

# Razvoj interaktivnog izložbenog prostora s dinamičnim grafičkim sadržajem u virtualnoj stvarnosti renderiranjem u realnom vremenu

---

**Bošanjak, Mladen**

**Master's thesis / Diplomski rad**

**2017**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Graphic Arts / Sveučilište u Zagrebu, Grafički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:216:431453>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-13**



*Repository / Repozitorij:*

[Faculty of Graphic Arts Repository](#)



**SVEUČILIŠTE U ZAGREBU**  
**GRAFIČKI FAKULTET**

# **DIPLOMSKI RAD**

Mladen Bošnjak



Sveučilište u Zagrebu  
Grafički fakultet

Smjer: Multimedij

# DIPLOMSKI RAD

**Razvoj interaktivnog izložbenog prostora s  
dinamičnim grafičkim sadržajem u virtualnoj  
stvarnosti renderiranjem u realnom vremenu**

Mentor:

doc. dr. sc. Tibor Skala

Student:

Mladen Bošnjak

Zagreb, 2017.

## **SAŽETAK**

U ovom radu opisuju se početci virtualne stvarnosti, od pojave samog izraza „virtualna stvarnost“, do onog što danas smatramo pod tim pojmom. Objašnjavaju se razlike između suvremenih uređaja za virtualnu stvarnost i njihovi nedostaci. Navode se tehnike koje se koriste za optimizaciju VR aplikacija zbog visokih hardverskih zahtjeva. Kao demonstracija aplikacije za virtualnu stvarnost izrađuje se aplikacija u pokretaču (*eng. 3d game engine*) Unity. U aplikaciji korisniku se omogućuje kretanje kroz virtualni izložbeni prostor izgledom temeljen na unutrašnjosti Grafičkog Fakulteta Sveučilišta u Zagrebu. Opisuje se cijeli proces izrade od stvaranja projekta do završnog testiranja i omogućavanja kretanja na drugim računalima. Integriraju se i omogućavaju opisane optimizacije. Pronalazi se rješenje za specifičan fiziološki i psihološki osjećaj nelagode prilikom kretanja i radi malog broja sličica u sekundi u virtualnoj stvarnosti. Zbog cilja što vizualno impresivnijeg završnog proizvoda, implementiraju se tehnike unaprjeđenja vizualnog izgleda i postavlja se što realističnije svjetlosno okruženje u virtualnoj sceni u kojoj se krajnji korisnik nalazi kada koristi aplikaciju. Kako bi aplikacija imala dodatnu funkcionalnost, omogućuje se apliciranje grafičkog sadržaja dohvaćenog s interneta na teksture u sceni, što aplikaciju pretvara u svojevrsni digitalni izložbeni prostor. Navode se i objašnjavaju moguće drugačije primjene stvorene aplikacije, primjerice na suvremenim operativnim sustavima i uređajima.

### **ključne riječi:**

*virtualna stvarnost*

*aplikacija*

*optimizacija*

*interaktivnost*

*real time renderiranje*

## SADRŽAJ

1. Uvod.....	1
2. Teorijski dio .....	3
2.1. Virtualna stvarnost .....	3
2.1.1. Nedostatci prve generacije vr uređaja .....	4
2.1.2. Specifični fiziološki i psihološki osjećaj nelagode prilikom kretanja i radi malog broja sličica u sekundi u virtualnoj stvarnosti .....	5
2.1.3. Budućnost virtualne stvarnosti .....	7
2.2. Tehnike unaprjeđivanja vizualnog izgleda scene .....	8
2.2.1. Globalna iluminacija.....	8
2.2.2. Post-processing efekti .....	9
2.3. Tehnike optimizacije aplikacija renderiranih u realnom vremenu .....	12
3. Praktični dio.....	16
3.1. Izrada aplikacije za virtualnu stvarnost u pokretaču unity .....	16
3.1.1. Stvaranje novog projekta i postavljanje scene .....	16
3.1.2. Unošenje vlastitih 3d modela i apliciranje materijala.....	18
3.2. Postavljanje svjetala i svjetlosnog okruženja u sceni .....	24
3.3. Dohvaćanje grafičkog sadržaja sa interneta i apliciranje istog kao teksture u aplikaciji.....	27
3.4. Postavljanje prostora na internetu za dohvaćanje grafičkog sadržaja....	28
4. Finaliziranje .....	29
4.1. Testiranje izrađene aplikacije.....	29
4.1.1. Stvaranje <i>builda</i> aplikacije kako bi bilo moguće pokretanje na drugim računalima .....	30
5. Potencijal integracije izrađene aplikacije u druge sustave i uređaje za virtualnu stvarnost .....	31
6. Zaključak .....	33
7. Literatura .....	35
8. Manje poznati pojmovi, akronimi i skraćenice .....	36
9. Popis slika .....	38

## 1. UVOD

Iako se izrada aplikacija za virtualnu stvarnost čini kao kompliciran posao, tvrtke koje prednjače u izradi VR uređaja pobrinule su se da prelazak s izrade 3d aplikacija poput računalnih igara i simulacija na izradu aplikacija za virtualnu stvarnost bude što bezbolniji. Najpopularniji pokretači za izradu interaktivnih aplikacija su Unity i Unreal Engine. Oba podržavaju izradu aplikacija za virtualnu stvarnost kroz SteamVR SDK za HTC Vive i Oculus SDK za Oculus Rift. Uz to, suprotno od konzola za igre poput Sonyjevog PlayStationa, svaki Vive i Rift, uključujući verzije za prodaju potrošačima, mogu služiti za izradu aplikacija. Nije potrebna posebna verzija uređaja, niti posebne licence za izradu aplikacija.

U ovom radu koriste se 3d modeli koji predstavljaju dio unutrašnjosti Grafičkog Fakulteta izrađeni za završni rad preddiplomskog studija dizajna grafičkih proizvoda naslova Proces izrade interaktivne 3d vizualizacije prostora napisanog pod mentorstvom doc. dr. sc. Tajane Koren Ivančević u kojem je detaljno opisan proces izrade 3d modela. Uz postojeće, izrađuju se novi modeli koji unapređuju izgled scene i dodaje se nekoliko novih virtualnih prostorija.

Interaktivne 3d vizualizacije koriste se primjerice u arhitekturi kako bi projektanti klijentima iz prve ruke prikazali kako će neko rješenje izgledati u stvarnosti. U ovom radu cilj je otići korak dalje tako što se korisnik, umjesto da gleda u vizualizaciju kroz ekran, nalazi u samom virtualnom prostoru i može se njime potpuno slobodno kretati. Korištenjem Unity pokretača postiže se cilj stvaranja izložbenog prostora kojeg korisnik može iskusiti u virtualnoj stvarnosti putem uređaja HTC Vive. Radovi izloženi u izložbenom prostoru povlače se s bilo koje lokacije na internetu. Mogu se promijeniti jednostavnom zamjenom grafičkog sadržaja na serveru. Time se dobiva smisao i funkcionalnost aplikacije i pridodaje joj se komercijalni potencijal izvan okvira samog diplomskog rada.

Daljnijim proširivanjem mogućnosti aplikacije moguće bi bilo unaprijediti sustav izloženih radova tako da se uz dvodimenzionalne radove mogu izlagati 3D radovi poput 3D modela ili video sadržaja izrađenih od strane studenata Grafičkog Fakulteta. Studenti bi mogli u virtualnoj stvarnosti vidjeti svoje modele u pravoj veličini i sagledati ih iz svih kutova. Uz ručno izrađene 3D modele aplikaciju je moguće proširiti kako bi se omogućilo izlaganje skulptura i instalacija prenesenih u 3D fotogrametrijom. Time bi virtualni izložbeni prostor bilo moguće pretvoriti u punokrvni muzej s izloženim radovima svih vrsta umjetnika, neovisno jesu li njihovi radovi stvoreni tradicionalnim tehnikama ili digitalni.

Virtualni prostor stvoren za ovaj rad moguće je zamijeniti ili čak korisniku dati opciju da bira izgled prostora u kojem se nalazi. U virtualnoj stvarnosti ne postoji ograničenje izgleda prostora, te bi bilo moguće izlagati radove na nevjerojatnim virtualnim lokacijama poput površine planeta Marsa ili u dubinama oceana.

## 2. TEORIJSKI DIO

### 2.1. VIRTUALNA STVARNOST

Naziv „virtualna stvarnost“ (*eng. Virtual Reality, skraćeno VR*) prvi puta se pojavljuje u kolekciji eseja *Le Théâtre et son double* Antonina Artauda iz 1958. u kojoj autor tim nazivom opisuje iluzionističku prirodu likova i objekata u kazalištu. Termin „virtualno“ od 1959 se koristi kao nešto što ne postoji fizički, no takvim se čini u softveru. [1]

Danas se virtualnom stvarnošću nazivaju iskustva koja doživljavamo kroz *VR* uređaje (*eng. headset*) poput trenutno vodećih HTC Vive od kompanije HTC i Oculus Rift kompanije Facebook.

Palmer Luckey 2012. započeo je Kickstarter kampanju za uređaj Rift kroz svoj startup Oculus [2]. Kampanja je skupila preko 2.5 milijuna američkih dolara. Prije izlaska komercijalne verzije *headseta* kompanija Facebook kupila je Oculus za 2 milijarde dolara [3]. Facebook vjeruje u društvenu aplikaciju ove tehnologije za lakše i prirodnije spajanje ljudi na velike udaljenosti. Valve je paralelno s tim razvijao vlastiti *headset* sličnih hardverskih sposobnosti, no superiornijeg sistema kretanja kroz virtualnu stvarnost. U suradnji s HTC-om ostvarili su masovnu proizvodnju hardvera. Njihov proizvod nazvan je HTC Vive, vidljiv na slici 1 [4].





Slika 1. HTC Vive headset s controllerima i baznim stanicama za praćenje korisnika u prostoru.

### 2.1.1. Nedostatci prve generacije VR uređaja

Ne računajući uređaje za mobilni VR poput Google Cardboarda i Samsung GearVR-a koji koriste korisnikov pametni telefon kao centralni dio hardvera uređaja, sagledavaju se nedostaci VR uređaja na koje se fokusiramo u ovom radu.

Najveći nedostatak uređaja HTC Vive i Oculus Rift je rezolucija ekrana u *headsetovima*. Iako koriste ekrane s jednom od najvećih gustoća piksela trenutno dostupnih u komercijalnim proizvodima. Sveukupno 1200 puta 2160 piksela (1080 puta 1920 po oku) na ekranu od 5.7 inča, no kada takav ekran približimo očima na udaljenost koju koristimo u *headsetima*, očito je vidljiva mreža piksela koja sačinjava ekran.

Cijena je velika prepreka potencijalnom korisniku VR tehnologije zbog cijene samih VR uređaja koji su još uvijek preko 700 američkih dolara (za vrijeme pisanja ovog rada cijena Oculus Rift uređaja s Touch Controllerima i sensorima pala je na 500 američkih dolara što pokazuje pozitivan trend u smanjenju cijena VR uređaja [5]). Uz to minimalna konfiguracija računala potrebnog za današnje uređaje virtualne stvarnosti također je puno zahtjevnija i time skuplja od laptopa ili stolnog računala prosječnog potencijalnog korisnika.

