

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1257

ANIMIRANE KINEMATIČKE STRUKTURE

Kristijan Šimunić

Zagreb, lipanj 2010.

Sadržaj

1. Uvod.....	1
2. Animacija	2
2.1. Tradicionalna animacija	2
2.2. Računalna animacija.....	3
3. Kinematika	6
3.1. Unaprijedna kinematika.....	9
3.2. Inverzna kinematika	10
4. Implementacija.....	17
4.1. Uvod.....	17
4.2. Referentne slike	18
4.3. Izgradnja mreže modela (MESH).....	19
4.4. Dodavanje materijala	29
4.5. Dodavanje kose	29
4.6. Izgradnja kostura	30
4.7. Povezivanje kostura i mreže	33
4.8. Inverzna kinematika	34
4.9. Animacija	35
4.10. Učitavanje modela u Ogre	38
5. Zaključak.....	40
Literatura	41
Sažetak	42
Summary.....	43

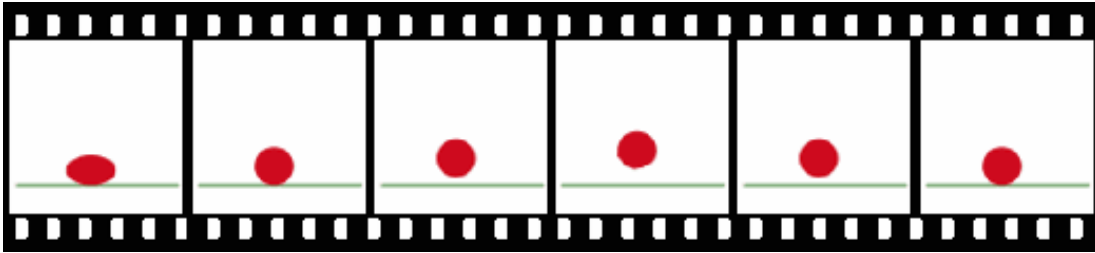
1. Uvod

Svakodnevno se susrećemo s animacijom. Međutim znamo li kako se ona ostvaruje, koje su tehnike njezine izrade, koliko posla iziskuje, itd. Ovim radom odgovorit ćemo na ta pitanja. Prikazat ćemo cijeli postupak stvaranja animacije kretanja čovjeka, točnije hodanja. Najprije ćemo se upoznati s pojmom animacije, tipovima animacije te tehnikama animiranja. Budući da ćemo koristiti skeletalnu tehniku animiranja za ostvarivanje kretanja čovjeka, moramo razumjeti način na koji se ona ostvaruje. Iz toga razloga nakon animacije govorit ćemo o kinematici tj. hijerarhijskim kinematičkim lancima, direktnoj i inverznoj kinematici. Nakon toga slijedi poglavlje o samom stvaranju animacije. Koristit ćemo se programski alatom Blender iz razloga što je besplatan, a opet kvalitetan alat za izradu 3D modela i animacija. Krenut ćemo s skicama zamišljenog modela „nabildanog“ muškaraca te ga pokušati ostvariti prema zamisli. Najprije ćemo napraviti vanjski oblik (mrežu), obojati ga i dodati kosu. Da bi ga mogli animirati morat ćemo napraviti kostur, dodati ga mreži te omogućiti korištenje inverzne kinematike radi lakšeg stvaranja animacije. Nakon toga slijedi stvaranje same animacije te učitavanje modela u Ogre (eng. „Open-source 3D Graphic Rendering Engine“).

Nakon proučavanja ovoga rada čitatelj će se upoznati s principima stvaranja animacije i posjedovati dovoljno znanja za njihovo razumijevanje, ako ne i stvaranje.

2. Animacija

Animacija je danas širom rasprostranjena. Prisutna je u filmovima, računalnim igrama, televizijskim programima, internetskim stranicama, na svim uređajima koji posjeduju ekran. Razvojem tehnologije i tehnike animiranja razvijaju se i same animacije tj. postaju kvalitetnije, realnije.



Slika 2.1 Animacija

Animacija je brzi prikaz slijeda dvodimenzionalnih slika, odnosno poza trodimenzionalnih modela kako bi ostvarili iluziju pokreta. Zbog fenomena tromosti vida, dolazi do iluzije kretanja, u čemu leži osnova stvaranja animacije.

2.1. Tradicionalna animacija

Tradicionalna animacija („celuloidna“ animacija) je najviše korištena u filmovima u 20-tom stoljeću. Pojedine sličice tradicionalno animiranog filma su fotografije ili crteži, koje su prvo crtane na papiru. Da bi se dobila iluzija kretanja, svaki crtež se malo razlikuje od onog prethodnog. Crteži su projicirani ili fotokopirani na transparentne acetatne listove zvane celuloid, koji su popunjeni bojama na suprotnoj strani od crteža (Slika 2.1.1a).



Slika 2.1.1 a) Tradicionalna animacija b) Stop animacija

Stop animacijom (Stop-motion) opisuje se animacija koja nastaje fizičkim manipuliranjem stvarnih objekata fotografirajući ih sličicu po sličicu, stvarajući tako dojam kretanja. Postoje mnogi tipovi stop animacije, obično nazvani po tipu materijala na kojem se stvara animacija (glinena, cutout, siluetna, grafička, model, go, objektna, lutkarska animacija) (Slika 2.1.1b).

2.2. Računalna animacija

Računalna animacija je umijeće kreiranja pokretnih slika upotrebom računala. To je potkategorija računalne grafike i animacije te ju možemo podijeliti na 2D i 3D animaciju.

2D animacija

Likovi su stvoreni na računalu koristeći 2D bitmap grafiku ili 2D vektorsku grafiku. Ovo uključuje automatizirane računalne verzije tradicionalnih animacijskih tehnika poput *tweeninga* (ubacivanja međusličica), *morphinga* (transformacije), *onion skinninga*, interpolirane rotoskopije itd. Primjeri su analogna računalna, flash i PowerPoint animacija.



Slika 2.2.1 2D animacija

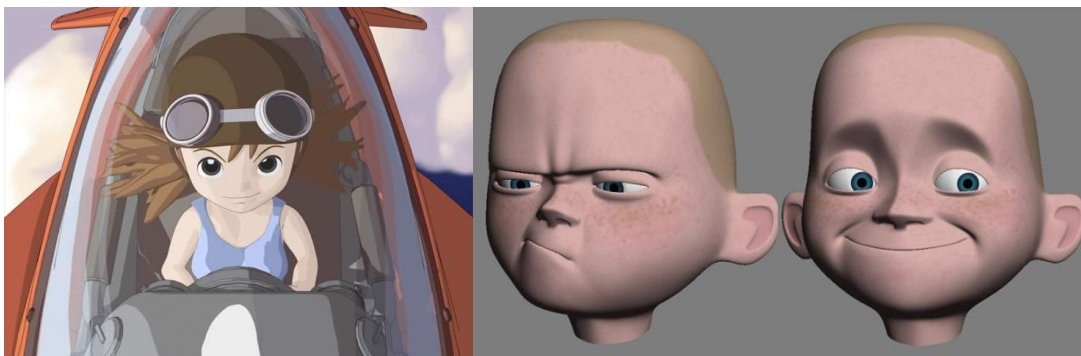
3D animacija

Ukratko ćemo objasniti neke od 3D tehnika animacije koje se danas koriste u praksi.

Cel-sjenčanje (cel-shading ili toon shading) je vrsta nefotorealističnog renderiranja s namjenom da računalna grafika izgleda kao da je crtana rukom. Cel-sjenčanje se koristi da bi se oponašao stil stripova ili animiranih

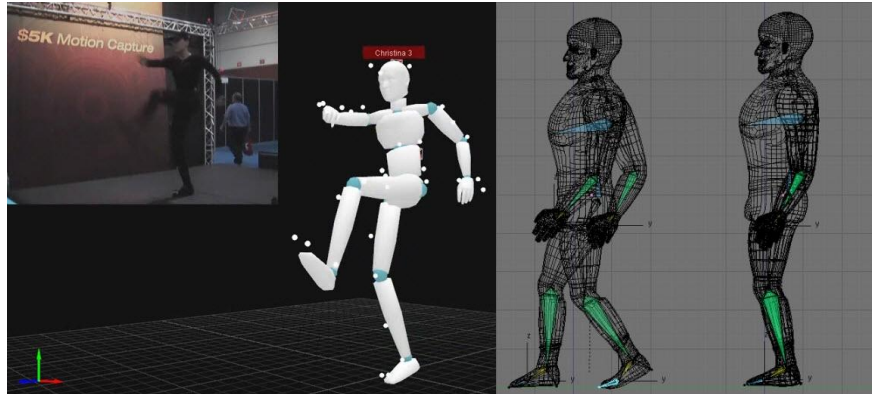
filmova. Iako rezultat cel-sjenčanja ima jednostavan izgled ručnog crteža, proces je prilično kompleksan.

Animacija izobličavanjem (Morph target) je metoda koja u svakoj ključnoj sličici (keyframe) pamti lokacije vrhova koji se mijenjaju prilikom mijenjanja izgleda mreže. Ovisno o vrsti prikaza, vrhovi će ići po stazama da bi popunili vrijeme između ključnih sličica, ili će se direktno prebacivati iz jedne lokacije u drugu lokaciju. Prvi pristup daje ljepše, a time i bolje rezultate te se zato i češće koristi. Ova metoda se često koristi pri animaciji tkanina, kože i izraza lica. Negativna strana ove metode je što je prilično vremenski zahtjevna, kako za izradu tako i za računalo.



Slika 2.2.2 a) Cel-sjenčanje b) Animacija izobličavanjem

Hvatanje pokreta (Motion capture) je proces snimanja pokreta iz stvarnog svijeta i njihovo prevođenje u digitalni oblik praćenjem ključnih točaka u vremenu. Na objekt čiji se pokreti žele uhvatiti stavljaju se markeri na ključne točke. Oni mogu biti zvučni, inercijski, LED, magnetski, reflektivni ili kombinacija nekih od navedenih. Prate se u dva puta većoj frekvenciji od frekvencije željenog pokreta u milimetarskoj preciznosti. Dobiveni podaci se obrađuju te se izračunavaju pozicije, kutovi, brzine i akceleracije omogućavajući preciznu digitalnu reprezentaciju pokreta. Prednosti korištenja ovog načina animacije su rezultati u realnom vremenu, realnost pokreta, količina posla neovisna o kompleksnosti i dužini pokreta te velike količine podataka u kratkom vremenu. Nedostatak su potreba opreme, ljudi i prostora. U slučaju pogreške potrebno je ponoviti cijeli postupak, nije moguće hvatanje nemogućih pokreta, modeli moraju biti istih proporcija inače će doći do „čudnih“ pokreta.



Slika 2.5 a) Motion capture animacija b) Skeletalna animacija

Skeletalna animacija se najčešće koristi za animaciju kralježnjaka ili u našem slučaju čovjeka, u kojoj je lik predstavljen od dva dijela: mreže (mesh) koja se prikazuje i kostura (skeleton). Ova tehnika se koristi tako da se konstruiraju kosti. Svaka kost ima trodimenzionalnu transformaciju (uključuje poziciju, veličinu i orijentaciju), i opcionalno „roditeljsku“ kost. Sve kosti zajedno čine kostur koje su povezane hijerarhijski u obliku stabla. Budući da ćemo se mi koristiti ovom tehnikom animacije detaljnije ćemo objasniti ovaj tip animacije u narednim poglavljima.

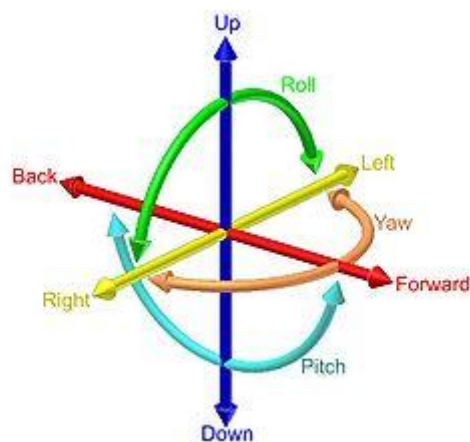
3. Kinematika

Pojam kinematike preuzeli smo iz terminologije korištene u robotici. U robotici njime označavamo granu mehanike koja se bavi proučavanjem gibanja tijela ne uzimajući u obzir sile pod čijim se djelovanjem to gibanje zbiva. Za to je zadužena dinamika koja zglobovima i vezama dodjeljuje masu i na njih djeluje silom, te analizira utjecaje sile na položaj sustava.

Dakle možemo reći da je kinematika grana grafičkog oblikovanja modela, gdje vodimo računa o pokretljivosti, njezinim ograničenjima i povezanosti različitih dijelova modela.

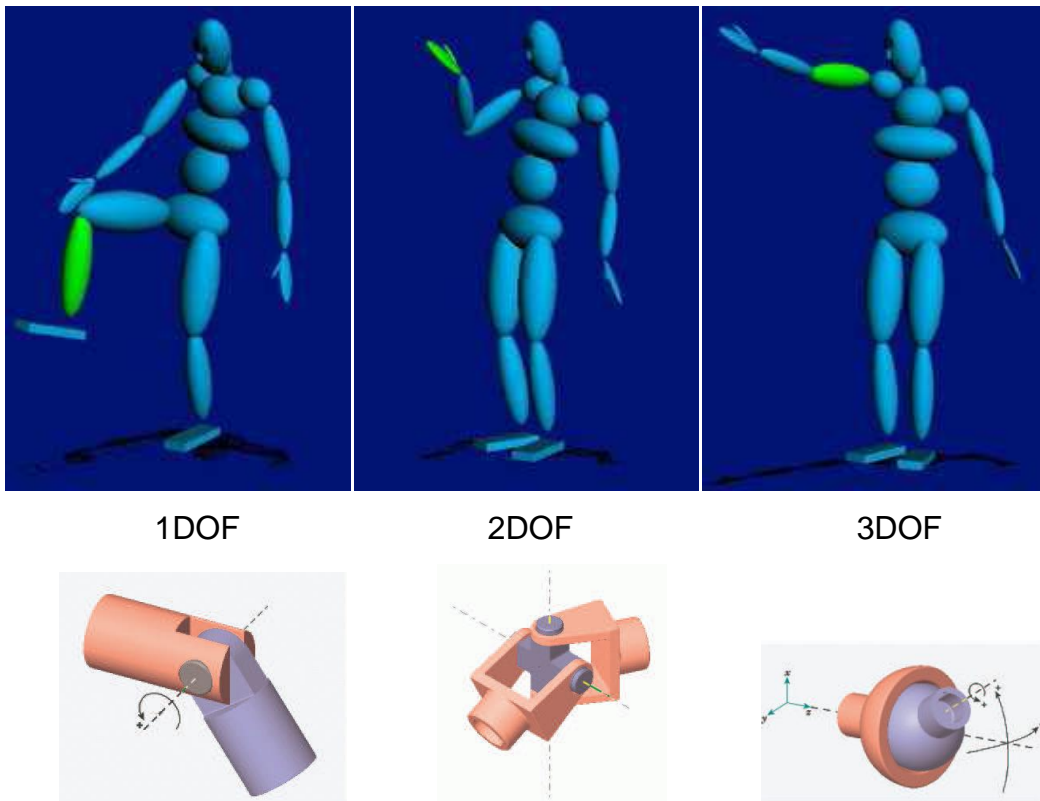
Model kinematičkih lanaca je jedan od jednostavnijih modela pristupa animaciji. Kinematički lanci se prirodno predstavljaju hijerarhijskim odnosom između članova, u našem slučaju kostiju, formiranih u stablastu strukturu. U modelu čovjeka, čvorovi stabla, segmenti su npr. potkoljenica, nadlaktica, ključna kost, itd. Segmenti su kruti elementi, dok grane imaju ulogu zglobova. Svaki čvor može imati više grana. U modelu čovjeka iz čvora trupa „izlaze“ grane vrata, ramena, itd.

Za svaki zglob razlikujemo stupanj slobode (DOF – eng. „degrees of freedom“). Postoje rotacijski i translacijski stupnjevi slobode u skladu s mogućnostima kretanja. Po ograničenju stupnja slobode razlikujemo 3 stupnja slobode kod rotacije. To su rotacija oko X, Y i Z osi. Isto tako razlikujemo i 3 stupnja slobode kod translacije po X, Y i Z osi (Slika 3.1).



