

Izrada humanoidnog 3D lika za video igru

Blaslov, Heidi

Undergraduate thesis / Završni rad

2020

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:312869>

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

Download date / Datum preuzimanja: **2024-08-04**



Repository / Repozitorij:

[University North Digital Repository](#)





**Sveučilište
Sjever**

Završni rad br. 663/MM/2019

Izrada humanoidnog 3D lika za video igru

Heidi Blaslov, 2227/336

Varaždin, lipanj 2020. godine



Sveučilište Sjever

Odjel za Multimediju, oblikovanje i primjenu

Završni rad br. 663/MM/2019

Izrada humanoidnog 3D lika za video igru

Student

Heidi Blaslov, 2227/336

Mentor

doc. dr. sc. Andrija Bernik

Varaždin, lipanj 2020. godine

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL **Odjel za multimediju**

STUDIJ **preddiplomski stručni studij Multimedija, oblikovanje i primjena**

PRISTUPNIK **Heidi Blaslov**

MATIČNI BROJ **2227/336**

DATUM **23.10.2019.**

KOLEGIJ **Računalna animacija**

NASLOV RADA **Izrada humanoidnog 3D lika za video igru**

NASLOV RADA NA ENGL. JEZIKU **Creating a humanoid 3D video game character**

MENTOR **doc.dr.sc. Andrija Bernik**

ZVANJE **Docent**

ČLANOVI POVJERENSTVA

1. **mr.sc. Dragan Matković, v. pred. - predsjednik**
2. **doc.art. Robert Geček - član**
3. **doc.dr.sc. Andrija Bernik - mentor**
4. **pred. Nikolina Bolčević Horvatić, dipl.ing. - zamjenski član**
5. _____

Zadatak završnog rada

BROJ **663/MM/2019**

OPIS

Cilj ovog rada je prikazivanje koraka izrade 3D humanoidnog lika u alatu Blender s posebnim naglaskom na njegovu optimizaciju za video igru. U radu će se opisati zahtjevi koje postavlja tip video igre za koju je lik osmišljen te će biti objašnjeni korišteni pojmovi, alati, tehnike i postupci. U praktičnom dijelu rada bit će izrađen 3D humanoidni lik za video igru s demonstracijom u alatu Unity uz upotrebu animacija s Unity Asset Store-a kako bi se dočaralo iskustvo koje taj lik treba pružiti.

U radu je potrebno:

- Opisati alate Blender i Unity te platformu Unity Asset Store.
- Nabrojati zahtjeve koje postavlja tip video igre za koju je lik osmišljen.
- Objasniti sve korake izrade 3D humanoidnog lika uz praktični primjer.
- Opisati tehnike korištene za optimizaciju lika za video igru.
- Izraditi demonstraciju završenog lika u alatu Unity uz dodavanje besplatnih animacija.

ZADATAK URUČEN

02.07.2020.



POTPIS MENTORA

Bernik

Predgovor

Interes za 3D modeliranje u meni se pojavio na kolegijima 3D modeliranja i računalne animacije zbog kojih sam poželjela proširiti svoje znanje i vještine na području 3D umjetnosti. Izrada 3D humanoidnog lika za video igru za mene je na brojne načine predstavljala velik izazov, ali time i priliku za učenje.

Ovim putem želim iskazati zahvalnost mentoru doc. dr. sc. Andriji Berniku na omogućavanju realizacije ovog završnog rada i mentoru stručne prakse Dejanu Koberu na korisnim savjetima vezanim uz izradu praktičnog dijela završnog rada.

Sažetak

Završni rad obrađuje temu izrade i optimizacije 3D modela za video igre. Video igre definiraju brojne zahtjeve koje svaki 3D model treba ispunjavati kako bi performanse u video igri bile zadovoljavajuće. Opisana je važnost reduciranja broja poligona od kojeg se geometrija sastoji i objašnjene su tehnike kojima se to postiže. Posebna pažnja posvećena je na ekonomično korištenje tekstura kod postizanja dopadljivog izgleda modela. U izradi praktičnog dijela završnog rada korišteni su alati Blender i Unity. Navedene su funkcionalnosti i pogodnosti korištenja tih alata.

U praktičnom dijelu rada izrađen je 3D humanoidni lik u alatu Blender. Blender je besplatni i open source 3D alat za stvaranje koji u cijelosti obuhvaća proces 3D stvaranja: modeliranje, skulpturiranje, teksturiranje, animiranje itd. Po završetku stvaranja lika, izrađena je jednostavna demonstracija upotrebe lika u razvojnom okruženju za video igre Unity uz upotrebu animacija s Unity Asset Store digitalne trgovine kako bi se dočaralo iskustvo koje taj lik treba pružiti.

Ključne riječi: 3D modeliranje, Blender, Unity, humanoidni lik, video igra

Abstract

The topic of this final project is creating and optimizing 3D models for video games. Video games demand a number of requirements that every 3D model needs to meet in order for the video game to perform well. The importance of reducing the number of polygons of which the geometry consists of is described and the techniques used to achieve that are explained. Special attention is paid to the economical use of textures in achieving a pleasing appearance of the model. Softwares Blender and Unity were used in the practical part of the final project. The tools and benefits of using these softwares are listed.

In the practical part of the final project, a 3D humanoid figure in the Blender softwer was created. Blender is a free and open source 3D creation tool that fully encompasses the 3D creation process: modeling, sculpting, texturing, animating, etc. Upon completion of character creation, a simple demonstration of character use was made in the Unity video game engine using animations downloaded from the Unity Asset Store. This simple demonstration was made to show how the charecter would look like in a full game.

Keywords: 3D modeling, Blender, Unity, humanoid character, video game

Popis korištenih kratica

3D	Three-dimensional Trodimenzionalan
2D	Two-dimensional Dvodimenzionalan
LOD	Level of Detail Razina detalja modela
API	Application Programming Interface Aplikacijsko programsko sučelje
VR	Virtual Reality Virtualna stvarnost

Sadržaj

1.	Uvod.....	1
2.	Optimizacija 3D modela za video igre.....	2
2.1.	Optimizacija geometrije	4
2.2.	Optimizacija tekstura.....	8
3.	Alati korišteni u izradi lika za video igru.....	13
3.1.	Blender 2.8	14
3.2.	Unity 2019.....	16
4.	Proces izrade lika za video igru	18
4.1.	Reference.....	19
4.2.	Skulptura	22
4.3.	Optimizacija modela	32
4.4.	Teksture	40
4.5.	Kostur	51
4.6.	Demonstracija u razvojnom okruženju Unity	53
5.	Zaključak.....	60
6.	Literatura.....	62
7.	Popis slika	65
8.	Prilozi.....	67

1. Uvod

3D modeliranje je proces izrade matematičkog prikaza bilo kojeg objekta u tri dimenzije putem specijaliziranog softvera. Gotov proizvod se naziva 3D model. 3D model je geometrija sastavljena od skupa točaka u 3D prostoru, koje su povezane različitim geometrijskim cjelinama poput rubova, trokuta, četverokuta, zakrivljenih površina itd. 3D modeli se mogu kreirati ručno, algoritamski ili skeniranjem. Njihove se površine mogu dodatno definirati mapiranjem tekstura. [1]

Postoje tri popularna načina za izradu 3D modela. Poligonalno modeliranje odnosi se na postavljanje skupa točki u 3D prostoru, nazvane verteksi, koje su povezane linijskim segmentima kako bi tvorili mrežu poligona. Većina 3D modela danas je izrađena u obliku poligonalnih modela, jer su fleksibilni i jer ih računala mogu jako brzo prikazati. Međutim, poligoni su ravni i mogu stvoriti dojam zakrivljene površine samo korištenjem velikog broja poligona. Modeliranje krivuljama stvara površine definirane krivuljama na koje utječu upravljačke točke. Digitalno skulpturiranje postaje sve popularnije i šire korišteno, a vještina 3D skulpturiranja korisna je u procesu kreiranja 3D modela za video igre.

3D modeli široko su upotrebljivi. Mnoge su računalne igre koristile unaprijed renderirane slike 3D modela kao 2D teksture za objekte prije nego što su ih računala mogla prikazati u stvarnom vremenu. S povećavanjem hardverskih sposobnosti današnjih računala, video igre su prešle iz 2D grafike u 3D prostore, a kvaliteta prikaza trodimenzionalnih svjetova u video igrama godinama neprestano raste. Video igra je multimedij u kojem je često jednako bitna kvalitetna izvedba priče, interakcije, vizualnih elemenata, dizajna zvukova i glazbe itd. Za mnoge je video igre iznimno bitan živopisan svijet i zanimljiv dizajn okoline. Ipak, dizajn i izvedba glavnog i sporednih likova u video igri jedan je od najvažnijih aspekata kreiranja video igre. Glavni lik je poveznica između igrača i svijeta u video igri, pa tako lik može tu vezu učvrstiti ili ju slomiti. [2]

U sljedećim poglavljima su navedeni zahtjevi koje video igre postavljaju u izradi 3D modela i predstavljene su tehnike kojima se zadovoljavaju ti zahtjevi. Kako bi se temeljito opisao i pojasnio proces izrade modela optimiziranih za video igru, u praktičnom dijelu je izrađen 3D humanoidni lik po današnjim standardima u industriji video igara.

2. Optimizacija 3D modela za video igre

Zahvaljujući poboljšanjima u tehnologiji, danas je većina velikih video igara napravljena pomoću 3D objekata. Prilikom izrade 3D modela za video igre važno je razumjeti koji zahtjevi postoje i kako raditi unutar njih kako bi se stvorila visokokvalitetna video igra. Iako je kvaliteta grafike u video igrama u porastu, još uvijek postoje velika ograničenja koja se moraju uzeti u obzir pri radu na video igrama. S nedavnim porastom 3D mobilnih igara niska ograničenja poligona koja su nekada bila prisutna opet postaju uobičajena. Ipak, s novim tehnologijama sve je češće moguće renderiranje u realnom vremenu sve većeg broja poligona.

Video igre su vrlo jedinstven medij jer su interaktivne, za razliku od mnogih drugih oblika izražavanja. Poput gotovo svih umjetnosti, zahtijevaju strpljenje, praksu i odgovarajuće alate. Softveri za modeliranje omogućuju stvaranje lokacija, likova i drugih objekata za video igre. Video igre danas često sadrže zanimljive lokacije prepune detalja. Dobro razvijeni svijet u video igrama može oživjeti igru, a često je jedan od glavnih elemenata koji čine igru. Zato svijet u video igri vrlo brzo može postati pun raznih objekata koji trebaju biti dovoljno dobro optimizirani. Većina video igara stvara ravnotežu između stvarnog života i fikcije. Mnoge video igre koriste 3D modele objekata iz stvarnog života. 3D modeli mogu biti stvoreni od fotografija koje se unose u softver za modeliranje kako bi se stvorio digitalni prikaz. [3]



Slika 2.1. Kingdom Come: Deliverance svijet [4]

Jedan od najsloženijih i najzanimljivijih dijelova dizajna videoigara je stvaranje likova. Prilikom izrade likova s 3D alatom, važno je znati na koje detalje vrijedi obratiti pozornost kako ne bismo video igru preplavili nepotrebnim informacijama. Kod izrade 3D modela za filmove, često je jedino ograničenje vrijeme, dok smo u video igrama ograničeni snagom hardvera na kojem se igra. Za razliku od filmova, video igre ne mogu biti prethodno renderirane. Filmski uvod u video igru i povremene scene koje su prethodno renderirane gotovo uvijek izgledaju mnogo bolje nego modeli dok igramo, zbog toga što je još uvijek nemoguće prikazati tu količinu detalja u realnom vremenu. Dok igramo video igru, na zaslonu se može prikazivati na tisuće različitih objekata u isto vrijeme koji zauzimaju puno procesorske snage. Iskustvo igranja može pogoršati zaostajanje od samo nekoliko sekundi. [5]

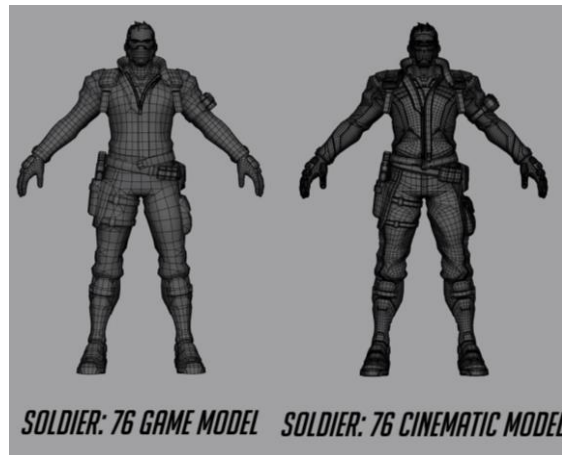


Slika 2.2. Razlika kvalitete modela za film (lijevo) i video igru (desno) [6]

Optimizacija 3D modela je važan dio stvaranja video igara jer su memorija i grafička kartica računala kao i njegova središnja procesorska jedinica ograničeni u snazi. Ako se njihove mogućnosti premaše, brzina kadrova u video igri će biti mala čime se jako smanjuje zadovoljstvo i nesmetanost igranja video igre.

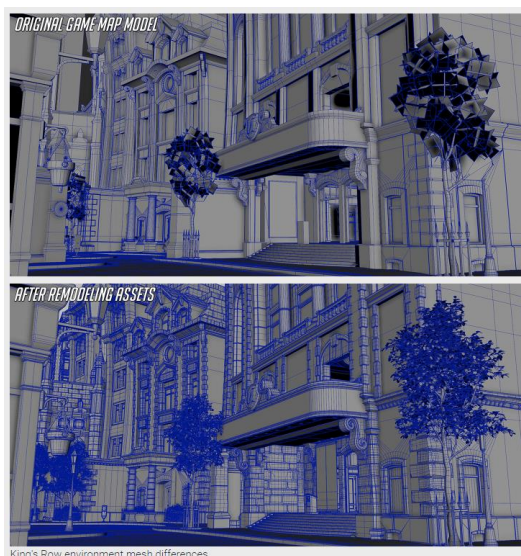
2.1. Optimizacija geometrije

Poligonski proračun je ukupan broj trokutastih poligona koji su potrebni za izradu nekog 3D objekta. Što je veći broj poligona, to je potrebno više vremena za prikaz tog objekta. To je zato što svaki verteks, rub i lice modela zahtijeva uporabu računalnih resursa. [5]



Slika 2.3. Usporedba lika za video igru (lijevo) i film (desno) [7]

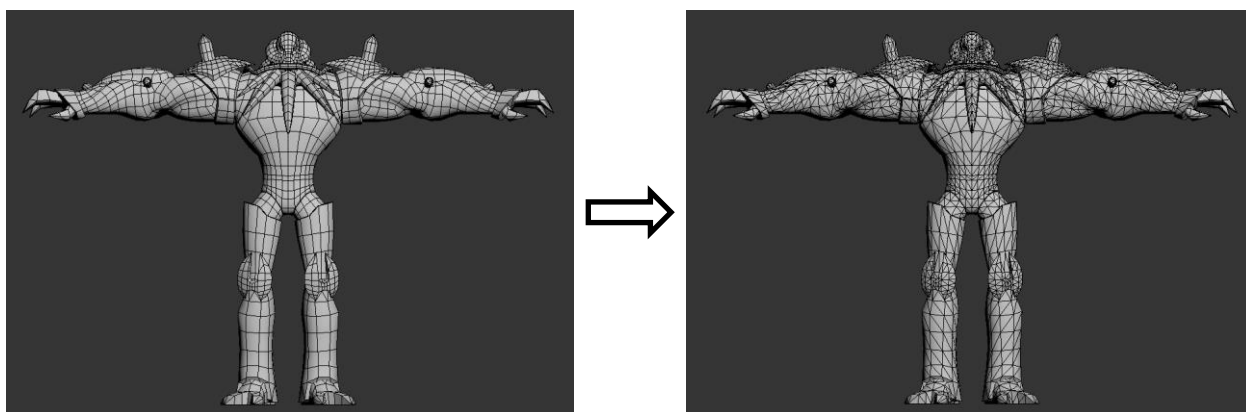
Broj poligona koji se koristi ovisi o vizualnoj kvaliteti modela koja je potrebna i platformi za koju se model izrađuje. Što se više poligona koristi u geometriji modela, to je model detaljniji i izgleda realnije i prirodnije. Zato je potrebno pronaći pravu količinu poligona s kojom objekt dobro izgleda, a optimiziran je za video igru.



Slika 2.4. Usporedba okoliša za video igru (gore) i film (dolje) [7]

2.1.1. Triangulacija

Model se mora sastojati isključivo od trokuta i četverokuta. Kada se model unese u video igru svaki poligon je trianguliran (pretvoren u dva ili više trokuta) za lakše izračunavanje prilikom prikazivanja. Trianguliranje poligona je jednostavan proces, ali što više strana ima poligon, to je više načina triangulacije i proces postaje kompliciraniji. Pri izradi modela bitno je pripaziti da je model u cijelosti izgrađen od trokuta i četverokuta. [5]



Slika 2.5. Triangulacija modela [8]

2.1.2. Silueta objekta

Broj poligona se može smanjiti pojednostavljenjem siluete objekta. Na taj se način izdvajaju glavni oblici koji čine taj objekt bez da postane previše složen. Također se treba uzeti u obzir jesu li neki dijelovi modela uopće važni i može li ih se potpuno izbrisati. Na sljedećim je slikama primjer redukcije oblika fotoaparata na dijelove koji čine siluetu fotoaparata jednostavnijom. [5]



Slika 2.6. Pojednostavljena silueta fotoaparata [5]

2.1.3. Nepotrebni poligoni

Na modelima za video igre je potrebno ukloniti nevidljive poligone iz modela. Kada je u pitanju broj poligona i redukcija tog broja u video igrama, sve se svodi na ono što će igrač na kraju moći vidjeti u sceni i iz koje perspektive će igrač to vidjeti. Ako igrač ne može vidjeti određene strane modela zbog načina na koji se model koristi u igri, ti se poligoni mogu ukloniti. [9]

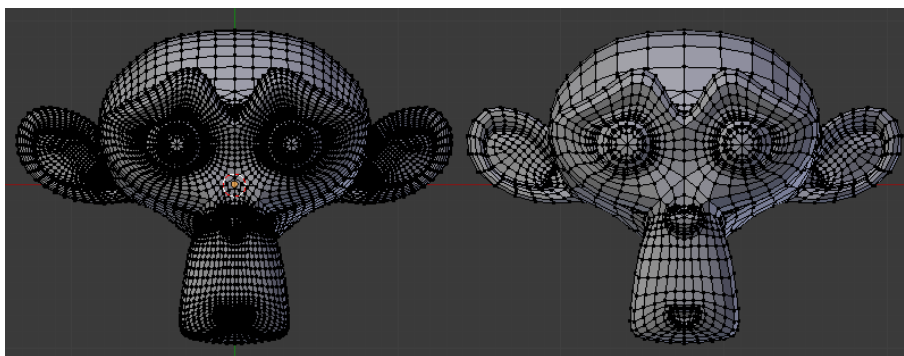
Nakon izrade modela potrebno je razmotriti gdje bi se sve mogli nalaziti poligoni koji su skriveni unutar geometrije tijela i time bespotrebni, ili koji se zbog načina korištenja u video igri nikada neće moći vidjeti. Ako se npr. dio poligona glave nalazi ispod kose lika, ti se poligoni mogu obrisati. U video igrama u kojima se oružja mogu pokupiti s tla, često se izrađuje inačica oružja s obrisanim poligonima koji su okrenuti prema tlu i inačica oružja sa svim poligonima kada ga igrač pokupi s tla. [5]



Slika 2.7. Razlika modela oružja iz trećeg (lijevo) i prvog lica (desno) [5]

2.1.4. Smanjenje broja rubnih petlji

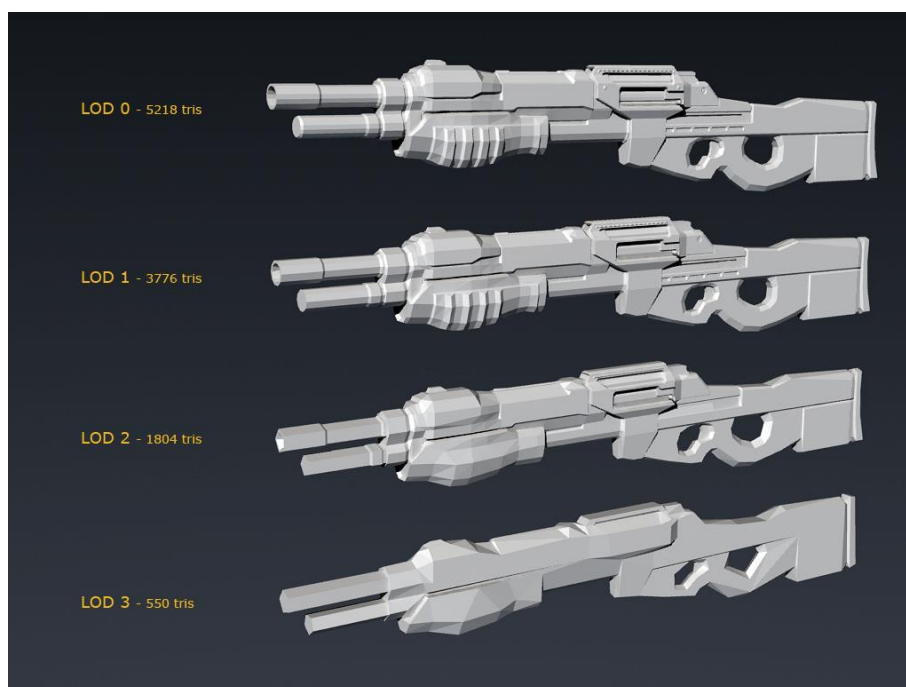
Svaki model kompleksnije geometrije sagrađen je od brojnih rubnih petlji (*engl. edge loop*). Za svaki model koji se izrađuje za video igru je potrebno razmotriti je li razina podjele modela, odnosno broj rubnih petlji previsok. Ukoliko je, nekoliko je načina na koje se geometriji može smanjiti broj rubnih petlji. Najjednostavniji način je brisanjem svake druge rubne petlje, no u mnogim alatima za izradu 3D modela postoje modifikatori koji to automatski čine.



Slika 2.8. Reduciranje rubnih petlji [10]

2.1.5. LOD modeli

Tehnika koja se često koristi u video igrama je podešavanje razine detalja (*engl. level of detail*) svakog modela. Ponekad postoji nekoliko razina detalja za pojedinačni objekt. Što je igrač udaljeniji od modela, to će objekt imati manje detalja s obzirom da nije dovoljno blizu da prepozna razliku. Kako se igrač približava, model niže razlučivosti zamjenjuje se s detaljnijim. Potrebno je kreirati nekoliko inačica modela krenuvši od najdetaljnijeg. Potom se reducira broj rubnih petlji za svaki sljedeći LOD model. [11]



Slika 2.9. Modeli različitog LOD [12]

2.2. Optimizacija tekstura

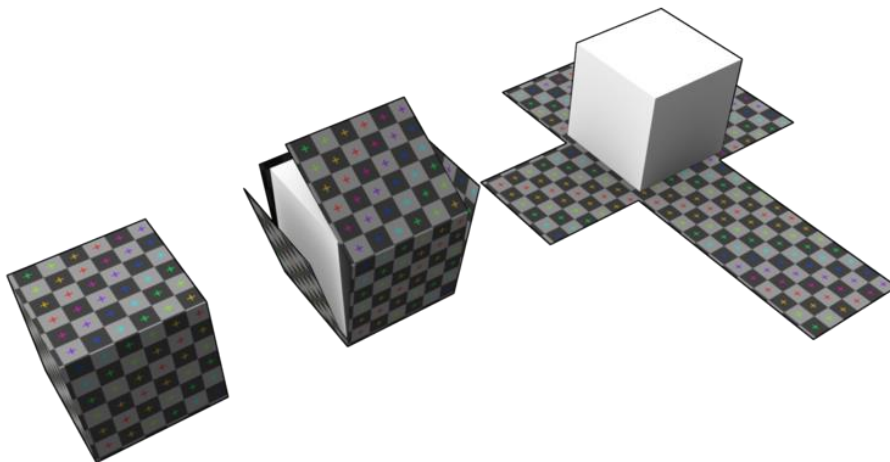
Kada je riječ o modeliranju za video igre, često se potrebno osloniti na detaljne teksture kako bi se poboljšala kvaliteta modela. Brojni se detalji ne trebaju modelirati, već se mogu dodati uz pomoć tekstura.



