



Универзитет „Св. Кирил и Методиј“ – Скопје
Факултет за Електротехника и Информациски Технологии



Лист, камен, ножички

Семинарска работа по предметот Мрежно Програмирање



Изработиле:

Радица Јованова 5/2020

Стефани Павлеска 105/2020

Предметни професори:

проф. д-р Марија Календар

проф. д-р Марко Порјазоски

Скопје, 2024

Идејата на нашата семинарска беше да покажеме имплементација на клиент-сервер комуникација и употреба на сокети во реален и интересен пример.

1. Фајл `server.py`

Во фајлот `server.py` имплементиран е сервер за игра со повеќе играчи користејќи сокети и нитки. Прво ја дефинираме IP адресата на сокетот и портата, а потоа го креираме сокетот користејќи IPv4 адресирање (`AF_INET`) и TCP протоколот (`SOCK_STREAM`). Откога серверот ќе почне да работи печати порака дека чека конекции. Следно дефинираме функција `“threaded_client”` која е одговорна за обработка на комуникацијата со клиентот во посебна нитка. Додека пак функцијата `“accept_connections”` е одговорна за прифаќање на дојдовни конекции од клиентите и работа со нив.

2. Фајл `network.py`

Во фајлот `network.py` дефинирана е класата `Network` која претставува интерфејс за мрежна комуникација на клиентска страна во клиент-сервер архитектура. Првата функција `“__init__”` е конструктор кој ги иницијализира атрибутите на класата кога се креира објект од класа `Network`. Следната функција `“getP”` враќа бројка на играчот кој е доделен на клиентот. Функцијата `“connect”` е одговорна за поврзување на клиентот со серверот. Последната функција `“send”` е одговорна за испраќање податоци до серверот и примање одговор.

