

# Dokumentacija za projekat “Crazy Cars”

---

Distribuirani računarski sistemi u elektroenergetici  
2020/2021

Studenti:

Ivan Marčeta PR158/2017  
Aleksa Bajčetić PR119/2017  
Đorđe Jelić PR129/2017  
Marija Mitrić PR90/2017  
Željana Mirković PR128/2017

# Sadržaj

1. Uvod.....	3
2. Dizajn.....	4
2.1. Rad aplikacije.....	5
2.2. Korišteni paketi .....	7
3. Zaključak .....	8

# 1. Uvod

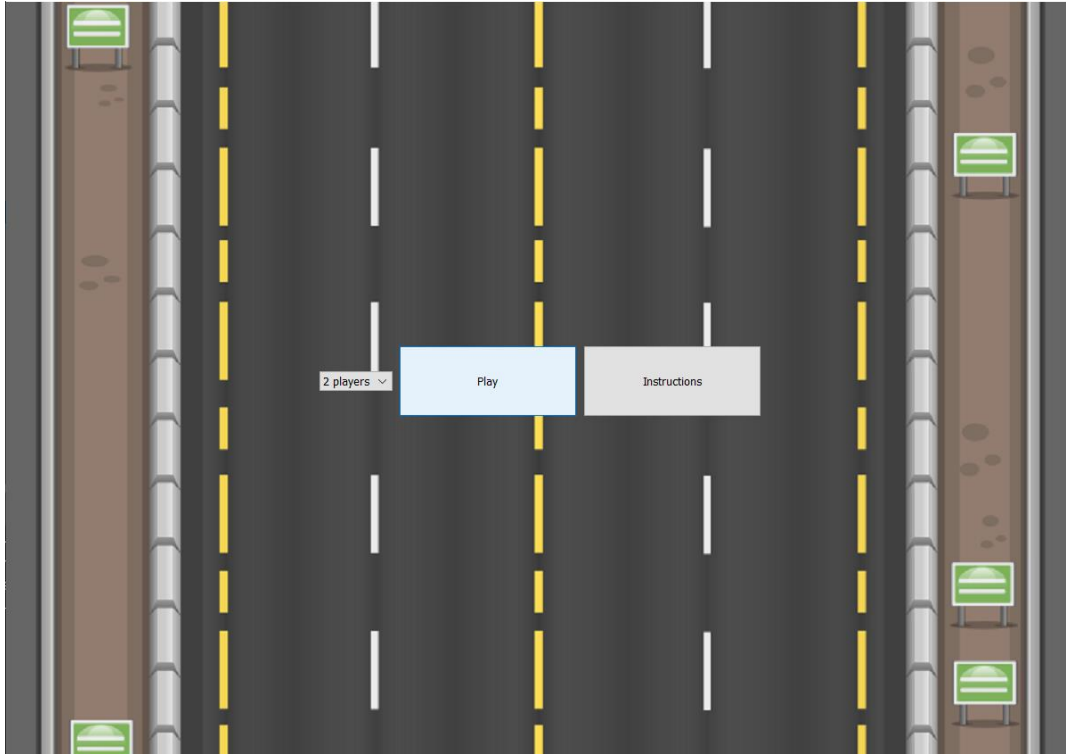
Projekat "Crazy Cars" predstavlja igricu napravljenu po uzoru na 2D Car Racing.

Ideja igre 2D Car Racing je da se korisnik nalazi u ulozi automobila, koga kontroliše uz pomoć strelica desno i lijevo. U centralnom dijelu ekrana se nalaze druga motorna vozila, u ulozi neprijatelja. Cilj igrača jeste da zaobiđe neprijatelje i pokupi novčiće koji se nalaze na putu. Ukoliko se sudari sa neprijateljem, igra se završava.

U nastavku će biti opisana realizacija igre "Crazy Cars".

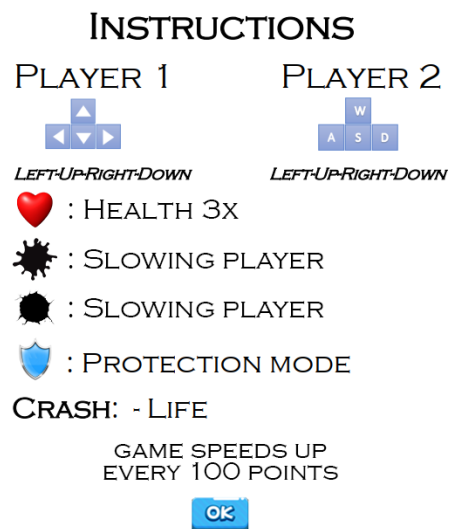
## 2. Dizajn

Aplikacija je razvijena za rad u dva režima. U prvom, korisnicima je omogućena igra u multiplayer režimu u okviru istog računara. Dok drugi režim omogućava igranje turnira, u kome može učestvovati 4 ili 8 igrača.

