**PRIRODOSLOVNO-MATEMATIČKI FAKULTET   
SVEUČILIŠTE U SPLITU**

ZAVRŠNI INFORMATIČKI PROJEKT

**Interaktivna animacija u Blenderu - Crvenkapica**

Josipa Mrša

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

Split, rujan 2019

Sadržaj

[1 Uvod 3](#_Toc19812602)

[2 Pokretanje projekta 4](#_Toc19812603)

[2.1 Instalacija okruženja (Blender) 4](#_Toc19812604)

[2.2 Otvaranje projekta u Blenderu 4](#_Toc19812605)

[3 Struktura projekta 5](#_Toc19812606)

[3.1 Likovi u projektu 6](#_Toc19812607)

[4 Korištenje Blender Game Enginea 7](#_Toc19812608)

[4.1 Položaj i kretanje lika 7](#_Toc19812609)

[4.2 Radnje lika 8](#_Toc19812610)

[4.3 Programske skripte 9](#_Toc19812611)

[4.3.1 Korištene programske biblioteke 9](#_Toc19812612)

[4.3.2 Lokalne i globalne koordinate objekta u svijetu 10](#_Toc19812613)

[4.3.3 Rad sa scenom i njenim objektima 10](#_Toc19812614)

[4.3.4 Upravljanje kamerom u projektu 11](#_Toc19812615)

[4.3.5 Upravljanje položajem objekta na sceni 13](#_Toc19812616)

[5 Zaključak 15](#_Toc19812617)

# Uvod

Cilj ovog projekta bio je pokazati mogućnosti Blendera kao alata za izradu interaktivnog sadržaja, odnosno pogonskog sklopa za izradu videoigrica, uz njegove mogućnosti 3D modeliranja i animacije.

Blender je besplatno te *open-source* okruženje za 3D modeliranje, koje sadrži kompletan 3D cjevovod za izradu animacije, simulacija, praćenje pokreta (engl. *motion tracking*), obradu videozapisa te izradu računalnih igara. Moguće ga je koristiti i na Windows i na Linux platformama, te je dostupan u 32-bitnoj i 64-bitnoj verziji.[[1]](#footnote-1) U ovom projektu korištena je inačica Blendera 2.79.

Blender spada u programe umjetničkog 3D modeliranja, koji se najčešće koriste za razne umjetničke industrije kao filmska industrija i industrija video-igrica, za razliku od programa za oblikovanje pomoću računala (engl. *CAD software*) koji se najčešće koriste kod izrade vjernih modela dijelova, alata i raznih arhitektonskih projekata te čiji je krajnji rezultat 2D crtež ili 3D model kojeg je kasnije moguće 3D isprintati. Pod primjerom takvog programa mogu se navesti SolidWorks ili Autodesk Inventor

# Pokretanje projekta

Da bi se projekt mogao pokrenuti, potrebno je na računalo skinuti Blender verzije 2.79., u 64-bitnoj verziji.

## 

## Instalacija okruženja (Blender)

Blender je dostupan na Windows, macOS i Linux sustavima. Potrebno je uvijek provjeriti da su grafički procesori ažurirani te da postoji podrška za OpenGL. Instalacija je vrlo jednostavna – potrebno je ući na Blenderove službene stranice, te kliknuti na sekciju za preuzimanja (engl. *Download*). Tu se može naći najnovija verzija Blendera (trenutno 2.80), međutim ispod dugmeta za preuzimanje te verzije postoji mali link na stranicu, koja vodi na sekciju gdje se može preuzeti Blender 2.79. Klikom na prvi link u tekstu dolazi se do direktorija sa raznim verzijama – potrebno je odabrati verziju za željeni operacijski sustav, te odabrati 32-bitnu ili 64-bitnu verziju, ovisno o operacijskom sustavu.

Odabirom verzije preuzima se sažeta datoteka sa svim što je potrebno za rad Blendera na računalu. Kad se datoteka u potpunosti preuzme, potrebno ju je raspakirati na proizvoljno mjesto na računalu, te naći izvršnu datoteku programa. Nije potrebna nekakva kompleksnija instalacija, već se program odmah pokreće te je spreman za rad.

