

Stvaranje fraktalnih struktura pomoću 3d printera

Barbić, Nina

Master's thesis / Diplomski rad

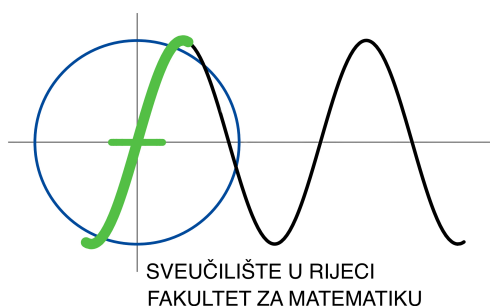
2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:196:280230>

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

Download date / Datum preuzimanja: **2024-05-16**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Mathematics - MATHRI Repository](#)



Sveučilište u Rijeci – Fakultet za matematiku

Diplomski sveučilišni studij matematike – smjer nastavnički

Nina Barbić

Stvaranje fraktalnih struktura pomoću 3D printera

Diplomski rad

Rijeka, 2022.

Sveučilište u Rijeci – Fakultet za matematiku

Diplomski sveučilišni studij matematike – smjer nastavnički

Nina Barbić

Stvaranje fraktalnih struktura pomoću 3D printera

Diplomski rad

Mentor:

doc. dr. sc. Bojan Crnković

Rijeka, 2022.

SAŽETAK

U ovom radu razmatra se način izrade fraktala Tepiha Sierpińskog te Mengerove spužve i njezinog negativa pomoću 3D printera. Rad je podijeljen na četiri glavna dijela. Prvi dio rada posvećen je fraktalima. Osim što će se saznati što su to fraktali, koja su njihova svojstva te na koji način ih je moguće podijeliti, pružit će se sažet pregled njihove povijesti. Također, predstaviti će se nekoliko poznatih fraktala. U drugom dijelu rada dat će se uvid u vrlo zanimljivu tehnologiju razvijenu 1980-ih godina, 3D printanje. Dat će se sažet pregled povijesti 3D printanja. Navest će se najpoznatije tehnologije 3D printanja te materijali koji se koriste pri 3D printanju. Detaljno će se opisati proces 3D printanja te tehnologija koja je korištena u praktičnom dijelu rada. Treći dio rada namijenjen je upoznavanju sa softverom Blender kojeg će se koristiti u praktičnom dijelu rada pri 3D modeliranju. Opisane su tehnike 3D modeliranja. Predstavljen je dodatak Animacijski čvorovi pomoću kojeg će se u praktičnom dijelu izraditi 3D modeli. Četvrti dio rada ujedno je njegov praktični dio. Detaljno će se opisati postupak kojim su izrađeni 3D modeli fraktala Tepiha Sierpińskog te Mengerove spužve i njezinog negativa. Ostatak provedenog procesa 3D printanja kao što su spremanje STL datoteke, rezanje 3D modela i naknadna obrada samo će se ukratko opisati. Na kraju rada priložit će se slike provedenih postupaka.

Ključne riječi: fraktali, 3D printanje, 3D modeliranje, 3D model, Blender, Tepih Sierpińskog, Mengerova spužva, Animacijski čvorovi

SADRŽAJ

| | |
|--|----|
| 1. UVOD | 1 |
| Slika 1.1. Benoit Mandelbrot (1924.-2010.) | 1 |
| 2. FRAKTALI | 2 |
| 2.1. Što su fraktali? | 2 |
| Slika 2.1. <i>Paprat</i> | 2 |
| 2.2. Svojstva fraktala | 3 |
| 2.2.1. Oblikovanje iteracijom | 3 |
| Slika 2.2. <i>Prva četiri koraka konstrukcije Trokuta Sierpinskog</i> | 3 |
| 2.2.2. Samosličnost | 3 |
| Slika 2.3. <i>Samoslična krivulja</i> | 3 |
| 2.2.3. Fraktalna dimenzija | 4 |
| Slika 2.4. <i>Samosličnost linije, kvadrata i kocke</i> | 4 |
| Slika 2.5. <i>Nulta i prva iteracija Tepiha Sierpinskog</i> | 5 |
| 2.3. Poznati fraktali i njihova povijest | 5 |
| 2.3.1. Weierstrassova funkcija | 5 |
| Slika 2.6. <i>Graf Weierstrassove funkcije</i> | 6 |
| 2.3.2. Cantorov skup | 6 |
| Slika 2.7. <i>Prva četiri koraka konstrukcije Cantorovog skupa</i> | 7 |
| Slika 2.8. <i>Prva tri koraka konstrukcije Cantorove prašine</i> | 7 |
| Slika 2.9. <i>Prva tri koraka konstrukcije Cantorovog oblaka</i> | 7 |
| 2.3.3. Kochova krivulja i pahuljica | 8 |
| Slika 2.10. <i>Prva četiri koraka konstrukcije Kochove krivulje</i> | 8 |
| Slika 2.11. <i>Prva četiri koraka konstrukcije Kochove pahuljice</i> | 9 |
| 2.3.4. Trokut Sierpinskog | 9 |
| Slika 2.12. <i>Prva četiri koraka konstrukcije Trokuta Sierpinskog</i> | 9 |
| Slika 2.13. <i>Tetraedar Sierpinskog</i> | 10 |
| 2.3.5. Tepih Sierpinskog | 11 |
| Slika 2.14. <i>Prva četiri koraka konstrukcije Tepiha Sierpinskog</i> | 11 |
| 2.3.6. Mengerova spužva | 12 |
| Slika 2.15. <i>Prva četiri koraka konstrukcije Mengerove spužve</i> | 12 |

| | |
|---|----|
| Slika 2.16. <i>Komplementi prvih triju iteracija Mengerove spužve</i> | 14 |
| 2.3.7. Julijini skupovi | 14 |
| Slika 2.17. <i>Četiri primjera Julijinih skupova</i> | 15 |
| 2.3.8. Mandelbrotov skup | 15 |
| Slika 2.18. <i>Mandelbrotov skup</i> | 15 |
| 2.4. Podjela fraktala | 16 |
| 2.4.1. Podjela fraktala prema stupnju samosličnosti | 16 |
| 2.4.2. Podjela fraktala prema načinu njihova nastanka | 16 |
| 3. 3D PRINTANJE | 16 |
| 3.1. Što je 3D printanje? | 16 |
| Slika 3.1. <i>Original Prusa i3 MK3S 3D printer</i> | 17 |
| 3.2. Primjena 3D printanja | 17 |
| 3.3. Tehnologije 3D printanja | 17 |
| 3.3.1. FDM tehnologija | 17 |
| Slika 3.2. <i>Ilustracija 3D printanja pomoću FDM printera s dvije mlaznice</i> | 18 |
| 3.4. Povijest 3D printanja | 18 |
| 3.4.1. Temelji | 18 |
| 3.4.2. Komercijalizacija | 19 |
| 3.4.3. Masovna upotreba | 19 |
| 3.5. Proces 3D printanja | 20 |
| 3.5.1. 3D modeliranje | 20 |
| 3.5.2. Rezanje | 21 |
| Slika 3.3. <i>Početni zaslon PrusaSlicer softvera</i> | 21 |
| 3.5.3. 3D printanje | 22 |
| 3.5.4. Naknadna obrada | 22 |
| 3.6. Materijali za 3D printanje | 22 |
| 4. BLENDER | 23 |
| Slika 4.1. <i>Službeni logotip Blendera</i> | 23 |
| 4.1. Korisničko sučelje | 23 |
| Slika 4.2. <i>Uvodni zaslon inačice 2.93</i> | 23 |
| Slika 4.3. <i>Izgled sučelja nakon otvaranja Blendera</i> | 24 |
| Slika 4.4. <i>Izbornik Add</i> | 25 |
| Slika 4.5. <i>Dio uređivača 3D prikaz, namijenjen za navigaciju</i> | 25 |

| | |
|---|-----------|
| Slika 4.6. Bočna traka uređivača 3D prikaz | 26 |
| Slika 4.7. Izbornik modifikatora | 27 |
| 4.2. Načini uređivanja objekata | 27 |
| Slika 4.8. Izbornik načina uređivanja objekata | 28 |
| 4.3. Tehnike 3D modeliranja..... | 28 |
| Slika 4.9. Blenderovi geometrijski primitivi | 28 |
| Slika 4.10. Primjer 3D modela stvorenog pomoću poligonalne tehnike | 29 |
| Slika 4.11. Primjer NURBS krivulje i 3D modela stvorenog pomoću NURBS tehnike | 29 |
| Slika 4.12. Zaobljavanje površine objekta pomoću subdivizijske tehnike | 29 |
| Slika 4.13. 3D modeliranje u Blenderu pomoću Python skripte | 30 |
| 4.4. Animacijski čvorovi | 30 |
| 4.4.1. O dodatku | 30 |
| 4.4.2. Radni prostor | 30 |
| Slika 4.14. Radni prostor namijenjen za korištenje Animacijskih čvorova | 31 |
| Slika 4.15. Izbornici Add (lijevo) i Subprograms (desno) | 31 |
| 4.4.3. Izvršenje stabla čvorova | 31 |
| 4.4.4. Programi i potprogrami | 32 |
| 4.4.5. Petlja..... | 32 |
| Slika 4.16. Model izvršenja petlje u Animacijskim čvorovima | 32 |
| 4.4.6. Čvorovi..... | 32 |
| Slika 4.17. Čvor Invoke Subprogram i njegovo dodavanje | 33 |
| Slika 4.18. Čvor Loop Input..... | 33 |
| Slika 4.19. Čvor Distribute Matrices s opcijama Grid i Circle te jedan primjer rasporeda lokacija | 34 |
| Slika 4.20. Čvor Math i njegove operacije | 35 |
| 5. PRAKTIČNI DIO IZRADE TEPIHA SIERPINSKOG POMOĆU 3D PRINTERA | 36 |
| 5.1. 3D modeliranje u Blenderu | 36 |
| 5.1.1. Nulta iteracija..... | 36 |
| Slika 5.1. Početno sučelje Blendera..... | 36 |
| Slika 5.2. Strelica za otvaranje Bočne trake | 36 |
| Slika 5.3. Model nulte iteracije Tepiha Sierpińskog | 37 |
| 5.1.2. Prva iteracija | 37 |
| Slika 5.4. Normalizacija dimenzija manjoj „kocki“ | 38 |
| Slika 5.5. Dodavanje modifikatora | 38 |

| | |
|--|-----------|
| Slika 5.6. Odabrane postavke modifikatora Boolean | 39 |
| Slika 5.7. Model prve iteracije Tepiha Sierpińskog | 39 |
| Slika 5.8. Model prve iteracije, promatran pod pravim kutom | 40 |
| 5.1.3. Druga iteracija | 40 |
| Slika 5.9. Dodavanje novog radnog prostora | 40 |
| Slika 5.10. Izgled sučelja nakon dodavanja novog radnog prostora | 41 |
| Slika 5.11. Odabir vrste uređivača | 41 |
| Slika 5.12. Odabir načina izvršenja stabla čvorova | 41 |
| Slika 5.13. Stablo čvorova za drugu iteraciju | 42 |
| Slika 5.14. Skup objekata stvorenih pomoću Animacijskih čvorova | 42 |
| Slika 5.15. Postavke modifikatora Boolean | 43 |
| Slika 5.16. Izgled sučelja nakon izrade modela druge iteracije Tepiha Sierpińskog | 43 |
| Slika 5.17. Model druge iteracije, promatran pod pravim kutom | 44 |
| Slika 5.18. Vizualizacija uz pomoć čvora 3D Viewer: čvor Distribute Matrices | 45 |
| 5.1.4. Treća iteracija | 45 |
| Slika 5.19. Stabla čvorova za treću iteraciju | 46 |
| Slika 5.20. Skup objekata stvorenih pomoću Animacijskih čvorova | 47 |
| Slika 5.21. Izgled sučelja nakon izrade modela treće iteracije Tepiha Sierpińskog | 47 |
| Slika 5.22. Model treće iteracije, promatran pod pravim kutom | 48 |
| Slika 5.23. Vizualizacija uz pomoć čvora 3D Viewer; čvor Distribute Matrices | 48 |
| Slika 5.24. Vizualizacija uz pomoć čvora 3D Viewer; redom čvorovi Transform Matrix, Replicate Matrix, Combine Matrix List | 49 |
| 5.1.5. Četvrta iteracija i nadalje | 50 |
| Slika 5.25. Skup objekata stvorenih pomoću Animacijskih čvorova | 50 |
| Slika 5.26. Izgled sučelja nakon izrade modela četvrte iteracije Tepiha Sierpińskog | 50 |
| Slika 5.27. Model četvrte iteracije, promatran pod pravim kutom | 51 |
| 5.2. Rezanje u PrusaSlicer-u | 51 |
| 5.3. 3D printanje pomoću Prusa i3 MK3S 3D printera i naknadna obrada | 51 |
| 6. PRAKTIČNI DIO IZRADE MENGEROVE SPUŽVE I NJEZINOG NEGATIVA POMOĆU 3D PRINTERA | 51 |
| 6.1. 3D modeliranje u Blenderu | 51 |
| 6.1.1. Nulta iteracija | 51 |
| Slika 6.1. Gotov model nulte iteracije Mengerove spužve | 52 |

| | |
|--|-----------|
| 6.1.2. Prva iteracija | 52 |
| Slika 6.2. <i>Izgled sučelja nakon dodavanja novog radnog prostora</i> | <i>53</i> |
| Slika 6.3. <i>Stabla čvorova za prvu iteraciju</i> | <i>53</i> |
| Slika 6.4. <i>Skup objekata, stvorenih pomoću Animacijskih čvorova</i> | <i>54</i> |
| Slika 6.5. <i>Postavke modifikatora Boolean</i> | <i>54</i> |
| Slika 6.6. <i>Izgled sučelja nakon izrade modela prve iteracije Mengerove spužve</i> | <i>55</i> |
| Slika 6.7. <i>Postupak spajanja objekata dobivenih pomoću Animacijskih čvorova</i> | <i>55</i> |
| Slika 6.8. <i>Gotov model negativa prve iteracije Mengerove spužve</i> | <i>56</i> |
| Slika 6.9. <i>Gotov model prve iteracije Mengerove spužve</i> | <i>56</i> |
| Slika 6.10. <i>Rotirane kopije objekta</i> | <i>57</i> |
| 6.1.3. Druga iteracija | 58 |
| Slika 6.11. <i>Prvo stablo čvorova za drugu iteraciju</i> | <i>58</i> |
| Slika 6.12. <i>Drugo stablo čvorova za drugu iteraciju</i> | <i>59</i> |
| Slika 6.13. <i>Posljednje stablo čvorova za drugu iteraciju</i> | <i>59</i> |
| Slika 6.14. <i>Skup objekata, stvorenih pomoću Animacijskih čvorova</i> | <i>59</i> |
| Slika 6.15. <i>Izgled sučelja nakon izrade modela druge iteracije Mengerove spužve</i> | <i>60</i> |
| Slika 6.16. <i>Postupak spajanja objekata dobivenih pomoću Animacijskih čvorova</i> | <i>60</i> |
| Slika 6.17. <i>Gotov model negativa druge iteracije Mengerove spužve</i> | <i>61</i> |
| Slika 6.18. <i>Gotov model druge iteracije Mengerove spužve</i> | <i>61</i> |
| Slika 6.19. <i>Stvoreni objekti pomoću Animacijskih čvorova</i> | <i>62</i> |
| 6.1.4. Treća iteracija | 62 |
| Slika 6.20. <i>Prvi par stabala čvorova za treću iteraciju</i> | <i>63</i> |
| Slika 6.21. <i>Drugi par stabala čvorova za treću iteraciju</i> | <i>63</i> |
| Slika 6.22. <i>Posljednji par stabala čvorova za treću iteraciju</i> | <i>64</i> |
| Slika 6.23. <i>Skup objekata, stvorenih pomoću Animacijskih čvorova</i> | <i>64</i> |
| Slika 6.24. <i>Izgled sučelja nakon izrade modela treće iteracije Mengerove spužve</i> | <i>65</i> |
| 6.1.5. Daljnje iteracije | 67 |
| 6.2. Rezanje u PrusaSlicer-u | 67 |
| 6.3. 3D printanje pomoću Prusa i3 MK3S 3D printera i naknadna obrada | 67 |
| 7. ZAKLJUČAK | 68 |
| 8. LITERATURA | 69 |
| 9. IZVORI SLIKA | 70 |
| 10. PRILOZI | 71 |

1. UVOD

Geometrijski objekti kao što su npr. Cantorov skup, Trokut Sierpińskog i Kochova krivulja poznati su u matematici već više od jednog stoljeća. Svi ti objekti dijelili su neke zajedničke karakteristike koje su matematičarima bile vrlo intrigantne. Jedno je bilo sigurno, a to je da ti objekti nisu pripadali svijetu klasične euklidske geometrije. Uz objekte koje danas nazivamo fraktalima vežu se imena brojnih matematičara. Budući da je povijest fraktala vrlo bogata, u ovom radu navest će se samo dio matematičara koji su pridonijeli načinu na koji ih danas promatramo.

Tek unatrag četrdeset godina zahvaljujući francuskom matematičaru Benoitu Mandelbrotu fraktali su dobili svoj službeni naziv i definiciju. Uz to, Mandelbrot je primijetio nešto što nitko drugi nije, a to je da fraktali nisu samo matematičke zanimljivosti, već geometrija prirode. Pojavila se takozvana fraktalna geometrija, geometrija koja je prekršila sve euklidske zakone svijeta koji je stvorio čovjek. U svojoj knjizi *Fraktalna geometrija prirode* koju je objavio 1982. godine, Mandelbrot je istaknuo mnoge pojave fraktalnih objekata u prirodi. Još jedan ogroman doprinos Benoita Mandelbrota jest što je uz pomoć računala omogućio prikaz fascinantnih primjera fraktala koje nitko prije nije ni mogao zamisliti.



Slika 1.1. Benoit Mandelbrot (1924.-2010.) [33]

Osim fraktala, pozornost javnosti unatrag četrdeset godina privukao je razvitak tehnologije 3D printanja. Donedavno, za mnoge je ta tehnologija graničila sa znanstvenom fantastikom. Bilo je teško pojmiti da bi uređaj koji danas nazivamo 3D printerom mogao uopće postojati, upravo zbog beskrajnih mogućnosti njegove primjene.

Ovaj diplomski rad poveznica je između fraktala i tehnologije 3D printanja.

2. FRAKTALI

2.1. Što su fraktali?

Izraz fraktal osmislio je 1978. godine Benoit Mandelbrot. Potječe od latinske riječi *fractus*, što znači slomljen, razlomljen. [10] Postoji više različitih definicija fraktala, no sa sigurnošću se može reći da fraktali sadrže sljedeće tri karakteristike: samosličnost, fraktalnu dimenziju i oblikovanje iteracijom.

U prirodi je lako pronaći fraktalne strukture, a kao primjer promotrimo sliku 2.1. Može se primijetiti da je struktura paprati vrlo složena i detaljna te da su listovi oblikovani poput malih kopija grana. Zapravo je cijela paprat uglavnom građena od istog osnovnog oblika koji se iznova ponavlja u sve manjim veličinama. Nepravilnost i fragmentacija¹ još neke su karakteristike fraktalnih objekata, koje sprječavaju njihov opis tradicionalnim geometrijskim izrazima.



Slika 2.1. Paprat [14]

Najpreciznija definicija fraktala bila bi ona koju je dao matematičar Benoit Mandelbrot, a prema njemu su fraktali skupovi točaka čija je fraktalna dimenzija veća od topološke dimenzije. [4] Intuitivno, fraktali su složeni objekti koji se mogu podijeliti na beskonačno mnogo

¹ Fragmentacija – raskidanje neke cjeline u dijelove

dijelova. Bez obzira koji dio fraktala se promatra, njegova se složenost ne smanjuje i može se uočiti kako se iste strukture iznova i iznova ponavljaju.

2.2. Svojstva fraktala

2.2.1. Oblikovanje iteracijom

Fraktalna geometrija je iterativno² definirana geometrija - može se opisati u obliku koraka koji uključuju rezultate prethodnih koraka. Kao primjer navedenog, može se promotriti vrlo poznati fraktal Trokut Sierpińskog (slika 2.2.). Na prikazanoj slici može se uočiti da iz jednog trokuta, od kojeg započinje konstrukcija tog fraktala, u svakom novom koraku konstrukcije nastaje sve više manjih trokuta.

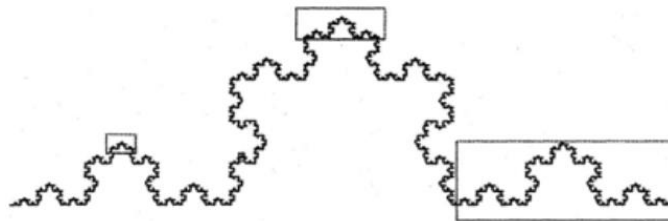


Slika 2.2. Prva četiri koraka konstrukcije Trokuta Sierpińskog [12]

U nastavku ovog diplomskog rada može se pronaći opisan način konstrukcije Trokuta Sierpińskog, nakon čega se može zaključiti da se ovaj fraktal generira³ u beskonačnom broju ponavljanja. Upravo to svojstvo fraktala još se naziva i oblikovanje iteracijom. Može se zaključiti da kad god se promatraju fraktali, neizravno se razmatra koncept beskonačnosti; fraktalna struktura beskonačno se reproducira unutar konačnog prostora na sve kompleksnije strukture.

2.2.2. Samosličnost

Fraktali dijele svojstvo da su sami sebi slični jer su njihovi proizvoljno mali dijelovi slični cjelini (slika 2.3.). Opisano svojstvo naziva se samosličnost. Inače, ne sadrže svi objekti isti stupanj samosličnosti. Kod nekih je ona potpuno savršena, dok je kod nekih ona približna i ne odmah uočljiva. Svojstvo samosličnosti vrlo lako može se uočiti u prirodi. Upravo paprat prikazana na slici 2.1. sadrži svojstvo samosličnosti.



Slika 2.3. Samoslična krivulja [29]

² Iteracija (lat. *Iterare* – ponoviti) – ponavljanje

³ Generirati (lat. *Generare* – stvarati) – stvarati

2.2.3. Fraktalna dimenzija

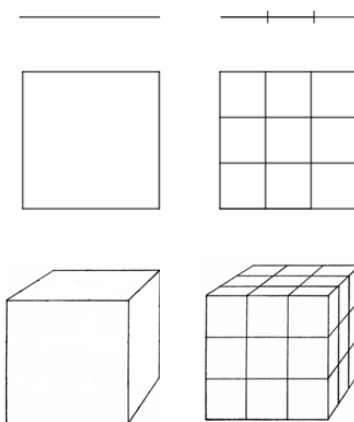
Prije nego što se navede što je to fraktalna dimenzija, potrebno je podsjetiti se one na koju svatko pomisli kada se spomene pojam dimenzije nekog objekta. Topološka dimenzija je dimenzija koja se većinom smatra uobičajenom dimenzijom i ona je uvijek jednaka nekom pozitivnom cijelom broju ili nuli. Označava broj „stupnjeva slobode”, odnosno broj smjerova u kojima se unutar nekog objekta može kretati. Objekti klasične euklidske geometrije imaju topološku dimenziju. Riječ je o nultodimenzionalnoj točki, jednodimenzionalnoj krivulji, dvodimenzionalnoj plohi te trodimenzionalnom tijelu.

Fraktali su jedinstveni po tome što je njihova dimenzija vrlo rijetko cjelobrojna. Fraktalnom dimenzijom se opisuje u kojoj mjeri neki fraktal ispunjava prostor u kojem se nalazi. [12] Fraktalna dimenzija je vrijednost koja ostaje konstantna, bez obzira na mjerilo u kojem se promatra neki fraktal. [11] Kao što je već navedeno, Mandelbrotova definicija fraktala govori da su fraktali skupovi točaka čija je fraktalna dimenzija veća od topološke dimenzije. Fraktalnu dimenziju moguće je odrediti pomoću više različitih metoda, a ovdje će se razmatrati metoda sličnosti. Ova metoda koristi se isključivo na potpuno samosličnim fraktalima. Potpuno samoslični fraktali su fraktali čiji dijelovi izgledaju kao njegove umanjene kopije. Metodu sličnosti moguće je upotrijebiti za određivanje dimenzije i geometrijskih objekata koji nisu fraktali, recimo segmenta linije, kvadrata i kocke.

Promotrimo liniju (slika 2.4.). Ako ju umanjimo na $1/3$ njezine duljine, trebat ćemo $3 = 3^1$ manjih sukladnih linija kako bismo dobili izvornu liniju.

Promotrimo zatim kvadrat (slika 2.4.), koji je također samosličan. Ako umanjimo duljinu stranice tog kvadrata na $1/3$ njegove duljine, dobit ćemo kvadrat čija je površina 9 puta manja. Stoga nam je potrebno $9 = 3^2$ manjih sukladnih kvadrata za popunjavanje izvornog kvadrata.

Zatim promotrimo kocku (slika 2.4.). Ako umanjimo duljinu brida te kocke na $1/3$ njegove duljine, dobit ćemo kocku čiji je volumen 27 puta manji. Stoga nam je potrebno $27 = 3^3$ manjih sukladnih kocki za popunjavanje izvorne kocke.



Slika 2.4. Samosličnost linije, kvadrata i kocke

Pretpostavimo da općenito imamo objekt kojeg je moguće u potpunosti popuniti pomoću manjih kopija sebe, bez praznina ili preklapanja. Označimo s k omjer duljine stranice izvornog objekta i duljine stranice njegove umanjene kopije, a s n broj manjih sukladnih elemenata za popunjavanje izvornog objekta. Kakav je odnos između n i k ? Za jednodimenzionalni segment linije vrijedi $n = k^1$, za dvodimenzionalni kvadrat $n = k^2$, a za trodimenzionalnu kocku $n = k^3$. Općenito vrijedi $n = k^d$, gdje d predstavlja dimenziju promatranog objekta. Ukoliko je promatrani objekt fraktal, d označava njegovu fraktalnu dimenziju. Logaritmiranjem izraza $n = k^d$ dobiva se:

$$d = \frac{\log n}{\log k}$$

U nastavku će se pomoću metode sličnosti odrediti fraktalna dimenzija jednog fraktala, poznatog kao Tepih Sierpińskog. Na slici 2.5. je prikazana njegova prva iteracija:



Slika 2.5. Nulta i prva iteracija Tepiha Sierpińskog [28]

Vidljivo je da prva iteracija Tepiha Sierpińskog sadrži osam svojih umanjenih kopija, stoga zaključujemo da je $n = 8$. Također, primjećujemo da je duljina stranice svake umanjene kopije trostruko manja od duljine stranice izvornog kvadrata, stoga je $k = 3$. Zaključujemo da fraktalna dimenzija Tepiha Sierpińskog mora biti jednaka broju d za koji vrijedi $3^d = 8$. Logaritmiranjem dobivamo

$$d = \frac{\log 8}{\log 3} \approx 1.893.$$

2.3. Poznati fraktali i njihova povijest

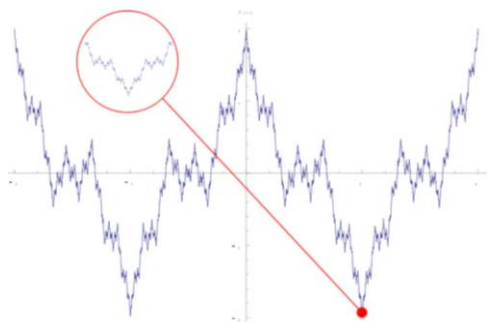
Povijest fraktala seže u 17. stoljeće, kada je njemački matematičar Gottfried Wilhelm Leibniz (1646.-1716.) uveo pojam samosličnosti. No, svojstvo samosličnosti Leibniz je uočio tek na dužini. Gotovo dvjesto godina proteklo je prije nego što je njemački matematičar Karl Weierstrass (1815.-1897.) 1872. godine predstavio funkciju čiji bi se graf (slika 2.6.) prema današnjim standardima smatrao fraktalom. Ta funkcija danas se naziva Weierstrassovom funkcijom.

2.3.1. Weierstrassova funkcija

Weierstrassova funkcija je funkcija

$$f(x) = \sum_{n=0}^{\infty} a^n \cos(b^n \pi x),$$

pri čemu je $0 < a < 1$, b pozitivan neparan cijeli broj i $ab > 1 + \frac{3}{2}\pi$. Weierstrassovu funkciju njegovi suvremenici nazivali su čudovišnom funkcijom, budući da je to bio prvi objavljeni primjer funkcije s neintuitivnim svojstvom da je neprekidna u svakoj točki svoje domene, ali nigdje diferencijabilna.



Slika 2.6. Graf Weierstrassove funkcije na intervalu $[-2, 2]$ [30]

Tijekom 19. i 20. stoljeća brojni drugi matematičari bavili su se istraživanjem i prikazom fraktalnih oblika. Međutim, nisu bili u potpunosti svjesni njihove složenosti. U nastavku će se predstaviti neke od tadašnjih tvorevina, koje se danas jednom riječju nazivaju fraktalima.

2.3.2. Cantorov skup

Skup točaka, danas poznat pod imenom Cantorov skup otkrio je 1874. godine irski matematičar Henry John Stephen Smith (1826.-1883.), a 1883. godine javnosti ga je predstavio utemeljitelj teorije skupova, njemački matematičar Georg Cantor (1854.-1918.).

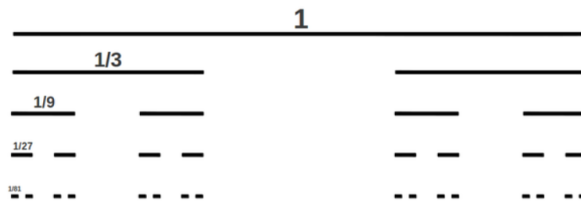
Cantorov skup ima vrlo jednostavnu konstrukciju i zanimljiv je u mnogim granama moderne matematike. Konstrukcija (slika 2.7.) započinje od jediničnog segmenta $[0, 1]$. Taj segment zatim se podijeli na tri jednaka podsegmenta te se ukloni srednji podinterval $(\frac{1}{3}, \frac{2}{3})$. Time ostaju dva segmenta $[0, \frac{1}{3}]$ i $[\frac{2}{3}, 1]$, pri čemu su oba duljine $\frac{1}{3}$. U idućem koraku se svaki od dva preostala zatvorena intervala dijeli na tri jednaka podintervala. Dobivaju se sljedeći segmenti: $[0, \frac{1}{9}]$, $[\frac{2}{9}, \frac{1}{3}]$, $[\frac{2}{3}, \frac{7}{9}]$ i $[\frac{8}{9}, 1]$. Pri tome je svaki duljine $\frac{1}{9}$.

Postupak se ponavlja. Općenito, nakon n koraka postoji 2^n segmenata, od kojih svaki ima duljinu $\frac{1}{3^n}$. Lako je zaključiti da je duljina Cantorovog skupa jednaka nuli. Naime, duljine uklonjenih segmenata su redom: $\frac{1}{3}, \frac{2}{3^2}, \frac{2^2}{3^3}, \dots, \frac{2^n}{3^{n+1}}$. Njihov zbroj jednak je duljini cijelog segmenta:

$$\frac{1}{3} \sum_{n=0}^{\infty} \left(\frac{2}{3}\right)^n = \frac{1}{3} \frac{1}{1 - \frac{2}{3}} = 1$$

Postavlja se pitanje ostavlja li postupak uklanjanja išta od jediničnog segmenta $[0, 1]$? Na primjer, točke kojima su pridruženi brojevi 0 i 1 očito nikada nisu uklonjene ovim postupkom.

Daljnijim razmatranjem konstrukcije Cantorovog skupa može se zaključiti da on sadrži beskonačno mnogo točaka.

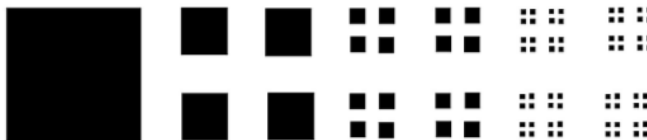


Slika 2.7. Prva četiri koraka konstrukcije Cantorovog skupa [12]

Topološka dimenzija Cantorovog skupa jednaka je nuli, dok je njegova fraktalna dimenzija približno jednaka

$$d = \frac{\log 2}{\log 3} \approx 0.6309.$$

Ravninski analogon Cantorovog skupa jest **Cantorova prašina**. Konstrukcija Cantorove prašine (slika 2.8.) započinje od kvadrata. Taj kvadrat se zatim podijeli se na 9 sukladnih kvadrata te se uklanjaju svi oni kvadrati koji ne sadrže vrh prvotnog. Ostaju 4 sukladna kvadrata. U idućem koraku postupak se ponavlja, odnosno vrši se podjela svakog od ta 4 kvadrata. Slijedi ponavljanje opisanog postupka beskonačno mnogo puta nad preostalim kvadratima.

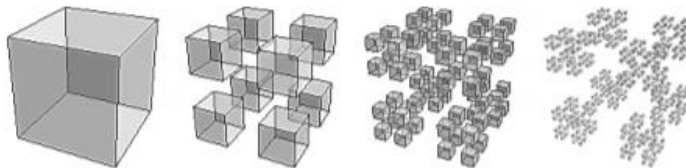


Slika 2.8. Prva tri koraka konstrukcije Cantorove prašine [12]

Topološka dimenzija Cantorove prašine jednaka je nuli, dok je fraktalna dimenzija približno jednaka

$$d = \frac{\log 4}{\log 3} \approx 1.2619.$$

Prostorni analogon Cantorovog skupa jest **Cantorov oblak**, čija konstrukcija (slika 2.9.) započinje od kocke. Ta kocka zatim se podijeli na 27 kocaka međusobno jednakih volumena. Kao i kod Cantorove prašine, uklanjaju se oni objekti koji ne sadrže vrh prvotnog objekta. Nakon provedenog postupka, ostaje 8 kocaka međusobno jednakih volumena. U idućem koraku vrši se podjela svake od tih 8 kocaka. Opisan postupak ponavlja se beskonačno mnogo puta nad preostalim kockama.



