



Politechnika Wrocławska

Platformy programistyczne .Net i Java

Snake

Prowadzący: dr inż. Aneta Górniak

Grupa Wtorek, 18:55 - 20:35

Wojciech Bukała, Przemysław Erbert

14.06.2024

Spis treści

1	Wprowadzenie	2
2	Struktura aplikacji	2
3	Interfejs graficzny	2
4	Kolekcje i algorytmy	2
5	Wielowątkowość	2
6	Sztuczna inteligencja	2
7	Obsługa zdarzeń z klawiatury i myszy	3
8	Zapis i odczyt danych	3
9	Dokumentacja javadoc	3
10	Diagram klas UML	3
11	Podsumowanie	4

1. Wprowadzenie

Do wykonania projektu wybraliśmy grę Snake

2. Struktura aplikacji

Aplikacja składa się z następujących plików:

- **App.java** - główna klasa uruchamiająca grę.
- **SnakeGame.java** - klasa odpowiedzialna za główną logikę gry oraz rysowanie elementów graficznych.
- **Snake.java** - klasa reprezentująca węża w grze.
- **Tile.java** - klasa reprezentująca pojedynczy segment ciała węża oraz inne obiekty na planszy takie jak jabłko, i żaba.

3. Interfejs graficzny

Aplikacja korzysta z biblioteki Swing do wyświetlania elementów graficznych. Główne okno gry jest tworzone i konfigurowane w klasie **App.java**. Plansza gry jest wyświetlana w komponencie **SnakeGame**, który dziedziczy po klasie **JPanel** i nadpisuje metodę **paintComponent** w celu rysowania elementów gry.

4. Kolekcje i algorytmy

Do przechowywania segmentów ciała węża wykorzystano klasę **ArrayList** z biblioteki standardowej Javy. Każdy wąż jest reprezentowany jako obiekt klasy **Snake**, która posiada listę **ArrayList<Tile>** przechowującą segmenty ciała węża. Segmenty te są aktualizowane w czasie rzeczywistym w metodzie odpowiedzialnej za ruch węża.

Dodatkowo, losowe generowanie przeszkód oraz owoców na planszy odbywa się za pomocą metod wykorzystujących kolekcje do przechowywania i zarządzania tymi obiektami. Algorytmy obsługujące ruch węża oraz detekcję kolizji są zaimplementowane w sposób umożliwiający dynamiczną aktualizację stanu gry.

5. Wielowątkowość

Gra obsługuje wielowątkowość, gdzie każdy wąż (w tym AI) oraz żaba są sterowane w oddzielnych wątkach. Wątki te komunikują się między sobą, zapewniając płynność rozgrywki.

6. Sztuczna inteligencja

Sztuczna inteligencja (AI) w grze została zaprojektowana, aby umożliwić podstawową rozgrywkę. Każdy wąż sterowany przez AI działa na osobnym wątku, co pozwala na równoległe przetwarzanie ich ruchów.

AI węży działa na zasadzie prostego algorytmu, który stara się zbierać najbliższe owoce i unikać przeszkód. Algorytm ten działa w następujący sposób:

1. Wąż określa swoją aktualną pozycję oraz pozycje najbliższego owocu.
2. Na podstawie różnicy współrzędnych obliczana jest nowa trajektoria ruchu węża.
3. Wąż zmienia swoją prędkość w kierunku najbliższego owocu, jednocześnie sprawdzając, czy na jego drodze nie znajduje się przeszkoda lub inny wąż.
4. Jeśli wykryta zostanie przeszkoda, AI zmienia kierunek ruchu, aby jej uniknąć.

Żaba, jako ruchomy obiekt, jest również kontrolowana przez AI, które unika węży oraz przeszkód na planszy.

7. Obsługa zdarzeń z klawiatury i myszy

Gra obsługuje zdarzenia z klawiatury i myszy za pomocą mechanizmów wbudowanych w bibliotekę Swing. Sterowanie węzem gracza odbywa się za pomocą klawiatury. W klasie **SnakeGame** dodano nasłuchiwanie zdarzeń klawiatury (**KeyListener**), które reagują na naciśnięcie klawiszy kierunkowych, zmieniając kierunek ruchu węża.

Menu gry oraz interaktywne elementy, takie jak przyciski do restartu gry, są obsługiwane za pomocą myszy. Dodano odpowiednie nasłuchiwanie zdarzeń myszy (**MouseListener**), które reagują na kliknięcia i wykonują odpowiednie akcje, takie jak rozpoczęcie nowej gry.

8. Zapis i odczyt danych

Gra umożliwia zapis i odczyt danych, takich jak tabela najlepszych wyników. Dane te są zapisywane i odczytywane z pliku za pomocą klas **FileWriter** i **BufferedReader** z biblioteki standardowej Javy.

Mechanizm zapisu wyników działa w następujący sposób:

1. Po zakończeniu gry wynik gracza jest zapisywany do pliku tekstowego.
2. Wyniki są przechowywane w uporządkowanej formie, aby można było łatwo je odczytać i wyświetlić na ekranie.
3. Podczas uruchamiania gry, plik z wynikami jest odczytywany, a wyniki są wyświetlane w odpowiedniej sekcji interfejsu użytkownika.

9. Dokumentacja javadoc

Aplikacja została udokumentowana za pomocą komentarzy w kodzie, na podstawie których wygenerowano dokumentację.

10. Diagram klas UML