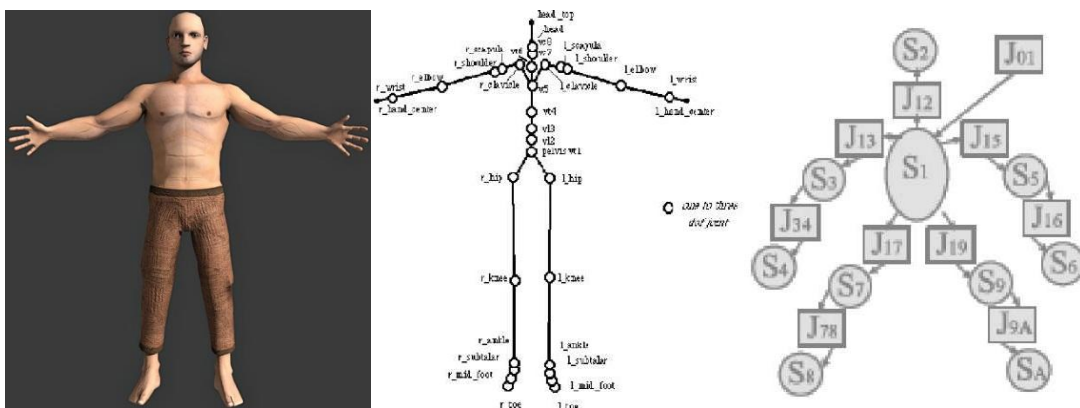
Slika 3.1 Rotacijski i translacijski stupnjevi slobode

Kod kostura čovjeka razlikujemo zglobove s maksimalno tri rotacijska stupnja slobode kao što je prikazano na slici ispod.



Slika 3.2 Rotacijski stupnjevi slobode kod čovjeka

Pri izgradnji hijerarhijskog modela, određeni objekt moramo prikazati pomoću grafa stablaste strukture. To ćemo ostvariti pomoću apstraktne reprezentacije našeg objekta. Na primjeru čovjeka Slika 3.3.



Slika 3.3 a) Model b) Apstraktna reprezentacija modela c) Graf stabla

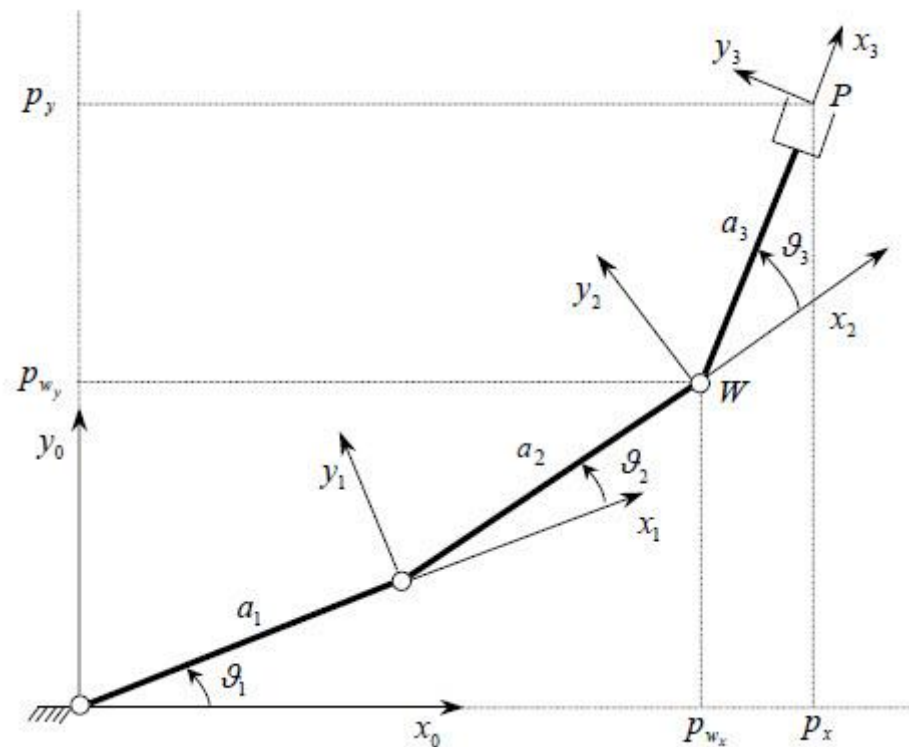
Apstraktnu reprezentaciju modela radimo tako da označimo bitne točke modela, poput zglobova te ih povežemo nepokretnim dijelovima, čvorovima. Zglobu dodjeljujemo isti stupanj slobode koji ima i zglob u stvarnosti. Tako na primjer, koljeno kao zglob, modeliramo s granom stupnja slobode jedan (1 DOF). Vrat kao zglob, predstavljamo granom stupnja slobode tri (3DOF). Na slici 3.3b vidimo jedan primjer apstraktne reprezentacije modela. Sada radimo graf stabla prema izgledu te apstrakcije.

Izradu grafa provodimo pomoću transformacija iz izvornog čvora. Uzmimo, npr. da nam je segment trupa izvorni čvor. Na Slici 3.3b to je segment S1. Iz tog segmenta izvodimo, npr. segmente bedrene kosti, S7 i S9. Bitno je uočiti da segmenti izvedeni iz roditeljskog segmenta ovise o poziciji i obliku roditelja. Iz segmenta bedrene kosti S7 izveden je segment potkoljenica, S8. Pozicija ovog segmenta ovisit će o poziciji i obliku segmenta S7. Kako segment S7 ovisi o segmentu S1, možemo zaključiti da pozicija segmenta S8 ustvari ovisi i o segmentu S7 i o segmentu S1. Ova se pojava naziva *nizanje transformacija*.

Pojmom nizanja transformacija ostvarili smo pasivni oblik hijerarhijskog modela. U tom obliku možemo odrediti i koje se karakteristike nasljeđuju kroz čvorove: da li će segment-dijete imati iste teksture, boju, oblik, itd. kao segment roditelj. Međutim, postavlja se pitanje kako ćemo ostvariti gibanje određenih segmenata, te kakav će biti njihov utjecaj na segmente-djecu, odnosno segmente-roditelje. Tu dolazimo do razdiobe međusobnih utjecaja kinematičkih segmenata na dvije vrste: unaprijedna kinematika, te inverzna kinematika.

3.1. Unaprijedna kinematika

Unaprijedna kinematika za manipulator (manipulator je prikladan termin preuzet iz robotike, a odnosi se na promatrani model koji ima jedan nepomični zglob – bazu, i skup povezanih pomičnih zglobova) zadaje položaje i kutove svih zglobova, te na osnovu njih izračunava položaj krajnje točke manipulatora. Ovo je trivijalan zadatak sa samo jednim rješenjem.



Slika 3.1.1 Unaprijedna kinematika

Kompletno određena poza 3D grafičkog modela specificirana je poznim vektorom (eng. „pose vector“). Poznii vektor čuva pozicije svih zglobova u modelu.

Unaprijedna kinematika, kao princip određivanja položaja jedinica strukture modela, postavlja se kao jednostavno rješenje, ali ne i efikasno. Za ostvarenje pokreta potrebno je mnogo ulaznih podataka, a kao rezultat dobivamo pokrete koji nisu prirodni. Zbog lakšeg modeliranja, te zbog činjenice da ukoliko želimo doći do željene točke s nekim segmentom S_n , moramo postepeno pomicati svaki segment koji je roditelj segmentu S_n , ova je metoda istisnuta od strane inverzne kinematike.

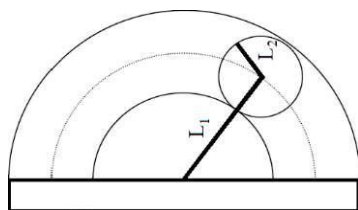
3.2. Inverzna kinematika

Inverzna kinematika radi postupak obrnut unaprijednoj kinematici – za zadanu krajnju točku manipulatora ona pronalazi položaje i kutove svih zglobova takve da manipulator bude u traženoj krajnjoj točki, ako je ista dohvatljiva, ili što je moguće bliže ako nije dohvatljiva. Ovo je mnogo složeniji problem, redovito s beskonačno mnogo rješenja.

Inverzna kinematika je, očito, puno upotrebljivija i zanimljivija od unaprijedne kinematike – za bilo kakvu animacijsku tehniku, puno nam je interesantnije da možemo samo zadati krajnju točku na kojoj mora biti ruka (npr., dohvatiti šalicu) nego da moramo točno navoditi sve položaje zglobova. Isto tako, snimanjem stvarnih pokreta čovjeka 3D senzorima za potrebe analize, dovoljno je snimati samo položaj krajnje točke, a ne položaje svih zglobova.

Za shvaćanje principa rada inverzne kinematike prvo ćemo objasniti pojam dohvatljivog radnog prostora (eng. „reachable workspace”). Na slici 3.2.1 vidimo jednostavan primjer, s označenim radnim prostorom. Tanke pune linije označavaju doseg segmenta S2, dok isprekidana predstavlja doseg segmenta S1. Matematički bi radni prostor mogli opisati sljedećom formulom:

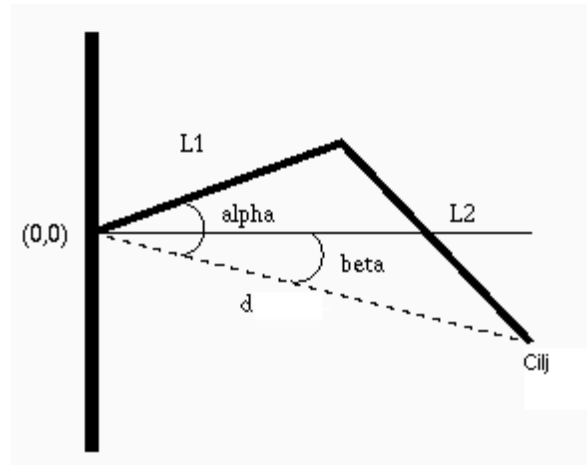
$$L_1 - L_2 \leq \text{radni prostor} \leq L_1 + L_2 \quad (3.2.1)$$



Slika 3.2.1 - Dohvatljivi radni prostor

Važno je napomenuti da neki zglobovi imaju ograničenja u pokretima. Na Slici 3.2.1 čvor L_1 može se pomicati samo 180 stupnjeva. Ovo je vrlo česta pojava u oblikovanju 3D modela.

Za prikaz principa rada inverzne kinematike koristit ćemo najjednostavniji primjer: dva segmenta povezana međusobno i zajedno na čvrstu točku:



Slika 3.2.2 - Princip inverzne kinematike

Segmenti su kruti elementi modela, te imaju fiksnu duljinu. Uz poznate duljine i zadan položaj vrha manipulatora, odredimo nepoznate kutove α i β .

Kut β računamo preko x i y koordinata točke *Cilj*:

$$\cos\beta = \frac{x}{\sqrt{x^2 + y^2}} \quad (3.2.2)$$

Kosinusovim poučkom

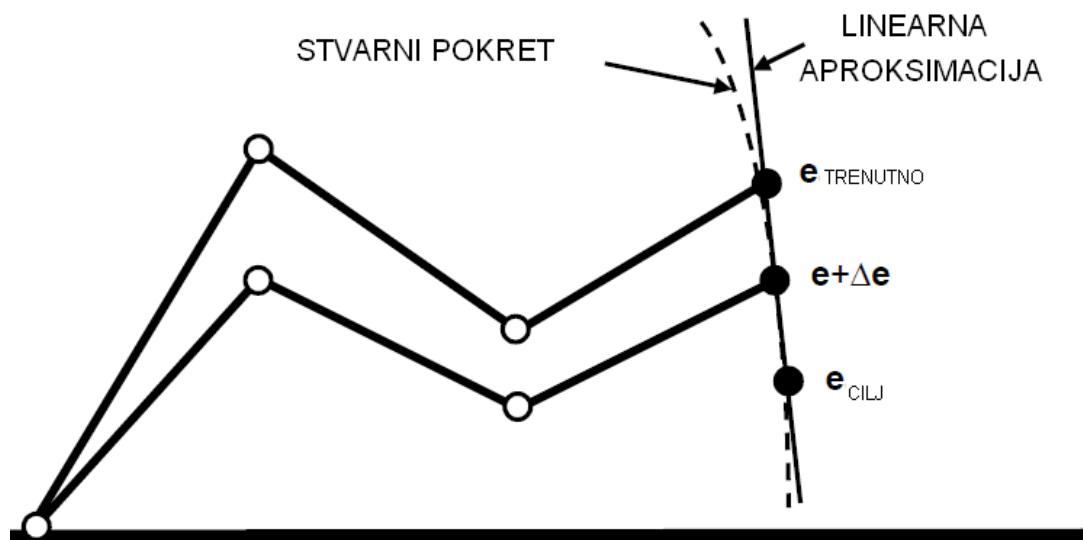
$$c^2 = a^2 + b^2 - 2ab\cos\alpha \quad (3.2.3)$$

i (3.2.2) računamo kut α :

$$\cos(\alpha - \beta) = \frac{L_1^2 + x^2 + y^2 + L_2^2}{2L_1L_2} \quad (3.2.4)$$

Ovim postupkom dobivamo samo dva rješenja. Ona su simetrična u odnosu na liniju koja spaja ishodište lanca i točku cilja. U gore spomenutim primjerima, govorimo o isprekidanoj liniji d. Bitno je napomenuti da se slučaj s dva rješenja pojavljuje samo kada upravljamo modelom s dva čvora. U složenijim primjerima broj rješenja se penje do beskrajno velikih brojki.

Glavna problematika inverzne kinematike, zbog nelinearnosti funkcija sinus i kosinus je teška aproksimacija linearnih pokreta. Linearni pokreti su, ipak, česta pojava u stvarnom svijetu.



Slika 3.2.3 - Odnos linearne aproksimacije i stvarnog pokreta

U nastavku ćemo se pozabaviti nekim češće korištenim rješenjima ovog problem s naglaskom na Jakobijevu matricu.

Jakobijeva matrica – Jakobijan

Jakobijeva matrica funkcije opisuje orijentaciju tangentne ravnine zadane funkcije u određenoj točki. U našem slučaju, Jakobijan je solucija za rješavanje problema linearne aproksimacije inverzne kinematike. Prvo definirajmo par vektora koji će nam pomoći u objašnjavanju ovog problema:

$$\theta = \begin{bmatrix} \theta_1 \\ \dots \\ \theta_n \end{bmatrix} \quad (3.2.5)$$

gdje je θ vektor kutova rotacije cijelog kinematičkog lanca. θ_i predstavlja kut rotacije zgloba i u odnosu na zglob $i-1$, tj. u odnosu na izvor ako je $i=1$. Zatim, definirajmo vektor e koji će predstavljati poziciju kraja lanca:

$$e = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} \quad (3.2.6)$$

$$e = f(\theta) \quad (3.2.7)$$

U unaprijednoj kinematici, vrijedi jednakost (3.2.7) koja omogućuje izračun pozicije ciljne točke na osnovu zadanih kutova svih segmenata modela. Jakobijan linearno modelira kretanje kraja lanca (promjenu vektora e) u odnosu na trenutne promjene položaja segmenata sustava.