## Otvaranje projekta u Blenderu

Projekt se treba preuzeti, te spremiti na prikladno mjesto na računalu. Kad se Blender pokrene, iz izbornika „Datoteka“ (engl. *File*), treba odabrati opciju „Otvori…“ (engl. *Open…*) te pronaći datoteku projekta (s nastavkom .blend). Otvara se projekt s postavkama koje je za projekt ili okruženje definirao zadnji korisnik. Da bi se projekt otvorio sa zadanim postavkama okruženja, potrebno je odznačiti opciju „Učitaj UI“ (engl. *Load UI*) prije nego se otvori u Blenderu.

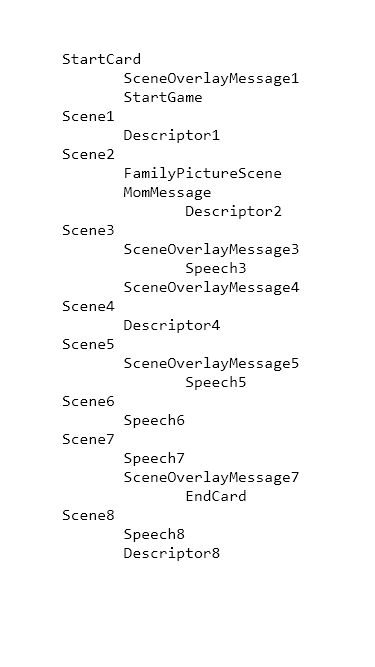
Jednom kad je projekt otvoren u Blender okruženju, u gornjoj traci alata obavezno pogonski sklop prebaciti u „Blender Game“ ako on nije već odabran. Kad je sve postavljeno, prebaciti trenutnu aktivnu scenu na scenu imena StartCard i na tipkovnici pritisnuti tipku P za pokretanje projekta.



Slika 2.1 Postavke radnog okruženja za projekt

# Struktura projekta

Projekt je podijeljen na 8 logičkih dijelova, odnosno osam cjelovitih razina ili scena, od kojih neke imaju pomoćne scene ili međuscene vezane uz njih. U Blenderu verzije 2.79 trenutno je moguće grupirati modele u scene, ali nije moguće grupirati te scene u logičke grupe, stoga na slici moguće je vidjeti kako bi grupacija modula približno izgledala:



Slika 3.1 Logička struktura projekta

Početna scena (StartCard) služi kao uvodna scena za preostalih osam glavnih. Scene naziva SceneOverlayMessage su scene koje sadrže kopije scena tog rednog broja, nisu interaktivne te najčešće služe kao okvir za odvijanje dijaloga, naracije ili animacije likova. Scene naziva Descriptor scene su koje sadrže opise, pitanja i zagonetke, odnosno okviri su za interaktivni sadržaj. Scene naziva Speech sadrže animacije dijaloga između likova scene, a ostale scene različitih naziva (MomMessage i FamilyPictureScene) sadrže informacije ili o glavnom zadatku lika, ili detaljnije zadatke. Projekt se pokreće od početne scene.

## Likovi u projektu

Projekt je napravljen kao interaktivna animacija poznate dječje bajke Crvenkapica. Stoga, likovi kojima se u projektu može upravljati su lik Crvenkapice, a na kasnijim razinama, lik lovca. S ostalim ključnim likovima, koji uključuju antagonista priče (vuk), lik bake, itd., nije moguće upravljati, ali je, pomoću lika s kojim se upravlja, moguće ostvariti interakciju s njima. Glavni likovi i neki sporedni likovi imaju svoju pripadnu animaciju, koja je ostvarena ili korištenjem kostura (engl. *Rigging*) ili samo pomoću transformiranja lokacije, rotacije i veličine lika.

