

# Animacijski čvorovi u Blenderu

---

Varšava, Marko

Undergraduate thesis / Završni rad

2019

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:386012>

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

*Download date / Datum preuzimanja:* **2024-09-21**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studij**

**ANIMACIJSKI ČVOROV I U BLENDERU**

**Završni rad**

**Marko Varšava**

**Osijek, 2019.**

# SADRŽAJ

<b>1. UVOD</b> .....	1
<b>1.1. Zadatak završnog rada</b> .....	1
<b>2. PROGRAMSKI PAKET BLENDER</b> .....	2
<b>2.1. Glavne značajke</b> .....	2
<b>2.2. Sučelje</b> .....	2
<b>3. ANIMATION NODES</b> .....	3
<b>3.1. Sučelje</b> .....	4
<b>3.2. Izvršenje</b> .....	8
<b>3.3. Osnovni čvorovi</b> .....	9
<b>4. ANIMACIJA</b> .....	20
<b>4.1. Slaganje Lego kockica</b> .....	20
<b>4.2. Podizanje stambenog naselja</b> .....	28
<b>4.3. Prateći kvadrati</b> .....	33
<b>4.4. Render</b> .....	40
<b>5. ZAKLJUČAK</b> .....	42
<b>LITERATURA</b> .....	43
<b>SAŽETAK</b> .....	44
<b>ABSTRACT</b> .....	45
<b>ŽIVOTOPIS</b> .....	46
<b>PRILOZI</b> .....	47

# 1. UVOD

Tema ovog završnog rada je objasniti za što i kako se koristi program Blender. Objasniti čemu služi i kako se koristi dodatak za Blender Animation Nodes. Objasniti korištenje web alata za slaganje 3D modela od Lego kockica Mecabricks. Na početku rada u drugom poglavlju objašnjen je programski paket Blender. U trećem poglavlju opisan je dodatak Animation Nodes i njegova primjena, kako ga dodati u Blender te kako ga koristi. Objasnjene su osnovni čvorovi koji se koriste u praktičnom dijelu završnog rada koji se nalazi u četvrtom poglavlju. U petom poglavlju nalazi se zaključak.

## 1.1. Zadatak završnog rada

Objasniti izradu animacije u programu Blender uz pomoć dodatka Animation Nodes. Animirati objekte, postaviti scenu, rasvjetu i kameru. Objasniti postavke *rendera* i dobiti vizualno zanimljivu 3D animaciju.

## 2. PROGRAMSKI PAKET BLENDER

### 2.1. Glavne značajke

Blender je besplatan program otvorenog koda, nudi širok raspon alata uključujući 3D modeliranje, *renderiranje*, animaciju, montažu videozapisa, vizualne efekte, teksturiranje, te mnogo vrsta simulacije. Može se koristiti za pravljenje 3D vizualizacija poput fotografija, videa i interaktivnih video igara. Blender je više-platformska aplikacija, podržana na Linux, macOS i MS-Windows sustavima. Sučelje Blendera koristi *OpenGL* grafičko sučelje kako bi pružio isto iskustvo na svim podržanim hardverima i platformama. Sučelje je moguće prilagoditi pomoću Python skripti. U ovom završnom radu korištene su inačice programa Blender 2.79 i 2.8 Beta.


### 2.2. Sučelje

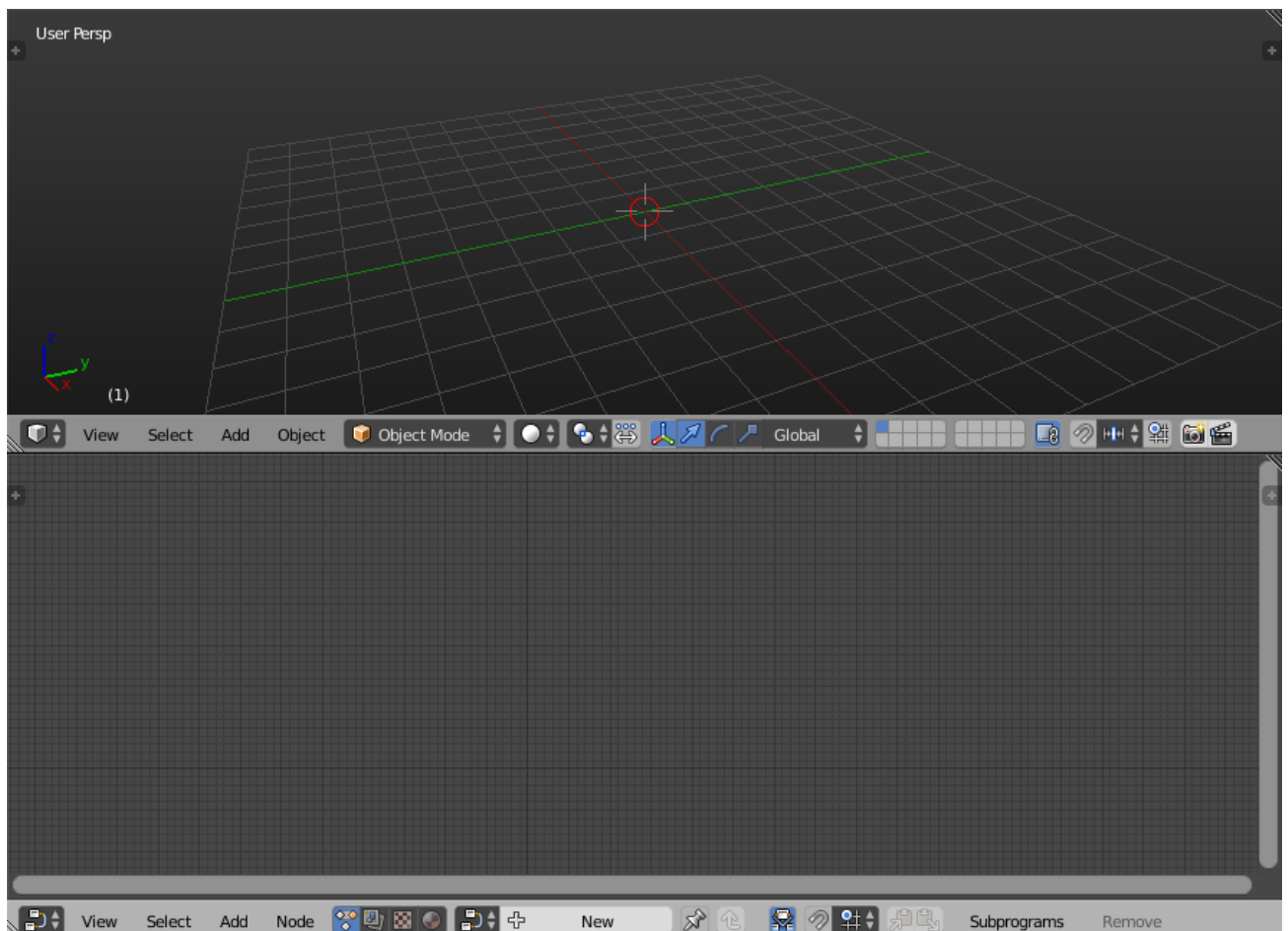
Blenderovo korisničko sučelje ujednačeno je na svim platformama. Glavni elementi sučelja su prozori, ekrani, područja i editori. Sučelje se može prilagoditi tako da odgovara specifičnim potrebama promjenom izgleda zaslona. Zasloni su organizirani u jedno ili više područja, a svako područje sadržava editor. Početni ekran podijeljen je u pet područja koja sadržavaju sljedeće editore: na vrhu se nalazi *Info Editor*, veliki 3D prikaz, vremenska crta na dnu, *Outliner* u gornjem desnom kutu i u donjem desnom kutu editor svojstva. Editori su podijeljeni u regije. Regije mogu imati manje strukturne elemente poput kartica i panela s gumbovima i kontrolama postavljenih unutar njih.

### 3. ANIMATION NODES

Dodatak za Blender Animacijski Čvorovi (engl. Animation Nodes) je sustav za vizualno skriptiranje pomoću čvorova dizajniran za pokretnu grafiku. Razvio ga je Jacques Lucke. Za potrebe ovog rada koristi se verzija *Animation Nodes 2.0*. Mnogi dijelovi Animacijskih čvorova su prepisani koristeći Cython programski jezik. To je jezik koji miješa Python i C programski jezik i prikladan je za razvoj dodataka s visokim performansama u Blenderu.

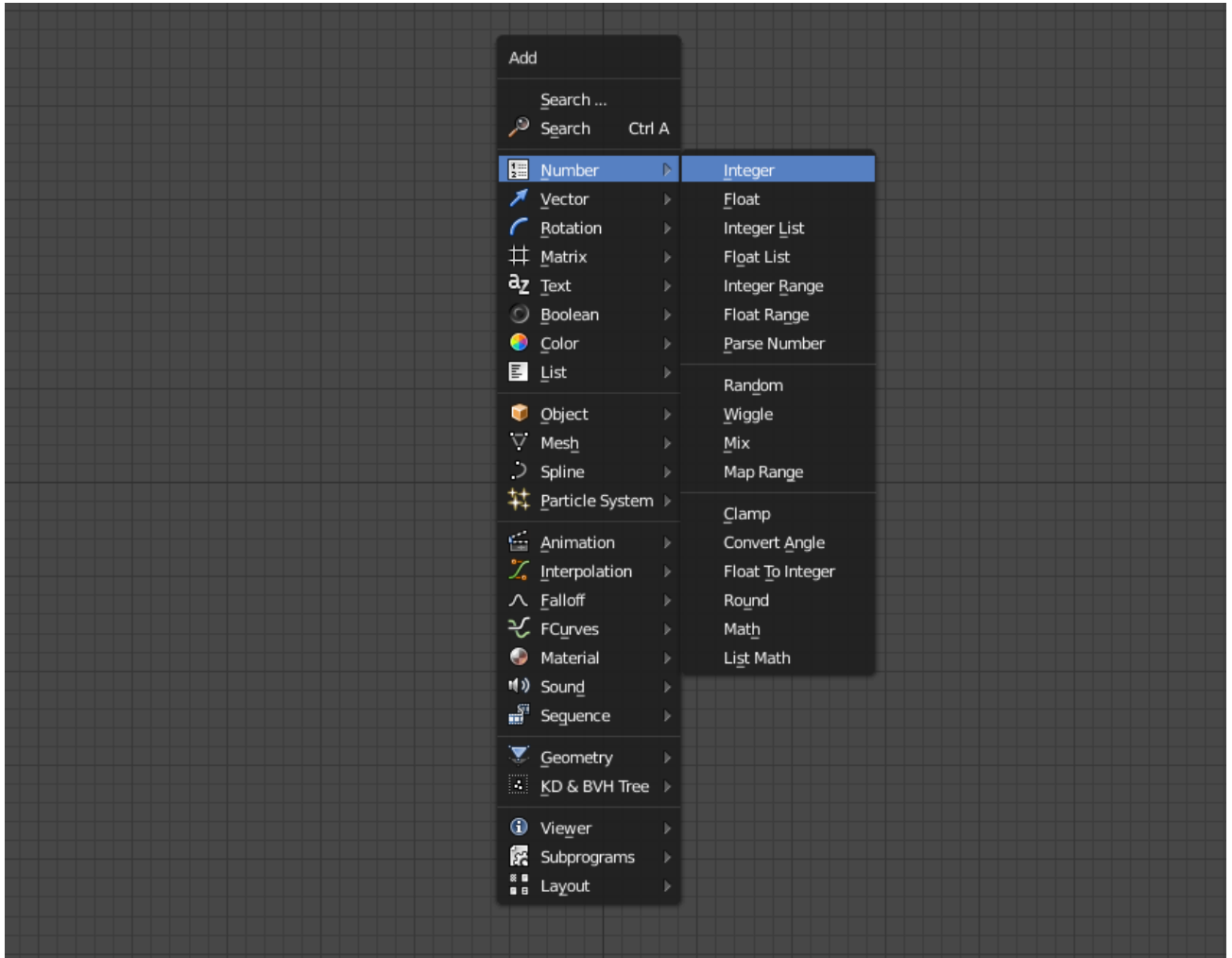
#### 3.1. Sučelje

Većina sučelja se nalazi u *Node Editoru* (Slika 3.1.) dok je ostatak sučelja postoji u drugim regijama poput 3D prikaza. Animacijski čvorovi imaju svoj editor čvorova, odaberemo ga klikom na *Dope Sheet* ikonu  u alatnoj traci. Novo stablo čvorova dodaje se klikom na *New*.



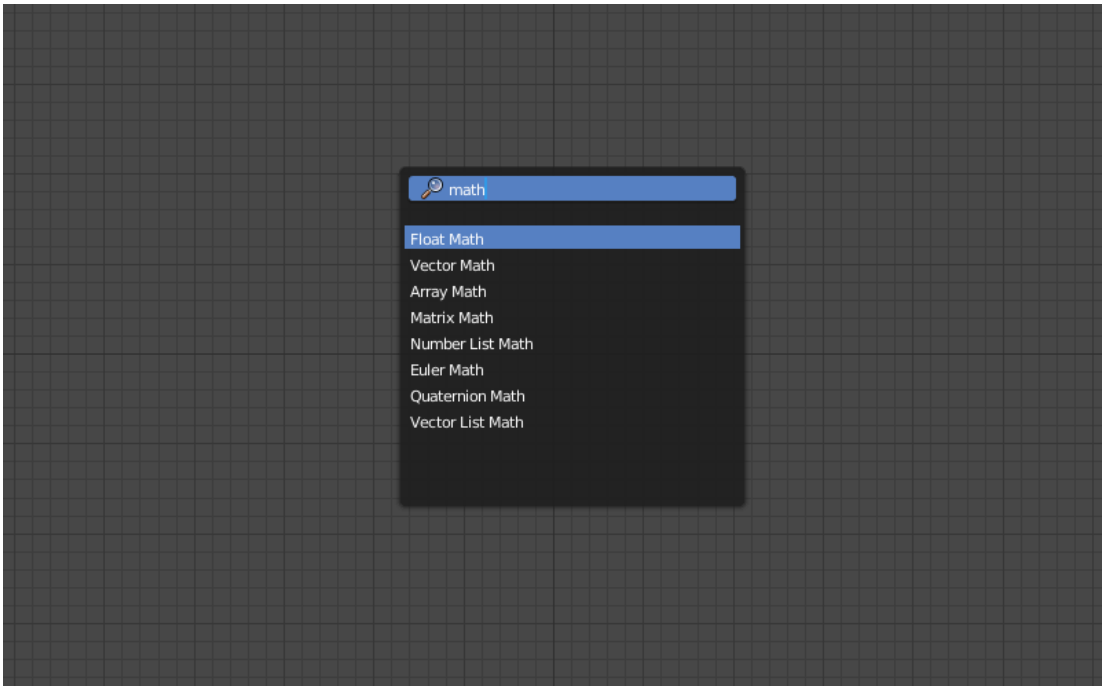
**Slika 3.1.** *Node Editor Animacijskih čvorova*

*Node Menu* (Slika 3.2.) služi za dodavanje novog čvorova, možete otvoriti izbornik čvorova pritiskom na *Node* u alatnoj traci ili pomoću kratice na tipkovnici Shift+A. Izbornik sadržava grupu podizbornika od kojih svaki sadržava listu čvorova podijeljenih po tipu u kategorije. Izbornik sadržava sve čvorove u Animacijskim čvorovima osim nekih pasivnih čvorova. Pasivni čvorovi su čvorovi koji se mogu dodati jedino preko izbornika za pretraživanje (engl. Search menu). Najčešće su dodani automatski gdje god su potrebni, na primjer, *Parse Number* čvor je automatski dodan kada pokušamo spojiti tekst s brojem.



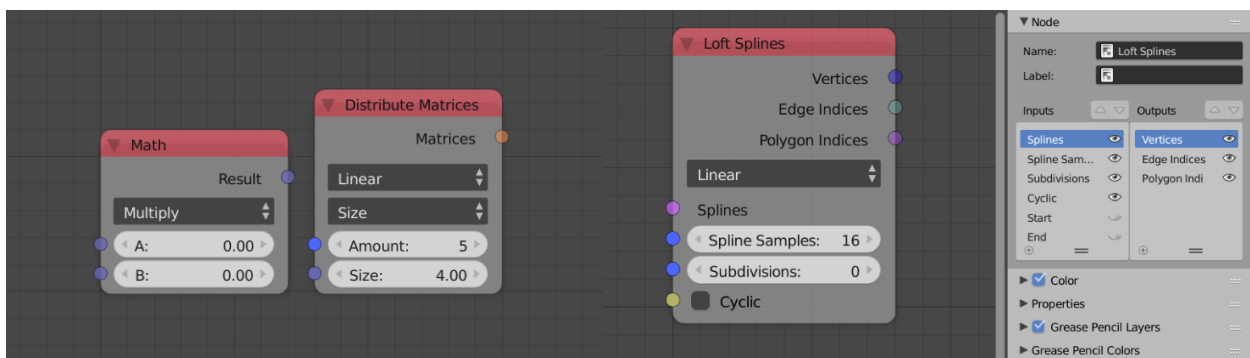
**Slika 3.2.** *Node Menu Animacijskih čvorova*

Izbornik za pretraživanje (engl. Search menu) može biti otvoren pomoću kratice na tipkovnici Ctrl+A (Slika 3.3.), ona omogućava brzi pronalazak i umetanje čvora po njegovom imenu. Neki čvorovi imaju višestruka imena za brži pronalazak. Isto tako neki čvorovi imaju predefimirane opcije, na primjer, traženje *add*, *subtract*, *multiple* će dodati *Math* čvor i postaviti ga na potreban tip.



