

Snake Ai

Kacper Maciejczak 259657, Jakub Cebula 264317

14 czerwca 2024

Projekt Java prosta gra z interfejsem graficznym



1 Założenia Projektowe

Projekt zakłada utworzenie aplikacji desktopowej z interaktywnym interfejsem graficznym.

Aplikacja będzie wykorzystywać wielowątkowość w celu zapewnienia płynności działania.

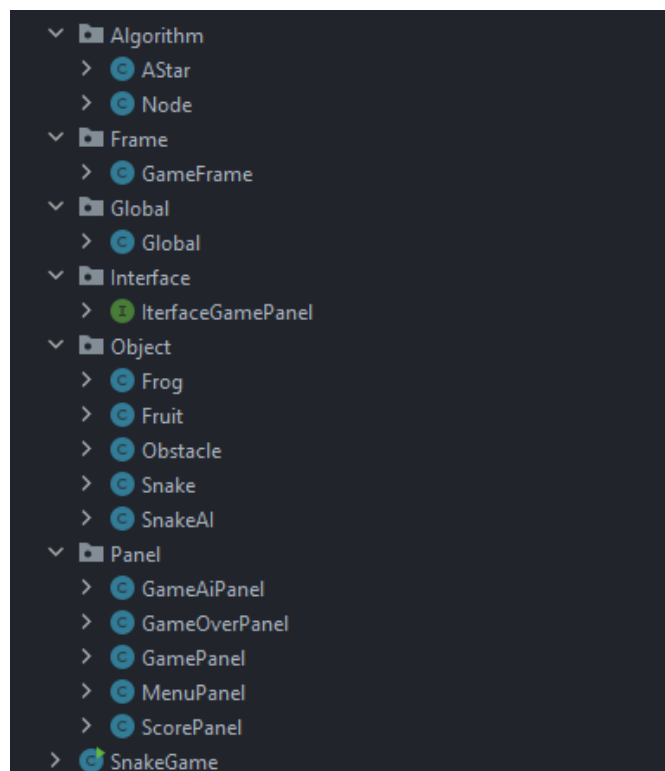
Interfejs graficzny będzie tworzony przy użyciu biblioteki Swing, co zapewni intuicyjną i responsywną obsługę użytkownika.

Aplikacja będzie reagować na zdarzenia z klawiatury i myszy, umożliwiając użytkownikowi wygodne korzystanie.

Dane o najlepszym wyniku będą zapisywane do pliku tekstowego. Proces zapisu i odczytu danych będzie realizowany przy użyciu odpowiednich funkcji z biblioteki standardowej języka programowania.

1.1 Podział projektu

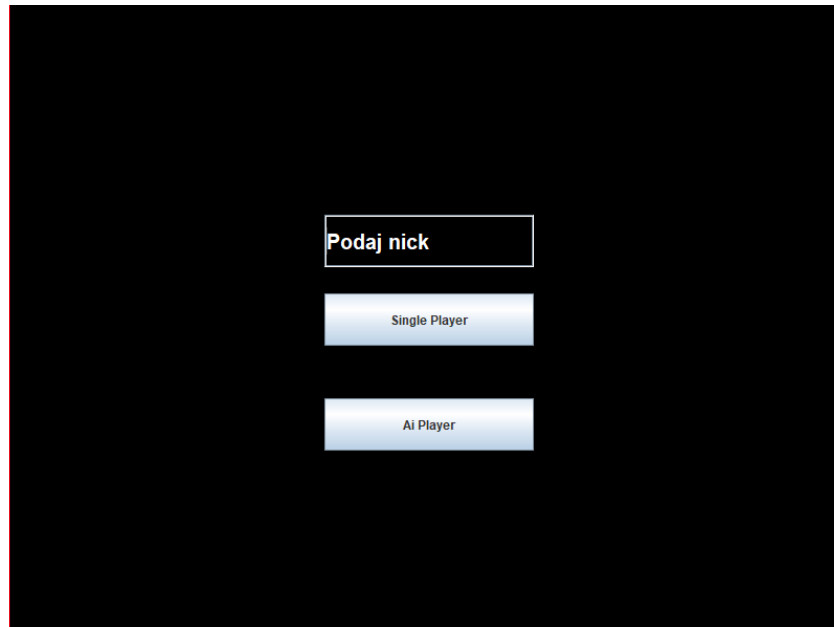
Dla lepszej przejrzystości kodu odpowiednie funkcje zostały podzielone na odpowiednie moduły.



