

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 860

SIMULACIJA HODA PAUKA

Željko Kelava

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 860

SIMULACIJA HODA PAUKA

Željko Kelava

Zagreb, lipanj 2023.

ZAVRŠNI ZADATAK br. 860

Pristupnik: **Željko Kelava (0036529933)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentorica: prof. dr. sc. Željka Mihajlović

Zadatak: **Simulacija hoda pauka**

Opis zadatka:

Proučiti načine kretanja pauka. Izraditi model pauka te pripadni skeletni model koji omogućuje upravljanje pokretima pauka. Posebice obratiti pažnju na složenije primjere vizualnih učinaka. Izraditi realistične animacije kretanja pauka. Razmotriti mogućnosti korištenja skripti u upravljanju pokretima. Diskutirati utjecaj različitih parametara. Načiniti ocjenu rezultata i implementiranih algoritama. Izraditi odgovarajući programski proizvod. Koristiti grafički programski alat Blender. Rezultate rada načiniti dostupne putem Interneta. Radu priložiti algoritme, izvorne kodove i rezultate uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 9. lipnja 2023.

Sadržaj

Uvod	1
1. Korištene tehnologije i alati.....	2
1.1. Blender	2
1.2. Python proširenje bpy	2
2. Izrada modela pauka.....	3
2.1. Tijelo pauka	3
2.2. Noge pauka	9
2.3. Materijal pauka	13
3. Izrada skeleta pauka.....	15
3.1. Stvaranje kostiju	15
3.2. Inverzna kinematika	18
3.3. Spajanje skeleta i modela	20
4. Simuliranje hoda.....	23
4.1. Primjer Izvođenja	24
4.2. Funkcija micanja	25
4.3. Funkcija rotiranja.....	28
Zaključak	30
Literatura	31
Sažetak.....	32
Summary.....	33

Uvod

Simuliranje hoda životinja može biti korisno u puno različitih područja kao što su razvoj robotske tehnologije, biomehanike i animacija. Koristi se za analizu njihove građe i fiziologije, dizajniranje robotskih sustava inspiriranih kretanjem pravih životinja te unaprjeđenje animacija u filmskoj industriji i industriji videoigara.

Za rješavanje ovakvog problema potrebno je dobro razumjeti anatomsku strukturu životinje. Osim toga potrebno je razumijevanje mehanizama njihovih pokreta i koordinacije nogu. Današnja tehnologija omogućuje inženjerima da stvore precizne modele i simulacije koje vjerno repliciraju pokrete tijela životinja. Uzimajući u obzir kako različiti dijelovi tijela rade zajedno, kako mišići djeluju te kako se energija prenosi u udove moguće je stvoriti realistične modele hodanja životinja. Te informacije je onda moguće iskoristiti za izradu robota koji se ponašaju kao prave životinje.



Prikaz četveronožnog robota Spot

Kroz ovaj rad opisan će se postupak izrade modela pauka, izrade skeletnog modela i njihovo spajanje. Također opisan će se postupak pomicanja nogu te izrada skripte koja simulira hodanje pauka.

1. Korištene tehnologije i alati

Ovaj rad koristi alat Blender. Pomoću Blendera izradit ćemo 3D model pauka, a zatim ćemo napraviti i skeletni model koji ćemo pridružiti 3D modelu. Na kraju ćemo koristiti Python proširenje (*module*) bpy pomoću kojega ćemo izraditi animacije.

1.1. Blender

Blender je besplatni program otvorenog kôda. Koristi se za stvaranje animiranih filmova, vizualnih učinaka, interaktivnih 3D aplikacija, virtualne stvarnosti i još puno toga. Značajke koje Blender pruža su 3D modeliranje, UV preslikavanje, teksturiranje, rastersko uređivanje slika, izrada skeleta, simulacija fluida, simulacija čestica, animiranje, iscrtavanje, video uređivanje itd. Veliki broj značajki čini ga popularnim kod umjetnika, dizajnera i animatora. Blenderova licenca omogućuje korištenje alata u bilo koju svrhu uključujući edukacijsku i komercijalnu.

Verzija Blendera korištena u ovom radu je Blender 3.5.0.

1.2. Python proširenje bpy

Python je interpretacijski programski jezik visoke razine poznat po jednostavnosti i čitljivosti. Jedan je od „lakših“ programskih jezika za početnike. Unatoč tome koristi se u velikome broju različitih područja uključujući analizu podataka, umjetnu inteligenciju, razvoj web poslužitelja, itd. Velik broj vanjskih proširenja doprinose njegovoj popularnosti.

Blender pruža proširenje bpy. To nam omogućuje da koristimo Python kako bi pisali kôd za upravljanje svim elementima scene. Moguće je transformirati objekte, postavljati svojstva kao materijale i modifikatore te postavljati *keyframeove* animacije.

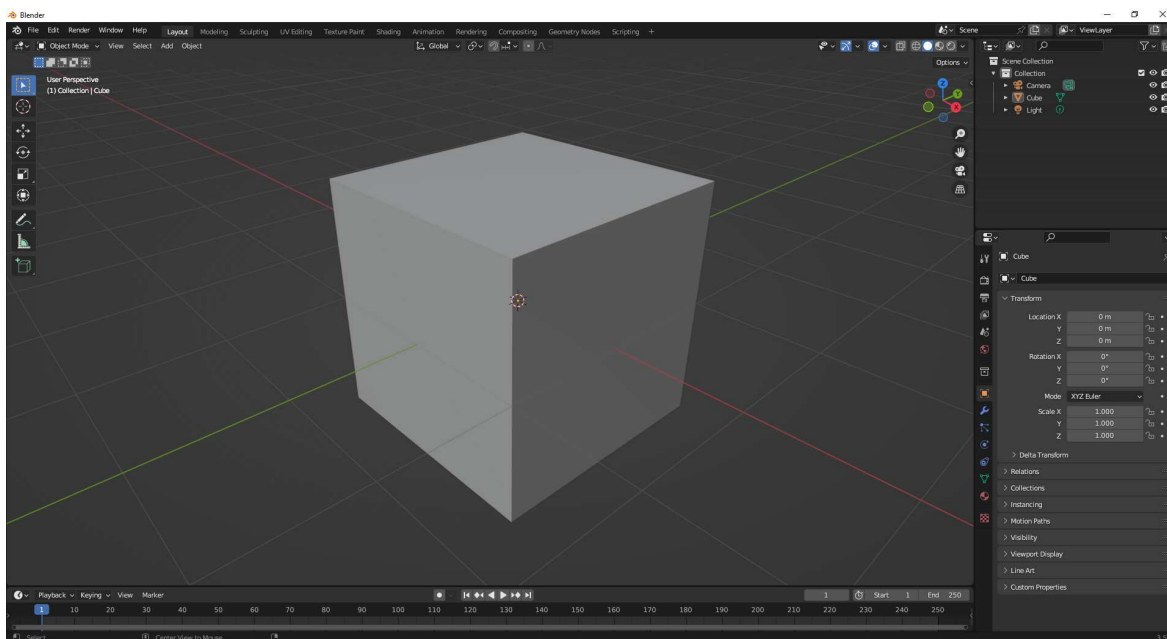
Verzija Pythona korištena u ovom radu je Python 3.7.

2. Izrada modela pauka

Pauk je životinja koja pripada koljenu člankonožaca. To koljeno specifično je po kolutićavom tijelu i člankovitim nogama. Tijelo im je prekriveno hitinskim oklopom. Do sada je otkriveno preko 50,000 vrsta pauka. Anatomski je prepoznatljiv po tome što mu je tijelo podijeljeno na dva dijela spojena drškom. Ima osam nogu, a svaka noga je podijeljena na 7 dijelova.

2.1. Tijelo pauka

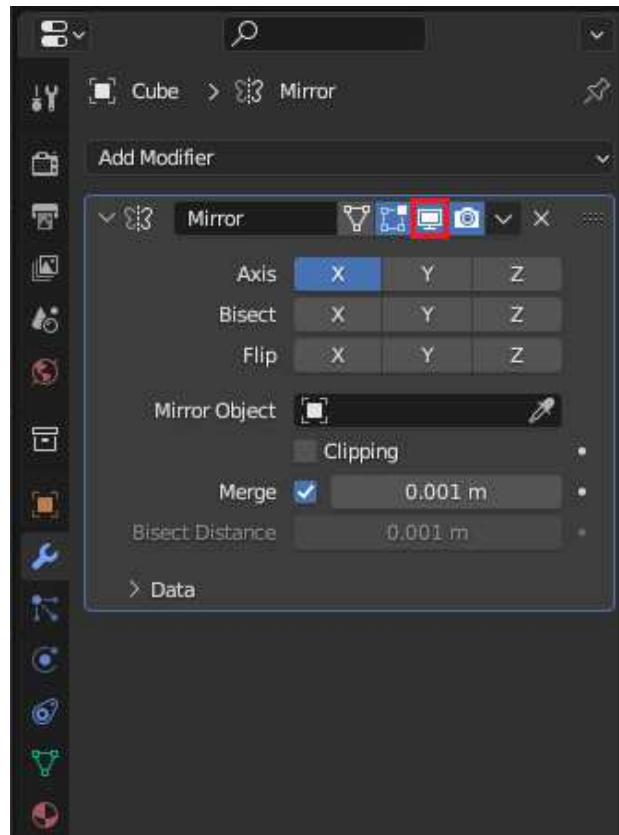
U ovom dijelu opisat ćemo izradu modela tijela. Kada prvi put upalimo Blender i napravimo novi projekt dobijemo običnu kocku (Slika 2.1). Dodatno, kocka se može dodati u scenu tako da na vrhu ekrana izaberemo opciju *Add -> Mesh -> Cube*. Na sličan način dodaju se i ostali objekti u scenu kao svjetlo, kamera, krivulje, itd.



Slika 2.1: Prikaz početne kocke

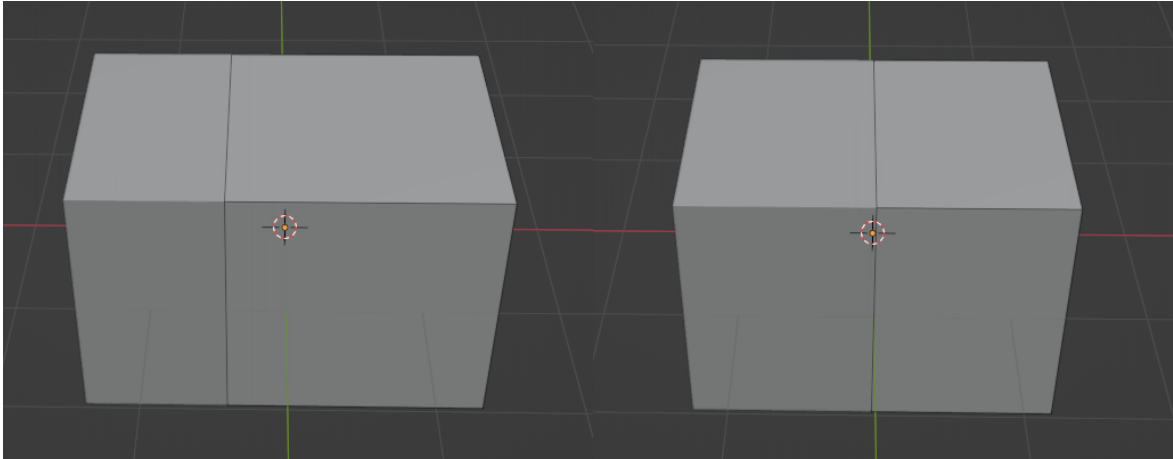
Sada ćemo na kocku dodati *modifier*. S izbornika na desnoj strani ekrana treba odabrati plavu ikonu koja izgleda kao ključ. Zatim opciju *Add Modifier* i pod izbornikom *Generate* odabrati *Mirror* modifier. Ovaj atribut koji smo dodali na objekt će nam služiti da sve promjene koje napravimo na jednoj strani modela budu zrcaljane na drugu stranu

također. Treba postaviti da se zrcaljenje odvija po x-osi. Opciju *Clipping* još nećemo uključiti. Sve postavke koje je potrebno postaviti prikazane su na slici 2.2.



