

# Geometrija i optimizacija 3D modela za videoigre

---

Plantić-Petak, Andrej

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:502978>

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

Download date / Datum preuzimanja: **2025-03-11**



Repository / Repozitorij:

[University North Digital Repository](#)





**Sveučilište  
Sjever**

Završni rad br. 932/MM/2024

# **Geometrija i optimizacija 3D modela za videoigre**

**Andrej Plantić Petak, 0336052286**

Varaždin, rujan 2024. godine



**Sveučilište  
Sjever**

**Odjel za Multimediju, oblikovanje i primjenu**

**Završni rad br. 932/MM/2024**

# **Geometrija i optimizacija 3D modela za videoigre**

**Student**

Andrej Plantić Petak

**Mentor**

doc.dr.sc. Andrija Bernik

Varaždin, rujan 2024. godine

## Prijava završnog rada

### Definiranje teme završnog rada i povjerenstva

ODJEL	Odjel za multimediju	
STUDIJ	preddiplomski stručni studij Multimedija, oblikovanje i primjena	
PRISTUPNIK	Andrej Plantić-Petak	NATIČNI BROJ 0336052286
DATUM	20.9.2024.	KOLLOVI 3D modeliranje
NASLOV RADA	Geometrija i optimizacija 3D modela za videoigre	

NASLOV RADA NA ENGL. JEZIKU	Geometry and optimization of 3D models for video games
-----------------------------	--

MENTOR	doc.dr.sc. Andrija Bernik	ZVANJE	Docent
--------	---------------------------	--------	--------

ČLANOVI POVJERENSTVA	
1.	mr.sc. Dragan Matković, v.pred. - predsjednik
2.	mag.rel.publ. Nikola Jozić, pred. - član
3.	doc.dr.sc. Andrija Bernik - mentor
4.	doc.art.dr.sc Robert Geček - zamjenski član
5.	

### Zadatak završnog rada

PROJ

OPIS

Tema završnog rada sastoji se od teorijskog i praktičnog dijela. Cilj teorijskog dijela je objasniti koji alati i postupci se koriste kod optimizacije geometrije kod videoigara, te se na praktičnom primjeru koriste spomenute tehnike koje su potkrijepljene detaljnim objašnjenjem.

Teorijski dio objašnjava osnove 3D modeliranja, njezinu složenost i značaj kod optimizacije. Istražuju se grane optimizacije geometrije koje se danas koriste kod izrade modernih videoigara, te se elementi optimizacije prikazuju na primjerima. Cilj praktičnog dijela je prikaz procesa modeliranja, te upotreba spomenutih načina optimizacije, te njihova implementacija na model.

Kod izrade modela korišteni su program za modeliranje Blender i program specijaliziran za teksture Adobe Substance Painter.

ZADATAK DUBUČEN

20. 09. 2024.



MENTORA

*Bernik*

# Predgovor

Tema završnog rada je detaljan prikaz postupaka kod izrade 3D modela kacige za korištenje u videoigrama, te je detaljno objašnjena geometrija i optimizacije geometrije za videoigre. Također su objašnjene tehnike optimizacije poput retopologije i poligonske redukcije i kreiranja potrebnih mapa.

# Sažetak

Ovaj završni rad obuhvaća cjelokupan proces izrade modela za upotrebu u videoigrama u ovome slučaju 3D model kacige. U radu se objašnjava geometrija modela, njegova optimizacija i utjecaj na performanse u videoigrama. Također se objašnjava optimizacija topologije, korištenje različitih mapa kod teksturiranja modela, kao i optimizacija za *real-time rendering* i na kraju se daju primjeri kvalitetno optimiziranih igara.

**Ključne riječi:** optimizacija, mape, renderiranje, 3D, model, topologija, videoigre

# Summary

This final work covers the entire process of creating a model for use in videogames, in this case a 3D model of a helmet. The paper explains the geometry of the model, its optimization and the impact on performance in video games. It also explains topology optimization, the use of different maps when texturing models, as well as optimization for real-time rendering, and finally examples of well-optimized games are given.

Keywords: optimization, maps, rendering, 3D, model, topology, video games

## Popis korištenih oznaka i kratica

3D	Trodimenzionalnost
LOD	Razina detalja
FPS	Broj sličica u jednoj sekundi
GPU	Grafička procesorska jedinica
CPU	Jedinica za obradu računala
PC	Osobno računalo
AAA	visokobudžetne igre



## Tablica sadržaja

1. Uvod.....	10
1.2. Cilj rada .....	11
1.3. Značaj optimizacije 3D modela u videoigrama .....	11
2. Osnove geometrije u 3D modeliranju .....	12
2.1. Tipovi geometrijskih primitiva .....	13
2.2. Topologija 3D modela .....	13
2.3. Geometrijska složenost i njen utjecaj na performanse .....	14
3. Optimizacija geometrije za videoigre.....	15
3.1. Poligonska redukcija i važnost low-poly modela .....	15
3.1.1. Retopologija: Postupak i alati .....	16
3.2. UV mapiranje .....	19
3.2.1. Diffuse i albedo mape .....	22
3.2.2. Mape za sjaj i hrapavost modela .....	23
3.2.3. Normalne mape za očuvanje detalja .....	24
3.3. LOD – razina detalja .....	25
3.4. Geometrijska optimizacija za real-time rendering.....	25
3.4.1. Occlusion, frustum i backface Culling .....	26
3.4.2. Teselacija i dinamičko prilagođavanje rezolucije.....	27
3.5. Upotreba geometrijskog instanciranja i modifikacija postojećih geometrija .....	28
4. Primjeri optimizirane geometrije u videoigrama.....	29
4.1. AAA naslovi i upotreba optimizirane geometrije .....	29
4.2. Indie igre: Balans između kvalitete i performansi.....	30
5. Blender .....	31
5.1. Sučelje programa Blender.....	31
6. Postupak izrade 3D modela kaciće optimizirane za videoigru.....	32
6.1. Izrada baze kaciće .....	32

6.2. Izrada rogova.....	37
6.3. Izrada krune .....	38
6.4. Izrada ornamenata .....	39
6.5. Izrada detalja.....	40
6.6. Završne radnje.....	41
9. Proces teksturiranja .....	42
9.1. UV odmotavanje .....	42
9.2. Kreiranje <i>normal mape</i> .....	43
9.3 Kreiranje tekstura u programu substance painter .....	44
10. Zaključak.....	45
11. Literatura.....	47
12. Popis slika.....	48

# 1. Uvod

Kod teorijskog dijela završnog rada se objašnjavaju glavni koraci u optimizaciji modela kako bi se koristili u videoigrama.

Loša geometrija u videoigrama pogoršava performanse videoigre zbog ogromne količine nepotrebnih polgiona koji se moraju renderirati, što smanjuje broj sličica u sekundi koje kvare cjelokupan doživljaj kod igrača. Loša optimizacija znači neurednu i neresponzivnu igru koja se neće svidjeti igraču. Kreatori videoigara pokušavaju postići zlatnu sredinu između dobro optimizirane igre i njezine grafike.

Kod praktičnog dijela se modelira kaciga na kojoj će se upotrijebiti neki od navedenih optimizacija geometrije i način teksturiranja, te prikaz i značaj normalnih mapa. Program u kojem se modelira je *blender*, a korišten program za teksturiranje je *substance painter*.

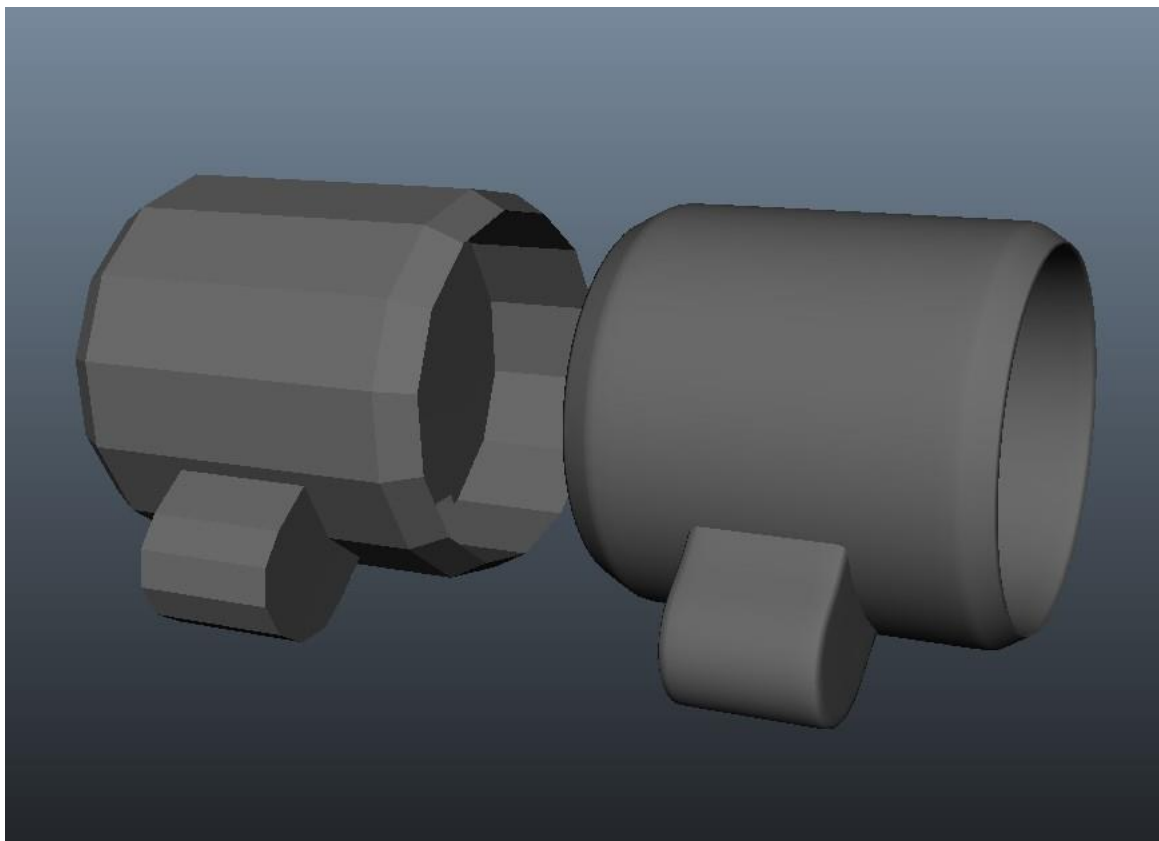
Završnim radom se želi objasniti tehnike optimizacije koje je potrebno upotrijebiti kod modeliranja, te prikazati kako kreirati 3D model spreman za uporabu u videoigrama.

## 1.2. Cilj rada

Cilj ovog završnog rada je objasniti geometrijsku optimizaciju 3D modela, kreiranje različitih mapa, korištenje ostalih tehnika optimizacije modela, utjecaj optimizacije na performanse i prikazati proces izrade 3D modela.

## 1.3. Značaj optimizacije 3D modela u videoigrama

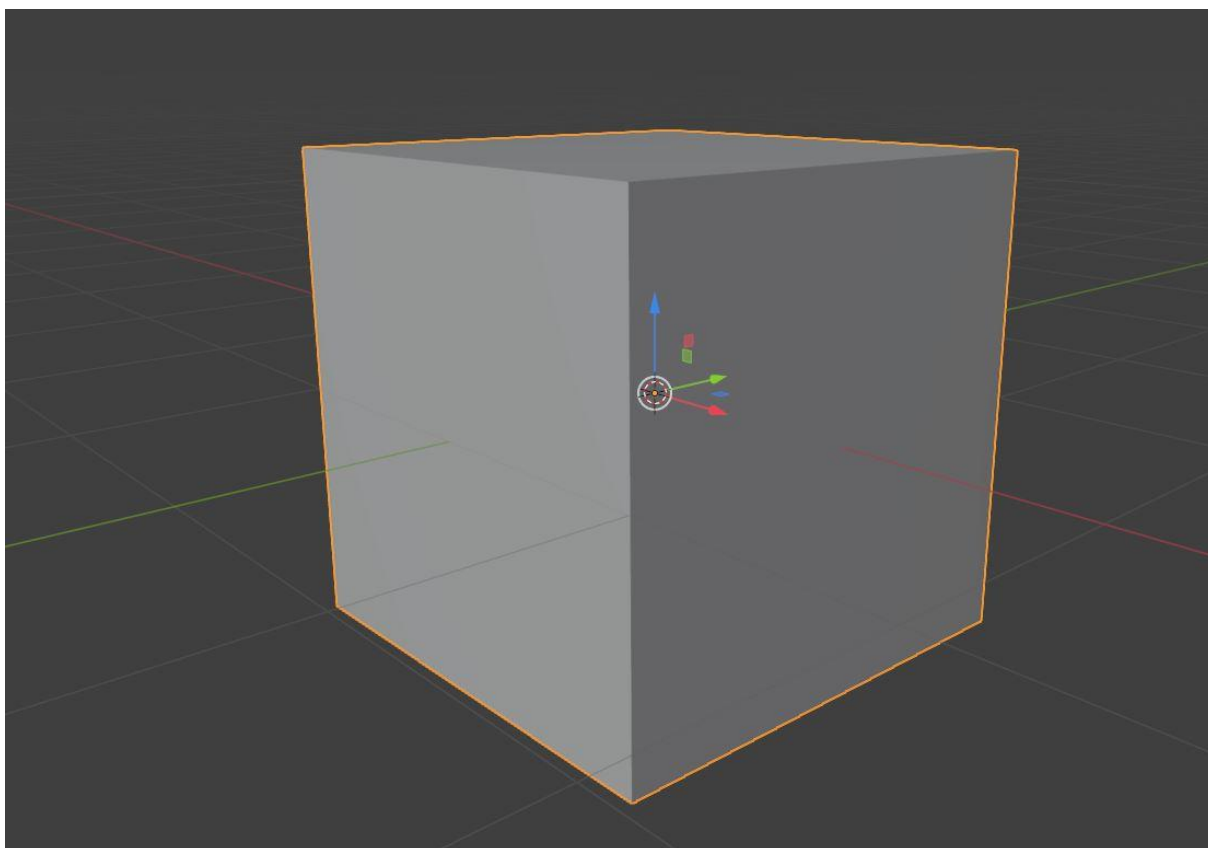
U današnje vrijeme sve više se koristi 3D modeliranje u svakodnevnom životu. Bilo to kod videoigara, filmova, arhitekture ili animacije. Proporcionalno korištenju programa raste i potražnja za različitim modelima, samim time rastu i kompleksnost i mogućnosti programa za 3D modeliranje. Zbog velikih mogućnosti koje nude programi, izrađuju se kompleksniji 3D modeli koji ako nisu pravilno optimizirani specifično kod videoigrama mogu znatno utjecati na kvalitetu igre. Kvaliteta 3D modela se određuje gustoćom poligona koji tvore taj 3D model, u svijetu modeliranja taj termin se naziva *poly-count*. Ako model ima više poligona izgledat će glađe nego li model s manje poligona koji ima više šiljastih rubova. Proporcionalno kreiranju više poligona se povećava količina resursa potrebna da računalo može procesirati taj model. Primjerice ako na sceni postoje 100 različitih ne optimiziranih modela koji imaju veliki *poly-count*, tada je potrebno znatno više resursa da bi se učitala scena. Zato postoje tehnike koje koriste modele s manjim *poly-countom*, a izgledaju isto kao i *high-poly* modeli.



