

# Izrada 3D ljudskog modela u Blenderu i animiranje u Unityju

---

Stipić, Dan

Master's thesis / Diplomski rad

2019

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:216:042427>

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

*Download date / Datum preuzimanja:* **2025-03-20**



*Repository / Repozitorij:*

[Faculty of Graphic Arts Repository](#)



SVEUČILIŠTE U ZAGREBU

GRAFIČKI FAKULTET

DAN STIPIĆ

IZRADA 3D LJUDSKOG MODELA U  
BLENDERU I ANIMIRANJE U UNITYJU

DIPLOMSKI RAD

Zagreb, 2019.



Sveučilište u Zagrebu  
Grafički fakultet

DAN STIPIĆ

# IZRADA 3D LJUDSKOG MODELA U BLENDERU I ANIMIRANJE U UNITYJU

DIPLOMSKI RAD

Mentor:

prof. dr. sc. Nikola Mrvac

Student:

Dan Stipić

Zagreb, 2019



## Sažetak

U ovome radu opisan je postupak izrade 3D ljudskog modela koji je prilagođen za korištenje unutar sučelja za video igre. Objasnjavat će se postupak izrade 3d modela u Blenderu, na koje pojedinosti se treba paziti kako bi model bio spreman za animiranje, te zatim se nastavlja raditi u game engineima ili modulima za video igre poput Unityja ili Unreala. U ovom slučaju, koristit će se Unity sučelje, te će se prikazati postupak pripremanja modela za kretanje i interakciju unutar 3D svijeta Unity sučelja. Važan segment će biti orijentiran oko topologije (raspored poligona koji omogućuje deformacije na ispravan način) te kako se manipulira istom kako bi smanjili broj poligona i uštedjeli na memoriji a u isto vrijeme zadržali adekvatan broj da tijekom animiranja 3d modela, kretnje djeluju realistično. Pravilna topologija je ključna tijekom postavljanja kostiju, kako bi odnos između mreže(*mesh*) i kostiju bio ispravan.

# SADRŽAJ

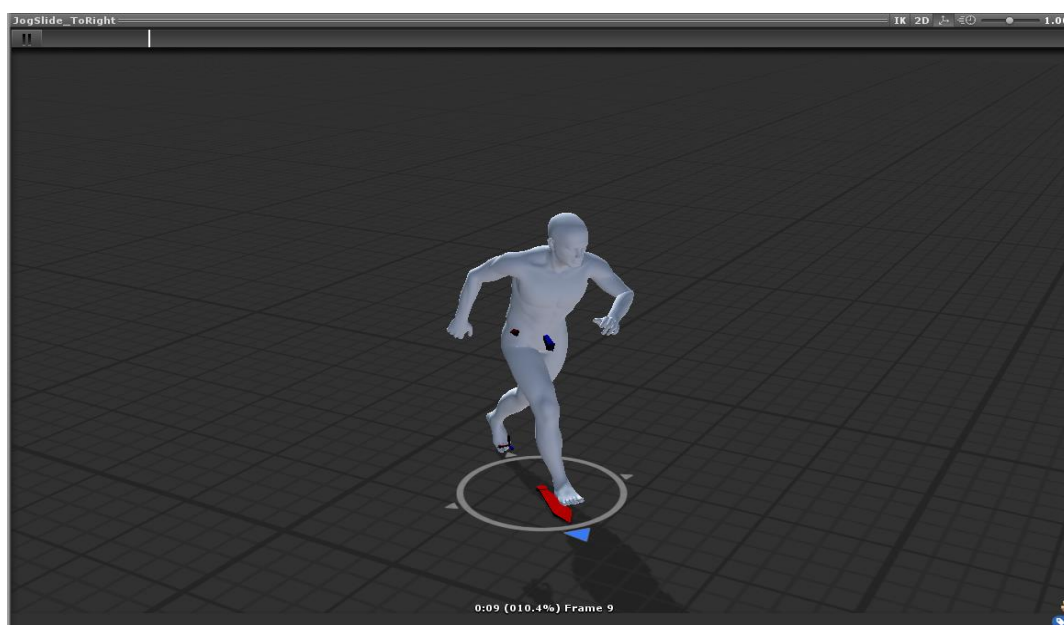
<b>1. Uvod .....</b>	<b>1</b>
1.1 Blender .....	2
1.2 Unity .....	3
<b>2. Teoretski dio.....</b>	<b>4</b>
2.1. Proces izrade 3D lika.....	4
2.2. 3D modeliranje .....	5
2.3. Rigging lika za animiranje.....	5
2.4. Struktura kvalitetnog 3D modela.....	6
2.4.1. Broj poligona .....	6
2.4.2. Početna pozicija .....	7
2.4.3. Uredna topologija.....	7
2.4.4. Pravila imenovanja.....	9
<b>3. Eksperimentalni dio .....</b>	<b>10</b>
3.1. Alati u Blenderu .....	10
3.2. Izrada 3D modela .....	15
3.3 Rigging proces.....	19
3.3.1 Armatura .....	22
3.3.2 Ograničenja.....	24
3.3.3 Upravljači .....	29
3.4 Prednja(FK) i inverzna(IK) kinematika .....	31
3.5 Vertex grupe .....	33
<b>4. Rezultati i rasprava .....</b>	<b>35</b>
4.1. Ubacivanje modela u Unity .....	35
<b>5.Zaključak.....</b>	<b>38</b>
<b>6.Literatura .....</b>	<b>40</b>



# 1. Uvod

Tijekom razvoja video igri postoji više načina za stvaranje i animiranje 3D modela. U ovom radu će se istraživati dizajniranje i animiranje ljudskog 3D modela koristeći računalni program za 3D grafiku Blender te sučelja za razvoj igara Unity pomoću kojega će se animirati sami 3D model. Istražit će se fizikalna svojstva te njihovo utjecanje na samu animaciju, obradit će se topološka optimizacija. Obradit će se i problematika organizacije poslova stvaranja 3D modela u video industriji koji je podijeljen u dva dijela kako bi se optimizirao proces stvaranja finalnog proizvoda.

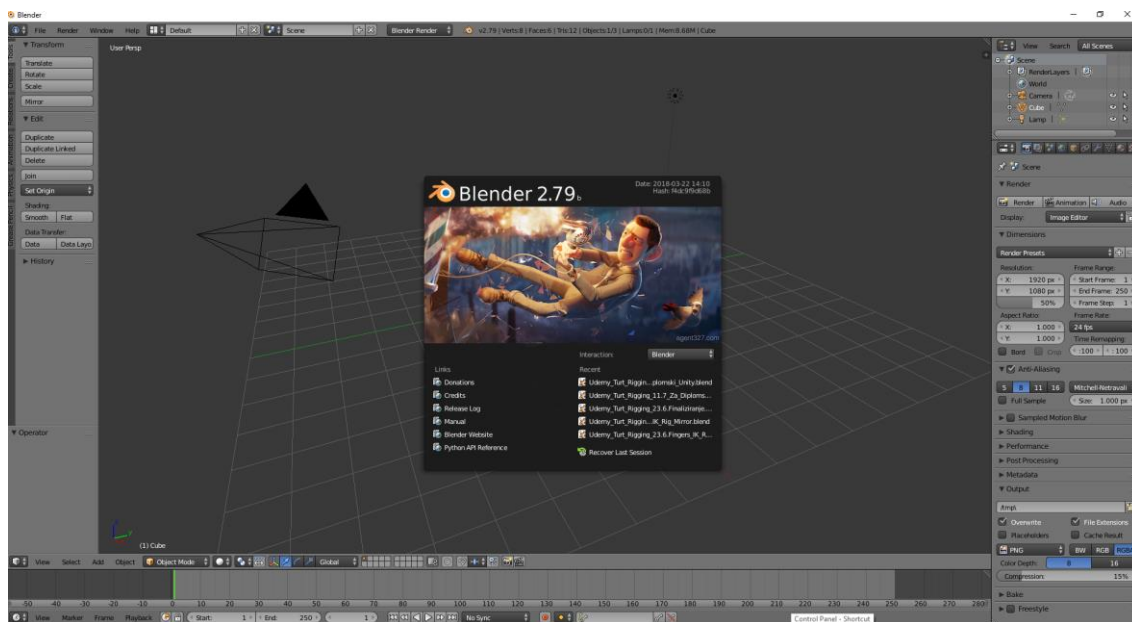
Modeliranje i animiranje je jedan dio posla, koji će se obaviti u Blenderu, no korištenje tih modela u igrama, je zaseban posao koji se nastavlja raditi u game engineima ili modulima za video igre poput Unityja ili Unreala. Za rad, koristit će se Unity sučelje, te će se prikazati postupak pripremanja modela za kretanje i interakciju unutar 3D svijeta Unity sučelja i na što se mora pripaziti prije prebacivanja modela iz Blender programa u Unity program kako bi se izbjegle greške i komplikacije pri konverziji modela.



*Slika 1, Animiranje modela u Unity sučelju*



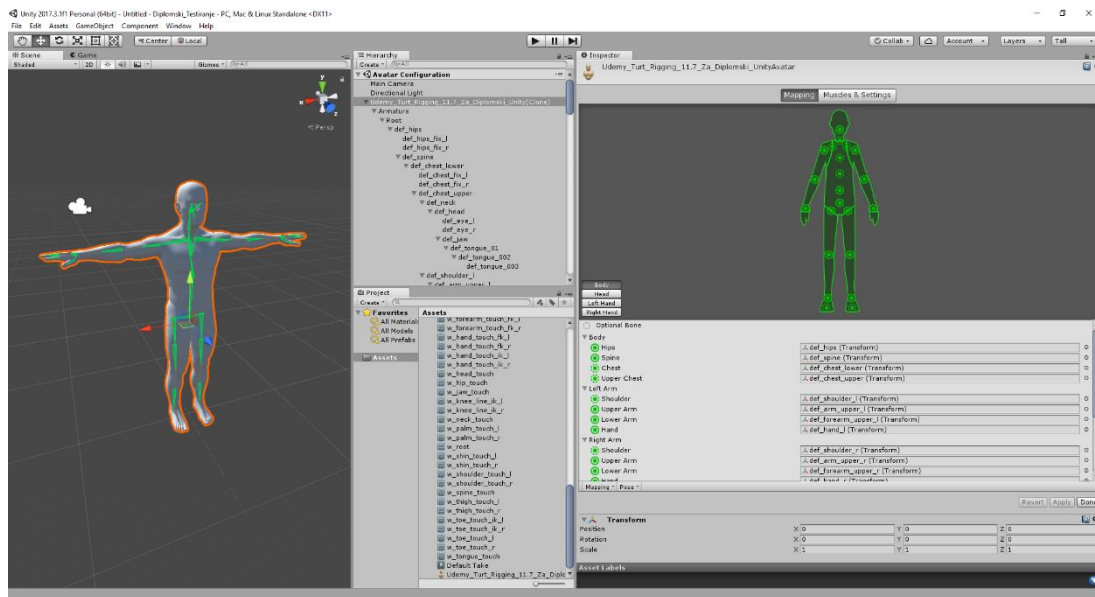
## 1.1 Blender



*Slika 2, Primjer izgleda sučelja programa*

Blender je besplatan open-source program za izradu 3D kompjuterske grafike. Pomoću njega su rađeni 3D modeli te montiranje(*rigging*) koje služi kako bi dali mogućnost kontrole navedenog modela, naj češće za potrebe animiranja.

## 1.2 Unity



Slika 3, Izgleda Unity sučelja

Unity je besplatan pokretač(engine) razvijen od strane Unity Technologies-a. Pomoću toga engine-a su renderirani modeli za testiranja lika, postavljanje animacija kako bi se lik mogao kretati ovisno o potrebi, te provjera da li je lik kompatibilan s Unity sučeljem.

## 2. Teoretski dio

### 2.1. Proces izrade 3D lika

Kreiranje 3D lika je veći zahvat nego što se čini na prvi pogled. U većim studijima, posao je podijeljen među više umjetnika zbog vremena koje je potrebno i širokog spektra vještina koje su potrebne za izradu jednog lika. U manjim projektima, dizajn likova, 3D modeliranje i animiranje mogao bi raditi jedan 3D umjetnik no s velikim limitom na broj likova koje bi se moglo dodati u igru, čisto radi vremena i zahtjevnosti. Najčešće je bolje imati različitog umjetnika odgovornog za zaseban korak unutar kreativnog procesa. Na ovaj način, posao se izvodi puno brže i teret posla se bolji podijeli na cjeline. Primjer, dizajner likova radi na novom primjerku, dok se 3D umjetnik bavi s kreiranjem 3D verzije prvog lika. Manji timovi najčešće nisu te sreće da imaju dovoljno veliki tim da se adekvatno podijele poslovi, pružajući umjetniku da se usavršava u svojem području umjesto da radi sve poslove upitne kvalitete. Prema timu koji je kreirao Overwatch od Blizzard Entertainment, posao je najčešće podijeljen na sljedeći način: jedna osoba se bavi s kreiranjem koncepta lika, druga radi 3D model, treća osoba se bavi s rigging procesom a rad na animiranju dobiva četvrta osoba. Tijekom rada na modelu nije bilo potrebe za kreiranjem pretjerano stiliziranog koncepta, pošto je glavni zadatak bio više rad na klasičnom i pojednostavljenom liku kako bi se lakše mogli prikazati problemi i zahtjevi pri radu na ostalim koracima. Mogu se koristiti referentne slike kao pozadinske slike preko kojih se počinje modelirati lika. [1]

## 2.2. 3D modeliranje

Kreiranje modela kreće od jednostavnijih oblika poput kocke. Dodajući loop cutove odnosno rezove i pomičući točke, 3D umjetnik može kreirati tijelo modela. Udovi se mogu modificirati od drugih kocki koji se produže da stvore kvadre koji se kasnije spajaju s tijelom kada su spremni. Glava se najčešće započinje od kocke ili se radi poligon po poligon prema referentnoj slici, ovisno o situaciji. Ako je lik ili objekt simetričan, umjetnik može koristiti alate za zrcaljenje kako bi se kreirali savršeno simetričan lik. Kada je alat za zrcaljenje aktivan, sve promjene na jednoj strani se automatski kopiraju na drugu stranu, znatno ubrzavajući proces izrade lika.