Rysunek 1: Paczki

1.2 Ekran główny

Główny panel aplikacji zawiera dwie opcje gry: Singleplayer oraz Ai. Umożliwia również graczowi wprowadzenie pseudonimu, który zostaje zapisany w pliku z najlepszymi wynikami.

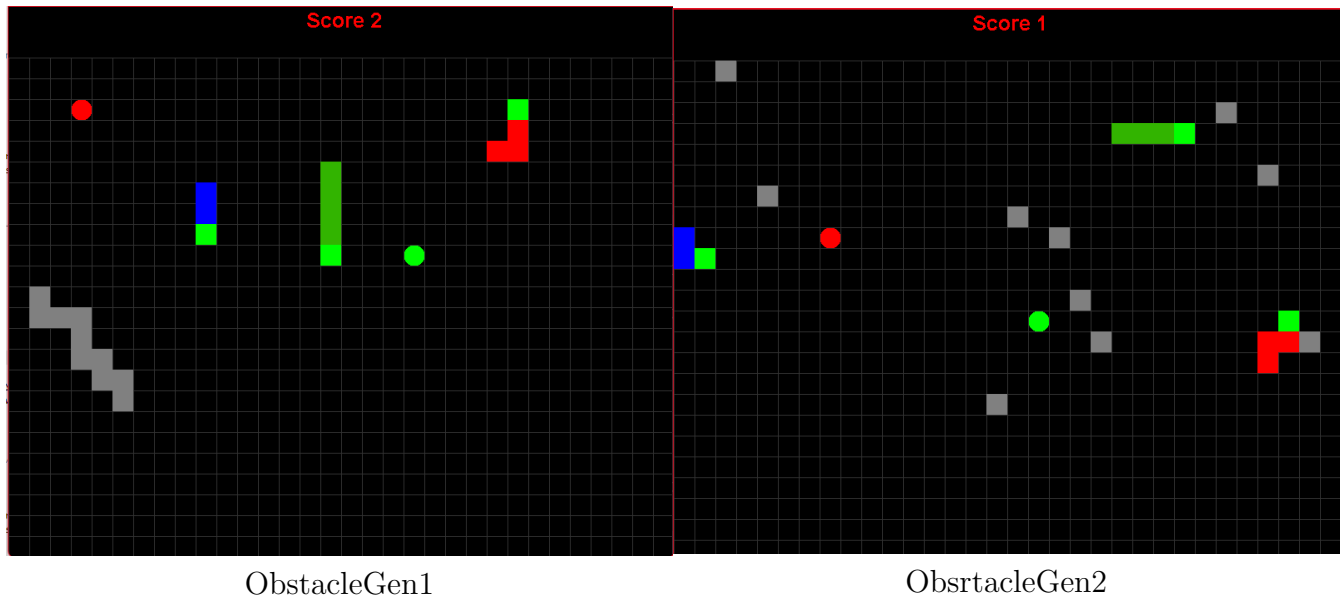


Rysunek 2: Ekran główny

1.3 Single Player

Po wciśnięciu przycisku zaczyna się gra Single Player w której sterujemy zielonym węże.

Aby zacząć rozgrywkę musimy kliknąć przycisk "D" a następnie "Space"



- Niebieski węź - Ai sterowany algorytmem najkrótszej ścieżki A* dąży do pokonania gracza, zablokowania mu drogi
- Czerwony węź - Ai sterowany algorytmem najkrótszej ścieżki A*, jego celem jest zbieranie jabłek
- Szare elementy - Przeszkoda po wjechaniu kończy grę
- Czerwone kółko - Jabłko po zjedzeniu zwiększa długość węży, przez gracza zwiększa jeszcze wynik.
- Zielone kółko - Żaba gdy wykryje w pobliżu węża zaczyna uciekać, działa tak samo jak jabłko.

1.4 Algorytm

Algorytm zaczyna od ustawienia punktu początkowego (głowy węża) i punktu końcowego (jabłka).

Każdy węzeł na planszy ma przypisane trzy wartości:

- gCost: rzeczywisty koszt dotarcia do tego węzła od punktu początkowego.

