

CAR GAME

CS324 – Skripting jezici

Profesor: Student:

Nemanja Zdravković Andrijana Jovanović 3918

**Sadržaj**

1. Uvod i teoretska postavka izabrane teme ………………………………… 3

2. Primer/studija slučaja primene izabrane teme ……………………………. 6

3. Zaključak …………………………………………………………………. 15

**1. Uvod i teoretska postavka izabrane teme**

**Instalacija:**

Za izradu projektnog zadatka sam koristila **PyGame** biblioteku. Da bi se na racunaru pokrenuo program koji je napisan uz koriscenje biblioteke PyGame, potrebno ju je prethodno instalirati. Preduslov je naravno da na racunaru imamo instaliran Python. Kada je na racunaru instaliran Python, mozemo preci na instalaciju biblioteke PyGame. Dovoljno je da u komandrnoj liniji otkucamo pip install pygame.

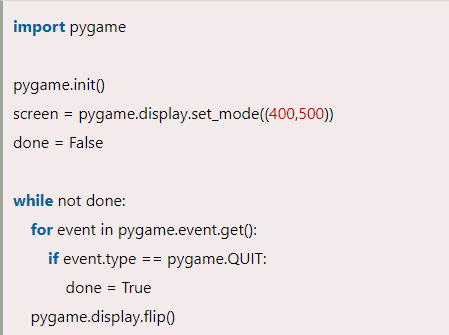
**PyGame biblioteka:**

Pygame je višeplatformski skup Python modula koji se koristi za kreiranje video igara. Sastoji se od kompjuterske grafike i biblioteka zvuka dizajniranih da se koriste sa programskim jezikom Python. Pygame dodaje funkcionalnost povrh odlične SDL biblioteke. Ovo nam omogućava da kreiramo potpuno opremljene igre i multimedijalne programe na jeziku Python.

Pygame je veoma prenosiv i radi na skoro svakoj platformi i operativnom sistemu.

Jednostavan primer pygame:

Evo jednostavnog programa pygame koji daje osnovnu ideju o sintaksi



**import pygame** - Ovo omogućava pristup pygame okviru i uvozi sve funkcije pygame-a.

**pygame.init()** - Ovo se koristi za inicijalizaciju svih potrebnih modula pygame-a.

**pygame.display.set\_mode((širina, visina))** - Ovo se koristi za prikaz prozora željene veličine. Povratna vrijednost je Surface objekat koji je objekt na kojem ćemo izvoditi grafičke operacije.

**pygame.event.get()-** Ovo se koristi za pražnjenje reda događaja. Ako to ne pozovemo, prozorske poruke će se početi gomilati i igra će po mišljenju operativnog sistema prestati da reaguje.

**pygame.QUIT** - Ovo se koristi da prekinemo događaj kada kliknemo na dugme za zatvaranje u uglu prozora.

**Pygame Surface**

Pygame Surface se koristi za prikaz bilo koje slike. Surface ima unaprijed definiranu rezoluciju i format piksela. Boja površine je po defaultu crna. Njegova veličina je definirana prosljeđivanjem argumenta veličine.

**Pygame Clock**

Vremena su predstavljena u milisekundama (1/1000 sekundi) u pygame-u. Pygame sat se koristi za praćenje vremena. Vreme je bitno za stvaranje pokreta, reprodukciju zvuka ili reagovanje na bilo koji događaj. Vreme ne računamo u sekundama. Brojimo ga u milisekundama.

**Pygame Blit – blit()**

Pygame blit je proces prikazivanja objekta igre na površini, a ovaj proces se naziva blitting. Kada kreiramo objekt igre, moramo ga renderovati. Ako ne renderujemo objekte igre i ne pokrenemo program, onda će nam dati crni prozor kao izlaz. Ova funkcija se koristi za crtanje jedne slike u drugu.

**Pygame Rect**

Rect se koristi za crtanje pravougaonika u Pygameu. Pygame koristi Rect objekte za skladištenje i manipulaciju pravougaonim oblastima. Rect se može formirati iz kombinacije vrednosti levo, vrha, širine i visine. Također se može kreirati od Python objekata koji su već Rect ili imaju atribut pod nazivom "rect".

Rect() funkcija se koristi za izmenu položaja ili veličine pravougaonika. Vraća novu kopiju Rect-a sa izmenama. U originalnom pravougaoniku se ne dešavaju nikakve izmene.

**Pygame Draw**

Pygame pruža geometrijske funkcije za crtanje jednostavnih oblika na površini. Ove funkcije će raditi za prikazivanje površina u bilo kojem formatu.

