**PRIRODOSLOVNO-MATEMATIČKI FAKULTET   
SVEUČILIŠTE U SPLITU**

Josipa Mrša

**Low-Poly Landscape**

Informatički projekt

**Studij:** Preddiplomski studij

**Studijska grupa:** Informatika

**Predmet**: Računalna grafika

**Ak. god.**: 2017./2018.

**Nastavnik**: Doc. dr. sc. Hrvoje Kalinić

Split, 2018

# UVOD

Cilj mog projekta bio je spojiti izradu modela okoliša u Blenderu sa funkcionalnosti PyOpenGL-a i PyGame modula. Svi prikazani modeli napravljeni su u 3D grafici, ali način na koji su u projektu poslagani daje simulaciji 2D izgled.

Također, modeli su izrađeni Low Poly tehnikom, što znači da su se za njihovu izradu koristio malen broj poligona za postizanje osnovnih prepoznatljivih oblika. Ova se tehnika najčešće koristi kod „real-time“ aplikacija (npr. igrice) kao način optimizacije performanse kod računala i konzola, za razliku od modela s velikim brojem poligona koji se često koriste kod izrade animacija u filmovima i serijama. Glavni razlog je što su igrice interaktivne, stoga je potrebno smanjiti opterećenje, dok su animacije u filmovima statične te ne iziskuju korisničku interakciju jednom kad se iscrtaju (s njima se ne upravlja niti ih se može više modificirati).

# IZRADA MODELA U BLENDERU

Blender je besplatni i open source alat za izradu 3D modela, koji podržava modeliranje, „rigging“ (montiranje armature), animaciju, iscrtavanje, kompoziciju, simulaciju i snimanje pokreta te ima interni „game engine“ te podržava i izradu igara. Svi modeli u projektu izrađeni su u Blenderu.

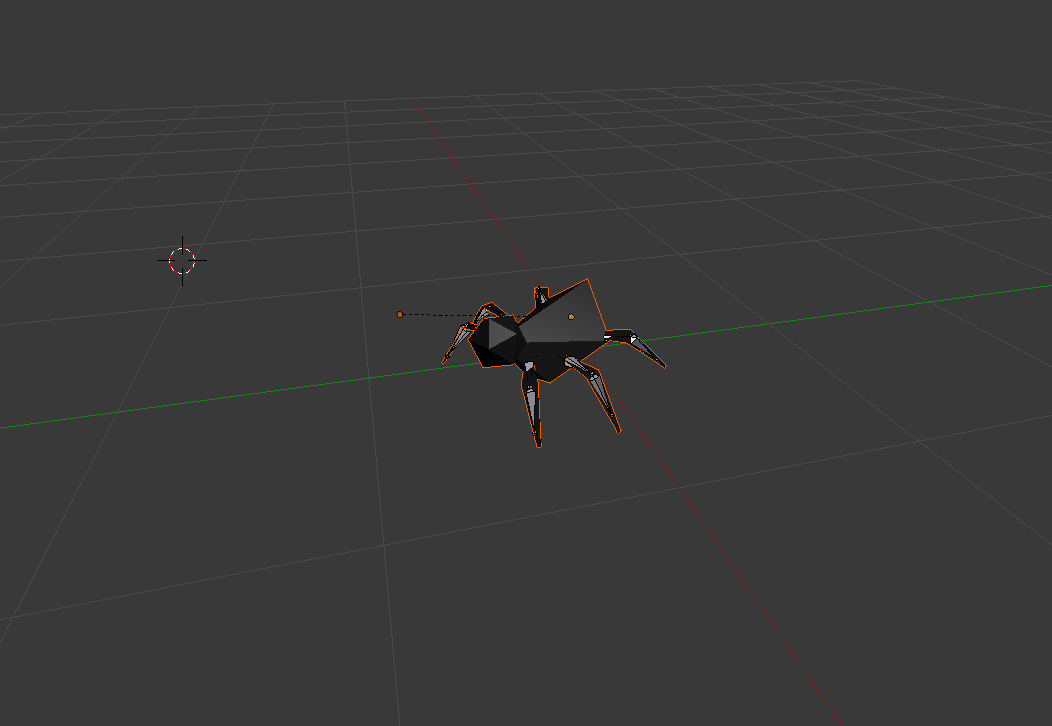
Prvi model koji sam izradila u Blenderu bio je pejzaž prateći slijedeći tutorijal:

