

Dokumentacija projekta Nutella Galaga

Distribuirani računarski sistemi u elektroenergetici

Tim F5

**Dejana Grujić, Aleksandra Kovač, Dina Petrušić, Marina
Dimitrijević, Jelena Mavrak** PR 148/2015, PR 85/2015, PR
82/2015, PR 34/2015, PR 9/2015

Sadržaj

1. Uvod	2
2. Opis rada aplikacije.....	3
3. Python	7
4. PyQt5	8
5. Zaključak	8

Uvod

Projekat Nutella Galaga predstavlja igricu napravljenu po uzoru na arkadnu igru Galaga.

Ideja igre Galaga je da se korisnik nalazi u ulozi aviona, koga pomera horizontalno pri kraju ekrana, kontrolišući ga na strelice levo i desno. U centralnom delu ekrana se nalaze alien-si, u ulozi neprijatelja. Oni ispaljuju projekte na nasumičan period, sa nepredvidive pozicije. Ukoliko pogode igrača, on gubi 1 od 3 života. Cilj igrača je da pogodi neprijatelje projektilima koje izbacuje klikom na strelicu na dole ili taster space i tako osvaja bodove. Kada se reši svih alien-a, igrač prelazi na sledeći nivo, ponovo ima 3 života i broj neprijatelja kao na početku igrice. Razlika u nivoima se ogleda u povećanju brzine kretanja neprijatelja. Igrač je zaštićen sa nekoliko odbrambenih bunkera, koje neprijatelji moraju da unište da bi došli do igrača. Igrica se završava u slučaju da igrač izgubi sve živote, ako su se neprijatelji spustili do kraja ekrana i tako pobedili igrača ili kada igrač ostane bez poslednjeg bunkera.

Za razliku od Galaga, u igrici Nutella Galaga je omogućen rad sa 2 igrača. Pri startovanju aplikacije, oba igrača se nalaze u uglovima okvira za dozvoljeno kretanje. Korisnici pokreću svoje avatare horizontalno na strelice levo i desno, odnosno na tastere A i D. Umesto alien-a, neprijatelji igrača su nutelle, postavljene u formaciji 3x10, koje se kreću po zadatoj šemi od jedne do druge ivice dozvoljene oblasti ekrana aplikacije. Oni napadaju igrače ispaljivanjem projektila nasumično i nezavisno jedni od drugih. Kao u originalu, da bi igrač prešao na sledeći nivo, neophodno je da pogodi svakog neprijatelja projektilima koje izbacuje na strelicu na gore tj. taster W. Novim nivom se povećava brzina kojom projektil leti ka igraču i šansa da će neprijatelj ispaliti projektil, dok se životi igrača ponovo inicijalizuju na 3. Kada projektili nutella pogode igrača, broj njegovih života se smanjuje. Takođe, dodata je još jedna uloga u projektu, velika nutella koja se pojavljuje iznad ostalih nutella i vraća sve živote igraču koji je pogodi. Igrica se završava u slučaju da oba igrača izgube sve živote i tada se za porednika proglašava igrač koji je najduže ostao u igrici.

