



Politechnika Wrocławska

---

# Platformy Programistyczne NET i Java: Aplikacja okienkowa - Snake

**Wtorek 18:55**

---

## **Sprawozdanie**

Jędrzej Kuśnierz, Nr indeksu: 263494

Maciej Kościński, Nr indeksu: 264318

# 1 Cel projektu

Celem projektu było stworzenie gry w Snake'a.

## 2 Opis działania aplikacji

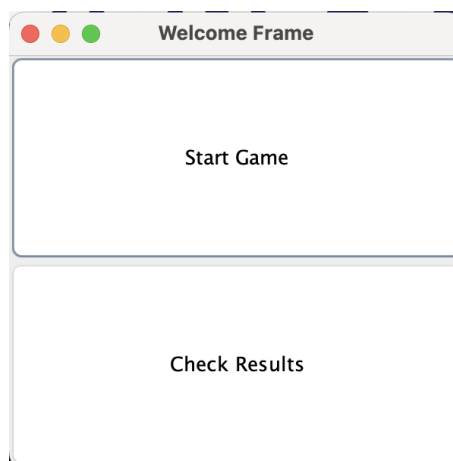
### 2.1 Ogólne działanie

Projekt to gra w węża, zrealizowana w języku Java przy użyciu biblioteki Swing do obsługi interfejsu graficznego. Gra oferuje możliwość rywalizacji z komputerowo sterowanymi wężami. Projekt zawiera różne klasy reprezentujące elementy gry, takie jak gracz, komputer, owoce i żaby. Wyniki gry są zapisywane i mogą być przeglądane w osobnym oknie wyników. Aplikacja wykorzystuje wielowątkowość, operacje wejścia/wyjścia do zapisywania i odczytywania wyników z pliku tekstowego.

### 2.2 Zasady gry

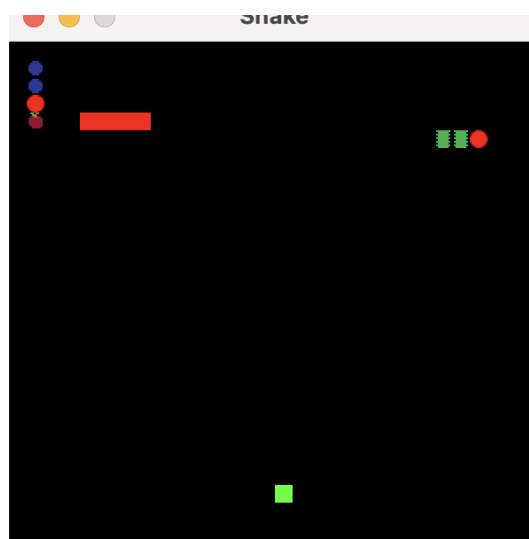
- Gracz/Komputer zbierają 2 typy nagród: jabłka/żaby. Jabłka są statyczne i generują się w losowym miejscu na mapie, gdzie pozostają do zebrania ich przez gracza. Żaby co każdy "cykl" przeskakują w losowo wybrane miejsce w najbliższym otoczeniu - są znacznie cięższe do zebrania. Z tego też powodu dają 2 punkty
- Celem jest zebranie jak największej ilości punktów.
- Węża komputera można blokować, jeżeli się to uda i uderzy w ciało twojego węża będzie zamrożony na 5 sekund.
- Wąż komputera podąża tylko za jabłkami i unika występujących na mapie obiektów.

### 2.3 Menu



*Menu główne gry*

## 2.4 Gra



*Wygląd gry. Niebieski wąż to komputer, zielony to gracz. Jabłko w lewym górnym rogu. Żaba w dolnej części ekranu*

Każdy z obiektów obsługiwany jest przez osobny wątek a komputer ma zaprojektowany algorytm który podąża za owocem dodatkowo omijając przeszkody.

