

**Jesenji semestar 2023/2024**

**Predmet: CS324 - Skripting Jezici**

Ko želi da bude milioner

Profesor: Faruk Selimović Student: Uroš Lazarević

Asistent: Tamara Vukadinović Datum: 08.02.2025

[**1. Uvod 2**](#_xrnjfjciv9il)

[1.1. Postavka zadatka 2](#_mkm85x4flrkc)

[1.2. Gameplay i mekahina 2](#_vcmhl0avgsol)

[**2. Metodologija 4**](#_lft2gpm0qs3w)

[2.1. Teoretske osnove 4](#_ix1tszwue6yv)

[2.2. Koncept pelje igre 4](#_nbleh7ulh8zt)

[2.3. Tkinter 4](#_sfm3trtx4pzw)

[2.4. PIL 5](#_nvzae5yv2ncd)

[2.5. CSV 5](#_yt4w8paiuug1)

[**3. Arhitektura 6**](#_z9zh37agiuum)

[**4. Zaključak 10**](#_vqefzjkbe69s)

[**5. Literatura 11**](#_ped888eqyzo6)

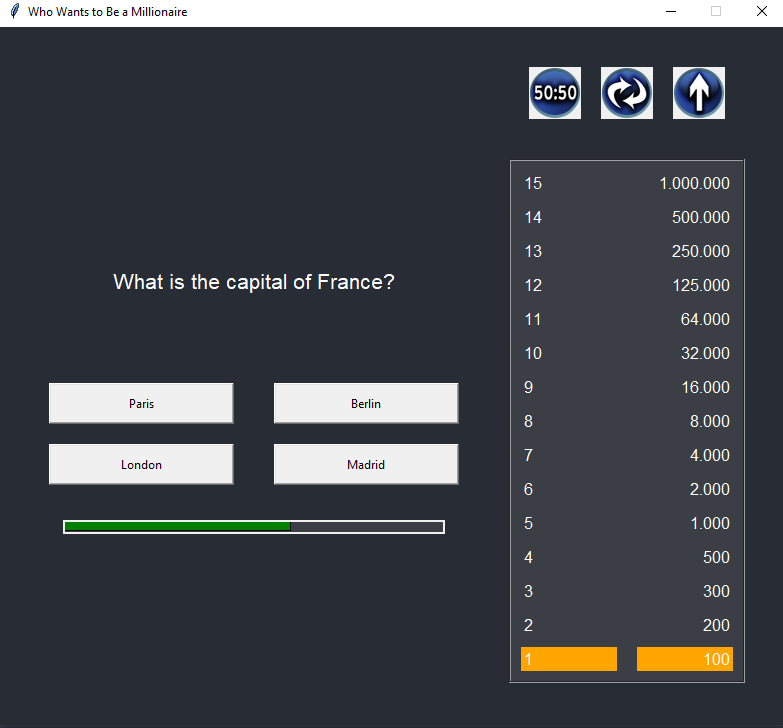
# Uvod

## Postavka zadatka

Iz predmeta CS324, Scripting jezici, projektni zadatak je video igra “Ko želi da bude milioner”. Igra je izradjena u Python programskom jeziku korišćenjem PyCharm razvojnog okruženja. Za izradu projekta korišćene su biblioteke tkinter, csv, random i PIL.