# Korištenje Blender Game Enginea

U gornjem lijevom dijelu trake za odabir načina rada, za radno okruženje odabire se radno okruženje logike igre (engl. *Game Logic*), gdje je moguće vidjeti detaljnu strukturu projekta, te strukturu svake scene u projektu. Naravno, ako je potrebno napraviti i pregledati unaprijed promjene na projektu, obavezno je potrebno prebaciti pogonski sklop na Blender Game pogonski sklop, koji služi za rad s interaktivnim projektima.

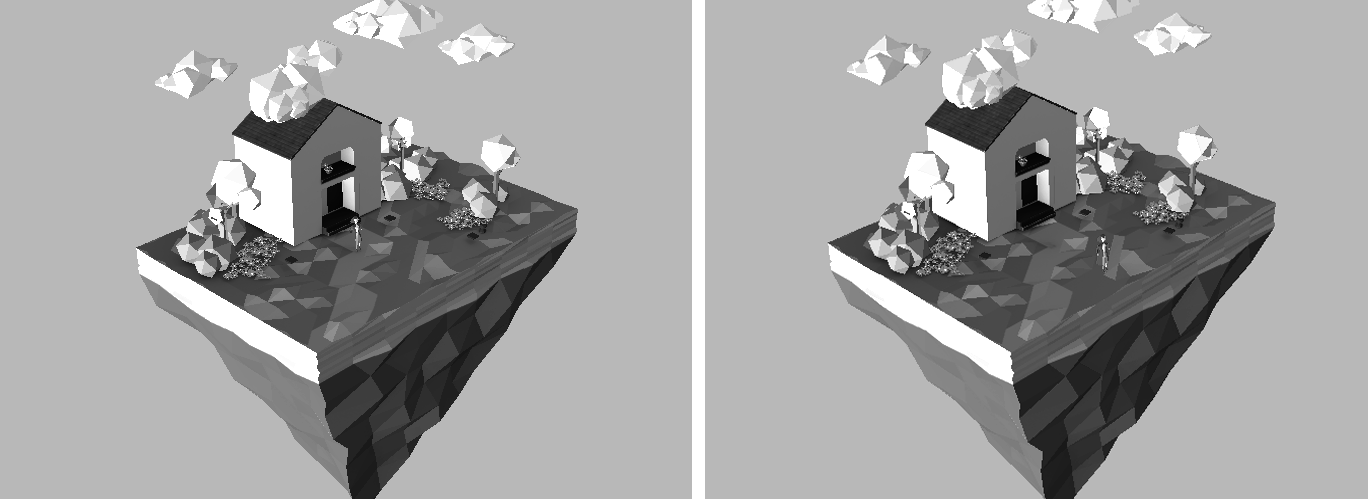
Strukture koje možemo vidjeti za neke objekte scene sastoje se od logičkih blokova. Logički blokovi su „gotovi“ blokovi koda koji se dalje mogu povezivati u kompleksne strukture, a sve u svrhu kraćenja posla pisanja skripti. Oni modeliraju neka jednostavna ponašanja, primjerice kretanje ili sudaranja objekata na sceni. Imamo tri glavna tipa logičkih blokova:

* Senzori (engl. *Sensors*) – primaju signale iz scene, koje prosljeđuju blokovima povezanim s njima
* Kontroleri (engl. *Controllers*) – upravljaju signalima između senzora i pokretačkih blokova
* Pokretači (engl. *Actuators*) – obično modeliraju neku radnju koja se treba izvršiti na primljeni signal

U ovom projektu logički blokovi su uglavnom korišteni za modeliranja kretnji likova, mogućih radnji s likovima, komunikacije među likovima, te kod sudara likova sa drugim objektima scene. Isto tako, koriste se i za pozivanje i pokretanje svih postojećih skripti.