Slika 2.9. Prva tri koraka konstrukcije Cantorovog oblaka [12]

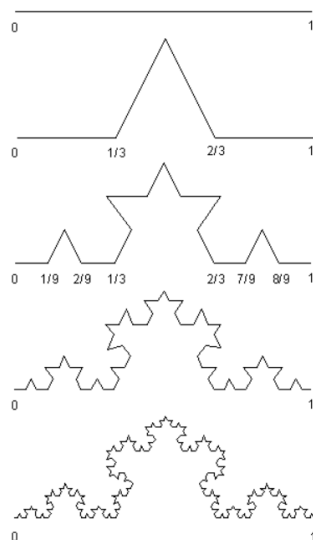
Fraktalna dimenzija Cantorovog oblaka približno je jednaka

$$d = \frac{\log 8}{\log 3} \approx 1.8928.$$

2.3.3. Kochova krivulja i pahuljica

Godine 1904., više od dva desetljeća nakon što se Cantorov skup prvi put pojavio, švedski matematičar Helge von Koch (1870.-1924.) predstavio je fraktal koji se danas naziva Kochova krivulja. Njezina konstrukcija (slika 2.10.) je iterativna kao i kod Cantorovog skupa. Proviđi se na sljedeći način:

Započinje se od dužine, koju se podijeli na tri sukladna dijela. Nad srednjim dijelom konstruira se jednakostranični trokut, nakon čega se srednji dio dužine uklanja. Time se dobivaju četiri nove sukladne dužine. U idućem koraku konstrukcije slijedi provođenje prethodno opisanog postupka nad svakom od četiriju novonastalih dužina. Isti postupak ponavlja se beskonačno mnogo puta, čime nastaje Kochova krivulja.



Slika 2.10. Prva četiri koraka konstrukcije Kochove krivulje [31]

Može se primijetiti da se nakon svakog koraka konstrukcije duljina Kochove krivulje poveća za faktor $\frac{4}{3}$. Dakle, nakon n -tog iterativnog koraka konstrukcije Kochove krivulje njezina duljina će biti

$$l_n = \left(\frac{4}{3}\right)^n.$$

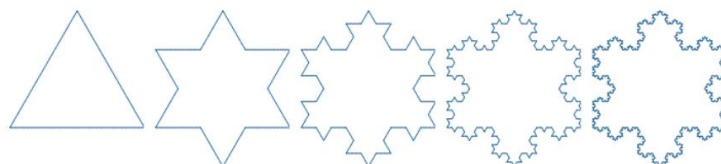
Konačno, može se zaključiti da Kochova krivulja ima beskonačnu duljinu jer vrijedi

$$l = \lim_{n \rightarrow \infty} l_n = \lim_{n \rightarrow \infty} \left(\frac{4}{3}\right)^n = \infty.$$

Topološka dimenzija Kochove krivulje jednaka je jedan, dok je njezina fraktalna dimenzija približno jednaka

$$d = \frac{\log 4}{\log 3} \approx 1.2619.$$

Slijedi primjer krivulje koja je vrlo slična Kochovoj krivulji. Ovaj put riječ je o zatvorenoj krivulji koja se naziva **Kochova pahuljica**. Konstrukcija Kochove pahuljice (slika 2.11.) započinje od triju dužina koje je potrebno postaviti tako da čine jedan jednakostraničan trokut. Postupak kojim je konstruirana Kochova krivulja provodi se na svakoj toj dužini, čime je dobiven prvi korak konstrukcije Kochove pahuljice. Isti postupak ponavlja se nad novonastalim dužinama. Kochova pahuljica dobiva se nakon beskonačno mnogo provedenih koraka konstrukcije.



Slika 2.11. Prva četiri koraka konstrukcije Kochove pahuljice [32]

Topološka dimenzija Kochove pahuljice jednaka je jedan, dok je njezina fraktalna dimenzija približno jednaka

$$d = \frac{\log 4}{\log 3} \approx 1.2619.$$

Duljina Kochove pahuljice je beskonačna. S druge strane, zanimljivo je da ta zatvorena fraktalna krivulja zatvara konačnu površinu.

2.3.4. Trokut Sierpińskog

Trokut Sierpińskog otkrio je 1915. godine poljski matematičar Waclaw Sierpiński (1882.-1969.). Ovaj fraktal konstruira (slika 2.12.) se na sljedeći način. Zapčinje se od jednakostraničnog trokuta. Svakoj stranici tog trokuta potrebno je odrediti njezino polovište. Uklanjanjem trokuta čiji su vrhovi ujedno polovišta stranica početnog trokuta dobit će se prva iteracija ovog fraktala. Opisan postupak ponavlja se beskonačno mnogo puta nad preostalim trokutima.



Slika 2.12. Prva četiri koraka konstrukcije Trokuta Sierpińskog [12]

Topološka dimenzija Trokuta Sierpińskog jednaka je jedan, dok je njegova fraktalna dimenzija približno jednaka

$$d = \frac{\log 3}{\log 2} \approx 1.5849.$$

U nastavku će se razmotriti opseg i površina Trokuta Sierpińskog. Neka je duljina stranice njegove nulte iteracije jednaka a .

Opseg tog trokuta jednak je $3a$, dok je njegova površina jednaka $\frac{a^2\sqrt{3}}{4}$. Promotri li se prva iteracija, vidljivo je da se ona sastoji od 3 jednakostranična trokuta, pri čemu svaki od njih ima duljinu stranice $\frac{a}{2}$. Opseg O i površina P prve iteracije jednaki su

$$O_1 = 3 \cdot \left(3 \cdot \frac{a}{2}\right) = \frac{3^2 a}{2}$$

$$P_1 = 3 \cdot \frac{\left(\frac{a}{2}\right)^2 \sqrt{3}}{4} = 3 \cdot \frac{\frac{a^2}{4} \sqrt{3}}{4} = \frac{3}{4} \cdot \frac{a^2 \sqrt{3}}{4}$$

Druga iteracija sastoji se od 9 jednakostraničnih trokuta, pri čemu svaki od njih ima duljinu stranice $\frac{a}{4}$. Izračunajmo njihov opseg i površinu:

$$O_2 = 9 \cdot \left(3 \cdot \frac{a}{4}\right) = \frac{3^3 a}{2^2}$$

$$P_2 = 9 \cdot \frac{\left(\frac{a}{4}\right)^2 \sqrt{3}}{4} = \frac{3^2}{4^2} \cdot \frac{a^2 \sqrt{3}}{4}$$

Općenito vrijedi:

$$O_n = \left(\frac{3}{2}\right)^n \cdot 3a$$

$$P_n = \left(\frac{3}{4}\right)^n \cdot \frac{a^2 \sqrt{3}}{4}$$

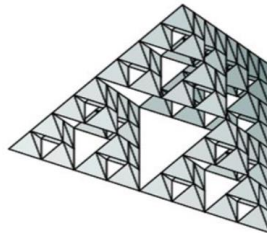
Kada n teži u beskonačnost, tada je

$$O = \lim_{n \rightarrow \infty} O_n = \lim_{n \rightarrow \infty} \left(\frac{3}{2}\right)^n \cdot 3a = 3a \lim_{n \rightarrow \infty} \left(\frac{3}{2}\right)^n = \infty$$

$$P = \lim_{n \rightarrow \infty} P_n = \lim_{n \rightarrow \infty} \left(\frac{3}{4}\right)^n \cdot \frac{a^2 \sqrt{3}}{4} = \frac{a^2 \sqrt{3}}{4} \lim_{n \rightarrow \infty} \left(\frac{3}{4}\right)^n = 0$$

Iz prethodnog računa je vidljivo kako broj iteracija Trokuta Sierpińskog teži u beskonačnost, njegov opseg teži u beskonačnost a površina nuli. Stoga se može reći da je opseg ovog fraktala beskonačan, a površina mu je jednaka nuli.

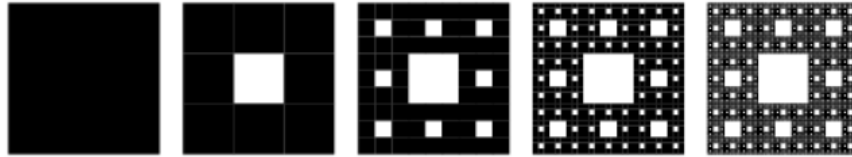
Prostorni analogon Trokuta Sierpińskog jest **Tetraedar Sierpińskog** (slika 2.13.).



Slika 2.13. Tetraedar Sierpińskog [10]

2.3.5. Tepih Sierpińskog

Tepih Sierpińskog još jedan je primjer fraktala kojeg je otkrio Waclaw Sierpiński. Konstrukcija (slika 2.14.) ovog fraktala započinje od kvadrata, kojeg se podijeli na 9 kvadrata međusobno jednakih površina. Uklanjanjem središnjeg kvadrata dobiva se prva iteracija. U idućem koraku preostalih 8 kvadrata dijeli se na 9 kvadrata jednakih površina te se uklanja središnji kvadrat iz svake grupe od 9 kvadrata. Beskonačnim ponavljanjem opisanog postupka dobiva se Tepih Sierpińskog.



Slika 2.14. Prva četiri koraka konstrukcije Tepiha Sierpińskog [28]

Topološka dimenzija Tepiha Sierpińskog jednaka je jedan, dok je njegova fraktalna dimenzija približno jednaka

$$d = \frac{\log 8}{\log 3} \approx 1.8928.$$

Ako se promotri slika 2.14., moglo bi se zaključiti da Tepih Sierpińskog nakon beskonačno mnogo koraka ima beskonačan opseg i površinu koja je jednaka nuli. U nastavku će se dokazati tvrdnja o površini. Neka je duljina stranice nulte iteracije jednaka a . Površina tog kvadrata jednaka je

$$P_0 = a^2$$

Površina prve iteracije dobije se nakon što se od površine kvadrata prethodne iteracije oduzme površina kvadrata sa stranicom duljine $\frac{a}{3}$:

$$P_1 = a^2 - \frac{a^2}{9} = \frac{8a^2}{9}$$

Površina druge iteracije dobije se nakon što se od površine P_1 oduzme površina 8 kvadrata sa stranicom duljine $\frac{a}{9}$:

$$P_2 = \frac{8a^2}{9} - \frac{8a^2}{81} = \frac{64a^2}{81} = \frac{8^2a^2}{9^2}$$

Površina treće iteracije dobije se nakon što se od površine P_2 oduzme površina 64 kvadrata sa stranicom duljine $\frac{a}{81}$:

$$P_3 = \frac{8^2a^2}{9^2} - \frac{8^2a^2}{9^3} = \frac{(9 \cdot 64 - 64)a^2}{9^3} = \frac{8^3a^2}{9^3}$$

Općenito vrijedi:

$$P_n = \frac{8^n a^2}{9^n} = \left(\frac{8}{9}\right)^n a^2$$

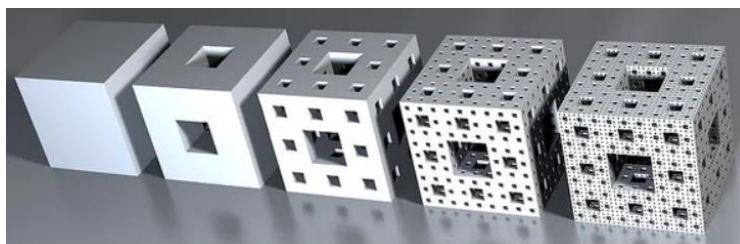
Niz površina P_n je geometrijski niz s kvocijentom $\frac{8}{9} < 1$, a to znači da je konvergentan i ima sljedeći limes:

$$P = \lim_{n \rightarrow \infty} P_n = \lim_{n \rightarrow \infty} \left(\frac{8}{9}\right)^n a^2 = a^2 \lim_{n \rightarrow \infty} \left(\frac{8}{9}\right)^n = 0$$

2.3.6. Mengerova spužva

Prostorni analogon Tepiha Sierpińskog jest Mengerova spužva. Na slici je moguće primijetiti da je svaka strana Mengerove spužve upravo Tepih Sierpińskog. Ovaj fraktal dobio je ime po austrijskom matematičaru Karlu Mengeru (1902.-1985.), koji ga je opisao 1926. godine.

Konstrukcija Mengerove spužve (slika 2.15.) započinje od kocke, koja se podijeli na 27 kocaka međusobno jednakih volumena. Prva iteracija nastaje uklanjanjem središnje kocke te 6 kocaka koje se nalaze u središtima strana prvotne kocke. Ovaj postupak ponavlja se beskonačno mnogo puta nad preostalim kockama.



Slika 2.15. Prva četiri koraka konstrukcije Mengerove spužve [27]

Prva iteracija Mengerove spužve sadrži 20 svojih umanjenih kopija, dok je duljina brida svake umanjene kopije trostruko manja od duljine brida izvorne kocke. Stoga je fraktalna dimenzija Mengerove spužve približno jednaka

$$d = \frac{\log 20}{\log 3} \approx 2.7268.$$

Nakon što je određena fraktalna dimenzija Mengerove spužve, treba razmotiti kolika bi bila njezina topološka dimenzija. Promotri li se slika 2.15., vidljivo je da svaka nova iteracija Mengerove spužve ima sve manji volumen. Broj rupa se povećava iz iteraciju u iteraciju. Iako je možda iznenađujuće, topološka dimenzija Mengerove spužve jednaka je jedan. Primijetimo da vrijedi da je fraktalna dimenzija veća od topološke dimenzije, o čemu govori Mandelbrotova definicija fraktala.

U nastavku će se razmotriti oplošje i volumen Mengerove spužve. Neka je duljina brida početne kocke jednaka a . Oplošje te kocke iznosi:

$$O_0 = 6a^2$$

Ukoliko se promotri slika 2.15., vidljivo je da se svih 6 strana prve iteracije Mengerove spužve mogu podijeliti na 8 kvadrata čija je duljina stranice $\frac{a}{3}$. Zbroj njihovih površina jednak je $6 \cdot 8 \cdot \left(\frac{a}{3}\right)^2$. Preostaje još i pridodati površinu kvadrata koji okružuju svih 6 rupa, nastalih uklanjanjem kocaka duljine brida $\frac{a}{3}$ u prvom koraku konstrukcije. Budući da svaku rupu okružuju 4 kvadrata čija je duljina stranice $\frac{a}{3}$, ukupna površina tih kvadrata jednaka je $6 \cdot 4 \cdot \left(\frac{a}{3}\right)^2$. Stoga oplošje prve iteracije iznosi:

$$O_1 = 6 \cdot 8 \cdot \left(\frac{a}{3}\right)^2 + 6 \cdot 4 \cdot \left(\frac{a}{3}\right)^2 = 72 \cdot \frac{a^2}{9}$$

Druga iteracija sadrži $8^2 = 64$ kvadrata duljine stranice $\frac{a}{9}$. Njihova ukupna površina jednaka je $6 \cdot 8^2 \cdot \left(\frac{a}{9}\right)^2$. Također, druga iteracija sadrži 20 kocaka međusobno jednakih volumena. Svaka od njih sadrži 6 rupa koje su nastale uklanjanjem kocaka duljine brida $\frac{a}{9}$ u drugom koraku konstrukcije. Stoga je svaka od tih rupa omeđena s 4 kvadrata čija je duljina stranice $\frac{a}{9}$. Ukupna površina svih tih kvadrata jednaka je $20 \cdot 6 \cdot 4 \cdot \left(\frac{a}{9}\right)^2$. Preostaje još i pridodati površinu kvadrata koji okružuju rupe nastale tijekom uklanjanja kocaka u prvom koraku konstrukcije. Svaku od tih 6 rupa okružuju 8 kvadrata čija je duljina stranice $\frac{a}{9}$. Zbroj njihovih površina jednak je $6 \cdot 4 \cdot 8 \cdot \left(\frac{a}{9}\right)^2$. Stoga oplošje druge iteracije iznosi:

$$O_2 = 6 \cdot 8^2 \cdot \left(\frac{a}{9}\right)^2 + 20 \cdot 6 \cdot 4 \cdot \left(\frac{a}{9}\right)^2 + 6 \cdot 4 \cdot 8 \cdot \left(\frac{a}{9}\right)^2 = 1056 \cdot \frac{a^2}{81}$$

Općenito vrijedi

$$O_n = 4 \cdot \left(\frac{8}{9}\right)^n \cdot a^2 + 2 \cdot \left(\frac{20}{9}\right)^n \cdot a^2$$

Kada n teži u beskonačnost, tada je

$$O = \lim_{n \rightarrow \infty} O_n = \lim_{n \rightarrow \infty} \left(4 \cdot \left(\frac{8}{9}\right)^n \cdot a^2 + 2 \cdot \left(\frac{20}{9}\right)^n \cdot a^2 \right) = 0 + \infty = \infty$$

Može se zaključiti da Mengerova spužva nakon beskonačno mnogo koraka ima beskonačno oplošje. Preostaje izračunati volumen Mengerove spužve.

Volumen nulte iteracije iznosi

$$V_0 = a^3$$

Početna kocka dijeli se na 27 kocaka čija je duljina bridova jednaka $\frac{a}{3}$. Prva iteracija nastaje uklanjanjem 7 kocaka, stoga volumen prve iteracije iznosi

$$V_1 = a^3 - 7 \cdot \left(\frac{a}{3}\right)^3 = a^3 - \frac{7a^3}{27} = \frac{20a^3}{27}$$

U idućem koraku svaka kocka duljine brida $\frac{a}{3}$ dijeli se na 27 kocaka duljine bridova $\frac{a}{9}$. Druga iteracija nastaje uklanjanjem 7 kocaka iz svake od kocke duljine brida $\frac{a}{3}$, stoga volumen druge iteracije iznosi

$$V_2 = \frac{20a^3}{27} - 7 \cdot 20 \cdot \left(\frac{a}{9}\right)^3 = \frac{20a^3}{27} - \frac{140a^3}{729} = \frac{400a^3}{729} = \frac{20^2 a^3}{27^2}$$

Općenito vrijedi

$$V_n = \left(\frac{20}{27}\right)^n a^3$$

Kada n teži u beskonačnost, tada je

$$V = \lim_{n \rightarrow \infty} V_n = a^3 \lim_{n \rightarrow \infty} \left(\frac{20}{27}\right)^n = 0$$

Zanimljivo je vidjeti izgled komplementa prvih triju iteracija Mengerove spužve (slika 2.16.). Prikazani objekti često se nazivaju i negativima Mengerove spužve. Nakon beskonačno mnogo ponavljanja Mengerova spužva ima volumen jednak nuli, dok njezin komplement (negativ) ima volumen jednak a^3 .



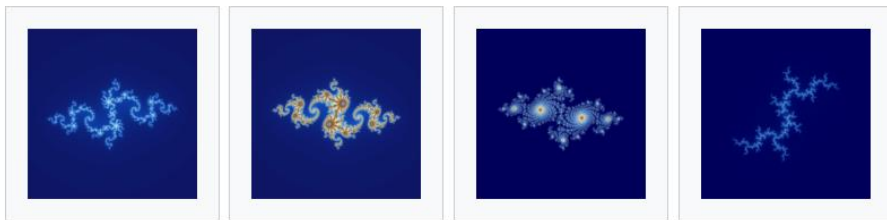
Slika 2.16. Komplementi prvih triju iteracija Mengerove spužve

2.3.7. Julijini skupovi

Julijini skupovi dobili su ime po francuskom matematičaru Gastonu Juliji. Inače, Gaston Julia (1893.-1978.) bio je francuski matematičar koji je zajedno sa svojim sunarodnjakom i kolegom Pierreom Fatouom (1878.-1929.) proučavao skupove koji se mogu stvarati iterativnim postupkom u kompleksnoj ravnini. Budući da u to doba nisu postojala računala, te skupove nije bilo moguće vizualno predložiti. Nakon što je Julijin rad bio objavljen 1918. godine u znanstvenom časopisu *Mémoire sur l'iteration des fonctions rationnelles*, Julia je stekao veliki ugled među svojim kolegama. No vrlo brzo njegov rad bio je zaboravljen. Sedamdesetih godina prošlog stoljeća rad Gastona Julije oživio je i popularizirao matematičar Benoit Mandelbrot. Inspiriran Julijinim radom, Mandelbrot je kao zaposlenik IBM-a uz pomoć računalne grafike uspio prikazati prve slike Julijinih skupova. Julijini skupovi određeni su rekurzivnom funkcijom

$$f(z_n) = z_{n+1} = z_n^2 + c, n \in \mathbb{N}_0$$

pri čemu je broj $c = a + bi$ fiksni kompleksan broj te $z_n = x_n + y_n i$ proizvoljan kompleksan broj.



Slika 2.17. Četiri primjera Julijinih skupova [26]

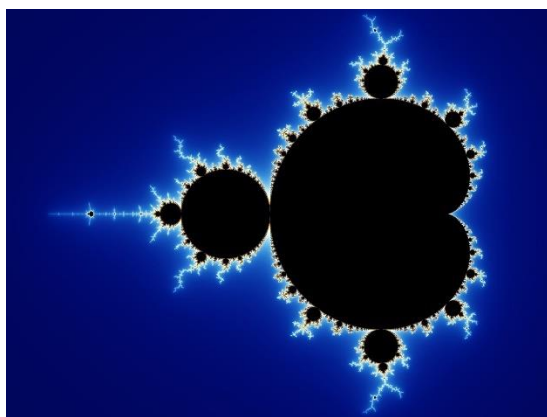
2.3.8. Mandelbrotov skup

Očito je po kojem matematičaru je ovaj fraktal dobio ime. Mandelbrotov skup određen je rekursivnom funkcijom

$$f(z_n) = z_{n+1} = z_n^2 + c, n \in \mathbb{N}_0$$

pri čemu je $c = a + bi$ proizvoljan kompleksan broj te $z_0 = 0 + 0i$. Prilikom proučavanja Julijinih skupova Mandelbrot je uvidio da će, ukoliko u navedenu rekursivnu funkciju uvrsti različite vrijednosti broja c , dobiti različite Julijine skupove. Dobivao je povezane i nepovezane Julijine skupove, pri čemu povezan Julijin skup upravo predstavlja jedan Mandelbrotov skup. Ako $|f(z_n)|$ za neki proizvoljan broj $c \in \mathbb{C}$ teži u beskonačnost, kaže se da točka (a, b) koja je pridružena broju $c \in \mathbb{C}$ ne pripada Mandelbrotovom skupu. Ukoliko $|f(z_n)|$ ne teži u beskonačnost, točka (a, b) pridružena broju $c \in \mathbb{C}$ pripada Mandelbrotovom skupu.

Prethodno je navedeno da je upravo prvu računalno generiranu sliku Mandelbrotovog skupa stvorio Benoit Mandelbrot. Kada se promatra slika Mandelbrotovog skupa na zaslonu računala, zapravo se promatra skup piksela u boji. Svaki piksel predstavlja određenu točku u kompleksnoj ravnini. Ako točka (a, b) pripada Mandelbrotovom skupu, točka se boji crnom bojom, dok se u suprotnom boji nekom drugom bojom. Intenzitet boje svake točke koja ne pripada Mandelbrotovom skupu ovisi o broju ponavljanja nakon kojeg se utvrdilo da ona ne pripada tom skupu. Slika 2.18. prikazuje jedan Mandelbrotov skup.



Slika 2.18. Mandelbrotov skup [12]

Rubna krivulja Mandelbrotovog skupa jest kardioda. Slike Mandelbrotovog skupa pokazuju sve finije detalje pri sve većim uvećanjima, a uvećanja je moguće ponoviti beskonačno

mnogo puta. Stoga se može reći da je glavna karakteristika Mandelbrotovog skupa njegova beskrajna složenost i upravo zbog toga je toliko zanimljiv i fascinant.

2.4. Podjela fraktala

Fraktale je moguće podijeliti u odnosu na stupanj samosličnosti i način njihova nastanka.

2.4.1. Podjela fraktala prema stupnju samosličnosti

Potpuno samoslični fraktali su fraktali čiji dijelovi izgledaju kao njihove umanjene kopije. Primjeri takvih fraktala su Trokut Sierpińskog i Tepih Sierpińskog.

Kvazi samoslični fraktali su fraktali čiji dijelovi nisu u potpunosti jednaki cijelom fraktalu, već se pojavljuju u iskrivljenom obliku. Najpoznatiji takvi fraktali su Mandelbrotov skup i Julijini skupovi.

Statistički samoslični fraktali su fraktali koji ne sadrže vlastite umanjene kopije te imaju najmanji stupanj samosličnosti. Unatoč najmanjem stupnju samosličnosti, ovoj vrsti fraktala i dalje se može odrediti fraktalna dimenzija. Statistička samosličnost prisutna je u prirodi, na primjer može se uočiti na drvetu, školjci, pahuljici.

2.4.2. Podjela fraktala prema načinu njihova nastanka

Iterativni fraktali su fraktali koji nastaju kopiranjem te translacijom i/ili rotiranjem kopije. Primjeri iterativnih fraktala su Trokut Sierpińskog i Tepih Sierpińskog.

Rekurzivni fraktali su fraktali koji su određeni nekom rekurzivnom funkcijom. Primjer rekurzivnog fraktala je Mandelbrotov skup. Pomoću rekurzivne funkcije određuje se pripada li neka točka kompleksne ravnine tom skupu ili ne.

Slučajni fraktali su fraktali koji se mogu uočiti u prirodi. Riječ je o fraktalima s najmanjim stupnjem samosličnosti.

3. 3D PRINTANJE

3.1. Što je 3D printanje?

3D printanje (engl. *3D printing*) ili trodimenzionalni ispis je postupak aditivne proizvodnje kojim se stvara neki fizički objekt. Uređaj kojim se vrši 3D printanje naziva se 3D printer. Prilikom aditivne proizvodnje, objekt se stvara polaganjem uzastopnih slojeva materijala koji čine konačan proizvod. 3D ispis uključuje sve proizvodne tehnologije koje se temelje na automatiziranom i računalno kontroliranom stvaranju trodimenzionalnih objekata, od jednog ili više različitih materijala, putem aditivnog procesa koji se temelji na skrućivanju.



Slika 3.1. *Original Prusa i3 MK3S 3D printer* [6]

3.2. Primjena 3D printanja

Aditivna proizvodnja danas je prisutna u automobilskoj, zrakoplovnoj, vojnoj, inženjerskoj, građevinskoj, stomatološkoj i medicinskoj industriji. Također, sve češće se koristi i u mnogim drugim područjima kao što su obrazovanje, arhitektura, biotehnologija, dizajn itd. U obrazovnom sektoru ona nudi izvrsnu priliku za obogaćivanje iskustva učenja učenicima i studentima, pružajući praktičan pristup učenju. Na primjer, 3D ispis se uvelike koristi u medicinskim školama i fakultetima za stvaranje opipljivih 3D modela kao vizualnih 3D tiskanih pomagala za učenje.

3.3. Tehnologije 3D printanja

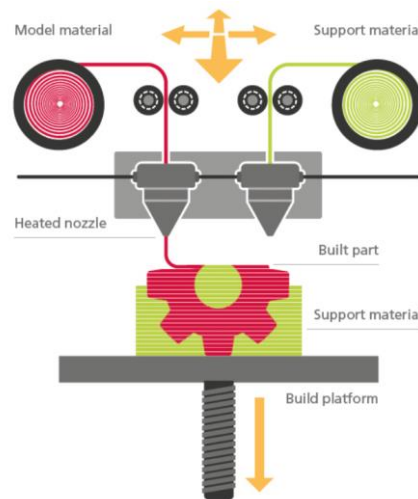
Općenito, postoje različite tehnologije 3D printanja te svaka zahtijeva određenu vrstu materijala za 3D ispis. No, svim tim tehnologijama zajedničko je da stvaraju trodimenzionalni objekt gradeći ga sloj po sloj, sve dok se ne dovrši cijeli objekt. Treba naglasiti da svaka ima svoje prednosti i nedostatke u odnosu na ostale. Riječ je o mnogim tehnologijama, a neke od njih su stereolitografija (SL/SLA), selektivno lasersko sinteriranje (SLS), tehnologija taložnog očvršćivanja (FDM), laminarna objektna proizvodnja (LOM), direktno lasersko sinteriranje metala (DMLS) itd. Najpoznatije su FDM, (SL/SLA) i SLS tehnologije. U ovom radu pobliže će se opisati jedino FDM tehnologija, budući da će se ona koristiti prilikom izrade fraktalnih objekata.

3.3.1. FDM tehnologija

FDM tehnologija najčešće je korištena tehnologija 3D ispisa i često je prva tehnologija kojoj su korisnici izloženi. Puni naziv ove tehnologije glasi „fused deposition modeling”, što u prijevodu na hrvatski jezik znači „tehnologija taložnog očvršćivanja“. Prilikom korištenja FDM tehnologije, objekt se gradi selektivnim odlaganjem rastopljenog materijala sloj po sloj. Debljina pojedinog sloja varira od 0.05 do 0.3 mm ukoliko se koristi standardna mlaznica debljine 0.4mm. [6] Tijekom korištenja FDM tehnologije upotrebljava se plastika koja se naziva filament. Filament je namotan

na kolut te nakon što mlaznica (engl. *Nozzle*) dosegne željenu temperaturu, nit se dovodi u glavu pisača, gdje se topi.

Otopljeni materijal izlazi iz mlaznice u obliku tankih niti i taloži se sloj po sloj, po unaprijed zadanim lokacijama. Nakon što je izašao iz mlaznice, materijal se hladi i skrućuje. Katkad se hlađenje materijala ubrzava upotrebom ventilatora za hlađenje koji je pričvršćen na glavu pisača. Kada je sloj završen, platforma za izgradnju (engl. *Build platform*) pomiče se prema dolje (kod nekih printera glava za istiskivanje pomiče se prema gore) i nanosi se novi sloj. Postupak nanošenja ponavlja se sve dok se ne dobije cjelokupan objekt. Tiskani objekt mehanički se odvaja od platforme, dok se potporni materijal (engl. *Support material*) ručno odvaja od objekta ili ispire odgovarajućim otapalom.



Slika 3.2. Ilustracija 3D printanja pomoću FDM printera s dvije mlaznice [19]

3.4. Povijest 3D printanja

3.4.1. Temelji

Najstarije tehnologije 3D ispisa prvi put su se pojavile krajem 1980-ih, a zvale su se tehnologije brzog prototipiranja (eng. *Rapid prototyping*). One su izvorno zamišljene kao brza i isplativija metoda za razvoj proizvoda u industriji. Prvu patentnu prijavu za tu tehnologiju podnio je dr. Kodama u Japanu 1981. godine. Njegovo istraživanje uključivalo je slojeve materijala koji tvore 3D objekt. Međutim, Kodamina patentna prijava nije napredovala do potpunog patenta jer autor nije uspio podnijeti potpunu prijavu prije isteka roka od godinu dana. Unatoč tome, objavljeni rad potvrđuje njegov povijesni doprinos na tom polju.

Nekoliko godina kasnije, troje francuskih istraživača Olivier de Witteu, Jean-Claude Andréu i Alain le Méhautéu podnijeli su patent za stereolitografski postupak (SLA). Oni su nastojali

stvoriti sustav koji pomoću lasera stvrdnjava tekuće monomere⁴ u krute tvari. Slično kao i Kodama, nisu mogli podnijeti patent za ovu tehnologiju. Godine 1986. Charles Hull podnio je prvi patent za stereolitografske uređaje. Hull je razvio sustav za stvaranje 3D objekata pomoću fotoosjetljive smole i UV lampe. Godine 1988. je suosnovao 3D Systems Corporation - jednu od najvećih i najplodnijih tvrtki koje danas djeluju u sektoru 3D ispisa. Ta tvrtka predstavila je prvi komercijalni SLA 3D pisač 1987. godine.

Stereolitografija je ipak imala konkurenciju u prostoru 3D ispisa, sa suparničkim tehnologijama u razvoju. Na primjer, 1988. godine Carl Deckard na Sveučilištu u Teksasu podnio je patent za tehnologiju selektivnog laserskog sinteriranja (SLS). Umjesto UV svjetla, SLS je laserom pratio i učvršćivao slojeve polimernih⁵ prahova. Scott Crump, suosnivač tvrtke Stratasys, 1989. godine podnio je patent za FDM tehnologiju – danas najpoznatiju tehnologiju 3D ispisa.

3.4.2. Komercijalizacija

Tvrtka Stratasys je 1991. izdala svoj prvi FDM 3D pisač. FDM dijelovi bili su jači i kemijski otporniji, ali SLA dijelovi mogli su se stvoriti brže i točnije. Sljedeće godine 1992., DTM Inc je predstavio svoj prvi SLS 3D pisač.

Tijekom 1990-ih i 2000-ih godina aditivna je proizvodnja stekla svoje uporište kao atraktivna alternativa tradicionalnim proizvodnim procesima, pa je tako brzo počela utjecati na širok spektar industrijskih, znanstvenih, obrazovnih i društvenih djelatnosti. Time su nastali različiti novi pojmovi za istu stvar, poput brzog prototipiranja, brzog alata, brzog lijevanja i brze proizvodnje. Međutim, 1993. Emanuel Sachs na Tehnološkom institutu u Massachusettsu osmislio je izraz „trodimenzionalni ispis“ i zaštitni znak 3DP™, koji je vrlo brzo prihvaćen na globalnoj razini kao izraz za sve slične procese proizvodnje aditiva.

3.4.3. Masovna upotreba

Godine 2002. bilo je moguće ispisati potpuno funkcionalni životinjski bubreg. Rast baze korisnika olakšan je brojnim aktivnostima i organizacijama, a financiranje zajednice temeljeno je na internetu (tzv. *Crowdsourcing*). Tijekom 2004. i 2005. godine dogodili su se počeci vjerojatno najvažnijeg događaja u povijesti 3D ispisa. Naime, 2005. godine dr. Adrian Bowyer osnovao je RepRap inicijativu kojoj je bio cilj razviti uređaj kojim bi 3D tehnologija postala općedostupna. Godine 2008. RepRap je realizirao projekt 3D samokopiranja izradom printera koji je mogao ispisati gotovo sve vlastite komponente. Novi cilj tog projekta bio je usmjeren na širenje jeftinog 3D ispisa diljem svijeta.

Od 2008. godine upotreba 3D pisaa naglo je porasla. Stvarale su se zajednice i tiskarske usluge koje su omogućavale amateru da iskoristi ispis 3D objekata po niskoj cijeni. Od 2009. godine, paketi za izgradnju vlastitog 3D printera bili su prisutni na tržištu po povoljnim cijenama.

⁴ Monomer je osnovna konstitucijska jedinica polimera.

⁵ Polimeri su tvari građene od makromolekula. Postoje prirodni i sintetički polimeri. Vuna, pamuk i svila su prirodni polimerni materijali.

Od 2011. godine 3D ispis primjenjivao se u gotovo svim oblicima industrije. Dosegao je vrlo visoku razinu u istraživanju, proizvodnji i potrošnji. Tijekom 2012. godine izrađena je potpuno prilagođena proteza čeljusti i od tada se svake godine događaju iznenađujuća dostignuća na ovom polju.

3.5. Proces 3D printanja

Cijeli proces 3D ispisa moguće je podijeliti unutar sljedećih četiri koraka:

1. korak - stvaranje 3D digitalne datoteke objekta
2. korak - rezanje
3. korak - 3D printanje
4. korak - naknadna obrada

Prvi korak 3D ispisa jest stvaranje trodimenzionalne digitalne datoteke objekta kojeg želimo ispisati. Najčešći način stvaranja digitalnog modela jest **3D modeliranje**. No, treba napomenuti da se taj proces može zaobići na nekoliko načina.