Jakobijan J je matrica parcijalnih derivacija čitavog kinematičkog lanca u odnosu na vektor e . J_i , i -ti stupac Jakobijeve matrice predstavlja iznos vektora e u odnosu na kut određenog segmenta θ_i .

$$J = \frac{\partial \mathbf{e}}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial e_x}{\partial \theta_1} & \frac{\partial e_x}{\partial \theta_2} & \dots & \frac{\partial e_x}{\partial \theta_n} \\ \frac{\partial e_y}{\partial \theta_1} & \frac{\partial e_y}{\partial \theta_2} & \dots & \frac{\partial e_y}{\partial \theta_n} \\ \frac{\partial e_z}{\partial \theta_1} & \frac{\partial e_z}{\partial \theta_2} & \dots & \frac{\partial e_z}{\partial \theta_n} \end{bmatrix}$$

Slika 3.2.4 Jakobijeva matrica

U inverznoj kinematici, vrijedi jednakost:

$$\boldsymbol{\theta} = f^{-1}(e) \quad (3.2.8)$$

Za infinitezimalne vrijednosti promjene vektora kuta $\boldsymbol{\theta}$ i vektora pozicije ciljne točke e , vrijedi jednakost:

$$\Delta \boldsymbol{\theta} = J^{-1} \Delta e \quad (3.2.9)$$

Manipulacijom SVD (singularne dekompozicije vrijednosti) i jednadžbe normale, dolazimo do izračuna infinitezimalne promjene vektora kutova.

$$J \Delta \boldsymbol{\theta} = \Delta e \quad (3.2.10)$$

$$J^T J \Delta \boldsymbol{\theta} = J^T \Delta e \quad (3.2.11)$$

$$V S^2 V^T \Delta \boldsymbol{\theta} = J^T \Delta e \quad (3.2.12)$$

$$\Delta \boldsymbol{\theta} = V S^{-2} V^T \Delta e \quad (3.2.13)$$

Algoritam za izračun kutova vektora θ :

- Izračunaj Jakobijan J
- Izračunaj pomak $\Delta\theta$
- $\theta = \theta + \Delta\theta$
- Ažuriraj sve zglobove i čvorove na osnovu vektora θ
- Ponavljaj dok e_{trenutno} nije unutar prostora tolerancije e_{cilj}

Pseudo inverzna Jakobijeva matrica

Implicitno ili eksplicitno invertiranje Jakobijana J u gore predloženoj soluciji je skupo i redundantno u pogledu sustava. Popularno rješenje ovog problema je pseudo inverzna matrica J^+ :

$$J^+ = (J^T J)^{-1} J^T \quad (3.2.14)$$

umjesto matrice J^{-1} . Takva je matrica jedinstvena, međutim nestabilna u blizini singulariteta, te vremenski zahtjevna za izračunavanje. Singulariteti su točke u kojoj su vrijednosti pozicije segmenata beskonačne. Matrica nije matematički točna kao inverzna (ukoliko ista postoji), ali greška može biti smanjena ukoliko uzimamo manje korake po iteraciji.

Transponirana Jakobijeva matrica

Koncept ove matrice je jednostavan. Umjesto pseudo-inverzne matrice koristimo transponiranu. Korištenje ove matrice nije matematički točno, međutim teži ka rješenju. Također, nema problema sa singularitetima, a i računanje nije toliko zahtjevno. Međutim, javljaju se problemi pri skaliranju segmenata modela.

DLS metoda (Damped least squares)

Jakobijeva i slične matrice su osjetljive na singularitet sustava. Te metode osciliraju na visokim amplitudama blizu singulariteta, kako algoritmi pokušavaju konvergirati ka nedohvatljivim ciljevima. DLS metoda kompenzira te probleme uvođenjem konstante prigušenja A .

CCD algoritam

CCD algoritam (eng. „Cyclic-Coordinate Descent“) je iterativni optimizacijski postupak koji početni problem postavlja na slijedeći način. Pretpostavimo da u manipulatoru s n zglobova točka $\mathbf{P}_k = (x_k, y_k, z_k)$ označava krajnju točku manipulatora u njegovom trenutnom stanju, a točka $\mathbf{P}_c = (x_c, y_c, z_c)$ označava ciljnu točku koju želimo dohvatiti. Rješavanje inverzne kinematike se može svesti na definiranje funkcije pogreške, odnosno kvadrata udaljenosti između te dvije točke:

$$E(\mathbf{q}) = \|\mathbf{P}_c - \mathbf{P}_k\|^2 \quad (3.2.15)$$

i zatim primjenjivanja nekog od optimizacijskih algoritama za nalaženje minimuma te funkcije. Stanje \mathbf{q} za koje je funkcija pogreške minimalna je stanje u kojem je krajnja točka najbliža ciljnoj. U slučaju da je cilj dohvatljiv, to je upravo stanje manipulatora koje taj cilj i dohvaća.

Algoritam se može proširiti i na zadovoljavanje uvjeta orijentacije, a ne samo položaja, tako da se definira i funkcija pogreške orijentacije koja se sumira s pogreškom položaja te se ukupna pogreška pokušava minimizirati. Uvjeti orijentacije nisu proučavani jer se rijetko pojavljuju u praktičnoj upotrebi.

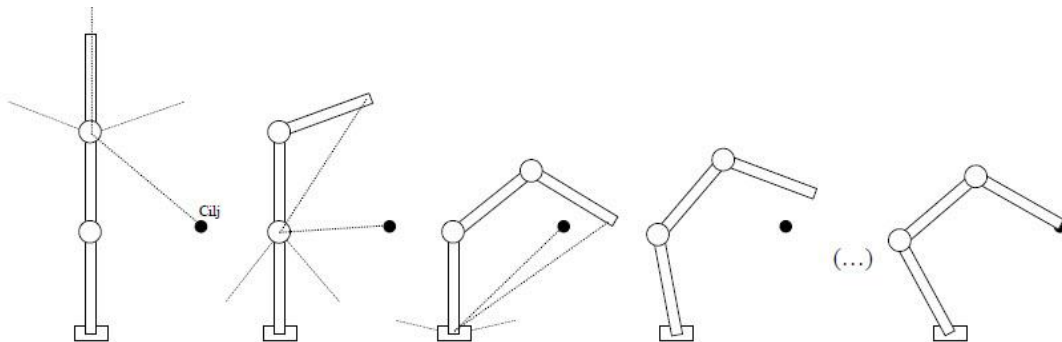
Ono što razlikuje ovaj algoritam od rješavanja Jakobijeve matrice i što algoritmu daje ime, je činjenica da se zglobovi obilaze sekvencijalno, odnosno jedan po jedan. Time se problem optimizacije višedimenzionalne funkcije svodi na sekvencijalno optimiranje jednodimenzionalnih funkcija.

Algoritam u svakoj iteraciji polazi od najudaljenijeg zgloba n u manipulatoru i prelazi redom sve zglobove do baze, u svakom koraku optimirajući funkciju pogreške, odnosno zakrećući trenutni zglob u onaj položaj u kojem će se krajnja točka manipulatora što više približiti ciljnoj. Svaka promjena trenutnog zgloba $/i/$ utječe i na sve prethodno posjećene zglobove $/i/+1$, $/i/+2$, itd.

Nakon jedne iteracije, odnosno nakon jednog obilaska svih zglobova u lancu, manipulator će biti u poziciji koja je bliža ciljnoj točki, ali za dostizanje

dostizanje optimalnog položaja (dohvatljive točke najbliže ciljnoj) potrebno je je više iteracija. Empirija pokazuje da je, ovisno o početnom stanju manipulatora, 10 do 50 iteracija dovoljno za dostizanje ciljne točke s određenom zadanom preciznošću.

Za jednostavan primjer manipulatora s tri zgloba (sa ograničenim pokretima), algoritam radi na slijedeći način:



Slika 3.2.5 Prva četiri koraka algoritma i konačno stanje dohvaćanja cilja.

U prvom koraku, zadnji zglob u lancu se postavlja u optimalan položaj. Zglob je ograničen (tanke crtane linije) pa se ne može okrenuti do položaja apsolutno najbližeg cilju, ali dolazi do svog maksimalnog otklona. Zatim se prethodni, tj. srednji zglob postavlja tako da se kraj manipulatora što više približi cilju. Nakon njega zakreće se i prvi zglob, odnosno baza manipulatora, i to u smjeru od cilja, jer je u tom smjeru optimalni položaj. U idućoj iteraciji, ponovo se podešava daljnji zglob, itd.

4. Implementacija

U ovom poglavlju prikazat ćemo postupak izrade 3D modela čovjeka sa svim potrebnim strukturama za animaciju te izradu animacije kretanja (hodanja) čovjeka. Koristit ćemo skeletalni model s hijerarhijskim kinematičkim lancima i inverznom kinematikom, a za to će nam poslužiti programski alat Blender (Slika 4.1.1).



Slika 4.1.1 Blender

4.1. Uvod

U slijedećih nekoliko poglavlja proći ćemo kroz sve korake izrade modela i animacije. Od izgradnje mreže modela (eng. „mesh“) do same animacije. Na slici 4.1.2 možemo vidjeti model čovjeka u završnoj fazi.

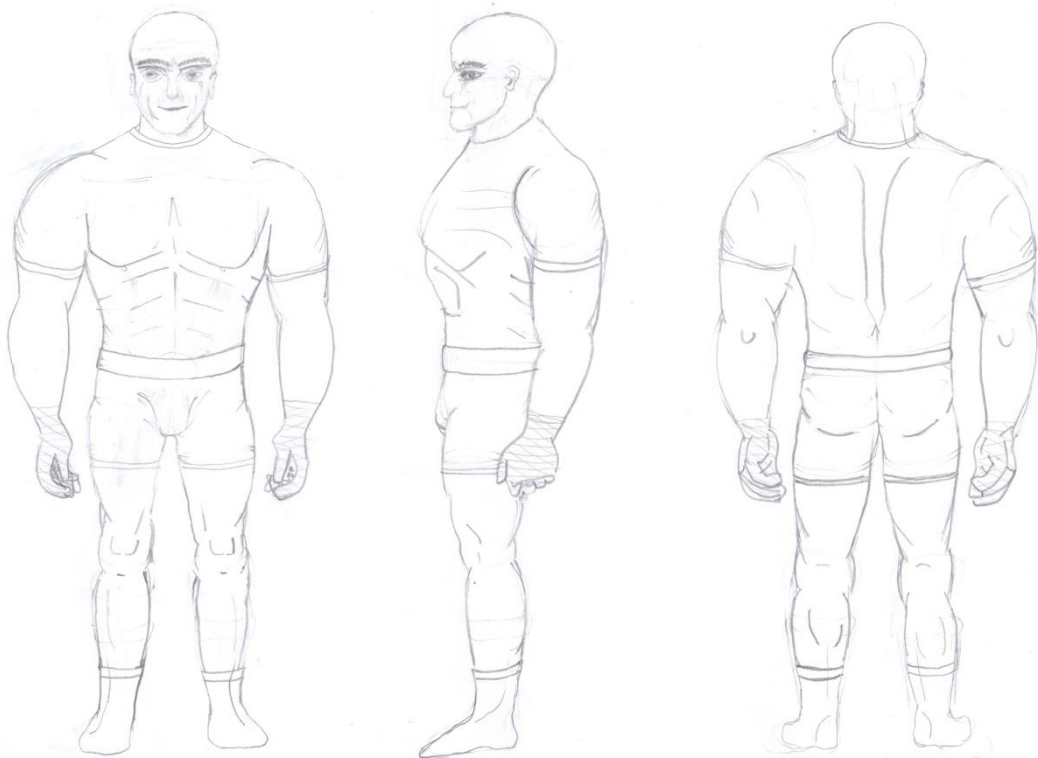


Slika 4.1.2 Gotov model čovjeka

4.2. Referentne slike

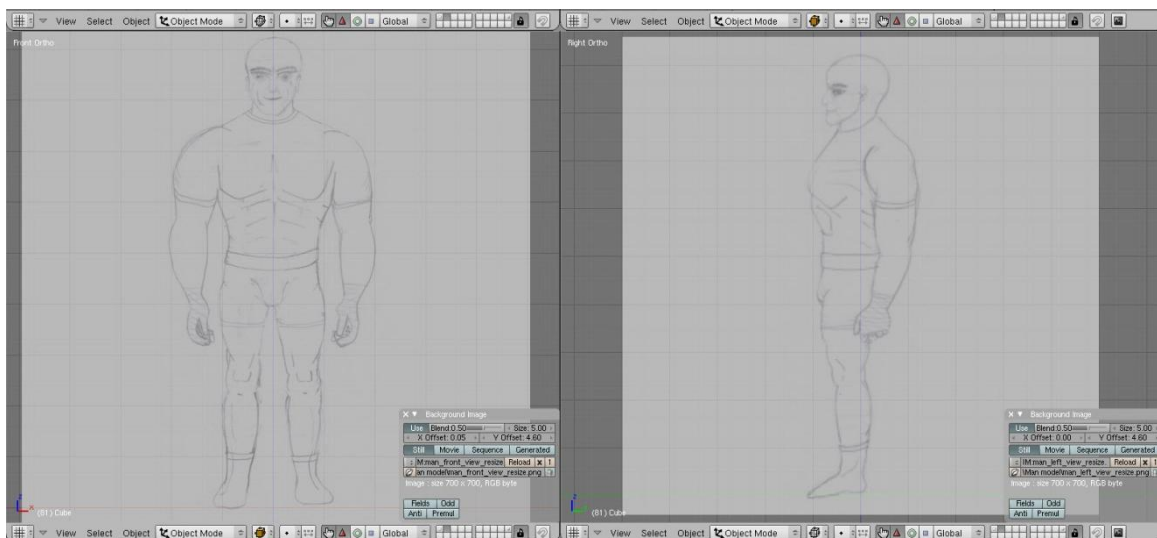
Koristeći referentne slike modela možemo si uvelike olakšati izradu mreže istog. Pri tome možemo koristiti proizvoljan broj slika. Međutim za izradu su nam dvije najbitnije, a to su slika modela slikanog sprijeda i s boka. Ako želimo vjerodostojniji model uvijek možemo koristiti referentne slike modela i iz drugih perspektiva kao na primjer model slikan odozgo ili recimo s leđne strane, zatim slike pojedinih dijelova tijela također slikanih iz različitih perspektiva.

U ovome radu koristit ćemo dvije referentne slike. Model slikan sprijeda i s bočne strane. Prvi korak je nacrtati zamišljeni model u željenim perspektivama (slika 4.2.1).



Slika 4.2.1 Skice modela u različitim perspektivama

Drugi korak je ubaciti referentnu sliku svakog pogleda u odgovarajući pogled na model u blenderu (slika 4.2.2). Naravno slike moraju biti u istoj rezoluciji i model na njima mora biti jednake veličine.



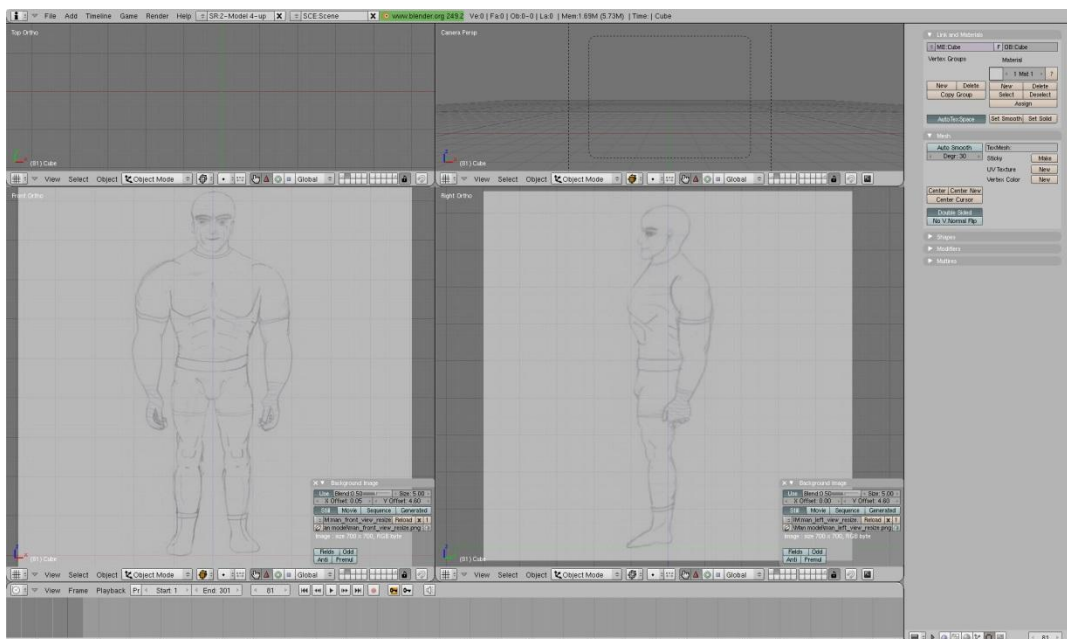
Slika 4.2.2 Referentne slike u blenderu

Kad se referentne slike nalaze u blenderu pozicioniramo ih u koordinatnom sustavu tako da model stoji na x-osi, z-os čini osnu simetriju čovjeka, a y-os prolazi sredinom bočnog pogleda na model.