<https://cgi.tutsplus.com/tutorials/secrets-to-creating-low-poly-illustrations-in-blender--cg-31770>

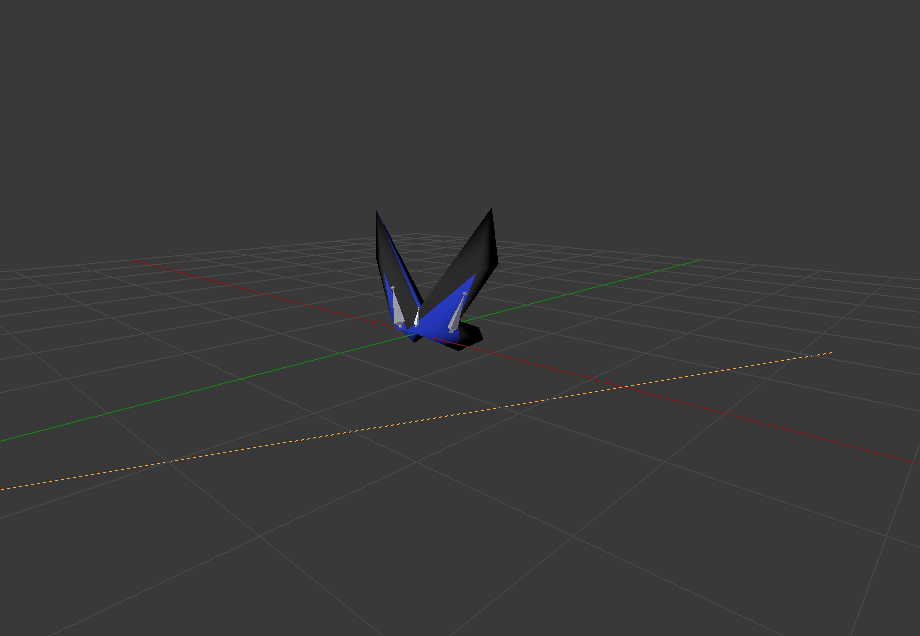


Slika - Renderirani 3D pejzaž u Blenderu

Idući model je bio model pauka, te nakon njega model leptira. Oba modela su u Blenderu bila animirana, međutim u PyOpenGL projektu nisu eksportirane njihove animacije zbog jednostavnosti. Oba modela sastavljena su od par poligona, bez neke posebne reference. Da bi se modeli mogli animirati, potrebno je bilo montirati armaturu, odnosno „kostur“ pomoću kojeg se dijelovi poligona mogu pomicati, rotirati i skalirati neovisno o ostalim dijelovima poligona. To omogućuje izradu jednostavnih, ali i kompleksnih i glatkih animacija kod modela.



Slika - Model pauka



Slika - Model leptira

Kada se svi elementi spoje, dobijemo konačni render projekta u Blenderu.



Slika - Konačni render

Jedini model koji se ne vidi u ovom renderu su zvijezde, koje se kasnije koriste samo u PyOpenGL-u. Da se zadrži prepoznatljivi izgled, površine modela su potpuno matirane te je gradivni element modela trokut, za razliku od klasičnog kvadrata.

# UPRAVLJANJE OBJEKTIMA PUTEM PYOPENGL-A I

# PYGAME-A

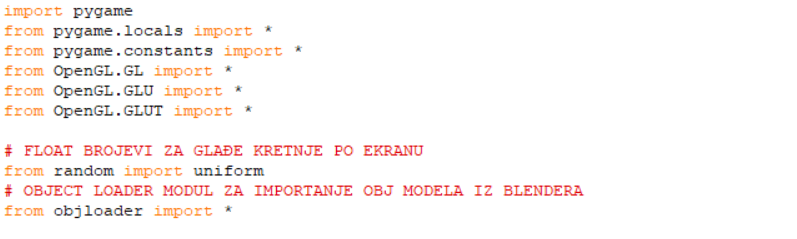
Modeli iz Blendera eksportirani su u OBJ obliku, stoga je bilo potrebno pronaći ili izraditi modul koji će importirati modele u OBJ obliku i omogućiti upravljanje modelima unutar PyOpenGL prozora.

Modul koji se koristi u ovom projektu nađen je na stranicama od PyGame-a:

<http://www.pygame.org/wiki/OBJFileLoader>

Prva datoteka sadrži sam kod koji importira sam objekt i materijal objekta. Druga datoteka sadrži kod projekta te također služi kao preglednik i okvir za upravljanje objektima.

