

Modeliranje, teksturiranje i animiranje lika videoigre u programskom alatu Blender

Šiško, Tin

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:211:112834>

Rights / Prava: [Attribution 3.0 Unported](#)/[Imenovanje 3.0](#)

Download date / Datum preuzimanja: **2024-05-20**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Tin Šiško

**MODELIRANJE, TEKSTURIRANJE I
ANIMIRANJE LIKA VIDEOIGRE U
PROGRAMSKOM ALATU BLENDER**

ZAVRŠNI RAD

Varaždin, 2023.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Tin Šiško

Matični broj: 0016146603

Studij: Informacijski i poslovni sustavi

**MODELIRANJE, TEKSTURIRANJE I ANIMIRANJE LIKA
VIDEOIGRE U PROGRAMSKOM ALATU BLENDER**

ZAVRŠNI RAD

Mentor/Mentorica:

Doc. dr. sc. Mladen Konecki

Varaždin, kolovoz 2023.

Tin Šiško

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvatanjem odredbi u sustavu FOI-radovi

Sažetak

U ovom radu će se prikazati i opisati izrada 3D modela čovjeka za videoigre u programu Blender, te način na koji se 3D model prenosi u program za izradu videoigara Unity.

Izrada 3D modela podijeljena je u 5 faza: prikupljanje referentnih slika, digitalno kiparenje, retopologija i izrada ostalih objekata, teksturiranje i animiranje.

Prikupljanje referentnih slika je prva faza u kojoj će se prikupiti slike na kojima se nalaze objekti koji pomažu u razumijevanju oblika i materijala 3D modela. Nakon što se prikupe referentne slike, izrađuje se digitalni kip s visokom razinom detalja, ali i velikim brojem poligona. Budući se radi o modelu namijenjenom za primjenu u video igrama, potrebno ga je izraditi s što manjim brojem poligona kako se ne bi naštetilo performansama igre. Stoga je potrebno napraviti retopologiju digitalnog kipa, odnosno konačnu topologiju 3D modela s malim brojem poligona koja će se koristiti u videoigri. Pomoću konačne mreže tijela napraviti će se ostali odjevni predmeti. Kada se završi s izradom topologije, objekt je spreman za animiranje, odnosno izradu animacija poput hodanja i trčanja. Za ručno bojanje modela, potrebno je stvoriti sliku i povezati je s materijalom koji je dodijeljen modelu. Kada su sve faze izrade modela gotove, model se može izvesti iz Blender-a i prebaciti u Unity za daljnje korištenje u procesu izrade video igara.

Ključne riječi: Blender, 3D modeliranje, retopologija, teksturiranje, animacija, videoigre, Unity

Sadržaj

1. Uvod	1
2. Metode i tehnike rada	2
3. Osnovni pojmovi 3D modeliranja.....	3
4. Blender	5
4.1. Korisničko sučelje Blender-a	5
4.1.1. 3D prikaz.....	6
4.1.1.1. Načini rada u 3D prikazu	7
4.1.1.2. Načini sjenčanja u 3D prikazu	8
4.1.2. Lista objekata u sceni.....	9
4.1.3. Postavke	9
4.1.4. Vremenska crta	10
5. Prikupljanje referentnih slika	11
6. Digitalno kiparenje	12
7. Retopologija.....	16
8. Teksturiranje	21
9. Animiranje.....	31
9.1. Inverzna kinematika	32
10. Primjena 3D modela u Unity	36
11. Zaključak	44
Popis literature	45
Popis slika	47

1. Uvod

3D modeliranje je proces izrade trodimenzionalne digitalne reprezentacije nekog objekta koji može biti iz stvarnog svijeta ili zamišljen od strane čovjeka. Ima široku primjenu u raznim granama industrije poput medicine, zabavne industrije, autoindustrije, građevine, vojne industrije itd. Kako bi 3D modeli ispunili svoju svrhu i bili privlačni korisnicima, 3D dizajneri moraju imati znanja iz raznih područja poput ergonomije, anatomije, umjetnosti, tehnologije, psihologije itd. Budući je 3D modeliranje vrlo interesantno zanimanje jer se radi o spoju tehnologije i umjetnosti, postoji velika zainteresiranost ljudi za rad u tom području.

Tijekom izrade 3D modela za videoigre potrebno je voditi računa o što manjim troškovima resursa uređaja na kojemu će se igrati igra. Dobar primjer toga je vidljiva razlika u razini detalja videoigara koje se igraju na računalu i na pametnim mobitelima. Postoje razne tehnike koje se primjenjuju kako bi se dodatno uštedili resursi, a što manje narušio izgled samog modela poput glatkog sjenčanja (engl. Shade smooth), korištenja mapa normala (engl. Normal map) i sakrivanja određenih nevidljivih dijelova objekata.

U sljedećim poglavljima bit će detaljnije opisan program Blender, faze izrade 3D modela čovjeka i implementacija modela u program Unity.

Razlog odabira ove teme je moja velika zainteresiranost za područje videoigara, te prethodno iskustvo u korištenju programa Blender za izradu 3D modela. Smatram da je uloga grafike u videoigramima od velike važnosti jer njezina atraktivnost privlači igrača da isproba video igru i da je nastavi igrati. Uloga 3D modelera je prenesti zamisli dizajnera iz 2D slike u trodimenzionalni prostor.

Postoje razni programi za izradu 3D modela poput industrijskog standarda ZBrush-a koji se najčešće koristi za digitalno kiparenje, Autodesk Maya-e, Blender-a, Cinema 4D, 3Ds Max, itd. Razlog zašto se u ovom radu koristi Blender je to što je besplatan, a osim toga ima i veliku zajednicu korisnika te tako i veliku količinu edukativnog sadržaja.

2. Metode i tehnike rada

Za izradu osnovnog oblika tijela čovjeka izraditi će se više jednostavnih odvojenih objekata koji će predstavljati dijelove tijela poput glave, vrata, trupa, ruku i nogu. Nakon što se pojedini dijelovi tijela oblikuju, mogu se sjediniti u jedan objekt. Kako bi se detaljnije izradilo tijelo i kosa čovjeka koristi se digitalno kiparenje, odnosno razvlačenje i oblikovanje poligona na način da njihov oblik što pobliže prikazuje oblik sa referentnih slika.

Proces retopologije provodi se pomoću modifikatora koji omogućavaju nedestruktivnu obradu mreže objekta i ubrzavaju sam proces izrade topologije. Kako bi se dodatno smanjio broj poligona, lančić ogrlice će biti izrađen uz pomoć alfa kanala slike koja će prikazivati dio ogrlice.

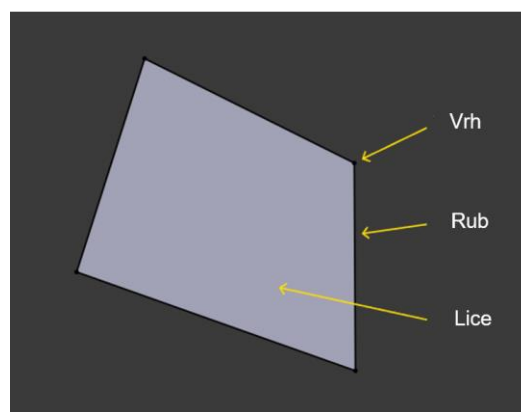
Prije početka ručnog bojanja tekstura prenijeti će se detalji s digitalne skulpture na sliku koja će služiti kao početne smjernice za sjene i svjetla koja se nalaze na tijelu.

Za izradu 3D modela, njegovo teksturiranje i animiranje korišten je program Blender, dok je za izradu slike ogrlice korišten program za crtanje Krita, a za izradu jednostavne video igre korišten je program Unity.

3. Osnovni pojmovi 3D modeliranja

Radi boljeg razumijevanja sadržaja ovog rada potrebno je objasniti neke osnovne pojmove koji će se spominjati u nastavku ovog rada.

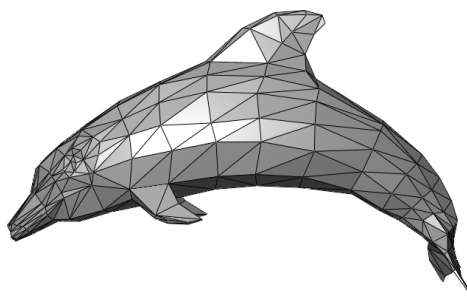
3D modeli se sastoje od 3 elementa, a to su: vrhovi (engl. Vertices), rubovi (engl. Edges) i lica (engl. Faces). Vrhovi su točke, odnosno koordinate u 3D prostoru, dok su rubovi ravne linije koje povezuju dva vrha. Lica su 3 ili više povezanih rubova koji tvore ispunjenu površinu i ona čine 3D model vidljivim [1].



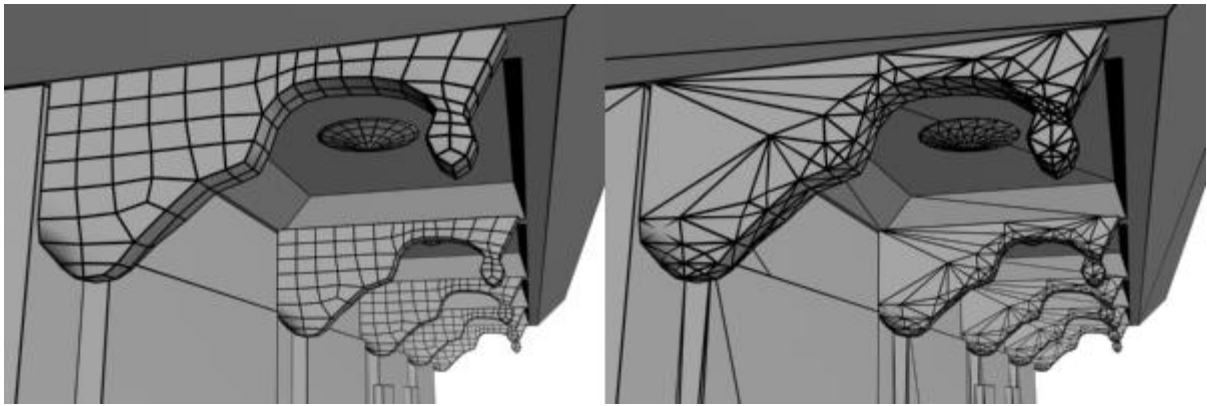
Slika 1: Osnovni elementi 3D modela (Izvor: Blender User Manual, 2023)

Mreža 3D modela je skup svih točaka, rubova i lica koji sačinjavaju 3D model i ona predstavlja oblik 3D modela [2].

Topologija mreže 3D modela se odnosi na raspodjelu i povezanost točaka, rubova i lica koji sačinjavaju 3D model. Važnost ispravne topologije se očituje u načinu na koji model reagira na svjetlost, koliko je jednostavno urediti postojeći oblik modela, koliko je jednostavno pripremiti model za teksturiranje, animiranje i animirati ga [3].



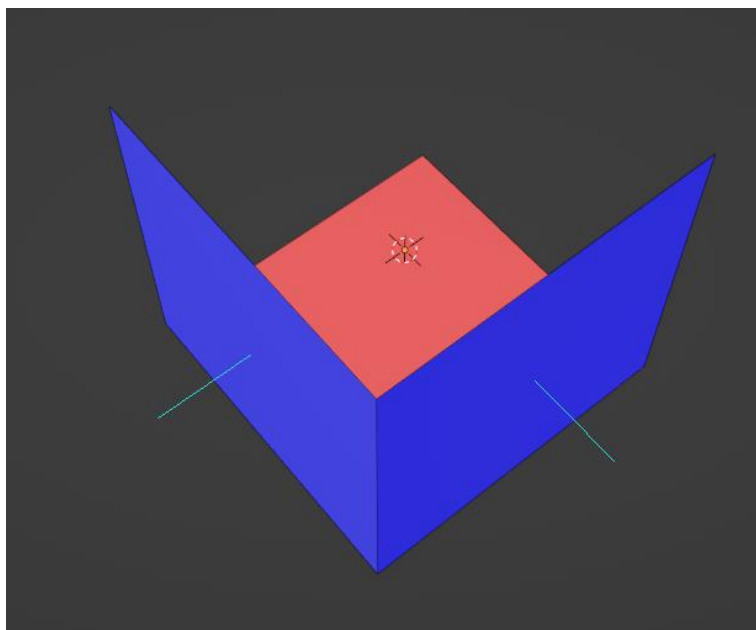
Slika 2: Mreža 3D modela (Izvor: Wikipedia, 2023)



Slika 3: Različite topologije modela istih oblika (Izvor: Turbosquid 3D Resources, 2023)

Svako lice ima unutarnju i vanjsku stranu, a strana je određena smjerom na koji gleda normala okomita na lice. Blender ima mogućnost provjere orijentacije lica tako da se vanjska (vidljiva strana) prikazuje plavom bojom, a unutarnja strana prikazuje crvenom bojom. Također moguće je prikazati i smjer normale svijetloplavom crtom koja izlazi okomito iz lica s vanjske strane. Posebice pri izradi 3D modela za video igre važno je obratiti pozornost na ispravan smjer normali lica, zbog mogućih anomalija tijekom glatkog sjenčanja ili teksturiranja.

3D modeli unutar Unity-a po standardu imaju vidljivu samo jednu stranu lica (vanjsku) dok je druga strana transparentna. Moguće je korištenje skripti koje omogućavaju da materijal objekta ipak bude vidljiv sa obje strane lica.



Slika 4. Normale lica u Blender-u

4. Blender

Blender je besplatni program otvorenog koda, koji omogućava kreativnim i inovativnim ljudima da stvaraju 3D sadržaj, te da ga objavljuju i prodaju. Također nudi mogućnost korištenja raznih funkcionalnosti poput modeliranja, izrade kostura za animiranje, animiranja, simulacija, renderiranja, kompozicije i izrade igara [4].

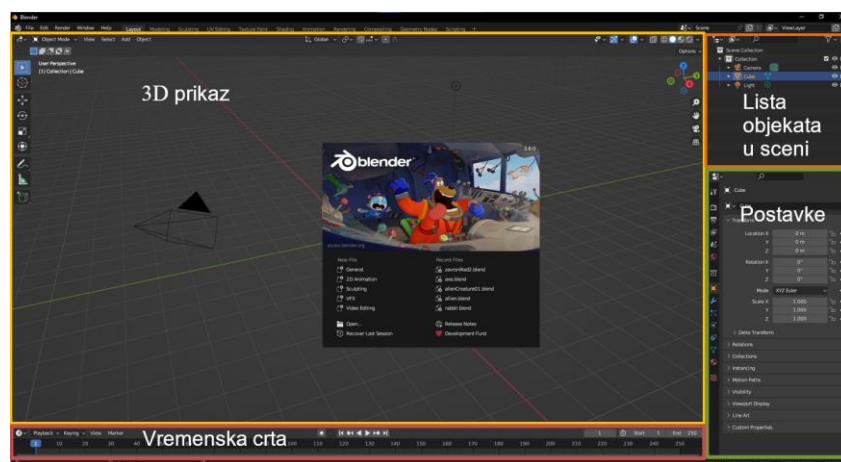
Moguće je koristiti Blenderov API za pisanje skripti u Pythonu kako bi se stvorili novi alati ili izmijenili postojeći.

4.1. Korisničko sučelje Blender-a

Korisničko sučelje Blender-a vizualno izgleda komplicirano zbog velikog broja alata i mogućnosti koje sam program nudi. Međutim ovisno o projektu koji se radi, potrebno je znati služiti se samo određenim alatima i mogućnostima. Korištenje prečaca znatno olakšava i ubrzava sam rad u programu.

Nakon pokretanja Blendera, na sredini ekrana se nalazi prozorčić koji nudi stvaranje novog projekta ili pokretanje nekog prethodno korištenog projekta. Za izlazak iz tog prozorčića potrebno je kliknuti bilo gdje drugdje na ekran.

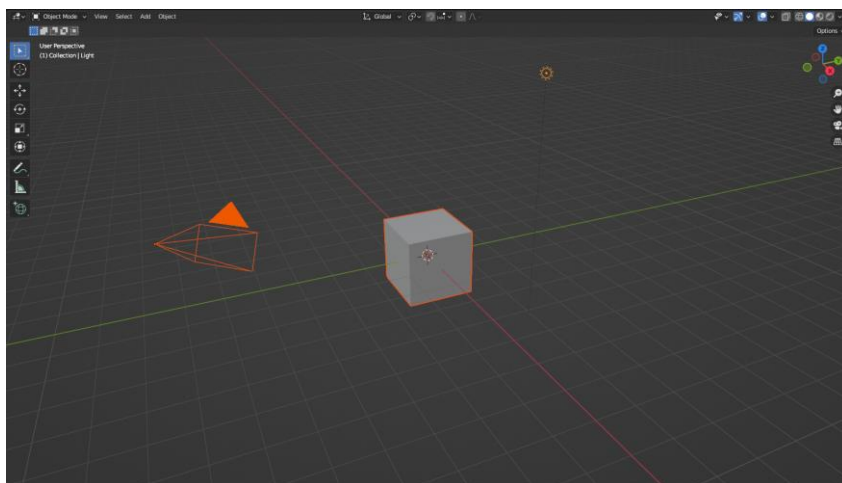
Početno korisničko sučelje se sastoji od 4 prozora od kojih se 3 koriste tijekom cijelog procesa izrade 3D modela, a to su 3D prikaz (zauzima većinu ekrana), lista objekata u sceni (u gornjem desnom kutu sučelja) i postavke (smještene ispod liste objekata). Četvrti prozor se nalazi ispod 3D prikaza, a predstavlja vremensku crtu koja služi za prikaz vremenskih okvira tijekom prikaza animacija.