4.3. Izgradnja mreže modela (MESH)

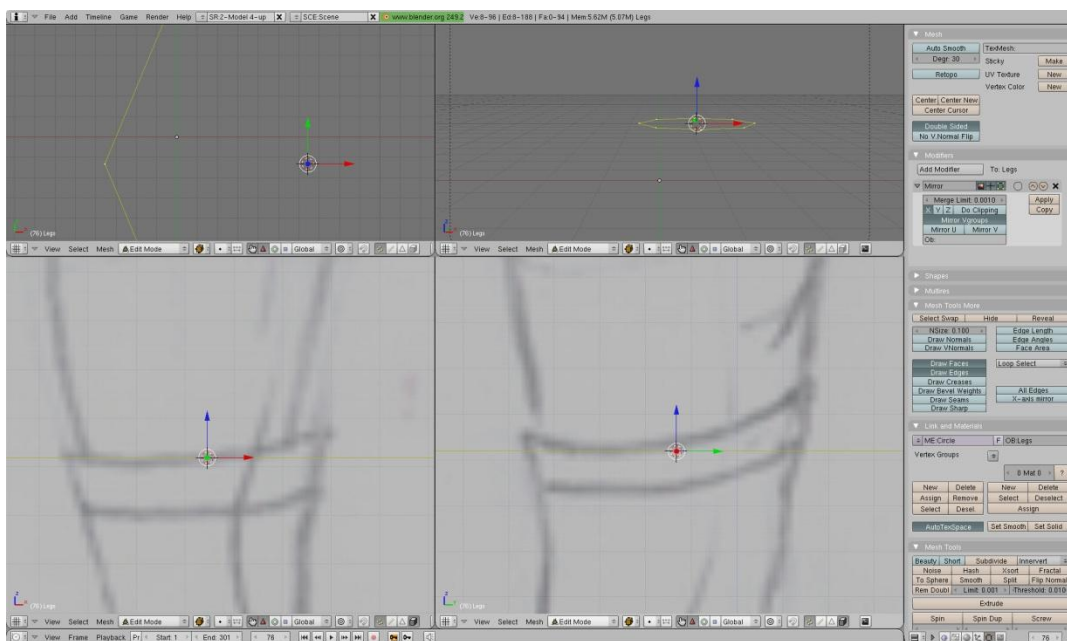
Izgradnju mreže modela podijelit ćemo po dijelovima tijela, te ćemo ih nakon završetka modeliranja povezati u konačni oblik mreže modela čovjeka. Prije nego krenemo na modeliranje pojedinih dijelova važno je napomenuti da ćemo se koristiti jednim „trikom“ u blenderu. Koristit ćemo modifikator „Mirror“ primijenjen na Y-Z ravninu. Na taj način ćemo mrežu koju ćemo napraviti na desnoj strani čovjeka kopirati na lijevu stranu. Time ćemo uštedjeti puno vremena, a isto tako dobiti simetriju koja je prisutna kod ljudi.

Sučelje Blendera namjestit ćemo tako da imamo 3 pogleda na model: prednji (Front), bočni-desni (Right) i odozgor (Top). Također dodat ćemo još jedan pogled (Camera) koji će imati proizvoljni pogled na model kako bi vidjeli kako izgleda to što smo izmodelirali iz svih kutova (slika 4.3.1).



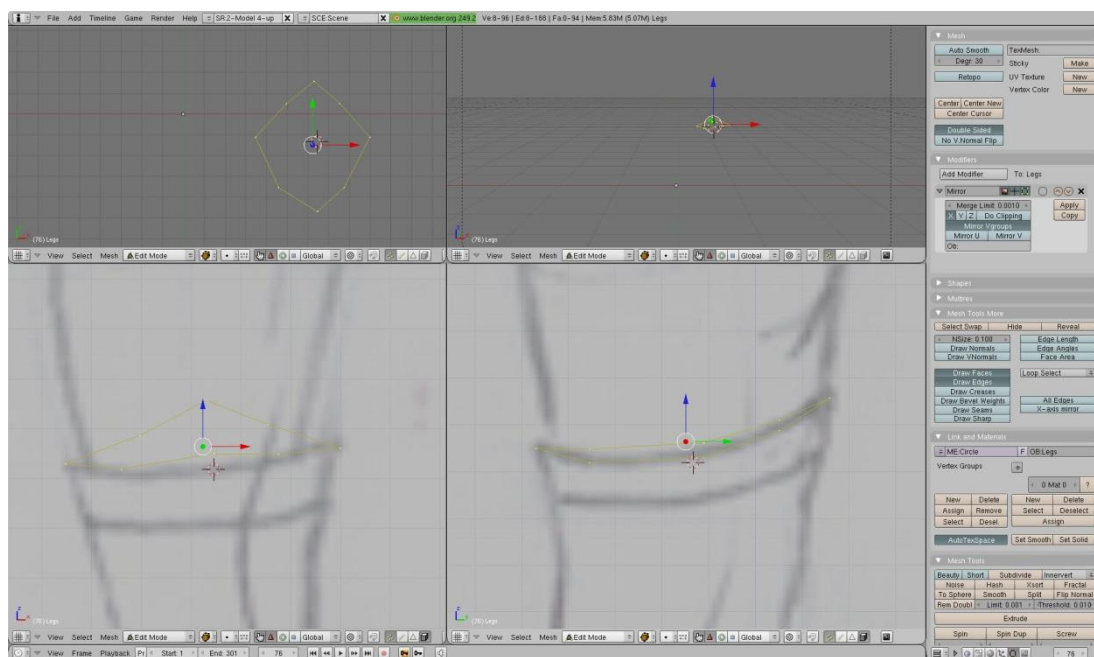
Slika 4.3.1 Sučelje blendera

Krenut ćemo od nogu, točnije čizama. Formiranje mreže započeti ćemo mnogokutom (blender: Circle) s 8 vrhova. To će biti dovoljno vrhova da dobijemo željeni oblik. Pozicioniramo kursor u prednjem i bočnom pogledu da odredimo središte mnogokuta. Zatim odemo u pogled odozgo i tamo dodamo mnogokut (blender: Space->Add->Circle). Tako ćemo dobiti mnogokut polegnut u X-Y ravni, jer želimo s mnogokutom napraviti krug oko noge (slika 4.3.2).



Slika 4.3.2 Dodavanje mnogokuta

Kad smo dodali mnogokut skaliramo (blender: S) ga u prednjem i bočnom pogledu tako da rubovi mnogokuta odgovaraju rubovima na referentnim slikama. Nakon toga rotiramo (blender: R) u bočnom pogledu, jer vidimo kako gornji rub čizme lagano pada s desna na lijevo. Skaliramo još malo, ako je potrebno. Nakon toga u bočnom pogledu uzmemo (blender: G) krajnji lijevi vrh i namjestimo ga na pravu poziciju, isto tako učinimo i s krajnjem desnim te popravimo vrhove u sredini. Ponovimo postupak i u prednjem pogledu (Slika 4.3.3).

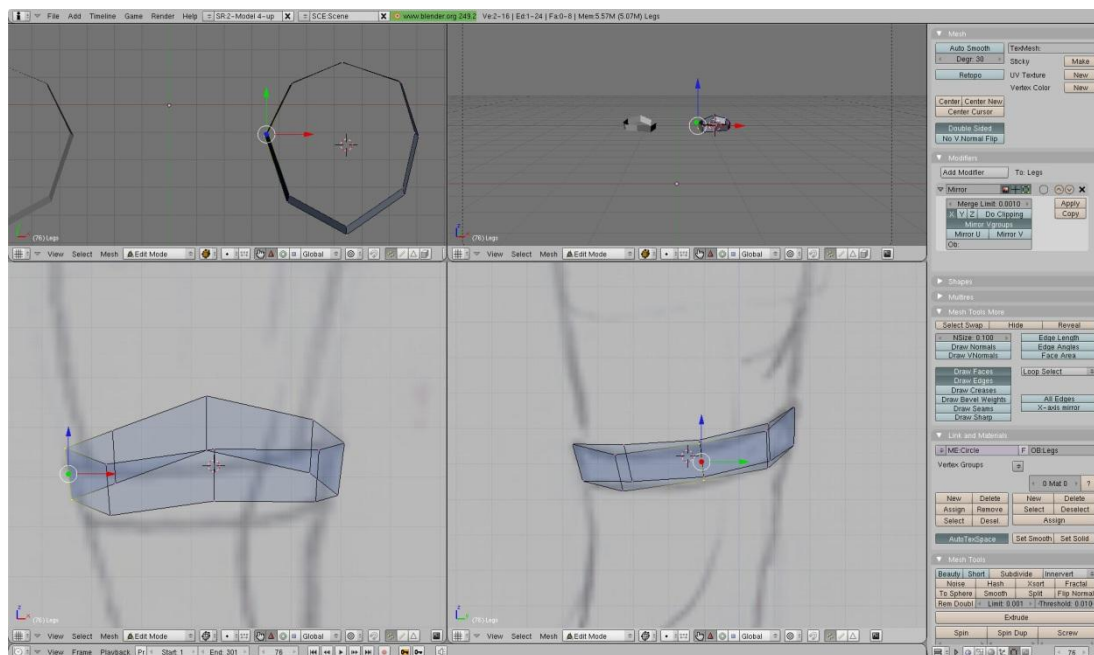


Slika 4.3.3 Prilagođavanje vrhova rubovima referentnih slika

U pogledu odozgo vidimo da mnogokut koji predstavlja obris vrha čizme nije dovoljno „okrugao“. Iako nam se u prednjem i bočnom pogledu čini da je sve u redu, u pogledu odozgo vidimo da moramo još malo poraditi na razmještanju vrhova da bi dobili željeni oblik.

Na ovom primjeru, a tek smo počeli modelirati, vidimo da je podjela sučelja na prisutna tri odnosno četiri pogleda vrlo korisna. Isto tako treba napomenuti da sve što radimo u jednom pogledu (modifikacije) u realnom vremenu se reflektira i na ostale poglede te na taj način u svakom trenutku vidimo što se događa s modelom u onim dijelovima koji trenutno nisu vidljivi u dotičnom pogledu.

Sljedeći korak je proširivanje mreže odnosno „izvlačenje“ vrhova (blender: Extrude). Iako trenutna mreža ne izgleda kao takva uskoro ćemo to promijeniti. Selektiramo sve vrhove (blender: A) ili bridove (dva susjedna vrha) mreže koje želimo proširiti, u našem slučaju svi prisutni vrhovi, te ih izvučemo (blender: E->Only Edges) (Slika 4.3.4).

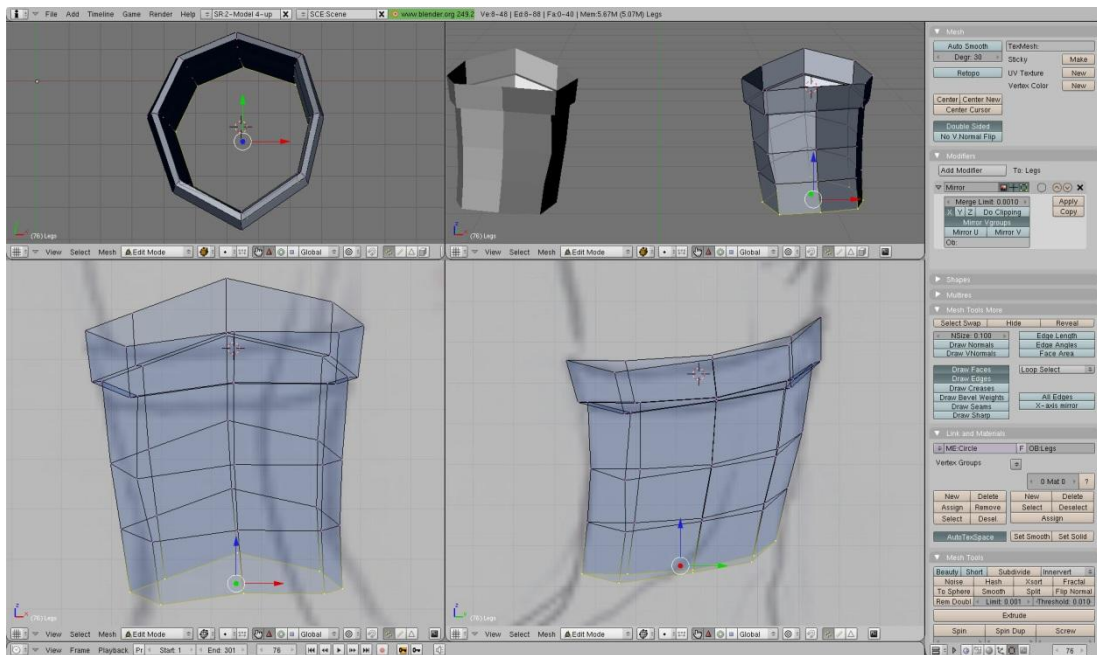


Slika 4.3.4 Izvlačenje bridova

Razlika između izvlačenja vrhova i bridova je što izvlačenjem vrhova dobivamo bridove između novog vrha i vrha kojeg izvlačimo, a izvlačenjem bridova dobivamo površine (Faces) omeđene novim bridom i bridom kojeg izvlačimo.

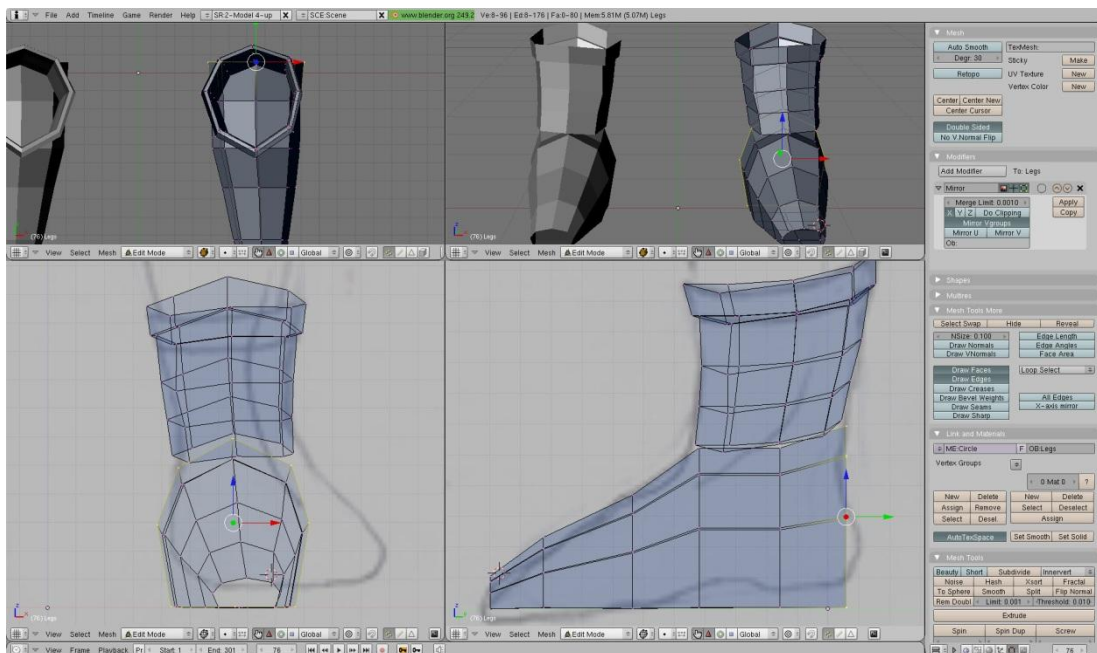
Visinu izvlačenja bridova određujemo ovisno o obliku kojeg modeliramo. Što je površina ravnija to visina može biti veća, a što je površina nepravilnija, to je visina manja. Isto tako treba raspodijeliti visine tako da se na zglobu modela uvijek nalazi „prsten“ (bridovi raspoređeni u krug) oko kojeg će se mreža savijati kod animacije kretanja.