Bežična tehnologija za komercijalne svrhe nije još na dovoljnoj razini da bi podržavala VR uređaje zbog količine informacija koje ti uređaji prenose. Trenutno se HTC Vive spaja na računalo s jednim HDMI kablom za protok vizuala, dva USB 3.0 kabela za podatke i jednim kablom u utičnicu za dovod struje. To uvelike narušava ugodu korisniku kod korištenja aplikacija zato što je konstantno svjestan da je spojen kablom kod kretanja.

Jedini današnji VR uređaj koji omogućuje kretanje na veličini cijele prostorije je HTC Vive, no osposobljavanje prostorije za kretanje (*eng. Room Setup*), je kompliciran proces koji uključuje bušenje zidova kako bi se na njih mogli montirati senzori ili kupovinu dodatnih stalaka za senzore.

#### 2.1.2. Specifični fiziološki i psihološki osjećaj nelagode prilikom kretanja i radi malog broja sličica u sekundi u virtualnoj stvarnosti

Osobe koje su izrađivale (*eng. developers*) dosadašnja iskustva za računala poput proizvođača računalnih igara suočavaju se s novim setom problema kod izrade aplikacija za virtualnu stvarnost.

Glavni od tih su mučnina kod tradicionalnog načina kretanja kroz 3d prostor, visoki zahtjevi uređaja zbog rezolucije ekrana u *headsetovima* i renderiranja svake sličice u sekundi dva puta (jednom za svako oko). Uz to aplikacije za virtualnu stvarnost moraju se konstantno renderirati u 90 sličica u sekundi. Padanje broja sličica u sekundi (*eng. framerate*) ispod tog praga rezultira trenutačnom mučninom kod korisnika.

Novi problem koji se pojavio kod izrade aplikacija za virtualnu stvarnost jest taj da tradicionalni načini kretanja u 3d aplikacijama izazivaju mučninu kod korisnika.

Pod tradicionalnim načinom kretanja smatra se kretanje pritiskom na tipku ili pomicanje palice na igraćoj kontrolnoj ploči (*eng. gamepad*) kod čega kamera u aplikaciji simulira kretanje u stvarnosti, glatkim pomicanjem u željenom smjeru. Iako trenutno nije objašnjen točan razlog zbog čega dolazi do mučnine, kod testiranja aplikacija za virtualnu stvarnost uviđa se kada dolazi do mučnine [6]. Kada se tijelo korisnika u virtualnoj stvarnosti počne pomicati u nekom smjeru, mozak korisnika kroz oči javlja da se tijelo pomiče, no unutarnje uho korisnikovog tijela svjesno je da je tijelo u stvarnosti nepomično te šalje različite signale od očiju mozgu. Kod primitka tih različitih signala mozak korisnika aktivira mučninu.

Kod pada *frameratea* ispod 90 sekundi u virtualnoj stvarnosti većina korisnika doživljava mučnine. Razlog ovog fenomena također nije u potpunosti objašnjen od strane znanstvenika.

Kako bi se izbjegao tradicionalan način kretanja, postoji više opcija koje se mogu koristiti u virtualnoj stvarnosti kako bi se izbjegla mučnina kod korisnika. Kretanje kroz prostor na veličini prostorije koje je prvotno uvedeno s uređajem HTC Vive omogućuje kretanje korisnika u prostoru maksimalne veličine od 4 metra četvornih uz pomoć dediceranih baznih stanica u kutovima prostorije. Korisnikove kretnje u stvarnom svijetu direktno se transliraju u virtualnu stvarnost. Problem kod ovog načina kretanja je limitiran prostor kretanja.

Kako bi se izbjegla limitiranost prostora kretanja uvedena je teleportacija. Korisnik uperi ručni upravljač (*eng. controller*) u smjeru u kojem se želi teleportirati i pritiskom na određeni gumb aplikacija translirira korisnikovu poziciju u virtualnu stvarnost na novo odabranu poziciju. Ovaj način kretanja eliminira mučninu zbog toga što se korisnik trenutno pojavi na novoj lokaciji što u mozgu ne izaziva mučninu jer unutarnje uho to doživljava kao da se korisnik nije pomaknuo.

### 2.1.3. Budućnost virtualne stvarnosti

Microsoft je za svoj uređaj miješane stvarnosti Hololens izradio sustav za praćenje pozicije korisnika (*eng. tracking*) gdje se za praćenje korisnika u prostoru umjesto vanjskih senzora kao kod Oculus Rifta i HTC Vivea koriste senzori koji su u samom uređaju (*eng. inside-out tracking*) [7].

Taj sustav Microsoft daje na korištenje kompanijama koje stvaraju vlastite uređaje za miješanu i virtualnu stvarnost poput Lenovo-a, Acera, Asusa, HP-a i drugih.

Njihovi uređaji imat će veću rezoluciju ekrana, no umjesto OLED tehnologije, koristit će LCD što će smanjiti cijenu uređaja, no sniziti kvalitetu slike koju korisnik doživi kroz uređaj. Ovi uređaji još uvijek će se smatrati prvom generacijom uređaja za virtualnu stvarnost.

Velike kompanije poput Valve-a, Microsofta i Oculus (Facebook) već rade na sljedećoj generaciji uređaja, no malo je poznato o njihovim namjerama za sljedeću generaciju. Vrlo je vjerojatno da će novi uređaji imati puno veću rezoluciju ekrana (4K), manje vrijeme odaziva, *inside-out tracking* kakav već sada koristi Microsoft i vrlo vjerojatno će uređaji biti u potpunosti bežični. Kako bi VR uređaji postali normalna stvar u kućanstvu kao što je to danas osobno računalo ili konzola za igranje računalnih igara, potrebno je ispuniti sve navedene uvjete, te uz njih smanjiti cijenu uređaja.

Problem kod ovih predviđanja jest potreba za puno jačim pokretačima 3d grafike. Niti današnja računala, niti ona skorije budućnosti nemaju grafičke procesore dovoljno snažne za renderiranje realističnih 3d prostora u 4K rezoluciji dva puta po slici u sekundi (*eng. frame*) (koliko je potrebno za virtualnu stvarnost), te će tehnologija grafičkih procesora morati znatno napredovati u skorije vrijeme.

Moguće rješenje nedostatka hardverske snage računala nalazila bi se u samom *headsetu*, poput uređaja koji bi pratio korisnikov pogled (*eng. eye tracker*) te na mjestima ekrana u koja korisnik gleda povećao gustoću renderiranih piksela, a gustoću piksela u periferiji vida smanjio. Slično kao što funkcioniraju ljudske oči

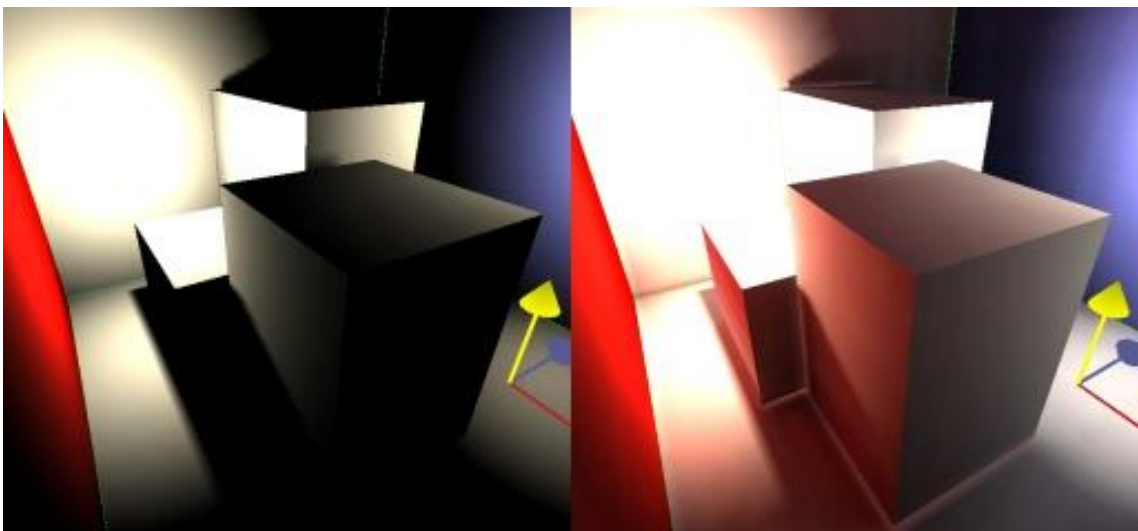
gdje oštro vidimo samo 20 stupnjeva oko točke u koju smo uperili pogled. Ljudske oči ne bi bile sposobne primijetiti manju gustoću piksela u perifernom vidu jer same oči ne vide jasno taj dio vidnog polja.

## 2.2. TEHNIKE UNAPRJEĐIVANJA VIZUALNOG IZGLEDA SCENE

### 2.2.1. Globalna iluminacija

U računalnoj grafici do nedavno za izvore svjetlosti izračunavala se samo svjetlost koja direktno pada na površine. Globalna iluminacija je sistem koji izračunava kako se svjetlost odbija od površina na druge površine (indirektno svijetlo) [8]. Primjerice kada na objekt žarko crvene boje pada svjetlost ,iako sam ne emitira svjetlost, objekti koji su mu u blizini zbog difuzne svjetlosti odbijene s njega postaju crveni, kao što je vidljivo na slici 2.

Iako je kod tradicionalnih 3d aplikacija moguće kalkulirati globalnu iluminaciju za svaku sličicu u sekundi, za *VR* aplikacije to je trenutno nedostižno zbog prevelikih zahtjeva samog *VR* uređaja. Iz tog razloga globalna iluminacija izračunava se prije pokretanja aplikacije, u procesu njenog stvaranja.



Slika 2. Lijevo na slici vidljiva je scena bez kalkulirane globalne iluminacije, desno je ista ta scena s unaprijed kalkuliranom globalnom iluminacijom.

Nedostatak ove tehnike prikazivanja globalne iluminacije je nemogućnost mijenjanja intenziteta, rotacije ili boje svjetlosti. Budući da je globalna iluminacija unaprijed izračunata, ona neće reagirati na promjene u osvjetljenju, što dovodi do fizikalno nepravilnog izgleda scene.

### 2.2.2. Post-processing efekti

Niz tehnika kojima *developeri* uljepšavaju izgled scena u 3d aplikacijama (*eng. post-processing effects*) nazivaju se tako zato što uzimaju informacije iz već renderirane sličice i na temelju nje prikazuju efekte[9].

SSAO (*skraćénica za eng. screen space ambient occlusion*) i njegove različite iteracije služe za simuliranje suptilnih kontaktnih sjena između objekata u sceni. Na slici 3 vidljiv je sam SSAO efekt s isključenim bojama i ostalim svjetlosnim efektima u sceni.



