas rozgrywki
- strzałka w dół - odpowiada za skręcanie w dół podczas rozgrywki
- strzałka w górę - odpowiada za skręcanie w górę podczas rozgrywki

W przypadku myszy:

- obsługa przycisków po zakończeniu rozgrywki

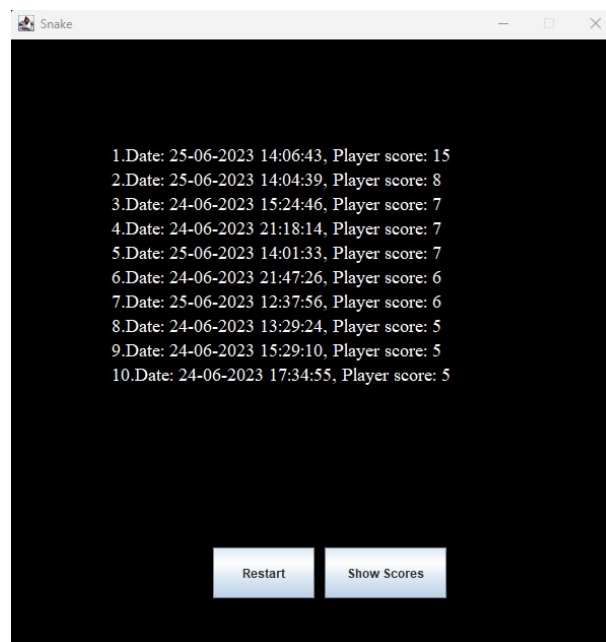
## 7 System zliczania punktów

Gra posiada system zliczania punktów gracza oraz SI, posiada również system zliczania długości rozgrywki. Wszystkie te wyniki zapisywane są do pliku .txt, który stanowi podstawę dla wszystkich rozgrywek - na wzór klasycznych gier arkadowych.



Zrzut ekranu 6: Plik .txt z wynikami

Program wybiera z pliku tekstowego 10 najlepszych wyników i pokazuje je po naciśnięciu przycisku "Show Scores" po zakończeniu rozgrywki.



Zrzut ekranu 7: Wyświetlanie 10 najlepszych wyników

## 8 Diagram UML oraz dokumentacja

Zarówno diagram UML, jak i dokumentacja zostały wykonane przy pomocy IntelliJ IDEA.



Zrzut ekranu 8: Diagram UML

Strona główna dokumentacji Javadoc:

OVERVIEW PACKAGE CLASS TREE INDEX HELP	
Packages	
Package	Description
Constants	
Frames	
Objects	
Panels	
Threads	

Zrzut ekranu 9: Dokumentacja Javadoc



## 9 Linki

- Repozytorium projektu: <https://github.com/kgawerda/Snek>
- Przykład działania aplikacji: <https://youtu.be/K5udXloTXT8>
- Zasady gry: [https://en.wikipedia.org/wiki/Snake\\_\(video\\_game\\_genre\)](https://en.wikipedia.org/wiki/Snake_(video_game_genre))