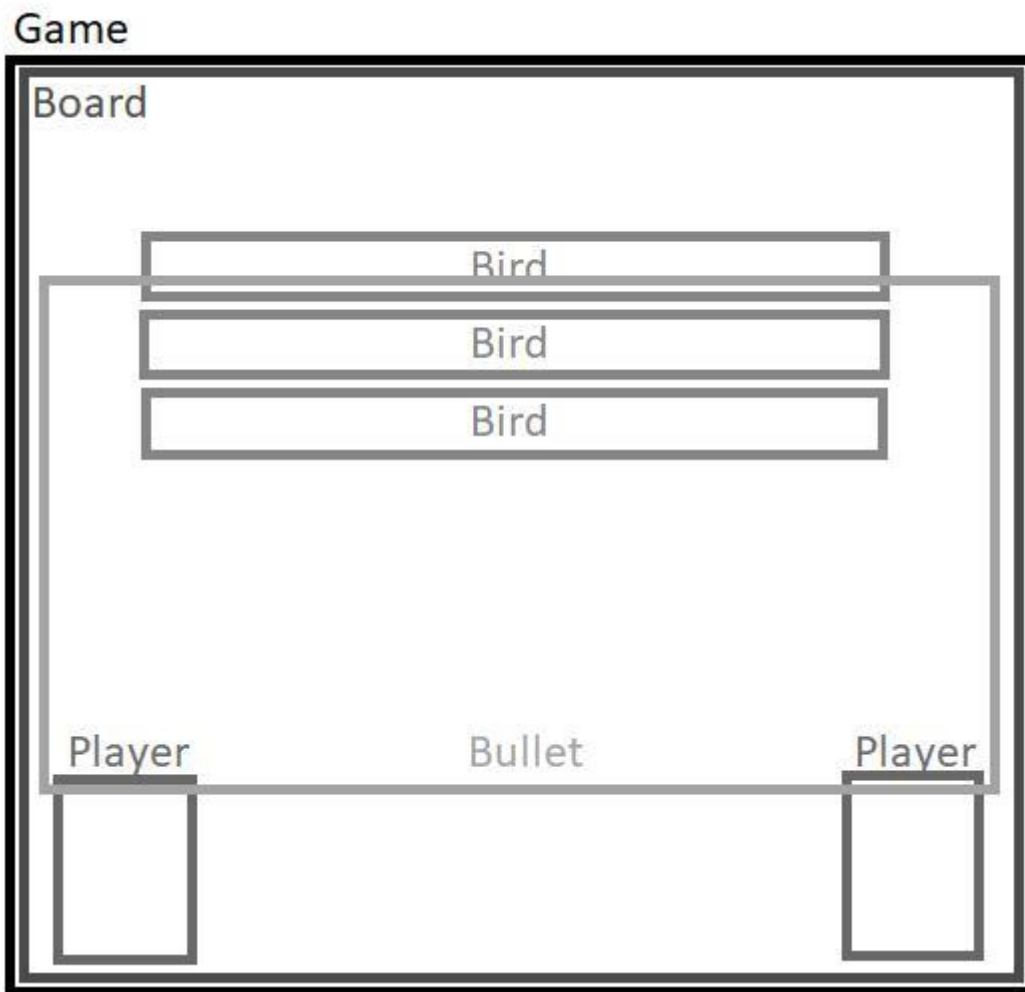
Realizacija projekta Nutella je izvršena korišćenjem Python jezika, multiprocessing biblioteke i PyQt5 okvira.

U narednim poglavljima će biti prikazan opis rada aplikacije, kao i kako i zbog čega su korišćeni navedeni programski jezik, biblioteka i okvir.

Opis rada aplikacije

Igricu čini 5 celina:

- Game ○
- Board ○
- Player
- Nutella ○
- Bullet



Slika 1 – Prikaz celina aplikacije

Game je klasa QMainWindow-a sa ulogom osnovne aplikacije koja sadrži klasu Board, sačinjenu od Player, Nutella i Bullet klasa i na kojoj se prikazuje tok

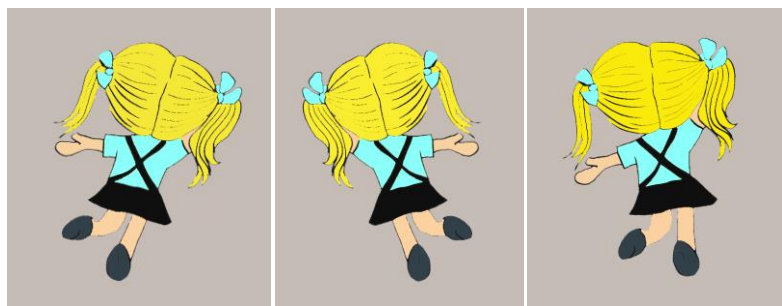
izvršavanja aplikacije. U njoj je postavljeno centriranje aplikacije na ekranu, naziv aplikacije, ikona i pozadina vizuelnog identiteta igrice.