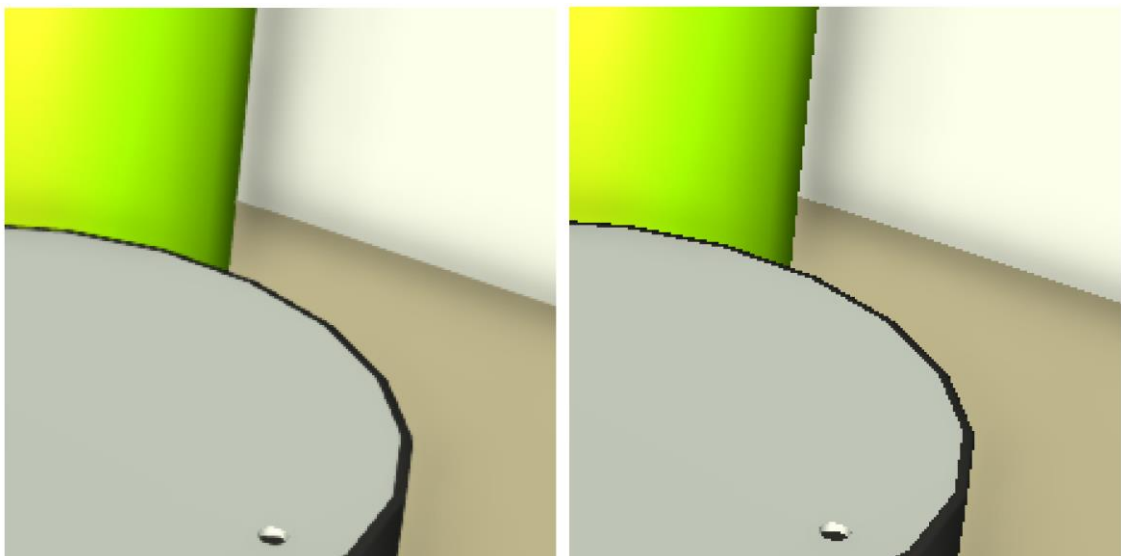
Slika 3. Scena iz računalne igre Assassin's Creed: Syndicate s isključenim bojama i vidljivim samo SSAO efektom.

*Bloom* efekt simulira način na koji vrlo intenzivno svjetlo promatraču izgleda kao da izlazi iz okvira objekta koji emitira svjetlost, kao što je vidljivo na slici 4.



Slika 4. Lijevo je prikazan model renderiran u pokretaču Unreal Engine bez bloom efekta, desno s bloom efektom.

*Anti-aliasing* vrlo je bitan *post-effect* kod virtualne stvarnosti zbog relativno male rezolucije ekrana u *headsetovima*. Radi na principu uspoređivanja zadnjih nekoliko renderiranih sličica u sekundi gdje pronalazi razliku u poziciji piksela na rubovima 3d objekata. Bazirano na prošloj renderiranoj sličici zamućuje boje na rubovima objekata kako bi se izbjegla osjetna oštrina u razlici boje u pikselima. Demonstracija *Anti-aliasinga* prikazana je na slici 5.



Slika 5. Lijevo je prikazan približeni dio renderirane scene s *Anti-aliasing* efektom. Desno je ista scena bez korištenja efekta.



U aplikacijama za virtualnu stvarnost ne koristimo sve efekte koji se koriste u tradicionalnim 3d aplikacijama iz više razloga. *Depth of field* koji filmski zamućuje prednji ili pozadinski plan, ne koristi se zato što u VR-u sam korisnik određuje što će biti u fokusu svojim pogledom. *Motion blur* koji zamućuje objekte koji se pomiču velikom brzinom također se ne koristi jer kvaliteta slike mora biti uvijek oštra.

Kromatska aberacija simulira efekt filmske kamere tako što prema rubovima renderirane scene sve više zamjenjuje crvene, zelene i plave piksele koristi se u drugačije svrhe kod virtualne stvarnosti. U tradicionalnim aplikacijama koristi se za dobivanje filmskog efekta, dok se u aplikacijama virtualne stvarnosti koristi kako bi se umanjilo stvaranje stvarne kromatske aberacije koju korisnik vidi kroz leće VR uređaja. Na slici 6 očita je kromatska aberacija na lijevom gornjem kutu VR displaya.



Slika 6. Prikaz kromatske aberacije na lijevom gornjem kutu VR displaya.

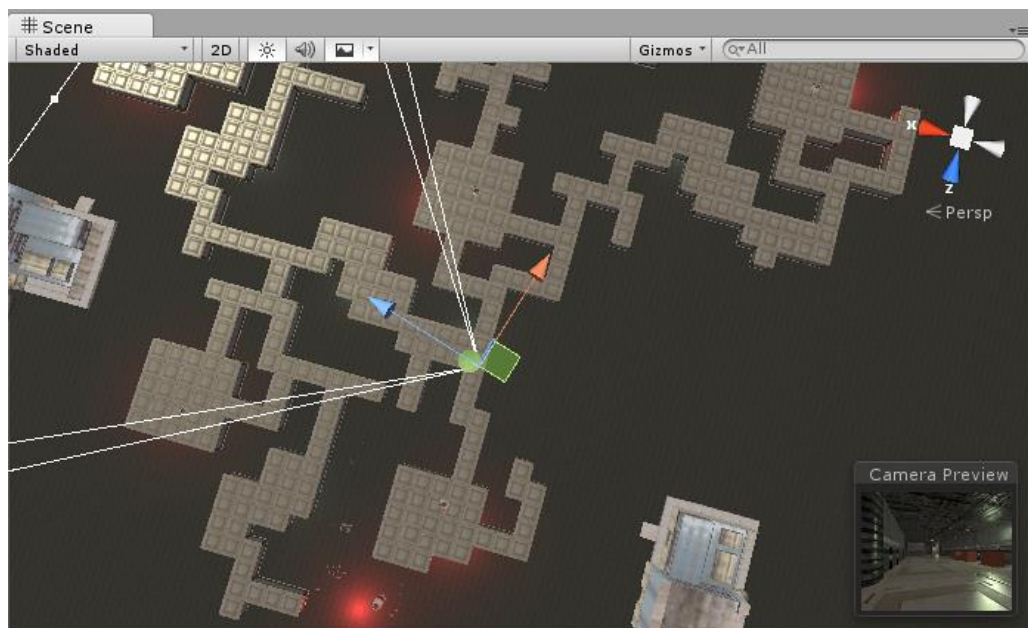


## 2.3. TEHNIKE OPTIMIZACIJE APLIKACIJA RENDERIRANIH U REALNOM VREMENU

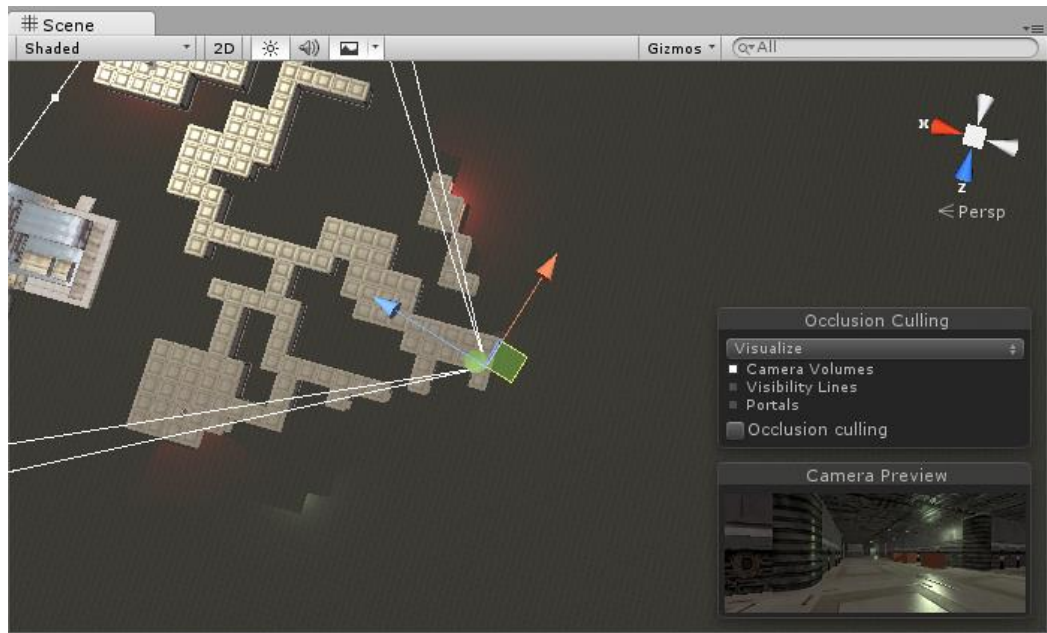
Kako bi se dostupna snaga hardvera na kojoj se pokreće aplikacija iskoristila što je efikasnije moguće, razvile su se mnoge tehnike optimizacije renderiranja u realnom vremenu [10]. Aplikacija razvijena u eksperimentalnom dijelu ovog rada koristi mnoge tehnike optimizacije kako bi ju bilo moguće renderirati u 90 sličica u sekundi na rezoluciji od 2150x1200 piksela.

### *Frustum culling*

*Frustum culling* funkcionira tako da isključuje renderiranje objektima koji su izvan pogleda kamere. To ne znači da isključuju renderiranje objekata koji nisu vidljivi zato što ih zaklanja neki bliži objekt. Za to se koristi *occlusion culling*. Na slikama 7 i 8 demonstriran je *frustum culling*.



Slika 7. Bijelim obrubom prikazano je vidno polje korisnika. Bez upotrebe frustum cullinga renderiraju se i objekti koji nisu u vidnom polju.

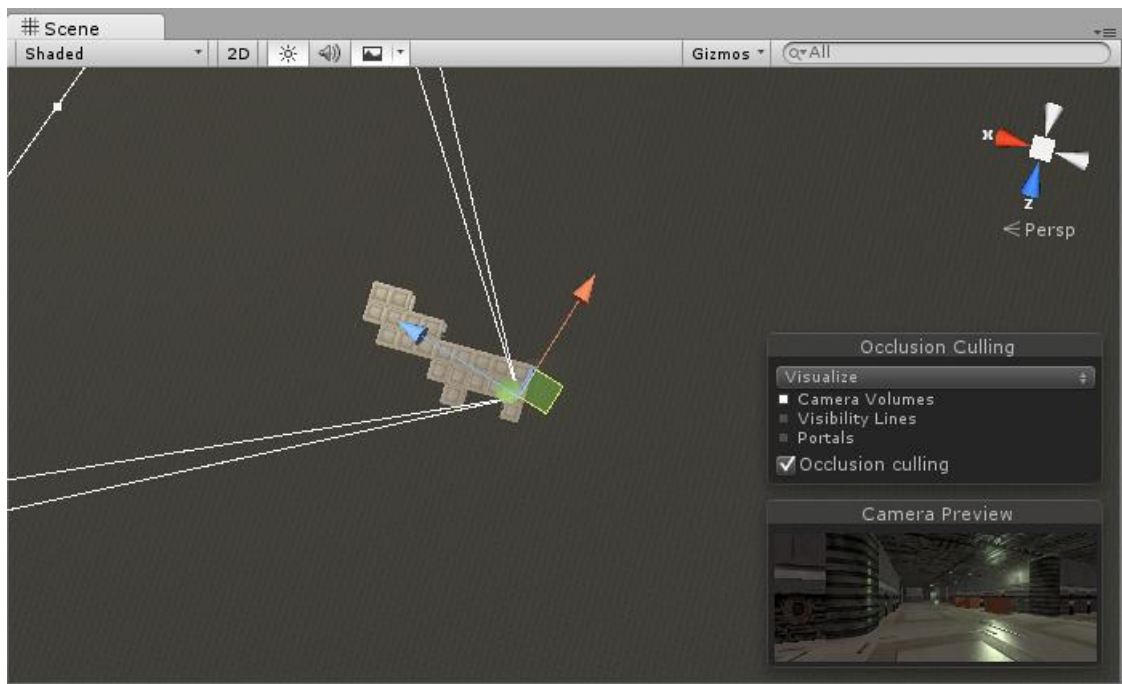


Slika 8. Korištenjem frustum cullinga, objekti koji nisu u vidnom polju korisnika izostavljeni su kod renderiranja, čime se uvelike ubrzava renderiranje.