Da bi istaknuli rub čizme izvući ćemo bridove, ali tako da ćemo ih pozicionirati na ista mjesta na kojima su bridovi iz kojih ih izvlačimo, te ćemo ih samo skalirati. Smanjit ćemo radijus mnogokuta i dobit ćemo rub. Nastavimo izvlačiti bridove dok ne dođemo do gležnja (Slika 4.3.5)



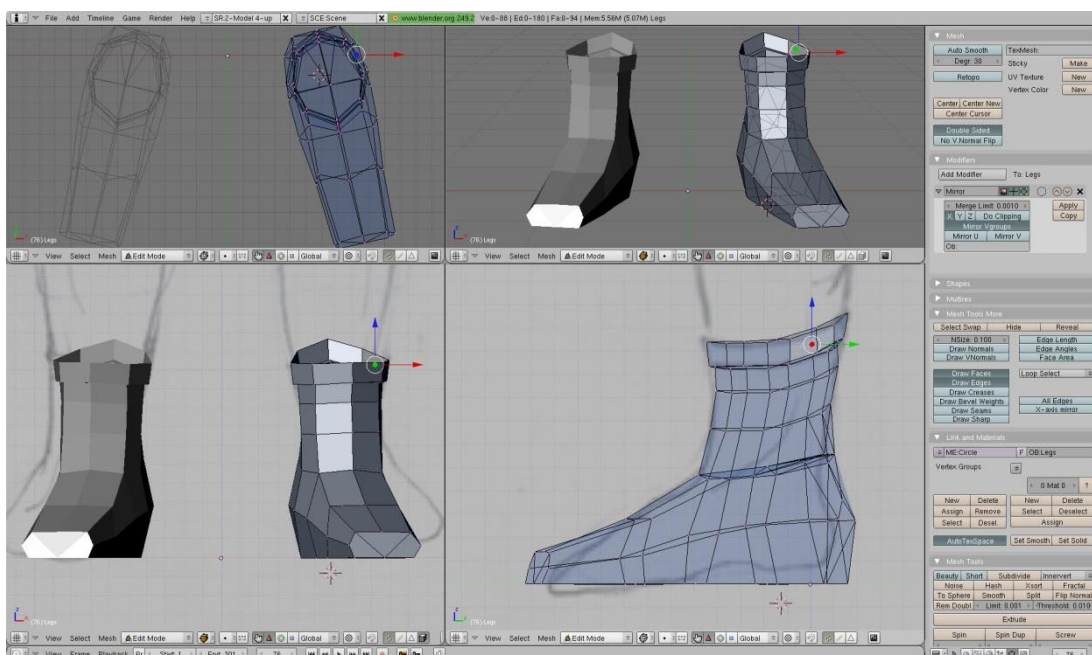
Slika 4.3.5 Gornji dio čizme

Kao što smo dodali mnogokut kad smo počeli raditi čizmu isto tako dodamo jedan mnogokut (s također 8 vrhova) ali ovaj put na prednjem dijelu čizme (kod prstiju) i to u prednjem pogledu. Skaliramo, rotiramo i modificiramo vrhove dok se ne prilagodimo referentnoj slici. Zatim izvlačimo taj mnogokut prateći oblik stopala dok ne dođemo do pete (Slika 4.3.6).



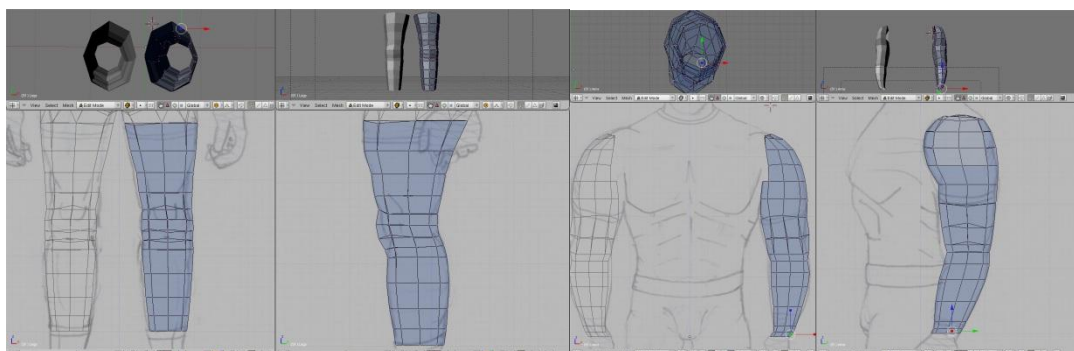
Slika 4.3.6 Donji dio čizme

Sad kad imamo grubi oblik čizme, prvo što moramo napraviti je povezati gornji s donjim dijelom čizme. To radimo tako da odaberemo dva vrha koja želimo spojiti i spojimo ih (blender: W->Merge->At Center) ili na sredini ili u jednom od ta dva vrha ovisno kako nam odgovara. Također moramo popuniti rupe kod prstiju i pete tako da odaberemo četiri vrha i pretvorimo ih u površinu (blender: F). Time smo napravili desnu čizmu, a isto tako zahvaljujući modifikatoru „Mirror“ i lijevu. Nakon malo dotjerivanja dobijemo čizme (Slika 4.3.7).



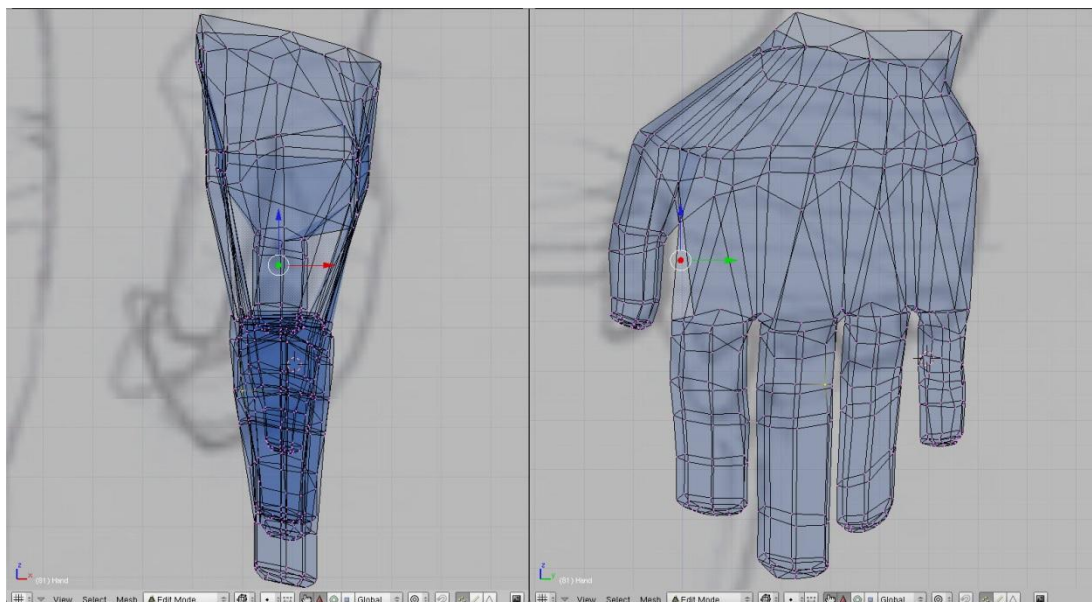
Slika 4.3.7 Čizme

Sljedeće na redu su noge. Dodamo mnogokut s 8 vrhova iznad vrha čizama i izvlačenjem prema gore do prepona dobijemo noge (Slika 4.3.8a). Na isti način radimo i ruke. Dodamo mnogokut od 8 vrhova kod ručnog zgloba i izvlačimo prema gore sve do ramena (Slika 4.3.8b).



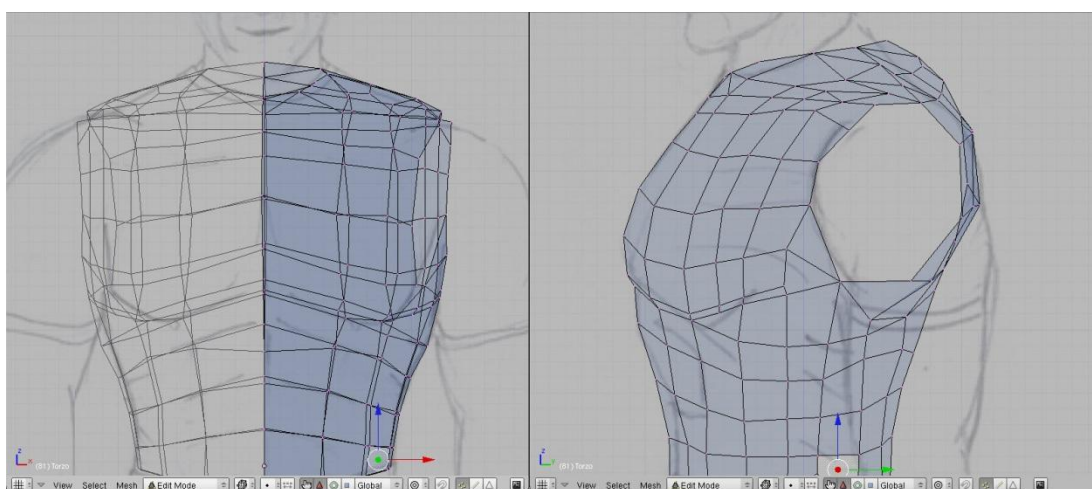
Slika 4.3.8 a) Noge b) Ruke

Da bi dovršili ruke moramo napraviti još šaku i prste. Prvo napravimo jedan prst, recimo kažiprst i zatim ga umnožimo 5 puta te svaki prilagodimo da dobijemo sve prste jedne ruke. Zatim povežemo krajeve prstiju s ručnim zglobom i na taj način dobijemo šaku (Slika 4.3.9).



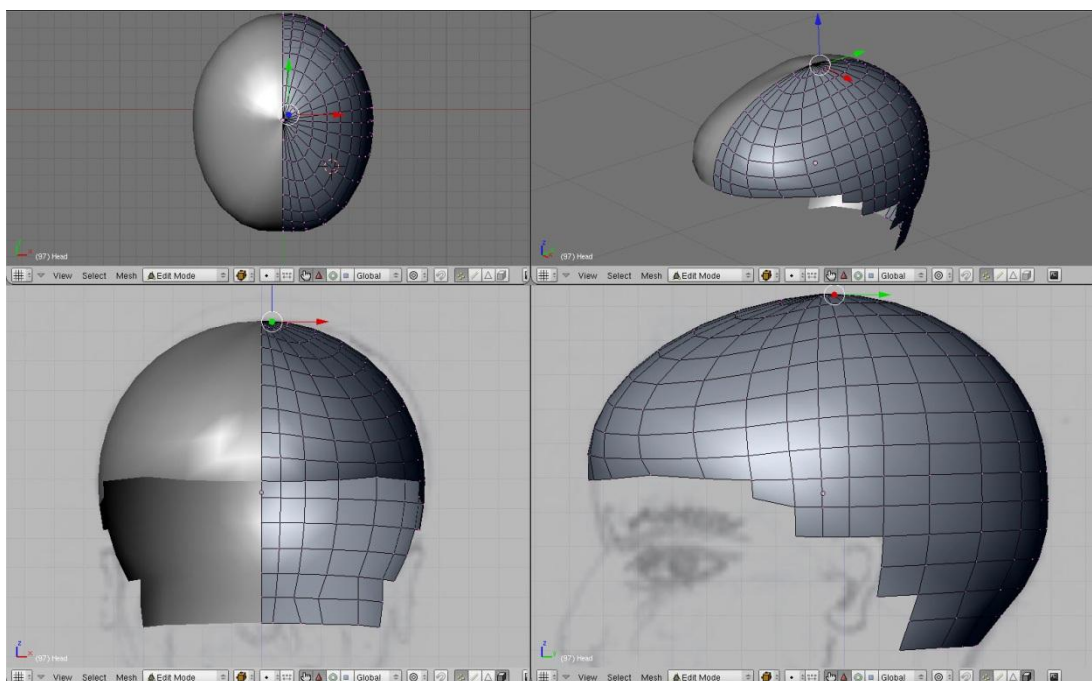
Slika 4.3.9 Ruka

Za trup ćemo koristiti mnogokut s 16 vrhova kako bi bolje izrazili oblik tijela, jer se sada radi o većoj površini. Dodamo ga oko struka. Izbrišemo vrhove lijeve polovice mnogokuta, jer će se oni ionako kopirati. Te izvlačimo prema dolje da dobijemo zdjelicu, i prema gore da dobijemo torzo (Slika 4.3.10).



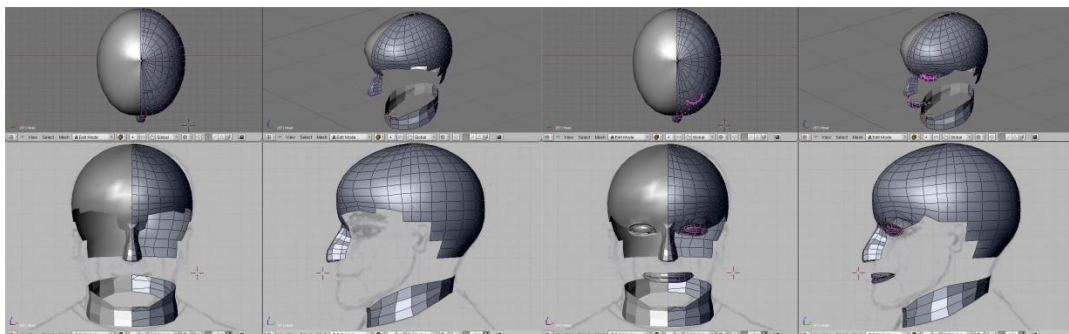
Slika 4.3.10 Torzo

Najsloženiji dio mreže je područje glave. Dodat ćemo NURBS sferu (blender: Add->Surface->NURBS Sphere) kako bi je oblikovali. Iz kugle koju imamo na početku kontrolnim točkama NURBS sfere prilagođavamo se referentnim slikama. Naravno ne možemo zahvatiti cijelu glavu nego samo gornji dio. Kada postignemo zadovoljavajući oblik NURBS sferu moramo pretvoriti u mrežu (blender: Object Mode: Surface->Convert Object Type). Sada obrišemo lijevu polovicu i višak vrhova na desnoj strani (Slika 4.3.11).



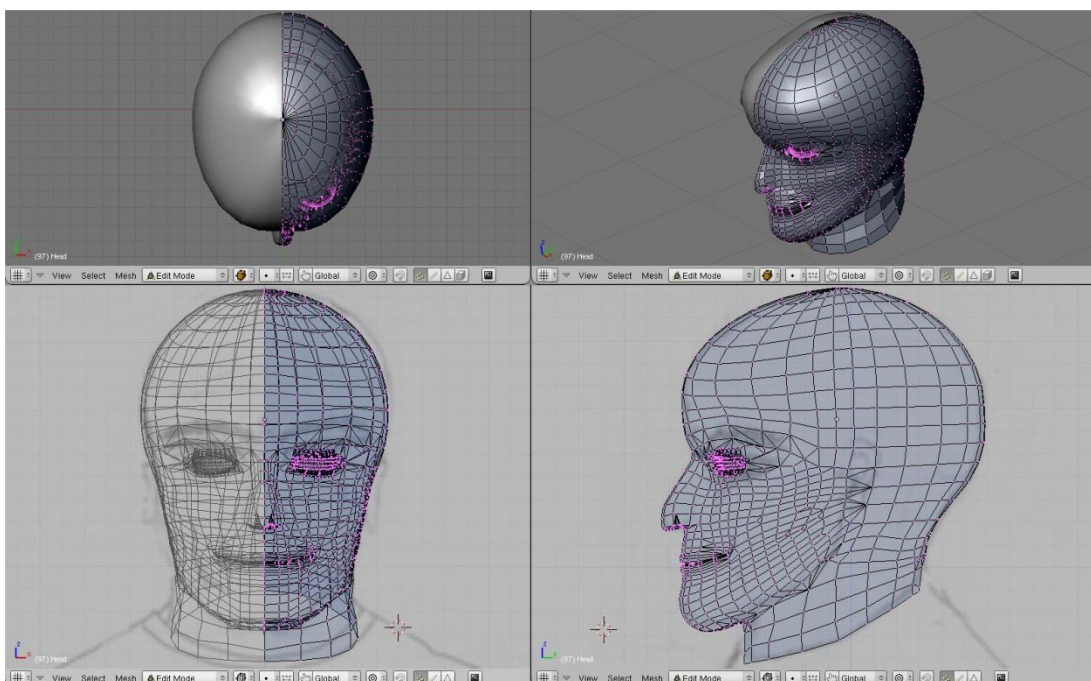
Slika 4.3.11 Oblikovanje mreže gornjeg dijela glave

Selektiramo krajnje bridove čela i izvlačimo ih prema dolje oblikujući čelo do kraja. Izvlačenjem vrha u bočnom pogledu pratimo oblik nosa. Zatim izvlačenjem dobivenih bridova u desno u prednjem pogledu oblikujemo cijeli nos. Dodamo mnogokut s 16 vrhova u pogledu odozgo kod donjeg dijela vrata. Obrišemo lijevu polovicu i izvlačimo prema gore dok ne dobijemo cijeli vrat (Slika 4.3.12a). Usnice radimo također dodavanjem mnogokuta s 8 vrhova. Brišemo lijevu polovicu i izvlačenjem prema gore i dolje oblikujemo gornju i donju usnicu. Oko oblikujemo pomoću sfere pretvarajući je u mrežu i brisanjem viška vrhova (Slika 4.3.12b).