## 2.3. Rigging lika za animiranje

3D model je prvobitno statična mreža koja se ne može animirati. Kada se radi na humanoidima ili kreaturama kompleksnijih oblika izuzetnu važnost ima dobar sistem za micanje udova. Kreiranje sistema za micanje lika koristeći kosti se zove rigging dok se struktura može nazvati rig, armatura ili kostur. Dva glavna tipa kostiju koji se koriste su deformacijske kosti i kontrolne kosti. Deformacijske kosti su kosti koje su direktno povezane s određenim dijelom 3D mreže. Kretanjem, rotiranjem, ili skaliranjem deformacijske kosti će u isto vrijeme utjecati na okolnu mrežu 3D modela. Kontrolne kosti s druge strane, se koriste kao kontrolne poluge za druge kosti, direktno ili indirektno. Kada se kreira sistem za pokretanje zglobova, lociranih na krajevima kostiju, postoje 2 glavna tipa i oni su Forward Kinematics (FK) i Inverse Kinematics (IK). Forward Kinematics je sistem gdje se kosti miču jedna po jedna kako bi se složila poza.

Inverse Kinematics je sistem gdje kost na samom kraju lanca diktira logično kretanje ostalih kostiju niže u lancu pomoću programa. Primjerice, ako se namješta poza prstiju na ruci, jedino je potrebno micati i rotirati vrh prsta kako bi se dobio adekvatan rezultat. IK sistem predstavlja jedan od problema pri izradi projekta, zna biti ponekad kompliciran i kretanje znaju biti nepredvidljive ako se ne adekvatno primireni rig. IK sistem je veoma koristan kada se radi s udovima dok je FK efektivan kod rada s prstima i kičmom. Kada je armatura završena, model je spreman za proces weight paintinga. Blender ima postavku koja omogućuje automatsko spajanje kostura i

modela te veže zasebne kosti za određeni dio mreže modela no postavljanje automatskih težina na mrežu se treba kasnije dodatno namještati i mijenjati. Klasičan primjer problema automatskih težina je kada se lik i njegova odjeća ne kreću adekvatno tijekom animiranja, udovi prolaze kroz vlakno hlača ili kada kost zahvati veće područje mreže tijela nego što je namijenjeno za određenu kost. Rigging proces može biti poprilično dug i zahtjevan proces, ovisno o kompleksnosti zadatka.

## **2.4. Struktura kvalitetnog 3D modela**

Postoji dosta mogućih poteškoća pri izradi 3D lika. Topologija može biti problematična i neuredna, nepotrebno veliki broj poligona, rig može imati tendenciju lomiti model tijekom animiranja. Preporučljivo je znati koje su dobre navike izrade 3D modela prije nego što se upušta u izradu kompleksnih likova.

### **2.4.1. Broj poligona**

U slučaju likova u video igrama, topologija i optimizacija su izuzetno važne. Važnost dolazi iz same činjenice renderira u pravom vremenu, što znači da svaka slika koja se vidi se kreira simultano kako korisnik kontrolira svojeg lika i prati sve što se događa na ekranu. Stvaranje tih slika zahtjeva određenu snagu procesuiranja i vremena ovisno o detaljima likova i okruženja. Ako je model pretjerano kompleksan, korisnik bi trebao čekati određeno vrijeme prije nego što bi se rezultat izrenderira na ekranu, zahtijevalo bi previše vremena. Sad je pitanje, kolika je optimalna količina detalja za likove? Sve ovisi o konzoli ili mediju na kojem bi se igra konzumirala.

Unity korisničke instrukcije preporučuju idealan broj poligona od 300 do 1500 za mobilne igre, dok desktop platforme mogu podnijeti broj od 1500 do 4000. Sve to može ovisiti o broju likova simultano na ekranu, što ih je više to se treba više prilagođavati odnosno smanjivati broj poligona po liku. Za razliku od video igara koje se renderiraju u pravom vremenu, filmovi nemaju taj problem pošto se renderiraju

prije gledanja. Što znači da vrijeme renderiranja nije problem u ovom slučaju, tako da detalji mogu biti visoki. Važno je uzeti u obzir potrebe određene igre kada se radi na likovima. Razne pametne prilagodbe ovisno o okruženju i tonu igre, mogu znatno reducirati broj poligona.

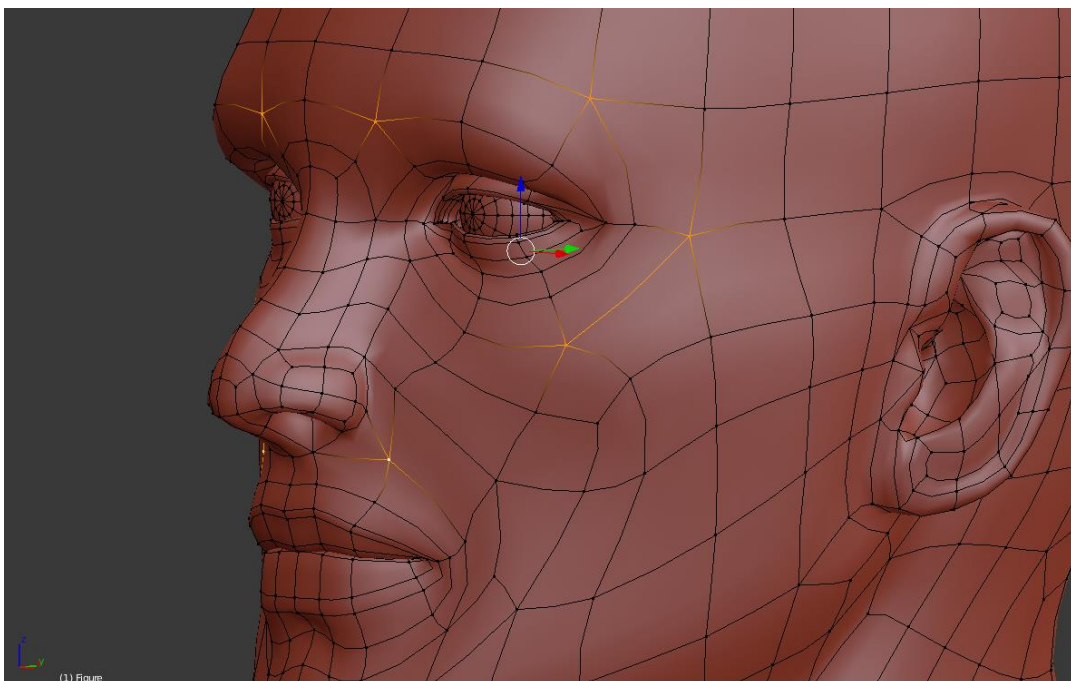
### **2.4.2. Početna pozicija**

U 3D igrama, likovi su najčešće modelirani u pozi u kojoj su ruke raširene sa strane, odnosno u takozvanoj T-pози. Poanta ove poze je olakšati modeliranje, postavljanje riga i animiranje. Raširene ruke olakšavaju i poprilično naporan proces bojanja težina na određene dijelove tijela. Unatoč svojim prednostima, postoje i mane vezane uz ovu pozu. U T-pози, teže je za animacijski softver da zaključi u kojem smjeru bi se ruka trebala kriviti te je rastezanje primjetno pri spuštanju ruku niz tijelo. U novijim igrama može se primijetiti da su ruke nešto niže spuštene. Promjena je relativno nova, te je jedna od mogućih objašnjenja činjenica da rijetko koji lik u igrama ima potrebu dizati ruke ravno iznad glave što znači da je srednja vrijednost raspona kretnje ruku nešto niža. Isto tako, imajući ruke nešto niže u početnoj pozi, čini topologiju ramena više prirodnijom te reducira rastezanje na ramenima tijekom micanja ruku.

### **2.4.3. Uredna topologija**

Kada se model ubaci u game engine, poput Unity-a, dolazi do triangulacije modela. Drugim riječima, svaki poligon će biti konvertiran u trokutaste poligone. Ako se model već sastojao od trokutova ili četverokuta sve bi trebalo biti pod kontrolom. Program će pretvoriti četverokute u trokute. Preporuka je da se ne miješaju trokutovi i četverokuti, nego da se pridržava jednoj vrsti poligona. U većina slučajeva četverokuti su bolji izbor. Problemi mogu doći ako ima poligona s više od četiri brida odnosno n-gon. Što ima više bridova na poligonu, to postoji više načina za engine da triangulira

te poligone te su rezultati nepredvidljivi. Kada se kreiraju likovi, dobra je ideja prilagoditi topologiju prema prirodnim mišićnim linijama. Ovo omogućuje bolje reagiranje mreže modela pri animiranju lika. Još jedan od mogućih problema kod topologije mogu stvarati pole-ovi. Pole je točka ili vertex koji spaja pet ili više točaka. Pole-ovi su korisni kod određenih lokacija na licu tijekom modeliranja, ali generalno mogu stvarati probleme ako umjetnik ne zna kamo ih smjestiti. Pole-ovi poremete tok rubnih petlji, čineći probleme kasnije kada se hoće dodati dodatne rubne petlje. Rubne petlje na licu modela bi idealno trebale stvarati koncentrične kružnice oko očiju i usta, to se podrazumijeva kao uredna i logična topologija lica.



*Slika 4, Prikaz korisnih pole-ova na licu*

Korisna preporuka za rad na 3D modelu je ukloniti nepotrebne poligone koji nisu vidljivi. U igrama iz prvog lica, nije potrebno imati nepotrebne poligone ispod puške lika poput nogu ili ako lik ima odjeću modeliranu na već izmodelirano golo tijelo, nije potrebno zadržati površinu modela ispod odjeće, samo čini broj poligona višim bez potrebe.

#### 2.4.4. Pravila imenovanja

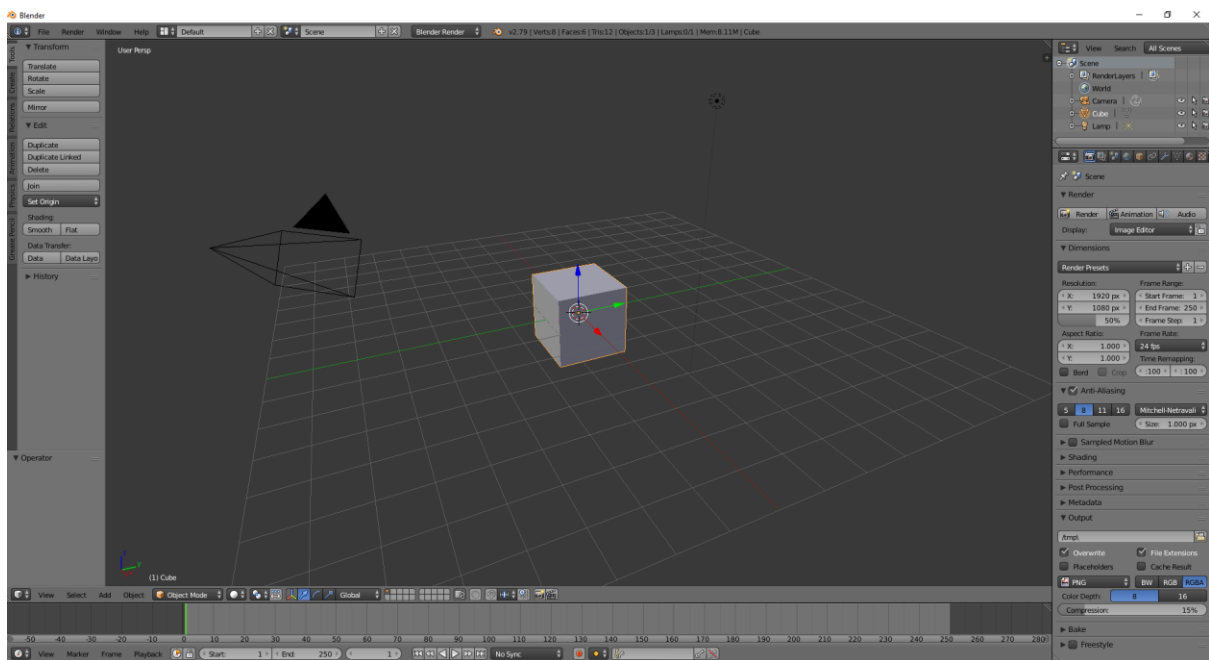
Svaki objekt i materijal koji se koristi bi trebao imati ime kako bi se lakše prepoznao. Nazivajući sve je posebno važno tijekom rigging procesa. Kostii nisu unikatnog izgleda, te se tijekom rada na njima ubrza i olakša posao ako moramo raditi samo na jednoj strani i prepusti se Blenderu da zrcali poredak i broj kostiju na drugu stranu, a u isto vrijeme i naslijedi ime ali s prikladnim sufiksom. Primjerice kada kopiramo kost lijevog ramena pod imenom „def.shoulder.l“, Blender će zrcaliti ramenu kost i promijeniti joj automatski ime na „def.shoulder.r“, zato što prepoznaje taj sufiks i zna što treba napraviti pri kopiranju na drugu stranu.