#### Okluzija (*eng. occlusion culling*)

Occlusion culling je jedan od novijih i naprednijih načina za optimizaciju scene u Unity-u i drugim pokretačima. Funkcionira tako da isključuje renderiranje objekata koji nisu vidljivi zato što su iza nekog drugog objekta.

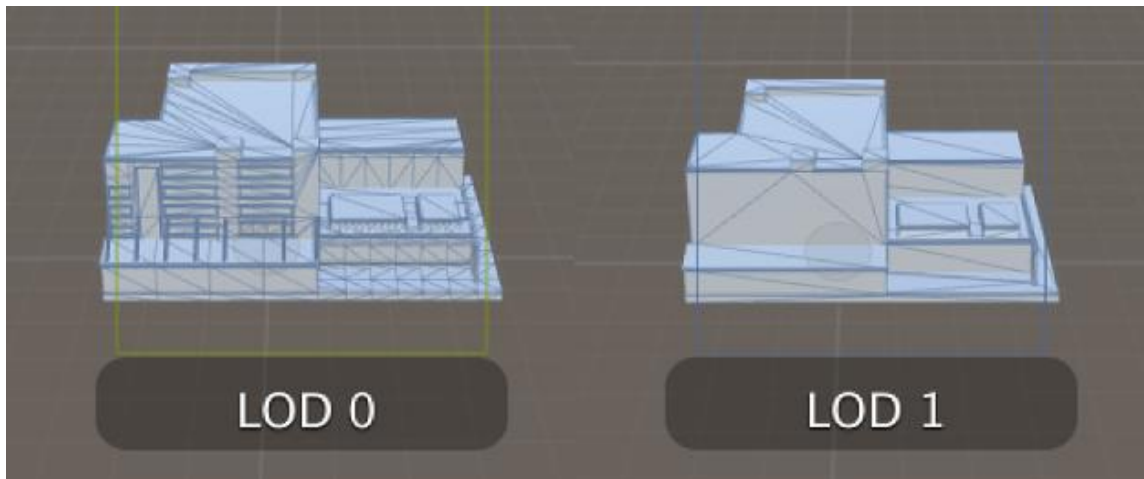
Tradicionalno u računalnoj grafici objekti koji su najdalje od kamere renderiraju se prvi i bliži objekti se renderiraju nakon toga. To nije optimalno zato što u puno slučajeva dalje objekte ne vidimo zato što ih zaklanjaju bliži objekti, te dalje objekte nije potrebno renderirati. Slika 9 demonstrira učinak *occlusion cullinga*.



Slika 9. Većinu objekata u vidnom polju korisnika zaklanjaju kocke. Upotrebom occlusion cullinga objekti koji se ne vide ne renderiraju se.

### *Level of detail*

Objekti koji su daleko od kamere renderirani su u manje piksela od objekata koji su bliži kameri. Budući da na udaljenijim objektima korisnik ne vidi detalje istom oštrinom kako ih vidi na bližim objektima, moguće je zamijeniti izgled objekta kada je na određenoj udaljenosti kako bi se smanjilo potrebno procesiranje na grafičkom procesoru uređaja za taj objekt. Potrebno je ručno u programu za modeliranje izraditi više modela istog objekta, svakog sa sve manje detalja. Kao što je vidljivo na slici 10. Kod spremanja modela potrebno je nazvati objekt „nazivobjekta\_LOD0“ koji će Unity prepoznati kao najdetaljniju verziju objekta i koju će renderirati kada je objekt vrlo blizu kamere. Model objekta s manje detalja potrebno je nazvati „nazivobjekta\_LOD1“ i tako nadalje.



Slika 10. Lijevo je prikazan model kakav se pojavljuje kad je blizu kamere. Desno je prikazan isti model kakav se pojavljuje kada je daleko od kamera. Primjetno je veliko smanjenje broja poligona od kojih je model izrađen.

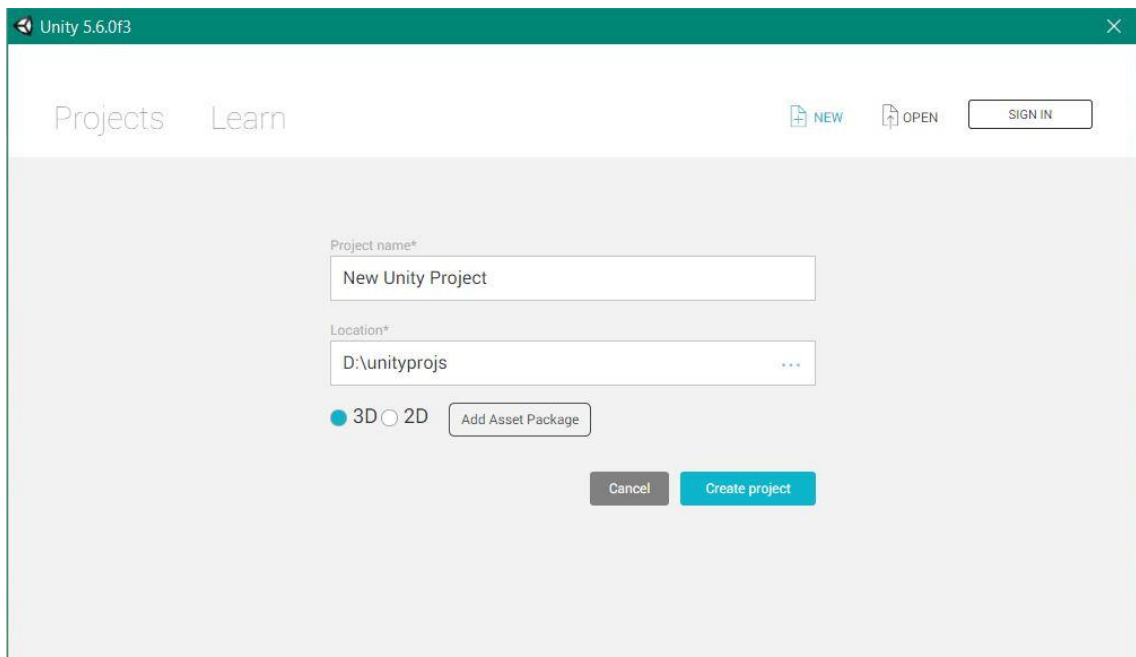
### 3. PRAKTIČNI DIO

#### 3.1. IZRADA APLIKACIJE ZA VIRTUALNU STVARNOST U POKRETAČU UNITY

Praktični dio ovog rada izrađuje se u pokretaču Unity zbog njegove svestranosti i lakoće izrade aplikacija. Unity je najpopularniji pokretač 3d grafike na svijetu i služi za izradu mnogih popularnih 3d aplikacija namijenjenim za korištenje na osobnim računalima, mobitelima i virtualnoj stvarnosti.

##### 3.1.1. Stvaranje novog projekta i postavljanje scene

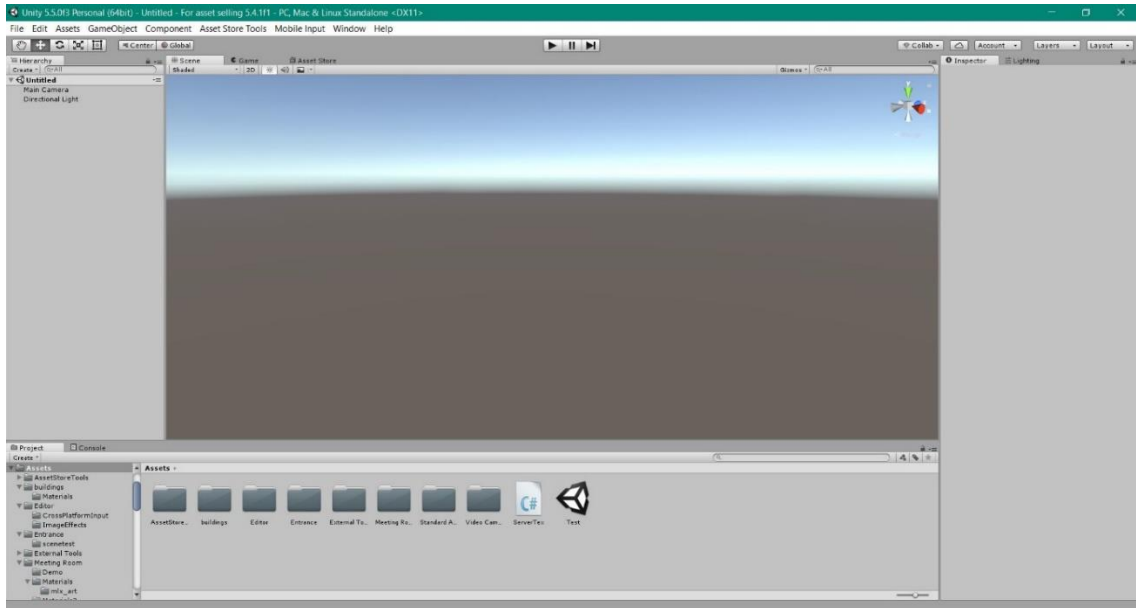
Kod pokretanja Unity-a javlja se prozor kao u slici 11 u kojem je moguće otvoriti postojeći projekt ili stvoriti novi.



Slika 11. Prozor pokretača Unity u kojem se odabire vrsta projekta.

Pritiskom na ikonu New otvara se prozor u kojem se bira ime projekta, lokaciju na kojoj će se nalaziti na disku te je li projekt 3D ili 2D. Projekt ovog diplomskog rada je u 3 dimenzije te se iz tog razloga bira opcija 3D.

Unity za scene koristi .scene fajlove. Izrađuje se novi .scene pritiskom na File>New Scene u glavnoj traci izbornika u pokretaču. Izgled standardnog Unity sučelja prikazan je na slici 12.



Slika 12. Standardno sučelje pokretača Unity.

U glavnom prozoru vidljivo je standardno Unity nebo. U hijerarhiji scene vidljive su dvije stavke – Main Camera, kamera kroz koju će korisnik vidjeti virtualni svijet i Directional Light, svjetlo koje simulira sunce.

