

Distribuirani računarski sistemi – projekat Turn snake game

AUTORI: STEFAN BEŠOVIĆ, STEFAN ĐUROVIĆ, JELENA BEADER, PREDRAG GLAVAŠ

Sadržaj

Uvod i pravila igre.....	1
Pokretanje igre	1
Arhitektura aplikacije	3
Arhitektura serverske aplikacije	3
Arhitektura klijentske aplikacije	3
Arhitektura komunikacije	6
Moguća poboljšanja	6

Uvod i pravila igre

U okviru projekta realizovana je turn based multiplayer igra sa zmijama. Igru mogu igrati od 2 do 4 igrača na više računara u okviru LAN mreže. Svaki igrač na početku igre može imati od jedne do četiri zmije koje kontroliše kada na njega dođe na red putem strelica na tastaturi, a menja aktivnu zmiju pritiskom tastera TAB.

Ukoliko igrač zmijom udari u svoju zmiju gubi igru, ukoliko udari u zid ili protivničku zmiju gubi svoju zmiju. Ukoliko svojom zmijom okruži glavu protivničke zmije, protivnička zmija će biti uništena.

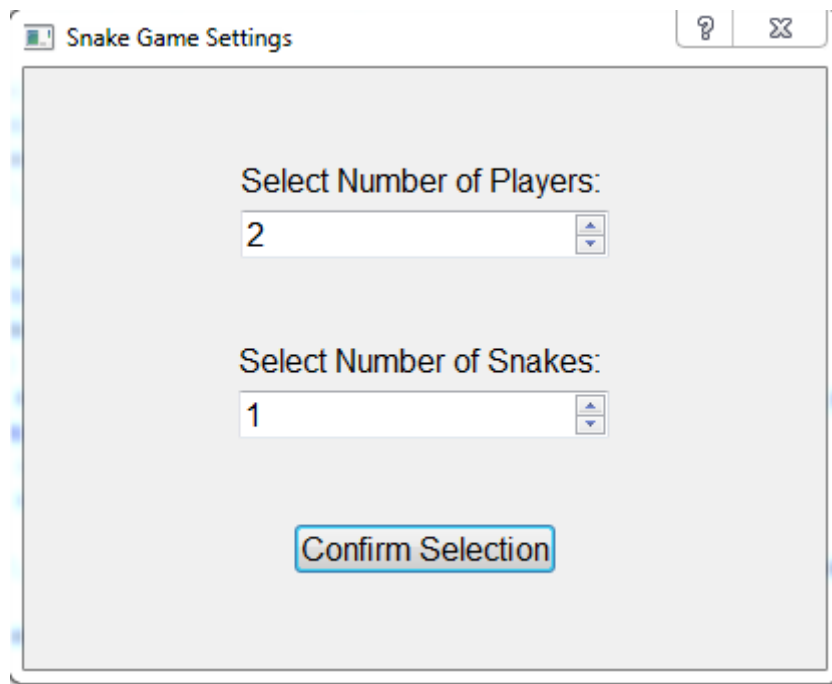
Jedenjem hrane koja je narandžaste boje igračeva zmija dobija još jedan korak po potezu i produžava se za 1 deo. Jedenjem hrane plave boje igrač može biti ili nagrađen dobijanjem dodatne zmije ili kažnjen gubitkom poteza na aktivnoj zmiji trajno. Plava hrana se pojavljuje svakih 30 sekundi i ostaje na tabli dok god ne bude pojedena, nakon čega se pojavljuje opet za 30 sekundi.

Hrana se kreće po tabli nasumično 1-3 polja pravolinijski nakon što svi igrači završe svoje poteze u jednoj rundi.

Pobednik je igrač koji poslednji ostane sa živim zmijama na tabli, bez obzira na broj poena koje imaju drugi igrači.