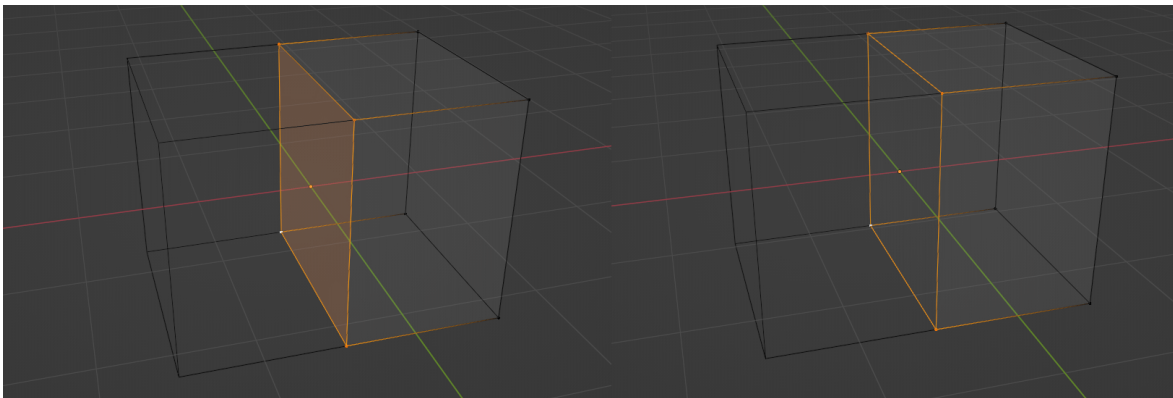
Slika 2.2: Postavke za *Mirror modifier*

Do sada smo radili u načinu rada *Object mode*. Taj način rada služi da upravljamo objektima kao cjelinama što uključuje micanje, skaliranje i rotiranje. Sada je potrebno odabrati opciju rada *Edit mode*. U ovom načinu rada ćemo činiti sve promjene nad našim modelom. Promjene koje radimo su modificiranje geometrije tijela. To može biti micanje vrhova, bridova ili dodavanje novih poligona. Da bi naš *Mirror modifier* radio ispravno potrebno je kocku pomaknuti po x-osi toliko da naš zrcalni model ne dodiruje pravi. Zatim je potrebno uključiti opciju *Clipping* i ponovno približiti kocku prema sredini tako da se prava i zrcalna kocka spoje u jednu cjelinu. Ovo smo morali napraviti tako da rub naše kocke leži na yz-ravnini (Slika 2.3).



Slika 2.3: S lijeve strane pogrešno postavljeno zrcaljenje, a s desne ispravno

Problem koji se sada pojavljuje je da unutar našeg modela postoji ploha koja ga siječe na 2 dijela. To je posljedica toga da smo krenuli raditi model od kocke. Problem se lagano može riješiti. Potrebno je isključiti *Mirror modifier*, i pobrisati srednju plohu. Zrcaljenje možemo isključiti pritiskom na tipku označenu crvenim kvadratom na slici 2.2. Sada treba selektirati plohu koju želimo obrisati. Na vrhu ekrana možemo odabrati način selektiranja koji želimo koristiti. Potrebno je odabrati način rada s plohama. Sada lijevom klikom miša na srednju plohu ona postaje selektirana. Nakon toga, pritiskom tipke *x* na tipkovnici otvori se izbornik. Odabirom na opciju *Faces* briše se srednja ploha. Ponovno treba uključiti zrcaljenje. Ako sada promijenimo prikaz scene na *Wireframe*, možemo vidjeti da više nema plohe u sredini (Slika 2.4).

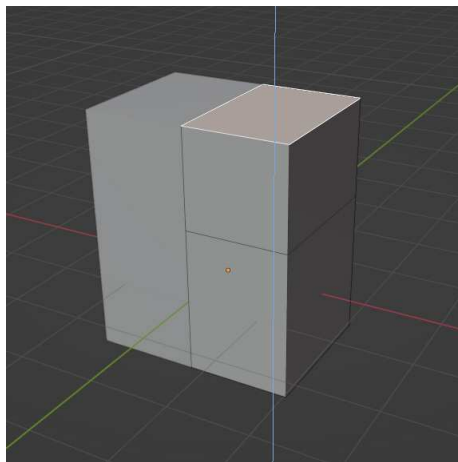


Slika 2.4: S lijeve strane postoji ploha u sredini, a s desne ne

Sada možemo početi modelirati tijelo pauka. Korisno je znati tipkovne prečace za translaciju, rotaciju i skaliranje. Lijevim klikom miša odabiremo vrh, brid ili plohu. Ako pri odabiru držimo tipku *SHIFT*, onda će se naša postojeća selekcija proširiti s novim objektom. Svi odabrani objekti obojeni su narančastom bojom. Ako želimo translirati

objekte pritisnut ćemo tipku *g* na tipkovnici. Ako želimo tu translaciju napraviti samo po nekoj osi onda ćemo nakon pritiska tipke *g*, pritisnuti još *x*, *y* ili *z*. Znači pritiskom kombinacije tipki *g* i *z*, odabrane objekte transliramo po *z*-osi. Prečac za rotaciju je tipka *r*, a za skaliranje tipka *s*. Također nakon tipki *r* i *s* može se pritisnuti *x*, *y* ili *z* tako da se operacija odvija samo po jednoj osi.

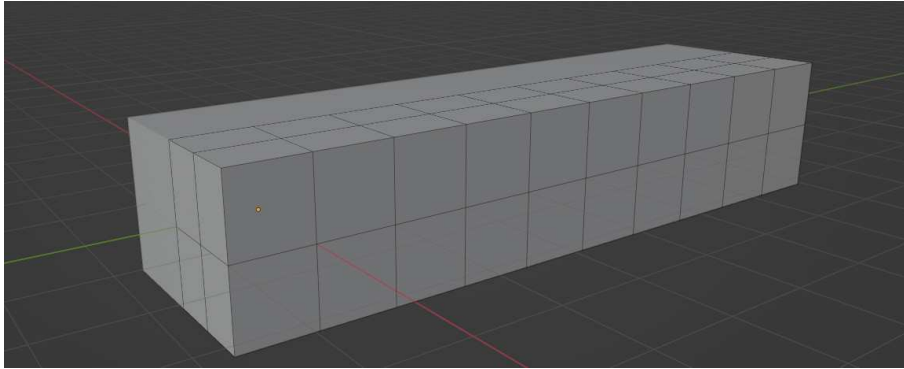
Još jedna operacija koja nam je potrebna je operacija *Extrude Region*. Prečac je tipka *e*. Ova operacija iz naše selekcije stvara nove točke, bridove ili plohe. Način na koji ju koristimo je da odaberemo plohu koju želimo „izvući“ (Slika 2.5). Ovom operacijom dodajemo novu geometriju tijela.



Slika 2.5: Primjer korištenja operacije *Extrude Region*

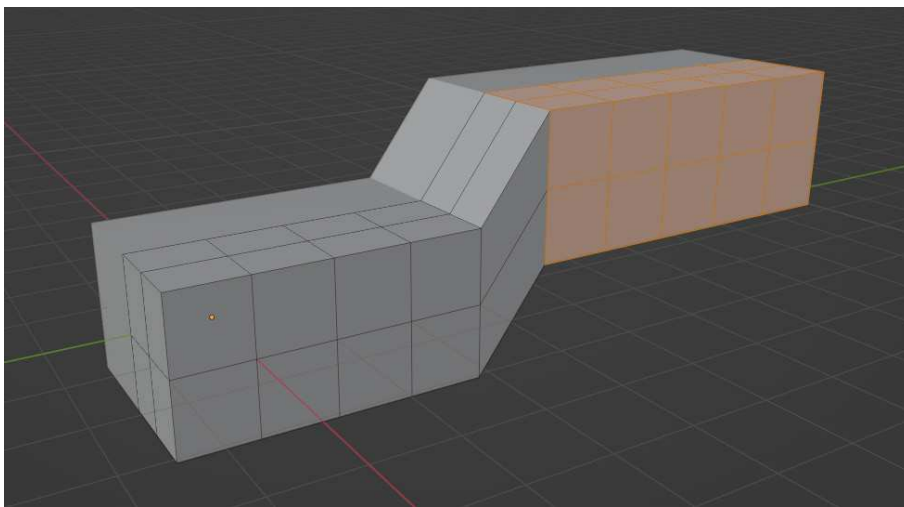
Zadnja operacija koju ćemo koristiti u modeliranu je *Subdivide*. Odabirom neke plohe i pritiskom desnog klika pojavljuje se izbornik u kojem možemo odabrati opciju *Subdivide*. Tom operacijom dijelimo plohu na više dijelova. Služi nam da detaljnije možemo doraditi naš model.

Od naše kocke želimo otprilike dobiti izgleda tijela pauka. Prvo ćemo nad par ploha napraviti *Subdivide*, a zatim koristeći operaciju *Extrude* izvući tijelo pauka u dubinu da dobijemo okvirni oblik tijela (Slika 2.6)



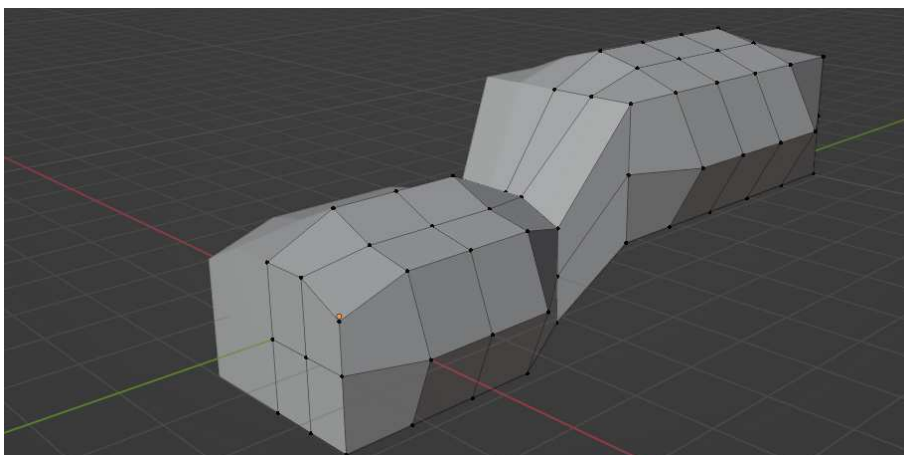
Slika 2.6: Prikaz početnog oblika tijela

Sada ćemo odabrati stražnje poligone i translirati ih po z-osi (Slika 2.7)



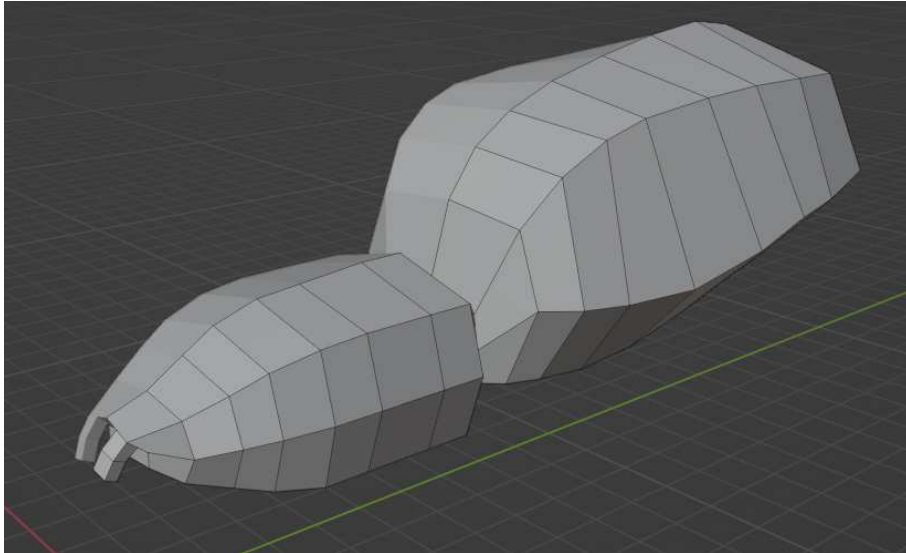
Slika 2.7: Prikaz transliranja stražnjeg dijela modela

Naš model daleko je od izgleda pauka, ali do sada smo upoznali sve potrebne operacije da ga modeliramo. Nizom operacija skaliranja i transliranja vrhova možemo zaobliti model tako okvirno podsjeća na tijelo pauka (Slika 2.8).