Slika 1.3.1. Razlika između modela s više poligona (desno) i modela s manje poligona (lijevo)

## 2. Osnove geometrije u 3D modeliranju

Geometrija u 3D modeliranju se sastoji od točaka (vertex), bridova (edges) i stranica (faces). Pomoću osnovnih elemenata se manipulira 3D modelom u odabranom programu za 3D modeliranje. Programi za 3D modeliranje za orijentaciju na sceni koriste koordinatni sustav koji koristi osi X,Y,Z, svaka os ima drugačiju boju. Os X je crvene boje, os Z je plave boje i os Y je zelene boje.



Slika 2.1. Primjer koordinatne osi na primitivu kocke

Osnovni alati za manipulaciju su pomicanje (move), rotacija (rotate), skaliranje (scale) i (transform) koji sadrži sva 3 elementa odjedanput. Postoje 2 osnovna moda manipulacije *object mode* i *edit mode*.

Kod *object moda* se upravlja modelom u cjelini, dok u *edit modeu* se mogu manipulirati točke, bridovi i stranice.

Glavne vrste modeliranja geometrije korištene u izradi videoigara su:

Modeliranje čvrstih objekata odnosno (*hard surface modeling*) to je način modeliranja s ciljem ispravnosti geometrijskih površina. Model se izrađuje u potpunosti odnosno model ima svoju masu koja je matematički izračunata. Primjerice u toj kategoriji su modeli oružja, robota, vozila ili arhitekture. [1]

Površinsko modeliranje predstavlja vizualni prikaz vanjskog izgleda modela i njegove površine. Definira vanjske krivulje modela. Za razliku od čvrstog modeliranja, model je geometrijski i fizički neispravan, bez definiranih svojstava mase i bez debljine. Što olakšava modificiranje modela. Međutim, površinski modeli su iznutra šuplji. U toj kategoriji su primjerice modeli čovjeka, drva, biljke, životinje. [1]

## 2.1. Tipovi geometrijskih primitiva

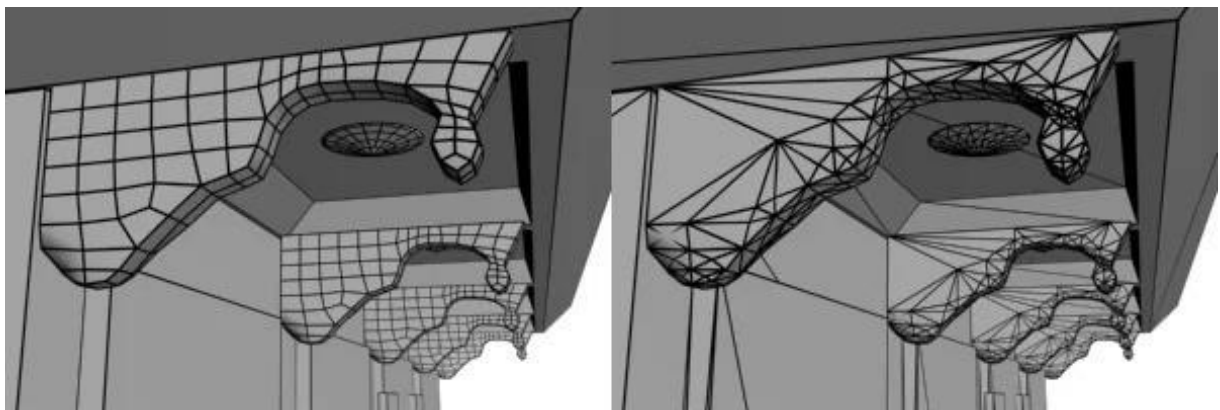
U 3D modeliranju geometrijski primitivi su osnovni oblici koji se koriste kao baza za stvaranje složenijih modela. Različiti softveri za 3D modeliranje nude drugačije skupove primitiva. Najčešće korišteni geometrijski primitivi su: *plain i cube*.

Ploha (*plain*) je jedna stranica, koja se sastoji od četiri točke i četiri brida. Model nije trodimenzionalni jer nema debljinu. Pomoću modela mogu se izraditi podovi, ploče stolova ili ogledala.

Kocka (*cube*) je standardna kocka koja sadrži osam točaka, dvanaest bridova i šest stranica. modeli koji se mogu izraditi od kocke uključuju kutije ili sanduke.[2]

## 2.2. Topologija 3D modela

Topologija je u osnovi izgled 3D modela, odnosno način na koji su točke i bridovi postavljeni da bi se stvorila površina. Kod pravilne topologije žičani pregled modela (*wireframe*) je organiziran i čist. Što znači da je jasan tok i smjer crta koje se protežu kroz *wireframe-a* nekog modela se naziva *edge-flow*. [5]

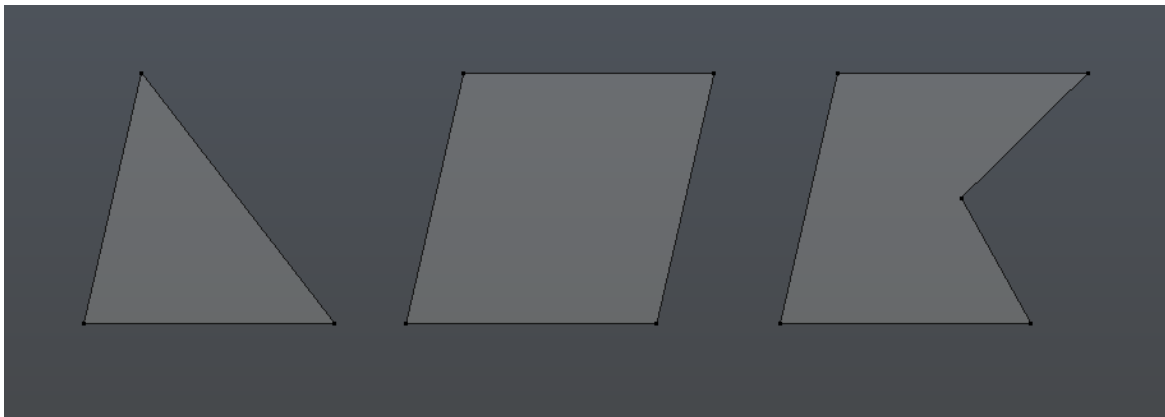


Slika 2.2.1. Razlika između modela s topologijom kvadrata (lijevo) i topologije trokuta (desno)

Postoje 3 vrste topologije kod poligona, to su: trokut, četverokut i n-gon.

Četverokut (quad-based topology) je način izrade modela gdje se pazi da poligoni koji čine model budu četverokuti, odnosno da imaju jednu stranicu, omeđenu s 4 brida. Četverokutasta topologija je preporučljiva kod modeliranja zbog lakšeg popravka modela, te njegove manipulacije.[1]

Trokutasta topologija (triangle-based topology) je način izrade modela gdje se poligoni sastoje od trokuta, odnosno stranice omeđene s 3 brida. Trokutasta topologija se najčešće koristi kod pokretača igara. N-gon topologija sadrži stranicu sa više od 4 bridova. Najzahtjevniji je od navedenih vrsta, nije podržan u svim programima i može se čudno ponašati u pokretačima igara. [4]



Slika 2.2.2. Izgled poligona spomenutih topologija

## 2.3. Geometrijska složenost i njen utjecaj na performanse

Dobra topologija olakšava rad s modelom, minimizira izbočine i moguće neželjene deformacije. Također zbog dobre topologije sam model je optimiziraniji, čime koristi manje resursa i memorije. Topologija je posebno važna kod renderiranja u stvarnom vremenu (real-time rendering), s modelima koji imaju deformirajuće animacije, poput modela likova i s modelima koji imaju visoko reflektirajuće površine, primjerice automobili.

Površine s velikom zakrivljenošću obično zahtijevaju više poligona za podupiranje strukture. Primjerice područje koje zahtijeva izvrsnu topologiju je ljudsko lice. Ako topologija nije dobro izrađena, animacije lica izgledaju neprirodno.

Model s najoptimalnijom poligonalnom strukturom za igre u potpunosti je napravljen od četverokuta. Četverokutasta topologija najpouzdaniji je način modeliranja, zato što poligoni dobro rade s modifikatorima.

Dok se četverokuti koriste kod izrade modela trokutasta topologija služi za rješavanje topoloških problema, primjerice na područjima gdje se model deformira. Zajedničko korištenje više vrsta topologije na jednome modelu se naziva hibridna topologija. Iako je za modeliranje najlakše koristiti četverokutastu topologiju pokretači igara primjerice *unreal engine* prilikom uvoza modela automatski pretvara topologiju modela u optimiziraniju topologiju za pokretače igara kao što je trokutasta topologija.

## 3. Optimizacija geometrije za videoigre

Optimizacija geometrije bitan je aspekt dizajna videoigre jer ima ključnu ulogu u određivanju performanse i vizualne kvalitete igre. Korištenje optimizirane geometrije može uvelike poboljšati iskustvo igrača pružanjem glatke i brze igre, kao i realistične i detaljne grafike.

Jedna od ključnih prednosti korištenja optimiziranih geometrija u dizajnu videoigara je poboljšanje performansi. Broj poligona koje je potrebno prikazati na zaslonu u bilo kojem trenutku može biti prilično velik. Smanjenjem broja poligona i pojednostavljivanjem oblika dobiva se na boljoj performansi videoigre.

Drugi važan aspekt optimizacije geometrije je vizualna kvaliteta. Pozornost igrača često je usmjerena na likove, okolinu i druge objekte u svijetu igre. Objekti moraju biti detaljni i uvjerljivi kako bi u potpunosti uronili igrača u igru. Optimiziranjem geometrije objekata, omogućuje se učinkovitiji prikaz detalja i održavanja vizualnih detalja.

### 3.1. Poligonska redukcija i važnost low-poly modela

Današnje videoigre sadrže ogromnu količinu modela na sceni koji se sastoje od velike količine poligona. Kako bi se scena optimizirala i omogućila prikaz više objekata odjednom, potrebno je koristiti metodu poligonske redukcije. Poligonska redukcija je način optimizacije modela u kojoj se smanjuje broj poligona određenog modela kako bi se povećala ukupna performansa. Jedna od najkorištenijih tehnika poligonske redukcije je retopologija. Videoigre koje su izrađene s velikim budžetom često kako bi prikazale svoj raskoš vrlo detaljnih i visoko kvalitetnih scena moraju kod izrade svojih modela koristiti njezine *low-poly* verzije. Svrha *low-poly* modela je da se na njih mogu preslikati detalji *high-poly* modela, gdje se u procesu ne gubi na kvaliteti.

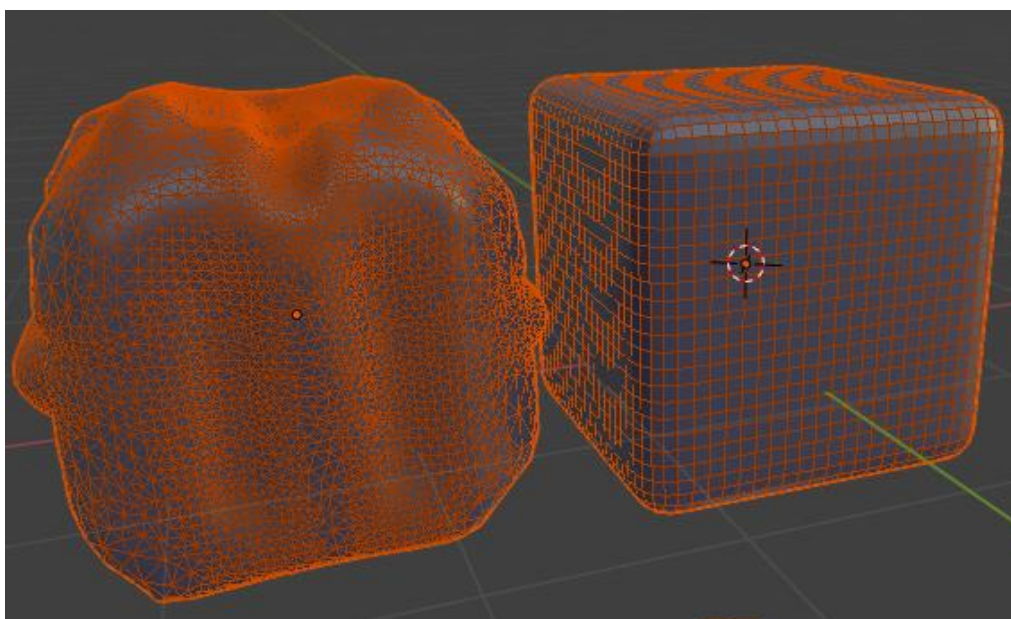


### 3.1.1. Retopologija: Postupak i alati

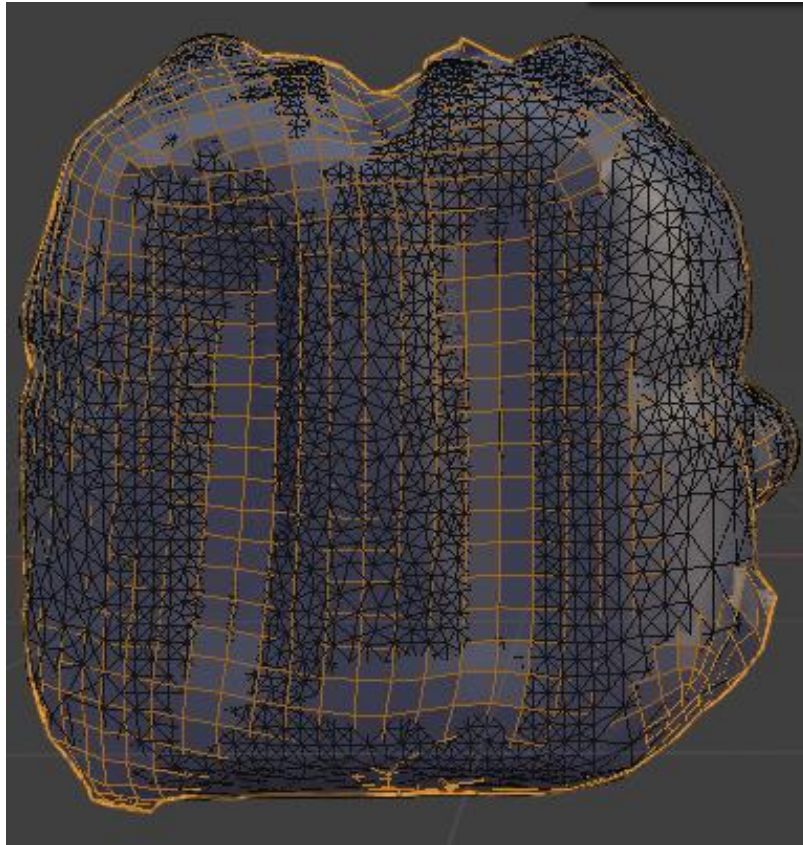
Retopologija se koristi kod promjene topologije 3D modela, s ciljem smanjenja ukupnog broja poligona. Retopologija se najčešće koristi kada se optimizira model izrađen tehnikom kiparstva ili 3D skeniranjem. Tehnikama kiparstva i 3D skeniranja se ne može direktno utjecati na broj poligona tokom izrade. Zato što se kod kiparstva koristi kist kojim se oblikuje model, a poligoni se matematički izračunavaju. Kod 3D skeniranja kada se skenirani model ubaci u program za modeliranje, najčešće se sastoji od velikog broja poligona, te je samim time teško naknadno manipulirati takvim modelom. Retopologija se može napraviti ručno manipuliranjem geometrije ili putem automatiziranih metoda. [4]

Jedna od tehnika retopologije je *Remeshing* što je tehnika koja automatski ponovno gradi geometriju s ujednačenijom topologijom. Nudi mogućnosti odabira topologije, količinu poligona i njihovo usmjerenje. Može se dodavati ili brisati količina poligona odnosno o definiranoj rezoluciji.