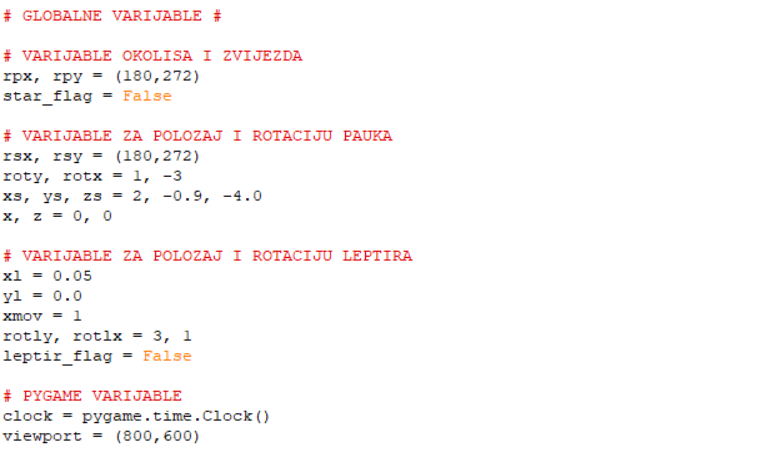
Osim OBJLoadera, korišten je i PyGame modul. U srži OBJLoader modula korišten je PyGame, no iako u samom projektu nije bilo potrebno koristiti PyGame modul, odabran je za korištenje zbog mogućnosti koje su u tom modulu malo bolje izvedene nego kod samog PyOpenGL-a.



Slika - Moduli

Osim nabrojenog, korišteni su klasični moduli za OpenGL funkcije, te funkcija uniform iz random modula, koja pronalazi nasumični decimalni broj iz zadanog ranga brojeva, čija će uloga u projektu biti kasnije objašnjena.

Također, pretvorbom Blender modela u OBJ datoteke, dobivaju se samo glavni objekti i njihovi materijali, dok posebnih efekata, kad se objekt importa u PyOpenGL neće biti.



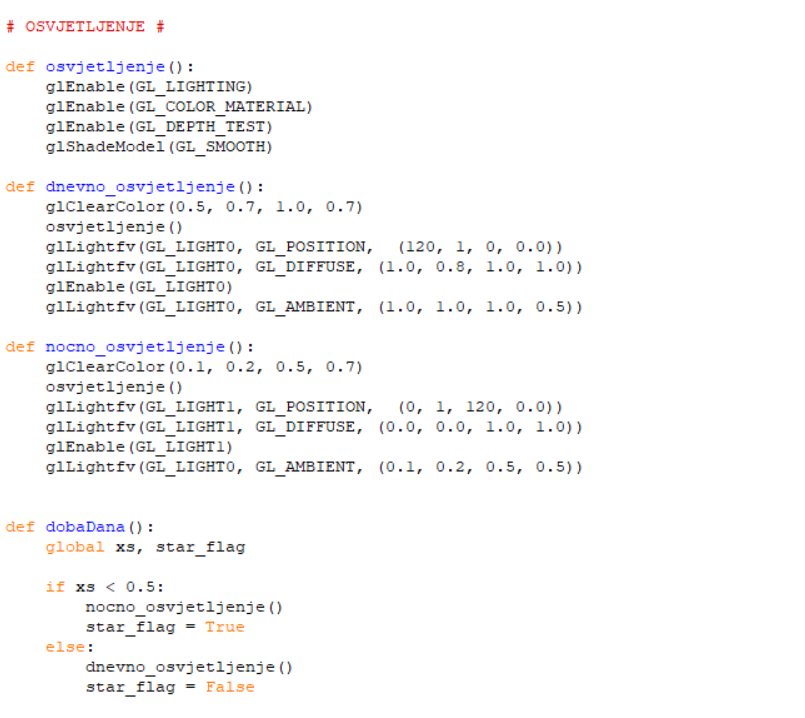
Slika - Globalne varijable

Ispod modula navedene su neke globalne varijable koje se koriste u kasnijim funkcijama, raspoređene prema objektima / funkcijama u kojima se koriste. Kod varijabla okoliša i zvijezda, rpx i rpy služe za kut rotacije okoliša, dok star\_flag varijabla služi za pojavu zvijezda na ekranu.