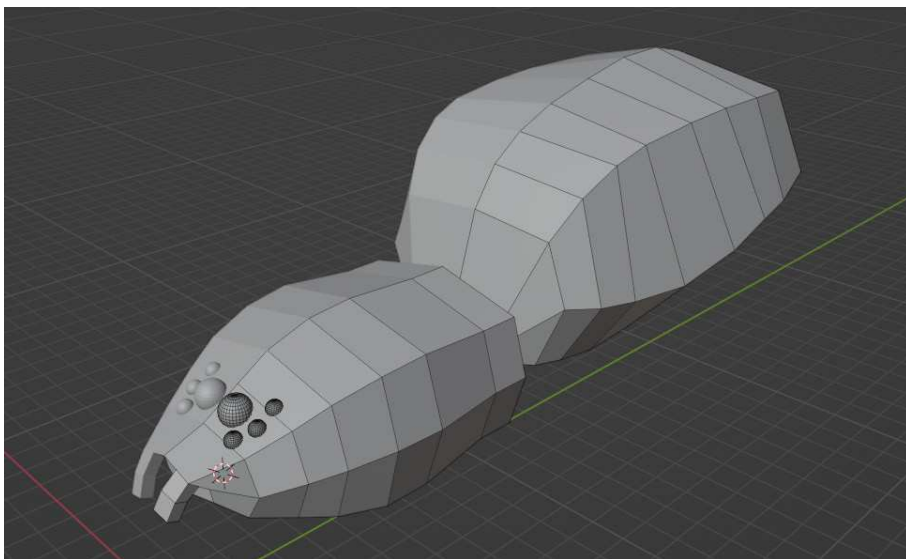
Slika 2.8: Prikaz grubog oblika pauka

Ako nastavimo primjenjivati ove postupke transliranja i skaliranja vrhova s vremenom možemo dobiti model koji izgleda kao tijelo pauka. Naravno možemo još uvijek koristiti operacije *Extrude* i *Subdivide*. Na slici 2.9 vidi se konačna geometrija tijela pauka.



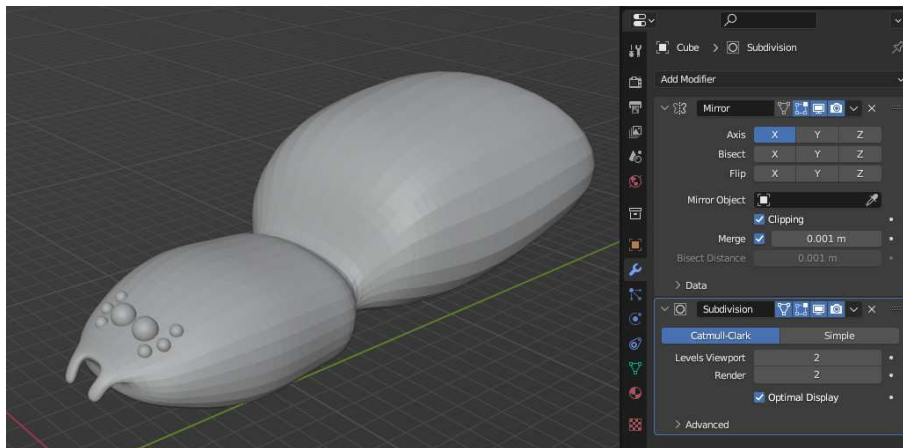
Slika 2.9: Konačni prikaz tijela pauka

Još preostaje pauku dodati oči. To činimo tako da dok smo u *Edit mode* načinu rada na vrhu ekrana izaberemo opciju *Add -> UV Sphere*. Sfera će se dodati na poziciju *3D Cursor*. Treba paziti da se *3D Cursor* pomakne iz ishodišta sustava jer ćemo inače dodati sferu koju će naš *Mirror modifier* zrcaliti i nećemo ju moći maknuti iz ishodišta. Dodanu sferu ćemo skalirati i translirati tako da ona čini oko pauka. To ćemo ponoviti još tri puta za ukupno osam očiju (Slika 2.10).



Slika 2.10: Prikaz tijela pauka s očima

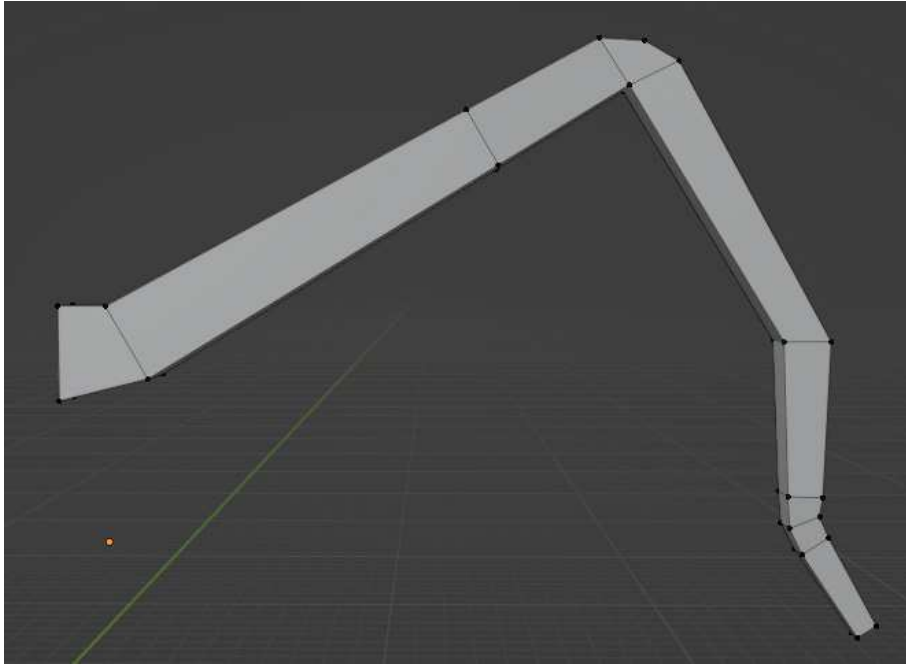
Trenutačni model pauka izgleda grubo i nije detaljan. Postoji jednostavno rješenje za ovaj „problem“. Isto kao što smo dodali *Mirror modifier*, dodat ćemo *Subdivision modifier*. Ovaj atribut ne mijenja originalnu geometriju objekta, ali iterativno dodaje nove vrhove i poligone tako da naš objekt postane više gladak. Možemo odrediti koliko iteracija podjele želimo s obzirom na detalje koje želimo imati. Za ovaj model dovoljno je odabrati 2 iteracije da tijelo pauka izgleda glatko. Dobili smo konačni model tijela pauka kojem treba još dodati noge (Slika 2.11).



Slika 2.11: Prikaz konačnog modela pauka

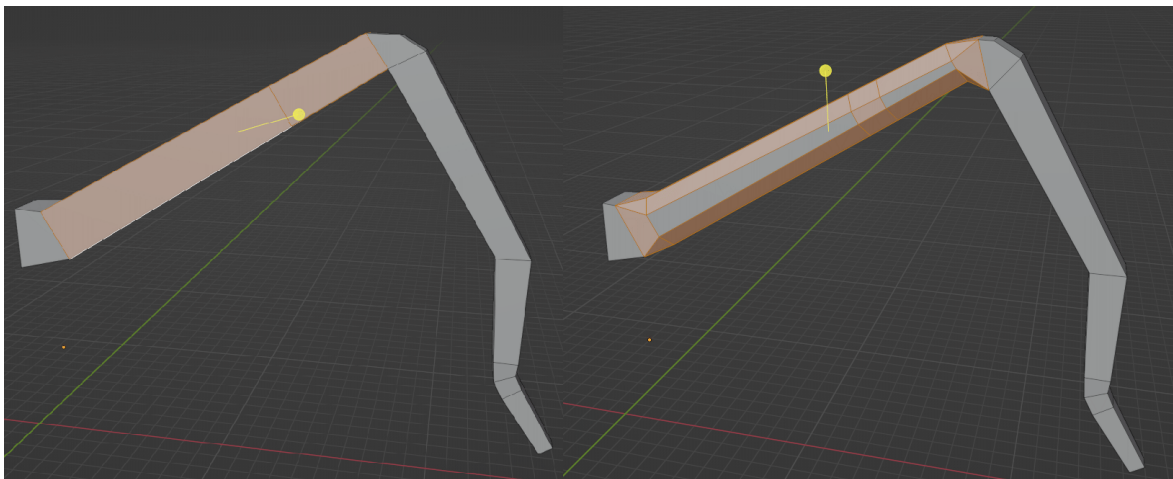
2.2. Noge pauka

Da bi dovršili cijeli model pauka ostaje nam još modelirati noge. Modelirat ćemo jednu nogu i onda ju kopirati. Prebacit ćemo način rada u *Object mode* i s desne strane pritisnuti ikonu oka da sakrijemo model tijela. Dodat ćemo novu kocku tako da na vrhu ekrana izaberemo opciju *Add -> Mesh -> Cube*. Dok imamo selektiranu kocku prebacit ćemo način rada ponovno u *Edit mode*. Sada treba koristiti iste načine modeliranja kao i kod izrade tijela. Jednostavnim rotiranjem ploha, korištenjem alata *Extrude* i skaliranjem moguće je napraviti model noge kao na slici 2.12.



Slika 2.12: Prikaz modela noge

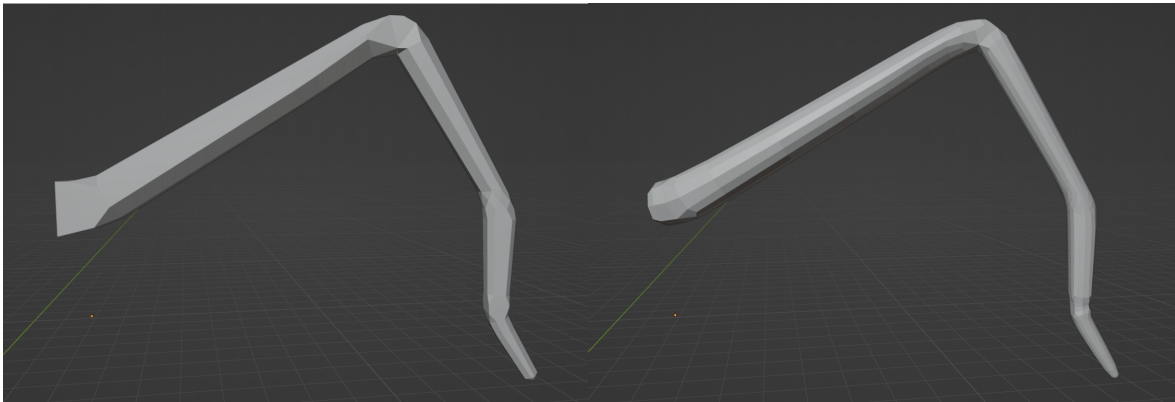
Trenutni model noge je grub i „kockast“. Koristit ćemo alat *Bevel* da bi zaoblili model. Kada koristimo alat *Bevel* nad plohom dodajemo novu geometriju. Dodaje se zaobljeni profil oko plohe tako da se originalna ploha smanji, a točke koje su činile te plohu sada čine 4 nove plohe koje su spojene s originalnom. Ako odaberemo više ploha od jednom možemo dobiti učinak zaobljenosti koji želimo. Ovaj postupak napraviti ćemo na više dijelova noge da dobijemo više „cilindričan“ oblik. Slika 2.13 prikazuje rezultat korištenja alata *Bevel*.



Slika 2.13: Prikaz modela noge prije (slika lijevo) i nakon (slika desno) korištenja alata *Bevel*

Ovaj postupak zaobljenosti mogli smo dobiti dodavanjem *Subdivision modifiera*. To ćemo i napraviti, ali tek kada sami zaoblimo noge alatom *Bevel*. Blenderov atribut nije

savršen i bolje radi ako ima veći broj točaka po kojima može računati svoj algoritam za podjelu. Slika 2.14 prikazuje konačan model noge prije korištenja *Subdivision modifiera* i nakon.