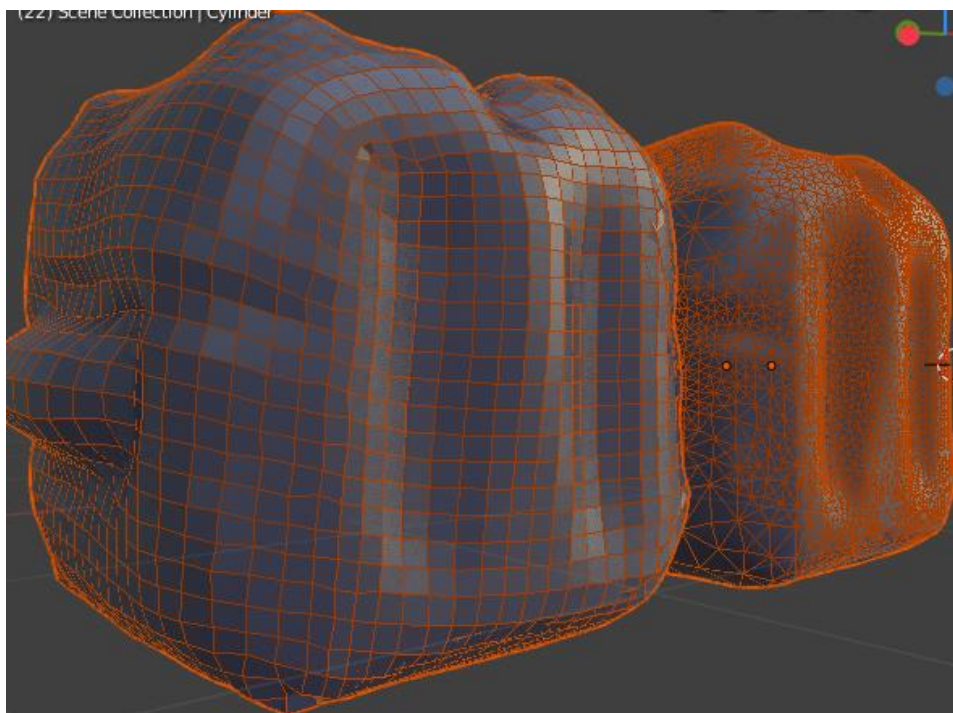
*Softwrap* je tehnika koja nam omogućuje da omotamo jednostavni model oko kompliciranijeg modela. Postupak se sastoji od kreiranja jednostavnijeg modela s manje poligona, njegovom manipulacijom da izgleda kao i kompliciraniji model, te jednostavniji model omotamo oko kompliciranijeg. Finalni produkt je model istog oblika s manjom, i ljepše posloženom topologijom.



Slika 3.1.1.1. Prikaz modela s puno poligona (lijevo) i modela s manjim brojem poligona (desno)

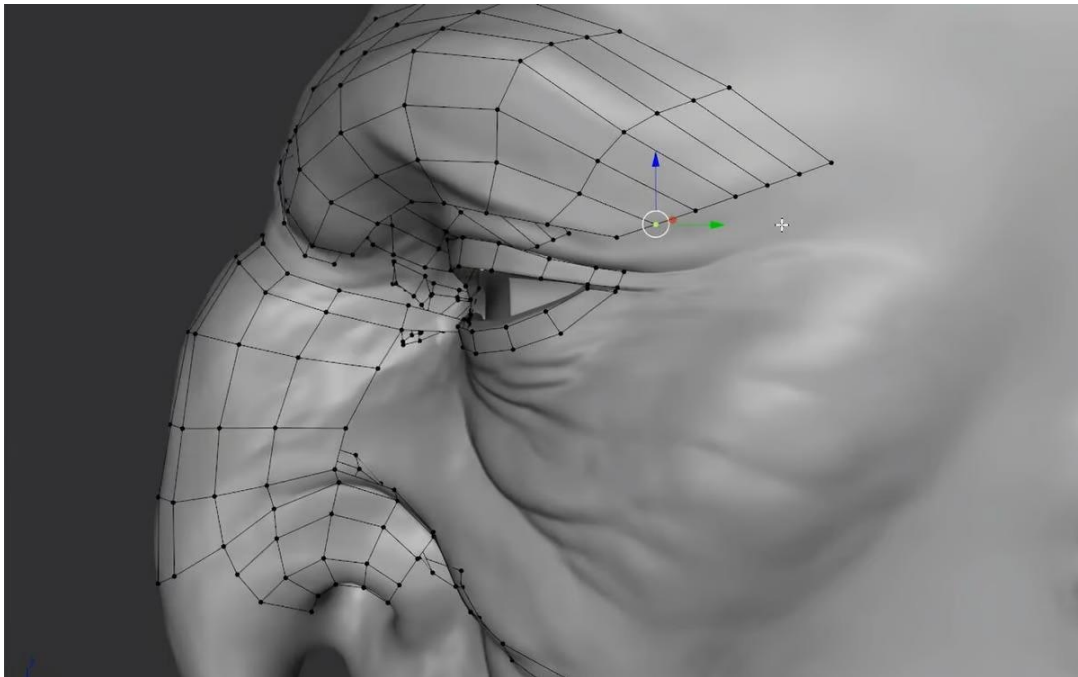


*Slika 3.1.1.2. Prikaz stapanja između 2 modela, gdje se pokušava replicirati jednak izgled jednostavnog modela naspram kompliciranijeg modela*



*Slika 3.1.1.3. Razlika između modela nakon korištenja softwrap-a*

*Retopo MT* je tehnika retopologije koja omogućuje ručno „crtanje“ poligona. Koristi crtanje kvadratnom topologijom, manipuliranjem točaka poligona, te njihovim spajanjem.



Slika 3.1.1.4. Prikaz crtanja topologije na modelu

*Quadwrap* je poluautomatska tehnika koja se koristi kod jednostavnog kreiranja topologije kod modela oblika cilindra, gdje precizno tehnikom *click and drag* automatski iscrta topologiju oko modela.

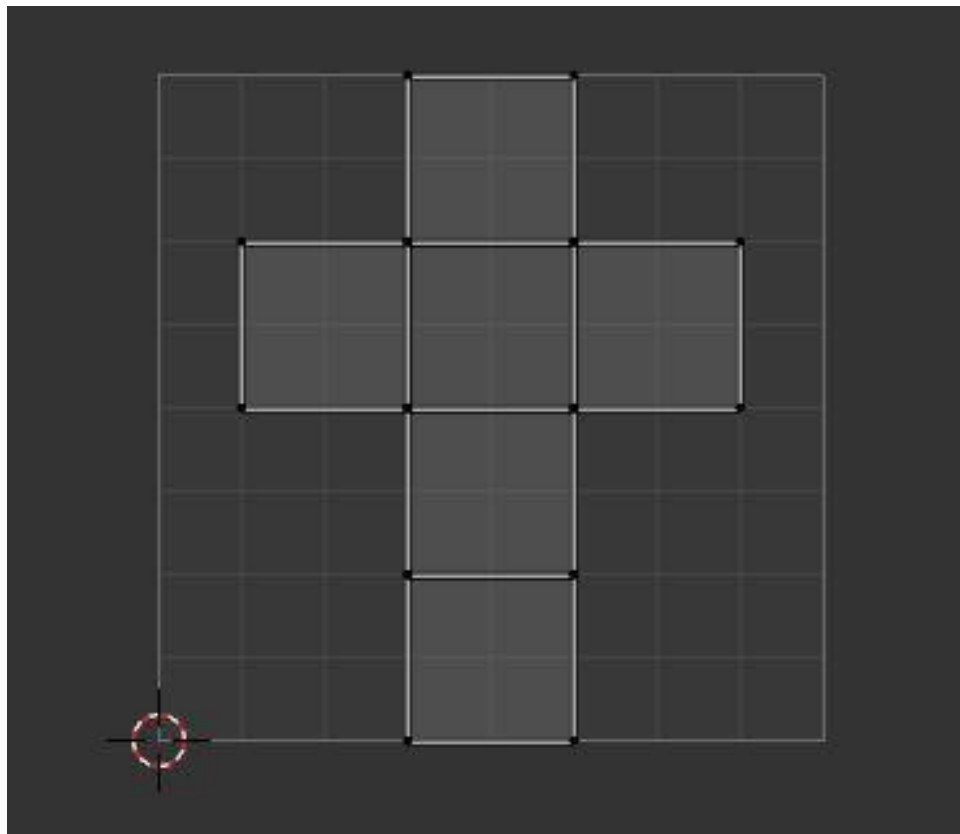


3.1.1.5. Prikaz poluautomatskog crtanja topologije na modelu

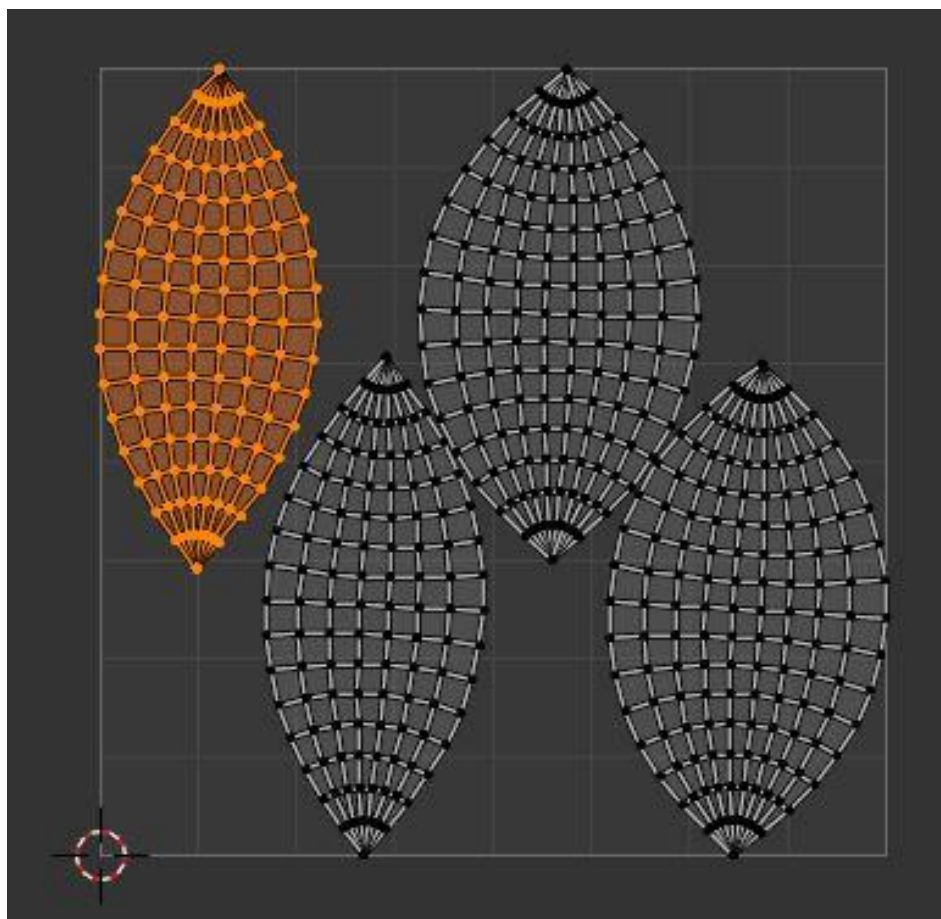
## 3.2. UV mapiranje

UV mapiranje je proces 3D modeliranja gdje se 3D model projicira na 2D podlogu. Kada se model projicira na dvodimenzionalnu podlogu, svaki poligon, brid i vrh se prenose na UV kartu. Tekstura se zatim postavlja na UV kartu, na temelju postavljenih neomotanih poligona. Pokretači igara koriste dvije koordinate teksture za mapiranje širine i visine teksture. U se koristi za širinu, V se koristi za visinu. Koordinate teksture temelje se u mjerilu od 0,0 do 1,0, pri čemu je 0,5 središnja točka. Kada model ima UV ljestvicu udaljenosti veću od 1, tekstura se tada raspoređuje preko modela. UV karte se sastoje od UV ljski ili UV otoka, oboje imaju isto značenje. Kada se model odmota, obično se razbije na nekoliko dijelova.

