RAZVALI BUNDEVU

Tehnička dokumentacija

Autor: Vjekoslav Radošević

UVOD

Riječi o autoru

Zovem se Vjekoslav Radošević, učenik sam 4.5 razreda srednje škole Gimnazija Andrije Mohorovičića Rijeka. Kao završni projekt svog školovanja u GAM-u odlučio sam napraviti video-igru u programskom jeziku Python. Učenje Pythona bio je jedan od glavnih djelova plana i programa nastave informatike te smatram kako imam dovoljno znanja kako bih izvršio svoj zadatak. Radeći program, naučio sam mnogo o Pythonu, kao i o modulu Pygame u kojemu je napravljena igrica.

Ideja projekta

Ideju o izradi video-igre u Pythonu dobio sam upravo igrajući video-igru te je moj cilj bio napraviti dvodimenzionalnu igru sa jednostavnim sustavom kretanja i borbe. Temelj igre je kreacija lika kojega je moguće pomicati, a zadatak je pobijediti sve umjetno kontrolirane elemente na ekranu koji imaju unaprijed zadani patent kretanja. Video-igra je kratkoga trajanja sa grafički jednostavnim elementima koji su prezentirani naizmjeničnim prikazivanjem slika. Važno je napomenuti kako nisam autor slika likova, slike su preuzete sa interneta.

DETALJAN OPIS RADA

Mehanike video-igre

KRETANJE LIJEVO I DESNO:

Jedna od glavnih značajki video igre je kretanje glavnoga lika po x-osi. Lik se može kretati u lijevo pritiskom lijeve strelice na tipkovnici pri čemu se njegova trenutna pozicija mijenja u negativnom smijeru na x-osi. Također se lik može kretati i u desno pritiskom desne strelice na tipkovnici pri čemu se njegova vrijednost pozicije na x-osi povećava.

BRZINA:

Svi pomični elementi na ekranu kreću se određenim brzinama ukoliko je programirano da se konstantno kreću ili je dana naredba za to. Sve su brzine elemenata postaju fiksne jednom kada je video-igra pokrenuta te njihove vrijednosti nije moguće mijenjati dok video-igra traje. Iznos brzine označava broj piksela za koji se element pomakne u jedinici vremena. Predznak brzine označava smijer kretanja lika po x-osi.

SKAKANJE:

Skok je način kako lika pomicati po y-osi. U ovoj video-igri skok je izveden po kvadratnoj funkciji što znači da se lik, ukoliko je u pokretu, za vrijeme skakanja giba po putanji parabole. Brzina će na početku skoka biti veća nego u tjemenu parabole zbog djelovanja negativne akceleracije u smjeru gibanja lika. Nakon prolaska kroz tjeme parabole brzina lika se povećava budući da na njega djeluje pozitivna akceleracija u smjeru kretanja prema dolje. Važno je napomenuti kako se lik pri kretanju prema gore giba u negativnom smjeru y-osi, a pri padu se giba u pozitivnom smjeru y-osi. Razlog tomu je koordinatni sustav pygame prozora čije početne koordinate (0,0) započinju u gornjem lijevom kutu.

IZLAZAK IZ PROZORA:

Kretanje glavnoga lika, kao i ostalih umjetno kontroliranih elemenata, nije dozvoljeno izvan veličine prozora video-igre. Naravno, kada bi bilo dozvoljeno da element napusti prozor nebi se javila pogreška u programu nego bi samo izgubili pogled na spomenuti element. Taj je problem riješen tako da je liku zabranjeno kretanje u zadanom smijeru ukoliko zbroj njegove lokacijae na x-osi i širine lika premašuje širinu prozora video-igre. Ovaj postupak vrijedi i za lijevi i za desni rub prozora.

UMJETNO KONTROLIRANI ELEMENTI:

Likovi koje igrač nije u mogućnosti kontrolirati, kontroirani su od strane računala. Putanje tih likova su unaprijed zadane kao i sve ostale radnje koje su atributi njihovih klasa. Likovi se stvaraju na zadanoj poziciji u koordinatnom sustavu i kreću se po x-osi dok im vrijednost ordinate ostaje ne promijenjena. Vrijednosti brizna kretanja likova su zadane kao i zamišljena „snaga“ koju svaki neprijateljski element ima. Kretanje tih likova izvan granica pozora nije dozvojeno te se ne određuje na temelju zbroja položaja i širine lika već su krajnje točke na x-osi zadane svojom varijablom.

PROJEKTILI:

Glavni lik ima mogućnost ispaljivanja projektila koji mogu naštetiti snazi neprijateljskih likova. Projektili su dizajnirani kao krugovi te nemaju nikakve popratne slike tj. izgled projektila na cijeloj njihovoj putanji ostaje nepromijenjen. Projektili se gibaju isključivo po x-osi a smijer i broj prođenih piksela u jedinici vremena ovisi o zadanoj brzini projektila. Broj projektila na ekranu je ograničen, ali vrlo jednostavno promjenjiv u kodu programa. Broj projektila je ograničen petljom koja ispituje broj projektila na ekranu te ne dopušta ispaljivanje novoga ukoliko se broj vidljivih projektila nije smanjio. Projektili se brišu ukoliko su dotaknuli okvir neprijatelja ili ukoliko je zbroj radijusa i pozicije projektila premašio širinu prozora.