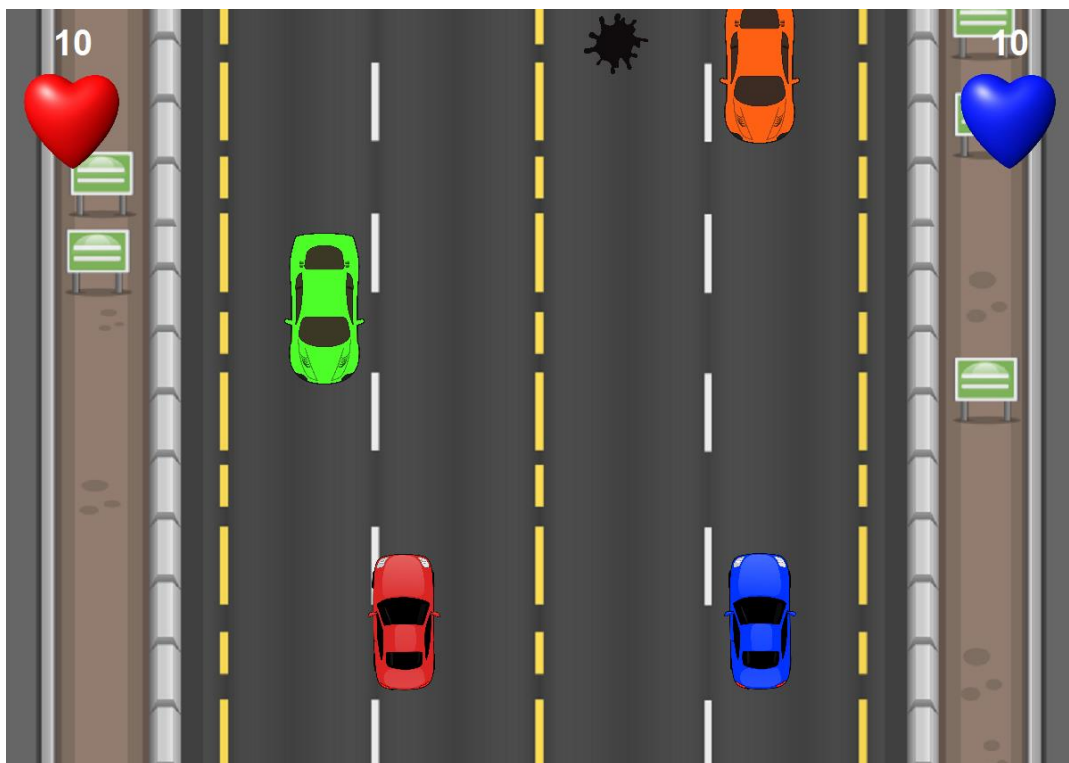


Slika 1. Home page

Na slici 1 prikazan je početni izgled aplikacije, na kome imamo tri opcije. Prva opcija nam služi za izbor broja igrača za igranje turnira, druga opcija predstavlja pokretanje samog turnira. Na trećoj opciji se nalaze instrukcije (Slika 2), koje korisniku ukratko objašnjavaju kako igra funkcioniše.



Slika 2. Instrukcije



Slika 3. Prozor igre

Može se uočiti nekoliko cjelina koje su od značaja. U gornjim dijelovima prozora nalazi se broj trenutno sakupljenih poena I preostalih života za oba igrača, što čini prvu cjelinu. Druga cjelina sadrži sam interfejs aplikacije, odnosno polje za igru.

## 2.1. Rad aplikacije

Aplikacija inicijalno pokreće dva igrača I više padajućih objekata. Igrači primaju komande putem tastature, I to koristeći strelice za jednog, **w**, **a**, **s**, **d** tastere za drugog. Pritiskom na bilo koji od pomenutih tastera šalje se komanda za kretanje komponente koja je zadala datu komandu. Komanda se dalje analizira I obrađuje ukoliko je to moguće. Skor se povećava za 10 svake sekunde tokom trajanja igre. Kada igrač skupi 100 poena, poveća se brzina objekata koji padaju I sakupljanje postaje teže.