Slika 5. Početno sučelje Blender-a

4.1.1. 3D prikaz

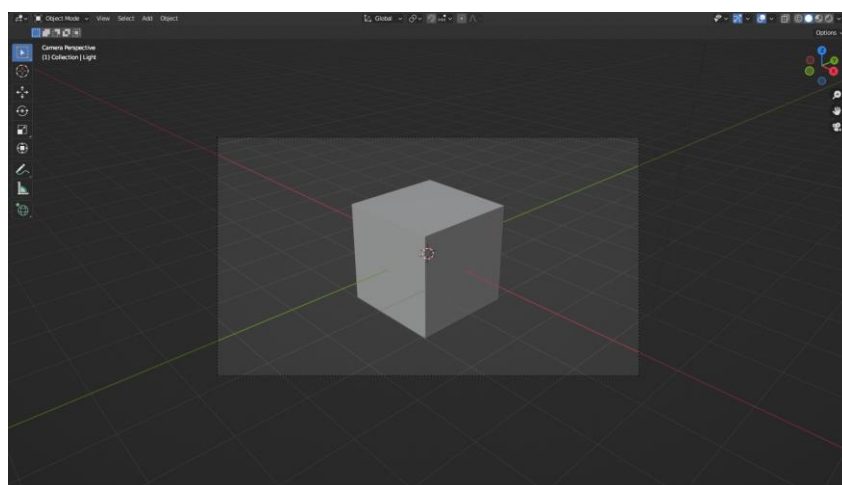
3D prikaz se koristi za prikazivanje i postavljanje scene, izradu 3D modela, bojanje tekstura, bojanje utjecaja kostiju na dijelove modela, pozicioniranje kostiju i u ovom prozoru se provodi najviše vremena. Kada se stvori novi projekt, u 3D prikazu se nalaze 3 objekta, a to su kamera, kocka i svjetlo (na slici 6.).



Slika 6. 3D prikaz scene

Odabrani objekt će imati žuti obrub, a može se odabrati i više objekata pri čemu će prethodno odabrani imati narančasti obrub, a zadnje odabrani žuti.

Kamera prikazuje dio scene koji će se vidjeti (svjetliji pravokutnik na slici 7.) nakon renderiranja, tj. stvaranje slike ili nekog videa scene.



Slika 7. Pogled iz kamere

4.1.1.1. Načini rada u 3D prikazu

U 3D prikazu postoji 7 načina rada koji omogućavaju izvođenje različitih akcija nad objektima u sceni, a to su:

- Objektni način rada
- Način uređivanja
- Način kiparenja
- Bojanje vrhova
- Bojanje težine
- Bojanje tekstura
- Način poziranja

Objektni način rada omogućava pozicioniranje, rotaciju i skaliranje objekata u globalnom prostoru. Postoje postavke objekata u sceni koje se mogu mijenjati samo u objektnom načinu rada.

Način uređivanja omogućava izmjenu mreže objekta, odnosno stvaranje i pomicanje novih točaka, rubova i lica, promjenu orijentacije lica, rotaciju i skaliranje rubova i lica u lokalnom prostoru. Prečac za izmjenu između objektnog načina i načina uređivanja je tipka TAB.

Način kiparenja omogućava korištenje kistova za digitalno kiparenje koje na različite načine pomiču i oblikuju mrežu objekta. Neki od korisnih kistova su oštro crtati (engl. Draw Sharp), glinene trake (engl. Clay Strips), napuhati (engl. Inflate), zgrabiti (engl. Grab), naborati (engl. Crease) i ugladiti (engl. Smooth). U ovom načinu rada moguće je na 2 načina izmijeniti topologiju mreže. Jedan od načina je pomoću alata Dinamična topologija (engl. Dynamic Topology), koji omogućava da se tijekom prelaska kista preko površine objekta poveća ili smanji broj poligona na tom području. Jedna od važnijih karakteristika ovog alata je da pretvara sva lica u trokute. Nedostatak ove tehnike je da pri velikim promjenama može doći do usporavanja rada računala. Prednost korištenja ovog alata je u tome da se na pojedini dio modela može postaviti točno željena gustoća lica. S druge strane postoji alat za ponovno stvaranje mreže (engl. Remesh) koji će postaviti gustoću mreže na jednaku vrijednost za cijeli model. Prednost ove tehnike je urednija topologija (lica su četverokuti) i jednaka brzina rada računala tijekom korištenja kistova, a nedostatak se nalazi u nepotrebnom opterećenju računala s visokom razinom lica na dijelovima na kojima ona nisu potrebna.

Bojanje vrhova nudi mogućnost vrlo jednostavnog bojanja 3D modela na način da se u određenu boju mogu obojati vrhovi i prostor oko njih. Primjena ovog načina rada je korisna

ako želite na brzinu obojati digitalni kip (koji ima veliki broj vrhova, a time i dobru rezoluciju za bojanje).

Bojanje težine se koristi za povezivanje odnosa između kostiju i dijelova 3D modela. Moguće je odrediti koliko će pojedina kost pri pokretanju utjecati na pokretanje određenog dijela 3D modela. Kada se odabere neka kost, lica na modelu su obojana u različite boje koje mogu varirati između plave, zelene, žute i crvene. Hladnije boje predstavljaju slabi ili nikakav utjecaj kosti na taj dio mreže objekta, a toplije boje znače da kost ima utjecaj na taj dio mreže objekta.

Bojanje tekstura omogućava bojanje slika koje su povezane sa materijalom 3D modela, na način da prelaskom kista preko 3D modela ili slike promjena bude vidljiva i na modelu i na slici. Kako bi prethodno bilo moguće, potrebno je napraviti još neke međukorake koji će kasnije u poglavlju teksturiranja detaljnije biti objašnjeni.

Način pozicioniranja služi za postavljanje 3D modela pomoću kostiju u određene poze. U tom načinu rada moguće je kostima dodijeliti ograničenja što se tiče lokacije i rotacije. Ovaj način rada se koristi i tijekom animiranja za stvaranje ključnih slika.

4.1.1.2. Načini sjenčanja u 3D prikazu

Postoje 4 načina sjenčanja u 3D prikazu, a to su:

- Žičani okvir
- Solidni način
- Pregled materijala
- Renderirani način

Žičani okvir (engl. Wireframe) prikazuje samo rubove 3D modela na sceni bez ispune lica. Korisno ga je koristiti kada nismo sigurni kako izgleda dio topologije koji se nalazi unutar nekog objekta.

Solidni način prikazuje lica objekta jednostavnim sjenčanjem s ciljem isticanja geometrijskih osobina objekta, a dobro ga je koristiti za modeliranje i kiparenje.

Pregled materijala prikazuje lica objekta s odabranim materijalima i po standardu koristi HDRI sliku za osvjetljenje objekta. HDRI slike za razliku od običnih digitalnih slika imaju sposobnost pohraniti stvarne vrijednosti svjetline stvarnog svijeta [5]. Ovaj način rada je dobar za pregledavanje materijala i bojanje tekstura s malim opterećenjem računala.

Renderirani način omogućava finalni prikaz scene s svim svjetlima i njihovim efektima. Ovaj način prikazuje u stvarnom vremenu kako bi izgledala scena kada se želi stvoriti slika ili video scene [6].

4.1.2. Lista objekata u sceni

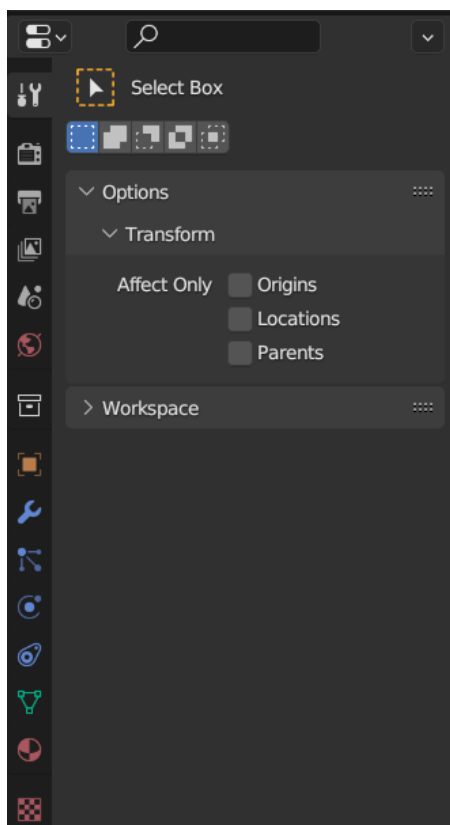
Lista objekata u sceni služi za prikaz hijerarhije svih objekata koji postoje u sceni trenutno otvorenog projekta. Objekte je moguće podijeliti u grupe, odabrati koji objekti će biti prikazani u sceni, koji objekti su selektivni i koji objekti će se renderirati u konačnoj slici ili videu. Vrste objekata se razlikuju po ikoni koja se nalazi uz njih.



Slika 8. Lista objekata u sceni

4.1.3. Postavke

Prozor s postavkama nudi mnogo mogućnosti za promjenu postavki trenutno odabranog alata, postavki renderiranja scene, postavki scene, odabranog objekta, modifikatora, čestica, fizike itd. Pomoću postavki možemo mijenjati način na koji modifikatori utječu na odabrani objekt i njihov redoslijed primjene. Ovisno o načinu rada 3D prikaza u kojemu se nalazimo mijenjaju se postavke koje su ponuđene. Na primjer postavke za promjenu jačine kista za digitalno kiparenje se prikazuju samo kada se nalazimo u načinu za digitalno kiparenje.



Slika 9. Prozor s postavkama

4.1.4. Vremenska crta

Vremenska crta prikazuje na kojem vremenskom okviru se trenutno nalazimo i nudi mogućnost za promjenu trenutnog vremenskog okvira, pokretanje i zaustavljanje animacije, te odabir početnog i završnog vremenskog okvira koji će se pustiti.

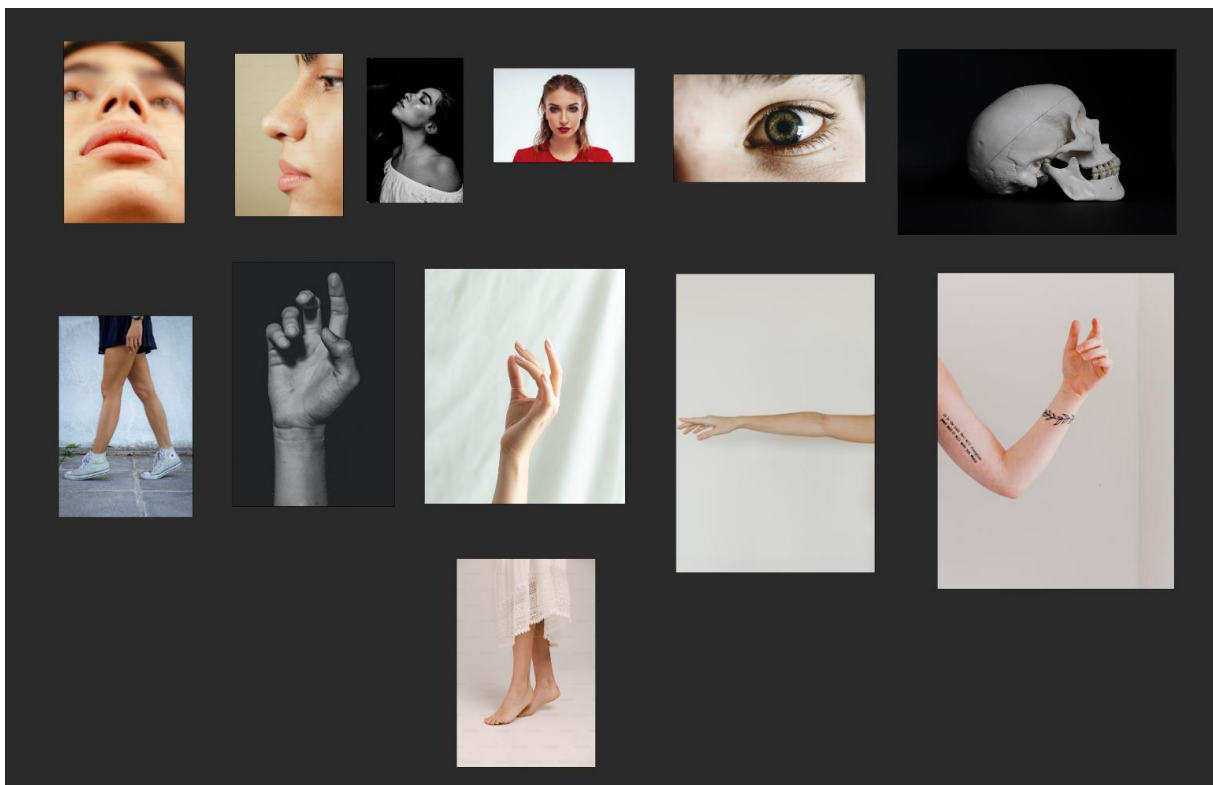


Slika 10. Vremenska crta

5. Prikupljanje referentnih slika

Kao što je i u programiranju prije izrade programskih rješenja potrebno izraditi dokumentaciju kako bismo imali točne zahtjeve korisnika i funkcionalnosti koje aplikacija mora ispuniti, tako je i u 3D modeliranju korisno prikupiti referentne slike kako naši modeli ne bi svojim izgledom odstupali od želja kupaca.

Referentne slike omogućavaju 3D modeleru da shvati kakav bi oblik neki objekt trebao imati, te kako bi različiti materijali trebali izgledati i reagirati na svjetlost. U slučaju da 3D modeler nema specifično zadane smjernice kako nešto treba izgledati, on može referentne slike koristiti kao izvor ideja kako bi samostalno kreirao jedinstveni dizajn 3D modela. Pri odabiru slika često nije dovoljno prikupiti samo jednu sliku željenog objekta, nego je potrebno prikupiti slike objekta iz različitih kutova i osvjetljenja. Postoje razni programi koji omogućavaju prikupljanje slika na jedno mjesto i njihovo prikazivanje. Za prikupljanje referentnih slika u ovom radu se koristi program PureRef radi svoje jednostavnosti.



Slika 11. Skup referentnih slika u programu PureRef (Slike preuzete sa stranice Unsplash)

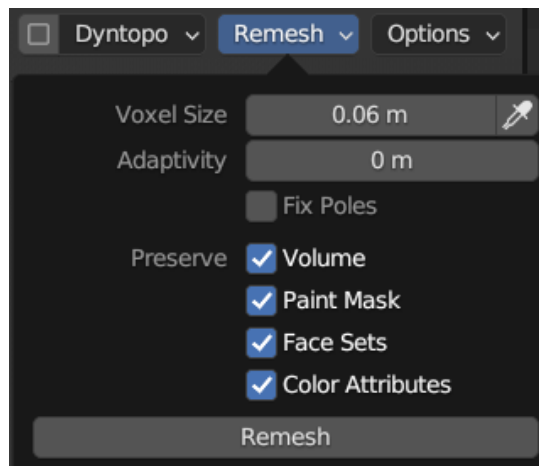
6. Digitalno kiparenje

Kako bi se lakše postigao početni oblik tijela čovjeka, stvorit će se više jednostavnih geometrijskih oblika koji će predstavljati udove. Stvaranje novih geometrijskih objekata se postiže korištenjem prečaca Shift+A > Mesh ili putem izbornika na vrhu prozora 3D prikaza Add > Mesh. Pomicanje objekata u prostoru postiže se korištenjem prečaca G nakon što se odabere objekt, dok se rotiranje postiže pritiskom tipke R, a skaliranje pritiskom tipke S. Za dodavanje novih točaka, rubova i lica objektima potrebno je prijeći u način uređivanja, te odabrati željene točke, rubove i lica i na kraju pritisnuti prečac E kako bi se pojavili novi dijelovi objekta [8].