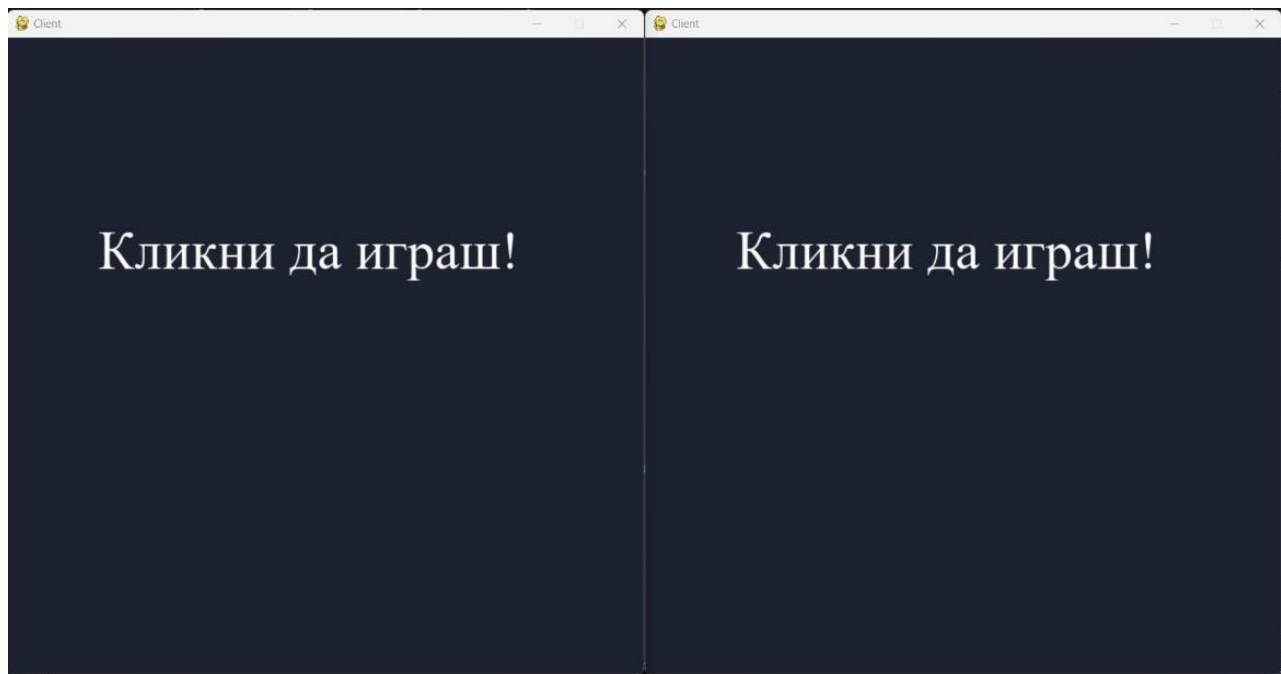
3. Фајл `game.py`

Овој фајл ја содржи класата `Game` каде што е прикажана логиката на играта. Прво имаме конструктор каде што при креирање на објект од класата, се иницијализираат атрибутите. Функцијата `“get_player_move(self, p)”` го враќа потегот на играчот `p`, додека пак функцијата `“play(self, player, move)”` се користи за извршување на потег од страна на играчот `player`. `“connected”` функцијата кажува дали може да започне играта односно дали се поврзани два играчи. `“bothWent”` функцијата кажува дали двата играчи веќе направиле потег. Функцијата `“winner”` одредува кој играч е победник според потегот кој што е направен, `“resetWent”` функцијата се користи за ресетирање на потезите на играчите.

4. Фајл client.py

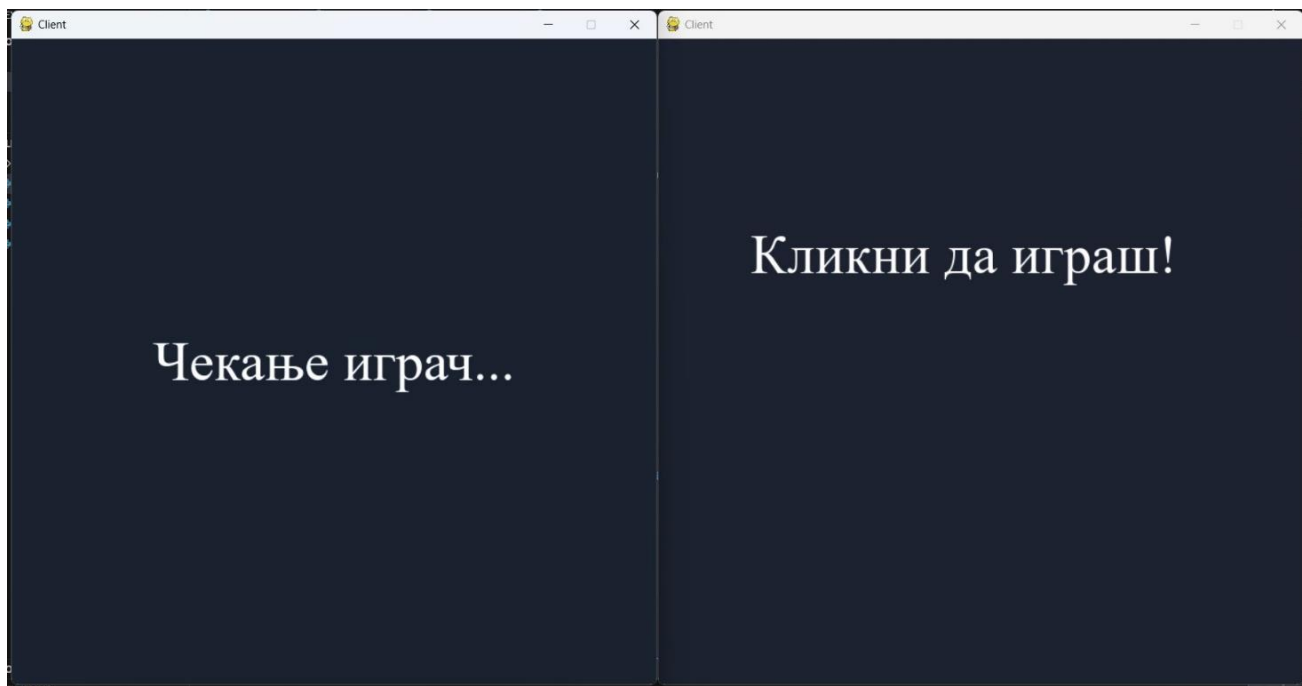
Во кодот на овој фајл дефинирана е класа Button за креирање на копчиња во графички кориснички интрфејс користејќи ја библиотеката Pygame. Прво го креираме прозорецот со претходно дефинирана висина и ширина. Потоа ги иницијализираме атрибутите на објектот за копче како што се текстот, позицијата (x и y) и бојата. Функцијата “**draw**” го црта копчето на прозорецот, а функцијата “**click**” прави проверка на клик на копчиња.