OKVIRI:

Svaki pokretni element na prozoru ima svoj okvir koji predstavlja područje na kojem element djeluje. Okviri glavnog lika i neprijatelja imaju ublik uspravnih pravokutnika, a a okvir projektila ima oblik kružnice. Do interakcije elemenata dolazi ukoliko se dotaknu njihovi okviri. Doticanje okvira provjerava se trenutnim lokacijama okvira tj. ako se u bilo kojem trenutku okviri preklapaju. Okviri ne odgovaraju u potpunosti obliku likova budući da se slike ne poklapaju idealno pri njihovoj promjeni, najveći se pomak primjećuje pri prelasku slike za u desno sa slikom za u lijevo i obratno. Okvir projektila savršeno se poklapa sa njegovim oblikom budući da se radi o krugu.

PREKID VIDEO-IGRE:

U gornjem desnom kutu prozora nalazi se gumb za prekidanje video-igre koji je dostupan u svakome trenutku. Ukoliko je glavni lik pogođen, video-igra se kratkotrajno zamrzne kako bi igrač vidio što se dogodilo. Pri zamrzavanju igre također ju je moguće prekinuti istim postupkom, a programsko rješenje će biti objašnjeno u nastavku.

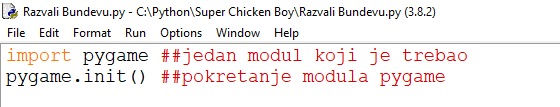
IZMJENA I NEVIDLJIVOST PROTIVNIKA:

Video-igra započinje pojavljivanjem glavnog lika i običnog neprijatelja koji su vidljivi. Glavni neprijatelj također je pozvan pri pokretanju igre no on se ne crta te njegova svojstva tda još nisu važeća. Kada neprijatelj izgubi svu svoju snagu, on se prestane osvježavati na ekranu i njegova svojstva nestaju. Ukoliko nema neprijatelja na ekranu, pojavljuje se novi na zadanoj poziciji sa jednakim atributima kao i prošli neprijatelj budući da pripadaju istoj klasi. Nakon pet poraženih neprijatelja aktivira se crtanje glavnog neprijatelja na ekranu, a njegova svosjtva postaju važeća. Nakon pobjede nad glavnim neprijateljem pojavljuje se naslov na sredini prozora koji obavještava kako je video-igra završena. Pri tome prozor se ne zatvara, čeka se naredba za prekid video-igre.

TEHNIČKA DOKUMENTACIJA

MODUL PYGAME:

Video-igra je napravljena u programskom jeziku Python, a modul koji korišten je Pygame. Nakon što je pygame modul unešen u sučelje potrebno ga je pokrenuti funkcijom „pygame.init()“.

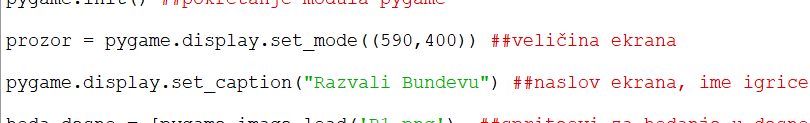


Nakon pokretanja modula može se započeti sa izradom video-igre nakon koje moramo zaustaviti pygame modul sa naredbom „pygame.quit()“.



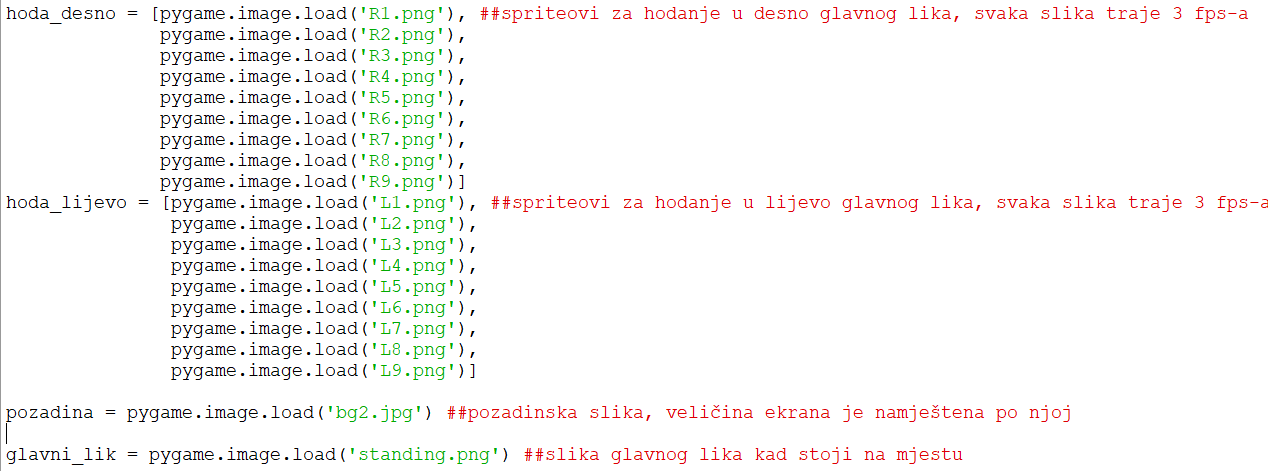
IZRADA PROZORA:

Prozor u kojem će se odvijati video igra napravljen je funkcijom „pygame.display.set\_mode()“, a potrebne su dvije viličine kao argumenti funkcije. Prvi argument je šrina prozora, a drugi je njegova visina. U gornjem lijevom kutu prozora može se dodati njegovo ime koje označava ime video-igre. Dodaje se naredbom „pygame.display.set\_caption()“ a unutar zagrade se upisuje string tj. ono što želimo napisati.