**Slika 3.3.** Search Menu otvoren pomoću kratice

Postavke čvora koje korisnik često koristi postavljene su unutar čvora (Slika 3.4.). Na primjer, matematičke operacije i obrasci distribucije su opcije čvora. Opcije koje korisnik ne koristi često nalaze se u panelu *Advanced Node Settings* zbog preglednijeg korisničkog sučelja.

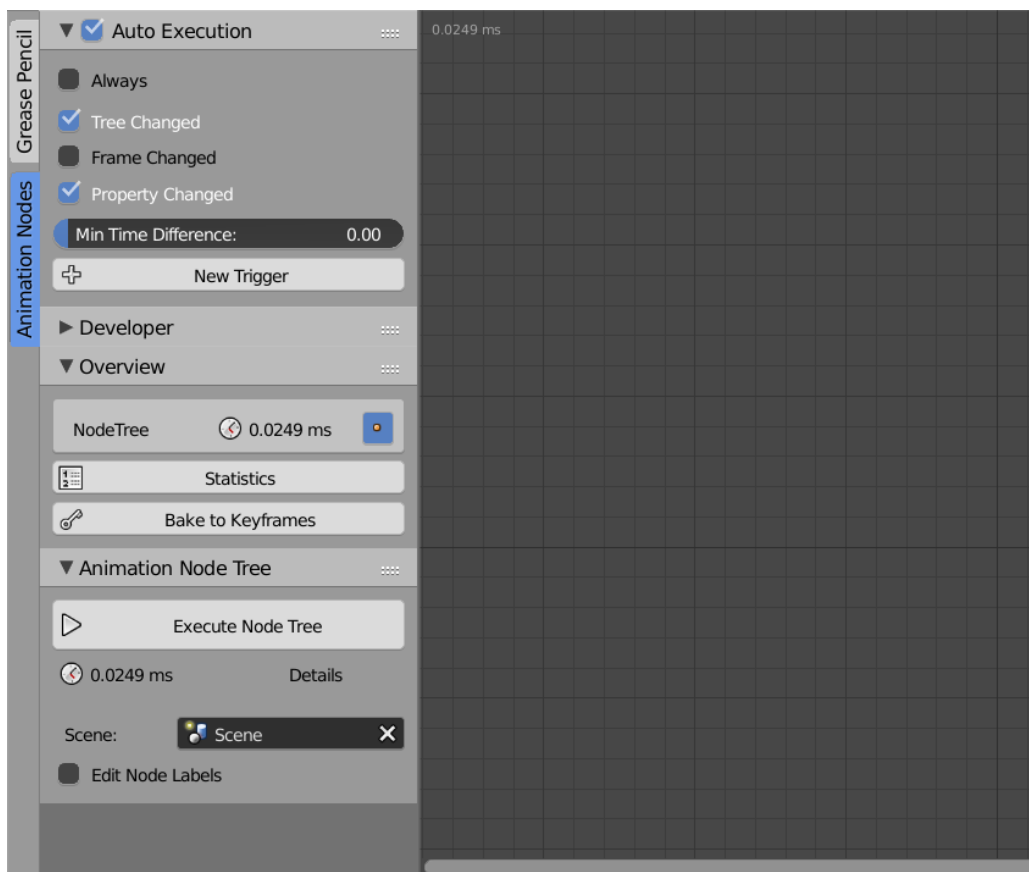


**Slika 3.4.** Prikaz postavki unutar čvorova i Node panela



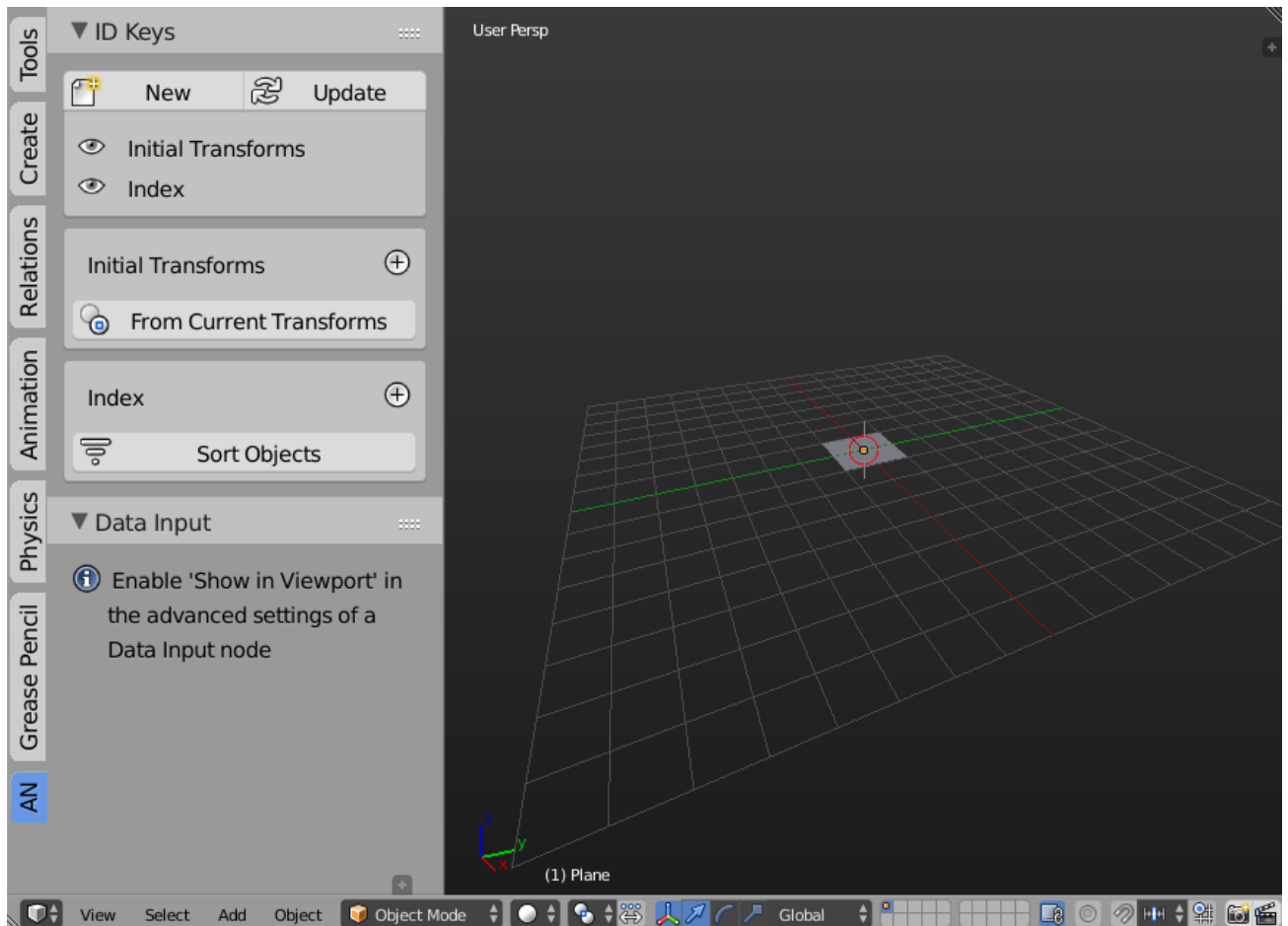
Ulazi i izlazi čvora mogu se sakriti u *Node panelu* pritiskom na oko pokraj imena željenog ulaza ili izlaza (Slika 3.4.). Neki čvorovi poput potprograma omogućavaju promjenu mjesta ulazima/izlazima koristeći strelice. Također tom panelu možete promijeniti imena i oznake čvora. Imena moraju biti unikatna.

Izbornik alata u *editoru* čvorova u Animacijskim čvorovima sadržava postavke i informacije samog stabla. *Auto Execution* panel sadržava opcije vezane uz automatsku izvršenje stabla čvorova, objašnjeno u nastavku. *Developer* panel sadržava opcije za pregledavanje, profiliranje i uklanjanje pogrešaka u stablu čvorova. Unutar *Overview* panela nalaze se informacije o stablu kao što je vrijeme izvršenja. Gumb *Statistics* prikaže tablicu u kojoj se nalaze sve informacije o brojevima i tipovima čvorova u stablu. *Animation Node Tree* panel sadržava gumb za ručno izvršenje te odabir scene kojemu stablo čvorova pripada (Slika 3.5.). Opcija *Edit Node Label* omogućava uređivanje oznake unutar svakog čvora.

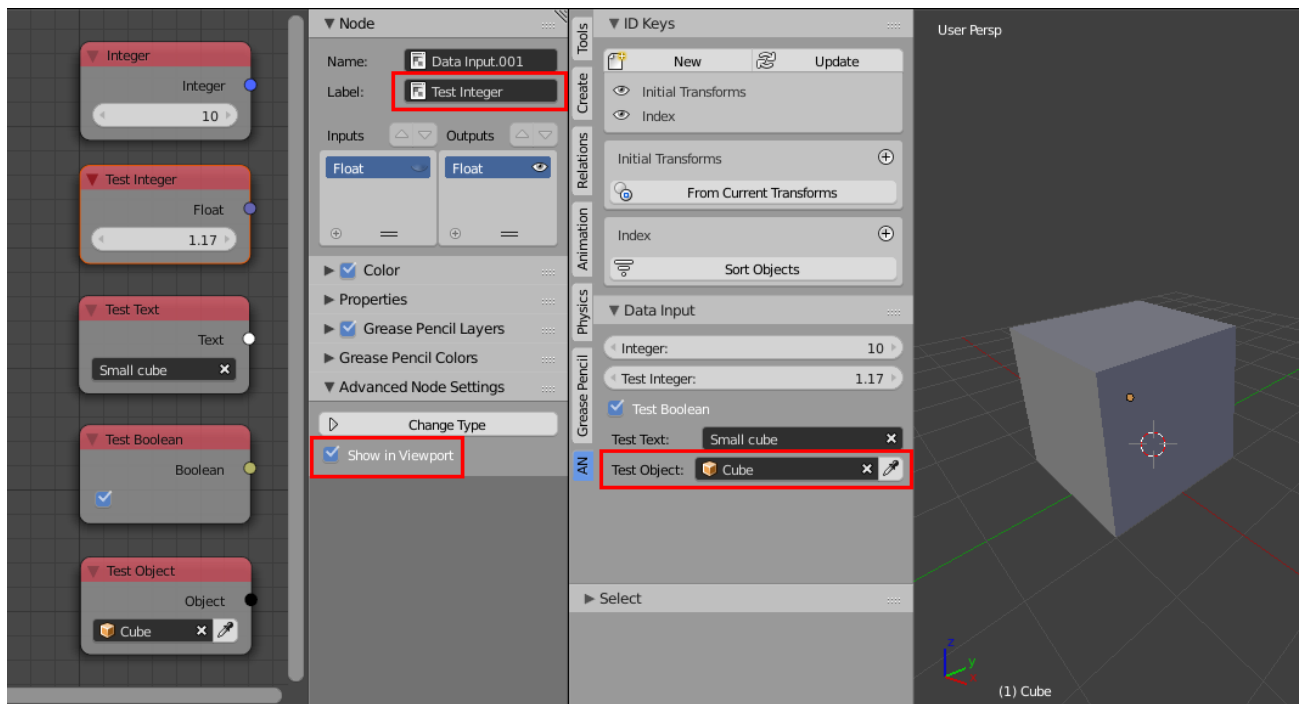


**Slika 3.5.** *Animation Node Tree* panel

Unutar alatne trake *editora* 3D prikaza nalazi se kartica *Animation Nodes* s opcijama vezanim za *Object ID Key* čvor i panel *Data Input* (Slika 3.6.). Ako je opcija *Show in Viewport* unutar panela *Advanced Node Settings* uključena unutar panela *Data Input* pojavit će se svojstva čvorova za unos podataka (Slika 3.7.). Naziv tog svojstva definiran je oznakom čvora ako je postavljena. Ta opcija je korisna kada želimo promijeniti parametre izravno iz 3D prikaza bez potrebe za odlazanjem u *Node editor*.



**Slika 3.6.** *Animation Node* kartica unutar 3D prikaza



Slika 3.7. Prikaz opcije *Show in Viewport*

### 3.2. Izvršenje

Animacijski čvorovi pretvaraju stablo čvorova u skriptu, koja se izvršava na dalje opisane načine. Prema zadanom, Animacijski čvorovi izvršavaju stablo čvorova što je više moguće za što bolje i glađe ažuriranje scene. Međutim, to iscrpljuje procesor i usporava druga područja Blendera. Nepotrebno je izvršavati stablo ako se ništa nije promijenilo u stablu ili sceni. Animacijski čvorovi pružaju automatsko i ručno izvršavanje sustava koje odgovaraju vašim potrebama. Automatsko izvršavanje je kontrolirano s opcijama opisanim u nastavku.

*Always* opcija je omogućena prema zadanom i izvršava stablo čvorova što je više moguće. Tu opciju ne treba koristiti sve dok ona nije potpuno potrebna. Isključivanjem opcije *Always*, dobivamo sljedeće opcije izvršavanja stabla čvorova.

*Tree Changed* opcija koja će izvršiti stablo svaki put kada se dogodi promjena u stablu, to jest kada čvor bude dodan ili uklonjen iz stabla.

*Frame Changed* opcija koja će izvršiti stablo svaki put kada se promjeni trenutni okvir (engl. frame) scene. Ova opcija bi trebala biti uključena ako stablo ovisi o trenutnom okviru.

*Property Changed* opcija koja će izvršiti stablo svaki put kada se promjeni svojstvo jednog od čvorova, to jest kada se promjeni ulaz ili opcija bilo kojeg čvora.

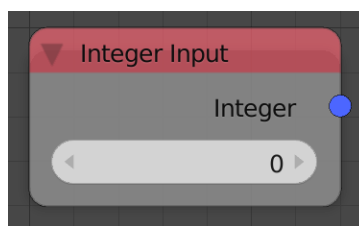
Vjerojatno je da trebate izvršiti stablo čvorova svaki put kad nešto promijenite u svom stablu, pa bi ta opcije *Tree Changed* i *Property Changed* trebale biti uključene.

*Minimum time difference* je vrijednost koja definira vrijeme između svaka dva uzastopna izvršenja. Ako je vrijednost postavljena na jednu sekundu, stablo se neće izvršiti sve dok ne prođe sekunda, čak i ako je jedan od uvjeta iznad zadovoljen, a to su promjena okvira ili svojstva.

### 3.3. Osnovni čvorovi

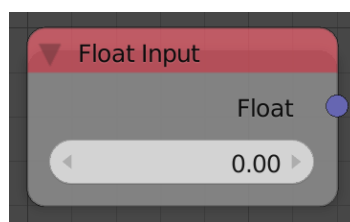
U ovom poglavlju objašnjeni su najčešće korišteni čvorovi i čvorovi korišteni u praktičnom dijelu rada.

Iz prve kategorije *Number*, čvor *Integer* (Slika 3.8.) daje za izlaz cijeli broj.



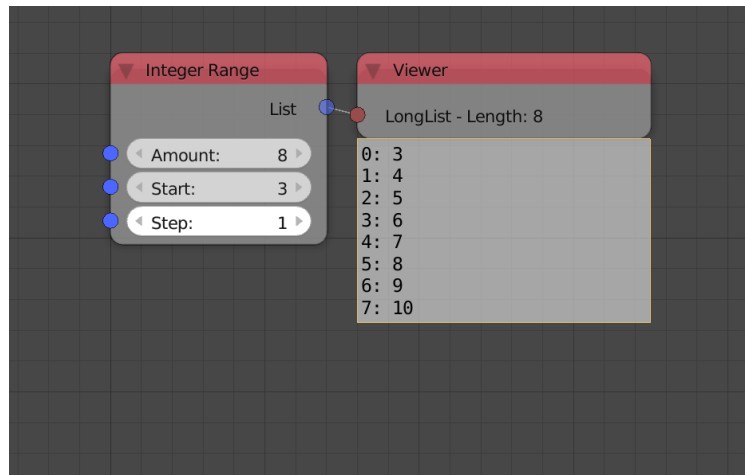
**Slika 3.8.** *Integer Input* čvor

*Float* čvor (Slika 3.9.) daje kao izlaz decimalni broj.



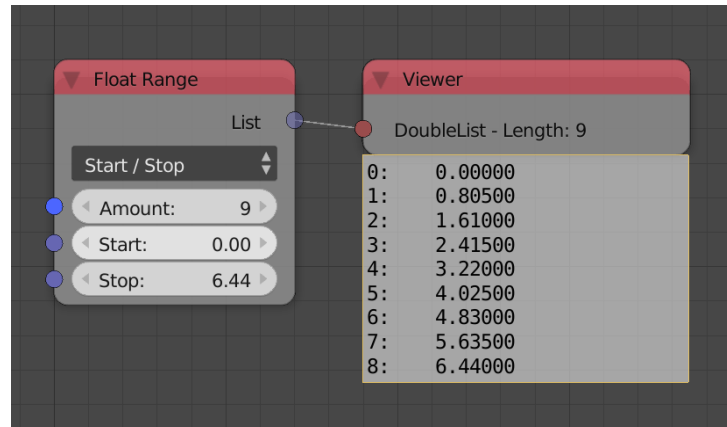
**Slika 3.9.** *Float Input* čvor

*Integer Range* čvor (Slika 3.10.) se koristi za generiranje liste cijelih brojeva koja predstavlja aritmetički niz. Čvor ima tri ulaza, količina (engl. Amount), start i korak. Količina određuje duljinu aritmetičkog niza koja je ujedno duljina izlazne liste cijelih brojeva. Start predstavlja vrijednost prvog broja u nizu. Korak predstavlja razliku između dva uzastopna elementa niza.



