

Spis treści

Memorsy!	1
Główne założenia protokołu.....	1
Połączenie, rozłączenie.....	1
Sposób obsługi klientów przez serwer	2
Przepływ danych w aplikacji	2
Rodzaje pakietów	2
Generowanie mapy	2
Przykładowa rozgrywka	2
Logowanie.....	2

Memorsy!

Program jest realizacją gry w „memories”. Gra polega na znalezieniu na planszy dwóch pasujących do siebie kart.

Aplikacja umożliwia grę na dwóch poziomach (6 ,12 kart w talii).

Główne założenia protokołu

Protokół realizowany jest z użyciem szyfrowanego połączenia.

Wszystkie komunikaty są kończone znakiem końca linii – '\r\n'.

Tablice są formatowane za pomocą biblioteki pickle¹.

Strony komunikacji porozumiewają się za pomocą kodów : mapowane z pliku 'codes.py',

poza wyjątkiem gdy klient chce opuścić rozgrywkę, nie wysyła kodu dla "QuitGame" tylko komunikat tekstowy

Takie rozwiązanie eliminuje konflikt z zapytaniem o treść karty, która przysłana jest poprzez zapytanie zależne od pozycji karty.

Połączenie, rozłączenie

Komunikacja jest realizowana z użyciem szyfrowanego połączenia. Po połączeniu, klient pobiera i sprawdza ważność certyfikatu serwera. Jeśli jest ważny, nawiązywane jest połączenie szyfrowane, realizowane przy użyciu modułu ssl.²

Jeśli klient chce rozłączyć się z serwerem, wysyłamy kod komunikatu 'QuitGame'. (Poza wyjątkiem opisanym w założeniach).

Po zakończeniu, zerwaniu połączenia z serwerem klient jest usuwany z listy.

W przypadku błędu serwera (zerwania połączenia), aplikacja klienta zamyka się.

¹ <https://docs.python.org/2/library/pickle.html>

² <https://docs.python.org/2/library/ssl.html>

Sposób obsługi klientów przez serwer

- Każdy klient jest obsługiwany w niezależnym wątku.
- Dla każdego klienta generowana jest mapa zależna od poziomu, klient generuje pustą mapę.
- Mapa przechowywana jest tylko po stronie serwera, klient wysyła zapytania o zawartość kart.
- Klient wysyła tablice z pozycjami kart, z zapytaniem czy do siebie pasują.
- Gdy skończą się karty tj. klient odgadnie wszystkie możliwe pary, serwer wysyła komunikat informujący o wygranej.
- Po skończeniu gry, klient ma możliwość zaczęcia kolejnej rozgrywki.

Przepływ danych w aplikacji

Rodzaje pakietów

Protokół wykorzystuje trzy różne sposoby przekazywania informacji.

- Informacja o wybranym poziomie rozgrywki jest przekazywana jako wartość typu integer.
- Zapytanie o kartę, należy przekazać string zawierający pozycję karty.
- Sprawdzenie dwóch kart – tablica id: [id, id], przekształcona za pomocą modułu pickle³.
- Wszystkie komunikaty przekazywane są za pomocą kodów stanu (plik 'codes.py').

Generowanie mapy

- Mapa jest generowana z losowo wybranych kart z pliku '/store/cards.txt'.
- Karty losowane są przy pomocy modułu random⁴.

Przykładowa rozgrywka

Klient – C, Serwer – S.

- K wysyła wybrany poziom
- S odpowiada czy przygotował grę, jeśli tak obie strony generują mapy
- K wysyła zapytania o zawartość kart
- S odpowiada tekstem.
- K wysyła parę kart [id, id]
- S odpowiada czy karty do siebie pasują.
- Gdy wszystkie karty zostały odkryte – S wysyła komunikat „GameOver”
- Klient kończy rozgrywkę, lub rozpoczyna nową (wysyła - „QuitGame” / „NewGame”).
- NewGame -> pkt 1 / QuitGame -> zamknięcie połączenia z C

Logowanie

Wszystkie czynności podejmowane w aplikacji, błędy, ostrzeżenia logowane są do plików.

- server.log - serwer
- client.log - klient

³ <https://docs.python.org/2/library/pickle.html>

⁴ <https://docs.python.org/2/library/random.html>