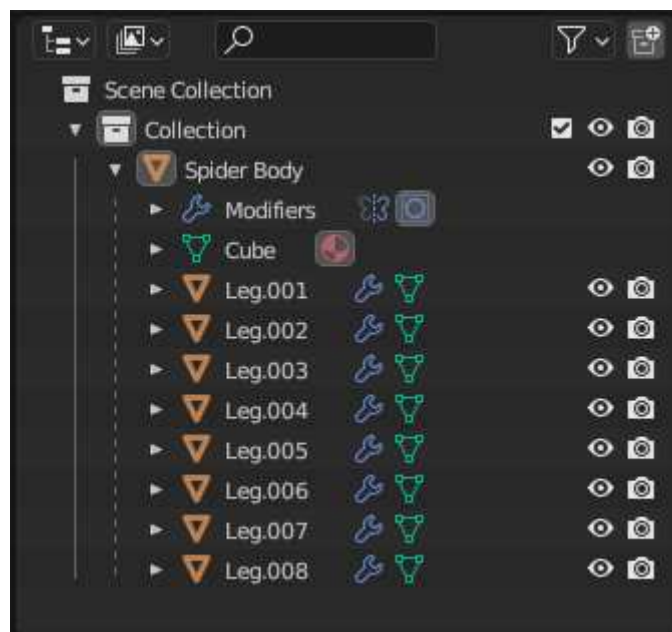
Slika 2.14: Konačni model noge bez (slika lijevo) i sa (slika desno) *Subdivision modifierom*

Sada u istoj sceni imamo jedan model noge i model tijela pauka. Način rada treba prebaciti u *Object mode*. Kopirat ćemo nogu pauka 3 puta i postaviti noge na tijelo. Translacijom, skaliranjem i rotacijom, noge ćemo postaviti samo s jedne strane pauka. Drugu stranu dobit ćemo tako da zrcalimo noge po x-osi. Kod operacije zrcaljenja važna je ishodišna točka objekta *Origin*. Svaki zasebni objekt u Blenderu ima ishodišnu točku (*Origin*). Kada odabiremo više objekata, koordinate koje Blender ispisuje su zapravo koordinate točke koja je prosjek svih ishodišnih točaka odabranih objekata. Uzmimo za primjer da jedna ishodišna točka ima x koordinatu -5, a druga točka x koordinatu 5, ako selektiramo oba objekta, njihova x koordinata je 0. To znači da možemo odabrati 4 noge pauka s jedne strane, pročitati njihovu x koordinatu, zrcaliti noge preko x-osi i onda za te nove 4 noge postaviti na x koordinatu suprotne vrijednosti. Zrcaljenje se odvija pritiskom tipki *CTRL + m*, a zatim tipka *x*, da bismo zrcalili preko x-osi. Rezultat ovih operacije je da imamo simetrično tijelo pauka i njegovih 8 nogu (Slika 2.15).



Slika 2.15: Prikaz konačnog modela pauka s nogama

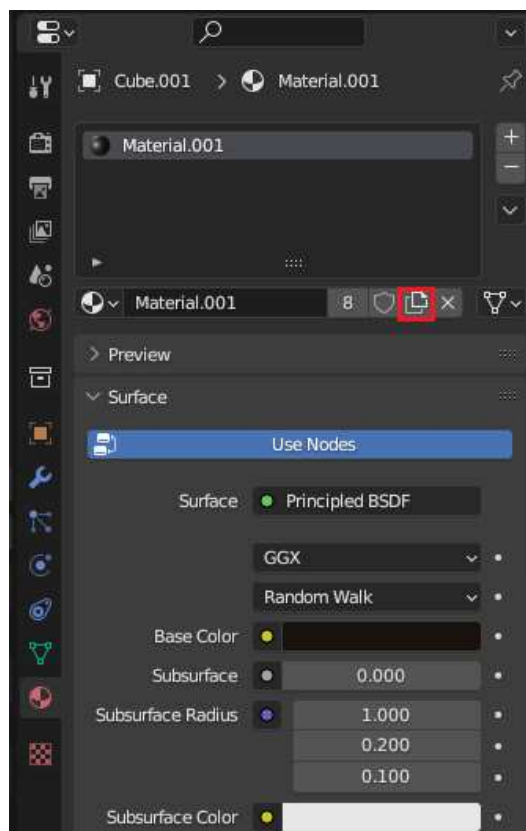
Potrebno je još promijeniti hijerarhiju objekata, odnosno potrebno je tijelo pauka napraviti roditeljem noga. To se pravi tako da prvo mišem odaberemo sve noge, a kao zadnji objekt izaberemo tijelo pauka. Zatim se kombinacijom tipki *CTRL + p* otvara izbornik u kojem možemo izabrati opciju *Set parent to Object (Keep Transform)*. S desne strane ekrana u hijerarhiji možemo vidjeti odnos između objekata (Slika 2.16).



Slika 2.16: Prikaz hijerarhije objekata

2.3. Materijal pauka

Da bi dovršili cijeli model pauka ostaje mu još dodati materijal. Prvo ćemo mišem odabrati jednu nogu pauka. Zatim ćemo s desne strane ekrana pod *Properties* odabrati ikonu crvene sfere. Pritiskom na tipku *New* dodajemo novi materijal za odabranu nogu. U izborniku pod *Surface* potrebno je promijeniti vrijednost *Base Color* (Slika 2.17). Nakon što promijenimo boju, tu promjenu još uvijek nećemo vidjeti. Potrebno je prebaciti scenu u način rada *Material Preview* ili *Rendered*. Sada kada želimo pridružiti taj materijal nekoj drugoj nozi, potrebno je samo iz padajućeg izbornika odabrati materijal koji smo već napravili.



Slika 2.17: Vrijednosti parametara za materijal noge

Za bojanje tijela postupak je sličan. Jedina razlika je što oči pauka pripadaju istom objektu kao i cijelo tijelo, a želimo da oči budu drugačije boje od tijela. Potrebno je na objektu tijela izraditi još jedan materijal pritiskom na gumb označen crvenim kvadratom na slici 2.17. Tijelo sada ima 2 materijala. Pritiskom tipke *a* na tipkovnici odabiru se svi poligoni objekta. Sve poligone koji nisu dio očiju potrebno je maknuti iz trenutnog odabira. Pritiskom na tipku *Assign* oči su obojene bojom drugog materijala. Rezultat dodjeljivanja materijala pauku vidi se na slici 2.18.



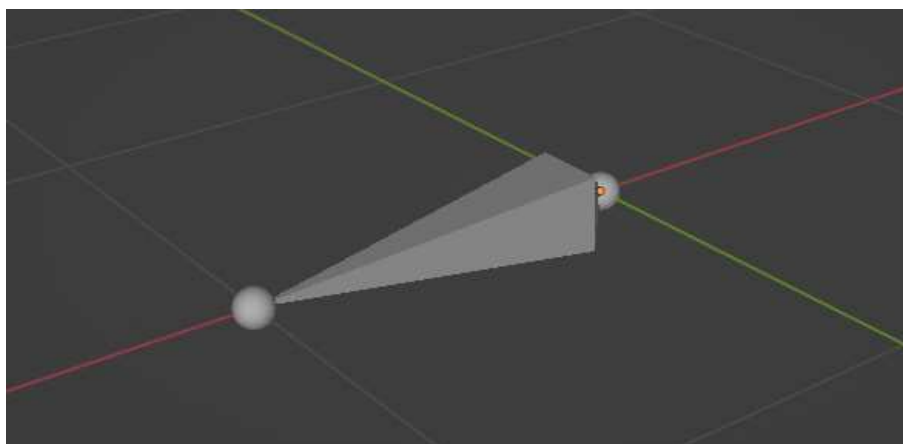
Slika 2.18: Rezultat dodavanja materijala modelu pauka

3. Izrada skeleta pauka

U ovom dijelu rada opisat ćemo način na koji možemo izraditi skeletalni model. Skeletalni model, poznat i kao armatura, služi nam prvenstveno za 3D animaciju. Sastoji se od serije kostiju povezanih zajedno. Pomicanjem kostiju možemo stvarati kompleksne pokrete, deformacije i animacije likova ili objekata u Blenderu. Moguće je postaviti ograničenja za svaku kost, poput rotacijskih ili translacijskih granica, kako bi se kontrolirao raspon njihovih pokreta. Nakon izrade skeleta potrebno ga je pridružiti modelu pauka.

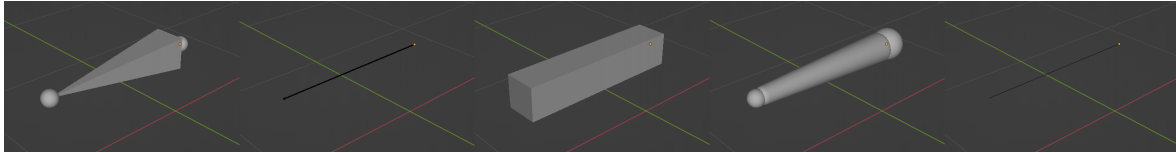
3.1. Stvaranje kostiju

Na početku ćemo postaviti način rada u *Object mode*. Zatim ćemo u scenu dodati prvu kost tako da na vrhu ekrana izaberemo opciju *Add -> Armature*. Kost koju smo dodali sastoji se od tri dijela (Slika 3.1). To su dvije sfere i piramida koja ih povezuje. Ukupno izgleda kao velika strelica. Dvije sfere predstavljaju početak i kraj kosti, a srednji dio određuje smjer i duljinu kosti. Dodanu kost rotirat ćemo tako da leži na svojoj strani.



Slika 3.1: Prikaz kosti

Postoji više različitih prikaza kostiju koji mogu biti korisni, ovisno o vrsti posla ili zadatka koji se obavlja (Slika 3.2).

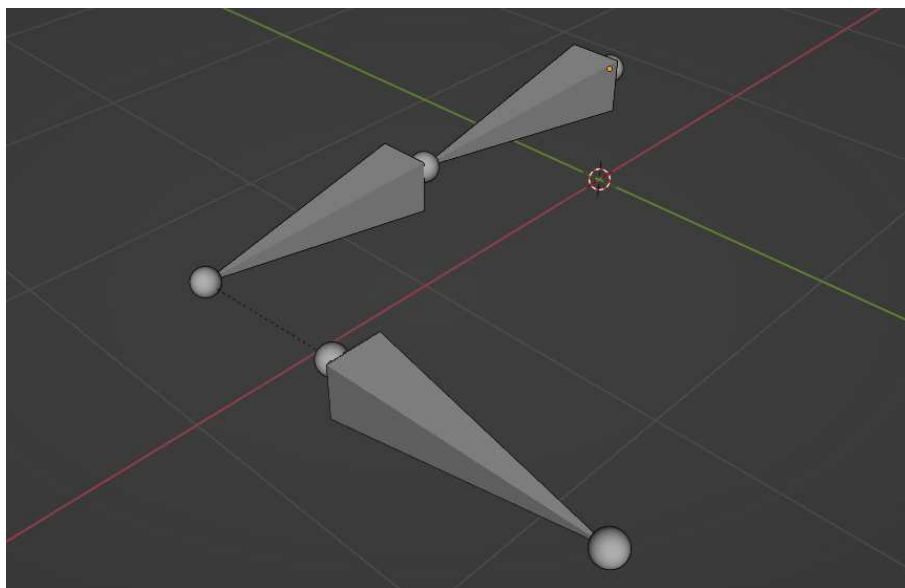


Slika 3.2: Različiti prikazi iste kosti u Blenderu

Da bismo na ovu kost dodali još kostiju potrebno je prijeći u način rada *Edit mode*. Kost koju smo već stvorili koristit ćemo da kontroliramo stražnji dio pauka. Mišem treba odabrati prednju sferu te koristiti alat *Extrude* kako bi dodali novu kost koja se nastavlja na postojeću. Sada imamo dvije kosti koje su povezane jedna na drugu. Ove dvije kosti koristit će se za upravljanjem tijela pauka.