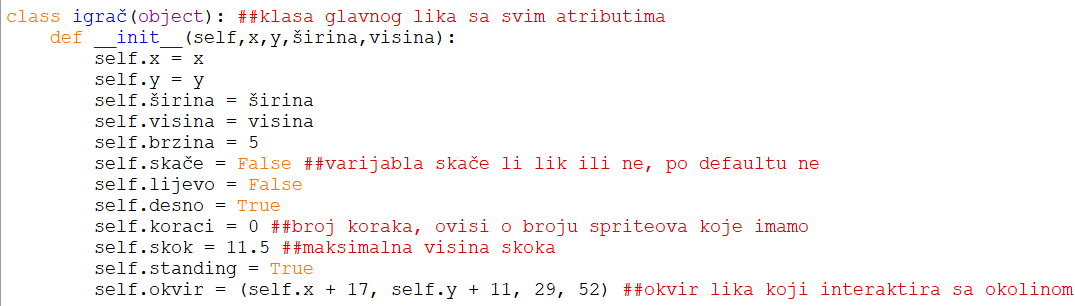
POVEZIVANJE SLIKA SA PROGRAMOM:

Likovi se kreću po prozoru izmjenom slika sa kojima su povezani te svakim korakom dolazi do promjene slike. Kako bi povezali slike sa programom, koristimo funkciju pygame.image.load te ih dodajemo u liste koje poslije koristimo kod animiranja hodanja. Posebno je dodana slika stajanja glavnoga lika te pozadinska slika.



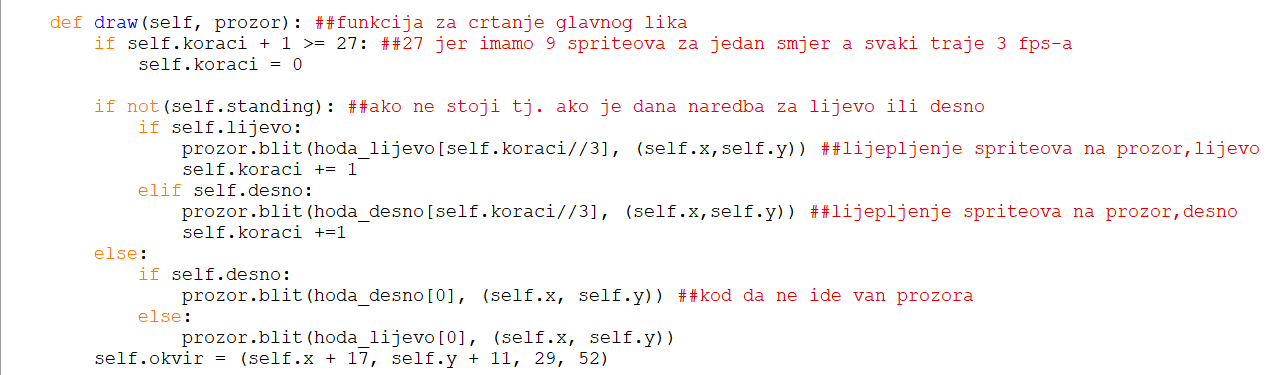
KLASA IGRAČA I NJEGOVI OSNOVNI ATRIBUTI:

Mogućnosti glavnoga lika spremljene su u klasu kako bi kod bio pregledniji a igrica bolja pri izvođenju. U klasi igrača nalaze se sve osnovne postavke koje određuju njegove proporcije, početnu poziciju, brzinu, varijable za hodanje lijevo i desno, maksimalna visina skoka, broj koraka, vrijednost ukoliko lik stoji veličina njegovog okvira.



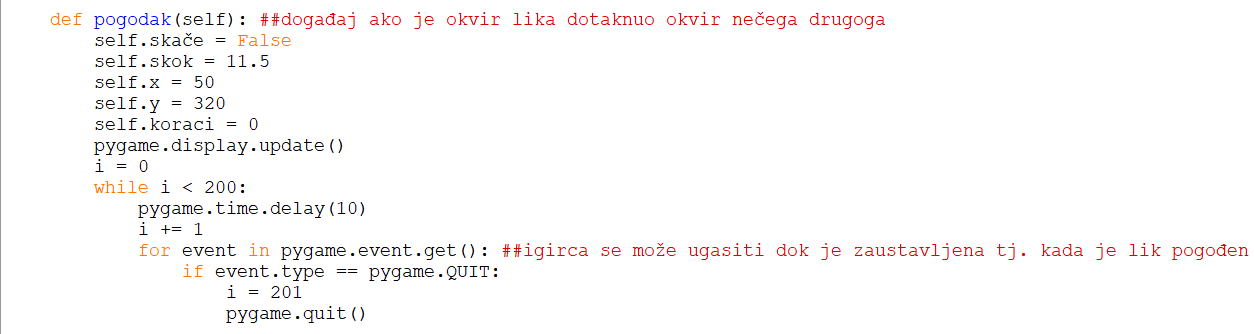
FUNKCIJA ZA CRTANJE GLAVNOGA LIKA:

Crtanje glavnoga lika odnosi se na izmjenu slika tj. spriteova ovisno o položaju u kojem se nalazi glavni lik. Svaka slika prikazuje se na prozoru tri fps-a od njih ukupno 27 te ju tada mijenja sljedeća koja prikazuje novi korak ili zaustavljanje. Koja će se slika prikazati ovisi o komandi koja je pritisnuta a u slučaju da nije niti jedna, postavlja se slika mirnoga lika koja traje do početka kretanja. Ukoliko se lik gibao, a u zadanom trenutku stoji, biti će prikazana stajaća slika za lijevo ili desno kako bi igrač znao u kojem je smjeru lik okrenut kako bi mogao ispaljivati projektile u željenom smjeru. Problem hodanja riješen je na način da ukoliko likne stoji, na ekran se lijepe slike iz liste hodanja lijevo ili desno ovisno o broju koraka. Broj koraka ovdje služi kao brojač te se koristi kao rastuća varijabla u funkciji rednog broja elementa u listi. Ukolio broj koraka prijeđe zadani fps od 27, on se resetira na početnu vrijednost koja iznosi nula te proces lijepljenja slika kreće od početka liste. Rješenje problema da lik ne izlazi iz prozora objašnjeno je u mehanikama video-igre.