Slika 2.10. Jednostavni modeli s teksturama [11]

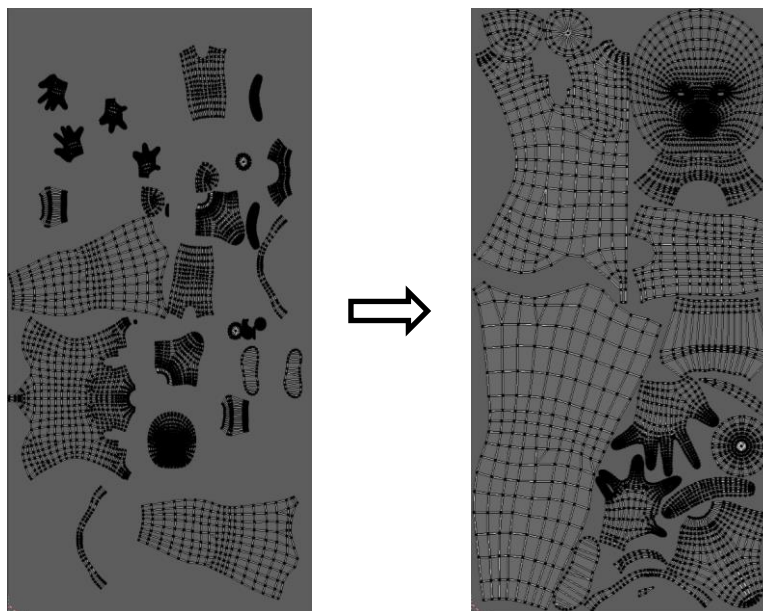
2.2.1. Optimizacija UV mapa

UV mape služe za projiciranje 2D tekstura na površinu 3D modela. Slova u i v označavaju osi 2D teksture jer se slova x , y i z već koriste za označavanje osi u 3D prostoru. Kako bi se 2D tekstura primijenila na 3D model, model je potrebno odmotati (*engl. UV unwrap*) na UV mapu. [13]



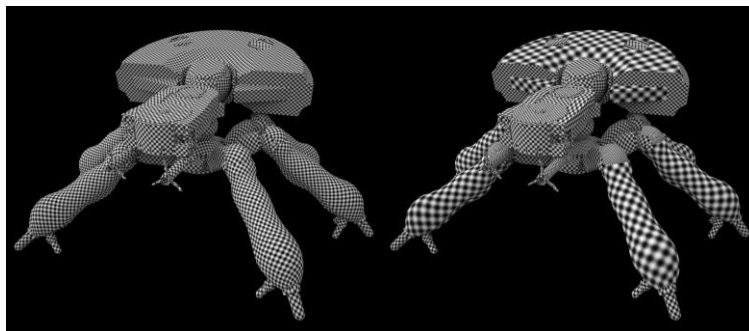
Slika 2.11. Vizualizacija UV odmotavanja [14]

Kako bi se iskoristila što veća površina 2D teksture, potrebno je odmotane dijelove modela složiti na UV mapu zadane rezolucije na što optimalniji način, što znači da ostane što manje praznog prostora. Sav prazan prostor su neiskorišteni pikseli koji ne služe ničem osim što iskoristavaju memoriju.



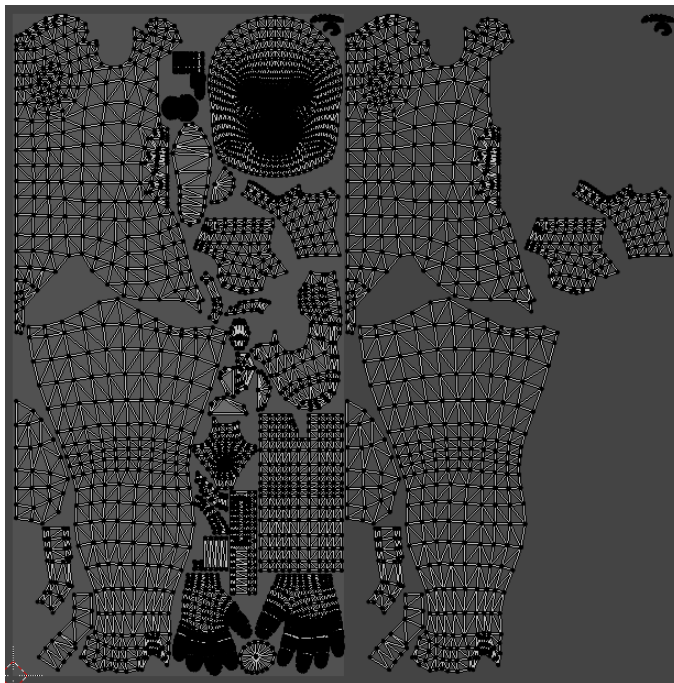
Slika 2.12. Optimalno iskorišten UV prostor

Poželjno je što više objekata odmotati na istu UV mapu kako bi bilo potrebno što manje teksturnih mapa, no to ovisi o rezoluciji koju želimo da svaki pojedinačni objekt ima. Dobra praksa je da svi objekti imaju istu rezoluciju teksture (*engl. texel density*), kako ne bi bilo dijelova s višom i manjom rezolucijom teksture. Ipak, ponekad se to pravilo može prekršiti ako npr. želimo da lice lika ima bistriju teksturu s više detalja nego ostatak tijela. Način na koji se određuje texel density pojedinog objekta je povećavanjem ili smanjivanjem UV otoka (*engl. UV island*), odnosno odmotanih dijelova objekata na UV mapi, kako bi pokrio veći ili manji prostor teksture. [15]



Slika 2.13. Ispravan (lijevo) i neispravan (desno) texel density [16]

Način na koji se UV prostor dodatno može iskoristiti je tako da se dijelovi objekta koji se ponavljaju ili koji su simetrični poslože jedan iznad drugoga. UV otoci se mogu zrcaliti i posložiti tako da se savršeno preklapaju i čitaju istu informaciju o teksturi.



Slika 2.14. UV mapa sa zrcaljenjem [17]

2.2.2. Optimizacija teksturnih mapa

Način na koji se najviše može uštediti memorija kada se radi o optimizaciji teksturnih mapa je smanjenjem rezolucije teksturne slike. Odabrana veličina teksturne slike uvelike ovisi o potrebama koje ta tekstura treba ispunjavati.

Za početak je potrebno objasniti pravilo biranja veličine koja je potencija broja 2 (*engl. the power of two rule*). Ovo pravilo nužno je pratiti kako bi razvojno okruženje u kojem se izrađuje video igra trošilo manje resursa na optimizaciju teksture, jer svaku teksturu čija veličina nije definirana po *power of two* pravilu treba prilagoditi potenciji broja 2. Iako su resursi potrošeni ispravljajući jednu teksturu možda minimalni, na razini cijele video igre stotine neispravnih tekstura može opteretiti razvojno okruženje. [18]

The power of two rule je skup jednostavnih kriterija koji osiguravaju da je veličina teksturne mape u standardiziranim dimenzijama. Drugim riječima, svaka slika koja je veličine 8, 16, 32, 64, 128, 256, 512, 1024, 2048 (ili više za modernije igre) piksela u smjeru širine i visine, smatra se valjanom i pravilno optimiziranom za brzo učitavanje u video igru i obradu u memoriji. Moderne računalne igre daju prednost kvadratnim (1:1) ili ograničenim pravokutnim (2:1 i 1:2) dimenzijama radi jednostavnije i brže obrade. [18]

Na sljedećoj slici je prikazana vizualizacija utjecaja pravila potencije broja 2 na obradu slike u razvojnom okruženju. Na lijevoj strani su teksture ispravne, a s desne strane su teksture neispravne veličine. Razlika veličine teksture s lijeve i desne strane označena je crvenom bojom. Ta površina predstavlja količinu prostora koju bi tekstura trebala zauzeti ali nije, što prisiljava postojeću teksturu da se prilagodi na valjanu veličinu. [18]

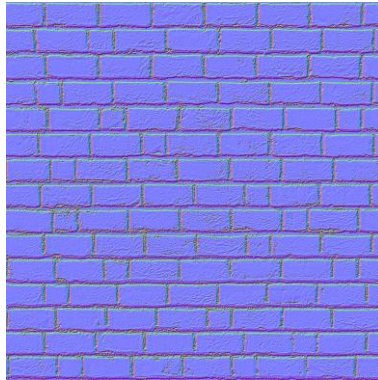


Slika 2.15. Vizualizacija količine prostora koja se treba obraditi [18]

Sljedeća stvar bitna pri određivanju veličine teksturne mape za pojedini objekt je njegova svrha u video igri. Bitno je odlučiti kakva je „bistrina“, odnosno kvaliteta teksture potrebna za koji objekt, kako se resursi ne bi nepotrebno trošili na teksture visokih rezolucija na malim ili jako udaljenim objektima. Pojam *texel density* u ovom slučaju ponovno postaje relevantan jer prikazuje vrijednost broja piksela po jednom centimetru. Što je veći *texel density* na objektu, to je veća kvaliteta teksture na tom objektu. *Texel density* dobije se dijeljenjem broja piksela na teksturi po duljini jednog centimetra na prikazu objekta u video igri. U strateškim video igrama objekti su vrlo udaljeni i dovoljan je *texel density* od 1.28 px/cm. U video igrama s kamerom u trećem licu potreban je nešto veći *texel density* od 5.12 px/cm, dok su u video igrama s kamerom u prvom licu objekti najbliži igraču i potreban je najveći *texel density* od 10.24 px/cm. Ove vrijednosti nisu univerzalne, ali prikazuju razliku u prosječnim *texel density* vrijednostima prema žanru video igre. Najčešće rezolucije tekstura u video igrama se kreću od 512 x 512 px do 2048 x 2048 px, a u modernim video igrama mogu biti i veće. [15]

2.2.3. Optimizacija upotrebom mapa normala

Mapa normala je slika koja pohranjuje informacije o površinskom nagibu na geometriji skulpture za svaki piksel. Crveni, zeleni i plavi kanal slike kontroliraju smjer normale svakog piksela. Mapa normala primjenjuje se na model nižeg broja poligona i stvara dojam detalja koji postoje na skulpturi. [19]



Slika 2.16. Mapa normala [20]

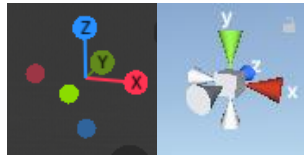
Model s jednakim brojem poligona može djelovati puno detaljnije upotrebom mape normala. Na taj način se dobivaju potrebni detalji bez smanjivanja performansi zbog prevelikog broja poligona.



Slika 2.17. Model niske razlučivosti s mapom normala i model visoke razlučivosti [21]

3. Alati korišteni u izradi lika za video igru

U praktičnom dijelu rada korišteni su alati Blender i Unity jer su besplatni, vrlo široko korišteni i iznimno kompatibilni. Osim toga, oba se alata neprestano ažuriraju i nedavno su redizajnirani za još bolje iskustvo. Unity pruža mogućnost uvoženja izvornih .blend datoteka izrađenih u alatu Blender. To znači da automatski prepoznaje model, materijale i teksture, armaturu, animacije i ostalo što sadrži uvezena .blend datoteka. Jedna razlika na koju je potrebno obratiti pozornost je razlika u koordinatnim sustavima između ova dva alata. Dok je u alatu Blender os z usmjerena prema gore, u alatu Unity je to os y. [22]



Slika 3.1. Razlika koordinatnih sustava u Blender-u (lijevo) i Unity-u (desno)

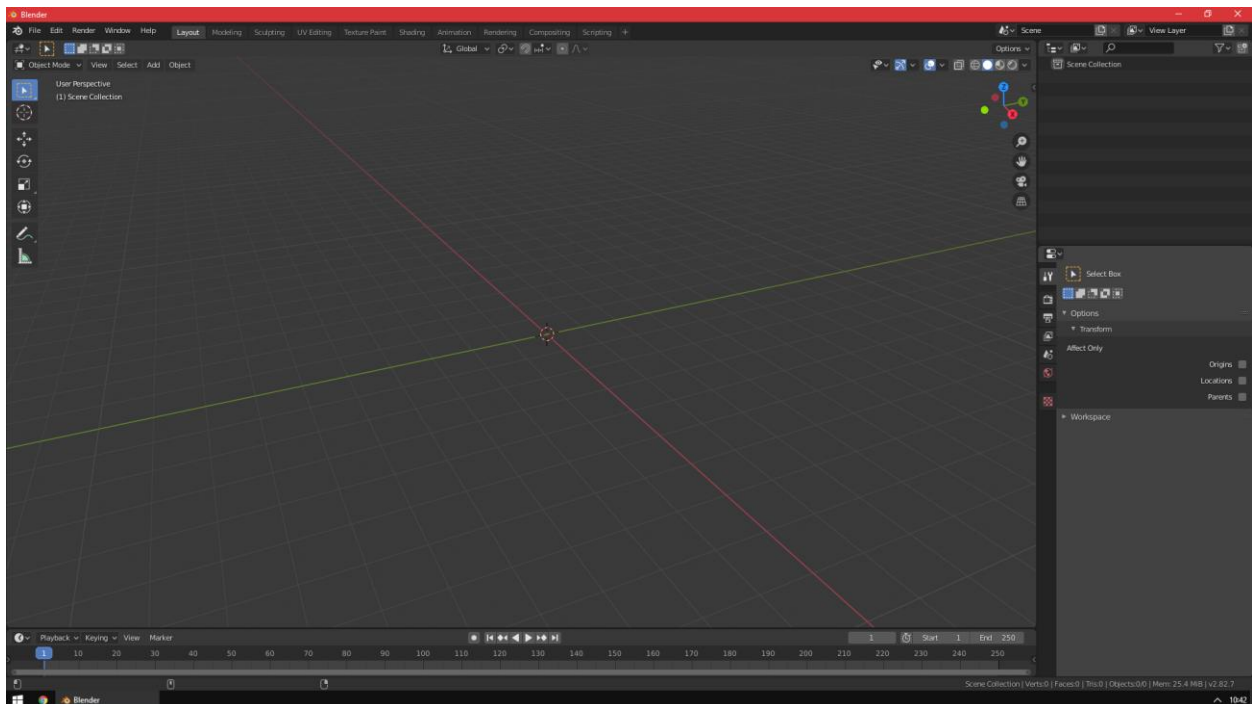
3.1. Blender 2.8

Blender je besplatni i open source 3D alat za stvaranje koji se može koristiti u bilo koje svrhe, uključujući komercijalne i obrazovne svrhe. U cijelosti obuhvaća procese 3D stvaranja: modeliranje, skulpturiranje, animiranje, simulacije, renderiranje, kompozitiranje, video obradu i izradu video igara. Napredni korisnici koriste Blenderov API za Python skripte kako bi prilagodili aplikaciju i izradili specijalizirane alate.



Slika 3.2. Blender logo [23]

Blender je dobro prilagođen pojedincima i malim studijima kojima je pogodan jedinstveni proces izrade. Javnost je ovlaštena za izradu malih i velikih promjena u kodu Blender-a, što dovodi do mnoštva novih alata i ispravki programskih pogrešaka. Institucija Blender trenutno zapošljava petnaestoro ljudi koji rade na softveru i kreativnim projektima tvrtke Blender kako bi testirali Blender u proizvodnim okruženjima. [23]



Slika 3.3. Blender korisničko sučelje

Blender 2.80 sadrži redizajnirano korisničko sučelje koje se sastoji od tamne teme i modernog seta ikona. Korisničko sučelje intuitivnije je nego ikad. Sadrži izbornike koji omogućuju brz pristup često korištenim alatima i potpuno nove radne prostore za brzo započinjanje izrade skulptura, tekstura i ostalog. Mogu se prilagoditi za stvaranje vlastitog učinkovitog radnog okruženja. Također, u sklopu Blender-a 2.8 predstavljen je novi Eevee renderer koji je fizički utemeljen renderer u stvarnom vremenu. Sadrži napredne značajke kao što su refleksije, rasipanje svjetlosti ispod površine, meke i kontaktne sjene, zamagljivanje pokreta kamere itd. Eevee materijal Principled BSDF kompatibilan je s mnogim razvojnim alatima za video igre. [24]

Kao i svaki program, Blender ima svoje prednosti i mane s obzirom na druge 3D softvere. Najbitnije prednosti alata Blender su te što je softver besplatan i što se jako široko koristi tako da se lako mogu pronaći video uputstva i potrebna literatura. Također, cijeli proces izrade 3D lika za video igru može se napraviti koristeći isključivo Blender, a pritom je i vrlo dobro optimiziran za daljnji rad s modelom u programu Unity.

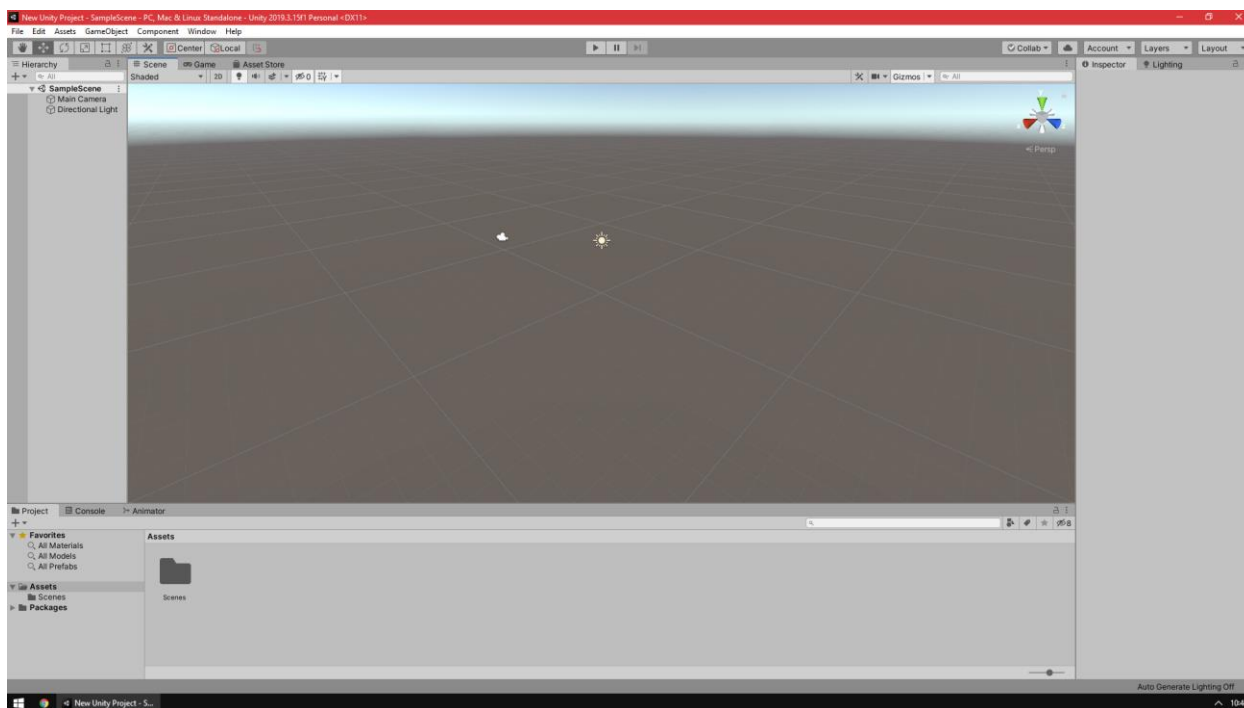
3.2. Unity 2019

Unity je razvojni alat za više platformi koji je razvio Unity Technologies, a prvi put je najavljen i objavljen u lipnju 2005. Posljednja stabilna verzija, Unity 2019.3, objavljena je u svibnju 2020. godine. [25]



Slika 3.4. Unity logo [26]

Alat podržava više od 25 platformi. Može se koristiti za stvaranje trodimenzionalnih, dvodimenzionalnih i VR igara, kao i raznih simulacija i drugih iskustava. Unity alat su usvojile industrije i izvan videoigara, poput filmske, automobilske, inženjerske i građevinske. Alat nudi primarni skriptni API za C# programski jezik. Za 3D video igre Unity omogućuje kompresiju tekstura i postavke razlučivosti za svaku platformu koju alat podržava, a pruža i podršku za bump mape, mape refleksije, okluziju okoline itd. Od 2018. godine Unity se koristi za kreiranje približno polovice novih mobilnih igara na tržištu i za 50% video igara općenito. [27]

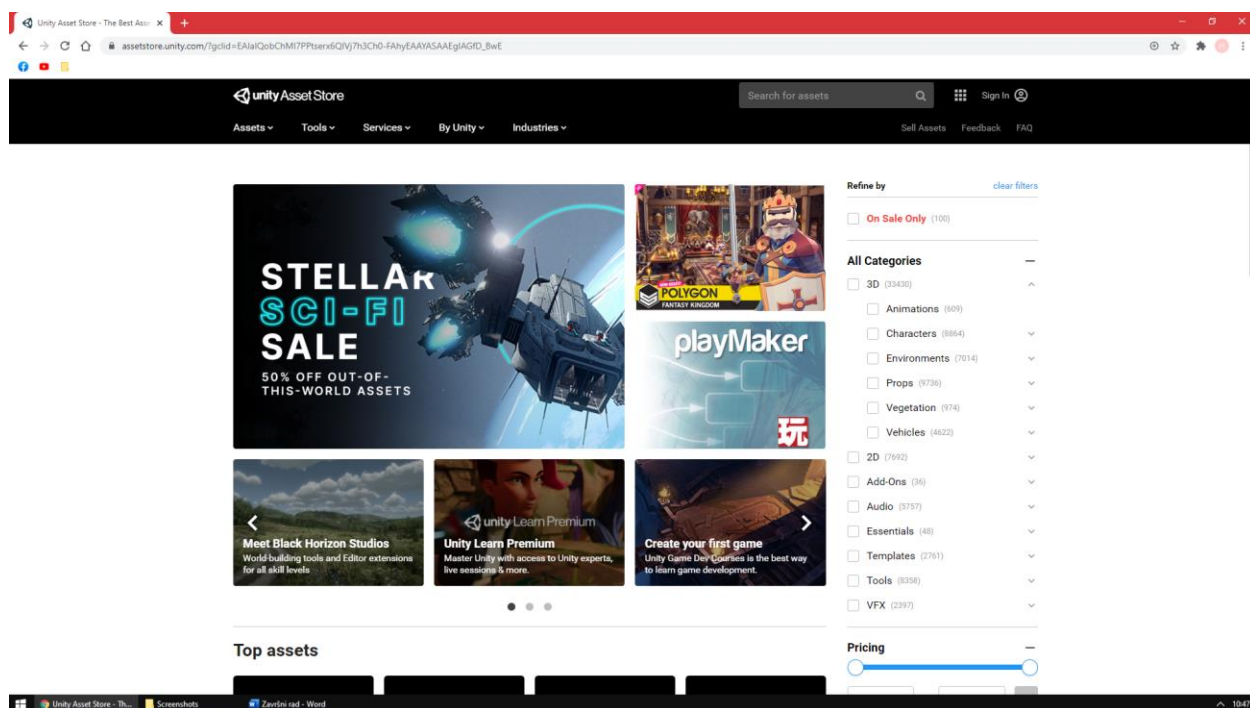


Slika 3.5. Unity korisničko sučelje

Besplatna verzija Unity razvojnog okruženja Unity Personal sadrži najnoviju inačicu osnovne razvojne platforme Unity i resurse za započinjanje projekta i učenje o raznim mogućnostima unutar alata. Kriteriji za korištenje besplatne verzije Unity alata su:

- ostvareno je manje od 100 000 USD godišnjeg bruto prihoda, bez obzira koristi li se u komercijalne svrhe ili interni projekt
- nisu prikupljena sredstva za izradu projekta veća od 100 000 USD
- trenutno se ne upotrebljava Unity Plus ili Pro verzija

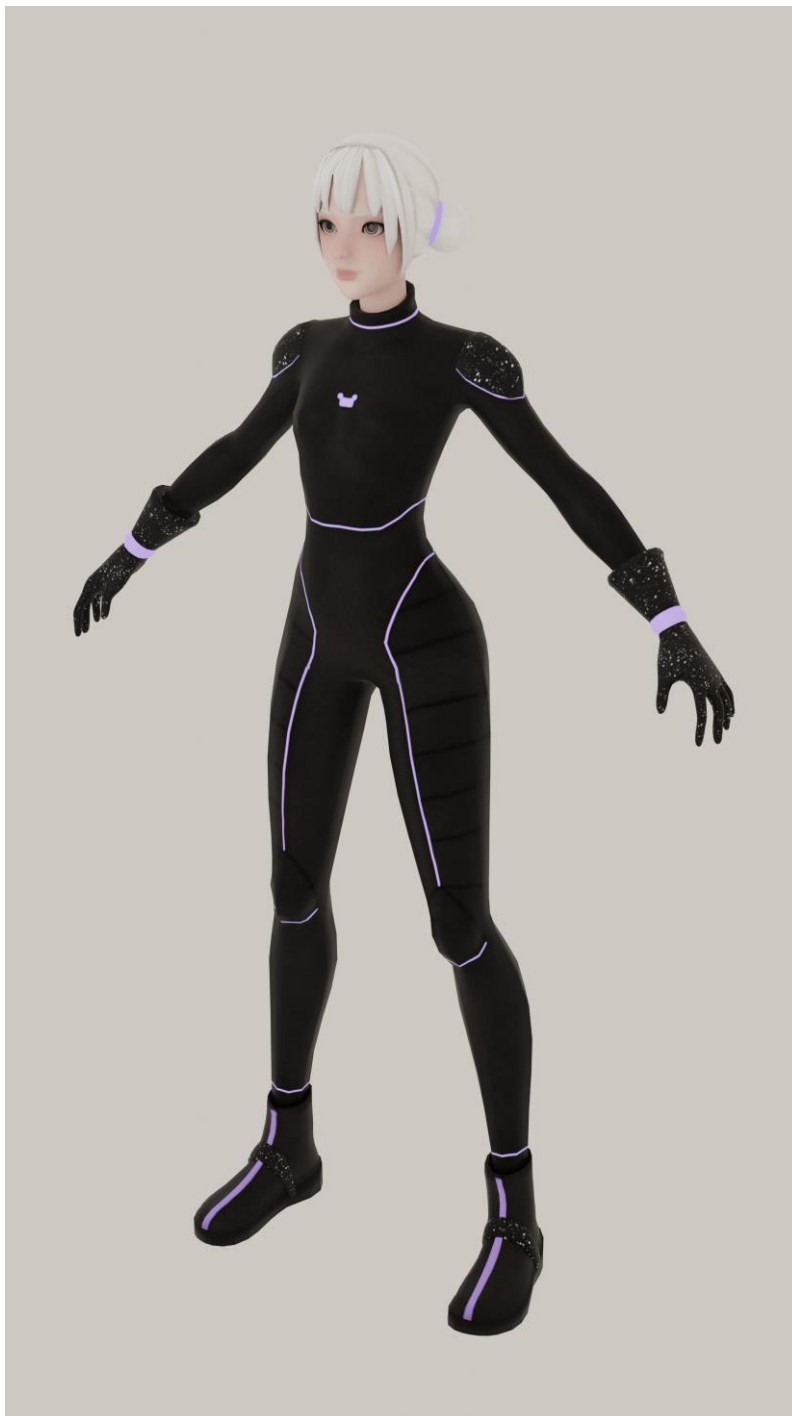
Unity Asset Store je online trgovina na kojoj kreatori mogu razvijati i prodavati imovinu (*engl. asset*) za slobodnu upotrebu drugim proizvođačima video igara. To uključuje 3D i 2D objekte, animacije i alate koji se mogu kupiti ili besplatno preuzeti. Unity Asset Store pokrenut je 2010. godine, a do 2018. godine kroz digitalnu trgovinu bilo je oko 40 milijuna preuzimanja.



Slika 3.6. Unity Asset Store

4. Proces izrade lika za video igru

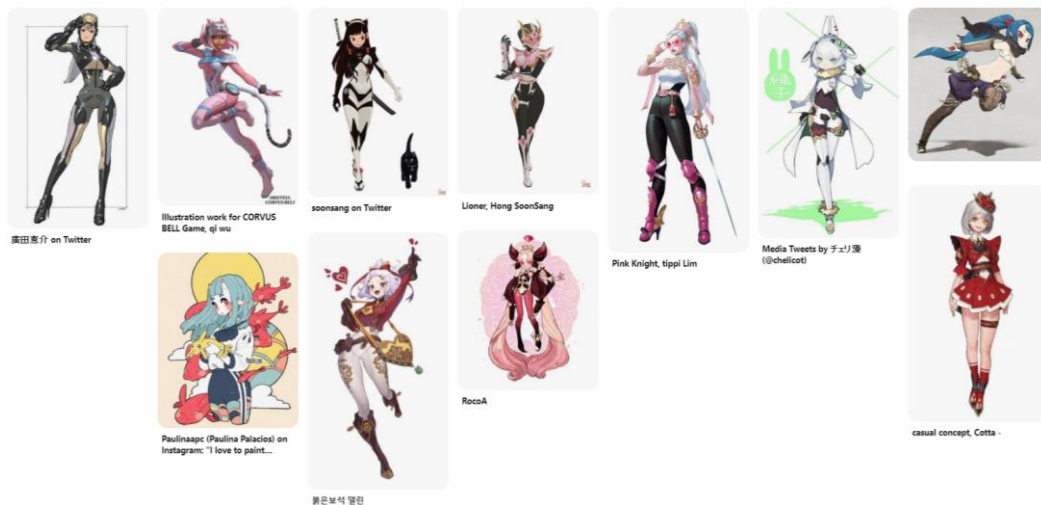
U praktičnom dijelu rada izrađen je humanoidni 3D lik za video igru. Proces izrade modela podijeljen je na izradu referenci, izradu skulpture, optimizaciju modela, izradu teksturnih mapa i materijala i dodavanje kostura kako bi se model mogao animirati. Nakon toga se izrađuje demonstracija gotovog lika u video igri koja je napravljena u razvojnom okruženju Unity.



Slika 4.1. Gotov model lika

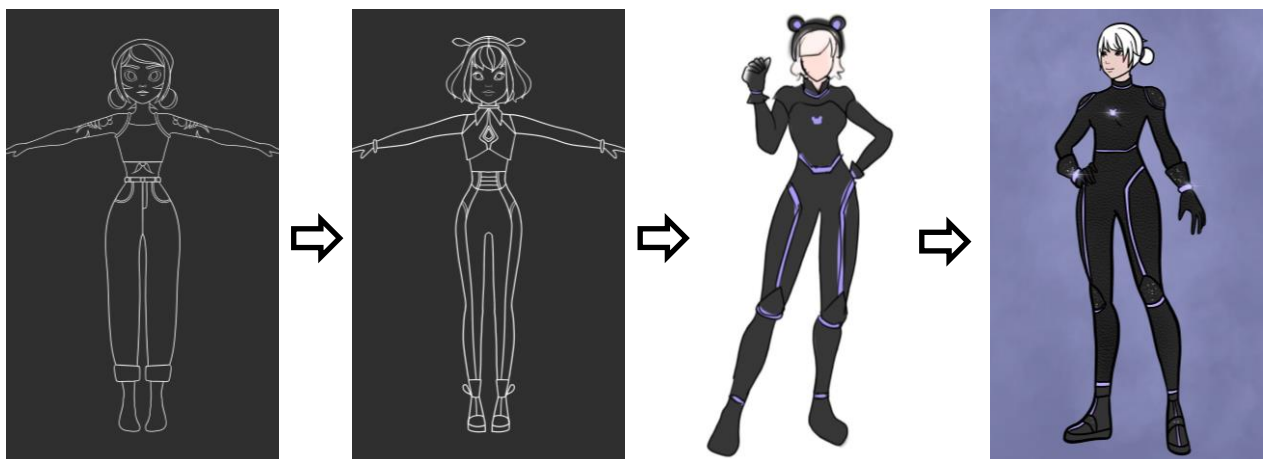
4.1. Reference

Prvi korak u izradi bilo čega je stvaranje ideje o tome što želimo stvoriti. Lik za video igru koji će se izraditi je mlada djevojka bijele kose, smeđih očiju, u crnom odijelu kojeg ističu svijetleće ljubičaste linije. Pojam *concept art* odnosi se na ilustracije nekog objekta, bića ili okoline koji vizualiziraju ideju. Kako bi se izradio dobar concept art, često je vrlo korisno napraviti kolekciju slika koje nas inspiriraju i daju ideju o tome kako bi naš concept art trebao izgledati.



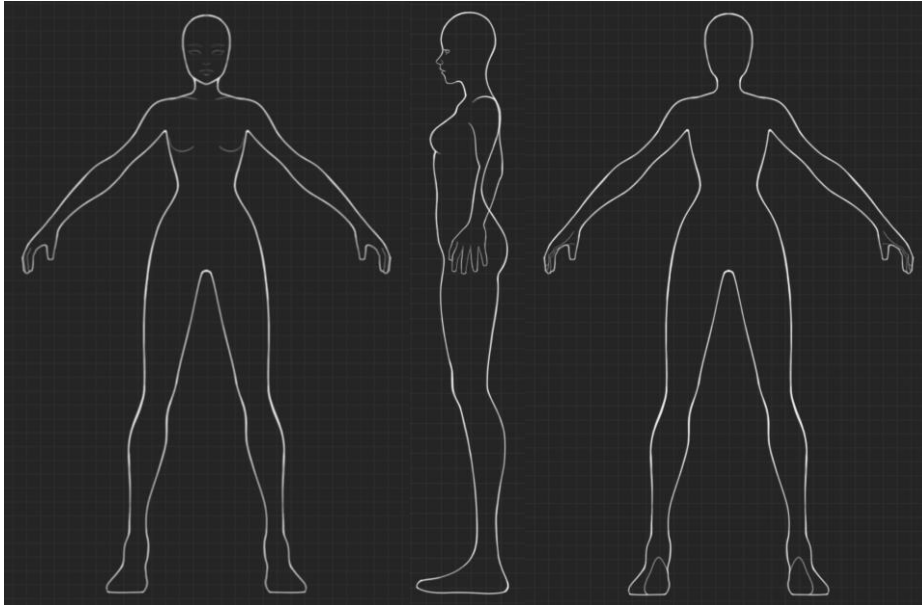
Slika 4.2. Kolekcija slika za inspiraciju

Potom se ilustriraju razne ideje. Ponekad je potrebno proći kroz mnogo iteracija dok se ne dođe do završnog concept art-a s kojim smo zadovoljni. Na sljedećoj slici su prikazane faze izrade concept art-a za lika koji će se izrađivati u praktičnom dijelu ovog rada.

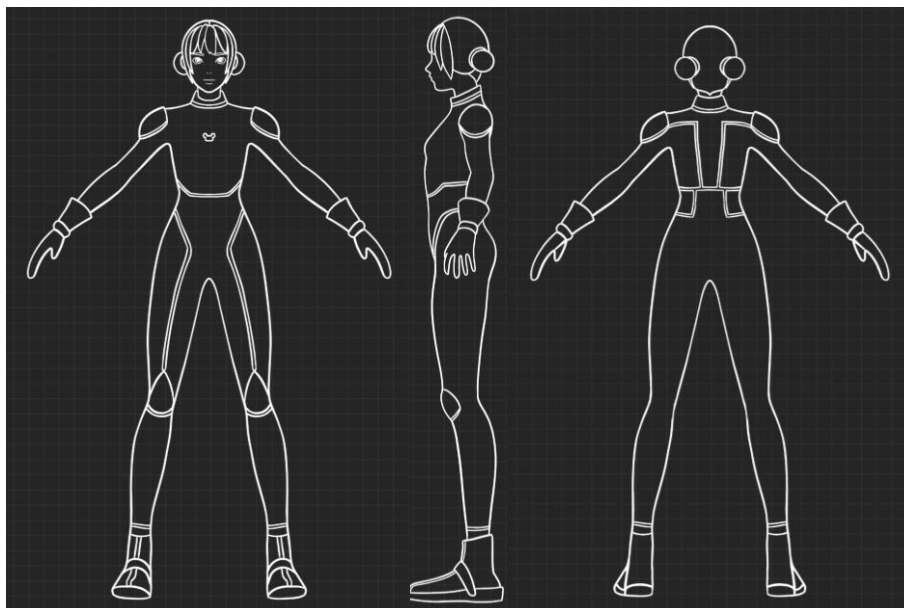


Slika 4.3. Iteracije za concept art

Nakon što postoji jasna ideja o tome kako će lik izgledati potrebno je izraditi referentne slike koje su upotrebljive u izradi lika u 3D prostoru. Dakle, potrebno je lika nacrtati u pogledu sprijeda, odostraga i sa strane. Izradile su se referentne slike s obrisom tijela bez ikakvih detalja, kako detalji ne bi smetali u procesu skulpturiranja tijela. Posebno su izrađene i referentne slike sa svim detaljima koje će lik imati. Te se slike uvoze u Blender projekt u kojem će se izrađivati lik. [28]



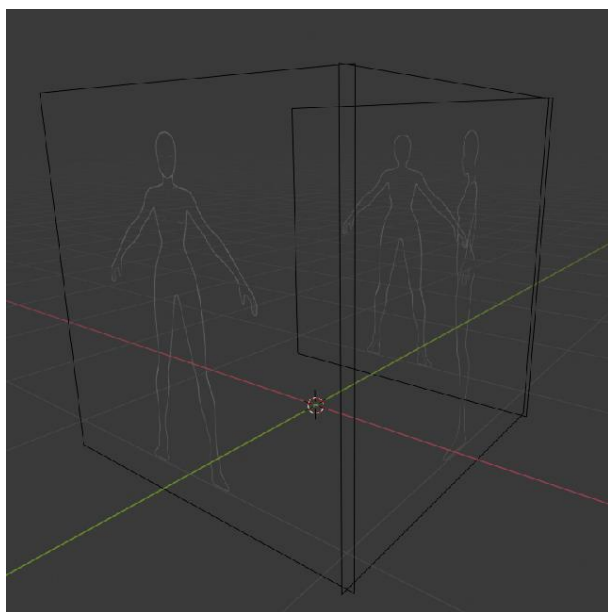
Slika 4.4. Reference bez detalja



Slika 4.5. Reference s detaljima

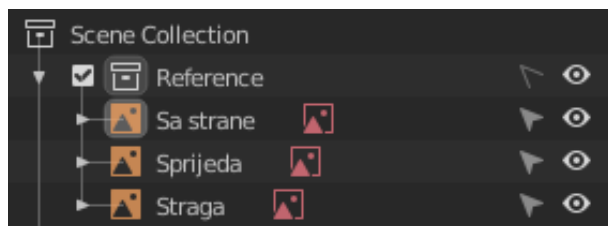
Pokretanjem programa Blender automatski se kreira nova scena koja sadrži kocku, kameru i svjetlo. Za početak, obrisano je sve što se nalazilo u početnoj sceni. Naredbom *Add > Image > Reference* u praznu scenu su dodane 3 reference za proporcije lika sprijeda, odostraga i sa strane.

U *Object Data Properties* prozoru moguće je povećati prozirnost slike klikom na opciju *Use Alpha* i promjenom vrijednosti za *Opacity* (neprozirnost) slike. Referencama je potrebno povećati prozirnost kako bi se kroz bijele obrise tijela na slikama mogli vidjeti rubovi 3D modela kojega se modelira prema ovim referencama. Reference se postavljaju na način da se može vidjeti kako model treba izgledati u pogledu sprijeda, odostraga i s boka.



Slika 4.6. Reference u 3D prostoru

Outliner prozor sadrži popis svih objekata koji se nalaze u sceni. Svaki projekt je bitno dobro organizirati tako da se objekti imenuju i grupiraju u kolekcije kako bi se više objekata odjednom moglo označiti, sakriti, ili kako bi im se omogućila / onemogućila selekcija. Napravljena je nova kolekcija koja sadrži samo reference i onemogućena joj je selekcija.



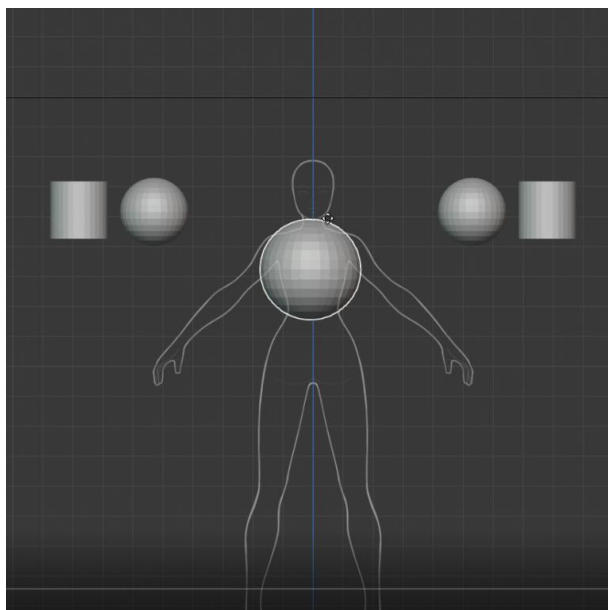
Slika 4.7. Organizacija scene

4.2. Skulptura

Proces skulpturiranja podijeljen je u četiri faze: priprema za izradu referentnog modela, izrada referentnog modela, priprema za izradu skulpture i izrada skulpture.

4.2.1. Priprema za izradu referentnog modela

Gotovo se svaki dio ljudskog tijela može prikazati kuglama ili valjcima. Ruka se na primjer može pojednostaviti na valjke za nadlakticu i podlakticu, a kuglu za lakat. Zato se ti objekti mogu koristiti kako bi se započelo s modeliranjem tijela. Naredbama *Add > Mesh > Cube / UV Sphere* dodane su dvije kugle i jedan valjak. Prva kugla služi za oblikovanje dijelova torza. Druga kugla i valjak služe za oblikovanje udova, a na sebi imaju dodan *Mirror Modifier*.



Slika 4.8. Priprema za referentni model

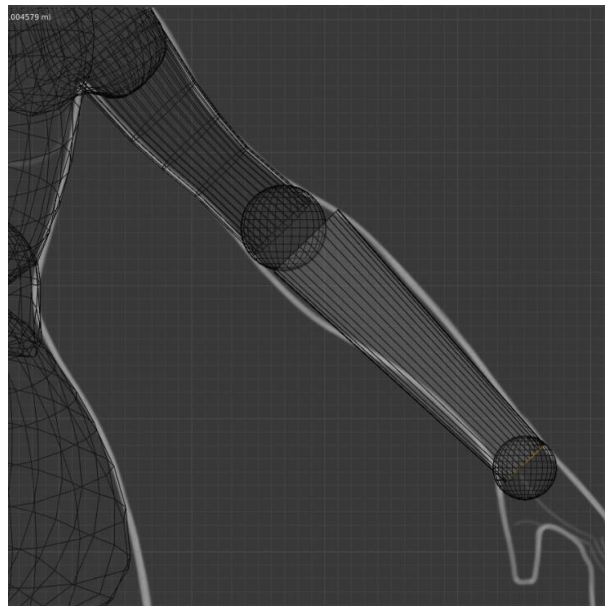
Mirror Modifier jedan je od mnogih modifikatora u sklopu alata Blender. Ovaj modifikator zrcali geometriju objekta s pozitivne strane odabrane osi na negativnu. S obzirom da je ljudsko tijelo simetrično, upotrebom ovog modifikatora dovoljno je modelirati samo jednu polovicu tijela. Nalazi se u *Modifier Properties* prozoru i vrlo se jednostavno može dodati na selektirani objekt. Svaki modifikator sadrži brojne opcije. U ovom slučaju, pozornost je potrebno obratiti samo na *Mirror Object* opciju. Riječ je o objektu čija lokacija i rotacija definira plohu prema kojoj se objekt

zrcali. U ishodište koordinatnih osi dodan je objekt bez geometrije naredbom *Add > Empty > Plain Axes*, preimenovan je u *Mirror* i dodan kao *Mirror Object* u modifikator. Na taj način su dobiveni zrcaljeni objekti koji se lako mogu duplicirati i nije potrebno za svaki novi valjak ili kuglu ponovno dodavati *Mirror Modifier*. Time je završena priprema za izradu referentnog modela. [32]

4.2.2. Izrada referentnog modela

Referentni model je izgrađen od dupliciranih i deformiranih pripremljenih kugli i valjaka. Korišten je jedan tip deformiranja za valjke, a drugi za kugle, no za oba su korišteni osnovni alati promjene lokacije, rotacije i skaliranja kako bi objekti otprilike dobili željenu poziciju u prostoru.

Kako bi se deformirali valjci na način da prate prirodnu zakrivljenost listova, nadlaktica i sl., selektiran je jedan od valjaka u *Object Mode* načinu interakcije, a potom je uključen *Edit Mode*. *Edit Mode* dozvoljava rad sa selektiranim verteksima, rubovima ili licima. Sadrži i brojne korisne alate za manipulaciju s geometrijom objekta. U ovom slučaju je korišten *Loop Cut* alat koji dodaje rubnu petlju između postojećih rubnih petlji. Nova rubna petlja zatim se može smanjiti ili povećati s obzirom na izgled koji se želi postići.



Slika 4.9. *Edit Mode* način rada

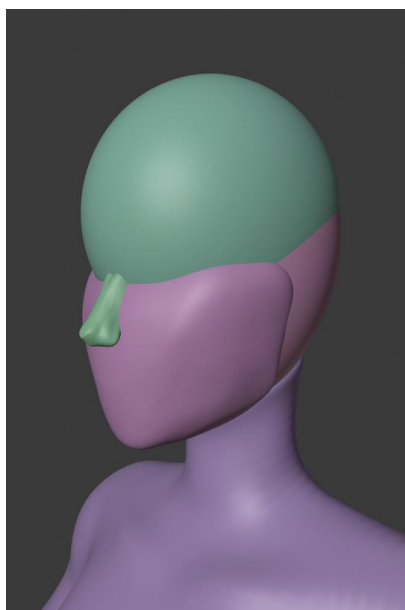
Kugle se deformiraju na nešto drugačiji način. Selektirana je kugla u *Object Mode* načinu i uključen je *Sculpt Mode*. *Sculpt Mode* je način interakcije s objektom specijaliziran za skulpturiranje, što znači da sadrži mnoštvo kistova za manipulaciju s objektom. U fazi izrade referentnog modela korišten je samo *Grab* kist koji hvata i povlači dio geometrije određen radijusom i jačinom kista.

Referentni model je gotov onda kada smo zadovoljni sa siluetom i generalnim oblikom modela. U ovoj fazi detalji nisu bitni, kao ni urednost topologije. Dobrodošlo je i odskakanje od početne reference jer jednom kada 2D sliku prenesemo u 3D prostor, proporcije nam mogu djelovati čudno. Slobodni smo raditi velike promjene u silueti lika, što je u kasnijim fazama mnogo teže.