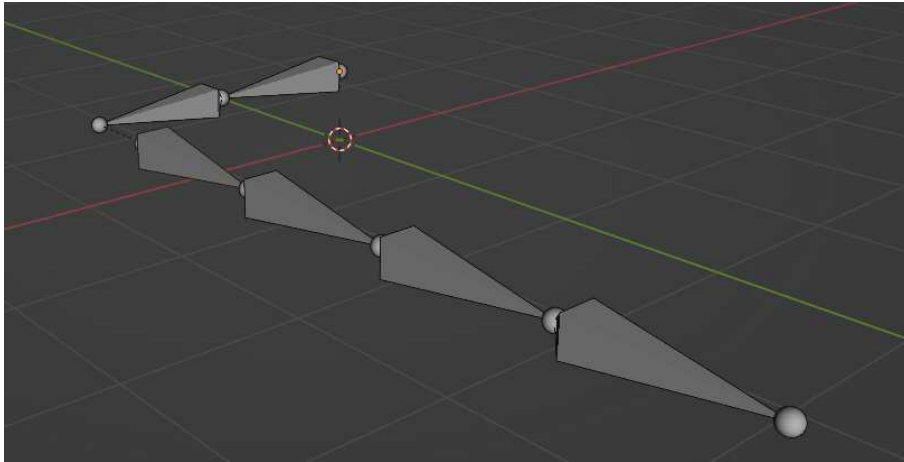
Sljedeće što je potrebno napraviti je dodati kosti koje će predstavljati noge pauka. Kost ne moraju nužno biti povezane jedna za drugu, ali još uvijek mogu biti dio istog skeleta. Prvo ćemo izraditi kosti za jednu nogu pauka.

Dok smo u načinu rada *Edit mode*, dodat ćemo novu kost tako da na vrhu ekrana izaberemo opciju *Add -> Single Bone*. Novu kost translirat ćemo i rotirati tako da otprilike dođe na mjesto gdje bi počela noga pauka (Slika 3.3). Nije bitno da ovaj postupak bude precizan jer ćemo kasnije svaku kost ručno postaviti na njeno mjesto. Mišem ćemo odabrati novu kost, a zatim kost koja predstavlja tijelo i pritisnuti kombinaciju tipki *CTRL + p*. U otvorenom izborniku izabrat ćemo *Make Parent Keep Offset*. To smo napravili tako da kosti mogu imati utjecaj jedna na drugu, a ne moraju biti spojene direktno.



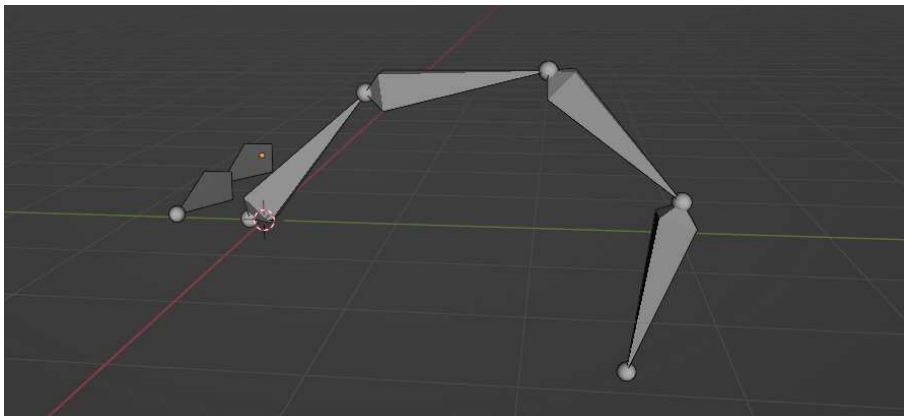
Slika 3.3: Prikaz kosti koje predstavljaju tijelo i početak paukove noge

Sada možemo na ovu kost noge dodati još kostiju koristeći alat *Extrude* jer želimo da nove kosti budu povezane za postojeću. Noga našeg 3D modela pauka ima 4 glavna dijela. Zbog toga će svaka noga imati 4 kosti (Slika 3.4).



Slika 3.4: Kostiju za tijelo pauka i jedne njegove noge

Prelaskom u *Pose mode* način rada već sada moguće je upravljati kostima. Potrebno je posebno odabrati svaku kost i postaviti njenu rotaciju (Slika 3.5).



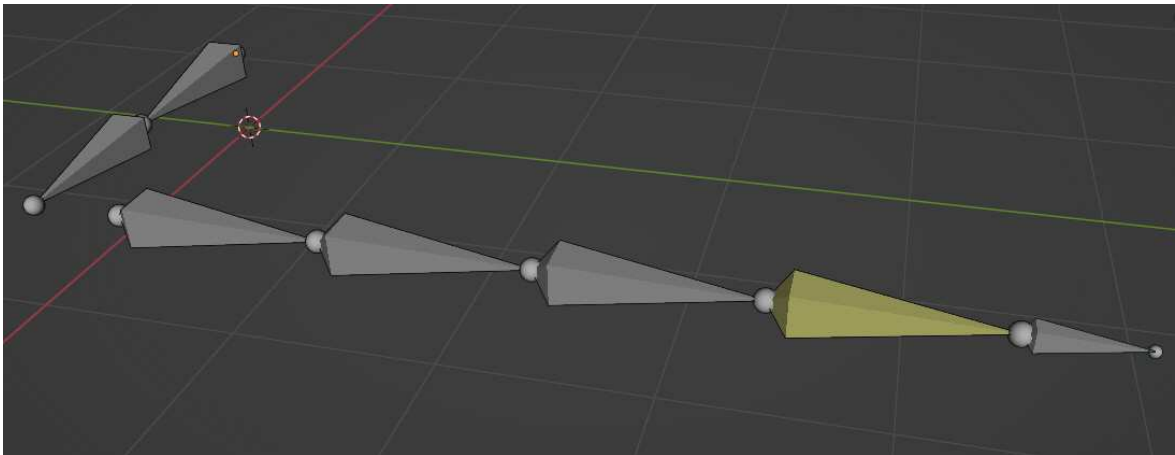
Slika 3.5: Ručno postavljene kostiju u obliku paukove noge

Ovaj postupak micanja kostiju je iznimno spor, ne učinkovit i težak za raditi. Treba nam način da pomaknemo samo zadnju kost, a sve ostale kosti se onda same rotiraju. Za micanje kostiju koristit ćemo postupak koji se zove inverzna kinematika (engl. *inverse kinematics*). To je proces u kojem se računaju pozicije zglobova u odnosu na poziciju početka i kraja lanca kostiju. Sljedeće potpoglavlje opisuje način izrade takvog sustava.

3.2. Inverzna kinematika

Na postojeće kosti noge našeg skeleta dodat ćemo još jednu kost. Ta kost bit će upravljačka kost. Njena pozicija određuje poziciju ostalih kostiju u lancu. Način rada prebacit ćemo u *Edit mode* i mišem ćemo odabrati zadnju kost noge te alatom *Extrude* napraviti još jednu manju kost. Toj novoj manjoj kosti moramo maknuti roditelja. Pritiskom tipki *ALT + p* i odabirom *Clear Parent* dobijemo taj rezultat.

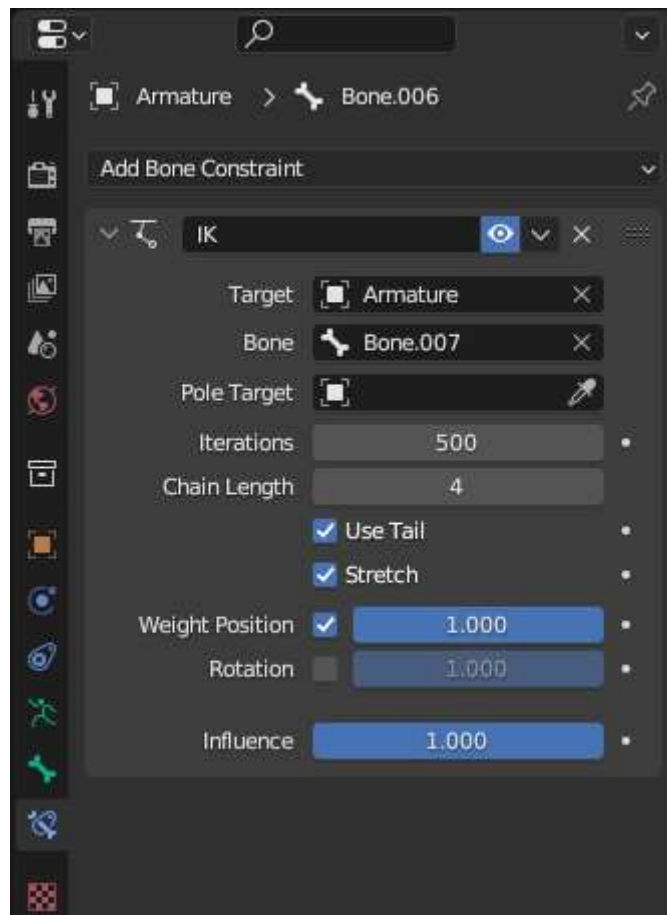
Sljedeće što moramo napraviti je prebaciti način rada u *Pose mode*. Prvo mišem odabiremo manju kost koja će biti upravljačka. Zatim proširimo odabir na kost kojom će prva kost upravljati. Pritiskom tipki *SHIFT + i* te odabirom *Add IK To Active Bone* dodali smo *IK Controller*. Rezultat dodavanja *IK Controller*-a vidi se na slici 3.6.



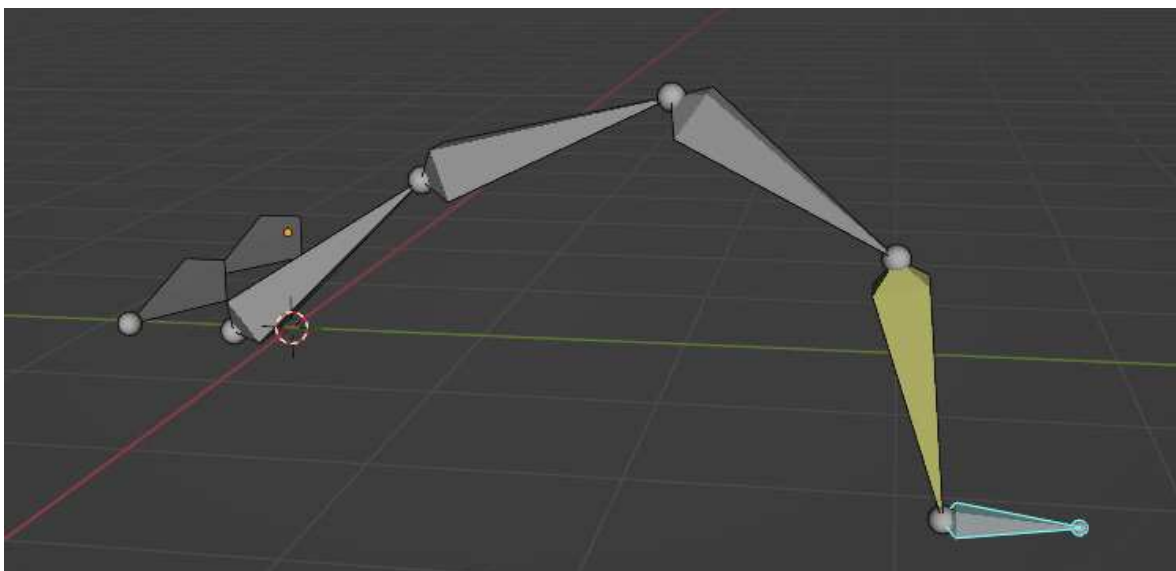
Slika 3.6: Prikaz skeleta koji koristi inverznu kinematiku

Da bi pomicanje nogu radilo ispravno potrebno je odabrati kost žute boje. Zatim je pod *Properties* -> *Bone Constraint Properties* potrebno vrijednost polja *Chain Length* postaviti na 4 (Slika 3.7). Ta vrijednost određuje duljinu lanca kojim *IK Controller* upravlja. Prijašnja vrijednost bila je 0, odnosno *IK Controller* je upravljao svim kostima u lancu uključujući i kostima tijela.

Ako sada prebacimo način rada u *Pose mode* i promijenimo poziciju *IK Controller*a cijeli lanac kostiju koji predstavljaju nogu će se pomaknuti (Slika 3.8).