Drugi način stvaranja trodimenzionalne digitalne datoteke jest putem **3D skeniranja**. 3D skeniranje je tehnologija blisko povezana s 3D ispisom, koja analizira objekt iz stvarnog svijeta i stvara njegovu digitalnu repliku. 3D skeniranje naširoko se koristi za inženjerske zadatke od strane stručnjaka u industriji. Nakon što je postojeći objekt digitaliziran, imamo mogućnost i promijeniti ga prije ispisa. Za ovaj proces potreban je 3D skener.

Treći način stvaranja trodimenzionalne digitalne datoteke jest **pisanje vlastitog G-koda**. G-kod je programski jezik sastavljen od niza naredbi koje razumije svaki 3D printer. On kontrolira gdje i kako pomicati pisač, kojom brzinom se koristiti, koju temperaturu postaviti itd. Korisno je imati osnovno znanje o G-kodu kako bi se razumjelo kako 3D pisač radi te kako bi se eventualno otklonile poteškoće tijekom 3D printanja. No, treba napomenuti da pisanje vlastitog G-koda nije nimalo jednostavno i može čak oštetiti 3D pisač ukoliko je neispravno izvedeno.

Četvrti način na koji se može dobiti trodimenzionalna digitalna datoteka jest **preuzimanje 3D modela iz internetske biblioteke**. Naime, web stranice poput Thingivers, CrabCad, YouImagine i MyMinifactory Shapeways dostupne su za preuzimanje 3D datoteke koje su modelirali drugi korisnici. Te datoteke su u većini slučajeva spremne za 3D ispis.

Za potrebe ovog diplomskog rada, detaljnije će biti opisan proces 3D printanja pri kojem se trodimenzionalna digitalna datoteka izrađuje 3D modeliranjem.

3.5.1. 3D modeliranje

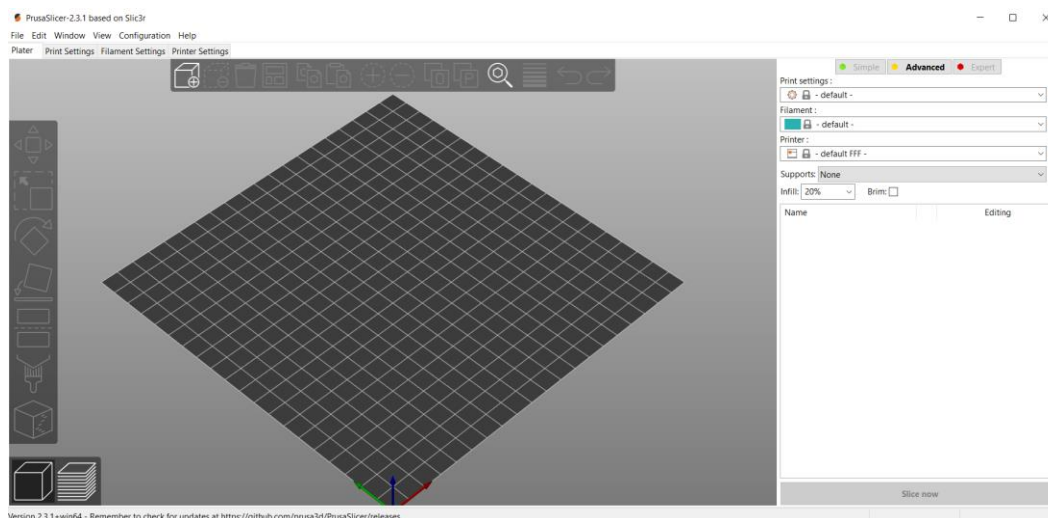
3D modeliranje je proces 3D računalne grafike, odnosno stvaranje matematičkog prikaza bilo kojeg željenog trodimenzionalnog objekta unutar posebnog softvera. Rezultat 3D modeliranja naziva se 3D model. Danas postoji veliki broj takvih softvera - neki su besplatni, a neki su dostupni tek nakon kupnje. Svi oni su slični, ali imaju različita sučelja i koriste se u različite svrhe. Neki od njih su Blender, TinkerCad, Autodesk Fusion 360, SketchUp Free, AutoCad, SolidWorks, Maya, itd. Za potrebe ovog diplomskog rada koristit će se Blender.

Nakon što je izrađen 3D model, datoteka se pohranjuje u odgovarajući format (npr. STL, OBJ, FBX, 3MF). Najčešći format 3D datoteke jest STL, a njezina upotrebljiva ekstenzija je .stl. Puni naziv kratice STL glasi „Standard Triangle Language file“, odnosno „datoteka standardnog jezika trokuta“. STL datoteka ima takav naziv jer opisuje površine objekta kroz trokutaste „mreže“. Danas gotovo svaki softver za 3D modeliranje na tržištu može generirati STL datoteku iz gotovog 3D modela. Potrebno ju je samo izvesti iz korištenog programa za 3D modeliranje.

3.5.2. Rezanje

Da bi 3D printer izradio objekt, potrebne su mu upute. Izraz „rezanje“ (engl. *Slicing*) odnosi se na korištenje računalnog softvera za pretvaranje digitalne 3D datoteke u upute za 3D ispis. Naime, taj softver generira G-kod na temelju izrađene 3D datoteke, kojeg razumije svaki 3D printer. Softveri za rezanje nazivaju se „sliceri“, odnosno na hrvatskom jeziku „rezači“. Razlog takvog naziva jest upravo u tome što ti softveri izrežu 3D model u vodoravne dvodimenzionalne slojeve, koji će se kasnije ispisivati, jedan po jedan. Mnogo je dostupnih „slicer“ programa na tržištu, a neki od njih su PrusaSlicer, Cura, IdeaMaker, Slic3r, Simplify3D. Onaj koji će se koristiti u praktičnom dijelu ovog diplomskog rada jest PrusaSlicer.

Korištenje programa za rezanje prilično je jednostavno. Naime, nakon što je program pokrenut, potrebno je uvesti .stl format (ili neki drugi, po potrebi) modela. Time bi se model trebao pojaviti na radnoj površini softvera. Osim što „slicer“ izrezuje 3D model, taj program sadrži i postavke koje je potrebno odabrati, a to su: visina sloja, brzina i temperatura ispisa, gustoća ispune, pozicija potpornog materijala, vrsta materijala, dimenzije isprintanog objekta, kvaliteta samog ispisa itd. S obzirom na odabrane postavke, softver za rezanje će izračunati koliko će materijala printer morati istisnuti i koliko će za to trebati vremena.



Slika 3.3. Početni zaslon PrusaSlicer softvera

3.5.3. 3D printanje

Kako bi se moglo započeti s 3D printanjem, prvo je potrebno prenijeti datoteku G-koda na 3D printer. Postoji nekoliko načina na koje korisnici 3D pisača to mogu učiniti. Najčešće je riječ o umetanju SD kartice u 3D pisač, spajanju 3D pisača na računalo pomoću USB kabela ili pak korištenju specijaliziranih softvera. Slijedi priprema 3D printera za rad. Prije nego što se započne s ispisom, potrebno je provjeriti materijal i boju. FDM 3D printeri omogućuju odabir različitih veličina mlaznica; manja mlaznica daje detaljnije ispise, dok veća mlaznica smanjuje duljinu ispisa. Većinu 3D pisača nije potrebno nadzirati nakon što je ispis započeo. Prilikom printanja, stroj će slijediti automatizirane upute G-koda. Ovisno o veličini ispisanog objekta, pisaču i korištenim materijalima, ispis je gotov za nekoliko minuta ili pak nekoliko sati. Gotov objekt potrebno je uzeti s platforme.

3.5.4. Naknadna obrada

Naknadna obrada je završna faza 3D ispisa. Nju čine sljedeći postupci (vrlo rijetko je potrebno provođenje svih za isti objekt): uklanjanje potpornog materijala, brušenje, bojanje, poliranje, zavarivanje/sastavljanje.

3.6. Materijali za 3D printanje

Broj raspoloživih materijala za 3D printanje brzo raste jer potražnja tržišta za pojedinim materijalima i mehaničkim svojstvima (čvrstoća, tvrdoća, rastezljivost, elastičnost) potiče neprestani razvoj znanosti o materijalima. Zbog toga je nemoguće dati potpuni pregled svih materijala za 3D printanje. Svaki 3D printer kompatibilan je samo s određenim materijalima. Plastika je najčešće korišten materijal za 3D ispis – dolazi u obliku niti (FDM 3D printeri), smole (SLA 3D printeri) i praha (SLS 3D printeri). Većina plastičnih materijala za 3D ispis može se koristiti i u osobnoj primjeni i u onoj profesionalnoj. Postoje različite vrste filamenata, a neki od njih su PLA, ABS, ASA i PETG. Moguće je korištenje i drugih materijala kao što su metal, keramika, papir te čak i čokolada. Općenito, kada se odlučuje koje materijale odabrati za 3D ispis nekog proizvoda treba svakako uzeti u obzir njegovu funkciju, dimenzije, dizajn te sami printer kojeg se planira koristiti.

4. BLENDER

Blender je besplatan softver otvorenog koda za 3D računalnu grafiku, koji pruža širok opseg primjene. Najčešće se upotrebljuje pri izradi 3D modela, računalnih igara, animiranih filmova, vizualnih efekata, interaktivnih 3D aplikacija, arhitektonskom projektiranju te uređivanju interijera. Nudi brojne usluge, kao što su: 3D modeliranje, teksturiranje, renderiranje, simuliranje, animacija, skriptiranje, UV uređivanje, uređivanje videozapisa, komponiranje itd.



Slika 4.1. Službeni logotip Blendera [20]

Blender je razvio Ton Roosendaal, nizozemski programer i direktor tvrtke „Not a Number Technologies” (NaN). Godine 1994. pokrenuta je prva verzija Blendera, koji je javno objavljen 1998. godine. Isprva je bio dostupan kao shareware softver. Nakon bankrota tvrtke 2002. godine, Blenderov razvoj je prestao. Budući da nije mogao kupiti prava od NaN-ovih podupiratelja, Roosendaal je odlučio pokrenuti neprofitnu organizaciju Blender Foundation, s namjerom da Blender postane softver otvorenog koda. Time je Blender od 2005. godine postao besplatan za korištenje. Danas se Blender izdaje pod licencom *GNU General Public License*. Uz besplatno korištenje, ta licenca korisnicima omogućuje uređivanje koda i distribuciju izmijenjenih verzija. Blender se neprekidno razvija od strane stotina programera, umjetnika i ostalih ljubitelja ovog programa diljem svijeta. Upravo takva zajednička primjena znanja pomaže Blenderu da postane sve kvalitetniji i napredniji. Dostupno je mnogo verzija ovog programa, a u ovom diplomskom radu korištena je inačica 2.93.

Općenito, Blender je kompatibilan s operativnim sustavima Windows, macOS i Linux. Nakon preuzimanja softvera, postupak instalacije je jednostavan. Treba napomenuti da je prilično izazovno naučiti se koristiti Blenderom zbog mnogih mogućnosti koje nudi. Za napredno korištenje Blendera trebat će godine iskustva. Također, treba svakako naglasiti da veliku pomoć pri upoznavanju s njime predstavljaju brojni tutorijali koji su dostupni na službenoj web stranici blend.org ili pak odgovarajućim platformama (npr. YouTube).

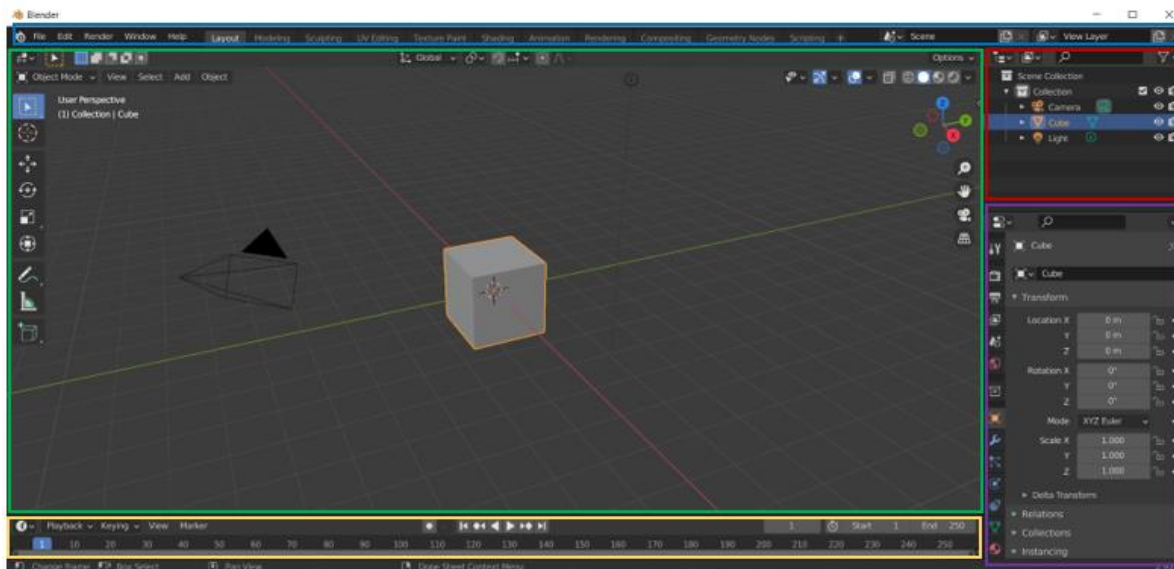
4.1. Korisničko sučelje

Prilikom pokretanja Blendera, u sredini prozora pojavljuje se uvodni zaslon. Zaslon sadrži mogućnosti za stvaranje novih projekata ili otvaranje nedavno otvorenih .blend datoteka. Također, ponuđeno je nekoliko oblika primjene programa, a za potrebe ovog rada odabire se oblik *General*.



Slika 4.2. Uvodni zaslon inačice 2.93

Zatim je moguće vidjeti cjelokupno sučelje Blendera. Sredina sučelja (tzv. scena) sastoji se od kocke (engl. *Cube*), kamere (engl. *Camera*), svjetla (engl. *Light*) i koordinatne mreže. Scena je prostor u kojem se model vizualizira i modificira.



Slika 4.3. Izgled sučelja nakon otvaranja Blendera

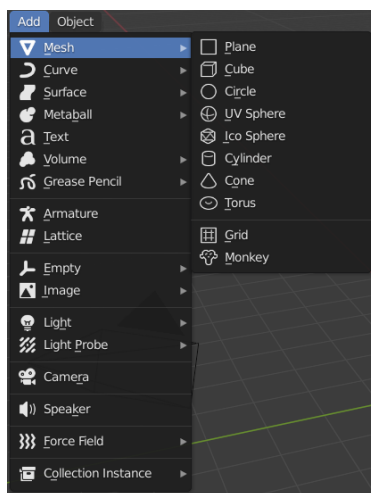
Izgled sučelja nakon otvaranja Blendera moguće je vidjeti na slici 4.3. Podijeljeno je unutar pet uređivača:

1. Info ploča (engl. *Info panel*) - plavi okvir
2. 3D prikaz (engl. *3D viewport*) - zeleni okvir
3. Zbirka scene (engl. *Scene collection*) - crveni okvir
4. Svojstva (engl. *Properties*) – ljubičasti okvir
5. Vremenska crta (engl. *Timeline*) - žuti okvir

Uređivač *Info ploča* obuhvaća opće izbornike *File*, *Edit*, *Render*, *Window* i *Help*. Desno od izbornika nalaze se radni prostori, a bira se onaj koji odgovara zadatku koji će se obavljati u Blenderu. Vrlo je korisno prebacivati se između različitih radnih prostora unutar iste datoteke. Riječ je o sljedećim radnim prostorima: *Layout*, *Modeling*, *Sculpting*, *UV Editing*, *Texture Paint*, *Shading*, *Animation*, *Rendering*, *Compositing*, *Geometry Nodes*, *Scripting*. Prikazano sučelje na slici sadrži radni prostor *Layout*.

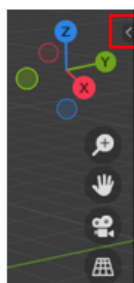
Uređivač *3D prikaz* koristi se za interakciju s 3D scenom u različite svrhe i ujedno je najznačajniji tip uređivača. Upotrebljuje se unutar gotovo svih ostalih radnih prostora. Sastoji se od alatne trake, zaglavlja i bočne trake. Zaglavlje ovog uređivača sadrži različite izbornike (*Mode*, *View*, *Select*, *Add*, *Object*), kontrole za transformaciju, način prikaza objekata te postavke sjenčanja. Alatna traka nalazi se na lijevoj strani uređivača, a sadrži kursor te alate za premještanje,

skaliranje, rotiranje, mjerenje i komentiranje objekata na sceni. Pomoću izbornika *Add* na scenu je moguće dodati neki od ponuđenih geometrijskih primitiva (kocka, krug, sfera, valjak, stožac itd.). Za potrebe ovog rada u praktičnom dijelu dodavati će se jedino kocka (engl. *Cube*).



Slika 4.4. Izbornik *Add*

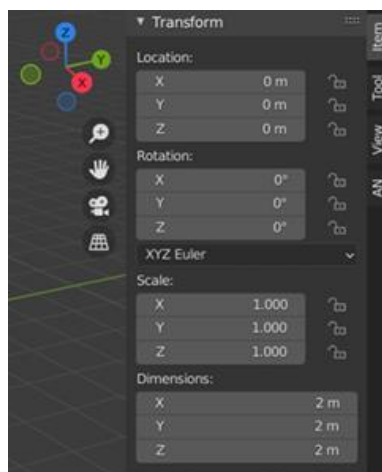
Desna strana uređivača namijenjena je za navigaciju. Općenito, navigacija u 3D prostoru Blendera vrši se korištenjem miša, tipkovnih prečaca te već spomenute desne strane uređivača *3D prikaz*. Na slici 4.5. moguće je vidjeti oznake X (crvena boja), Y (zeleno) i Z (plava boja) osi. Pomicanjem miša gore-dolje ili lijevo-desno pokraj bilo koje od tih oznaka, promijenit će se kut pod kojim se promatraju scena i objekti koje ona sadrži. Također, na slici su prisutna četiri gumba, navedena od vrha do dna. Prvi gumb služi za „zumiranje” 3D prikaza, odnosno bolju vidljivost objekata na sceni. Drugi gumb pomiče pogled gore, dolje, lijevo i desno. Treći gumb prikazuje trenutnu scenu, promatranu iz gledišta aktivne kamere na sceni. Četvrti gumb mijenja projekciju kamere za prikaz.



Slika 4.5. Dio uređivača *3D prikaz*, namijenjen za navigaciju

Bočna traka nije vidljiva tijekom otvaranja Blendera. Do nje se dolazi klikom na strelicu, označenu crvenom bojom na slici 4.5. Bočna traka (slika 4.6.) prikazuje postavke transformacije određenog objekta na sceni. Općenito, kako bi bilo moguće upravljati željenim objektom, potrebno ga je prvo označiti klikom unutar scene ili *Zbirke scene*. Kada je objekt označen, bridovi tog objekta poprimit će narančastu boju. Transformacije se odnose na brojne operacije koje se mogu izvoditi nad odabranim objektom. Svaki objekt može se premještati, rotirati i skalirati u *Object*

Mode načinu uređivanja (o tom načinu rada govorit će se malo kasnije). Riječ je o trima osnovnim transformacijama u Blenderu koje je moguće provesti nad objektima.

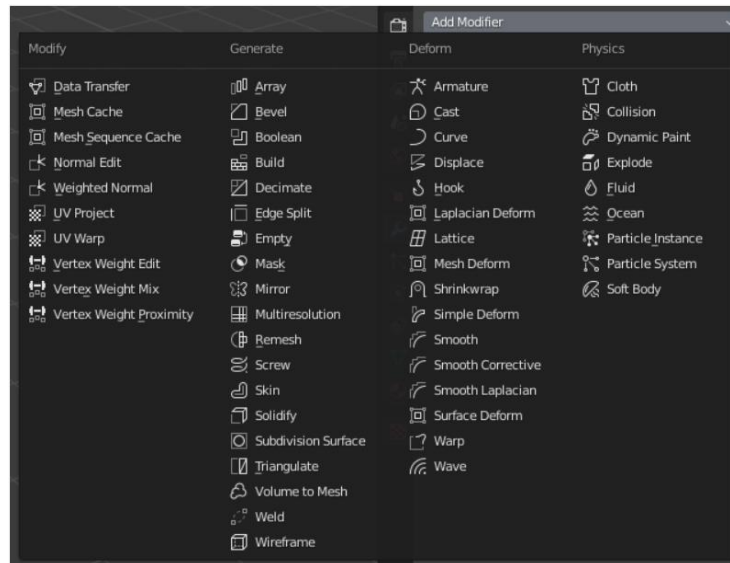


Slika 4.6. Bočna traka uređivača 3D prikaz

Nakon što je objekt označen, premještanje objekta provodi se pritiskom tipke „G“ na tipkovnici. Kada je to učinjeno, miš je potrebno pomicati u smjeru položaja na sceni na koji ga se želi premjestiti. Premještanjem objekta promijeniti će se njegova lokacija (engl. *Location*), a njezine koordinate moguće je vidjeti unutar Bočne trake. Rotacija (engl. *Rotation*) objekta provodi se pritiskom tipke „R“ na tipkovnici, a obavlja na isti način opisan kao kod premještanja objekta. Skaliranje objekta obavlja se pomoću tipke „S“ na tipkovnici. Osim pomoću tipkovnice, transformacije objekata moguće je provesti pomoću Alatne trake ili pak unošenjem promjena unutar Bočne trake.

Unutar *Zbirke scene* moguće je vidjeti popis objekata koji pripadaju sceni. Klikom na odgovarajući objekt i pritiskom tipke „Delete“ na tipkovnici moguće ga je izbrisati. Također, klikom na ikonu oka koja se nalazi desno od naziva svakog objekta može se odabrati hoće li objekt biti vidljiv na sceni ili ne.

Uređivač *Svojstva* omogućuje uređivanje mnogih podataka, koji utječu na scenu i njezine objekte. S lijeve strane vidljiv je stupac ikona, putem kojih se odabiru odgovarajuće postavke. U praktičnom dijelu ovog diplomskog rada, unutar ovog uređivača jedino će se koristiti odjeljak *Modifier Properties*. Taj odjeljak sadrži padajući izbornik *Add Modifier*, u kojem je moguće dodati željeni modifikator. Općenito, modifikatori koriste algoritam za izmjenu objekta bez potrebe za njegovim ručnim uređivanjem. Rezultat primjene modifikatora obično je složeniji od prethodno opisanih transformacija. Na jedan objekt moguće je primijeniti više modifikatora.



Slika 4.7. Izbornik modifikatora

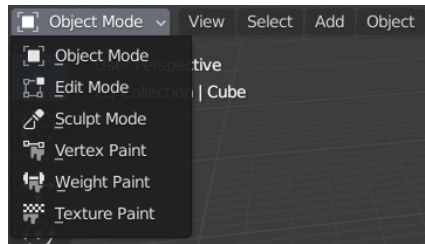
Tijekom izrade 3D modela fraktala Tepiha Sierpińskog i Mengerove spužve isključivo će se koristiti modifikator *Boolean*. Pomoću tog modifikatora moguće je obavljati tri različite operacije nad odabranim objektima – uniju (engl. *Union*), presjek (engl. *Intersect*) ili razliku (engl. *Difference*). Ukoliko je želja da promjene koje je učinio modifikator budu trajne, potrebno ga je primijeniti. Za potrebe ovog rada koristit će se jedino operacija razlike. Način korištenja ovog modifikatora pobliže će se opisati tijekom same izrade spomenutih 3D modela.

Uređivač *Vremenska crta* služi za animiranje. Za potrebe ovog diplomskog rada on neće biti potreban.

4.2. Načini uređivanja objekata

U Blenderu je moguće uređivati objekte na šest načina, a to su *Object Mode*, *Edit Mode*, *Sculpt Mode*, *Vertex Paint*, *Weight Paint* i *Texture Paint*. Odabir načina uređivanja pojedinog objekta ovisi o samom objektu. Na primjer, kameru i svjetlo moguće je uređivati koristeći samo jedan način - *Object Mode*. Kocka se pak na primjer može uređivati na svih šest navedenih načina. Ona pripada vrsti takozvanih mrežastih objekata koji se sastoje od zasebnih skupina elemenata: vrhova (engl. *Vertices*), bridova (engl. *Edges*) i strana (engl. *Faces*).

Navedene elemente nije moguće odabrati i uređivati sve dok je zadan način *Object Mode*. Riječ je o glavnom načinu koji je uključen u Blender prema zadanim postavkama. U tom načinu rada transformacije se primjenjuju na cjelokupan objekt. S druge strane, u načinu uređivanja *Edit Mode* promjene utječu na pojedinačne elemente. Prelazak na drugi način uređivanja vrši korištenjem padajućeg izbornika u zaglavlju uređivača *3D prikaz* (slika 4.8.).



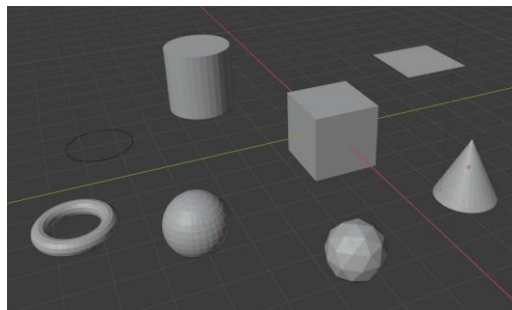
Slika 4.8. Izbornik načina uređivanja objekata

U praktičnom dijelu rada koristiti će se jedino način uređivanja *Object Mode*.

4.3. Tehnike 3D modeliranja

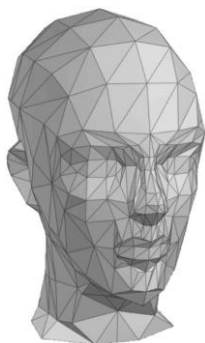
Unutar Blendera moguće je na više načina izraditi 3D modele. Postoji nekoliko tehnika 3D modeliranja. Na odabir same tehnike utječu dizajn i primjena konačnog proizvoda, kao i sposobnosti i vještine njegovog kreatora. U nastavku će se spomenuti samo nekoliko tehnika koje je moguće koristiti unutar Blendera.

Objektno modeliranje (engl. *Object modeling*) najjednostavniji je oblik modeliranja. Modeliranje započinje od jednog ili više geometrijskih primitiva, a mijenjajući ih uz pomoć osnovnih transformacija, modifikatora te ostalih ponuđenih Blenderovih alata (kopiranje, istiskivanje, rezanje, spajanje itd.) stvaraju se složeniji 3D modeli.



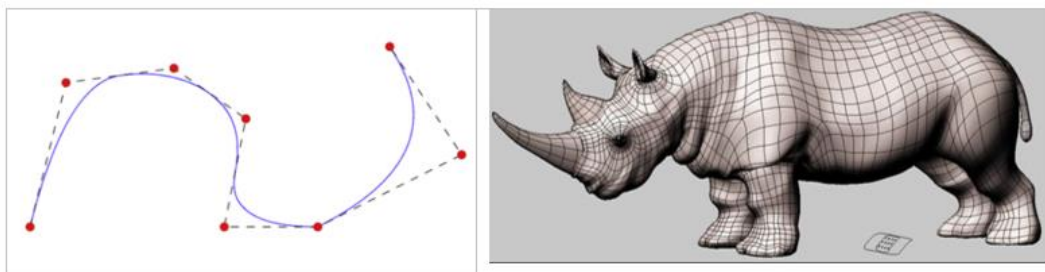
Slika 4.9. Blenderovi geometrijski primitivi

Poligonalno modeliranje (engl. *Polygonal modeling*) naziv je dobilo po poligonu, ključnoj komponenti takozvanog poligonalnog 3D modela. Općenito, geometrija poligona sastoji se od točaka 3D prostora te ravnih linija koje ih povezuju (rubova). Model nastao ovom tehnikom modeliranja definiran je skupom međusobno povezanih poligona, pri čemu su poligoni međusobno povezani zajedničkim rubnim točkama. Skup tih poligona naziva se poligonalna mreža, a kreiranjem poligonalne mreže definira se željeni oblik 3D modela. Nedostatak ove tehnike modeliranja jest u tome što nije moguće dobiti zakrivljene površine. Općenito, značajke poligonalnog modeliranja čine ga idealnim za slučajeve upotrebe u kojima preciznost nije toliko bitna, već je naglasak na vizualizaciji.



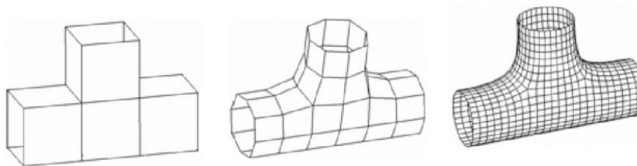
Slika 4.10. *Primjer 3D modela stvorenog pomoću poligonalne tehnike* [21]

NURBS modeliranje (engl. *NURBS modeling*) je tehnika modeliranja kojom se 3D modeli stvaraju pomoću krivulja. NURBS je skraćenica za *Non Uniform Rational B-Spline*, a riječ je o parametarski definiranoj krivulji koja je zadana težinskim kontrolnim vrhovima. Svaka krivulja ima svoj početak, kraj i zakrivljenost. Manipulacijom kontrolnim vrhovima 3D dizajner kontrolira zakrivljenost krivulja. Nakon što su kreirane sve krivulje, slijedi stvaranje takozvanih zakrpa (engl. *Patches*) između odgovarajućih kontrolnih vrhova. Međusobnim spajanjem nastalih zakrpa dobiva se savršeno glatka površina NURBS 3D modela.



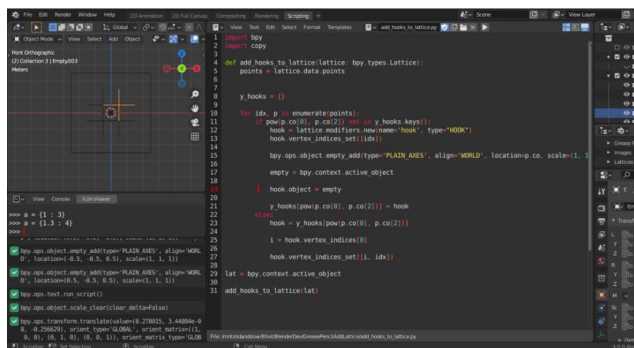
Slika 4.11. *Primjer NURBS krivulje i 3D modela stvorenog pomoću NURBS tehnike* [22] [23]

Subdivizijsko modeliranje (engl. *Subdivision modeling*) je tehnika modeliranja u kojem stvaranje 3D modela započinje primjenom poligonalne tehnike modeliranja. Nakon što je kreirana željena poligonalna mreža objekta, koristi se NURBS matematika prilikom koje se vrši podjela svakog postojećeg poligona na četiri manja poligona. Tim postupkom zaoblit će se oštri dijelovi objekta. Treba napomenuti da se opisana podjela vrši nad poligonalnom strukturom koja se u cijelosti (ili prvenstveno) sastoji od četverokuta. Postupak podjele moguće je ponoviti koliko god se ne dobije zadovoljavajući izgled 3D objekta. Što će se dalje ponavljati postupak, to će broj kontrolnih vrhova objekta biti veći.



Slika 4.12. *Zaobljavanje površine objekta pomoću subdivizijske tehnike* [24]

Proceduralno modeliranje (engl. *Procedural modeling*) je najnaprednija tehnika modeliranja. Za razliku od prethodno navedenih tehnika, korištenjem ove tehnike 3D model izrađuje se algoritamski. Poznato je da Blender koristi Python konzolu. Inače, Blender ima ugrađeni Python interpreter koji se aktivira prilikom pokretanja Blendera i ostaje aktivan sve dok Blender radi. Radni prostor *Scripting* korisniku omogućuje pisanje vlastite Python skripte. Time su bitno proširene mogućnosti 3D modeliranja unutar Blendera. Postoji još jedan vrlo zanimljiv oblik proceduralnog modeliranja, tijekom kojeg se koristi Blenderov dodatak Animacijski čvorovi. Upravo će se Animacijski čvorovi koristiti u praktičnom dijelu ovog diplomskog rada.



Slika 4.13. 3D modeliranje u Blenderu pomoću Python skripte [25]

4.4. Animacijski čvorovi

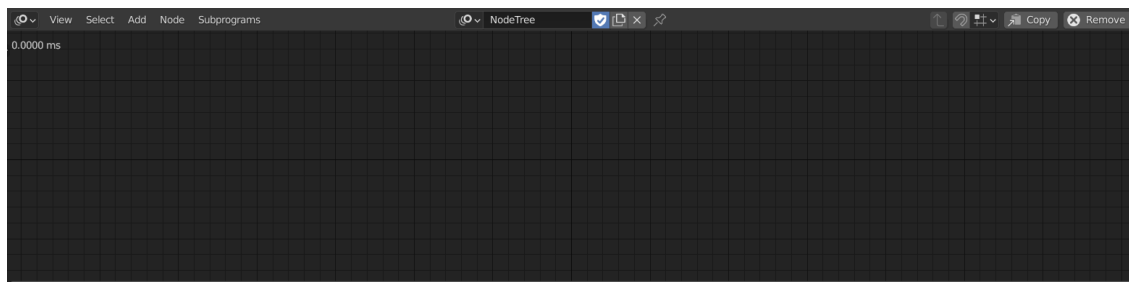
4.4.1. O dodatku

Dodatak Animacijski čvorovi (engl. *Animation Nodes*) sustav je vizualnog skriptiranja, baziran na čvorovima. Mnogi dijelovi Animacijskih čvorova prepisani su koristeći Cython programski jezik. Taj jezik kombinira Python i C programske jezike. [17] Godine 2015. njemački programer Jacques Lucke započeo je razvoj Animacijskih čvorova. Na internetu je dostupan službeni priručnik o Animacijskim čvorovima koji sadrži vodič za instalaciju dodatka, opis funkcije svakog čvora te mnogobrojne tutorijale za samostalno učenje. Budući da Animacijski čvorovi nisu unaprijed instalirani u Blenderu, ovaj dodatak potrebno je naknadno instalirati kako bi se mogao koristiti.

Postoji mnogo čvorova sa širokim opsegom primjene. Potrebno ih je dobro razumjeti kako bi njihova primjena bila uspješno realizirana. Nakon što su spojeni na odgovarajući način, skup tih čvorova čini jedno stablo čvorova (engl. *Node Tree*).