Postoje mnoge metode za odmotavanje modela na UV karti. Primjerice, aktivni poligoni, bridovi i točke mogu se projicirati na UV kartu na temelju *viewport-a* odnosno prozora programa. Ova se metoda naziva planarnim mapiranjem i najbolje funkcionira za objekte koji su relativno ravni. Okrugli objekti obično imaju iskrivljene UV karte, čim je površina manje okomita u odnosu na kameru, to se manje vidi na UV karti. Ako se cijeli model odmota planarnom metodom, odmotat će se i poligoni koji nisu vidljivi kameri, što uzrokuje preklapanje UV ljski. Savršeno preklapajuće UV ljske mogu se koristiti za simetrične modele kako bi se uštedio UV prostor, ali slučajna preklapanja UV ljski obično uzrokuju probleme u fazi teksturiranja.[4]



Slika 3.2.1. Prikaz potpuno pravilne i ravne preslike kocke na UV mapu

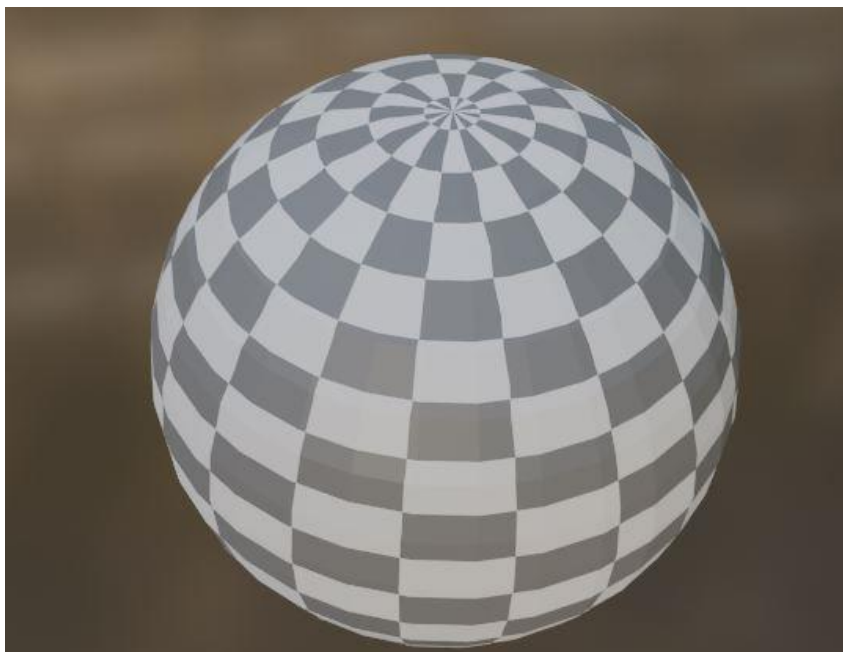


Slika 3.2.2. Prikaz neravne UV mape kugle

Druga metoda je ručno označavanje UV šavova u 3D softveru. *Wireframe* se odvaja po šavovima. Ručno postavljanje šavova poželjno je za složene modele jer daje veću kontrolu nad postupkom odmatanja. Kod ove metode najbolje je paziti na količinu istezanja u UV karti. Šavovi se mogu lako vidjeti na teksturama. Šavove je najbolje postaviti na mjesta koja se ne mogu vidjeti u igri. Nakon što se cijeli model odmota, potrebno je provjeriti veličinu UV otoka nanošenjem karirane teksture na model. Ova tekstura otkriva potencijalno rastezanje i neželjene šavove. [4]



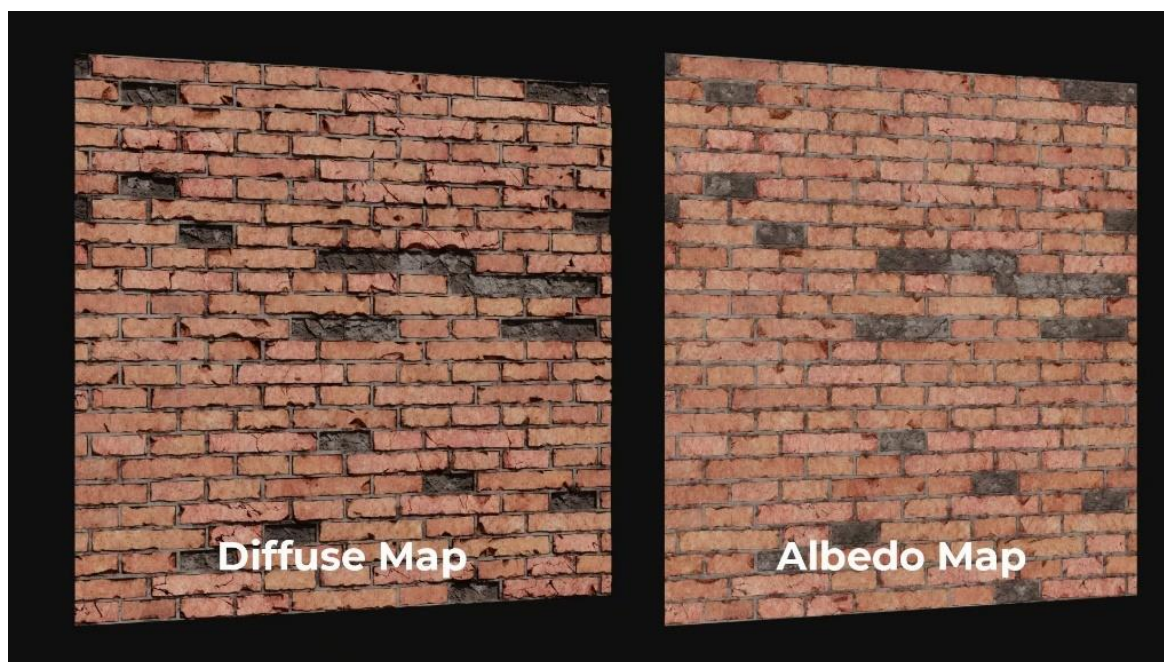
*Slika 3.2.3. Prikaz karirane teksture na modelu kugle gdje se vide izrađeni šavovi*



*Slika 3.2.4. Prikaz karirane teksture na modelu kugle bez dodavanja šavova (vide se rastezanja teksture)*

### 3.2.1. Diffuse i albedo mape

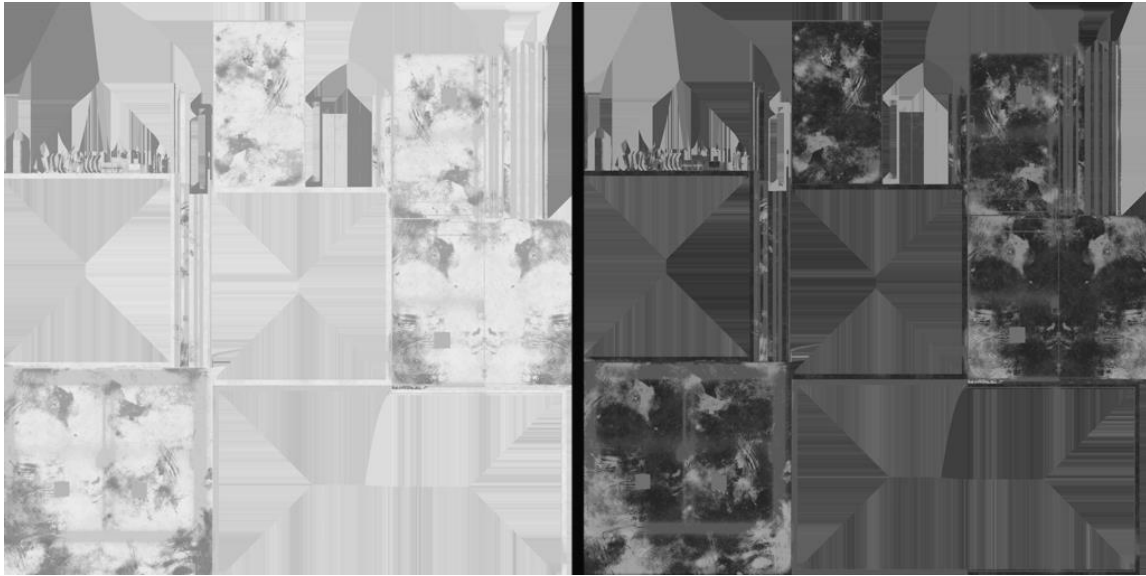
Za utjecanje na boju modela postoje dvije mogućnosti, karta difuzne teksture i karta albedo teksture. Razlika između korištenja difuzne karte ili albedo karte je uklanjanje ambijentalne okluzije i usmjerenog svjetla u albedo kartama. Ambijentalna okluzija je tehnika koja približno određuje količinu sijena. Na kartama albeda nema svjetla ili sjena, što rezultira preciznijim teksturiranjem. [4]



*Slika 3.2.1.1. Razlika teksture zida između diffuse i albedo mape*

### 3.2.2. Mape za sjaj i hrapavost modela

Hrapavost površine, obično se kontrolira pomoću mapa hrapavosti ili sjaja. Obje su karte u sivim tonovima, s vrijednostima u rasponu između 0 i 1. Mapa hrapavosti diktira koliko je površina hrapava, a čim je površina hrapavija, to manje reflektira. Bijela područja na karti označavaju hrapavu površinu, dok tamniji tonovi označavaju glatke i sjajnije površine i obrnuto.

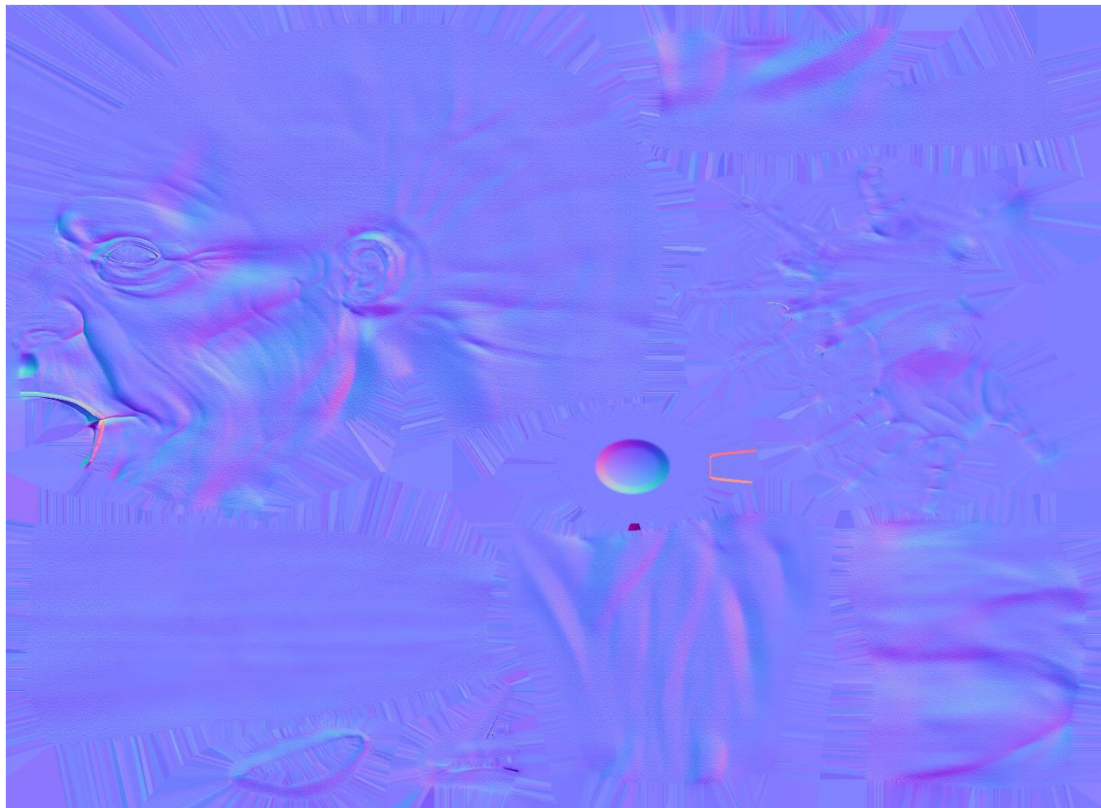


*Slika 3.2.2.1. Mapa hrapavosti teksture*



### 3.2.3. Normalne mape za očuvanje detalja

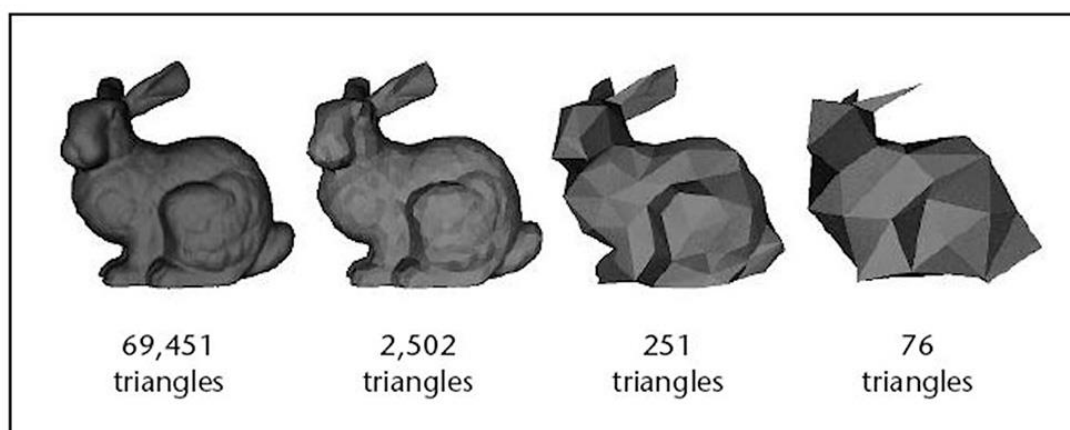
Karta normala se koristi kako bi se lažirali detalji na modelu. Mapa normala koristi RGB kanale za promjenu normala, a svaki kanal odgovara x, y i z-osi, odnosno različite boje u normalnim kartama oponašaju neravnine i šupljine. Kako je normalne karte vrlo teško izraditi ručno, često se izrađuju prijenosom detalja s vrlo detaljnog modela na model s manjim brojem poligona „pečenjem“. Pečenje (*texture baking*) je izraz koji se koristi za spremanje detalja na mapi teksture koji se zatim prenose na model.