Функцијата “**menu_screen()**” преставува почетен екран на играта. Изгледот на почетниот екран е прикажан на Слика 4.1

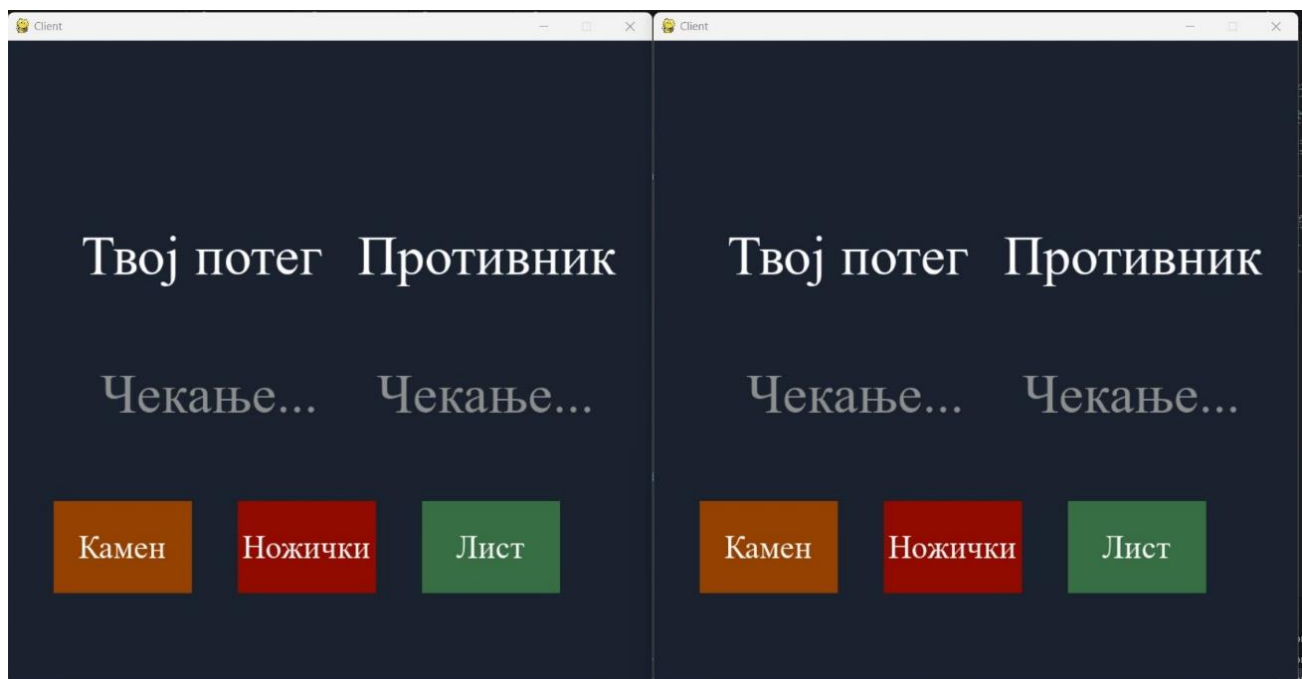


Слика 4.1 Почетен екран

Тука играчот може да избере да ја започне играта или да ја напушти со затворање на прозорецот. Кога едниот клиент ќе кликне да игра на негово прозорче се испишува порака “Чекање играч” како што е прикажано на Слика 4.2. Кога и другиот клиент ќе кликне да ја започне играта ќе се појави прозорчето и кај двата клиент каде што ќе можат да ја започнат играта, а тоа е прикажано на Слика 4.3.