**Pygame tekst i font**

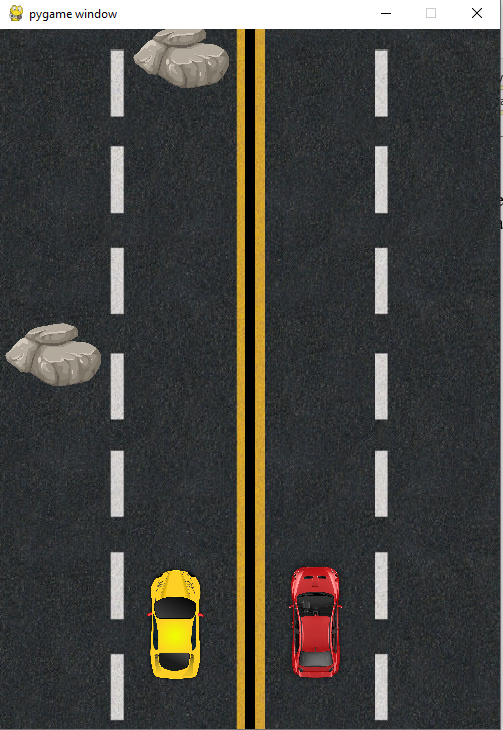
Pygame takođe pruža mogućnosti za prikazivanje fonta i teksta. Možemo učitati fontove iz sistema pomoću funkcije pygame.font.SysFont (). Objekti fontova su kreirani pomoću pygame.font.Font ().

**Pygame Collision**

Collision detection je proces kada se dva objekta na ekranu sudare. Na primer, ako je igrač pogođen neprijateljskim metkom, tada može izgubiti život ili, program mora znati kada igrač dodirne novčić kako bi ga automatski pokupio.

**2. Primer/studija slučaja primene izabrane teme**

Na samom pocetku pokazacu vam kako izgleda aplikacija kada se pokrene.

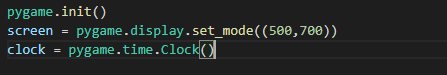


Za ovu igru su potrebna dva igrača. Jedan igrač pomera zuti automobil (A dugme -na levo i D dugme - na desno), a drugi igrač pomera crveni automobil(LEFT – za levo i RIGHT – za desno). Na putu se nasumično stvaraju stene i igrači moraju da ih izbegavaju. Ukoliko igrač sa zutim automobilom udari u stenu, pobednik je onda crveni automobil. Kao sto je prikazno na slici ispod.



**Kod projektnog zadatka:**

Prvo importujemo pygame biblioteku.



**Pygame.init()** se koristi za inicijalizaciju svih potrebnih modula pygame-a.

Screen nam omogucava prikaz nase aplikacije gde podesavamo visinu i sirinu prozora.

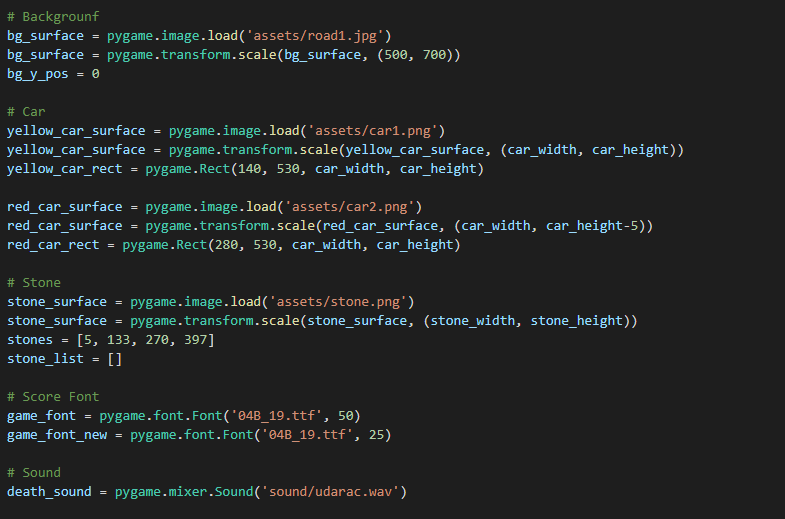
Na sledecoj slici ucitavamo slike koje sun am potrebne kao sto je pozadina, zuti automobil, crveni automobil, stena, font i zvuk. Za ucitavanje svega sto smo nabrojali se koristi **pygame.image.load()** – koji ucitava slike.