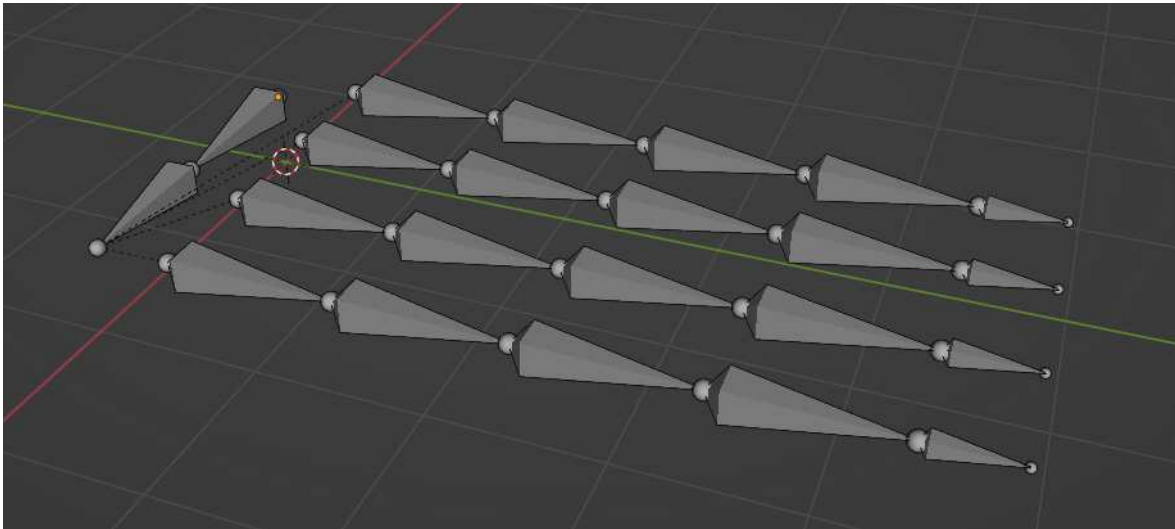
Slika 3.7: Postavke za *Bone Constraint Properties*



Slika 3.8: Pomicanje noge pomoću *IK Controller*-a

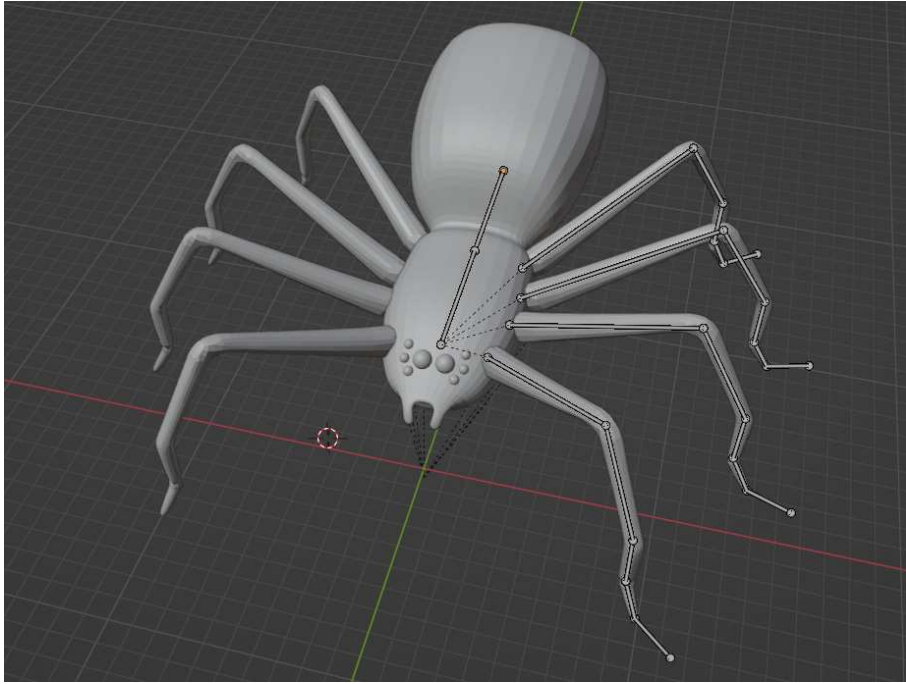
3.3. Spajanje skeleta i modela

Prije nego što odlučimo spojiti model sa skeletom, potrebno ga je prvo završiti. Treba prijeći u *Edit mode* način rada. Odabrat ćemo svaku kost noge i pritiskom tipki *SHIFT + d* kopirati trenutni odabir. Kostići ćemo kopirati 3 puta tako da imamo noge samo s jedne strane. To radimo zato što ćemo svaku kost trebati namjestiti u zglobove modela pauka. Naš model je simetričan stoga ćemo namjestiti kosti samo s jedne strane a onda ih zrcaliti na drugu. Slika 3.9 prikazuje izgled kostiju kada ih kopiramo 3 puta.



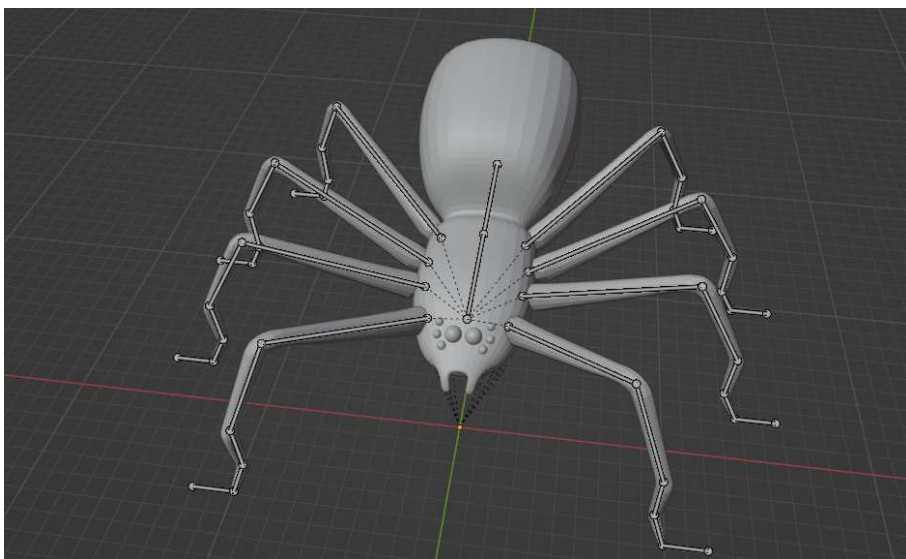
Slika 3.9: Prikaz kostiju noga samo s jedne strane

Da bismo znali gdje postaviti kosti treba pritisnuti ikonu oka pored modela pauka kako bi on ponovno postao vidljiv. Sada u sceni imamo skelet i 3D model. Potrebno je translirati zglobove kostiju tako da se oni poklope sa zglobovima modela (Slika 3.10). Što preciznije postavimo kosti na model to će animacija kasnija biti bolja. Treba napomenuti da se ove promjene moraju raditi u *Edit mode* načinu rada, a ne u *Pose mode*. Korisno je promijeniti način prikazivanja kostiju tako da ih lakše možemo pozicionirati.



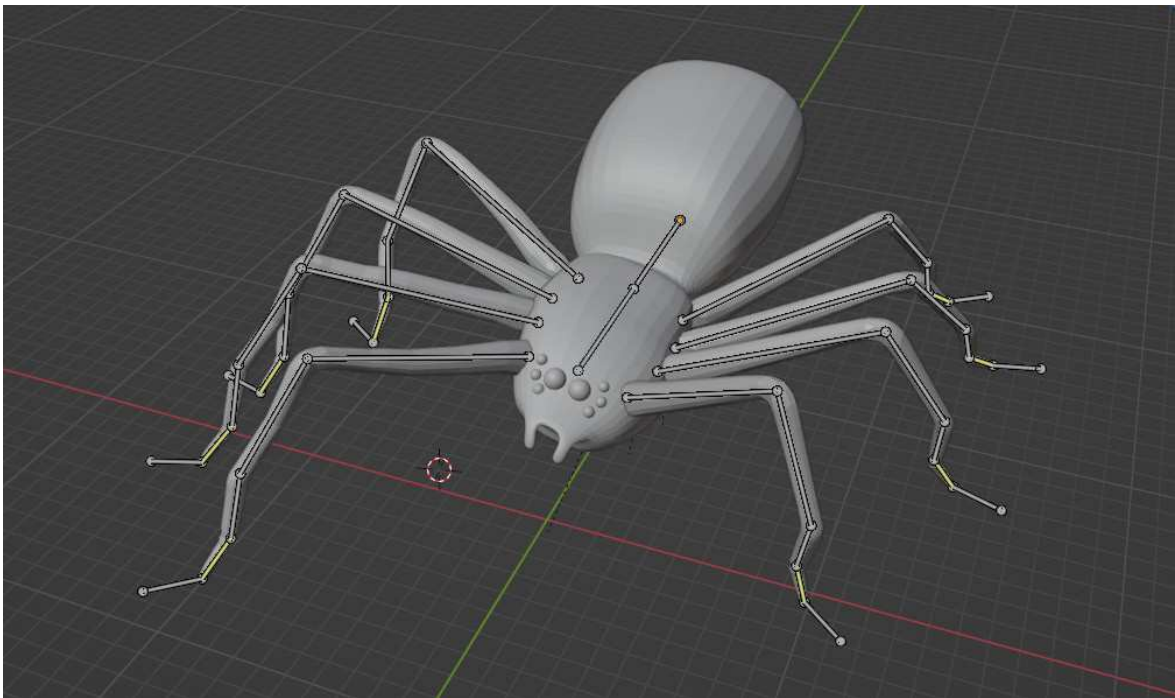
Slika 3.10: Prikaz polovice skeleta postavljenog na model pauka

Sljedeće što želimo napraviti je kopirati kosti s jedne strane modela na drugu. Ako odaberemo sve kosti koje želimo kopirati i onda pritisnemo desni klik, pojavit će se opcija *Symmetrize*. Klikom na tu opciju neće se ništa dogoditi. To je zato što Blender koristi imena kostiju kako bi napravio simetrične kosti. Ime kosti mora završavati sa „L“ ili „R“. Moguće je ručno preimenovati kosti, ali također je moguće koristiti Blenderov alat za automatsko stvaranje imena. Odabiranjem kostiju i pritiskom na desni klik miša otvara se izbornik. Potrebno je izabrati *Names -> Auto-Name Left/Right*. Sada možemo izabrati opciju *Symmetrize*. Slika 3.11 prikazuje konačnu poziciju kostiju na modelu.



Slika 3.11: Prikaz cijelog skeleta na modelu pauka

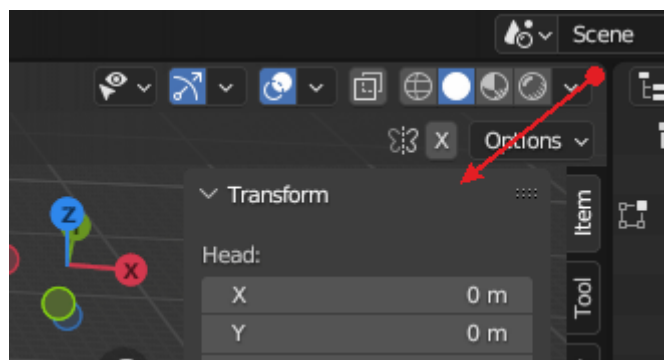
Sada kada su kosti skeleta postavljene na dobro mjesto potrebno ih je još spojiti. Za spajanje skeleta s modelom potrebno je prvo prijeći u *Object mode* način rada. Prvo treba odabrati sve objekte koji cine model pauka. To su tijelo i 8 noga. Zatim je kao zadnji objekt u odabir potrebno dodati i sami skeletni model. Zatim se kombinacijom tipki *CTRL + p* otvara izbornik. Potrebno je odabrati *Armature Deform -> With Automatic Weights*. Blender će sam izračunati koliko se koji vrh objekta mora micati kada se miču kosti. Ako sada način rada prebacimo u *Pose mode* i pomaknemo *IK Controller-e*, možemo micati noge kako got želimo (Slika 3.12).