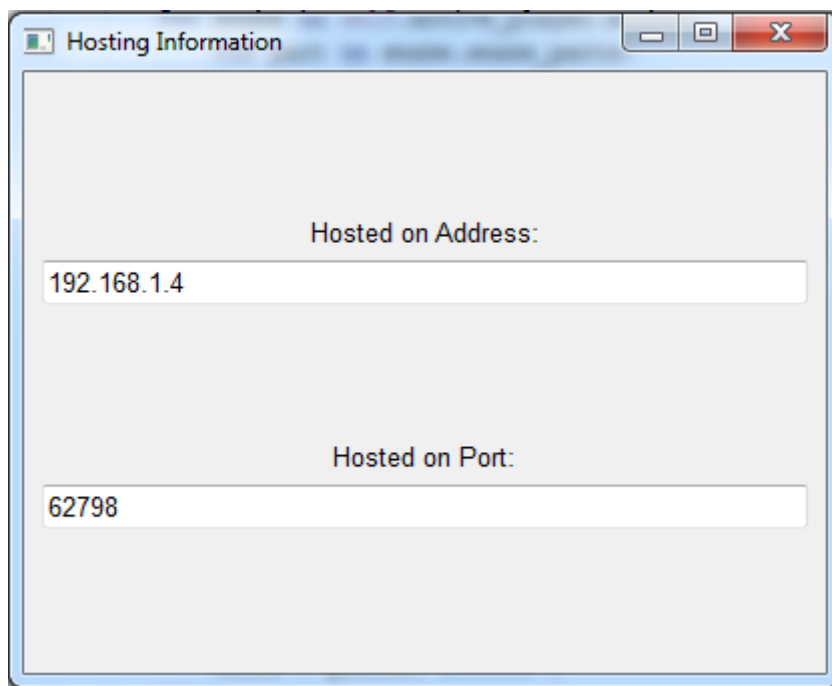
Pokretanje igre

Igra se pokreće prvo pokretanjem serverske aplikacije i izborom parametara igre.



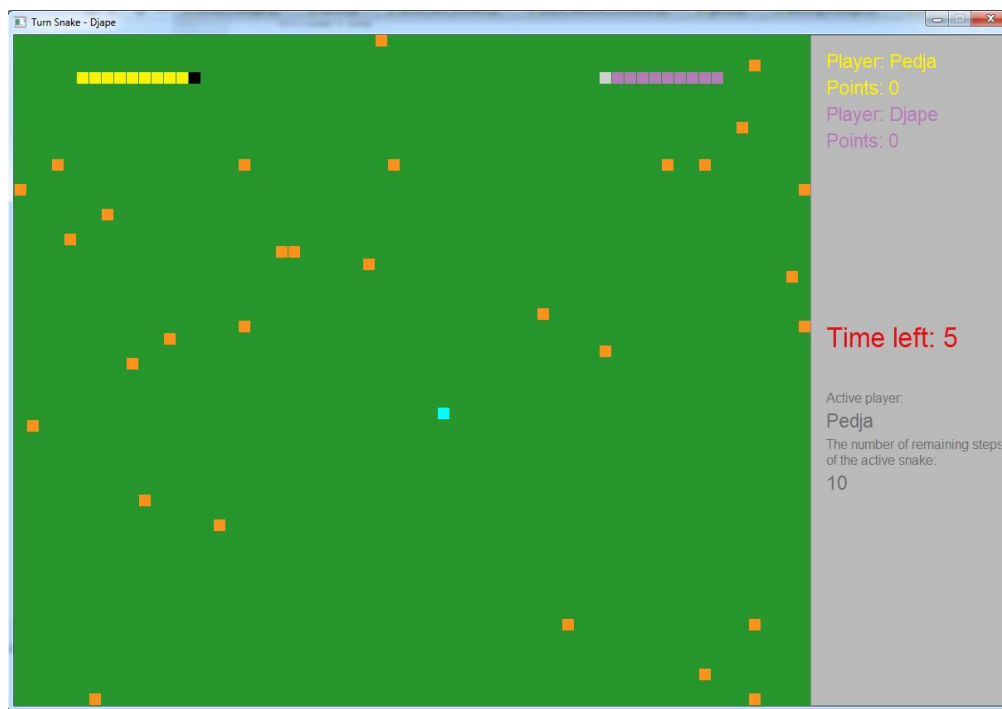
slika 1. Izgled prozora sa parametrima igre na serveru

Nakon toga izlazi prozor sa informacijama na kojoj IP adresi i portu je igra hostovana, ti podaci moraju biti saopšteni svim igračima kako bi mogli da se priključe.



slika 2. Izgled prozora sa adresom servera igre

Sada se igri mogu priključiti klijenti koji žele da igraju pokretanjem klijentske aplikacije na svojim računarima. Računari moraju biti u istoj LAN mreži kao i računar sa serverskom aplikacijom. Klijenti po pokretanju aplikacije unose IP i port servera a zatim i nadimak pod kojim će igrati. Nadimak mora biti jedinstven, a ako nije server će ih na to upozoriti i tražiti da ga ponovo unesu. Kada se svi igrači konektuju, igra će započeti.