Varijable za objekt pauka, osim kuta rotacije, sadrže i varijable za postavljanje pauka na ekran (xs, ys, zs), kretanje samog pauka (x, z; po x i z osi) i rotaciju pauka prema koordinatnim osima (rotx, roty).

Varijable za objekt leptira sadrže varijable za postavljanje položaja (xl, yl), pomak leptira po x osi (xmov), rotaciju leptira (rotly, rotlx) i bool varijablu za pokretanje / zaustavljanje kretanja leptira (leptir\_flag).

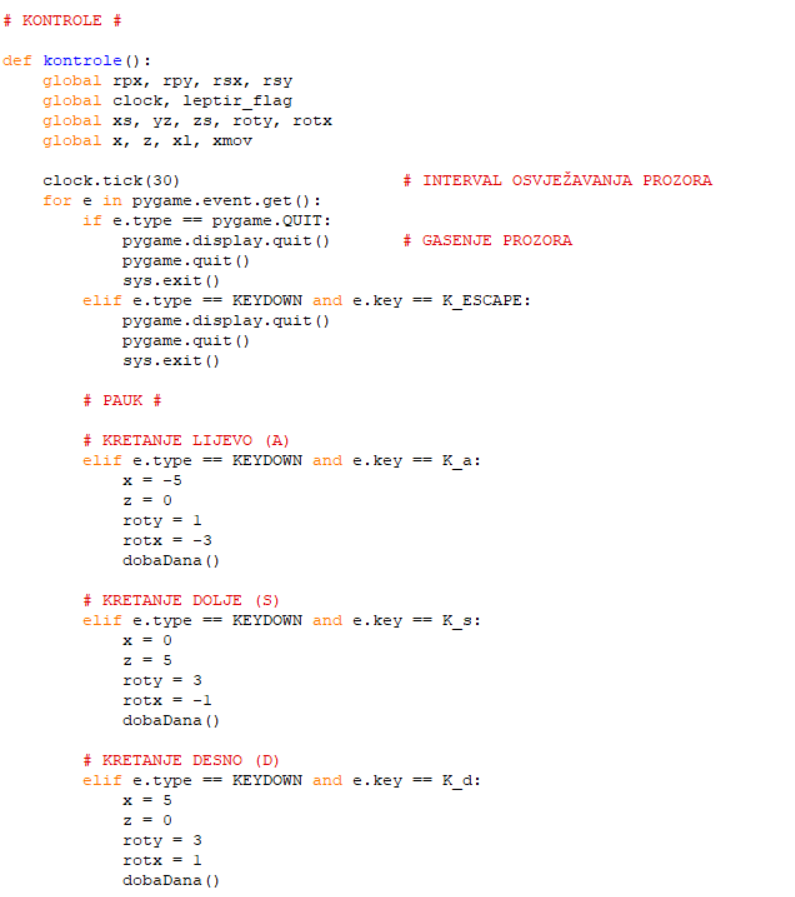
Ostale varijable tiču se PyGame prozora, gdje viewport služi za veličinu prozora (širina i visina) te interval osvježavanja prozora (clock).