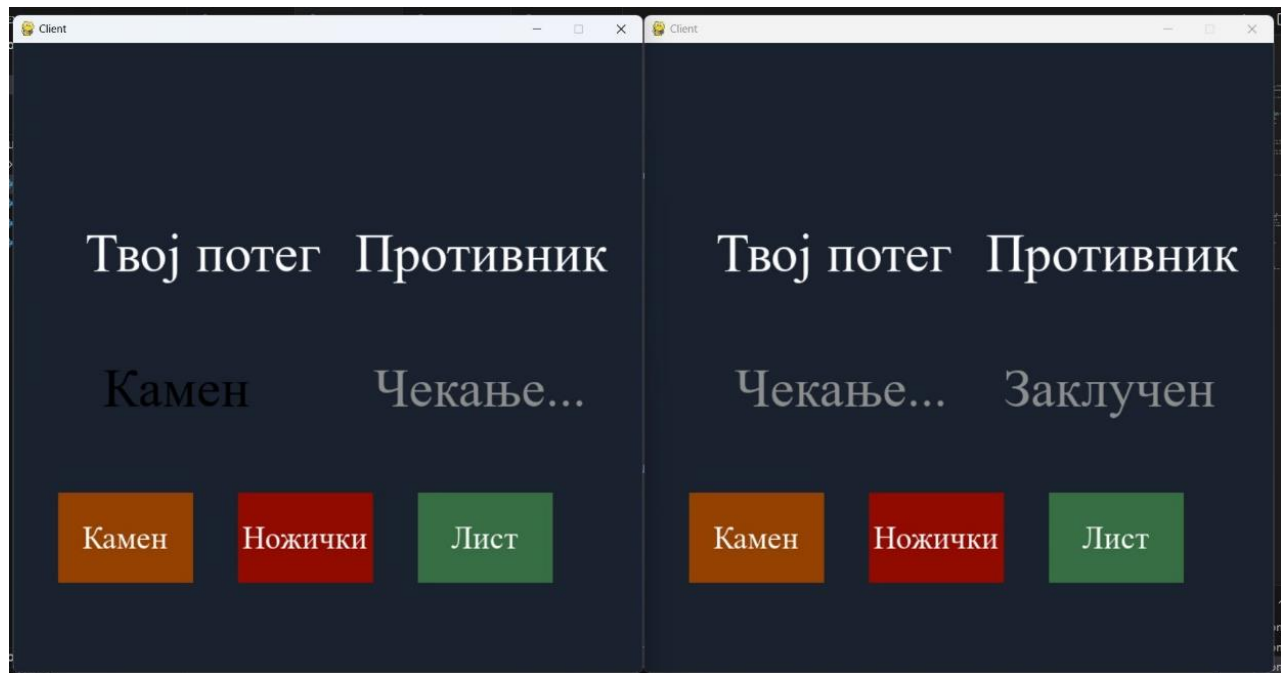


Слика 4.2 Чекање играч



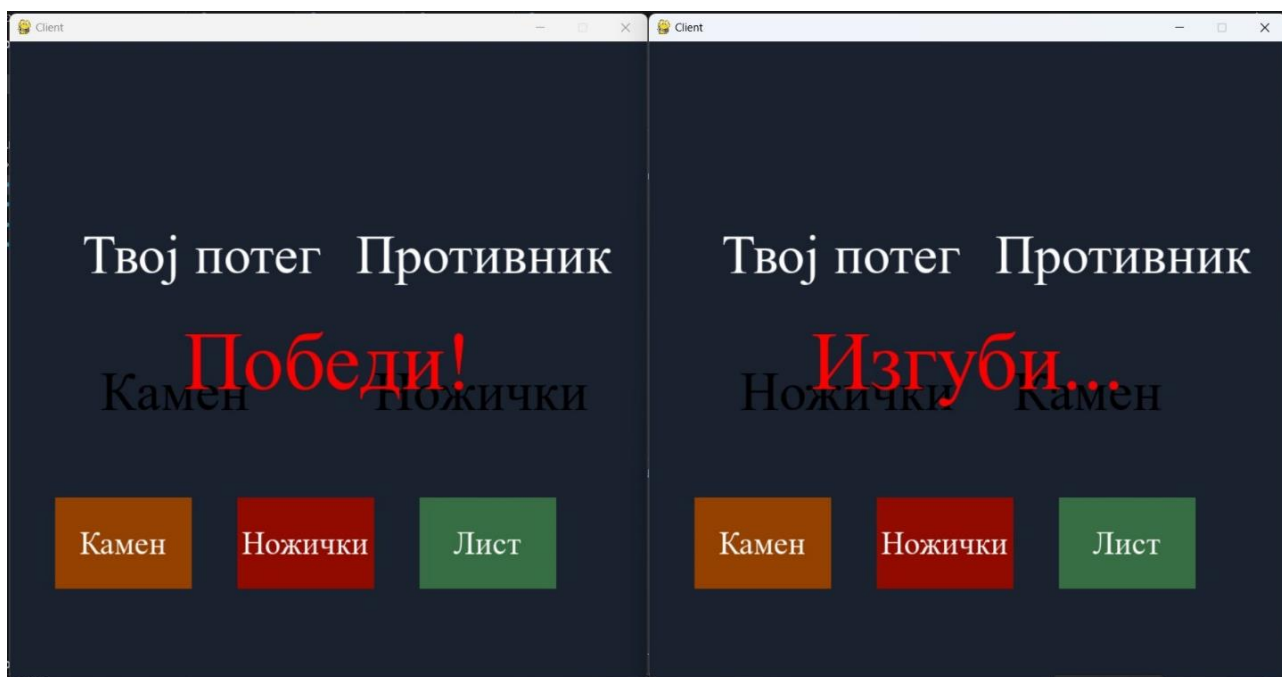
Слика 4.3 Започнување на игра

Ако клиентите кои што играат се Клиент1 и Клиент2. Доколку Клиент1 избере една од трите опции (Камен, Ножички или Лист) во делот Твој потег на неговит прозорец ќе се прикаже избраната опција додека пак кај прозорецот на Клиент2 статусот на делот Противник се менува од “Чекање..” во “Заклучен” со што Клиент2 знае дека сега тој треба да селектира една од понудените опции како што е прикажано на Слика 4.4. Оваа логика е прикажана во функцијата “**redarWindow**”.