- hCost: heurystyczna estymacja kosztu dotarcia z tego węzła do punktu końcowego.
- fCost: suma gCost i hCost, czyli całkowity przewidywany koszt dotarcia z punktu początkowego do punktu końcowego przez ten węzeł ($fCost = gCost + hCost$).

Score 0															
	F:15 G:1 H:14	F:15 G:2 H:13	F:15 G:3 H:12	F:15 G:4 H:11	F:15 G:5 H:10	F:15 G:6 H:9	F:15 G:7 H:8	F:15 G:8 H:7	F:15 G:9 H:6	F:15 G:10 H:5	F:15 G:11 H:4	F:15 G:12 H:3	F:17 G:13 H:4	F:19 G:14 H:5	F:21 G:15 H:6
F:15 G:1 H:14	F:15 G:2 H:13	F:15 G:3 H:12	F:15 G:4 H:11	F:15 G:5 H:10	F:15 G:6 H:9	F:15 G:7 H:8	F:15 G:8 H:7	F:15 G:9 H:6	F:15 G:10 H:5	F:15 G:11 H:4	F:15 G:12 H:3	F:15 G:13 H:2	F:17 G:14 H:3	F:19 G:15 H:4	F:21 G:16 H:5
F:15 G:2 H:13	F:15 G:3 H:12	F:15 G:4 H:11	F:15 G:5 H:10	F:15 G:6 H:9	F:15 G:7 H:8	F:15 G:8 H:7	F:15 G:9 H:6	F:15 G:10 H:5	F:15 G:11 H:4	F:15 G:12 H:3	F:15 G:13 H:2	F:15 G:14 H:1	F:17 G:15 H:2	F:19 G:16 H:3	F:21 G:17 H:4
F:15 G:3 H:12	F:15 G:4 H:11	F:15 G:5 H:10	F:15 G:6 H:9	F:15 G:7 H:8	F:15 G:8 H:7	F:15 G:9 H:6	F:15 G:10 H:5	F:15 G:11 H:4	F:15 G:12 H:3	F:15 G:13 H:2	F:15 G:14 H:1	F:15 G:15 H:0	F:17 G:16 H:1	F:19 G:17 H:2	F:21 G:18 H:3
F:17 G:4 H:13	F:17 G:5 H:12	F:17 G:6 H:11	F:17 G:7 H:10	F:17 G:8 H:9	F:17 G:9 H:8	F:17 G:10 H:7	F:17 G:11 H:6	F:17 G:12 H:5	F:17 G:13 H:4	F:17 G:14 H:3	F:17 G:15 H:2	F:17 G:16 H:1	F:19 G:17 H:2	F:21 G:18 H:3	F:23 G:19 H:4
F:19 G:5 H:14	F:19 G:6 H:13	F:19 G:7 H:12	F:19 G:8 H:11	F:19 G:9 H:10	F:19 G:10 H:9	F:19 G:11 H:8	F:19 G:12 H:7	F:19 G:13 H:6	F:19 G:14 H:5	F:19 G:15 H:4	F:19 G:16 H:3	F:19 G:17 H:2	F:21 G:18 H:3	F:23 G:19 H:4	F:25 G:20 H:5
F:21 G:6 H:15	F:21 G:7 H:14	F:21 G:8 H:13	F:21 G:9 H:12	F:21 G:10 H:11	F:21 G:11 H:10	F:21 G:12 H:9	F:21 G:13 H:8	F:21 G:14 H:7	F:21 G:15 H:6	F:21 G:16 H:5	F:21 G:17 H:4	F:21 G:18 H:3	F:23 G:19 H:4	F:25 G:20 H:5	F:27 G:21 H:6
F:23 G:7 H:16	F:23 G:8 H:15	F:23 G:9 H:14	F:23 G:10 H:13	F:23 G:11 H:12	F:23 G:12 H:11	F:23 G:13 H:10	F:23 G:14 H:9	F:23 G:15 H:8	F:23 G:16 H:7	F:23 G:17 H:6	F:23 G:18 H:5	F:23 G:19 H:4	F:25 G:20 H:5	F:27 G:21 H:6	F:29 G:22 H:7
F:25 G:8 H:17	F:25 G:9 H:16	F:25 G:10 H:15	F:25 G:11 H:14	F:25 G:12 H:13	F:25 G:13 H:12	F:25 G:14 H:11	F:25 G:15 H:10	F:25 G:16 H:9	F:25 G:17 H:8	F:25 G:18 H:7	F:25 G:19 H:6	F:25 G:20 H:5	F:27 G:21 H:6	F:29 G:22 H:7	F:31 G:23 H:8
F:27 G:9 H:18	F:27 G:10 H:17	F:27 G:11 H:16	F:27 G:12 H:15	F:27 G:13 H:14	F:27 G:14 H:13	F:27 G:15 H:12	F:27 G:16 H:11	F:27 G:17 H:10	F:27 G:18 H:9	F:27 G:19 H:8	F:27 G:20 H:7	F:27 G:21 H:6	F:29 G:22 H:7	F:31 G:23 H:8	F:33 G:24 H:9
F:29 G:10 H:19	F:29 G:11 H:18	F:29 G:12 H:17	F:29 G:13 H:16	F:29 G:14 H:15	F:29 G:15 H:14	F:29 G:16 H:13	F:29 G:17 H:12	F:29 G:18 H:11	F:29 G:19 H:10	F:29 G:20 H:9	F:29 G:21 H:8	F:29 G:22 H:7	F:31 G:23 H:8	F:33 G:24 H:9	F:35 G:25 H:10
F:31 G:11 H:20	F:31 G:12 H:19	F:31 G:13 H:18	F:31 G:14 H:17	F:31 G:15 H:16	F:31 G:16 H:15	F:31 G:17 H:14	F:31 G:18 H:13	F:31 G:19 H:12	F:31 G:20 H:11	F:31 G:21 H:10	F:31 G:22 H:9	F:31 G:23 H:8	F:33 G:24 H:9	F:35 G:25 H:10	F:37 G:26 H:11