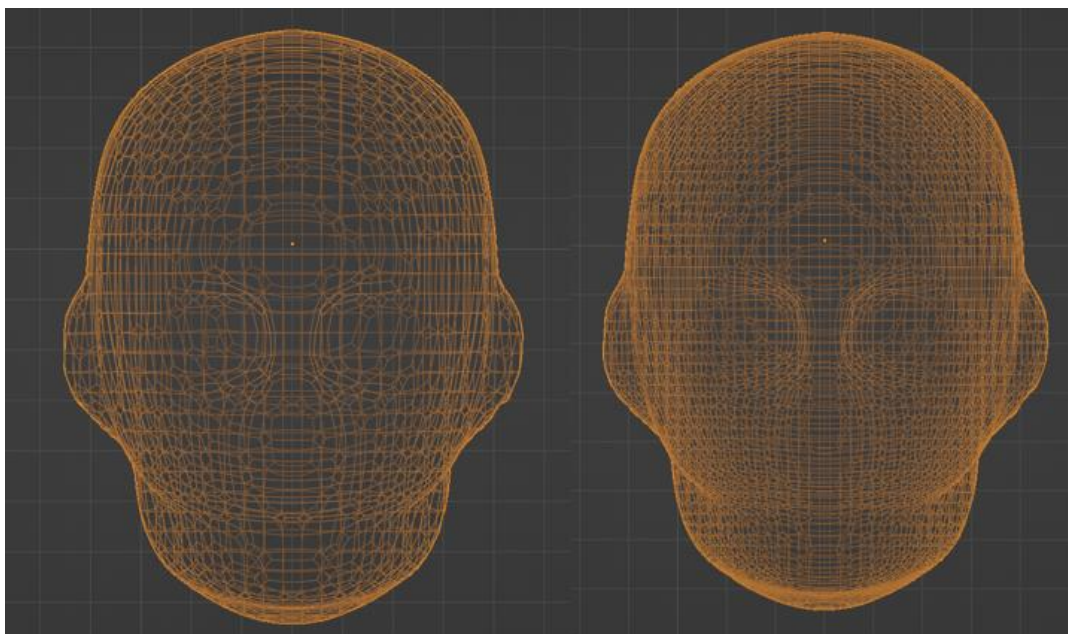


Slika 12. Početni oblik tijela čovjeka

Kada smo zadovoljni s početnim oblikom tijela, možemo odabrati pojedini dio tijela i prijeći u način kiparenja gdje možemo detaljnije urediti objekte. Običnim razvlačenjem postojeće mreže objekta vjerojatno će se ostati bez dovoljnog broja lica za uređivanje detalja, te je stoga potrebno povećati broj lica. Kako bi se povećao broj lica odabranog objekta, potrebno je primijeniti alat Remesh koji se nalazi u gornjem desnom kutu prozora 3D prikaza za kiparenje. Kako bi se povećala gustoća lica potrebno je smanjiti vrijednost Voxel Size i pritisnuti gumb Remesh [9].



Slika 13. Remesh alat

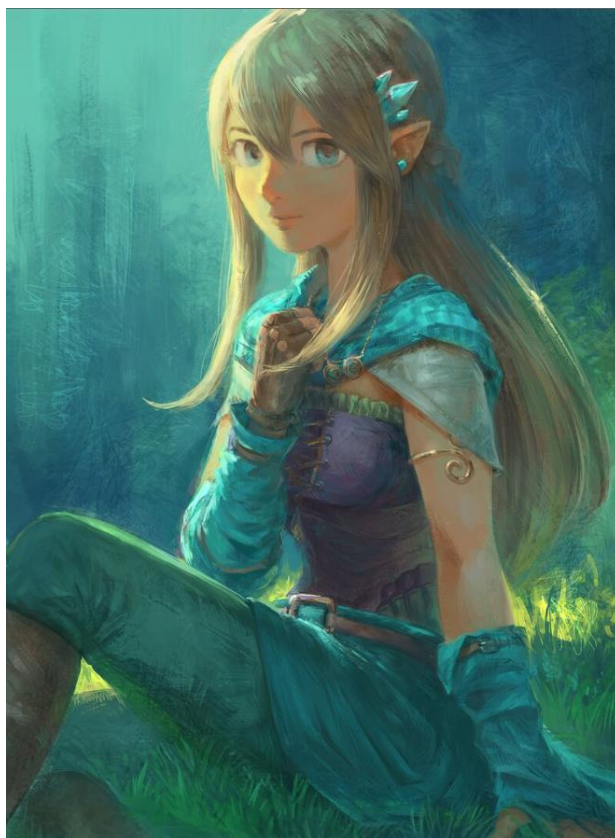


Slika 14. Dva objekta s različitom gustoćom poligona

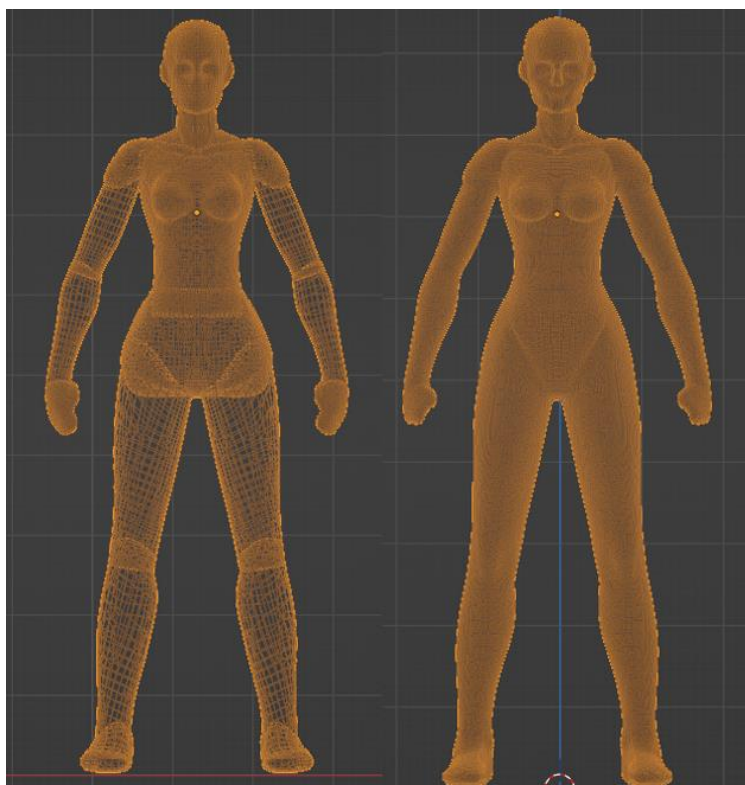
Kada se stvore detaljniji oblici udova potrebno ih je spojiti u zajednički objekt tako da ih se odabere i stisne prečac CTRL+J [9].

Nakon povezivanja objekata u jedan zajednički objekt, oni još uvijek nisu fizički spojeni, nego se međusobno preklapaju. Kako bi se uklonila preklapanja, odnosno stvorila jedinstvena mreža za sve objekte, koristi se prethodno spomenuti alat Remesh. Kako bi Remesh ispravno spojio sve objekte, svaki objekt mora biti potpuno zatvoren, odnosno vodonepropustan [9].

3D model koji se izrađuje u ovom završnom radu temelji se na konceptu lika Sare autora Davida Revoya koji je uredio dizajn originalnog autora Mandia Paugha [7].

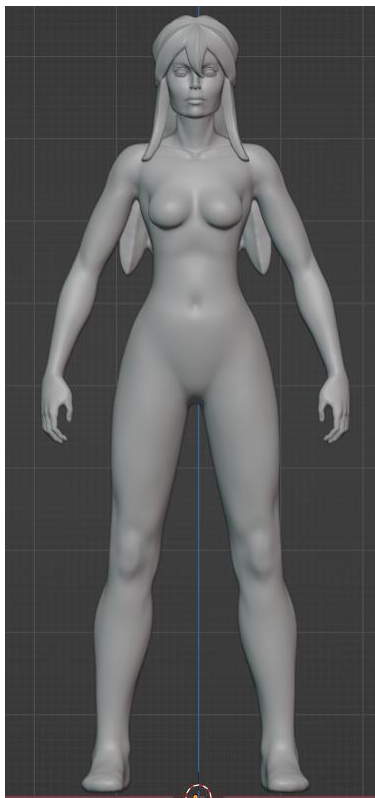


Slika 15. Koncept lika Sare (Izvor: David Revoy, 2022)



Slika 16. Objekti s različitim mrežama i spojeni objekti

Nakon što se sve spoji u jednu mrežu potrebno je nastaviti s oblikovanjem detalja i povećavanjem gustoće mreže do željene razine. Kosa će biti napravljena kao zasebni objekt također pomoću digitalnog kiparenja.



Slika 17. Završeni digitalni kip čovjeka

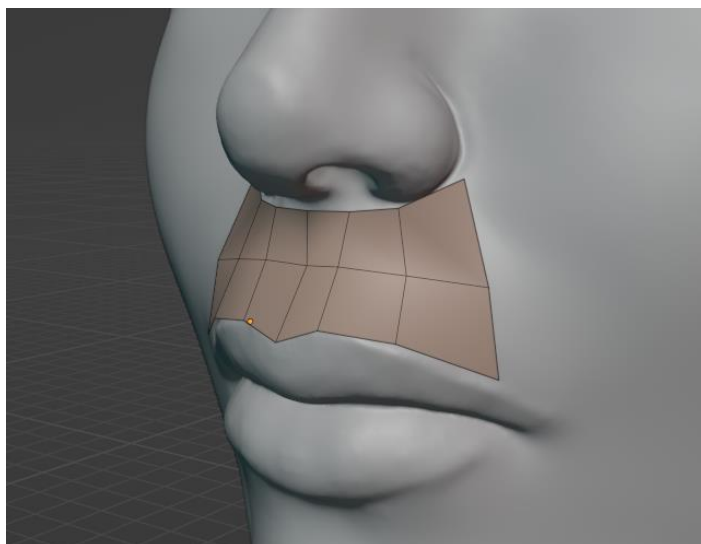
Konačni digitalni kip čovjeka ima 11 miliona lica, što znatno usporava rad računala, zauzima puno prostora memorije i otežava prelazak na fazu animiranja i korištenja u video igrama. Blender ima modifikator koji može smanjiti broj lica na željenu razinu, međutim ne nudi željeni oblik topologije koji pogoduje animiranju likova.

Tvorci jednog od najpoznatijih programa za razvoj igara – Unreal Engine 5, 2020. godine su predstavili revolucionarnu tehnologiju Nanite koja omogućava korištenje statičkih (nepokretnih) objekata s velikim brojem lica u video igrama [10].

7. Retopologija

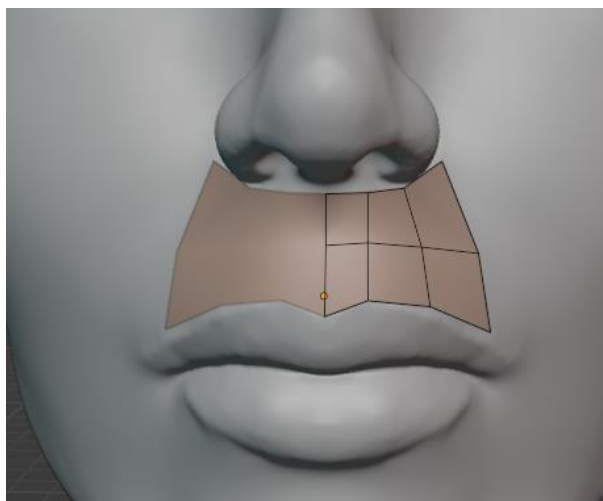
Retopologija je proces izrade jednostavnije i čišće topologije mreže iz objekta s velikim brojem poligona kako bi se omogućilo lakše daljnje korištenje objekta. Postoji mnogo programa koji mogu u potpunosti samostalno i brzo napraviti retopologiju objekta, međutim dobiveni rezultat nije dobar kao ručna retopologija. Ručna retopologija je većini 3D modelera vjerojatno najmanje drag dio 3D modeliranja jer je vrlo monoton.

U Blender-u postoje dva modifikatora koja ubrzavaju proces retopologije, a to su Skupljajuća folija (engl. Shrinkwrap) i Ogledalo (engl. Mirror). Skupljajuća folija omogućava lijepljenje lica nove mreže preko lica odabranog objekta. Na taj način nije potrebno voditi previše računa o lokaciji na kojoj se nalaze dijelovi mreže novog objekta [11].



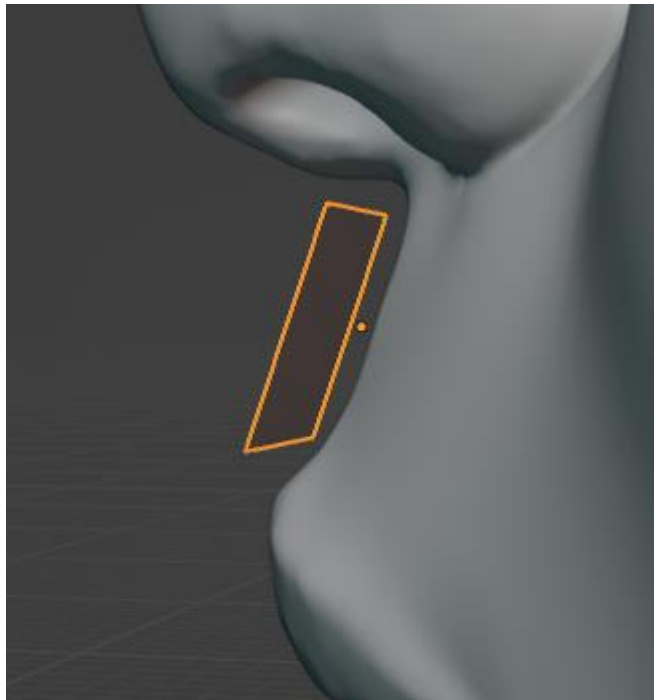
Slika 18. Zalijepljena lica novog objekta pomoću Skupljajuće folije

Ogledalo se koristi kada imamo objekt koji je simetričan po jednoj ili više osi. Stoga da se ne obavljaju iste radnje na simetričnom dijelu mreže, prethodno navedeni modifikator će zrcaliti mrežu po zadanim osima.



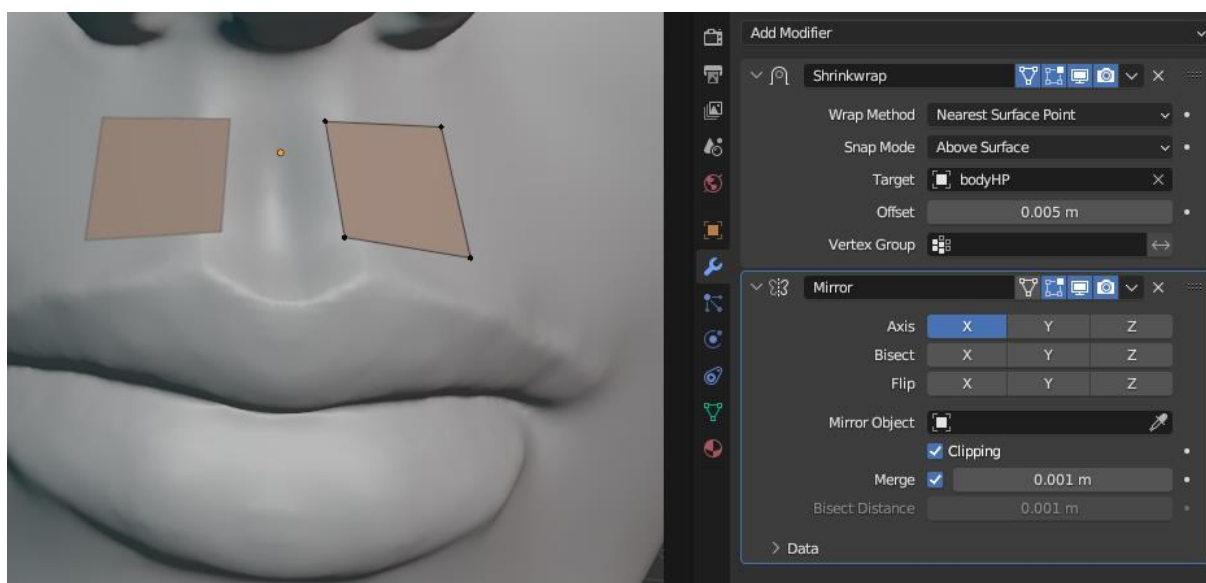
Slika 19. Zrcaljenje mreže objekta pomoću Ogledala

Prvi korak retopologije je stvaranje novog objekta, najčešće ravnine (engl. Plane). Lice ravnine je potrebno postaviti i rotirati što bliže dijelu digitalnog kipa na kojemu se želi započeti retopologiju. Ako se lice ne postavi pravilno, modifikator Skupljajuća folija bi mogao na krivi način zalijepiti lice.



