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

Sveučilišni preddiplomski studij

Proceduralno generiranje građevina u *Blenderu* primjenom geometrijskih čvorova

Završni rad

Filip Jakirac

Osijek, 2023.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 08.09.2023.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Filip Jakirac
Studij, smjer:	Računalno inženjerstvo
Mat. br. Pristupnika, godina upisa:	R 4355, 11.10.2021.
OIB Pristupnika:	86960825699
Mentor:	izv. prof. dr. sc. Časlav Livada
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Proceduralno generiranje građevina u Blenderu primjenom geometrijskih čvorova
Znanstvena grana rada:	**Obradba informacija (zn. polje računarstvo)**
Zadatak završnog rad:	U radu je potrebno napraviti aplikaciju koristeći Blender za proceduralno generiranje građevina (zgrade, mostovi, ograde...) primjenom geometrijskih čvorova (engl. geometry nodes) Primjeri: https://twitter.com/Pavel_Oliva/status/1591839944980529156 https://paveloliva.gumroad.com/l/buildify
Prijedlog ocjene završnog rada:	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene od strane mentora:	08.09.2023.
Datum potvrde ocjene od strane Odbora:	24.09.2023.

Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 25.09.2023.

Ime i prezime studenta:	Filip Jakirac
Studij:	Računalno inženjerstvo
Mat. br. studenta, godina upisa:	R 4355, 11.10.2021.
Turnitin podudaranje [%]:	3

Ovom izjavom izjavljujem da je rad pod nazivom**: Proceduralno generiranje građevina u Blenderu primjenom geometrijskih čvorova**

izrađen pod vodstvom mentora izv. prof. dr. sc. Časlav Livada

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Sadržaj

1. UVOD

3D modeliranje jedna je od disciplina koja je nastala te je postupno poboljšavana razvojem računalnih sustava. Ona obuhvaća korištenje računala i računalnog softvera za kreiranje i realističan prikaz raznorodnih 3D objekata na zaslonu računala pomoću širokog raspona različitih programskih rješenja. Počeci računalnog 3D modeliranja javljaju se 1970-ih godina korištenjem modeliranja žičanog okvira [1] (engl. *wireframe modelling*) na način formiranja međusobno povezanih linija u vrhove i rubove u trodimenzionalnom koordinatnom sustavu. Nastankom i razvojem složenijih i sposobnijih računalnih sustava, omogućeno je dizajniranje kompleksnih objekata s površinom i detaljima visoke rezolucije. Neki od najpoznatijih računalnih alata današnjice korišteni u području 3D modeliranja su *Autodesk Fusion 360, 3ds Max, SolidWorks* i *Blender*.

U ovom radu razmotrit će se mogućnosti proceduralne izgradnje raznih građevina u programu *Blender*. U prvom dijelu rada, opisat će se proces modeliranja složenih 3D objekata te prikazati potencijalna ograničenja i probleme vezane uz njega.

U drugom dijelu rada, bit će prikazano područje rada te alati koji nude efikasniji način za stvaranje takvih objekata u računalu.

Treći dio rada prikazat će programsko rješenje zadatka te detaljno opisati postupak proceduralnog stvaranja različitih elemenata pomoću ugrađenih funkcionalnosti programa *Blender*. Opisat će se pojmovi koje je važno razumjeti kako bi se takav pristup problemu ispravno shvatio i koristio.

U zaključku rada prikazat će se sažet opis problema i korištenog programskog rješenja kao načina da se smanje ili u potpunosti zaobiđu problemi koji nastaju tradicionalnom tehnikom modeliranja složenih 3D objekata u za to namijenjenim računalnim programima.

1.1. Zadatak rada

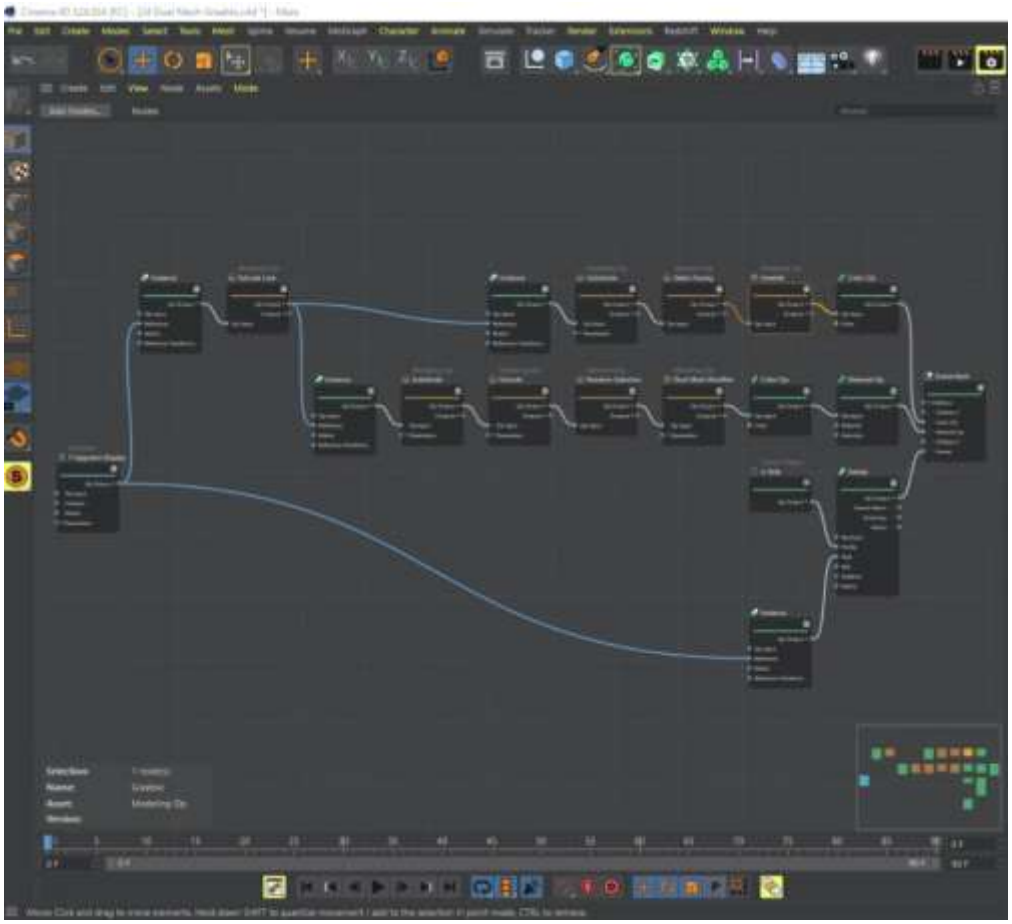
Zadatak ovog rada upotreba je geometrijskih čvorova (engl. *geometry nodes*) u svrhu proceduralne izgradnje 3D modela različitih građevina. Dodatno, navesti neke slične programe koji omogućuju takvu funkcionalnost kao i detaljno opisati proces stvaranja takvog programskog rješenja.

2. PREGLED PODRUČJA RADA

U ovom dijelu rada prikazat će se nekoliko primjera programa koji se mogu koristiti za usporedbu s rješenjem prikazanim u ovom radu. Na kraju poglavlja opisat će se korišteni program i obrazložiti zašto je odabran za korištenje pri izradi ovog projekta.