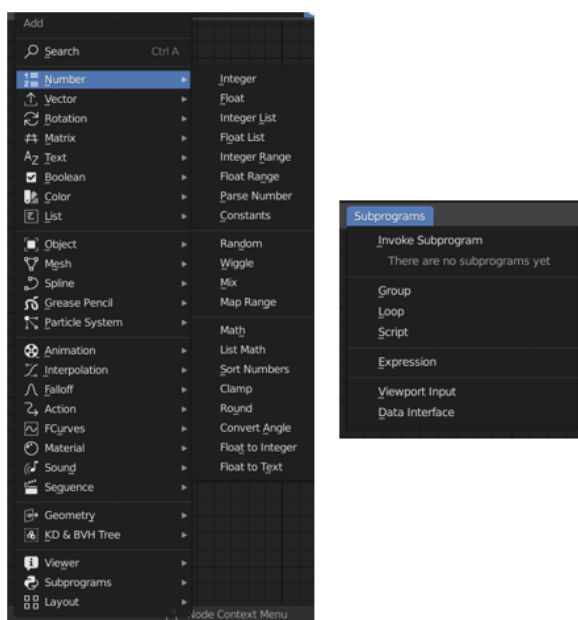
4.4.2. Radni prostor

Rad s Animacijskim čvorovima omogućen je unutar posebnog radnog prostora.



Slika 4.14. Radni prostor namijenjen za korištenje Animacijskih čvorova

Unutar prikazanog radnog prostora dodaju se čvorovi koji su potrebni za rad. Postoje dva načina za dodavanje čvorova, a to su korištenje izbornika *Add / Subprograms* ili pak tipkovnog prečaca *Shift + A*.



Slika 4.15. Izbornici *Add* (lijevo) i *Subprograms* (desno)

4.4.3. Izvršenje stabla čvorova

Općenito, Animacijski čvorovi pretvaraju stablo čvorova u skriptu koju je moguće izvršiti na nekoliko načina.

Opcija *Always* je uključena prema zadanim postavkama i provodi izvršavanje stabla čvorova što je češće moguće za najbolje i najglade ažuriranje scene. No to opterećuje procesor i ometa druga područja u Blenderu. Nema smisla izvršavati stablo ukoliko se ništa nije promijenilo u stablu čvorova ili na sceni. Osim ako je apsolutno potrebna, ovu opciju je bolje ne koristiti. [17]

Opcija *Tree Changed* ukoliko je omogućena, izvršiti će stablo čvorova svaki put kada se stablo čvorova promijeni, odnosno kada se dodaje ili ukloni neki čvor. [17]

Opcija *Frame Changed* ukoliko je omogućena, izvršiti će stablo čvorova svaki put kada se promijeni trenutni okvir scene. [17]

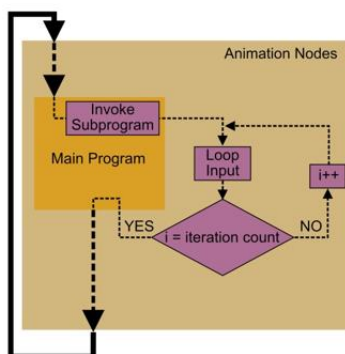
Opcija *Property Changed* ukoliko je omogućena, izvršiti će stablo čvorova svaki put kada se promijeni svojstvo, odnosno ulaz ili postavka bilo kojeg čvora. [17]

4.4.4. Programi i potprogrami

Prema priručniku za Animacijske čvorove, svako stablo čvorova može se smatrati programom. U programiranju, za ponavljanje određenih naredbi praksa je koristiti petlje. Isto pravilo vrijedi i za Animacijske čvorove. Potprogrami (engl. *Subprograms*) omogućuju korisniku stvaranje određenog bloka funkcija, koje se mogu pozvati s bilo kojeg mjesta u stablu čvora pomoću odgovarajućeg čvora potprograma za pozivanje *Invoke Subprogram*. Svaki potprogram ima svoju boju; svi čvorovi u istom potprogramu dijele istu boju čvora, što olakšava prepoznavanje pripadnosti svakog čvora. Općenito, dostupne su tri vrste potprograma: petlje, grupe i skripte. Ostali čvorovi koji nisu dio potprograma obojani su najčešće sivom bojom. [16]

4.4.5. Petlja

Loop je poseban tip potprograma, sličan petljama u programiranju. Njegova uloga je brojanje iteracije koja definira koliko puta će se petlja ponoviti. Model izvršavanja petlje u Animacijskim čvorovima prikazan je na slici. Iznos iteracije korisnik može definirati ručnim unosom vrijednosti unutar ulaza čvora *Invoke Subprogram*. [16]

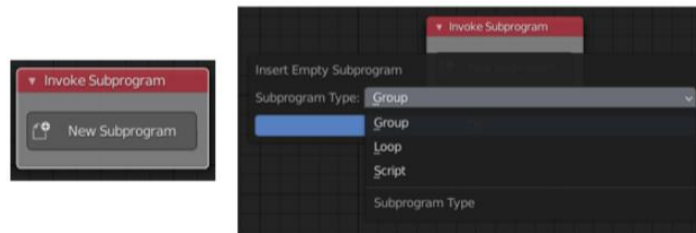


Slika 4.16. Model izvršenja petlje u Animacijskim čvorovima [16]

U nastavku će se dati opis samo onih čvorova koji će se koristiti u praktičnom dijelu ovog diplomskog rada.

4.4.6. Čvorovi

Čvor potprograma *Invoke Subprogram* nalazi se unutar izbornika *Subprograms*. Taj čvor odrađuje isti zadatak (ili niz operacija) za svaki objekt posebno. Nakon što je dodan unutar radnog prostora, klikom na *New Subprogram* unutar samog čvora, moguće je odabrati tri tipa ovog potprograma. U praktičnom dijelu rada isključivo će se birati tip *Loop*. Odabirom tipa *Loop*, potprogram *Invoke Subprogram* automatski će stvoriti čvor potprograma *Loop Input*.



Slika 4.17. Čvor *Invoke Subprogram* i njegovo dodavanje

Čvor potprograma **Loop Input** dopušta ponavljanje u čvorovima. Iteracija (engl. *Iteration*) je čin izvršavanja grupe čvorova više puta. Svaka iteracija ima pridružen cijeli broj koji se zove indeks (engl. *Index*).



Slika 4.18. Čvor *Loop Input*

Čvor potprograma *Loop Input* može imati parametre, odnosno vrijednosti koje su konstantne za sve iteracije osim ako se izričito ne mijenjaju. Vrijednosti koje se mijenjaju iz iteracije u iteraciju definirane su onim što je poznato kao iterator. Iteratori i parametri se mogu dodati klikom na gumbove plus *New Iterator* i *New Parameter*. Ukoliko je dodan novi iterator, unutar čvora *Loop Input* stvorit će se crni prozor iznad gumba *New Iterator*. Ukoliko je dodan novi parameter, unutar čvora *Loop Input* stvorit će se crni prozor iznad gumba *New Parameter*. Generator je popis koji je u početku prazan i omogućuje mu dodavanje vrijednosti prilikom svake iteracije.

Parametri su konstantni za sve iteracije, međutim, može se promijeniti njihova vrijednost u određenoj iteraciji koristeći ono što je poznato kao operator ponovnog dodjeljivanja, Čvor tog operatora naziva se **Reassign Loop Parametar**. Njega se dodaje klikom na gumb plus *New Generator Output* unutar čvora *Loop Input*.

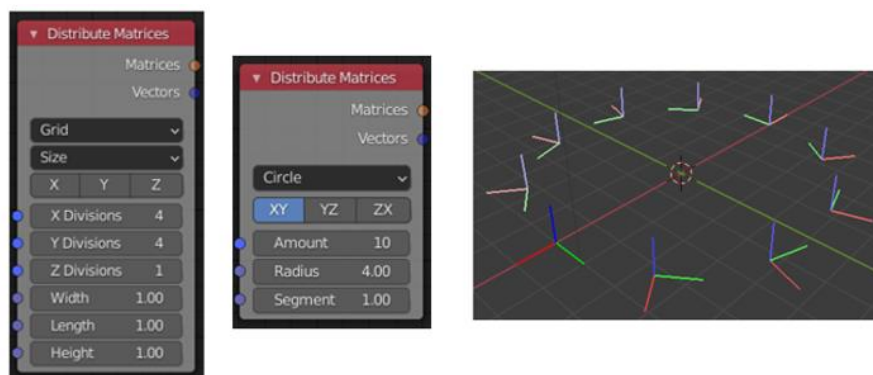
Čvor **Matrix List** sadrži listu matrica koje predstavljaju lokacije buduće stvorenih objekata. Provođenjem daljnjih iteracija, ulaz *Matrix List* čvora sadržavat će sve veći broj lokacija. Dodaje ga se klikom na gumb plus *New Generator Output* unutar čvora *Loop Input*.

Čvor **Combine Euler** nalazi se unutar izbornika *Add*, odnosno kategorije *Rotation*. Ovaj čvor provodi rotaciju objekta oko x, y ili z - osi, nakon što su zadovoljeni određeni unaprijed definirani uvjeti.

Čvor **Combine Matrix List** nalazi se unutar kategorije *List*. Ovaj čvor se koristi za kombiniranje dviju lista matrica.

Čvor **Compose Matrix** nalazi se unutar kategorije *Matrix*. Čvor *Compose Matrix* formira matricu transformacije koja se može koristiti za primjenu određenih transformacija (premještanje, rotacija, skaliranje). [17]

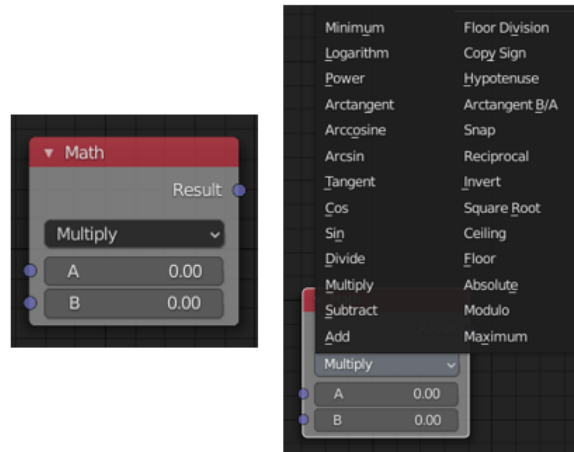
Čvor **Distribute Matrices** nalazi se unutar izbornika *Add*, odnosno kategorije *Matrix*. Taj čvor stvara listu matrica koje predstavljaju lokacije buduće stvorenih objekata, koji će biti raspoređeni na određen način u prostoru. Postoji više opcija čvora *Distribute Matrices* koje je moguće odabrati, a upravo se njima definira način na koji će lokacije biti raspoređene u prostoru. Moguće ih je rasporediti duž x osi (opcija *Linear*), 3D rešetke (opcija *Grid*), kruga (opcija *Circle*), mreže (opcija *Mesh*), spirale (opcija *Spiral*) itd. U praktičnom dijelu koristit će se isključivo opcija *Grid*. Ta opcija ima dvije podopcije, *Size* i *Step*. Koristiti će se podopcija *Size* koja omogućuje definiranje dimenzija rešetke (*Width*, *Length*, *Height*), odnosno broj matrica na x, y i z osi (*X Division*, *Y Division*, *Z Division*).



Slika 4.19. Čvor *Distribute Matrices* s opcijama *Grid* i *Circle* te jedan primjer rasporeda lokacija

Čvor **Get Length** automatski se stvara nakon što se spoje čvorovi *Invoke Subprogram* i *Object Instancer*. Čvor *Get Length* računa koliko kopija treba stvoriti čvor *Object Instancer*.

Čvor **Math** nalazi se unutar izbornika *Add*, odnosno kategorije *Number* i obavlja ukupno 26 matematičkih funkcija. Izlaz čvora rezultat je jedne od operacija nad ulazima. [17] Operacije koje će se koristiti u praktičnom dijelu su množenje (*Multiply*) i dijeljenje bez ostatka (*Floor Division*).



Slika 4.20. Čvor *Math* i njegove operacije

Čvor **Matrix Math** nalazi se unutar kategorije *Matrix*. Taj čvor omogućuje množenje matrica transformacija. Množenjem se dobiva nova matrica transformacije. Treba imati na umu da množenje matrica nije komutativno, stoga treba voditi računa o redoslijedu postavljenih čvorova.

Čvor **Object Instancer** nalazi se unutar kategorije *Object*. Taj čvor dinamički stvara i kopira objekt. Postoje tri opcije unutar ovog čvora, a to su *Copy from Source*, *Copy Full Object* te *Deep Copy*. Buduće da se tijekom modeliranja u praktičnom dijelu ispostavilo da niti jedna od ovih opcija koja je uključena/isključena ne radi promjenu na sceni, sve opcije su mogle biti i isključene.

Čvor **Object Matrix Output** nalazi se unutar kategorije *Object*. Taj čvor transformira ulazne objekte po matrici ulaznih transformacija.

Čvor **Object Transforms Output** nalazi se unutar kategorije *Object*. Ovaj čvor postavlja lokaciju, rotaciju i skalu ulaznog objekta na izlaznu transformaciju. [17] Klikom na potvrdne okvire unutar čvora definira se željena transformacija.

Čvor **Replicate Matrix** nalazi se unutar kategorije *Matrix*. Ovaj čvor više puta kopira ulaznu matricu, a zatim transformira svaku novu kopiju na temelju odgovarajuće transformacijske matrice.

Čvor **Transform Matrix** nalazi se unutar kategorije *Matrix*. Taj čvor transformira ulaznu matricu matricom ulazne transformacije.

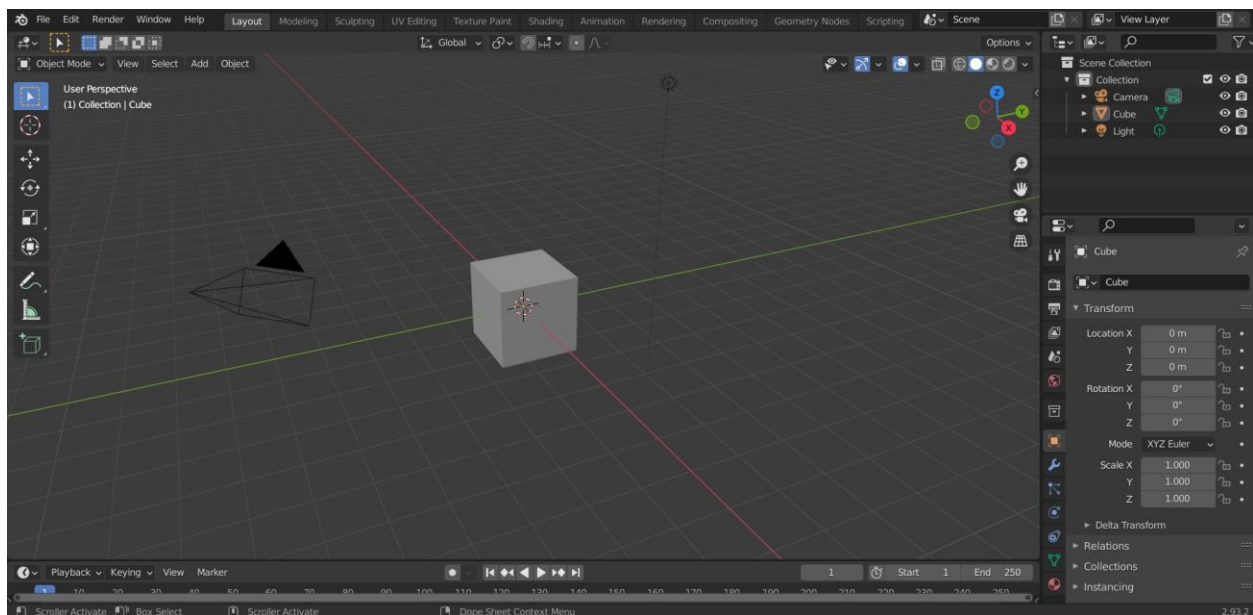
Čvor **3D Viewer** nalazi se unutar kategorije *Viewer*. Ovaj čvor vizualizira radnje čvorova koji pripadaju kategorijama *Matrix* i *Vector*. Vektori se crtaju kao točke, a matrice kao obojene osi čija se zasićenost smanjuje kako se indeks matrice povećava.

5. PRAKTIČNI DIO IZRADE TEPIHA SIERPIŃSKOG POMOĆU 3D PRINTERA

5.1. 3D modeliranje u Blenderu

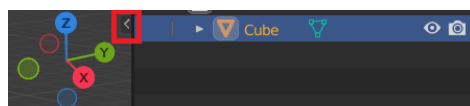
5.1.1. Nulta iteracija

Otvaranjem Blendera, na sceni su prikazani osnovni elementi njegovog trodimenzionalnog prostora: kocka (engl. *Cube*), kamera (engl. *Camera*) i svjetlo (engl. *Light*).



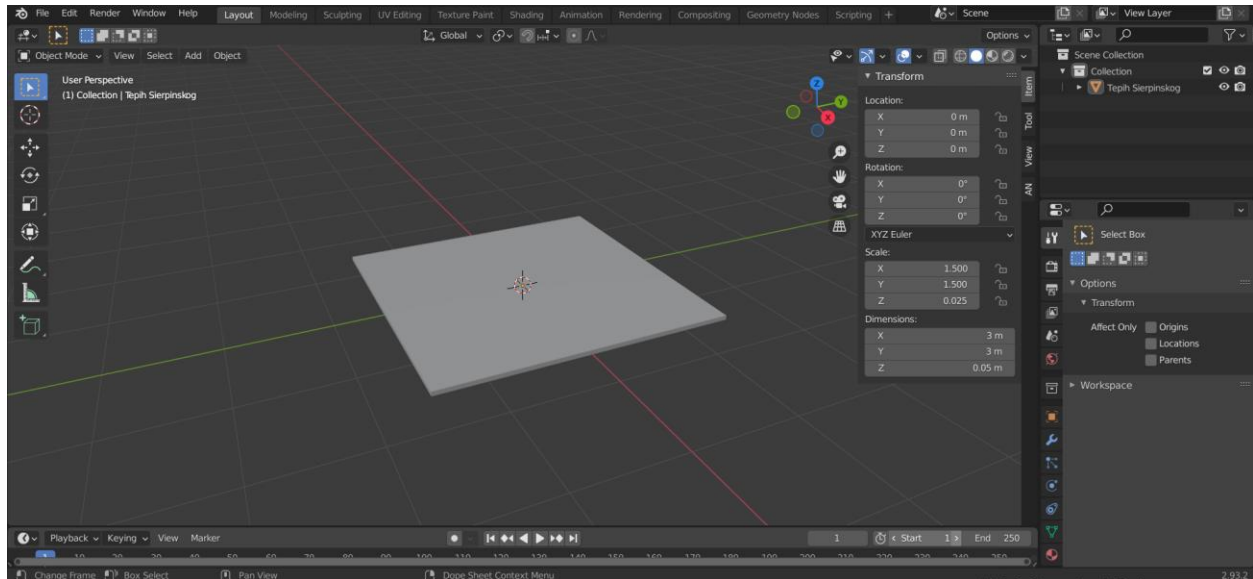
Slika 5.1. Početno sučelje Blendera

Budući da tijekom izrade modela neće biti potrebni niti kamera niti svjetlo, unutar *Scene Collection* može ih se obrisati. To se radi na način da se lijevim klikom miša označi kamera te na tipkovnici pritisne *Delete*. Isto se zatim napravi za svjetlo. Kocki koja je ostala na sceni odabrat će se sljedeće dimenzije: 3 x 3 x 0.05 m. Bitno je naglasiti da su one proizvoljne. Izborom ovakvih dimenzija prilagodit će se izgled izvorne kocke kako bi više nalikovala na kvadrat, koji će se kasnije moći printati. Promjena dimenzija provodi se unutar *Scene Collection*, tako da se klikom označi „Cube“, a zatim pritisne mala strelica prikazana na slici 5.2. Time će se otvoriti Bočna traka unutar koje se može birati lokacija, način rotacije i skaliranja te dimenzije određenog objekta u prostoru.



Slika 5.2. Strelica za otvaranje Bočne trake

Kocka s izmijenjenim dimenzijama trenutno se naziva „Cube“, a unutar *Scene Collection* preimenovat će se u „Tepih Sierpińskog“. To se provodi tako da se dva puta brzim potezom pritisne na naziv „Cube“ te se zatim umjesto njega upiše onaj željeni. Gotov model prikazan je na slici 5.3. On ujedno predstavlja nultu iteraciju fraktala Tepiha Sierpińskog. Preostaje spremanje kreiranog 3D projekta na računalo, a biti će spremljen u obliku datoteke s ekstenzijom .blend.



Slika 5.3. Model nulte iteracije Tepiha Sierpińskog

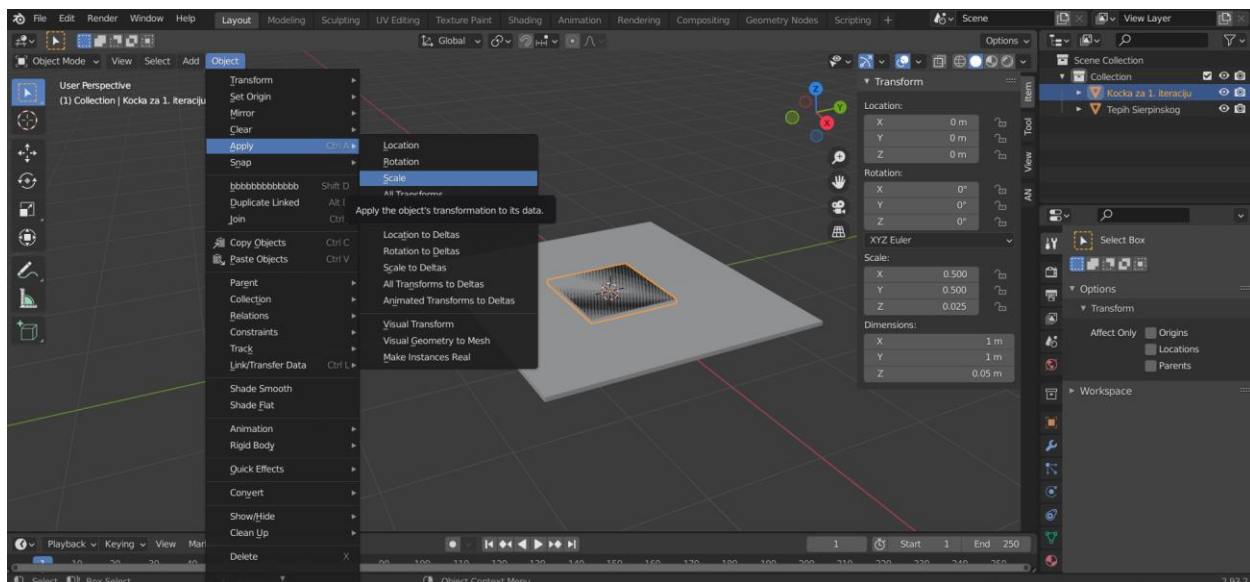
5.1.2. Prva iteracija

1. Dodavanje novog objekta na scenu

Prvu iteraciju fraktala Tepiha Sierpińskog izradit će se provođenjem određenih operacija na prethodno izrađenom modelu nulte iteracije. Stoga je potrebno otvoriti spremljenu 3D datoteku modela nulte iteracije. Prvo što će se napraviti jest dodavanje „kocke“, kojoj će se izmijeniti dimenzije u: 1 x 1 x 0.05 m. Preimenovat će se u „Kocka za 1. iteraciju“. Nju se dodaje s ciljem da se pomoću nje napravi središnja rupa, koju sadrži prva iteracija fraktala Tepiha Sierpińskog.

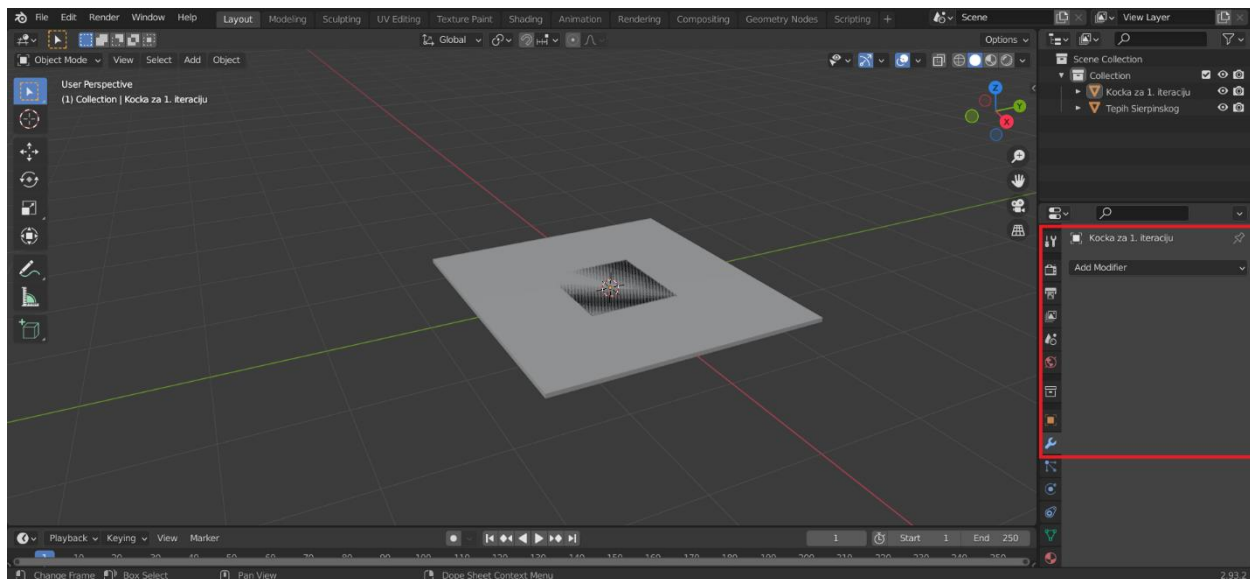
2. Dodavanje modifikatora

Središnja rupa stvorit će se pomoću modifikatora *Boolean*, odnosno korištenjem operacije razlike (engl. *Difference*). Taj modifikator primijenit će se na objekt „Tepih Sierpińskog“. Primjenom tog modifikatora, oduzet će se objekt „Kocka za 1. iteraciju“ od objekta „Tepih Sierpińskog“. Time će se dobiti rupa prve iteracije Tepiha Sierpińskog. Prije nego što se primijeni spomenuti modifikator, potrebno je normalizirati dimenzije objekta „Kocka za 1. iteraciju“. To se provodi tako da se unutar *Scene Collection* lijevim klikom miša označi „Kocka za 1. iteraciju“ te zatim unutar izbornika *Object* pritisne *Apply* → *Scale*.



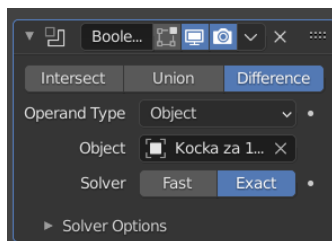
Slika 5.4. Normalizacija dimenzija manjoj „kocki“

Time će označena „kocka“ poprimiti skalu 1 za svaku od triju koordinatnih osi. Sljedeći korak je dodavanje modifikatora *Boolean*. No prije dodavanja modifikatora, potrebno je unutar *Scene Collection* lijevim klikom miša označiti „Tepih Sierpińskog“. To se izvodi jer je inače prije dodavanja modifikatora uvijek potrebno označiti onaj objekt od kojeg će se vršiti oduzimanje. Nakon toga potrebno je kliknuti na ikonu plavog ključa odjeljka *Modifier Properties* (slika 5.5.). Klikom na tu ikonu, desno će biti prikazana opcija *Add Modifier*. Nakon pritiska lijevom klikom miša na *Add Modifier*, pojavit će se izbornik modifikatora.



Slika 5.5. Dodavanje modifikatora

Odabrat će se modifikator *Boolean*. Nakon odabira modifikatora, pojavit će se prozor u kojem je potrebno odabrati operaciju koju će modifikator obavljati (odabire se razlika, engl. *Difference*), vrstu operandu (odabire se objekt, engl. *Object*) te konkretan objekt koji ima ulogu operandu (odabire se „Kocka za 1. iteraciju“). Operandom se smatra onaj objekt koji se želi oduzeti od nekog drugog objekta. Odabrane postavke moguće je vidjeti na sljedećoj slici.

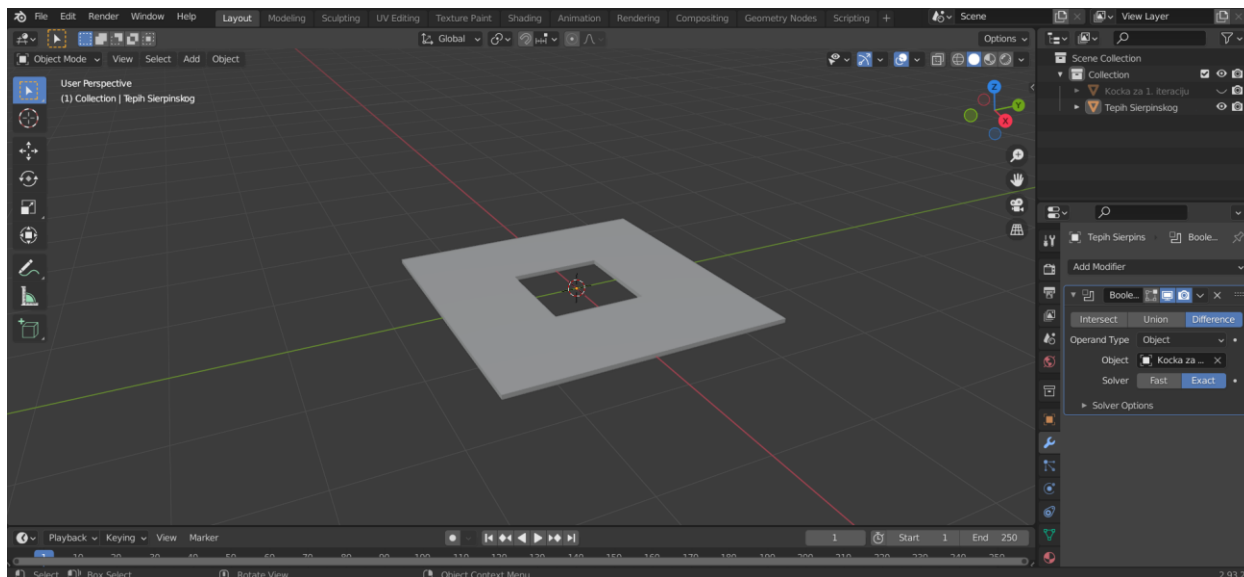


Slika 5.6. Odabrane postavke modifikatora *Boolean*

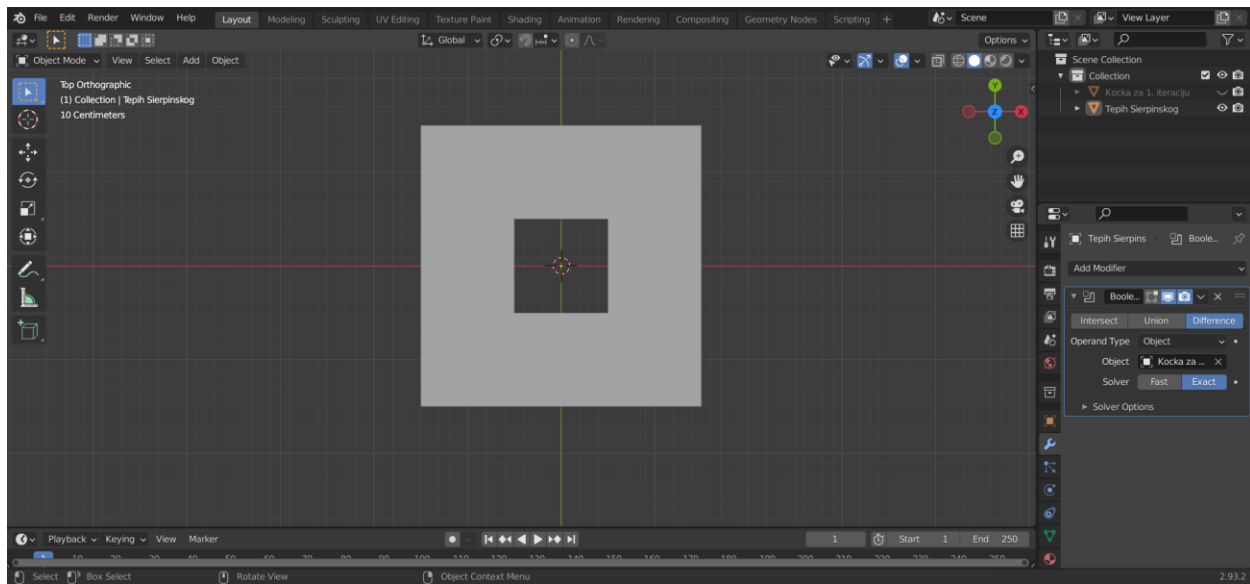
Nakon odabira postavki modifikatora, potrebno ga je primijeniti. To se provodi klikom na ikonu strelice koja se nalazi unutar postavki modifikatora, iznad riječi *Difference*. Pritiskom na tu ikonu, pojavit će se opcije *Apply* i *Duplicate*. Pritiskom na *Apply*, primijenit će se odabrani modifikator.

3. Rad s objektom i spremanje 3D projekta na računalo

Kako bi se vidjela napravljena rupa, unutar *Scene Collection* potrebno je ugasiti vidljivost elementa „Kocka za 1. iteraciju“. To se postiže klikom na ikonu oka kraj „Kocka za 1. iteraciju“ unutar *Scene Collection*. Element na sceni „Tepih Sierpińskog“ upravo predstavlja prvu iteraciju fraktala Tepiha Sierpińskog, a prikazan je na slici 5.7. Treba napomenuti da element „Kocka za 1. iteraciju“ u ovom trenutku više nije potreban. Stoga ga je potrebno izbrisati prije pripreme modela za 3D printanje. Izrađen 3D projekt potrebno je spremiti na računalo.