## 3. Eksperimentalni dio

### 3.1. Alati u Blenderu

Modeliranje mreže(mesh modeling), skup točkica(vertex), bridova(edges) i poligona(faces) koji čine oblik 3D objekta, je naj fundamentalniji korak u 3D razvojnom procesu. Blender podržava mnoge tehnike modeliranja, mesh modeling je bazična i naj češća tehnika modeliranja. Modeliranje mreže tipično počinje s primitivnom mrežom(mesh primitive) oblika kocke, kugle, cilindra, ovisno o želji korisnika, te se zatim manipulira mrežom kako bi stvorili veći i kompleksniji oblik.

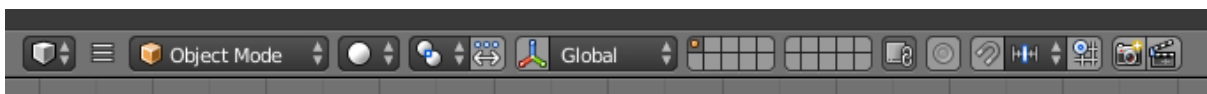


*Slika 5, početni zaslon Blender sučelja*

3D preglednik ima tri osnovne metode koje daju opciju kreiranja, editiranja i manipuliranja mreže modela. Svaka od tri metode ima razne alate, neki od alata se mogu naći u jednoj ili više metoda. Metode koje se koriste za modeliranje: Objektna metoda (*Object mode*), Obradna metoda (*Edit mode*), Oblikovna metoda (*Sculpt Mode*).

### Objektna metoda

Operacije objektna metode u Object modu imaju efekta na cijeli objekt. Object mode ima sljedeći izgled u 3D pregledniku:



*Slika 6, zaglavlje objektna metode*

### Obradna metoda

Operacije u obradnoj metodi utječu samo na geometriju selektiranog objekta, ali nema utjecaja na globalne postavke poput lokacije i rotacije objekta. Može se samo modificirati mreža objekta koji obrađujete. Kako bi modificirali druge objekte, potrebno je izaći iz obradne metode, ući u objektnu metodu te zatim selektirati traženi objekt i onda se vratiti u obradnu metodu. Obradna metoda ima sljedeći izgled u 3D pregledniku:



*Slika 7, zaglavlje obrađivačke metode*

Dio ponuđenih alata i metoda će se koristiti u ovom projektu, pošto nisu svi potrebni, od ponuđenih u objektnoj i obradnoj metodi koristi će se sljedeći:



### 1. Tipovi obradnika (*editor type*)

Editor type služi kako bi se prebacivali iz jednog u drugi obradnik, primjerice obradnik gdje radimo na objektima je 3D preglednik (*3d view*), u njemu se radi većina posla vezana za modeliranje. Drugi obradnik koji će se koristiti su grafikonski obradnik (*graph editor*) i u slučaju da želimo posebne pokrete za model, koristi će se planerska ploča (*dope sheet*). Grafikonski obradnik je zaslužan za dodavanje posebnih mogućnosti za kosti i prilagođavanje odnosa između drugih kostiju koje će se postavljati za model. To se radi pomoću upravljača (*drivers*), oni omogućuju više vrijednosti na istim kostima, primjerice prebacivanje iz jednog moda kontrole kostiju u drugi poput inverzne kinematike (*inverse kinematics*) i prednje kinematike (*forward kinematics*). Njihova uloga će biti važna kada se dođe do koraka postavljanja kostiju za 3D model.

### 2. Blender metoda (*blender mode*)

Služi za prebacivanje iz objektno metode i obradne metode, te služi kao pokazivač u kojoj metodi je trenutno postavljen.

### 3. Metoda prikaza za 3d preglednik (display method for 3D view)

Daje opciju da objekte promatramo na više načina kako bi nam olakšalo ili dalo na znanje kako bi otprilike izgledao nakon obrade. Zadana metoda pregleda je solidni pregled (*solid view*), druge koje će se koristiti su mrežni pregled (*wireframe*) i prikazni pregled (*rendered*).

Wireframe omogućuje pregled objekata poput mreže i olakšava selektiranje određenih točaka koje inače nisu vidljive iz prednje perspektive. Rendered view nam prikazuje kako će objekt izgledati pod boljim uvjetima i postavkama, pošto je to dosta zahtjevan proces, rendered view prikazuje pod nižom kvalitetom od krajnjeg produkta, kako bi mogli rotirati kameru oko objekta i pregledavati kako će izgledati bez da čini pre veliki teret za računalo.

#### **4. Glavna točka (*pivot center*)**

Uloga glavne točke je odrediti kako će se i oko koje točke širiti ili rotirati objekt. Pivot center ima više opcija. Medijan ili centralna vrijednost(*medijan point*) nam omogućuje da manipuliramo većim brojem objekata kroz zajedničku srednju točku, dok individualno podrijetlo(*individual origins*) za svaki selektiran objekt odredi zasebnu središnju točku po kojoj će se manipulirati. 3D pokazivač(*3D cursor*) kao glavnu točku nam omogućuje da sami odredimo lokaciju točke oko koje će se manipulirati selektiranim objektima.

#### **5. 3D manipulacijska naprava(*3D manipulator widget*)**

Prikazuje osi X,Y i Z u 3D prostoru čim selektirate ili kliknete s 3D pokazivačem, te kao drugu ulogu nam omogućuje da namjestimo prema kojem će se prostoru orijentirati osi, bilo to prema lokalnom(*local space*), koji predstavlja prostor selektiranog objekta, dok globalni(*global space*), predstavlja prostor cijele scene ili prema normalama objekta(*normal*).

#### **6. Slojevi(*layers*)**

Omogućuje spremanje raznih objekata po više slojeva kako bi scena tijekom rada bila urednija.

## **7. Proporcionalna obrada**(*proportional editing*)

Proportional edit je način transformiranja selektiranih elemenata (poput točaka) dok u međuvremenu ta manipulacija elementom utječe na okolne elemente.

Primjerice, pomicanjem jedne točke uzrokuje micanje okolnih ne selektiranih točaka u određenom krugu. Neselektirane točke locirane bliže selektiranoj točki micat će se više za razliku od udaljenih, odnosno micat će se proporcionalno relativno lokaciji selektiranog elementa.

## **8. Hvatanje**(*snap*)

Omogućuje hvatanje objekata i dijelova mreže za različite tipove elemenata scene tijekom transformacije selektiranog objekta. Snap ima više opcija na što će se zahvatiti, ovisno o potrebi, može se izabrati da to bude lice objekta, brid ili točka, dok inkrement daje opciju da se veže za rešetku scene koja je podijeljena na kocke.

## **9. Vrste selekcije**(*selection mode*)

Omogućuje nam da biramo između 3 selekcije, točke(*vertex*), brida(*edge*) ili plohe /lica(*face*).

## **10. Limitiranje selekcije na vidljivo**(*limit selection to visible*)

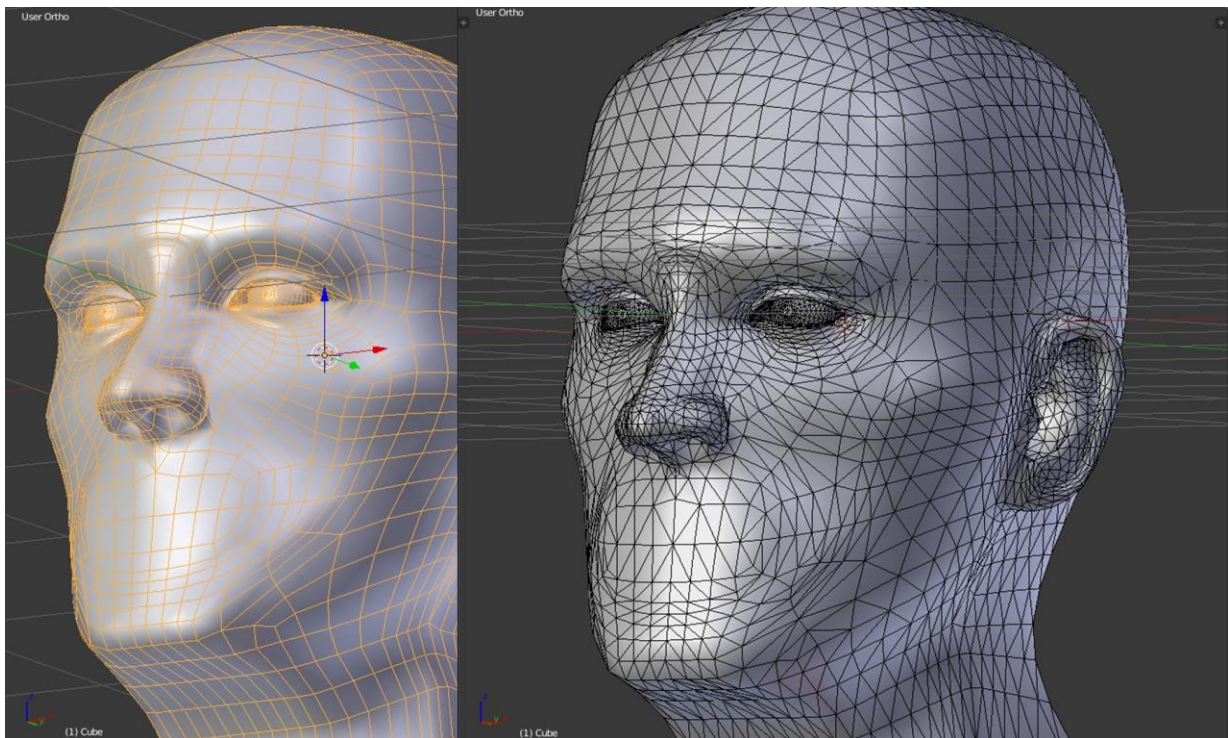
Objekti su relativno prozirni, vidljive su točke, bridovi i lice s druge strane relativno na kut gledanja. [2] [3]

## 3.2. Izrada 3D modela

Proces izrade 3D modela osobe zahtjeva vremena, znanja i vježbe. Na žalost ne postoji lagan ili brz pristup koji će dovesti do idealnog rezultata, no postoje načini i tehnike koje će ubrzati i olakšati proces izrade koje je dobro znati pri formiranju lika. Kvaliteta izrade lika omogućuje lakše i efikasnije korištenje u drugim stadijima izrade, poput postavljanja kostura i animiranje, te lakše adaptiranje gotovog rada za druge programe poput Unityja. Raspored poligona je krucijalan kako bi postigli realistične crte lica modela, mora postojati sklad a to postizemo tako da preorijentiramo poligone kako bi stvarale kružne uzorke na licu.

Poligoni su 2D oblici s najmanje 3 brida i 3 kuta, a u 3D grafici se najčešće koriste dvije vrste poligona, trokuti(*triangle*) i četverokuti(*quad*). Pošto su trokuti najjednostavniji poligoni, svaki objekt se može podijeliti na trokute ali se trokut ne može podijeliti na ništa drugo osim na trokute. Zbog ovog razloga, crtanje trokutova je znatno lakše nego crtanje poligona višega reda, stvarajući manje posla za računalo. Iz ovog razloga su trokutovi toliko često korišteni u kompjutorskoj grafici.