## 2.5 Tabela wyników

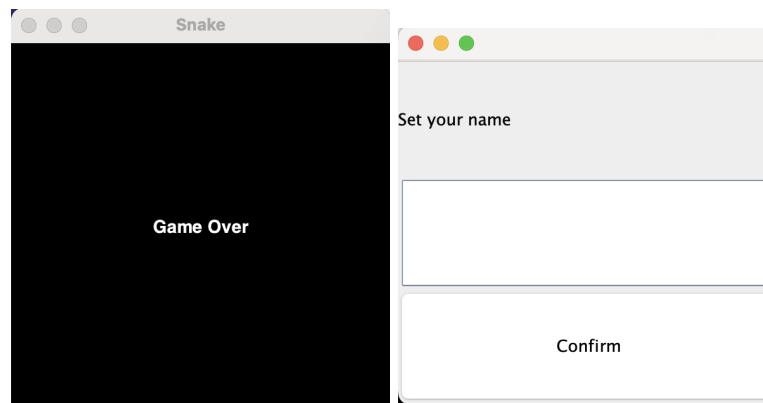
Istnieje możliwość wyświetlenia tabeli z wynikami:

Results Frame	
Name	Result
jedrek	2
ja	4
ez	3
x	0
jedrzej	3
x	0

*Tabela z wynikami*

## 2.6 Koniec gry

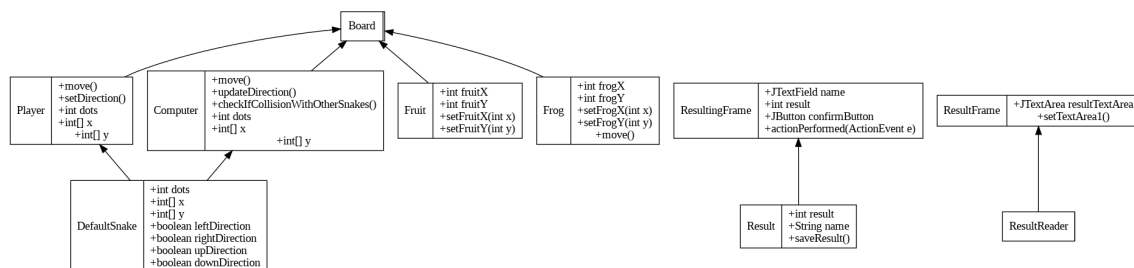
Po porażce wyświetla się napis "Game Over" a użytkownik ma możliwość zapisać swój wynik w tabeli



*Game Over i zapisywanie wyniku*

### 3 Diagram UML

Stworzyliśmy diagram klas UML



*Diagram klas UML*

### 4 Dokumentacja

Wykorzystując komentarze wygenerowaliśmy dokumentację javadoc.

### Podsumowanie

#### Kluczowe elementy programu

- **Klasa Board:** Jest to centralny element gry, który rozszerza `JPanel` i implementuje `ActionListener` do obsługi aktualizacji gry. Odpowiada za renderowanie planszy, ruch gracza i węży sterowanych przez komputer, wykrywanie kolizji oraz śledzenie punktacji. Metody takie jak `locateFruit`, `locateFrog` i `locateObstacle` są używane do dynamicznego rozmieszczania elementów gry na planszy.
- **Klasa Player:** Reprezentuje węża sterowanego przez gracza. Rozszerza klasę `DefaultSnake`, która zawiera wspólne właściwości i metody dla wszystkich węży. Klasa ta obsługuje ruch węża na podstawie wejścia z klawiatury oraz jego interakcję z owocami i żabami.
- **Klasa Computer:** Reprezentuje węże sterowane przez komputer. Również rozszerza klasę `DefaultSnake` i zawiera logikę odpowiedzialną za ruch AI, aktualizację kierunku oraz wykrywanie kolizji z innymi wężami i przeszkodami.

- **Klasa DefaultSnake:** Klasa bazowa dla węży, zawierająca podstawowe właściwości (takie jak pozycje segmentów węża) oraz metody wspólne dla węży sterowanych przez gracza i komputer.
- **Klasa Fruit:** Reprezentuje owoc, który może być zbierany przez węże, co powoduje ich wzrost oraz zwiększenie punktacji.
- **Klasa Frog:** Reprezentuje żabę, która pojawia się na planszy i może być zbierana przez węże. Zebranie żaby daje dodatkowe punkty. Żaba przemieszcza się co pewien czas, co jest kontrolowane przez licznik w klasie **Board**.
- **Klasa Result:** Reprezentuje wynik gracza, w tym jego punktację i imię. Zawiera metody do zapisywania wyniku do pliku tekstowego.
- **Klasa ResultReader:** Zawiera metody do odczytywania wyników z pliku tekstowego i zwracania ich jako listę obiektów **Result**.
- **Klasy ResultFrame i ResultingFrame:** Okna Swing, które wyświetlają wyniki gry oraz pozwalają graczom na wprowadzenie swojego imienia i zapisanie wyniku. **ResultFrame** wyświetla listę wyników, a **ResultingFrame** pozwala na zapisanie nowego wyniku po zakończeniu gry.