

UNIVERZITET U NOVOM SADU

**Fakultet tehničkih nauka u Novom Sadu**

Distribuirani računarski sistemi u elektroenergetici

|  |  |
| --- | --- |
| Asistent:  Mihailo Vasiljević | Studenti:  Jelić Igor PR-4/2016  Telečki Karolina PR -81/2016  Lukić Zlatko PR-18/2017  Živanović Nikola PR-21/2017  Kondić Doroteja PR-28/2017 |

Sadržaj

[Snakes 3](#_Toc62771996)

[Detaljan opis igre 3](#_Toc62771997)

[1.1. Speed 4](#_Toc62771998)

[1.2. Players 4](#_Toc62771999)

[1.3. About 4](#_Toc62772000)

[1.4. Start 4](#_Toc62772001)

[1.5. Exit 5](#_Toc62772002)

[Dodatna objašnjenja 5](#_Toc62772003)

[Izgled table 6](#_Toc62772004)

[Implementacija 6](#_Toc62772005)

# Snakes

**Snakes** je strateška igrica bazirana na jednoj od najpopularnijih igara ikada.

Korisnik ima mogućnos da izabere od 2 do 4 igrača. Na početku igre, igrač bira da li želi da igra s jednom ili više. Korisnik takođe ima mogućnost da izabere brzinu, odnosno vreme trajanja poteza (10s-15s). **Cilj** igre je da igrač zarobi protivničku zmiju. Kada zmijica pojede jabuku njena dužina se povećava za jedan. Svaka zmija ima mogućnost da napravi onoliko koraka koliko je dugačka. U igrici se pojavljuju i dve pečurke, zelena i crvena. Ukoliko se neka zmijica nadje 5 polja u svim pravcima od zelene pečurku njena dužina će se povećati za dva, dok ukoliko pojede crvenu, njena dužina će se smanjiti za jedan. Prilikom sudara zmijice sa zidom, drugom zmijom ili sa samom sobom, zmija umire.

Igra omogućava korisniku da igra i kao single player, u tom slučaju gore pomentua pravila ne važe. Ovde, igrač nema mogućnost da unapred smišlja poteze. Zmijica se kreće sama, u smeru koji joj zada korisnik pomoću tastera strelica. Cilj ovakve igre jeste da zmija pojede što više jabuka i da pritom ne udari u zid ili samu sebe.

https://lh4.googleusercontent.com/iygTiK9kaP-_O_8jVDBtfJe9CLaZqQzvacSlaHJIniqmT4NA4X-aUpnG7aXRBPytfZMu7mPj9qSiGZoUW_n8pCrg-ajGgcLXyW8rmOGY_1H3iFN3ZM8066tw94ykuYdzythiAPIhttps://lh3.googleusercontent.com/5lpr1BuK8ngfQxlhcD6mGURx_tRFrXEYVFN_kyNsIjwkaqqDlfqXF09foS8Q5OsLkK_FI5BD5P1TRLA2exodGDjr41VM_kgEDu9IZC1IbzLxSEXSKa5Q0cFz7eDZRFa-47QG70whttps://lh3.googleusercontent.com/sM7SM6RYHD2c_pXMBAud17qR4o3JaGfyNJkrrdyVtycB46VtHwEHWiK6EPuykJQbvBNdPJJuwquIwFrLUvl3RPUOyh91ofELyO9s9yzODOmGuNsEaRwaaa088MvUD4qnZX7w44Ihttps://lh4.googleusercontent.com/zfPPLDFAXGAtDnvVIy1x6utmVRMHaJt5ZOV6wpaFGMnKf-kiPu7R8lc-EnIu2iZJu-rjDDGOFxCdlc931eries_EsLKIaTBo4HX79tMhQpyMBD_VzNhWJmrTASr8DwF7-bLqG3Y

Detaljan opis igre

Prilikom pokretanja igrice, otvara se početni prozor *(slika 1)*.



*Početni prozor (slika 1)*

U nastavku sledi objašnje uokvirenih stavki sa slike.