Slika - Funkcije osvjetljenja

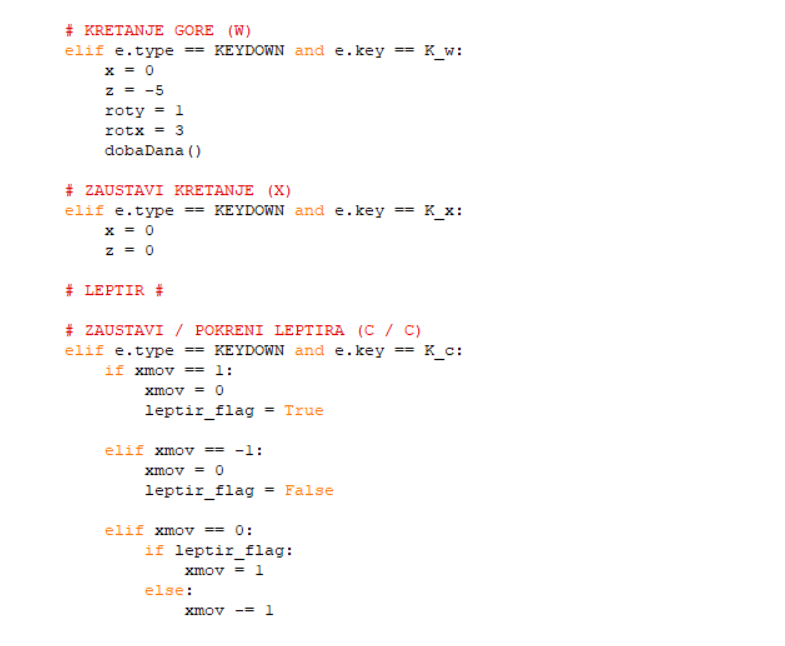
U funkcijama osvjetljenja koriste se OpenGL moduli. U glavnoj funkciji osvjetljenje postavljamo položaj svjetla, omogućavamo osvjetljenje, sjenčanje i utjecaj osvjetljenja na objekte. Za sjenčanje je preporučeno staviti parametar „GL\_SMOOTH“ s obzirom da najviše pogoduje modelima iz Blendera. Također, ne moramo imati samo jedan izvor svjetla nego više njih, kao što su ovdje GL\_LIGHT0 i GL\_LIGHT1. Ovdje su za svjetla postavljeni položaji, jačina te razine ambijentalnog i difuznog svjetla. Moguće je još postaviti i razinu reflektirane svjetlosti te fiksno (spot) osvjetljenje. Također, prigodne boje osvjetljenja i samog okoliša postavljene su za obje funkcije osvjetljenja.

Funkcija dobaDana() mijenja dan i noć s obzirom na položaj pauka na mapi. Također upravlja pojavom zvijezda na nebu kod noćnog osvjetljenja.



Slika - Upravljanje objektima

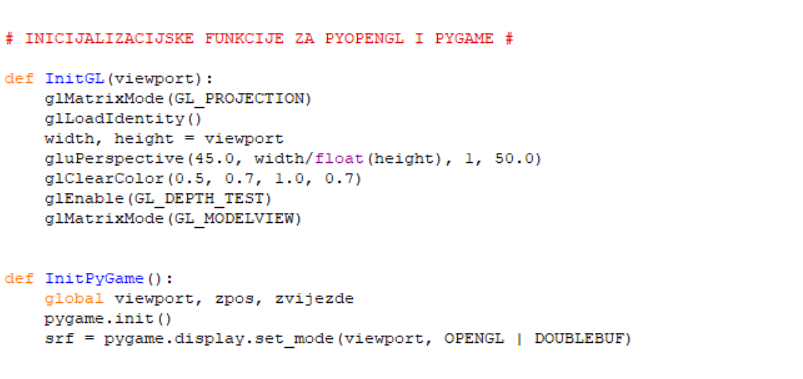
Funkcija kontrole() brine se o upravljanju likovima na mapi i samim prozorom. U PyGame modulu imamo posebne tipove događaja ovisno o tome što korisnik pritisne. Tako da za zatvaranje prozora postoji zaseban događaj PyGame.QUIT, koji omogućuje pritiskom na X potpuni izlaz iz PyGame prozora. Isto tako je izlaz iz prozora definiran pritiskom tipke ESC i tipom događaja KEYDOWN (kad je tipka pritisnuta). Tipovi događaja još mogu biti na otpust tipke (KEYUP), na pokret miša (MOUSEMOTION) i na pritisak / otpuštanje tipke miša (MOUSEBUTTONDOWN / MOUSEBUTTONDOWN). Što se tiče kretanja pauka, jednom kad korisnik pritisne tipke W-A-S-D, pauk se može kretati gore-lijevo-dolje-desno. Kontrolu kretnje gore-dolje vrši varijabla z, dok lijevo-desno vrši varijabla x. Pauk se rotira ovisno o pritisnutoj tipci. Na svaki pritisak tipke se dodatno poziva metoda dobaDana za provjeru položaja na x-osi pauka.



Slika - Upravljanje objektima

Pritiskom na tipku X, pauk se zaustavlja na mjestu.

Leptir se pomiče konstantno, a jedino upravljanje njime vrši se na tipku C, gdje možemo zaustaviti ili pokrenuti leptira. Ukoliko nam je pomak po x-osi 1, leptir se zaustavlja te se njegova zastavica postavlja na True. Zastavica nam je bitna za pamćenje smjera leptira (ako je pomak bio 1, leptir ide nadesno, dok za -1 ide nalijevo). Stoga, jednom kad pomak postane 0, znat ćemo u kojem smjeru leptir treba ići kod idućeg pritiska tipke.



