



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
NOVI SAD



Distribuirani računarski sistemi u elektroenergetici - Primenjeno softversko inženjerstvo -

Lucky4

PR34/2017 Vesna Prica

PR35/2017 Aleksa Matić

PR4/2017 Nikola Nikolić

PR15/2017 Tamara Jović

Novi Sad, 2021.

Sadržaj

1. Zadatak.....	3
2. Opšti rad aplikacije	4
3. Prednosti i mane	9
4. Zaključak	10

1. ZADATAK

Turn Snake

Napisati turn based igricu koja kombinuje klasičnu snake igricu i turn based strategiju.

Uloge:

- Dva do četiri igrača koji planiraju strategiju kretanja zmija
- Zmije
- Hrana

Pravila:

- Svaki igrač posjeduje jednu do više zmija kojima mogu da planiraju putanju
- Cilj igrača je da zarobi suparničke zmije
- Zmija umire ako glavom udari u zid ili u tijelo druge zmije
- Zmija posjeduje dužinu i broj koraka po potezu
- Zmija može proširiti dužinu i broj koraka skupljanjem hrane
- Hrana se kreće pravolinijski od 1 do 3 koraka po potezu

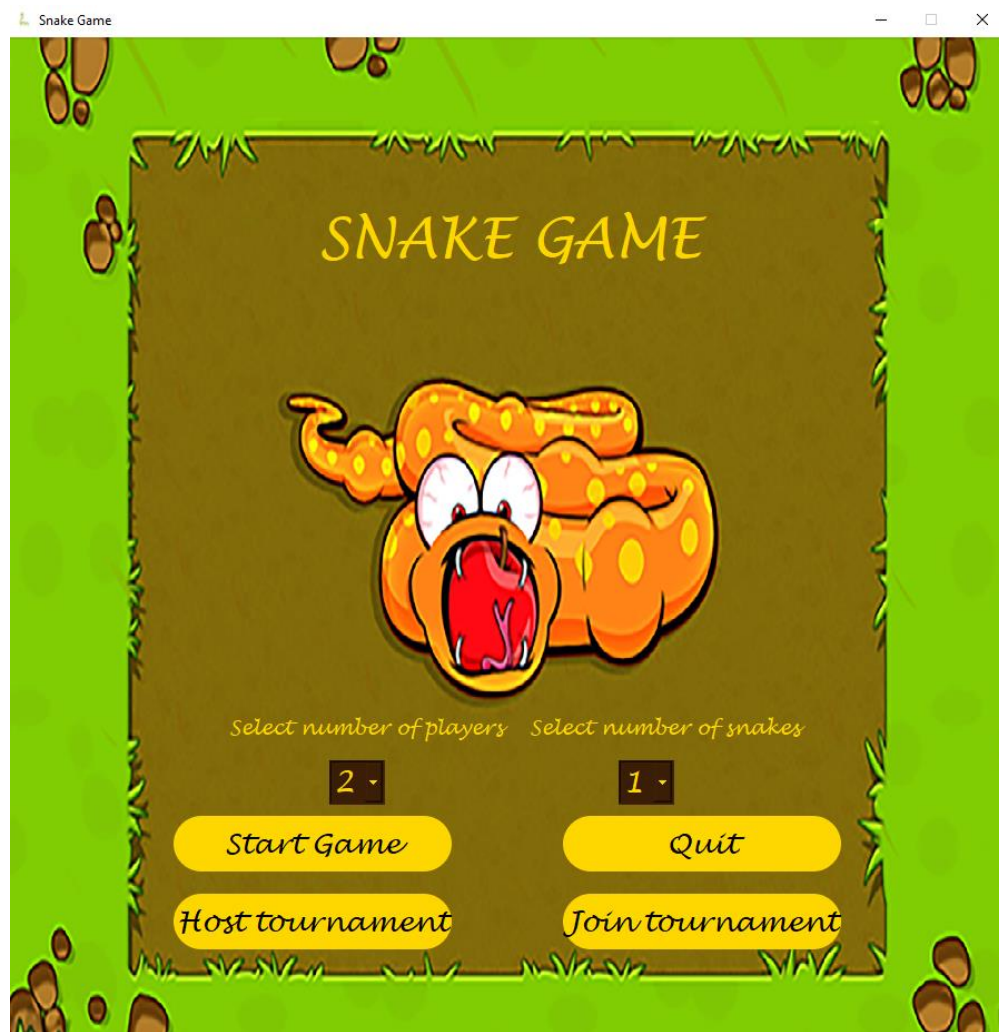
Za realizaciju koristiti Python3, PyQt5, multiprocessing. Raditi u timovima od 4 člana.