1.1. Speed

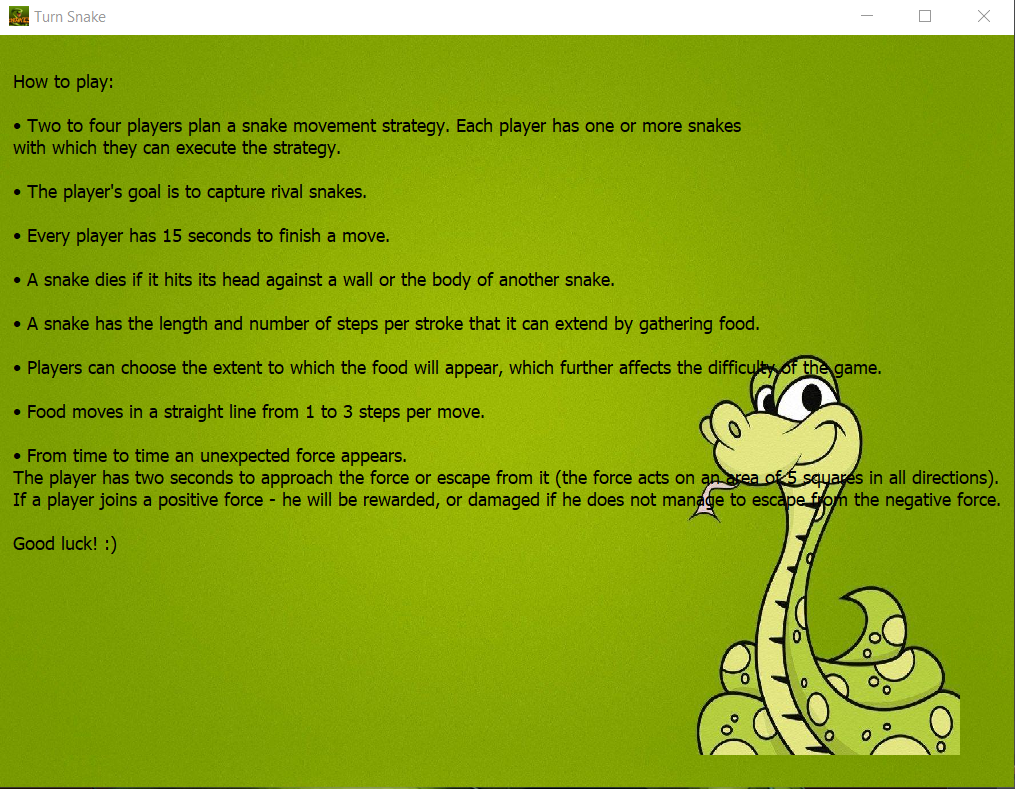
Predstavlja vreme trajanja, za koje igrač koji je na potezu treba da odigra određeni broj koraka.

1.2. Players

Korisnik ima mogućnost da izabere broj igrača (minimalan broj je 2, a maksimalan 4). Takođe, korisnik ima mogućnost da odabere “Classic snake” pri čemu igra sam (način igranja ovakve Zmije je objašnjen u drugom pasusu poglavlja “Snakes”).

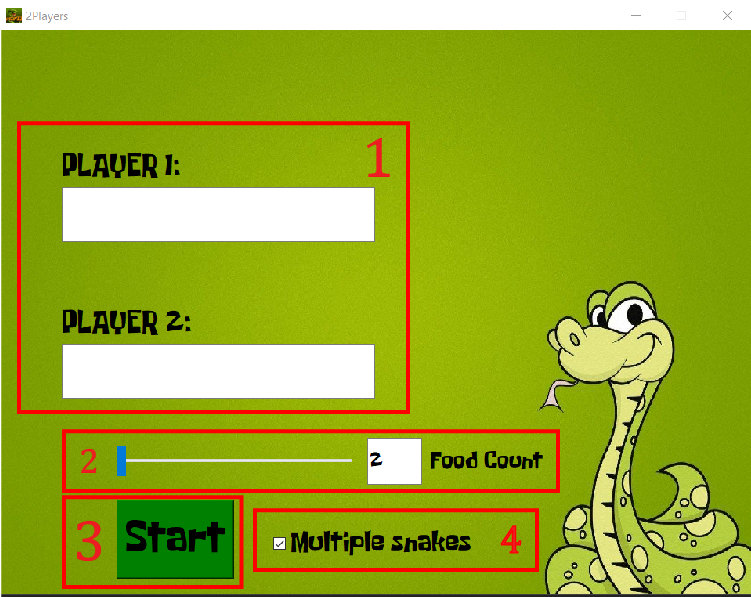
1.3. About

Sadrži informacije o načinu igranja igrice *(slika 2).*

*Prozor “About” (slika 2)*

1.4. Start

Otvara se novi prozor sa podešavanjima *(slika 3).*



*Podešavanja (slika 3)*

Objašnjenje polja sa **slike 3**:

1. **Player ½** - Unos imena igrača
2. **Food Count** - Bira se količina jabuka (hrane) koja će se prikazati prilikom startovanja igrice
3. **Start** - Početak igre
4. **Multiple snakes** - Korisnik bira da li želi da ima jednu ili više zmija (maksimalan broj zmija koje jedan igrač može da ima u jednom trenutku su dve. Ukoliko izgubi jednu, moze da se podeli ponovo ako je duži od 10 polja).

1.5. Exit

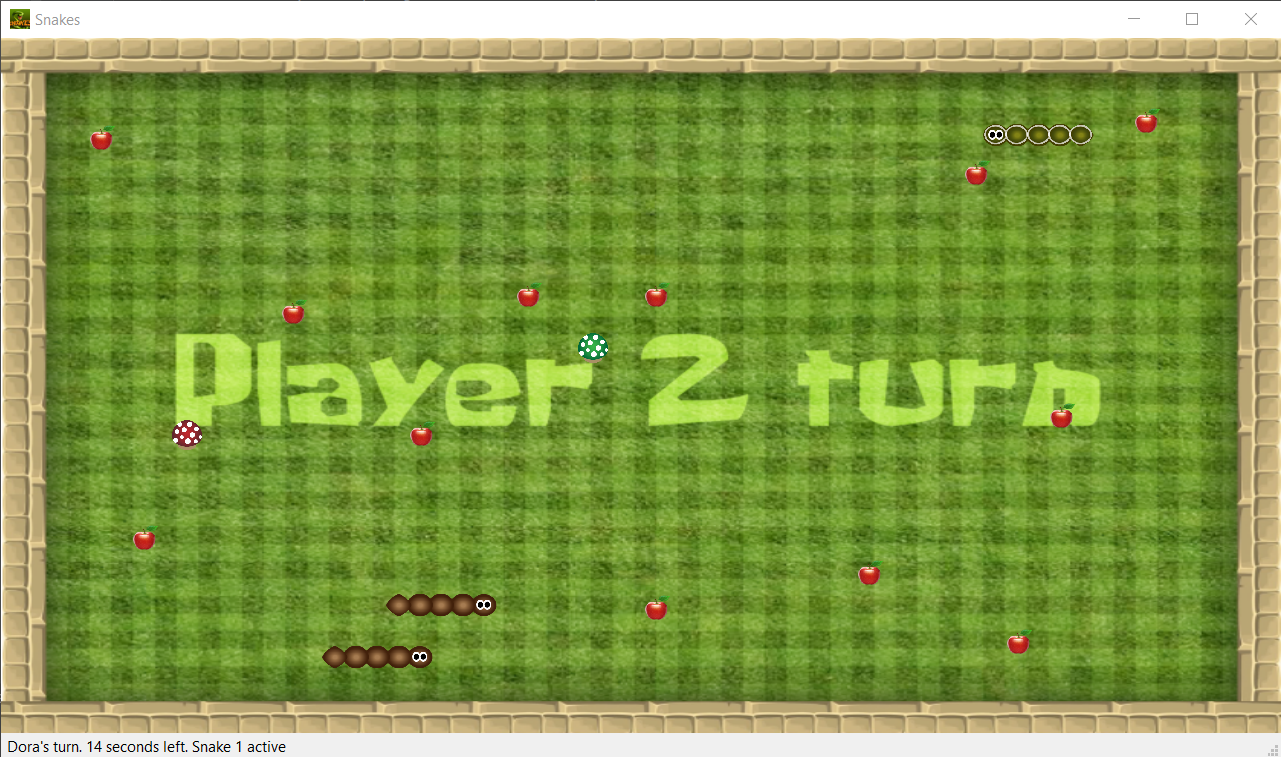
Omogućava izlazak iz igrice.