**pygame.Rect() –** omogucava da kreiramo pravougaonik na odredjenim (x,y) kordinatama u visini i sirini automobila.

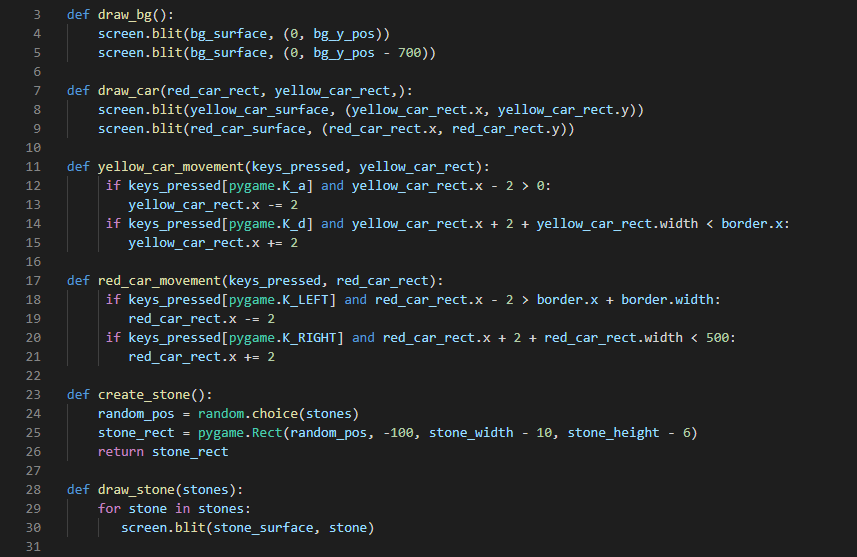
**pygame.transform.scale()** - nam omogucava da podesimo zeljenu visinu i sirinu slike.

Za ucitanje fonta koristimo **pygame.fonr.Font()** I u zagradi stavljamo naziv naseg fonta koji se nalazi u folderu i zadajemo velicinu fonta.

**pygame.mixer.Sound() –** ucitavanje zvuka.



Objasnjenje svake od metoda:



**Def draw\_bg() -** ovom metodom pomocu **blit()** omogucavamo da se na ekran prikaze nasa pozadina na odredjenim pozicijama.

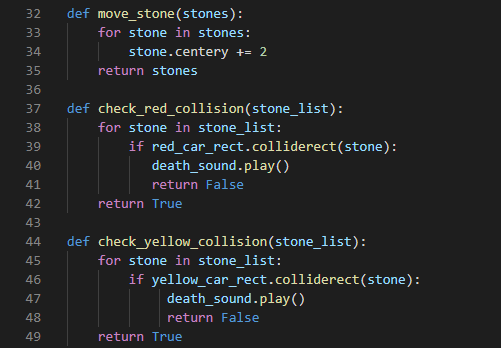
**Def draw\_car() –** takodje ovom metodom vrsimo prikaz crvenog i zutog automobila na ekranu koji smo smestili u odredjeni pravougaonik na odredjenim (x,y) kordinatama.

**Def yellow\_car\_momvement(keys\_pressed, yellow\_car\_rect) –** proveravamo da li je igrac kliknuo ‘A’ dugme na tastaturi i da li je x kordinata pravougaonika u kom je smesten automobil veca od 0 (kako automobil ne bi izasao iz ekrana). Ukoliku je uslov ispunjen onda se pravougaonik automobila pomera za 2 na levo.

Drugi uslov je ako je pritisnuto ‘D’ dugme na tastaturi i da je x kordinata+2+sirina pravougaonika zutog automobile manja od border-a, onda se ce automobile pomerati na desno do border linije.

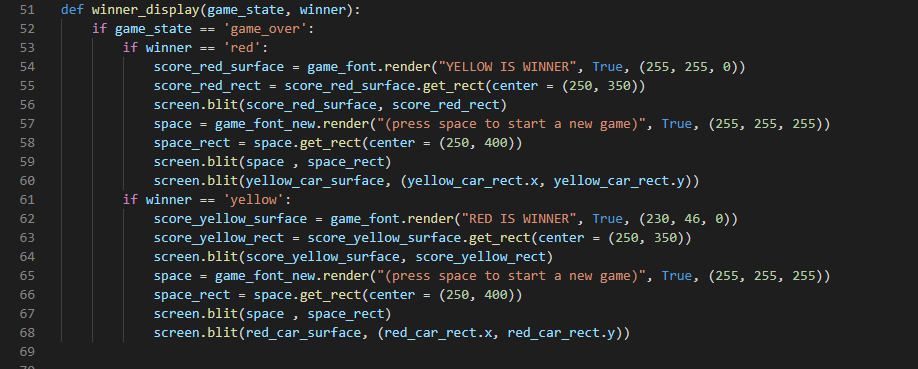
**Def create\_stone()** – omogucava nam da se stene random kreiraju. Iz zadate liste biraju se nasumicno brojevi koji ce pokazivati x kordinatu stvaranja stene na putu.

**Def draw\_stone(stones)** – prolazi se kroz listu u kojoj se nalaze (x, y) kordinate stena. Uzima se po jedna (x,y) kordinata (odnosno po jedan pravougaonik u koji se ubacuje ucitana slika nase stene) i prikazuje na ekran.