Slika 4.10. Referentni model tijela

Za referentni model glave korištene su kugle koje su *Grab* kistom deformirane na način da izrade početni izgled glave.



Slika 4.11. Referentni model glave

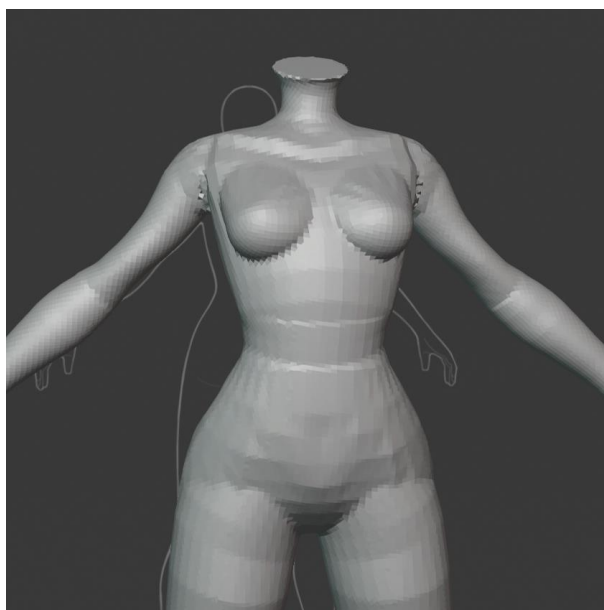
4.2.3. Priprema za izradu skulpture

Iako referentni model daje dobru ideju o tome kako će model na kraju izgledati, ta geometrija neiskoristiva je za skulpturiranje iz nekoliko razloga:

- tijelo se sastoji od više odvojenih objekata
- velik dio geometrije nalazi se unutar tijela
- geometrija nije pripremljena na način da opcija za simetriju na kistovima pravilno radi
- geometrija se ne sastoji od približno jednakih četverokuta

Prije nego što se može početi skulpturirati, trebaju se riješiti ti problemi. Za početak je potrebno selektirati sve objekte od kojih se tijelo sastoji i naredbom *Object > Join* spojiti ih u jedan. No, samo spajanje objekata ne rješava se geometrije koja se nalazi unutar njih, odnosno geometrije koja se ne vidi. To često izaziva probleme kod skulpturiranja i onemogućava izradu glatkih prijelaza između objekata. *Remesh* alat koji se nalazi u zaglavlju *Sculpt Mode* načina interakcije

služi upravo tome da generira novu topologiju prema vidljivom dijelu modela, čime se gubi nepotrebna unutarnja geometrija. Koristeći zadane postavke, potrebno je odabrati opciju *Remesh*.



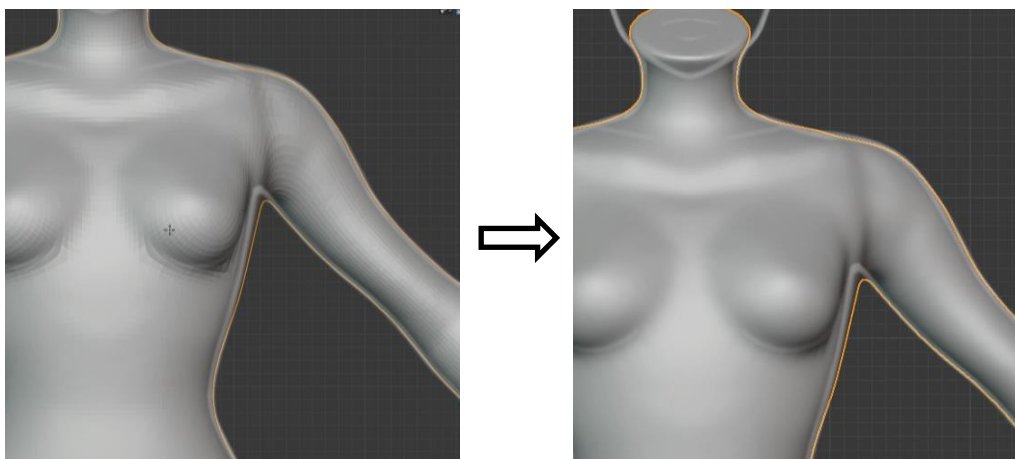
Slika 4.12. Geometrija nakon opcije *Remesh*

Kako bi opcija za simetriju na kistovima za skulpturiranje pravilno radila, vrlo je bitno primijeniti dvije opcije na dobiveni model. Obje se opcije nalaze u *Object Mode* načinu rada. Prva opcija premiješta težište modela na isto mjesto na kojem je *Mirror Object* koji se koristi za *Mirror Modifier*, a primjenjuje se naredbom *Object > Set Origin > Origin to 3D Cursor*. Druga opcija primjenjuje na objekt sve transformacije koje su na njemu rađene i vraća vrijednosti lokacije i rotacije na 0, a veličine na 1. Primjenjuje se naredbom *Object > Apply > All Transforms*.

Iako je u ovoj fazi model donekle spreman za skulpturiranje, bio bi bolji kada bi geometrija bila sačinjena samo od četverokuta. *Remesh* alat generirao je geometriju koja se sastoji od trokuta i četverokuta. Trokuti nisu poželjni u skulpturi modela jer mogu izazvati neočekivane probleme kod skulpturiranja s nekim kistovima i uzrokovati stezanje dijela geometrije koje je vrlo očito i loše izgleda.

Kako bi se generirala topologija koja se sastoji samo od četverokuta, koristi se besplatna probna verzija proširenja za Blender zvana *Quad Remesher*. [29] Proširenja za Blender su skripte pisane Pythonom koje na osnovne funkcionalnosti programa nadodaju nove funkcije, elemente korisničkog sučelja, operatore i sl. Blender je *open-source* program koji dopušta korisnicima da kreiraju vlastita proširenja, zbog čega postoji mnoštvo besplatnih proširenja za razne potrebe.

Proširenje se lako dodaje u Blender klikom na *Edit > Preferences > Add-ons > Install*. Nakon instalacije *Quad Remesher* proširenja, potrebno je selektirati model tijela i podesiti opcije za generiranje geometrije. *Quad Count* odnosno broj četverokuta treba postaviti na 100 000 kako bi se stvorio vrlo detaljan model tijela, pogodan za skulpturiranje. Zatim je uključena opcija simetrije po x osi i klikom na gumb *REMESH IT* generirala se nova geometrija. Postoje dva načina na koje objekt može biti sjenčan: *Smooth* (glatki) i *Flat* (ravni). Klikom na opciju *Object > Shade Smooth* završena je priprema modela za skulpturiranje.



Slika 4.13. Razlika između Flat i Smooth načina sjenčanja

4.2.4. Izrada skulpture

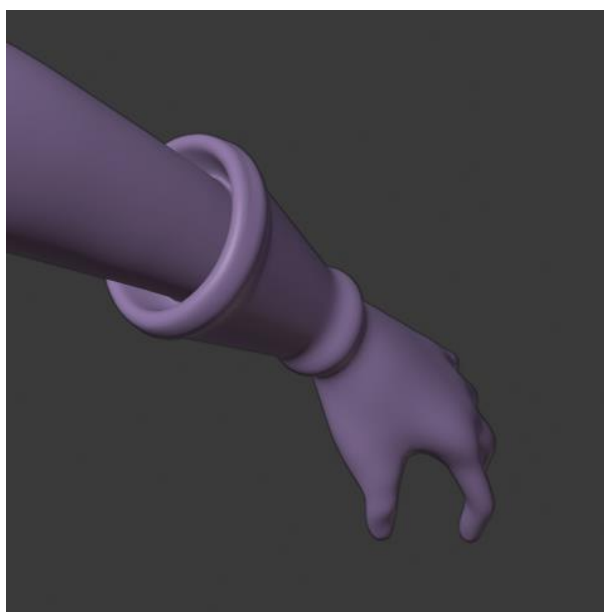
Izrada skulpture podrazumijeva dodavanje detalja na model. Posebnu pažnju potrebno je obratiti na izgled koljena, laktova, dlanova i stopala s obzirom da su ti dijelovi u fazi izrade referentnog modela bili zanemareni. Dodatno su doručeni detalji na leđima, listovima i ključnoj kosti. I dalje su korištene reference kako se skulptura ne bi previše razlikovala od skice lika.

Kist koji je najviše bio korišten je *Draw* (crtanje). Taj kist, kao i mnogi ostali, može dodavati geometriju klikom na gumb +, ili ju oduzimati klikom na gumb -. Također je moguće postaviti radijus i jačinu djelovanja kista. Korišteni su i kistovi *Crease* (nabiranje), *Smooth* (uglađivanje) i *Grab* (hvatanje). Kist *Crease* nabire geometriju prema unutra. Korišten je kod skulpturiranja dijela usta na kojem se spajaju gornja i donja usna i kod oblikovanja nosnica. Kist *Smooth* je vrlo koristan kist koji uglađuje površinu i smanjuje njene nepravilnosti.



Slika 4.14. Gotova skulptura tijela i glave

Zatim je u *Object Mode* načinu rada dodan valjak koji je pomoću *Extrude Region* alata oblikovan u rukavicu. *Extrude Region* alat stvara i pomiče novu geometriju po zadanoj normali. Nova geometrija stvorena pomoću tog alata je potom smanjena ili povećana s obzirom na stožasti oblik rukavice kojeg je trebalo postići. Zatim je na objekt rukavice dodan modifikator *Mirror*, ali i modifikator *Subdivision Surface*. *Subdivision Surface* modifikator dodaje novu geometriju između već postojeće i time ju proglašuje što stvara mekše prijelaze i glađi izgled. Potom je rukavica spojena s ostatkom tijela i Smooth kistom je ugladen prijelaz između tijela i ovratnika.



Slika 4.15. Skulptura rukavice

Oblik ovratnika također je izrađen pomoću valjka koji je oblikovan po želji na način na koji je bila oblikovana i rukavica.



Slika 4.16. Skulptura ovratnika

Izrada čizmi započela je s dodavanjem ravnine u scenu naredbom *Add > Mesh > Plane*. Izbrisani su svi osim jednog verteksa. Preostali verteks je postavljen tamo gdje bi trebao biti vrh nožnog palca i *Extrude Region* alatom dodani su novi verteksi koji su okružili stopalo. Prvi i posljednji verteks su spojeni tako što su selektirani, a zatim je odabrana naredba *Vertex > Merge Vertices > At Center*. Dodana su lica između verteksa i dobivena je ploha nalik na trag kojeg na zemlji ostavlja čizma. Zatim su selektirane sve plohe i alatom *Extrude Region* im je dodana debljina i dobiven je đon čizme.

Kako bi se izradio ostatak čizme, treba selektirati skulpturu tijela i u *Edit Mode* načinu rada uključiti selekciju lica i selektirati alatom *Select Box* samo lica koja grade lijevo stopalo i gležanj. Kako selekcija ne bi bila ograničena samo na vidljiva lica koja se nalaze sprijeda, potrebno je odabrati *Toggle X-Ray* ili *Wireframe* opciju u načinima prikaza modela, jer ti načini prikaza dopuštaju selekciju svih lica koji se nađu u *Select Box* alatu. Nakon što su stopalo i gležanj selektirani, potrebno ih je duplirati naredbom *Mesh > Duplicate*, a zatim ih odabirom na *Mesh > Separate > Selection* odvojiti od skulpture tijela čime to postaje zasebni objekt. Potom se taj objekt poveća *Scale* alatom do željene veličine čizme.

Debljina gornjeg dijela čizme dobije se na način da se u *Edit Mode* načinu rada odabere gornji rub čizme i alatom *Inset Faces* kreira novi rub čiji se radijus smanji *Scale* alatom. Ponovnim korištenjem *Inset Faces* alata kreira se još jedan rub koji se *Move* alatom premjesti prema dolje kako bi se stvorio dojam šupljeg prostora u čizmi. Detalj na prednjem dijelu čizme dobiven je selekcijom linije lica na prednjem dijelu čizme i korištenjem alata *Extrude Region*.

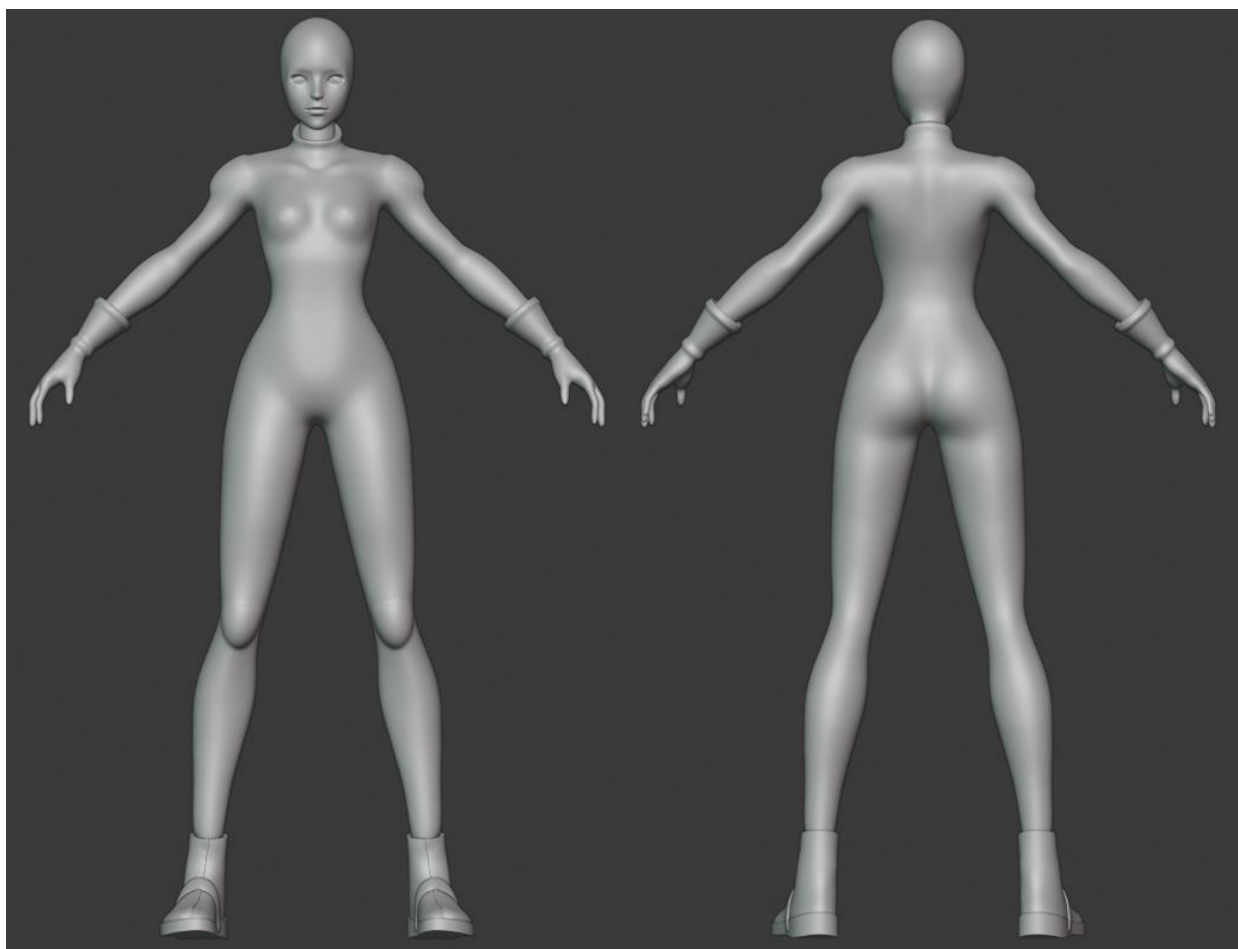
Pojas preko čizme izrađen je korištenjem krivulja. U scenu su naredbama *Add > Curve > Bezier* i *Add > Curve > Circle* dodani Bezierova krivulja i kružnica. Odabirom na Bezierovu krivulju, u *Object Data Properties > Geometry* prozoru moguće je dodati kružnicu kao *Bevel Object*. *Bevel Object* je objekt koji krivulji daje debljinu s obzirom na svoj oblik i veličinu. Deformiranjem kružnice mijenja se i oblik Bezierove krivulje.

Kako bi se dobio željeni oblik pojasa, kružnicu je potrebno oblikovati u oblik graha, selekcijom jednog od 4 segmenta u *Edit Mode* načinu i pomicanjem istog bliže centru kružnice. Bezierovoj krivulji u *Edit Mode* načinu potrebno je dodati još segmenata za kontrolu krivulje selektiranjem već postojećih i odabirom naredbe *Segments > Subdivide*. Zatim je moguće pomicanjem segmenta prilagoditi oblik krivulje da prati oblik čizme. Kako bi određeni dijelovi pojasa bili širi ili uži, odabire se segment na tom dijelu i alatom *Radius* mijenja se njegova širina.



Slika 4.17. Skulptura čizmi

U Sculpt Mode načinu rada korištenjem kista *Draw* dodani su detalji za štitnike za koljena i ramena. Time je završena faza izrade skulpture za lika za video igru.



Slika 4.18. Gotova skulptura lika

4.3. Optimizacija modela

Nakon završetka izrade skulpture s visokom rezolucijom poligona i mnoštvom detalja, bitno je kreirati novi model po uzoru na skulpturu. Novi model niske rezolucije treba dakle imati znatno manji broj poligona. Dok se skulptura sastojala od 670 000 trokuta, model optimiziran za video igru ne bi trebao sadržavati više od 15 000 trokuta. Za takvo drastično smanjenje broja poligona koristi se proces koji se naziva retopologija, a riječ je o ručnom kreiranju nove geometrije iznad skulpture. S obzirom da za neke dijelove modela nije bila potrebna detaljna verzija s puno poligona, ti su dijelovi ostavljeni za izravno modeliranje s malim brojem poligona. Riječ je o očima, trepavicama, obrvama, i kosi. [30]

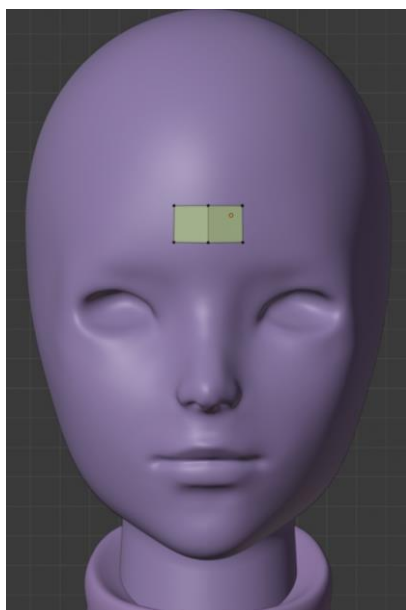
Idealni model za video igru trebao bi imati geometriju koja se sastoji od što manjeg broja poligona koji i dalje oblikuju taj model na prepoznatljiv i čitak način, ne bi trebao imati duple poligone ili poligone unutar geometrije koji se ne vide i trebao bi se sastojati isključivo od četverokuta i trokuta. Također, geometrija treba organski slijediti oblik tijela zbog animacija koje će kasnije biti primijenjene na modelu.

4.3.1. Retopologija

Proces retopologije započinje se pripremom jednog poligona na površini skulpture. Naredbom *Add > Mesh > Plane* u scenu se dodaje četverokut. On se zatim skalira, rotira i premjesti otprilike na poziciju na kojoj se započinje retopologija – npr. na čelo ako se radi o retopologiji lica. Poligon se smješta blago desno od centra lica s obzirom da će se nova topologija zrcaliti po x osi. Veličina tog poligona otprilike će odrediti veličinu svih ostalih poligona od kojih će se nova geometrija sastojati, jer je poželjno da poligone gradimo u sličnoj veličini.

Poligonu je zatim potrebno dodati dva modifikatora. *Mirror* modifikator dodaje se na isti način na koji je to bilo napravljeno kod skulpturiranja. Zatim se dodaje *Shrinkwrap* modifikator koji lijepi poligon na odabrani *Target* objekt. Kao *Target* objekt odabire se dio skulpture koji se retopologira. Na taj način nova geometrija savršeno prijanja uz geometriju skulpture, što uvelike smanjuje i olakšava posao retopologije. U opcijama modifikatora potrebno je uključiti opciju *On Cage* kako bi prikaz verteksa u *Edit Mode* načinu rada odgovarao položaju verteksa koji odlučuje modifikator.

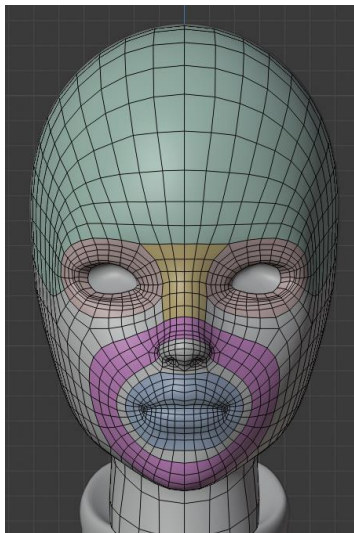
Još nekoliko opcija dodatno olakšava proces retopologije. Odabirom na opciju *Object Properties > Viewport Display > In Front* nova će se geometrija prikazivati ispred geometrije skulpture, iako su zapravo u istoj ravnini. Na taj način se jasnije vidi nova geometrija. Osim toga, dodatno možemo poboljšati vidljivost odabirom opcije *Random* za boju materijala u *Viewport Shading* opcijama i uključivanjem opcije *Backface Culling* kako bi se smanjio broj informacija koje se prikazuju. Za dodatnu preciznost može se uključiti opcija *Snap to Face* u zaglavlju scene kako bi se dodatno osiguralo lijepljenje poligona na skulpturu.



Slika 4.19. Početak retopologije lica

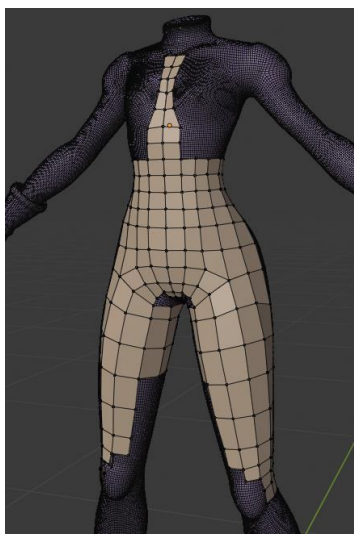
U *Edit Mode* načinu rada se zatim odabire opcija *Edge Select* i selektiraju se rubovi iz kojih se potom alatom *Extrude Region* istiskuju nova lica. S obzirom da ovom liku neće biti potrebno animirati lice, pažljivo postavljanje petlji oko očiju, nosa i usta nije nužno s obzirom da se ta geometrija neće deformirati. Ipak, na taj se način poligoni vrlo organski savijaju oko lica i stvaraju geometriju na kojoj neće biti nepravilnosti kod osvjetljavanja. Ovaj je način retopologije lica pogodan za stvaranje dobre UV mape. Kako bi se dobila takva geometrija lica, potrebno je prvo postaviti najveće petlje oko očiju, nosa i usta i bitne linije koje određuju topologiju lica, kao što su petlja za centar lica, petlja za rub čeljusti i sl. Nakon što se postave glavne petlje, praznine se brzo mogu popuniti korištenjem *F2* dodatka za Blender. [31] Odabirom jednog vršnog verteksa dobije se novi poligon stvaranjem novog verteksa. Položaj novog verteksa izračunava se na temelju položaja tri postojeća verteksa.

Osim toga, novi poligoni mogu se stvarati upotrebom već spomenutog *Extrude Region* alata, a dva se susjedna poligona mogu spojiti odabirom verteksa koje želimo spojiti i korištenjem naredbe *Vertex > Merge Vertices > At Center*.