## Položaj i kretanje lika

Položaj je određen koordinatama lokacije i rotacije, definiranim kod spuštanja lika na scenu. Lik se u tom slučaju može pomaknuti bilo gdje na sceni, te tamo gdje se nalazi će se nalaziti i pri pokretanju igre. Međutim, ukoliko se pomakne van terena, te pokrene igra, lik se smješta na najbližu moguću točku prije definirane granice kretanja. Svaki lik kojim se može upravljati za svaku scenu ima točno određeno polje kretanja – preko logičkih blokova koje definiraju granice. Granice su određene prema xy ravnini.



Slika 4.1 Položaj lika na sceni

Kretanje lika modelirano je u potpunosti preko logičkih blokova. Pritiskom tipke W na tipkovnici, lik se pomiče unaprijed. Pritiskom tipke A, lik se rotira ulijevo, a pritiskom tipke D, lik se rotira udesno. Pritiskom na tipku razmaka, lik će skakati. Za svaku od pritisnutih tipki W, A ili D se pokreće i animacija hoda lika.

## Radnje lika

Osim kretanja lika, on može imati i neke druge kontrole. Ovisno o tome s kojim objektima se sudara, otvara se mogućnost korištenja tipki S, K, M ili X. Tipka S služi za skupljanje ili uzimanje ključnih predmeta, te se igraču prikazuju upute ukoliko ima mogućnost koristiti ovu tipku. Tipka K služi za prebacivanje na iduću razinu ili gašenje međuscena, tamo gdje se igraču nudi ova opcija. M i X tipke služe za otkrivanje i sakrivanje glavnog zadatka ove igrice, da se igrača podsjeti što mu je zadatak. Taj zadatak moguće je pregledavati od druge do četvrte scene.

## Programske skripte

U ovom projektu postoje dvije skripte, odnosno dva tipa skripte, od kojih jedna upravlja položajem kamere, a druga upravlja položajima objekata na sceni. Za svaku skriptu koja upravlja položajem neke scene opisa, definirana je zasebna skripta koja je strukturno potpuno ista, ali se koriste drugi parametri i objekti, prilagođeni sceni na kojoj se ta skripta izvršava.

### Korištene programske biblioteke

U ovom projektu korištene programske biblioteke uključuju:

* *Blender Game Engine* modul (BGE)
* Modul za rad s fontovima (BLF)
* Modul za kompleksne matematičke operacije

BGE je programska biblioteka koja sadrži sve potrebno za rad sa Blenderovim pogonskim sklopom za igre. Ona je razbijena na nekoliko podgrupa sa pripadajućim klasama i funkcijama, a neke od najčešće korištenih uključuju tipove, logiku, iscrtavanje, događaji, itd. U ovom projektu korišten je samo modul logike. Taj modul bavi se upravljanjem logičkim blokovima, objektima i scenama preko skripte, odnosno preko njega možemo pristupiti definiranim logičkim blokovima i mijenjati ih preko skripte.

BLF je programska biblioteka koja sadrži sve funkcije vezane uz manipulaciju teksta i fontova u projektu

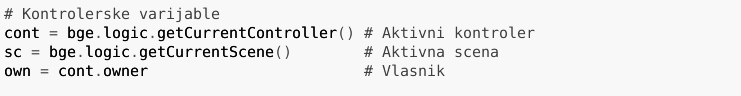
Modul za kompleksne matematičke operacije postoji već kao programska biblioteka u samom Pythonu.

### Lokalne i globalne koordinate objekta u svijetu

Kod upravljanja objektom javljaju se dva bitna tipa koordinata. Lokalne koordinate su statične koordinate, koje se ne mijenjaju pomicanjem, rotiranjem ili skaliranjem objekta kojim se upravlja. Globalne, odnosno koordinate svijeta, su koordinate koje pri pokretanju projekta odgovaraju lokalnim koordinatama, ali pomicanjem, rotacijom ili skaliranjem lika se mijenjaju. Koriste se kod modeliranja kretanja lika ili kod objekta koji treba pratiti kretanje objekta na sceni.