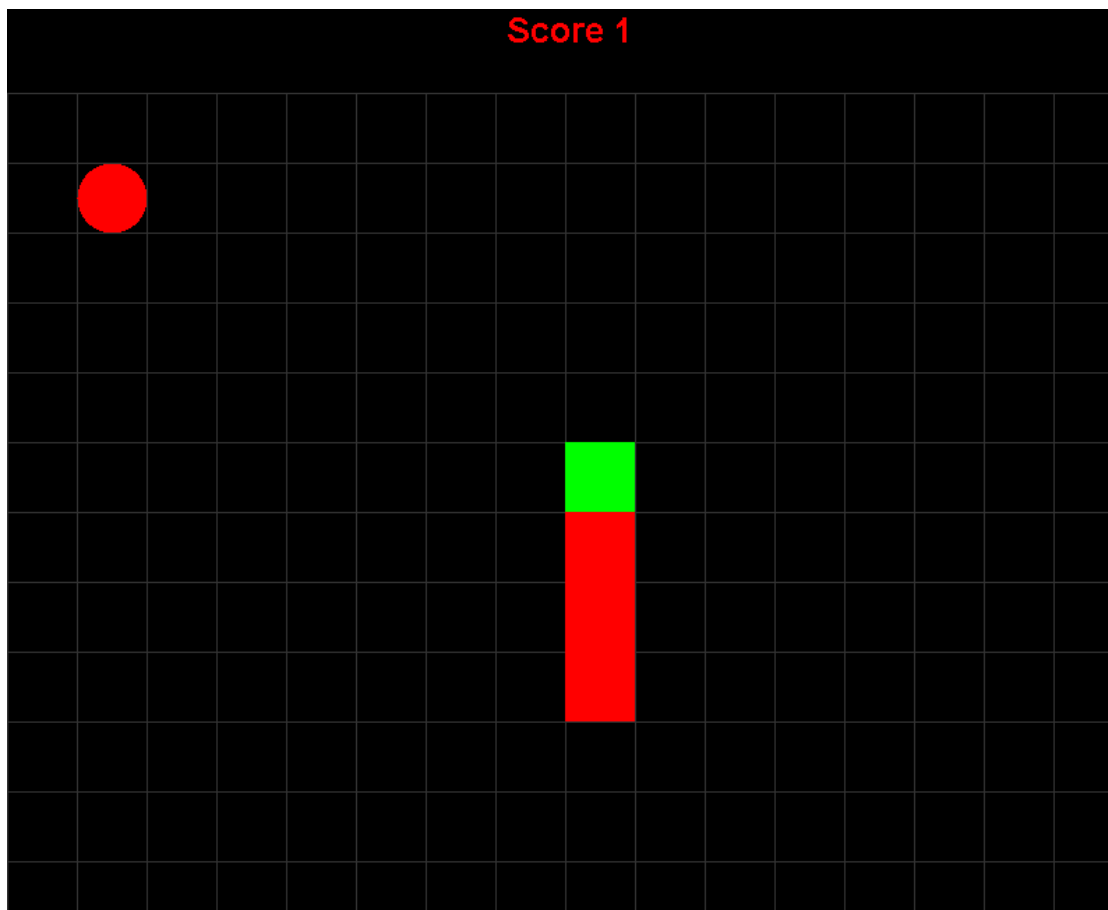
Rysunek 3: Wizualizacja

W każdej iteracji algorytm wybiera węzeł z najniższym fCost z listy otwartej. Ten węzeł jest przenoszony do listy zamkniętej. Sprawdzani są wszyscy sąsiedzi bieżącego węzła: Dla każdego sąsiada obliczane są nowe wartości gCost, hCost i fCost. Jeśli sąsiad nie jest w liście otwartej, jest do niej dodawany. Jeśli sąsiad jest już w liście otwartej, ale nowa ścieżka do niego jest tańsza (mniejszy gCost), aktualizowane są jego koszty