**Slika 3.10.** *Integer Range* čvor

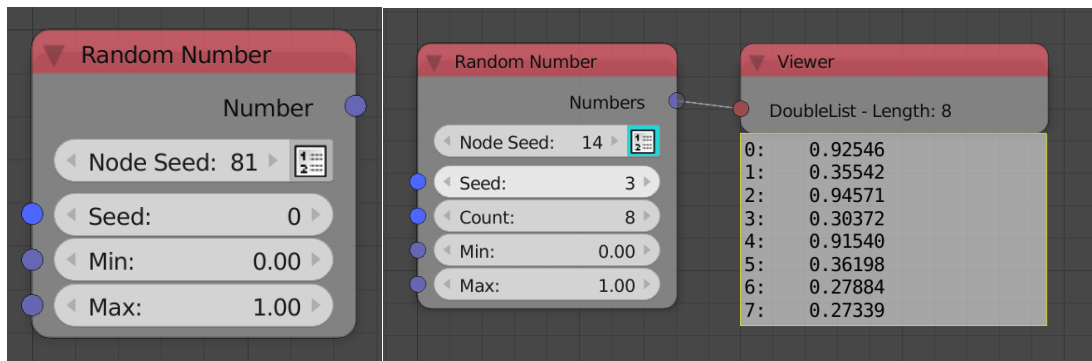
*Float Range* čvor (Slika 3.11.) služi za generiranje liste decimalnih brojeva koja predstavlja aritmetički niz. Čvor ima dvije opcije generiranja brojeva. *Start / Step* opcija omogućava definiranje niza s početnom vrijednosti i korakom koji predstavlja razliku između dva uzastopna elementa. Dok druga *Start / Stop* opcija omogućava definiranje niza s početnom i krajnjom vrijednosti. U ovom slučaju, razlika dva uzastopna elementa je automatski prilagođena, recipročna je vrijednosti *Stop* ulaza.



**Slika 3.11.** *Integer Range* čvor s *Viewer* čvorom

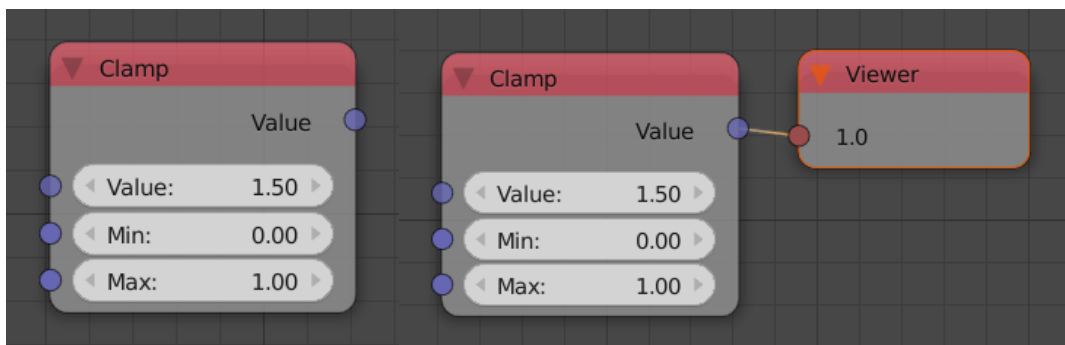
*Random Number* čvor (Slika 3.12.) generira nasumičan decimalni broj između zadane minimalne i maksimalne vrijednosti. Ulaz *Seed* je sjeme nasumičnog generatora, gdje drukčiji broj sjemena generira drukčiji nasumičan broj. Čvor još sadrži opciju sjemena čvora (engl. Node Seed), to dodatno sjeme omogućava dobivanje različitog nasumičnog broja ako ulaz sjemena nije slobodan, to jest, tada je nasumičan broj ovisan o svojstvu kojem nije moguće upravljati. Na primjer, kada koristimo više *Random Number* čvorova u petlji dok koristimo indeks kao sjeme, promjenom opcije sjemena čvora dobit ćemo različite broj za svaki čvor. Čvor omogućava kreiranje liste

nasumičnih brojeva. Opcija se uključuje klikom na ikonu pored sjemena čvora. Uključivanjem opcije pojavljuje se ulaz za veličinu liste (Count).

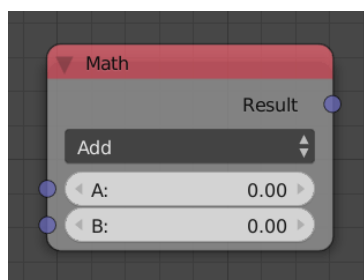


**Slika 3.12.** *Random number* čvor

*Clamp* čvor (Slika 3.13.) ograničava vrijednost na određeni raspon. Bilo koja vrijednost izvan definiranog raspona bit će srezana, to jest, ako je ulazni broj veći od maksimalne vrijednosti bit će postavljen na maksimalnu vrijednost čvora, a ako je manji bit će postavljen na minimalnu vrijednost. Vrijednost koju želimo ograničiti postavljamo na ulaz *Value*.



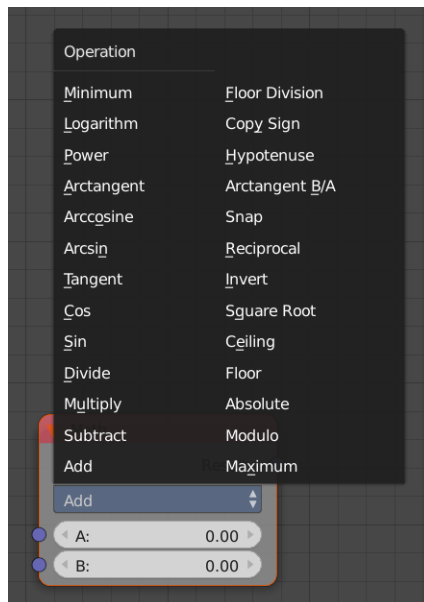
**Slika 3.13.** *Clamp* čvor



**Slika 3.14.** *Math* čvor

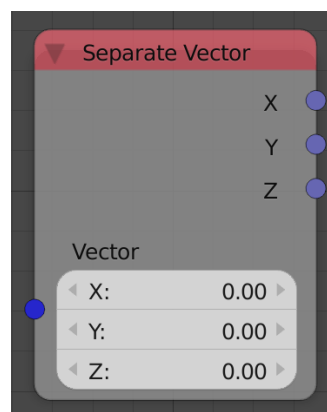
*Math* čvor (Slika 3.14.) obavlja 26 matematičkih operacija. Izlaz čvora je rezultat jedne od operacija nad ulazima. Podržane operacije su zbrajanje (*Add*), oduzimanje (*Subtract*), množenje (*Multiply*), dijeljenje (*Divide*), sinus (*Sine*), kosinus (*Cosine*), tangens (*Tangent*), arkus sinus

(*Arcsine*), arkus kosinus (*Arccosine*), arkus tangens (*Arctangent*), potenciranje (*Power*), logaritmiranje, minimum, maksimum, modul, apsolutno, *Floor*<sup>1</sup>, *Ceiling*<sup>2</sup>, korijen (*Square Root*), promjenu predznaka (*Invert*), inverzija (*Raciprocal*), *Snap*<sup>3</sup>, *Arctangent B/A*<sup>4</sup>, *Hypotenuse*<sup>5</sup>, *Copy Sign*<sup>6</sup>, *Floor Division*<sup>7</sup>.



**Slika 3.15.** Dostupne operacije Math čvora

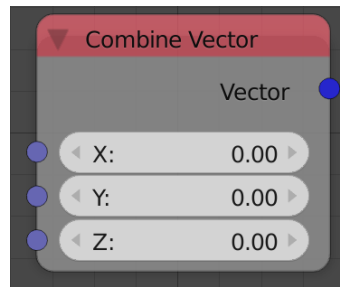
Iz druge kategorije *Vector*, *Separate Vector* čvor (Slika 3.16.) uzima 3D vektor i vraća njegove X, Y i Z komponente.



**Slika 3.16.** *Separate Vector* čvor

<sup>1</sup>Floor – vraća cijeli broj ulaza A, <sup>2</sup>Ceiling – vraća sljedeći cijeli broj ulaza A, <sup>3</sup>Snap – Zaokružuje ulaz A na cijeli višekratnik koraka (engl. Step size) , <sup>4</sup>Arctangent B/A – vraća arkus tangens ulaza B podijeljenim s ulazom A, <sup>5</sup>Hypotenuse – vraća duljinu hipotenuze gdje su ulazi A i B stranice trokuta, <sup>6</sup>Copy Sign – vraća ulaz A s predznakom od ulaza B, <sup>7</sup>Floor Division – vraća cijeli broj ulaza A podijeljenim s ulazom B

*Combine Vector* čvor (Slika 3.17.) sastavlja 3D vektor iz X, Y i Z komponente.



**Slika 3.17.** *Combine Vector* čvor

Iz kategorije *Matrix*, *Distribute Matrices* čvor (Slika 3.18.) kreira listu matrica predstavljenu lokacijom objekta distribuiranog negdje u prostoru. Matrice mogu biti distribuirane u redu, mreži, krugu ili zasnovane na prilagođenoj distribuciji.

Opcija *Linear* distribuira matrice po X osi centrirane u središte svijeta. Postoje dvije vrste linearne distribucije *Size* i *Step*. *Size* distribuira objekte matrica po veličini željene linije, dok druga opcije *Step* distribuira objekte po definiranom razmaku između dva uzastopna objekta. Ulaz količine (engl. Amount) definira broj matrica, ulaz veličine (engl. Size) definira duljinu linije, a ulaz razmaka (engl. Distance) definira razmak između dva uzastopna objekta.

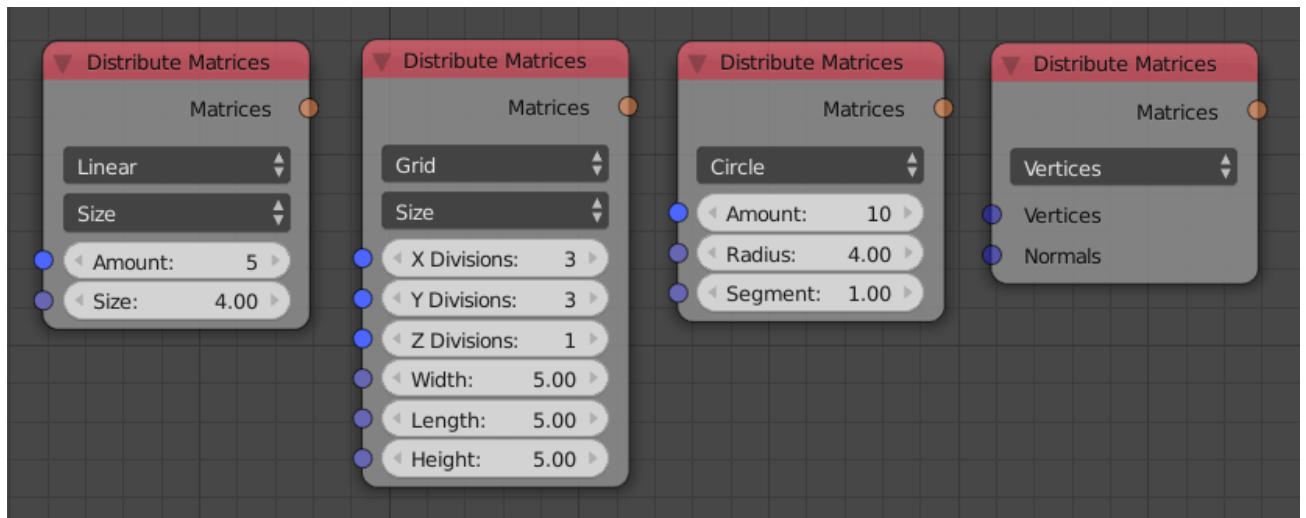
Opcija *Grid* distribuira objekte matrica po 3D rešetkama s pozitivnom Z osi centrirane u središte svijeta. Postoje iste dvije opcije kao i kod linearne distribucije *Size* i *Step*. *Size* distribuira objekte matrica po veličini željene mreže, a opcije *Step* distribuira objekte po definiranom razmaku između dva uzastopna objekta. Ulazi X podjeljaka (engl. X Division), Y podjeljaka (engl. Y Division) i Z podjeljaka (engl. Z Division) definiraju broj objekata po X osi, odnosno Y i Z osi. Ulazi širina (engl. Width), duljina (engl. Length) i visina (engl. Height) definiraju veličinu rešetke. Ulazi X udaljenost (engl. X Distance), Y udaljenost (engl. Y Distance) i Z udaljenost (engl. Z Distance) definiraju razmak između dva uzastopna objekta po X osi, odnosno Y i Z osi.

Opcija *Circle* distribuira objekte matrica po kružnici centrirane u središte svijeta. Ulaz količine (engl. Amount) definira broj matrica, ulaz radius definira radius kružnice, te ulaz odlomak (engl. Segment) definira omjer koji opisuje kompletnost kruga, pri čemu 1 znači potpuni krug, a 0 znači nema kruga.

Opcija *Vertices* distribuira objekte matrica bazirane na listi vektora koje određuju lokacije objektima i drugom listom vektora koje određuju orijentaciju objekata. Ulaz vrhovi (engl. Vertices) je lista vektora koja predstavlja lokacije matrica. Ulaz normale (engl. Normals) je lista



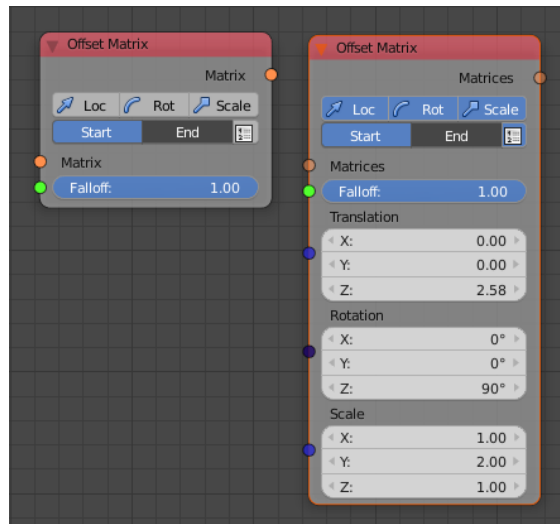
vektora koja predstavlja orijentaciju matrica. Imajte na umu da obje liste moraju imati istu duljinu, a broj izlaznih matrica bit će jednak duljini liste.



**Slika 3.18.** *Distribute Matrices* čvor s različitim opcijama

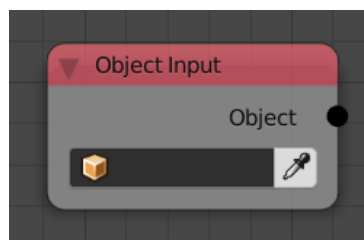
*Offset Matrix* čvor (Slika 3.19.) izmjeni matricu zasnovanu na ulaznom odstupanju (engl. offset) translacije, rotacije i skale (engl. scale) s ulazim *falloff* faktorom. Opcije *Loc/Rot/Scale* omogućavaju pomak lokacije, rotacije i skale. Opcija *Start* će pomaknuti matrice zasnovane na ulaznom odstupanju s ulaznim *falloff* faktorom. Opcija *End* će pomaknuti matrice zasnovane na ulaznom odstupanju s inverznim ulaznim *falloff* faktorom, drugim riječima, sve matrice će biti promijenjene bazirano na parametrima odstupanja, a zatim će matrice biti vraćene na početni položaj s *falloff* faktorom.

Ulazi u čvor su matrice koje želimo pomicati, *falloff* faktor, vektor translacije, vektor rotacije te vektor skale. Izlaz čvora su nove matrice poslije odstupanja. Čvor nudi napredne postavke odstupanja definiranih po globalnim i/ili lokalnim osima translacije, odnosno točke zakretanja rotacije.



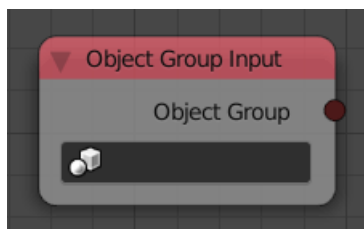
**Slika 3.19.** *Offset Matrix* čvor s različitim opcijama

*Object Input* čvor (Slika 3.20.) vraća ulazni objekt. Klikom na ikonu kapaljke postavlja se za ulazni objekt aktivni objekt u 3D prikazu.



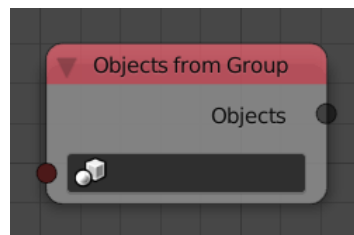
**Slika 3.20.** *Object Input* čvor

*Object Group Input* čvor (Slika 3.21.) vraća ulaznu grupu objekata.