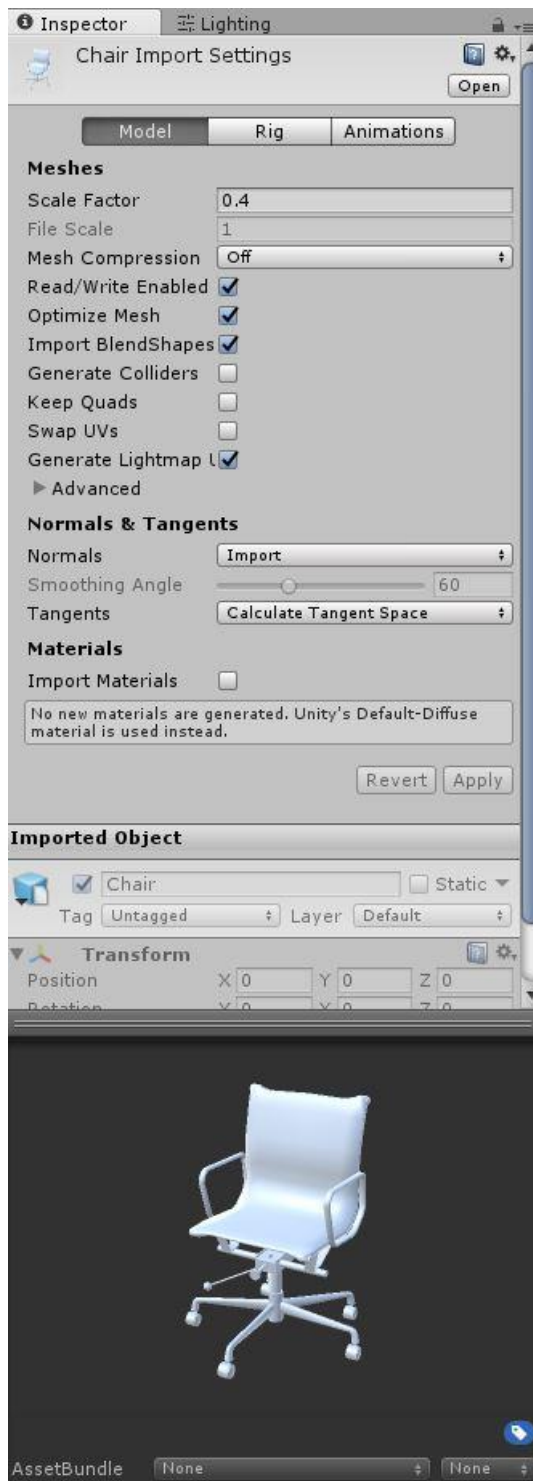
### 3.1.2. Unošenje vlastitih 3d modela i apliciranje materijala

Unaprijed izrađeni modeli u programu za modeliranje (u slučaju ovog rada Blender) unose se u Unity najčešće u .fbx ili .obj formatu. Ako Unity ne podržava nativni format programa u kojem je model izrađen, potrebno je izvesti (*eng. export*) model u jedan od formata koje Unity podržava.

Odvlačenjem fajla modela u Assets prozor Unityja ili pohranjivanjem fajla u folder na disku gdje se nalaze podaci projekta model se ubacuje u projekt. Unity nakon detekcije novog modela počinje proces uvoženja (*eng. import*) modela u projekt. Pritiskom na ikonu modela u Assets prozoru Unityja, moguće je mijenjati postavke *importa* ako je to potrebno.

U slučaju modela stvorenih za aplikaciju ovog diplomskog rada, potrebno je promijeniti nekoliko postavki *importa* u Inspector prozoru vidljivom na slici 13. Veličina modela (*eng. Scale Factor*), potrebno je smanjiti kako bi bio realne veličine naprema korisniku u virtualnoj stvarnosti.

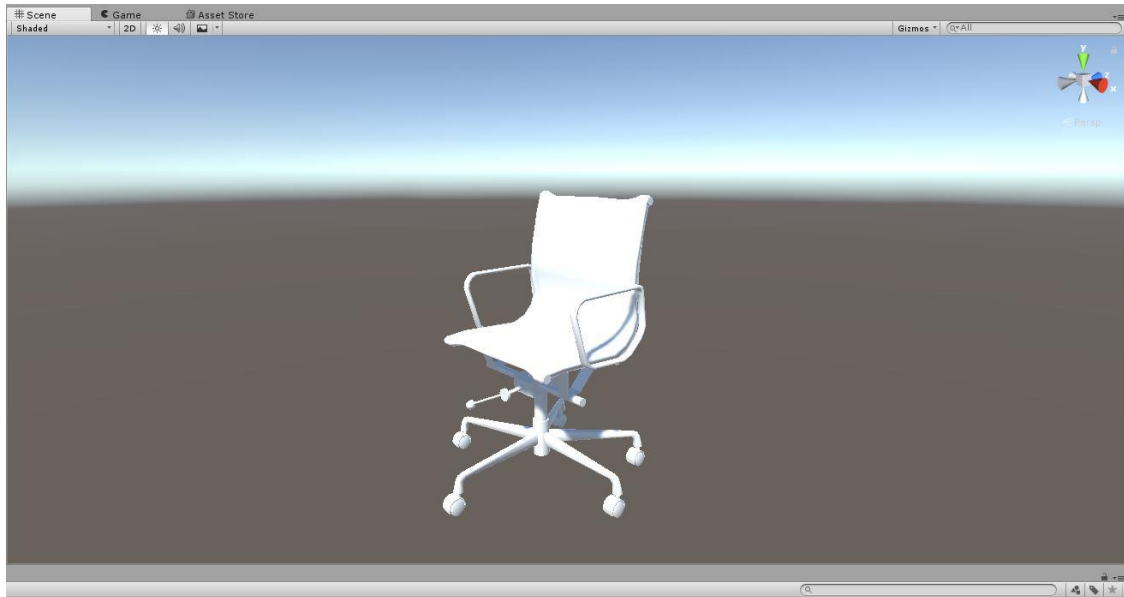
*Generate Lightmap Uvs* potrebno je omogućiti kako bi pokretač znao kako aplicirati direktnu i difuznu svjetlost na model kod kalkulacije svjetlosti. *Import materials* potrebno je onemogućiti budući da se materijali stvaraju u Unity-u, a ne koriste se materijali generirani u programu za modeliranje u kojem je model stvoren.



Slika 13. Inspector prozor pokretača Unity.



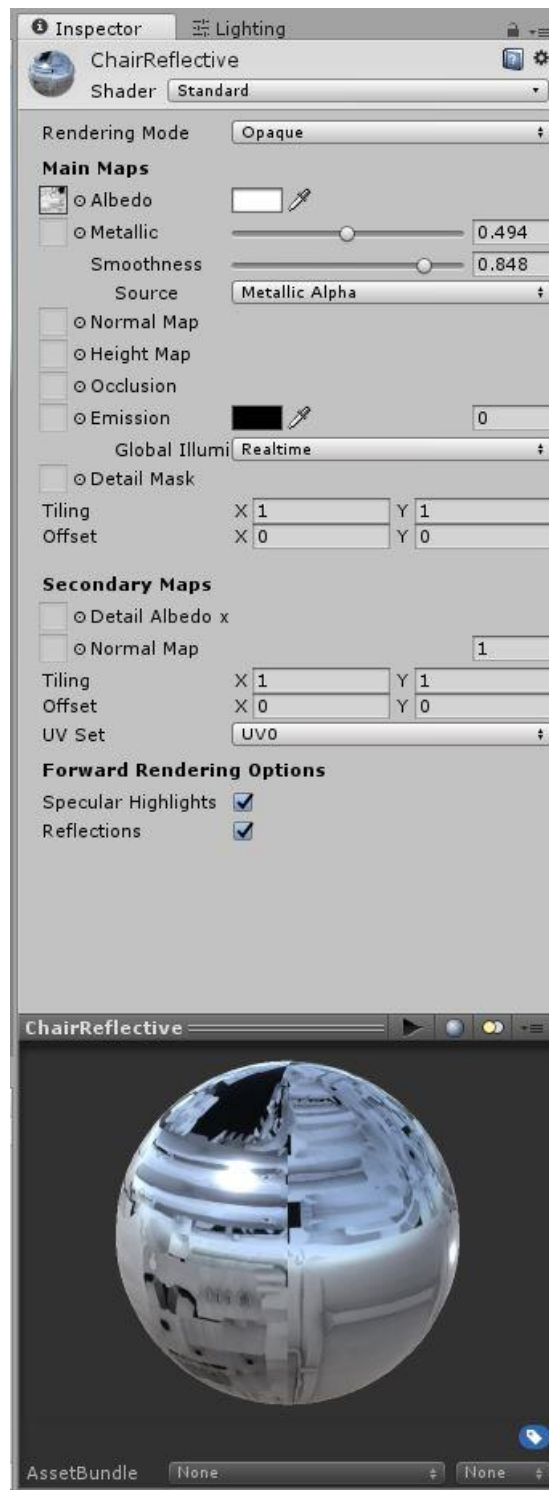
Odvlačenjem *importantog* modela u Scene prozor Unity-a, model se vizualno pojavljuje u sceni te je dodan u hijerarhiju scene. Vidljivo na slici 14.



Slika 14. Model dodan u scenu. Osvijetljen je Directional Light-om koji je stvoren prilikom stvaranja scene i indirektno osvjetljen difuznom svjetlošću automatski generiranog neba.

Teksture za model *importantne* su na isti način kao i sam model te se pojavljuju u Assets prozoru Unity-a. Novi materijal stvara se pritiskom desnog klika na Assets prozor i odabirom Create>Material.

U Inspector prozoru pokretača vidljive su postavke materijala, vidljivo na slici 15. Albedo je tekstura koja djeluje na model kao sama boja materijala. Dijelovi UV mape pobožani crvenom bojom na modelu će izgledati kao crveni materijal. *Slider* Metallic određuje koliko je materijal metalan (koliko će svjetlosti površina odbijati iz svoje okoline). *Slider* Smoothness određuje koliko je materijal gladak (koliko će jasno odbijati svjetlost). Materijal poput stakla ili zrcala imat će smoothness i metallic *slider* do kraja, budući da su vrlo glatki materijali, dok će materijal poput željeza imati metallic *slider* do kraja, no smoothness *slider* vrlo nisko zbog mikroskopskih nepravilnosti na materijalu koji difuzno raspršuju svjetlost te je nemoguće jasno vidjeti svoj odraz u takvom materijalu. Model s korektno apliciranim materijalom prikazan je na slici 16.



Slika 15. Inspector prozor materijala.



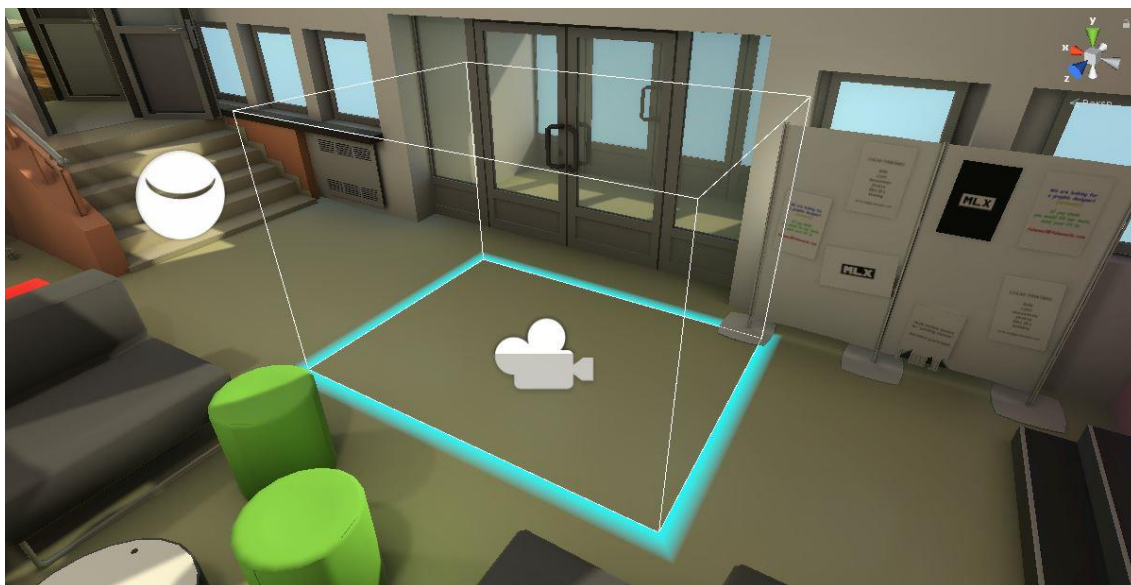
Slika 16. Model u sceni projekta s dodanim materijalom.