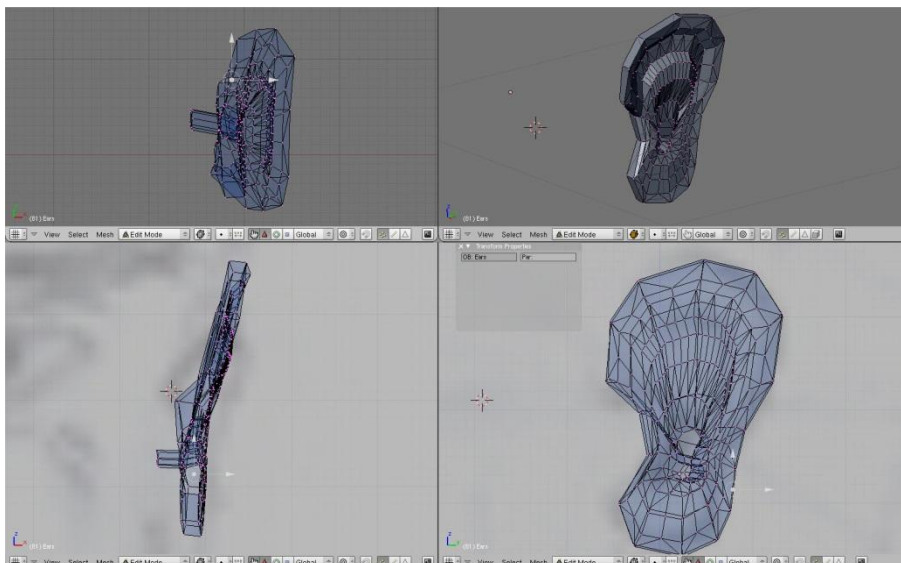
Slika 4.3.12 a) Nos i vrat b) Usta i oči

Preostalo je povezati modelirane dijelove glave. Krenemo od vrata prvo dovršavajući stražnji dio glave, a zatim također od vrata, lice (čeljust i obraze). Da bi izrazili obraze u Sculpt načinu rada pomoću alata Inflate (I) izvlačimo mrežu oko obraza kao da ju napuhavamo i zatim zaglađujemo pomoću alata Smooth (S). Rezultat je slika 4.3.13.



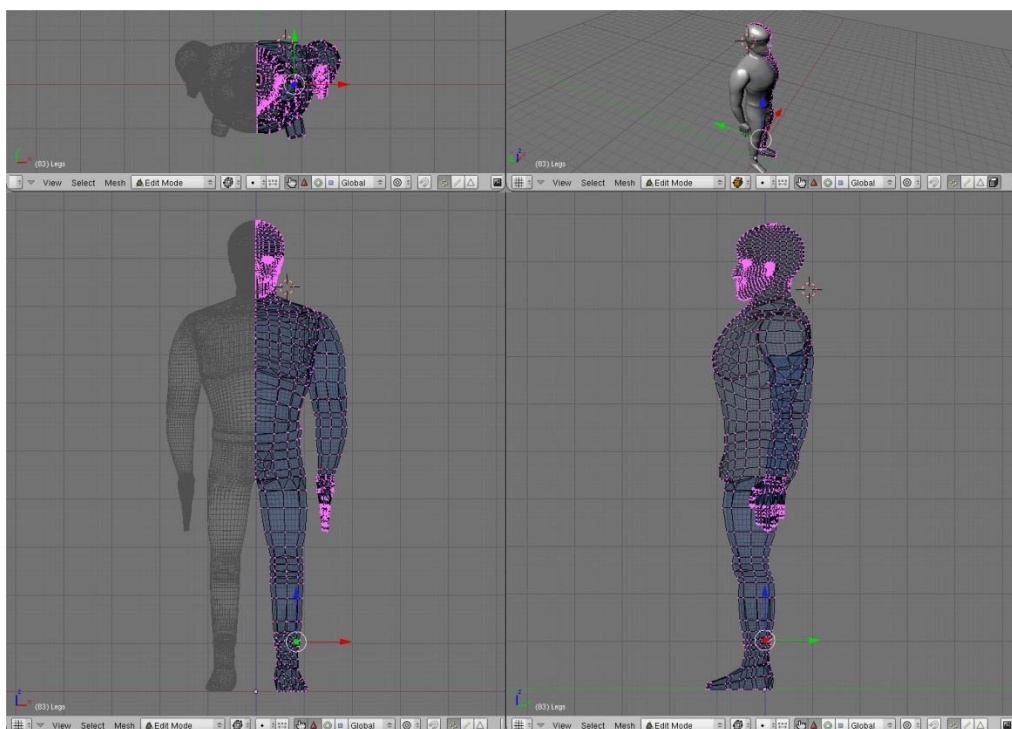
Slika 4.3.13. Model glave

Ne smijemo zaboraviti uši. U bočnom pogledu dodamo mnogokut s 8 vrhova i modificiramo vrhove prateći oblik uha te u prednjem pogledu izvučemo bridove jednom kako bi dobili debljinu uha. Zatim izvlačenjem u bočnom pogledu stvaramo unutrašnjost uha te završavamo stvaranjem ušne šupljine (Slika 4.3.14).



Slika 4.3.14 Uho

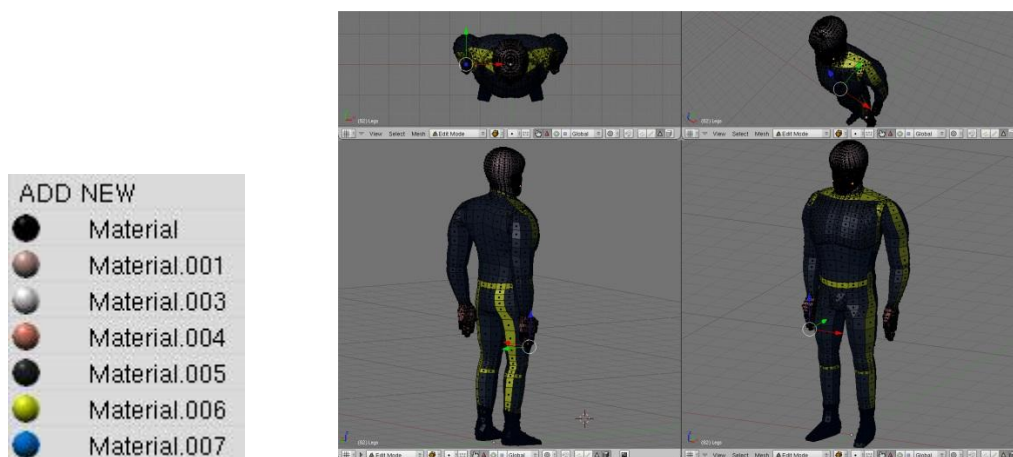
Ovime smo izmodelirali sve dijelove tijela te ih sada moramo spojiti u jednu kompaktnu mrežu. Otvorimo novu datoteku. Dodamo čizme te noge pomoću opcije „append“ u blenderu, ali tako da dodamo kao kopiju objekta a ne poveznicu. Sada ta dva objekta moramo spojiti u jednu mrežu (blender: ctrl+J) i povezati vrhove („zašiti“) gdje se mreže spajaju. Na isti način na koji smo dodali noge dodajemo redom ostale dijelove tijela i spajamo ih s postojećom mrežom. Rezultat je mreža modela čovjeka (Slika 4.3.15).



Slika 4.3.15 Mreža gotovog modela čovjeka

4.4. Dodavanje materijala

Materijale dodajemo mreži tako da selektiramo površinu (face) kojoj želimo dodijeliti materijal (vrhove koji omeđuju površinu), odaberemo materijal i dodijelimo ga selektiranoj površini. Istovremeno možemo dodijeliti isti materijal više površina tako da ih selektiramo i onda dodijelimo materijal. Svaki materijal ima jednu boju tako da ćemo napraviti onoliko materijala koliko boja nam je potrebno za naš model (Slika 4.4.1a).

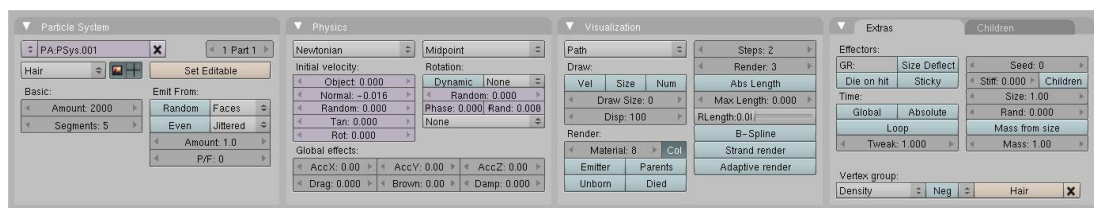


Slika 4.4.1 a) Materijali b) Model s dodijeljenim materijalima

Nakon dodjeljivanja materijala svim površinama mreže model izgleda kao na slici 4.3.1b.

4.5. Dodavanje kose

Za dodavanje kose koristit ćemo se alatom ugrađenim u blender. Pod „Particle Buttons“ u izborniku „Particle System“ odaberemo opciju „Hair“. Moramo odrediti na koji dio mreže stavljamo kosu stvarajući grupu vrhova s imenom „Hair“, zatim količinu, vrstu, duljinu, smjer i tip stvaranja kose (Slika 4.5.1).



Slika 4.5.1 Opcije stvaranja kose

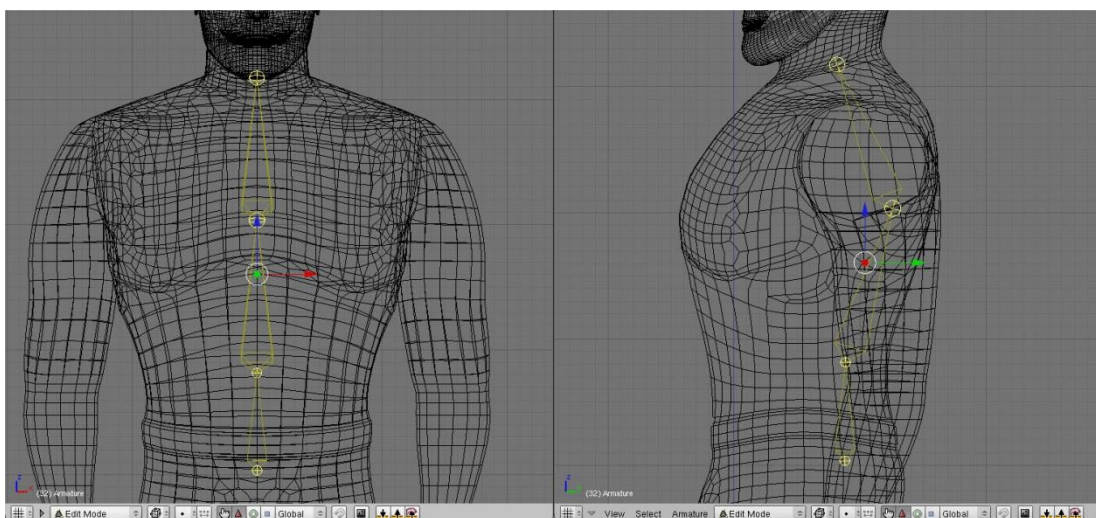
Rezultat je dobivena kosa na slici 4.5.2.



Slika 4.5.2 Kosa

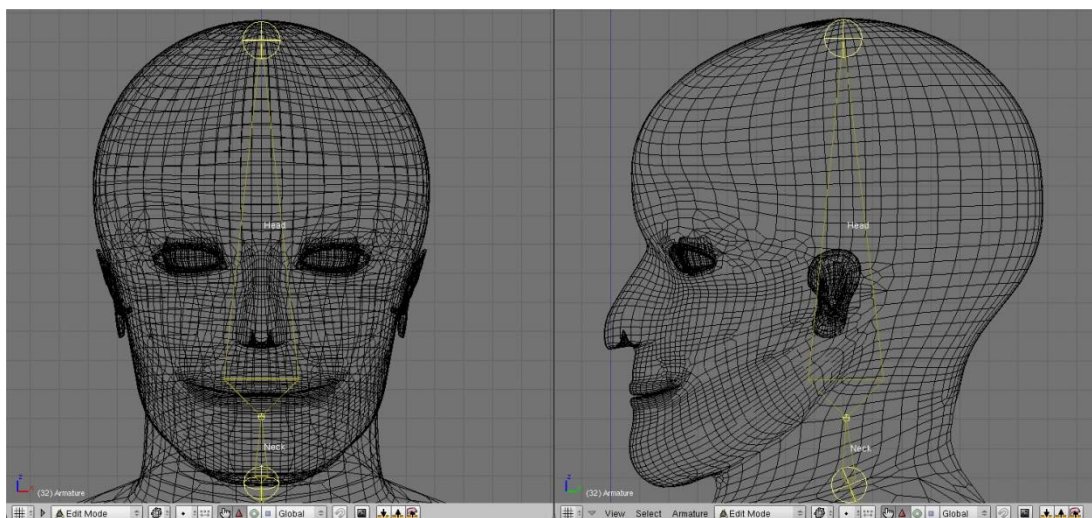
4.6. Izgradnja kostura

Da bi mogli animirati kretanje čovjeka potreban nam je kostur (blender: Armature). Nećemo raditi kostur identičan kosturu čovjeka nego ćemo ga pojednostaviti u onoj mjeri koliko je dovoljno za prikazivanje animacije kretanja čovjeka. Krećemo od dodavanja (blender: Add->Armature) izvorišne kosti (root) modelu čovjeka. Pozicionirat ćemo je u području donjeg dijela kralježnice te iz nje izvući (blender: E) prema gore još dvije kosti tako da te tri kosti čine kralježnicu čovjeka (Slika 4.6.1).



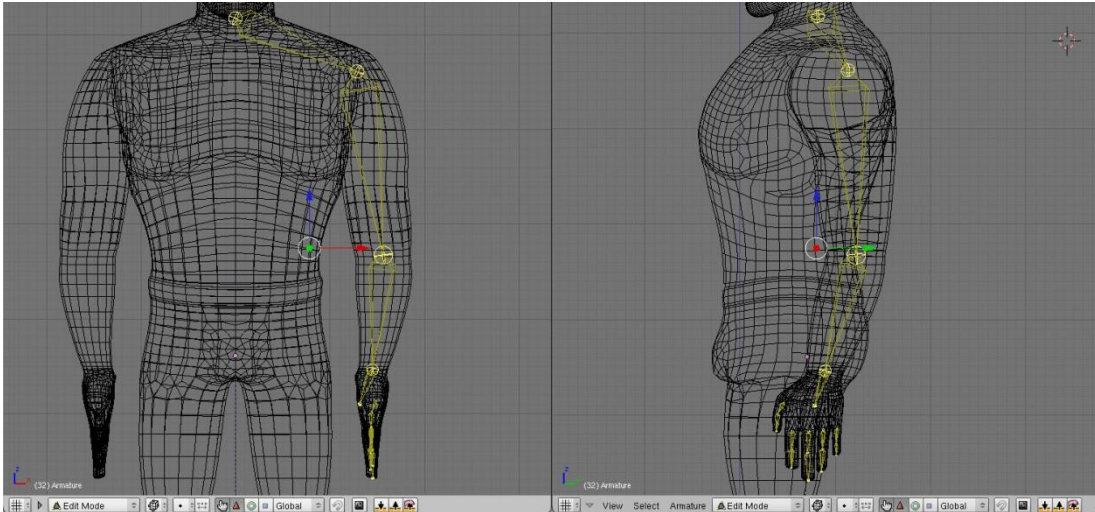
Slika 4.6.1 Kralježnica čovjeka

Blender je kosti nazvao redom: „Root“, „Root.001“, „Root.002“ te ćemo ih tako i ostaviti. Iz kosti „Root“ izvući ćemo kost „Waist“, a iz kosti „Root.001“ kost „Chest“. Sada ćemo iz kosti „Root.002“ izvući dvije kosti prema gore. Prvu ćemo nazvati „Neck“, a drugu „Head“ koje će nam poslužiti za micanje vrata i glave (Slika 4.6.2).