## Dodatna objašnjenja

1. Igrač počinje igricu sa jednom zmijicom.
2. Ukoliko je otkačena stavka “Multiple snakes” igrač može dodati još jednu zmijicu pritiskom na **taster S**.
3. Za menjanje zmijica jednog igrača, potrebno je pritisnuti **taster Tab**.
4. Igrač pomera zmijice pomoću **strelica**.

## Izgled table

Prilikom pokretanja igre otvara se sledeći prozor *(slika 4)*.



*Board (slika 4)*

U donjem desnom uglu “Table”, moguće je videti koji igrač je na potezu i koliko vremena ima da odigra trenutni potez.

Za bonus i malus su implementirane dve pečurke *(izgled prokazan na slici 5)*



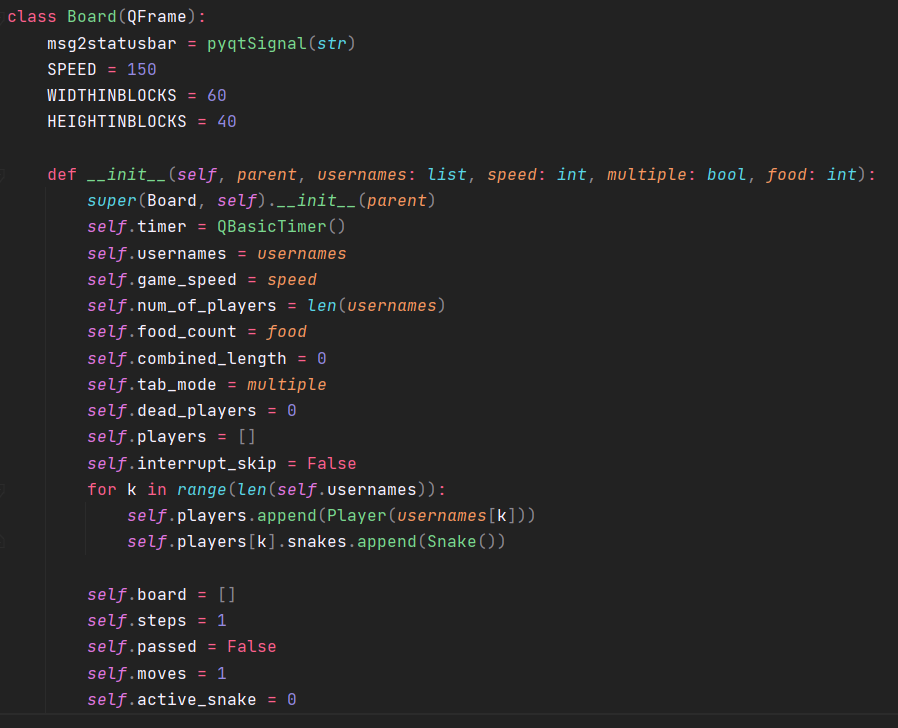
*Bonus i malus (slika 5)*

Ukoliko se zmijica nađe glavom unutar 5 polja u svim pravcima od zelene pečurke, povećaće se za dva, a ukoliko pojede crvenu pečurku smanjiće se za jedan. Vreme postojanja ovih pečurki je ograničeno na dve sekunde.

# 

# Implementacija

Početna logika igrice kreće iz fajla splash.py koji je zadužen za sami prikaz splash screen-a. U zavisnosti od toga koliko igrača je izabrano na početnom ekranu, pokreće se odgovarajuća klasa za unos korisničkih imena. Iz te klase pokreće se game.py skripta, tj. klasa SnakeGame. Ukoliko se odabere klasična zmijica, pokreće se klasa ClassicSnake, ne omogućujući unos korisničkog imena. Obe klase pozivaju zasebne board elemente, SnakeGame Board u fajlu board.py, dok ClassicSnake poziva fajl classic\_snake.py.