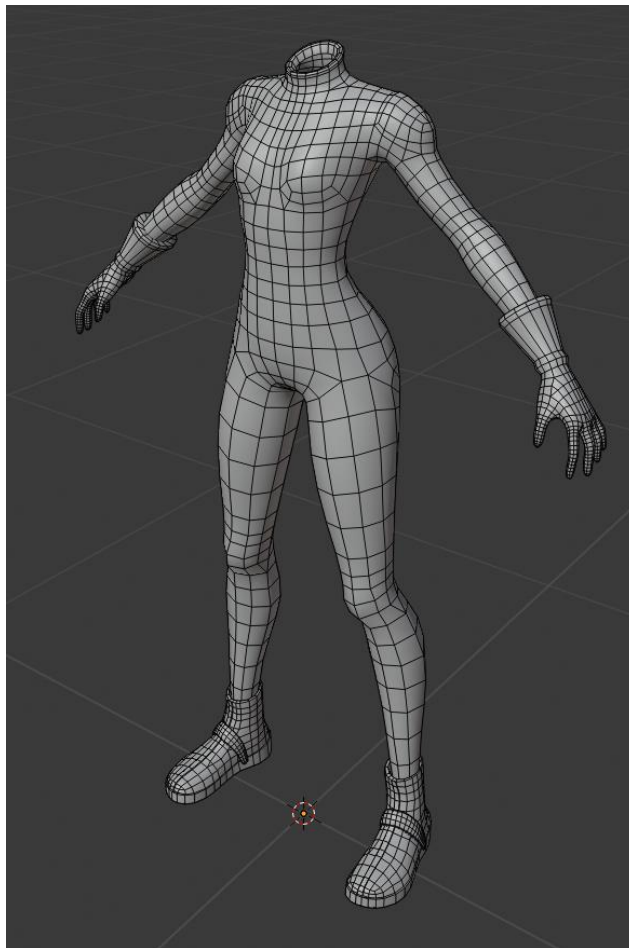
Slika 4.20. Retopologija lica

Ovakav proces retopologije potrebno je ponoviti za tijelo, rukavice i čizme. Izrađuje se poseban objekt za svaki od njih, i za svaki se ponavlja postupak dodavanja potrebnih modifikatora i razmišljanja o najpovoljnijem postavljanju petlji. Kod tijela je potrebno gušće rasporediti petlje na mjesta na kojima će dolaziti do savijanja, odnosno na zglobove. Također, potrebno je pripaziti na to da se koristi što manje trokuta, jer trokuti mogu izazvati razne deformacije uslijed savijanja.



Slika 4.21. Retopologija tijela u procesu

Nakon što je retopologija dovršena za svaki od dijelova skulpture, primijenjeni su modifikatori što znači da je geometrija trajno promijenjena i ne ovisi više o skulpturama. Zatim je za svaki od objekata uključena opcija *Object > Shade Smooth* kako bi sjenčanje objekta bilo glatko.



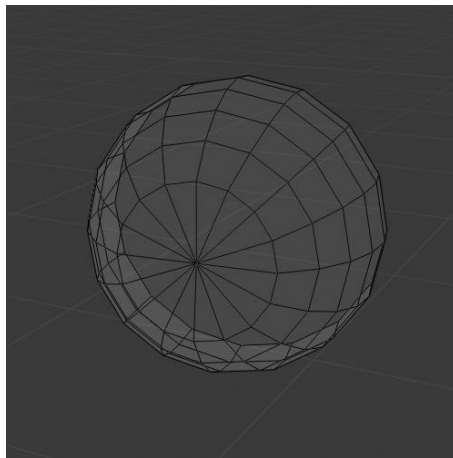
Slika 4.22. Gotova retopologija tijela

4.3.2. Modeliranje

Kao što je već navedeno, za neke dijelove lika kao što su oči, trepavice, obrve i kosa nisu bile potrebne skulpture, odnosno verzije s visokom rezolucijom poligona. Ti su dijelovi odmah modelirani s malim brojem poligona.

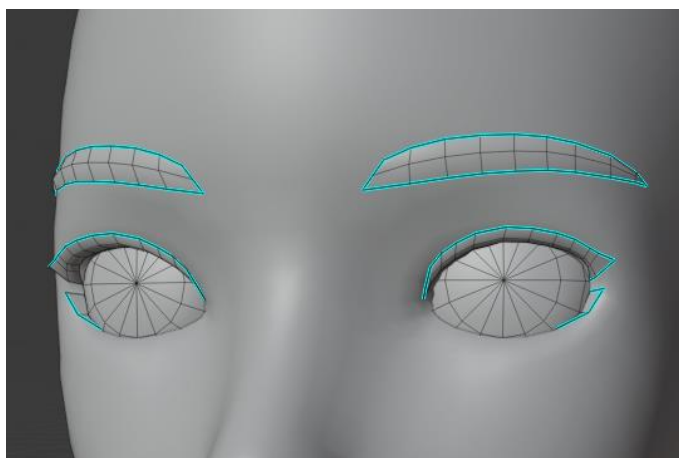
Model očiju napravljen je dodavanjem kugle u scenu naredbom *Add > Mesh > UV Sphere*. Potom je izbrisan donji dio kugle i obrisana je svaka druga rubna petlja naredbom *Mesh > Delete > Dissolve Edges* kako bi se prorijedio broj poligona. Kako bi se dobio ravan vrh kugle selektiran je vršni verteks kugle zajedno sa verteksima koji sačinjavaju najbližu rubnu petlju i upotrebom

tipkovnih prečaca $S + Z + O$ poravnati su po z osi. Primijenjen je *Mirror* modifikator, uključeno je glatko sjenčanje za objekt i model je smješten u očnu šupljinu.



Slika 4.23. Model oka

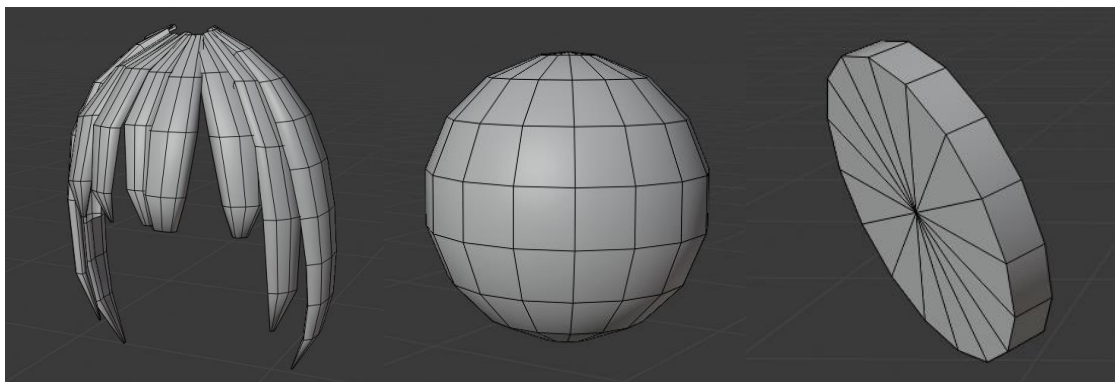
Trepavice su izrađene u *Edit Mode* načinu objekta za lice odabirom dijela rubne petlje očiju i korištenjem alata *Extrude Region*. Rubni dijelovi trepavica selektirani su i odabrana je opcija *Edge > Mark Sharp* kako bi ti dijelovi u sjenčanju objekta djelovali oštro. Takvi rubovi poprime plavu boju u *Edit Mode* načinu rada. Model za obrve izrađen je dodavanjem ravnine u scenu i brisanjem svih osim jednog verteksa. Preostali verteks postavljen je na mjesto na kojem se treba nalaziti rub obrve. *Extrude Region* alatom dodani su novi verteksi kojima je iscrtan oblik obrve, a prvi i posljednji verteks su spojeni naredbom *Vertex > Merge Vertices > At Last*. Selekcijom četiri verteksa i tipkovnim prečacem *F* stvori se lice. Na taj način se popuni model obrve.



Slika 4.24. Modeli očiju, trepavica i obrvi

Model za kosu izgrađen je pomoću krivulja na sličan način kao i model za pojas na čizmama. Dodana je Bezierova krivulja i kružnica koja je blago formirana u kvadratni oblik. Kružnica je korištena kao *Bevel Object*. Bezierove krivulje su duplicirane i smještene na način da tvore oblik šiški, a svakoj je zasebno u *Edit Mode* načinu interakcije mijenjana duljina i oblik. Nakon što su sve krivulje postavljene na željeno mjesto, svakoj je krivulji primjenjena opcija *Object > Convert to > Mesh from Curve/Meta/Surf/Text* koja krivulju pretvara u geometriju. Prije odabira na tu opciju potrebno je smanjiti rezoluciju krivulje postavljanjem *Resolution Preview U* opcije u *Object Data Properties* prozoru s 12 na 2.

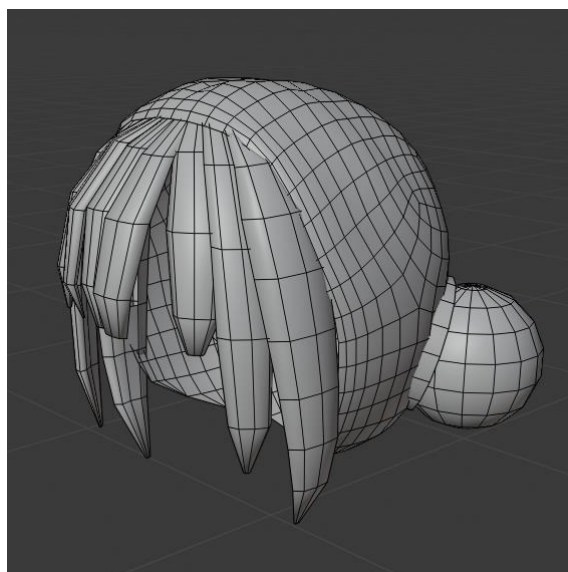
Model za punđe izrađen je dodavanjem kugle u scenu i brisanjem svake druge rubne petlje kako bi se razrijedio broj poligona. Dodan je *Mirror* modifikator. Model za gumice za kosu izrađen je dodavanjem valjka u scenu koji je skaliran da odgovara željenom obliku. Također je obrisano svaki drugi rub na plaštu valjka kako bi se smanjila razlučivost modela. Baza valjka ima više od četiri verteksa. To lice je obrisano, selektirani su svi verteksi koji su ga okruživali, i alatom *Inset Faces* i *Scale* stvoreni su četverokuti koji se gotovo spajaju u sredini baze valjka. Selektirani su svi verteksi koji se gotovo spajaju i iskorištena je opcija *Vertex > Merge Vertices > At Center*.



Slika 4.25. Modeli šiški, punđe i gumice za kosu

Model za ostatak kose izrađen je na nešto drugačiji način. U scenu je dodana kugla koja je potom u *Sculpt Mode* načinu interakcije skulpturirana u oblik kose *Grab* i *Draw* kistovima. Potom je objekt selektiran u *Object Mode* načinu rada i klikom na tipkovni prečac *N* otvoren je izbornik u kojem je moguće postaviti *Quad Remesh* opcije. *Quad Count*, odnosno broj četverokuta, postavljen je na 600, uključena je simetrija po x osi, i odabrana je opcija *REMESH IT*. Tako je na brz i jednostavan način dobivena optimizirana geometrija za model kose. Automatsko generiranje geometrije vrlo je korisno onda kada se taj objekt neće deformirati u animacijama, jer tada nije

bitno da su rubne petlje postavljene tako da odgovaraju deformacijama. Nasumičnost automatski generirane geometrije u tom je slučaju sasvim zadovoljavajuća.



Slika 4.26. Model kose

4.3.3. Čišćenje modela

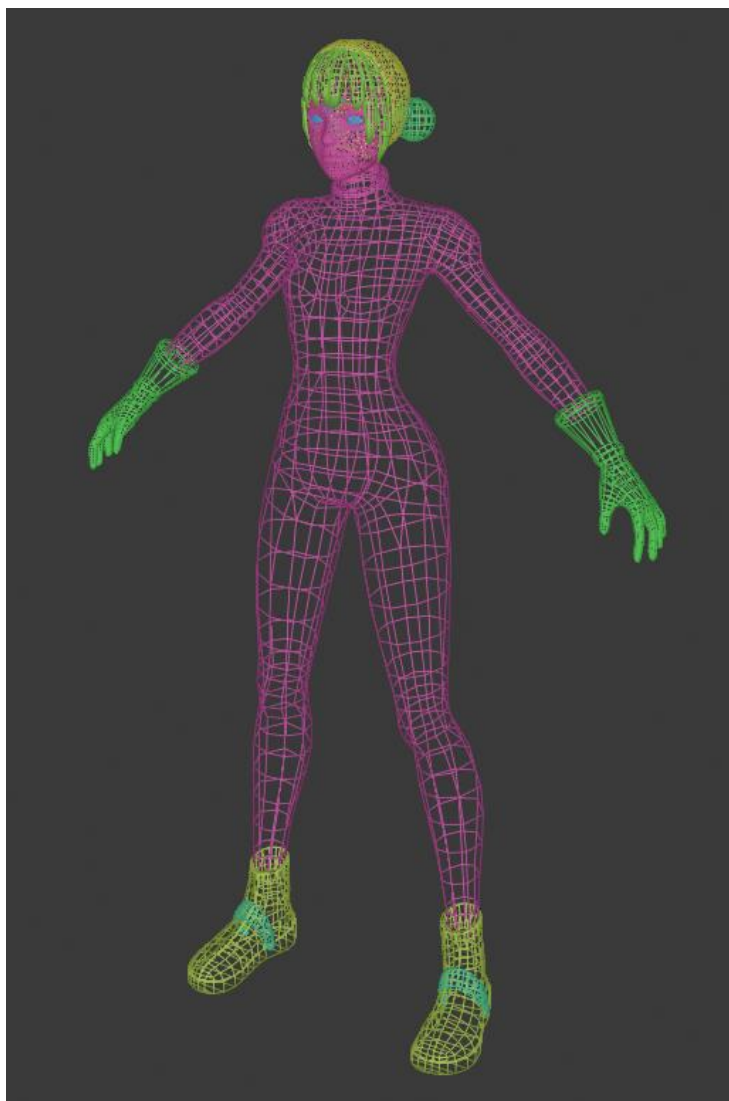
Po završetku s retopologijom i modeliranjem svih potrebnih objekata koji tvore lika, dobivena je dosta dobro optimizirana geometrija pogodna za kasnije animiranje. No, geometrija se dodatno može optimizirati brisanjem nepotrebnih poligona koji se nalaze unutar geometrije i time se zapravo nikada ne vide.

Na mjestu na kojem se podlaktica spaja s modelom rukavice, obrisani su višak poligona koji i otkriva rukavicu. Isto je tako na mjestu gdje se list spaja s modelom čizme obrisana nekolicina nepotrebnih poligona. Poligoni koji sačinjavaju model glave, a nalaze se ispod kose, nepotrebni su i zato su izbrisani. Također, uklonjeni su i poligoni koji tvore model vrata, a ulaze u model tijela.

Sljedeći korak u čišćenju geometrije je selektiranje svih objekata i odlazak u *Edit Mode* način rada. Zatim se selektiraju svi verteksi i klikom na *Vertex > Merge Vertices > By Distance* brišu se dupli verteksi koji se nalaze na identičnoj lokaciji. Potom su za svaki objekt primijenjeni svi korišteni modifikatori čime je trajno promijenjena geometrija.

Objekte nije potrebno međusobno spajati jer je u sljedećem koraku izrade lika potrebno izraditi UV mape za svaki od objekata, što je mnogo preglednije i lakše raditi ako su objekti odvojeni. Također, korisno je znati da model lika ne mora biti jedinstvena tzv. „vodonepropusna geometrija“, odnosno geometrija sačinjena od poligona koji su svi međusobno spojeni u jednu cjelinu. „Vodonepropusna geometrija“ odnosi se na geometriju koja u imaginarnom slučaju u kojem je ispunjena vodom tu vodu ne bi propuštala. U slučaju lika za video igru, nije potrebno spajati vršne vertekse podlaktice s vršnim verteksima modela rukavice, jer pri tom procesu broj lica na modelu ostaje isti i performanse u video igri time neće biti poboljšane. [32]

Geometrija je gotova onda kada ima urednu topologiju pogodnu za animiranje sastavljenu od niskog broja poligona i koja je očišćena do razine u kojoj nema nepotrebnih nevidljivih ili duplih poligona, rubova ili verteksa.



Slika 4.27. Gotova optimizacija modela

4.4. Teksture

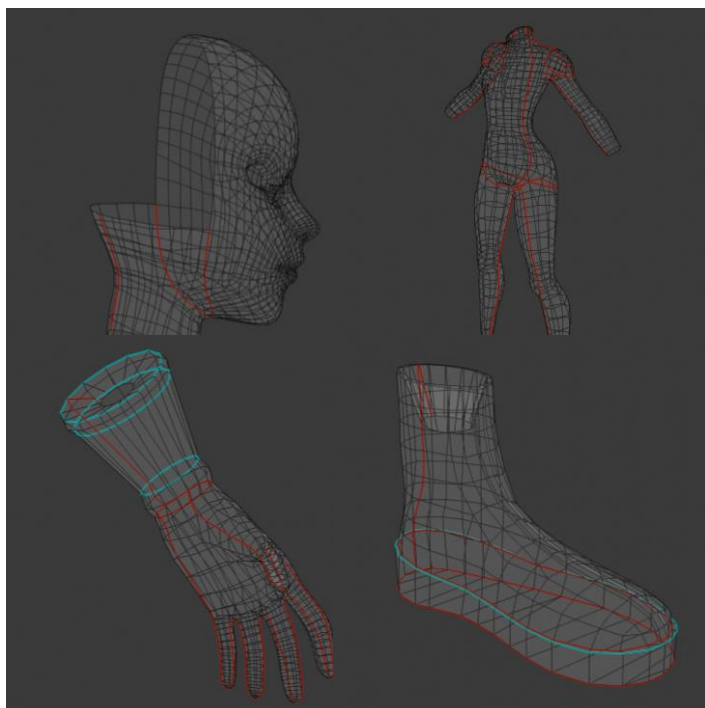
Kako bi model lika dodatno zaživio, na njega je potrebno dodati materijale s teksturama. Taj se proces sastoji od izrade UV mape, difuzne mape, mape emisije i mape normala, koje se zatim dodaju na materijale koje lik sadrži. [11]

4.4.1. UV mapa

Nakon što je geometrija gotova, potrebno ju je „odmotati“ na UV mapu. Taj se proces započinje postavljanjem šavova na model. Postavljanjem šavova sami definiramo na kojim će se mjestima geometrija rašiti kako bi se odmotala na 2D prostor sa što manje deformacija. Šavovi se obično postavljaju na mjesta na kojima se najmanje vide, s obzirom da na tim mjestima može doći do primjetnih prijelaza u teksturi. Zbog toga je poželjno postavljati što manji broj šavova, a da se geometrija i dalje lijepo odmota. [13]

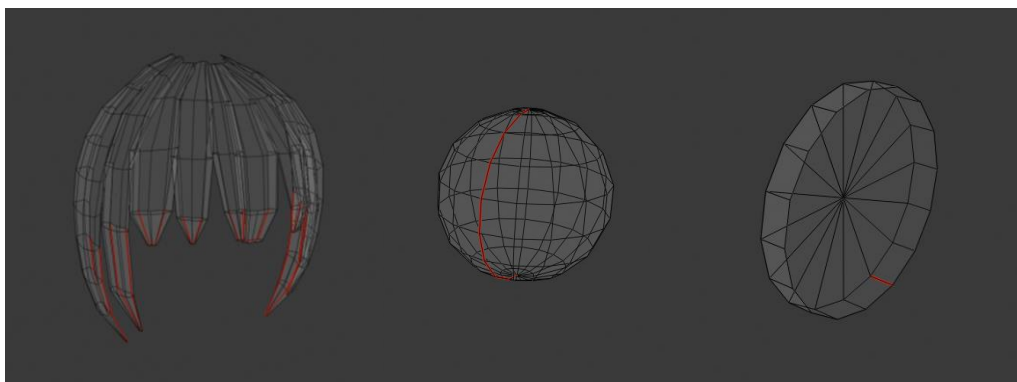
Šav se postavi odabirom rubova u *Edit Mode* načinu rada dok je odabran *Edge Select* način odabira i klikom na opciju *Edge > Mark Seam*. Odabran rub zatim poprima crvenu boju i označava mjesto na kojem će se geometrija rašiti. Šavovi se mogu označiti samo na lijevoj polovici modela a zatim se opcijom *Mesh > Symmetrize* preslikaju na desnu polovicu.

Jedan šav na glavi postavljen je tako da odvoji lice od vrata, a drugi šav postavljen je na stražnjoj strani vrata. Na ruci je šav smješten tako da odvoji gornju od donje polovice dlana. Drugi šav odvaja gornji dio rukavice od dlana, a treći je položen uzduž gornjeg dijela rukavice. Na čizmi su šavovima odvojeni gornji i donji dio đona i postavljen je treći šav na stražnjem dijelu čizme. Na svakom udu je smješten dugi šav na najmanje primjetnom dijelu i napravljeni su šavovi koji odvajaju trup od udova. Stavljene su šavovi po sredini leđa, uz šavove za štitnike za ramena.



Slika 4.28. Šavovi za tijelo i glavu

Šavovi na šiškama postavljeni su po unutarnjem rubu šiški kako se ne bi primjetili. Isto tako, šav na pundi postavljen je sa strane kugle koja je okrenuta prema glavi. Objektu za gumicu za kosu dovoljan je jedan mali šav na plaštu valjka. Za sve ostale modele nisu potrebni šavovi jer se i bez njih mogu odmotati na UV mapu bez većih deformacija.

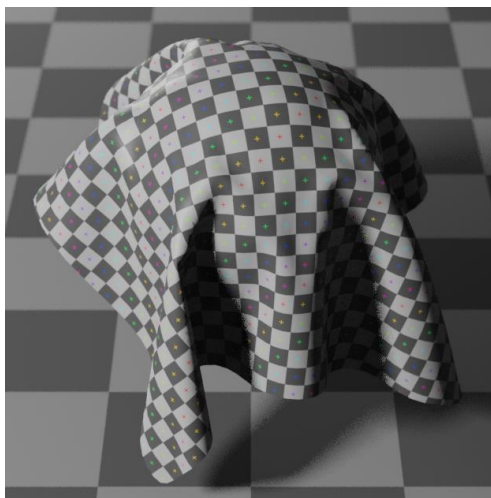


Slika 4.29. Šavovi za kosu

Za model lika potrebne su dvije UV mape: jedna za kosu i jedna za sve ostalo. UV mapa za kosu treba biti rezolucije 1024 x 1024 px, a UV mapa za sve ostalo 1024 x 2048 px. Kako bi se odredila veličina UV mape, izrađuju se privremene slike željenih rezolucija na koje se primjenjuje tekstura s uzorkom šahovnice. Tekstura s uzorkom šahovnice omogućuje brzu provjeru ispravnosti

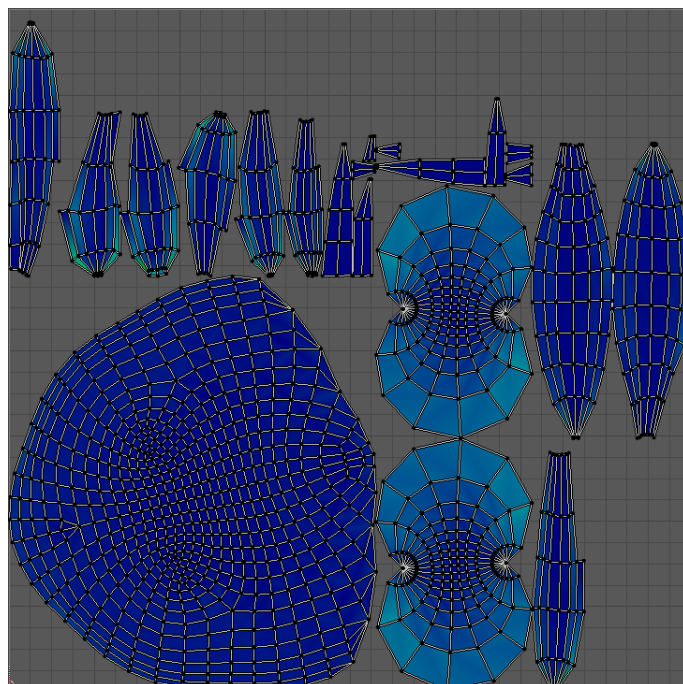
UV mapa i provjeru razlučivosti teksture na dijelovima modela. Slike se izrađuju klikom na opciju *Image > New* u izborniku *UV Editor* prozora. Slika za kosu nazvana je *kosa_sahovnica*, dok je slika za sve ostalo nazvana *lik_sahovnica*. Odabrana je rezolucija, nakon čega je za opciju *Generated Type* odabran *UV Grid*, odnosno na slici je generirana tekstura šahovnice.