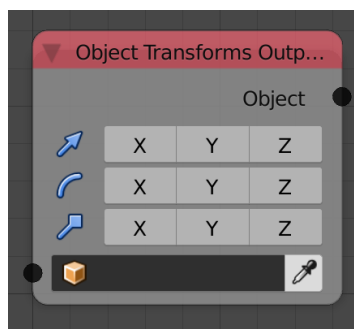
**Slika 3.21.** *Object Group Input* čvor

*Object From Group* čvor (Slika 3.22.) vraća listu objekata koji su uključeni u ulaznu grupu.



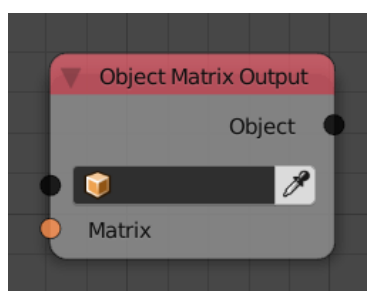
**Slika 3.22.** *Object From Group* čvor

*Transform Outputs* čvor (Slika 3.23.) postavlja lokaciju, rotaciju i skalu ulaznog objekta na izlaznu transformaciju. Korištenje svojstva pali se klikom na potvrdne okvire (engl. checkbox).



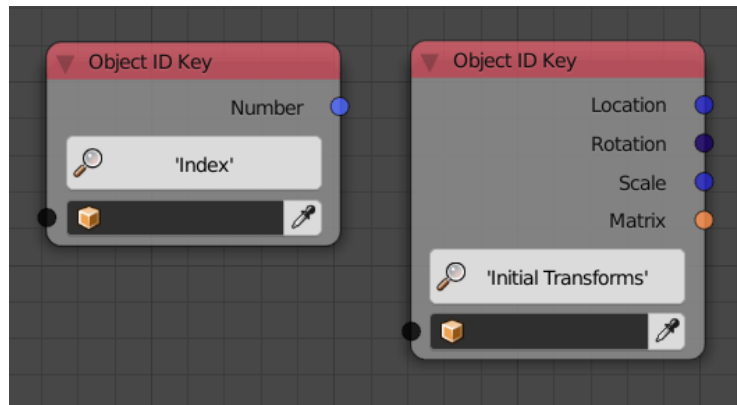
**Slika 3.23.** *Object Transform Output* čvor

*Object Matrix Output* čvor (Slika 3.24.) postavlja transformacije ulaznog objekta na matricu ulazne transformacije.



**Slika 3.24.** *Object Matrix Output* čvor

*Object ID Keys* čvor (Slika 3.25.) vraća vrijednost specifičnog identifikacijskog ključa (engl. ID key). Ključevi su vrijednosti koje su postavljene na objekte i one su konstantne sve dok se ne ažuriraju. Primjer bi bio postavljanje početne transformacije objekta u ID ključ, a zatim mu pristupiti iz različitog razloga korištenjem *Object ID Keys* čvora.



**Slika 3.25.** *Object ID Key s različitim opcijama*

*Object Instancer* čvor (Slika 3.26.) dinamički stvara i duplicira objekte. Izlaz čvora je lista koja sadržava instancirane objekte.

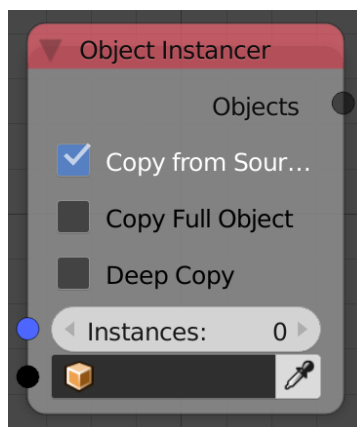
Opcija kopiraj iz izvora (engl. Copy from source) je uključena po zadanom i omogućuje instanciranje objekta sa svojim podacima, međutim, ako je opcija isključena čvor će kreirati prazne blokove podataka koje je moguće popuniti i kontrolirati pomoću drugih čvorova kao poput *Copy Object Data* čvora i *Object Output* čvora.

Opcija tip objekta (engl. Object Type) je jedino vidljiva kada je opcija kopiraj iz izvora isključena. Omogućava definiranje tipa blokova podataka koji koje objekt treba nositi. Pa tako podaci mogu biti prazan objekt (engl. Empty), mreža (engl. Mesh), krivulje (engl. Curve) i drugo.

Opcija kopiraj cijeli objekt (engl. Copy full object) će kopirati modifikatore i ograničenja s objektom.

Opcija duboka kopija (engl. Deep copy) će kopirati podatke mreže tako da svaki blok podataka ima jednog korisnika. Novi objekt bit će neovisan o izvoru.

Opcije kopiraj cijeli objekt i duboka kopija moguće su jedino kad je opcija kopiraj iz izvora isključena.



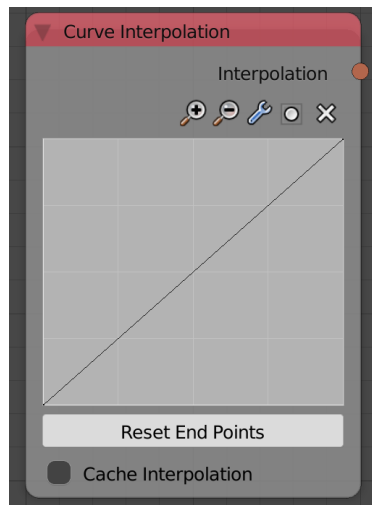
**Slika 3.26.** *Object Instancer* čvor

*Delay Falloff* čvor (Slika 3.27.) stvara *falloff* zasnovan na indeksu koji povezuje decimalni broj koji se povećava kako se vrijeme povećava tamo gdje se objekti s indeksima koji su bliži nuli počinju povećavati prije onih koji su udaljeniji od nule. Korisnik može kontrolirati stopu povećanja u odnosu na vrijeme i indekse. Ulaz vrijeme (engl. time) predstavlja trenutno vrijeme. Ulaz odgoda (engl. delay) predstavlja vrijeme između početka animacije dva uzastopna objekta. Ulaz trajanje (engl. duration) je vrijeme koje potrebno svakom objektu da dostigne svoju maksimalnu vrijednost (1). Interpolacija (engl. Interpolation) je svojstvo koje mijenja izlaz po brzinskoj funkciji animacije. Ulaz *Offsets*, ako je omogućen, određuje koliko se svaki objekt pomaknuo u vremenu.



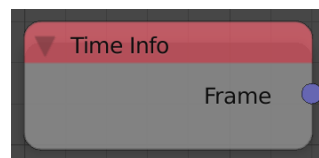
**Slika 3.27.** *Delay Falloff* čvor i predefinirane interpolacije

*Curve Interpolation* čvor (Slika 3.28.) uzima točke i vraća interpolaciju koja je predstavljena krivuljom.



**Slika 3.28.** *Curve Interpolation* čvor

*Time Info* čvor (Slika 3.29.) vraća informaciju o reprodukciji scene. Izlazi su mu trenutni okvir (engl. Current Frame), početni okvir (engl. Start Frame), zadnji okvir (engl. End Frame) i frame rate<sup>1</sup>.



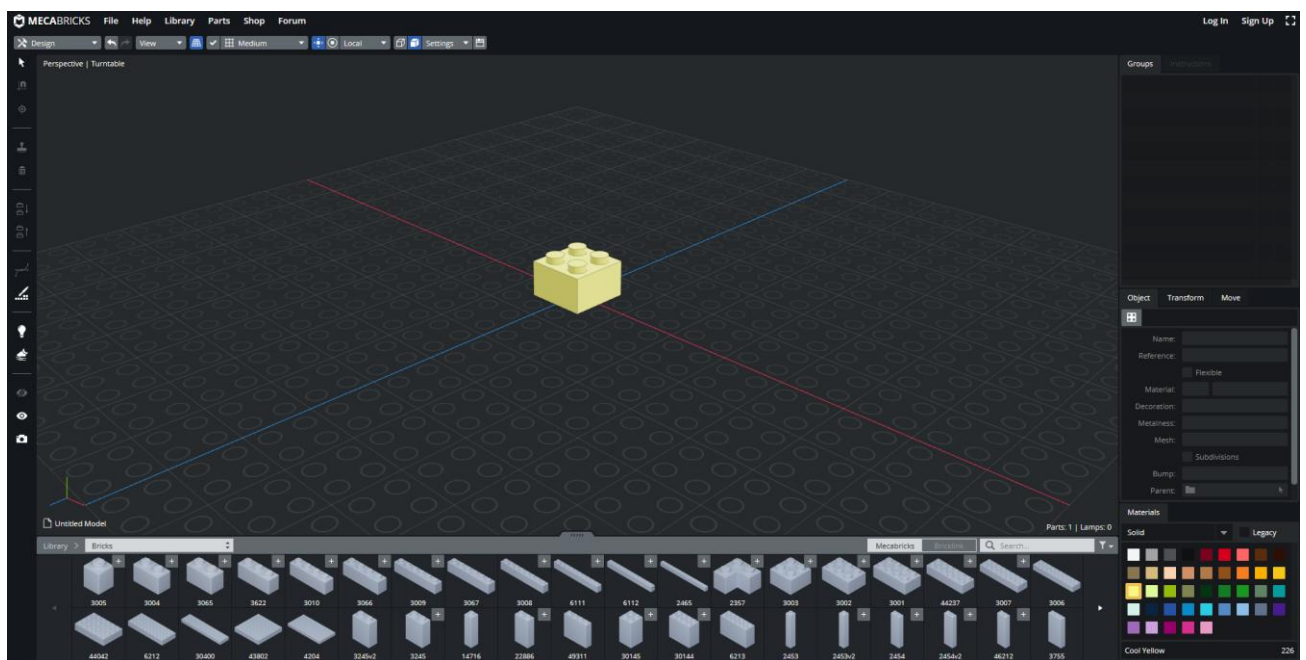
**Slika 3.29.** *Time Info* čvor

<sup>1</sup>frame rate – označava broj okvira u sekundi u videozapisu

## 4. ANIMACIJA

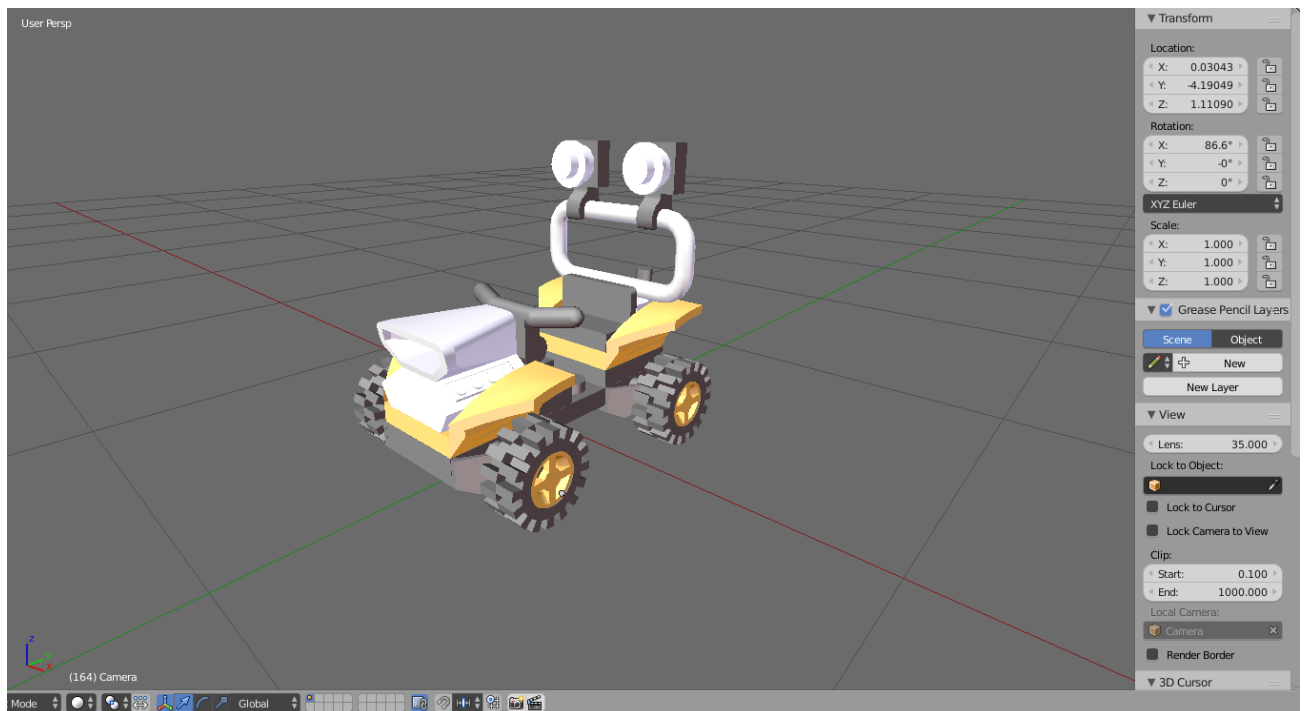
### 4.1. Slaganje Lego Kockica

Model koji će se animirati potrebno je složiti, a za to se koristi besplatan online alat za preglednike Mecabricks (Slika 4.1.). Alat omogućava pravljenje LEGO 3D modela i mogućnost objavljivanja tako da bude dostupno svima. Nije potrebna instalacija nikakvog softvera niti dodatka u preglednik. Jedini zahtjev je imati preglednik i grafičku karticu koja podržava *WebGL*, JavaScript API za renderiranje interaktivne 2D i 3D grafike. Model se izrađuje u radionici (engl. Workshop) gdje imamo dostupno preko 10000 dijelova.



Slika 4.1. Sučelje Mecabricks alata

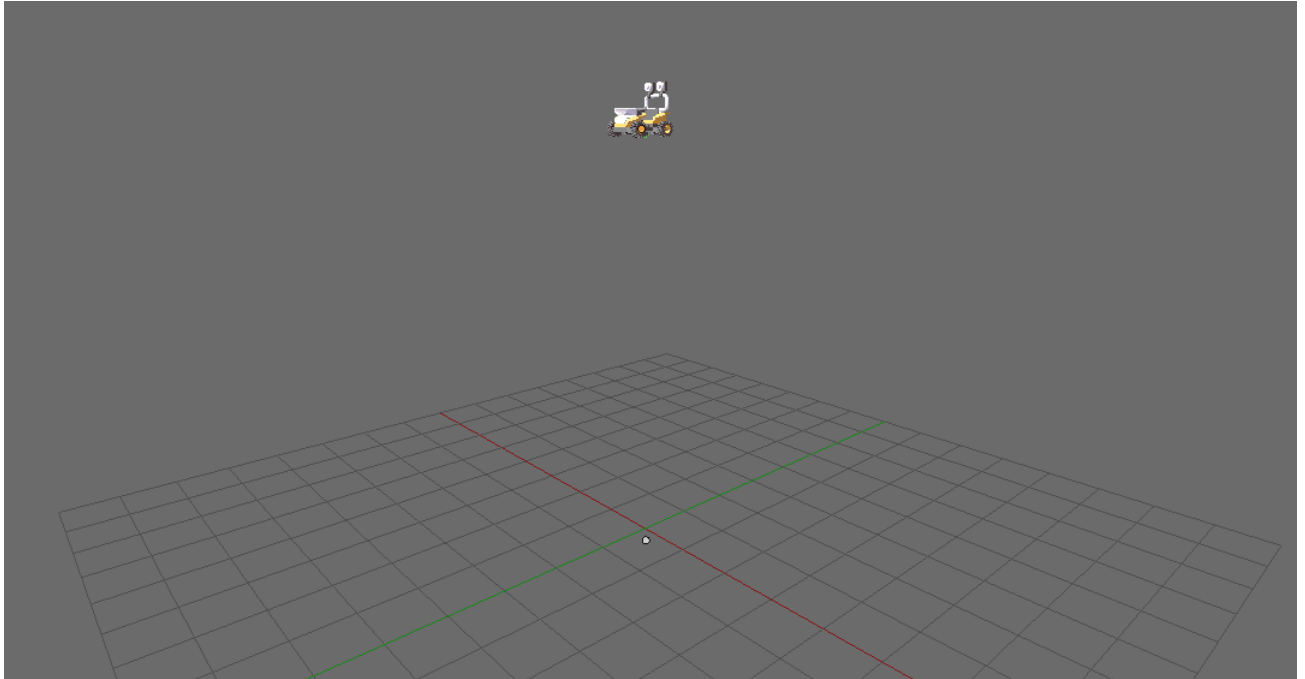
Nakon što se napravio model, spremi se dokument u formatu Collada(.dae) za Blender. U Blenderu se pravi novi prazni dokument. Kraticom Ctrl + A označi se sve na sceni i pritiskom na tipku X na tipkovnici brišu se svi elementi. Uvozi se dokument s Mecabricksa klikom na padajući izbornik s trake glavnog izbornika *File / Import / Collada /* pronađe se model i pritisne se gumb import COLLADA u gornjem desnom kutu (Slika 4.2.). Označe se svi objekti modela i grupiraju.