Slika 3.12: Prikaz pauka koji je rotiran i ima pomaknute noge

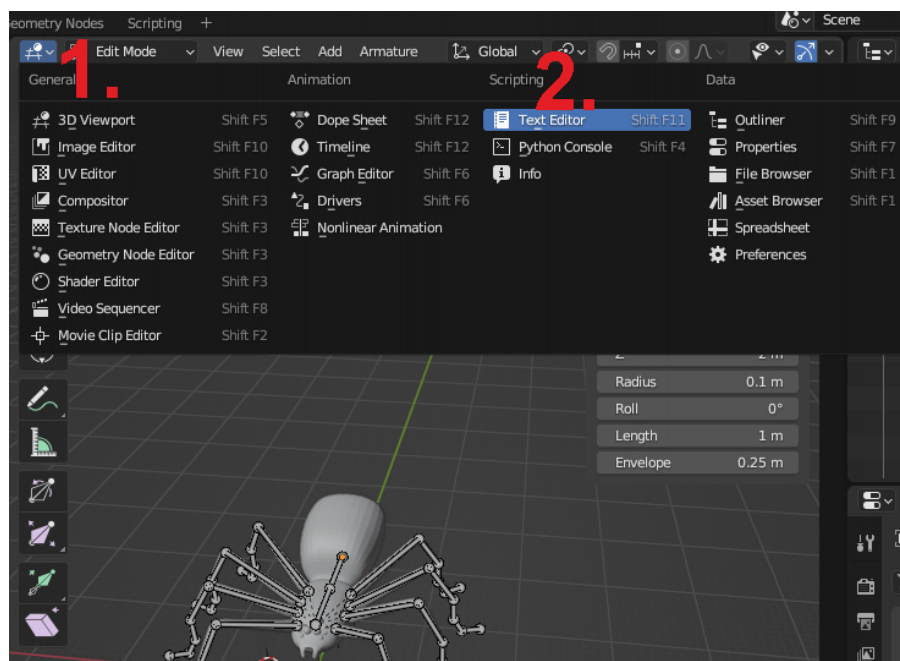
4. Simuliranje hoda

U ovom dijelu rada koristit ćemo Python biblioteku `bpy` kako bismo simulirali hodanje pauka. Napraviti ćemo sistem gdje možemo unijeti naredbe koje će pauk izvršiti. Naredbe koje ćemo programirati su micanje u naprijed i rotiranje. Za instalaciju biblioteke `bpy`, potrebno je u komandnoj liniji napisati: `pip install bpy`. Da bi započeli pisati kôd u Blenderu potrebno je otvoriti novi „prozor“. Iz gornjeg desnog ili lijevog ruba ekrana mišem treba povući prema sredini ekrana (Slika 4.1).



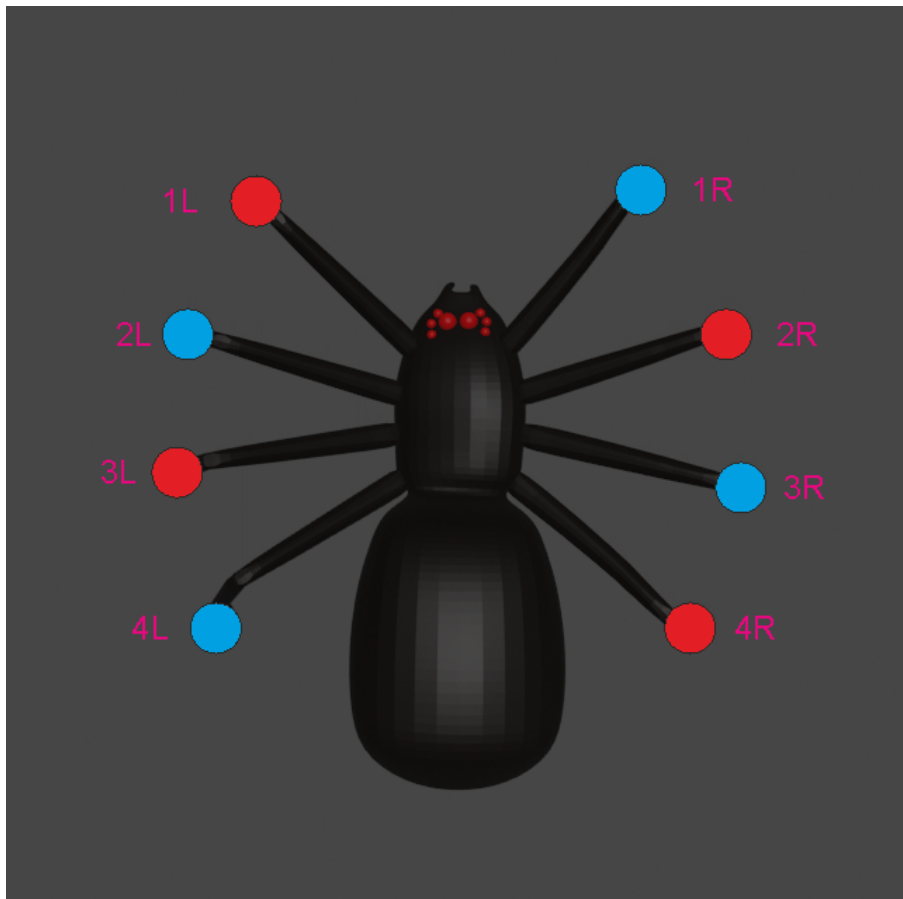
Slika 4.1: Dodavanje novog prozora

U novom prozoru potrebno je *Editor Type* promijeniti na *Text Editor* (Slika 4.2). Sada u ovom prozoru možemo pisati python kôd.



Slika 4.2: Promjena prozora u *Text Editor*

Način na koji pauk hoda na prvu može izgledati komplicirano. Hod se zapravo sastoji od samo dva dijela. U prvom dijelu pomiče se jedan komplet noga, a u drugom drugi. Na slici 4.3 vidi se koje noge se miču zajedno. Također je napisano ime svake noge koje se koristi u programu za kretanje.

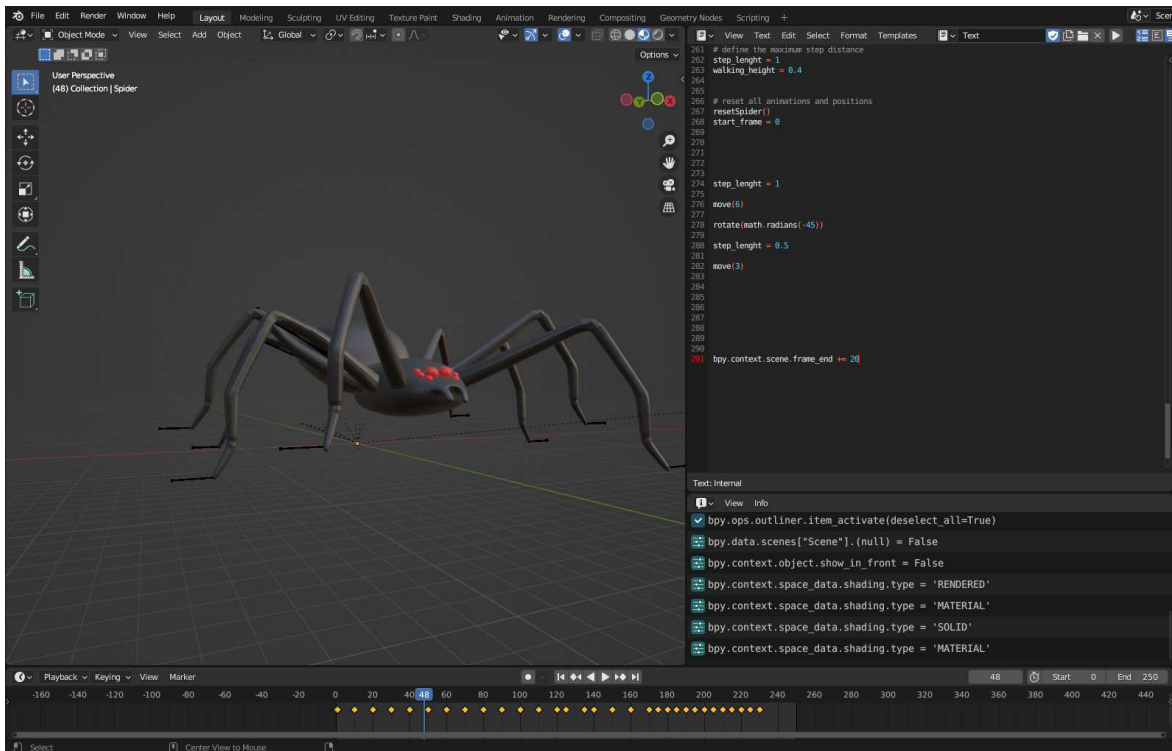


Slika 4.3: Noge obojene istom bojom kreću se zajedno

4.1. Primjer Izvođenja

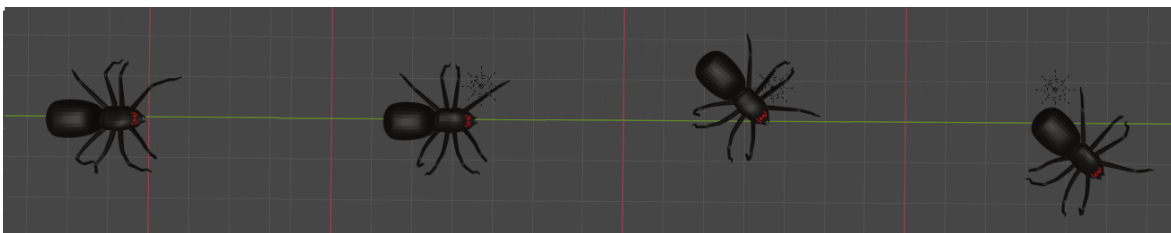
Primjer naredbi koje možemo zadati pauku su:

```
step_lenght = 1
move(6)
rotate(math.radians(-45))
step_lenght = 0.5
move(3)
```



Slika 4.4: Primjer izvođenja simulacije hoda pauka

Slika 4.4 prikazuje izgled okoline u kojoj se odvija simulacija. S lijeve strane slike je prikaz simulacije. S desne strane je kôd koji postavlja *keyframeove* kroz vrijeme ovisno o naredbama koje smo mu zadali. Na dnu ekrana nalazi se *Timeline*. To je vremenska traka koja žutim točkicama prikazuje *keyframeove*. Više dijelova iste animacije mogu se vidjeti na slici 4.5



Slika 4.5: Prikaz pauka kroz vrijeme animacije

4.2. Funkcija micanja

Funkcija `move()` kao parametar prima udaljenost za koju će se pauk pomaknuti u smjeru gledanja. Udaljenost može biti bilo koji pozitivan racionalni broj. U ovom potpoglavlju proći ćemo kroz glavne dijelove funkcije.

Prvo što funkcija računa je broj koraka kojih pauk treba napraviti. Broj koraka ovisi i duljini jednog koraka i ukupnoj udaljenosti koju treba prijeći. Kôd za računanje broja koraka:

```
# calculate number of steps
number_of_steps = distance // step_lenght + 1
if (distance % step_lenght == 0):
    number_of_steps = distance // step_lenght
```

Broj koraka jednak je udaljenosti podijeljeno s duljinom koraka. Broj koraka mora biti cijeli broj. Bitno provjeriti da li je duljina koraka višekratnik udaljenosti.

Nakon računanja koraka potrebno je animirati pokret. Animacija se određuje *keyframeovima*. *Keyframe* je trenutak u vremenu u kojem je zabilježeno stanje nekog objekta kao što je pozicija i rotacija. Potrebno je imati najmanje dva *keyframea* u različitim trenutcima za animaciju. Animacija je onda definirana prijelazom objekta iz jednog stanja u drugo. Prijelaz se može interpolirati na različite načine.

Prije nego što pomaknemo noge pauka potrebno je njihove trenutne pozicije označiti *keyframeom*. Isječak kôda zadužen za taj dio:

```
main_bone.keyframe_insert("location", frame = start_frame)
for i in range(len(controller_bones)):
    controller_bones[i].keyframe_insert("location", frame =
start_frame)
```

Varijabla `main_bone` ima referencu na kost tijela pauka, a lista `controller_bones` sadrži reference na *IK Controllere* za pomicanje nogu. `start_frame` je prvi trenutak u vremenu trenutne naredbe.