Slike je potom potrebno primijeniti na objekte kreiranjem materijala za kosu i materijala za sve ostalo. Selekcijom na model šiški i odlaskom u *Material Properties* prozor, moguće je klikom na *New* dodati novi materijal za taj objekt. Materijal se zatim preimenuje u *kosa_materijal*. Pod dodatnim opcijama za *Base Color*, odnosno baznu boju, odabire se opcija *Image Texture*, tj. slikovna tekstura. Klikom na gumb za pretraživanje postojećih slika otvara se izbornik koji sadrži slike koje su već kreirane i odabire se slika *kosa_sahovnica*. Zatim se selektiraju svi objekti koji tvore kosu pazeći da je objekt šiški selektiran posljednji, nakon čega se odabirom na opciju *Object > Make Links > Materials* materijal *kosa_materijal* vrlo brzo primijeni na sve selektirane objekte. Za model tijela kreiran je materijal nazvan *lik_materijal* koji na sebi ima *lik_sahovnica* slikovnu teksturu. Zatim je istim procesom primijenjen na sve ostale objekte koji tvore lika.



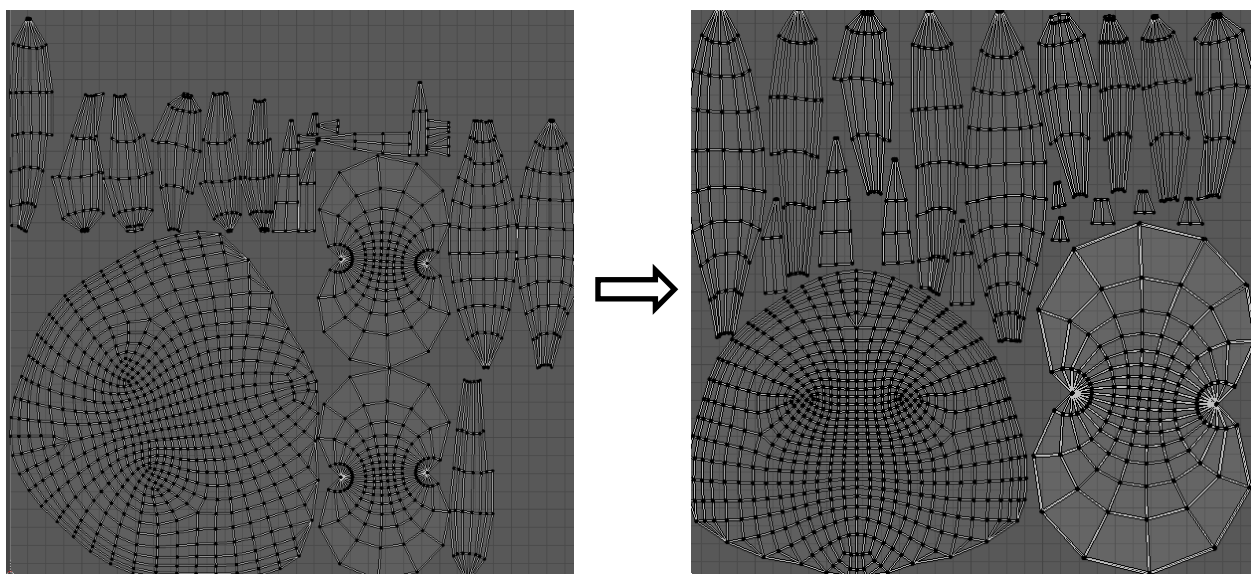
Slika 4.30. Pregled materijala

Nakon što su materijali tako postavljeni, moguće je krenuti s UV odmotavanjem. Prvo je kreirana UV mapa za objekte koji tvore kosu. Selektirani su modeli šiški, punđi i ostatka kose, nakon čega su u *Edit Mode* načinu rada odabrana lica koja ti objekti sadrže. Klikom na opciju *UV > Unwrap* stvara se UV mapa povrh teksture *kosa_sahovnica*. Valjanost odmotavanja može se provjeriti uključivanjem opcije *Stretching* u *View > Display > Overlays*. Uključivanjem te opcije na UV mapu se prikazuje količina rastezanja teksture. Crvena boja ukazuje na veliko rastezanje i nju je potrebno izbjegavati. Plava i blago zelena boja ukazuju na to da je odmotavanje uredno.



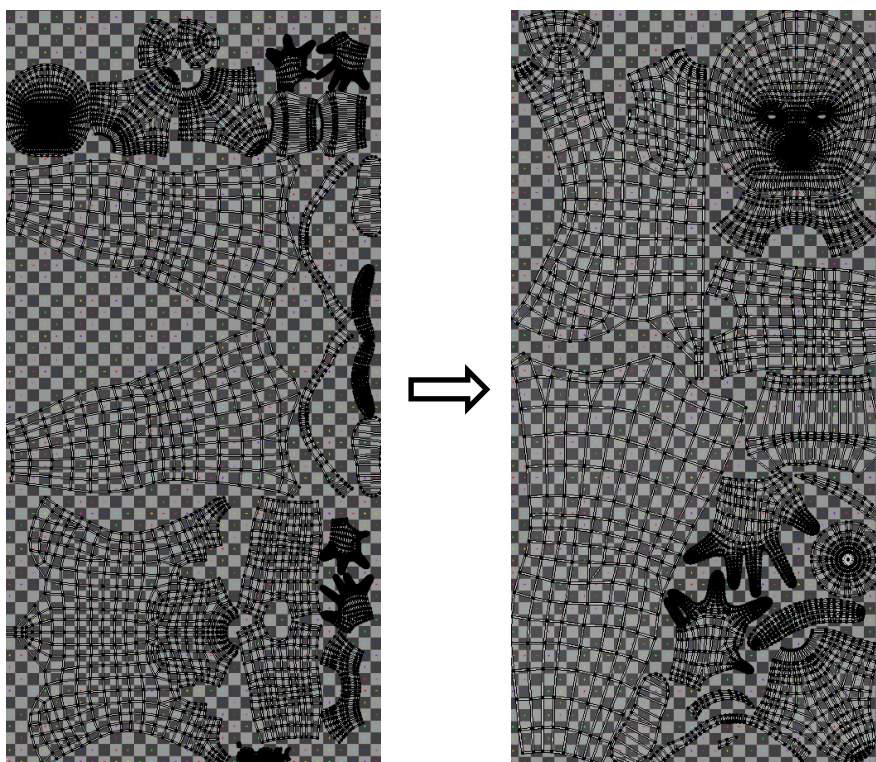
Slika 4.31. UV mapa s prikazom rastezanja

UV otoci su automatski raspoređeni tako da odgovaraju definiranoj veličini teksture koja je primijenjena na materijal. Ipak, prostor UV mape može se bolje iskoristiti ako se UV otoci ručno poslože. Kod namještanja UV otoka nastoji se pokriti što više prostora, kako bi svaki pojedini objekt obuhvaćao što veći dio teksture, što rezultira višom kvalitetom. Također, UV verteksi ne smiju prelaziti granice kreirane privremene teksture. Kako bi se prostor što bolje iskoristio, UV otoci za punđe su postavljeni jedan iznad drugoga, s obzirom da će imati identičnu teksturu.



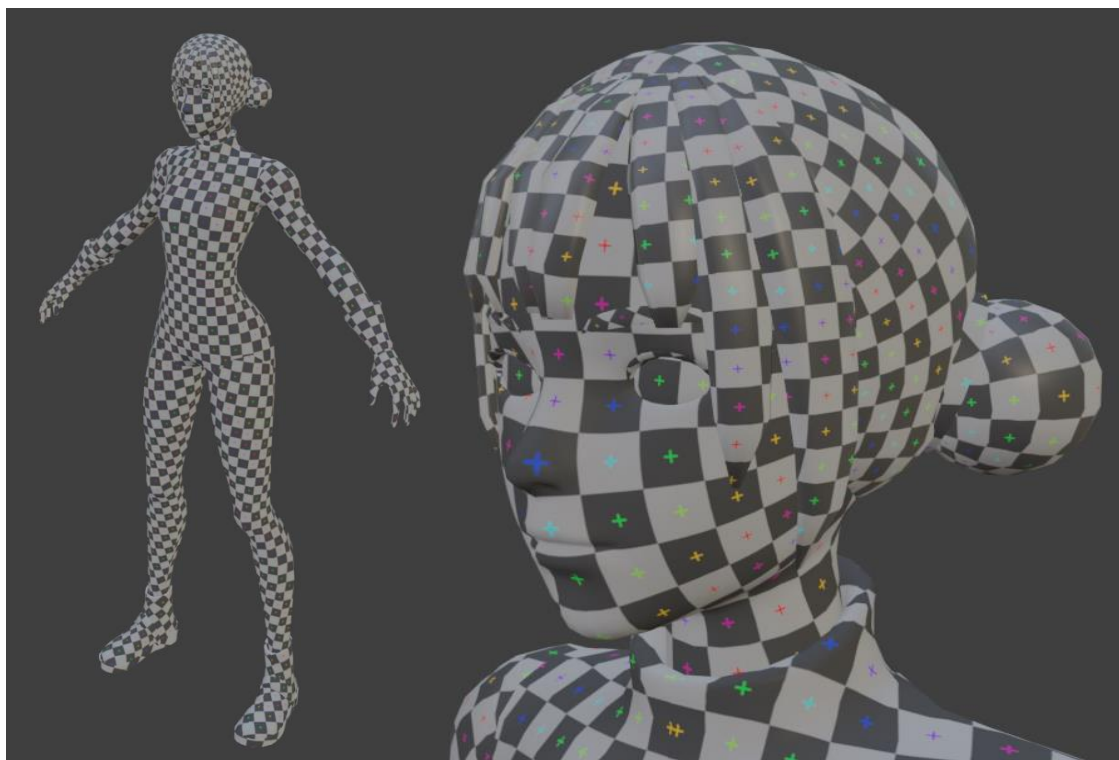
Slika 4.32. Ručno slaganje UV mape za kosu

Isti se proces ponavlja za sve ostale objekte koji se odmotavaju na mapi rezolucije 1024 x 2048 px. Kako bi se iskoristio što veći prostor i s obzirom na činjenicu da je model savršeno simetričan, na UV mapi su posloženi samo UV otoci za lijevi dio tijela. Zatim su UV otoci za desni dio tijela zrcaljeni opcijom *UV > Mirror > X Axis* i stavljeni točno iznad UV otoka za lijevi dio tijela. Na taj način lijeva i desna polovica tijela prikazuju isti dio teksture, a rezolucija teksture za svaki pojedini objekt gotovo je dvostruko veća. Kako bi se UV otok za trup podijelio na lijevu i desnu polovicu, odabrana je selekcija lica i selektirana su samo lica na desnoj strani trupa. Klikom na tipkovni prečac *Y* taj se dio otoka odvaja i može se zrcaliti.



Slika 4.33. Ručno slaganje UV mape za tijelo

Tekstura s uzorkom šahovnice u ovom je procesu jako korisna jer omogućava pregled veličine teksture na pojedinim objektima. U *Viewport Shading* opciji u zaglavlju scene nalaze se opcije za način sjenčanja modela u sceni. Odabirom na opciju *Material Preview* na objektu postaje vidljiv materijal koji je kreiran zajedno s teksturom koja se nalazi na tom materijalu. Na taj je način moguće pri namještanju UV otoka vidjeti da li je šahovnica na svim objektima podjednake veličine. Izrada UV mapa gotova je onda kada svaki objekt ima uredno odmotanu geometriju posloženu na UV prostor željene rezolucije, na način da se ne preklapa s objektima koji će se nalaziti na istoj teksturi. Zahvaljujući teksturi s uzorkom šahovnice, odmah je vidljivo da je proces UV odmotavanja pravilno izrađen.

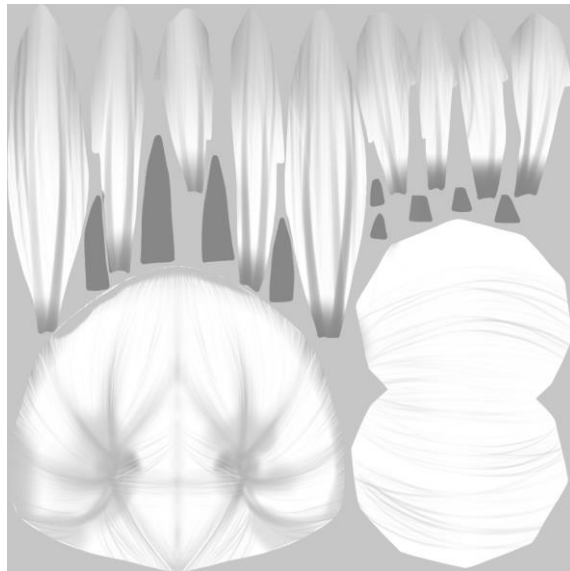


Slika 4.34. Model nakon izrade UV mapi

4.4.2. Difuzna mapa

Difuzna mapa najčešća je vrsta tekstura koja definira boju i uzorak objekta. Kako bi se kreirala difuzna mapa, prvo je potrebno izvesti UV mape u .png formatu kako bi se mogle koristiti u programu za slikanje. Za to se odabire opcija *UV > Export UV Layout* u *UV Editor* prozoru. Potrebno je odabrati mjesto spremanja slike, naziv slike i *Fill Opacity* odnosno neprozirnot ispune na slici postaviti na 0. Izvozi se raspored UV mape za selektirane objekte, pa je zato potrebno prvo izvesti UV mapu sa selektiranim objektima za kosu, a zatim UV mapu sa selektiranim objektima za sve ostalo. [33]

Za stvaranje difuzne mape korištena je aplikacija Procreate, no može se koristiti bilo koji alat za digitalnu obradu fotografija. Procreate je alat za digitalno slikanje koji je razvijen za iOS i iPadOS. [34] U aplikaciji su otvorene UV mape koje su potom obojane i teksturirane raznim kistovima. Jednom kada je teksturiranje završeno, slika se sprema u .jpg formatu. Spremljena je slika za kosu koja je nazvana *kosa_d* (slovo *d* stoji za difuziju) i slika za sve ostalo koja je nazvana *lik_d*.

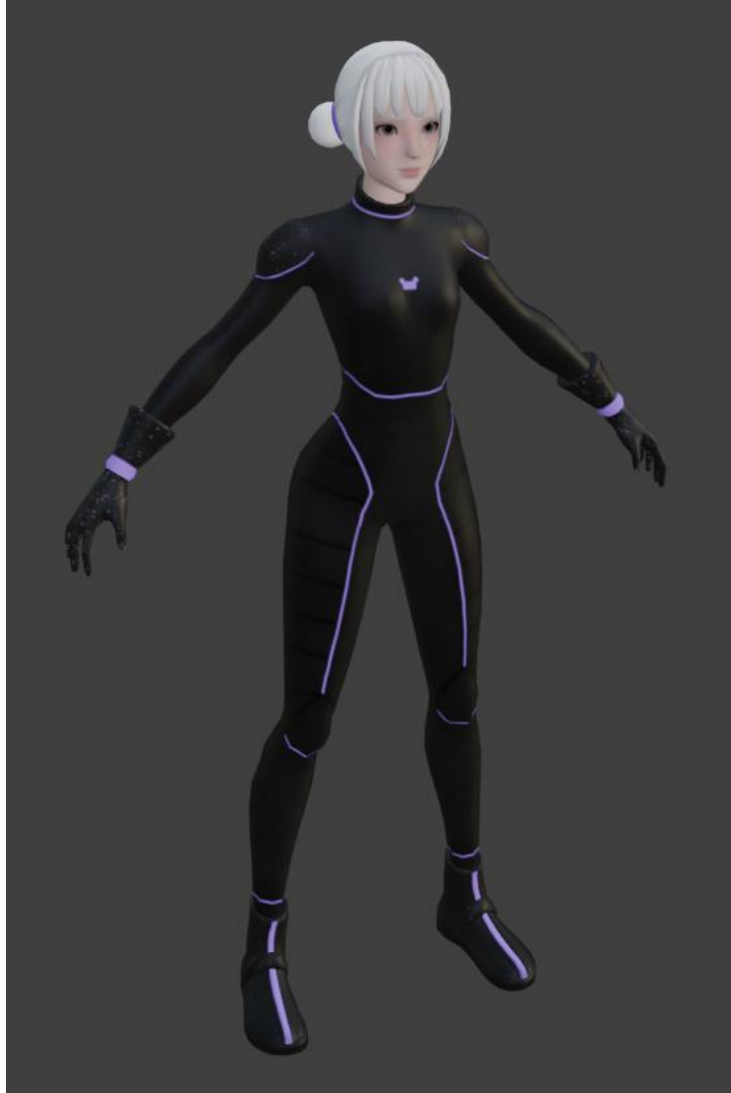


Slika 4.35. Difuzna mapa za kosu



Slika 4.36. Difuzna mapa za tijelo

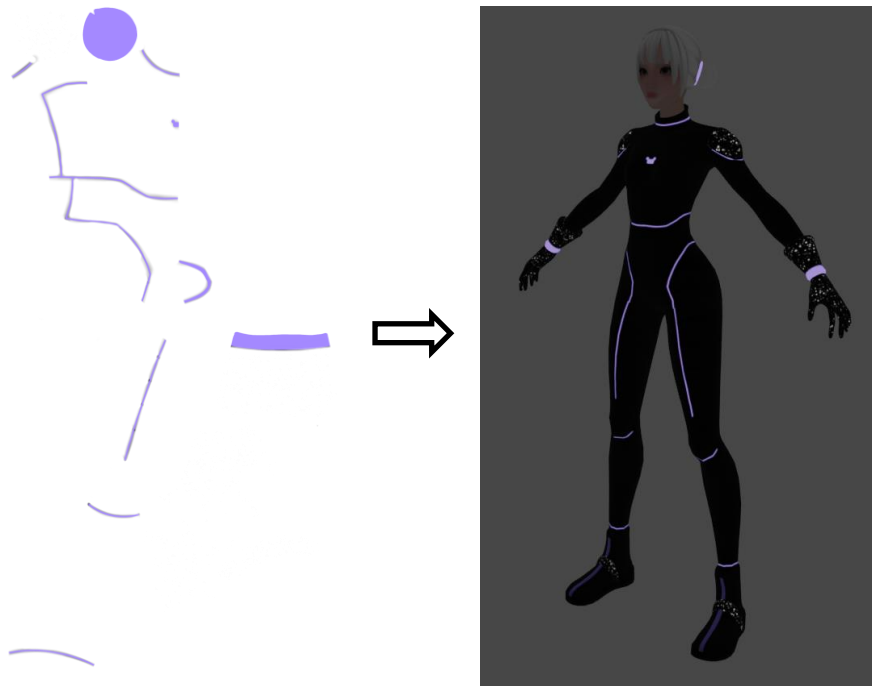
Slika *kosa_d* se postavi kao difuzna mapa za materijal *kosa_materijal* tako što se odabere kao *Image Texture* umjesto privremene slike *kosa_sahovnica* koja je ranije kreirana u *Material Properties* prozoru. Slika *lik_d* na isti se način primijeni na *lik_materijal*.



Slika 4.37. Model s difuznim mapama

4.4.3. Mapa emisije

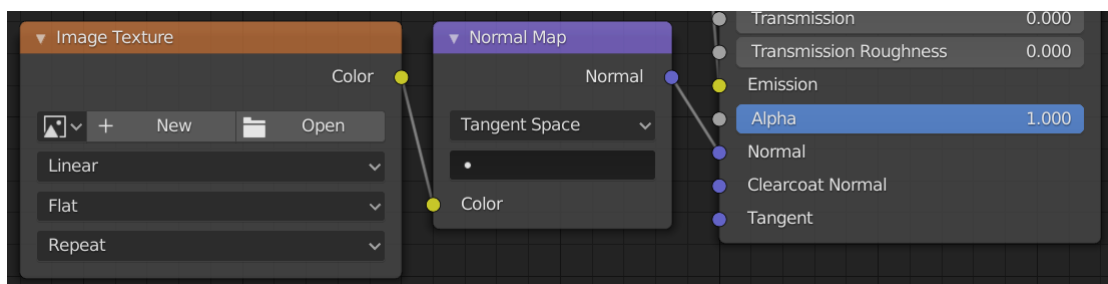
Mapa emisije određuje svjetlosni izvor na modelu pomoću teksture, a koristi se za stvaranje dojma svjetlećih ljubičastih traka na odijelu lika. U aplikaciji Procreate izdvojeni su ljubičasti dijelovi na difuznoj mapi lika i izvezeni iz aplikacije u .jpg formatu na prozirnoj pozadini. Slika je nazvana *lik_e* (slovo *e* stoji za emisiju) i postavljena je kao *Image Texture* odnosno slikovna tekstura u opciji za emisiju na materijalu *lik_materijal*. [35]



Slika 4.38. Mapa emisije i njen utjecaj na izgled modela u mraku

4.4.4. Mapa normala

Mapa normala izrađuje se samo za materijal za tijelo i glavu, odnosno za *lik_materijal*. Otvoren je *Shading* radni prostor specijaliziran za izradu materijala. Ovaj radni prostor sadrži *Shader Editor* prozor u kojem je moguć rad s čvorovima (*engl. Node*) koji uređuju materijal. Izrađuje se novi čvor za slikovnu teksturu naredbom *Add > Texture > Image Texture*. Zatim se dodaje čvor koji tu teksturu čita kao mapu normala naredbom *Add > Vector > Normal Map*. Izlaz za boju na *Image Texture* čvoru spaja se na ulaz za boju na *Normal Map* čvoru. Zatim se izlaz za normalu na *Normal Map* čvoru spaja na ulaz za normalu na čvoru *Principled BSDF* koji sadrži sve ulaze i postavke za *lik_materijal*. Na taj je način postavljena baza za dodavanje mape normala na lika, i potrebno je još samo kreirati samu mapu normala koja se potom ubacuje u *Image Texture* čvor.