**Slika 4.2.** Model unutar Blendera

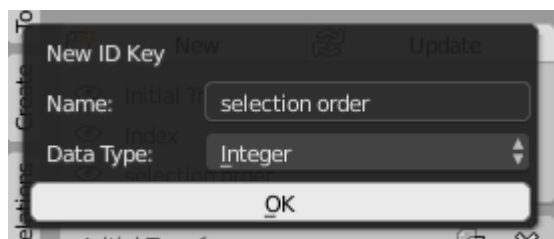
Na ekran se dodaje još jedno područje te promijeni *Editor* u *Node Editor*. U donjem dijelu *Editora* mijenja se tip čvorova u *Animation*. Prvi čvor kojeg se dodaje je *Object from Group*, njegov ulaz je prethodno grupirani model. Izlaz *Object from Group* čvora spaja se na ulaz *Object Matrix Output* i na *Object ID Key* čvor sa svojstvom *Initial Transforms*. *ID Keys* se naročito koriste kada se žele transformirati neki objekti relativno na njihovu izvornu poziciju, u ovom slučaju to će biti krajnja lokacija animiranih objekata. Čvor *Object Matrix Output* transformira ulazni objekt po svojstvima transformacije ulazne matrice. Preostaje spojiti izlaz *Object ID Keys* čvora *Matrices* s ulazom *Offset Matrix* čvora kako bi se postavila početna pozicija za animaciju modela, a izlaz *Offset Matrix* spaja se na ulaz *Object Matrix Output* čvora. U *Offset Matrix* čvoru odabire se lokacija te povećava vrijednost Z osi. To će podići model izvan kadra (Slika 4.3.).





**Slika 4.3.** Podizanje modela po Z osi

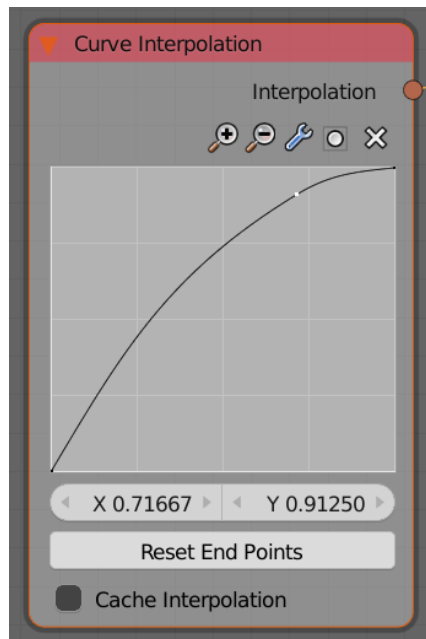
Dodaje se *Delay Falloff* čvor kojem su za ulaz *Time* postavljeni okviri iz *Time Info* čvora, to će „oživiti“ animaciju. Ulaz *Offset* izlaz je novog *Object ID Key* čvora sa svojstvom redosljed selekcije. To svojstvo se mora napraviti selektirajući model redosljedom kojim se želi da se kockice spuštaju, zatim u kartici *AN* unutar 3D prikaza pod panelom *ID Keys* se pravi nova lista ključeva po selekciji pritiskom na *New*. Ključu se dodjeljuje ime i za tip podatka postavlja cijeli broj (engl. Integer) (Slika 4.4.).



**Slika 4.4.** Dodavanje novog ID ključa

Preostaje postaviti parametre *Delay Falloff* čvora kako bi animacija imala odgovarajuće trajanje i dodati interpolacijski čvor koji će usporiti dijelove kada se približe početnoj lokaciji što će učiniti animaciju zanimljivijom (Slika 4.5.). Interpolacijski čvor je čvor koji uzima vrijednost između 0 i 1 i vraća vrijednost isto između 0 i 1. Na primjer, funkcija može biti kvadratna funkcija ( $f(x)=x^2$ ), kada je ulaz 0.5 izlaz će biti 0.25. *Curve Interpolation* čvor predstavlja interpolaciju pomoću

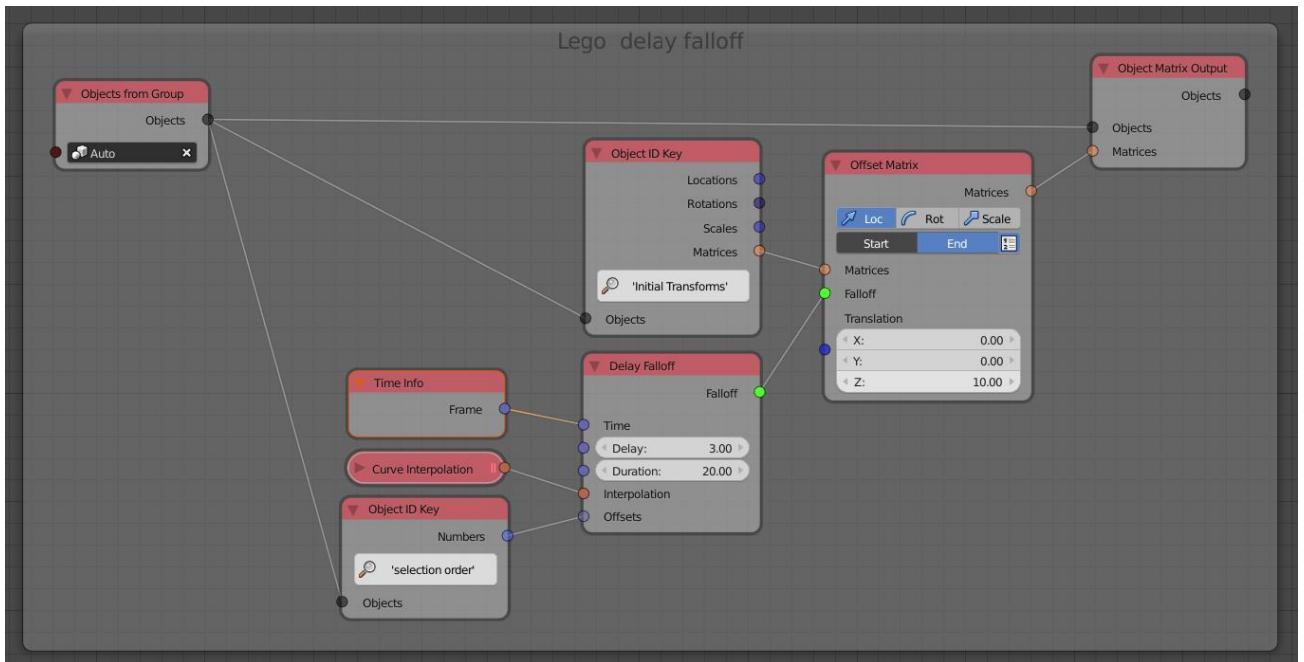
ulaznih točaka krivulje. Pritiskom na *play* ikonu u području vremenske crte (engl. Timeline) dijelovi modela padaju prema dole (Slika 4.6.).



**Slika 4.5.** Izgled krivulje interpolacije

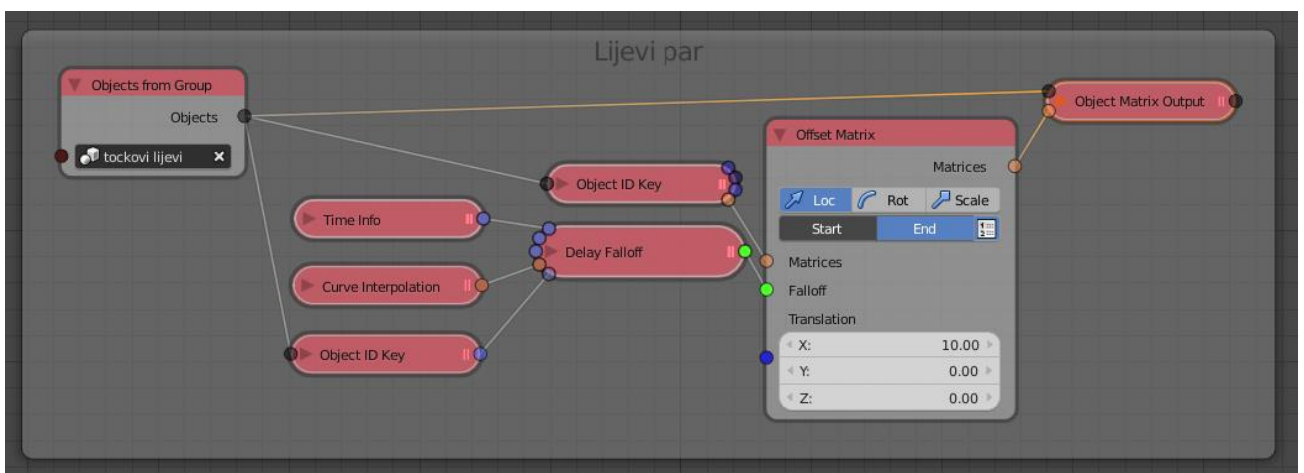


**Slika 4.6.** Spuštanje kockica na početnu poziciju

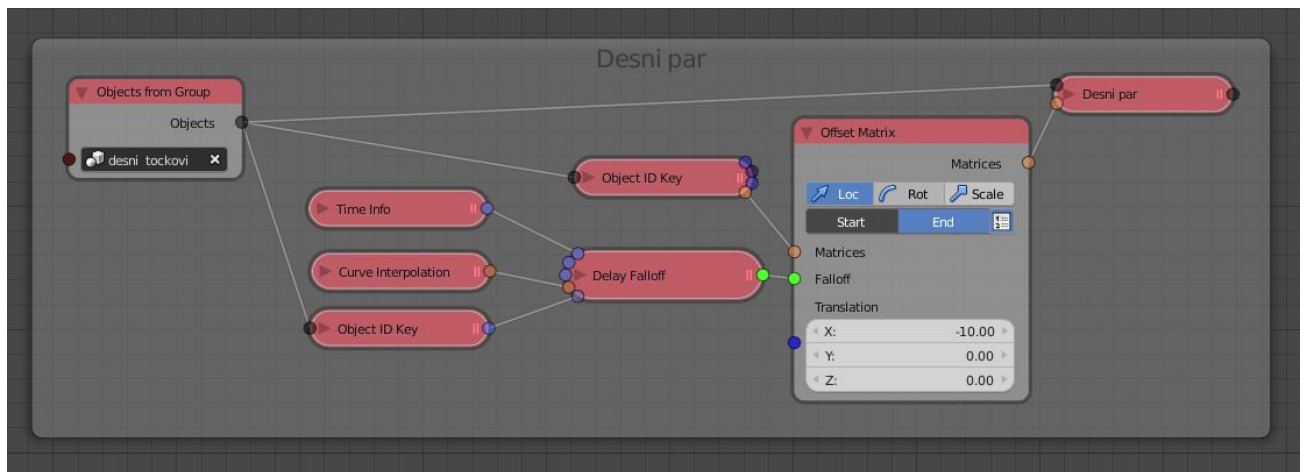


**Slika 4.7.** Gotovo stablo Animacijskih čvorova

Možemo primijetiti kako kotači prolaze kroz osovine, zbog toga će se kotačima promijeniti smjer dolaska na scenu. Potrebno je grupirati lijevi par i desni par kotača. Kopirati cijelo stablo sa slike (Slika 4.6.) za svaki par kotača posebno. U *Object from Group* čvoru postaviti maloprije napravljene grupe, te u *Offset Matrix* čvoru lokaciju Z vratiti na 0, a povećavati vrijednost X osi za lijevi par kotača i smanjiti vrijednost X osi za desni par kotača (Slika 4.8. i 4.9.).

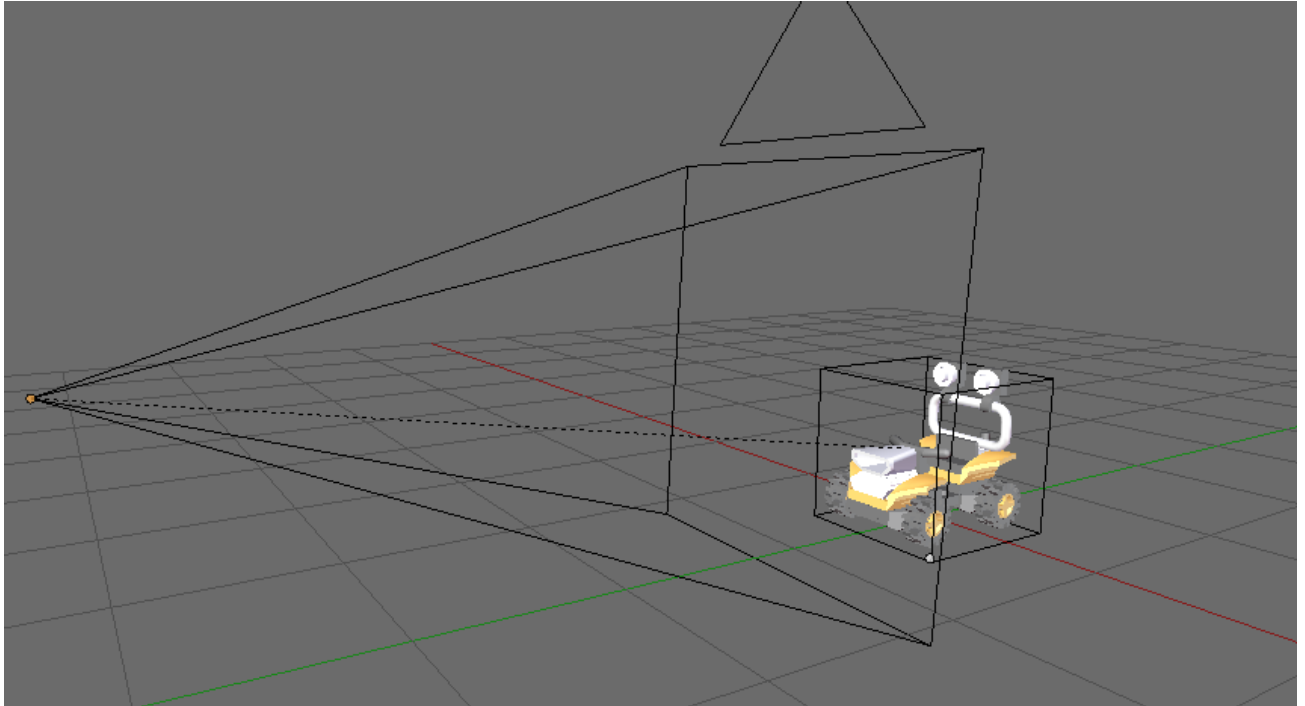


**Slika 4.8.** Stablo čvorova za lijevi par kotača

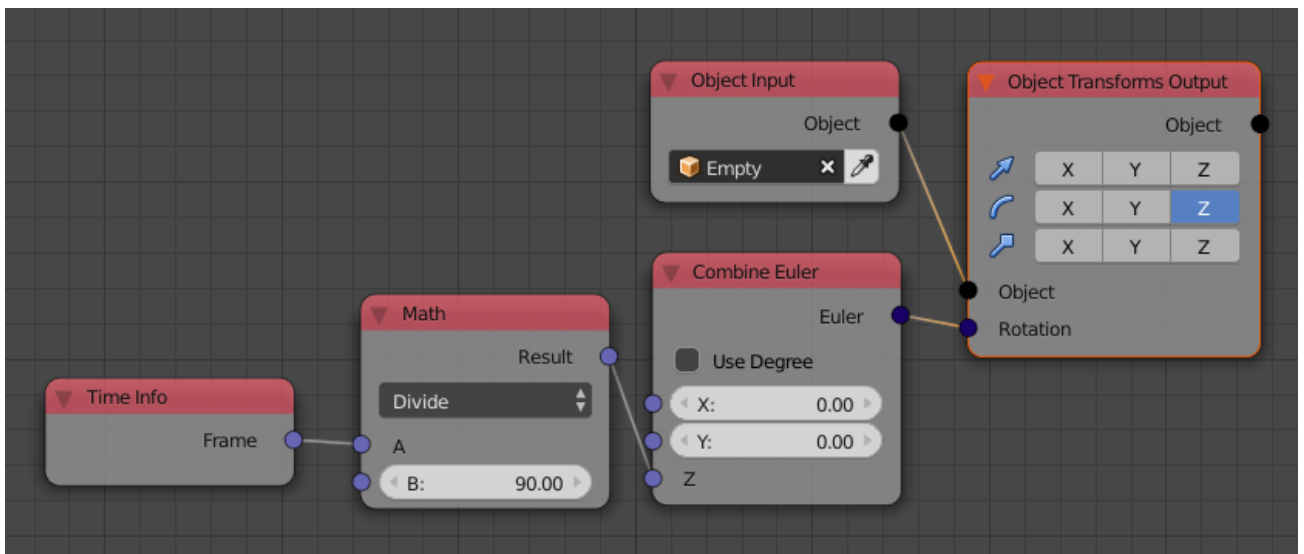


**Slika 4.9.** Stablo čvorova za desni par kotača

Rotaciju cijelog modela oko središta postiže se rotacijom kamere oko modela. Za to je potreban prazni objekt u sredini modela. Stvara se prazna kocka (engl. Empty cube), kameru se postavlja prema središtu modela, selektira se kamera, zatim uz držanje tipke *Shift* selektira praza kocka, pritisne se *Ctrl + P* te odabere *Object*. S ovime se povezala kamera s praznim objektom, sada je dovoljno rotirati prazan objekt a kamera će ga pratiti orijentirana prema objektu (Slika 4.10.). Pravi se novo stablo, u *Object Input* čvor postavlja se prazan objekt i spaja ga se na *Object Transform Output* čvor. Uključuje se rotacija po Z osi. Spaja se *Time Info* čvor u ulaz *Object Transform Output* čvora rotacije. Automatski se ubacuje *Combine Euler* čvor u kojem treba premjestiti vezu s X na Z kako bi se prazan objekt vrtio po Z osi. Za usporavanje rotacije koristi se *Math* čvor s operacijom dijeljenja.

