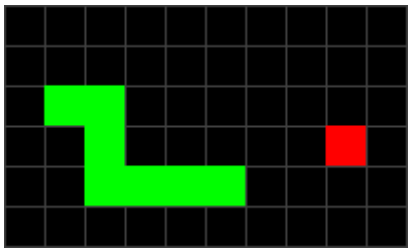
Dokumentacja projektu gry Snake



Autor Krzysztof Garbicz

Nazwa projektu: Projekt końcowy z przedmiotu Język programowania

REV	Data	Zmiany
0.1	25.06.2023	Krzysztof Garbicz kgarbicz@student.agh.edu.pl

Spis treści

1. Wprowadzenie

Gra Snake jest klasyczną grą wideo, w której gracz kontroluje węża poruszającego się po planszy. Celem gry jest zjedzenie jak największej ilości jedzenia, aby zdobyć punkty i unikać kolizji z własnym ciałem oraz granicami planszy. Ten dokument opisuje implementację gry Snake w języku C++ z wykorzystaniem biblioteki SFML (Simple and Fast Multimedia Library).

2. Funkcjonalności

Główne funkcjonalności gry Snake obejmują:

1. **Poruszanie wężem**: Gracz może sterować wężem za pomocą klawiszy strzałek, aby zmieniać kierunek poruszania się węża.
2. **Zbieranie jedzenia**: W grze generowane są jedzenie w postaci czerwonych prostokątów. Gdy wąż zetknie się z jedzeniem, zdobywa punkty, a jedzenie jest ponownie generowane w innym miejscu na planszy.
3. **Kolizja z granicami planszy**: Jeśli głowa węża dotknie granic planszy, gra kończy się.
4. **Kolizja z własnym ciałem**: Jeśli głowa węża zetknie się z dowolną częścią swojego ciała, gra kończy się.
5. **Super jabłko**: Okazjonalnie generowane jest super jabłko w postaci fioletowych migających prostokątów, które może być zebrane przez węża, dodając dodatkowe punkty. Super jabłko jeżeli nie zostanie zebrane w ciągu 10 sekund znika z planszy
6. **Efekty dźwiękowe**: Gra posiada dźwięki dla jedzenia, końca gry, zebrania super jabłka oraz muzykę.

3. Analiza problemu

Celem gry Snake jest zapewnienie interaktywnej i zabawnej rozgrywki, która wymaga refleksu i umiejętności strategicznych od gracza. Główne problemy, które należy rozwiązać podczas implementacji tej gry, to:

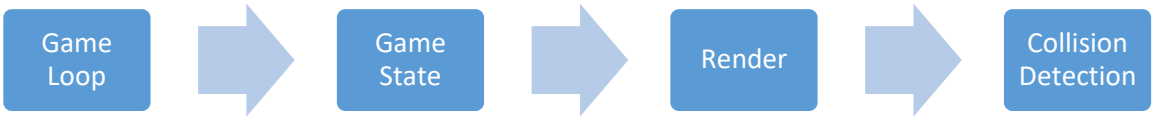
1. **Obsługa interakcji gracza**: Gra musi reagować na ruchy gracza i interpretować naciśnięcia klawiszy strzałek w celu odpowiedniego poruszania wężem.
2. **Generowanie i aktualizacja elementów na planszy**: Jedzenie i super jabłko muszą być generowane w losowych miejscach na planszy, a ich położenie powinno być aktualizowane w odpowiednich momentach.
3. **Kolizje**: Gra musi sprawdzać kolizje między głową węża, jego ciałem, granicami planszy oraz jedzeniem i super jabłkiem.
4. **Punkty i wynik**: Gra musi śledzić liczbę zdobytych punktów i wyświetlać wynik dla gracza.
5. **Wyświetlanie grafiki i tekstu**: Gra musi wyświetlać planszę, węża, jedzenie, super jabłko, tekst informacyjny i inne elementy graficzne w oknie gry.