2.1. *Cinema* 4D

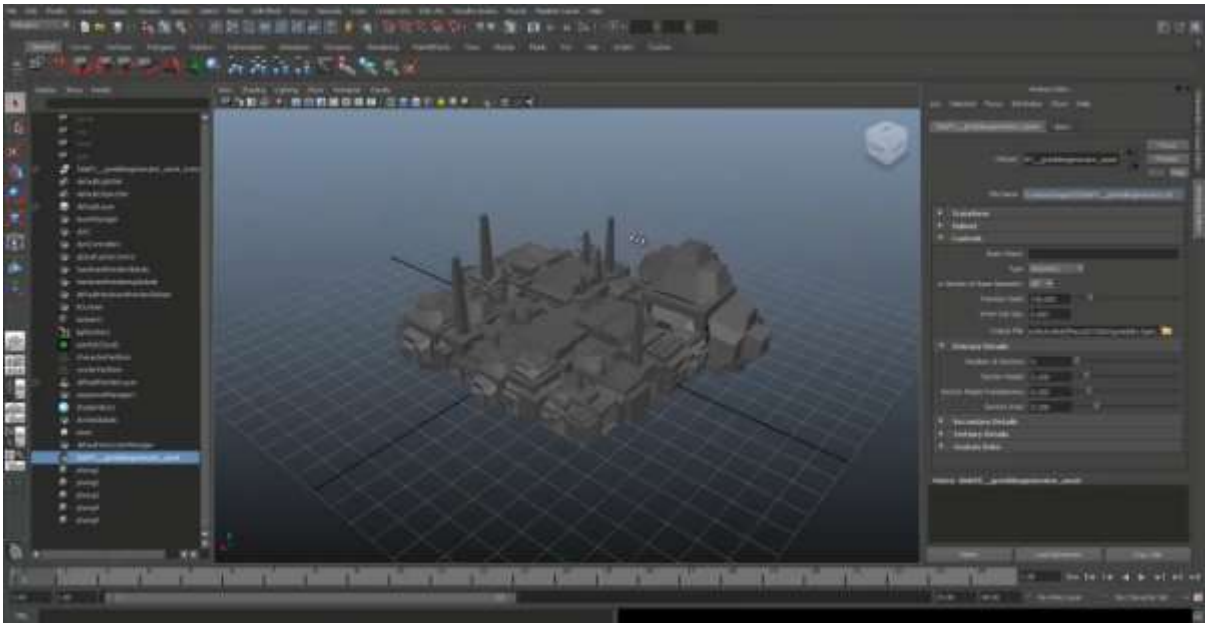
Cinema 4D popularan je program namijenjen 3D modeliranju koji nudi mogućnost proceduralnog generiranja objekata. Jedna od prednosti ovog programa je što nudi jednostavne ugrađene funkcije za proceduralno generiranje objekata koji se mogu proizvoljno mijenjati te koristiti u raznim projektima. Koristi sustav *Xpresso* [2] kako bi korisnici imali vizualni prikaz odnosa povezanih čvorova korištenih za generiranje objekata. Rad sa čvorovima u programu prikazan je slikom 2.1. Koristi se i programskim jezikom *Python* i omogućuje izmjenu postojećih kao i dodavanje novih funkcionalnosti. Prema [3], *Python* je pogodan za korištenje za razne potrebe, uključujući razvoj video igara, strojno učenje ili razvoj grafičkih sučelja.



Slika 2.1. *Rad s čvorovima u Xpresso sustavu programa Cinema 4D* [4].

2.2. *Houdini*

Houdini je program tvrtke *SideFX* namijenjen 3D modeliranju stvoren na principu proceduralnog načina rada [5] (engl. *procedural node based workflow*). Korisnicima nudi rukovanje čvorovima pri modeliranju objekata i često se koristi u filmskoj industriji i industriji video igara. U usporedbi s ostalim alatima na tržištu, hardverski je zahtjevniji i teže je ovladati njegovim funkcionalnostima pa je u odnosu na ostale programe manje popularan. Korisničko sučelje programa *Houdini* prikazano je na slici 2.2.



Slika 2.2. *Korisničko sučelje programa Houdini* [6].

2.3. *Blender*

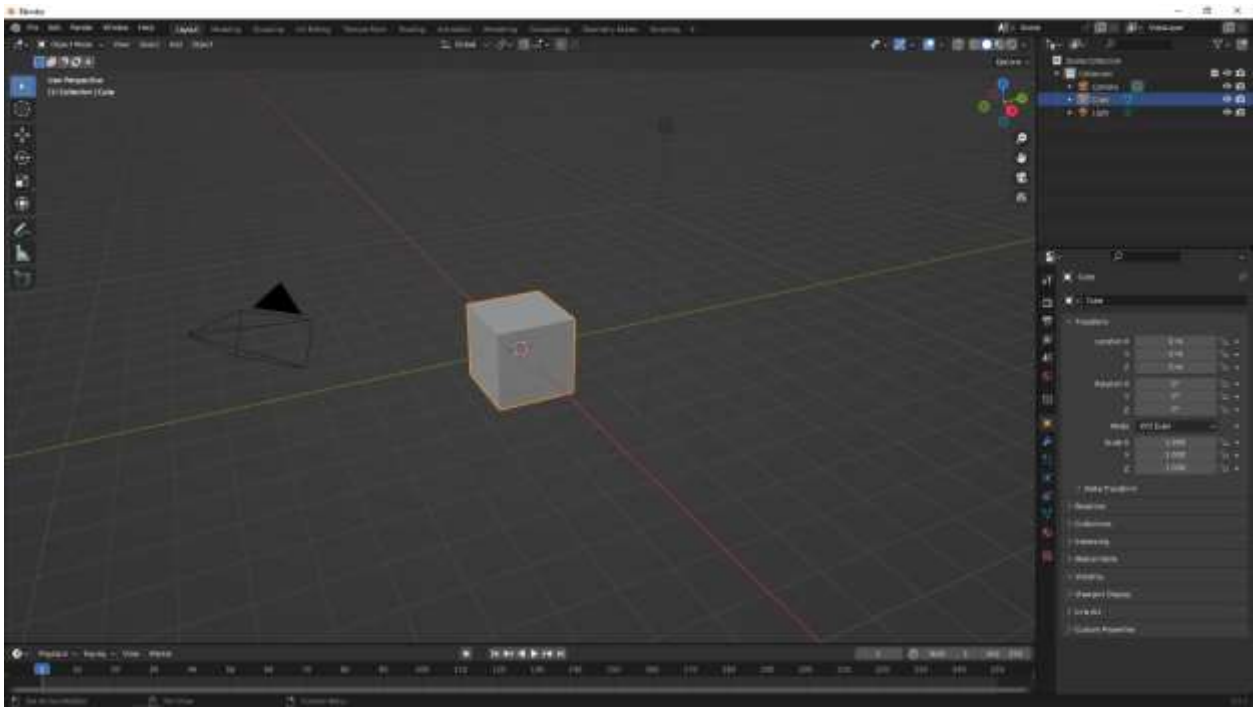
Blender je besplatan softver otvorenog koda (engl. *open source software*) što znači da svim korisnicima pruža mogućnost izmjene i nadogradnje programa kao i dodavanje novih korisnih funkcionalnosti te njihovo međusobno dijeljenje. Korisničko sučelje aplikacije prilikom početnog pokretanja prikazano je slikom 2.3. *Blender* sadrži širok raspon alata za modeliranje i omogućuje korisnicima obavljanje raznih operacija poput:

- Poligonalnog modeliranja
- Skulptiranja
- UV mapiranja
- Animacije objekata

Kao i prethodno navedeni programi, *Blender* uključuje kompatibilnost rada s ulančanim čvorovima pri kreiranju kompleksnih objekata ili geometrije. S verzijom 2.39 u *Blender* je dodana funkcionalnost geometrijskih čvorova koja se koristi u ovom radu. Korištenje geometrijskih čvorova pri kreiranju vlastitog projekta korisniku nudi sljedeće:

- Proceduralno generiranje objekata (geometrije)
- Parametrizirano upravljanje svojstvima objekata
- Olakšano ponavljajuće instanciranje objekata
- Proceduralne animacije objekata ili njihovih dijelova

Spomenuto je kako *Blender* pruža mogućnost nadogradnje i izmjene vlastitih ugrađenih funkcionalnosti kao i dodavanje novih te njihovo dijeljenje s ostalim korisnicima. Ta se primjena koristi u izradi ovog rada te se opisuje dalje u radu.