i rodzic (poprzednik). Węzły w liście zamkniętej są oznaczone jako odwiedzone, aby algorytm nie wracał się do nich ponownie. Takie

podejście skraca czas przeszukiwania całego "labiryntu" co jest porządane w takich typach gier.

1.5 Ai game

Panel ten pokazuje sprawdza jak poprawność działania algorytmu A*

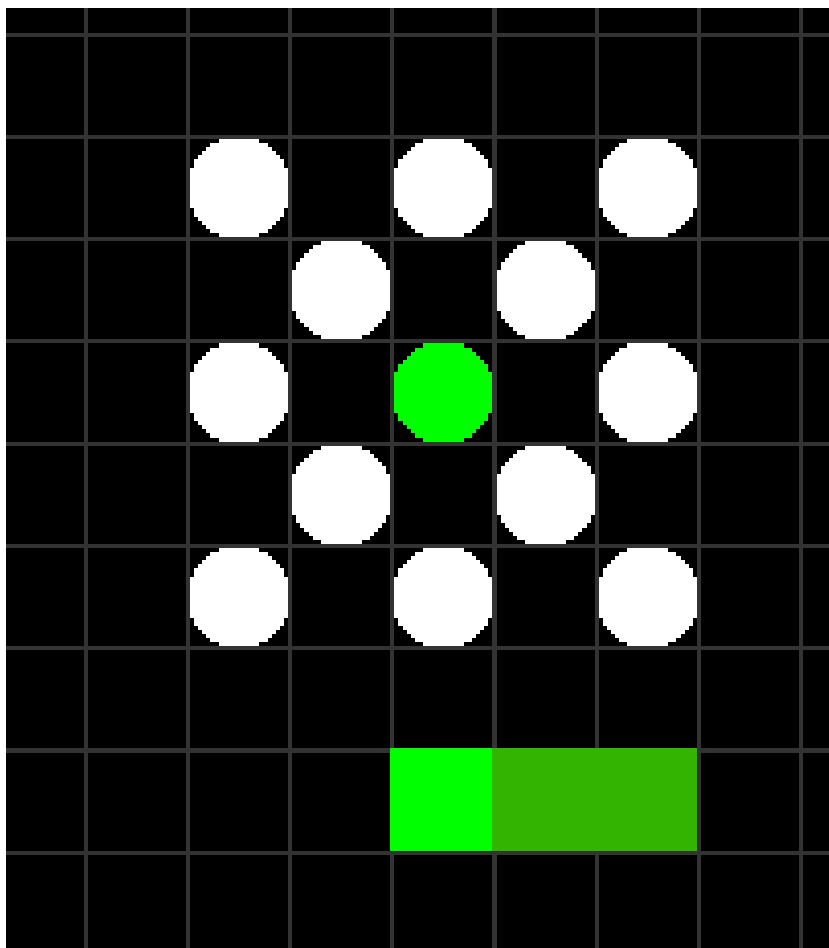


Rysunek 4: Ai Game

Wąż porusza się automatycznie, my tylko możemy go zastopować przyciskiem "Space" lub zakończyć rozgrywkę klikając "Escape"