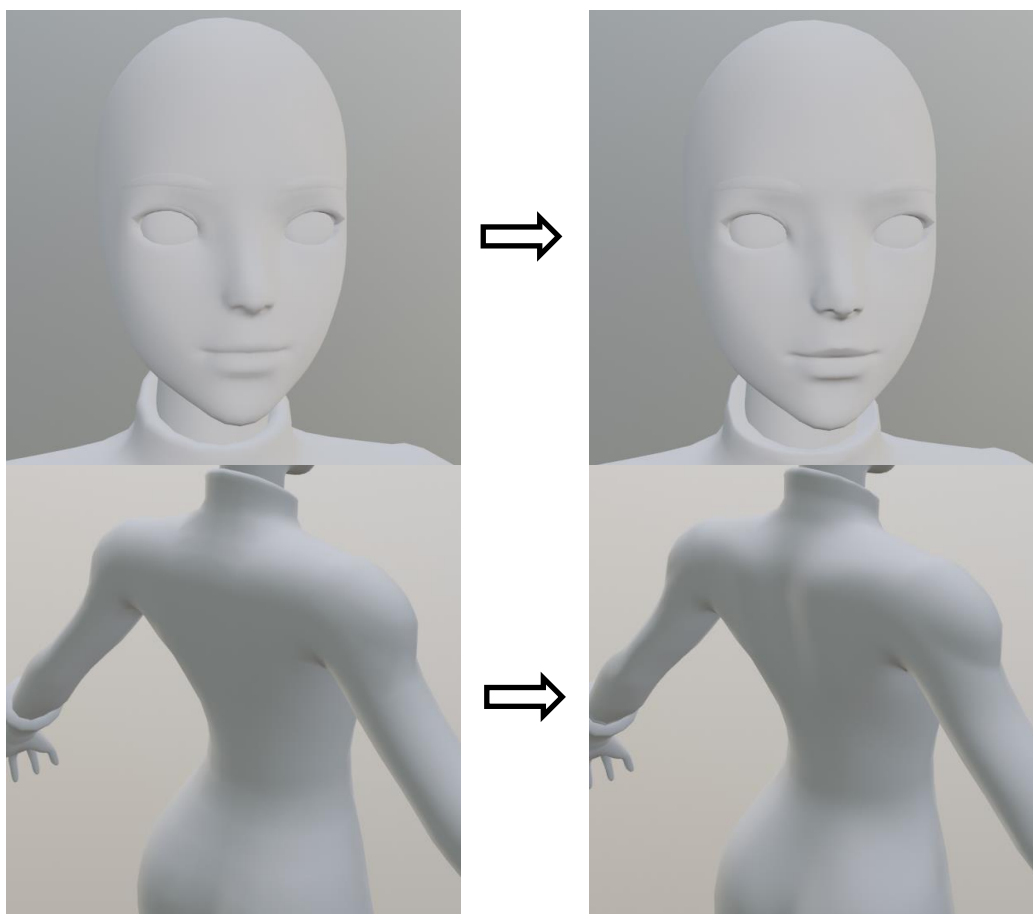


Slika 4.39. Postavljeni čvorovi za mapu normala

Prvi korak za stvaranje mape normala je selektiranje skulpture lika i potom selektiranje modela lika niže rezolucije u točno tom redoslijedu. Zatim se u *Shader Editor* prozoru selektira čvor *Image Texture* u kojem želimo pohraniti mapu normala. Odlaskom u *Render Properties* prozor koji sadrži postavke vezane uz renderiranje, odabire se opcija *Cycles* za *Render Engine*.

Cycles je jedan od nekoliko mehanizama za renderiranje koje Blender 2.8 sadrži. Dizajniran je za pružanje fizički utemeljenih rezultata s brojnim kontrolama i fleksibilnim čvorovima za izradu materijala. Samo je u tom načinu renderiranja moguće stvoriti mapu normala u alatu Blender 2.8.

Pod *Bake* odjeljkom u *Render Properties* prozoru, nalaze se potrebne postavke za kreiranje mape normala. Potrebno je označiti opciju *Selected to Active* koja će informacije sa selektirane skulpture prenijeti na aktivni, odnosno posljednje selektiran objekt, a to je model lika. *Ray Distance*, odnosno udaljenost zraka se postavlja na 0.02 m kako bi se uhvatili svi detalji na skulpturi. Klikom na gumb *Bake* kreira se mapa normala i postavlja u čvor *Image Texture*. Time na modelu mapa normala odmah postaje vidljiva jer su svi potrebni čvorovi povezani s čvorom za proizvodnju materijala.



Slika 4.40. Model prije i nakon dodavanja mape normala

4.4.5. Podešavanje materijala

Posljednji korak je podešavanje opcija za individualne materijale koji se nalaze na liku. Uz materijale *kosa_materijal* i *lik_materijal* koji su već navedeni, izrađen je i treći materijal koji je odabran samo za oči, a nazvan je *oci_materijal*.

Na materijalu *kosa_materijal* refleksija (*engl. Specular*) je postavljena na vrijednost 0.3 kako bi djelovala sjajnije, dok je na ostalim materijalima ta vrijednost postavljena na 0.05. Hrapavost (*engl. Roughness*) na materijalu za oči je postavljena na 0, dok je na ostalim materijalima postavljena na 0.5. Također, na materijalu za oči dodan je bezbojni lak (*engl. Clearcoat*) kako bi poprimio jaki sjaj, postavljanjem njegove vrijednosti na 1.



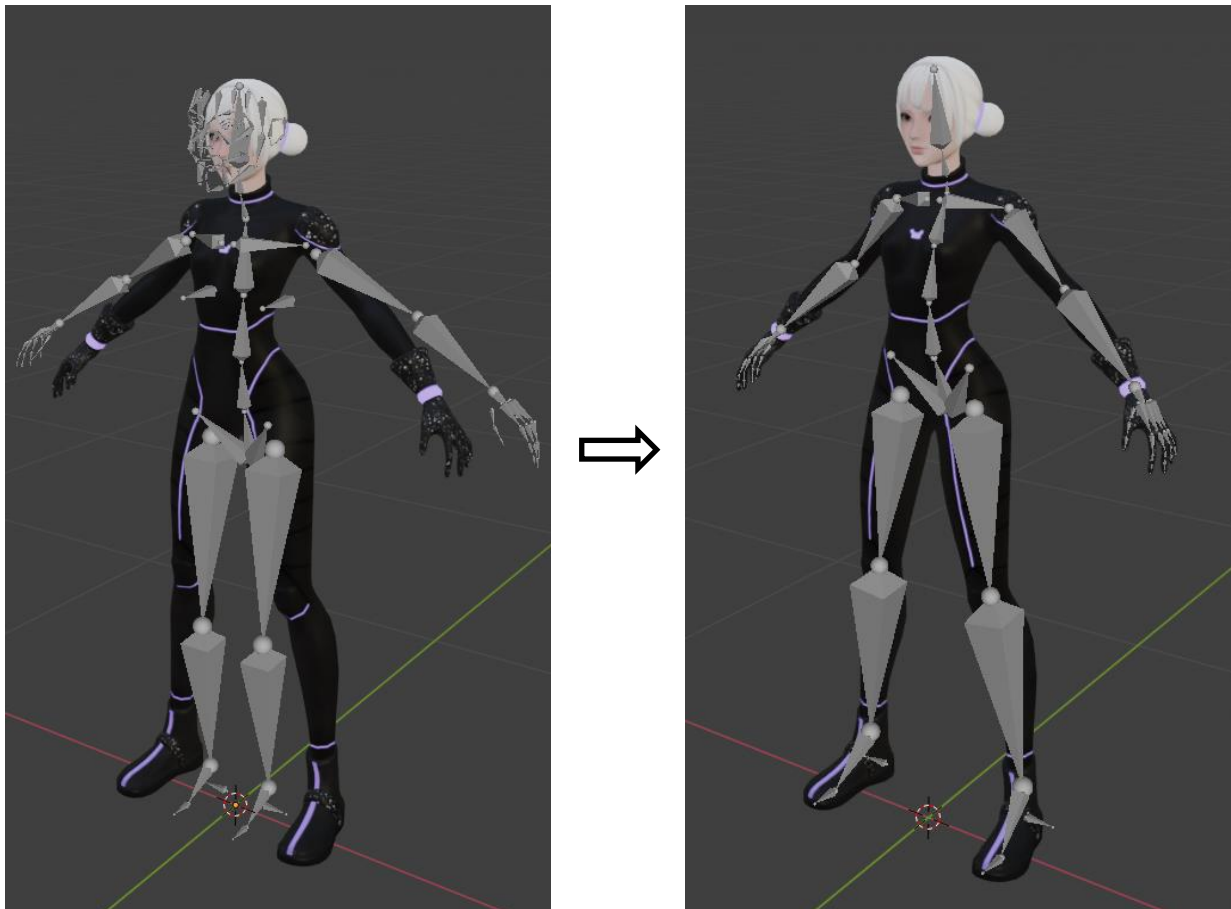
Slika 4.41. Model nakon podešavanja materijala

4.5. Kostur

Kostur u Blender-u može se smatrati sličnim pravom kosturu jer baš kao i pravi kostur, sastoji se od mnogih kostiju. Te se kosti mogu pomicati i sve što je povezano s njima kretat će se i deformirati na sličan način. Izrada kostura nužan je korak za animiranje lika. [36]

4.5.1. Izrada kostura

Besplatni dodatak za Blender 2.8 zvan *Rigify* sadrži već gotov ljudski kostur koji je kompatibilan s alatom Unity. To znači da alat Unity prepoznaje sve kosti u *Rigify* ljudskom kosturu i na njega vrlo brzo može primijeniti gotove animacije. Kostur se dodaje u scenu naredbom *Add > Armature > Human (Meta-Rig)*. U *Edit Mode* načinu rada mogu se izbrisati nepotrebne kosti, a sve ostale posložiti na način da savršeno prijanjaju modelu lika.

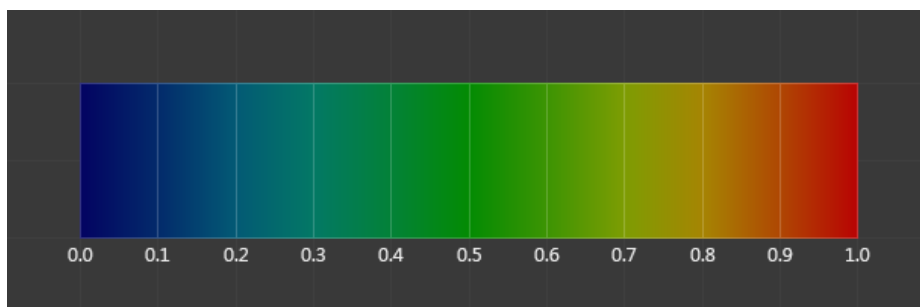


Slika 4.42. Rigify kostur prije i nakon prilagođavanja modelu

4.5.2. Težinsko bojanje

Kako bi kosti pomicala geometriju tijela, potrebno je kostur spojiti na tijelo. Za početak su odabrani svi objekti koji sastavljaju tijelo i naredbom *Object > Join* spojeni su u jedan. Zatim je selektiran dobiveni objekt tijela, a potom kostur i odabirom na naredbu *Object > Parent > With Automatic Weights* kostur je spojen na tijelo upotrebom automatskog težinskog bojanja.

Težinsko bojanje odnosi se na bojanje težina po verteksima modela lika s obzirom na željenu količinu deformiranja geometrije za pojedinu kost. Odlaskom u *Weight Paint* način interakcije, moguće je kistom postaviti željenu „težinu“ geometrije za selektiranu kost. Težina modela skalira se od 0 do 1, gdje vrijednost 0 određuje da se geometrija ne deformira pomicanjem selektirane kosti i prikazana je plavom bojom, dok vrijednost 1 određuje da se geometrija potpuno pomiče zajedno sa selektiranom kosti i prikazana je crvenom bojom. Vrijednosti između 0 i 1 teže ka zelenoj boji i određuju stupanj kojim se geometrija deformira pomicanjem kosti. [37]



Slika 4.43. Vrijednosti težinskog bojanja prikazane bojom [38]

S obzirom da je kostur s geometrijom povezan automatskim težinskim bojanjem, većina posla već je obavljena. Selektiranjem i rotiranjem svake pojedine kosti provjerava se ispravnost deformiranja geometrije. *Draw* kistom i podešavanjem željene težine i radijusa kista, boja se dio geometrije koji je potrebno ispraviti. Selektiranjem kostura i odlaskom u *Pose Mode* način interakcije može se provjeriti ispravnost modela izrađivanjem probnih poza. Ukoliko nema grešaka u pomicanju geometrije s kostima, model lika je završen.

4.6. Demonstracija u razvojnom okruženju Unity

Kako bi se ispitala ispravnost dovršenog lika za video igru, napravljena je vrlo jednostavna aplikacija u razvojnom okruženju Unity u kojoj je moguće pokretati lika i narediti pokretanje animacije mahanja rukom.

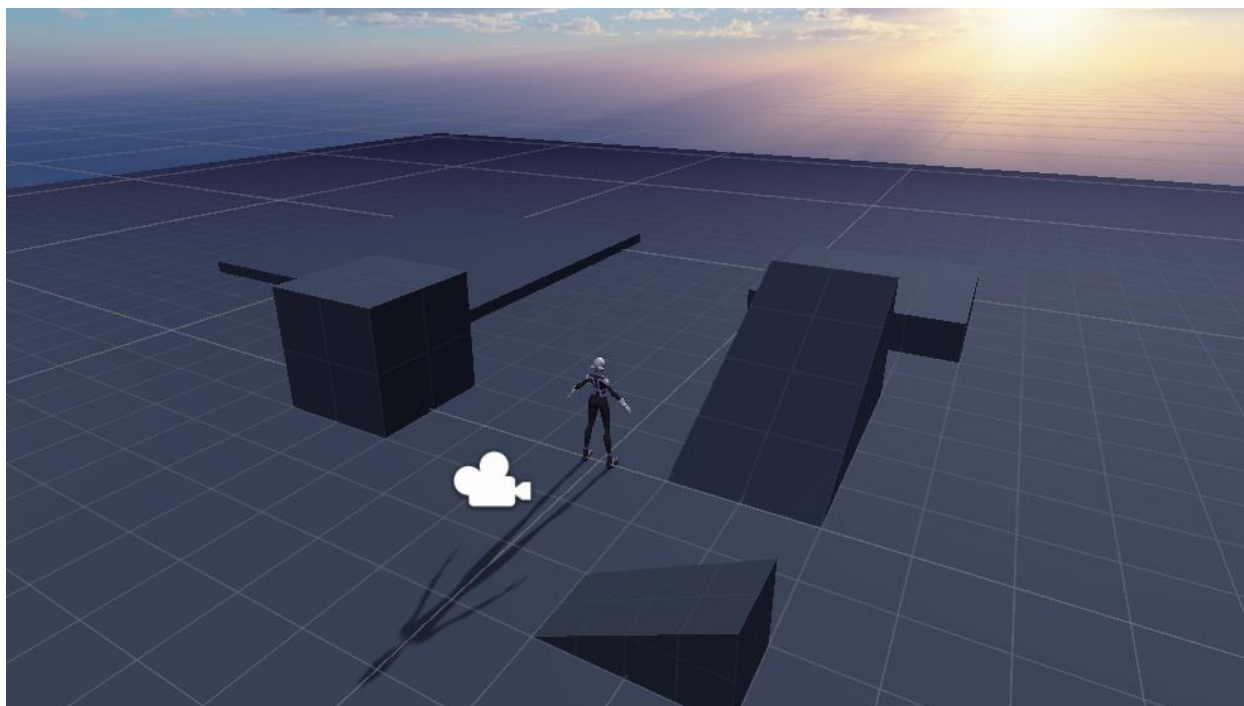
4.6.1. Postavljanje projekta

Otvaranjem alata Unity, prvo se otvara aplikacija Unity Hub u kojoj se kreira novi projekt klikom na gumb *New*. Odabire se predložak za 3D projekt i unese se ime i lokacija na koju će se spremi projekt. Kako bi projekt od početka bio dobro organiziran kreirani su folderi *Animacije*, *Lik*, *Scena* i *Skripte*. U folder *Lik* dodana je .blend datoteka u kojoj je izrađen model lika za video igru, zajedno s folderom koji sadrži sve potrebne teksture. Klikom na ikonu lika dostupne su razne postavke u prozoru *Inspector*. U kartici *Rig* je potrebno odabrati *Animation Type* > *Humanoid* i kliknuti gumb *Configure* kako bi Unity prepoznao sve kosti koje lik sadrži.

4.6.2. Scena

Pokretanjem novog projekta automatski je kreirana scena koja sadrži kameru i svjetlo. Dodano je još jedno svjetlo vrste *Point Light* koje je postavljeno tako da osvjetljava suprotnu stranu od već postojećeg svjetla. Boja novog svjetla promijenjena je u plavo-ljubičastu u *Inspector* prozoru pod opcijom *Color*.

Unity Asset Store sve je veća trgovina koja sadrži razne modele, animacije i tehnologije koje izrađuju ljudi diljem svijeta. Iz trgovine je uvezen paket *AllSky Free* koji sadrži razne *Skybox* materijale koji mijenjaju izgled neba u sceni. Odabirom na *Window* > *Rendering* > *Lighting Settings* moguće je odabrati novo nebo za *Skybox Material* opciju. Odabran je *Epic_BluSunset* materijal za nebo. S Unity Asset Store trgovine preuzet je paket *Standard Assets* koji sadrži mnoge objekte za prototipiranje. U scenu su dodani modeli za tlo, kao i kocke i platforme kako bi popunile scenu. Zatim se u sredinu scene dodaje model lika.



Slika 4.44. Izgled scene u alatu Unity

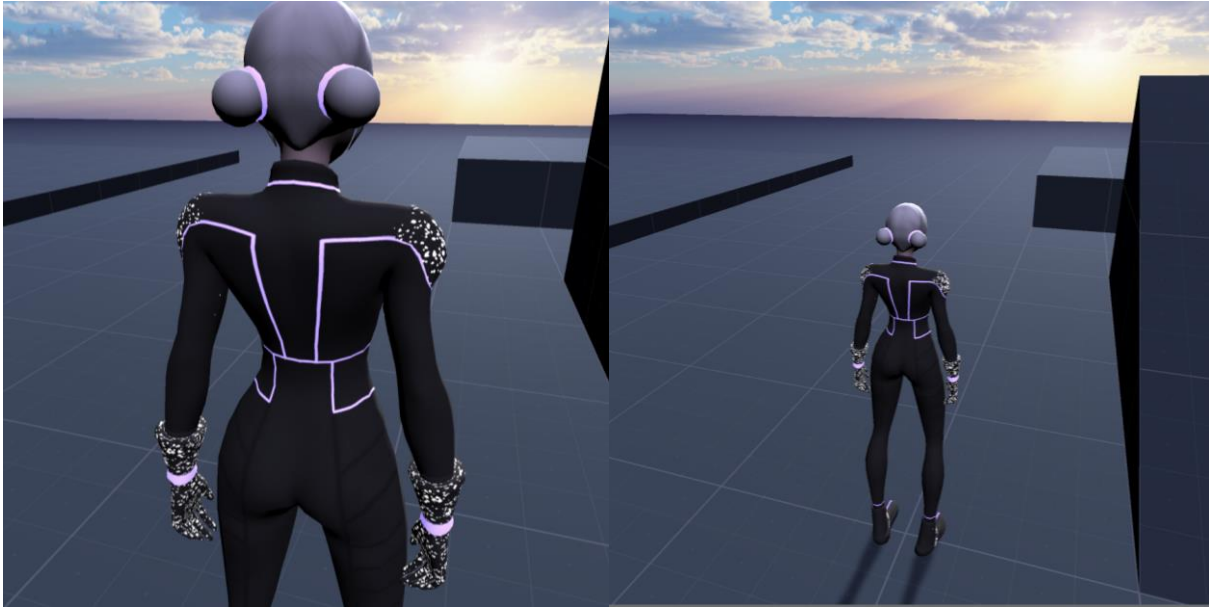
4.6.3. Kamera

U folderu *Skripte* kreirana je nova C# skripta koja je nazvana *ThirdPersonCamera*. Ova skripta služi za postavljanje kamere u trećem licu. Za početak je u *Start* funkciji koja se poziva samo jednom napisan kod koji zaključava kursor i koji ga čini nevidljivim, jer kursor u aplikaciji neće biti potreban.

```
Cursor.lockState = CursorLockMode.Locked;  
Cursor.visible = false;
```

U *Update* funkciji koja se poziva jednom po svakoj slici (*engl. frame*), napisan je kod koji omogućuje približavanje i udaljavanje kamere. Postavljena je brzina i glatki prijelaz pozicije kamere, kao i minimalna i maksimalna udaljenost od igrača.

```
scrollAmount = Input.GetAxis("Mouse ScrollWheel") * 3;  
wantedDistance -= scrollAmount;  
wantedDistance = Mathf.Clamp(wantedDistance, 0.7f, 2);  
dstFromTarget = Mathf.Lerp(dstFromTarget, wantedDistance, 0.04f);
```



Slika 4.45. Minimalna i maksimalna udaljenost kamere od lika

LateUpdate funkcija pokreće se na kraju svake slike i poziva se nakon što je pozvana *Update* funkcija. U njoj je definirano kretanje kamere po horizontali i vertikalni. Postavljeni su glatki prijelazi kamere i minimalna i maksimalna pozicija kamere po vertikalni. Dok se igrač ne kreće, kamerom se može kružiti oko igrača, a kada igrač krene hodati, kreće se u smjeru u kojem je okrenuta kamera.

```
 yaw += Input.GetAxis ("Mouse X") * 8;  
 pitch -= Input.GetAxis ("Mouse Y") * 8;  
 pitch = Mathf.Clamp(pitch, -40, 85);  
  
 currentRotation = Vector3.SmoothDamp (currentRotation, new Vector3  
 (pitch, yaw), ref rotationSmoothVelocity, rotationSmoothTime);  
 transform.eulerAngles = currentRotation;  
  
 transform.position = target.position - transform.forward *  
 dstFromTarget;
```



Slika 4.46. Mogućnost okretanja kamere oko lika

Kako bi se skripta *ThirdPersonCamera* uopće primijenila na kameru u sceni, selektirana je kamera i naredbom *Add Component > ThirdPersonCamera* u *Inspector* prozoru na nju je dodana skripta. Kao meta koju kamera prati kreiran je „dijete“ objekt lika, koji je izrađen naredbom *Game Object > Create Empty Child*. Objekt je postavljen u visini ramena lika. U postavkama kamere se potom „dijete“ objekt lika može postaviti kao *Target* objekt kojeg kamera prati.

4.6.4. Kretanje

U folderu *Skripte* kreirana je skripta *PlayerController* koja služi za omogućavanje kretanja lika i provođenje željenih animacija. Lik treba imati jednu animaciju dok stoji, drugu dok se kreće, i treću kada igrač pritisne gumb *E*, na koji lik maše rukom.