Slika 3.2.3.1. Prikaz normal mape bora na čovjekovom licu

### 3.3. LOD – razina detalja

Razina detalja (*Level Of Detail*) odnosi se na razinu složenosti 3D modela i prvenstveno se koristi u renderiranju u stvarnom vremenu za video igre. Pomaže smanjiti količinu detalja pojednostavljivanjem poligona i tekstura kako se udaljavaju od kamere. LOD pomaže u upravljanju računalnim zahtjevima renderiranja složenih scena, omogućujući učinkovitiju izvedbu uz očuvanje vizualne kvalitete. Primjerice kada se objekt promatra izdaleka, možda će morati biti vidljive samo osnovne značajke kao što su boja i geometrija. Općenito, što je bliže kameri, bit će potrebna veća razina detalja. [6]



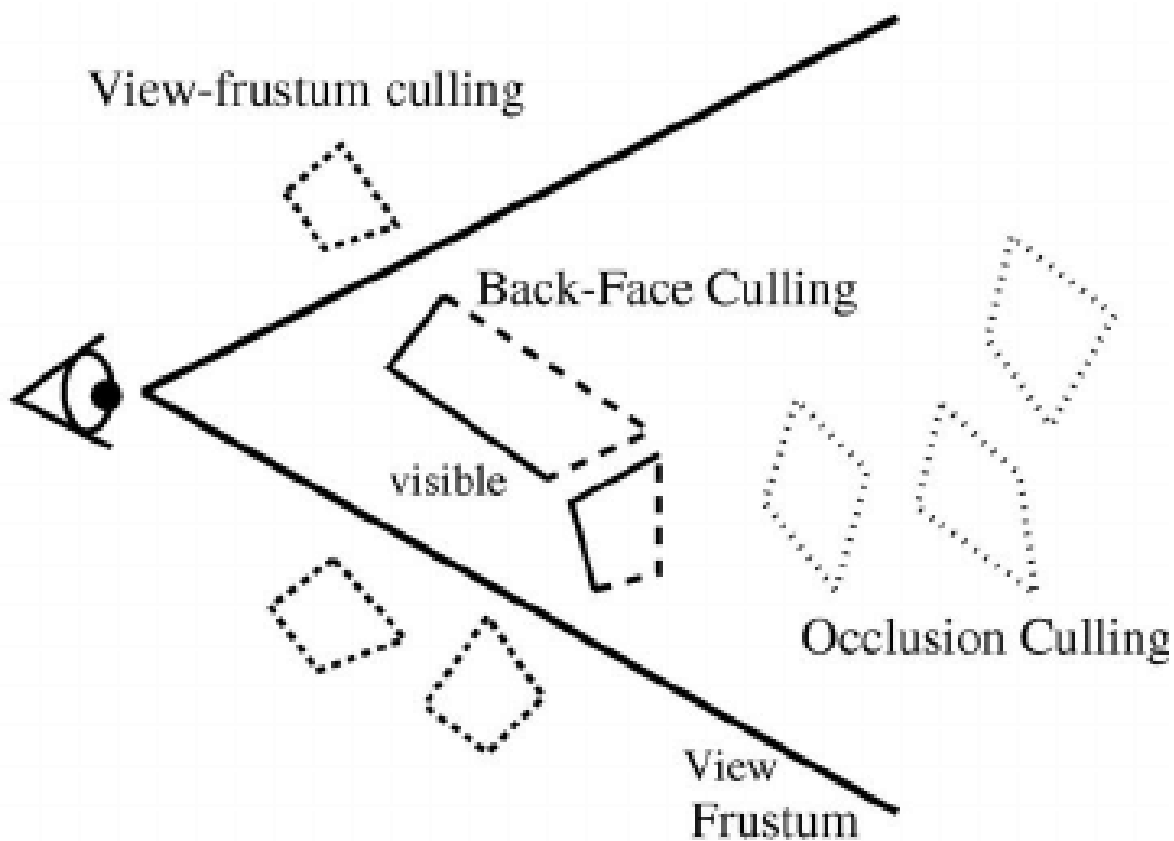
Slika 3.3.1. Prikaz LOD-a (kamera je blizu na lijevom primjeru, dok na desnom kamera najdalje od objekta)

### 3.4. Geometrijska optimizacija za real-time rendering

Koncept renderiranja je automatsko generiranje fotorealistične slike. Stopa generiranja je dovoljno brza da slijed slika izgleda kao animacija koja nešto simulira. Ciklus renderiranja odvija se dovoljno velikom brzinom da promatrač ne vidi pojedinačne slike, već uranja u dinamički proces. Brzina kojom se slike prikazuju mjeri se u sličicama u sekundi (*Frames Per Second*). Na oko 6 FPS-a, osjećaj interaktivnosti počinje rasti, 24 FPS-a bi moglo biti prihvatljivo i svakako je u stvarnom vremenu. Video igre ciljaju na 30, 60, 72 ili više FPS-a. Od oko 72 FPS-a i više, ljudsko oko ne može otkriti nikakve razlike u brzini prikaza.

### 3.4.1. Occlusion, frustum i backface Culling

*Occlusion Culling* je tehnika koja onemogućuje renderiranje objekata kada ih kamera trenutno ne vidi jer su zaklonjeni drugim objektima. Objekti koji su najudaljeniji od kamere se crtaju prvi, a bliži objekti se crtaju preko njih, ta tehnika se naziva „*overdraw*“, odnosno precrtavanje. *Frustum Culling* onemogućuje renderiranje objekata koji se nalaze izvan vidokruga kamere. *Backface culling* Onemogućuje renderiranje objekata čiji su poligoni obrnuto okrenuti od usmjerenja kamere. [7]



Slika 3.4.1.1. Prikaz Culling tehnika

### 3.4.2. Teselacija i dinamičko prilagođavanje rezolucije

Teselacija je tehnika koja uključuje rastavljanje složenih površina na manje dijelove kojima se lakše upravlja. Omogućuje detaljniji prikaz 3D objekata dijeleći ih na manje geometrijske komponente, kao što su trokuti ili četverokuti, odnosno teselacija povećava razinu detalja u grafici dodavanjem više poligona. Primarni utjecaj teselacije na grafiku u igrama je razina detalja koju dodaje okruženjima i likovima u igri. Rastavljanjem površina na manje komponente, teselacija omogućuje glađe krivulje, zamršene teksture i realistične oblike. Teselacija je posebno učinkovita u prikazivanju složenih površina, kao što su krajolici, voda i organske strukture. Primjerice u igri koja prikazuje planinski teren, teselacija se može upotrijebiti za dodavanje dodatnih detalja stijenama i liticama, stvarajući impresivniji i vizualno privlačniji krajolik.



*Slika 3.4.2.1. Prikaz razlike teksture poda s teselacijom (desno) i bez teselacije (lijevo)*

Dinamička razlučivost prilagođava kvalitetu prikaza broja piksela na zaslonu. Rezolucija se prilagođava kada ima previše objekata na zaslonu. Kako bi se kod renderiranja scene FPS stabilno održavao na željenoj razini, proporcionalno se smanjuje rezolucija zaslona kako bi se postigli željeni rezultati.



*Slika 3.4.2.2. Prikaz promjene rezolucije kod dinamične razlučivosti*

### 3.5. Upotreba geometrijskog instanciranja i modifikacija postojećih geometrija

Geometrijsko instanciranje je prikazivanje mnogo malih objekata, često svaki sličan sljedećem, sa samo malim razlikama kao što su boja, položaj i orijentacija. Primjerice svako stablo u velikoj šumi može biti geometrijski vrlo slično, ali se mogu razlikovati po boji ili visini. Međutim, renderiranje velikog broja malih objekata, od kojih je svaki napravljen od nekoliko poligona, predstavlja veliko opterećenje za današnje GPU-ove. Cilj instanciranja je minimizirati broj promjena stanja i teksture i prikaz istih poligona više puta unutar iste serije u jednom pozivu. Samim time se minimizira vrijeme CPU-a potrošeno na slanje i oslobađa se vrijeme da CPU može raditi druge zadatke.

## 4. Primjeri optimizirane geometrije u videoigrama

Današnje videoigre sve su realističnije. Tehnologija vezana uz videoigre toliko je napredovala, da scene iz videoigara izgledaju kao da su slikane kamerom u stvarnome životu. Doduše za pokretanje takvih igara potrebno je imati PC ili igraču konzolu koja je u stanju pokretati takve igre. U dizajnu videoigara optimizacija je jedna od najvažnijih elemenata kod kreiranja kvalitetnog igraćeg iskustva.

### 4.1. AAA naslovi i upotreba optimizirane geometrije

*Triple A* naslovi su videoigre koje su nastale uz veliki budžet, te ciljaju na što boljim grafikama kako bi prikazale svu raskoš svojeg virtualnog svijeta. Dobar primjer optimizirane AAA igre je *Metal Gear Solid V*. Igra je izašla 2015. godine i prima puno pohvala o svojem optimiziranom svijetu. Primjerice igra je optimizirana tako što su svi modeli low-poly, a refleksije modela su se nadomjestile pametno stiliziranom teksturom koja oponaša refleksiju. Videoigra je zapravo sastavljena od modela, tekstura efekata niske kvalitete koji su odlično zamaskirani tehnikama optimizacije.



Slika 4.1.1. Primjer scene iz igre Metal Gear Solid V



Slika 4.1.2. Primjer scene iz igre Metal Gear Solid V

## 4.2. Indie igre: Balans između kvalitete i performansi

Indie igre je naziv za videoigre koje su izrađene od strane manjih timova ljudi, većinom s manjim budžetom. Cilj takvih igara nije prezentacija grafičkih elementa, već se više fokusira na priču, *gameplay* i svaka indie igra ima svoj stil po kojem je kreirana. Jedna od vrlo popularnih indie igra je Factorio. Čiji se koncept svodi na izradu i održavanje tvornica. Factorio odlično koristi tehniku instanciranje gdje se isti modeli i procesi ponavljaju, pa se samim time uvelike dobije na performansi.



Slika 4.2.1. Primjer scene iz igre Factorio

## 5. Blender

Blender je besplatan program za 3D modeliranje otvorenog koda. Neke od mogućnosti su mu: modeliranje, *rigging*, animacija, simulacija, renderiranje, sastavljanje i *motion tracking*. Blender je vrlo prikladan za pojedince koji se tek opuštaju u svijet 3D modeliranja.

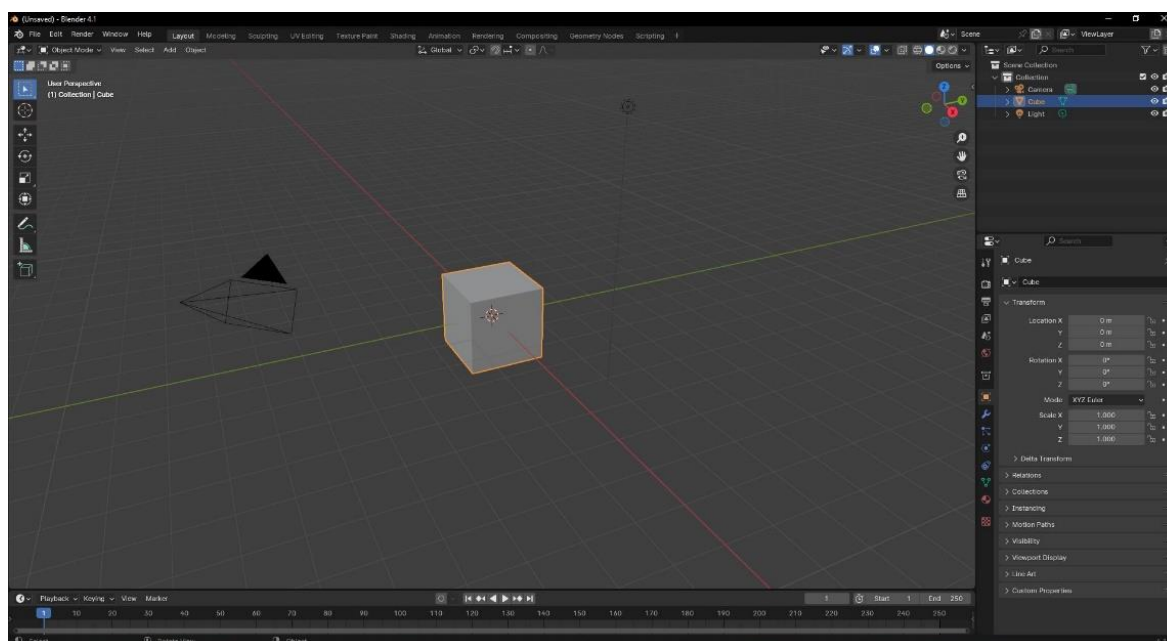
### 5.1. Sučelje programa Blender

Blenderovo sučelje je podijeljeno u tri glavna dijela:

Gornja traka na samom vrhu sastoji se od glavnog izbornika koji se koristi za spremanje, uvoz i izvoz datoteka, konfiguriranje postavki i renderiranje.

Glavni radni prostor se sastoji od statusne trake na dnu, koja prikazuje prijedloge prečaca i relevantne statistike.

Radni prostori su u biti unaprijed definirani raspoređeni prozora. Svaki radni prostor sastoji se od skupa područja koja sadrže uređivače koji su namijenjeni za obavljanje određenih zadatak kao što su modeliranje, animiranje ili skriptiranje.



Slika 5.1.1. Sučelje programa Blender

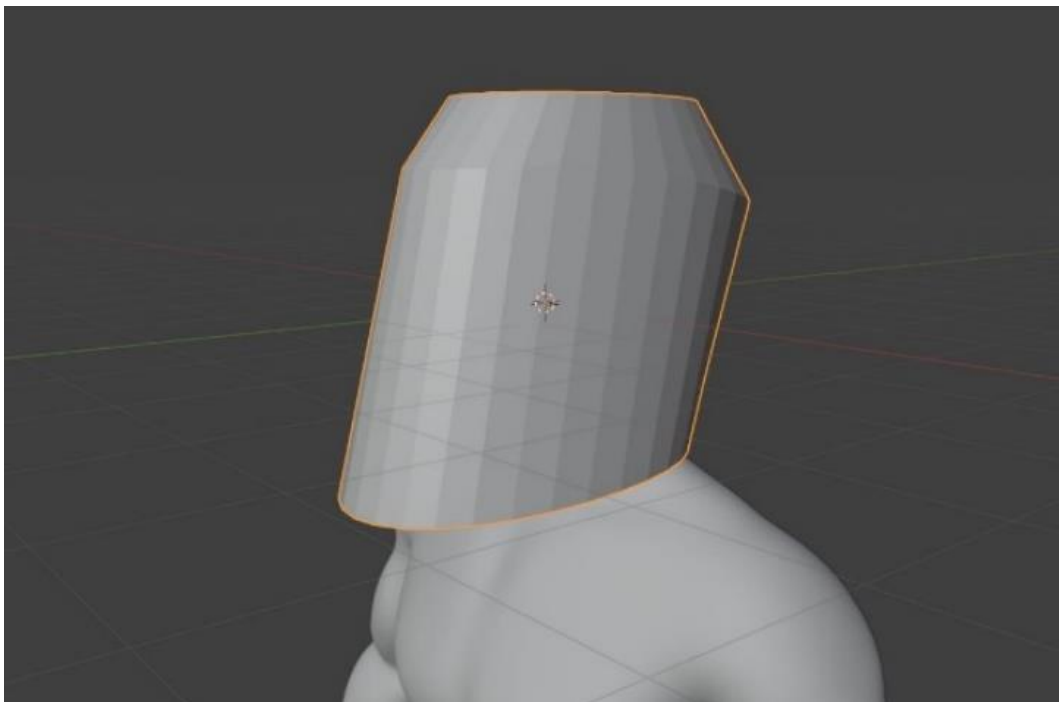


## 6. Postupak izrade 3D modela kacige optimizirane za videoigru

Praktični dio rada se bavi s opisivanjem koraka koji su potrebni da se izradi 3D model kacige. Sam proces zahtjeva poznavanje programa Blender, te poznavanje kako optimizirati model za korištenje u videoigrama.