Slika 5.7. Model prve iteracije Tepiha Sierpińskog



Slika 5.8. Model prve iteracije, promatran pod pravim kutom

5.1.3. Druga iteracija

1. Dodavanje novog objekta na scenu

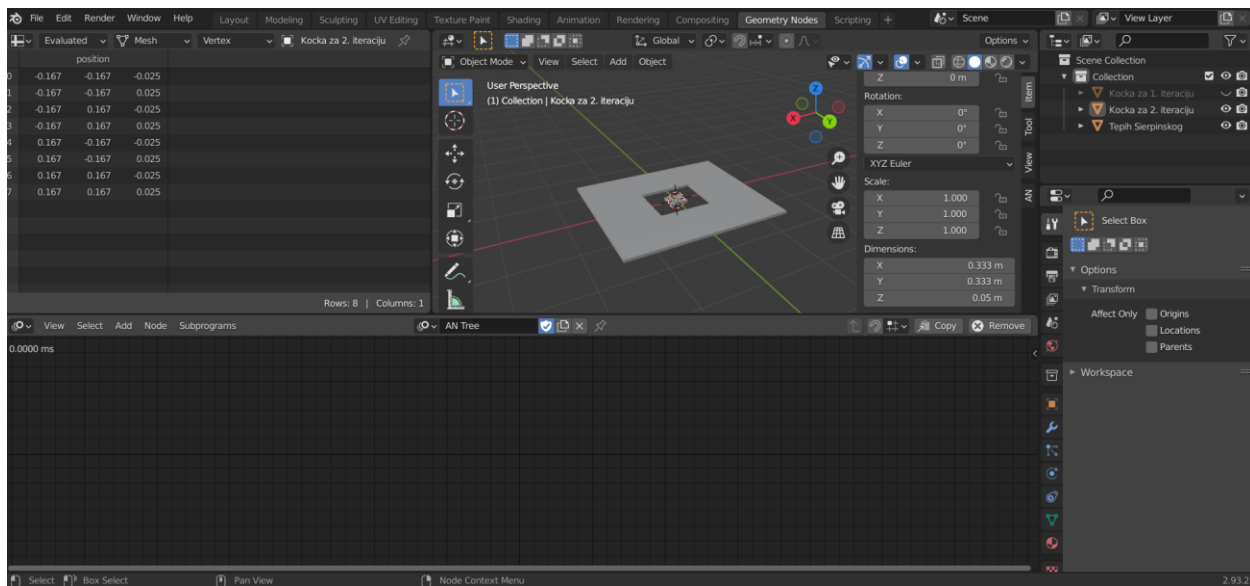
Druga iteracija izradit će se provođenjem određenih radnji nad kreiranom modelu prve iteracije. Stoga je potrebno otvoriti spremljenu 3D datoteku modela prve iteracije. Stvaranje modela druge iteracije postići će se korištenjem Animacijskih čvorova. No, prvo što će se napraviti jest dodavanje nove kocke na scenu, nakon čega će joj se izmijeniti dimenzije u $1/3 \times 1/3 \times 0.05$ m. Preimenovat će se u „Kocka za 2. iteraciju“ te normalizirati njezine dimenzije. To se izvodi na jednak način koji je opisan pri izradi prve iteracije ovog fraktala. Ideja je da se upravo pomoću te „kocke“ izrade dodatne rupe koje nedostaju kako bi element scene „Tepih Sierpińskog“ poprimio izgled druge iteracije Tepiha Sierpińskog.

2. Dodavanje novog radnog prostora

Kako bi se mogli koristiti Animacijski čvorovi, potrebno je dodati novi radni prostor (slika 5.9.).

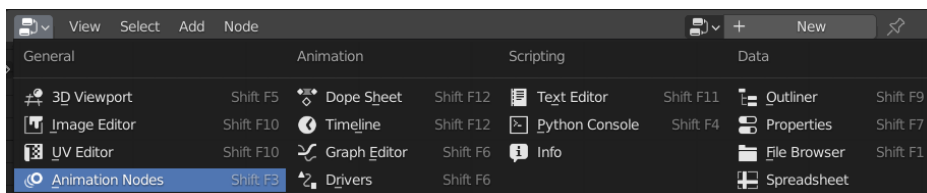


Slika 5.9. Dodavanje novog radnog prostora



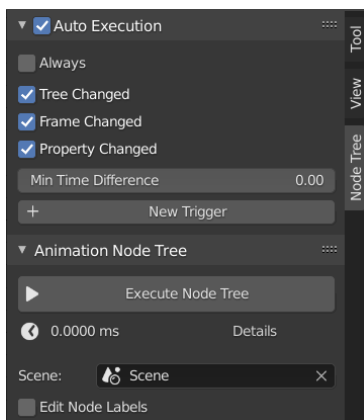
Slika 5.10. Izgled sučelja nakon dodavanja novog radnog prostora

Nadalje, potrebno je odabrati vrstu uređivača (engl. *Editor type*) u gornjem lijevom kutu novog radnog prostora (slika 5.11.). Između ponuđenog, odabrat će se Animacijski čvorovi.



Slika 5.11. Odabir vrste uređivača

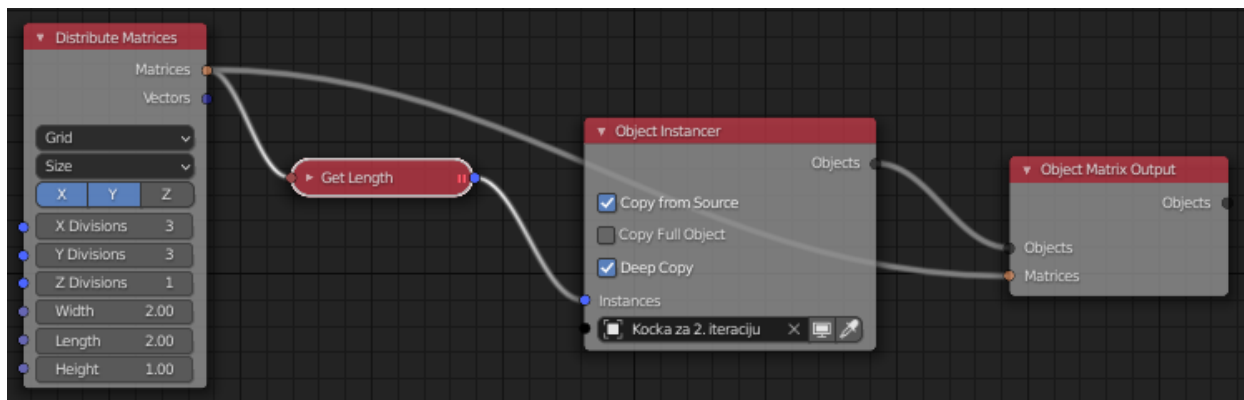
Kako bi se moglo stvoriti stablo čvorova, potrebno je kliknuti na *Add* → *New Node Tree*. Još samo preostaje odabrati načine na koje će Animacijski čvorovi izvršavati stablo čvorova. Odabrat će se *Tree Changed*, *Frame Changed* i *Property Changed* (slika 5.12.).



Slika 5.12. Odabir načina izvršenja stabla čvorova

3. Gradnja stabla Animacijskih čvorova

Unutar prethodno dodanog radnog prostora izgradit će se stablo Animacijskih čvorova, a ono se sastoji od sljedećih čvorova: *Distribute Matrices*, *Get Length*, *Object Instancer* te *Object Matrix Output*. Nakon što su svi dodani unutar radnog prostora, čvorove je potrebno rasporediti i spojiti prema slici 5.13. te odabrati prikazane postavke.

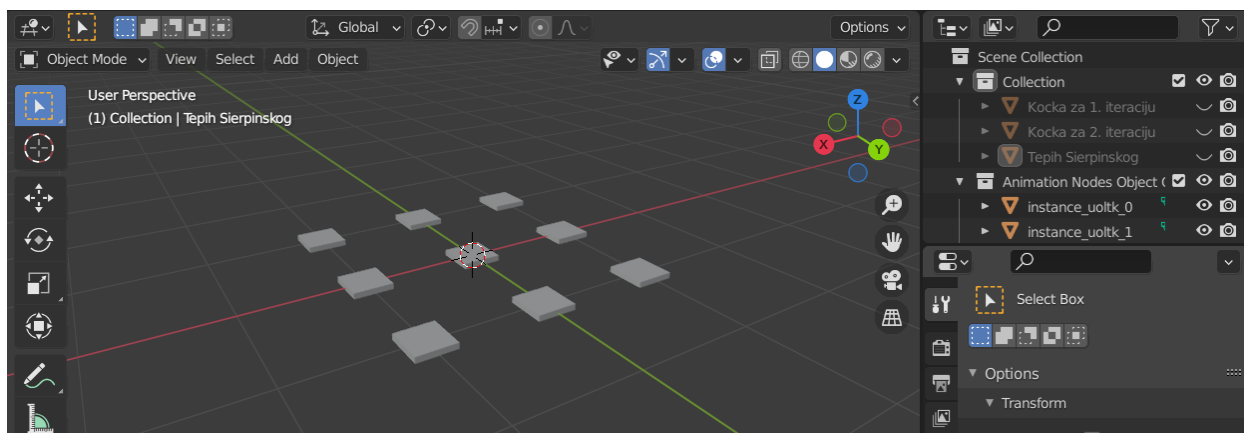


Slika 5.13. Stablo čvorova za drugu iteraciju

Preostaje izvršiti još nekoliko koraka kako bi se izradio model druge iteracije Tepiha Sierpińskog. Unutar čvora *Object Instancer* potrebno je unijeti „Kocka za 2. iteraciju“.

4. Rad sa stvorenim objektima i spremanje 3D projekta na računalo

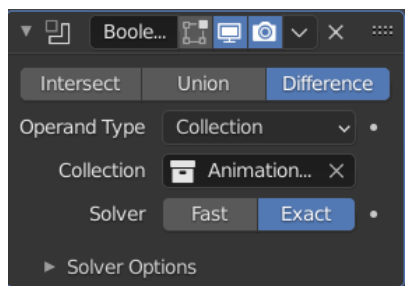
Nakon postavljanja „Kocka za 2. iteraciju“ u čvor *Object Instancer*, unutar *Scene Collection* pojavit će se folder *Animation Nodes Object Container*. Kako bi se dobio uvid što on točno predstavlja, potrebno je isključiti vidljivost svih ostalih objekata na sceni (slika 5.14.).



Slika 5.14. Skup objekata stvorenih pomoću Animacijskih čvorova

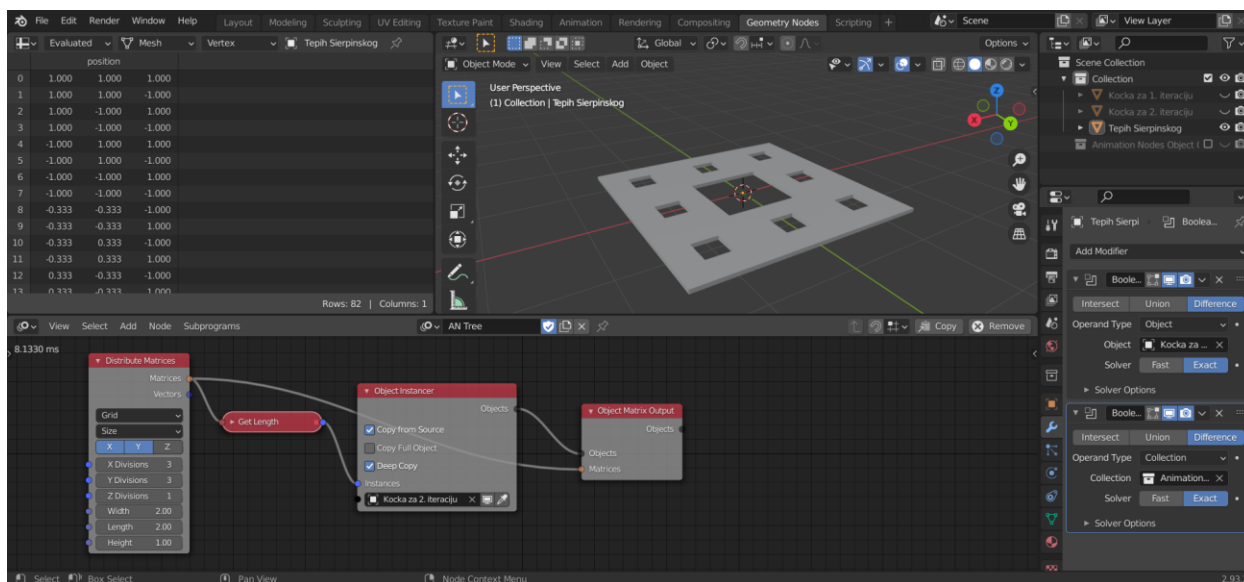
Naime, objekti na slici upravo su objekti koji su generirani pomoću prethodno stvorenog stabla Animacijskih čvorova. Iduće što je potrebno napraviti jest dodavanje modifikatora *Boolean*. Pomoću njega je cilj oduzeti skup stvorenih objekata od objekta „Tepih Sierpińskog“, kako bi se dobila druga iteracija fraktala Tepiha Sierpińskog. Dodavanje modifikatora *Boolean* provest će se

na isti način koji je opisan pri izradi prve iteracije. Ne smije se zaboraviti da je prije dodavanja modifikatora potrebno označiti objekt „Tepih Sierpińskog“ unutar *Scene Collection*, jer će se od njega vršiti oduzimanje. Nakon što je dodan modifikator, potrebno je odabrati postavke koje su prikazane na slici 5.15. Bitno je napomenuti da će se u ovom slučaju oduzimati skup objekata, stoga je pod *Operand Type* potrebno odabrati *Collection*.

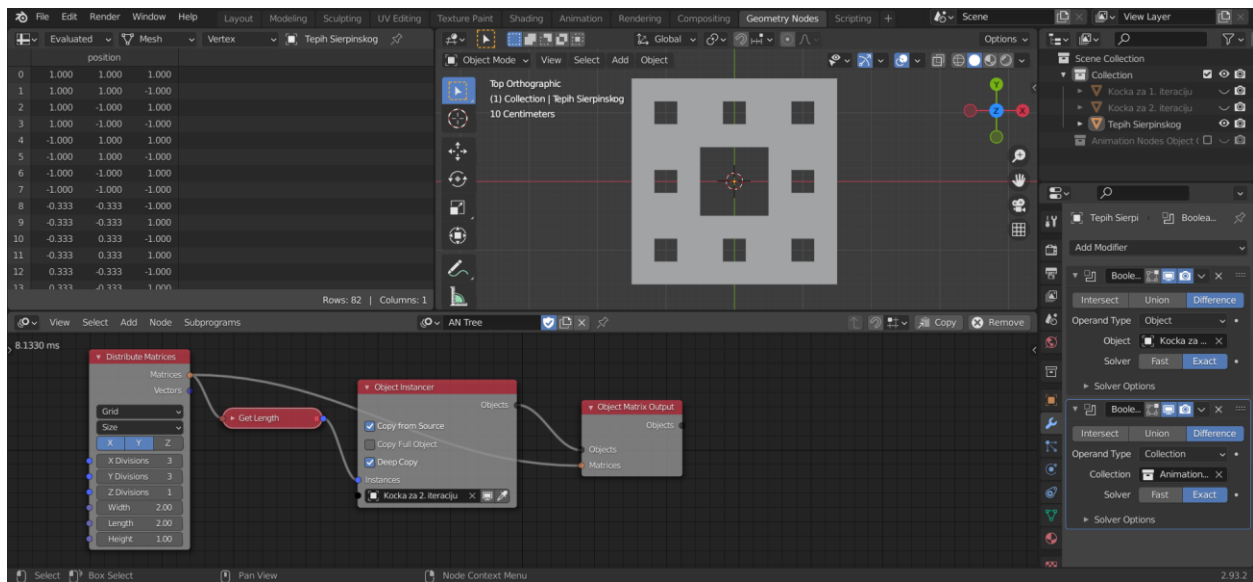


Slika 5.15. Postavke modifikatora Boolean

Nakon odabranih postavki modifikatora, dobit će se rupe druge iteracije fraktala Tepiha Sierpińskog. Preostaje još samo unutar *Scene Collection* isključiti folder *Animation Nodes Object* te vidljivost objekata „Kocka za 1. iteraciju“ i „Kocka za 2. iteraciju“. Na sceni će jedino ostati vidljiv objekt „Tepih Sierpińskog“ koji je poprimio izgled druge iteracije fraktala Tepiha Sierpińskog (slika 5.16.). Elementi „Kocka za 1. iteraciju“ i „Kocka za 2. iteraciju“ u ovom trenutku više nisu potrebni. No, ne smije ih se još brisati jer su potrebni za izradu modela daljnjih iteracija. Izrađen 3D projekt potrebno je spremiti na računalo. Prije pripreme modela „Tepih Sierpińskog“ za 3D printanje potrebno je primijeniti modifikator te zatim izbrisati sve ostale objekte sa scene.



Slika 5.16. Izgled sučelja nakon izrade modela druge iteracije Tepiha Sierpińskog

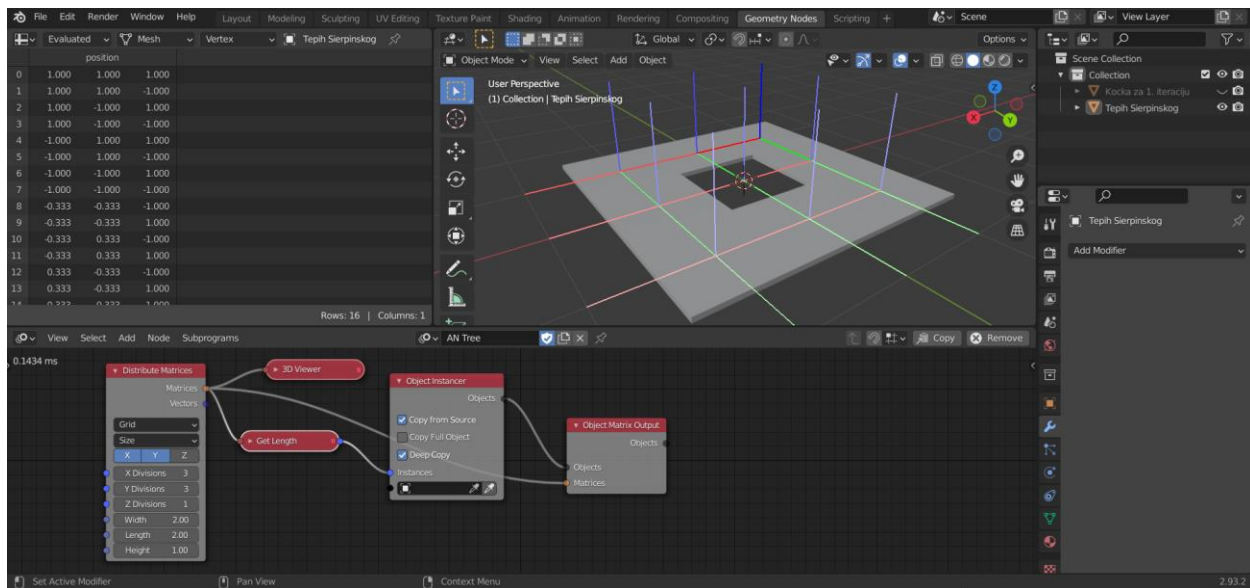


Slika 5.17. Model druge iteracije, promatran pod pravim kutom

5. Objašnjenje načina na koji je izgrađeno stablo Animacijskih čvorova

Za objašnjenje načina na koji je izgrađeno stablo biti će od velike koristi čvor *3D Viewer*. Taj čvor će se stoga dodati uz kreirano stablo. Izlaz *Matrices* čvora *Distribute Matrices* spojiti će se na ulaz toga čvora, pri čemu će se dobiti vizualizacija onoga što on obavlja (slika 5.18.). Naime, čvor *Distribute Matrices* stvara listu matrica koje predstavljaju lokacije buduće stvorenih objekata, koji će biti raspoređeni na određen način u prostoru. U ovom slučaju ti objekti biti će izgledom identični objektu „Kocka za 2. iteraciju“.

Kao podsjetnik, na početku je navedeno da je ideja upravo pomoću te „kocke“ izraditi dodatne rupe koje nedostaju kako bi element scene „Tepih Sierpińskog“ poprimio izgled druge iteracije Tepiha Sierpińskog. Odabrana opcija *Grid* čvora *Distribute Matrices* raspoređuje matrice na 3D rešetki, a sve se to odvija na modelu prve iteracije. Dimenzije rešetke zadane su na taj način zbog dvaju uvjeta; prvi je da udaljenost između svake dvije susjedne lokacije mora biti jednaka, a drugi je raspoređivanje tih devet lokacija unutar 3 retka i 3 stupca.



Slika 5.18. Vizualizacija uz pomoć čvora 3D Viewer: čvor *Distribute Matrices*

Čvor *Get Length* na temelju odabranih dimenzija rešetke i broja matrica računa koliko je objekata potrebno stvoriti. Čvor *Object Instancer* dinamički stvara i kopira objekt. Budući da je cilj stvoriti objekte jednake veličine kao što je „Kocka za 2. iteraciju“, upravo taj element navodi se unutar čvora *Object Instancer*. Čvor *Object Matrix Output* transformira ulazni objekt („Kocka za 2. iteraciju“) po svojstvima transformacije ulazne matrice. Drugim riječima, objekt „Kocka za 2. iteraciju“ kopira se potreban broj puta i zatim se raspoređuje po prethodno definiranim lokacijama. Time se dobiva skup objekata prikazanih na slici 5.14.

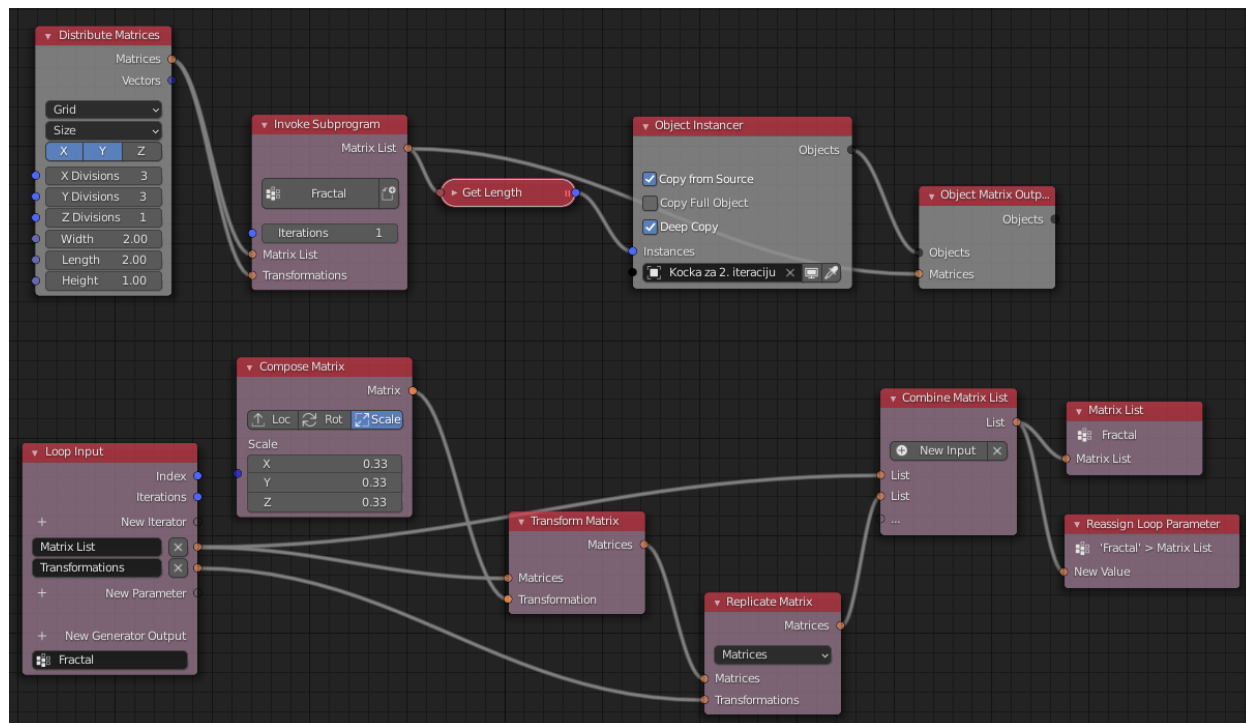
5.1.4. Treća iteracija

Model treće iteracije Tepiha Sierpińskog izradit će se provođenjem određenih transformacija na kreiranom modelu druge iteracije. Stoga je potrebno otvoriti prethodno spremljenu 3D datoteku modela druge iteracije. Treća iteracija Tepiha Sierpińskog također će se izraditi pomoću Animacijskih čvorova. Ovaj put nije potrebno dodavanje nove kocke na scenu jer će se za stvaranje rupa svih idućih iteracija koristiti objekt „Kocka za 2. iteraciju“. Nakon otvaranja 3D datoteke modela druge iteracije, odabrat će se sljedeći načini izvršenja stabla: *Tree Changed*, *Frame Changed* i *Property Changed*. Stabla treće iteracije gradit će se pored izgrađenog stabla druge iteracije.

1. Gradnja stabala Animacijskih čvorova

Potrebno je izgraditi dva stabla Animacijskih čvorova. Prvo stablo sastoji se od čvorova *Distribute Matrices*, *Invoke Subprogram*, *Get Length*, *Object Instancer* te *Object Matrix Output*, dok se drugo sastoji od čvorova *Loop Input*, *Compose Matrix*, *Transform Matrix*, *Replicate Matrix*, *Combine Matrix List*, *Matrix List* te *Reassign Loop Parametar*. Nakon što su svi dodani unutar

radnog prostora, čvorove je potrebno rasporediti i spojiti prema slici 5.19. te odabrati prikazane postavke.

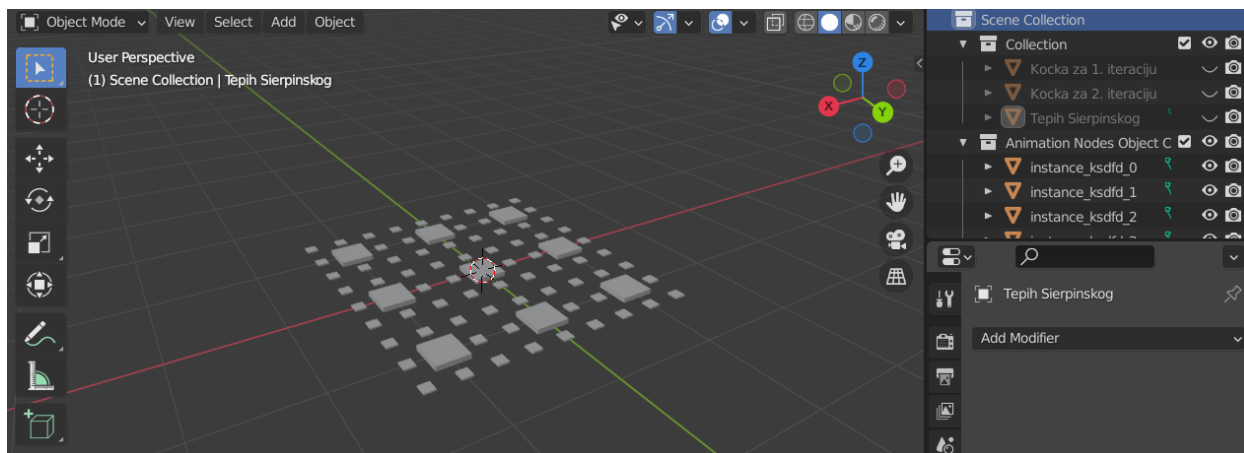


Slika 5.19. Stabla čvorova za treću iteraciju

2. Rad sa stvorenim objektima i spremanje 3D projekta na računalo

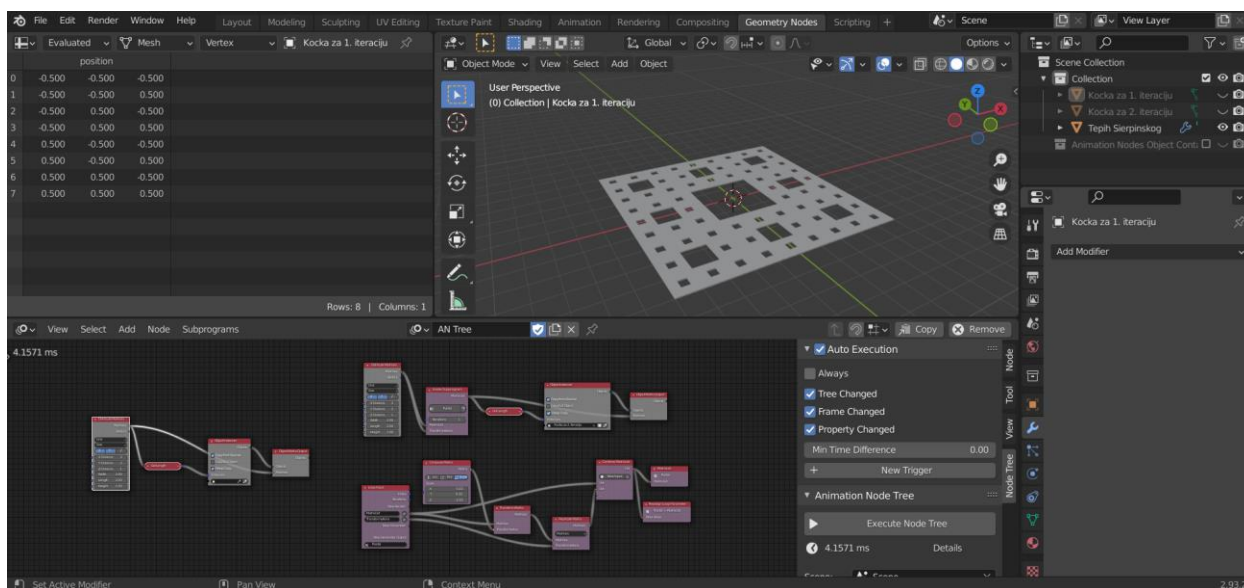
Nakon što su izgrađena stabla prema prethodnim uputama, potrebno je unutar čvora *Object Instancer* unijeti „Kocka za 2. iteraciju“. Zatim je unutar čvora *Invoke Subprogram* potrebno upisati vrijednost 1. Time će se dogoditi promjena na sceni, odnosno unutar foldera *Animation Nodes Object Container* stvorit će se dodatni objekti. Kako bi se dobio uvid što točno predstavlja folder, potrebno je isključiti vidljivost svih ostalih objekata na sceni (slika 5.20.).

Objekti na slici upravo su objekti koji su generirani pomoću prethodno stvorenog stabla Animacijskih čvorova. Nije potrebno dodavati novi modifikator jer će onaj koji je dodan u drugoj iteraciji automatski napraviti potrebne promjene, u ovom slučaju na temelju stabla čvorova treće iteracije.

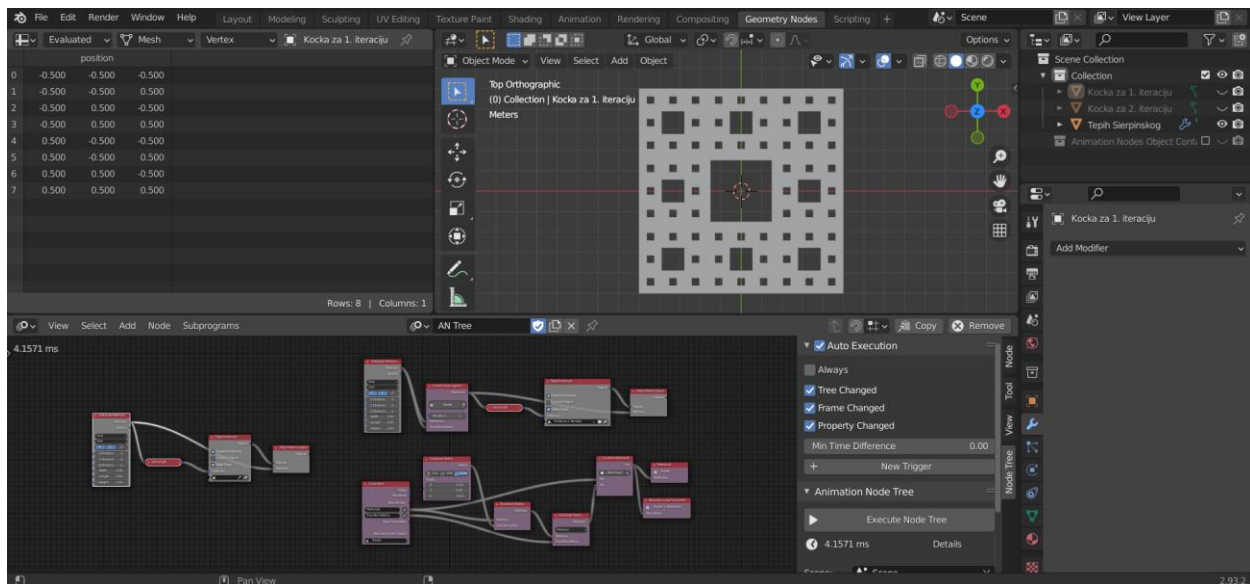


Slika 5.20. Skup objekata stvorenih pomoću Animacijskih čvorova

Unutar *Scene Collection* zatim će se isključiti folder *Animation Nodes Object* te vidljivost objekata „Kocka za 1. iteraciju“ i „Kocka za 2. iteraciju“. Na sceni će jedino ostati vidljiv objekt „Tepih Sierpińskog“, koji je poprimio izgled treće iteracije fraktala Tepiha Sierpińskog (slika 5.21.). Izrađen 3D projekt treba spremiti na računalo. Prije pripreme modela „Tepih Sierpińskog“ za 3D printanje potrebno je primijeniti modifikator te izbrisati sve ostale objekte sa scene.



