

Programowanie współbieżne

Projekt

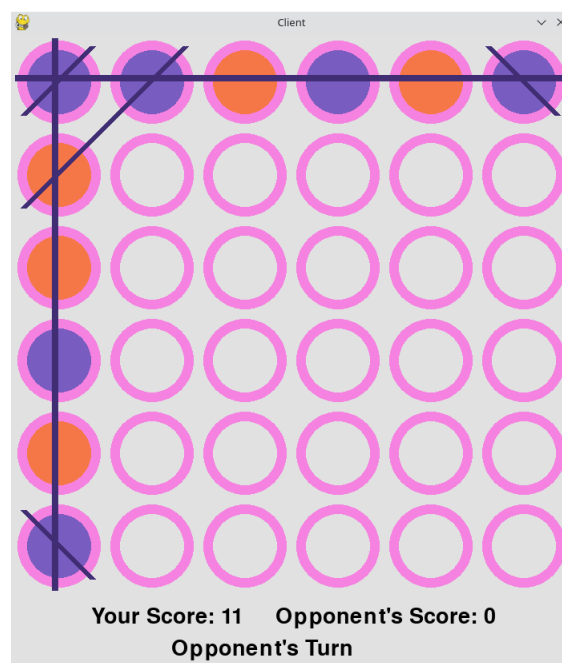
Celem projektu jest stworzenie gry multiplayer "Ziemniaki" w oparciu o architekturę klient-serwer. Program ma zapewnić interfejs graficzny dla graczy oraz centralny serwer, którego celem jest zarządzanie sesjami gry. Istotne jest również zabezpieczenie przed konfliktami dostępu do planszy oraz koordynacja ruchów graczy.

1. Zasady gry:

- każdy gracz posiada swój kolor
- gracze naprzemiennie zamalowują kolejne pola swoim kolorem
- gracz który jako ostatni zamaluje pole w linii przez całą planszę (pionowo, poziomo lub ukośnie) dostaje tyle punktów ile było pól jego koloru w danej linii (jeśli jednocześnie kończymy kilka linii to sumujemy punkty z wszystkich)
- ten kto zdobędzie najwięcej punktów wygrywa

2. Zastosowane rozwiązanie:

Komunikacja między klientami a serwerem będzie odbywać się przy użyciu gniazd TCP z wbudowanego modułu python **sockets**. Umożliwi to stałe połączenie i obustronną wymianę danych między klientami a serwerem. Do obsługi wielu równoległych połączeń wykorzystana zostanie klasa **Thread** z wbudowanego modułu **threading**. Każdy gracz po nawiązaniu połączenia zostanie przypisany do nowej lub już istniejącej sesji gry. Utworzony zostanie dla niego nowy wątek oraz przypisany numer gracza, co umożliwi w dalszej kolejności synchronizację dostępu do planszy. Po rozpoczęciu gry gracz może zmienić stan gry tylko i wyłącznie, gdy wyznaczona jest jego kolej.



3. Opis użytkowania programu:

Po uruchomieniu aplikacji pojawia się okno z informacją o kliknięciu w celu połączenia z serwerem. Następnie pojawia się widok planszy lub informacja o oczekiwaniu na połączenie przeciwnika. Pod widokiem planszy widnieje informacja o wyniku gracza oraz przeciwnika. W przypadku oczekiwaniu na ruch przeciwnika, również pojawia się stosowna informacja. Po zamalowaniu całej planszy pojawia się informacja o wygranej lub przegranej, a następnie po kilku sekundach gra zostaje zresetowana i rozpoczęta na nowo.

Wszystkie kliknięcia w planszę walidowane są zarówno po stronie serwera jak i po stronie klienta, dlatego do serwera wysyłane są tylko prawidłowe ruchy gracza, gdyby jednak użytkownik spróbował wysłać nieprawidłowy ruch manualnie, zostanie on zwalidowany i odrzucony przez serwer.