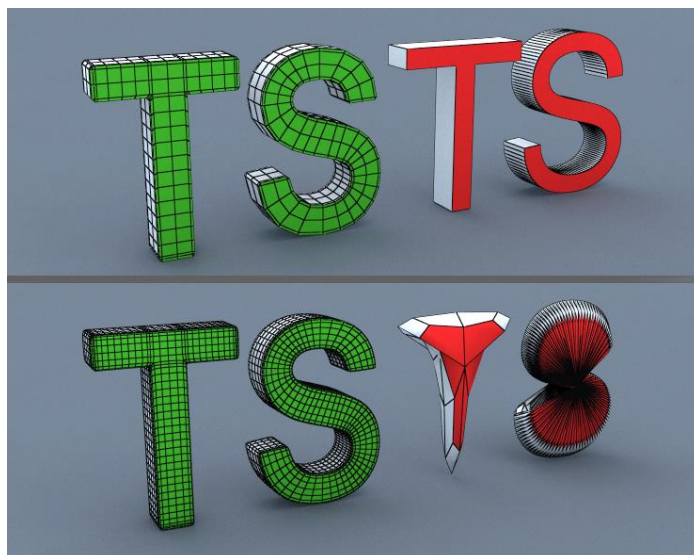
Slika 20. Objekt ravnina postavljen blizu digitalnog kipa

Kako bi se lice novog objekta zalijepilo na digitalni kip, potrebno je dodati modifikator Skupljajuća folija na novi objekt. Vrijednost svojstva Snap Mode potrebno je postaviti na Above surface kako bi se lice postavilo na vanjsku stranu površine, a kao vrijednost za svojstvo Target treba odabrati digitalni kip. Vrijednost Offset je u ovom slučaju postavljena na 0.005 m što znači da je lice udaljeno za 5 milimetara od površine digitalnog kipa. Poželjno je da lice ne prelazi os simetrije kako se ne bi pojavila preklapanja pri korištenju modifikatora Ogledalo. Zatim možemo dodati modifikator Ogledalo i postaviti mu vrijednost osi na X i uključiti opciju Clipping kako bi se pri prelasku preko osi simetrije vrhovi spojili.



Slika 21. Vrijednosti modifikatora Shrinkwrap i Mirror

Sljedeći korak je stvaranje novih lica i njihovo pozicioniranje na željena mjesta. Kako bi se oblikovala što bolja topologija, poželjno je na određena mjesta postaviti petlje lica i iz njih stvarati nove dijelove mreže. Dijelovi objekta koji će se tijekom animacije deformirati, trebali bi imati veću gustoću lica od ostatka tijela kako bi ti dijelovi što bolje zadržali oblik. Za objekte koji će se animirati bilo bi najbolje da su izrađeni od četverokutnih lica, međutim dijelovi tijela koji se neće previše deformirati mogu sadržavati i trokute. Poligoni koji imaju više od 4 vrha nisu poželjni u 3D modeliranju jer uništavaju tok poligona i uzrokuju probleme u renderiranju, glatkom sjenčanju i primjeni nekih modifikatora [12].



Slika 22. Primjer problema uzrokovanih korištenjem poligona s više od 4 vrha (Izvor: Turbosquid 3D Resources, 2023)

Proces retopologije završava kada se prekrije cijeli digitalni kip, odnosno novonastali objekt ima isti oblik kao i stari, ali s puno manje detalja. Nakon odrađene retopologije tijela, nova mreža tijela čovjeka ima 3182 lica, što je normalna vrijednost za 3D modele koji se koriste u video igrama. Kako bi se prenijeli detalji s digitalnog kipa na novi objekt, potrebno je izraditi texture na kojima će se ti detalji nalaziti. Za dodavanje glatkog sjenčanja objektu, potrebno je odabrati objekt i u gornjem lijevom kutu u izborniku Object odabrati opciju Shade Smooth.

Nakon odrađene retopologije tijela, potrebno je izraditi preostale modele odjevnih predmeta. Budući su ti objekti svojim oblikom slični obliku tijela na kojem se nalaze, moguće je iskoristiti dijelove mreže tijela kako bismo iz nje napravili mrežu odjeće. Potrebno je odabrati lica koja svojim oblikom reprezentiraju odjevni predmet i pritisnuti prečac CTRL+D koji će duplicirati odabrana lica. Nakon toga treba pritisnuti prečac P i odabrati Selection kako bi se odabrani dio pretvorio u zasebni objekt. Novonastali objekt potrebno je dodatno oblikovati i stvoriti još lica kako bi se objekt razlikovao od tijela čovjeka.

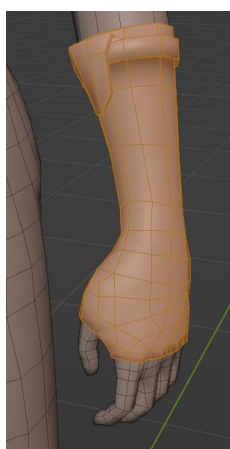
Koristan modifikator za postavljanje debljine objektima se naziva Solidify. Može se koristiti u slučaju kada objekt nema debljinu, nego mu se površina sastoji od jednog sloja lica. Postavljanjem Solidify modifikatora stvaraju se novi poligoni, čiju udaljenost od postojeće mreže možemo definirati u postavkama modifikatora.

Solidify modifikator može znatno otežati proces davanja utjecaja kostima jer se problem očituje u neželjenom preklapanju različitih objekata. Stoga je ponekad bolji način stvaranja prividnog volumena objekta, tako da se na rubovima objekta stvore dodatna lica koja

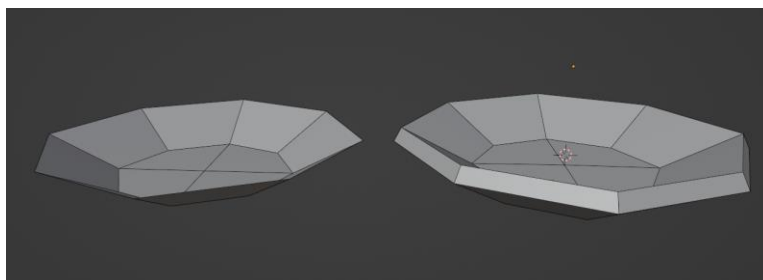
će zatvoriti prostor između objekata. Ako je korišten Solidify modifikator, poželjno je izbrisati unutarnja lica koja se ne vide.



Slika 23. Digitalni kip i rezultat retopologije



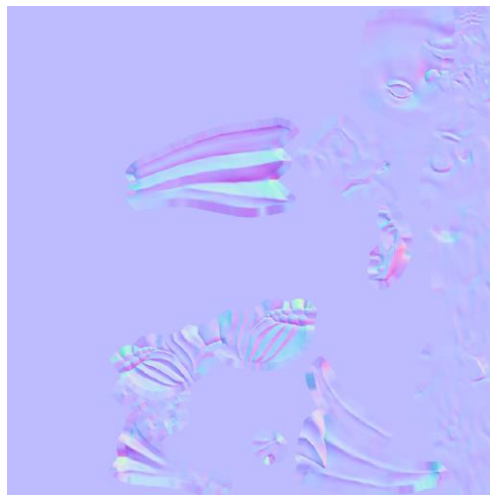
Slika 24. Rukavica nastala odvajanjem od mreže tijela



Slika 25. Primjer korištenja Solidify modifikatora

8. Teksturiranje

Teksturiranje je proces izrade tekstura, odnosno slika koje će predstavljati detalje na 3D modelu. Postoje razne vrste tekstura koje nude različite načine sjenčanja 3D modela. Neke od često korištenih vrsta su Diffuse map, Roughness map, Metallic map i Normal map. Diffuse map je tekstura koja omogućava prikaz boja 3D modela. Roughness map omogućava prikaz hrapavosti određene površine, odnosno kako se reflektira svjetlost sa površine. Metallic map govori je li objekt sačinjen od metala ili nije. Normal map je vrlo korisna tekstura koja omogućava lažiranje neravnina i udubljenja. Pomoću nje je moguće prikazati model s malim brojem poligona vizualno vrlo sličnim digitalnom kipu.

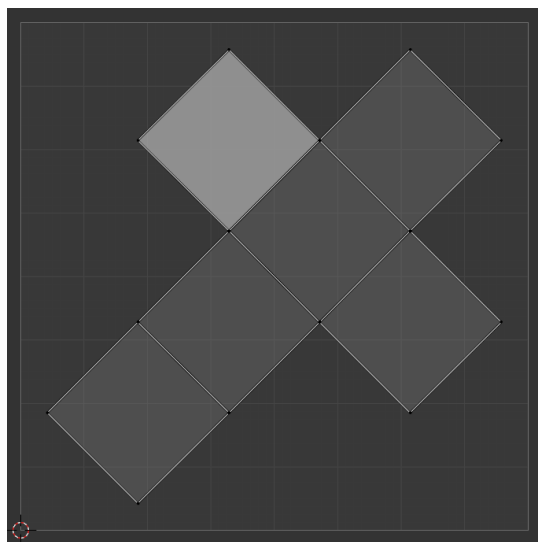


Slika 26. Primjer teksture Normal map



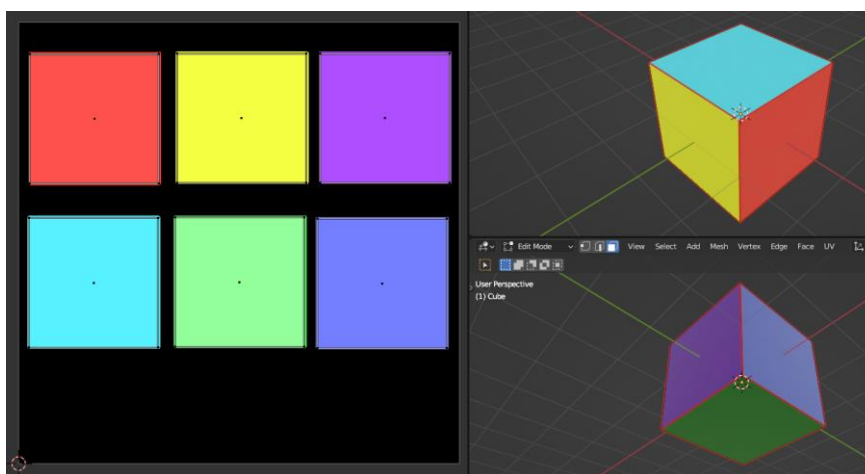
Slika 27. Digitalni kip i model s malim brojem poligona koji koristi Normal map

Prije izrade tekstura za 3D model koji će se koristiti u video igrama potrebno je stvoriti UV mapu (UV se odnosi na dvodimenzionalni prostor). UV mapa nastaje razmotavanjem 3D mreže i njezinim preslikavanjem u 2D prostor [13]. Na slici 28. se vidi prikaz UV mape jednostavnog geometrijskog tijela kocke.



Slika 28. Primjer UV mape kocke

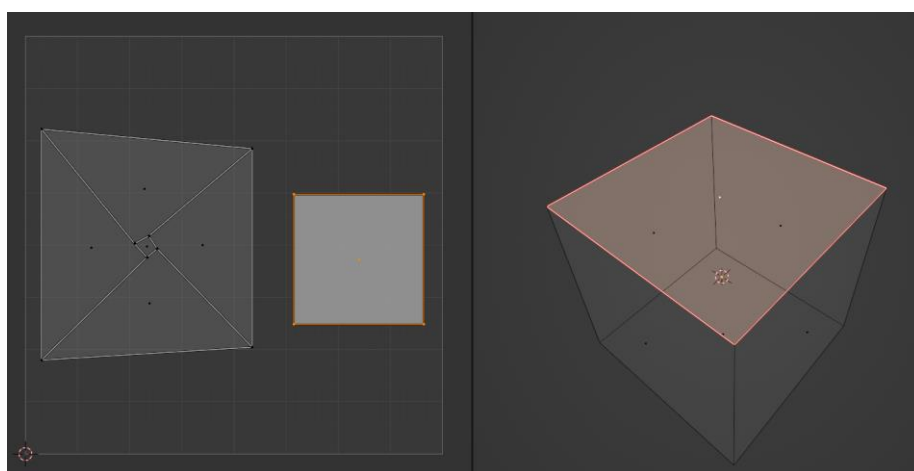
Razmotavanje mreže u 2D prostor je potrebno zbog toga što su teksture 2D slike koje će ukrašavati 3D model. UV mapa govori programu koji dio slike će se prikazati na kojem dijelu 3D modela. Složeniji 3D modeli se na UV mapi moraju podijeliti u više odvojenih dijelova koji se nazivaju UV otocima (engl. UV islands) kako ne bi došlo do neispravnog prikaza tekstura [13].



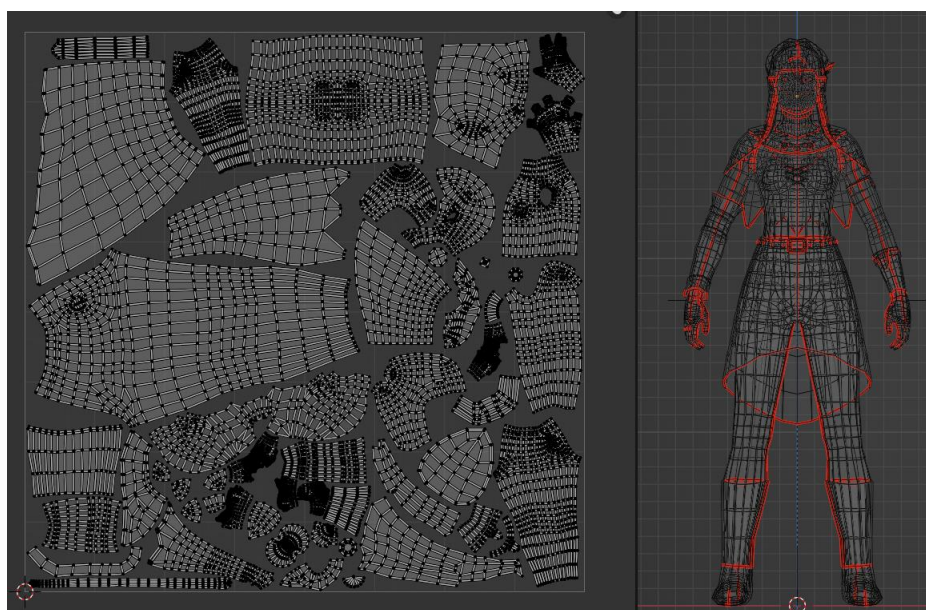
Slika 29. Svako lice kocke predstavlja zasebni UV otok

Kako bi se razmotala mreža u dijelove potrebno je prijeći u način uređivanja i odabrati rubove koji će predstavljati granice između dijelova mreže. Nakon što odaberemo rubove potrebno je pritisnuti prečac U i odabrati Mark Seam, što će označiti rubove crvenom bojom. Crveno označeni rubovi nazivaju se šavovima (engl. Seams) i predstavljaju granicu između dva lica koja će na UV mapi biti odvojena [13]. Ako se odabere petlja rubova, objekt se dijeli na nova dva UV otoka koja su odvojena crvenim rubovima (primjer na slici 30.).

Nakon što se postave svi šavovi, potrebno je pritisnuti prečac U i odabrati Unwrap kako bi se stvorila nova UV mapa na temelju odabranih šavova. UV mapu se može vidjeti u prozoru UV editor ili promjenom radnog prostora na UV Editing koji se nalazi na vrhu sučelja.



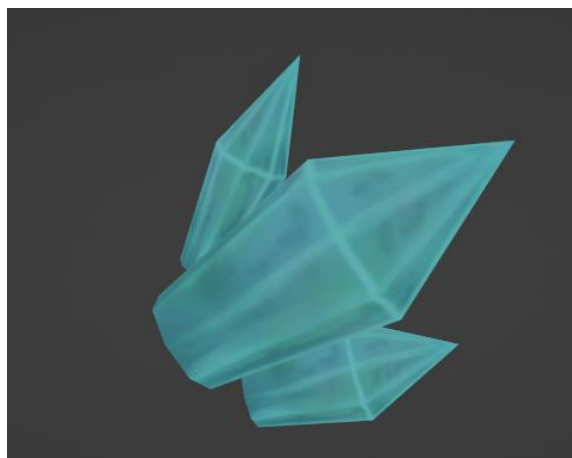
Slika 30. Dva UV otoka



Slika 31. UV mapa 3D modela čovjeka

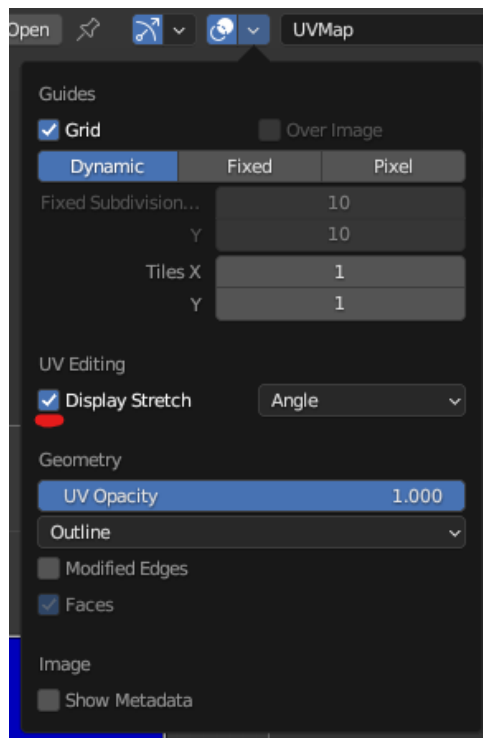
Za izradu kvalitetne UV mape potrebno je pripaziti na par stvari. Što više prostora pojedini otok zauzima na UV mapi to je veću razinu detalja moguće prikazati na tom prostoru 3D modela. Stoga je potrebno da otoci, odnosno dijelovi 3D modela koje igrač najviše vidi zauzimaju najveći prostor UV mape. UV otoci i njihova lica se ne bi smjeli preklapati, jer će time doći do prikaza jednog dijela teksture na više mjesta 3D modela.