*Inicijalizacioni deo klase Board unutar board.py fajla (slika 6)*

Za potrebe praćenja trajanja poteza kao i odbrojavanje preostalog vremena za potez, koristi se timer iz biblioteke **threading**. Prilikom pokretanja igrice pozivaju se iz fajla helpers.py iz klase PerpetualTimer. Kao argumente primaju vremenski interval nakon kojeg bi se izvrsavale (repetitivno) i funkcija koju ce izvrsavati. Ukoliko igrač završi potez pre vremena, ima opciju da odmah preda potez protivniku pritiskom na **taster N**. Prilikom pritiska na **N**, oba tajmera se zaustavljaju te se odmah zatim ponovo pokreću kako bi poslužili narednom igraču.

U zavisnosti od izabranog broja igrača kreira se odgovarajući broj instanci klase Player, te se svakom igraču dodeljuje po jedna početna zmija **dužine 5**. Igrač broj jedan kreće iz donjeg levog, drugi iz gornjeg desnog, treći iz gornjeg levog, dok četvrti polazi iz donjeg desnog ćoška. Za iscrtavanje zmija pobrinuo se paintEvent koji pravi jasnu razliku između glave, tela i repa svake zmije. Za iscrtavanje se koristi QPainter. Pored zmija, ovaj event zadužen je za iscrtavanje hrane, bonusa i malusa, kao i “sjaja” trenutno aktivne zmije. Unutar klase board se nalazi i logika za kretanje zmije kao funkcija move\_snake (zmija može da se kreće onoliko polja koliko je duga). Za praćenje kretanja aktivne zmije zadužen je poseban thread. S obzirom da je igrica turn-based, nema potrebe da proveravanja kolizije budu unutar nekog posebnog tajmera ili threada gde bi uzalud trošile procesorsko vreme. Umesto toga, proveravanje kolizija se izvršava svaki put kada se zmija pomeri. Kada igrač konzumira hranu bar 5 puta, pritiskom na **taster S** može da stvori još jednu zmiju koju će zatim moći da kontroliše, pri čemu se već postojeća zmija smanjuje za dužinu novonastale zmije. Svakim deljenjem, brojač konzumirane hrane restartuje se na 0. Da bi se obezbedilo da igrač nema više od 2 aktivne zmije u jednom trenutku pobrinuće se funkcija check\_number\_of\_live\_snakes unutar player.py skripte. Prebacivanje na novonastalu zmiju vrši se pritiskom na **taster Tab** koji poziva funkciju change\_active\_snake iz snake.py fajla**.** Ma koliko da je trenutna zmija dugačka, novonastala zmija će biti početne dužine 5. Za to je zadužena funkcija split\_snake koja se nalazi unutar snake.py skripte. Pre nego što igrač uspe da stvori novu zmiju, mora se postarati da nova zmija neće naići ni na kakvu koliziju. Pri svakom pokušaju stvaranja nove zmije, poziva se funkcija check\_split\_collision iz collisions.py fajla. Ta funkcija vraća boolean koji, nakon izvršenih provera, vraća potvrdan ili odričan odgovor na igračev zahtev za stvaranjem nove zmije. Unutar tog fajla se takođe nalaze i ostale kolizije, kao što su kolizija zmije sa hranom (zmija raste, dropuje se nova hrana), sa zidom, sa samom sobom (samoubistvo) i sa drugom zmijom. Ukoliko dodje do sudara dveju zmija, umire samo ona koja je inicirala pokret koji je doveo do kolizije. Ukoliko dođe do smrti aktivne zmije, funkcije koje vode brigu o kolizijama prvo proveravaju da li igrač čija je zmija „poginula“ ima još neku aktivnu zmiju. Ukoliko je odgovor da, ta zmija postaje aktivna. U suprotnom, igrač biva eliminisan i potez se predaje sledećem igraču. Tokom igre se u random generisanim trenucima u interavalima od 5 do 20 sekundi pojavljuju i neočekivane sile predstvaljene klasama bonus.py i malus.py. To je omogućeno pozivima funkcija deus\_ex\_machine\_bonus i deus\_ex\_machine\_malus u okviru board.py fajla koje, takođe, nakon 2 sekunde primenjuju njihovo dejstvo.

Igrač koji poslednji ostane u polju sa bar jednom zmijom postaje pobednik.