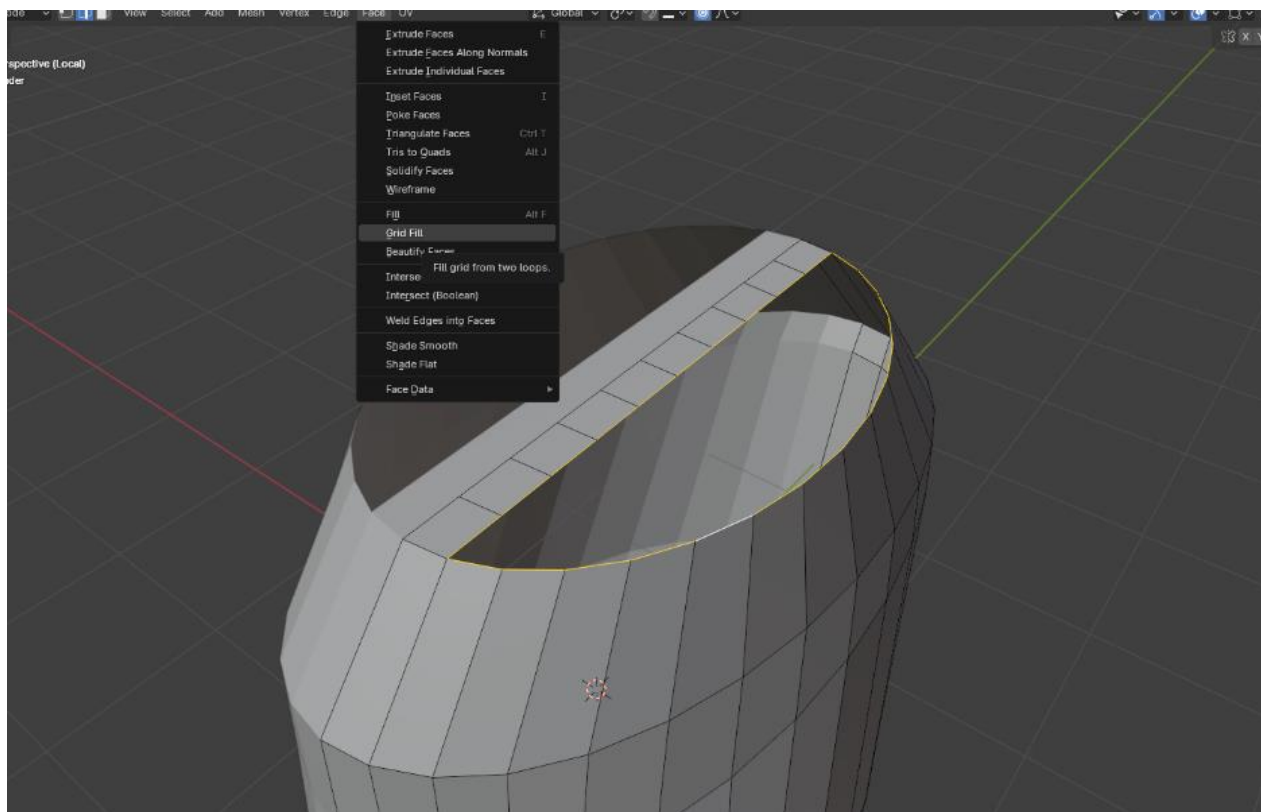
### 6.1. Izrada baze kacige

Izrada kacige počinje komandom `shift+A`, te biranjem geometrijskom tijela cilindra. Za referencu potrebnu u oblikovanju i skaliranju kacige potrebno je uvesti 3d model čovjeka. U programu blender za manipulaciju modela postoji glavni alati poput: *size*, *position* i *rotation*, odnosno promjena veličine, pozicije i rotacije objekta. Unutar *edit mode* prozora u programu blender može se manipulirati točkama (*vertexes*), bridovima (*edges*) i stranicama (*faces*). Nakon skaliranja potrebno je potvrditi promijene, kako bi se izbjegli budući problemi s nekim alatima, klikom na gumb *apply*.



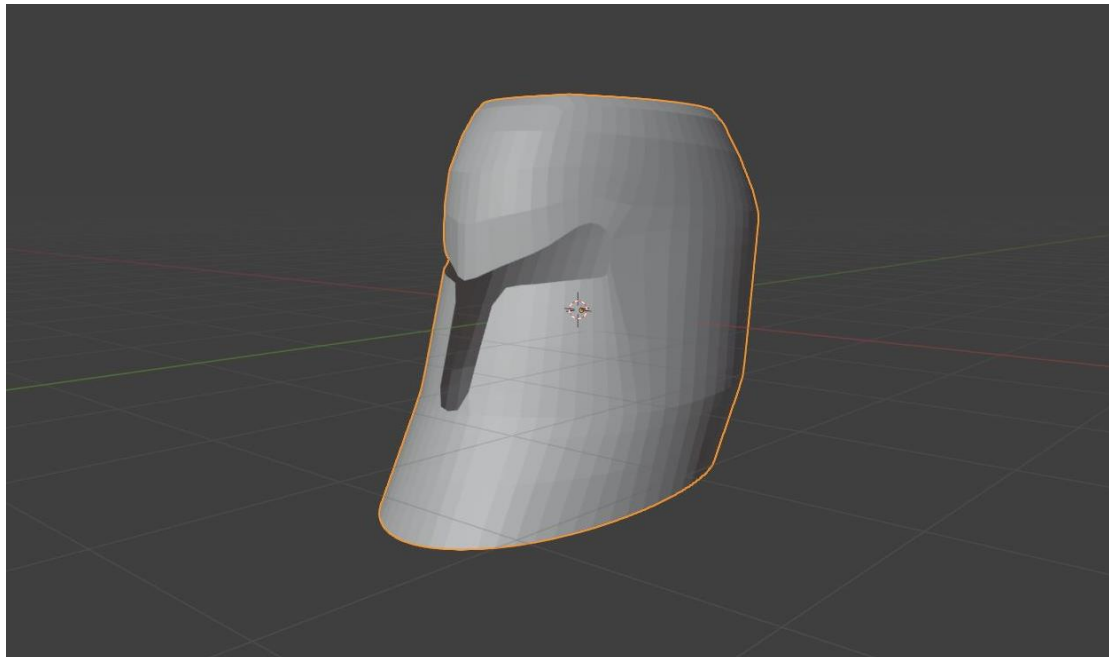
Slika 6.1.1. Dovršena baza kacige

Nakon početnog skaliranja u daljnjem procesu potrebno je koristiti naprednije alate poput funkcije *mirror* koja omogućuje podjelu modela na 2 jednaka dijela i omogućuje manipuliranje na obje strane odjednom, poput ogledala. Zatim se izbriše gornji dio kacige, te se stvori „most“ između dva kraja kacige kako bi se pomoću alata *grid fill* brže popunio dio bez *faceva*.



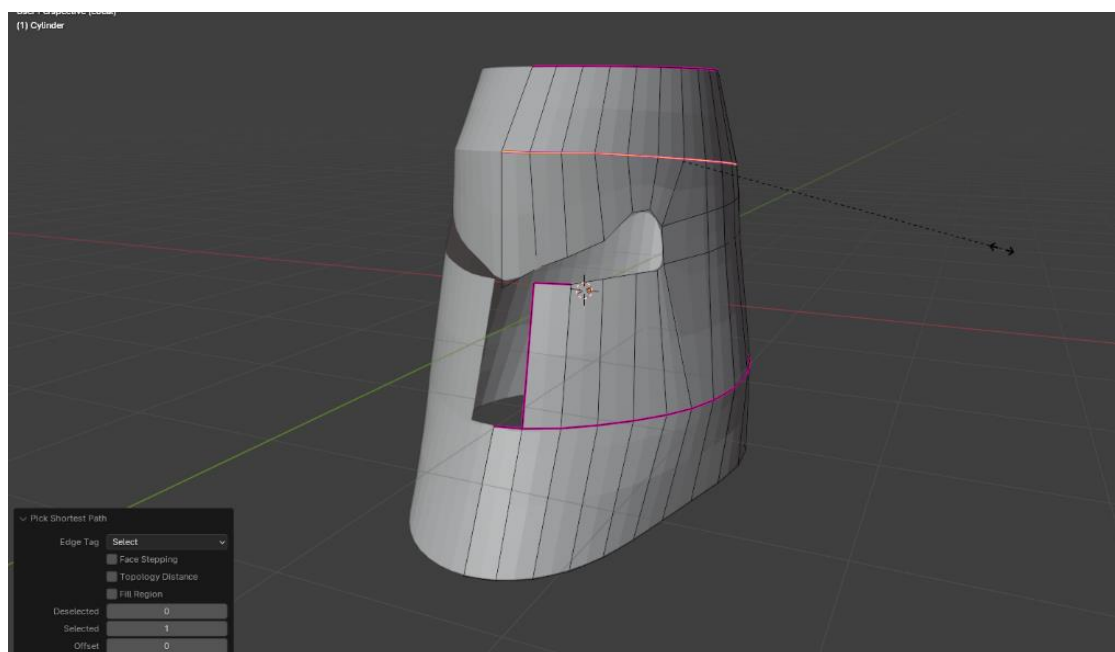
Slika 6.1.2. Korištenje alata *grid fill*

Zatim alatom *subdivision surface* stranice modela se dijele na manje dijelove kako bi se dobio gladak izgled modela.



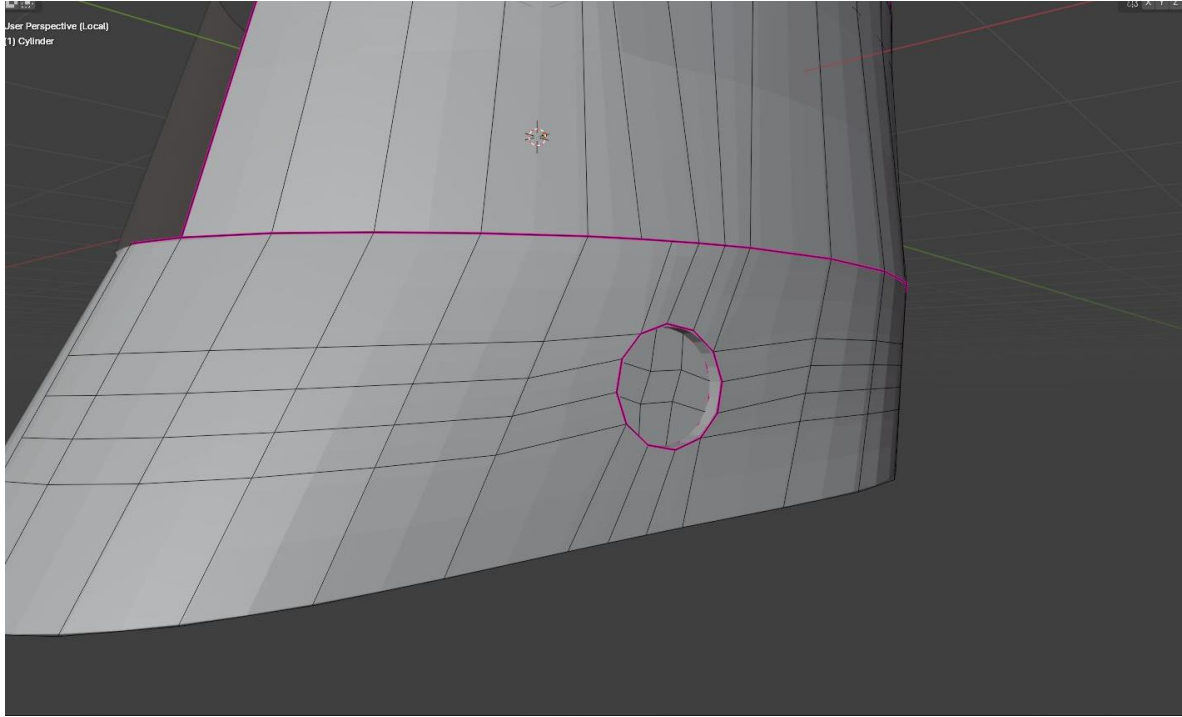
Slika 6.1.3. Izgled baze kacige nakon korištenja alata *subdivision surface*

Nakon korištenja alata *subdivision surface* potrebno je alatom *edge crease* odrediti koliko će rubovi modela biti oštri. Ljubičastom crtom na bridovima modela može se vidjeti čim su više izoštreni tim je jača ljubičasta boja.



Slika 6.1.4. Prikaz alata *edge crease*

Zatim se alatom *loopcut* dodaju dodatni bridovi koji se tada selektiraju i pomoću dodatnog alata *looptools circle* mogu pretvoriti u krug. Zatim se alatom *extrude* dobiveni krug udubljuje u kacigu.