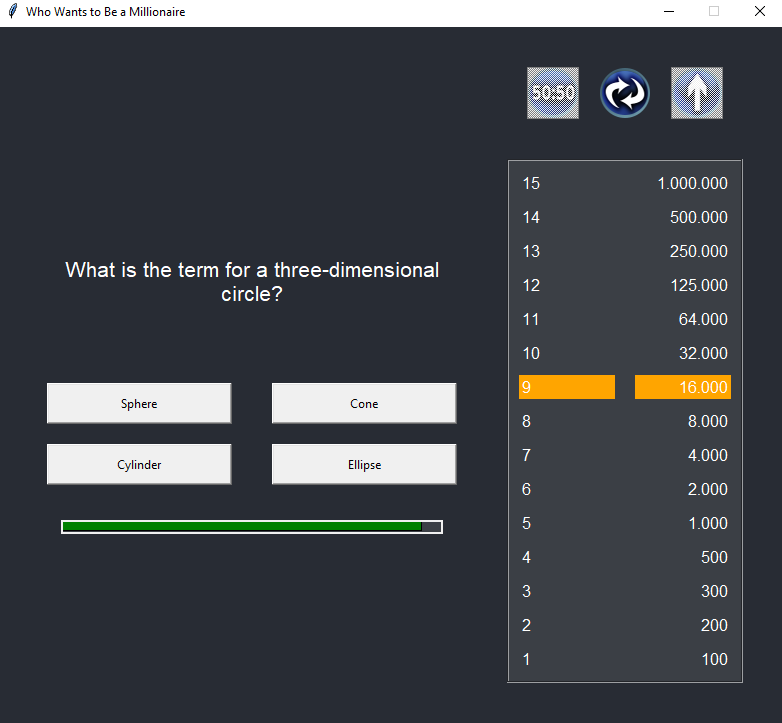
## Gameplay i mekahina

Rešenje projektnog zadatka je igra “Ko želi da bude milioner.” Kada se igra pokrene igraču se prikazuje user interfejs pomoću kog se igra odvija. Sa leve strane ima postavljeno pitanje, četiri dugmeta za odgovor i indikator preostalog vremena. Sa desne strane ima tri dugmeta za pomoć, kao i indikator na kome se pitanju igrač trenutno nalazi i koja je to suma novca.



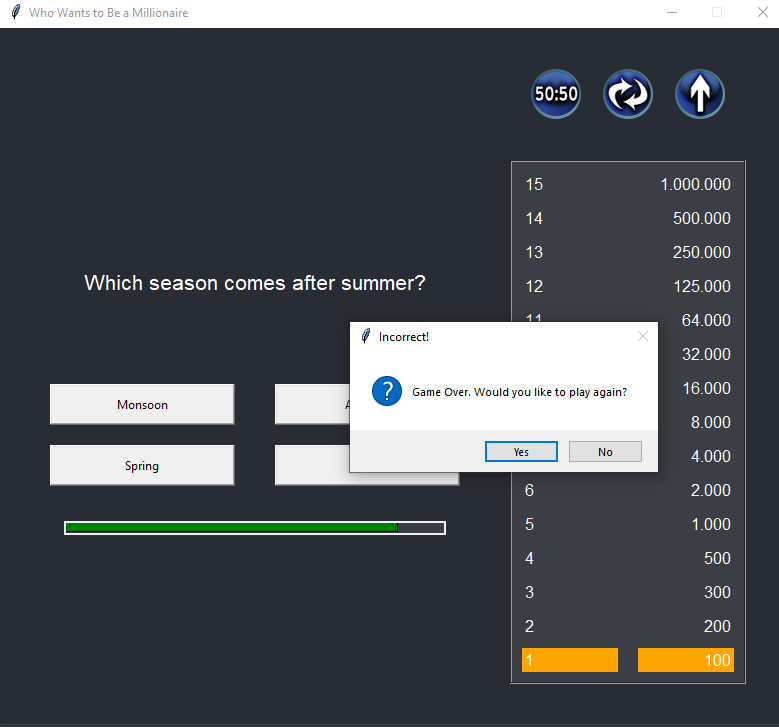
*1. inicijalni prikaz igre*

Ukoliko igrač odgovori tačno prosleđuje na sledeće pitanje. Igraču su takoće ponuđene 3 pomoći, 50-50, zamena i preskoči pitanje. 50-50 igraču briše dva odgovora i u ponudi mu ostaju jedno tačno i jedno netačno rešenje, zamena menja pitanje sa nekim drugim pitanjem iste težine, dok preskoči pitanje nastavlja igru dalje.