4. Struktura projektu

Główna struktura projektu gry Snake obejmuje następujące komponenty:

1. **Game** - klasa reprezentująca samą grę. Odpowiada za logikę gry, inicjalizację, obsługę zdarzeń i aktualizację stanu gry.
2. **Snake** - klasa reprezentująca węża. Odpowiada za poruszanie wężem, sprawdzanie kolizji i aktualizację ciała węża.
3. **Food** - klasa reprezentująca jedzenie. Odpowiada za generowanie zwykłych jabłek, sprawdzanie kolizji z wężem i aktualizację położenia zwykłych jabłek.
4. **SuperApple** - klasa reprezentująca super jabłko. Odpowiada za generowanie super jabłka, sprawdzanie kolizji z wężem i aktualizację położenia jabłka.
5. **GameWindow** - klasa reprezentująca okno gry. Odpowiada za wyświetlanie grafiki, obsługę zdarzeń klawiatury i aktualizację wyświetlanego stanu gry.
6. **SoundManager** - klasa reprezentująca menadżera dźwięku. Odpowiada za odtwarzanie efektów dźwiękowych w grze.

4.1 Architektura



(1)Handles User Input (2) Updates State (3) Render Graphics

1. **Game Loop:** Główna pętla gry, która odpowiedzialna jest za zarządzanie przepływem gry i komunikację między poszczególnymi modułami.

2. **Game State**: Moduł, który przechowuje stan gry, takie jak położenie węża, położenie pożywienia, wynik gracza itp. Odpowiada za aktualizację stanu gry na podstawie informacji z modułu obsługującego wejście użytkownika.

3. **Renderer**: Moduł odpowiedzialny za renderowanie grafiki na ekranie. Korzysta z informacji dostarczonych przez moduł stanu gry i rysuje elementy gry, takie jak wąż, pożywienie, tło itp.

4. **Collision Detection**: Moduł odpowiedzialny za wykrywanie kolizji między elementami gry, takimi jak wąż, ściany, pożywienie itp. Na podstawie wyników wykrywania kolizji podejmowane są odpowiednie akcje, takie jak zwiększenie wyniku, zakończenie gry itp.

5. Opis realizacji

Gra Snake została zaimplementowana w języku C++ z wykorzystaniem biblioteki SFML (Simple and Fast Multimedia Library), która zapewnia interfejs do obsługi grafiki, dźwięku i interakcji z użytkownikiem.

6. Instrukcje

Aby rozpocząć grę Snake, gracz powinien uruchomić program. Po uruchomieniu pojawi się okno gry, w którym gracz może zacząć sterować wężem za pomocą klawiszy strzałek. Głównym celem gry jest zdobycie jak największej liczby punktów poprzez zbieranie jedzenia i unikanie kolizji. Po przegranej rozgrywce gracz może rozpocząć kolejną rozgrywkę naduszając klawisz Enter. Gra rozpoczyna się z początkowymi ustawieniami.

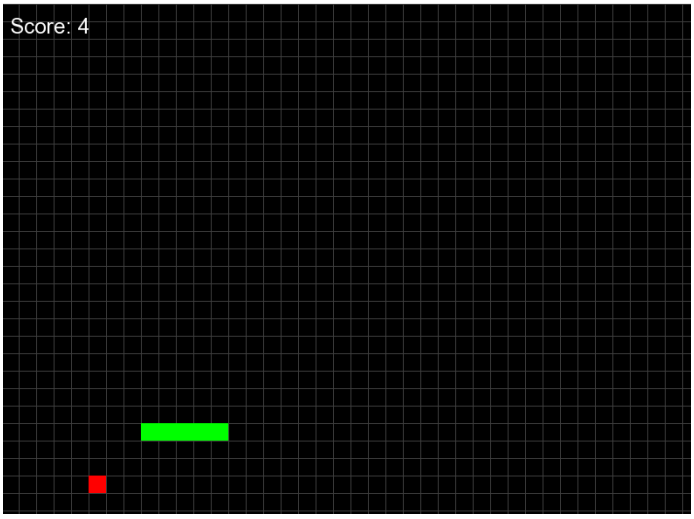
6.1 Sterowanie

Gracz może sterować wężem za pomocą klawiszy strzałek:

- Strzałka w górę: Porusza wężem do góry.
- Strzałka w dół: Porusza wężem w dół.
- Strzałka w lewo: Porusza wężem w lewo.
- Strzałka w prawo: Porusza wężem w prawo.

