

Projekt "Wieloosobowa gra czasu rzeczywistego"

Autorzy: Damian Krzyżelewski 151796, Michał Kulawiak 151840

Cel projektu

Celem projektu było stworzenie prostej gry wieloosobowej czasu rzeczywistego w stylu Agar.io, wykorzystując Pythona, Pygame (po stronie klienta) i C++ (po stronie serwera). Gra umożliwia graczom kontrolowanie kolorowych obiektów- piłek, zbieranie punktów i rywalizację z innymi graczami.

Struktura Projektu

Serwer (C++)

Klasa Game: Zarządza stanem gry, w tym informacjami o graczach, małych piłkach i kolizjach.

Funkcja Wątku Klienta: Obsługuje każde połączenie klienta w osobnym wątku, odbiera dane wejściowe gracza, aktualizuje stan gry i wysyła informacje z powrotem.

Funkcja main: Konfiguruje gniazdo serwera, nasłuchuje i uruchamia wątki dla każdego klienta tworząc dla każdej trójki graczy nową instancję gry.

Klient (Python)

Klasa Canvas: Obsługuje okno Pygame i rysowanie.

Klasa Player: Reprezentuje gracza w grze, z metodami do rysowania i poruszania się.

Klasa Game: Zarządza logiką gry, w tym interakcjami graczy, kolizjami i rysowaniem.

Klasa Network: Obsługuje komunikację z serwerem, wysyłając dane gracza i odbierając aktualizacje stanu gry.

W projekcie przesyłane są dane w formie tekstu między klientami a serwerem. Format przesyłanych danych obejmuje różne informacje, takie jak pozycje graczy, promienie graczy, pozycje małych piłek itp. Dane są przesyłane, jako ciągi znaków za pomocą protokołu tekstowego. Poniżej przedstawiono ogólny format przesyłanych danych:

1. Server wysyła początkowe pozycje graczy i kolory w formacie:
Np. 0:20,20,10|1:400,20,10|2:20,400,10?143,291|317,396@8,163,18|250,176,2|2,246,250
Gdzie:
 - 0:20,20,10|1:400,20,10|2:20,400,10 to początkowe pozycje graczy oddzielone | w formacie:
 - Id_Gracza:pozycja_X,pozycja_Y,radius
 - 143,291|317,396 to pozycje X i Y małych piłek do zjadania

- 8,163,18|250,176,2|2,246,250 to kolory kolejnych graczy oddzielone | w formacie r,g,b
2. Klient wysyła serwerowi tylko swoją pozycję w podobnym formacie:
id_gracza:pozycja,pozycja,radius
 3. Server odbiera od klienta jego pozycję, aktualizuje ją i odpowiada mu w formacie jak w pkt. 1 tylko bez informacji o kolorach, która była potrzebna tylko do inicjalizacji gry.

Jak Uruchomić Projekt

Należy mieć zainstalowaną bibliotekę Pygame.

Uruchom serwer: Skompiluj i uruchom plik serwera w C++.

Uruchom klienta: W network.py ustaw adres serwera i odpowiedni numer portu. Uruchom plik klienta w Pythonie. Każda gra rozpocznie się po podłączeniu się do niej trzech klientów.

Podsumowanie i wnioski

Projekt dostarczył nam cenne doświadczenie w programowaniu gier wieloosobowych oraz zarządzaniu komunikacją między klientem a serwerem. Jest dobrą bazą, która mogłaby być rozszerzana o dodatkowe funkcje, takie jak obsługa lobby, zapisywanie wyników, czy zaimplementowanie bardziej zaawansowanych mechanizmów znanych z gry Agar.io.