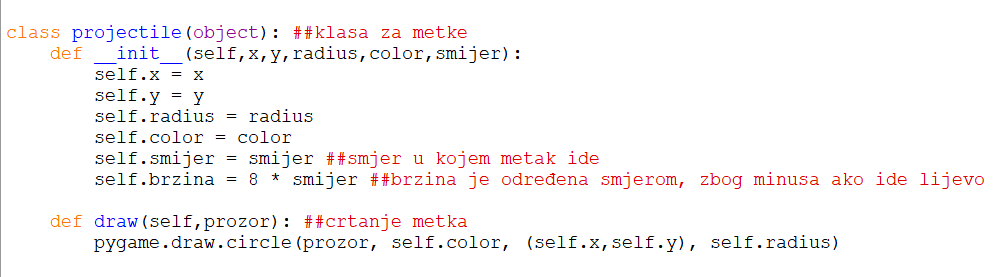
FUNKCIJA ZA INTERAKCIJU OKVIRA GLAVNOGA LIKA:

Ukoliko se okviri glavnoga lika i ostalih likova međusobno preklapaju, određeno je da se taj događaj tretira kao smrt glavnoga lika koji se vraća na početnu poziciju. Kada je lik dotaknut, program se kratkotrajno zamrzava kako bi igrač dobio uvid što se dogodilo. To je rješeno na način da se uvodi novi brojač koji dopušta pokretanje igre tek nakon što dođe do vrijednosti od 200. Za vrijeme zamrzavanja igre svejedno je moguće zatvoriti program tako da se pritiskom znaka X u kutu ekrana brojač odmah pomiče na 201 te odmrzava program. Kada je program ponovno u pokretu, on je u stanju izvršiti naredbu za zatvranje prozora.



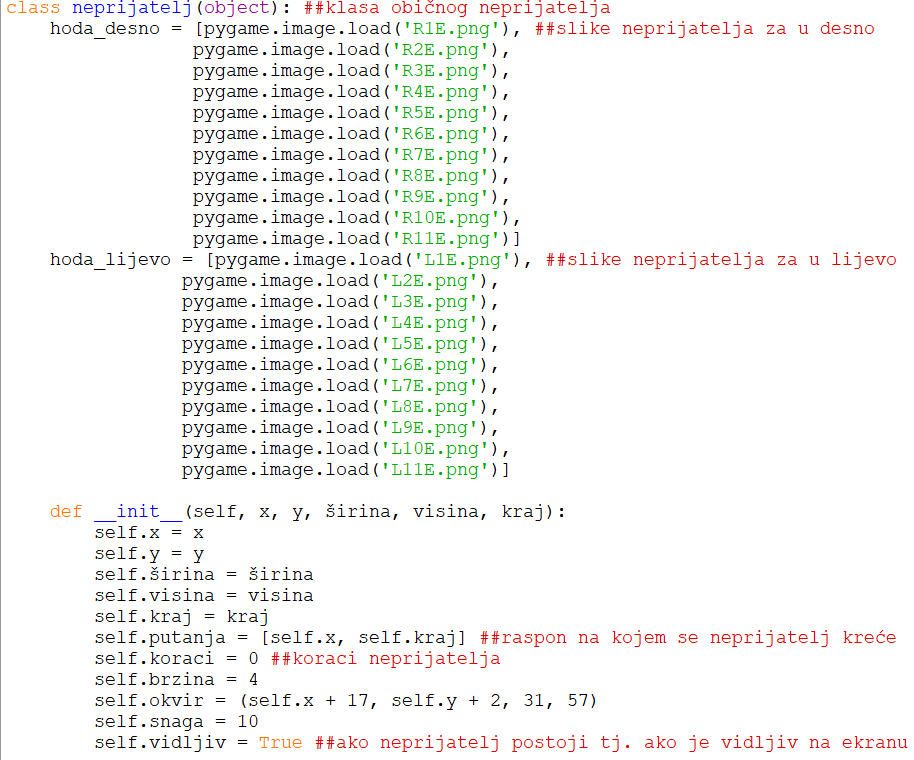
KLASA PROJEKTILA:

Glavni lik može ispaljivati projektile koji su definirani u svojoj klasi zbog preglednosti koda. Projektili su određeni početnim pozicijama na x i y-osi, radijusom, bojom, smjerom i brzinom. Ovdje nam je potrebna varijabla smjer budući da nemamo zasebne komande za ispaljivanje lijevo i desno nego to ovisi o smjeru u kojem gleda glavni lik. Smjer je svojim iznosom uvjek jednak 1, no važan je njegov predznak koji određuje kretanje projektila lijevo ili desno po x-osi. Brzina projektila stavljena je na iznos od 8 kako bi brzina metka bila veća od brzine kretanja ostalih elemenata na ekranu. Crtanje metka obavljeno je pozivanjem funkcije „pygame.draw.circle()“, a kao argumente potrebno je navesti površinu na kojoj će se prikazati, boju, lokaciju i radijus, budući da je projektil oblika kruga.