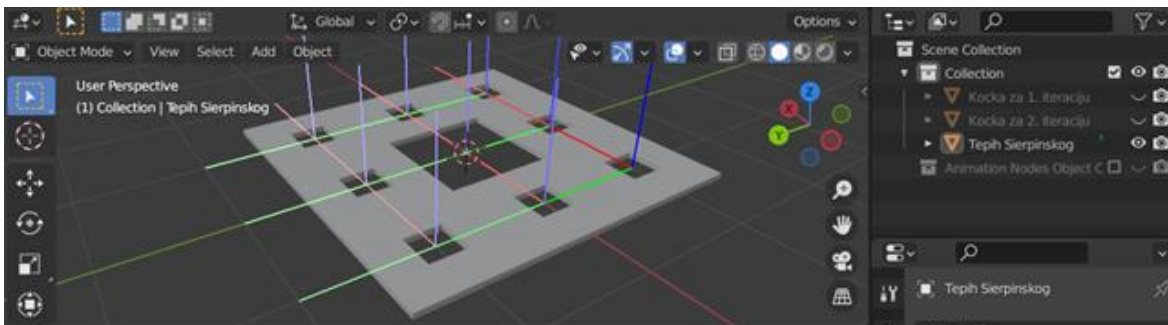
Slika 5.21. Izgled sučelja nakon izrade modela treće iteracije Tepiha Sierpińskog



Slika 5.22. Model treće iteracije, promatran pod pravim kutom

3. Objašnjenje načina na koji su izgrađena stabla Animacijskih čvorova

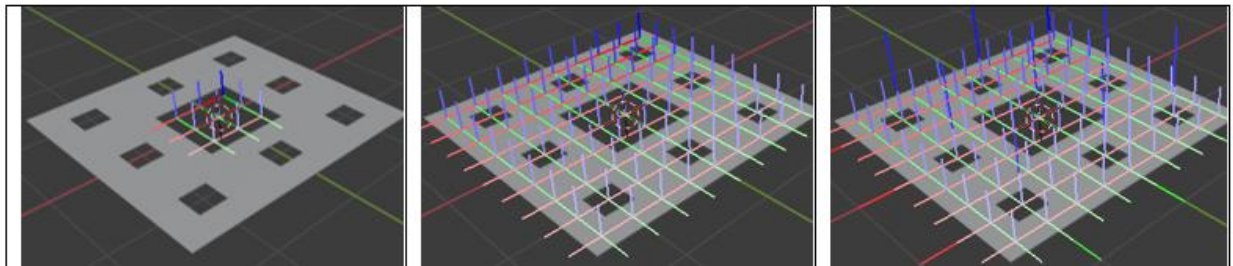
Prilikom objašnjavanja načina na koji su izgrađena stabla koristit će se čvor *3D Viewer*. U prvom stablu čvorova izlaz *Matrices* čvora *Distribute Matrices* potrebno je spojiti na ulaz čvora *3D Viewer*. Budući da su postavke čvora *Distribute Matrices* jednake kao i pri izradi druge iteracije, dobit će se jednake dimenzije rešetke te broj matrica po x i y - osi. Ovaj put sve se odvija na modelu druge iteracije (slika 5.23.).



Slika 5.23. Vizualizacija uz pomoć čvora *3D Viewer*; čvor *Distribute Matrices*

Kao što je vidljivo na slici, devet je stvorenih lokacija. To su lokacije na kojima su stvoreni objekti druge iteracije (slika 5.14.), jednake veličine kao i objekt „Kocka za 2. iteraciju“. Sada je cilj izraditi manje objekte, odnosno oko svake od tih devet lokacija potrebno je izraditi osam manjih. Stabla čvorova prikazana su na slici 5.19. Izlaz *Matrices* čvora *Distribute Matrices* spaja se na ulaze *Matrix List* i *Transformations* potprograma *Invoke Subprogram*. Taj potprogram pokreće novi, a to je *Loop Input*. Ulazi potprograma *Invoke Subprogram* ujedno su izlazi čvora *Loop Input*. Devet lokacija generiranih pomoću čvora *Distribute Matrices* ulaze u potprogram *Invoke Subprogram* i izlaze iz čvora *Loop Input* u drugom stablu. Izlaskom iz tog čvora, u nastavku

drugog stabla provest će se određene transformacije nad tih devet lokacija, s ciljem dobivanja dodatnih manjih objekata. Čvor *Compose Matrix* formira matricu transformacije koja se može koristiti za primjenu određenih transformacija (premještanje, rotacija, skaliranje). U ovom slučaju odabire se skaliranje konstantom $1/3$, jer je cilj kasnije stvoriti objekte trostruko manje veličine od onih koji su dobiveni u drugoj iteraciji. Čvor *Transform Matrix* transformira ulaznu matricu matricom ulazne transformacije. Ulazna matrica je *Matrix List* (izlaz čvora *Loop Input*), a matrica ulazne transformacije je matrica formirana čvorom *Compose Matrix*. Drugim riječima, čvor *Transform Matrix* provest će skaliranje konstantom $1/3$ nad udaljenosti između svake dvije lokacije koje su stvorene pomoću čvora *Distribute Matrices*. Devet lokacija sada se nalaze u središtu Tepiha (slika 4.7.). Čvor *Replicate Matrix* kopira listu početnih devet lokacija, zatim svaku novu kopiju transformira na temelju matrice transformacije (napravi devet trostruko manjih kopija) te dobivene kopije raspoređuje na devet početnih lokacija (definiranih pomoću čvora *Duplicate Matrices*). Rezultat koji se dobije pomoću *Replicate Matrix* čvora može se vidjeti na slici 5.24. Nadalje, čvor *Combine Matrix List* stvara uniju liste koja sadrži devet početnih lokacija te liste koja sadrži one nove, dobivene pomoću čvora *Replicate Matrix*. Dobiveni rezultat čvora *Combine Matrix List* može se vidjeti na slici 5.24.



Slika 5.24. Vizualizacija uz pomoć čvora 3D Viewer; redom čvorovi *Transform Matrix*, *Replicate Matrix*, *Combine Matrix List*

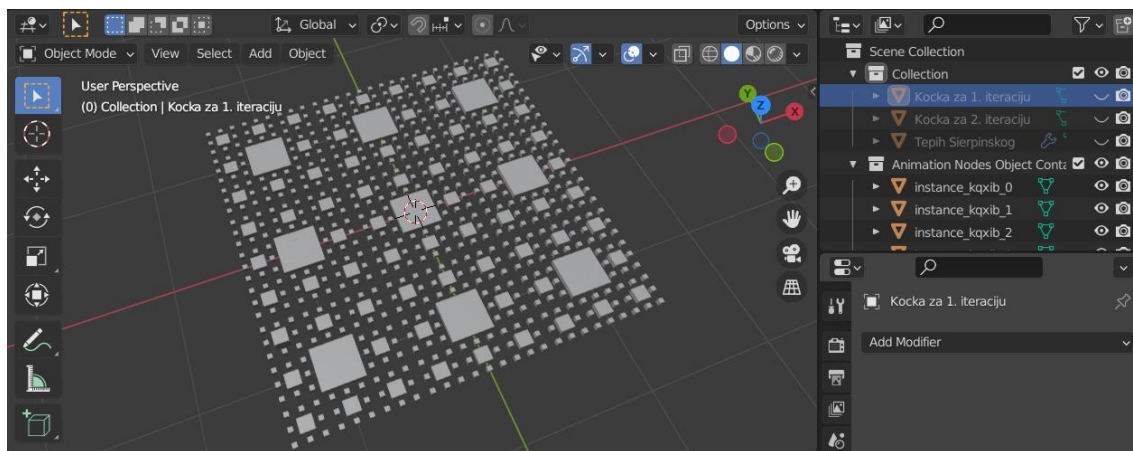
Čvor *Matrix list* je nova lista lokacija, kreirana nakon djelovanja *Combine Matrix List* čvora. Provođenjem daljnjih iteracija, ulaz *Matrix List* čvora sadržavat će sve veći broj lokacija. Čvor *Reassign Loop Parametar* ima funkciju da nakon svake uvećane vrijednosti iteracije čvoru *Combine Matrix List* pridodaje listu prethodno kreiranih lokacija. Potrebno je promotriti ponovo čvor *Matrix List* u drugom stablu. Čvor *Matrix List* je izlaz iz potprograma *Invoke Subprogram*. Izlaz iz potprograma *Invoke Subprogram* spojen je na čvorove koji služe kako bi se izradili odgovarajući objekti, koji će se pridružiti prethodno stvorenim lokacijama. Izlaz *Matrix List* potprograma *Invoke Subprogram* spojen je na ulaz *Instances* čvora *Object Instancer*. Između njih nalazi se čvor *Get Length*. Na temelju liste stvorenih lokacija, čvor *Get Length* računa koliko je objekata potrebno stvoriti.

Čvor *Object Instancer* zatim dinamički stvara objekte i radi njihove kopije. Unutar njega stavlja se element „Kocka za 2. iteraciju“, jer je cilj na temelju unaprijed definiranih transformacija dobiti objekte koji su od njega trostruko manje veličine. Čvor *Object Matrix Output* transformira ulazne objekte (kopije objekata koje je stvorio čvor *Object Instancer*) po matrici ulaznih

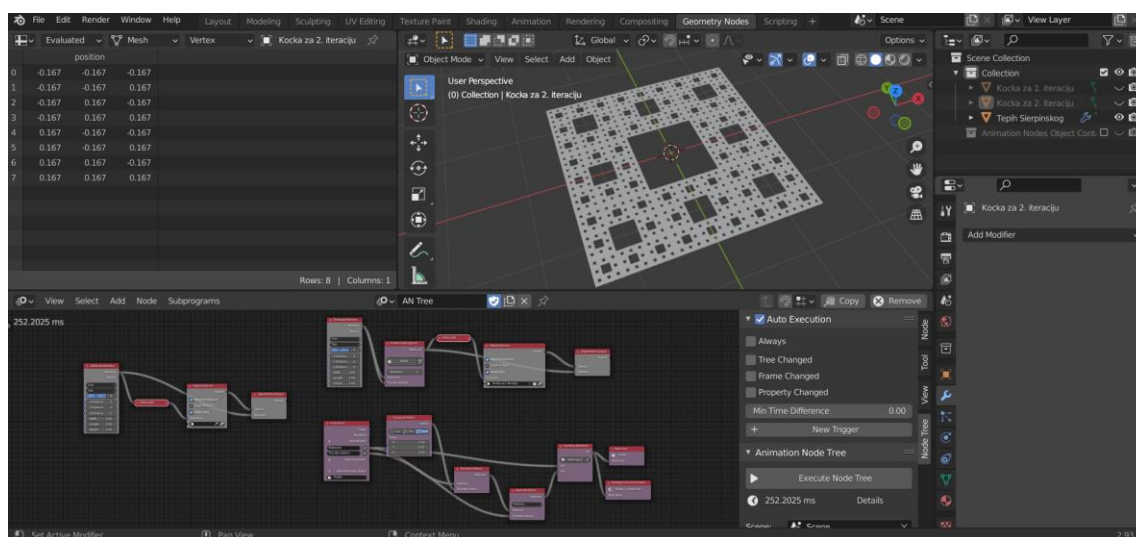
transformacija (liste matrica, odnosno lokacija). Jednostavnije rečeno, taj čvor na temelju prethodno stvorenih lokacija provodi transformaciju odgovarajućih objekata te ih zatim pridružuje pripadajućim lokacijama. Unutar ulaza *Iterations* potprograma *Invoke Subprogram* upisana je vrijednost 1, što znači da će se petlja izvršiti jedanput. Time će se stvoriti željeni objekti, prikazani na slici 5.20.

5.1.5. Četvrta iteracija i nadalje

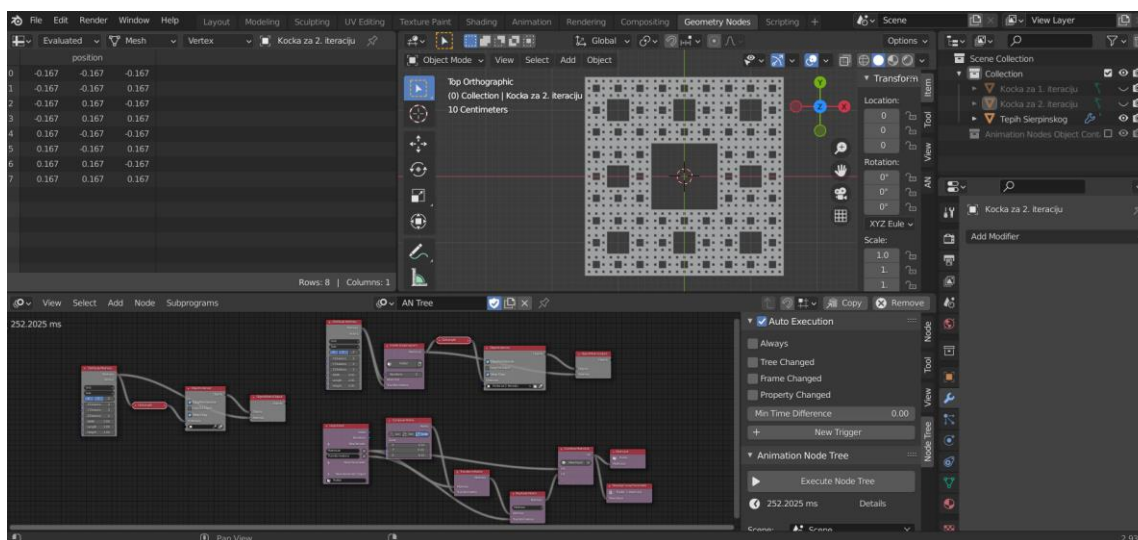
Model četvrte iteracije izrađuje se na jednak način kao i onaj treće iteracije. Jedino što je potrebno promijeniti jest da se unutar ulaza *Iterations* potprograma *Invoke Subprogram* umjesto vrijednosti 1 upiše vrijednost 2. Prilikom izrade četvrte iteracije, Blenderu će biti potrebno tri do pet minuta da izvrši stablo i izradi potrebne objekte. Daljnje iteracije moguće je izraditi na jednak način; za petu iteraciju postavila bi se vrijednost 3 itd. No nažalost, to je vrlo teško izvedivo zbog preopterećenog procesora. Slike izrađenog modela četvrte iteracije mogu se vidjeti u nastavku.



Slika 5.25. Skup objekata stvorenih pomoću Animacijskih čvorova



Slika 5.26. Izgled sučelja nakon izrade modela četvrte iteracije Tepiha Sierpińskog



Slika 5.27. Model četvrte iteracije, promatran pod pravim kutom

5.2. Rezanje u PrusaSlicer-u

Kako bi se mogli izrezati gotovi 3D modeli, potrebno je otvoriti svaku od spremljenih .blend datoteka te izvesti STL datoteku odabirući *File* → *Export* → *Stl* (.stl) u gornjem lijevom kutu programa. Unutar PrusaSlicer programa treba uvesti .stl format datoteke. Biraju se odgovarajuće postavke printanja. Rezanje modela provodi pritiskom na opciju *Slice now* unutar softvera. Preostaje izvesti datoteku G - koda, pritiskom na opciju *Export G-code*.

5.3. 3D printanje pomoću Prusa i3 MK3S 3D printera i naknadna obrada

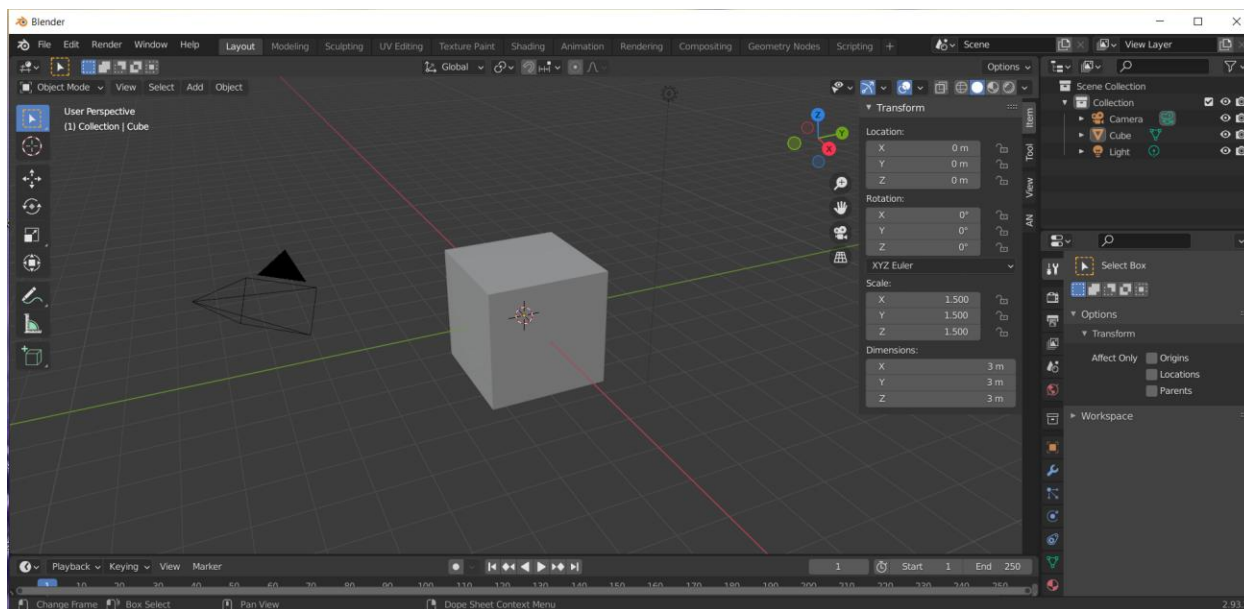
Slijedi 3D printanje modela. Nakon što je bio isprintan 3D objekt treće iteracije, niti jedan oblik naknadne obrade nije bio potreban.

6. PRAKTIČNI DIO IZRADE MENGEROVE SPUŽVE I NJEZINOJ NEGATIVA POMOĆU 3D PRINTERA

6.1. 3D modeliranje u Blenderu

6.1.1. Nulta iteracija

Kako bi se moglo započeti s radom, potrebno je otvoriti novu Blender datoteku. Kamera i svjetlo mogu se izbrisati, a način na koji se to provodi opisan je pri izradi fraktala Tepiha Sierpińskog. Kocki na sceni odabrat će se dimenzije 3 x 3 x 3 m te će se unutar *Scene Collection* objekt preimenovati u „Mengerova spužva“. Gotov model prikazan je na slici 6.1. - riječ je o nultoj iteraciji istoimenog fraktala. Kreiran 3D projekt potrebno je spremiti na računalo.



Slika 6.1. Gotov model nulte iteracije Mengerove spužve

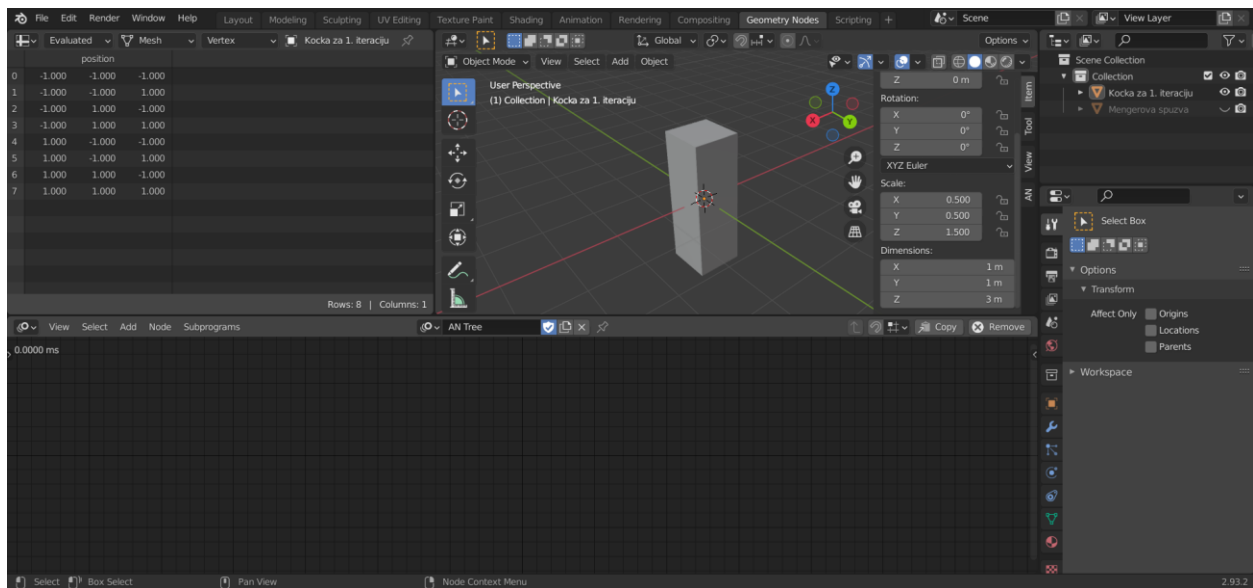
6.1.2. Prva iteracija

1. Dodavanje novog objekta na scenu

Model prve iteracije Mengerove spužve izradit će se provođenjem odgovarajućih radnji na prethodno izrađenom modelu nulte iteracije. Stoga je potrebno otvoriti spremljenu 3D datoteku modela nulte iteracije. Na scenu se dodaje kocka, kojoj se mijenjaju dimenzije u $1 \times 1 \times 3$ m te joj se dodjeljuje naziv „Kocka za 1. iteraciju“. Nadalje, provest će se normalizacija njezinih dimenzija unutar izbornika *Object*, a opis tog postupka izrečen je prilikom izrade fraktala Tepiha Sierpińskog. Inače, objekt „Kocka za 1. iteraciju“ dodaje se s ciljem stvaranja rupa koje sadrži prva iteracija Mengerove spužve, a može se vidjeti na slici 6.2.

2. Dodavanje novog radnog prostora

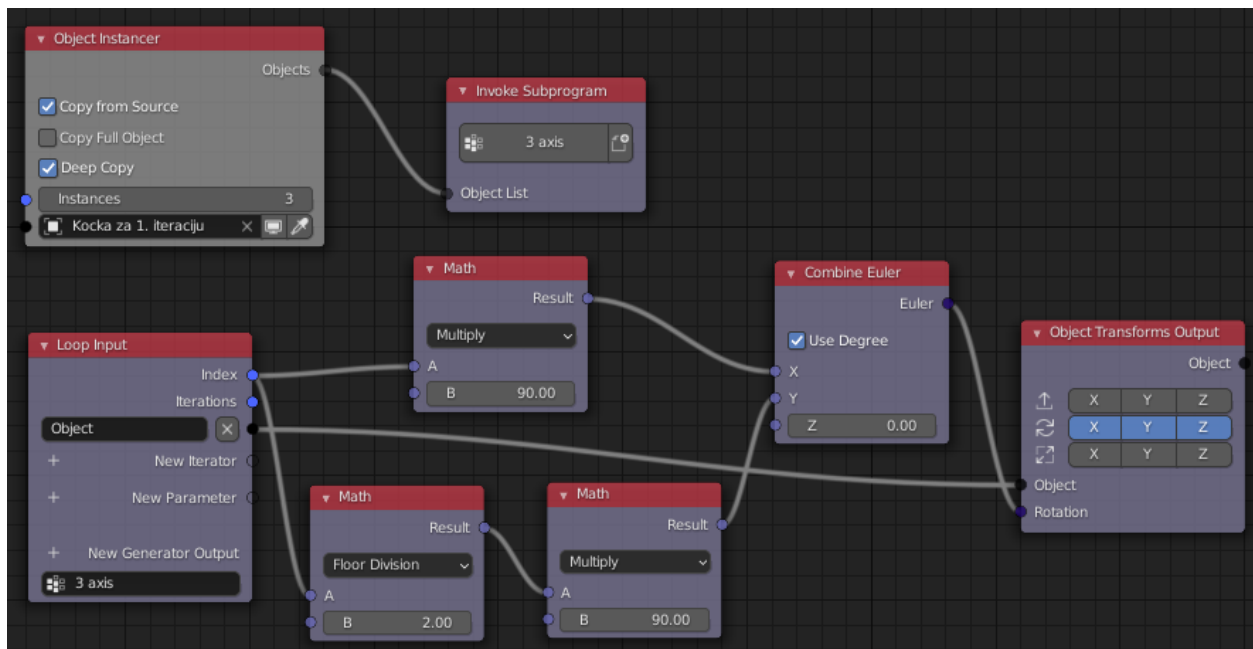
Iduće što je potrebno napraviti jest dodavanje novog radnog prostora, a sam postupak njegovog dodavanja opisan je pri izradi fraktala Tepiha Sierpińskog.



Slika 6.2. Izgled sučelja nakon dodavanja novog radnog prostora

3. Gradnja stabala Animacijskih čvorova

Slijedi gradnja dvaju stabala Animacijskih čvorova. Prvo stablo sadrži *Object Instancer* i *Invoke Subprogram* čvorove, dok drugo stablo sadrži čvorove *Loop Input*, *Combine Euler*, *Object Transforms Output* te tri čvora *Math*. Nakon dodavanja čvorova unutar radnog prostora, potrebno ih je rasporediti prema slici 6.3. te odabrati prikazane postavke.

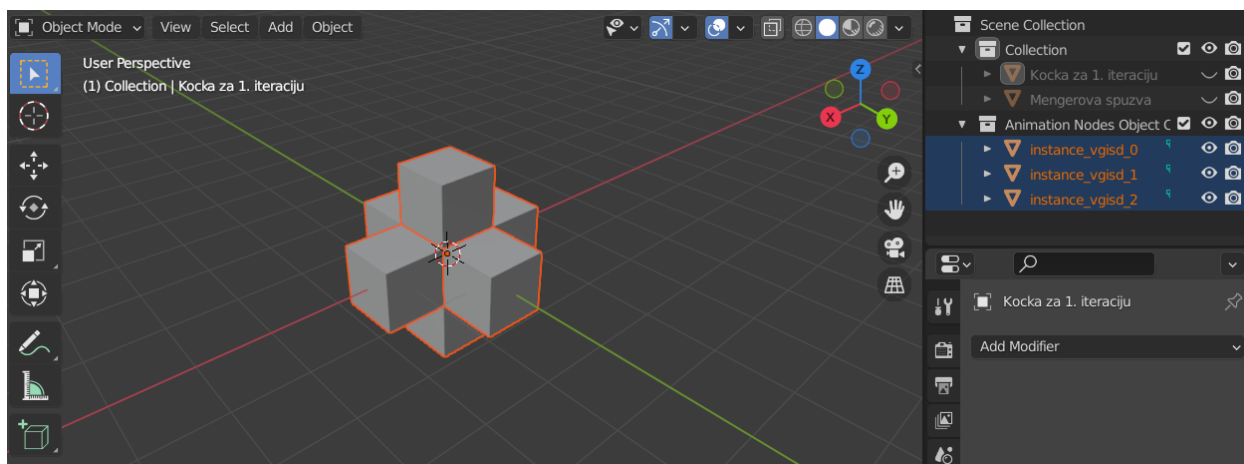


Slika 6.3. Stabla čvorova za prvu iteraciju

Provođenjem daljnjih koraka izradit će se prvo model negativa Mengerove spužve, a zatim i sama Mengerova spužva. Unutar čvora *Object Instancer* potrebno je unijeti „Kocka za 1. Iteraciju”.

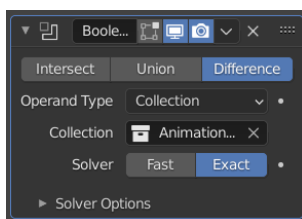
4. Rad sa stvorenim objektima i spremanje 3D projekata na računalo

Nakon upisivanja elementa „Kocka za 1. iteraciju” u čvor *Object Instancer*, unutar *Scene Collection* pojavit će se folder *Animation Nodes Object Container*. Riječ je o folderu koji sadrži tri objekta, stvorenih uz pomoć Animacijskih čvorova. Spajanjem tih objekata uz pomoć Blendrovog alata *Join* stvorit će se model negativa prve iteracije Mengerove spužve. Na sljedećoj slici mogu se vidjeti spomenuti objekti.



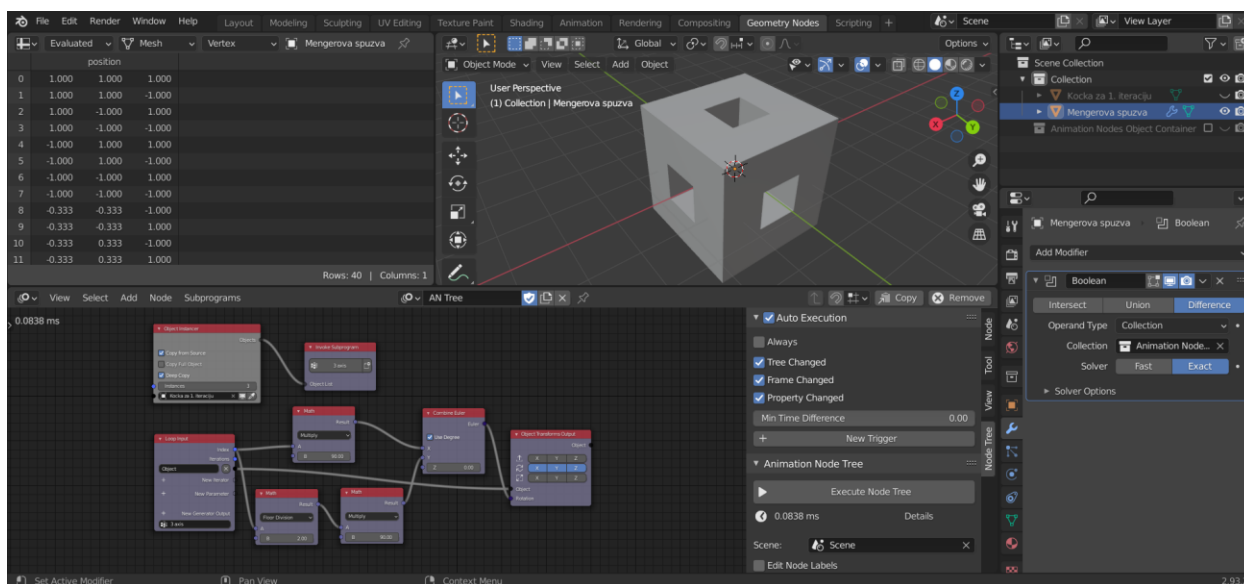
Slika 6.4. Skup objekata, stvorenih pomoću Animacijskih čvorova

Sljedeće što je potrebno napraviti jest dodavanje modifikatora *Boolean*. Uz pomoć tog modifikatora oduzet će se skup objekata stvorenih pomoću Animacijskih čvorova od objekta „Mengerova spužva”. Način dodavanja novog modifikatora opisan je pri izradi fraktala Tepiha Sierpińskog. Prije dodavanja modifikatora, potrebno je označiti objekt „Mengerova spužva” unutar *Scene Collection*. Nakon što je dodan, odabiru se postavke koje su prikazane na slici 6.5.



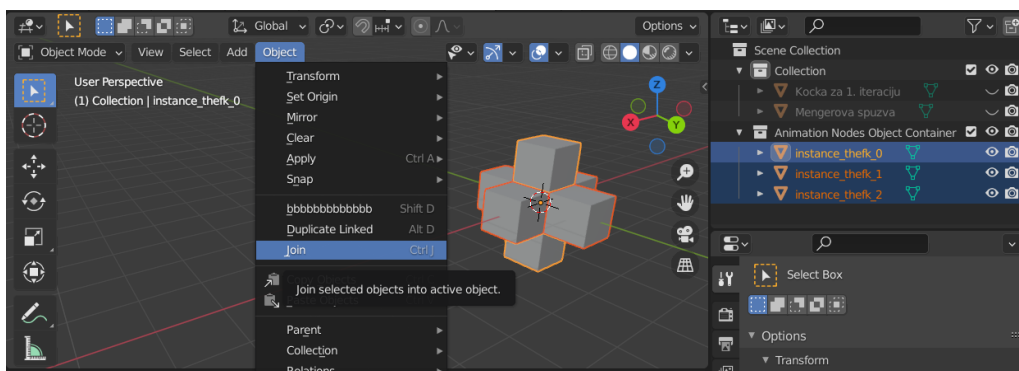
Slika 6.5. Postavke modifikatora *Boolean*

Odabirom navedenih postavki, objekt „Mengerova spužva” na sceni poprimit će izgled prve iteracije poznatog fraktala. Kako bi ga bilo moguće vidjeti, potrebno je isključiti vidljivost svih ostalih objekata na sceni (slika 6.6.).



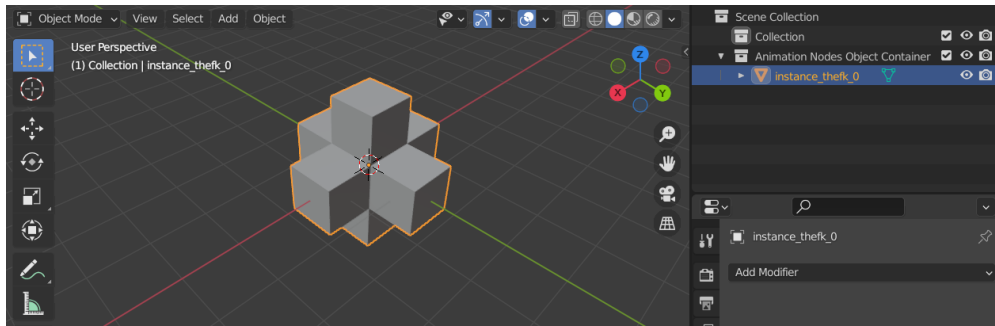
Slika 6.6. Izgled sučelja nakon izrade modela prve iteracije Mengerove spužve

Stvoreni 3D projekt može se spremiti na računalo, npr. pod nazivom *Mengerova spužva i negativ 1*. Preostaje se upoznati s koracima koje je potrebno provesti kako bi se kasnije oba modela (objekt „Mengerova spužva” i njegov negativ) mogla pohraniti u obliku STL datoteke. Krenut će se s radom na negativu. Prvo je potrebno označiti objekte koji su stvoreni uz pomoć Animacijskih čvorova, zatim unutar izbornika *Object* odabrati opciju *Join* (slika 6.7.) Opisanim postupkom dobit će se jedinstveni objekt koji predstavlja model negativa prve iteracije Mengerove spužve.



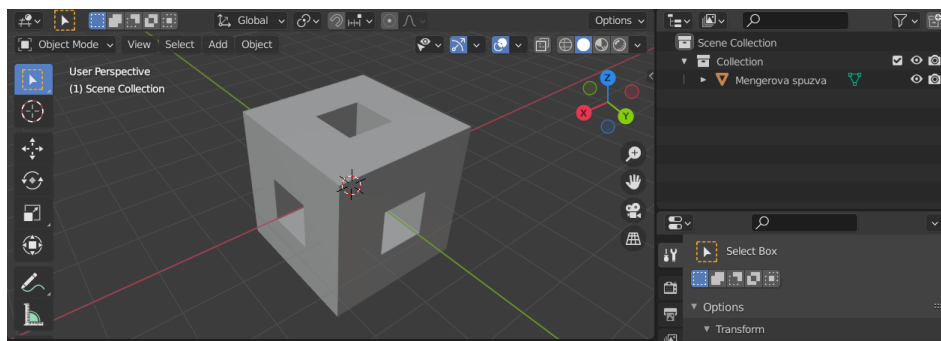
Slika 6.7. Postupak spajanja objekata dobivenih pomoću Animacijskih čvorova

Nadalje, unutar *Scene Collection* potrebno je izbrisati objekte „Kocka za 1. iteraciju” te „Mengerova spužva”. Time će na sceni ostati jedino objekt koji je namijenjen za 3D printanje (slika 6.8.). Kreirani 3D projekt potrebno je spremiti na računalo, na primjer pod nazivom *Negativ Mengerove spužve 1*.