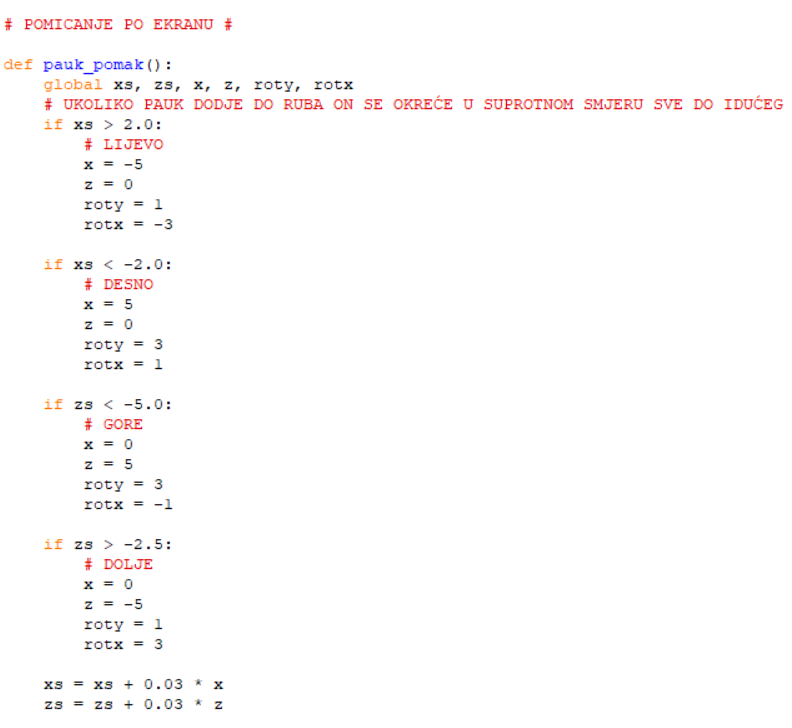
Slika - Inicijalizacijske funkcije za module

Inicijalizacijske funkcije brinu se o samom prikazu u prozoru programa. InitGL() funkcija se brine o inicijalizaciji PyOpenGL-a, a InitPyGame() o inicijalizaciji PyGame modula.

InitGL() prima parametar viewport, koji je u biti n-torka koja određiva veličinu prozora. glMatrixMode() prima 2 parametra (zasebno), koji se oba odnose na način pogleda na objekt – GL\_PROJECTION se odnosi na projekcijsku perspektivu, a GL\_MODELVIEW na direktan pogled na objekt. Zbog toga koristimo i funkciju gluPerspective() koja nam definira prostor projekcije.

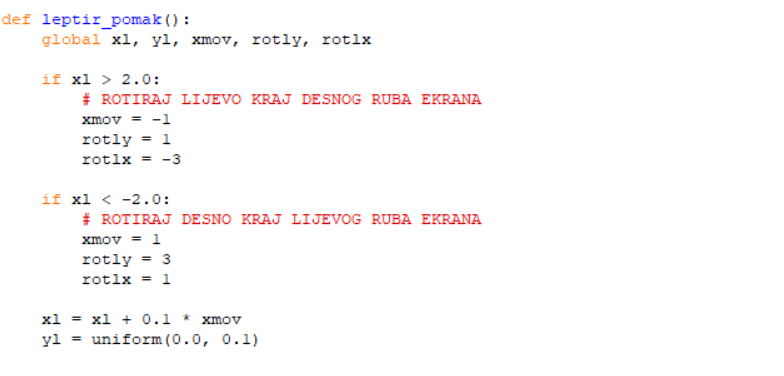
Sam prozor inicijalizira se s plavom bojom, a boja prozora se mijenja ovisno o funkcijama osvjetljenja (svjetloplavo ili tamnoplavo). U sklopu inicijalizacijske funkcije za PyOpenGL i funkcije osvjetljenja imamo GL\_DEPTH\_TEST, koji provjerava stanje depth buffera za ispravan prikaz osvjetljenja i objekata u prozoru.

Kod inicijalizacije za PyGame imamo pygame.init() koji inicijalizira sve importirane PyGame module, te kreiranje prozora sa podrškom za OpenGL s duplim bufferima.