Pošto se lik ne smije moći kretati dok maše rukom, kreirana je *if-else* selekcija koja odvaja kretanje od mahanja rukom. Ako trenutno igrač ima mogućnost kretanja, postavlja se brzina kretanja uz glatke prijelaze pri kretanju i gravitacija koja utječe na lika ako pada s neke visine.

```

if (canMove){

    Vector2 input = new Vector2(Input.GetAxisRaw("Horizontal"),
    Input.GetAxisRaw("Vertical"));
    Vector2 inputDir = input.normalized;

    if (inputDir != Vector2.zero){
        float targetRotation = Mathf.Atan2(inputDir.x, inputDir.y)
        * Mathf.Rad2Deg + cameraT.eulerAngles.y;
        transform.eulerAngles = Vector3.up *
        Mathf.SmoothDampAngle(transform.eulerAngles.y,
        targetRotation, ref turnSmoothVelocity, turnSmoothTime);
    }

    float targetSpeed = walkSpeed * inputDir.magnitude;
    currentSpeed = Mathf.SmoothDamp(currentSpeed, targetSpeed, ref
    speedSmoothVelocity, speedSmoothTime);
    velocityY += Time.deltaTime * gravity;
    Vector3 velocity = transform.forward * currentSpeed + Vector3.up
    * velocityY;
    controller.Move(velocity * Time.deltaTime);

    if (controller.isGrounded){velocityY = 0;}

    float animationSpeedPercent = .5f * inputDir.magnitude;
    animator.SetFloat("speedPercent", animationSpeedPercent,
    speedSmoothTime, Time.deltaTime);

    if (Input.GetKeyDown(KeyCode.E)){canMove = false;}
}

```

Lik prelazi u stanje nemogućnosti kretanja ukoliko igrač pritisne tipku E. Tada se u *else* dijelu izvodi animacija mahanja ruke na određeno vrijeme.

```

else{
    float animationSpeedPercent = 1;
    animator.SetFloat("speedPercent", animationSpeedPercent,
    speedSmoothTime, Time.deltaTime);

    timer += Time.deltaTime;
    if (timer > 1.5f){
        canMove = true;
        timer = 0;
    }
}

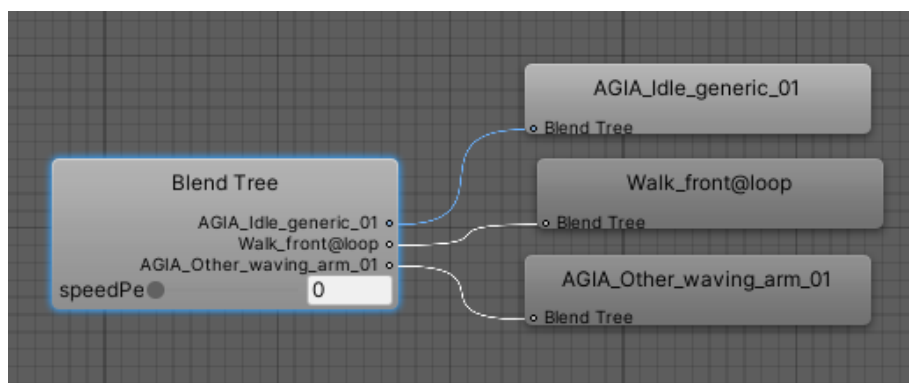
```

4.6.5. Animacije

Kako bi se na liku prikazivale animacije kao što su definirane u skripti za kretanje, prvo ih je potrebno besplatno preuzeti iz trgovine Unity Asset Store.

Odlaskom na *Asset Store* prozor direktno unutar Unity aplikacije, moguće je filtrirati pretraživanje na besplatne animacije. Odabrani su paketi s animacijama *Anime Girl Idle Animations Free* i *SD Unity-Chan Haon Custom*. Uvezene su animacije za mirovanje i mahanje rukom iz paketa *Anime Girl Idle Animations Free* i animacija za hodanje iz *SD Unity-Chan Haon Custom* paketa. Animacije su spremljene u folder *Animacije*.

Potom je u folderu *Lik* dodan novi kontroler za animacije desnim klikom na prozor i odabirom naredbe *Create > Animator Controller*. On se zatim dodaje na objekt lika kao *Controller*. Odlaskom u prozor *Animator* potrebno je kreirati *Blend Tree* koji kontrolira prijelaze u animacijama. Na njega se dodaju animacija za mirovanje, hodanje i mahanje rukom, čime redom poprimaju vrijednosti 0, 0.5 i 1. S obzirom da su po tim vrijednostima referencirane u skripti za kretanje, tako postavljene animacije već su vidljive na modelu.



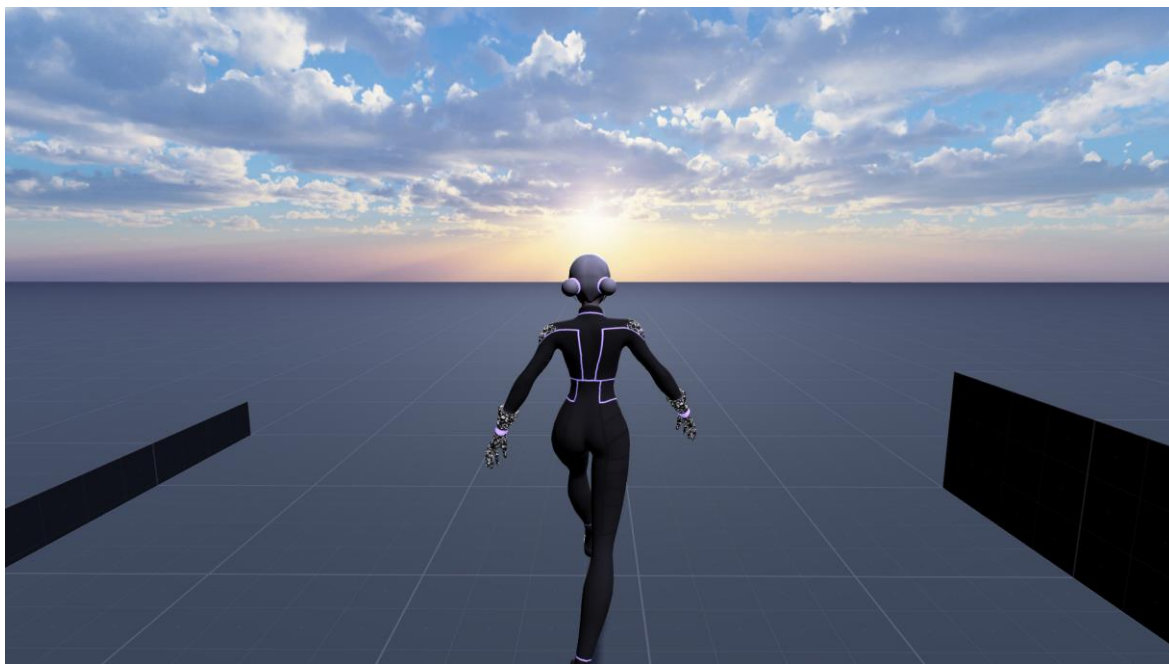
Slika 4.47. Blend Tree s dodanim animacijama

4.6.6. Kolizije

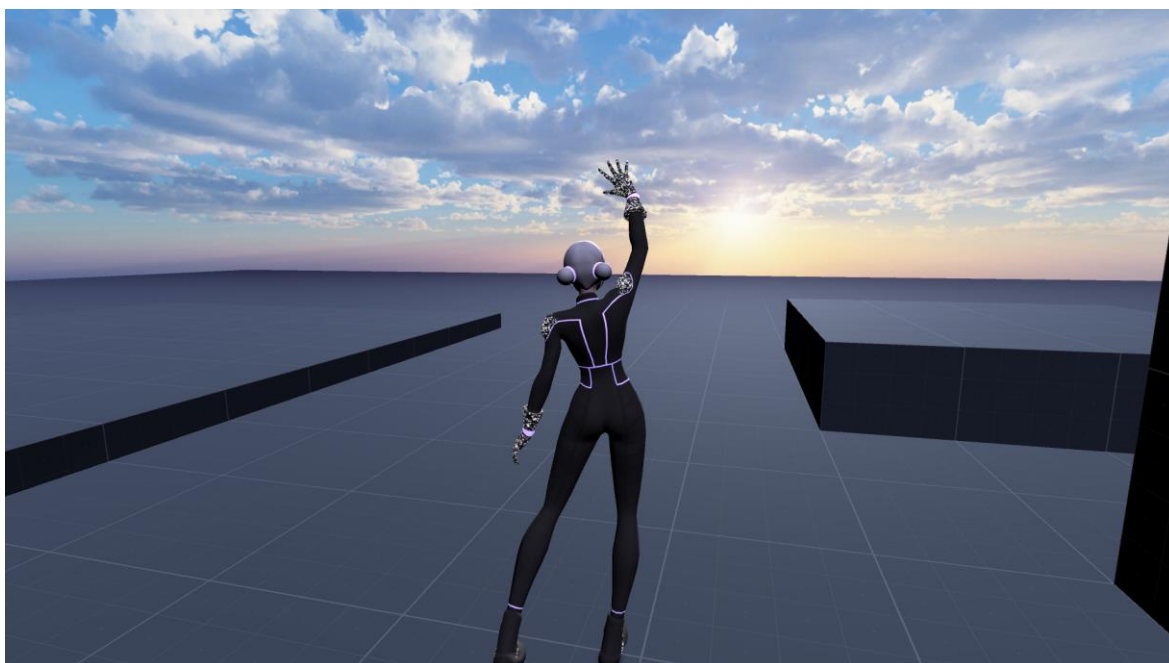
Dodavanje kolizije na objekt lika sprječava prolazanje lika kroz objekte. Dok je odabran lik u sceni, kolizija se dodaje klikom na *Add Component* i odabirom na *Character Controller*. Radijus kolizije podesi se na 0.22, a visina na 1.6, što je otprilike i visina lika.

4.6.7. Izrada aplikacije

Aplikacija se izvozi odabirom na *File > Build Settings*. Potrebno je odabrati *Add Open Scenes* i dodati kreiranu scenu. Scenu je potrebno selektirati u *Scenes In Build* prozoru. Zatim se odabire gumb *Build*. Nakon što se odabere folder u koju će se aplikacija spremiti, projekt se kreće izvoziti. Po završetku je dobivena .exe aplikacija koja sadrži gotovu scenu i spremna je za korištenje.



Slika 4.48. Snimak zaslona u gotovoj aplikaciji pri hodanju



Slika 4.49. Snimak zaslona u gotovoj aplikaciji pri mahanju rukom

5. Zaključak

Video igre od početka svog postojanja postavljaju brojne zahtjeve za dizajnere i programere kako bi bile igrive u realnom vremenu. Iako se snaga hardvera tokom godina iznimno povećala, današnja računala još su uvijek limitirana u mogućnostima prikaza 3D modela. Zato su osmišljene brojne tehnike koje rješavaju problem hardverskog ograničenja. U tehnike optimizacije spada pojednostavljivanje siluete modela, redukcija broja poligona, brisanje nevidljivih poligona, stvaranje LOD modela, ekonomično korištenje tekstura i upotreba mape normala.

Blender i Unity besplatni su alati koji su iznimno kompatibilni, ažurirani na današnje standarde industrije video igara i opremljeni svim potrebnim alatima za realizaciju ovakvog projekta. Blender i Unity sve su popularniji alati, što je rezultiralo velikim brojem lako dostupne literature i uputstava.

Proces izrade 3D lika za video igre zahtjeva stručno iskustvo u mnogim aspektima 3D modeliranja i umjetnosti općenito. Potrebno je krenuti od dojmive ideje koja odgovara potrebama video igre i stvoriti dobru vizualnu reprezentaciju izgleda lika. Potom se trebaju izraditi referentne slike koje su više tehničke prirode u usporedbi s concept art slikama. Lik se može u potpunosti kreirati unutar alata za 3D modeliranje Blender, uz upotrebu alata za izradu tekstura po izboru. Proces modeliranja lika sastoji se od skulptuiranja, izrade modela s niskim brojem poligona, teksturiranja i postavljanja kostura. Lik izrađen u alatu Blender lako se može implementirati u Unity projekt uz upotrebu animacija preuzetih s Unity Asset Store digitalne trgovine.



**Sveučilište
Sjever**



**SVEUČILIŠTE
SIEVER**



**IZJAVA O AUTORSTVU
I
SUGLASNOST ZA JAVNU OBJAVU**

Završni/diplomski 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, Heidi Blaslov, pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključiva autorica završnog rada pod naslovom Izrada humanoidnog 3D lika za video igru, te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

**Studentica:
Heidi Blaslov**

Heidi Blaslov

Sukladno Zakonu o znanstvenoj djelatnosti i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, Heidi Blaslov, neopozivo izjavljujem da sam suglasna s javnom objavom završnog rada pod naslovom Izrada humanoidnog 3D lika za video igru, čija sam autorica.

**Studentica:
Heidi Blaslov**

Heidi Blaslov

6. Literatura

- [1] Josh Petty: What is 3D Modeling & What's It Used For?, dostupno na:
<https://conceptartempire.com/what-is-3d-modeling/>,
pristupano 9.6.2020.
- [2] Skillshare Team: How to Design a Video Game Character, dostupno na:
<https://www.skillshare.com/projects/How-to-Design-a-Video-Game-Character/137870>,
pristupano 9.6.2020.
- [3] Tushar Tajane: 5 Ways 3D Modeling Is Used In Video Game Development, dostupno na:
<https://techzoom.org/five-ways-3d-modeling-used-video-game-development/>,
pristupano 9.6.2020.
- [4] PCGamesN: Kingdom Come: Deliverance, dostupno na:
<https://www.pcgamesn.com/kingdom-come-deliverance/kingdom-come-deliverance-release-date-beta-gameplay-combat>, pristupano 9.6.2020.
- [5] David Silverman: 3D Primer for Game Developers: An Overview of 3D Modeling in Games, dostupno na:
<https://gamedevelopment.tutsplus.com/articles/3d-primer-for-game-developers-an-overview-of-3d-modeling-in-games--gamedev-5704>, pristupano 9.6.2020.
- [6] Jerry Wu: Razlika kvalitete modela za film i video igru, dostupno na:
<https://www.pinterest.com/pin/353673376967906368/>,
pristupano 9.6.2020.
- [7] Behind The Cinematic: Jeff Chamberlain, dostupno na:
<http://www.behindthecinematic.com/people/jeff-chamberlain/>,
pristupano 9.6.2020.
- [8] XeNTax: Reverse Model Wireframe, dostupno na:
<https://forum.xentax.com/viewtopic.php?f=16&t=18005>,
pristupano 9.6.2020.
- [9] Laura Lohikoski, Elin Rudén: Optimization of 3D Game Models, dostupno na:
<https://www.diva-portal.org/smash/get/diva2:708048/FULLTEXT01.pdf>,
pristupano 9.6.2020.
- [10] Blender Stack Exchange: Is there an unsubdivide function in Blender?, dostupno na:
<https://blender.stackexchange.com/questions/21082/is-there-an-unsubdivide-function-in-blender>,
pristupano 9.6.2020.
- [11] Pluralsight: A Comparison of Modeling for Games and Modeling for Movies, dostupno na:
<https://www.pluralsight.com/blog/film-games/whats-the-difference-a-comparison-of-modeling-for-games->,
pristupano 9.6.2020.
- [12] Jordan Newman's Games Development Journal: Level of Detail, dostupno na:
<https://jnewman96.wordpress.com/year-2/oo-design/level-of-detail/>,
pristupano 9.6.2020.
- [13] Thomas Denham: What is UV Mapping & Unwrapping?, dostupno na:
<https://conceptartempire.com/uv-mapping-unwrapping/>,
pristupano 9.6.2020.

- [14] Wikimedia Commons: File:Cube Representative UV Unwrapping.png, dostupno na:
https://commons.wikimedia.org/wiki/File:Cube_Representative_UV_Unwrapping.png,
pristupano 9.6.2020.
- [15] Artyom Sergeev: Figuring out Texel Density, dostupno na:
<https://80.lv/articles/textel-density-tutorial/>,
pristupano 9.6.2020.
- [16] Chris Arnold: UV mapping for the SUITE - Quixel SUITE 2, dostupno na:
<https://nl.pinterest.com/pin/370561875571069210/>,
pristupano 9.6.2020.
- [17] Timothy Bermanseder: Technical and Visual Analysis of Overwatch, dostupno na:
<https://80.lv/articles/overwatch-technical-overview/>,
pristupano 9.6.2020.
- [18] KatsBits: Make Better Textures, The 'Power Of Two' Rule & Proper Image Dimensions, dostupno na:
<https://www.katsbits.com/tutorials/textures/make-better-textures-correct-size-and-power-of-two.php>,
pristupano 9.6.2020.
- [19] Polycount: Normal Map, dostupno na:
http://wiki.polycount.com/wiki/Normal_map,
pristupano 9.6.2020.
- [20] Learn OpenGL: Normal Mapping, dostupno na:
<https://learnopengl.com/Advanced-Lighting/Normal-Mapping>,
pristupano 9.6.2020.
- [21] Mattcoombe: G-SWAT Road Warrior - Low Poly vs High Poly, dostupno na:
<https://mattcoombe.cgsociety.org/76mw/g-swat-road-warrior->,
pristupano 9.6.2020.
- [22] Pablo Farias Navarro: How to Import Blender Models into Unity – Your One-Stop Guide, dostupno na:
<https://gamedevacademy.org/how-to-import-blender-models-into-unity-your-one-stop-guide/>,
pristupano 9.6.2020.
- [23] Blender: About, dostupno na:
<https://www.blender.org/about/>,
pristupano 9.6.2020.
- [24] Blender: Blender 2.80, dostupno na:
<https://www.blender.org/download/releases/2-80/>,
pristupano 9.6.2020.
- [25] Unity: Unity Core Platform, dostupno na:
<https://unity.com/products/core-platform>,
pristupano 9.6.2020.
- [26] PngFind: Unity Logo, dostupno na:
https://www.pngfind.com/mpng/ibJxwbR_unity-logo-unity3d-unity-logo-png-transparent-png/,
pristupano 9.6.2020.
- [27] Unity: Our Company, dostupno na:
<https://unity.com/our-company>,
pristupano 9.6.2020.

- [28] Dmitrij Leppée: ZBrush: Stvaranje 3D lika za igru ili cinematic, dostupno na:
<https://www.vidilab.com/how-to/1552-zbrush-stvaranje-3d-lika-za-igru-ili-cinematic/>,
pristupano 9.6.2020.
- [29] Exoside: Quad Remesher, dostupno na:
<https://exoside.com/quadremesher/>,
pristupano 9.6.2020.
- [30] Josh Petty: What is Retopology? (A Complete Intro Guide For Beginners), dostupno na:
<https://conceptartempire.com/retopology/>,
pristupano 9.6.2020.
- [31] Blender Manual: F2, dostupno na:
<https://docs.blender.org/manual/en/dev/addons/mesh/f2.html>,
pristupano 9.6.2020.
- [32] David Stutz: A Formal Definition of Watertight Meshes, dostupno na:
<https://davidstutz.de/a-formal-definition-of-watertight-meshes/>,
pristupano 9.6.2020.
- [33] Adam Johnson: Diffuse Maps, dostupno na:
<https://docs.cryengine.com/display/SDKDOC2/Diffuse+Maps>,
pristupano 9.6.2020.
- [34] Procreate, dostupno na:
<https://procreate.art/>,
pristupano 9.6.2020.
- [35] Blog Cat: Emission Map Guide For Artists, dostupno na:
<https://megacatstudios.com/blogs/press/emission-map-guide-for-artists>,
pristupano 9.6.2020.
- [36] Blender Manual: Armatures, dostupno na:
<https://docs.blender.org/manual/en/latest/animation/armatures/introduction.html>,
pristupano 9.6.2020.
- [37] Jaime Rios Velasco: Blender: Weight Paint – Simply Explained, dostupno na:
<https://all3dp.com/2/blender-weight-paint-simply-explained/>,
pristupano 9.6.2020.
- [38] Blender Manual: Weight Paint, dostupno na:
https://docs.blender.org/manual/en/latest/sculpt_paint/weight_paint/introduction.html,
pristupano 9.6.2020.

7. Popis slika

Slika 2.1. Kingdom Come: Deliverance svijet [4]	2
Slika 2.2. Razlika kvalitete modela za film (lijevo) i video igru (desno) [6]	3
Slika 2.3. Usporedba lika za video igru (lijevo) i film (desno) [7]	4
Slika 2.4. Usporedba okoliša za video igru (gore) i film (dolje) [7]	4
Slika 2.5. Triangulacija modela [8].....	5
Slika 2.6. Pojednostavljena silueta fotoaparata [5]	5
Slika 2.7. Razlika modela oružja iz trećeg (lijevo) i prvog lica (desno) [5].....	6
Slika 2.8. Reduciranje rubnih petlji [10].....	7
Slika 2.9. Modeli različitog LOD [12].....	7
Slika 2.10. Jednostavni modeli s teksturama [11].....	8
Slika 2.11. Vizualizacija UV odmotavanja [14]	8
Slika 2.12. Optimalno iskorišten UV prostor.....	9
Slika 2.13. Ispravan (lijevo) i neispravan (desno) texel density [16].....	9
Slika 2.14. UV mapa sa zrcaljenjem [17]	10
Slika 2.15. Vizualizacija količine prostora koja se treba obraditi [18]	11
Slika 2.16. Mapa normala [20]	12
Slika 2.17. Model niske razlučivosti s mapom normala i model visoke razlučivosti [21]	12
Slika 3.1. Razlika koordinatnih sustava u Blender-u (lijevo) i Unity-u (desno)	13
Slika 3.2. Blender logo [23].....	14
Slika 3.3. Blender korisničko sučelje.....	14
Slika 3.4. Unity logo [26]	16
Slika 3.5. Unity korisničko sučelje	16
Slika 3.6. Unity Asset Store.....	17
Slika 4.1. Gotov model lika	18
Slika 4.2. Kolekcija slika za inspiraciju	19
Slika 4.3. Iteracije za concept art	19
Slika 4.4. Reference bez detalja.....	20
Slika 4.5. Reference s detaljima.....	20
Slika 4.6. Reference u 3D prostoru	21
Slika 4.7. Organizacija scene	21
Slika 4.8. Priprema za referentni model.....	22
Slika 4.9. Edit Mode način rada.....	23
Slika 4.10. Referentni model tijela.....	24
Slika 4.11. Referentni model glave.....	25
Slika 4.12. Geometrija nakon opcije Remesh	26
Slika 4.13. Razlika između Flat i Smooth načina sjenčanja.....	27
Slika 4.14. Gotova skulptura tijela i glave	28
Slika 4.15. Skulptura rukavice	28
Slika 4.16. Skulptura ovratnika.....	29
Slika 4.17. Skulptura čizmi.....	30

Slika 4.18. Gotova skulptura lika.....	31
Slika 4.19. Početak retopologije lica.....	33
Slika 4.20. Retopologija lica.....	34
Slika 4.21. Retopologija tijela u procesu	34
Slika 4.22. Gotova retopologija tijela	35
Slika 4.23. Model oka	36
Slika 4.24. Modeli očiju, trepavica i obrvi.....	36
Slika 4.25. Modeli šiški, punde i gumice za kosu	37
Slika 4.26. Model kose	38
Slika 4.27. Gotova optimizacija modela	39
Slika 4.28. Šavovi za tijelo i glavu.....	41
Slika 4.29. Šavovi za kosu	41
Slika 4.30. Pregled materijala	42
Slika 4.31. UV mapa s prikazom rastezanja	43
Slika 4.32. Ručno slaganje UV mape za kosu	43
Slika 4.33. Ručno slaganje UV mape za tijelo.....	44
Slika 4.34. Model nakon izrade UV mapi.....	45
Slika 4.35. Difuzna mapa za kosu.....	46
Slika 4.36. Difuzna mapa za tijelo	46
Slika 4.37. Model s difuznim mapama	47
Slika 4.38. Mapa emisije i njen utjecaj na izgled modela u mraku	48
Slika 4.39. Postavljeni čvorovi za mapu normala	48
Slika 4.40. Model prije i nakon dodavanja mape normala.....	49
Slika 4.41. Model nakon podešavanja materijala.....	50
Slika 4.42. Rigify kostur prije i nakon prilagođavanja modelu.....	51
Slika 4.43. Vrijednosti težinskog bojanja prikazane bojom [38]	52
Slika 4.44. Izgled scene u alatu Unity.....	54
Slika 4.45. Minimalna i maksimalna udaljenost kamere od lika.....	55
Slika 4.46. Mogućnost okretanja kamere oko lika.....	56
Slika 4.47. Blend Tree s dodanim animacijama.....	58
Slika 4.48. Snimak zaslona u gotovoj aplikaciji pri hodanju	59
Slika 4.49. Snimak zaslona u gotovoj aplikaciji pri mahanju rukom	59

8. Prilozi

Aplikacija sa demonstracijom 3D humanoidnog lika u razvojnom okruženju Unity, dostupna za preuzimanje na:

<https://mega.nz/file/LIAySaiI#vU80JaQ7zyiPlfAYrnQJ3tD-6RXtGRJlxmI9OIgBE48>