Slika 2.3. *Korisničko sučelje Blendera.*

2.4. Cilj rada

Cilj ovog rada kreiranje je aplikacije koja nudi proceduralno generiranje različitih oblika zgrada i mostova korištenjem geometrijskih čvorova pomoću aplikacije *Blender* uz olakšane izmjene njihovih svojstava (dimenzija, teksture, komponenti). Potrebno je pojednostaviti

proces prilagodbe objekata željama korisnika uz što manje koraka i potrebnog znanja te omogućiti efikasnu izgradnju čitavih gradova dodavanjem gotovih aplikacijskih priključaka programu. Također, navesti i objasniti rad s priključcima, alatima i programima koji se koriste pri izradi aplikacije. Uz to, navesti funkcionalnosti projekta te kroz primjere prikazati njihovu praktičnu primjenu.

3. KORIŠTENE TEHNOLOGIJE

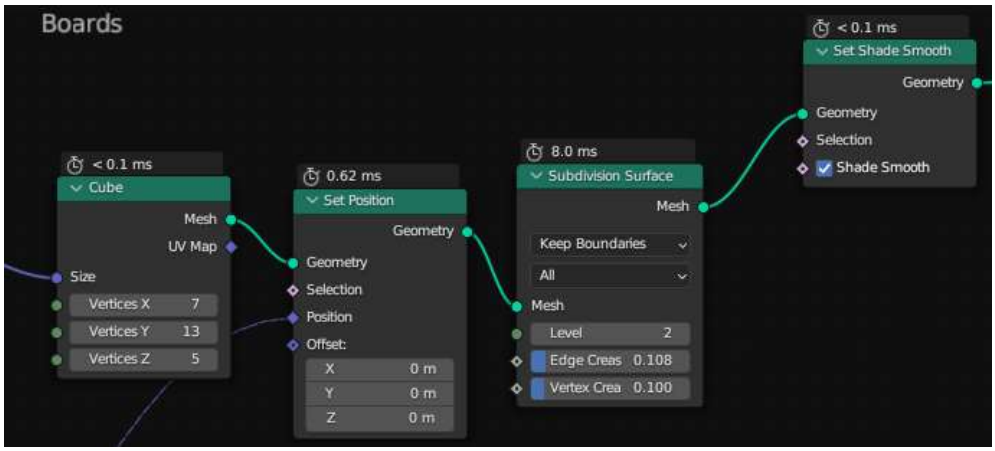
Kao što je spomenuto u prethodnom poglavlju, za izradu ovog projekta koristi se program *Blender*. U ovom dijelu rada opisat će se korišteni alati i priključci s pripadajućim funkcionalnostima i podprogramima.

3.1. Geometrijski čvorovi (engl. *geometry nodes*)

Rad s geometrijskim čvorovima [7] dodan je u program s verzijom 2.39. Korisničko sučelje (engl. *User interface*, UI) zasnovano je na poznatom principu rada s čvorovima koji je korisnicima prethodno bio dostupan u radu s ostalim podsustavima programa poput alata za sjenčanje (engl. *shading*) UV mapiranje ili čvorova materijala (engl. *material nodes*).

Pri radu s čvorovima, korisnici se oslanjaju na grafičko sučelje prikazano na slici 3.1. koje se sastoji od grupnih funkcija ili svojstava (čvorova) povezanih vezama. Čvorovi se mogu dodavati i povezivati ručno unutar korisničkog sučelja ili mogu biti unaprijed postavljeni kao dijelovi ugrađenih podustava kao što su sustavi za sjenčanje (engl. *shaders*). Sučelje čvorova koristi *Python API* (*Application Programming Interface*) kako bi se propisalo željeno ponašanje pojedinih čvorova i korisnicima pružilo sučelje za jednostavnu interakciju bez potrebe shvaćanja pozadinskog programskog koda.

Upotreba geometrijskih čvorova omogućuje korisnicima skraćen proces izgradnje složenih objekata ili čitavih cjelina na način da se propišu pravila za organizaciju jednostavnih objekata u cjelinu te se primjenom tih pravila na jednostavan objekt njegovim ponovljenim instanciranjem stvara kompleksna cjelina. Korisnik može propisana pravila parametrizirati i pružiti mogućnost jednostavne izmjene različitih svojstava kreiranih objekata (dimenzije, broj ponavljanja, pseudo-slučajne nepravilnosti pri generiranju).



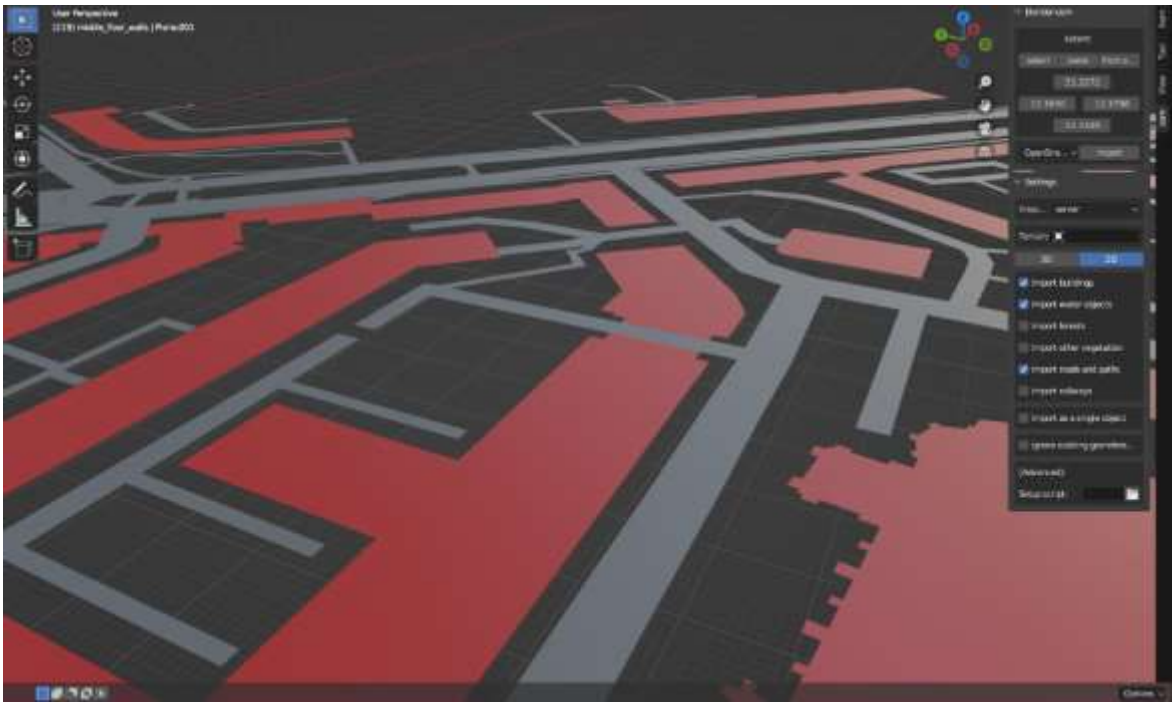
Slika 3.1. *Prikaz rada s čvorovima.*

3.2. *Open street map* (OSM)

Open street map dodatak je *Blenderu* pomoću kojeg je moguće u program uvesti podatke o stvarnoj lokaciji putem jednostavnog korisničkog sučelja. Pruža razna proširenja aplikacije korisnika od kojih su upotrebljena:

Uvoz podataka o dijelovima grada – OSM unosi podatke o odabranoj lokaciji putem geografskih koordinata o pozicijama ulica, željeznica, stambenih objekata te vodenih površina

Označavanje pozicija objekata – OSM označava predviđena mjesta za pojedine objekte različitim bojama kako bi korisnicima olakšao snalaženje u uvezenim podacima



Slika 3.2. *Prikaz uvezenih podataka putem OSM-a.*

Ostale funkcionalnosti koje OSM nudi su vizualizacija terena, pojednostavljen prikaz postojećih građevina te vizualizacija vegetacije uz skaliranje veličine uvezenih objekata. Primjer učitanih podataka prikazan je slikom 3.2., a dokumentacija dodatka i detalji o autorima uz dodatne funkcije koje se ne koriste pri izradi ovog rada nalaze se u popisu literature [8].