slika 3. Izgled prozora sa aktivnom igrom klijentske aplikacije

Arhitektura aplikacije

Arhitektura serverske aplikacije

Kao ključni delovi serverske aplikacije mogu se izdvojiti:

- Game loop
- Mrežni deo
- Menadžeri

Mrežni deo serverske aplikacije vrši prijem i slanje podataka od klijenata i klijentima. Server pravi po jednu nit za svakog klijenta preko koje šalje stanje igre. Stanje igre se dobija proračunima u Game loop-u i stavlja na odgovarajući queue svake niti koja šalje klijentu. Za prijem podataka server ima jednu nit gde multipleksiranjem utvrđuje sa kog soketa je stigla poruka. jedna nit je dovoljna jer u jednom momentu samo jedan igrač može biti na potezu.

Game loop je srž logike igre i petlja gde se na osnovu odgovarajućih parametara računa stanje igre i šalje klijentima. Objedinjuje i koristi sve ostale delove serverske aplikacije.

Menadžeri predstavljaju grupu klasa koja sadrži logiku za upravljanjem i kontrolom igre. Pomenuta logika se sastoji od metoda koje su zadužene za iscrtavanje elemenata igre, detekciju kolizije (preklapanja) pa do metoda za generisanje hrane i proveru završetka igre.

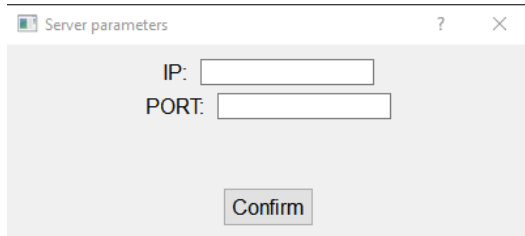
Arhitektura klijentske aplikacije

Delovi klijentske aplikacije su:

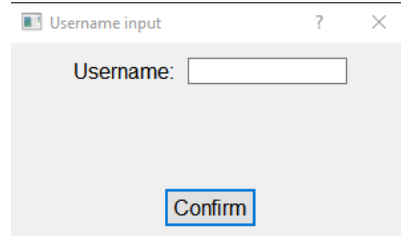
- Mrežni
- Grafički deo

Mrežni deo klijentske aplikacije vrši komunikaciju sa serverom. Postoji zasebna nit na klijentu koja prima podatke od servera i poziva metode na grafičkom delu aplikacije koje iscrtavaju stanje igre.

Grafički deo klijentske aplikacije predstavlja grafički interfejs kreiran korišćenjem Qt GUI frejmworka (PyQt5 biblioteka). Sastoji se od nekolicine prozora (slike 4., 5. 6.), svaki sa svojom posebnom namenom.



slika 4. prozor za kontekovanje na server



slika 5. prozor za unos korisnikovog imena



slika 6. prozor za igru

Prozorom za unos informacija za kontekovanjem (slika 4.), korisnik unosi adresu i port servera na kojem je hostovana igra, sam prozor implementira verifikaciju ulaza, tako da korisnik mora uneti validnu IP adresu i port, daljom proverom se utvrđuje da li je moguće povezati se na server, u suprotnom klijent biva obavešten upozorenjem da se ne može kontektovati na server.