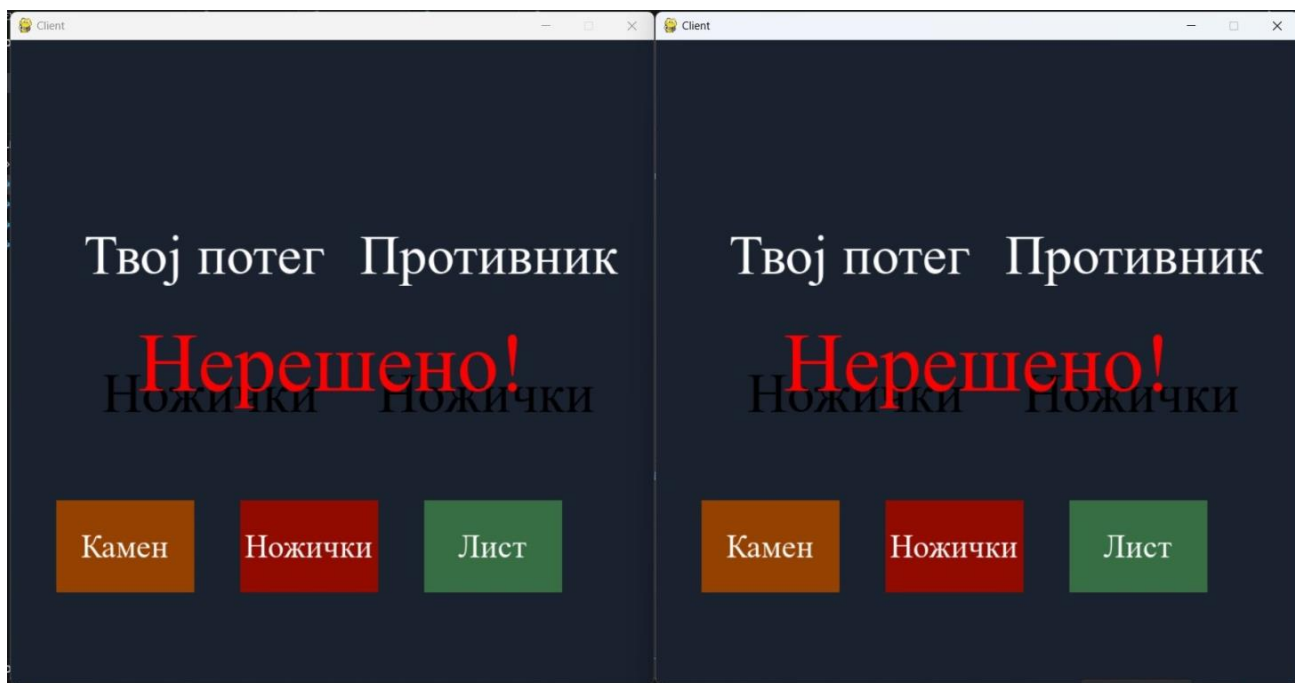


Слика 4.4 Селектирање од понудените опции

Откога и двата играчи ќе ја направат селекцијата се разгледува селектираната комбинација и се прикажува резултатот на соодветното прозорче со испишување на пораката “Победи!”, “Изгуби..” или “Нерешено!” како што е прикажано на Слика 4.5. и Слика 4.6. Ова логика е имплементирана во функцијата “**main**”.



Слика 4.5 Приказ на порака “Победи!” и “Изгуби..”



Слика 4.6 Приказ на порака “Нерешено!”