3.3. *Buildify* 1.0

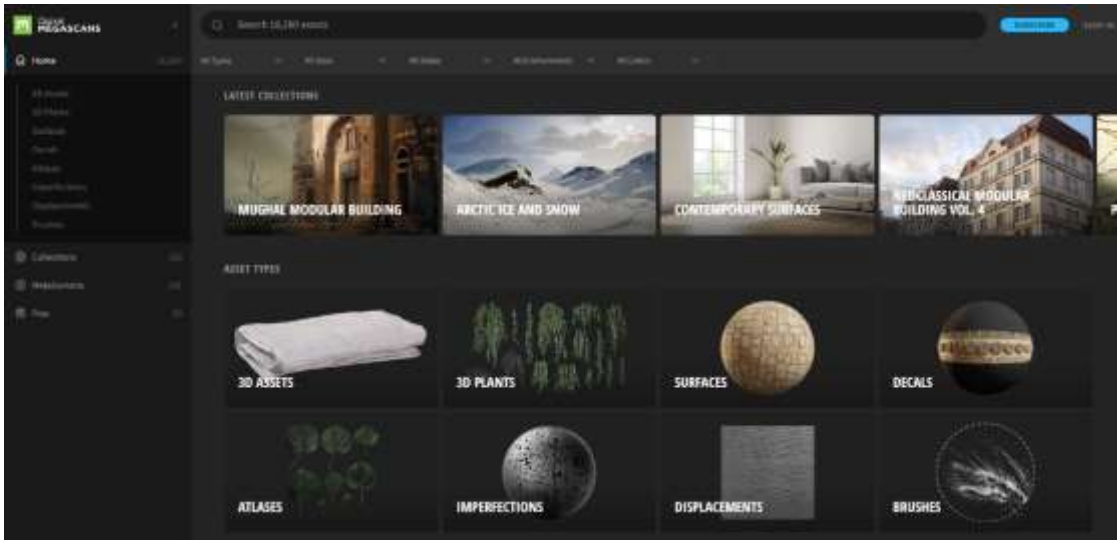
Buildify je korisnički projekt koji koristi geometrijske čvorove za proceduralno generiranje zgrada. Uz aplikaciju priloženi su pojednostavljeni elementi (dijelovi zgrade) koji se prema postavljenom skupu geometrijskih čvorova kombiniraju kako bi se od njih izgradila zgrada. Program se izvršava tako što se elementi za izgradnju objekta dodaju u pripadajuće skupine te se prema propisanom ponašanju u programskom rješenju instanciraju i dupliciraju kako bi stvorili cjelinu. U postavkama čvorova propisana je upotreba takvih elemenata na način da je svaki podsustav zadužen za upravljanje određenom skupinom elemenata i njihovo povezivanje s ostalima. U glavnom dijelu programa, podsustavi se povezuju i primjenom glavnog skupa čvorova na ravninu (2D objekt unutar aplikacije) iz nje se podiže zgrada.

Korisnicima je pružen skup parametara kojima mogu upravljati izgledom i svojstvima generirane zgrade poput broja katova, rasporeda i broja zidnih elemenata ili rasporeda elemenata na krovovima. Izmjenom tipova elemenata i svrstavanjem istih u određene skupine, korisnik može odabrati željeni izgled zgrade, a izmjenom oblika ravnine na koju se primjenjuje program, mijenja se oblik cijele zgrade.

Detalji o aplikaciji i autoru nalaze se u literaturi [9].

3.4. *Quixel bridge*

Quixel bridge aplikacija je koja služi kako bi se *Blender* projekt povezao s bazom podataka *Megascans* koja sadrži 3D modele, teksture i ostale komponente koje je moguće izraditi u *Blenderu*. Pomoću stranice *Quixel*, korisnici mogu izmjenjivati ili prodavati svoj rad drugima kako bi ih mogli koristiti u svojim projektima. Slika 3.3. prikazuje korisničko sučelje za pretraživanje *Megascans* baze podataka.



Slika 3.3. *Pretraživanje Megascans baze podataka.*

Nakon preuzimanja programa, potrebno ga je pokrenuti i povezati s *Blender* projektom. Pretraživanjem baze podataka elemenata moguće je uvesti određene elemente u projekt gdje će se kasnije moći koristiti kao dijelovi vlastitih objekata ili kao zasebne cjeline ovisno o ciljevima.

Važno je napomenuti da se preuzeti elementi u računalo pohranjuju lokalno te ukoliko korisnik želi dijeliti svoj projekt s drugima, potrebno je uz dijeljenu *Blender* aplikaciju pružiti pristup i mapi s korištenim elementima uvezenih programom *Quixel bridge*. Dodatni podaci o programu nalaze se u popisu literature [10].

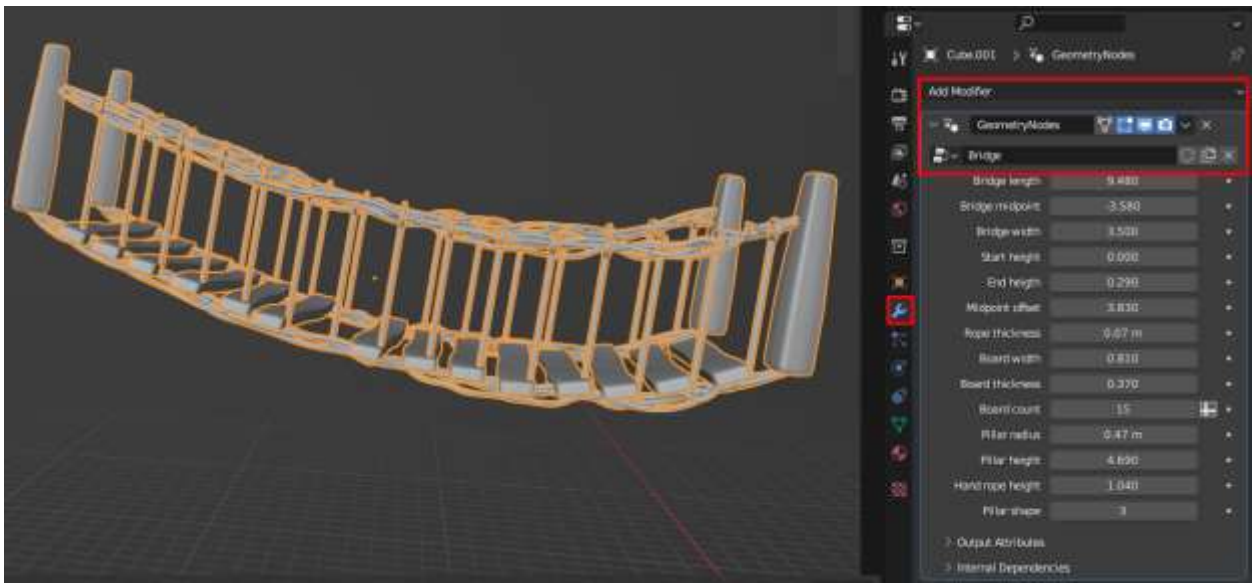
4. IMPLEMENTACIJA RJEŠENJA

U ovom poglavlju objasnit će se postupak kreiranja programskog rješenja i detaljno opisati koraci implementacije. Dodatno, predstavit će se upute za korištenje svih priključaka i dodataka korištenih pri izradi aplikacije. Kroz primjere će biti prikazani proceduralno generirani objekti te kako promjene različitih parametara utječu na njihov konačan izgled.

4.1. Proceduralno generiranje mostova

Set geometrijskih čvorova *Bridge* omogućuje parametrizirano proceduralno generiranje mostova, a u ovom dijelu rada opisat će se postupak stvaranja takve funkcionalnosti.