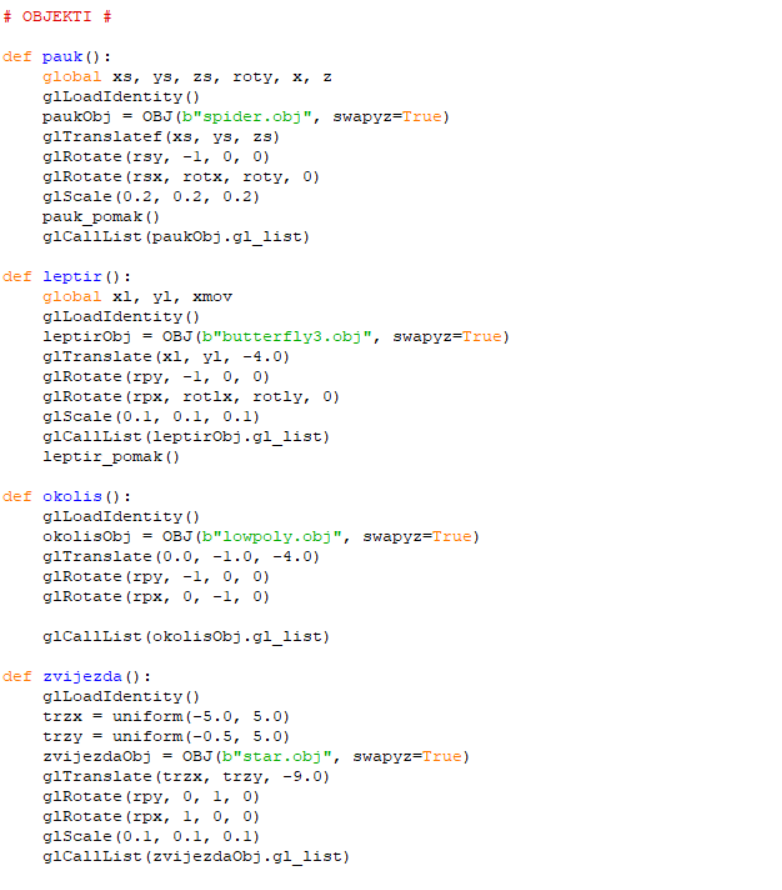
Slika - Pomicanje pauka po ekranu

Što se tiče pomicanja pauka po ekranu, korisničkim unosom on se kreće lijevo, desno, gore ili dolje. Jednom kad je tipka pritisnuta, pauk se kreće cijelim putom u određenom smjeru sve dok ne dođe do percipiranog ruba mape ili ruba ekrana. Ukoliko dođe do toga pauk se okreće i hoda u suprotnom smjeru. Na primjer, ukoliko je korisnik pritisnuo tipku W, pauk se kreće gore sve dok ne udari percipirani gornji rub (zs > -2.5). Jednom kad udari taj rub, pauk se okreće za 180 stupnjeva i nastavlja put kretajući se prema dolje, do percipiranog donjeg ruba (koji se nalazi pri kraju ekrana). Ista stvar se dešava kad udari i taj rub; opet se okreće i hoda prema gore. To će se dešavati sve dok korisnik sam ne odluči promjeniti smjer pritiskom na drugu tipku kretnje ili zaustavljanjem pauka. Stvarne koordinate po x-osi i z-osi se ažuriraju putem parametara x i z koji označavaju pomak po ekranu.