### Rad sa scenom i njenim objektima

Kako bismo radili sa scenom i njenim objektima, ono što je potrebno najprije dohvatiti su trenutno aktivni kontroler te trenutnu aktivnu scenu. Trenutno aktivni kontroler je logički blok kontrolera preko kojeg će se pozivati ta Python skripta, a trenutno aktivna scena je ona scena u kojoj se nalazi objekt koji sadrži trenutno aktivni kontroler. Vlasnik kontrole je upravo objekt koji je vezan uz taj kontroler.



Slika 4.2 Upravljačke varijable

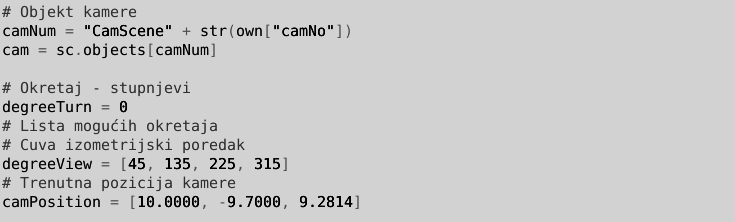
### Upravljanje kamerom u projektu

Kamere u svakoj sceni imaju dva bitna svojstva. Prvo svojstvo je redni broj kamere, a drugo je svojstvo također broj koji označava poziciju kamere u igri. Mogućih položaja kamere ima četiri, a svaki označava pogled iz jednog od kutova određenih terenom, da se očuva izometrija scene.

Kako kamera ne prati, odnosno nije vezana za neki objekt u sceni, koristi se promjena lokalnog položaja za promjene položaja na sceni.

Skripta koja pomiče kameru hvata signal iz senzora za dvije tipke: strelicu lijevo, te strelicu desno koje pokreću kameru ulijevo, odnosno udesno od prethodnog položaja.

Varijable vezane uz ovu skriptu vidljive su sa donje slike:

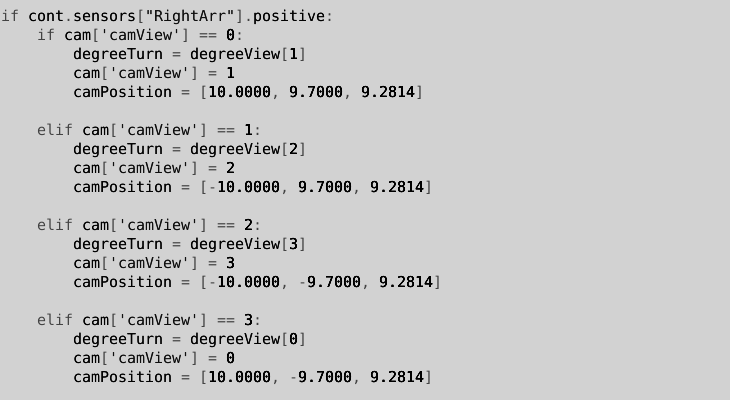


Slika 4.3 Svojstva i varijable za kameru

Dakle, svaka kamera koja koristi ovu skriptu ima svoj naziv, te redni broj koji se podudara sa brojem scene. Taj objekt za korištenje sprema se u varijablu.

* Varijabla degreeTurn prati koji je trenutni kut pogleda.
* Niz degreeView bilježi po kojim kutevima se kamera može rotirati
* Varijabla camPosition bilježi trenutnu poziciju kamere

Kad skripta primi signal iz jednog od senzora tipkovnice, prvo dohvati u kojem je trenutno položaju kamera. Tome služi varijabla camView. Ta varijabla poprima vrijednosti od nula do tri, što označava ujedno i indeks vrijednosti u nizu sa stupnjevima na koji će se postaviti rotacija kamere. Položaj kamere se također mijenja ovisno o vrijednosti varijable camView – ako zamislimo koordinatni sustav sa x-osi i y-osi, onda se položaj mijenja po kvadrantima. Na slici je prikazano što se događa pri dohvatu pozitivnog signala pritiskom desne strelice. Pritiskom na tipku desne strelice kamera se kreće suprotno od kazaljke sata. Analogno se definira i postupak kad skripta primi pozitivni signal iz pritiska tipke lijeve strelice, osim što se u tom slučaju kamera kreće u smjeru kazaljke sata.