Slika 6.1.5. Prikaz alata *loopcut* i *extrude*

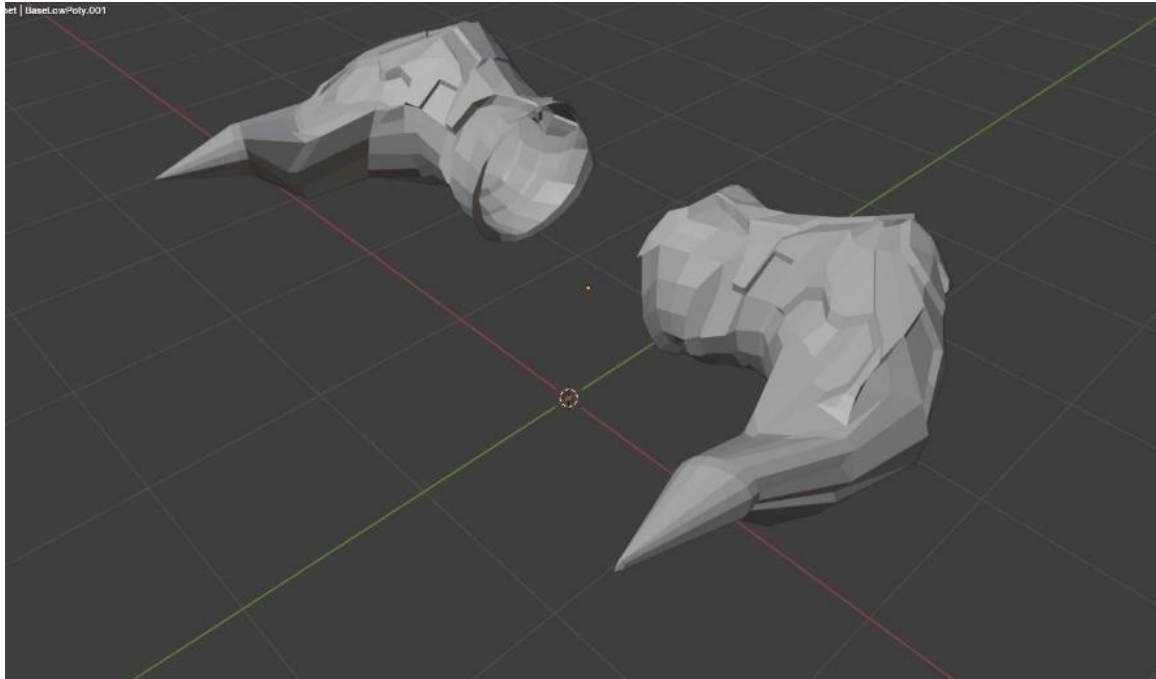
Korištenjem gore navedenih alata i dodataka kreira se model prema ideji. Kod kreiranja modela potrebno je imati nekakvu referencu prema kojoj se modelira, kako bi model bolje i realističnije ispao. Ako se ne koriste reference može doći do krivo formiranog modela u smislu njegove veličine ili veličine pojedinih dijelova.



*Slika 6.1.6. Finalni izgled baze kacige*

## 6.2. Izrada rogova

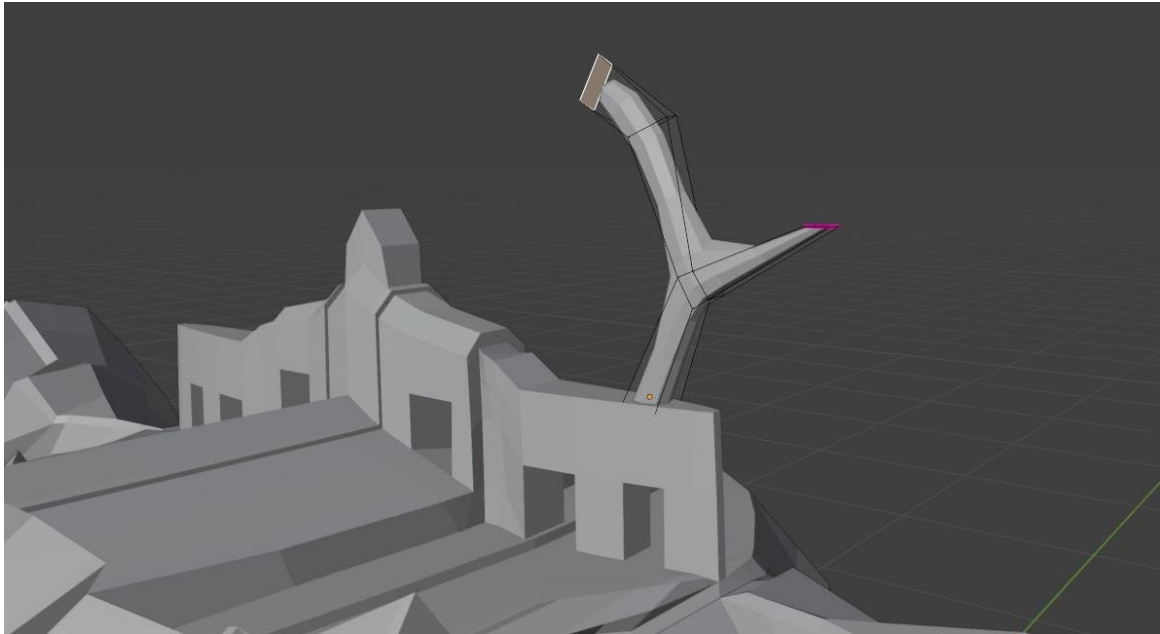
Nakon što se izradila baza kacige, pomoću alata *looptool* na isti način kako se izradilo kružno udubljenje kod kacige se s obje strane izreže dio gdje idu rogovi. Zatim selektiramo sve bridove na otvoru kacige i pomoću alata *looptool* oblikujemo pravilni krug. Zatim pomoću alata extrude pomoću bridova otvora kacige smo oblikovali rogove. Također zbog *subdivision-a* koji smo napravili sada lakše možemo oblikovati rogove.



Slika 6.2.1. Finalni izgled rogova

### 6.3. Izrada krune

Za izradu krune kreirana je kocka koja je *subdivide-ana* i *extrudana* na isti način kao i kod rogova.



Slika 6.2.2. Prikaz ekstrudanja kocke

## 6.4. Izrada ornamenata

Kod izrade ornamenta korišten je alat *bezier curve* koji omogućava kreiranje crvene linije koja se može proizvoljno oblikovati. Kada liniju oblikujemo po želji tada u postavkama linije povećamo njezinu dubinu na proizvoljnu veličinu. Pomoću tog alata su se oblikovali svi krivudavi dijelovi kacige.

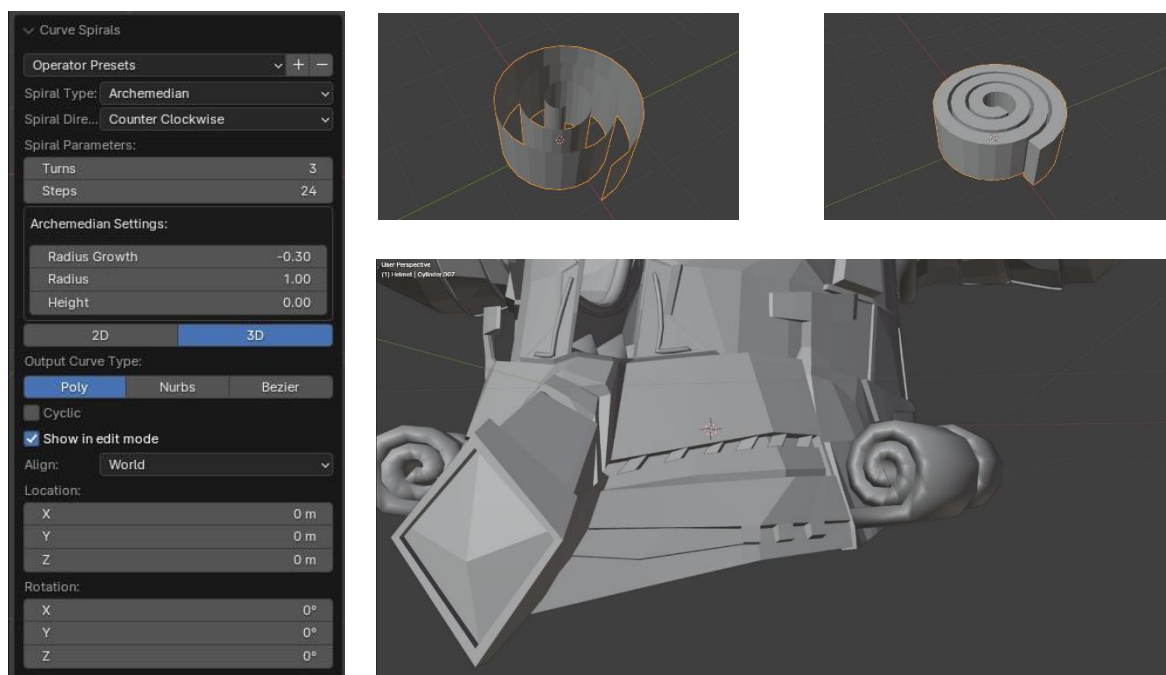


Slika 6.4.1. Prikaz alata *bezier curve*



## 6.5. Izrada detalja

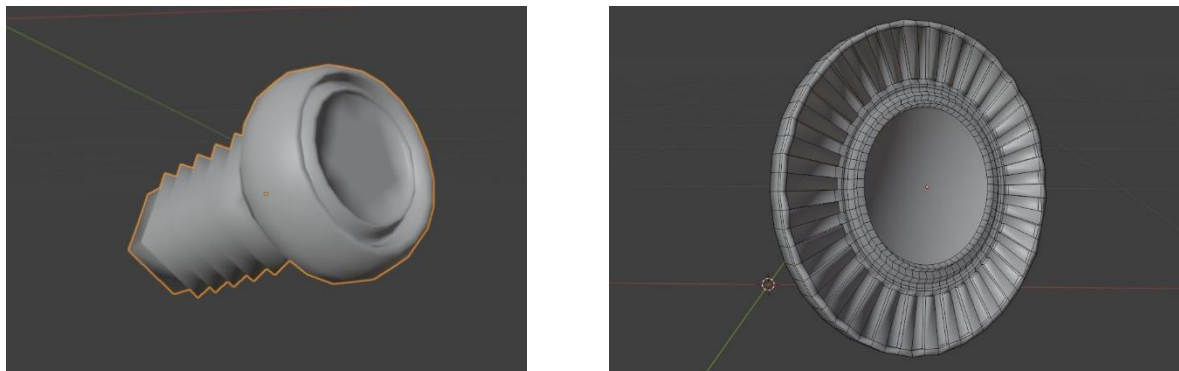
Kod kreiranja detalja izrađena je spiralna tkanina pomoću alata *curve spirals archemedian*. U postavkama spirale promjenom parametara i postavki dobiva se željeni oblik koji smo pretvorili u *mesh*, te se oblik *extrud-ao*. Te pomoću *modifera solidify* se povećala debljina objekta. Te opcijom *shade smooth* se površina objekta izgladila.



Slika 6.5.1. Prikaz postavka alata *curve spirals* i izgled modela

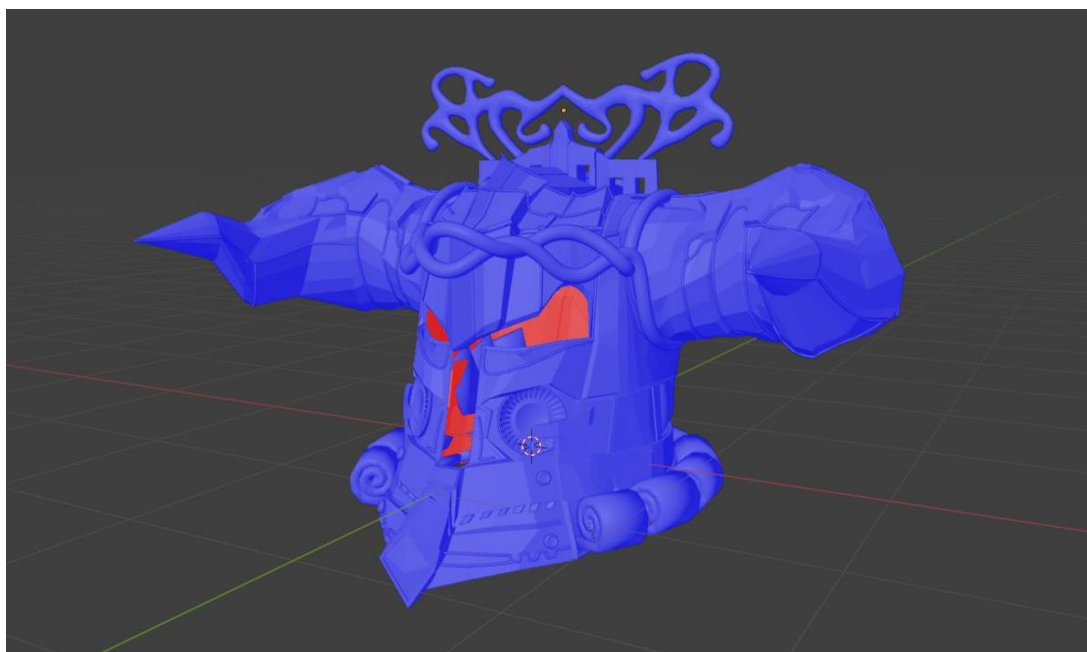
## 6.6. Završne radnje

U finalnom dijelu izrade modele kreirali su se vijci i oblici koji su popunili prazninu po modelu kacige radi dobivanja realističnosti i detalja.



Slika 6.6.1. i .6.6.2. Izgled vijka i jednog od detalja kacige

Dobra praksa je kod finala izrade modela potvrditi *scale*. Također potrebno je kod prozora *viewport overlays*, uključiti opciju *face orientation*, koja prikazuje da li su pravilno okrenute stranice modela. Plavom bojom je označen prednji dio stranice, dok crvenom bojom su označeni stražnji dijelovi stranica. Samim time kada bi se model koristio u videoigrama, crveni dio se ne bi prikazivao.



Slika 6.6.3. Prikaz opcije *face orientation*

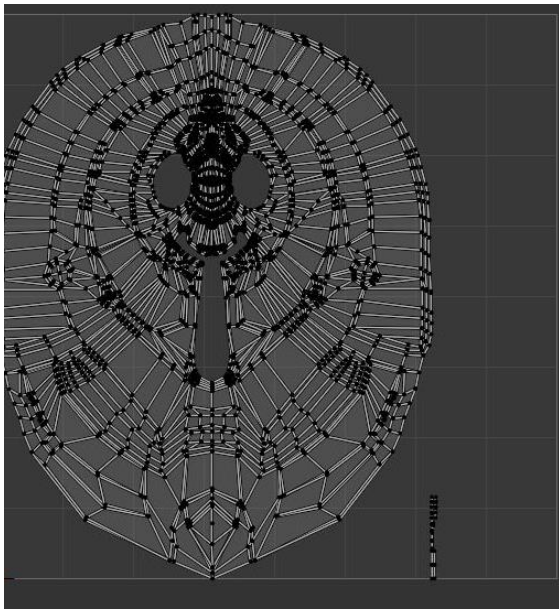
## 9. Proces teksturiranja

Nakon završetka izrade modela potrebno je preći na proces teksturiranja.

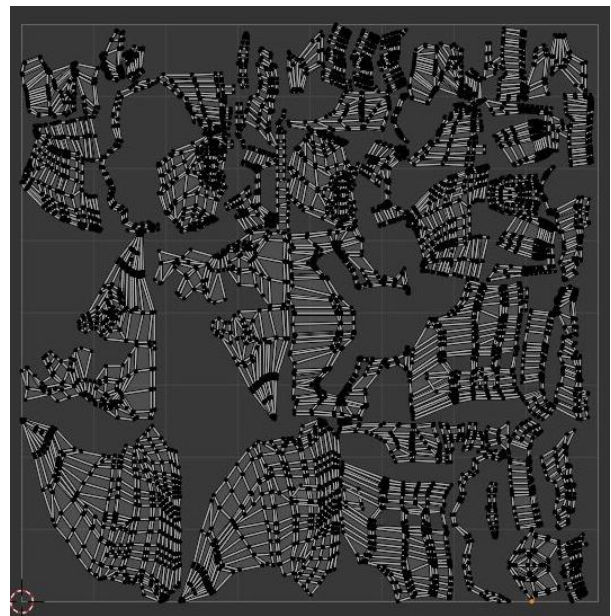
### 9.1. UV odmotavanje

Prije početka procesa kreiranja *normal mape*, potrebno je proći kroz proces *unwrappanja*.