Uloga quadova odnosno četverokuta u 3D grafici je više kao pomagalo korisnicima, pošto modeliranje s poligonima poput trokutova je znatno teže. Quadovi čine proces modeliranja bržim i preciznijim zato što nam olakšava posao stvaranja krivulja i oblina, definiranje topologije je znatno lakše. Topologija je organizacija, tok i struktura točaka, bridova i lica 3D modela. Game engine tijekom konverzije, kada se prebacuje lik u Unity program, pretvara kvadrate u trokute. [4] [5]

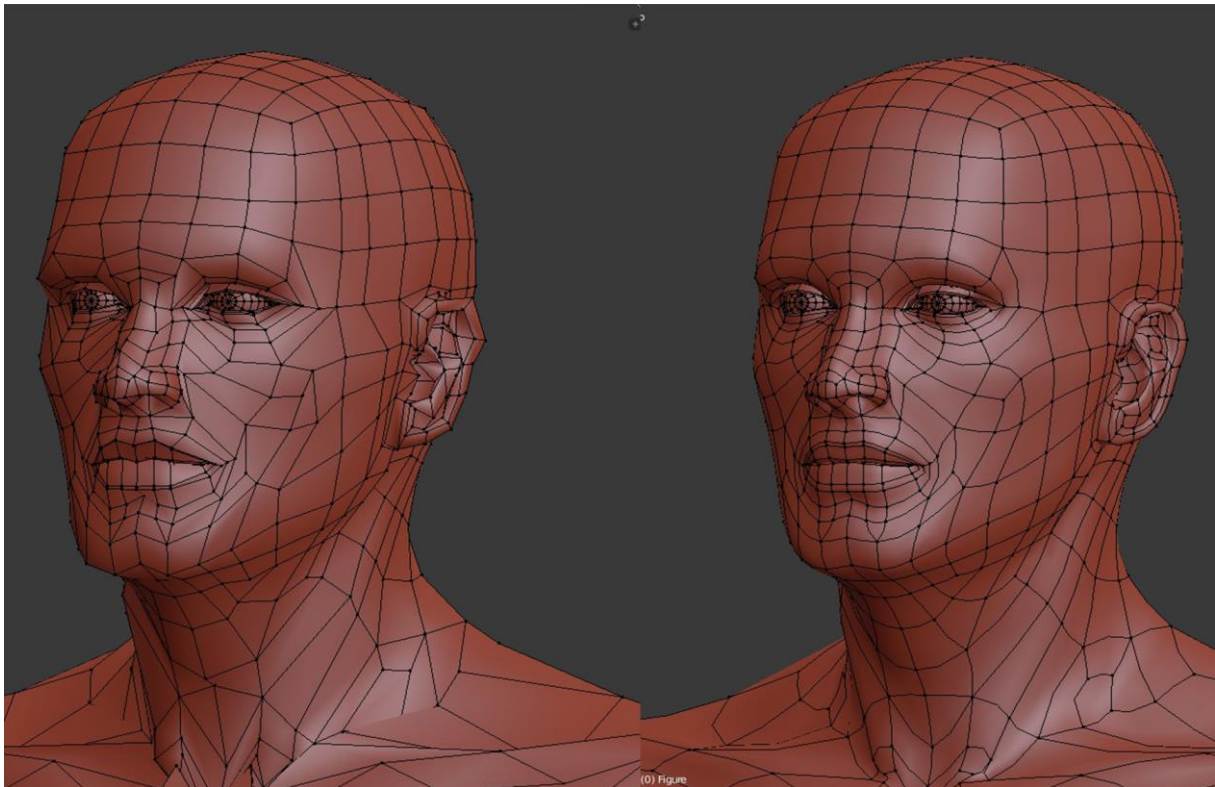


*Slika 9, Izgled mreže koristeći kvadrate i trokuteve*

Uređivanje topologije je važno iz još jednog razloga, kako bi smanjili broj poligona (*polycount*). Polycount je izuzetno važan za video igre, svaka kretnja i akcije objekata koje se odvijaju unutar igre se mora renderirati u stvarnom vremenu, čineći konstantni teret na računalo. Što manje poligona, to je manje posla za procesor. Nije više dovoljno da likovi izgledaju dobro, nego da su i ekonomični po svakom pogledu. U Blenderu kreće čišćenje poligona kada je osnovni oblik postignut, tada se može uklanjati višak koji ne bi trebao utjecati na kvalitetu i izgled modela. Koristeći rotiranje bridova (*rotate edges*), kratice `ctrl + A`, preusmjeravaju se bridovi ili lica da odgovaraju toku topologije i u isto vrijeme omogućuje da se uoči koji poligoni nisu potrebni.

Također važan aspekt rada na 3D modelu je korištenje modifikatora (*modifier*), njihova je uloga raditi većinu posla umjesto korisnika. Primjerice, modifikator podjela površine (*subdivision modifier*) se koristi kako bi podijelio plohe mreže modela u manje plohe dajući ugađeni izgled. Koristeći ovaj modifikator, omogućava modeliranje kompleksnih ugađenih površina s jednostavnom mrežom koja sadrži relativno malo točaka. Dopušta modeliranje mreža visoke rezolucije bez potrebe za spremanjem i održavanjem velikih količina podataka i daje ugađeni organski izgled objektu. Ovaj

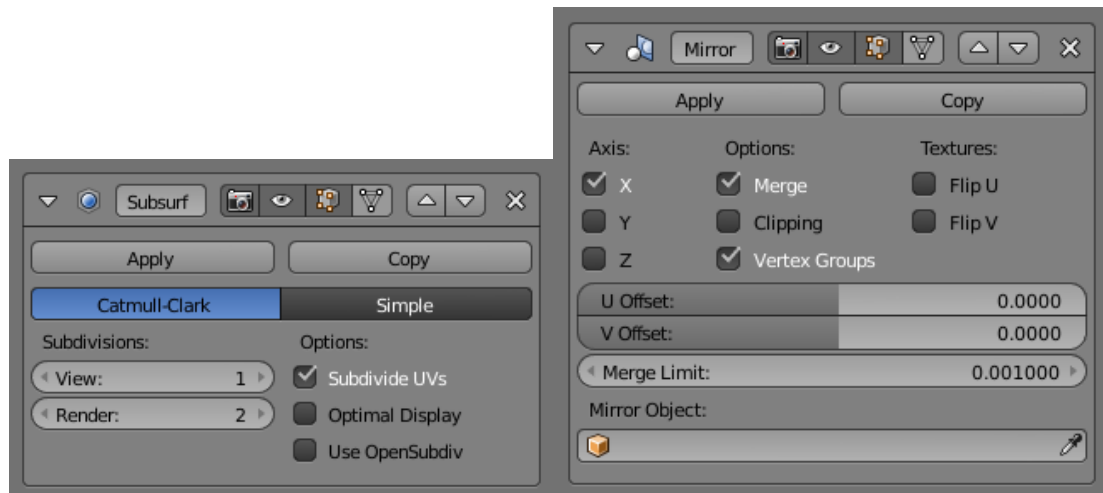
proces kreira virtualnu geometriju koja se generira bez da utječe na originalnu mrežu, ali se može konvertirati u pravu geometriju pritiskom gumba prihvatiti(*apply*) na prozoru modifikatora. Pod opcijama ovog modifikatora koje su od važnosti u ovom primjeru su podjela(*subdivision*), koji je podijeljen na view i render. Subdivision se može numerički odrediti, što je veći broj, to je veća podjela lica dajući više ugađeni izgled, ali i veće opterećenje za program. View pokazuje kako će te promjene izgledati u 3D pregledniku, dok render pokazuje kako će izgledati u render pregledniku. [6]



*Slika 10, Izgled modela bez i sa SubSurf modifikatorom*



Također važan modifikator koji se koristi je zrcaljenje (*mirror modifier*), njegova svrha je kako i njegovo ime upućuje, da zrcali mrežu objekta ovisno o lokalnoj osi, bila to X, Y ili Z, preko lokacijskog podrijetla objekta (*objekt origin*). Opcije koje se nalaze na ovom modifikatoru koje će biti od koristi su spajanje (*merge*) i *clipping*. Merge služi u slučaju da su dvije točke na istoj lokaciji, originalna točka i točka koja je zrcaljena, one će se spojiti u jednu. Clipping sprječava prelaženje točaka na zrcaljenu stranu tijekom rada u obradnoj metodi.



Slika 11, Subsurf i mirror modifier

### 3.3 Rigging proces

Montiranje(*Rigging*) je generalni termin za dodavanje controla objektu, tipično za potrebe animiranja. Rigging obično involvira korištenje jednog ili više sljedećih svojstava:

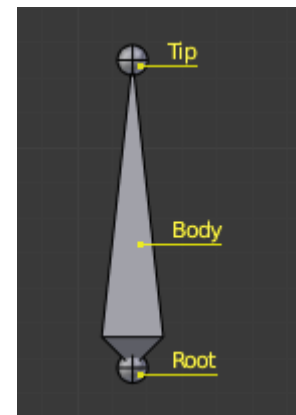
- **Armatura**(*Armature*)  
Ovo svojstvo omogućuje mrežnim objektima da imaju fleksibilne zglobove i često se koristi za animiranje kostura.
- **Ograničenja**(*Constraints*)  
Kako bi se kontroliralo kakve pokrete ili rotacije koje imaju smisla za određene kosti, primjerice ograničiti rotaciju podlaktice.
- **Modifikatori**(*Object Modifiers*)  
Rad na mreži dovodi do dosta deformacija, postoje razni modifikatori koji pomažu kontrolirati ishod.
- **Upravljači**(*Drivers*)  
Omogućuje opremi(*rig*) da kontrolira više različitih vrijednosti odjednom, također omogućuje neke postavke da se automatski ažuriraju bazirano na promjenama na drugim lokacijama.

**Kosti** su osnovni element svake armature. Kosti u armaturi se generalno mogu klasificirati unutar dva različita tipa: Deformacijske(*Deforming Bones*) i Kontrolne(*Control Bones*) kosti. [7]

## Struktura kosti

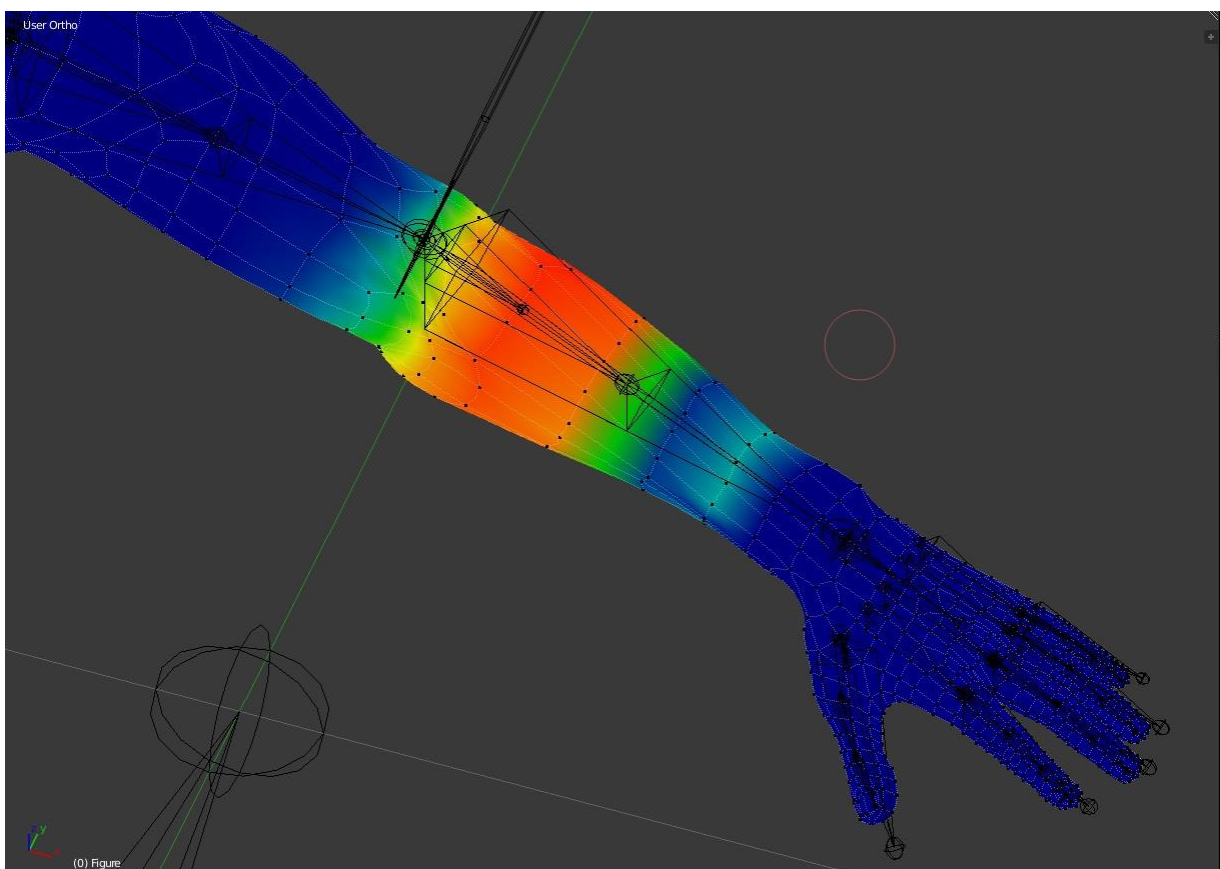
Kosti imaju 3 elementa:

1. "Početni zglob" (*root* ili *head*)
2. "Tijelo" (*body*)
3. "Krajnji zglob" (*tip* ili *tail*)