*2.prikaz igranja igre*

Ako igrač u nekom trenutku izabere netačan odgovor izlazi mu poruka da je pogrešio i nudi mu se opcija da igra ispočetka ili da prekine igru. U suprotnom ako dodje do kraja prikazuje se poruka pobede i nude se iste opcije.

**

*3. prikaz gubitka igre*

# Metodologija

## Teoretske osnove

Da bi se izradila ovakva igra neophodno je razumeti gradivo koje je pređeno na predavanjima i vežbama, kao i dodatne biblioteke poput tkinter

## Koncept pelje igre

Petlja igre je primarni mehanizam koji pomera igru unapred u vremenu. U igrama postoje 3 osnovne faze: faza inicijalizacije, petlja igre i faza isključivanja. Kada se igra pokrene program ulazi u fazu inicijalizacije gde se učitavaju resursi potrebni za igru. Nakon toga igra ulazi u stanje pelje igre koje se izvršava dok igra traje. Kada se igra završi izvršava se faza isključavanja igre gde se oslobađaju resursi.

*4. logika petlje igre*

Petlja igre je sekvenca procesa koji se izvršavaju za vreme trajanja igre. To su ulaz, ažuriranje i scrtavanje. Proces ulaza ima za zadatak da kontroliše igru komandama koje igrač prosleđuje pomoću miša. Porces ažuriranja obuhvata sve zadatke potrebne za ažuriranja igre kao što su pozicija resorsa, kretanje resursa, da li su se dva resursa sudarila, kao i fiyika i sile u igri. Kada se sva izračunavanja završe potrebno je prikazati igraču rezultat. Za to je zadužen proces iscrtavanja.

## Tkinter

Tkinter je standardna biblioteka za pravljenje grafičkih interfejsa u Pythonu, dostupna bez dodatne instalacije. Svaka Tkinter aplikacija započinje kreiranjem glavnog prozora pomoću klase Tk(), unutar kojeg se dodaju različiti widgeti poput Label, Button, Entry, Text, Frame, Canvas, Listbox i drugih. Za raspoređivanje elemenata koriste se tri glavna menadžera rasporeda: pack() za jednostavno slaganje, grid() za organizaciju u tabele i place() za precizno pozicioniranje. Tkinter omogućava rad sa događajima i vezivanje funkcija na korisničke akcije, dok mainloop() osigurava da aplikacija ostane aktivna i reaguje na interakcije. Moduli messagebox i filedialog omogućavaju prikaz poruka i rad sa datotekama, dok tkinter.ttk donosi modernije stilizovane widgete. Korišćenjem varijabli poput StringVar, IntVar i BooleanVar, aplikacija može dinamički ažurirati podatke između elemenata. Tkinter je lagan i pogodan za manje do srednje složene aplikacije, pružajući jednostavan, ali moćan alat za razvoj desktop GUI interfejsa.