Prvo je kreiran početni 3D objekt, a zatim je na njega primijenjen modifikator (engl. *modifier*) te su u prozoru za uređivanje geometrijskih čvorova postupno dodavane povezane grupe čvorova zadužene za upravljanje dijelovima mosta. Postupak dodavanja modifikatora na objekt prikazan je slikom 4.1. Čvorove i pripadajuće veze nije potrebno grupirati, no kreiranjem ovakve aplikacije stvara se kompleksna struktura čvorova pa njihovo grupiranje osigurava lakše snalaženje pri izmjenama ili proširenjima. Grupa čvorova potrebna za proceduralno generiranje mostova prikazana je slikom 4.2.



Slika 4.1. *Dodavanje modifikatora na objekt.*

Grupa *Lower ropes* sastoji se od dva *Quadratic Bezier* objekta na čije su parametre (*start, middle, end*) priključeni čvorovi za upravljanje duljinom i zakrivljenosti objekata (*Combine* XYZ).

Ova grupa najvažniji je dio programskog rješenja jer služi kako bi se iz nje razvili ostali elementi koji čine most (daske, rukohvat, stupovi). Na nju se povezuje grupa *Lower rope thickness* koja upravlja debljinom užadi.

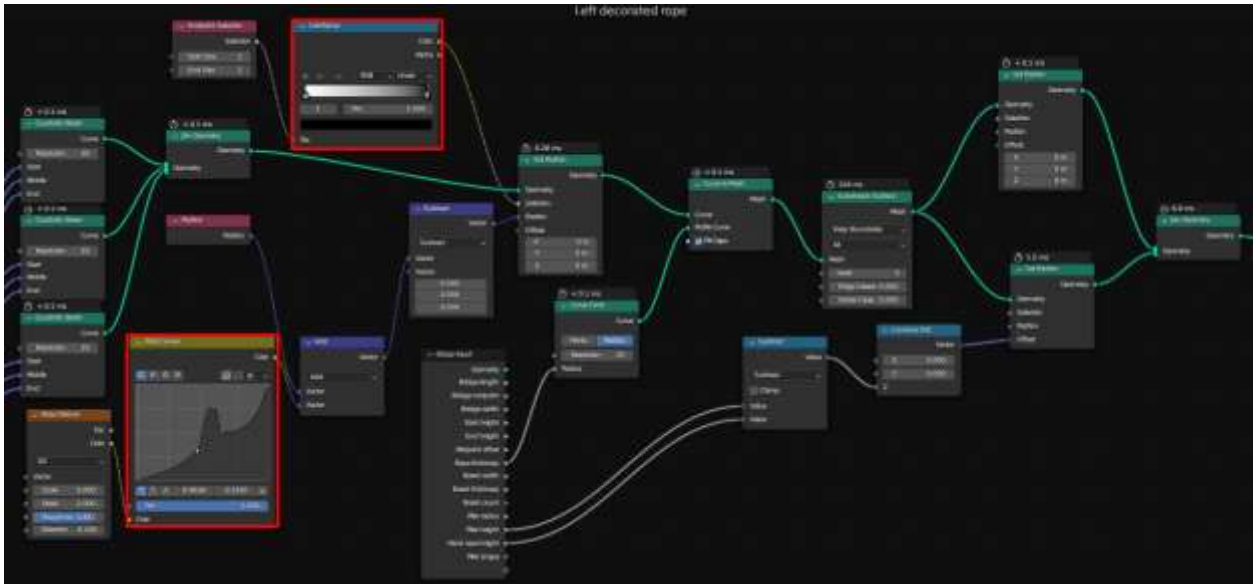
Grupa *Help middle rope* pomoćna je grupa i njeni elementi nisu vidljivi, a služi kako bi osigurala simetriju mosta. Predstavlja dužinu koja se pruža kroz sredinu mosta i koristi se kako bi se ostali elementi simetrično rasporedili na lijevu i desnu stranu.

Grupa *Boards* po mostu stvara daske koje se pružaju čitavom duljinom lijevog i desnog donjeg užeta. Grupa *Board arrangement* slaže daske po dužini mosta pomoću čvorova *Instance on points* i *Instance rotation,* a čvor *ColorRamp* koristi se kako bi se omogućio nasumičan izostanak pojedinih dasaka. U grupi *Board count* propisuje se gustoća postavljanja dasaka. Grupa *Board distortion* mijenja oblik i površinu postavljenih dasaka. Grupa *Delete overlap* uklanja po jednu dasku sa svakog kraja mosta kako ne bi došlo do preklapanja sa stupovima.

Grupa *Vertical ropes* podiže okomitu užad koja povezuje rukohvat s daskama. Njihova visina određena je kao razlika parametara visine stupova i visine rukohvata. Izražavanje pozicije elementa kao razlike ili zbroja parametara ostalih dijelova mosta osigurava pravilan odnos među elementima pri promjeni njihovih parametara.

Grupe *Left hand rope* i *Right hand rope* propisuju izgled i relativnu poziciju lijevog i desnog rukohvata u odnosu na donju užad. Pružaju se cijelom dužinom mosta i povezuju se na stupove na krajevima. U *Arrange left hand ropes* grupi propisuje se debljina i duljina lijevog rukohvata te se njegova širina povezuje sa širinom mosta što osigurava da izmjena širine mosta osigurava sukladno pomicanje pripadajućeg rukohvata. Analogno tome, kreira se *Arrange right hand ropes* grupa čvorova.

Grupe *Left decorated rope* i *Right decorated rope* modificiraju izgled rukohvata kako bi zaista izgledali kao užad. Ne mijenjaju teksturu užadi, već koriste čvorove distorzije kako bi dodali zakrivljenje na pripadajuće uže. Za to su korišteni *ColorRamp* i *RGB Curves* čvorovi. Za izmjenu izgleda rukohvata potrebno je otvoriti postavke geometrijskih čvorova jer takve izmjene znatno opterećuju računalo pa nisu izdvojene u skupu promjenjivih parametara kao na slici 4.1.



Slika 4.2. *Čvorovi izmjene izgleda rukohvata.*

Izgled i dimenzije stupova mosta postavljeni su u grupi *Pillar arrangement*. Zadan oblik stupova je zvijezda, a mijenja se izmjenom parametra *Pillar shape*. Čvor *Resample curve* stupovima daje izgled zaobljenog kamena, a kasnije se može i dodati tekstura površine. Veličina stupova može se promijeniti parametrom *Pillar radius*, a izmjenom u čvoru *ColorRamp* mijenja se debljina donjeg dijela stupa. Grupa *Rope props* povezuje se s grupom rukohvata i na njih dodaje prstenove čiji je radijus vezan za širinu rukohvata.

Neki od ostalih čvorova koji se često koriste, a nisu navedeni u opisu pojedinih grupa su: **Čvorovi matematičkih operacija (*Math node*)** – vraćaju rezultat matematičkih operacija povezanih parametara (zbroj, razliku ili umnožak)

Čvor krajnje točke (*Endpoint selection*) – odabire krajnju točku instanciranja objekta

Čvor postavljanja materijala (*Set material*) – postavlja teksturu povezanog objekta

***Noise texture* čvor –** stvara nepravilnosti na površini povezanog objekta

Čvor uklanjanja geometrije (*Delete geometry node*) – uklanja instancu objekta (dijela cjeline)