U slučaju da se želi uštediti prostor na UV mapi, mogu se u potpunosti preklapati oni 3D otoci koji predstavljaju više instanci istog objekta. Na taj način će različiti objekti koji imaju isti oblik, biti prikazani istim teksturama. Primjer toga su kristali različitih veličina koji se nalaze u kosi, a imaju isti oblik mreže.



Slika 32. Kristali koji imaju zajedničke UV otoke

Još jedan od mogućih problema je rastezanje lica na UV mapi, a uzrok tome je loš odabir šavova. Rastegnuta lica na UV mapi će uzrokovati izobličeni prikaz teksture na 3D modelu. Za rješavanje takvog problema potrebno je promijeniti šavove ili dodati nove šavove, ako oni koji već postoje nisu dovoljni za pravilno razmatanje mreže. U Blender-u postoji opcija koja omogućava prikaz rastezanja lica na UV mapi pomoću boja. Kako bi se pokrenuo takav način prikaza potrebno je otvoriti UV editor i među opcijama na vrhu odabrati padajući izbornik Overlays i omogućiti Display Stretch.



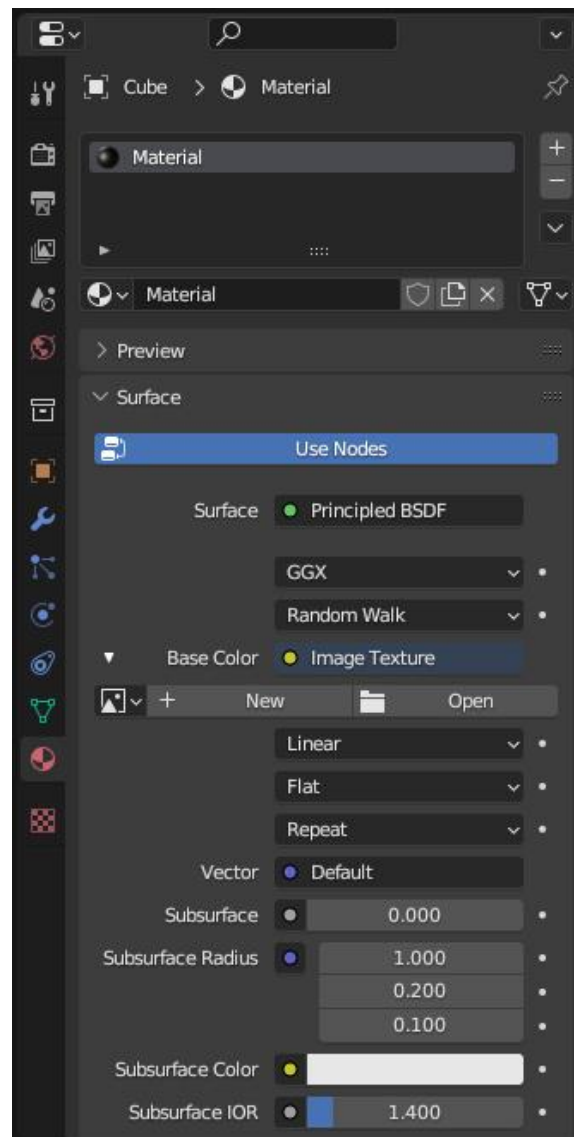
Slika 33. Izbornik Overlays

Nakon uključivanja opcije Display Stretch, lica na UV mapi će poprimiti boju između plave i crvene. Što je boja hladnija, znači da ima manje rastezanja, a ako je toplija znači da je rastezanje veće. Ako pomaknemo vrhove UV otoka obične kocke, na način da svojim oblikom više ne tvore kvadrat ili pravokutnik, boja lica će poprimiti toplije boje.



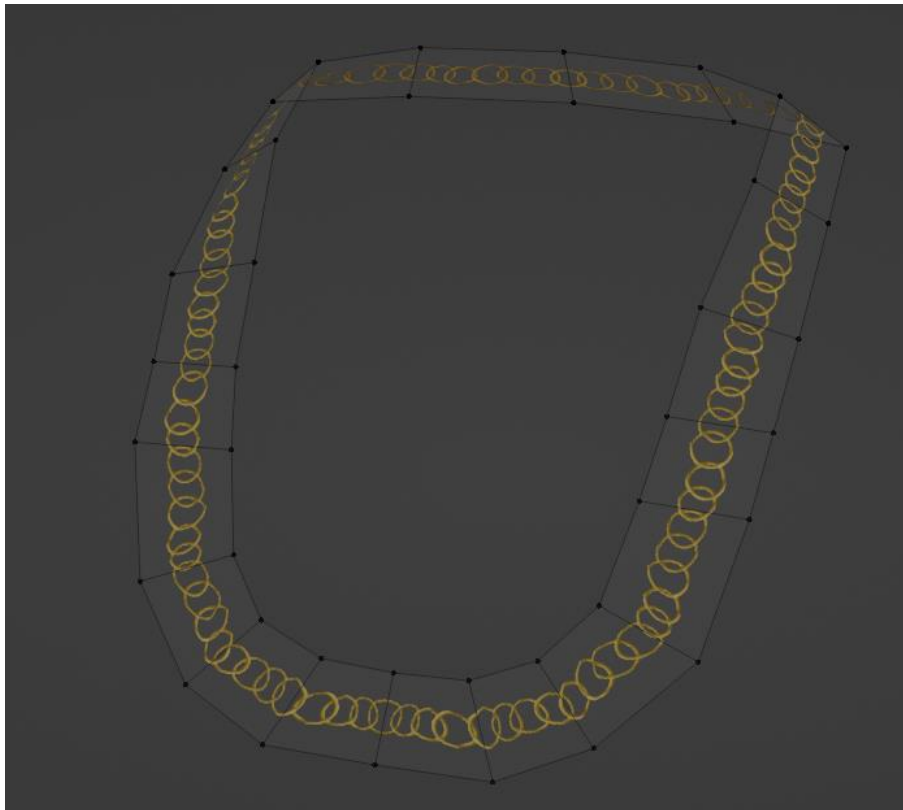
Slika 34. Rastegnuta UV lica kocke

Nakon završetka izrade UV mape, potrebno je stvoriti novu sliku i povezati je s 3D modelom. Kreiranje i povezivanje slike s modelom moguće je putem pristupanja izborniku Material koji se nalazi u prozoru Postavke. Kod padajućeg izbornika Surface potrebno je pritisnuti žuti kružić uz atribut Base Color i odabrati Image Texture.



Slika 35. Izbornik Material

Nakon odabira Image Texture objekt bi trebao poprimiti crnu boju (ako se nalazite u načinu za pregled materijala ili za renderiranje) zbog toga što nije odabrana ni jedna tekstura. Za kreiranje nove slike potrebno je pritisnuti gumb New i pripaziti da ima kvačica pokraj svojstva Alpha jer bez nje određeni dijelovi modela neće moći biti transparentni. Transparentnost se često koristi za smanjenje broja poligona kod objekata kao što su zastave, plaštev i razni objekti s većim brojem rupa [14].

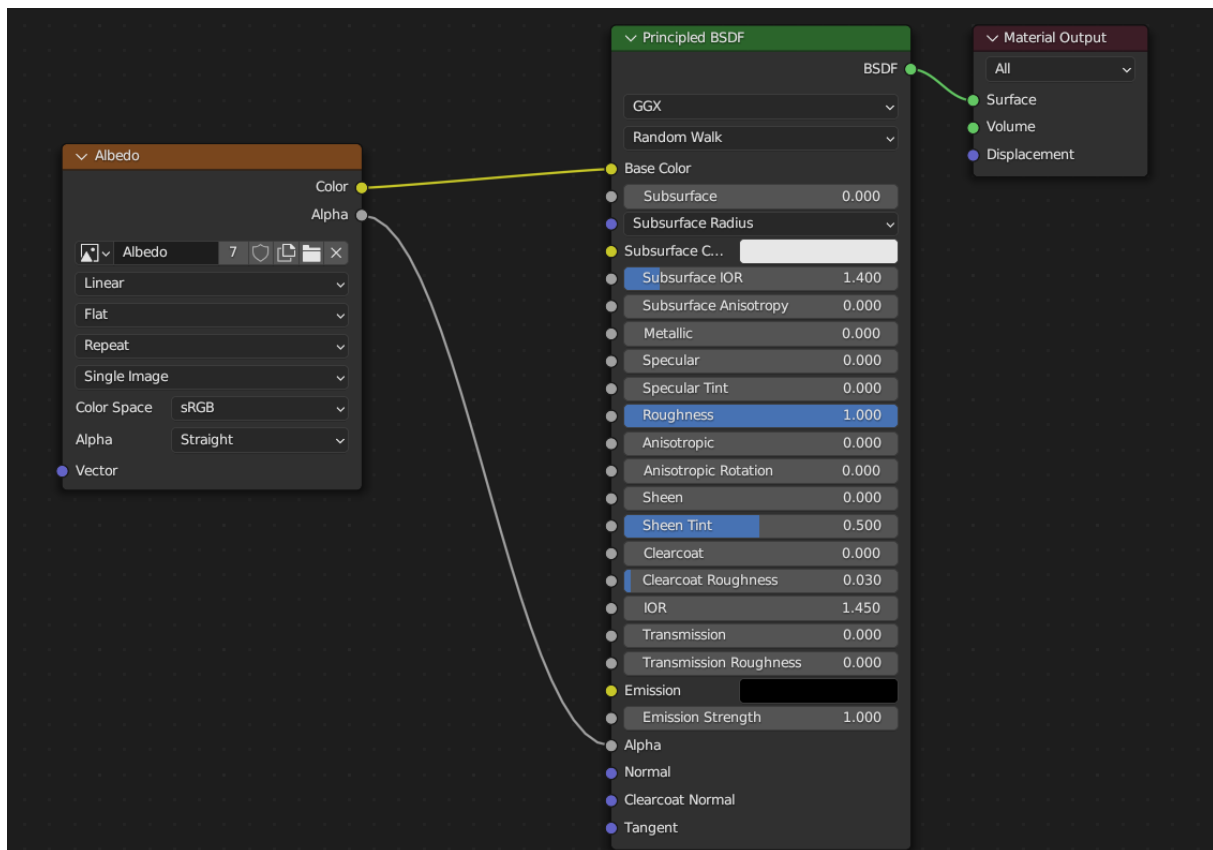


Slika 36. Transparentni objekt Lančić

Postavke materijala je moguće mijenjati i na pregledniji način putem Uređivača sjenčanja (engl. Shader Editor). Unutar prostora Uređivača sjenčanja mogu se koristiti razni čvorovi (engl. Nodes) koji će utjecati na prikaz materijala 3D modela.

Kako bi se stvorio novi čvor koji će sadržavati teksturu 3D modela, potrebno je pritisnuti prečac SHIFT+A i u tražilicu upisati Image Texture te povezati izlaze iz novonastalog čvora sa ulazima na Principled BSDF čvoru. Izlaz Color iz čvora slike se spaja sa ulazom Base Color čvora Principled BSDF, kako bi se prikazala boja 3D modela i izlaz Alpha sa ulazom Alpha za prikaz transparentnih dijelova.

U Uređivaču sjenčanja je također moguće izrađivati i proceduralne materijale koji svoj izgled mogu prilagođavati obliku 3D modela. Proceduralni materijali su korisni kada je potrebno izraditi teksture za velike površine poput mora i objekata čiji se oblik mijenja tijekom igranja igre. Za razliku od običnih tekstura koje se inače koriste, teksture proceduralnih materijala se generiraju tijekom igranja igre, te stoga troše i više računalnih resursa.



Slika 37. Čvorovi unutar Uređivača sjenčanja

Primjer prijenosa detalja sa digitalnog kipa na model s malim brojem poligona bit će prikazan putem stvaranja mape Ambijentalne okluzije (engl. Ambient Occlusion map) jer je ona korisna za izradu materijala čiji se svi detalji nalaze samo u jednoj ručno obojanoj slici.

Kako bi se prenijeli detalji potrebno je odraditi proces koji se zove pečenje tekstura (engl. Texture baking). Tijekom pečenja tekstura u Blender-u mogu se dogoditi razne greške koje će rezultirati neispravnim izgledom ispečene teksture. Najčešći problem s izgledom ispečene teksture se pojavljuje kada se preklapaju mreže digitalnog kipa i objekta s malim brojem poligona. Kako bi se te greške minimizirale korisno je napraviti objekt koji se zove Kavez (engl. Cage).

Kavez je duplikat modela s malim brojem poligona koji će svojim većim dimenzijama od originala prekriti i digitalni kip i objekt s malim brojem poligona [15]. Razlog korištenja kaveza je što se pomoću njega pravilno usmjeravaju zrake koje služe za hvatanje detalja sa digitalnog kipa. Kreiranje duplikata objekta s malim brojem poligona (kavez) se postiže pritiskom na prečac CTRL+D. Duplikat je potrebno skalirati na način da prekriva i digitalni kip i objekt s malim brojem poligona, ali je važno pripaziti da se pri skaliranju poligoni mreže ne

počnu preklapati. Preklapanje se obično pojavljuje na mjestima oko ušiju i između prstiju, te je stoga bolje te dijelove ručno skalirati pomicanjem poligona.

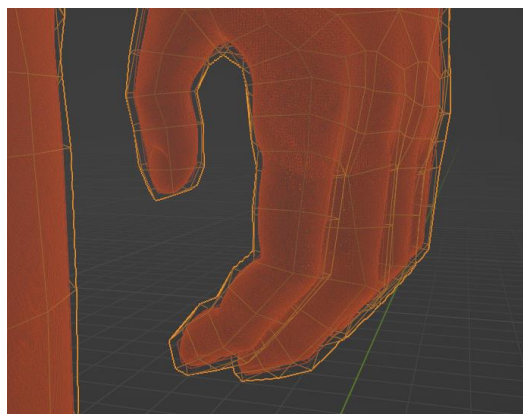
Za pečenje detalja prvo je potrebno odabrati čvor teksture unutar Uređivača sjenčanja na koju će se oni zapeći. Zatim treba odabrati digitalni kip i objekt s malim brojem poligona te u prozoru s postavkama odabrati izbornik Render. Potrebno je odabrati za Render Engine vrijednost Cycles i kod padajućeg izbornika Bake promijeniti vrijednost Bake Type na Ambient Occlusion, te staviti kvačice na Selected to Active i Cage. Za Cage Object je potrebno odabrati objekt koji se koristi kao kavez i pritisnuti gumb Bake. Ovisno o snazi računala potrebno je pričekati neko vrijeme dok Blender ne izradi ispečenu teksturu.

Mapa ambijentalne okluzije je tekstura koja se sastoji od crno bijelih nijansi koje predstavljaju sjene na objektima. Uobičajeno se miješaju s teksturama koje sadrže boje kako bi se istaknula trodimenzionalnost objekta [16]. Mapi ambijentalne okluzije se može dodijeliti boja kože korištenjem nekog od programa za obradu slike ili u Blenderu pomoću čvora koji se zove Miješanje (engl. Mix).

Ručno bojanje 3D modela se odvija unutar 3D prikaza pomoću načina rada za bojanje tekstura (engl. Texture Paint). Proces je relativno jednostavan, međutim zahtjeva dobro znanje iz područja umjetnosti poput teorije boja, kompozicije i perspektive. Za bojanje određene teksture potrebno je odabrati njezin čvor unutar uređivača sjenčanja. Zatim je potrebno u objektnom načinu rada odabrati željeni objekt i preći u način rada za bojanje tekstura.

Unutar načina rada za bojanje tekstura postoje 6 korisnih alata za bojanje, a to su kist za crtanje, kist za ublažavanje, kist za razvlačenje boje, kist za kloniranje, kist za ispunu objekta i kist za crtanje oblika pomoću maske.

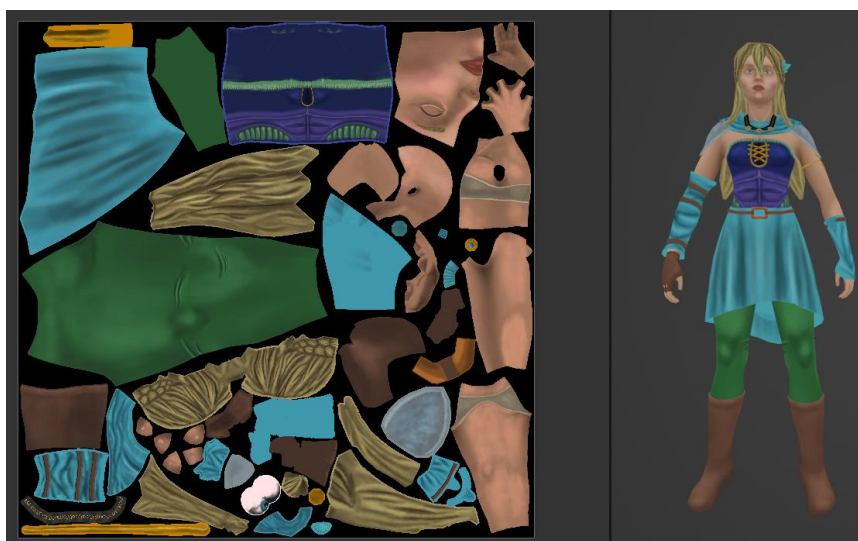
Lančić ogrlice je napravljen u programu za obradu slike Krita, na način da je pozadina lančića transparentna, odnosno alfa kanal je postavljen na 0. Slika je prenesena u Blender, a objektu koji predstavlja ogrlicu dodijeljena je druga UV mapa na kojoj su lica postavljena na način da se ne prepoznaje uzorak slike. Zatim je prikaz teksture te UV mape zapečen na zajedničku teksturu svih objekata.