### 3.1.3. Proširenje mogućnosti Unity pokretača korištenjem SteamVR i Arc Teleporter *plugina*

Kao pokretač VR sustava koristi se SteamVR proširenje pokretača (*eng. plugin*) za Unity od Valve-a, tvorca HTC Vive tehnologije [11]. Besplatni i plaćeni *plugini*, dodatni modeli, te skripte moguće je preuzeti s Unity-ovog dedicanog Asset Store-a. Oba *plagina* koji se koriste u ovom projektu potpuno su besplatni i moguće ih je koristiti u komercijalne svrhe.

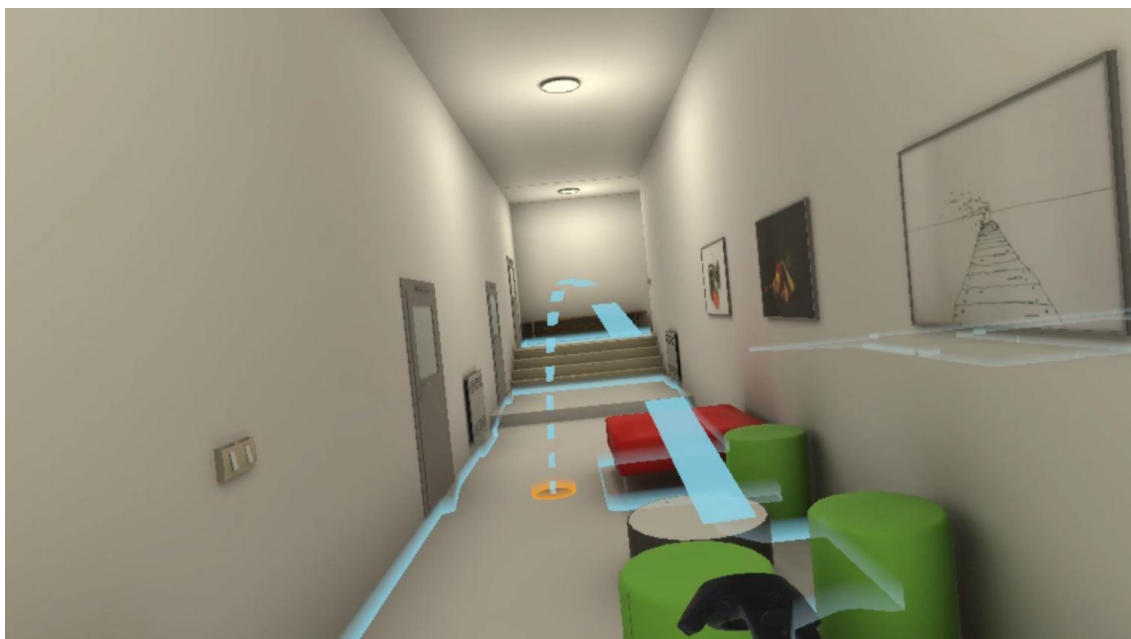
Kod stvaranja okruženja za podržavanje *VR-a*, potrebno je promijeniti postavke u pokretaču koje savjetuje *plugin*. *Plugin* stvara svoje datoteke u Asset datoteci projekta. U tim datotekama nalaze se sve skripte i resursi potrebni za pokretanje projekta na uređaju za virtualnu stvarnost.

Dovlačenjem resursa [Camera Rig] iz datoteke SteamVR *plagina* u scenu stvara se sustav koji sadrži svu logiku vezanu uz *VR* kao što je vidljivo na slici 17. Upravlja logikom *headseta* i ručnih *controllera* tako što njihovu lokaciju u stvarnom svijetu prenosi na virtualnu kameru i *controllere* u aplikaciji.



Slika 17. Camera Rig pozicioniran u sredinu scene.

Od ovog dijela izrade aplikacije nadalje, moguće je testiranje na VR uređaju, no nije moguće kretanje kroz prostor osim u kvadratu u kojem možemo hodati i u stvarnom prostoru. Kako bi se taj nedostatak ispravio, koristi se Arc Teleporter *plugin* [12]. Arc Teleporter omogućuje da se pritiskom na određenu tipku na *controlleru* stvori luk iz *controllera* prema točki u koju je uperen. To je demonstrirano na slici 18. Puštanjem tipke korisnik se teleportira na poziciju gdje pada luk. Ovom tehnikom moguće se s lakoćom teleportirati po cijeloj sceni bez stvaranja osjećaja mučnine kod korisnika.



Slika 18. Prikaz korištenja Arc Teleporter-a unutar aplikacije.

### **3.2. POSTAVLJANJE SVJETALA I SVJETLOSNOG OKRUŽENJA U SCENI**

Nakon postavljanja izrađenih modela u scenu potrebno je postaviti komponente u scenu koje će biti izvori svjetla, točke iz koje će se računati refleksije i generalne postavke kvalitete sjena i globalne iluminacije. Slike 19 i 20 prikazuju usporedbu scene bez i s kalkuliranom svjetlošću.



Slika 19. Scena bez postavljenih svjetala i Reflection Probe-ova. Nema izračunatu globalnu iluminaciju, niti sjene u sceni.



Slika 20. Scena s postavljenim komponentama za svjetlost i izračunatom globalnom iluminacijom i sjenama.

Glavni izvor svjetla bit će direkcionalno svjetlo (*eng. directional light*) koje simulira sunce i prolazi kroz prozore prostorije. Pomoćna svjetla bit će svjetla koja se šire u svim smjerovima iz jedne točke do određenog radijusa (*eng. point light*). Ona simuliraju svjetla iz žarulja u prostoriji. Svakom svjetlu moguće je mijenjati karakteristike poput intenziteta, boje i radijusa u kojem svijetli.



Unity za pridodavanje refleksija materijalima koji reflektiraju svjetlost (metal, sjajna plastika...) koristi Reflection Probe komponentu. Reflection Probe izračunava refleksije iz jedne točke u prostoru i pridodaje te refleksije svim objektima u ručno određenom prostoru. Prostor u kojem modeli dobivaju refleksije najčešće se određuje po prostorijama u sceni, kako bi u svakoj prostoriji modeli imali čim realističnije refleksije. Primjer je glavna prostorija ulaza u fakultet gdje jake refleksije daju prozori kroz koje prolazi danje svjetlo, dok za hodnik gdje uopće nema prozora treba postaviti drugi Reflection Probe kako bi modeli koji se nađu unutar radijusa tog Reflection Probe-a imali realistične refleksije. Na slici 21 prikazani su svi dijelovi sustava za osvjetljenje.

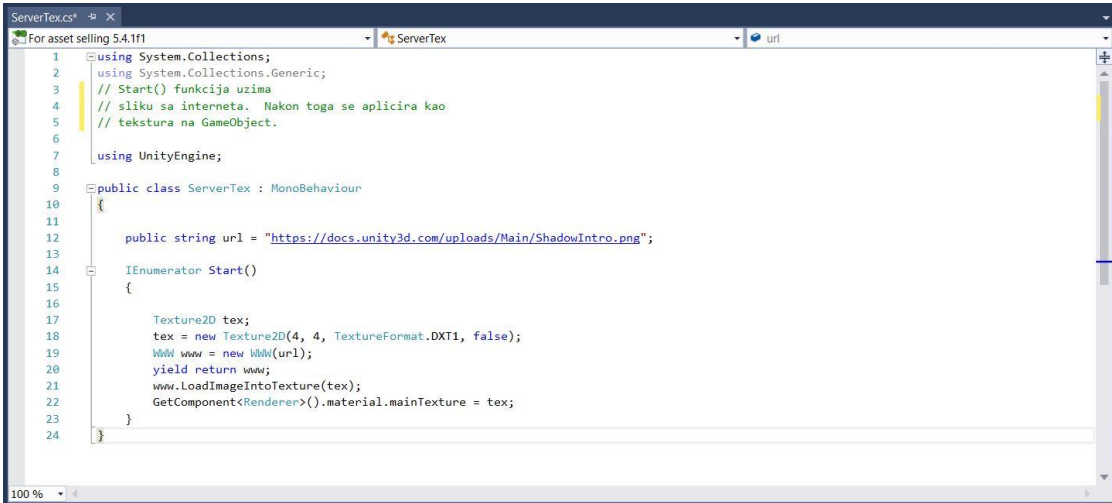


Slika 21. Lijevo je prikazano direkcionalno svjetlo koje simulira sunce, na sredini scene je Reflection Probe koji pridodaje refleksije modelima i desno su dva point svjetla koji simuliraju žarulje.

Postavke sjena i globalne iluminacije određuju se u Lighting kartici pojedine scene. Moguće je postaviti intenzitet refleksija, kvalitetu sjena, jačinu difuzne svjetlosti kod globalne iluminacije i veličinu tekstura u koje se spremaju informacije o sjenama i globalnoj iluminaciji.

### 3.3. DOHVAĆANJE GRAFIČKOG SADRŽAJA SA INTERNETA I APLICIRANJE ISTOG KAO TEKSTURE U APLIKACIJI

Budući da je Unity kao pokretač vrlo fleksibilan i dizajniran za jednostavno korištenje, nije potrebno puno koda kako bi se osposobilo dohvaćanje grafičkog sadržaja s interneta i apliciranje istog na određenu teksturu u aplikaciji. Kod je napisan u .cs skripti u jeziku C#. Kod se editira u vanjskom editoru po odabiru programera. Vanjski editor direktno je integriran s pokretačem i svaka izmjena automatski je vidljiva u oba programa. Za ovaj rad koristi se editor Visual Studio od kompanije Microsoft. U nastavku je na slici 22 prikazana cijela skripta.



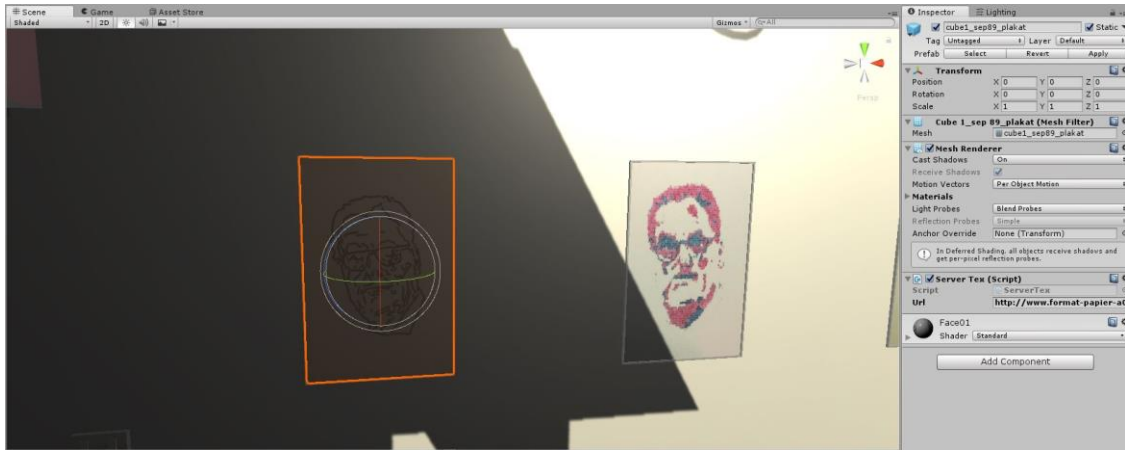
```
1 using System.Collections;
2 using System.Collections.Generic;
3 // Start() funkcija uzima
4 // sliku sa interneta. Nakon toga se aplicira kao
5 // tekstura na GameObject.
6
7 using UnityEngine;
8
9 public class ServerTex : MonoBehaviour
10 {
11
12     public string url = "https://docs.unity3d.com/uploads/Main/ShadowIntro.png";
13
14     IEnumerator Start()
15     {
16         Texture2D tex;
17         tex = new Texture2D(4, 4, TextureFormat.DXT1, false);
18         WWW www = new WWW(url);
19         yield return www;
20         www.LoadImageIntoTexture(tex);
21         GetComponent<Renderer>().material.mainTexture = tex;
22     }
23 }
24
```

Slika 22. Prikaz koda skripte u programu Visual Studio.