Slika 6.8. Gotov model negativa prve iteracije Mengerove spužve

Slijedi spremanje projekta s modelom Mengerove spužve. Potrebno je otvoriti spremljeni dokument *Mengerova spužva i negativ 1*. Unutar njega treba primijeniti prethodno dodani modifikator, a zatim unutar *Scene Collection* izbrisati objekt „Kocka za 1. iteraciju” te objekte dobivene pomoću Animacijskih čvorova. Time će na sceni ostati jedino objekt kojeg je cilj printati, a to je upravo prva iteracija Mengerove spužve. 3D projekt preostaje spremiti na računalo, na primjer pod nazivom *Mengerova spužva 1*.



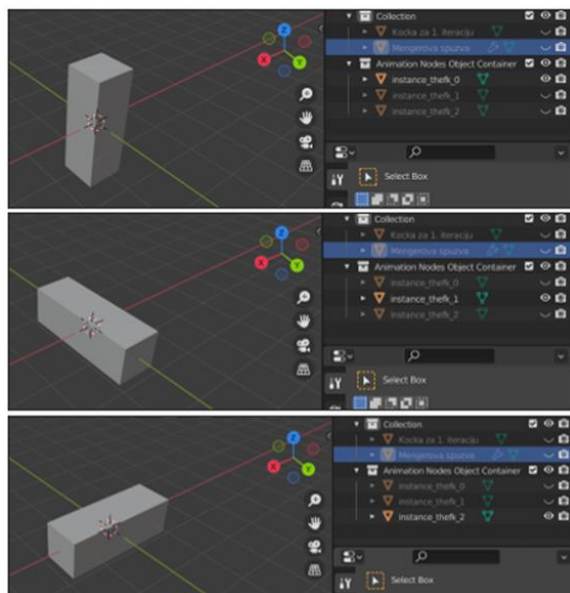
Slika 6.9. Gotov model prve iteracije Mengerove spužve

5. Objašnjenje načina na koji su izgrađena stabla Animacijskih čvorova

Pomoću stabala čvorova (slika 6.3.) stvorena su tri objekta (slika 6.4.). Spajanjem tih objekata izrađen je model negativa prve iteracije Mengerove spužve. U nastavku će se objasniti način na koji su stvorena ta tri objekta. Općenito, čvor *Object Instancer* dinamički stvara i kopira određeni objekt. Unutar čvora *Object Instancer* upisani su element „Kocka za 1. iteraciju” i vrijednost 3, budući da je cilj izraditi tri kopije objekta „Kocka za 1. iteraciju”. Stvorenim kopijama program samostalno dodjeljuje nazive.

Potprogram *Invoke Subprogram* pokreće novi, a to je *Loop Input*. Ulaz potprograma *Invoke Subprogram* ujedno je izlaz čvora *Loop Input*. Tri kopije koje su stvorene pomoću čvora *Object Instancer* ulaze u potprogram *Invoke Subprogram* i izlaze iz čvora *Loop Input* u drugom stablu. U nastavku drugog stabla cilj je provesti rotaciju stvorenih objekata, a ona se provodi uz pomoć određenih uvjeta. Ti uvjeti definirani su upravo pomoću čvorova *Math*. No prije nego što se objasni daljnji postupak, potrebno je promotriti sliku 6.10. Slika prikazuje tri kopije objekta „Kocka za 1. iteraciju” nakon što je nad pojedinima provedena rotacija. Desno od svakog objekta, unutar *Scene*

Collection moguće je pročitati njegov puni naziv, dodijeljen od strane samog programa. U nastavku, kada će se spominjati pojedini objekt, govorit će se o objektu (0), (1) ili (2) - na slici 6.10. moguće ih je vidjeti redom, odozgo prema dolje.



Slika 6.10. Rotirane kopije objekta „Kocka za 1. iteraciju”; odozgo prema dolje objekti (0), (1) i (2)

Izlaz *Index* čvora *Loop Input* spojen je na ulaz A čvora *Math*, koji obavlja matematičku operaciju množenja *Multiply*. Riječ je o tome da čvor *Loop Input* čvoru *Math* redom „dostavlja” indekse (0, 1, 2) stvorenih objekata, dok čvor *Math* na temelju svakog indeksa obavlja operaciju množenja. Ulazi A i B čvora *Math* predstavljaju faktore – ulaz A sadrži indekse 0, 1, 2, dok ulaz B sadrži vrijednost 90. Ovisno o dobivenom umnošku, pojedini objekt će se rotirati ili ne. Konkretno, za objekt (0) provodi se množenje; $0 \cdot 90 = 0$, za objekt (1); $1 \cdot 90 = 90$, za objekt (2); $2 \cdot 90 = 180$. Na temelju izračunatog umnoška, u nastavku stabla provest će se rotacija objekta (1) za 90° te objekta (2) za 180° oko x - osi. Izgled rotiranog objekta (1) može se vidjeti na slici 6.10.

Sljedeći čvor kojeg je potrebno objasniti jest čvor *Math*, koji obavlja operaciju dijeljenja bez ostatka *Floor Division*. On funkcionira na isti način kao i prethodno opisani čvor *Math*, samo što obavlja drugačiju operaciju. Za objekt (0) provodi se dijeljenje $0 : 2 = 0$, za objekt (1); $1 : 2 = 0$, za objekt (2); $2 : 2 = 1$. Budući da tijekom dijeljenja nije dobiven ostatak jedino kada je bilo riječ o objektu (2), u nastavku stabla provodit će se rotacija upravo tog objekta. Kako je cilj rotirati taj objekt za 90° , izlaz čvora *Math* s operacijom *Floor Division* spojen je na ulaz čvora *Math* s operacijom *Multiply*. Nakon provedene operacije množenja *Multiply* ispostaviti će se da je potrebna rotacija objekta (2) za 90° oko y - osi. Na slici 2.9. moguće je vidjeti izgled rotiranog objekta (2). Čvor *Combine Euler* na temelju izračunatih umnožaka definira rotaciju objekata (1) i (2), dok čvor *Object Transform Output* provodi prethodno definiranu rotaciju.

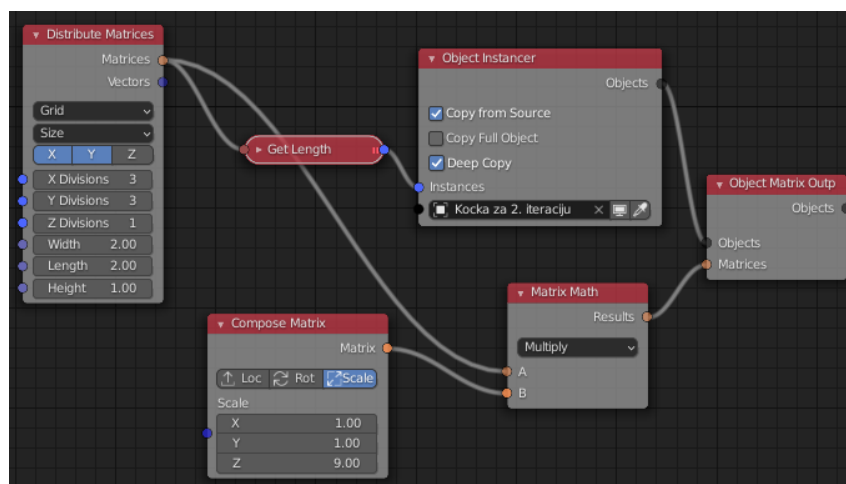
6.1.3. Druga iteracija

1. Dodavanje novog objekta na scenu

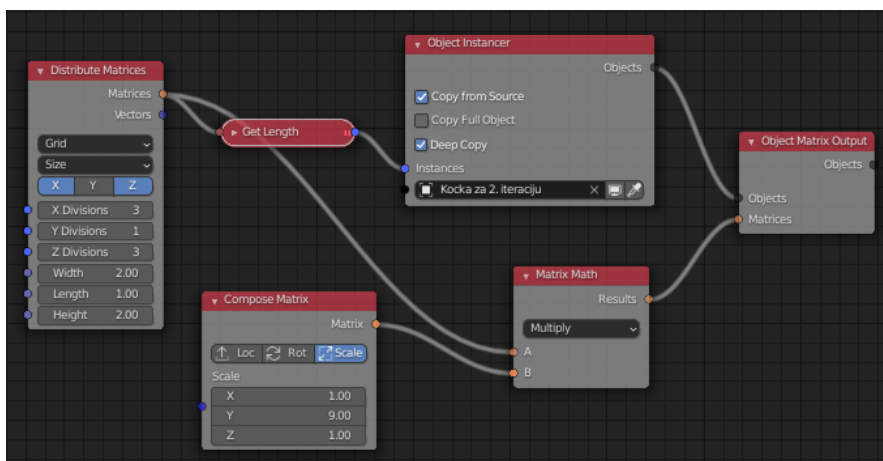
Izrada modela druge iteracije Mengerove spužve započinje unutar spremljenog 3D projekta *Mengerova spužva i negativ 1*. Stoga je potrebno otvoriti spomenutu 3D datoteku. Na scenu se dodaje nova kocka kojoj se izmjenjuju dimenzije u $1/3 \times 1/3 \times 1/3$ m te naziv u „Kocka za 2. iteraciju”. Objekt „Kocka za 2. iteraciju” dodaje se kako bi se stvorile rupe koje sadrži druga iteracija Mengerove spužve. Ponovo će se koristiti Animacijski čvorovi. Stabla druge iteracije gradit će se desno od stabala prve iteracije.

2. Gradnja stabala Animacijskih čvorova

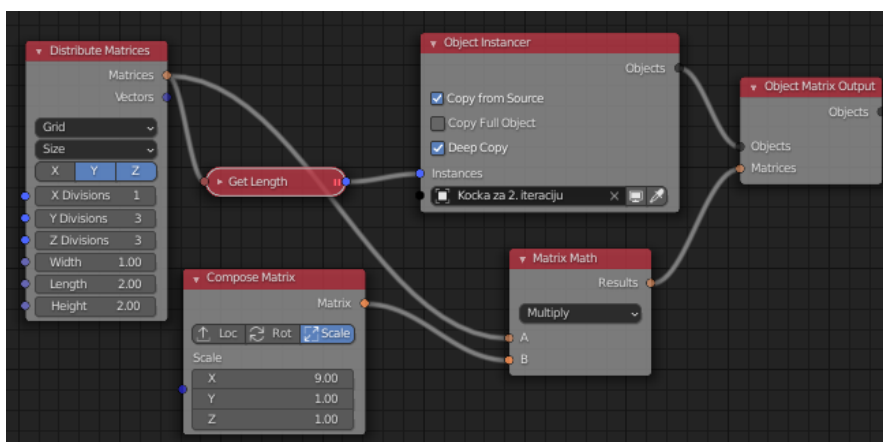
Slijedi gradnja triju stabala Animacijskih čvorova. Kreće se s gradnjom prvog stabla, a nakon što se ono izgradi, na isti način je potrebno izgraditi preostala dva. Sva tri stabla sadrže čvorove *Distribute Matrices*, *Get Length*, *Object Instancer*, *Compose Matrix*, *Matrix Math* te *Object Matrix Output*. Nakon što su dodani unutar radnog prostora, čvorove je potrebno rasporediti i spojiti prema slici 6.11. te unutar njih odabrati prikazane postavke. Inače, stabla čvorova razlikuju se u postavkama čvorova *Distribute Matrices* i *Compose Matrix* koje je moguće vidjeti na slikama 6.11. - 6.13.



Slika 6.11. Prvo stablo čvorova za drugu iteraciju; stvaranje skupa objekata u smjeru z - osi



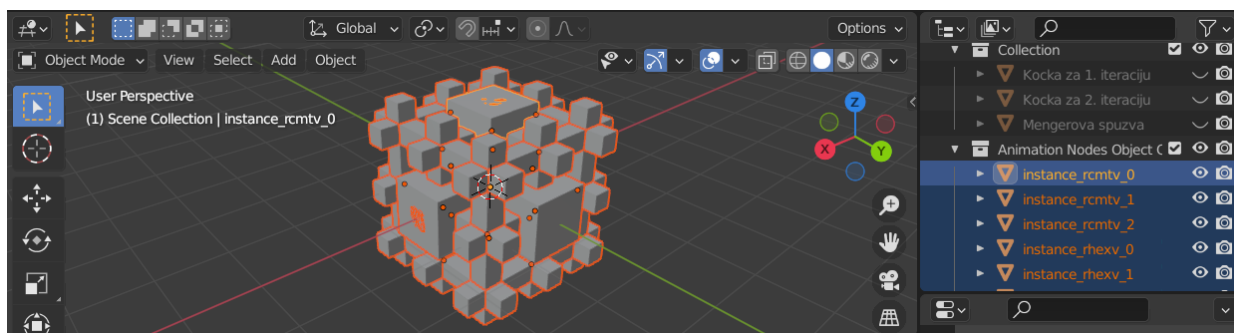
Slika 6.12. Drugo stablo čvorova za drugu iteraciju; stvaranje skupa objekata u smjeru y - osi



Slika 6.13. Posljednje stablo čvorova za drugu iteraciju; stvaranje skupa objekata u smjeru x - osi

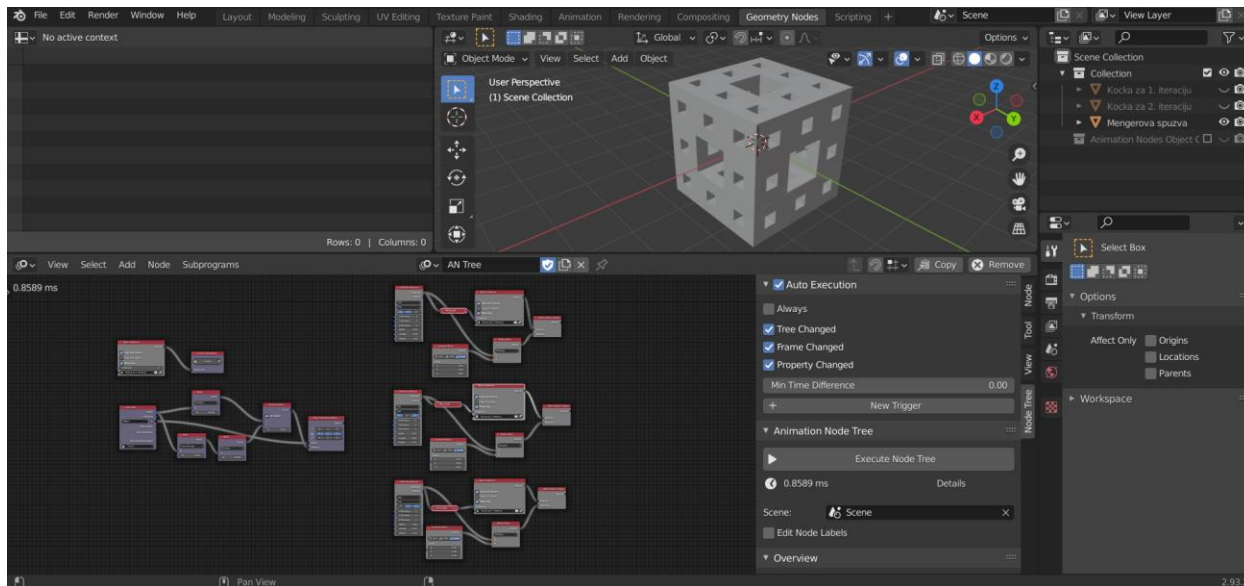
3. Rad sa stvorenim objektima i spremanje 3D projekata na računalo

Postavljanjem elementa „Kocka za 2. iteraciju” unutar sva tri čvora *Object Instancer*, dogodit će se promjena na sceni, odnosno unutar foldera *Animation Nodes Object Container* stvorit će se dodatni elementi. Primjenom alata *Join* na stvoreni skup objekata cilj je dobiti negativ druge iteracije Mengerove spužve. Na sljedećoj slici moguće je vidjeti stvoreni skup objekata.



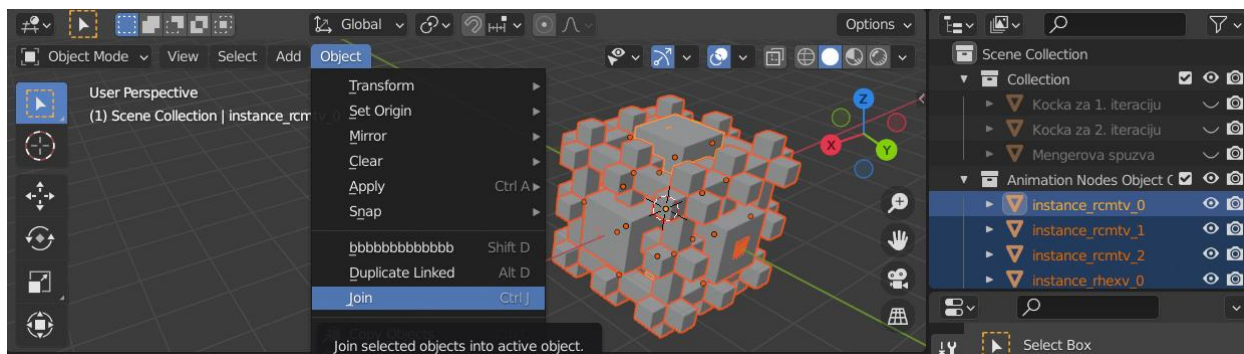
Slika 6.14. Skup objekata, stvorenih pomoću Animacijskih čvorova

Dodavanje novog modifikatora nije potrebno jer će prethodno dodan modifikator u prvoj iteraciji automatski napraviti promjene na sceni, u ovom slučaju na temelju stabala čvorova druge iteracije. Kako bi se mogao vidjeti novi izgled objekta „Mengerova spužva”, potrebno je unutar *Scene Collection* isključiti vidljivost svih ostalih objekata (slika 6.15.).

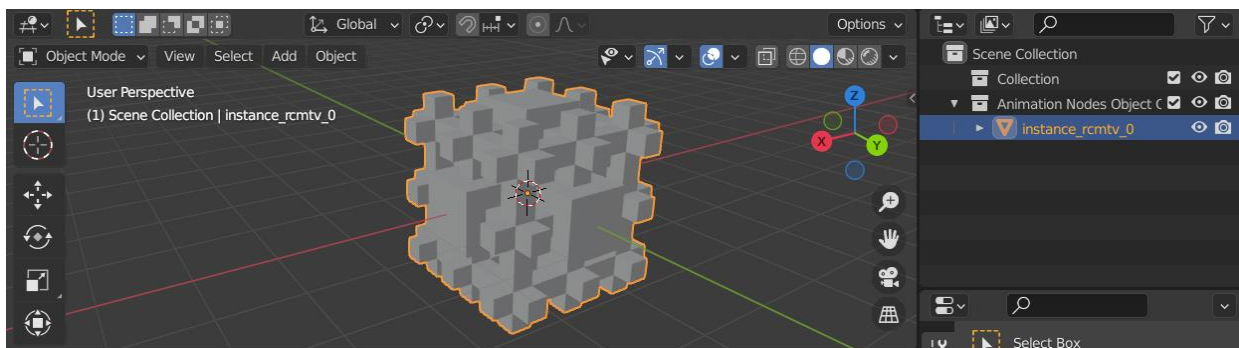


Slika 6.15. Izgled sučelja nakon izrade modela druge iteracije Mengerove spužve

Izrađen 3D projekt može se spremiti na računalo, npr. pod nazivom *Mengerova spužva i negativ 2*. Još preostaje pripremiti oba modela (objekt „Mengerova spužva” i njegov negativ) kako bi se svaki mogao zasebno spremiti. Postupak koji je potrebno provesti jednak je onom koji je opisan pri izradi prve iteracije, stoga se on neće ponovo opisivati. U nastavku se mogu vidjeti slike provedenog postupka.

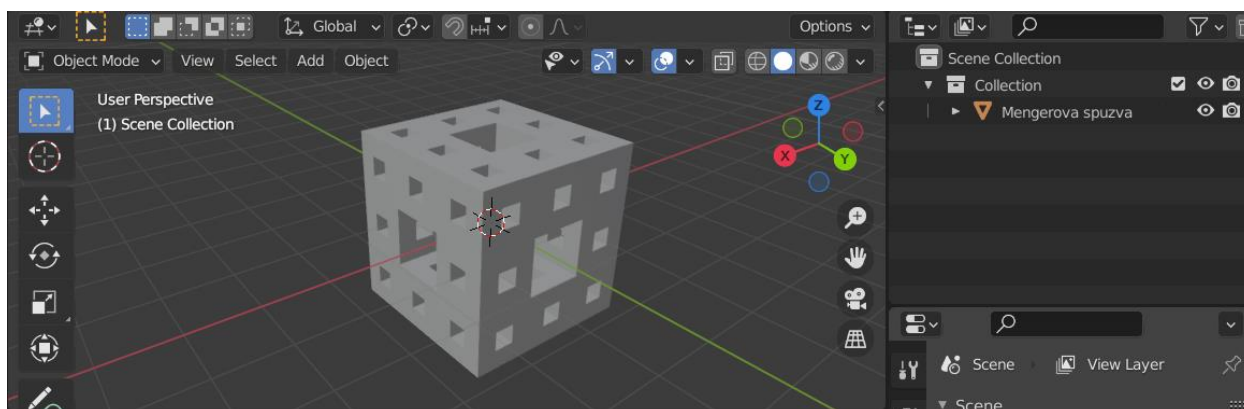


Slika 6.16. Postupak spajanja objekata dobivenih pomoću Animacijskih čvorova



Slika 6.17. Gotov model negativa druge iteracije Mengerove spužve

Na slici 6.17. nalazi se željeni objekt za 3D printanje. Stvoreni 3D projekt potrebno je spremiti na računalo, na primjer pod nazivom *Negativ Mengerove spužve 2*. Slijedi spremanje projekta s modelom Mengerove spužve. Nakon primjene modifikatora i brisanja suvišnih objekata unutar datoteke *Mengerova spužva i negativ 2.*, na sceni treba ostati jedino model druge iteracije Mengerove spužve (slika 6.18.). 3D projekt preostaje spremiti na računalo, na primjer pod nazivom *Mengerova spužva 2*.

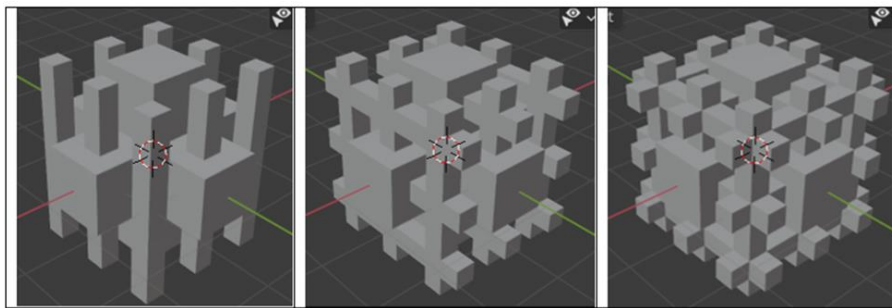


Slika 6.18. Gotov model druge iteracije Mengerove spužve

4. Objašnjenje načina na koji su izgrađena stabla Animacijskih čvorova

U nastavku će se objasniti jedino prvo stablo čvorova (slika 6.11.), budući da je postupak stvaranja objekata pomoću prvog stabla jednak onom koji se provodi pomoću ostala dva. Prvo stablo čvorova zaduženo je za stvaranje objekata u smjeru z - osi, dok su drugo i treće stablo zaduženi za stvaranje objekata u smjeru y i x - osi. Unutar prvog stabla moguće je primijetiti da su čvoru *Distribute Matrices* odabrane jednake postavke koje su odabrane prilikom izrade fraktala Tepiha Sierpińskog. Moguće je vidjeti da su izlazi čvorova *Distribute Matrices* i *Compose Matrix* spojeni na ulaze čvora *Matrix Math*. Čvor *Compose Matrix* formira matricu transformacije, pri čemu je odabrano skaliranje konstantom 9 po z - osi. Skaliranje je definirano na ovaj način jer je plan stvoriti objekte kojima je duljina visine devet puta veća od duljine brida objekta „Kocka za 2. iteraciju”. Dimenzije stvorenih objekata biti će $\frac{1}{3} \times \frac{1}{3} \times 3$ m. Kao podsjetnik, objekt „Kocka za 2. iteraciju” ima dimenzije $\frac{1}{3} \times \frac{1}{3} \times \frac{1}{3}$ m.

Unutar *Matrix Math* čvora provodi se množenje liste matrica čvora *Distribute Matrices* s matricom čvora *Compose Matrix*. Kao umnožak dobiva se nova lista matrica, a nalazi se unutar izlaza *Results* čvora *Matrix Math*. Provođenjem opisane operacije množenja povećat će se prostor unutar kojeg će se nalaziti lokacije buduće stvorenih objekata. Čvor *Object Instancer* stvara kopije objekta „Kocka za 2. Iteraciju”, dok čvor *Get Length* računa koliki broj kopija je potrebno stvoriti. Čvor *Object Matrix Output* transformira ulazne objekte (kopije objekata koje je stvorio čvor *Object Instancer*) po matrici ulaznih transformacija (liste matrica, odnosno lokacija). Jednostavnije rečeno, taj čvor na temelju prethodno stvorenih lokacija provodi transformaciju odgovarajućih objekata te ih zatim pridružuje pripadajućim lokacijama. Time će se stvoriti željeni objekti, a prikazani su na slici 6.19. (prvi skup objekata slijeva).



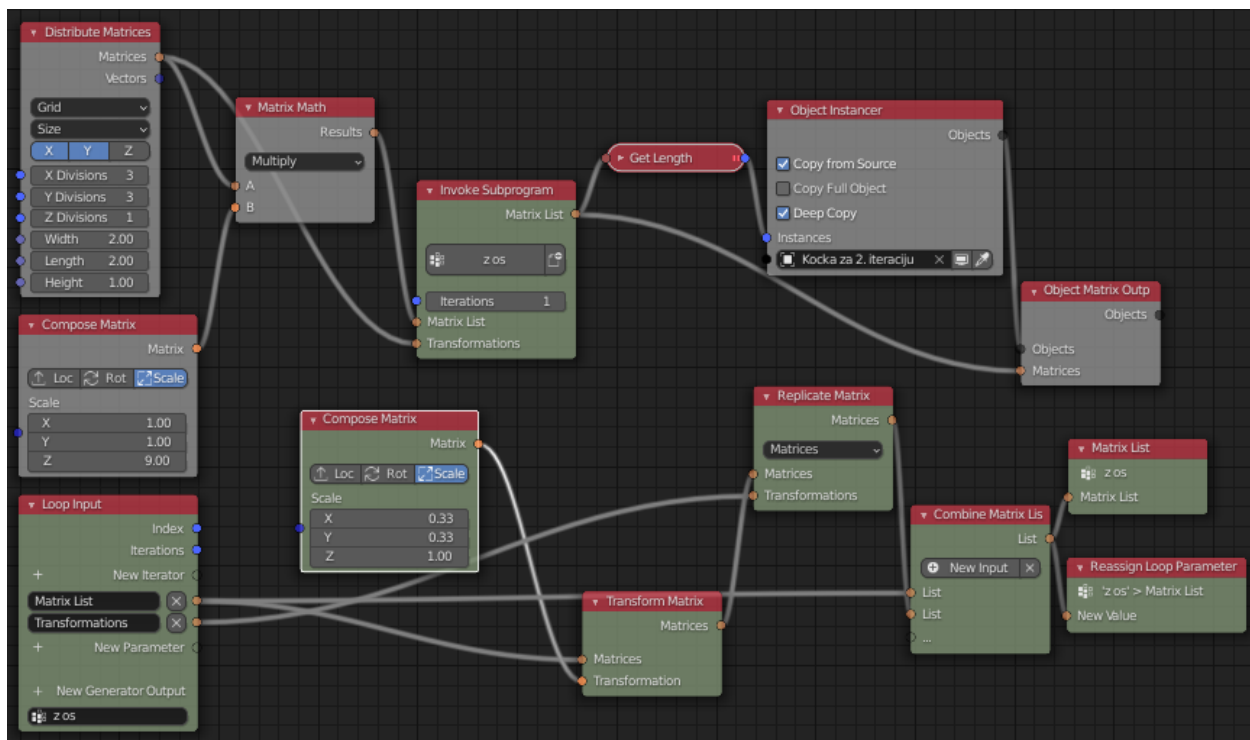
Slika 6.19. Stvoreni objekti pomoću Animacijskih čvorova slijeva nadesno; objekti prvog, drugog i trećeg stabla

6.1.4. Treća iteracija

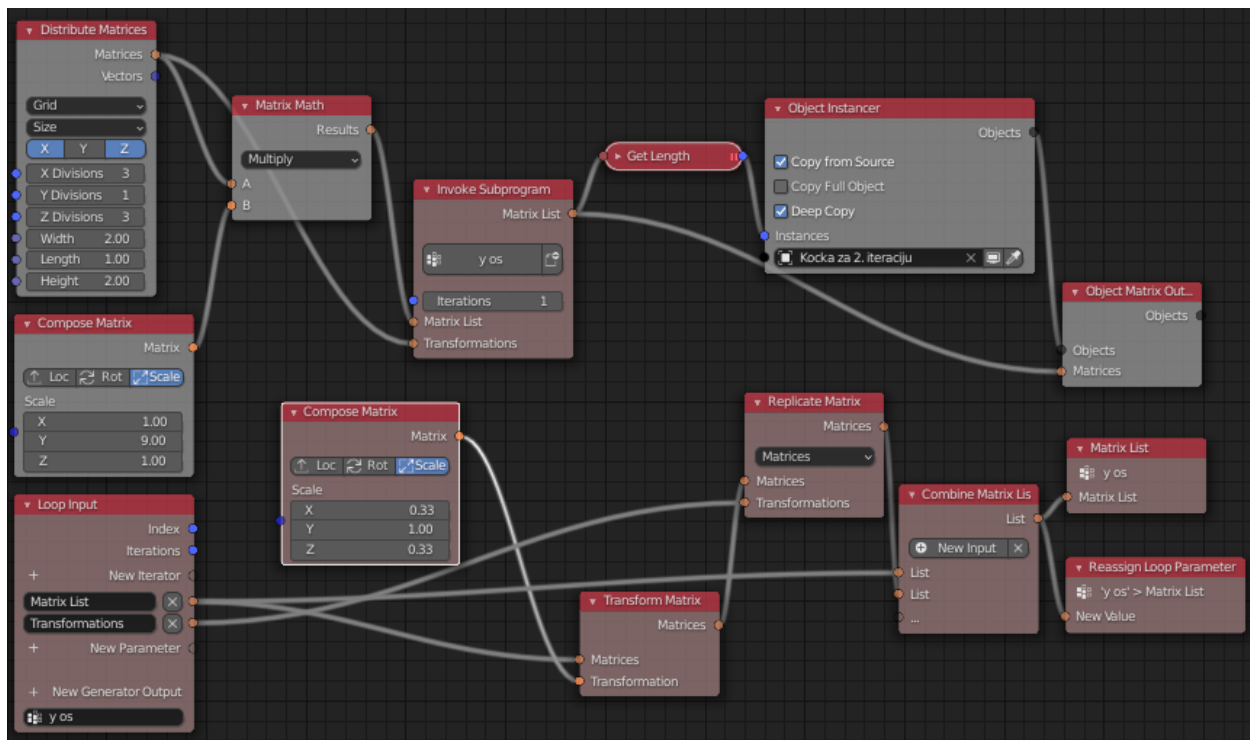
Model treće iteracije Mengerove spužve i njezinog negativa izradit će se vršenjem odgovarajućih transformacija unutar prethodno spremljenog 3D projekta *Mengerova spužva i negativ 2*. Stoga je potrebno otvoriti spomenutu 3D datoteku. Ovaj put nije potrebno dodavanje nove kocke na scenu jer će se za stvaranje rupa treće iteracije koristiti objekt „Kocka za 2. iteraciju“. Treća iteracija Mengerove spužve i njezinog negativa također će se izraditi pomoću Animacijskih čvorova. Stabla treće iteracije gradit će se desno od stabala druge iteracije.

1. Gradnja stabala Animacijskih čvorova

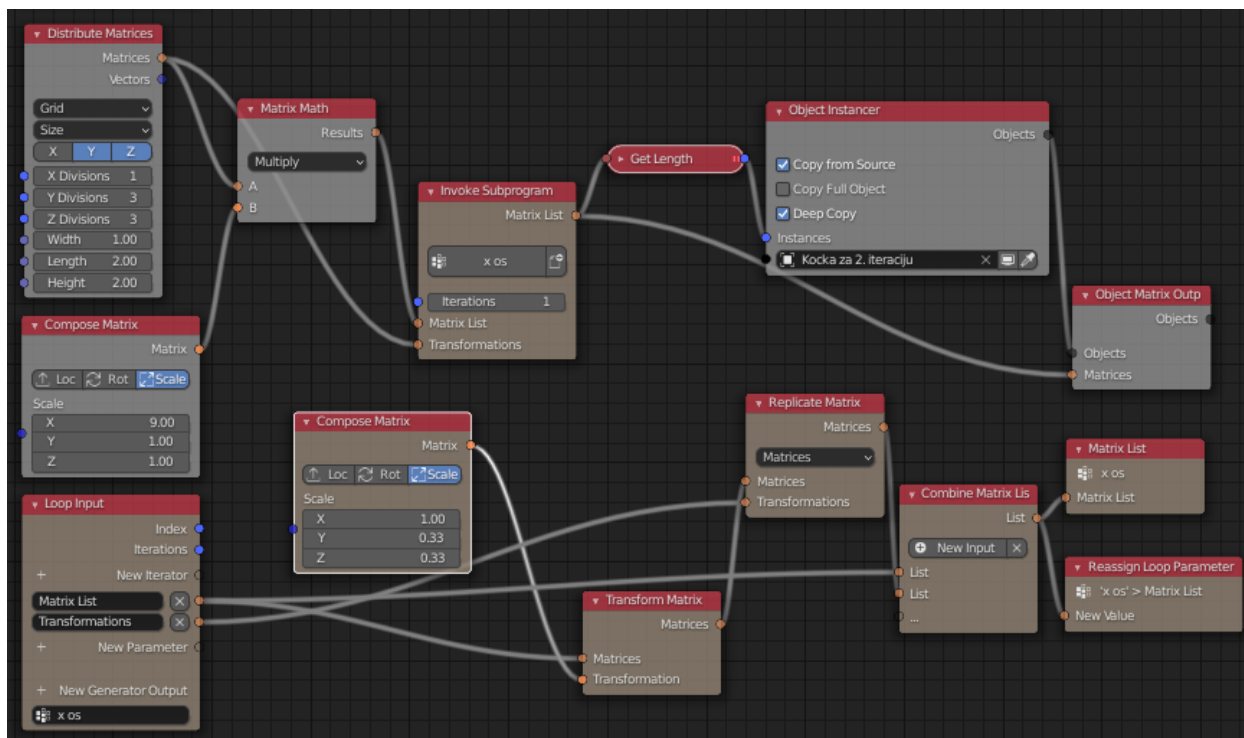
Kreće se s gradnjom prvog para stabala, a nakon što se oni izgrade, na isti način je potrebno izgraditi preostala dva para. Prvo stablo sadrži čvorove *Distribute Matrices*, *Compose Matrix*, *Matrix Math*, *Invoke Subprogram*, *Get Length*, *Object Instancer* te *Object Matrix Output*. Drugo stablo sadrži čvorove *Loop Input*, *Compose Matrix*, *Transform Matrix*, *Replicate Matrix*, *Combine Matrix List*, *Matrix List* te *Reassign Loop Parametar*. Nakon što su dodani unutar radnog prostora, čvorove je potrebno rasporediti i spojiti prema slici 6.20. te unutar njih odabrati prikazane postavke. Parovi se razlikuju u postavkama čvorova *Distribute Matrices* i *Compose Matrix*. Također, svaki par ima jedinstven potprogram *Invoke Subprogram*. Sva tri para stabala moguće je vidjeti na idućim slikama.