Napisati dokumentaciju u kojoj opisuje opšti rad aplikacije i rezimirati prednosti i mane korišćenja Python jezika, PyQt5 okvira i paralelizacije rada.

* tekst zadatka je podložen promjenama

2. OPŠTI RAD APLIKACIJE

Prilikom pokretanja aplikacije prikazuje se početni prozor (Slika 1) za odabir tipa igre.



Slika 1 Početni prozor

Prva opcija *Start Game* predstavlja klasičan tip igre sa više igrača koji komuniciraju preko mreže. Moguće je izabrati igru sa 2-4 igrača koji mogu imati 1-4 zmijske. Potrebno je da svaki igrač odabere ista podešavanja.

Na slici 2 je prikazana igra 4 igrača sa po 4 zmije.

Svaki igrač pomera jednu zmiju po potezu i ta zmija može da napravi koraka srazmerno sa svojom dužinom. Ukoliko igrač ne odigra potez u roku od 20sekundi, zmija se pomera za jedan korak u smeru trenutnog kretanja. Nakon što svaki tim odigra po jedan potez, hrana se pomera pravolinijski za 1-3 koraka nasumično.



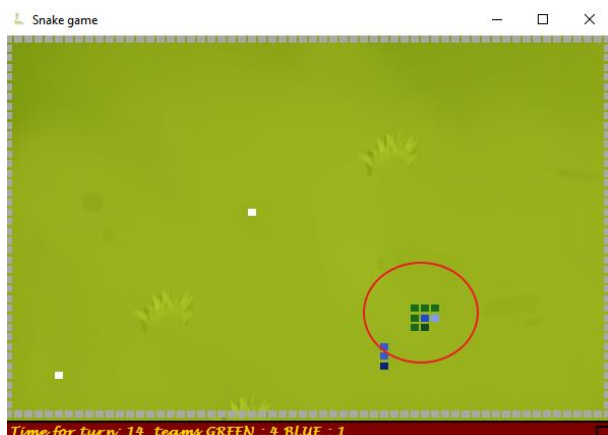
Slika 2 Prozor igre - 4 igrača po 4 zmije

Na slici 3. prikazan je element iznenađenja, odnosno tražena neočekivana sila u okviru proširenja zadatka. Element iznenađenja se pojavljuje na svakih 15 poteza i nestaje nakon dva poteza. Njegovo pojavljivanje na ekranu je nasumično. Postoje dve opcije ukoliko sila zahvati zmiju, zmija umre ili naraste. Dejstvo sile je takođe nasumično.

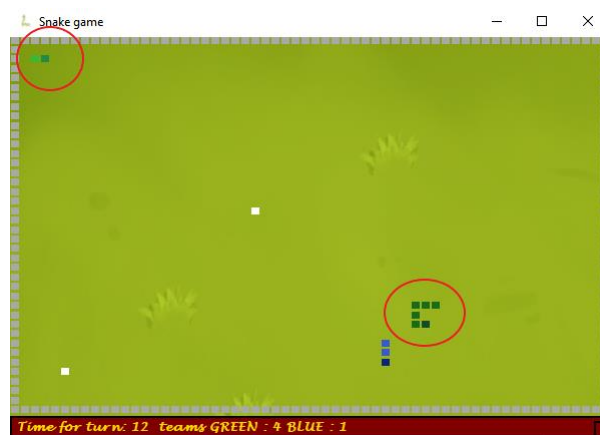


Slika 3 Surprise

Opkoljavanje je prikazano na slikama 4 i 5, gde se može videti pozicija kada je zmija plavog tima zarobljena (bez mogućnosti kretanja) i da nakon toga prelazi u zeleni tim.