Nakon spremanja skripte u programu Microsoft Visual Studio, Unity *importa* skriptu te provjerava ima li fatalnih pogrešaka u skripti. U slučaju fatalne pogreške skriptu je nemoguće pokrenuti dok se greška ne ispravi.



Kako bi se skripta pridodala objektu u sceni, odvlači se .cs fajl iz Asset prozora u Inspector prozor odabranog objekta kojem se želi pridodati skripta. Kao što je prikazano na slici 23.



Slika 23. Narančasto obrubljen odabrani objekt u Scene prozoru. Desno na slici su njegovi atributi u Inspector prozoru kod kojih se nalazi nova skripta.

Opcije koje su u skripti izložene Unity-ovom programu za uređivanje aplikacija (*eng. editor*) mogu se mijenjati direktno u *editoru* bez mijenjanja same skripte. Kao URL bira se slika na internetu koja će biti prikazana kao tekstura objekta koji na sebi ima skriptu.

### 3.4. POSTAVLJANJE PROSTORA NA INTERNETU ZA DOHVAĆANJE GRAFIČKOG SADRŽAJA

Kako bi u aplikaciji bilo moguće dohvatiti željeni grafički sadržaj s interneta i aplicirati ga kao teksture na modelima, potrebno je omogućiti serverski prostor na kojem će grafički sadržaj biti smješten. Za potrebe ovog diplomskog rada koristi se serverski prostor na Google Drive računu autora rada.

## 4. FINALIZIRANJE

Zadnji dio izrade aplikacije, nakon što je spremna za prvo pokretanje, uključuje testiranje aplikacije, uviđanje i ispravljanje grešaka i prostora za unaprjeđenje.

### 4.1. TESTIRANJE IZRAĐENE APLIKACIJE

Kod prvog testiranja u virtualnoj stvarnosti, dolazi se do zaključka da je prostor prevelik naprema tijelu korisnika. To funkcionira kod tradicionalnih 3d aplikacija gdje je u većini slučajeva kamera postavljena na razini ljudskih prsa, no u *VR-u* se čini kao da je prostorija dizajnirana za divove. Do ove greške je došlo zbog nepoštivanja realnih veličina kod izrade scene u Unity pokretaču. Unity kao mjeru koristi metrički sustav, no program za 3d modeliranje korišten za izradu modela koristi vlastite mjere. Iz tog razloga kod importa modela u pokretač, modeli nisu korektnih fizikalnih veličina. Potrebno je postavljanje referentne kocke od 1 metra četvornog pokraj 3d modela kojem znamo potrebnu veličinu poput stolice. Smanjenjem stolice na pravilnu veličinu naprema kocki i nakon toga smanjenjem svih ostalih modela za isti postotak, dolazi se do scene koja je korektnih fizikalnih veličina naprema korisniku. Svi modeli smanjeni su za 20%.

Drugi problem koji se zamjećuje kod testiranja jest da do izražaja dolazi relativno mala rezolucija *headseta* te je vidljiv svaki pojedini piksel (više nego u drugim *VR* aplikacijama). Iako je ovo hardverski problem, te ga nije moguće u potpunosti ukloniti softverskim putem, donekle ga je moguće riješiti korištenjem *anti-aliasing* post processing efekta opisanim u teorijskom dijelu ovog rada kako bi se ublažila razlika između boja piksela na rubovima 3d modela u sceni.

Nakon ekstenzivnog dodatnog testiranja i rješavanja sitnih vizualnih detalja, aplikacija je spremna za sljedeći korak.

Na slici 24 prikazana je aplikacija u upotrebi s prikazom korisnika i same aplikacije.



Slika 24. Lijevo korisnik s VR uređajem. Desno direktan prikaz aplikacije kroz display headseta.

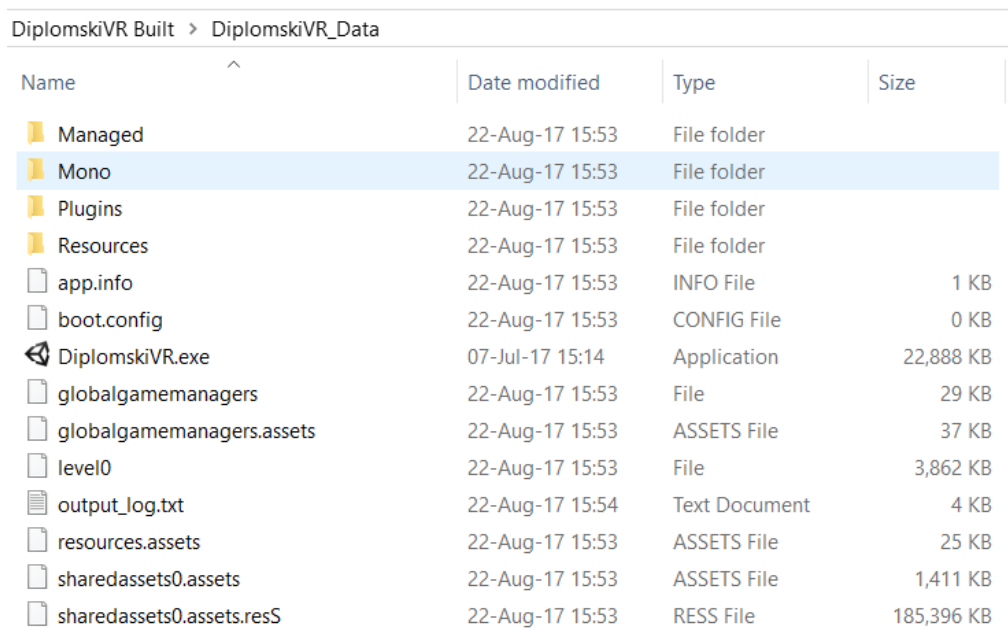
#### 4.1.1. Stvaranje *Builda* aplikacije kako bi bilo moguće pokretanje na drugim računalima

Kako bi aplikaciju bilo moguće pokrenuti bez da računalo na kojem se pokreće ima instaliran pokretač i popratne programe, potrebno je napraviti samostalnu verziju aplikacije (eng. *Build*). *Buildanjem* aplikacije stvara se samostalan (eng. *standalone*) fajl koji se može pokrenuti s računala. U slučaju Windows aplikacije to je .exe fajl, dok se kod Android aplikacije stvara .apk fajl koji je potrebno instalirati na pametni telefon korisnika kroz Google Play ili direktno s memorije pametnog telefona. Svaki operativni sustav i uređaj ima drugačiji sistem *standalone* aplikacija.

U Unity-u *standalone* .exe fajl za Windows kreira se kroz File>Build Settings prozor gdje se određuju postavke stvaranja *standalone* aplikacije. U Player Settings prozoru određuje se ime tvrtke koja proizvodi aplikaciju, ime same aplikacije, ikona, rezolucija u kojoj se pokreće, kompatibilnost s različitim verzijama Windows sustava i druge postavke.

Pritiskom na *Build* bira se prostor na disku gdje će se pohraniti sam .exe fajl i folder s podacima aplikacije. Ako aplikacija nema fatalnih grešaka, *build* prolazi uspješno i stvara se *standalone* aplikacija koju je moguće pokrenuti na drugim

računalima. Fajlovi i datoteke potrebne za distribuciju aplikacije prikazani su na slici 25.



Name	Date modified	Type	Size
Managed	22-Aug-17 15:53	File folder	
Mono	22-Aug-17 15:53	File folder	
Plugins	22-Aug-17 15:53	File folder	
Resources	22-Aug-17 15:53	File folder	
app.info	22-Aug-17 15:53	INFO File	1 KB
boot.config	22-Aug-17 15:53	CONFIG File	0 KB
DiplomskiVR.exe	07-Jul-17 15:14	Application	22,888 KB
globalgamemangers	22-Aug-17 15:53	File	29 KB
globalgamemangers.assets	22-Aug-17 15:53	ASSETS File	37 KB
level0	22-Aug-17 15:53	File	3,862 KB
output_log.txt	22-Aug-17 15:54	Text Document	4 KB
resources.assets	22-Aug-17 15:53	ASSETS File	25 KB
sharedassets0.assets	22-Aug-17 15:53	ASSETS File	1,411 KB
sharedassets0.assets.resS	22-Aug-17 15:53	RESS File	185,396 KB

Slika 25. Fajlovi buildane aplikacije za uključivanje aplikacije i prijenos na drugi uređaj.

## 5. POTENCIJAL INTEGRACIJE IZRAĐENE APLIKACIJE U DRUGE SUSTAVE I UREĐAJE ZA VIRTUALNU STVARNOST

Unity je svestran pokretač koji može *buildati* aplikaciju za mnogo uređaja i sustava poput Windows, OSX (Mac), Linux, Android, iOS i WebGL. Unity također podržava sustave kao što su dedicerane konzole za računalne igre Playstation 4, Xbox One, Nintendo Switch, no za njih je potrebno imati posebnu licencu i poseban uređaj s hardverom sličnim konzoli koji podržava *buildanje* za operativni sustav konzole (*eng. Dev Kit*).

Zbog svestranosti pokretača na kojem je izrađena aplikacija ovog diplomskog rada moguće je uz preinake i adaptaciju aplikacije prenijeti ju na druge sustave i uređaje. Najjednostavnije bilo bi prenijeti aplikaciju na drugi uređaj za virtualnu stvarnost poput Oculus Rift-a zato što bi način kretanja kroz sučelje aplikacije

ostao identičan. Za adaptaciju u tradicionalnu 3d aplikaciju za računala (Windows, Mac, Linux) sličnu računalnim igrama također ne bi bile potrebne prevelike preinake. Način kretanja morao bi se prenijeti na sustav unosa s mišem i tipkovnicom i sučelja bi se morala restrukturirati kako bi bila kompatibilna s tim načinom unosa.

Prebacivanje aplikacije na uređaje sa zaslonom na dodir bilo bi najkompliciranije zbog potpuno drugačijeg principa kretanja i navigacije. Budući da takav sustav unosa osim samog ekrana uređaja nema dedicerane komponente za unos (poput miša, tipkovnice, *controllera* ili *headseta*) potrebno je cijeli sistem navigacije prebaciti i vizualno prikazati na ekranu uređaja te nakon toga omogućiti da aplikacija reagira na korisnikove dodire. Bilo bi potrebno korisnikovu slobodu kretanja unutar virtualnog prostora u aplikaciji ograničiti na predefinirane točke kako bi korisniku bilo lakše koristiti aplikaciju.

Prednost tradicionalnih sustava nad sustavima virtualne stvarnosti jest taj što su puno duže u upotrebi, korisnici su upoznati s načinima unosa (tipkovnica i miš, ekran na dodir) i ulaženje u aplikaciju zahtijeva manje napora. Kod uređaja za virtualnu stvarnost potrebno je *headset* staviti na glavu korisnika, što kod prve generacije uređaja za virtualnu stvarnost još uvijek nije idealno riješeno, te je prosječnom korisniku često odbojno i neugodno stavljati na glavu nešto što mu zaklanja cijeli pogled. Kod tradicionalnih sustava ovi problemi ne postoje i puno je jednostavnije sjesti ili stajati pred uređajem i unositi komande.