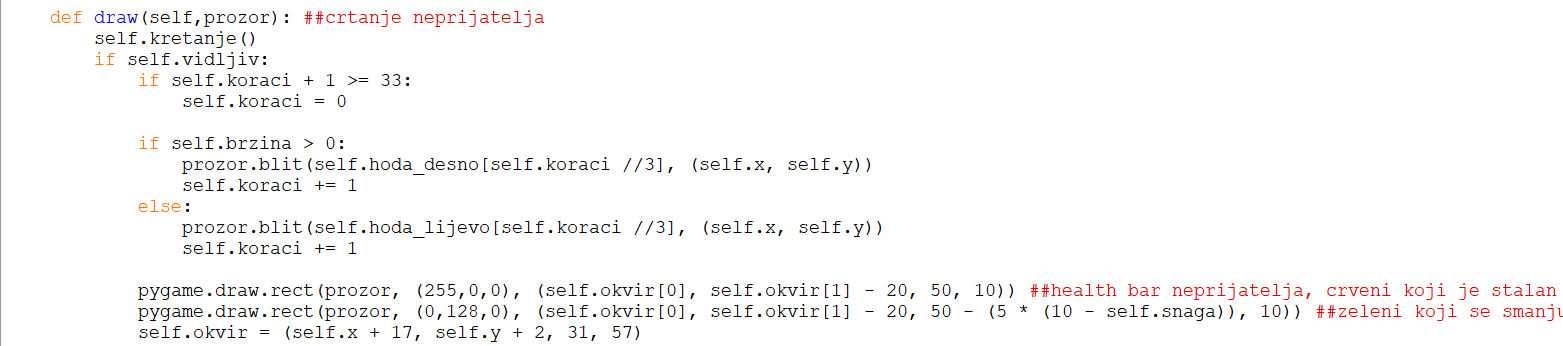
KLASA NEPRIJATELJA:

Klasa neprijatelja ima jednak broj slika za pomicanje u lijevo i u desno kao i glavni lik, stoga je također bilo moguće postaviti trajanje svake slike na 3 fps-a kao i kog glavnog lika. Jedina je razlika što neprjatelj nema sliku kada stoji u mjestu na početku igre budući da je konstantno u pokretu od jednoga kraja prozora prema drugome. Što se tiče neprijateljevih atributa neki su identični kao kod glavnoga lika kao na primjer njegova širina i visina koje su određene formatom slike.Nova varijabla je snaga neprijatelja čija je maksimalna vrijednost postavljena na 10 te će ona smanjivati ovisno o broju metaka koji dotaknu okvir neprijatelja. Sljedeća novina je varijabla „vidljiv“ koja, ukoliko je točna, dopušta crtanje neprijatelja na ekranu te omogućuje sve njegove funkcije koje interaktiraju sa okolinom. Varijabla „vidljiv“ postaje netočna kada neprijatelj izgubi svu snagu te omogućuje pojavljivanje novoga neprijatelja. Ukoliko bi varijabla „vidljiv“ bila stalno točna sve bi interaktivne funkcije neprijatelja bile konstantno aktivne što bi dovodilo do sudara neprijatelja sa glavnim likom iako je neprijatelj nevidljiv.



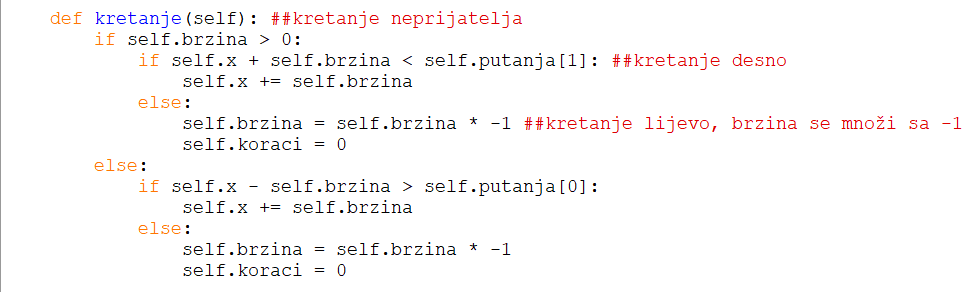
FUNKCIJA CRTANJA NEPRIJATELJA:

Funkcija crtanja neprijatelja radi jednako kao kod glavnoga lika, razlika je u implementaciji „health bara“ koji nam pokazuje koliko snage ima neprijatelj prije nego što nestane. Prikaz snage neprijatelja napravljen je pomoću dva pravokutnika koji se u početku preklapaju. Donji pravokutnik je crvene boje i on je stalan tj. prikazuje maksimalnu vrijednost snage. Gornji pravokutnik je zelene bolje koji je na početku jednake veličine kao i crveni, no njegova se površina smanjuje tako da se skraćuje njegova širina. To je postignuto na način da ukoliko okvir projektila interaktira sa okvirom neprijatelja, smanjuje se varijabla „snaga“. Razlika broja 10 i snage množi se faktorom 5 što se na kraju oduzima od širine zelenog pravokutnika. Kada širina zelenog pravokutnika postane jednaka nuli, varijabla „vidljiv“ postaje netočna.