Slika 4. Opkoljavanje



Slika 5. Prelazak zarobljene zmije u drugi tim

Igra se završava kada ostane jedan tim i svi korisnici dobijaju obaveštenje o kraju igre i o tome ko je pobednik, kao što je prikazano na slici 6.



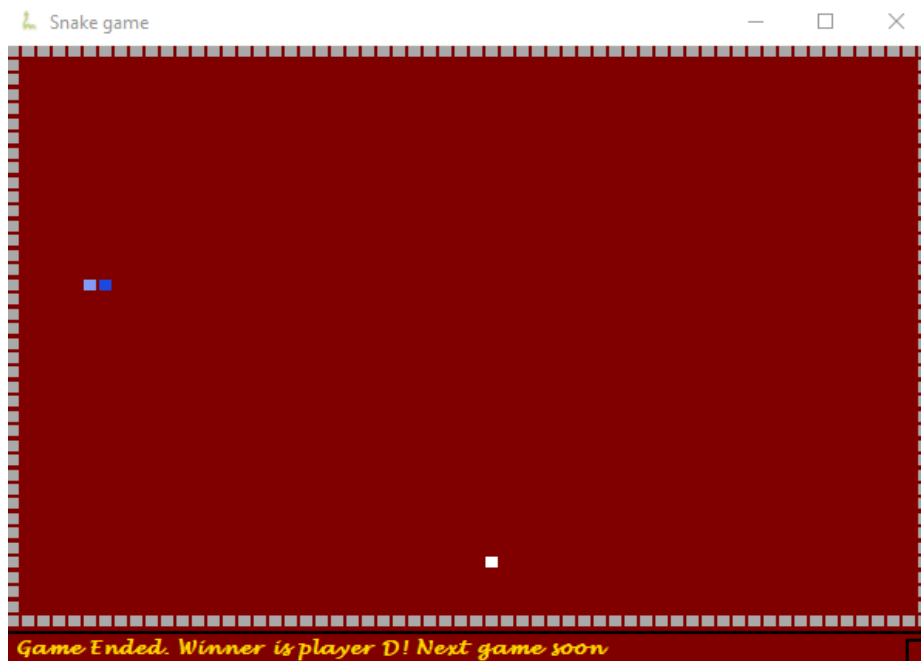
Slika 6. Kraj igre

Druga opcija predstavlja turnir za 4 igrača. Jedan korisnik pokreće aplikaciju odabirom *Host tournament* nakon čega se pruža opcija unosa imena igrača koji će učestvovati u turniru(slika 7), a ostali korisnici biraju opciju *Join tournament*.



