Mateusz Wyrzykowski 155883

Kacper Molenda 156014

Sprawozdanie sk2

Gra karciano-zręcznościowa „Buła Pizza Kot Ser Koza”

1. Opis projektu

Projekt polegał na stworzeniu wieloosobowej gry „zręcznościowej”, czyli takiej gdzie gracze jednocześnie muszą wykonać jakąś czynność, aby wygrać. Naszym wyborem została gra „Buła Pizza Kot Ser Koza”. W grze każdy gracz ma swój stos kart, gdzie każda z kart odpowiada którejś ze słów z tytułu gry. Gracze nie widzą co znajduje się w ich stosie. Zaczynając od pierwszego gracza wykładane są karty na wspólny stos na środku stołu, jednocześnie wypowiadając po słowa z tytułu gry w tej samej kolejności. Gdy wyłożona karta będzie zgadzać się ze słowami rymowanki, wszyscy gracze muszą jak najszybciej „zaklepać” wspólny stos. Ostatnia osoba, która to zrobi przegrywa i zbiera wszystkie karty ze wspólnego stosu do swojego stosu kart. Wygrywa osoba, która pozbędzie się wszystkich kart.

Projekt składa się z serwera napisanego w C oraz z klienta napisanego w Python. Komunikują się one za pomocą protokołu TCP.

Serwer napisany w C obsługuje wielowątkowość (pthread), gniazda TCP (socket, bind, listen, accept, send, recv), synchronizacji dostępu do danych (pthread_mutex, pthread_cond) oraz obsługę danych gry (tablica GameRoom) do zarządzania stanem pokoju, graczy, kart i tur.

Klient napisany w Python obsługuję interfejs graficzny zbudowany w Tkinter oraz gniazda TCP (socket).

2. Opis komunikacji pomiędzy klientem a serwerem

Komunikacja pomiędzy serwerem a klientem w tym projekcie opiera się na wymianie prostych wiadomości tekstowych. Każda wiadomość zawiera instrukcje lub informacje, które są przetwarzane przez serwer lub klienta.

2.1 Nawiązanie połączenia

Klient łączy się z serwerem poprzez socket TCP, używając IP serwera (127.0.0.1) oraz portu serwera (1100). Po nawiązaniu połączenia wysyła swoją nazwę gracza.

Serwer odbiera nazwę gracza i zapisuje ją w strukturze GameRoom. Sprawdza, czy istnieje dostępny pokój z wolnym miejscem, który nie zaczął jeszcze gry. Jeśli znajdzie taki pokój, to dodaje gracza do niego, zapisując dane w strukturze (player_sockets – gniazdo TCP klienta, player_names – nazwa gracza, player_count – ilość graczy w pokoju). Na koniec wysyła komunikat do gracza o dołączeniu do pokoju.

2.2 Komunikaty klienta do serwera

PLAYER_NAME:<nazwa_gracza> - Klient wysyła swoją nazwę użytkownika.

ZAGRAJ_KARTE - Klient informuje serwer, że chce zagrać kartę w swojej turze.

ZAKLEP - Klient próbuje zaklepać stos, gdy karta pasuje do rymowanki.

WYJSCIE - Klient informuje serwer, że chce opuścić grę.

2.3 Komunikaty serwera do klienta

LISTA_GRACZY:<lista_graczy_w_grze> - Serwer wysyła aktualną listę graczy w pokoju z ich nazwami i liczbą kart.

CARDS_COUNT:<n> - Informacja o liczbie kart w ręce gracza.

AKTUALNA_KARTA:<karta> - Serwer informuje o karcie zagranej na stos.

AKTUALNA_RYMOWANKA:<rymowanka> - Serwer wysyła aktualną rymowankę, która powinna być dopasowana.

SERWER:<wiadomość> - Ogólne komunikaty, np. o rozpoczęciu gry, zakończeniu tury lub wyjściu gracza.

GRATULACJE! WYGRAŁEŚ GRĘ! - Komunikat informujący ostatniego gracza, że wygrał grę.

3. Podsumowanie

Serwer obsługuje do 10 gier jednocześnie. W grze może znajdować się od 2 do 3 graczy (na start zawsze 3 graczy). Po stronie serwera należy rozdawanie i tasowanie kart, obsługa tur graczy, zaklepywanie kart, przywracanie pokoi graczy do pierwotnego stanu po rozegraniu gry. Równoczesna obsługa kilku graczy oparta jest na wątkach POSIX, co pozwala na komunikację z wieloma klientami na raz. Serwer obsługuje również rozłączenia graczy (gracz opuszcza grę, jego karty są rozdawane pomiędzy resztę graczy, a tura gracza przechodzi automatycznie do kolejnego aktywnego gracza).

Największymi problemami okazały się rozłączenia graczy (mechanizm przesuwania kolejki i redystrybucji kart działa w czasie rzeczywistym, uwzględniając tylko aktywnych graczy), obsługa wielu klientów (wielowątkowość) oraz połączenie turowości rozgrywki z jej zręcznościową częścią (dwa wątki – jeden od turowej rozgrywki, drugi od zręcznościowego zaklepywania stosu).