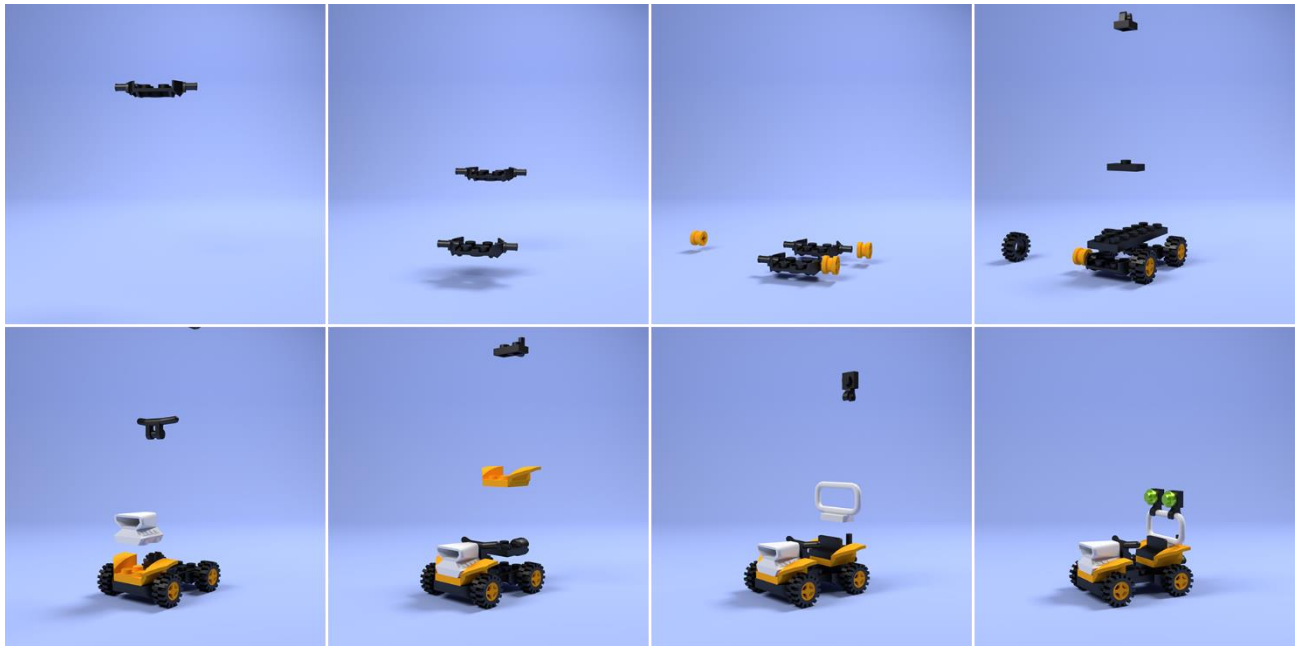


**Slika 4.10.** Prazna kocka povezana s kamerom



**Slika 4.11.** Stablo čvorova za rotaciju kamere

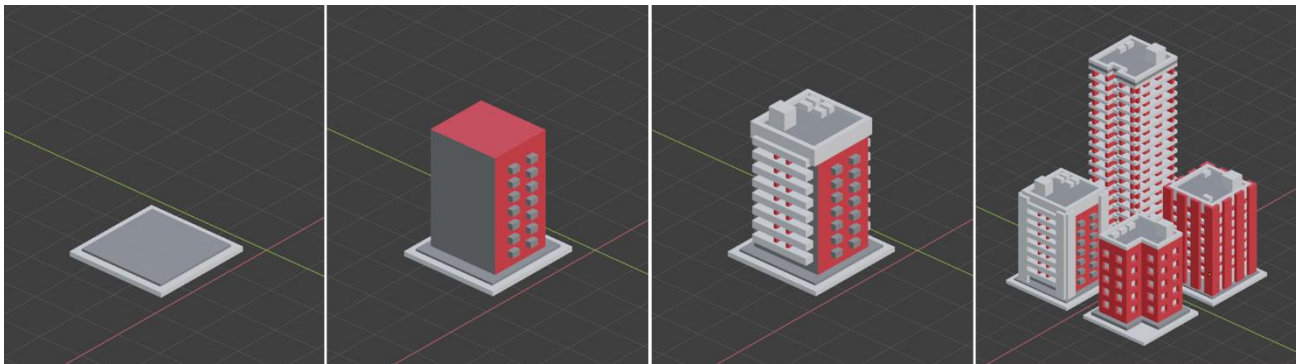
Preostaje pripraviti scenu i narediti postavke *rendera*, objašnjeno u zadnjem poglavlju.



**Slika 4.12.** Rednerana animacija slaganja *Lego kockica*

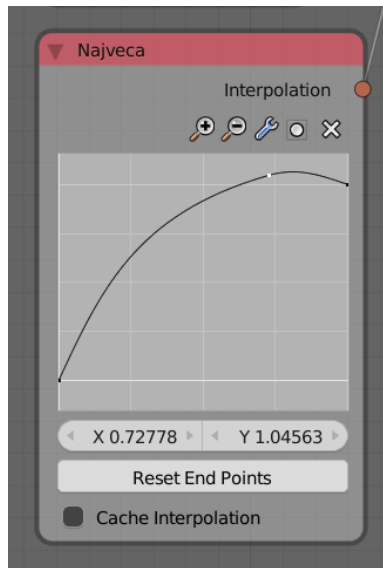
## 4.2. Podizanje stambenog naselja

Ova animacija napravit će se u *lowpoly* stilu s izometričnom kamerom. Za modeliranje koristi se Blender 2.8 jer je sučelje preglednije i jednostavniji je za korištenje novim korisnicima Blendera. Prošla animacija se nije animirala u njemu jer je još uvijek u Beta verziji i moguće su pogreške. Zgrade se modeliraju od kocke kojoj je potrebno premjestili podrijetlo (engl. origin) u centar donje stranice zbog animacije tako da se uđe u *Edit* mode pritiskom tipke *Tab*, zatim pritiskom tipke *G* pa tipke *Z* pomičemo kocku po *Z* osi. Izlazi se iz edit moda ponovnim pritiskom na *Tab*. Kocka se smanjuje po osima kraticom s tipkovnice *S* te pritiskom na jednu od tipki *X*, *Y* ili *Z* ovisno po kojoj je osi želi smanjiti, odnosno povećati kocku.

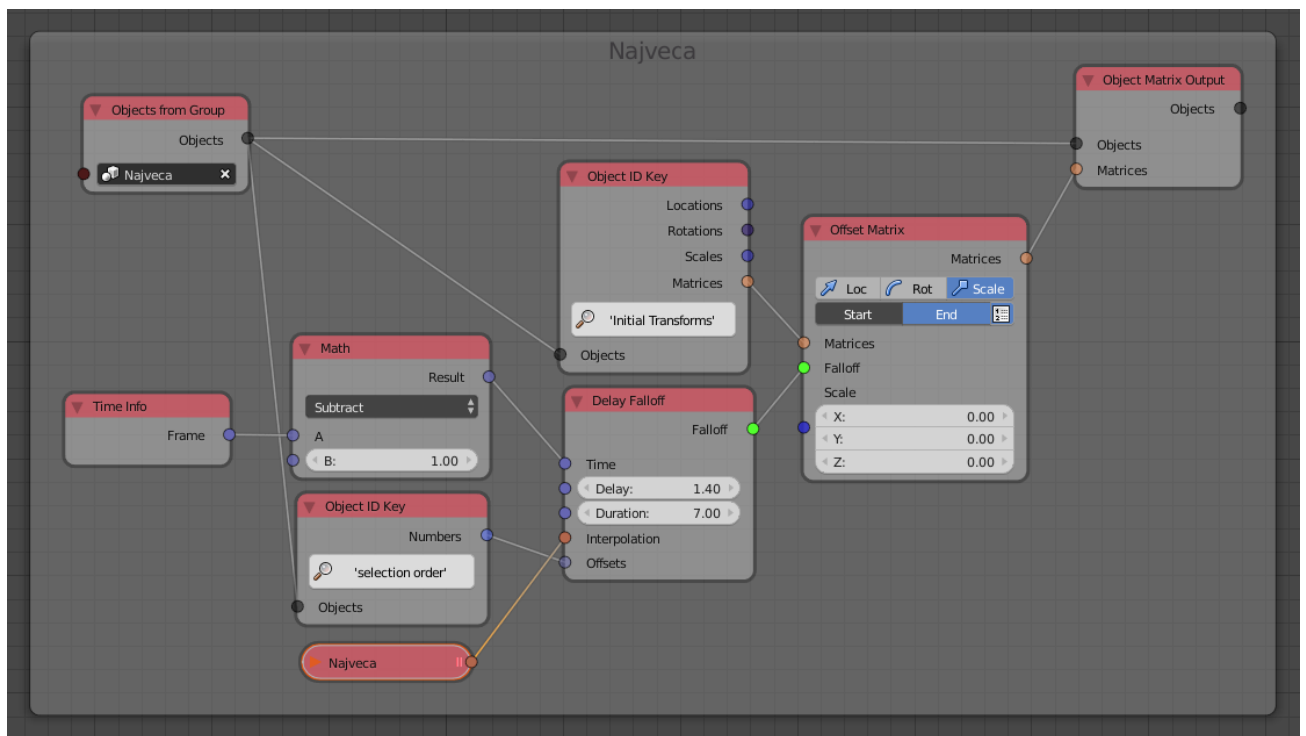


**Slika 4.13.** Modeliranje zgrada

Kada je modeliranje završeno sprema se model pritiskom na *File / Export / .xml* i uvozi ga se u novi dokument Blendera 2.79 gdje će se animirati. Grupira se svaka zgrada posebno. Pravi se novo stablo Animacijskih čvorova u kojeg se dodaje čvorove kao u prošlom primjeru (Slika 4.7.). Unutar *Object Input* čvora postavlja se prva zgrada, selektiraju se elementi zgrade željenim redoslijedom pojavljivanja te pravi cjelobrojni ID ključ. Unutar *Offset Matrix* čvora odabire se *X*, *Y* i *Z* komponenta skale. Ovime se postiglo nastajanje pojedinog elementa u sve tri dimenzije. Zbog boljeg vizualnog dojma, podizanjem krivulje interpolacije pri kraju preko vrijednosti 1 rezultat će preuveličavanjem nastajanja (Slika 4.14.).



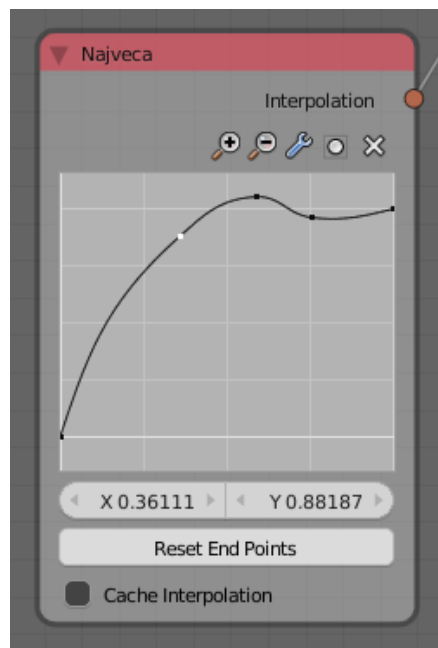
Slika 4.14. Izgled krivulje interpolacije



Slika 4.15. Stablo čvorova za animaciju najveće zgrade

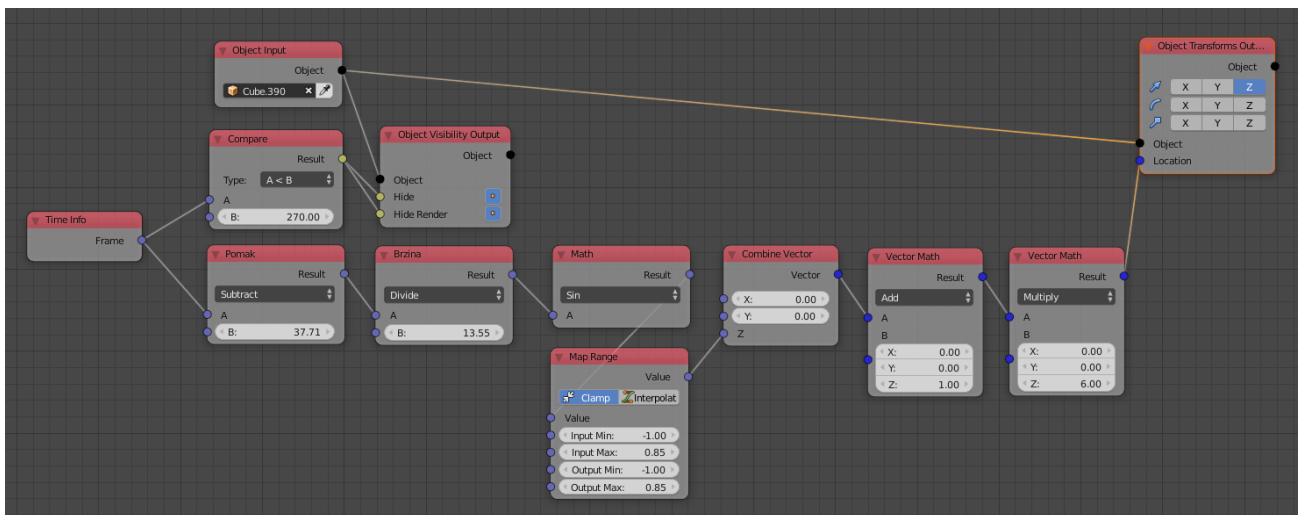


Isti proces ponovi se s preostalim zgradama i doda *Math* čvor s operacijom oduzimanja u vezu *Time Info* i *Delay Falloff* čvora kako bi se odgodili početak nastajanja svake zgrade. B ulaz predstavlja okvir u kojem će početi animacija. Isto se može postići s *Delay Time* čvorom. Nekim dijelovima zgrada nije potrebno skaliranje po sve tri osi, nego je vizualno ljepše skalirati samo po Z osi poput stupova zgrade. To se postiže kopiranjem cijelog stabla pojedine zgrade i postavljanjem vrijednosti 1 za X i Y ulaz u *Offset Matrix* čvor.

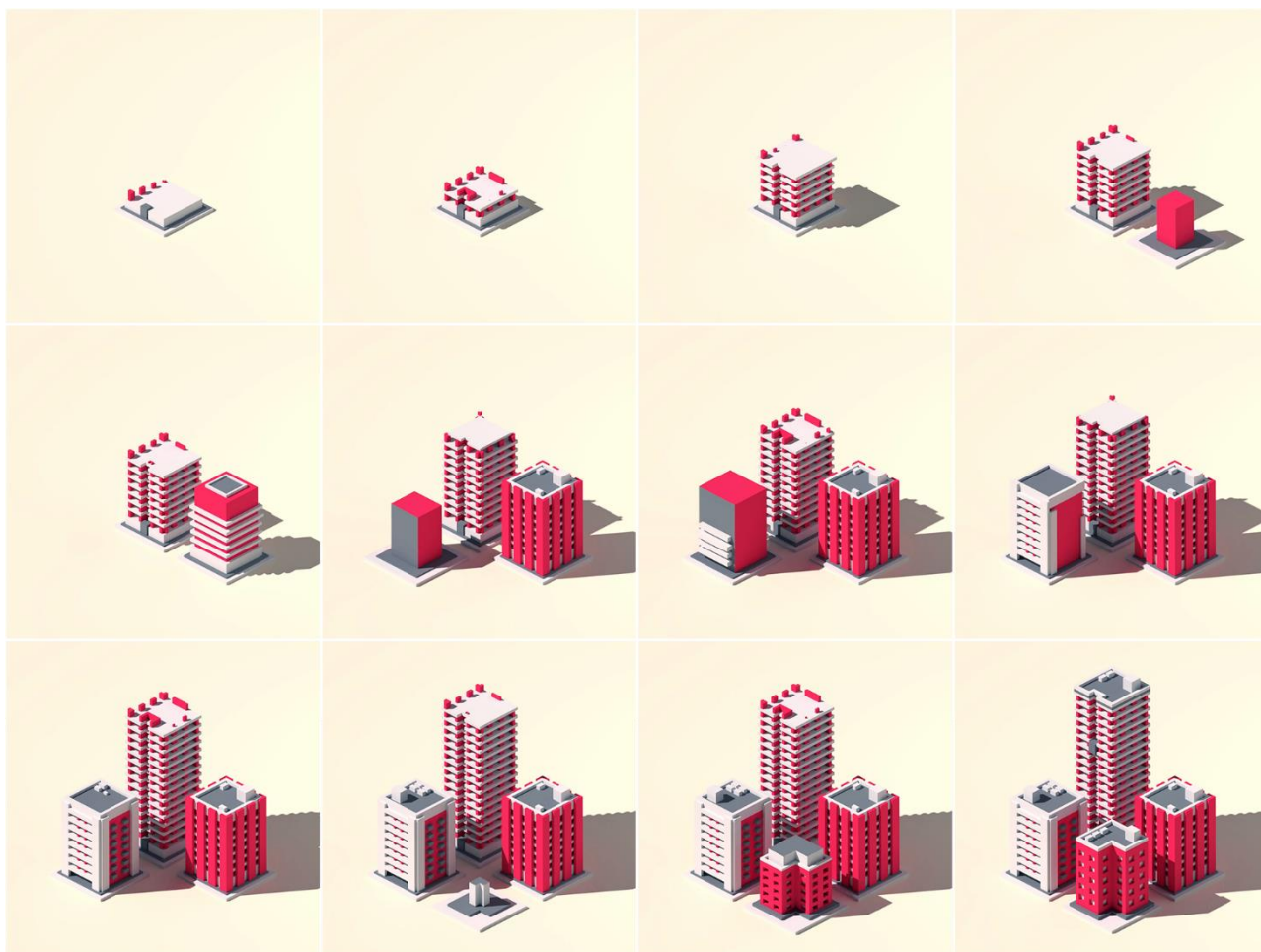


**Slika 4.16.** Izgled krivulje interpolacije s odskokom upotrijebljen na zadnjoj zgradi

Za animaciju dizala na najvećoj zgradi korištena je sinusna funkcija s dodatna dva *Math* čvora i dva *Vektor Math* čvora. Jedan *Math* čvor s operacijom sinus koji pomiče dizalo po Z komponenti određenom u *Object Transform Output* čvoru, drugi s operacijom dijeljenja koji predstavlja brzinu i treći s operacijom oduzimanja koji predstavlja pomak. Zaustavljanje lifta određeno je u *Map Range* čvoru s opcijom *Clamp* gdje sve iznad maksimalne vrijednosti (0.85) ostaje na toj maksimalnoj vrijednosti. Pomoću *Vektor Math* čvora i operacije množenja namještena je amplituda sinusnog signala. Dizalo je potrebno sakriti u prvom dijelu animacije, za to koristi logička operacija manje u *Compare* čvoru. Ulaz B predstavlja okvir pojavljivanja. Izlaz čvora je *boolean* spojen na *Object Visibility Output* čvor koji će sakriti objekt ako je ulaz *Hide* istinit, odnosno 1 .