Slika 38. Kavez za pečenje tekstura (žuta boja)



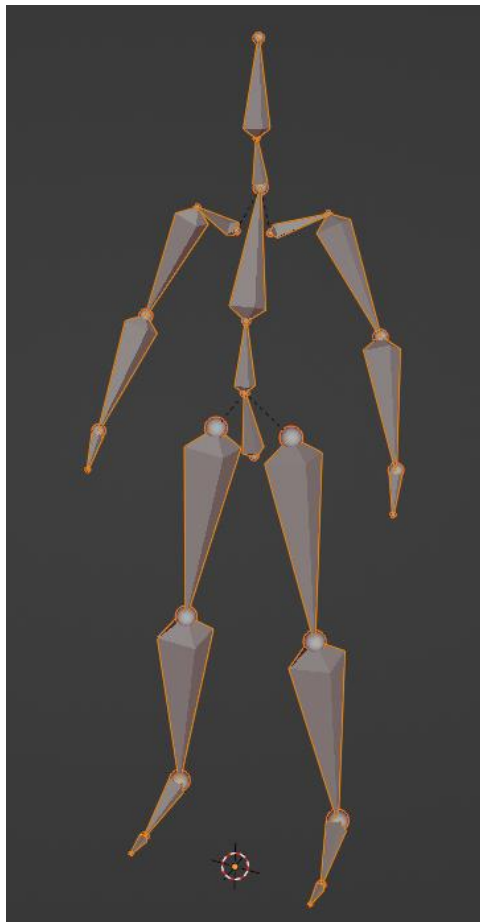
Slika 39. Mapa ambijentalne okluzije



Slika 40. Konačna tekstura 3D modela

9. Animiranje

Kako bi se 3D model mogao postaviti u određene poze potrebno je izraditi kosti i odrediti njihov utjecaj na 3D model (engl. Rigging). Za kreiranje kostiju potrebno je u objektnom načinu rada pritisnuti prečac CTRL+A i odabrati Armature > Single Bone. Nastalu kost je potrebno pozicionirati što bliže dijelu 3D modela na koji bi ona trebala utjecati. Kako bi kost uvijek bila vidljiva unutar objekata potrebno je u prozoru s postavkama uključiti opciju In Front koja se nalazi unutar izbornika Object pod padajućim izbornikom Viewport Display. Za stvaranje ostalih kostiju potrebno je prijeći u način rada za uređivanje i pritisnuti prečac E koji će iz odabrane kosti omogućiti izvlačenje nove kosti. Ako želimo odvojiti kost od njezinog roditelja potrebno je pritisnuti prečac ALT+P i odabrati Disconnect Bone. U slučaju da se želi postići nezavisnost kosti od njezinog roditelja, odnosno da kost ne prati lokaciju i rotaciju roditelja potrebno je pritisnuti prečac ALT+P i odabrati Clear Parent [17].

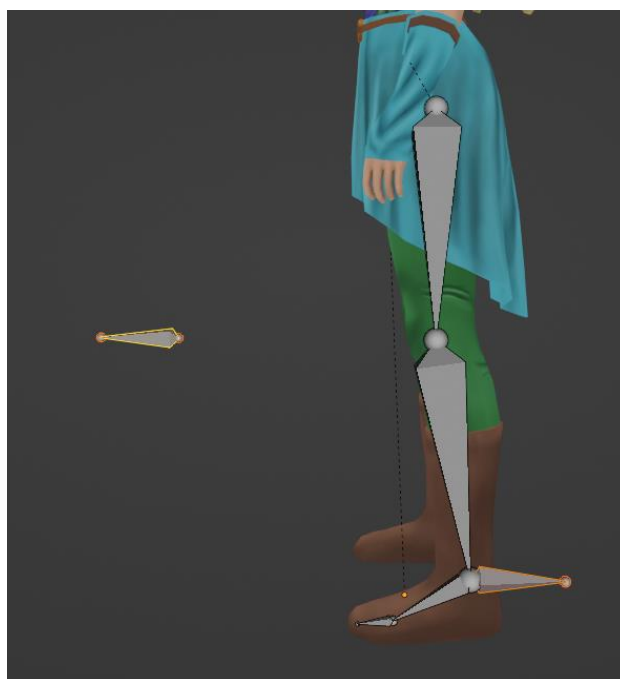


Slika 41. Kosti koje utječu na 3D model

Trenutni kostur i tijelo je moguće pomicati jedino koristeći princip direktne kinematike (engl. Forward Kinematics), odnosno ako se pomakne neka kost, njezine pomake i rotaciju prate i kosti koje se nastavljaju na nju. Ako se želi pozicionirati nekoliko povezanih kostiju, uobičajeno je da se s pozicioniranje kreće od početne kosti prema krajnjoj. Takav način animiranja je vrlo jednostavan i može se koristiti kod jednostavnijih kostura i animacija. Međutim kada postoji veći broj kostiju i složenije animacije, bilo bi potrebno svaku kost posebno pozicionirati i rotirati tijekom slaganja pojedine poze što oduzima puno vremena. Kod složenijih kostura, korisno je koristiti princip inverzne kinematike koji će biti objašnjen u nastavku.

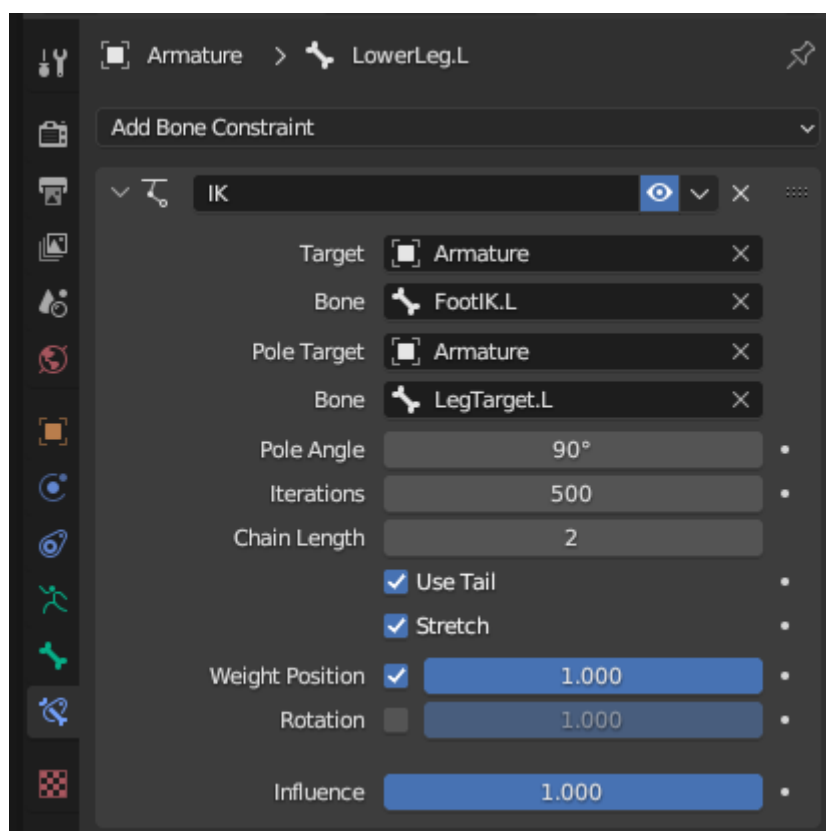
9.1. Inverzna kinematika

Za razliku od direktne kinematike, kod inverzne kinematike (engl. Inverse Kinematics) za pomicanje nekoliko povezanih kostiju, odabire se samo zadnja kost i njezinim pozicioniranjem Blender će pomaknuti ostale kosti na smislen način. Na taj način animator može pozicionirati 3D model poput lutke na koncu. Kako bi inverzna kinematika u Blender-u bila moguća potrebno je stvoriti još par kostiju koje neće imati utjecaj na 3D model, već na pomicanje samih kostiju. Za pomicanje nogu i ruku pomoću inverzne kinematike potrebno je za svaki ud stvoriti dodatne dvije kosti. Jedna kost nalaziti će se na početku kosti šake i stopala, a njezinim pomicanjem pomicati će se i kosti udova. Druga kost nalazi se na području laktova i koljena, a njezinim pomicanjem rotirat će se ruke i noge ovisno o položaju te kosti [17].



Slika 42. Kosti za inverznu kinematiku

Kako nove kosti ne bi utjecale na geometriju 3D modela, potrebno ugasiti opciju Deform koja se nalazi u izborniku Bone unutar prozora s postavkama. Zatim je potrebno dodati ograničenje za kosti, na način da se pređe u način za poziranje kosti (engl. Pose Mode) i odabere zadnja kost na koju bi trebala utjecati inverzna kinematika. Pod izbornikom Bone Constraints potrebno je dodati ograničenje Inverse Kinematics i ispuniti attribute koji se u njemu nalaze. Za attribute Target i Pole Target potrebno je staviti objekt koji sadrži kosti, za atribut Bone treba staviti kost koja će pomicati ostale kosti, a za drugi po redu atribut Bone treba staviti kost koja će rotirati ostale kosti. Pod atribut Chain Length potrebno je staviti broj povezanih kostiju na koje će inverzna kinematika utjecati. U slučaju da se koljeno ne savija prema kosti za rotaciju, potrebno je postaviti određeni kut na atribut Pole Angle koji će ispraviti taj problem [17].



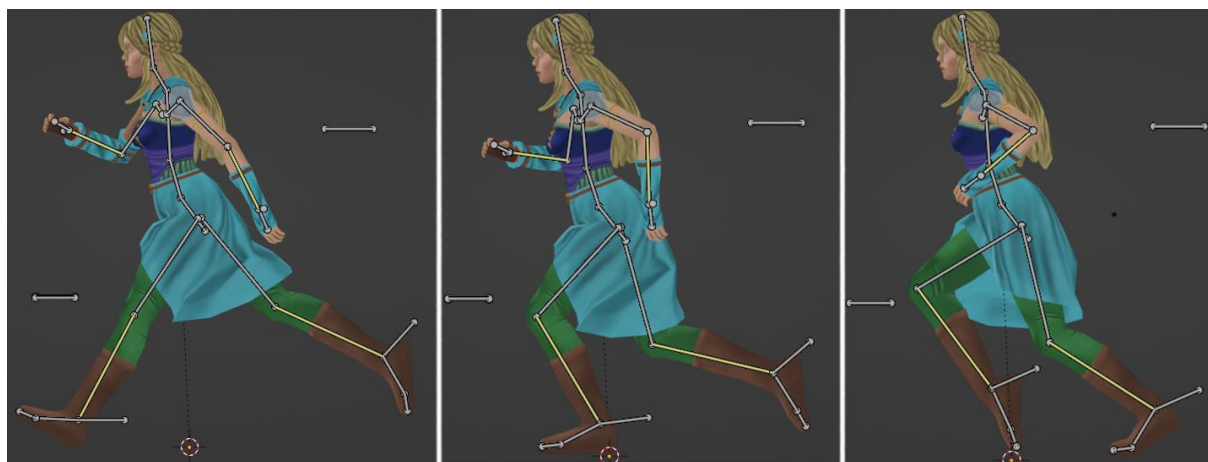
Slika 43. Ograničenje kostiju Inverzna kinematika

Kako se ne bi moralo dvaput raditi istu stvar za obje strane udova, Blender ima opciju za kopiranje kostiju prema osi simetrije. Stoga nakon što se izradi jedna strana kostura, moguće je odabrati te kosti i u načinu za uređivanje pritisnuti desnu tipku miša i odabrati opciju

Symmetrize. Kostima je prije korištenja te opcije potrebno dodijeliti na kraj njihovog imena sufiks .L (engl. Left) ili .R (engl. Right) ovisno o kojoj strani se radi.

Nakon što je cijeli kostur gotov potrebno je dodijeliti utjecaj kostima na 3D model. Kako bi se dodijelio utjecaj kostima potrebno je preći u objektni način rada, te odabrati prvo 3D model pa kosti i pritisnuti prečac CTRL+P i odabrati With Automatic Weights. Na ovaj način Blender će automatski dodijeliti utjecaj kostima, ali s malim greškama. Često se pojave greške na mjestima gdje su dva dijela tijela međusobno blizu (poput ruke i boka) te će kost ruke svojim pomicanjem utjecati i na bok. Takve greške se mogu popraviti uklanjanjem nevidljivih dijelova objekta i u načinu rada za bojanje težine (engl. Weight Paint) u koji se prelazi nakon što se prvo odabere objekt s kostima, a zatim 3D model. Prikaz utjecaja pojedine kosti se postiže pritiskom CTRL + lijeve tipke miša na određenu kost. Odabirom neke kosti dio 3D modela oko nje bi trebao dobiti topliju boju što znači da ta kost više utječe na taj dio. Kako bi se uklonio utjecaj koji kost ruke ima na bok trupa, potrebno je odabrati u prozoru s postavkama pod izbornikom Tools, kist Subtract i obojati taj dio boka u tamno plavo. Ako su svim kostima ispravno dodijeljeni utjecaji na dijelove 3D modela, može se preći na proces animiranja.

Animacije se sastoje od određenog broja sličica koje prikazuju objekt u različitim pozama. U Blender-u postoji prozor za rad s sličicama koji se zove Dope Sheet. Pomoću njega se može mijenjati trenutni okvir i spremati promjene poza u sličicama [18]. Animacije mogu imati velik broj sličica, a ručno pozicioniranje 3D modela za svaku sličicu bi oduzimalo jako puno vremena. Stoga se koriste ključni okviri (engl. Key Frames) koji omogućuju postavljanje ključnih poza, a Blender će u okvirima koji se nalaze između njih napraviti tranziciju između tih poza. Kako bi se dodatno uredile animacije, moguće je promijeniti neke okvire koji se nalaze u prostoru tranzicije i na njihovo mjesto ubaciti nove ključne okvire.



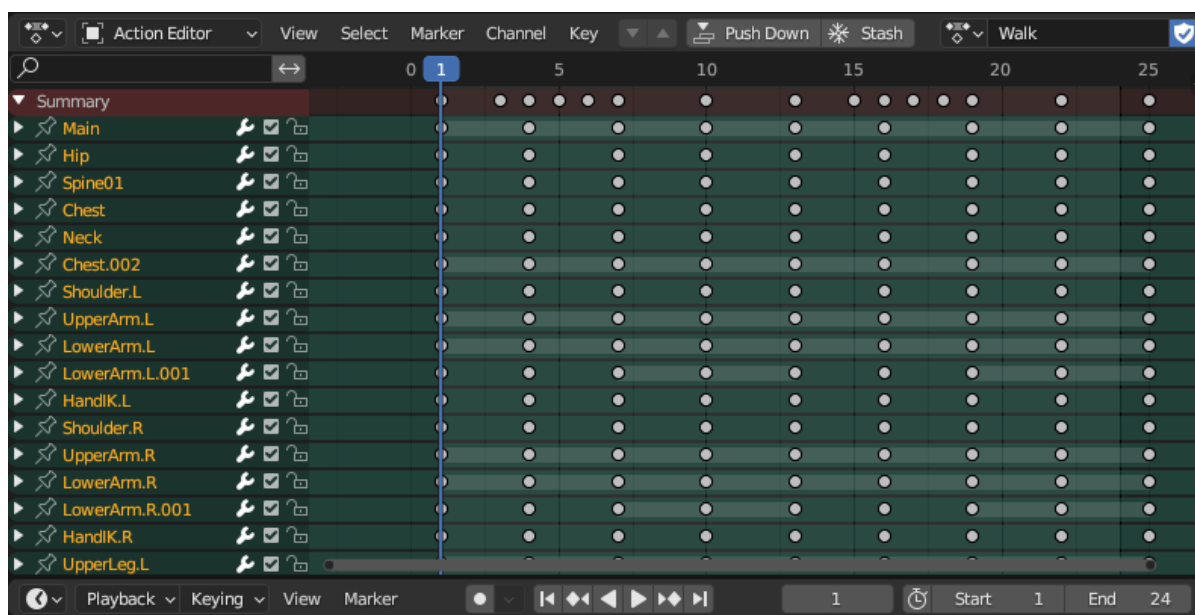
Slika 44. 3 ključna okvira iz animacije trčanja

Za stvaranje nove animacije potrebno je unutar Dope Sheet prozora pritisnuti gumb New i dati naziv animaciji.