Taj proces služi za kreiranje UV mape koja služi za projiciranje teksture na model. Potrebno je prijeći u *UV editing* prozor radi lakšeg snalaženja. Ako se model samo *unwrappa* bez ikakvog podešavanja tada će UV mapa biti poprilično neiskoristiva. Kod faze teksturiranja sama tekstura neće biti iste rezolucije na svim dijelovima modela. Kako bi se to izbjeglo potrebno je raditi rezove bridova modela pomoću opcije *mark seam* kako bi se lakše preslikao model na UV mapu. Radi bolje rezolucije model bi trebao biti rasprostranjen preko cijele UV mape.



Slika 9.1.1. Prikaz UV mape kacige bez upotrebe šavova



Slika 9.1.2. Prikaz UV mape kacige gdje su upotrjebljeni šavovi

## 9.2. Kreiranje *normal mape*

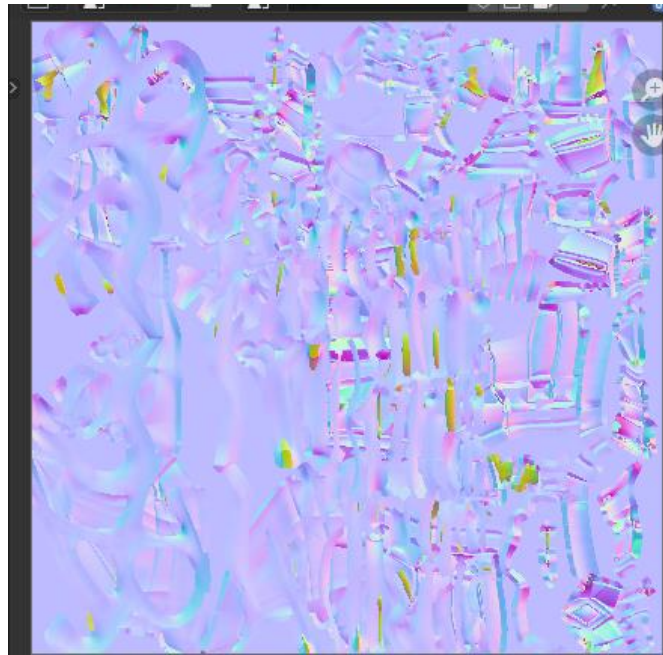
Kod optimizacije modela za videoigre potrebno je kreirati normalnu mapu. Svrha normalne mape je kopiranje podataka o odbijanju svjetla *high-poly* modela na *low-poly* model. Taj proces se naziva *baking normals*.

Duplicira se model, te se kod dobivenog modela poveća razina *subdivision surfaca*.



Slika 9.2.1. Prikaz kacige *high-poly* (lijevo) i *low-poly* (desno)

Zatim je modele potrebno preklopiti. U prozoru *shading* kreira se materijal i kreira se *image texture node* u kojem će biti pohranjena *normal mapa*. Stavi se naziv *noda*, rezolucija, te color space na *Non-Color*. Zatim u postavkama za *render* postavimo *render engine* na *cycles* zbog kvalitete i ne mogućnosti izrade normal mapa kod *EVEE render engine-a*. Zatim u postavka za *bake* stavimo *bake type* na *normal* i potvrdimo postavku *selected to active*. Zatim je potrebno zajedno selektirati prvo high poly model, a zatim low poly i klikne se *bake*.

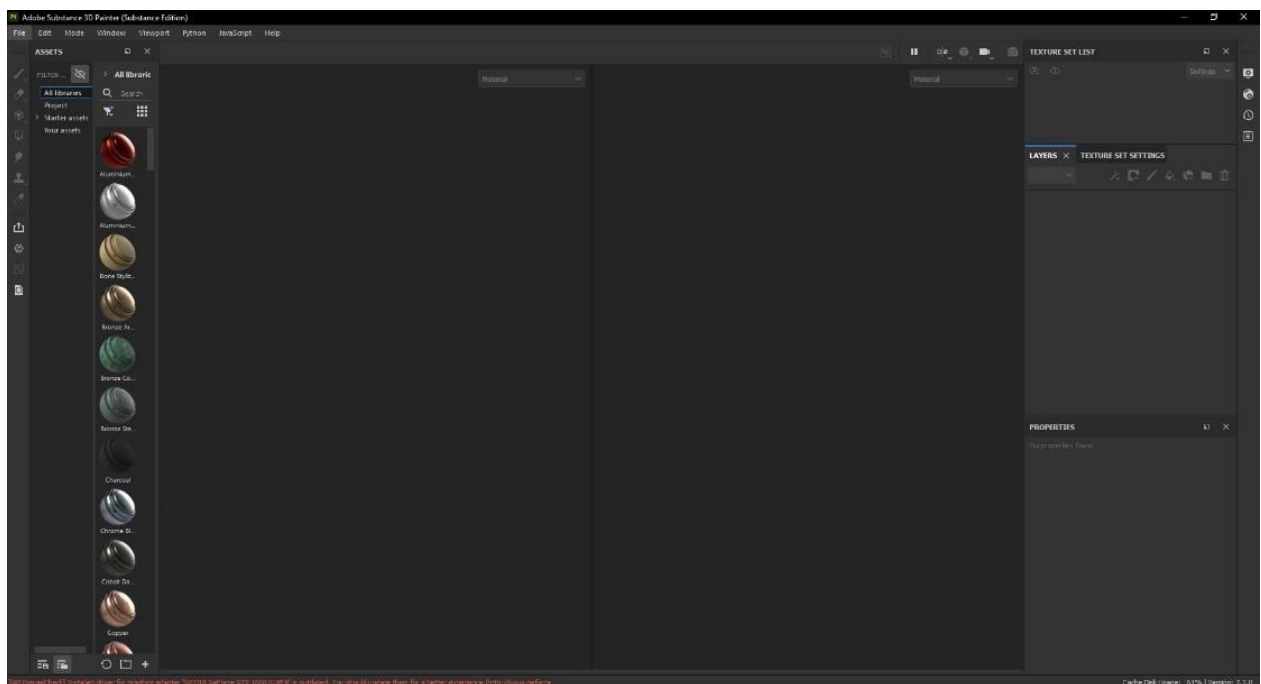


Slika 9.2.2. Prikaz normal mape kacige

Potrebno je model izvesti u formatu .fbx i normal mapu u formatu .png.

### 9.3 Kreiranje tekstura u programu substance painter

U program substance painter uvezemo model i njegovu normal mapu, tehnikom *drag and drop* biramo već kreirane teksture i stavljamo ih na model. Također svaka tekstura kreirana od programa može se dodatno podešavati u postavkama materijala.



Slika 9.3.1. Sučelje programa substance painter

## 10. Zaključak

Modeliranje je vrlo zahtjevan proces gdje je potrebno uložiti puno rada i truda, da bi se izradio kvalitetan model. Modeliranje je danas zastupljeno u svim medijima i sve se više koristi u filmovima, arhitekturi, medicini i posebice u videoigrama. Digitalno kreiranu produkciju, likove, okruženja, automobile vidimo svugdje u stvarnome životu. Tehnologija se razvija svakodnevno i neprestano donosi nove značajke što potiče njezine korisnike na stalno učenje nečeg novog. Sve je teže raspoznati stvarno od virtualnog zato što mogućnosti programa primjerice Blender koji je besplatan raste svakim danom. U radu su bili prikazani i objašnjeni elementi optimizacije kod specifično geometrije, te njezin utjecaj na dobru ili lošu optimizaciju kod videoigara, te važnost istih. U praktičnom primjeru se detaljnije pokazalo neki od spomenutih optimizacija, te se usput upoznao program Blender. Savjet čitateljima je da se vrijedi upoznati i malo dublje ući u svijet modeliranja jer su mogućnosti beskonačne. Ispočetka se čini teško, ali kada se pohvataju osnove bilo tko može napraviti kvalitetan model uz uloženi trud. Ovaj rad je bio izrađen s ciljem upoznavanja procesa modeliranja kroz teorijske i praktične primjere, te njegova optimizacija za korištenje kod videoigara.



*10.1 Finalni izgled 3D modela*

## 11. Literatura

- [1] William Vaughan: [digital] modeling, New Riders – 18.9.2024.
- [2] John M. Blain: The Complete Guide to Blender Graphics 3rd EDITION – Computer Modeling & Animation, CRC Press – 18.9.2024.
- [3] Arm: Unreal Engine Geometry best practices for Unreal Engine Issue01 – 18.9.2024.
- [4] Antti Lindsten: Exploring modern methods of 3d asset creation, Turku University of applied sciences – 18.9.2024.
- [5] Adam Ruch: Understanding videogames: 3 Perspective for Criticism – 18.9.2024.
- [6] <https://docs.blender.org/manual/en/latest/index.html> : Blender manual – 18.9.2024.
- [7] <https://docs.unity3d.com/Manual/index.html> : Unity manual – 18.9.2024.



## 12. Popis slika

Slika 1.3.1. Razlika između modela s više poligona (desno) i modela s manje poligona (lijevo).....	10
Slika 2.1. Primjer koordinatne osi na primitivu kocke.....	11
Slika 2.2.1. Razlika između modela s topologijom kvadrata (lijevo) i topologije trokuta (desno).....	12
Slika 2.2.2. Izgled poligona spomenutih topologija.....	13
Slika 3.1.1.1.Prikaz modela s puno poligona (lijevo) i modela s manjim brojem poligona (desno).....	15
Slika 3.1.1.2.Prikaz stapanja između 2 modela, gdje se pokušava replicirati jednak izgled jednostavnog modela naspram kompliciranijeg modela .....	15
Slika 3.1.1.3. Razlika između modela nakon korištenja softwrap-a.....	15
Slika 3.1.1.4. Prikaz crtanja topologije na modelu.....	16
Slika 3.1.1.5. Prikaz poluautomatskog crtanja topologije na modelu.....	16
Slika 3.2.1. Prikaz potpuno pravilne i ravne preslike kocke na UV mapu.....	17
Slika 3.2.2. Prikaz neravne UV mape kugle.....	17
Slika 3.2.3. Prikaz karirane teksture na modelu kugle gdje se vide izrađeni šavovi.....	18
Slika 3.2.4. Prikaz karirane teksture na modelu kugle bez dodavanja šavova (vide se rastezanja teksture).....	18
Slika 3.2.1.1. Razlika teksture zida između diffuse i albedo mape.....	19
Slika 3.2.2.1. Mapa hrapavosti teksture.....	19
Slika 3.2.3.1 Prikaz normal mape bora na čovjekovom licu.....	20
Slika 3.3.1. Prikaz LOD-a (kamera je blizu na lijevom primjeru, dok na desnom kamera najdalje od objekta).....	21
Slika 3.4.1.1. Prikaz Culling tehnika.....	22
Slika 3.4.2.1. Prikaz razlike teksture poda s teselacijom (desno) i bez teselacije (lijevo).....	23
Slika 3.4.2.2. Prikaz promjene rezolucije kod dinamične razlučivosti.....	23
Slika 4.1.1. Primjer scene iz igre Metal Gear Solid V.....	25
Slika 4.1.2. Primjer scene iz igre Metal Gear Solid V.....	25
Slika 4.2.1. Primjer scene iz igre Factorio.....	26
Slika 5.1.1. Sučelje programa Blender.....	27
Slika 6.1.1. Dovršena baza kacige.....	28
Slika 6.1.2. Korištenje alata grid fill.....	29
Slika 6.1.3. Izgled baze kacige nakon korištenja alata subdivision surface.....	30
Slika 6.1.4. Prikaz alata edge crease.....	30
Slika 6.1.5. Prikaz alata loopcut i extrude.....	31

Slika 6.1.6. Finalni izgled baze kacige.....	31
Slika 6.2.1. Finalni izgled rogova.....	32
Slika 6.2.2. Prikaz ekstrudanja kocke.....	32
Slika 6.4.1. Prikaz alata bezier curve.....	33
Slika 6.5.1. Prikaz postavka alata curve spirsals i izgled modela.....	34
Slika 6.6.1. i .6.6.2. Izgled vijka i jednog od detalja kacige.....	35
Slika 6.6.3. Prikaz opcije face orientation.....	35
Slika 9.1.1. Prikaz UV mape kacige bez upotrebe šavova.....	36
Slika 9.1.2. Prikaz UV mape kacige gdje su upotrjebljeni šavovi.....	36
Slika 9.2.1. Prikaz kacige high-poly (lijevo) i low-poly (desno).....	37
Slika 9.2.2. Prikaz normal mape kacige.....	37
Slika 9.3.1. Sučelje programa substance painter.....	38
Slika 10.1 Finalni izgled 3D modela.....	40

Sveučilište  
Sjever

UNIVERSITY  
OF NORTH  
DACOTA



SVEUČILIŠTE  
SJEVER

#### IZJAVA O AUTORSTVU

Završni/diplomski/specijalistički rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, ANDREJA PLAMIC - PETAK (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog/specijalističkog (obrisati nepotrebno) rada pod naslovom GEOMETRIJA I OPTIMIZACIJA 3D NOKRO ZA VIDEOKOLE (upisati naslov) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:  
(upisati ime i prezime)

Andreja Plamir Petak  
(vlastoručni potpis)

Sukladno članku 58., 59. i 61. Zakona o visokom obrazovanju i znanstvenoj djelatnosti završne/diplomske/specijalističke radove sveučilišta su dužna objaviti u roku od 30 dana od dana obrane na nacionalnom repozitoriju odnosno repozitoriju visokog učilišta.

Sukladno članku 111. Zakona o autorskom pravu i srodnim pravima student se ne može protiviti da se njegov završni rad stvoren na bilo kojem studiju na visokom učilištu učini dostupnim javnosti na odgovarajućoj javnoj mrežnoj bazi sveučilišne knjižnice, knjižnice sastavnice sveučilišta, knjižnice veleučilišta ili visoke škole i/ili na javnoj mrežnoj bazi završnih radova Nacionalne i sveučilišne knjižnice, sukladno zakonu kojim se uređuje umjetnička djelatnost i visoko obrazovanje.