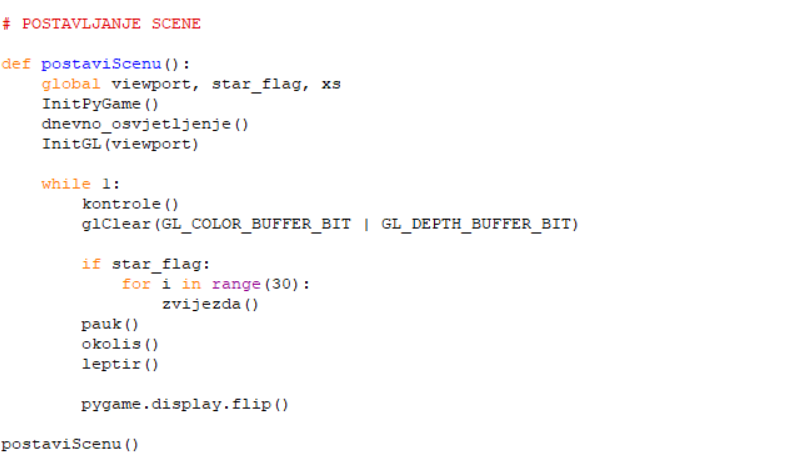
Slika - Pomicanje leptira po ekranu

Pomak leptira je sličnog principa kao i kod pauka, samo što korisnik ne upravlja smjerom leptira, nego samo njegovim zaustavljanjem i pokretanjem (opisano u kontrolama). Leptir ide u smjeru lijevo-desno, također se rotira ukoliko dođe do ruba ekrana i kreće u suprotnom smjeru. Razlika kod leptira je nadodana promjena y-koordinate, koja simulira animaciju kretnje leptira. Svakim njegovim pokretom nasumično se mijenja vrijednost njegove y-koordinate, simulirajući realističniji let leptira. Da izgleda realističnije, korišten je malen rang za funkciju generiranja nasumičnih decimalnih brojeva.



Slika - Importiranje objekata u PyOpenGL

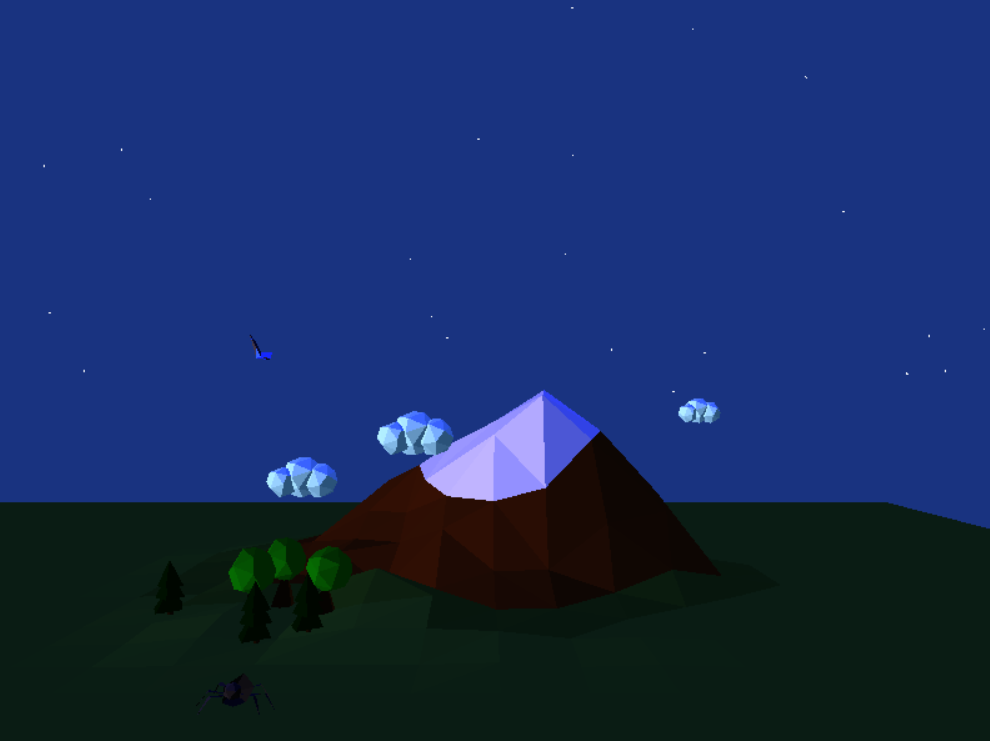
Ove funkcije importiraju objekte koji su korišteni u projektu. Tu se uglavnom postavljaju položaj, rotacija i skaliranje samih objekata. Kod importiranja zvijezde određene su nasumične koordinate (ponovno, pomoću decimalnih brojeva) s obzirom da nećemo imati samo jednu zvijezdu na nebu nego više njih. Funkcija glCallList() poziva objekt koji se kasnije učitava na ekran.



Slika - Postavljanje scene

Pri kraju koda imamo funkciju postavljanja scene, gdje se pozivaju inicijalizacijske funkcije, bitne globalne varijable i funkcija dnevnog osvjetljenja. Unutar petlje pozivamo naše objekte te pozivamo objekt zvijezde više puta da simuliramo izgled noćnog neba.

Funkcija pygame.display.flip() ažurira sadržaj prozora. S obzirom da koristimo PyGame u kombinaciji sa PyOpenGL-om, ova funkcija oponaša istu funkciju kao glSwapBuffers().



Slika - Noćni prizor