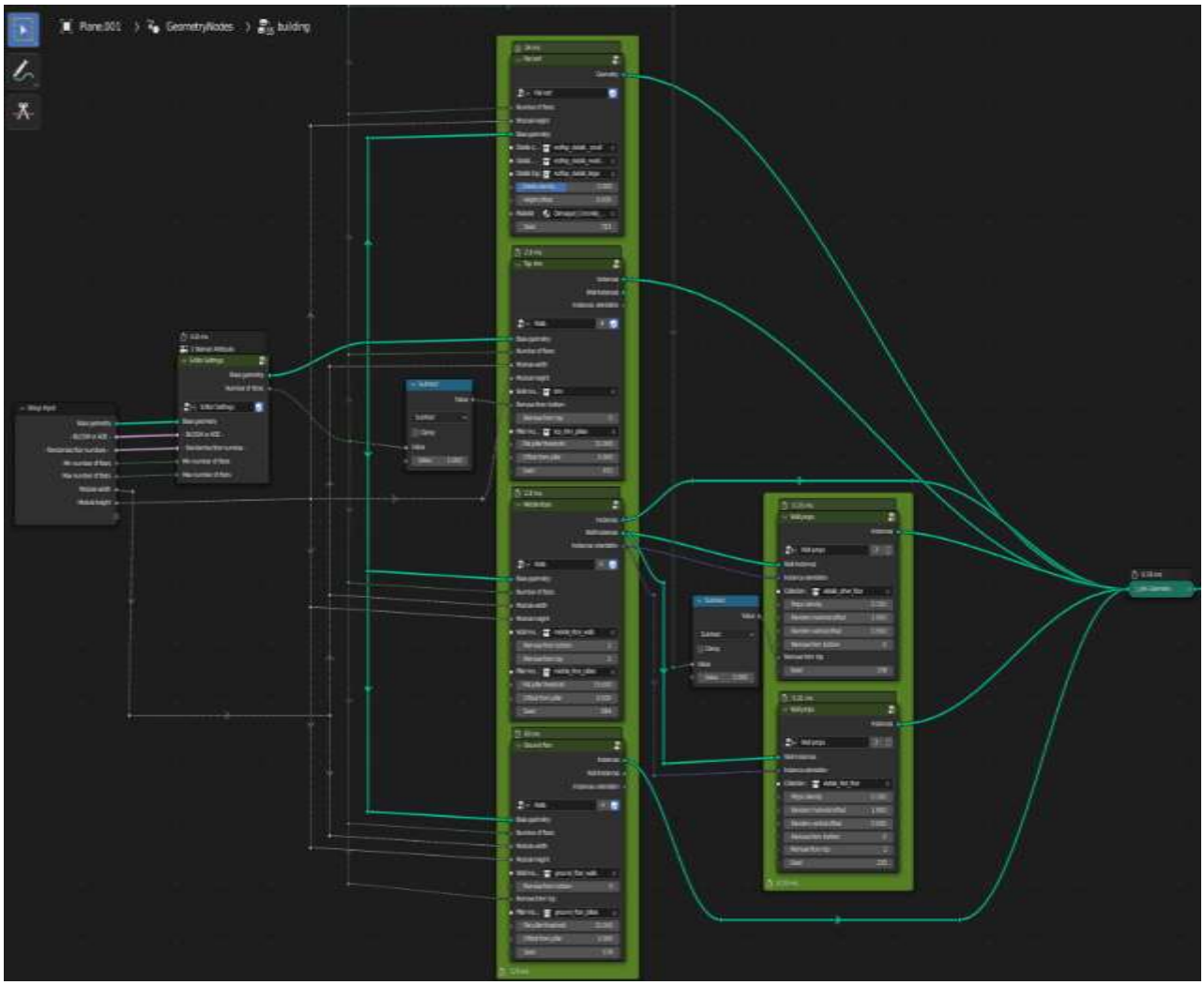
Čvor pridruživanja geometrije (*Join geometry node*) – spaja više geometrijskih objekata

Group input node – izdvaja željene parametre iz postavki geometrijskih čvorova i prikazuje ih u korisničkom sučelju kao na slici 4.1.

4.2. Proceduralna izgradnja zgrada

Buildify sadrži skup geometrijskih čvorova potrebnih za proceduralno generiranje zgrada. U ovom dijelu opisat će se rad s aplikacijom i objasniti način rada pripadajućih funkcija.

Set geometrijskih čvorova *building* sadrži podskupine funkcija zaduženih za pojedine dijelove zgrade koju generiraju i slažu u cjelinu. Program se izvodi tako da se, kao kod mosta, dodaje modifikator na objekt podloge. Program zatim proceduralno podiže zgradu tog oblika i omogućuje korisniku promjenu različitih parametara.

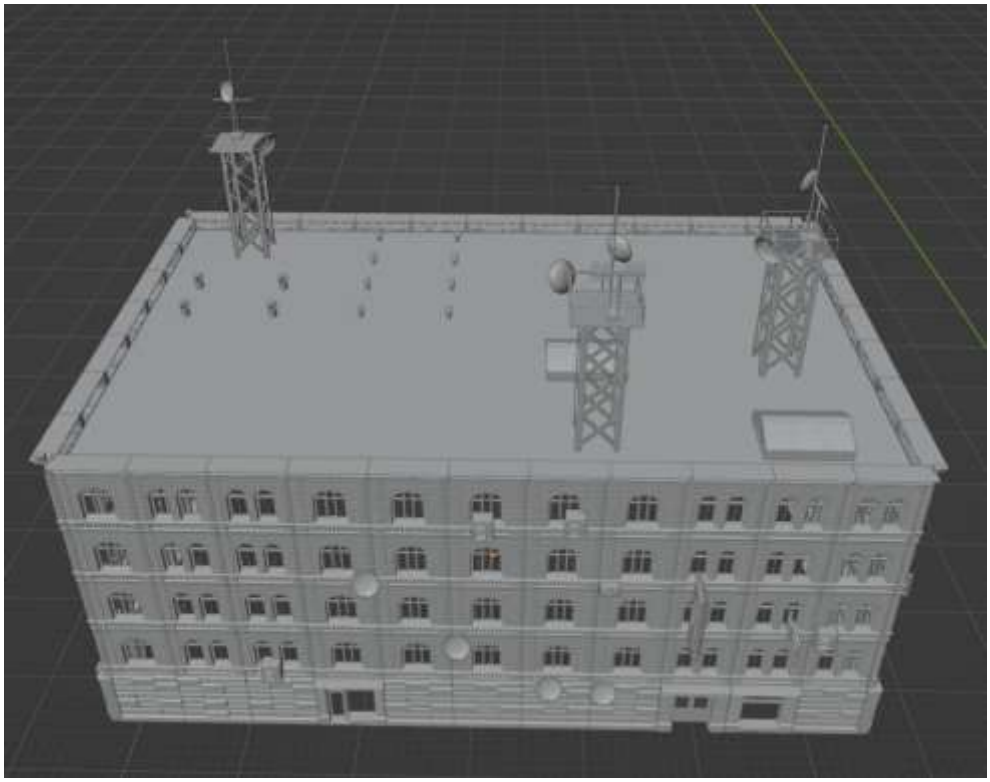


Slika 4.3. *Podsustavi geometrijskih čvorova building.*

Grupa *Top trim* slaže elemente obojene žuto, elemente srednjih katova plave boje raspoređuje grupa *Middle floors*, a prizemlje označeno crvenom bojom generiraju funkcije iz grupe *Ground floor*.

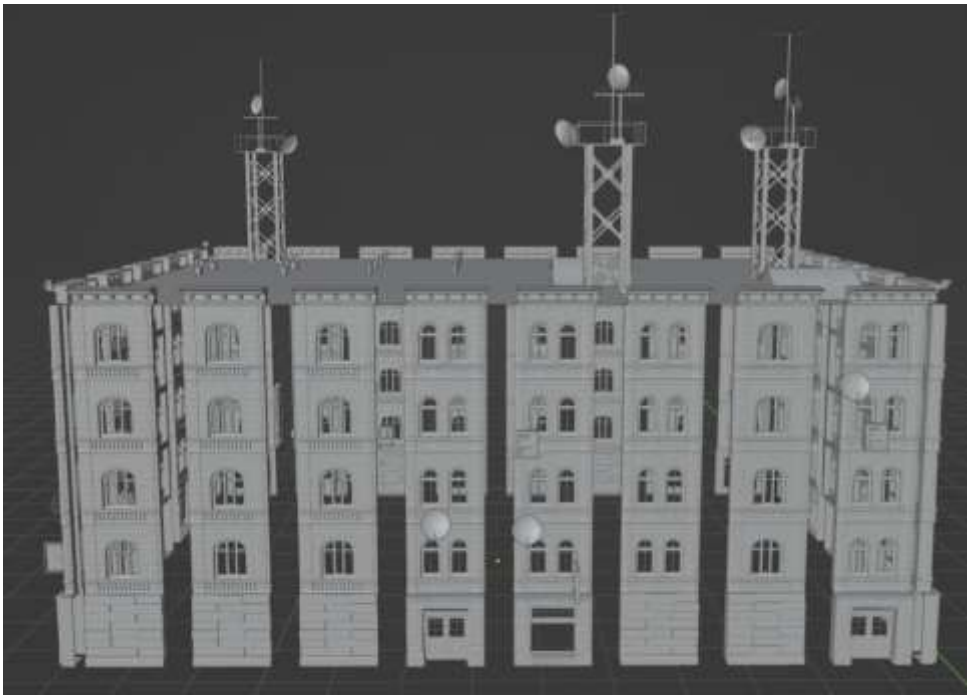
Slika 4.4. *Raspored odgovornosti podsustava po bojama.*

Ravan krov na vrh zgrade postavlja se pomoću podsustava *Flat roof*. Dijelovi zgrade označeni na slici 4.4. postavljaju se pomoću čvorova prikazanih na slici 4.3.