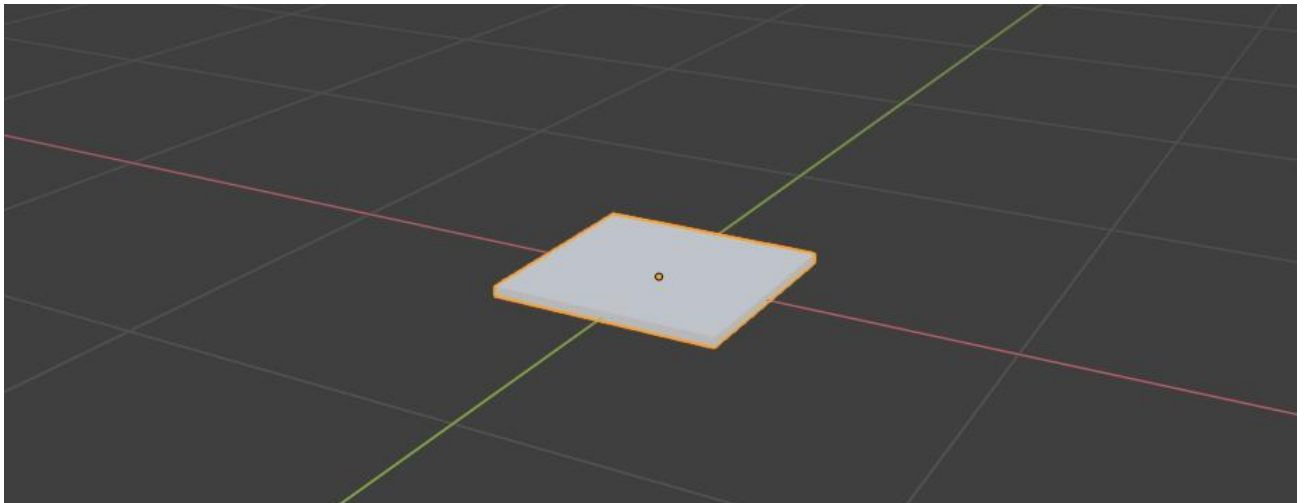
**Slika 4.17.** *Stablo čvorova za animaciju dizala na najvećoj zgradi*



**Slika 4.18.** *Izgleđ krajnje animacije podizanja stambenog naselja*

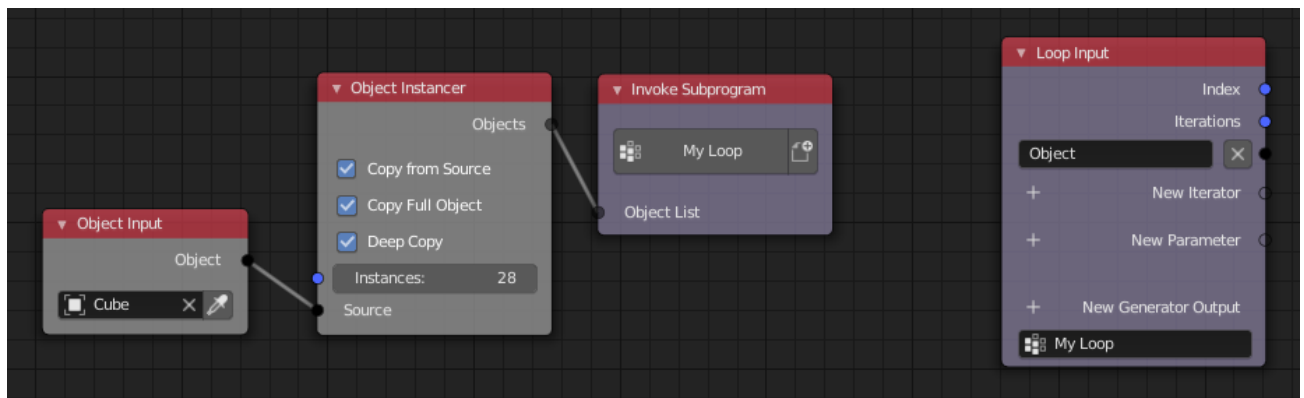
### 4.3. Prateći kvadrati

Za ovu animaciju koristiti se jedna kocka i jedan prazni objekt (prazna kocka). Pravi se novi dokument. U njemu se već nalazi kocka kojoj se mijenja visina pritiskom tipke S pa Z i pomicanjem miša smanji joj se visina (Slika 4.19.). Potrebno joj je primijeniti promjenu visinu klikom na padajući izbornik *Object / Apply / Scale*.



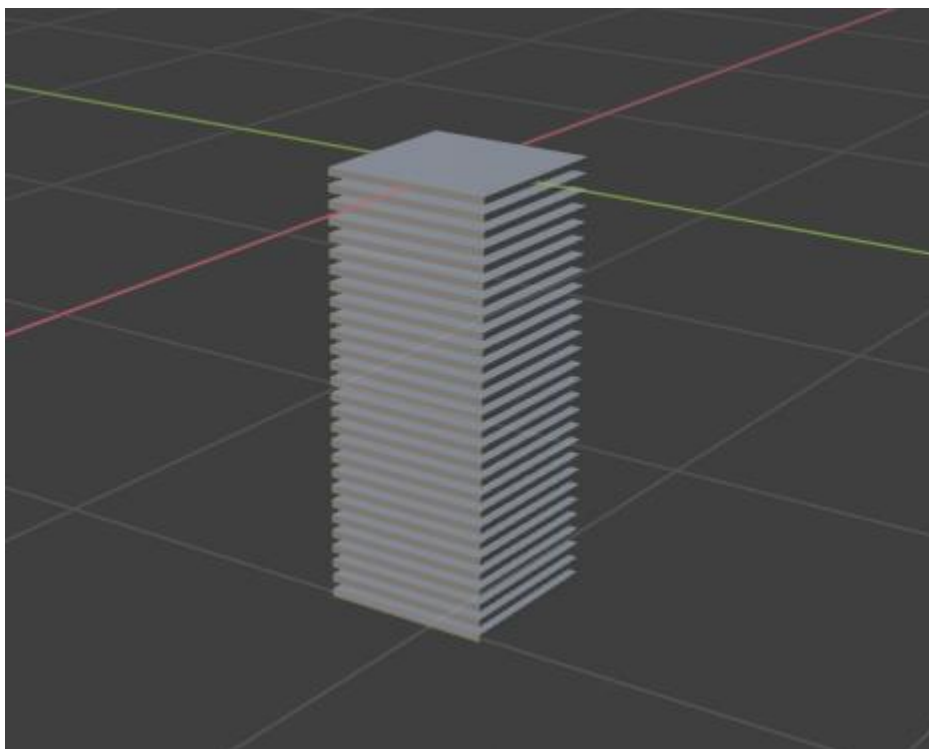
**Slika 4.19.** Kocka prilagođena za animaciju

Pravimo novo stablo Animacijskih čvorova. Unutar *Object Input* čvora stavlja se promijenjena kocka i spaja se izlaz čvora na ulaz *Object Instancer* čvora *Source*. Odaberu se opcije *Copy Full Object*, *Deep Copy* i poveća broj kopija (28). *Object Instacer* čvor je generirao kopije našeg objekta koje se sve nalaze na istom mjestu. Može se sakriti početna kocka jer više nije potrebna klikom na ikonu oka (za sakrivanje u 3D prikazu) i ikonu fotoaparata (za skrivanje u *renderu*). Kako bi upravljali svakim objektom zasebno potrebna je petlja potprograma *Invoke Subporgram*. Za to služi *Invoke Subprogram* čvor s tipom *Loop*. Automatski se dodaje *Loop Input* čvor u kojem se pravi nova lista ulaza klikom na plus pokraj *New Iterator* tipa *Object List*. Izlaz čvora *Object Instancer* spaja se na novo stvoreni ulaz *Invoke Subprogram* čvora *Object List* (Slika 4.20.). Time je završena izrada petlje, dalje se čvorovi spajaju u novo stablo potprograma *Loop Input*.

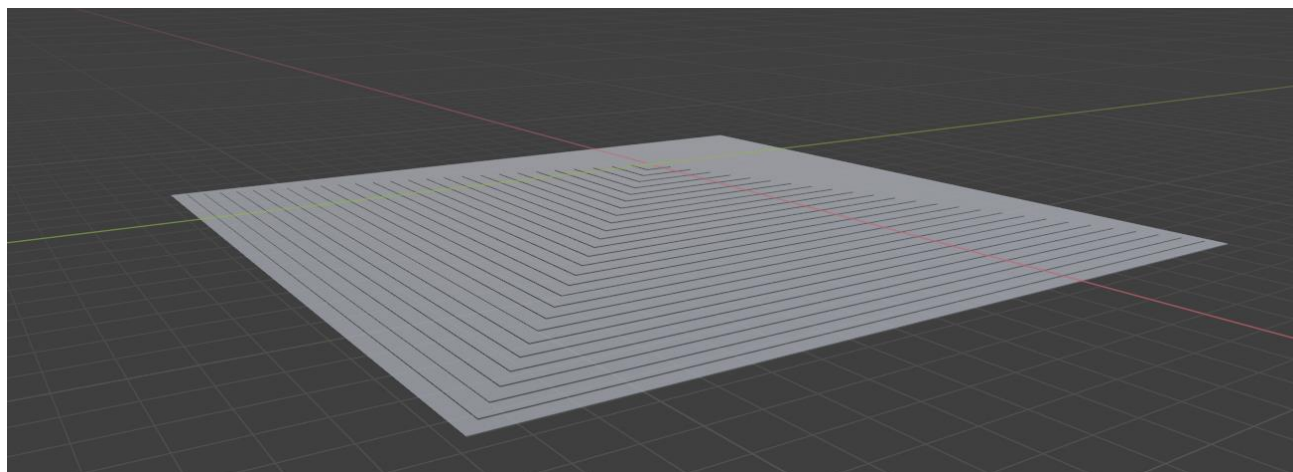


**Slika 4.20.** Potprogram s petljom

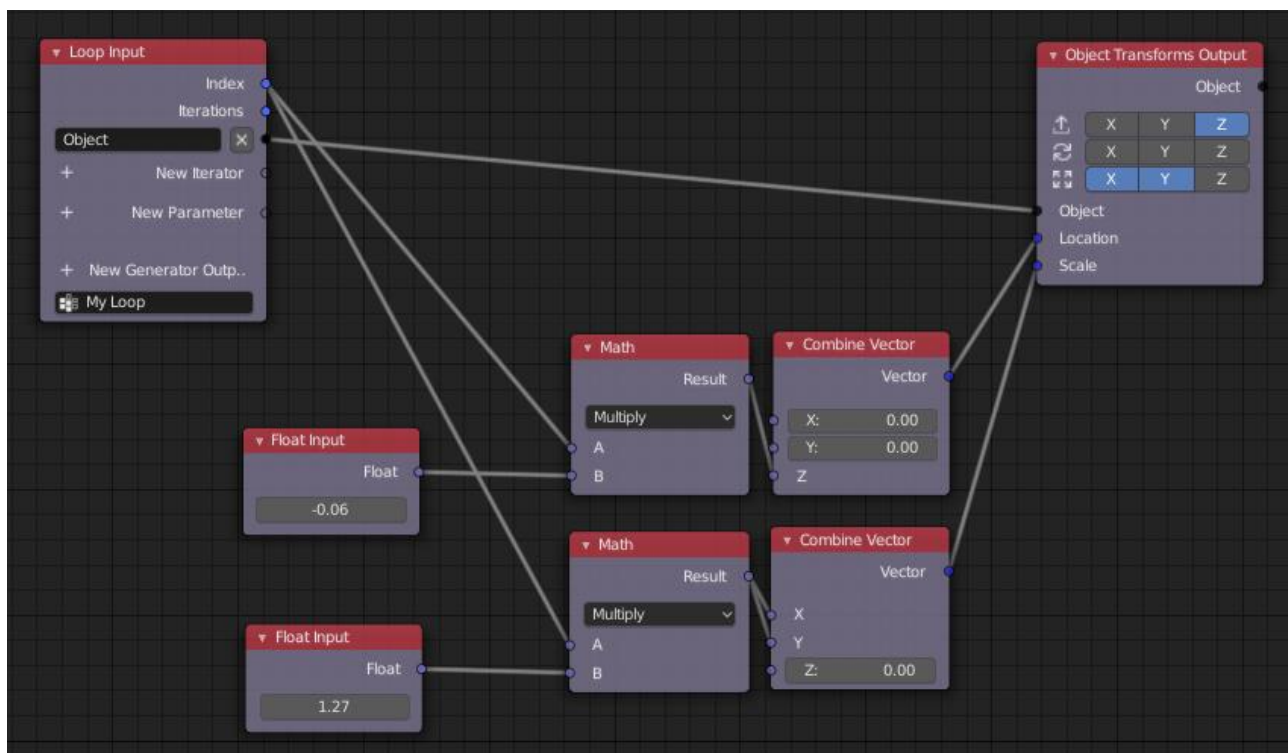
Objekte će se poslagati u red po Z osi tako da svaki sljedeći bude veći od prijašnjeg. Za to treba *Object Transform Output* čvor koji će transformirati objekte po ulaznim parametrima. Na njegov ulaz *Object* spaja se izlaz *Object* iz *Loop Input* čvora. Odabire se Z komponenta lokacije. Na ulaz lokacije spaja se izlaz *Index* preko *Math* čvora s operacijom množenja s malim negativnim decimalnim brojem spojen na Z komponentu vektora lokacije. Indeksi kreću od 0, što znači lokacija prvog objekta neće biti promijenjena, lokacija drugog će biti jedan put manja za veličinu decimalnog broja, lokacija drugog dva puta i tako dalje (Slika 4.21.). Objektima poslaganim u red potrebno je promijeniti dužinu i širinu. Odabere se X i Y komponenta skale u *Object Transform Output* čvoru i spaja se izlaz *Index* preko *Math* čvora s operacijom množenja spojen na X i Y komponentu vektora skale (Slika 4.22.). Po želji može se podiči sve kako objekti ne bi bili u negativnom dijelu Z osi ubacivanjem još jednog *Math* čvora s operacijom zbrajanja u granu gdje se mijenja Z komponenta vektora lokacije.