Ako se izrađuje više animacija, korisno je pritisnuti gumb pokraj naslova animacije koji se zove Fake User kako se animacije koje trenutno nisu učitane ne bi izbrisale pri izlasku iz programa.

Radi jednostavnijeg procesa animiranja korisno je imati referentne slike objekta iz stvarnog svijeta koji izvršava aktivnost koju će se animirati. Potrebno je napraviti pozu 3D modela, te zatim odabrati sve kosti, pritisnuti prečac I i odabrati Location & Rotation kako bi spremili lokaciju i rotaciju kostiju u ključni okvir. Zatim je potrebno prijeći na neki drugi okvir unutar Dope Sheet prozora i stvoriti novi ključni okvir s drugom pozom.

Ako je nastala animacija repetitivna poput hodanja i trčanja, potrebno je postaviti vrijednost atributa End (u donjem desnom kutu Dope Sheet prozora) na broj za jedan manji od zadnjeg okvira (budući su prvi i zadnji okvir jednaki). Animacije se pokreću pritiskom tipke Space ili na gumb Play Animation u prozoru Dope Sheet.

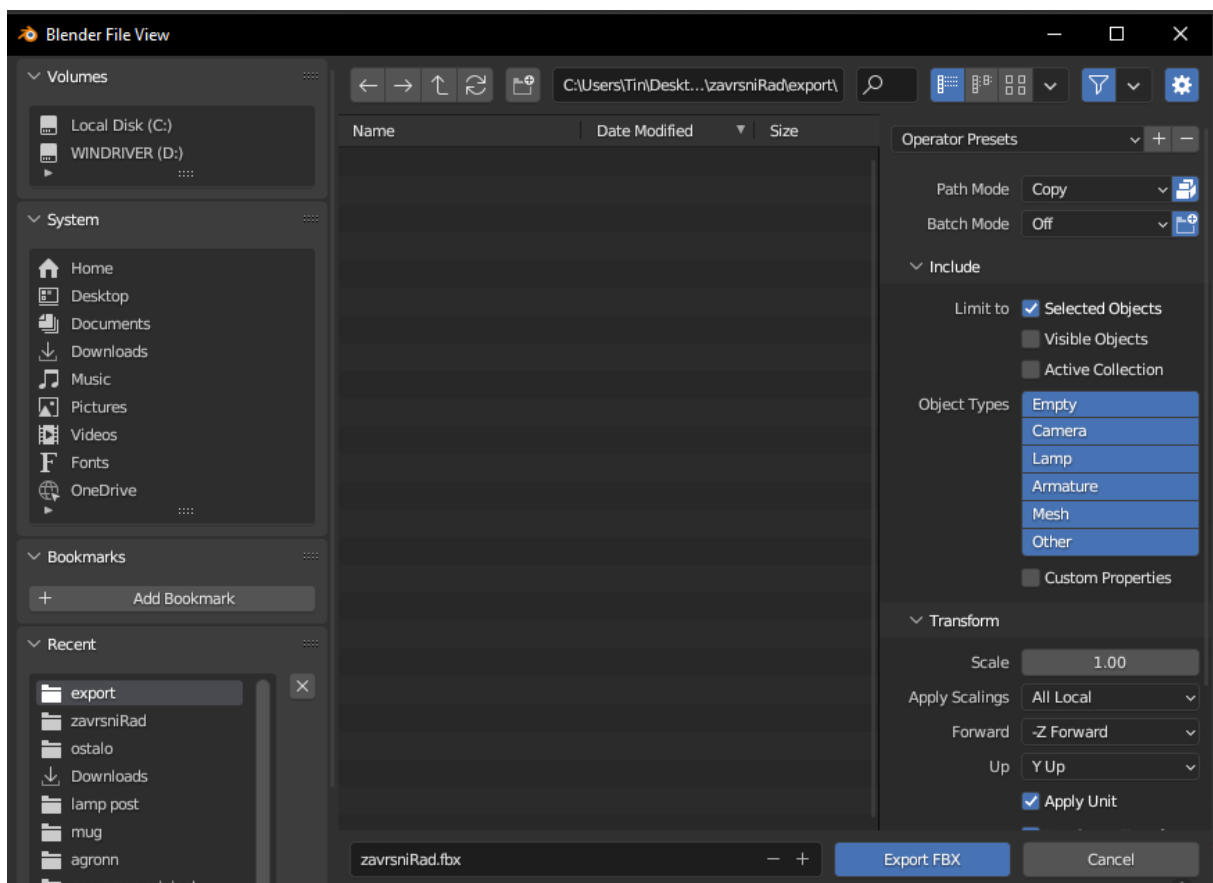


Slika 45. Dope Sheet prozor s animacijom za hodanje

10. Primjena 3D modela u Unity

Najjednostavniji način prenošenja 3D modela i animacija u Unity je na način da se prebaci .blend dokument u Unity-ev Project prozor. Međutim taj način se rijetko koristi jer .blend dokument zauzima više memorijskog prostora od uobičajenog .obj ili .fbx formata. Jednostavan format .obj je koristan za izvoz 3D modela (bez animacija i tekstura), dok je .fbx nešto složeniji i može izvesti 3D modele, animacije, teksture i razne druge podatke. Budući ovaj 3D model sadrži i animacije, u ovom projektu se za izvoz koristi .fbx format.

Za izvoz u Blender-u potrebno je odabrati objekte koje se želi izvesti, a zatim u izborniku odabrati File > Export > FBX (.fbx). U prozoru za izvoz potrebno je odabrati vrijednost Copy za Path Mode i pritisnuti gumb Embedded Textures koji se nalazi pokraj kako bi se spremile teksture u izvezenu datoteku. U slučaju da se želi izvesti samo odabrane objekte, potrebno je uključiti opciju Selected Objects.

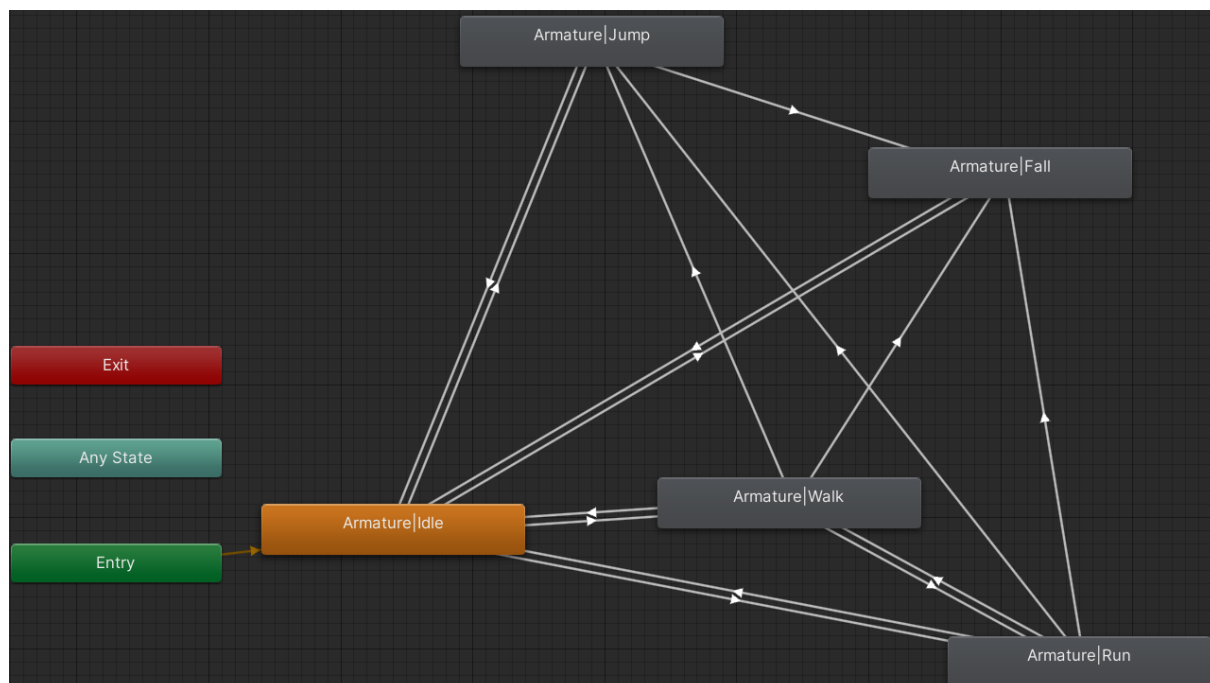


Slika 46. Prozor za izvoz 3D modela u Blender-u

Uvoženje objekta u Unity se može odraditi prevlačenjem i ispuštanjem .fbx dokumenta u prozor Project. Materijal uvezenog objekta na početku nema pridruženu teksturu s bojama, te je stoga potrebno u prozoru Inspector prijeći na izbornik Materials i pritisnuti gumb Extract Materials. Pritiskom prethodno navedenog gumba, u prozoru Project stvorit će se materijal i tekstura kao zasebni objekti pridruženi 3D modelu. Na materijalu je potrebno promijeniti svojstvo Rendering Mode na vrijednost Cutout kako bi se ispravno prikazali transparentni dijelovi.

Kako bi Unity znao koje animacije treba izvoditi u petlji potrebno je odabrati 3D model i u prozoru Inspector pod izbornikom Animations za svaku animaciju odabrati treba li opcija Loop Time biti uključena ili isključena. U ovom slučaju samo za animaciju skakanja treba isključiti tu opciju jer se ona izvodi samo jedanput kada se skoči.

3D model je potrebno ubaciti u scenu i dodijeliti mu komponentu Animator kako bi se mogla odrediti pravila izmjene animacija. Unutar animatora početno stanje je mirovanje, a iz njega se može prijeći u skakanje, hodanje, trčanje i padanje. Iz skakanja se može prijeći u mirovanje i padanje, a iz padanja u mirovanje. Iz hodanja se može prijeći u mirovanje, skakanje, trčanje i padanje. Iz trčanja se može prijeći u mirovanje, skakanje, hodanje i padanje.



Slika 47. Prikaz stanja u animatoru

Svakoj od veza unutar animatora je potrebno dodijeliti uvjete prelaska koji se ostvaruju pomoću parametara. Parametre je moguće stvoriti prelaskom na izbornik Parameters koji se nalazi u gornjem lijevom kutu prozora Animator i pritiskom na gumb +. Za ovaj projekt su stvorena 4 parametra, od kojih su tri tipa Bool, a jedan je Trigger. Bool parametri se koriste za mirovanje, hodanje, trčanje i padanje budući su to akcije koje nemaju točno određeno vrijeme trajanja. Trigger se koristi za akciju skakanja, jer se animacija skakanja može izvesti samo jedanput pri pritisku tipke Space. Razlika između parametra Bool i Trigger je u tome što se za Bool parametar mora u programskom kodu postaviti vrijednost istinitosti i laži, dok se kod Trigger-a nakon postavljanja istinitosti, vrijednost vraća u laž.

Kako bi se igrač mogao kretati po sceni i raditi određene akcije potrebno je stvoriti skriptu i dodijeliti je objektu igrača u sceni. Unutar skripte prvo se nalaze deklaracije varijabli koje će se kasnije koristiti. Kako bi se varijablama mogla postavljati vrijednost u prozoru Inspector potrebno ih je deklarirati kao javne, odnosno ispred njih napisati public.

Deklariranje varijabli:

```
//treba nam animator za pristup parametrima
public Animator playerAnimator;
//komponenta igraca kojom se liku daje sila za kretanje
public CharacterController controller;
//objekt start se odnosi na pocetnu poziciju igre
public GameObject start;
//objekt kanvas za prikaz teksta zavrsetka igre
public GameObject text;
//brzina hodanja
public float walkSpeed = 3f;
//brzina trcanja
public float runSpeed = 6f;
//vrijeme potrebno za okretanje
public float turnSmoothTime = 0.1f;
//varijabla u koju ce se spremiti trenutna brzina potrebna za
ugladiivanje rotacije igraca
float turnSmoothVelocity;
//brzina padanja objekta (gravitacija)
Vector3 velocity;
//vrijednost sile gravitacije
public float gravity = -9.81f;
//objekt (kugla) kojim se provjerava je li igrac u kontaktu s podlogom
public Transform groundCheck;
```

```

//radius kugle kojom se provjerava doticaj igraca s podlogom
public float groundDistance = 0.4f;
//sloj, odnosno svi objekti koji se smatraju podlogom
public LayerMask groundMask;
//istinito ako je igrač u doticaju s podlogom
bool isGrounded;
//visina skoka
public float jumpHeight = 3f;

```

Kod u nastavku se nalazi unutar metode Update koja se izvršava tijekom izmjene svake sličice. U varijablu `isGrounded` se sprema vrijednost istina, ako je metoda `Physics.CheckSphere` detektirala koliziju stvorene kugle i podloge. `groundCheck.position` i `groundDistance` definiraju lokaciju i radijus kugle kojom se provjerava kolizija. Varijabla `groundMask` definira s kojim slojem treba doći do kolizije. Ako je igrač pritisnuo tipku Space i ako se nalazi na tlu, lik će izvesti animaciju skoka i skočiti na prethodno definiranu visinu [19].

```

isGrounded = Physics.CheckSphere(groundCheck.position, groundDistance,
groundMask);
if(Input.GetButtonDown("Jump") && isGrounded){
    playerAnimator.SetTrigger("jump");
    velocity.y = Mathf.Sqrt(jumpHeight * -2 * gravity);
}

```

Zbog stalnog djelovanja sile teže na igrača, sila bi neprekidno rasla, te je stoga potrebno postaviti silu na malu vrijednost u slučaju da igrač stoji na podlozi [19]. Parametar koji govori je li igrač u zraku je postavljen na laž kako se ne bi izvodila animacija padanja.

```

if(isGrounded && velocity.y < 0){
    velocity.y = -2f;
    playerAnimator.SetBool("inAir", false);
}

```

U slučaju da je igrač u zraku postavlja se parametar za hodaње i trčanje na laž, a parametar za padanje se postavlja na istinu, čime se onemogućuje izvođenje animacije za hodaње i trčanje dok je igrač u zraku.

```

if (!isGrounded) {
    playerAnimator.SetBool("isWalking", false);
    playerAnimator.SetBool("isRunning", false);
    playerAnimator.SetBool("inAir", true);}

```

Varijable horizontal i vertical u sebi sadrže vrijednosti -1 ili 1, ovisno o tome jesu li pritisnute tipke za kretanje lijevo ili desno, naprijed ili nazad. Vektor direction sadrži vrijednost za 3 koordinatne osi kojim je definiran smjer kretanja igrača. Kako bi se očuvala stalna brzina kretanja, vektoru je potrebno dodati nastavak .normalized. Nastavak .normalized vektoru postavlja duljinu na vrijednost 1.