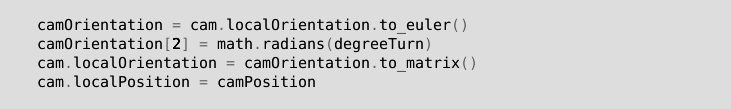


Slika 4.4 Okretanje kamere

Nakon toga, potrebno je promijeniti orijentaciju i položaj kamere. Za orijentaciju kamere, prvo je potrebno dohvatiti njenu trenutnu orijentaciju. Taj podatak spremljen je u obliku matrice, pa ga je potrebno pretvoriti u mjere gdje je moguće koristiti stupnjeve. To se ostvaruje pomoću funkcije to\_euler(), koja matricu trenutno pretvara u vektor Eulerovih mjera na x, y i z-osi.

Sada je moguće koristiti stupanj definiran varijablom degreeTurn. Ovdje je potrebno promijeniti rotaciju po z-osi, pa u vektoru, na zadnjem indeksu, mijenjamo vrijednost iz trenutne u onu koja se nalazi u varijabli degreeTurn. Kada je to izvršeno, promijenjeni vektor ponovno pretvaramo u matricu, te svojstvo orijentacije kamere localOrientation mijenjamo na ažuriranu.

Lokalni položaj kamere odmah vraća vrijednost u obliku vektora, pa svojstvu kamere localPosition dodjeljujemo samo vrijednost iz ažurirane varijable položaja.

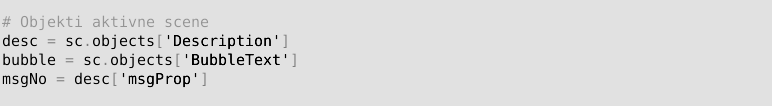


Slika 4.5 Pretvaranje vrijednosti

### Upravljanje položajem objekta na sceni

Ova skripta u biti upravlja položajem događaja koji se okidaju stajanjem na crvenim oznakama, odnosno namješta zadatke za igrača da budu iznad ili ispod objekta interesa.

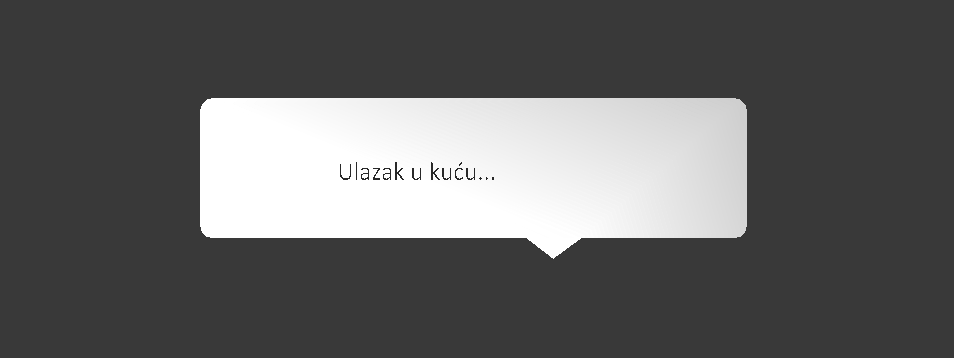
Uz već navedene početne varijable, potrebno je definirati i slijedeće:



Slika 4.6 Tekstualni objekti

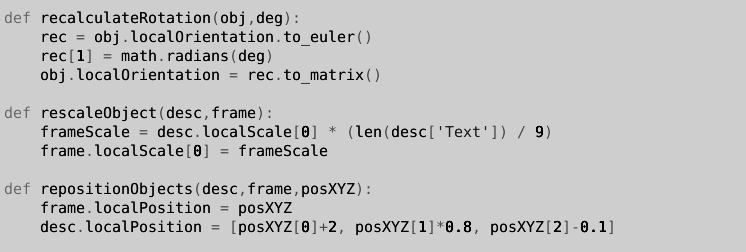
Dakle, potrebno je dohvatiti objekte nad kojima će se skripta izvršavati. Oni se dohvaćaju iz aktivne scene, odnosno scene u kojoj se nalazi kontroler koji pokreće ovu skriptu. Dohvaća se tekstualni objekt iz scene opisa i sprema pod varijablu desc. Njegov tekstualni okvir spremamo pod varijablu bubble.