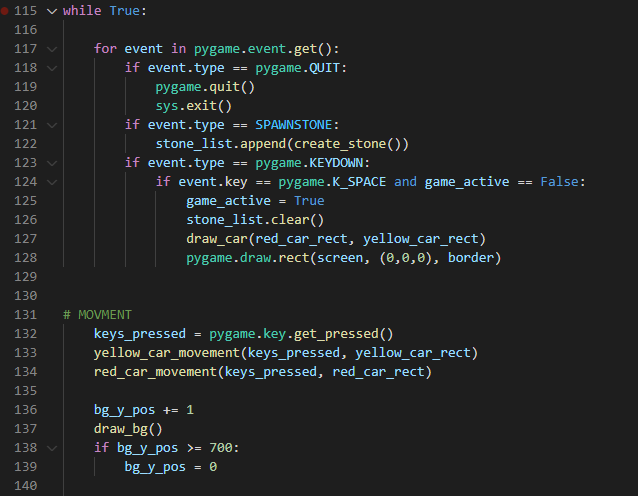


**Def move\_stone(stones)** – omogucava da se stena pomera. Prolazi se kroz listu u kojoj se nalaze pravougaonici stena(x,y, visina i sirina stene) i y kordinata ce se povecavati za 2 kako bi se stena pomerala po y kordinati (od vrha ekrana ka dnu ekrana).

**Def check\_red\_collision(stone\_list)** – prolazimo kroz listu u kojoj su smesteni pravougaonici stena(x, y, width, height). Pomocu metode **colliderect** proveravamo da li je pravougaonik zutog automobile dodirnuo pravougaonik stene I ukoliko jeste vraca se False i pokrecemo zvuk metodom **play().** Ukoliko nije doslo do dodira ova dva pravougaonika onda se vraca True.



**Def winner\_display(game\_state, winner) –** Proverava se da li je game\_state game\_over. Ukoliko jeste onda se proverava da li je winner == red, ako jeste onda se ucitava font koji zelimo, podesavamo sta zelimo da pise tim fontom i koje boje. Zatim se taj font smesta u pravougaonik na okredjenim kordinatama. Nakon toga metobom blit() postavaljamo nas font u pravougaonik i prikazujemo na ekran. Ovo sve isto vazi ukoliko je winner == yellow.



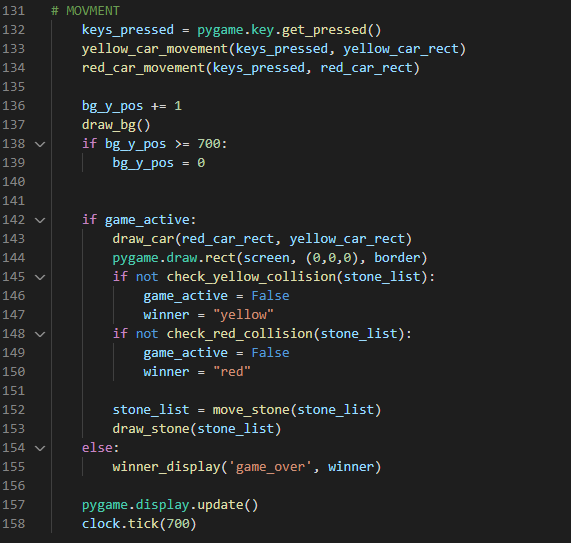
Unutar while prolazimo kroz sve event u pygame-u i ispitujemo da li je event.type == pygame.Quit i ukoliko jeste gasimo program.

Proveravamo da li je event.type == **SPAWNSTONE**(to je timer koji na svakih 500 milliseconds se kreira pravougaonik za stenu na odredjenim kordinatama).

Ukoliko je korisnik kliknuo **SPACE** i igra vise nije aktivna onda igra postaje aktivna, sve iz stone\_list se brise, na ekranu se pomocu metode draw\_car ponovo prikazuju automobile i iscrtava se border.

Metode koje su pozvane ispod: **yellow\_car\_movment(keys\_pressed, yellow\_car\_rect) i red\_car\_movment(keys\_pressed, red\_car\_rect)** omogucavaju pomeranje automobila levo desno na osnovu dugmeta koji je korisnik pritisnuo.

**Draw\_bg()** – omogucava prikaz slike puta na ekranu gde ce se y kordinata uvecavati za po 1 kako bi se vizulno dobilo da se put pomera.



Ukoliko je igra activna(True) – iscrtavace se automobile i border crne boje.

Proveravamo da li je doslo do udarca zutog automobila u stenu. Ako je doslo onda game\_active postaje = False i pobednik je crveni automobil. To sve isto proveravamo za crveni automobil i u tom slucaju pobednik je zuti automobil.

**3. Zaključak**

Glavna prednost pygame biblioteke je to što je jednostavnija od drugih za korišćenje i pogodna je za učenje programiranja kroz interesantan svet računarske grafike i računarskih igara.