Slika 4. Padajući objekti

Postoje tri vrste akcija pri interakciji igrača sa padajućim objektima:

- Pri interakciji igrača sa automobilima, igrač gubi život I vraća se u početni položaj (ukoliko nije izgubio I poslednji život, jer je u tom slučaju igra završena).
- Pri interakciji igrača sa uljem/flekom dolazi do usporavanja brzine igrača na 3 sekunde
- Pri interakciji igrača sa štitom, igrač postaje zaštićen od ostalih padajućih objekata na 7 sekundi

Igrači I padajući objekti imaju ograničeno kretanje u okviru glavnog prozora, tako da se ne mogu kretati van linija puta. Prilikom sudara da drugim automobilom ili kada pokupi zaštitu, igrač postaje untouchable, tj. ne može se desiti nikakva interakcija između igrača I objekata.

Ukoliko jedan od igrača izgubi sve živote, prikazuje se novi prozor u kome se nalazi naziv onog igrača koji je pobijedio (Slika 5). Ukoliko je u pitanju turnir, ovaj prozor će se pojaviti samo na kraju turnira.



*Slika 5. Prikaz pobjednika*

Klikom na bilo koji dio prozora sa slike 5, igrač se vraća na Home page gdje ponovo može da započne igru.

Kada se pokrene program, pokreću se dva procesa (main process, JobWorker process). Main process pokreće thread-ove I update-uje igricu, jobWorker šalje uputstvo na koji način sve da se radi. Imamo 6 thredova koji su parovi zaduženi su za blinkanje I untouchable automobile I jedan poseban thred koji update-uje skorove.

## 2.2. Korišteni paketi

U “Crazy Cars” aplikaciji korišteni su ugrađeni paketi kao što su “sys”, “time”, “threading”, kao i “multiprocessing”. Paket koji nije ugrađen, a korišten je “Pyqt5”. U daljem tekstu ukratko će biti dat opis svih korištenih paketa kao i funkcionalnosti koje oni pružaju, a koje su korištene u okviru aplikacije.

- “sys” – Ugrađeni paket koji pruža pristup promjenljivima ali i funkcijama korištenim i održavanim od strane interpretera. Paket je uvijek dostupan u okviru aplikacije Iz njega je korištena funkcija “sys.exit()”.
- “time” – Ugrađeni paket koji pruža razne vremenski povezane funkcije. Paket je uvijek dostupan u okviru aplikacije i iz njega je korištena funkcija “time.sleep(seconds)” koja suspenduje izvršavanje niti iz koje je pozvana za odgovarajući broj sekundi.
- “threading” – Ugrađeni paket koji konstruiše i pruža interfejse višeg nivoa nad nižim paketom “thread”. Paket je uvijek dostupan u okviru aplikacije Iz njega su korištene klase i funkcije “Thread” odnosno “threading.Thread(group=None, target=None, name=None, args=(), kwargs=())”. Parametri koji su bitni prilikom poziva jesu “target” u koji se smješta funkcija koja će biti izvršavana, kao i “args” u koji se smještaju potrebni parametri funkcije ukoliko postoje.
- “multiprocessing” – Ugrađeni paket koji pruža kreiranje procesa koristeći API sličan “threading” paketu. Paket “multiprocessing” pruža lokalnu ali i druge vrste konkurentnog rada efektivno zaobilazeći Iz ovog paketa u aplikaciji korištena je „Process“ klasa koja je zadužena za kreiranje „Process“ objekta i pozivanja njegove metode „Process.start()“. Takođe iz paketa „multiprocessing“ korišten je i “multiprocessing.Pipe” odnosno klasa koja implementira “Pipe” na “thread and process safe” način.
- “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.

### 3. Zaključak

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

Upotrebom PyQt5 okvira smo omogućili razvijanje nezavisnih komponenti koje su spojene sa Python kodom i na taj način čekaju korisnika da bi znale kako da reaguju (pristisak na određeni taster da bi se igrač pomjerio). Takođe, klase Qt okvira sadrže potrebne widget-e na kojima smo spojili sve komponente radi funkcionisanja aplikacije.

Kao potencijalno unapređenje nad aplikacijom tim vidi upotrebu "PyGame" paketa, kojim bi se olakšalo iscrtavanje ali i sinhronizaciju grafičkog interfejsa sa podacima, kao i bolja sinhronizacija između samih niti.