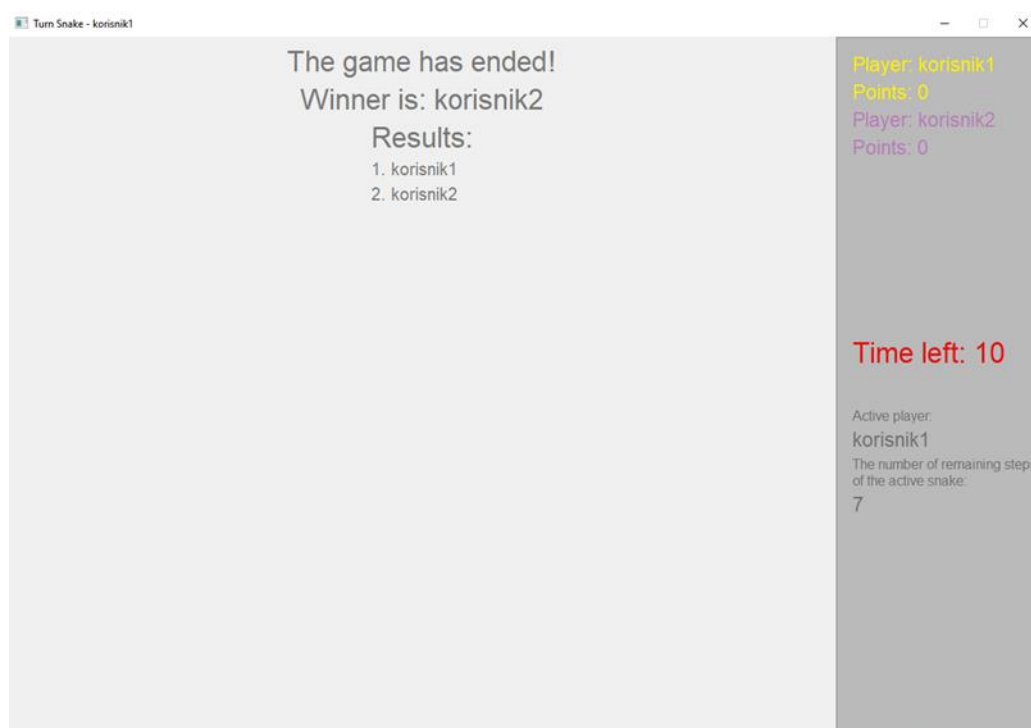
Slična stvar se odvija i sa izborom korisnikovog imena, po uspešnom povezivanju na server, klijentu se nudi da unese ime, i vrši se provera da dato ime već nije izabrano od starane suparničkog igrača. Po uspešnom konektovanju igrača i izborom imena, korisniku se prikazuje prozor za igru (slika 6.)

Za razliku od drugih klijentskih prozora, prozor za igru je složenija struktura koja objeđivanja više elemenata u okviru jednog prozora. U okviru prozora, pored polja (slika 7. element 1.) za igru imamo tabelu igrača (slika 7. element 2.), frejm za tajmer (slika 7. element 3.) , i frejm o trenutno aktivnom igraču i broju zmiya (slika 7. element 4.).



slika 7. gradivni elementi prozora za igru

U okviru prozora za igru se nalazi i prozor kraja igre (slika 8.), korisnicima se prikazuje pobednik, ali i tabela gde su svi igrači rangirani prema broju osvojenih bodova. Restart igre se obavlja automatski nakon 30 sekundi, posle uspešno završene igre.



slika 8. prikaz kraja igre

Arhitektura komunikacije

Klijenti i server komuniciraju uz pomoć TCP protokola iznad kog je implementiran protokol poruka sledećeg formata:

Dužina poruke (fiksno 4 bajta)	Tip poruke (fiksno 4 bajta)	Poruka (varijabilna dužina)
--------------------------------	-----------------------------	-----------------------------

Tip poruka je enumeracija koja govori prijemnoj strani šta je dobila od pošiljaoca kako bi mogla adekvatno odreagovati. Sama poruka je niz bajtova serijalizovan preko pickle biblioteke te je moguće poslati poruku proizvoljnog sadržaja (niz, heš mapa, objekat...)

Mehanizam za slanje je otporan na zagušenje bafera soketa te će pokušavati retransmisiju poruke dok god je ne pošalje.

Moguća poboljšanja

Svaki softver, pored svojih prednosti ima i mane, koje se mogu otkloniti potencijalnim poboljšanjima u budućnosti, sa tim slede predlozi kako unaprediti ovaj.

Bogatiji grafički interfejs klijentske aplikacije, gde bi se svi klijentski prozori objedinili pod jednim prozor, i tako centralizovali sve moguće akcije klijenta na jednom prozoru, nudeći različite prikaze u okviru istog prozora u zavisnosti od akcije klijenta bilo to povezivanje na server, unos imena, ili igra.

Ponuditi korisnicima veću slobodu pri izboru zmija, klijentu dozvoliti da bira boje svojih zmija (jedna boja za sve zmije).

Restilizovati izgled samih elemenata igre (zmija, hrane, pozadine), koristeći slike umesto geometrijskih elemenata za prikaz istih.

Dodavanje audio sadržaja u samu igru, od pozadinske muzike, do posebnih zvukova koje bi pratili pojedine radnje na GUI-ju, kao npr. otkucavanje tajmera koji je pri kraju, zvuk kada korisnik izgubi/pobedi.

Dodati mehanizam ponovnog povezivanja na server, ukoliko se veza prekine između klijenta i servera, klijent više nije u mogućnosti ponovo da uspostavi vezu i pridruži se već aktivnoj igri.

Dograditi sam kraj igre, ubaciti sistem glasanja za restartom igre (korisnici se izjašnjavaju da li bi želeli da ponove igru) umesto automatskog restarta igre.