Mateusz Napiórkowski 145251

Sprawozdanie sk2

Pan (gra karciana)

1. Opis projektu

Tematem projektu jest gra karciana Pan. Zasady gry zaimplementowanej w ramach projektu w dużej mierze pokrywają się z tymi, które można znaleźć na następującej stronie: <https://www.kurnik.pl/pan/zasady.phtml>. Zasadniczą różnicą jest to, że w mojej implementacji rozgrywki odbywać się mogą tylko między dwoma graczami oraz przegrany nie zdobywa kolejno literek P, A i N po kolejnych przegranych, lecz rozgrywana jest tylko jedna runda, na podstawie której od razu wyłaniany jest zwycięzca i przegrany.

Projekt został zrealizowany w architekturze klient-server. Serwer napisany został w języku C i wykorzystuje API bsd-sockets do komunikacji z klientami. Kod klienta napisany został w języku Python. Do komunikacji z serwerem wykorzystowane są metody z modułu socket, dostępnego w bibliotece standardowej języka. Klient posiada graficzny interfejs stworzony przy użyciu biblioteki Pygame.

2. Opis komunikacji pomiędzy serwerem i klientem

W celu umożliwienia komunikacji między serwerem a klientami po stronie serwera tworzony jest socket. Komunikacja odbywa się za pomocą protokołu IP w wersji czwartej, a za transmisję odpowiada protokół TCP. Usługa działa na porcie 1100. W celu zbindowania socketu z adresem i portem serwera wykorzystywana jest funkcja bind. Następnie uruchamiana jest funkcja listen w celu oczekiwania na połączenie z klientami. Serwer oczekuje na połączenie z dwoma klientami i akceptuje je. Deskryptory socketów obu klientów są następnie przekazywane w strukturze do funkcji uruchamianej w osobnym wątku, co możliwe jest dzięki funkcji pthread\_create.

Po uruchomieniu klienta również tworzy on jest socket, a następnie łączy się z serwerem za pomocą metody connect.

Do komunikacji z klientami serwer wykorzystuje następujące struktury danych:

* 13-bajtowa tablica, wykorzystywana na samym początku rozgrywki. Serwer losuje 12 kart dla każdego z graczy i zapisuje ich wartości na pierwszych 12 bajtach. Każdej z dostępnych w grze 24 kart przypisana jest wartośc liczbowa z przedziału 0-23. Na ostatnim bajcie zapisywana jest wartość 1 lub 2, by poinformować danego klienta czy w ramach rozgrywki jest on graczem nr 1 czy graczem nr 2.
* 3-bajtowa tablica. Pierwszy bajt o wartości 1 lub 2 informuje klientów, który z graczy może wykonać ruch. Na końcu rozgrywki bajt przyjmuje wartości 3 lub 4, gdzie 3 informuje klientów, że zwyciężył gracz 1, a 4 informuje o wygranej gracza 2. Drugi bajt informuje klientów jaka karta znajduje się aktualnie na szczycie stosu, na króry w ramach rozgrywki dokładane lub zabierane są karty. Bajt przyjmuje wartości z zakresu 0-24, gdzie 24 oznacza, że na stosie nie znajduje się jeszcze żadna karta. Trzeci bajt przyjmuje wartości z zakresu 0-24, by poinformować gracza ile kart posiada w danym momencie przeciwnik.
* 1 bajt o wartości 0 lub 1 informujący klienta czy ruch, który wykonał jest poprawny czy nie.
* 3-bajtowa tablica zawierająca wartości z przedziału 0-24. Według zasad gry, gracz może pobrać ze stosu 3 karty. Tablica informuje klienta, które z 24 dostępnych w grze kart zostały pobrane. Istnieje wyjątek, gdy gracz może pobrać mniej niż 3 karty i wtedy na pozostałych bajtach znajdzie się wartość 24.

Do komunikacji z serwerem klient wykorzystuje następującą strukturę danych:

* 25 bajtowa lista, służąca do informowania serwera o wykonanym ruchu. Pierwsze 24 bajty zawierają wartości binarne informujące, które z 24 kart zostały wybrane przez gracza. Oczywiście gracz zazwyczaj posiada mniej niż 24 karty, ale nie stanowi to problemu i na bajtach reprezentujących karty, których gracz nie posiada, znajdą się wartości 0. Jeśli ostatni bajt listy przyjmie wartość jeden, to oznacza to, że gracz zamiast dołożenia kart ze stosu chce je z niego pobrać.

3. Podsumowanie

Główny wątek serwera działa w pętli oczekując stale na 2 połączenia, by móc utworzyć oddzielny wątek obsługujący rozgrywkę między 2 graczami. Dzięki temu odbywać się może w tym samym czasie wiele rozgrywek naraz. Wykrycie i obsługa rozłączenia gracza nie została zaimplementowana z powodu błędów w zarządzaniu czasem podczas tworzenia projektu.

Interfejs graficzny klienta został napisany z pomocą modułu Pygame. W oknie programu wyświetlają się modele 2d takie jak: karty gracza, karta na szczycie stosu, karty przeciwnika(odwrócone), etykiety graczy i przyciski zatwierdzające wybrany przez gracza ruch. Mimo, że tworzenie interfejsu w tej bibliotece jest dość proste na tle rozwiązań z innych języków programowania, to przed implementacją właściwej logiki projektu dużą część czasu przeznaczoną na projekt zajęło zrozumienie działania tej biblioteki.

Jedną z innych niespodziewanych trudności było napisanie kodu po stronie serwera służącego do losowania 12 unikalnych kart dla każdego z graczy. W języku Python wystarczyłoby do tego wykorzystać metodę sample z modułu random, jednak język C nie posiada w swoich standardowych bibliotekach analogicznej funkcji.