FUNKCIJA KRETANJA NEPRIJATELJA:

Neprijatelja, za razliku od glavnoga lika, kontrolira računalo. Neprijatelj je stalno u pokretu te se giba od početne točke (varijabla x) do krajnje (varijabla kraj). Kako bi se mogao kretati u oba smjera, vrijednosti brzine moraju biti ponekad pozitivne (kretanje desno), ponekad negativne (kretanje lijevo). Ukoliko je neprijatelj došao do kraja svoje putanje, vrijednost brzine mora promjeniti predznak kako bi se neprijatelj kretao u drugom smjeru. Kada se promjeni predznak brzine, broj koraka se resetira na 0 te animacije kreću ispočetka, samo ovaj put iz druge liste.

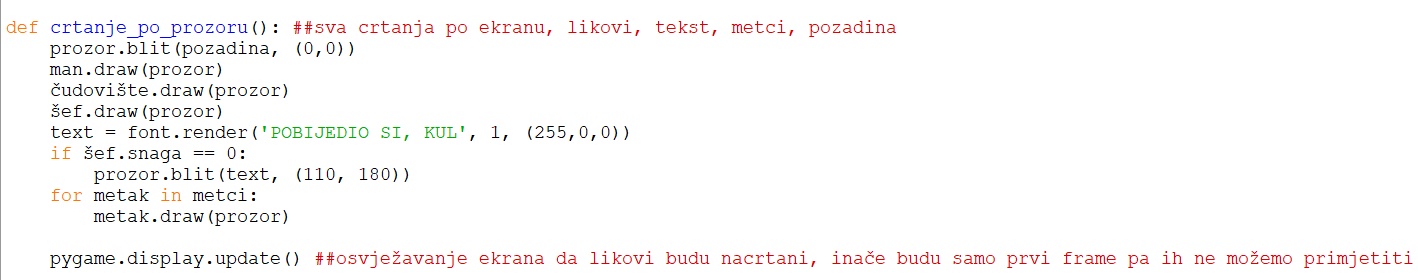


BOSS KLASA:

Boss je poslijednji neprijatelj u igrici i pojavljuje se nakon što se izmjeni 5 običnih neprijatelja. Boss je pozvan odmah pri pokretanju igrice no budući da je osnovno stanje njegove varijable „vidljivost“ stavljeno na False, on nije prisutan odmah na početku. Boss ima 9 umjesto 11 slika za kretanje u jednome smjeru te smo zbog toga morali prilagoditi maksimalan broj koraka prije nego što se on resetira na 0. Boss također ima mnogo šire pravokutnike od neprijatelja koji predstavljaju health bar, a rezultat toga je mnogo veća snaga bossa. Sve ostale vrijednosti bossa su jednake vrijednostima neprijatelja.

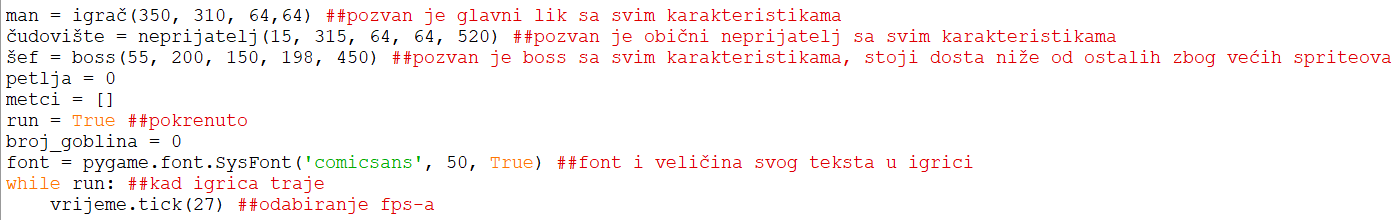
POSTAVLJANJE ELEMENATA NA PROZOR:

Nakon što izrađene klase svih elemenata te su pozvani likovi sa takvim vrijednostima, sve je potrebno postaviti na prozor. To nije potrebno raditi u zasebnoj funkciji, no ovdje je napravljeno kako bi se održala doslijednost programa. Funkcija „crtanje\_po\_prozoru()“ pozvana je pri kraju koda. Kako bi na prozor zalijepili sliku ili tekst potrebno je koristiti funkciju „.blit()“, a kako bismo nacrtali prije stvoreni element koristimo funkciju „.draw()“. Na prozoru se također pojavljuje tekst „POBIJEDIO SI, KUL!“ ukoliko je vrijednost snage bossa jednaka 0.

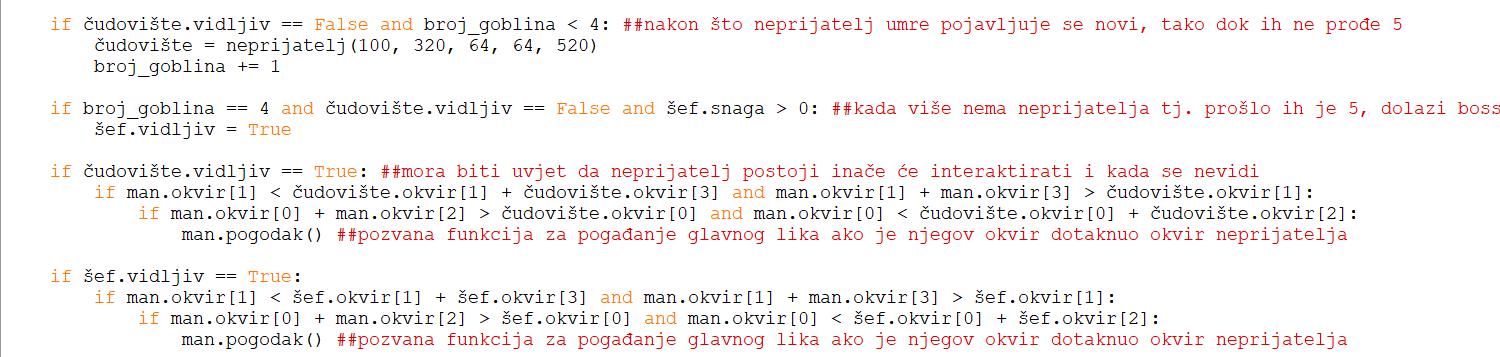


INAČICE KLASA:

U glavnom dijelu programa pozivaju se likovi te se stvara lista projektila (metci) u koju se spremaju projektili te se ispaljuju pozivanjem rednog broja elementa te liste. Varijabla broj\_goblina označava koliko je goblina bilo na ekranu te se svaki put poveća za 1 pri pojavi novog neprijatelja. „font“ je ime varijable u koju je spremljen oblik, veličina i boja svog teksta koji se pojavljuje u igri. „run“ je varijabla čije je osnovno stanje True i tako ostaje dokle god program radi, ta varijabla postaje False jedino kada se želi prekinuti izvedba programa.

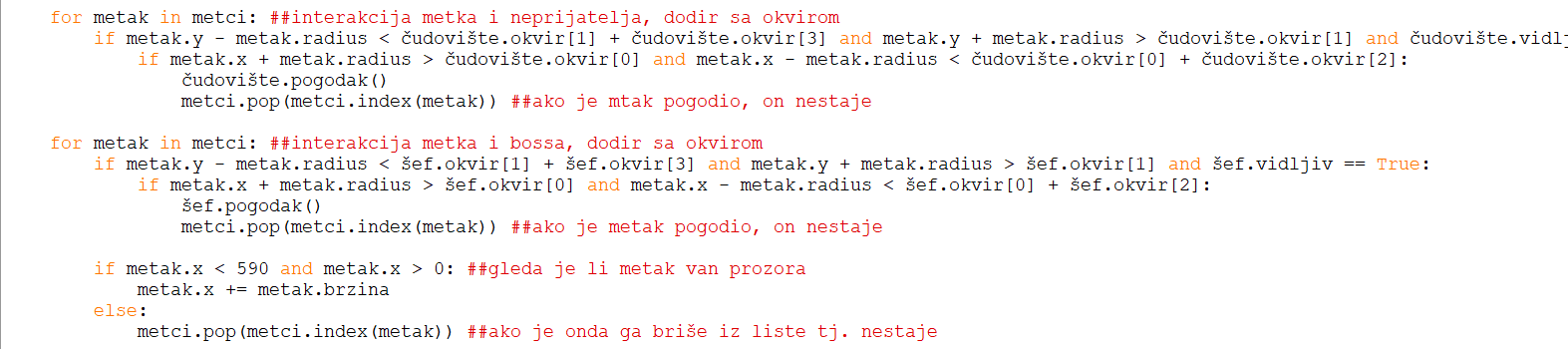


IZMJENA NEPRIJATELJA, POJAVLJIVANJE BOSSA, OKVIRI:

Neprijatelji se pojavljuju jedan za drugim dok ih ne prođe 5. To je učinjeno tako da ukoliko je neprijatelj ubijen tj. „nevidljiv“ te je brojač neprijatelja ispod 4 (budući da brojač počinje od 0), pojavljuje se poziva se identičan primjerak klase. Boss se pojavljuje ukoliko je brojač neprijatelja dosegao 4 a poslijednji je neprijatelj „nevidljiv“. Treći uvjet za pojavljivanje bossa je njegova snaga koja mora biti veća od nule jer u protivnom boss bi bio u stanju „vidljiv“ i nakon što je njegova snaga potrošena. Sljedeća dva uvjeta provjeravaju dodire okvira glavnoga lika i neprijatelja na temelju uspoređivanja x i y koordinati. Ukoliko se u bilo kojem trenutku pozicije igrača i neprijatelja preklapaju, poziva se funkcija za pogodak igrača.

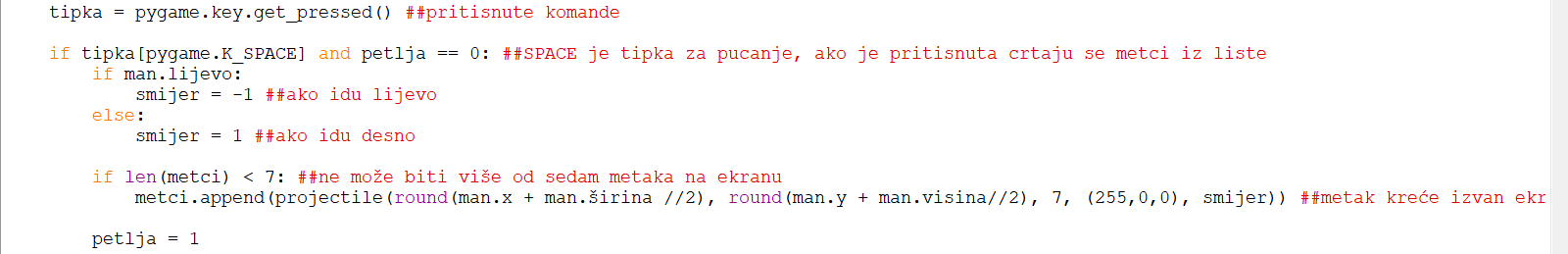
INTERAKCIJA PROJEKTILA, GRANICE PROZORA:

Provjeravanje preklapaju li se okviri projektila i neprijatelja rješeno je uspoređivanjem x i y koordinata okvira projektila sa okvirom neprijatelja. Ukoliko se dio okvira projektila nalazi unutar okvira neprijatelja, detektiran je pogodak te se poziva funkcija „pogodak“ za neprijatelja. U slučaju da je detektiran pogodak ili da je projektil došao do granice prozora, on se izbacuje iz liste projektila te se prestaje prikazivati.