Slika 6.20. Prvi par stabala čvorova za treću iteraciju; stvaranje skupa objekata u smjeru z - osi



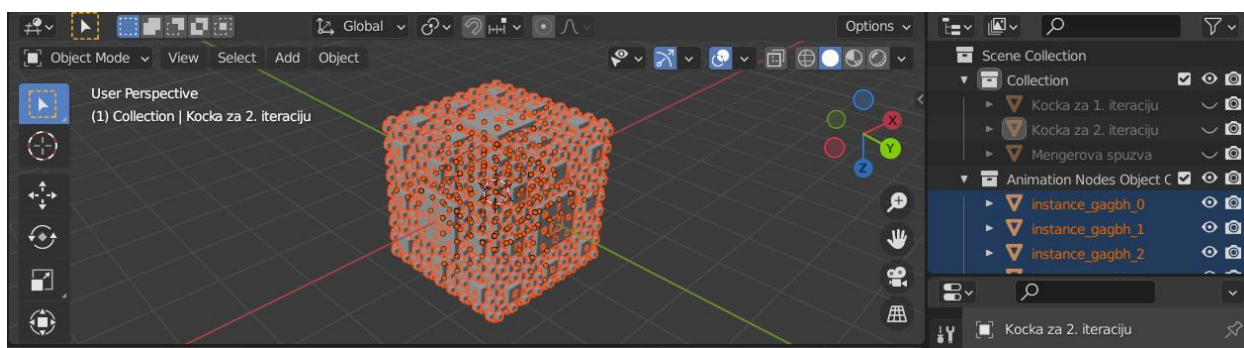
Slika 6.21. Drugi par stabala čvorova za treću iteraciju; stvaranje skupa objekata u smjeru y - osi



Slika 6.22. Posljednji par stabala čvorova za treću iteraciju; stvaranje skupa objekata u smjeru x - osi

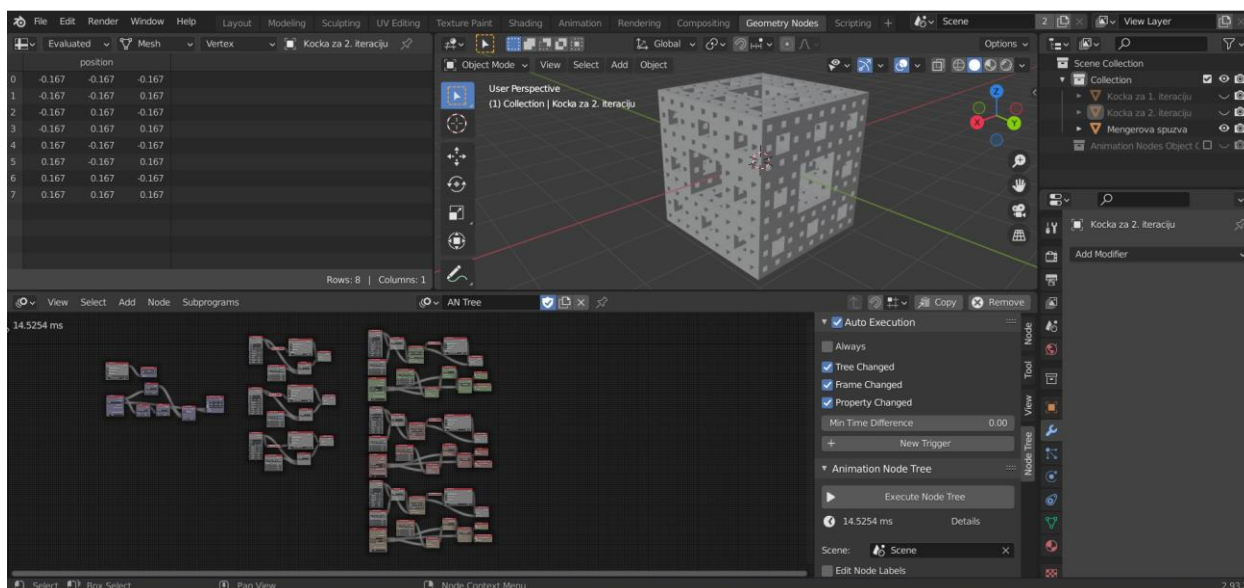
2. Rad sa stvorenim objektima i spremanje 3D projekata na računalo

Postavljanjem elementa „Kocka za 2. iteraciju” unutar sva tri čvora *Object Instancer* te upisivanjem vrijednosti 1 unutar ulaza *Iterations* čvora *Invoke Subprogram*, dogodit će se promjena na sceni. Istodobno će se unutar foldera *Animation Nodes Object Container* stvoriti dodatni elementi. Primjenom alata *Join* na stvoreni skup objekata cilj je dobiti negativ treće iteracije Mengerove spužve. Na sljedećoj slici moguće je vidjeti spomenuti skup objekata.



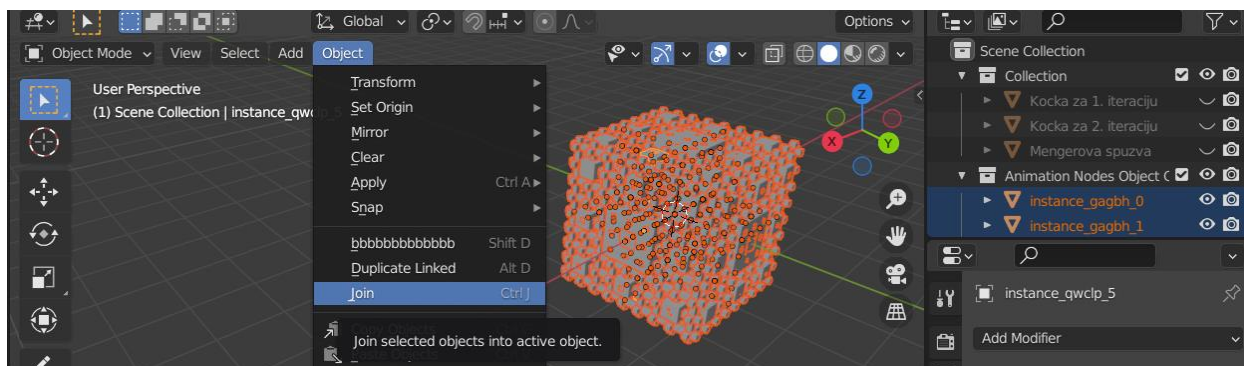
Slika 6.23. Skup objekata, stvorenih pomoću Animacijskih čvorova

Modifikator *Boolean* koji je dodan pri izradi prve iteracije automatski će izvršiti promjene na sceni, u ovom slučaju na temelju stabala čvorova treće iteracije. Novi izgled objekta „Mengerova spužva” prikazan je na slici 6.24., kojeg je moguće vidjeti nakon što je isključena vidljivost svih ostalih objekata unutar *Scene Collection*.

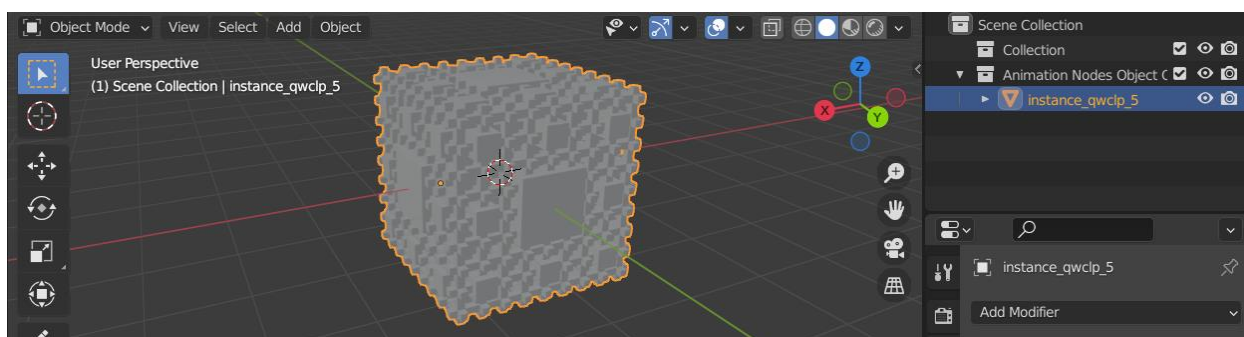


Slika 6.24. Izgled sučelja nakon izrade modela treće iteracije Mengerove spužve

Kreiran 3D projekt može se spremiti na računalo, npr. pod nazivom *Mengerova spužva i negativ 3*. Još preostaje pripremiti oba modela (objekt „Mengerova spužva” i njegov negativ) kako bi se svaki projekt mogao zasebno spremiti. Postupak koji je potrebno provesti jednak je onom koji je opisan pri izradi druge iteracije, stoga se on neće ponovo opisivati. U nastavku se mogu vidjeti slike provedenog postupka.

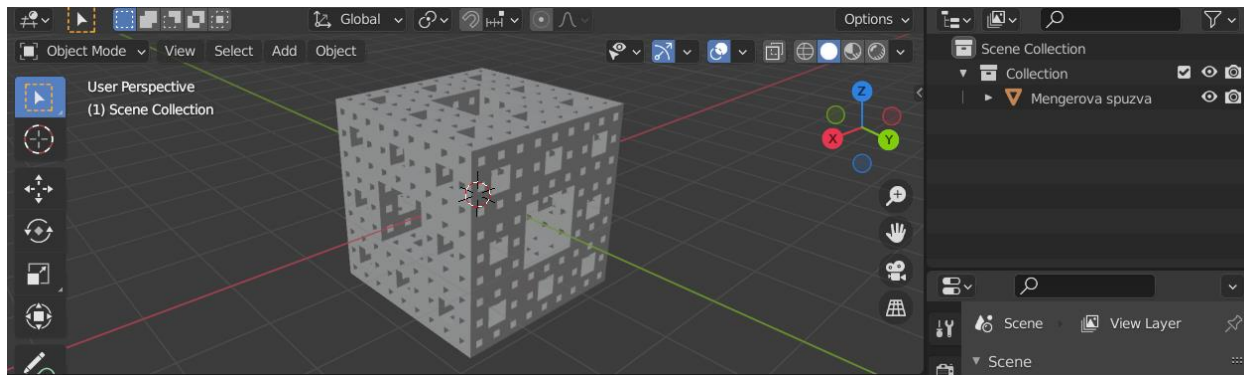


Slika 6.25. Postupak spajanja objekata dobivenih pomoću Animacijskih čvorova



Slika 6.26. Gotov model negativa treće iteracije Mengerove spužve

Stvoreni 3D projekt na slici 6.26. potrebno je spremi na računalo, na primjer pod nazivom *Negativ Mengerove spužve 3*. Za pripremu modela Mengerove spužve otvara se spremljen dokument *Mengerova spužva i negativ 3*. Nakon provedenog postupka brisanja suvišnih objekata sa scene, dobiva se gotov model treće iteracije Mengerove spužve (slika 6.27.).



Slika 6.27. Gotov model treće iteracije Mengerove spužve

Gotov 3D projekt može se spremi na računalo, na primjer pod nazivom *Mengerova spužva 3*.

3. Objašnjenje načina na koji su izgrađena stabla Animacijskih čvorova

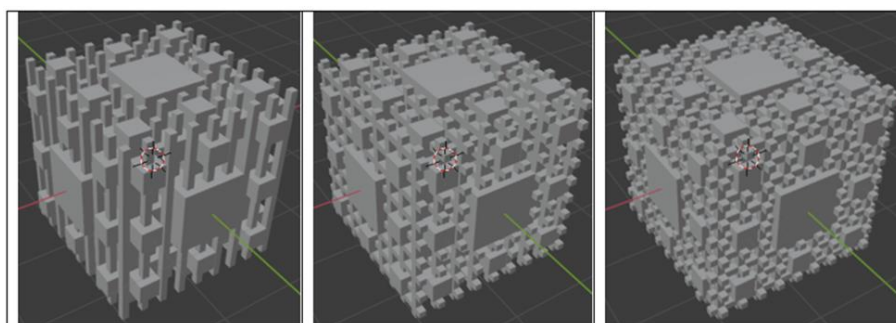
Objasniti će se jedino prvi par stabala čvorova (slika 6.20.). Prvi par stabala čvorova zadužen je za stvaranje objekata u smjeru z - osi. Općenito, ideja stvaranja modela treće iteracije Mengerove spužve slična je onoj koja se koristila pri izradi modela treće iteracije Tepiha Sierpińskog. Ukoliko se usporede stabla čvorova za izradu treće iteracije Tepiha Sierpińskog i Mengerove spužve, moguće je primijetiti da su ona vrlo slična. Kao što je opisano i pri izradi Tepiha Sierpińskog, pomoću čvora *Distribute Matrices* stvoreno je devet lokacija. Riječ je o lokacijama unutar kojih se nalaze objekti druge iteracije, pri čemu su dimenzije svakog objekta jednake $1/3 \times 1/3 \times 3$ m (Slika 6.19., prvi skup objekata slijeva). Sada je cilj izraditi manje objekte, tj. oko svakog od tih devet objekata potrebno je stvoriti osam manjih. Dimenzije manjih objekata biti će jednake $1/9 \times 1/9 \times 3$ m.

Ponovo je potrebno promotriti sliku 6.20. Čvor *Compose Matrix* formira matricu transformacije, pri čemu je odabrano skaliranje konstantom 9 po z - osi. Unutar *Matrix Math* čvora provodi se množenje liste matrica čvora *Distribute Matrices* s matricom čvora *Compose Matrix*. Kao umnožak dobiva se nova lista matrica. Skaliranje je definirano na ovaj način jer je plan stvoriti objekte kojima je duljina visine devet puta veća od duljine brida objekta „Kocka za 2. iteraciju”. Čvor *Compose Matrix* formira matricu transformacije, pri čemu je odabrano skaliranje konstantom $1/3$ po x i y - osi. Čvor *Transform Matrix* provest će skaliranje konstantom $1/3$ po x i y - osi nad listom matrica, formiranih pomoću *Matrix Math* čvora. Time će se automatski smanjiti udaljenost između devet početnih lokacija.

Čvor *Replicate Matrix* napraviti će devet trostruko manjih kopija lokacija te će dobivene kopije rasporediti na devet početnih lokacija. Čvor *Combine Matrix List* stvara uniju liste koja sadrži devet početnih lokacija te liste koja sadrži one dobivene pomoću čvora *Replicate Matrix*.

Čvor *Matrix list* je nova lista lokacija, kreirana nakon djelovanja *Combine Matrix List* čvora. Provođenjem daljnjih iteracija, ulaz *Matrix List* čvora sadržavat će sve veći broj lokacija. Čvor *Reassign Loop Parametar* će nakon svake uvećane vrijednosti iteracije čvoru *Combine Matrix List* pridodati listu prethodno kreiranih lokacija.

Čvor *Object Instancer* stvara kopije objekata. Unutar čvora *Object Instancer* unosi se element „Kocka za 2. iteraciju“ jer je na temelju unaprijed definiranih transformacija cilj dobiti objekte čije su dimenzije baza trostruko manje od dimenzija baze objekta „Kocka za 2. iteraciju“ te čija je visina devet puta veća od od duljine brida objekta „Kocka za 2. iteraciju“. Čvor *Object Matrix Output* na temelju prethodno stvorenih lokacija provodi transformaciju odgovarajućih objekata te ih zatim pridružuje pripadajućim lokacijama. Konačno, na slici 6.28. (prvi skup objekata slijeva) mogu se vidjeti stvoreni objekti.



Slika 6.28. Stvoreni objekti pomoću Animacijskih čvorova slijeva nadesno; objekti prvog, drugog i trećeg para stabala

6.1.5. Daljnje iteracije

Modeli daljnjih iteracija mogu se izraditi pomoću stabala čvorova treće iteracije. Na primjer, ukoliko bi se htio izraditi model četvrte iteracije, unutar ulaza *Iterations* čvora *Invoke Subprogram* potrebno je upisati vrijednost 2, i to unutar sva tri para stabala koja ga sadrže. Budući da će time Blender morati izraditi jako velik broj objekata, proces njihova stvaranja biti će vrlo spor.

6.2. Rezanje u PrusaSlicer-u

Kako bi se mogli izrezati gotovi 3D modeli, potrebno je otvoriti svaku od spremljenih .blend datoteka te izvesti STL datoteku odabirući *File → Export → Stl (.stl)* u gornjem lijevom kutu programa. Unutar PrusaSlicer programa treba uvesti .stl format datoteke. Zatim se biraju odgovarajuće postavke printanja. Rezanje modela provodi pritiskom na opciju *Slice now* unutar softvera. Preostaje izvesti datoteku G - koda, pritiskom na opciju *Export G-code*.

6.3. 3D printanje pomoću Prusa i3 MK3S 3D printera i naknadna obrada

Slijedi 3D printanje modela. Nakon što su isprintani 3D objekti, potrebno je ukloniti potporni materijal sa svakog objekta.

7. ZAKLJUČAK

U ovom radu upoznali smo se s dvije vrlo zanimljive teme, fraktalima i 3D printanjem. Osim samog upoznavanja s ovim dvjema temama, cilj rada bio je prikazati način izrade 3D fraktala Tepiha Sierpinskog te Mengerove spužve i njezinog negativa 3D printerom Prusa MK3S. 3D modele izradila sam pomoću besplatnog softvera za 3D modeliranje Blender. Blender se prilikom rada pokazao vrlo efikasnim i korisnim. Također, zahtjevan je za upotrebu prije nego što se ne nauči njime služiti. Budući da otprije nisam imala iskustva s 3D modeliranjem niti općenito procesom 3D printanja, mogu reći da je praktični dio rada za mene bio istovremeno vrlo izazovan i zanimljiv.

8. LITERATURA

- [1] AutoDesSys, *Subdivision Surfaces*, http://www.formz.com/manuals/formz9/Layout/WebHelp/10850_Subdivision_surfaces.html, (8.11.2021.)
- [2] Bodiš, I., *Fraktali*, Odjel za matematiku, Sveučilište J.J. Strossmayera u Osijeku, Diplomski rad, Osijek, 2012.
- [3] Cooper, F., *Introduction to Fractals and Julia Sets*, University of Warwick, Warwick, 2011.
- [4] Eklić, H., *Eksperimentalno određivanje fraktalne dimenzije*, <http://e.math.hr/old/frakdim/index.html>, (3.4.2021.)
- [5] Fractal Foundation, *The Mandelbrot Set, Exploring Infinity*, <http://fractalfoundation.org/OFC/OFC-index.htm>, (5.4.2021.)
- [6] Horvat, M., *Vizualizacija funkcija i ploha pomoću 3D printera*, Odjel za matematiku, Sveučilište u Rijeci, Diplomski rad, Rijeka, 2021.
- [7] Lakičević, L., *Osnove 3D modeliranja u Blenderu*, Odjel za informatiku, Sveučilište u Rijeci, Završni rad, Rijeka, 2020.
- [8] Lucke, J., *Animation Nodes*, <https://docs.animation-nodes.com/>, (10.7.2021.)
- [9] Ljubojević, D., *Modeliranje u programskom alatu Blender*, Fakultet organizacije i informatike, Sveučilište u Zagrebu, Varaždin, 2020.
- [10] Mikuš, A., *Trodimenzionalna fraktalna analiza pojava u kršu*, Prirodoslovno-matematički fakultet, Sveučilište u Zagrebu, Zagreb, 2016.
- [11] Novak, V., Trstenjak Petran, M., *Fraktali*, [8_Trstenjak_Petran_Novakm.pdf](#), (20.12.2021.)
- [12] Sabljak, A., *Klasični fraktali*, Odjel za matematiku, Sveučilište J.J. Strossmayera u Osijeku, Diplomski rad, Osijek, 2018.
- [13] STEMpedia, *Getting started with 3D printing*, <https://thestempedia.com/tutorials/getting-started-with-3d-printing/>, (20.4.2021.)
- [14] The Conversation, *Explainer: what are fractals?*, <https://theconversation.com/explainer-what-are-fractals-10865>, (8.4.2021.)
- [15] think3D, *Beginner's Guide to 3D Printing*, [beginners-guide-to-3d-printing.pdf](#), (19.4.2021.)
- [16] Tiss, R., *Creating Motion Graphics in Blender with Animation Nodes*, Tampere University of Applied Sciences, Tampere, 2017.
- [17] Varšava, M., *Animacijski čvorovi u Blenderu*, Fakultet elektrotehnike, računarstva i informacijskih tehnologija, Sveučilište J.J. Strossmayera u Osijeku, Završni rad, Osijek, 2019.

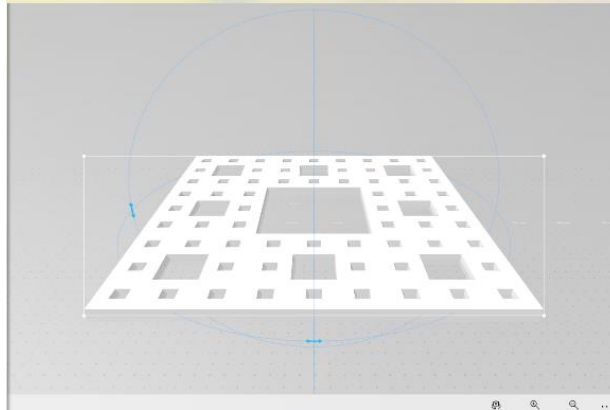
[18] 5agado, *Generating fractals with Blender and Animation nodes*,
<https://towardsdatascience.com/generating-fractals-with-blender-and-animation-nodes-3ae79d02bf40>, (7.7.2021.)

9. IZVORI SLIKA

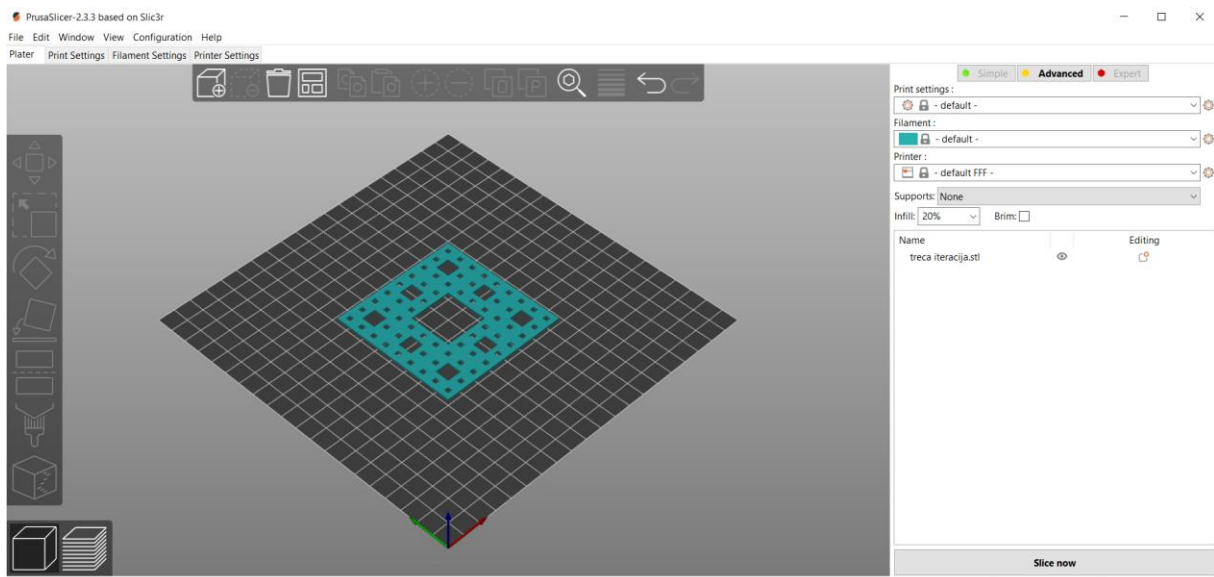
- [19] <https://rapidfab.ricoh-europe.com/technologies/fused-deposition-modelling/>
- [20] https://commons.wikimedia.org/wiki/File:Logo_Blender.svg
- [21] <https://kieranpbrown.wordpress.com/what-are-polygons-in-3d-modelling/>
- [22] https://www.researchgate.net/figure/Physical-mesh-and-control-mesh-of-a-quadratic-NURBS-curve-left-and-a-quadratic-NURBS_fig2_236627685
- [23] <http://www.3drender.com/nurbana/>
- [24] https://www.researchgate.net/figure/Figure12-Construction-process-of-a-subdivision-surface-a-Initial-mesh-b-Subdivide_fig1_320327453
- [25] <https://www.youtube.com/watch?v=cyt0O7saU4Q&t=395s>
- [26] https://en.wikipedia.org/wiki/Julia_set
- [27] <https://blogs.scientificamerican.com/roots-of-unity/a-few-of-my-favorite-spaces-the-menger-sponge/>
- [28] <https://www.chegg.com/homework-help/questions-and-answers/1-sierpinski-carpet-fractal-image-formed-follows-start-square-side-length-1-remove-center--q59564143>
- [29] https://sh.m.wikipedia.org/wiki/Datoteka:Koch_curve.svg
- [30] https://en.wikipedia.org/wiki/Weierstrass_function
- [31] https://www.wahl.org/fe/HTML_version/link/FE3W/c3.htm
- [32] [https://www.semanticscholar.org/paper/The-exact-\(up-to-infinitesimals\)-infinite-perimeter-Sergeyev/64430e35039f17ef3b516899435d34d55c250798/figure/0](https://www.semanticscholar.org/paper/The-exact-(up-to-infinitesimals)-infinite-perimeter-Sergeyev/64430e35039f17ef3b516899435d34d55c250798/figure/0)
- [33] https://sh.wikipedia.org/wiki/Beno%C3%AEt_Mandelbrot

10. PRILOZI

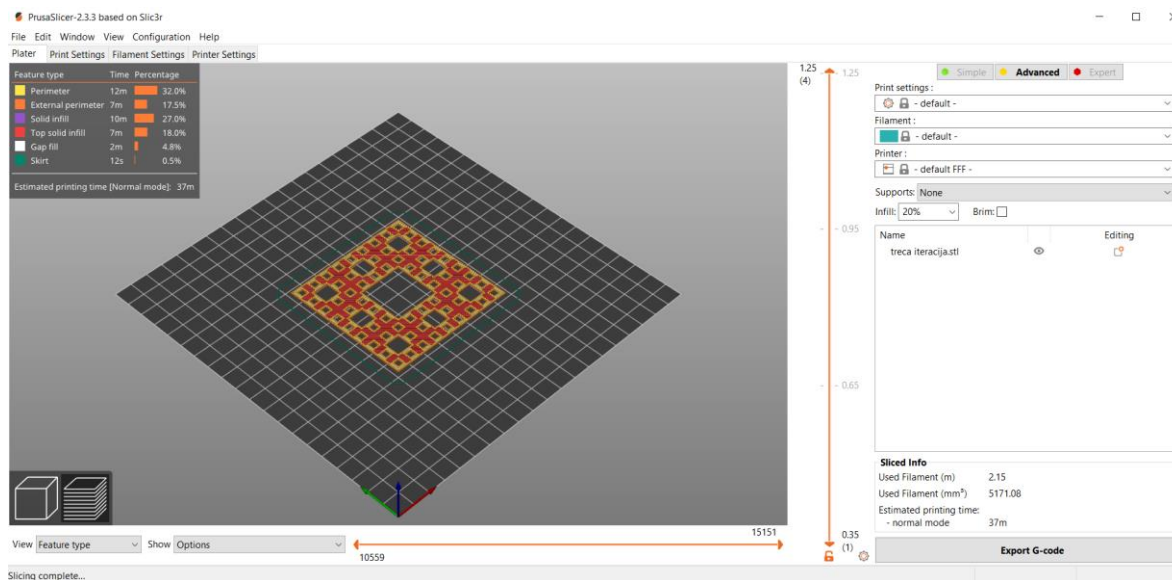
Prilog 1 – Prikaz STL datoteke: model treće iteracije Tepiha Sierpińskog



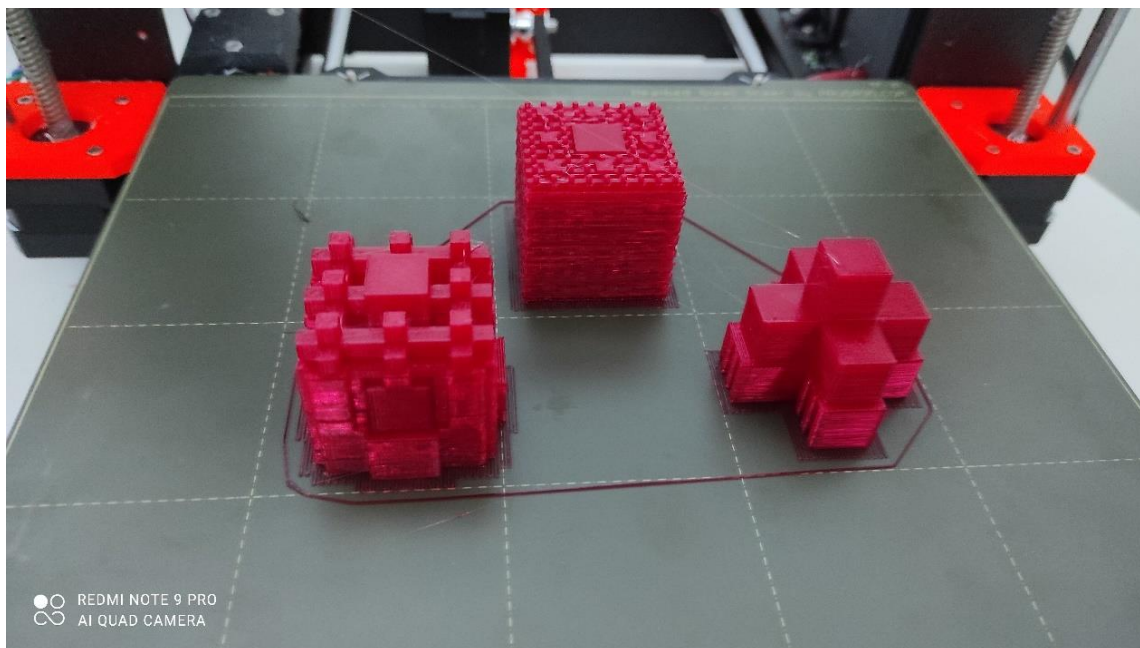
Prilog 2 – Prikaz rada u PrusaSlicer-u: model treće iteracije Tepiha Sierpińskog



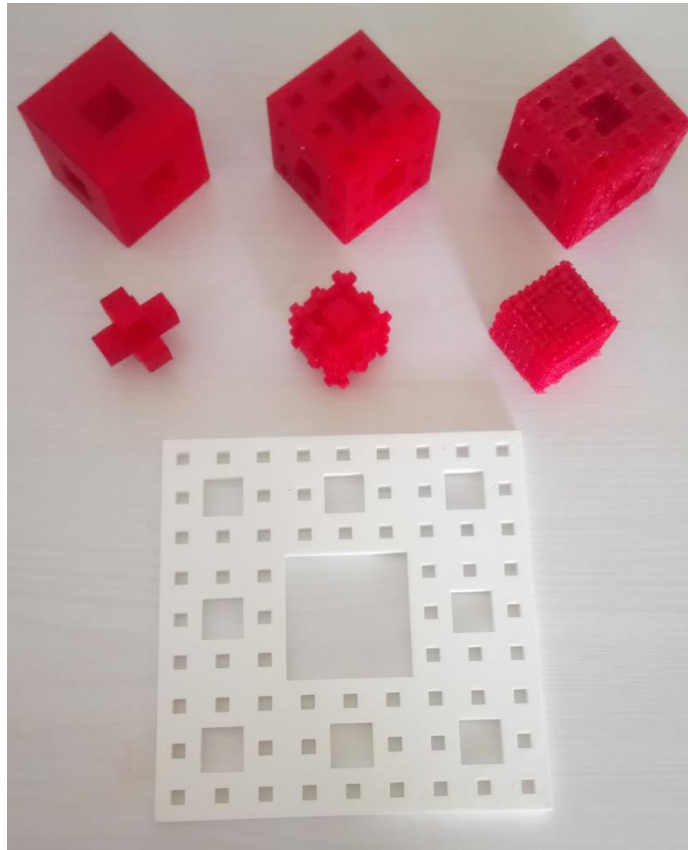
Prilog 3 – Prikaz izrezanog modela: model treće iteracije Tepiha Sierpińskog



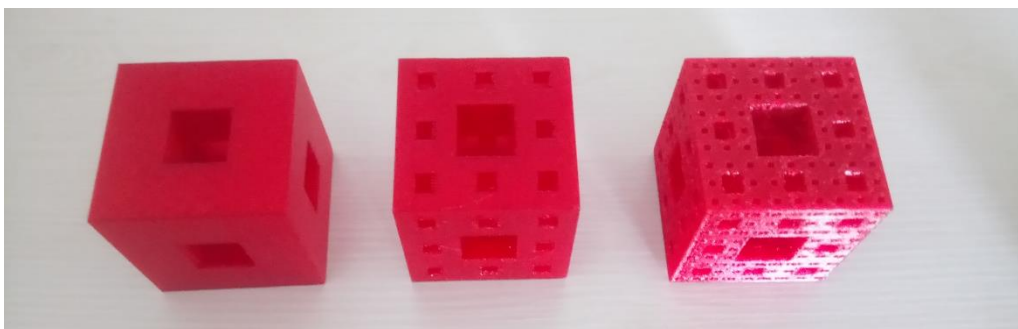
Prilog 4 – Modeli s potpornim materijalom: prve tri iteracije negativa Mengerove spužve



Prilog 5 – Isprintani modeli



Prilog 6 – Detalji prvih triju iteracija Mengerove spužve



Prilog 7 – Detalji prvih triju iteracija negativa Mengerove spužve