Parametar y vektora velocity odnosi se na vertikalnu os na kojoj djeluje gravitacija. Varijabla Time.deltaTime sadrži vrijeme izmjene dvije uzastopne sličice u sekundama i omogućava da promjena u brzini izmjene sličica ne utječe na konzistentnost igre. Funkcija controller.Move omogućuje promjenu lokacije igrača ovisno o vektoru koji joj se proslijedi.

```
float horizontal = Input.GetAxisRaw("Horizontal");
float vertical = Input.GetAxisRaw("Vertical");
Vector3 direction = new Vector3(horizontal, 0f, vertical).normalized;
velocity.y += gravity * Time.deltaTime;
controller.Move(velocity * Time.deltaTime);
```

Ako vektor smjera ima neku vrijednost, potrebno je odrediti kut prema kojemu se igrač kreće pomoću funkcije Math.Atan2. Funkcija Math.Atan2 za rezultat vraća arkus tangens kuta koji tvore vektori direction.x i direction.z [19]. Rezultat prethodne funkcije se prikazuje u radijanima, te je stoga potrebno pomnožiti s Mathf.Rad2Deg kako bi se vrijednost pretvorila u stupnjeve. Varijabla targetAngle sadrži stupanj na koji bi se igrač pritiskom tipki trebao rotirati. Kako bi se izgladila promjena rotacije igrača, potrebno je koristiti funkciju Math.SmoothDampAngle koja prima 4 parametra, a to su redom trenutni kut, kut na koji se treba rotirati, referenca na varijablu koja će sadržavati trenutnu brzinu i varijablu koja sadrži vrijeme potrebno za rotiranje objekta [19]. Korištenjem svojstva .eulerAngles rezultat varijable se prikazuje u stupnjevima umjesto kvaterniona. Promjenom vrijednosti transform.rotation mijenja se rotacija objekta u trodimenzionalnom prostoru. Ako je tijekom kretanja pritisnuta tipka Shift, brzina u određenom smjeru se množi sa vrijednosti varijable runSpeed i parametar za trčanje se postavlja na istinu, a za hodanje na laž.

```
if(direction.magnitude >= 0.1f && Input.GetKey(KeyCode.LeftShift)){
    float targetAngle = Mathf.Atan2(direction.x, direction.z) *
    Mathf.Rad2Deg;
    float angle = Mathf.SmoothDampAngle(transform.eulerAngles.y,
    targetAngle, ref turnSmoothVelocity, turnSmoothTime);
    transform.rotation = Quaternion.Euler(0f, angle, 0f);
    controller.Move(direction * runSpeed * Time.deltaTime);
}
```

```

if (isGrounded)
    playerAnimator.SetBool("isRunning", true);
    playerAnimator.SetBool("isWalking", false);
}

```

Inače ako nije pritisnuta tipka Shift, vrijednost brzine kretanja se množi s vrijednosti varijable `walkSpeed` i parametar za hodanje se postavlja na istinu, a za trčanje na laž.

```

else if (direction.magnitude >= 0.1f){
    float targetAngle = Mathf.Atan2(direction.x, direction.z) *
    Mathf.Rad2Deg;
    float angle = Mathf.SmoothDampAngle(transform.eulerAngles.y,
    targetAngle,
    ref turnSmoothVelocity, turnSmoothTime);
    transform.rotation = Quaternion.Euler(0f, angle, 0f);
    controller.Move(direction * walkSpeed * Time.deltaTime);
    if(isGrounded)
        playerAnimator.SetBool("isWalking", true);
        playerAnimator.SetBool("isRunning", false);
}

```

Ako se miruje parametri za hodanje i trčanje se postavljaju na laž.

```

else{
    playerAnimator.SetBool("isWalking", false);
    playerAnimator.SetBool("isRunning", false);
}

```

Kako bi se igrač tijekom sudara s preprekama ili završnim trofejom vratio na početnu poziciju potrebno je koristiti metodu `OnTriggerEnter()` koja se poziva kada dođe do kolizije. U slučaju da je naziv objekta s kojim je lik došao do kolizije `trophy` treba se prikazati pobjednički tekst koji će biti vidljiv dvije sekunde. U sceni se nalazi objekt `platno` (engl. `Canvas`) kako bi se na kraju igre prikazao tekst `Well done`. Objekt `platno` u hijerarhijskom prozoru ima objekt `dijete` `Text` koji unutar prozora `Inspector` ima svojstva za pozicioniranje teksta `Pos X` i `Pos Y`, te svojstvo `Text` za unos teksta koji će se prikazati na sceni.

Tijekom korištenja komponente `Character Controller` izravna promjena lokacije lika nije moguća, te je za teleportaciju lika potrebno privremeno ugasiti komponentu.

```

private void OnTriggerEnter(Collider other){
    if (other.name == "trophy"){
        text.SetActive(true);
        StartCoroutine(StartCounter());
        IEnumerator StartCounter()
        {
            yield return new WaitForSeconds(2);
            text.SetActive(false);
        }
    }
    controller.enabled = false;
    transform.position = start.transform.position;
    controller.enabled = true;
}

```

Logika za praćenje igrača kamerom je vrlo jednostavna i nalazi se u zasebnoj skripti. Javna varijabla player tipa GameObject će sadržavati objekt igrača iz scene, kako bi se mogla pratiti njegova lokacija. Preostale 3 javne varijable tipa float se odnose na pomak kamere od pozicije igrača.

```

public GameObject player;
public float x, y, z;

```

Unutar metode Update() se nalaze 3 varijable camx, camy i camz koje predstavljaju konačnu poziciju kamere u odnosu na igrača. Kamera prati igrača po dužoj ravni platforme i vertikalnoj osi, dok je bočna udaljenost na z osi stalna. Za postavljanje parametra transform.position potrebno je koristiti tip Vector3.

```

float camx = player.transform.position.x + x;
float camy = player.transform.position.y + y;
float camz = z;
transform.position = new Vector3(camx, camy, camz);

```

U sceni se nalazi prepreka sa sjekirama koje se rotiraju, a za njih također postoji zasebna skripta. Javna varijabla speed tipa float definira brzinu okretanja sjekire, a varijabla direction tipa int ima vrijednost 1 ili -1 i određuje smjer rotacije.

```

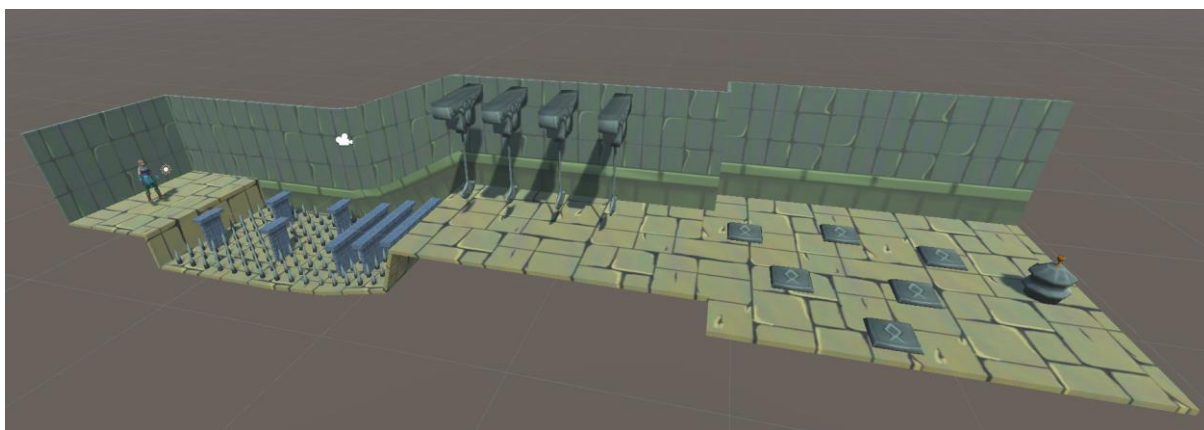
public float speed;
int direction = 1;

```


Varijabla `currentAngle` tipa `float` sadrži trenutni kut sjekire koji se nalazi između vrijednosti 0 i 360. Kako bi sjekira mijenjala smjer kada prijeđe 90 i -90 stupnjeva potrebno je sve stupnjeve veće od 180 umanjiti za 360 kako bi oni postali negativni.

```
float currentAngle = transform.rotation.eulerAngles.z ;  
    if (currentAngle > 180.0f)  
        currentAngle -= 360.0f;  
    if (currentAngle > 90 && direction == 1)  
        direction *= -1;  
    if (currentAngle < -90 && direction == -1)  
        direction *= -1;  
    transform.Rotate(0, 0, direction * speed*Time.deltaTime);
```

Vizualni dio scene se sastoji od platforme za kretanje, šiljaka, lebdećih ploča, sjekire, nagazne zamke i trofeja. Svi 3D objekti iz scene su izrađeni unutar Blender-a i na isti način poput lika igrača preneseni u Unity.



Slika 48. Prikaz scene igre

11. Zaključak

Zahvaljujući besplatnom programu Blender, danas svatko može iskazati svoju kreativnost i inovativnost kroz izradu 3D modela, 3D animacija, materijala, 2D animacija, vizualnih efekata i simulacija. Pomoću znanja i iskustva stečenog radom u Blenderu, moguće je izrađivati projekte visoke kvalitete koji doprinose zaradom uz niski trošak. Iako Blender svakim danom sve više napreduje i posjeduje dodatke koji uvelike olakšavaju i ubrzavaju rad, on još uvijek nije dorastao specijaliziranim industrijskim standardima poput ZBrush-a, Maya-e i Substance 3D Painter-a. U Blenderu je moguće izraditi 3D modele koji su svojom kvalitetom vrlo blizu onima koji se izrađuju u komercijalnim programima, međutim potrebno je puno više vremena za njihovu izradu. Stoga se Blender smatra odličnim za početnike, hobiste i male nezavisne studije kojima je budžet vrlo važan.

S razvojem tehnologije pojavljuju se novi programi i alati koji mogu znatno ubrzati proces izrade 3D modela, poput umjetne inteligencije ili fotogrametrije.

U ovom radu je prikazan jedan od mogućih načina izrade likova za video igre uz pomoć besplatnih programa koji su svima dostupni kao što su Blender i Krita. Izrađeni 3D model ima mali broj lica koja su glatko osjenčana, što je svojstveno modelima koji se koriste za video igre. Prikazan je način izrade digitalnog kipa i primjer njegovog korištenja za izradu teksture ambijentalne okluzije. Objašnjen je proces retopologije i pravila kojih se je potrebno pridržavati tijekom izrade modela s manjim brojem poligona. Zatim su objašnjeni svi koraci koje je potrebno odraditi prije izrade teksture poput izrade UV mape, dodjele materijala i izrade kaveza. Na početku poglavlja koje se odnosi na animiranje, prikazana je izrada kostura i objašnjeni su pojmovi poput direktne i inverzne kinematike. Kako bi se pozicioniralo 3D model u određenu pozu, bilo je potrebno kostima dodijeliti utjecaj nad mrežom 3D modela. U zadnjem poglavlju navedeni su mogući formati izvoza 3D modela za korištenje u programima za izradu igara, te način uvoženja 3D modela u program Unity. Nakon uvoza 3D modela potrebno je postaviti materijale 3D modela i opcije za animacije. Kako bi igrač mogao upravljati svojim likom, potrebno je liku dodijeliti komponente animator i skriptu za kretanje.

Postoje razni načini izrade 3D modela za video igre, a odabir jednog od njih ovisi o zadanim specifikacijama koje će 3D modeli imati u igri, poput izvođenja animacija, rastavljanja modela na više dijelova, modularnosti dijelova modela, ograničenim resursima i samoj vidljivosti modela u sceni.

Popis literature

- [1] Blender (bez dat.) "Structure" [Na Internetu]. Dostupno: <https://docs.blender.org/manual/en/2.80/modeling/meshes/structure.html> [pristupano 20.06.2023.].
- [2] "3D mesh" (2016.) [Na Internetu]. Dostupno: <https://www.techtarget.com/whatis/definition/3D-mesh> [pristupano 20.06.2023.].
- [3] TurboSquid (bez dat.) "Topology" [Na Internetu]. Dostupno: <https://resources.turbosquid.com/checkmate/checkmate-product-presentation/topology/> [pristupano 20.06.2023.].
- [4] Blender (bez dat.) "Features" [Na Internetu]. Dostupno: <https://www.blender.org/features/> [pristupano 20.06.2023.].
- [5] Mark Segasby (31.01.2022.) "What is a HDRI map? " [Na Internetu]. Dostupno: <https://www.lightmap.co.uk/blog/whatisanhdrimap/> [pristupano 02.07.2023.].
- [6] Blender (bez dat.) "Viewport Shading" [Na Internetu]. Dostupno: <https://docs.blender.org/manual/en/latest/editors/3dview/display/shading.html> [pristupano 20.06.2023.].
- [7] David Revoy (23.09.2022.) "Sara concept art" [Na Internetu]. Dostupno: <https://opengameart.org/content/sara-concept-art> [pristupano 10.07.2023.].
- [8] Andrew Price (03.12.2021.) "Blender Beginner Tutorial – Part 1" [Na Internetu]. Dostupno: https://www.youtube.com/watch?v=nloXOplUvAw&ab_channel=BlenderGuru [pristupano 10.07.2023.].
- [9] CG Boost (15.03.2021.) "Blender Sculpting Tutorial: Full Advanced Creature Creation Workflow" [Na Internetu]. Dostupno: https://www.youtube.com/watch?v=tQfFlzHJJ88&ab_channel=CGBoost [pristupano 20.07.2023.].
- [10] Epic Games (bez dat.) "Nanite Virtualized Geometry" [Na Internetu]. Dostupno: <https://docs.unrealengine.com/5.0/en-US/nanite-virtualized-geometry-in-unreal-engine/> [pristupano 20.07.2023.].
- [11] Grant Abbitt (02.10.2020.) "Retopology a Detailed Guide – Part 1 – The Setup" [Na Internetu]. Dostupno: https://www.youtube.com/watch?v=6sE-p8PcHDo&ab_channel=GrantAbbitt [pristupano 22.07.2023.].
- [12] TurboSquid (bez dat.) "Tris, Quads & N-Gons" [Na Internetu]. Dostupno: <https://resources.turbosquid.com/training/modeling/tris-quads-n-gons/> [pristupano 22.07.2023.].

- [13] Mara Calvello (27.04.2022.) "What Is UV Mapping? How It Makes 3D Models Come To Life" [Na Internetu]. Dostupno: <https://www.g2.com/articles/uv-mapping> [pristupano 22.07.2023.].
- [14] The CG Essentials (25.01.2022.) "How to Add TRANSPARENCY to Textures in Blender" [Na Internetu]. Dostupno: https://www.youtube.com/watch?v=udRLe6F6Z-0&ab_channel=TheCGEssentials [pristupano 01.08.2023.].
- [15] Grant Abbitt (03.02.2019.) "Baking Perfect Texture Maps Using a Cage" [Na Internetu]. Dostupno: https://www.youtube.com/watch?v=nJ0PM7m9TJc&ab_channel=GrantAbbitt [pristupano 01.08.2023.].
- [16] Grant Abbitt (19.02.2021.) "Axe – Detailed Game Objects – P18 – Ambient Occlusion Maps in Detail" [Na Internetu]. Dostupno: https://www.youtube.com/watch?v=ZVxBqYbYcPs&ab_channel=GrantAbbitt [pristupano 05.08.2023.].
- [17] Kekdot (10.03.2021.) „Easy Character Rigging with Inverse Kinematics | Blender 2.92 | Beginners tutorial“ [Na Internetu]. Dostupno: https://www.youtube.com/watch?v=hWfUe03lb5E&ab_channel=Kekdot [pristupano 05.08.2023.].
- [18] Blender (bez dat.) „Dope Sheet“ [Na Internetu]. Dostupno: https://docs.blender.org/manual/en/latest/editors/dope_sheet/introduction.html [pristupano 10.08.2023.].
- [19] Brackeys (24.05.2020.) „THIRD PERSON MOVEMENT in Unity“ [Na Internetu]. Dostupno: https://www.youtube.com/watch?v=4HpC--2iowE&ab_channel=Brackeys [pristupano 11.08.2023.].

Popis slika

Slika 1: Osnovni elementi 3D modela	3
Slika 2: Mreža 3D modela	3
Slika 3: Različite topologije modela istih oblika (Izvor: Turbosquid 3D Resources, 2023)	4
Slika 4. Normale lica u Blender-u	4
Slika 5. Početno sučelje Blender-a	5
Slika 6. 3D prikaz scene	6
Slika 7. Pogled iz kamere	6
Slika 8. Lista objekata u sceni	9
Slika 9. Prozor s postavkama	10
Slika 10. Vremenska crta.....	10
Slika 11. Skup referentnih slika u programu PureRef	11
Slika 12. Početni oblik tijela čovjeka	12
Slika 13. Remesh alat	13
Slika 14. Dva objekta s različitom gustoćom poligona	13
Slika 15. Koncept lika Sare (Izvor: David Revoy, 2022)	14
Slika 16. Objekti s različitim mrežama i spojeni objekti	14
Slika 17. Završeni digitalni kip čovjeka	15
Slika 18. Zalijepljena lica novog objekta pomoću Skupljajuće folije	16
Slika 19. Zrcaljenje mreže objekta pomoću Ogledala	17
Slika 20. Objekt ravnina postavljen blizu digitalnog kipa	17
Slika 21. Vrijednosti modifikatora Shrinkwrap i Mirror	18
Slika 22. Primjer problema uzrokovanih korištenjem poligona s više od 4 vrha (Izvor: Turbosquid 3D Resources, 2023)	19
Slika 23. Digitalni kip i rezultat retopologije.....	20
Slika 24. Rukavica nastala odvajanjem od mreže tijela	20
Slika 25. Primjer korištenja Solidify modifikatora.....	20
Slika 26. Primjer teksture Normal map	21

Slika 27. Digitalni kip i model s malim brojem poligona koji koristi Normal map	21
Slika 28. Primjer UV mape kocke	22
Slika 29. Svako lice kocke predstavlja zasebni UV otok	22
Slika 30. Dva UV otoka.....	23
Slika 31. UV mapa 3D modela čovjeka.....	23
Slika 32. Kristali koji imaju zajedničke UV otoke	24
Slika 33. Izbornik Overlays	25
Slika 34. Rastegnuta UV lica kocke	25
Slika 35. Izbornik Material	26
Slika 36. Transparentni objekt Lančić	27
Slika 37. Čvorovi unutar Uređivača sjenčanja.....	28
Slika 38. Kavez za pečenje tekstura (žuta boja)	30
Slika 39. Mapa ambijentalne okluzije	30
Slika 40. Konačna tekstura 3D modela	30
Slika 41. Kostii koje utječu na 3D model.....	31
Slika 42. Kostii za inverznu kinematiku.....	32
Slika 43. Ograničenje kostiju Inverzna kinematika	33
Slika 44. 3 ključna okvira iz animacije trčanja	34
Slika 45. Dope Sheet prozor s animacijom za hodanje	35
Slika 46. Prozor za izvoz 3D modela u Blender-u	36
Slika 47. Prikaz stanja u animatoru	37
Slika 48. Prikaz scene igre	43