Board je objekat klase *QFrame*, osnovne klase iz *PyQt5* okvira za pravljenje widget-a sa ramom. Ova klasa se koristi za kreiranje jednostavnog prostora za implementaciju aplikacije. U našem projektu, klasa *Board* sadrži sve neophodne labele da bi se napravila igrice, koje se zahvaljujući *QFrame* klasi nalaze na glavnom prozoru aplikacije. Pored labela, *Board* sadrži i metodu za postavljanje nutella u šemu kretanja, metodu za izvršavanje potrebnih promena tokom aplikacije, metodu za proveru da li je došlo do kontakta 2 labele, metodu za kretanje nutella, metodu za promenu sprite-a nutella, metodu za ispaljivanje projektila od strane nutella, metodu za inicijalizaciju i kretanje velike nutelle, metode za detekcije da li je taster pritisnut ili pušten nakon pritiska, metode za kretanje igrača, promenu izgleda tokom pokreta i ispaljivanje projektila, metodu za završenje igrice i prikaz pobednika, metodu za promenu nivoa, metodu za dobijanje informacije koji igrač je pogodio pticu i metodu za postavljanje timer-a da bi se znalo kada će se osvežiti trenutno stanje aplikacije.

Player klasa je tipa *QLabel* iz klase *QtWidgets* zadužena za postavljanje različitih objekata na *Board*. Ona predstavlja igrača zadatih dimenzija, dodeljenog broja života i postavljenog na board.

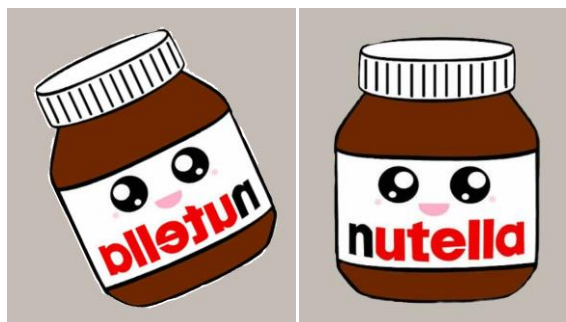


Slika 2 – Player 1 i njegove promene pri kretanju



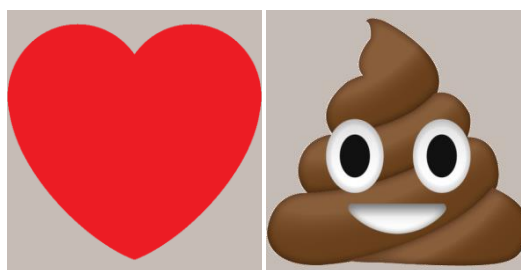
Slika 3 – Player 2 i njegove promene pri kretanju

Nutella klasa je labela koja predstavlja nutelle. Sadrži metode za postavljanje dimenzija na x i y osi Board-a, postavljanje i pomeranje nutella po Board-u.



Slika 4 – Nutella i njene promene pri kretanju

Bullet klasa služi za inicijalizaciju projektila koje igrač ili nutella ispaljuje. Ona poseduje metode za proveru koji projektil je u pitanju, u zavisnosti nad kojom labelom se poziva (igrač ili nutella) i njegovo postavljanje na Board.



Slika 5 – Projektili igrača i nutella

Pored ovih klasa, aplikacija ima još i klasu *EndGame* koja prikazuje kraj igre.

Pri pokretanju aplikacije, prikazuje se igrice smeštena na sredinu ekrana monitora. Oba igrača su postavljeni u uglove i naspram njih u gornjem uglu okvira se nalazi odgovarajući broj života. Ispod broja života se nalaze nutelle, postavljene u formaciju 3x10 koji pri startovanju igrice započinju svoje kretanje po šemi definisanoj u metodi klase Board i počinju sa nasumičnim ispaljivanjem projektila, što je takođe opisano u istoj klasi. Igrači se kreću i ispaljuju projektele potpuno nezavisno jedan od drugog, što je omogućeno upotrebom Timer event-a, koji generiše događaj nakon postavljenog vremenskog intervala.