Varijabla msgNo označava broj objekta koji nosi određenu poruku (pitanje ili zadatak za igrača). Preko te varijable određuje se položaj poruke na sceni.



Slika 4.7 Izgled tekstualnog okvira ili "oblačića"

Za ovu skriptu je definirano par metoda koje preračunavaju rotaciju, skaliranje i translaciju ova dva objekta.



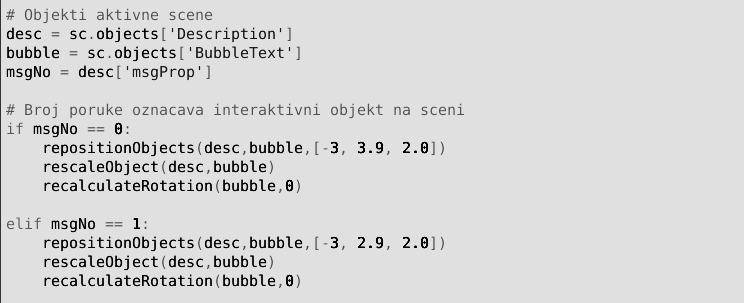
Slika 4.8 Transformacijske metode

Metoda recalculateRotation(obj, deg), ako je potrebna, radi istu zadaću kao i preračun rotacije kod skripte za pomicanje kamere. Prima dva parametra, objekt nad kojim se izvršava te stupanj za koliko se okreće. Potrebno ga je okretati po y-osi (da bude „naopako“ ili u normalnom položaju). Ponovno, lokalnu orijentaciju objekta se dohvati i preračuna u Eulerove mjere, ažurira i tako ažurirano postavi nazad kao orijentacija.

Metoda rescaleObject(desc, frame) prima opet dva parametra, koji su opis i tekstualni okvir, te uzima duljinu teksta opisa, pa matematičkim operacijama produljuje okvir po x-osi, odnosno skalira ga po x-osi.

Metoda repositionObjects(desc, frame, posXYZ) pomiče oba objekta, prima tri parametra: opis, okvir i potrebni položaj, okvir postavi na vrijednost potrebnog položaja, a tekst centrira pomoću te vrijednosti.

Sada sve što preostaje je provjeriti broj objekta, te pozvati prikladne vrijednosti lokacije, rotiranje i skaliranja objekta.



Slika 4.9 Određivanje položaja objekta teksta na sceni

# Zaključak

Ovaj projekt, na kraju, stvarno dokazuje da je Blender moguće koristiti kao alat za izradu jednostavnih interaktivnih sadržaja ili kao alat za izradu prototipa videoigri. Međutim, ne može se reći da trenutno konkurira postojećim alatima namijenjenim primarno kao pogonski sklop igre (primjerice Unity ili Unreal Engine). Stoga, za neke zahtjevnije projekte preporuka bi bila koristiti alate koji su primarno pogonski sklopovi igara, a Blender koristiti strogo kao alat modeliranja likova. Čini se da se s time slažu i autori Blendera, jer u najnovijoj stabilnoj verziji 2.80 Blender Game Engine više nije dostupan. Kao alternativu BGE-u, navodi se Godot Engine[[2]](#footnote-2), a dostupni su i razni vanjski dodaci koji obnašaju funkciju BGE-a.

1. Blender Foundation, „*About*“, <https://www.blender.org/about/> [↑](#footnote-ref-1)
2. Blender Wiki, *Blender 2.80: Removed Features*, <https://wiki.blender.org/wiki/Reference/Release_Notes/2.80/Removed_Features> [↑](#footnote-ref-2)