6.2 Punkty i wynik

Gra śledzi liczbę zdobytych punktów. Za każde zebranie jedzenia gracz otrzymuje 1 punkt, a za zebranie super jabłka otrzymuje 5 punktów. Wynik jest wyświetlany w prawym górnym rogu na ekranie gry.



Zrzut ekranu 1 z przykładowej rozgrywki

6.3 Kolizje

Gra sprawdza kolizje, które mogą wystąpić podczas rozgrywki:

- Kolizja z własnym ciałem: Jeśli głowa węża zetknie się z dowolną częścią swojego ciała, gra kończy się, a wyświetlany jest ekran końca gry.
- Kolizja z granicami planszy: Jeśli głowa węża dotknie granic planszy, gra kończy się, a wyświetlany jest ekran końca gry.
- Kolizja z jedzeniem: Jeśli głowa węża dotknie jedzenia, wąż rośnie o jedną jednostkę, a jedzenie zostaje zrespawnowane[1] w losowym miejscu na planszy.
- Kolizja z super jabłkiem: Jeśli głowa węża dotknie super jabłka, wąż rośnie o trzy jednostki, a super jabłko zostaje zrespawnowane w losowym miejscu na planszy.

[1] Termin często pojawiający się w grach, oznacza ponowne pojawienie się obiektu.

7. Metodologia rozwoju

Podczas tworzenia gry Snake zastosowano iteracyjny model rozwoju oprogramowania, który składał się z następujących etapów:

- **Analiza wymagań**: Na początku przeprowadzono analizę wymagań, aby zrozumieć funkcjonalności i zachowania oczekiwane od gry Snake. Zidentyfikowano główne elementy, takie jak sterowanie, logika gry, kolizje, punktacja itp.

- **Projektowanie**: Następnie przystąpiono do projektowania architektury gry. Określono strukturę klas, relacje między nimi oraz przepływ danych. Wybrano również bibliotekę SFML jako podstawę do obsługi grafiki, dźwięku i interakcji z użytkownikiem.

- **Implementacja**: Po zakończeniu etapu projektowania przystąpiono do implementacji poszczególnych komponentów gry. Każda klasa została starannie zaimplementowana, uwzględniając odpowiednie metody, funkcje i zmienne.

- **Testowanie**: Po zaimplementowaniu poszczególnych komponentów przeprowadzono testowanie, aby sprawdzić poprawność działania gry. Wykryte błędy i niedoskonałości były poprawiane, a proces testowania był powtarzany, aż do osiągnięcia oczekiwanego poziomu jakości.

- **Optymalizacja i poprawki**: Po przeprowadzeniu testów i uzyskaniu działającej gry, przystąpiono do optymalizacji kodu i poprawek. Zidentyfikowano miejsca, w których można było zoptymalizować działanie gry, oraz wprowadzono niezbędne poprawki.

- **Iteracja**: Cały proces, począwszy od analizy wymagań, projektowania, implementacji, testowania i poprawek, był powtarzany iteracyjnie, dopóki nie osiągnięto zamierzonego rezultatu.

8. Bibliografia

W trakcie tworzenia gry Snake korzystano z różnych źródeł informacji i materiałów pomocniczych. Poniżej znajduje się lista niektórych z tych źródeł:

1. Dokumentacja SFML - Oficjalna dokumentacja biblioteki SFML, dostępna na stronie: https://www.sfml-dev.org/documentation/

2. Strona internetowa C++ - Strona internetowa z dokumentacją języka programowania C++, zawierająca informacje na temat składni, bibliotek i dobrych praktyk: https://en.cppreference.com/

3. Artykuł na Wikipedii poświęcony grze komputerowej Snake: https://en.wikipedia.org/wiki/Snake_(video_game_genre)

9. Podsumowanie

Dokumentacja opisuje strukturę projektu gry Snake oraz instrukcje dotyczące sterowania, punktacji i kolizji. Gra została zaimplementowana w języku C++ przy użyciu biblioteki SFML. Wszystkie klasy i komponenty zostały opisane, a gracz może rozpocząć rozgrywkę, uruchamiając program. Powodzenia i miłej zabawy!