## PIL

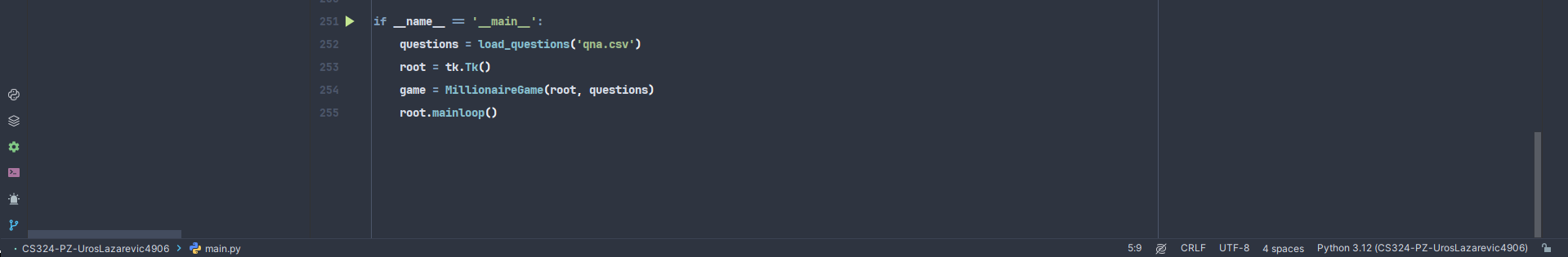
Pillow (PIL – Python Imaging Library) je popularna biblioteka za obradu slika u Pythonu, koja omogućava otvaranje, manipulaciju i spremanje slika u različitim formatima. Podržava rad sa formatima poput JPEG, PNG, BMP, GIF, TIFF i mnogim drugim. Koristi se za osnovne operacije poput promene veličine, rotacije, obrezivanja, filtriranja, konverzije između formata, kao i za naprednije tehnike poput crtanja po slikama, dodavanja teksta i rada sa prozirnošću. Klasa Image je osnovna komponenta za rad sa slikama, dok klasa ImageDraw omogućava crtanje oblika i teksta. Pillow pruža i ImageFilter modul za dodavanje efekata poput zamućenja, izoštravanja i ivica. Moduli ImageEnhance i ImageOps omogućavaju prilagođavanje kontrasta, osvetljenja i drugih parametara slike. Zahvaljujući svojoj jednostavnosti i širokoj podršci za rad sa slikama, Pillow se koristi u raznim aplikacijama, uključujući obradu fotografija, računarstvo u oblaku i veštačku inteligenciju.

## CSV

Modul csv u Pythonu omogućava jednostavno čitanje i pisanje CSV (Comma-Separated Values) datoteka, koje se često koriste za skladištenje tabelarnih podataka. Osnovne klase su csv.reader za čitanje i csv.writer za upis podataka u CSV format. Čitač parsira redove kao liste stringova, dok pisač omogućava dodavanje redova u CSV datoteku. Parametri poput delimiter**,** quotechar i escapechar omogućavaju prilagođavanje formata zapisa. Modul podržava i DictReader i DictWriter, koji rade sa podacima u obliku rečnika, omogućavajući jednostavno mapiranje kolona na ključeve. CSV fajlovi se otvaraju pomoću open() funkcije uz parametar newline='' kako bi se izbegli problemi sa novim redovima. Za rad sa različitim CSV formatima, može se koristiti csv.Dialect koji definiše pravila formatiranja. Modul csv je koristan za rad sa velikim količinama strukturiranih podataka, omogućavajući njihovu laku manipulaciju i integraciju sa bazama podataka i drugim aplikacijama.

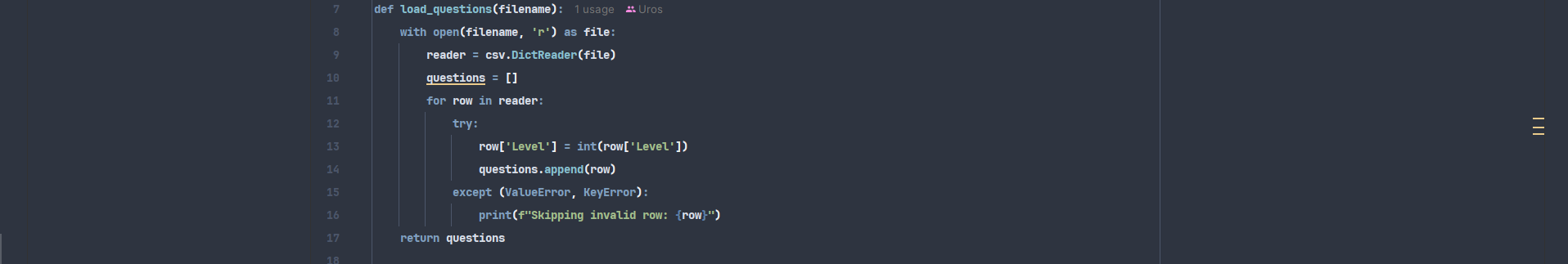
# Arhitektura

Pri pokretanju igre učitavaju se pitanja iz CSV datoteke, kreira se glavni prozor pomoću Tkintera-a i pokreće se instanca klase MillionareGame.