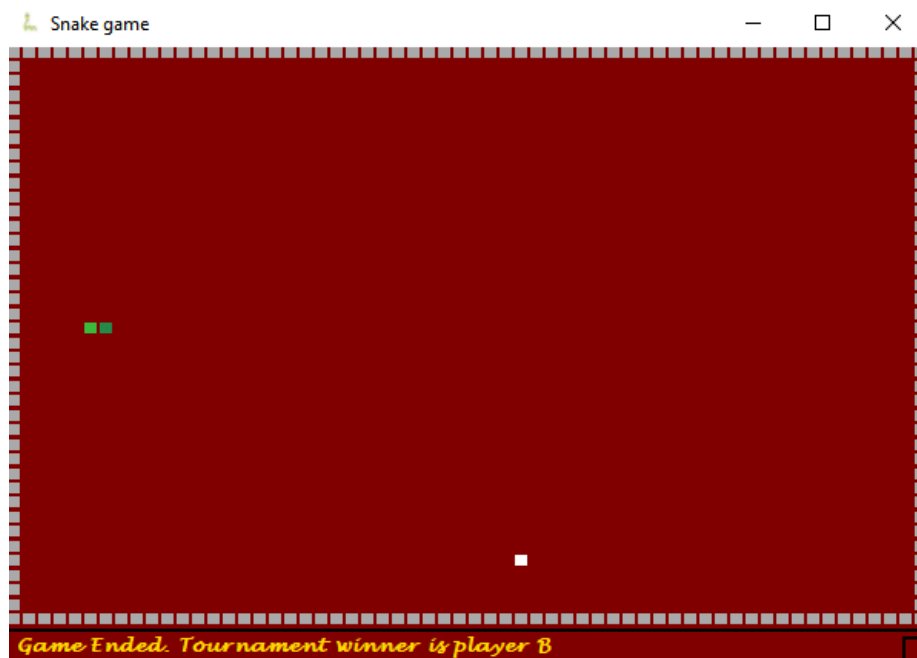
Slika 7. Unos imena igrača turnira

Turnir započinju igrači A i B, oba sa jednom zmijom. Nakon završetka te runde, ispisuje se pobjednik(slika 8) uz obaveštenja da sledeća igra počinje uskoro.



Slika 8. Završetak runde

Kada se završi druga runda, porednici igraju u finalu. Porednik finalne runde je porednik turnira i njegovo ime se ispisuje u statusbaru.



Slika 9. Završetak turnira

3. PREDNOSTI I MANE

Za razvoj ove igrice korišten je *Python* programski jezik, *PyQt5 framework* i *multiprocess* biblioteka za paralelizaciju rada.

1. *Python* programski jezik je interpretivni „visoki“ programski jezik koji podržava više tipova paradigmi, kao što su: objektno-orijentisana, imperativna, funkcionalna, proceduralna. Korištenje *Pythona* nam je mnogo olakšava sam razvoj igrice jer automatski vodi računa o stvarima kao što su *memory managment*, o tipovima podataka, lako skaliranje aplikacija, podržava *cross-platform* programiranje, itd.

Loše strane *Python-a* je pre svega što je *Python* spor, ima ograničenja kada je u pitanju pristup bazama podataka, gotovo je nemoguće napraviti kvalitetnu 3D grafiku, i nije najbolje za rešavanje zadataka koji zahtevaju optimizaciju korištenja memorije.

2. *Framework* koji je korišten za izradu ove igrice je *PyQt5*. *PyQt5* je *framework* pisan u programskom jeziku C++. Omogućava razvoj *cross-platform GUI-a* (*Graphical User Interface*). Sadrži oko 440 klasa i preko 6000 funkcija i metoda. Jedan od najvećih problema ovog *frameworka* je što je npr. na *Windows* platformi *Qt* aplikacija samo jedna prozor, bez kontrola tako da je otežano debugovanje.
3. *Multiprocessing* biblioteka, koja nam pruža kako lokalnu tako i distribuiranu konkurenciju. Zamenujući *Global Interpreter Lock* (mehanizam u *Python-u* koji garantuje da samo jedan *thread* može istovremeno da izvršava *bytecode*) koristeći podprocese umesto *thread-ova*. Zahvaljujući ovom modulu (modul je fajl u *Python-u* sa ekstenzijom *.py* koji sadrži definicije i izraze) koji dozvoljava programeru da u potpunosti iskoristi procesore na datoj mašini i samim tim paralelizuje izvršavanje zadataka. Biblioteka se koristi i na *Windows* i na *Unix* platformama.

Problem koji se javlja prilikom korištenja *multiprocess* biblioteke umesto *thread-va* je što se koriste procesi koji svaki ima zasebni memorijski prostor što otežava deljenje objekata i resursa između procesa, dok korištenjem *thread-ova* koristimo deljeni memorijski prostor i ovaj problem ne postoji. Kod *thread-ova* moramo zato koristiti neke od mehanizama zaključavanja kako ne bi više niti pisalo na istu memorijsku lokaciju, što bi dovelo do nekonzistentnog stanja prilikom izvršavanja aplikacije.

4. ZAKLJUČAK

Ovim projektom dobili smo spoj, paralelnog programiranja, grafike, cross platforme i videli kako to sve funkcioniše u okviru jednog zadatka. Uvideli smo značaj i efikasnost Python jezika u odnosu na druge do sada upoznate programske jezike.