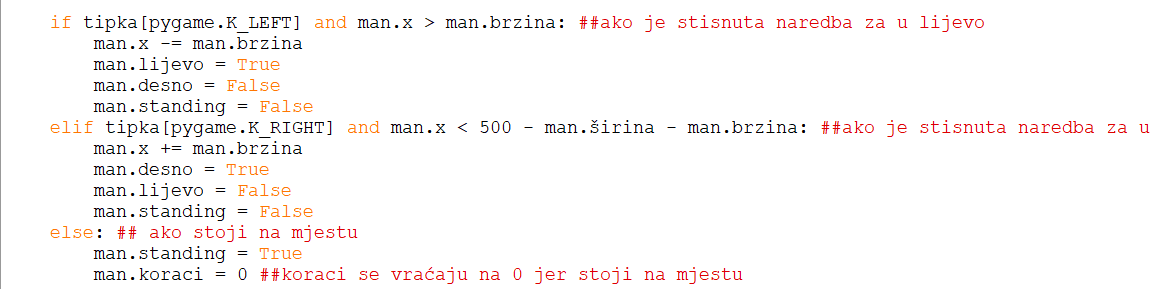
NAREDBA ZA PROJEKTILE:

U varijablu tipka spremljena je funkcija koja omogućuje reakciju elemenata na pritisak tipke. Kada je pritisnuta tipka „SPACE“ počinju se crtati projektili na ekranu. Smjer u kojem ide projektil ovisi o smjeru u kojem je ostavljen igrač te se smjer regulira množenjem sa -1. Maksimalan broj projektila postavljen je na 7 te ukoliko duljina liste dostigne 7, tipka za pucanje neće reagirati sve dok se broj projektila na ekranu ne smjanji. Početna pozicija projektila određuje se pozicijom igrača na što se dodaje širina njegova okvira sa te strane. Time je postignuto da projektili ne izlaze iz igrače nego njemu sa strane.



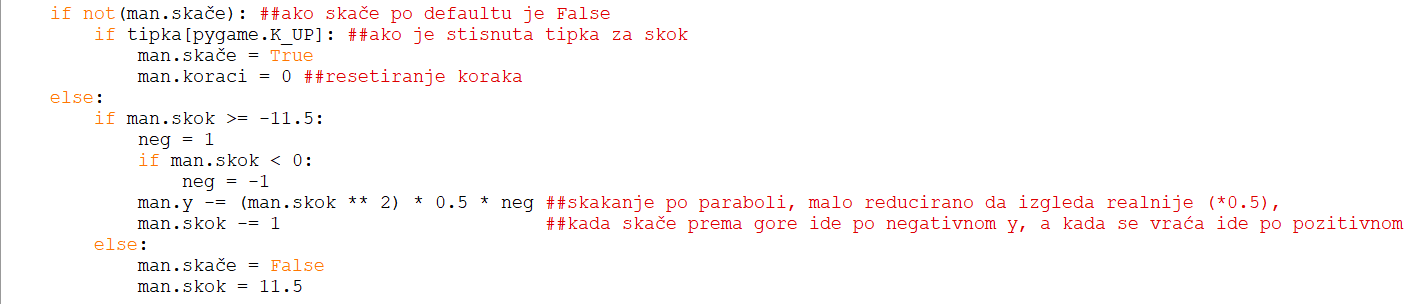
NAREDBE ZA KRETANJE LIJEVO I DESNO:

Naredbe za kretanje lika lijevo i desno mapirane su na tipke „LEFT“ i „RIGHT“ te funkcioniraju tako da sve ostale vrijednosti kretanja postanu „False“ osim odabrane naredbe koja postaje „True“. Ukoliko nije pritisnuta niti jedna tipka, glavni lik stoji, a broj koraka se resetira na 0 kako bi slike iz liste krenule od početka prilikom ponovnog hodanja.



SKAKANJE:

U klasi glavnoga lika definirane su dvije varijable koje se odnose na skakanje igrača. Varijabla „man.skače“ u osnovi je netočna te postaje točna pritiskom tipke „UP“ koja je mapirana za skok. Kada lik skoči, varijabla „man.skače“ postaje točna te ne dopušta ponovni ulazak u petlju za skakanje. Time je onemogućeno skakanje ukoliko je lik već u zraku. Varijabla „man.skok“ iznosom određuje maksimalnu visinu skoka tj. y koordinatu tjemena parabole. Lik skače po putanji parabole ukoliko je u pokretu te za svakoga skoka na početku ima najveću akceleraciju koja mjenja svoj predznak u tjemenu parabole. Igrač se giba negativnom brzinom u prvoj polovici skoka budući ide u smjeru negativne ordinate, a iznos brzine sa vremenom opada te dostiže 0 najvišoj točki. U drugom dijelu skoka brzina raste i dostiže maksimalnu vrijednost prije dodira sa tlom. Time sam pokušao izvesti simulaciju slobodnog pada, bez poklapanja akceleracije sa akceleracijom slobodnog pada.



ZATVARANJE PROGRAMA:

Na kraju je pozvana funkcija za crtanje svih elemenata na prozor koja je ranije bila definirana te završna linija koda koja zatvara modul pygame.