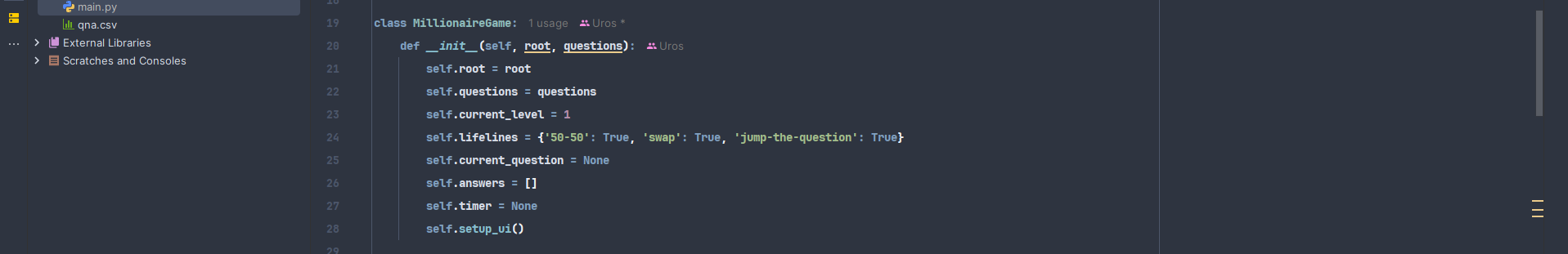
*4. inicijalizacija igre*

Funkcija load\_questions otvara CSV datoteku i čita pitanja koristeći csv.DictReader. Svaki red predstavlja jedno pitanje, a podaci se pretvaraju u rečnik. Ako postoji greška prilikom konverzije nivoa pitanja u broj, red se preskače uz ispis poruke.



*5. čitanje pitanja i odgovora iz csv datoteke*

Klasa MillionareGame kreira glavni prozor igre i postavlja početne vrednosti. Čuva listu pitanja, trenutni nivo igre i dostupne opcije za pomoć. Takođe, postavlja početnu vrednost za trenutno pitanje i listu mogućih odgovora.



*6. inicijalizacija klase MillionareGame*

Metoda setup\_up kreira grafički interfejs igre uključujući glavni okvir u kome se nalaze ostali elementi, levi okvir gde se prikazuje tekst pitanja, dugmad za odgovore i tajmer u obliku trake koja se smanjuje tokom odbrojavanja, desni okvir koji sadrži dugmad za pomoć i tabelu pitanja i nagrada koja prikazuje nivoe, potencijalne dobitke i indikator na kom se nivou trenutno nalazi igrač.

next\_question bira pitanje iz liste prema trenutnom nivou igre. Tekst pitanja i odgovori se postavljaju na odgovarajuća dugmat, a boje u tabeli nagrada se ažuriraju kako bi označile trenutno pitanje.



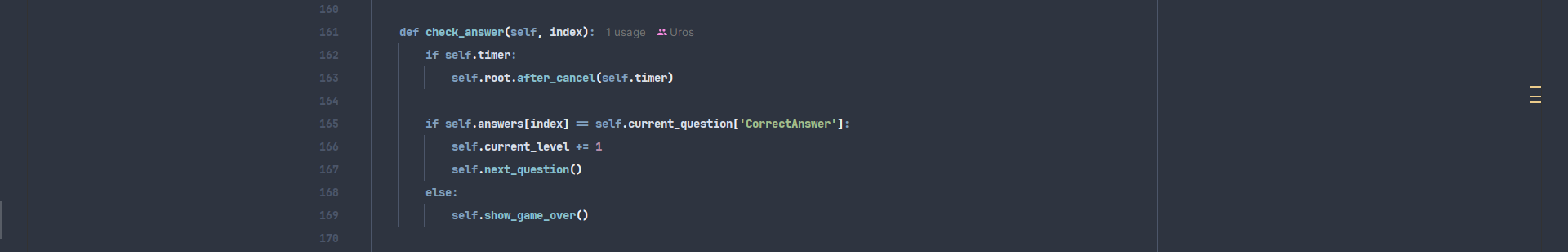
*7. metoda next\_question*

Tajmer se pokreće metodom start\_timer koja crta pravougaonik koji vizuelno prikazuje preostalo vreme. Funkcija update\_timer smanjuje širinu pravougaonika dok vreme ne istkene, nakon čega se igra završava.