## Deforming Bones

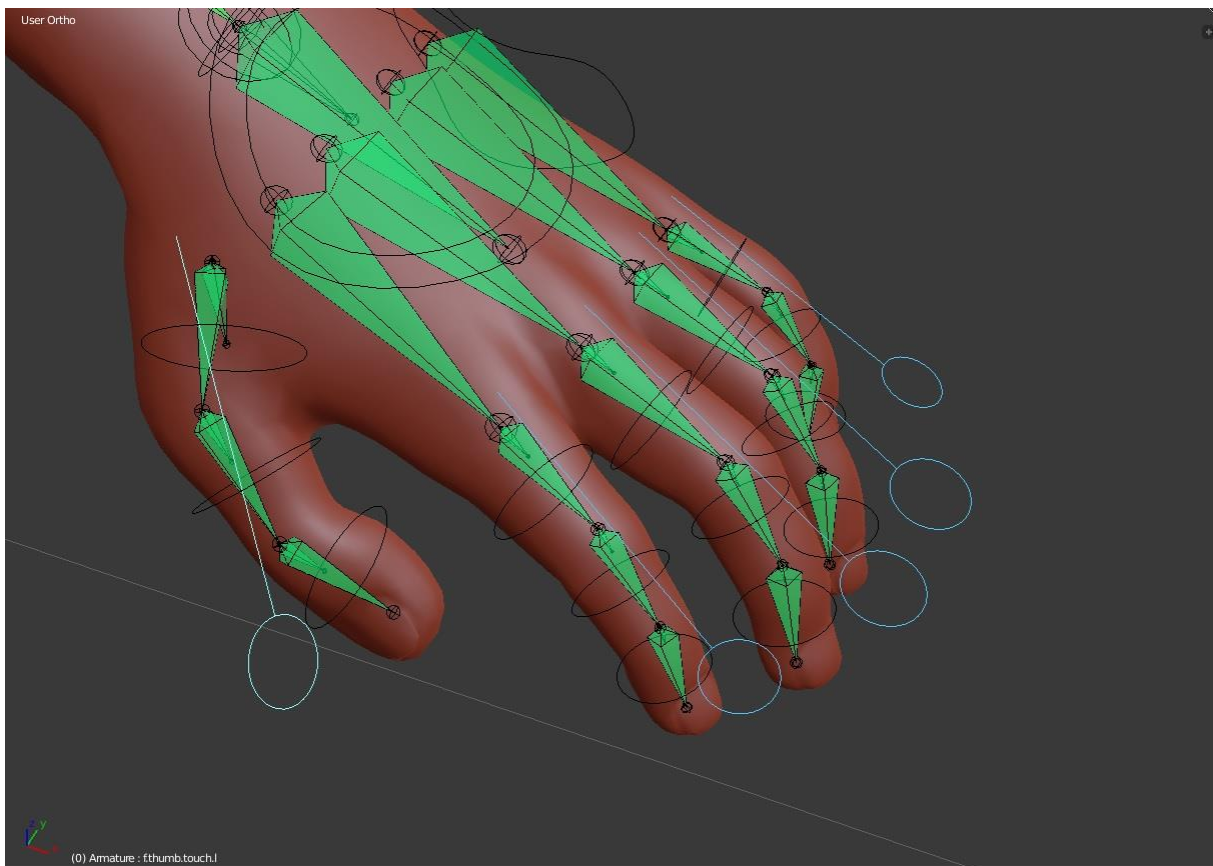
Kosti koje kada se manipulira dovode do manipuliranja okolnih točaka koje su dodijeljene toj specifičnoj kosti. Deformacijske kosti su direktno involvirane u mijenjanju pozicija točaka koje su dodijeljene toj kosti.



Slika 12, Deformacijska kost (podlaktica) i područje modela koji zahvaća

## Control Bones

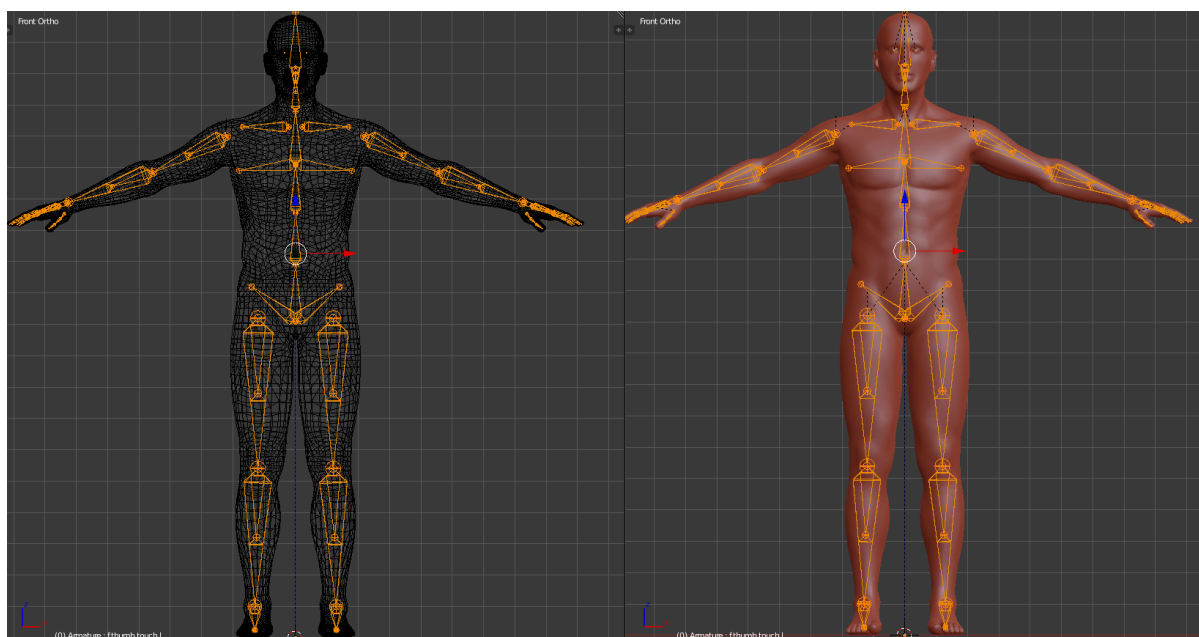
Kosti koje obavljaju ulogu slično polugama, tako da kontroliraju kako će se druge kosti ili objekti ponašati kada su oni manipulirani. Kontrolne kosti mogu na sebi imati posebne postavke koje omogućuju mijenjanje između dva načina rada, primjerice pomicanjem vrijednosti od 0 do 1 mijenja iz inverzne kinematike (*inverse kinematics*) u prednju kinematiku (*forward kinematics*), što dosta olakšava proces animiranja kada su oba načina rada potrebna. Kontrolne kosti nisu direktno korištene da bi mijenjale pozicije točaka, naprotiv, kontrolne kosti često nemaju nikakve točke dodijeljene njima. Kontrolnim kostima se mogu pridijeliti drugačiji oblici, ne moraju nužno izgledati kao klasične kosti koje Blender postavi.



Slika 13, Kontrolne kosti i njihov utjecaj na kosti prstiju

### 3.3.1 Armatura

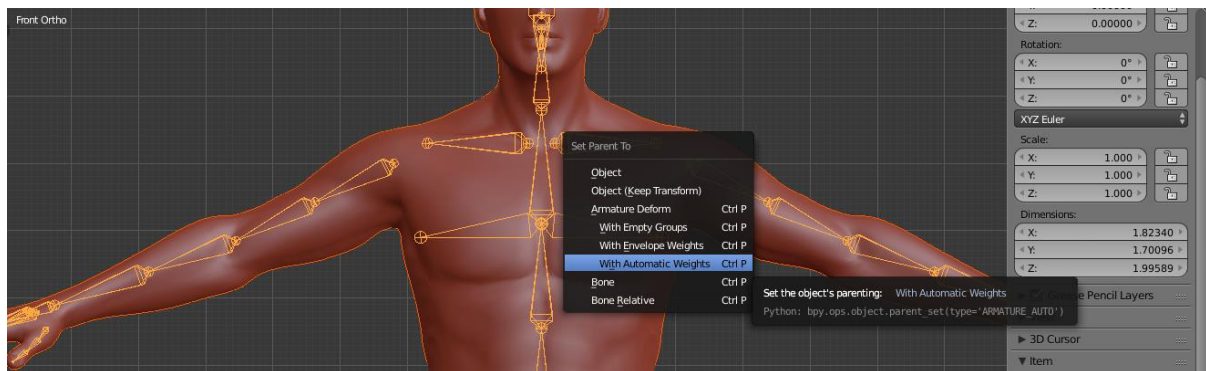
Armatura u Blenderu može se zamisliti kao nešto slično armaturi pravog kostura, i poput pravog kostura, armatura se može sastojati od mnogih kostiju. Ove kosti se mogu micati i sve što je pričvršćeno za njih ili povezano će se kretati i deformirati na sličan način. Armatura je tip objekta koji se koristi za rigging. Oprema(*rig*) su kontrole i konci koji kontroliraju marionetu. Armaturni objekti posuđuju dosta ideja od pravih kostura. Kako su armature dizajnirane za poze, bilo to za statičnu ili animiranu scenu, one imaju posebnu stanje koje se zove odmorna pozicija(*rest position*). To je zadani oblik armature, zadana pozicija/rotacija/omjer njihovih kostiju, koji su namješteni u obradnoj metodi(Edit Mode).



Slika 14, Izgled gotove armature na 3D ljudskom modelu

Kada je spremna armatura, dolazi se do zadnjeg koraka koji omogućuje korištenja armature za kontrole modela a to je postavljanje armaturne deformacije(*Armature Deform*). Selektiramo prvo model zatim armaturu te se stisne ctrl + P koji služi kako bi se dodijelio roditelj(*set parent to*), zatim se otvori prozor koji nam daje opciju Armature Deform s automatskim teretom(With Automatic Weights). Dodjela roditelja je namještanje odnosa između 2 objekta, roditelj je onaj koji diktira kretanje djeteta, u

ovom slučaju, kostur(armatura) je roditelj dok je model(mesh/mreža) dijete. Automatic Weights ili automatski teret je postavka koja radi posao namještanja utjecaja kostiju na model umjesto nas. Određivanja intenziteta kontrole određene kosti na njoj opredijeljen dio modela, primjerice koliku kontrolu će nadlaktična kost imati na taj dio ruke modela. Postavljanje tereta se može raditi „ručno“ odnosno korisnik može sam sebi odrediti terete. Pošto je automatic weights i dalje donekle limitiran, dobro je naknadno doraditi odnose u Bojanje Tereta (*Weight Paint*).



*Slika 15, Postavljanje automatskih tereta*

### 3.3.2 Ograničenja

Ograničenja(*constraints*) su način na koji možemo kontrolirati svojstva objekta (lokacija, rotacija, razmjer), koristeći obične statične vrijednosti(poput limit), ili drugi objekt, nazvan „meta“(poput kopiraj). Unatoč tome što su ograničenja korisna u statičnim projektima, njihov glavni razlog korištenja je animiranje. Mogu se kontrolirati animacije objekta kroz mete korištene ograničenjima(indirektno animiranje). Ove mete mogu onda kontrolirati vrijednosti vlasnika ograničenja, što dovodi do indirektnog animiranja vlasnika kroz animiranje meta. Još jedna korisna karakteristika ograničenja je da se može zadati koliki utjecaj(*influence*) određeni ograničivač može imati na određenu kost. Ograničivači mogu omogućiti da oči teniskog igrača prate tenisku loptu tijekom kretnje, omogućuju da se kotači busa rotiraju simultano, omogućuju ruci da lakše primi ručku mača te da mač potakne ruku da se miče sa zamahom itd. Ograničivači su odličan način da se doda sofisticiranost i kompleksnost rigu, ali treba biti pažljiv kada se dodaje više vrsta kako ne bi došlo do konfuzije pri interakciji između njih. Ograničivači U Blenderu rade s objektima i kostima.



Slika 16, Objektne ograničivači

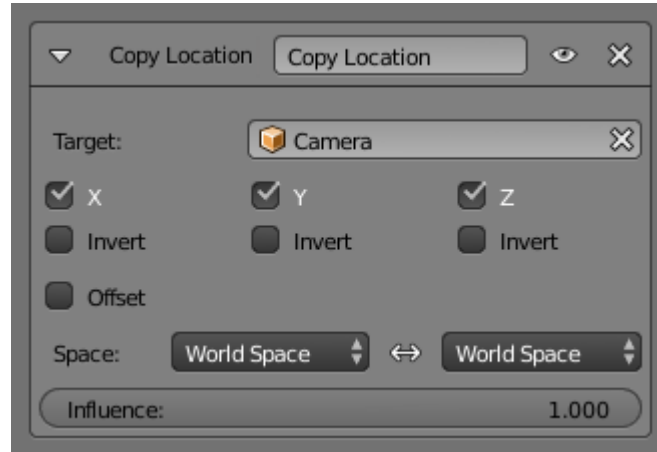


Slika 17, Ograničivači za kosti

Vrste ograničivača koje su bile korištene u ovom projektu su sljedeće: Kopiraj lokaciju(*copy location*), kopiraj rotaciju(*copy location*), kopiraj razmjer(*copy scale*), kopiraj transformaciju(*copy transform*), prigušeno praćenje(*damped track*), produži do(*stretch to*) i inverzna kinematika(*inverse kinematics*). Kopiraj lokaciju, rotaciju, razmjer i transformacija spadaju pod transformacijsku(transform) kategoriju ograničivača, dok prigušeno praćenje, produži i inverzna kinematika spadaju pod prateću(*tracking*) kategoriju.

**Kopiraj lokaciju**(*copy location*) ograničivač forsira vlasnika(kost na kojoj se nalazi ograničivač) da ima istu lokaciju kao meta(kost koja indirektno kontrolira vlasnika).

Opcije:



- Target

Data ID(ime objekta) korišten kako bi selektirao metu ograničivača, i nije funkcionalan(crveno stanje) kada nema selektiranog.