**Slika 4.21.** *Objekti poslagani u red po Z osi*

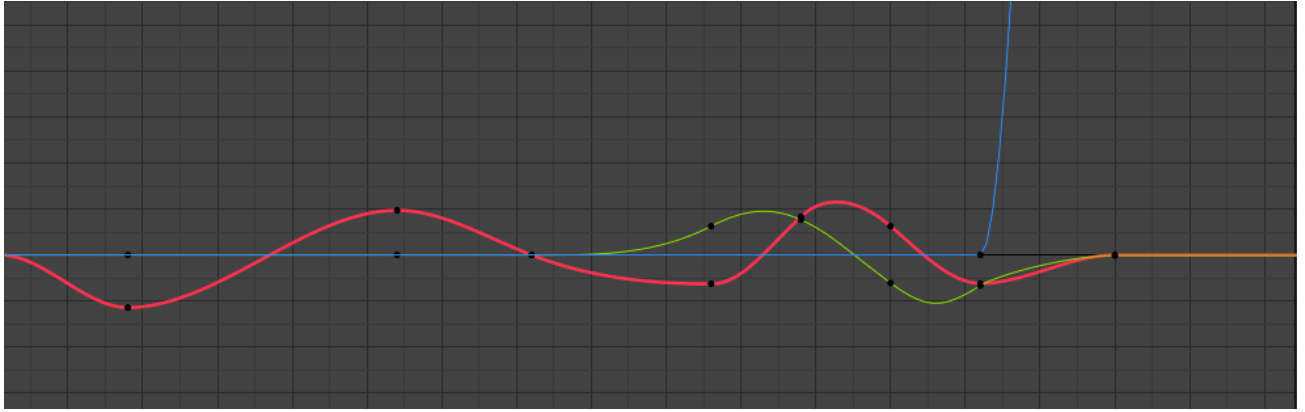


**Slika 4.22.** *Objekti poslagani u red po Z osi s promijenjenom visinom i širinom*



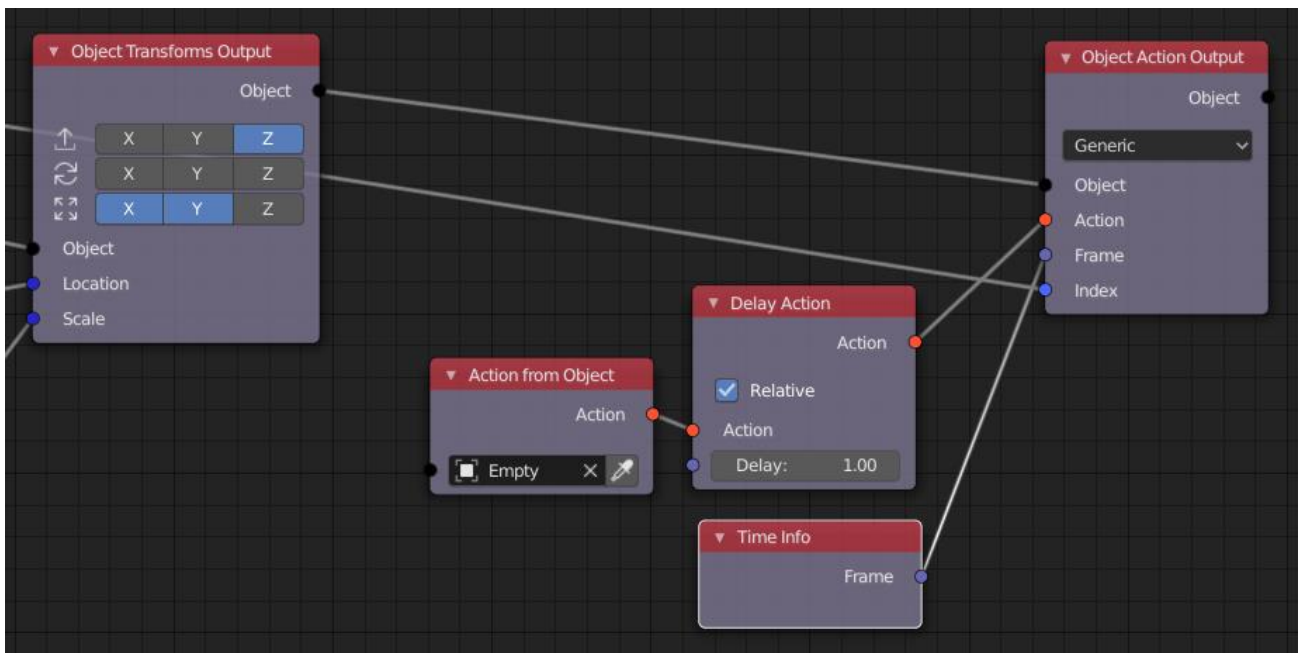
**Slika 4.23.** Izgled stabla čvorova za promjenu veličini i lokacije objektima

Objekti će se animirati da prate prvi objekt tako da svaki kasni za prethodim, zapravo svi prate radnju praznog objekta kojeg je potrebno ručno animirati. Dodaje se na scenu prazna kocka. Postavlja se pokazivač na početak vremenske crte, unutar *Properties Editor* pod karticom *Object* zabilježi se lokacija X i Y te rotacija Z klikom na točke pokraj ikona lokota, tako smo postavili *keyframeove*. *Keyframe* je točka u vremenu na kojoj se događa promjena neke vrijednosti, stavljaju se na početku i na kraju radnje te na ekstremima. Program će sam napraviti kontinuiranu promjenu veličine od jednog do drugog *keyframea*. Nakon promjene vrijednosti potrebno je potvrditi promjenu pritiskom na tipku I i odabiranjem *Available*. Radnju se može korigirati pomoću graf editora unutar vremenske crte (Slika 4.24).



**Slika 4.24.** Izgled grafa animirane prazne kocke (crveno lokacija po X osi, zeleno lokacija po Y osi, plavo rotacija po Z osi)

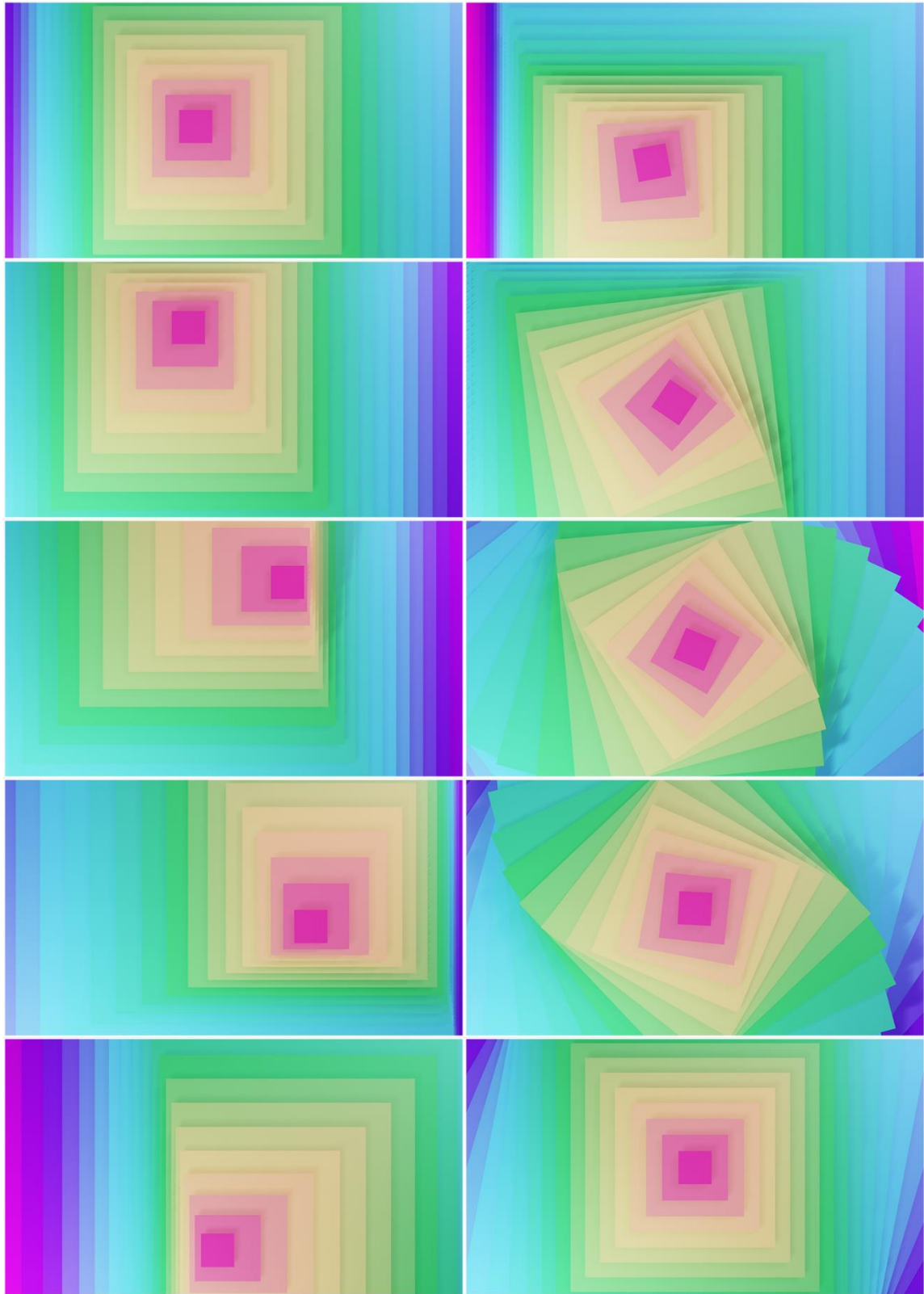
Za praćenje radnje praznog objekta služi *Object Action Output* čvor koji će transformirati objekt po ulaznoj radnji (u ovom slučaju kretanju prazne kocke). Na ulaz *Action* spaja se *Action from Object* čvor s praznom kockom, a na ulaz *Frame* postavlja se *Time Info* čvor. Sada svi objekti prate radnju prazne kocke. Potrebno je ubaciti kašnjenje u granu radnje pomoću *Delay Action* čvora s uključenom postavkom *Relative* i kašnjenjem (engl. Delay) od 1.00 (Slika 4.25.).



**Slika 4.25.** Proširenje stabla za praćenje radnje prazne kocke







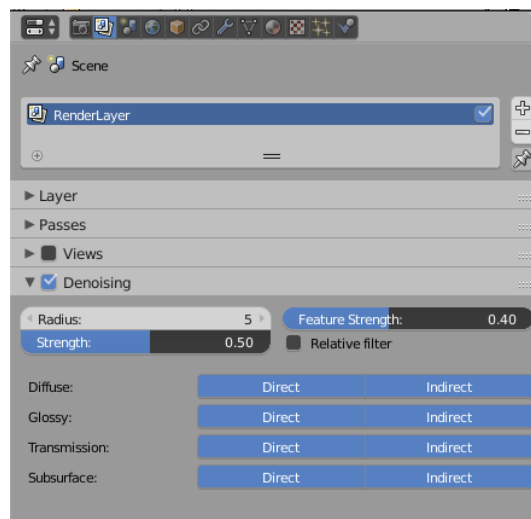
**Slika 4.27.** *Renderirana animacija pratećih kvadrata (poredano po stupcima)*

## 4.4. Render

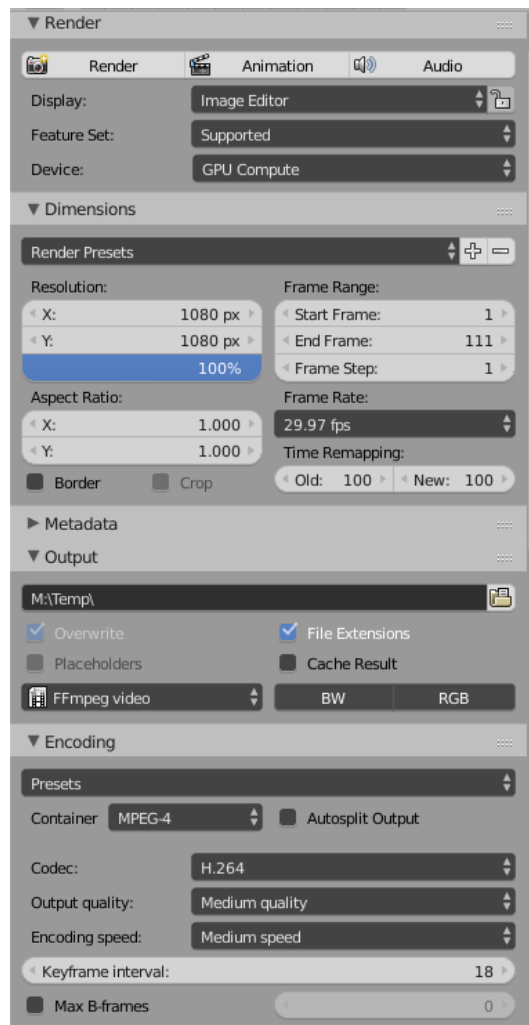
Za pozadinu se koristi ravnina (engl. plane) koja se povećava kako bi prekrila kadar, po potrebi istiskuju (engl. extrude) rubovi i zaobljuje s *Bevel Edge* alatom i *Subdivision surface* modifikatorom. Na scenu se dodaju svijetla.

*Render* je proces kojim računalo obrađuje 3D scenarij i kreira videozapis ili sliku. Za potrebe ovog rada koristi se *Cycles Render engine* koji radi na principu grafičke metode *path tracing*. Dizajniran za pružanje fizičkih utemeljenih rezultata za potrebe produkcije.

Animacija se *renderira* u Ffmpeg video formatu, MPEG-4 formatu digitalnog multimedijskog spremnika i H.264 kodekom (engl. Codec) (Slika 4.29.). Postavlja se željena rezolucija i duljina trajanja videozapisa određivanjem prvog i zadnjeg okvira. Povećanjem broja uzoraka (engl. sample) *rendera* videozapis će imati manje šuma (engl. Noise), ali će produžiti vrijeme *rendera*. Unutar Blendera postoji opcija uklanjanja dijela šuma sa slike (engl. Denoising). Opcija se nalazi unutar postavki *Render Layer* (Slika 4.28.). Uklanjanje šuma se obavlja na svakom okviru nakon što je *renderiran*. Zbog te opcije broj uzoraka *rendera* može biti niži, a da slika bude bez šuma. U ovom slučaju broj uzoraka je postavljen na 512.



**Slika 4.28.** Opcija *Denoising*



**Slika 4.29.** *Postavke rendera*

## 5. ZAKLJUČAK

Radom je prikazano da se u Blenderu uz pomoć Animacijskih čvorova može na jednostavan način vizualizirati mnoštvo proceduralnih animacija. Mogućnost lagane izmjene, već dovršenog projekta, još je jedna od prednosti koji ovaj dodatak čine efikasnim. Također, korištenje velikog broja objekata neće otežati postupak izrade animacije.

Animacije se mogu koristiti u širokom spektru primjene. Od izmišljenih do potpuno realnih pojava ili stvari. Primjeri slaganja Lego kockica i podizanje stambenog naselja mogu se interpretirati kao postupak slaganja automobila, odnosno redosljed gradnje stambenih objekata.

Na sličan način korisnici Blendera mogu dočarati događaje iz stvarnoga života. Odnosno, izrađene animacije mogu se koristiti za testiranja, predviđanja, razna istraživanja i simuliranje viđenog.

## LITERATURA

- [1] About Blender [online], Blender 2.79 Manual, dostupno na: [https://docs.blender.org/manual/en/dev/getting\\_started/about/index.html](https://docs.blender.org/manual/en/dev/getting_started/about/index.html) [15.8.2019.]
- [2] Installation [online], Animation Nodes Manual, dostupno na: [https://animation-nodes-manual.readthedocs.io/en/latest/user\\_guide/install/install.html](https://animation-nodes-manual.readthedocs.io/en/latest/user_guide/install/install.html) [15.9.2019.]
- [3] Interface [online], Animation Nodes Manual, dostupno na: [https://animation-nodes-manual.readthedocs.io/en/latest/user\\_guide/getting\\_started/interface.html](https://animation-nodes-manual.readthedocs.io/en/latest/user_guide/getting_started/interface.html) [15.9.2019.]
- [4] Execution [online], Animation Nodes Manual, dostupno na: [https://animation-nodes-manual.readthedocs.io/en/latest/user\\_guide/getting\\_started/execution.html](https://animation-nodes-manual.readthedocs.io/en/latest/user_guide/getting_started/execution.html) [15.9.2019.]
- [5] Nodes [online], Animation Nodes Manual, dostupno na: [https://animation-nodes-manual.readthedocs.io/en/latest/user\\_guide/nodes/index.html](https://animation-nodes-manual.readthedocs.io/en/latest/user_guide/nodes/index.html) [19.9.2019.]
- [6] Tutorials [online], BlenderDiplom, dostupno na: <https://www.blenderdiplom.com/en/tutorials.html> [19.9.2019.]
- [7] Jacques Lucke [online] YouTube dostupno na: <https://www.youtube.com/channel/UC5ABAUgEvBMmau-1xJsNw6w> [19.9.2019.]
- [8] Zach Hixson [online] YouTube, dostupno na: [https://www.youtube.com/channel/UC\\_Yypfyx5fKHpd640wPGFzQ](https://www.youtube.com/channel/UC_Yypfyx5fKHpd640wPGFzQ) [19.9.2019.]
- [9] Modeling [online], Blender 2.79 Manual, dostupno na: <https://docs.blender.org/manual/en/dev/modeling/index.html> [13.9.2019.]
- [10] Render [online], Blender 2.79 Manual, dostupno na: <https://docs.blender.org/manual/en/dev/render/index.html> [13.9.2019.]
- [11] MPEG-4 [online], Wikipedia, dostupno na: <https://hr.wikipedia.org/wiki/MPEG-4> [17.9.2019.]

## SAŽETAK

**Naslov:** Animacijski čvorovi u Blenderu

U završnom radu elaborira se korištenje dodatka za Blender *Animation Nodes*. Detaljno je objašnjen postupak izrade proceduralne animacije slaganja Lego kockica, podizanje stambenih zgrada i apstraktne animacije pomoću jednog objekta. Opisana je prilagodba postavki kamere i rendera.

**Ključne riječi:** animacija, Blender, čvor, Lego

## **ABSTRACT**

**Title:** Animation nodes in Blender

This final paper elaborates on the use of Blender Animation Nodes plugin. The process of creating a generic animation of stacking Lego cubes, erecting residential buildings and abstract animation using a single object is explained in detail. Also described is the adaptation of the camera and render settings.

**Key word:** animation, Blender, Lego, node



## **ŽIVOTOPIS**

Marko Varšava rođen je 18.8.1996. godine u Đakovu, živi u Osijeku. Pohađao je Osnovnu školu „Ivan Goran Kovačić“ te 2011. godine upisuje prirodoslovnu matematičku gimnaziju Antuna Gustava Matoša u Đakovu. 2015. godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

---

Marko Varšava

## **PRILOZI**

- [1] Videozapis: Slaganje Lego kockica
- [2] Videozapis: Podizanje stambenog naselja
- [3] Videozapis: Prateći kvadrati