Slika 6 – Prikaz aplikacije pri pokretanju

Ukoliko projektil nutelle pogodi igrača, taj igrač ostaje bez 1 života.



Slika 7 – Prikaz smanjenog broja života igrača nakon što je pogoden

Takođe, korišćenjem multiprocessing biblioteke, napravljena je dodatna instance klase Nutella, velika nutella, koja se kreće iznad i nezavisno od ostalih nutella. Njena uloga u igrici je vraćanje života onom igraču koji ju je pogodio



Slika 8 – Prikaz velike nutelle

Kada su sve nutelle pogođene, prelazi se na sledeći nivo. Moguće je dostići do 1000 nivoa, s tim što se na svakom novom nivou broj života igrača vraća na početnu vrednost od 3, povećava brzina kojom projektil leti ka igraču i šansa da će neprijatelj ispaliti projektil.

U momentu kada oba igrača izgube sve živote, aplikacija se završava pozivom metode klase EndGame I kao pobednik proglašava igrač koji je duže ostao u igrici.



Slika 10 – Prikaz proglašenja pobednika

Python

Python je interpretivni programski jezik višeg nivoa koji se fokusira na čitljivosti koda. Postao je popularan zbog svoje objektno-orijentisane funkcionalnosti, interaktivnosti, modularnosti, prenosivosti i velikog broja biblioteka sa automatskim upravljanjem memorijom.

Prednosti ovog programskog jezika su njegova snabdevenost standardnim bibliotekama koje pružaju mogućnost rada sa stringovima, web service alatima, protokolima i interfejsima operativnih sistema. Pored navedenog, podržava XML i slične markup jezike i pospešuje povećanje brzine i produktivnosti aplikacija.

Mane koje su karakteristične za Python jesu pre svega njegova jednostavnost, sa tačke gledišta prelaska sa učenja Python sintakse na ostale programske jezike. Iako je pogodan za razvoj desktop aplikacija i serverskih platform, vrlo je slab za razvoj proizvoda vezanih isključivo za mobilnu telefoniju. Zahteva više vremena provedenog na testiranju zbog svoje dinamičnosti i moguće je prijavljivanje grešaka iako aplikacija funkcioniše bez problema.

PyQt5

PyQt je GUI alat koji si koristi kao dodatak Python aplikacijama. Sadrži oko 440 klasa i njihovih funkcija koje mogu da pomognu programeru oko rada sa GUI widget-ima, SQL bazom podataka, XML parserom i abstrakcije mrežnih utičnica (sockets), regularnih izraza i razvoja funkcionalnih web browser-a.

Qt klase funkcionišu na signal/slot mehanizmu za komunikaciju između objekata i tako olakšavaju kreiranje komponenti koje se mogu iskoristiti na različite načine.

Zaključak

Zahvaljujući upotrebi Python programskog jezika, projektovanje aplikacije je vršeno paralelnim programiranjem, gde je akcenat bio na razdvajanju projekta na više komponenti i njihovom simultanom izvršavanju, nezavisno od ostalih celina.

Pri tome nam je pomoglo korišćenje multiprocessing biblioteke. Na taj način smo, korišćenjem Queue klase koja sadrži potrebne metode, uspeli da napravimo veliku pticu koja se kreće nezavisno od ostalih nutella uz provere da li su izvršene neophodne operacije za računanje njenog pomeranja.

Upotrebom PyQt5 okvira smo omogućili razvijanje nezavisnih komponenti koje su spojene (binding-om) sa Python kodom i na taj način čekaju korisnika da bi znale kako da reaguju (pristisak na određeni taster da bi se igrač pomerio) ili čekaju određeni momenat da izvršile akciju (nutelle izbacuju projekte u zavisnosti od Timer događaja). Takođe, klase Qt okvira sadrže potrebne widget-e na kojima smo spojili sve komponente radi funkcionisanja aplikacije.