- X,Y,Z

Ove kućice kontroliraju koje su osi aktivne. Invertiranje(invert) kućice omogućuju obrtanje određenih početnih koordinata.

- Offset

Kada je aktivan, ova controla omogućuje vlasnika da se translacija(koristeći svoje trenutne transformacijske postavke), relativno poziciji njegove mete.

- Space

Standardna konverzija između prostora.



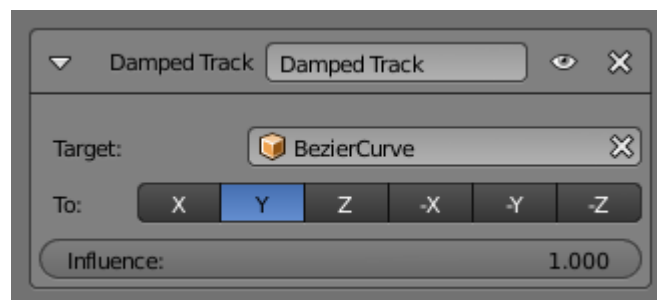
**Kopiraj rotaciju**(*copy rotation*) ograničivač forsira vlasnika da ujednači rotaciju s metom.

**Kopiraj razmjer**(*copy scale*) ograničivač forsira vlasnika da ujednači razmjer s metom.

**Kopiraj transformaciju**(*copy transform*) ograničivač forsira vlasnika da ima iste transformacije kao i meta ( ukratko, kopira lokaciju, rotaciju i razmjer sve u jednom ograničivaču).



**Prigušeno praćenje**(*damped track*) ograničivač, ograničava jednu lokalnu os vlasnika da je uvijek usmjerena prema meti. U drugim 3D Softwareima može se naći s imenom „Look at“ ograničivač.



- Target

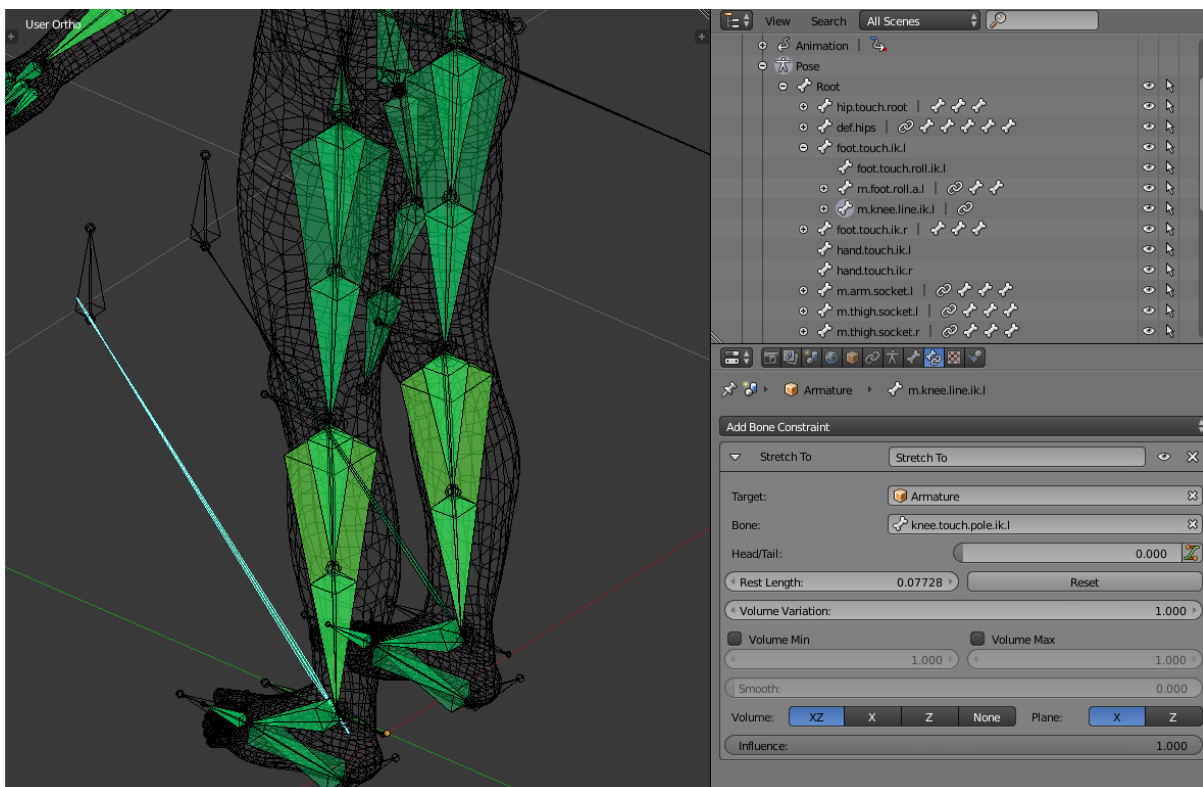
Data ID(ime objekta) korišten kako bi selektirao metu ograničivača, i nije funkcionalan(crveno stanje) kada nema selektiranog.

- To

Jenom kada vlasnik objekt primjeni Damped Track na sebe, mora se izabrati koju os objekta se želi usmjeriti prema meta objektu. Može se izabrati između 6 smjerova po

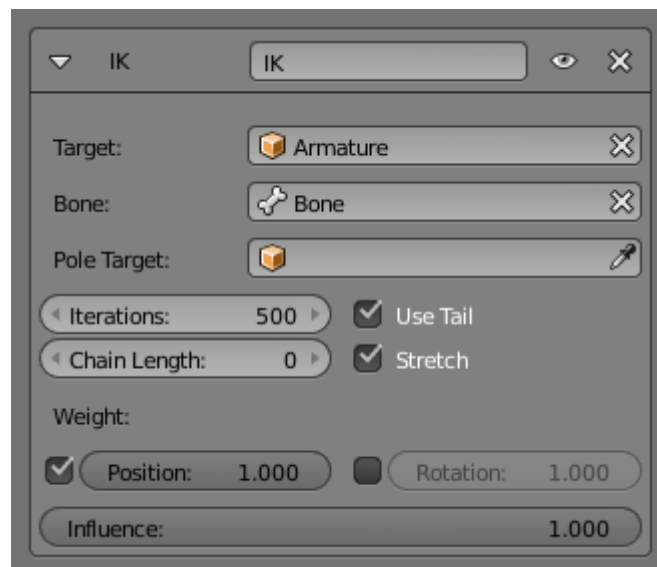
osima (-X,-Y,-Z, X, Y, Z). Negativni smjer osi uzrokuje objekt da je usmjeren u suprotnom smjeru uz duž te osi.

**Produži do**(*stretch to*) ograničivač omogućuje vlasniku da rotira i mijenja razmjer svoje Y osi prema meti. Međutim, pretpostavlja se da će Y os biti prateća i rastezljiva os i ne da opciju da koristite druge kao kod prethodnog ograničivača. Na kosturu modela se koristi više kao vizualni usmjerivač.



Slika 18, Označena kost(tirkizne boje) sa stretch to ograničivačem

**Inverzna kinematika**(inverse kinematics) ograničivač implementira inverznu kinematiku za armaturu. Ograničivač ove vrste je isključivo za kosti. Kako bi brzo postavili IK ograničenje s metom, selektira se kost u poznoj metodi(*pose mode*), te se pritisne Shift+I. [8]



#### Target

- Identifikacija korištena za selekciju armature.

#### Pole Target

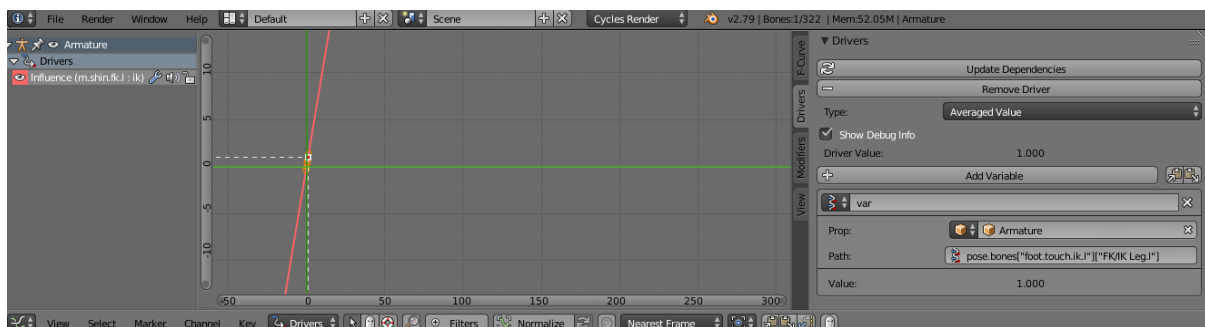
- Objekt za kontrolu smjera rotacije lanca kostiju.

#### Chain Length

- Broj kostiju koje su uključene u IK postavku. Kada je postavljeno na 0, uključuje sve kosti.

### 3.3.3 Upravljači

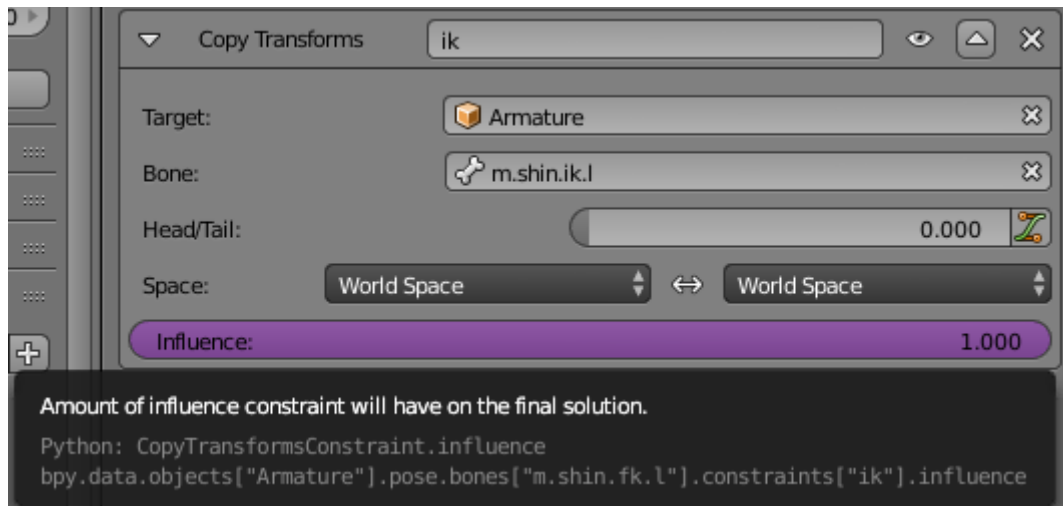
Upravljači su skripte čija je glavna uloga kontrola postavki s drugim postavkama. Primjerice rotacija jednog objekta je kontrolirana s lokacijom drugog objekta. Postoje nekoliko drugačijih načina za dodavanje upravljača u Blenderu. Kada su dodani upravljači, obično se modificiraju unutar grafikonskog obradnika(graph editor) s modalitetom namještenim na upravljači(drivers).



Slika 18, Graph Editor s modalitetom Drivers

Nakon što se postavi bazična armatura(osnovne kosti koje su potrebne za animiranje), može se postaviti složeniji i praktičniji sustav za kontroliranje kostiju. Po defaultu, kosti su namještene na Forward Kinematics(FK) logiku pokretanja lika, no možemo postaviti Inverse Kinematics(IK) sustav koji će nam ubrzati animiranje. Jedan i drugi sustav ima svoje prednosti i mane, tako da dobro dođe da možemo koristiti oboje na istoj armaturi, tu dolaze upravljači kao idealno rješenje. Upravljač će nam služiti kao alat koji će mijenjati iz jednog stanja u drugi, iz FK u IK. Nakon što postavimo FK i IK kosti, koje spadaju pod kontrolne kosti ili mehaničke kosti, možemo postaviti ograničivač kopiraj transformaciju, koji služi da poveže sustave, tako da FK prati IK i zatim dodamo upravljač na copy transform da imamo opciju da mijenjamo između jednog ili drugog sustava, odnosno da se samo IK kosti miču kada je vrijednost 1 a kada je vrijednost 0 onda se miču FK kosti. Kako bi ovo bilo moguće, jedan i drugi sustav moraju biti povezani s jednom zasebnom kosti koja ima ulogu poluge, u ovom slučaju je kost iznad stopala. Na njoj se izabere prilagođene karakteristike(*custom properties*) i stisne se dodaj, zatim se pojavi kontroler koji se pomiče od 0 do 1, kako bi mijenjao iz jednog stanja u drugo. Zatim se dodaje driver za svaku kost kojoj je potrebno (primjerice, kosti