Slika 4.5. *Prikaz krova zgrade.*

Modeli pomoću kojih se kreira zgrada mogu se stvoriti ili preuzeti s interneta i moraju biti prisutni unutar *Blender* aplikacije te pridruženi odgovarajućim spremnicima. Modeli moraju biti propisanih dimenzija kako bi se osiguralo pravilno ponašanje programa. Ukoliko dimenzije i orijentacija modela nisu pravilni, generirat će se zgrada s neželjenim deformacijama kao što je prikazano slikom 4.6. Također, s obzirom na to da aplikacija povezuje modele u cjelinu pomoću izvorišne točke (engl. *Point of origin*) objekta, važno ju je precizno postaviti kako bi se osiguralo pravilno ponašanje aplikacije. Izvorišna točka postavlja se na donju ravninu objekta te se pomiče horizontalno dok se ne postigne željeni rezultat.

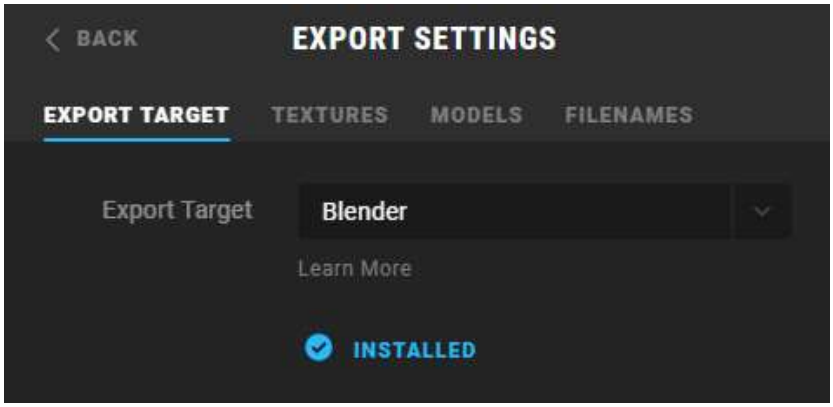


Slika 4.6. *Zgrada generirana od elemenata nepravilnih dimenzija.*

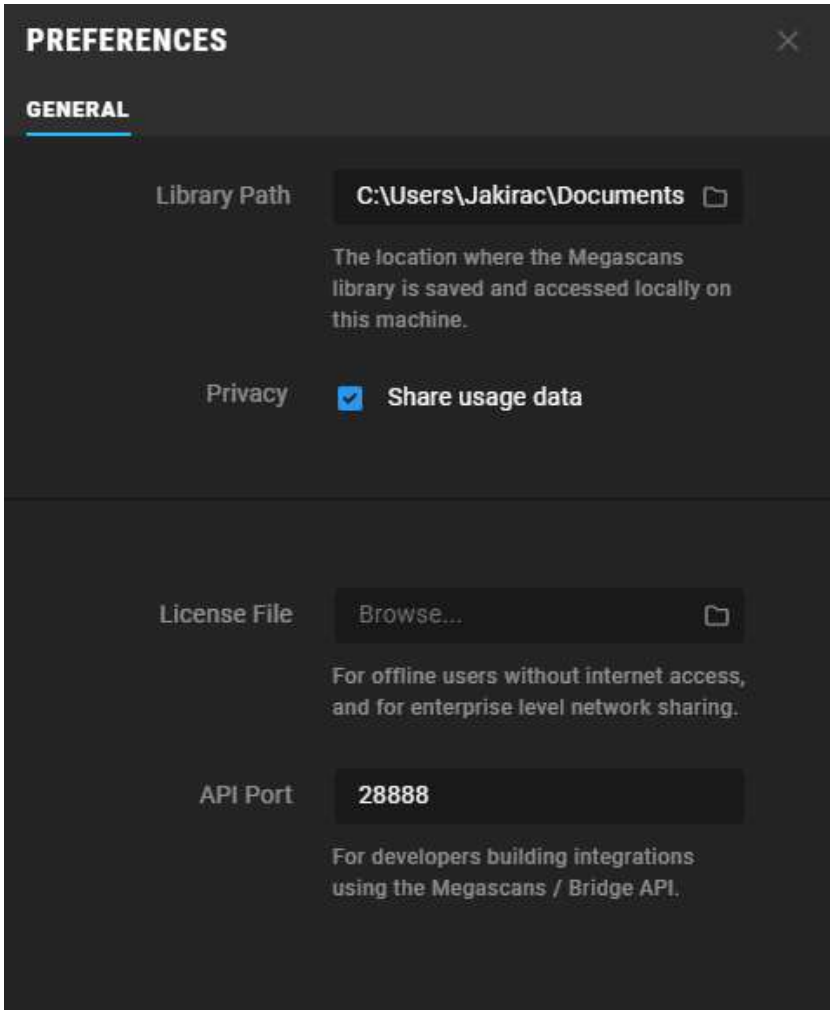
4.3. Dodavanje tekstura i modela iz *Megascans* biblioteke

Kroz prethodne primjere vidi se kako aplikacijom *Buildify* korisnik od nekoliko modela zidova, vrata ili prozora te jednog modela ravnog krova lako može generirati cijelu zgradu postavljanjem modifikatora na objekt baze. Ipak, takav pristup zahtijeva znanje kako napraviti 3D modele u *Blenderu*. To sužava krug ljudi kojima ovo programsko rješenje može biti korisno. Korištenje programa *Quixel bridge* nudi dodavanje gotovih modela koji se mogu koristiti pri generiranju zgrada. Tako se znatno smanjuje znanje potrebno da se stvore kompleksni objekti. Korisnici i dalje mogu sami stvarati svoje modele, no oni koji žele ih mogu preuzeti iz biblioteke kroz korisničko sučelje programa *Quixel bridge* prema uputama sa slika 4.7. i 4.8.

Prvo je potrebno preuzeti aplikaciju i prijaviti se sa svojim korisničkim računom. Nakon toga, moguće je pretraživanje *Megascans* biblioteke. Potrebno je odrediti apsolutnu putanju pohrane modela koji se žele preuzeti i postaviti *API port* na vrijednost 28888. U postavkama aplikacije potrebno je kao metu postaviti *Blender*, što će pokrenuti automatsko instaliranje potrebnih priključaka kako bi se aplikacija mogla povezati s *Blender* projektom.