Glavna petlja funkcije sada translira kosti i animira nove lokacije. Duljina jednog koraka je udaljenost podijeljena na broj koraka, odnosno varijabla `offset`. Varijabla `end_frame` pamti trenutak u kojem je kraj trenutnog koraka, a `middle_frame` trenutak koji je između prošlog i trenutnog koraka.

```
offset = distance / number_of_steps
end_frame = start_frame + i * int(offset*20) + int(offset*20)
middle_frame = int(end_frame - int((offset * 20)/2))
```

Sada imamo sve potrebno za animiranje koraka noge. Treba proći kroz listu kostiju koje kontroliraju noge i pomicati ih kroz vrijeme. Treba paziti da li prvi korak kojeg pauk uzima počinje s lijevom ili desnom stranom. To ovisi o koracima koje je pauk uzeo prije.

Ako je prvi puta započeo s lijevom nogom prvom i napravio 5 koraka, onda će pri sljedećem pozivu funkcije `move()` morati započeti s desnom nogom. Podatak je boolean spremljen u varijabli `left_or_right`.

U nekom koraku kost može ostati na podu ili se podići i pomaknuti u naprijed. Također treba napomenuti da se jedna noga mora pomaknuti za dvije duljine varijable `offset`. To je posljedica toga da se kost zapravo kreće u nazad gledano u odnosu na tijelo. Ako bi je pomaknuli samo jednom, onda bi zapravo kost došla na nultu poziciju. Ovo je jedan dio kôda koji upravlja jednim kompletom noga:

```
if (i % 2) == left_or_right:
    if bone.name in ["1L", "3L", "2R", "4R"]:
        # leg stays in place
        bone.keyframe_insert("location", frame = end_frame)
    else:
        # lift up the leg and move forwards
        if bone.name in ["1R", "3R"]:
            bone.location.x += offset
            bone.location.z -= walking_height
            bone.keyframe_insert("location", frame =
middle_frame)
            bone.location.x += offset
            bone.location.z += walking_height
            bone.keyframe_insert("location", frame =
end_frame)
        elif bone.name in ["2L", "4L"]:
            bone.location.x -= offset
            bone.location.z -= walking_height
            bone.keyframe_insert("location", frame =
middle_frame)
            bone.location.x -= offset
            bone.location.z += walking_height
            bone.keyframe_insert("location", frame =
end_frame)
```

Također, ovisno s koje strane tijela je noga, ima drugačije orijentiranu x koordinatu.

Na sličan način radi se i rotacija tijela koja simulira njihanje dok pauk hoda. Još jedina stvar koju treba napraviti u funkciji `move()` je izračunati točku u kojoj se nalazi *Origin* točka pauka nakon pomicanja. Ovaj podatak koristan je kod rotacije tijela.


```

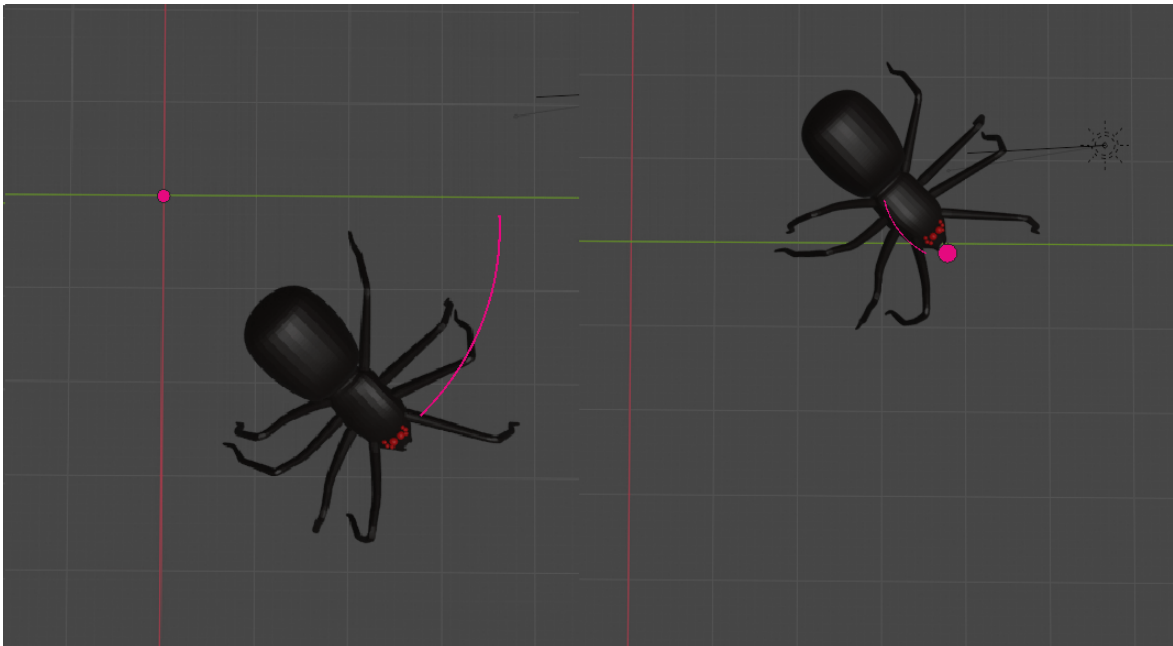
origin_point_x += distance * math.sin(total_angle)
origin_point_y += distance * math.cos(total_angle)

```

`total_angle` je ukupan kut za koji se pauk zarotirao, na početku je vrijednost te varijable 0.

4.3. Funkcija rotiranja

Funkcija `rotate()` kao parametar prima kut u radijanima. Pauk se zatim rotira tako da skoči na mjestu i u zraku se rotira za zadani kut. Za rotiranje tijela pauka ne možemo samo promijeniti njegov trenutni kut. To će pauka okrenuti oko njegove *Origin* točke koja je u ishodištu koordinatnog sustava. Način na koji zaobilazimo ovaj problem je tako da stvorimo nevidljivi objekt. Taj objekt ima *Origin* točku na mjestu gdje je stvoren. Nevidljivi objekt postat će roditelj objektu pauka. Rotacijom nevidljivog objekta ostvarili smo rotaciju i njegove djece, odnosno pauka. Slika 4.6 prikazuje razliku kada rotiramo oko ishodišta i kada rotiramo oko nevidljivog objekta.



Slika 4.6: Rotacija pauka za 45° oko različitih točaka

Na početku funkcije stvorit ćemo nevidljivi objekt. Njegova lokacija je definirana je varijablama `origin_point_x` i `origin_point_y`. Varijable su pomnožene 6 puta jer je pauk pri modeliranju skaliran 6 puta. Kôd za stvaranje nevidljivog objekta:

```

# make an empty arrow used for rotating
arrow = bpy.data.objects.new("Arrow", None)

```

```
bpy.context.collection.objects.link(arrow)
arrow.empty_display_type = 'SINGLE_ARROW'
arrow.location = (origin_point_x * 6, -origin_point_y * 6, 5)
```

Ako sada nevidljivi objekt napravimo roditeljem pauka, pozicija pauka će se resetirati na ishodište. Potrebno je pozvati naredbu kojom će se vrijednosti lokacije kostiju ažurirati. Naredba je:

```
bpy.context.view_layer.update()
```

Treba napomenuti da se pozivom ove funkcije izgube trenutne pozicije kostiju. Treba ih zapamtiti prije poziva i onda im dodijeliti zapamćenu vrijednost nakon.

Svaki put kad rotiramo pauka pravimo novi nevidljivi objekt. Varijabla `highest_parent` je referenca na „najviši“ objekt u lancu. Novi nevidljivi objekt postaje najviši roditelj. Kako bi pauk ostao na svom mjestu također je potrebno i kopirati matricu pauka u nevidljivi objekt. Ovaj isječak kôda je rješenje koje ponudio jedan korisnik na web-forumu [\[8\]](#). Kôd za postavljanje roditelja:

```
# make the empty arrow a parent of spider
highest_parent.parent = arrow
highest_parent.matrix_parent_inverse =
arrow.matrix_world.inverted()
highest_parent = arrow
```

Pomicanje kostiju u zrak radi se na sličan način kao kod funkcije micanja.

Zaključak

Izrada simulacije hoda životinje može biti kompliciran problem ovisno o detaljima koji se žele postići. U ovom radu simulacija je ostvarena više konceptualno.

Objašnjen je način izrade 3D modela u programskom alatu Blender. Opisani su primjeri korištenja alata za modeliranje te postupak izrade i dodjeljivanje jednostavne teksture modelu.

Pokazan je način izrade skeletnog modela. Dodavanje kostiju na već postojeće kosti i dodavanje novih nezavisnih kostiju. Opisan je način za micanje kostiju i izradu figura. Naknadno je objašnjena tehnika inverzne kinematike, odnosno izrada kontrolerskih kostiju kojima je jednostavnije moguće upravljati pojedinim lancima kostiju.

Nakon spajanja pauka i skeleta opisani su dijelovi Python kôda koji omogućuje upravljanje paukom. Opisan je postupak pripreme okoline za pokretanje Python skripti u Blenderu. Objasneni su zanimljivi isječci kôda za kretanje i rotaciju.

Datoteka Blender projekta simulacije hoda pauka javno je dostupna na poveznici: <https://gitlab.com/zzee/zavrsni-rad>

Literatura

- [1] Člankonošci, <https://www.enciklopedija.hr/natuknica.aspx?ID=13447>, pristupljeno 5. lipnja 2023.
- [2] RoBuilder, 23. svibnja 2021., *ABSOLUTE Beginner Basics.. Blender For Noobs*, <https://www.youtube.com/watch?v=Z8sg0nHNTTo>, pristupljeno 30. svibnja 2023.
- [3] PIXXO 3D, 24. svibnja 2021., *Tutorial: Blender MODELLING For Absolute Beginners - Simple Human*, <https://www.youtube.com/watch?v=9xAumJRKV6A>, pristupljeno 30. svibnja 2023.
- [4] Polyfjord, 12. prosinca 2022., *Tutorial: My New Rigging Workflow in Blender*, <https://www.youtube.com/watch?v=BiPoPMnU2VI>, pristupljeno 31. svibnja 2023.
- [5] Kekdot, 10. ožujka 2021., *Easy Character Rigging with Inverse Kinematics | Blender 2.92 | Beginners tutorial*, <https://www.youtube.com/watch?v=hWfUe03Ib5E>, pristupljeno 31. svibnja 2023.
- [6] *Blender as a Python Module*, <https://docs.blender.org/api/current>, pristupljeno 1. lipnja 2023.
- [7] Atom, 12. travnja 2010., *Hot to animate Pivot (origin) location?*, <https://blenderartists.org/t/hot-to-animate-pivot-origin-location/474924>, pristupljeno 2. lipnja 2023.
- [8] Justas, 4. veljače 2017., *How to make object A a parent of object B via Blender's Python API?*, <https://blender.stackexchange.com/questions/9200/how-to-make-object-a-a-parent-of-object-b-via-blenders-python-api>, pristupljeno 2. lipnja 2023.

Sažetak

Simulacija hoda pauka

U ovom radu opisan je postupak izrade 3D modela pauka, izrade skeletnog modela i njihovo spajanje u programskom alatu Blender. Nadalje je opisano simuliranje hoda koristeći Python biblioteku bpy. Detaljno je opisan proces izrade modela uz objašnjenja alata koji su pritom korišteni. Za upravljanje skeletom korišten je sistem upravljačkih kostiju (engl. *inverse kinematics*). Simulacija je ostvarena koristeći programski jezik Python kojim se kontrolirala lokacija upravljačkih kostiju. Moguće je zadati parametre simulacije kao duljinu i visinu koraka te duljinu puta i kut.

Ključne riječi: Blender, Python bpy, modeliranje, skeletni model, simulacija, pauk

Summary

Spider walk simulation

This paper describes the process of creating a 3D model of a spider, creating a skeleton model and combining them in the Blender software tool. Furthermore, walking simulation is described using the Python library bpy. The process of creating the model is described in detail with explanations of the tools used. Inverse kinematics was used to control the skeleton. The simulation was made using the Python programming language, which controlled the location of inverse kinematics controllers. It is possible to set the simulation parameters such as the length and height of the step, as well as the path length and angle.

Keywords: Blender, Python bpy, modeling, skeletal model, simulation, spider