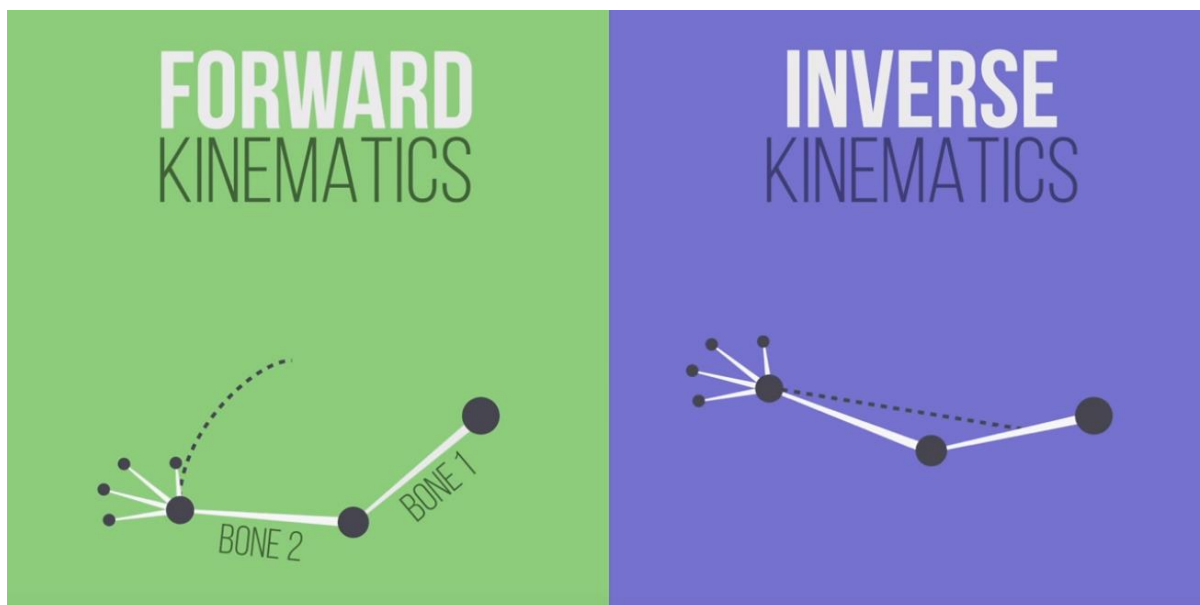
u nogama trebaju imati driver kako bi sustavi funkcionirali). Nakon što su driveri dodani, selektira se kost s karakteristikom za mijenjanje stanja te se kopira putanja podataka za tu kost(*copy data path*) te se nju unese u Graph editor pod prazno polje putanja ili *path* i tako ponoviti za svaki driver.



*Slika 20, Ljubičasta boja predstavlja dodani driver*

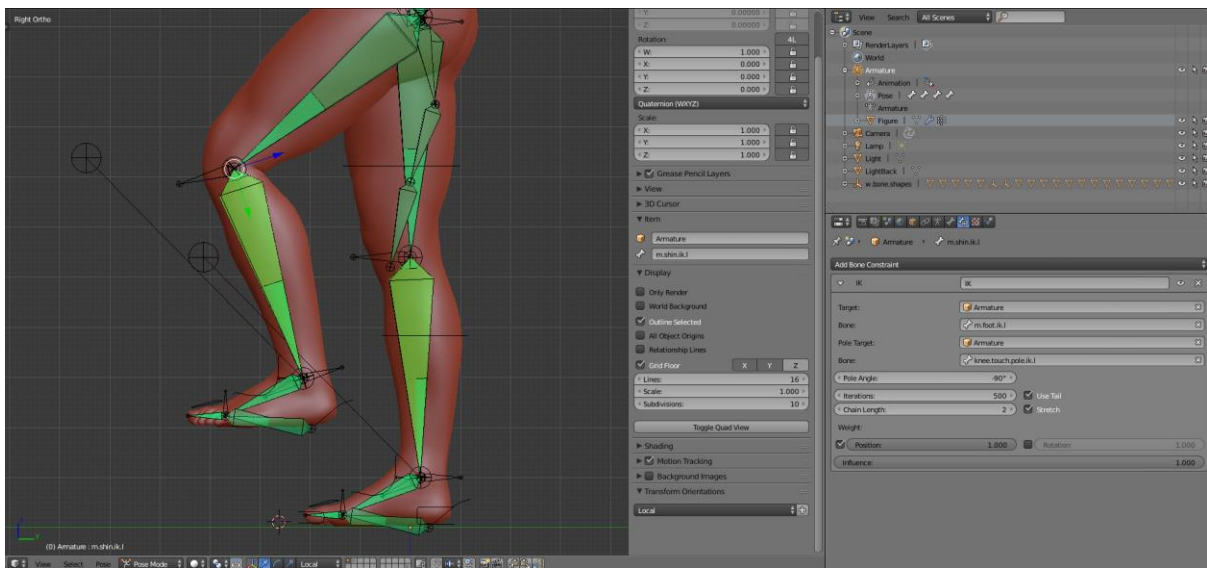
### 3.4 Prednja(FK) i inverzna(IK) kinematika

Tijekom postavljanja osnovnih kostiju, stvara se hijerarhija među tim kostima. Prva kost ili roditelj kost diktira rotaciju svih ostalih kostiju koje su konektirane s prvom kosti, te kosti su „djeca“ prve kosti, te prate sve promijene prve kosti. Lokacija, rotacija i veličina se kopira s roditelj kosti na djecu. Korisnik treba svaku kost zasebno namještati ili pustiti da roditelj kost diktira promjene. Taj sistem se naziva prednja kinematika(FK), putanja kretnje je u luku, daje mogućnost transformiranja svake kosti u lancu. Forward kinematics je koristan tijekom namještanja ruku za animaciju. Dok IK ima kretnje u linearnoj putanji dajući korisniku mogućnost animiranja više prirodnijih kretnji. IK radi računanje kretnji umjesto nas, kosti prate određenu kost koja se nalazi na kraju lanca te se djeca prilagođavaju roditelju ovisno o smjeru. S IK se gubi mogućnost rada s individualnim kostima, samo glavna kost se može micati. IK je jako koristan kod animiranja nogu.



*Slika 21, Razlika kretnja između FK i IK sustava*

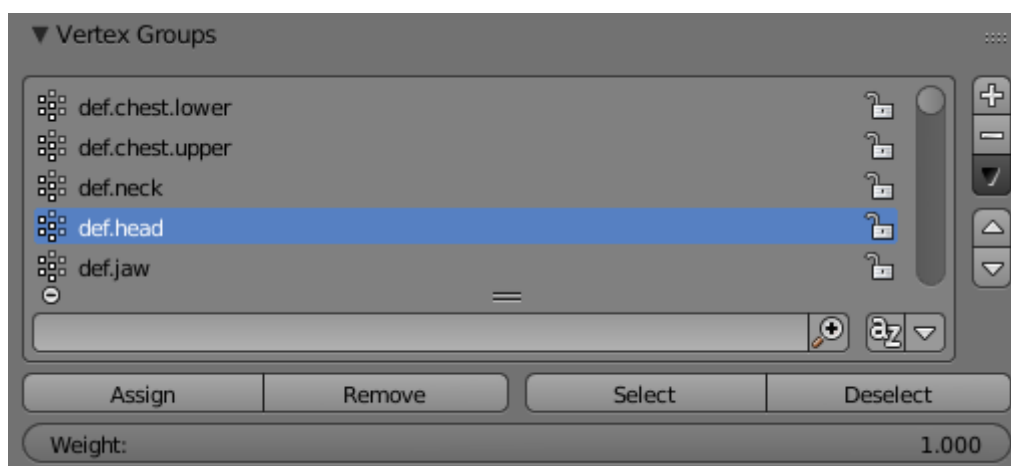
Međutim animiranje se većinom radi iz bočne perspektive te kako bi IK radio kako spada, prvo se odredi blokada na Y i Z osi kako bi se kosti samo micale po X osi te se mora se još dodati meta(*pole target*) kako bi usmjeravao kosti u pravom smjeru. Kost na kojoj se nalazi IK ograničenje ima 2 prazna polja u koje se upisuju imena za zadnju kost u lancu te za metu. Kada se odredi zadnja kost u lancu, ona diktira kretanje ostalih kostiju, no meta kada se selektira ona određuje smjer kretnje. Bez pole targeta, lanac bi bježao ne kontrolirano po X osi, uz put nam daje bolju kontrolu smjera kada gledamo iz prednje perspektive. Pole target ne smije imati roditelja među kostima lanca, mora biti zaseban objekt kako bi ovo funkcioniralo.



Slika 22, Funkcionalni IK sustav za noge

## 3.5 Vertex grupe

Grupe točaka (*Vertex Groups*) su primarno korištene za grupiranje i označavanje točaka koje pripadaju dijelu mreže objekta. Moguće je dodijeliti drugačije težinske vrijednosti (raspon je od 0.0 do 1.0) određenim točkama unutar grupe. Vertex grupe se najčešće koriste za Armature, odnosno kod rada na kostima.

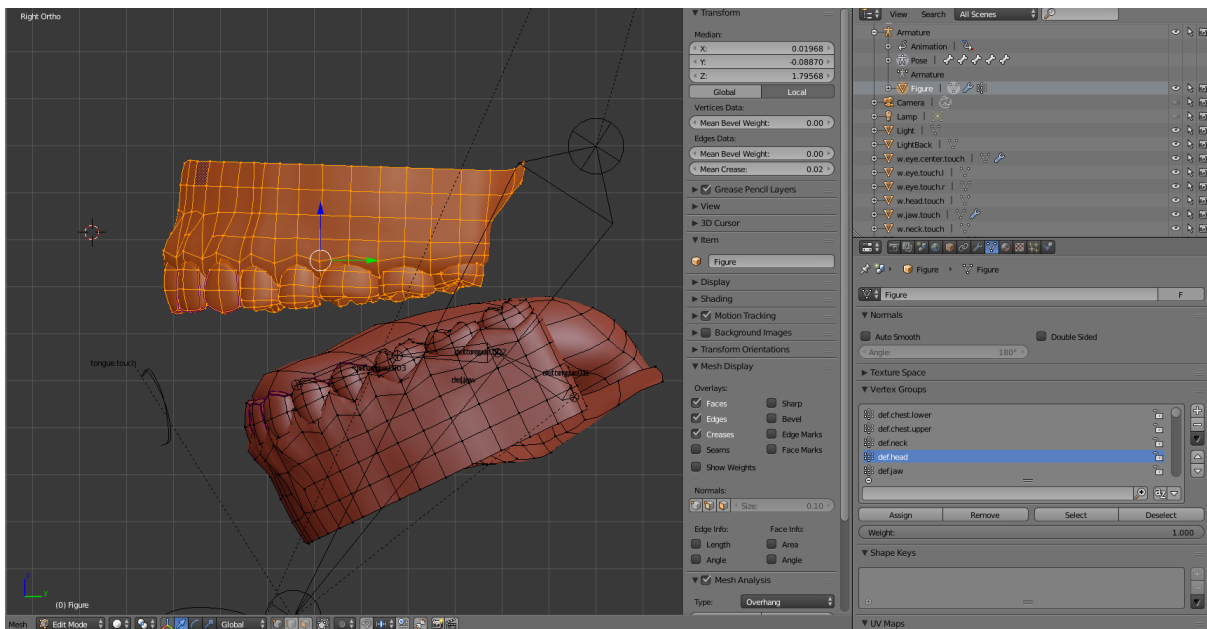


*Slika 23, Vertex grupe dodijeljene kostima*

Jedna od tipičnih scenarija gdje se koriste je tijekom „skinning“ faze. Skinning je dodjela dijela mreže na određenu kost kako bi dobili bolje i realističnije deformacije površine 3D modela. Kada se kreira vertex grupa, pojavi se grupa unutar Vertex Group liste s imenom po želji korisnika. Dodijeli (Assign) gumb omogućuje da se selektirane točke pridruže određenoj grupi na listi, dok Ukloni (Remove) radi suprotno, uklanja selektirane točke iz određene grupe. Selektiraj (Select) služi kako bi prikazao u 3D pregledniku koje su točke pridružene toj specifičnoj grupi, dok deselect miče selekciju. S desne strane Vertex Group prozora se nalazi tamno siva strelica koja kada se klikne otvara prozor s raznim dodatnim opcijama i postavkama, jedna od njih koja je jako korisna bila u ovom projektu je Ukloni Iz Svih Grupa (Remove from All Groups), ukratko, selektirane točke se miču iz svih grupa. Primjer gdje je bilo od velike koristi je bilo namještanje vertex grupa kod kostiju koje su kontrolirale kretanje čeljusti.