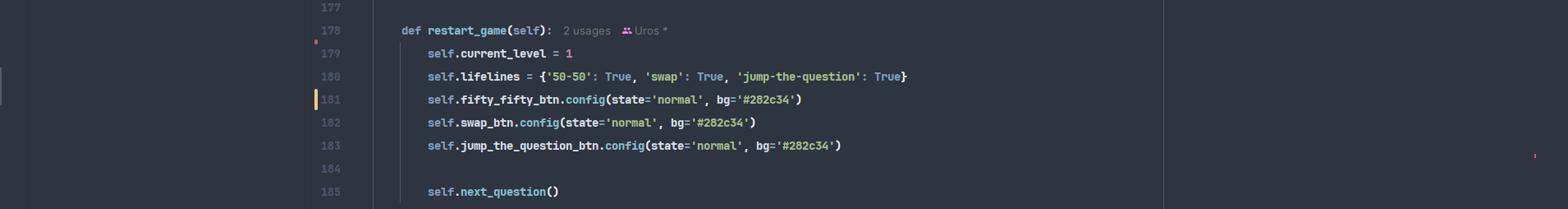
*8. metode za tajmer*

Kada igrač izabere odgovor, metoda check\_answer proverava da li je tačan. Ako je odgovor tačan nivo igre se povećava i učitava se novo pitanje. Ako je odgovor netačan, igra se završava i prikazuje se poruka u porazu.



*9. metoda check\_aanswer*

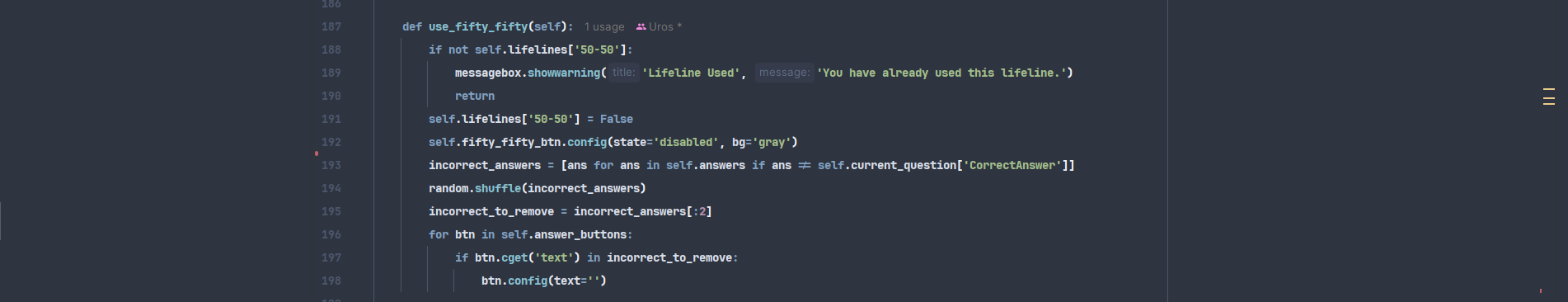
Ako igrač odgovori tačno na sva pitanja, prikazuje se poruka o pobedi sa opcijom ponovnog igranja. U slučaju poraza, šrikazuje se poruka sa istim izborom. Ako igrač želi da igra ponovo, poziva se metoda restart\_game, koja restartuje nivo igre i opcije pomoći.



*10. metoda restart\_game*

Igrač u svakom trenutku ima mogućnost aktiviranja pomoći. Nakon što igrač odabere jednu ili više, ista se ne može aktivirati do restartovanja igre.