## 6. ZAKLJUČAK

Izradom virtualnog izložbenog prostora s dinamički mijenjajućim izložbenim sadržajem prikazan je potencijal virtualne stvarnosti. Virtualna stvarnost budućnost je ljudske interakcije i načina na koji konzumiramo multimedijalne sadržaje.

Objašnjeni su počeci virtualne stvarnosti i iznenadno unaprjeđenje i stabilan rast prihvaćanja iste kao svakodnevnih alternative tradicionalnim sustavima. Kroz sagledavanje problema koji se pojavljuju u VR-u, pronađene su adekvatne alternative kojima je riješen problem mučnine.

Upotrebom moćnog alata Unity izrađen je virtualni izložbeni prostor izgledom temeljen na unutrašnjosti Grafičkog Fakulteta. Korištene su napredne tehnike vizualnog unaprjeđenja i optimizacije kako bi se postigla što kvalitetnija aplikacija. Kroz detaljno testiranje aplikacije uviđene su greške i potrebni popravci, nakon kojih je aplikacija dignuta na ispoliranu razinu spremnu za distribuciju na druga računala. U završnom dijelu rada vidljiv je potencijal adaptacije i proširenja izrađene aplikacije na druge sustave, gdje bi se mogla koristiti u različite svrhe.

Prikazan je samo jedan primjer kako virtualna stvarnost može unaprijediti ili komplementirati postojeće sustave. Primjerice, muzeji i druge ustanove gdje se izlažu grafički sadržaji ovakvu aplikaciju mogli bi imati kao stalnu instalaciju u samom muzeju ili kao servis na internetu koji je alternativa samoj fizičkoj ustanovi. Integracijom mrežnog sustava više korisnika mogli bi vidjeti jedni druge i vršiti interakciju čime bi se proširio društveni aspekt aplikacije.

Virtualna stvarnost nema ograničenja fizikalnog svijeta u kojem se nalazimo pa je moguće nadići funkcionalnosti postojećih muzeja tako da se doda interaktivni aspekt samim izloženim radovima, gdje korisnik može primjerice promijeniti boje ili veličinu radova.

U bliskoj budućnosti virtualna stvarnost će vrlo vjerojatno postati integralan aspekt ljudskih života. Iako mnogi gledaju na virtualnu stvarnost kao na nešto što povećava asocijalnost u društvu, evidentno je iz Facebook-ove akvizicije Oculus-a da virtualna stvarnost zapravo ima potencijal zblížiti ljudsku rasu. Telefon je prvi značajno unaprijedio komunikaciju između ljudi. Nakon toga slijedio je internet uz aplikacije za vizualno-auditivni razgovor poput Microsoftovog Skype-a. Virtualna stvarnost je sljedeći korak u tom smjeru. Umjesto da osobe koje kontaktiramo gledamo kroz ekran pametnog telefona ili računala, nalaziti ćemo se s njima u virtualnom prostoru gdje ih možemo vidjeti u prirodnoj veličini i vršiti interakcije koje do sada nisu bile moguće. Primjerice igrati društvene igre na virtualnom stolu, ili gledati film zajedno u virtualnom kinu.

Već u današnje vrijeme granice između ljudske percepcije virtualnog i stvarnog su sve manje. Igre u proširenoj stvarnosti poput Pokémon GO na pametnim telefonima u stvarni svijet ubacuju virtualne likove. Tehnologija Microsoftovog HoloLens-a prenosi proširenu stvarnost direktno na pogled korisnika. Napretkom ovih tehnologija i izradom sve više aplikacija, granice stvarnog i virtualnog sve više će se smanjivati.

U ne tako dalekoj budućnosti, uz pretpostavku da će se tehnologije potrebne za virtualnu i proširenu stvarnost unaprjeđivati trenutnom brzinom, nakon što se premosti problem simuliranja osjećaja dodira, neće postojati prepreke za stvaranje potpuno realističnog virtualnog prostora kojeg korisnik niti jednim osjetilom neće moći razlikovati od stvarnosti.

Ovaj rad samo je jedan primjer kako virtualna stvarnost može unaprijediti, a u budućnosti i zamijeniti postojeće aspekte ljudskog života.

## 7. LITERATURA

1. Hohstadt T. (2013). *The Age of Virtual Reality*, Lulu
2. <https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game> – *Kickstarter* (2. 8. 2017.)
3. <https://techcrunch.com/2014/07/21/facebooks-acquisition-of-oculus-closes-now-official/> - *Techcrunch* (2. 8. 2017.)
4. Casterson S. (2016). *HTC Vive: A Guide for Beginners*, CreateSpace Independent Publishing Platform
5. <https://www.theverge.com/2017/7/10/15943984/oculus-rift-touch-summer-sale-price-cut-vr-bundle> - *The Verge* (3. 8. 2017.)
6. Nite S. (2014). *Virtual Reality Insider: Guidebook for the VR Industry*, New Dimension Entertainment
7. Fuchs P. (2017). *Virtual Reality Headsets - A Theoretical and Pragmatic Approach*, CRC Press
8. Pharr M., Jakob W., Humphreys G. (2016) *Physically Based Rendering, Third Edition: From Theory to Implementation*, Morgan Kaufmann
9. Akenine-Moller T., Haines E., Hoffman N. (2008) *Real-Time Rendering, Third Edition*, CRC Press
10. Gregory J. (2014). *Game Engine Architecture, Second Edition*, CRC Press
11. <https://assetstore.unity.com/packages/tools/steamvr-plugin-32647> - *Unity Asset store, Steam VR plugin* (7.8. 2017.)
12. <https://assetstore.unity.com/packages/tools/vr-arc-teleporter-61561> - *Unity Asset Store, VR Arc Teleporter plugin* (7. 8. 2017.)



## 8. MANJE POZNATI POJMOVI, AKRONIMI I SKRAĆENICE

*VR* – skraćunica za riječ *Virtual Reality* (virtualna stvarnost).

Aplikacija - računalni program koji omogućuje korisnicima izvršavanje jednog ili više zadataka.

SDK - skup računalnih programa za izradu aplikacija namijenjenih za pokretanje na određenom hardveru ili operativnom sustavu.

3d model - poligonalni prikaz trodimenzionalnog objekta u digitalnoj grafici.

*3d game engine* – pokretač 3d grafike u realnom vremenu.

*Headset* – uređaj za virtualnu stvarnost koji korisnik nosi na glavi.

*Frame* – jedna sličica u sekundi.

Fotogrametrija - tehnika izrade 3d modela obrađivanjem fotografija objekta iz stvarnog svijeta.

Kickstarter kampanja - kampanja za prikupljanje novčanih sredstava za cilj koji je odredio tvorac kampanje.

Rezolucija ekrana - količina individualnih piksela sadržanih u ekranu.

Konfiguracija računala - hardverske karakteristike računala poput vrste procesora, grafičke kartice, količine radne memorije i tvrdog diska.

HDMI - audio/video sučelje za prijenos podataka. Većinom se koristi na za spajanje televizija i računalnih monitora na druge uređaje.

Renderiranje - stvaranje digitalne 2d slike iz 3d informacija na računalu.

Miješana stvarnost (*eng. augmented reality*) - tehnologija koja u stvarni svijet ubacuje računalno generiranu grafiku.

*Post-effect* – efekti u 3d grafici koji se dodaju nakon što je *frame* renderiran.

*Eye tracking* – praćenje smjera pogleda korisnika.

Tekstura - dvodimenzionalna slika koja definira neki aspekt materijala na 3d modelu poput boje ili količine refleksija.

*Plugin* – proširenje za računalni program.

*Google Drive* – servis tvrtke Google koji korisnicima pruža prostor na internetu za *upload* vlastitih datoteka.

*Build* – samostalna verzija aplikacije.

*Dev kit* – uređaj koji omogućuje stvaranje *builda* aplikacije za specifične platforme.

## 9. POPIS SLIKA

Slika 1. HTC Vive headset s controllerima i baznim stanicama za praćenje korisnika u prostoru.....	4
Slika 2. Lijevo na slici vidljiva je scena bez kalkulirane globalne iluminacije, desno je ista ta scena s unaprijed kalkuliranom globalnom iluminacijom.....	8
Slika 3. Scena iz računalne igre Assassin's Creed: Syndicate s isključenim bojama i vidljivim samo SSAO efektom.....	9
Slika 4. Lijevo je prikazan model renderiran u pokretaču Unreal Engine bez bloom efekta, desno s bloom efektom.....	10
Slika 5. Lijevo je prikazan približeni dio renderirane scene s Anti-aliasing efektom. Desno je ista scena bez korištenja efekta. ....	10
Slika 6. Prikaz kromatske aberacije na lijevom gornjem kutu VR displaya.....	11
Slika 7. Bijelim obrubom prikazano je vidno polje korisnika. Bez upotrebe frustum cullinga renderiraju se i objekti koji nisu u vidnom polju. ....	12
Slika 8. Korištenjem frustum cullinga, objekti koji nisu u vidnom polju korisnika izostavljeni su kod renderiranja, čime se uvelike ubrzava renderiranje.....	13
Slika 9. Većinu objekata u vidnom polju korisnika zaklanjaju kocke. Upotrebom occlusion cullinga objekti koji se ne vide ne renderiraju se. ....	14
Slika 10. Lijevo je prikazan model kakav se pojavljuje kad je blizu kamere. Desno je prikazan isti model kakav se pojavljuje kada je daleko od kamera. Primjetno je veliko smanjenje broja poligona od kojih je model izrađen.....	15
Slika 11. Prozor pokretača Unity u kojem se odabire vrsta projekta. ....	16
Slika 12. Standardno sučelje pokretača Unity. ....	17
Slika 13. Inspector prozor pokretača Unity.....	19
Slika 14. Model dodan u scenu. Osvijetljen je Directional Light-om koji je stvoren prilikom stvaranja scene i indirektno osvjetljen difuznom svjetlošću automatski generiranog neba.....	20

Slika 15. Inspector prozor materijala.....	21
Slika 16. Model u sceni projekta s dodanim materijalom.....	22
Slika 17. Camera Rig pozicioniran u sredinu scene.....	23
Slika 18. Prikaz korištenja Arc Teleporter-a unutar aplikacije.....	24
Slika 19. Scena bez postavljenih svjetala i Reflection Probe-ova. Nema izračunatu globalnu iluminaciju, niti sjene u sceni.....	25
Slika 20. Scena s postavljenim komponentama za svjetlost i izračunatom globalnom iluminacijom i sjenama.....	25
Slika 21. Lijevo je prikazano direkcionalno svjetlo koje simulira sunce, na sredini scene je Reflection Probe koji pridodaje refleksije modelima i desno su dva point svjetla koji simuliraju žarulje.....	26
Slika 22. Prikaz koda skripte u programu Visual Studio.....	27
Slika 23. Narančasto obrubljen odabrani objekt u Scene prozoru. Desno na slici su njegovi atributi u Inspector prozoru kod kojih se nalazi nova skripta.....	28
Slika 24. Lijevo korisnik s VR uređajem. Desno direktan prikaz aplikacije kroz display headseta.....	30
Slika 25. Fajlovi buildane aplikacije za uključivanje aplikacije i prijenos na drugi uređaj.....	31