

Zadanie: Gra Snake

Opis zadania:

Stwórz aplikację internetową w React, która symuluje klasyczną grę Snake. Gracz steruje wężem poruszającym się po planszy, zbierając jedzenie i unikając zderzenia ze sobą lub ze ścianami. Gra powinna przyspieszać w miarę postępu.

Założenia:

1. Plansza i Wąż:

- Plansza składa się z siatki o stałych wymiarach (np. 20x20).
- Wąż zaczyna jako mały segment (np. długość 3 pola) i porusza się w określonym kierunku (góra, dół, lewo, prawo) za pomocą strzałek na klawiaturze.
- Wąż rośnie za każdym razem, gdy zbierze jedzenie (dodawany jest nowy segment do węża).

2. Jedzenie:

- Jedzenie pojawia się losowo na planszy.
- Po zjedzeniu jedzenia wąż rośnie, a jedzenie pojawia się w nowym, losowym miejscu.

3. Mechanika gry:

- Gra kończy się, gdy wąż uderzy w ścianę lub zderzy się z własnym ciałem.
- W miarę zdobywania punktów prędkość węża rośnie, co zwiększa poziom trudności.
- Wynik jest liczony na podstawie liczby zebranych punktów (każde jedzenie to jeden punkt).

4. Sterowanie:

- Gracz steruje wężem za pomocą klawiatury (strzałki).
- Aplikacja musi reagować na naciśnięcia klawiszy i odpowiednio zmieniać kierunek węża.

5. Restart gry:

- Po zakończeniu gry wyświetlany jest przycisk „Zagraj ponownie”, który resetuje stan gry i umożliwia rozpoczęcie nowej rozgrywki.

6. Stylizacja:

- Wąż i plansza są stylizowane w minimalistyczny sposób – np. wąż jako zielone kwadraty, jedzenie jako czerwony kwadrat.
- Użyj CSS lub Bootstrapa do stylizacji elementów.

Wymagania techniczne:

- Użyj React do stworzenia komponentów gry.
- Wykorzystaj `useState` do przechowywania aktualnego stanu węża (jego segmentów), kierunku ruchu, pozycji jedzenia oraz wyniku.
- Użyj `useEffect` do obsługi animacji węża (aktualizacja jego pozycji w regularnych odstępach czasu).

- Zaimplementuj obsługę klawiatury, reagując na wciśnięcie klawiszy w celu zmiany kierunku węża.
- Dodaj logikę do sprawdzania kolizji (uderzenie w ścianę lub własny ogon).

Dodatkowe wyzwania (opcjonalnie):

- Dodaj tryby gry (np. łatwy, średni, trudny), w których plansza ma różne rozmiary lub prędkości początkowe węża.
- Zaimplementuj poziomy, gdzie wraz z postępem pojawiają się przeszkody na planszy, które wąż musi omijać.
- Dodaj animacje przy zjedzeniu jedzenia lub po zakończeniu gry.
- Przechowuj najlepszy wynik w lokalnej pamięci (`localStorage`) i wyświetlaj go graczowi.