* “50-50” uklanja dva netačna odgovora sa ekrana



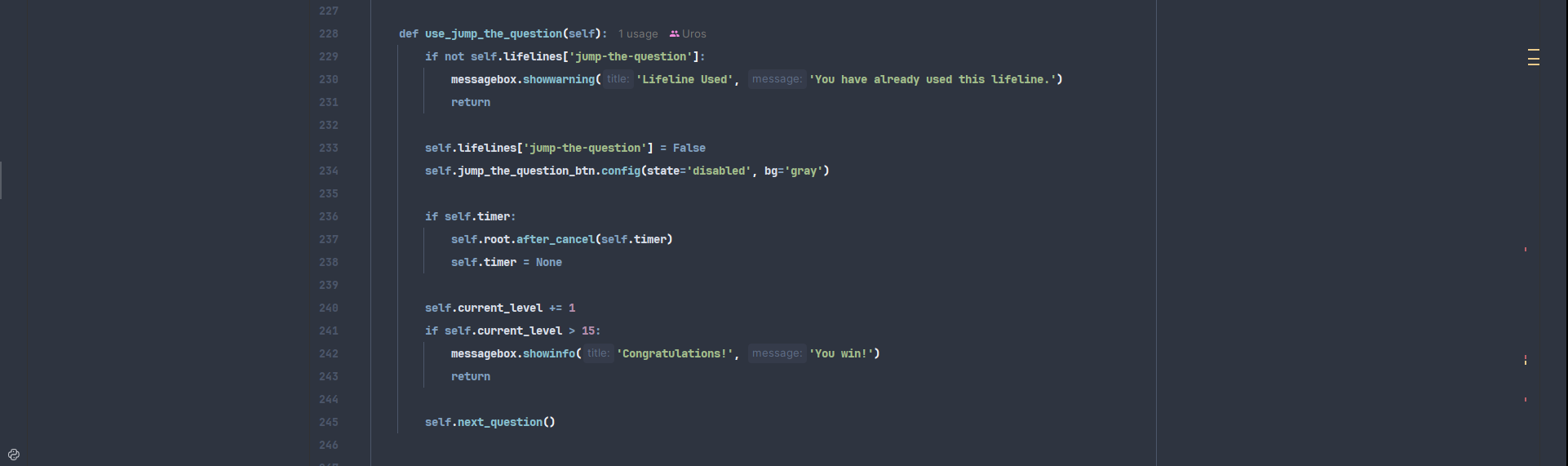
*11. metoda “50-50”*

* “zamena” menja trenutno pitanja sa drugim iz iste kategorije.



*12. metoda “zameni pitanje”*

* “preskoči pitanje” preskače trenutno pitanje i automatski prelazi na sledeći nivo. Ukoliko je to poslednje pitanje igrač automatski pobeđuje



*13. metoda “preskoči pitanje”*

# Zaključak

Igra koja je izrađena za projekat zadovoljava sve zahteve projektnog zadatka. Nakon izrade utvrio sam znanje stečeno na predavanjima i vežbama. Takođe sam morao neke stvari sam da istražim i naučim poput grafičkog interfejsa tkinter. Igru možemo usavršiti novim pitanjima i promenama kvaliteta života, poput dodavanja zvukova i menija.

# Literatura

* 1. [lams.metropolitan](http://lams.metropolitan.ac.rs/)
  2. [oreilly](https://www.oreilly.com/library/view/opengl-game-development/9781783288199/ch01s02.html)
  3. [gameprogrammingpatterns](https://gameprogrammingpatterns.com/game-loop.html#:~:text=A%20game%20loop%20runs%20continuously,control%20the%20rate%20of%20gameplay.)
  4. [Pillow](https://www.geeksforgeeks.org/python-pillow-a-fork-of-pil/)
  5. [tkinter](https://docs.python.org/3/library/tkinter.html)
  6. [csv](https://docs.python.org/3/library/csv.html)