Tijekom skinning procesa(koji je dio With Automatic Weights procesa), zna doći do grešaka kada se odabere opcija da računalo automatski stvori vertex grupe za svaku kost u armaturi. Ove probleme je preporučljivo srediti prije nego što se počne slagati naprednije armature poput IK i FK. Dobar primjer čestih grešaka koji se događaju kod rada na čeljusti je da se zubi gornjeg dijela čeljusti dodijele kostima donje čeljusti i obratno. Kako bi se to lagano korigiralo, problematične točke se selektiraju, zatim ih se ukloni iz svih grupa te se selektira grupa na kojoj su potrebni te se stisne Assign. Kako bi si taj proces olakšali, okolni mesh ili mrežu sakrijmo(*Hide*) te izoliramo čeljust kako bi se lakše selektirale problematične točke. [9] [10] [11]

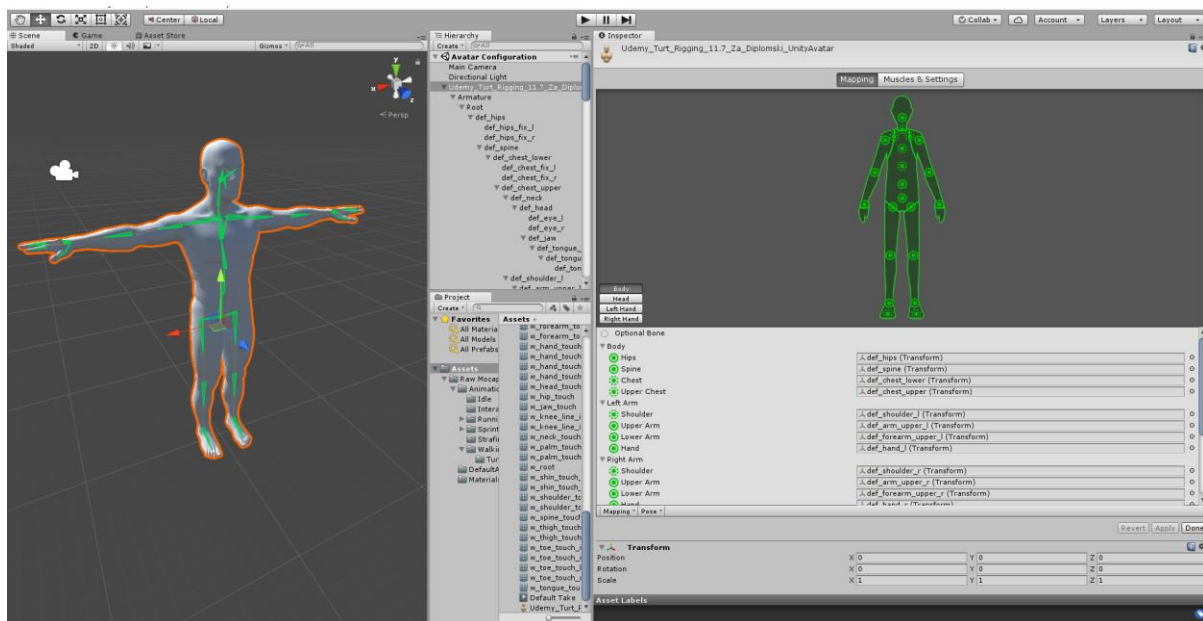


*Slika 24, Rad na čeljusti s grupama*

## 4. Rezultati i rasprava

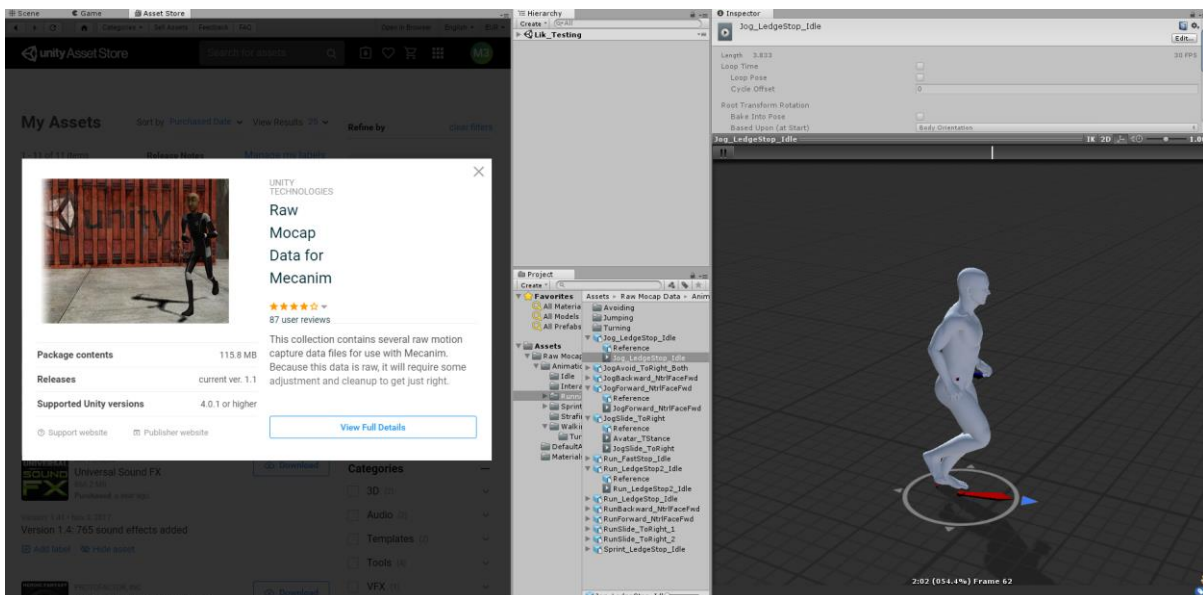
### 4.1. Ubacivanje modela u Unity

Zadnja faza je priprema modela za prebacivanje u Unity, u blenderu se prilagođavaju lokacija, rotacija i razmjer tako da se stisne Ctrl + A, te se primjeni (*apply*) lokacija, rotacija i razmjer kako model ne bi stvarao poteškoće tijekom konverzije, čineći ovaj korak model se postavlja u difoltno stanje. Unity kada konvertira model čita vrijednosti rotacije, lokacije i razmjera i koristi ih. Zadnji korak je pobrisati sve nepotrebne stvari osim modela i njegove armature, kamera, osvjetljenje i ostali objekti su nepotrebni. Nakon što je sve spremno, otvara se Unity i stvara se novi 3D projekt te se stvara folder sredstva(Assets) u koji se drag and dropa 3D model. Kada se očita, klikne se na model, otvara se prozor koji daje opcije animacijskih tipova, postavi se na humanoid te se klikne Apply. Tim korakom, Unity mapira i pregledava koje kosti su prisutne i koje će se koristiti za animiranje. Zatim se klikne na konfiguriraj (*Configure*), koji prikazuje gotovo mapiranje kostiju i prikazuje koje kosti se koriste na kojoj lokaciji.



Slika 24, Mapping u Unity sučelju

Unity neće koristiti sve kosti koje su na raspolaganju, primjerice donji dio ruke(Lower Arm) će koristiti jednu umjesto dvije kosti koje su na izboru, pošto nisu obje potrebne. Kako se pregledava kostur koji je Unity stvorio za lika, primijeti se da je sve kako spada osim kod donjeg dijela tijela, specifično kod nogu. Zna se dogoditi da Unity izabere krivu kost umjesto optimalne kosti. Primjerice kod noge, gornji dio noge(Upper Leg) radi kako spada, izabrana je viša bedrena kost kao kontrolna kost za to područje, dok za donji dio noge(Lower Leg) je izabrana donja bedrena kost (bedro i cjevanica imaju po 2 kosti) umjesto gornje kosti cjevanice, dok u isto vrijeme cjevanica kontrolira stopalo, to neće funkcionirati kako spada i mora se ručno promijeniti. Promjena je jednostavna, u polje gdje je upisana kriva kost, izabere se kost s liste koja paše toj lokaciji. Nakon što su sve kosti pregledane, i korisnik je zadovoljan s rezultatom, ispod liste se klikne gump Apply, pričekava se Unity da obavi svoje operacije te kada je gotov, stisne se Done, te je model spreman za korištenje. Skidanje besplatnog paketa s animacijama je sljedeći korak (Raw Mocap Data for Mecanim). Ukratko, to je kolekcija podataka snimaka gibanja(Motion Capture) ljudskog tijela u raznim kretnjama. Pošto je raw materijal, potrebno je malo prilagođavanja, no to je obavljeno u prošlom koraku. Kako bi koristili animacije, samo drag and dropamo izabranu animaciju u prozor gdje se nalazi model i automatski će ju primijeniti. [12] [13] [14]



Slika 25, Korištenje animacija na modelu

Završni produkt je uspješno ubačen i dobro reagira s Unity animacijama. Nakon što su isprobane sve osnovne animacije (trčanje, hodanje, skakanje, okretanje...), lik je spreman za korištenje unutar Unity baziranih video igri. Tijekom prijašnjih testiranja, primijetile su se tipični problemi neugodnih deformacija na tijelu modela, pogotovo kod trupa i zglobova. Kako bi se takve greške izbjegle i ispravile, potrebno je dobro pregledati kako reagira svaka kost sa svojim dodijeljenim sektorom 3D modela. Zna se dogoditi da dvije susjedne kosti dijele preveliki postotak verteksa unutar svojih zasebnih verteks grupa. Normalno, to se ne može uvijek izbjeći u potpunosti no dobro je držati neku razinu urednosti kada se radi na verteks grupama, kako bi se umanjile deformacije na 3D modelu. Pozicija kostiju je isto tako izuzetno važna, kako bi Blender pospajao određene kosti s dijelovima tijela na što efikasniji način. Primjerice, ako kost viri iz oka 3D modela, Blender će automatski povući i okolne točke oko oka te ih pridružiti toj verteks grupi, poput zraka koje isijavaju iz kosti u svim smjerovima sve dok ne pogode rub 3D modela. [15]

## 5. Zaključak

Cijeli proces izrade 3D Modela je dosta zahtjevan i dug proces, no uz prave savjete i informacije, izrada se može poprilično ubrzati i izbjeći moguće vremenski skupe greške te i u nekim slučajevima, ponavljanje cijelih koraka, poput rigging procesa. Čitajući o kreativnom procesu izrade likova unutar raznih velikih firmi koje se bave video igrama, primijeti se nevjerovatna prednost velikih timova, podjela posla i usavršavanje svakog stupnja izrade je ključna za napredak i ostvarivanje boljih i kvalitetnijih produkata u manje vremena sa zadovoljavajućim rezultatom. Male firme pate upravo zato što nemaju taj luksuz da se podijeli posao na više specijaliziranih sektora. Sloboda pojedinaca da postanu majstori u određenom koraku kreativnog procesa je uspjeh za firmu i pojedinca. No unatoč tome, trenutna situacija u industriji nije nikada bila bolja za manje studije da se rašire i pokrenu svoje uspješne naslove zbog sve veće povezanosti i kompatibilnosti među raznim programima koji se koriste pri izradi 3D likova i okruženja. Svaki stupanj izrade je jednako važan, bilo to sama izrada modela ili rigging proces, svaki korak se mora izvesti sa što većom preciznošću kako bi finalni produkt bio spreman za animiranje. Samo animiranje isto je ubrzano, ako nisu potrebne specifične kretnje koje se mogu ručno raditi u Blenderu, Unity ima popriličan broj paketa animacija koje se mogu iskoristiti za likove koji su stvoreni u Blenderu. Unity svojom jednostavnošću i nevjerovatnim izborom besplatnih i jeftinih asset paketa, čini idealno sučelje za početak testiranja likova. Faza rada u Unityju ovisi o dobro provedenim fazama u Blenderu. Animiranje u Blenderu je potrebno samo kada se radi o specifičnim animacijama koje se ne mogu naći u Unity sučelju ili ako je cilj animirati kompleksnije oblike modela s više udova, onda je potrebno koristiti Blender. Kako raste broj računalnih korisnika, tako i sve veći broj entuzijasta koji se bave ovim područjem, koji će moći uz minimalne troškove postići puno više pri izradi nego generacije prije nas. Blender je besplatan te se firma većinski razvija i raste preko donacija, no njezin utjecaj na gaming industriju je sve veći tako da interes sponzora raste svakim danom. Blender i Unity otvorili su vrata za novu generaciju game developera koji će promijeniti tržište. Što se tiče učenja i prakse s Unity i Blender sučeljima, internet je pun raznih korisnih

knjiga, stručnih članaka, zanimljivih YouTube kanala koji u današnje doba se pokazuju kao jedan od boljih izvora za istraživanje i učenje o ovim jako moćnim sučeljima. Također postoje profesionalni tečajevi koji se mogu kupiti za nisku cijenu preko interneta koji su se pokazali jako korisni. Uz ovakve moćne alate, ambiciozni kreativci mogu dogurati daleko.

## 6.Literatura

1. A Guide to Building a 3D Game Character, Elisabet Heikkila, 2017
2. Blender 3D: Noob to Pro, 2014
3. Beginning Blender, Lance Flavell, 2010
4. Modeling the Human Body in 3D: Data Registration and Human Shape Representation, Aggeliki Tsoli, 2014
5. Character Development in Blender 2.5, Jonathan Williamson, 2011
6. <https://www.udemy.com/blender-character-modeling-for-beginners-hd/learn/lecture/>
7. <https://www.blender.org/manual/>
8. [http://builder.openhmd.net/blender-hmd-viewport-temp/rigging/constraints/tracking/ik\\_solver.html](http://builder.openhmd.net/blender-hmd-viewport-temp/rigging/constraints/tracking/ik_solver.html)
9. [https://easyblend.org/html/modeling/meshes/vertex\\_groups/vertex\\_groups.html](https://easyblend.org/html/modeling/meshes/vertex_groups/vertex_groups.html)
10. [https://docs.blender.org/manual/en/latest/modeling/meshes/properties/vertex\\_groups/index.html](https://docs.blender.org/manual/en/latest/modeling/meshes/properties/vertex_groups/index.html)
11. Blender 2.5 Character Animation Cookbook, 2011
12. Unity 5.x By Example, Alan Thorn, 2016
13. Unity 5.x Cookbook, Matt Smith, 2015
14. <http://docs.unity3d.com/Manual/index.html>
15. <https://www.facebook.com/groups/tenthousandhours/>