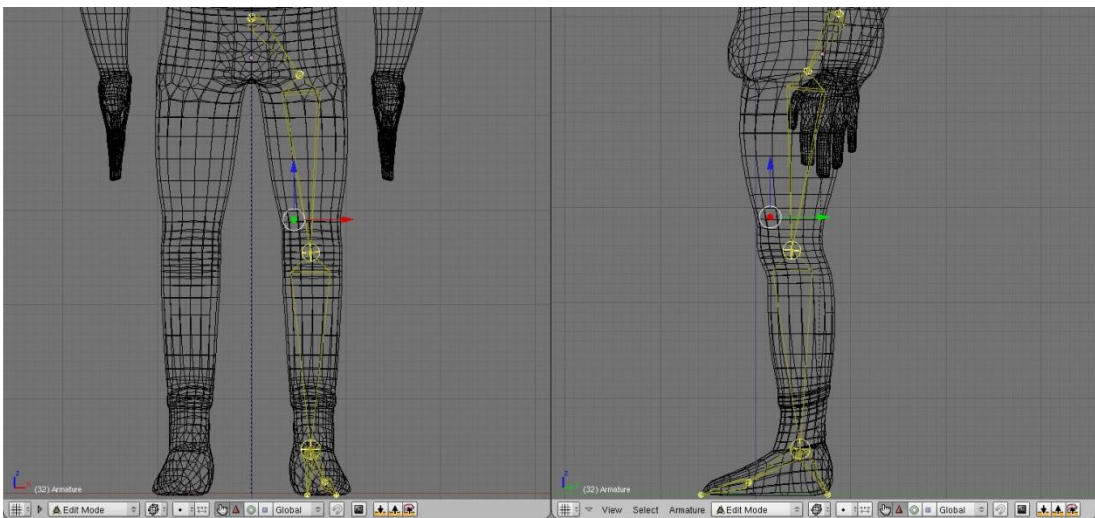


Slika 4.6.2 Kostri vrata i glave

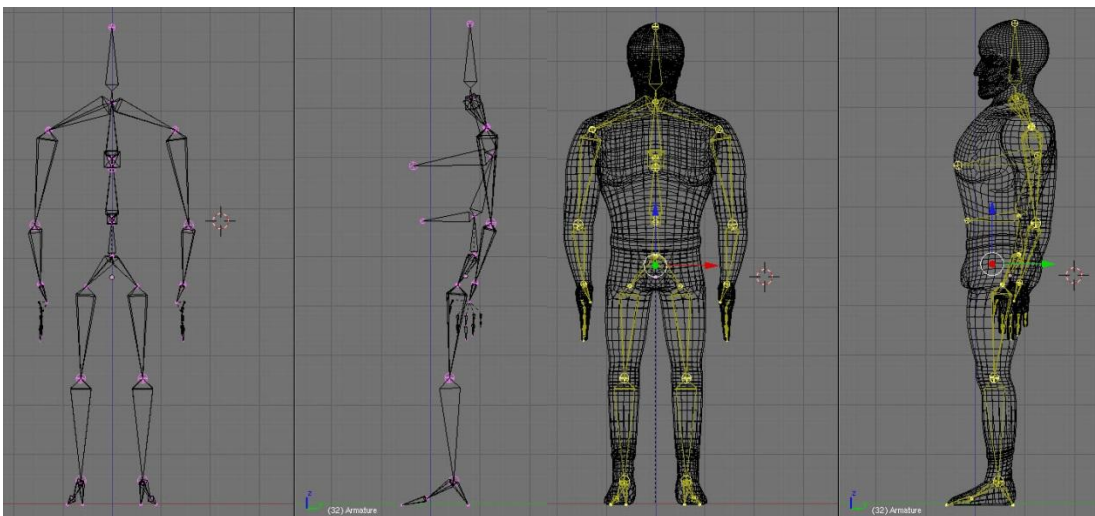
Nakon što smo napravili sredinu kostura sada ćemo napraviti lijevu stranu te ju zatim kopirati, okrenuti i na taj način dobiti desnu stranu. Iz kosti „Root.002“ izvući ćemo još jednu kost i nazvati je „Shoulder.L“ iz koje ćemo kasnije izvući kosti ruke. Posljednjoj kosti dodali smo nastavak „.L“ da bi obilježili da se radi o kosti lijeve strane kostura te da bi kasnije kod zrcaljenja lijeve strane blender automatski dodijelio istoj kosti samo s desne strane nastavak „.R“ umjesto „.L“. Iz kosti „Shoulder.L“ sada izvlačimo kosti „ArmHi.L“, „ArmLo.L“ i „Hand.L“. Kostri ne moraju nužno biti povezane te ćemo to iskoristiti prilikom stvaranja kosti prstiju. Za svaki prst napraviti ćemo dvije kosti za gornji i donji dio prsta. Imena kosti su sljedeća: „Thumb.L“, „Thumb.L.001“, „Pointer.L“, „Pointer.L.001“, itd. (Slika 4.6.3). Na sličan način ćemo napraviti i kosti nogu. Iz kosti „Root“ izvući ćemo redom kosti „Hip.L“, „LegHi.L“, „LegLo.L“, „Foot.L“, „Toes.L“ te iz „LegLo.L“ još jednu kost „Heel.L“ (Slika 4.6.4). Time smo napravili kosti lijeve strane. Sada ih označimo (kosti ruku i nogu), udvostručimo ih, okrenemo i pozicioniramo na desnu stranu te kažemo blenderu da zamjeni nastavke „.L“ s „.R“ i kostur je gotov (Slika 4.6.5).



Slika 4.6.3 Kostii ruke



Slika 4.6.4 Kostii noge



Slika 4.6.5 Kostur

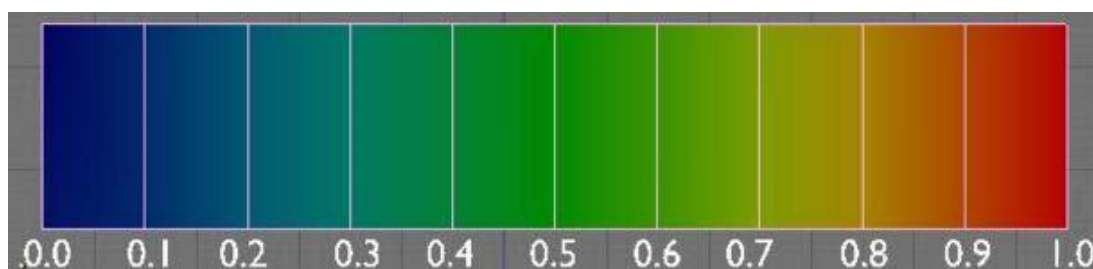
4.7. Povezivanje kostura i mreže

Prije nego povežemo kostur s mrežom moramo primijeniti modifikator „Mirror“ na mrežu tako da se stvore svi vrhovi na lijevoj strani. Kostur povezujemo s mrežom kako bi kostur kontrolirao mrežu. Tako ćemo pomicanjem kostiju pomicati i dio mreže koja je pod njezinom kontrolom te na taj način animirati kretanje. Povezivane ćemo postići odnosom roditelj-dijete dajući ulogu roditelja kosturu, a ulogu djeteta mreži (blender: ctrl + P). Prilikom stvaranja tog odnosa stvara se onoliko grupa vrhova koliko ima kostiju. Svaka grupa dobiva ime po imenu kosti na koju se odnosi (Slika 4.7.1).

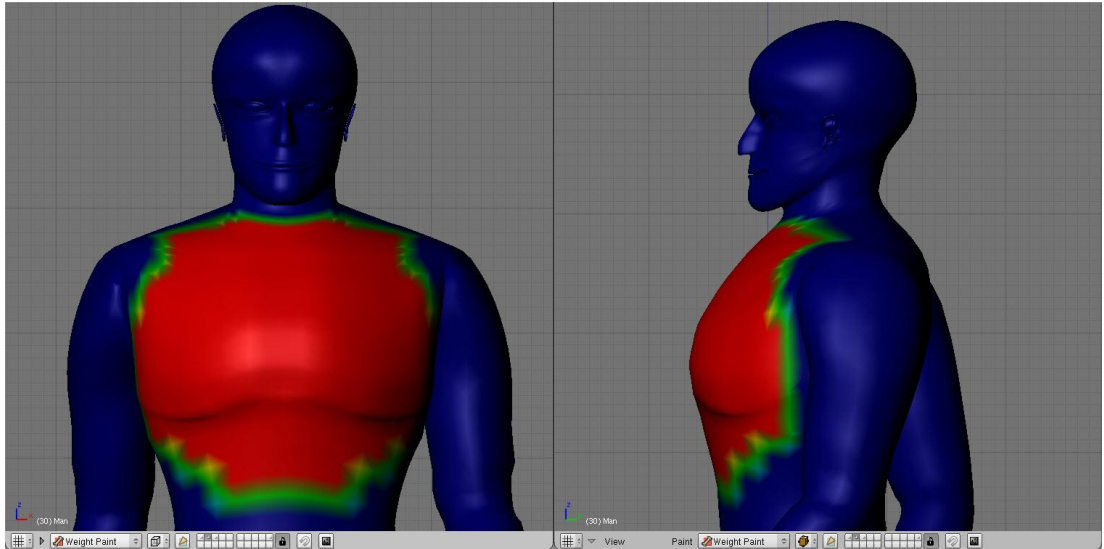
ArmHi.L	Hip.R	Ring.L			
ArmHi.R	LegHi.L	Ring.L.001	Foot.R	Neck	Thumb.L.001
ArmLo.L	LegHi.R	Ring.R	Hair	Pinky.L	Thumb.R
ArmLo.R	LegLo.L	Ring.R.001	Hand.L	Pinky.L.001	Thumb.R.001
Arms	LegLo.R	Root	Hand.R	Pinky.R	Toes.L
Boots	Legs	Root.001	Hands	Pinky.R.001	Toes.R
Chest	Middle.L	Root.002	Head	Pointer.L	Torzo
Ears	Middle.L.001	Shoulder.L	Heel.L	Pointer.L.001	Waist
Eye	Middle.R	Shoulder.R	Heel.R	Pointer.R	
Foot.L	Middle.R.001	Thumb.L	Hip.L	Pointer.R.001	

4.7.1 Popis grupa

Svakoj grupi moramo dodijeliti sve vrhove koje želimo da se pomiču zajedno s kostima s imenom te grupe. To možemo napraviti na dva načina. Prvi način je bojanjem (blender: Weight Paint Mode), tako da označimo grupu vrhova koju želimo editirati te kistom bojamo površinu modela. Crvena boja označava da će se ti vrhovi micati zajedno s kosti (koeficijent 1.0) dok plava boja označava da vrh nije u grupi tj. da se neće pomicati (0.0). Možemo bojati s bilo kojom bojom iz spektra (Slika 4.7.2).



Slika 4.7.2 Spektar s koeficijentima



Slika 4.7.3 Grupa vrhova „Chest“ u „Weight Paint Mode“

Na slici 4.7.3 možemo vidjeti kako to izgleda kada bojanjem editiramo grupu vrhova „Chest“.

Drugi način je da selektiramo sve vrhove u „Edit Mode“ koje želimo da su pod kontrolom određene kosti i dodijelimo ih grupi s imenom te kosti. Ovim načinom imamo veću kontrolu nad dodjeljivanjem vrhova određenim grupama međutim ne možemo na primjer dodijeliti vrhu težinski koeficijent 0.4. Tako da možemo kombinirati oba načina rada ovisno o potrebi.

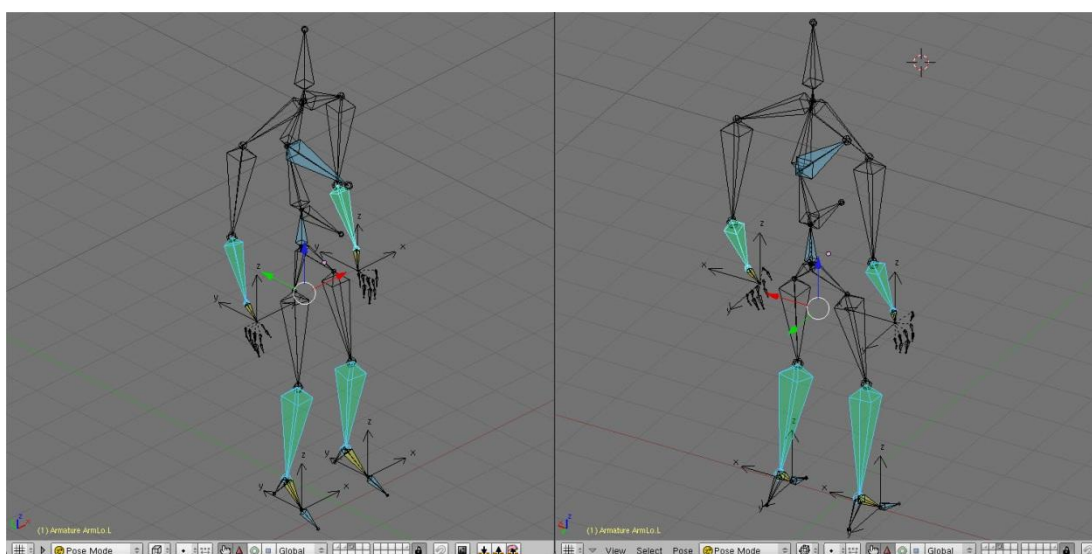
Važno je napomenuti da svaki vrh mora biti dodijeljen jednoj kosti inače nećemo moći eksportati model iz blendera u Ogre. To možemo provjeriti tako da selektiramo sve vrhove koji su dodijeljeni pojedinoj grupi te ih sakrijemo. Vrhovi koji ostanu nisu dodijeljeni niti jednoj grupi. Isto tako jedan vrh ne smije biti dodijeljen četiri ili više kostiju.

4.8. Inverzna kinematika

Kako bi si olakšali stvaranje animacije kretanja čovjeka dodat ćemo kostima „Foot.L“, „Foot.R“, „Hand.L“, „Hand.R“ ograničenje „IK Solver“ omogućujući time korištenje inverzne kinematike kod poziranja modela. Pri tome stvaramo i 4 objekta tipa „empty“ kojima ćemo kontrolirati ruke i noge. Svaki od tih objekata „empty“ stavljamo na vrh kosti kojima smo dodali ograničenje „IK Solver“ i nazovemo ih prikladnim imenima: „Leg.L“, „Leg.R“, „Arm.L“, „Arm.R“.

Zatim pod opcijom „Target:“ u ograničenju „IK Solver“ dodan kosti „Foot.L“ upišemo ime objekta empty „Leg.L“ koji smo pozicionirali na vrh te kosti „Foot.L“. Time smo omogućili kontroliranje lijeve noge pomoću objekta „Leg.L“. Isti postupak ponovimo za ostala tri ograničenja.

Još jedna bitna postavka ograničenja „IK Solver“ je „ChainLength“. Prvobitna postavka je nula što znači da će se pomicati sve kosti u lancu od kosti koja ima taj „IK Solver“ sve do izvorišne kosti. Mi to ne želimo te ćemo i za noge i ruke staviti broj tri. Tako smo odredili broj kostiju u lancu na koje će utjecati „IK Solver“ (Slika 4.8.1)



Slika 4.8.1 Ograničenja rotacije i inverzne kinematike

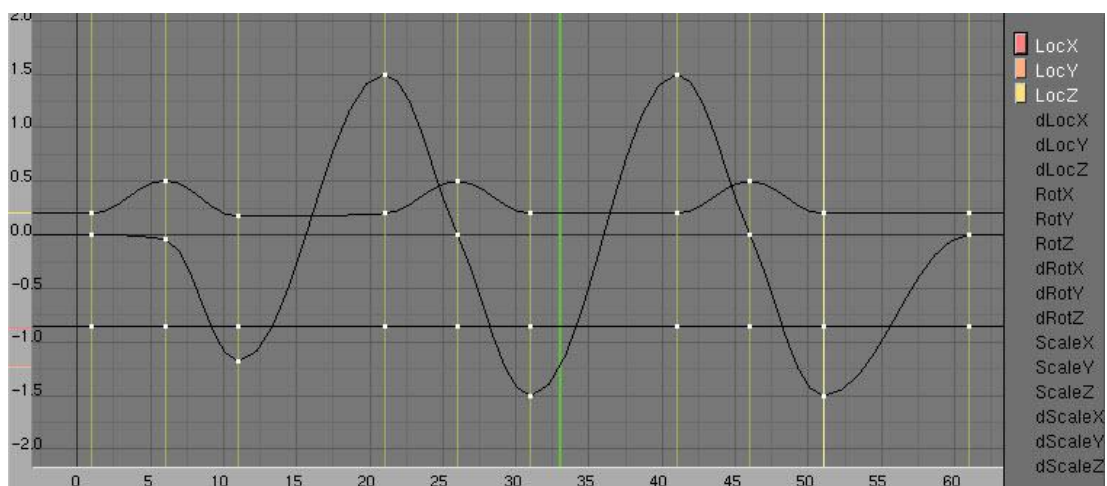
Kako bi pomicanje nogu i ruku prikazali što realnije, kostima „LegLo.L“, „LegLo.R“, „ArmLo.L“ i „ArmLo.R“ dodat ćemo ograničenje rotacije oko x, y i z osi tako da onemogućimo pokrete koji nisu mogući kod ljudi.