1.6 Żaba

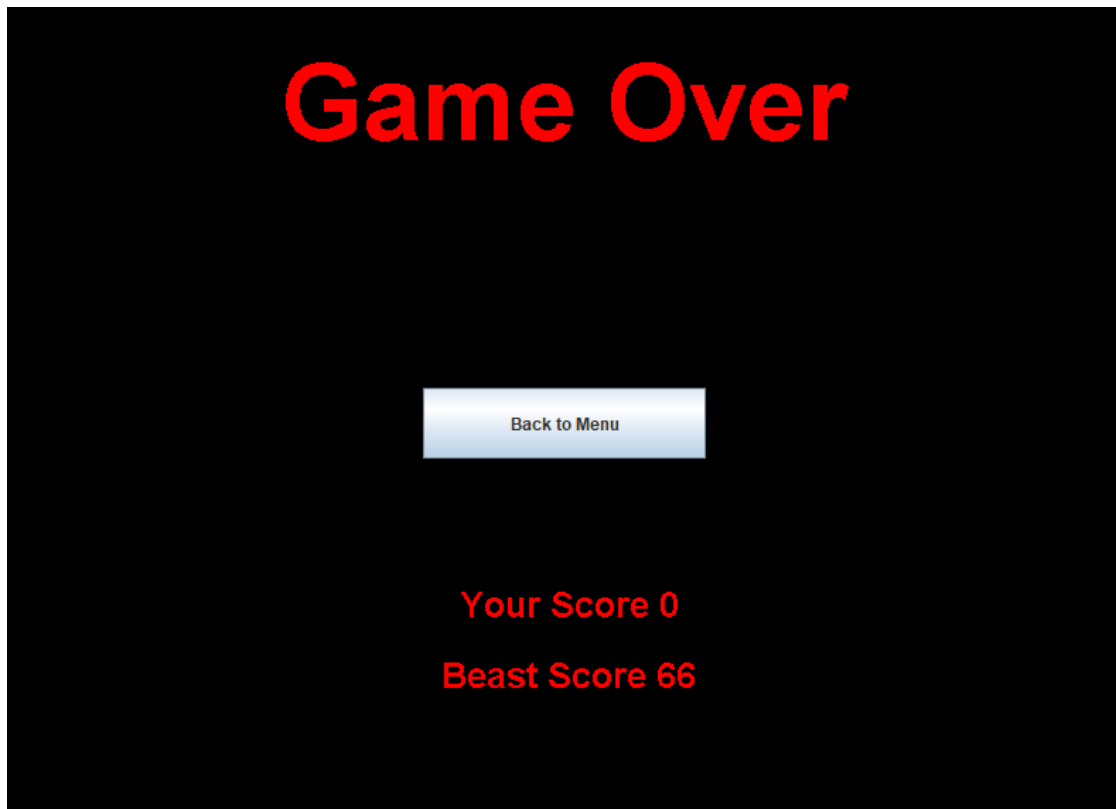
Żaba zawiera checkboxy, gdy jakikolwiek z węży wjedzie w jednego z nich żaba zaczyna uciekać - iść w tym samym kierunku co wąż, gdy nie ma ucieczki - jest pod ścianą zaczyna iść w losowe kierunki.



Rysunek 5: Żaba

1.7 Game Over Panel

Panel Game Over wyświetla informacje o wyniku ostatniej gry jak i najlepszego z pliku tekstowego.



Rysunek 6: Game Over

Przycisk Back to Menu wraca do ekranu głównego i restartuje wszystkie panele wraz ze wszystkimi wartościami.

1.8 Score Panel

Panel wyniku, który wyświetla się podczas gry, jest osobnym panelem aktualizującym się na bieżąco po zebraniu owoców. Aktualizacja panelu wyniku jest możliwa dzięki głównemu frame, który przechowuje i wywołuje odpowiednie panele.

2 Diagram Klas-UML