4.9. Animacija

Napravit ćemo animaciju kretanja čovjeka, točnije hodanja. Iz stajaćeg položaja krenut ćemo desnom nogom te ćemo se nakon nekoliko koraka zaustaviti. Animacija će se protezati kroz 61 sličicu (frames). Budući da koristimo inverznu kinematiku animaciju će biti vrlo lako napraviti.

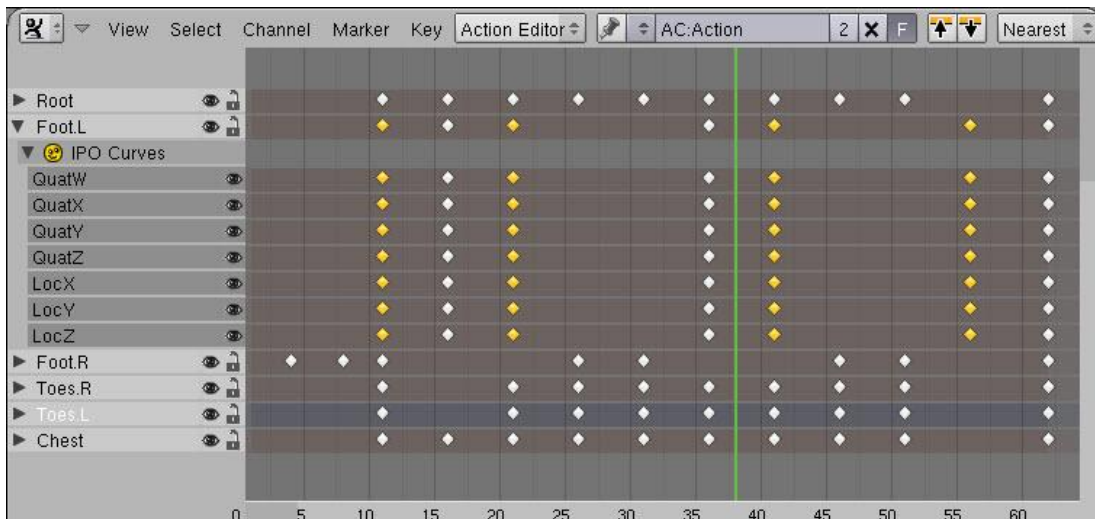
Prvi korak je da spremimo lokaciju (blender: l->Loc) sva četiri empty objekta u prvoj sličici u stajaćem položaju. Nakon toga pomaknemo se za 10 sličica unaprijed. Pomaknemo ruke i noge u položaj prvog koraka pomoću

empty objekata te spremimo njihovu lokaciju. Opet se pomaknemo za 10 sličica unaprijed, sada zamijenimo položaje ruku i nogu kako bi ostvarili drugi korak i opet spremimo lokaciju. Tako nastavimo do kraja animacije. Ovime smo ostvarili najosnovniju animaciju hodanja. U blenderu u „Ipo Curve Editor“ vidimo krivulje po kojima vidimo kako mijenjamo lokaciju empty objekata (Slika 4.9.1).



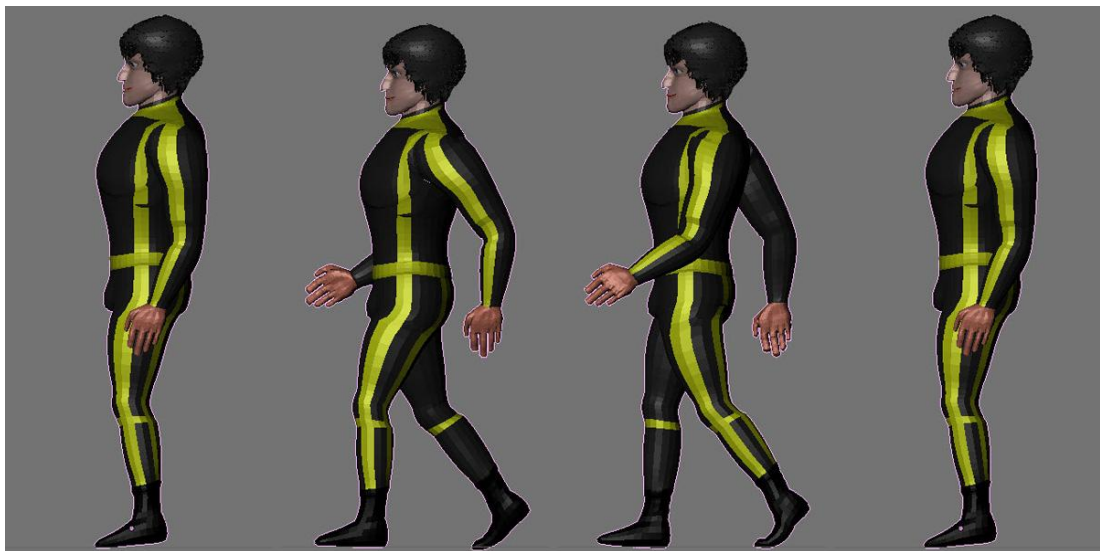
Slika 4.9.1 „Ipo Curve Editor“ za desnu nogu

Da čovjek ne hoda kao robot potrebne su dodatne modifikacije. Dodajemo geganje tako da izvorišnu kost pomaknemo u desno prije nego stanemo na desnu nogu i malo spustimo prema dolje, zatim vratimo u izvorišni položaj te pomaknemo lijevo i malo prema dolje prije nego stanemo na lijevu nogu. Pomicanjem kosti „Chest“ prema gore i dolje ostvarujemo disanje tijekom hodanja. Također moramo posvetiti pažnju na ponašanje stopala tokom hoda. Da bi ostvarili realniju animaciju svakih 5 sličica namjestimo izgled stopala pomicanjem i rotacijom kostiju stopala („Toes.L/R“, „Foot.L/R“) i spremimo njihovu lokaciju, ali i rotaciju. Spremljene ključne sličice (sličice u kojima smo spremili lokaciju i/ili rotaciju kostiju) za sve kosti vidljive su u „Action Editor“ (Slika 4.9.2).



Slika 4.9.2 „Action Editor“

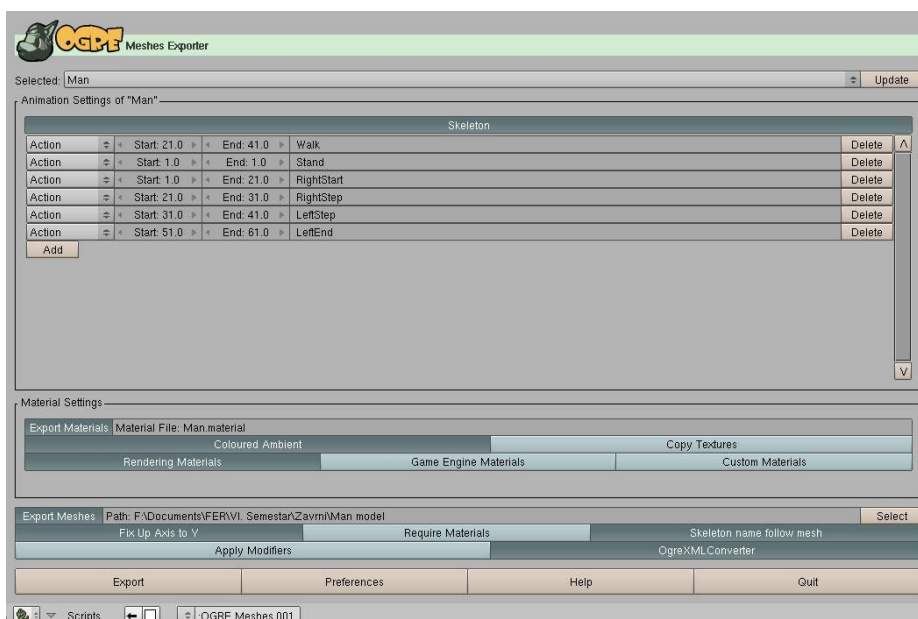
Potrebna su još dodatna dotjerivanja kao na primjer micanje glave, ramena i ostale sitnice međutim ovime smo pokazali osnove i nećemo ulaziti dublje u svijet animacije u blenderu.



Slika 4.9.3 Animacija hodanja

4.10. Učitavanje modela u Ogre

Da bi prikazali animaciju kretanja čovjeka u nekom drugom ambijentu učitat ćemo model u Ogre. Prvi korak je eksportiranje modela iz blendera. Za to će nam poslužiti skripta „OGRE Meshes Exporter“. Selektiramo mrežu modela i pozovemo skriptu na izvršavanje. Sada moramo namjestiti nekoliko postavki. Ako je model koji eksportiramo animiran imat ćemo opciju odabira svih animacija koje taj model posjeduje. Možemo odabrati bilo koju od njih i na temelju njih stvoriti animacije, koje će biti dostupne u Ogre-u preko imena koje im dodijelimo. Te nove animacije mogu biti identične ili sadržavati neki dio onih u blenderu definiranjem raspona sličica koje će nova animacija uzeti od njih. Zatim odaberemo način na koji će se eksportirati materijali. U našem slučaju „Game Engine Materials“. Odabrat ćemo i opcije „Fix Up Axis to Y“, kako bi uskladili koordinatne sustave i „OgreXMLConverter“ da se xml datoteke (ime_modela.mesh.xml, ime_modela.skeleton.xml) koje će generirati ova skripta automatski pretvore u datoteke iz kojih Ogre može učitati model. Te datoteke su „ime_modela.material“ koja sadrži podatke o materijalima modela, zatim „ime_modela.mesh“ koja sadrži podatke mreže kao popis vrhova, te „ime_modela.skeleton“ u kojoj se nalazi hijerarhijsko stablo kostura i zapis svih animacija dostupnih za dotični model.



Slika 4.10.1 Skripta „OGRE Meshes Exporter“

Kako u životu nije sve lako kao što se čini tako ni ovaj postupak nije iznimka. Nećemo uspješno eksportati model iz blendera ako:

- postoje vrhovi mreže koji nisu dodijeljeni niti jednoj kosti
- postoje vrhovi mreže koji su dodijeljeni četiri ili više kostiju
- modelu nisu dodijeljeni materijali

Budući da složeniji modeli imaju veliki broj vrhova rješavanje uzroka problema pod prvom točkom postaje mrkotrpan i dug posao. Da bi riješili taj problem moramo za svaku kost selektirati sve vrhove koji su njoj dodijeljeni te ih sakriti. Vrhovi koji ostanu nisu dodijeljeni niti jednoj kosti.

Također problem pri eksportanju mogu predstavljati dupli vrhovi koji su česta pojava u modeliranju. Međutim jednostavnom naredbom u „Edit Mode“ taj problem se brzo rješava (blender: A->W->Remove Doubles).

Nakon uspješnog eksportanja modela i učitavanja u Ogre zajedno sa stvaranjem ambijenta (teren, nebo, kućica, svjetlost, sjene) dobijemo prizor iz slike 4.10.2.



Slika 4.10.2 Učitavanje modela u Ogre

5. Zaključak

Stvaranje 3D animacije tehnikom skeletalnog animiranja iziskuje puno vremena i rada. Najveći dio vremena potreban je za stvaranje samog modela. Vrijeme potrebno za njegovo stvaranje se povećava ukoliko se pazi na svaki detalj kako bi bio što bolji, realniji. Također sama animacija tj. stvaranje poza i njihovo mijenjanje tokom vremena traži dosta posvećenosti i puno iskustva. Uvijek je prisutan osjećaj da treba unijeti neke promjene jer određeni pokret nije dovoljno realan. Kod animacije postoji mnogo trikova i dodatnih alata, koji nisu napomenuti u ovome radu jer nadilaze njegov zadatak, koji dodaju razne efekte animaciji, a uče se kroz nekoliko godina profesionalnim bavljenjem stvaranja animacije. U ovome radu nismo se bavili dodavanjem tekstura i njihovim stvaranjem, niti stvaranjem svijeta, tj. okoline, podešavanjem svijetlosti i ostalih sitnica koje itekako imaju utjecaja na konačnu animaciju kako bi ona bila što bolja i realnija.

Blender iako ne baš intuitivnog sučelja je vrlo moćan alat za stvaranje 3D modela i računalne animacije, a k tome je i besplatan. Naravno postoji još dosta stvari koje mu nedostaju da bi bio ravnopravan, ako ne i bolji od komercijalnih alternativa, međutim s obzirom na zajednicu koja stoji iza njega i brzinu razvijanja neće dugo još tako ostati.

Literatura

- Animation, <http://en.wikipedia.org/wiki/Animation>, 13.6.2010
- Cel-shaded animation, http://en.wikipedia.org/wiki/Cel-shaded_animation, 13.6.2010
- Morph target animation, http://en.wikipedia.org/wiki/Morph_target_animation, 13.6.2010.
- Motion Capture, http://en.wikipedia.org/wiki/Motion_capture, 13.6.2010
- Skeletal animation, http://en.wikipedia.org/wiki/Skeletal_animation, 13.6.2010
- Direct Kinematics, http://www-clmc.usc.edu/~cs545/Lecture_VIII.pdf, 14.6.2010
- Forward Kinematics: The Denavit-Hartenberg Convention, <http://www.cs.duke.edu/brd/Teaching/Bio/asmb/current/Papers/chap3-forward-kinematics.pdf>, 14.6.2010.
- Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods, <http://math.ucsd.edu/~sbuss/ResearchWeb/ikmethods/iksurvey.pdf>, 15.6.2010.
- Mike Tabaczynski - „Jacobian Solutions to the Inverse Kinematics Problem", <http://www.eecs.tufts.edu/~mtabac0a/IK/Proiect.pdf>, 15.6.2010.
- Zeljka Mihajlovic - „Unaprijedna i inverzna kinematika", http://www.zemris.fer.hr/predmeti/ra/predavanja/4_kinemat.pdf, 15.6.2010.
- Blender Essential Training, <http://www.lynda.com/home/DisplayCourse.aspx?lpk2=740>, 10.6.2010.

Sažetak

ANIMIRANE KINEMATIČKE STRUKTURE

Glavni zadatak ovog rada je upoznavanje s kinematičkim strukturama te njihovo animiranje na primjeru animacije kretanja čovjeka. Rad se sastoji od pet poglavlja, kroz koja se čitatelju pokušava objasniti ostvarenje glavnog zadatka.

Drugo poglavlje se bavi pojmom animacije, tipovima i tehnikama animiranja. Budući da se u radu koristimo 3D računalnom animacijom ona je i podrobnije objašnjena.

U trećem poglavlju upoznajemo se s kinematikom i njezinim tipovima: Unaprijedna i inverzna kinematika. Inverzna kinematika je detaljnije objašnjena jer je njezina primjena velika te se također koristi i u izradi animacije kretanja čovjeka.

U četvrtom poglavlju čitatelja se postepeno vodi kroz sve korake izrade modela čovjeka, svih potrebnih struktura za animaciju te samu animaciju kretanja čovjeka, tj. hodanja. Koristimo se programskim alatom Blender.

Ključne riječi: animacija, kinematičke strukture, direktna kinematika, inverzna kinematika, Blender, 3D model, Ogre, čovjek, hodanje.

Summary

ANIMATED KINEMATIC STRUCTURES

The main task of this work is to introduce the reader to kinematic structures and their animation on the example of human's walk. The work is composed of five chapters, which are trying to explain the realization of the task.

The second chapter deals with the term „animation“, types and techniques of animation. Since in this work we use 3D computer animation, it is explained in more detail.

In the third chapter we are introduced to the kinematics and its types: Forward and inverse kinematics. Inverse kinematics is explained in more detail because of its great application and is also used in the production of human's walk.

In the fourth chapter, the reader is gradually led step-by-step through man modeling, all the necessary structures for animation, and the animation of human's walk. We use a software tool Blender.

Keywords: animation, kinematic structures, direct kinematics, inverse kinematics, Blender, 3D model, Ogre, human, walk.