Slika 4.7. *Postavke izvoza podataka.*



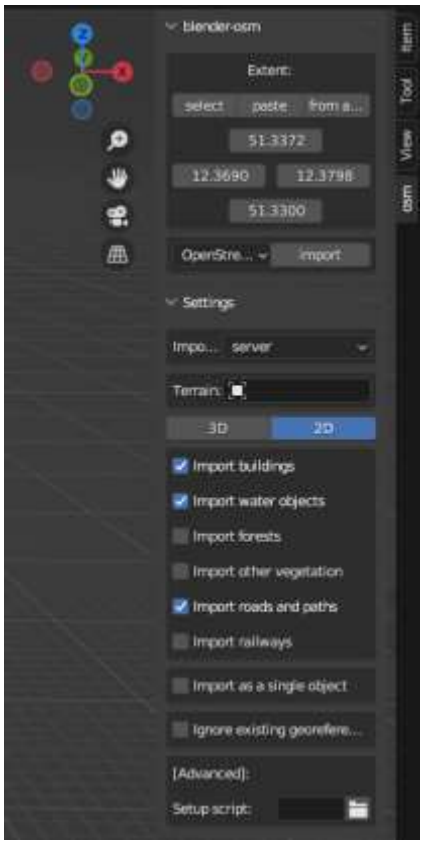
Slika 4.8. *Postavke programa.*

Nakon toga potrebno je dodati priključak u željeni *Blender* projekt unutar upravitelja priključcima što omogućuje prijenos podataka. Ovim postupkom omogućeno je dodavanje gotovih modela u *Blender* projekt pritiskom na gumb *Export* u korisničkom sučelju programa.

4.4. OSM integracija

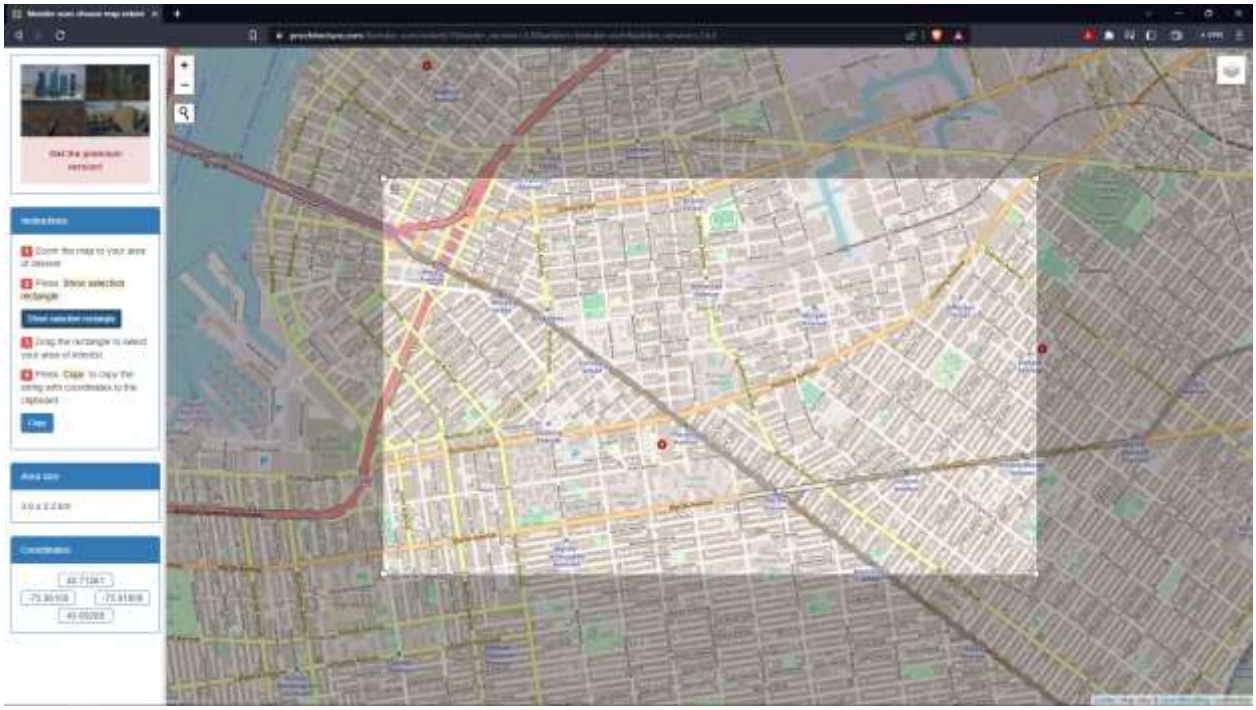
Priključak OSM koristi se kako bi ubrzao proces generiranja grada koristeći definirane setove geometrijskih čvorova. Priključak se preuzima sa stranice u literaturi [8], a u *Blender* projekt dodaje se pomoću upravitelja priključcima.

Nakon instaliranja priključka, sučelju se pristupa unutar *Blender* projekta. Pomoću geografskih koordinata određuje se geolokacija koja se vozi u projekt pritiskom na gumb *import* kako je prikazano na slici 4.9.



Slika 4.9. *Učitavanje željenih podataka korištenjem OSM-a.*

Drugi način učitavanja lokacije u projekt pokreće se pritiskom na gumb *select*. Tako se otvara internetski preglednik i povezuje s interaktivnim lokacijskim servisom. Tada se na karti odabiru podaci koji će biti učitani u projekt. Postupak je prikazan slikom 4.10.

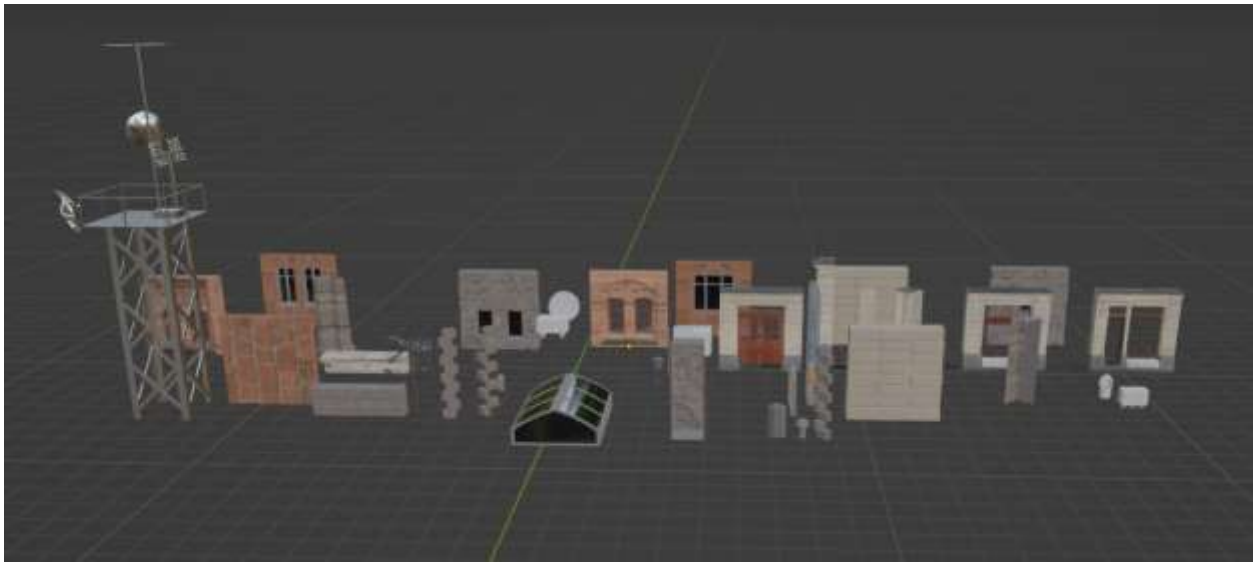


Slika 4.10. *Odabir lokacije za učitavanje.*

Primjer učitanih podataka o gradu prikazan je slikom 3.2. Primjenom modifikatora *building* na predviđena (crvena) mjesta učitana OSM-om, aplikacija generira zgrade i omogućuje izmjenu njihovih parametara.

4.5.Testiranje i prilagodba

Prilikom testiranja aplikacije, potrebno je pojedine dijelove zgrade svrstati u pripadajuće skupine kako bi aplikacija pravilno generirala zgrade. Ti objekti modeliraju se u *Blenderu* te moraju biti prisutni u projektu i ne ovise o ostatku projekta. Raspored elemenata po grupama i njihov izgled u *Blenderu* prikazan je slikama 4.11. i 4.12. U aplikaciji, nalaze se udaljeni od zgrada koje se generiraju, a moguće ih je i učiniti nevidljivima kada se postigne željeni izgled modela.

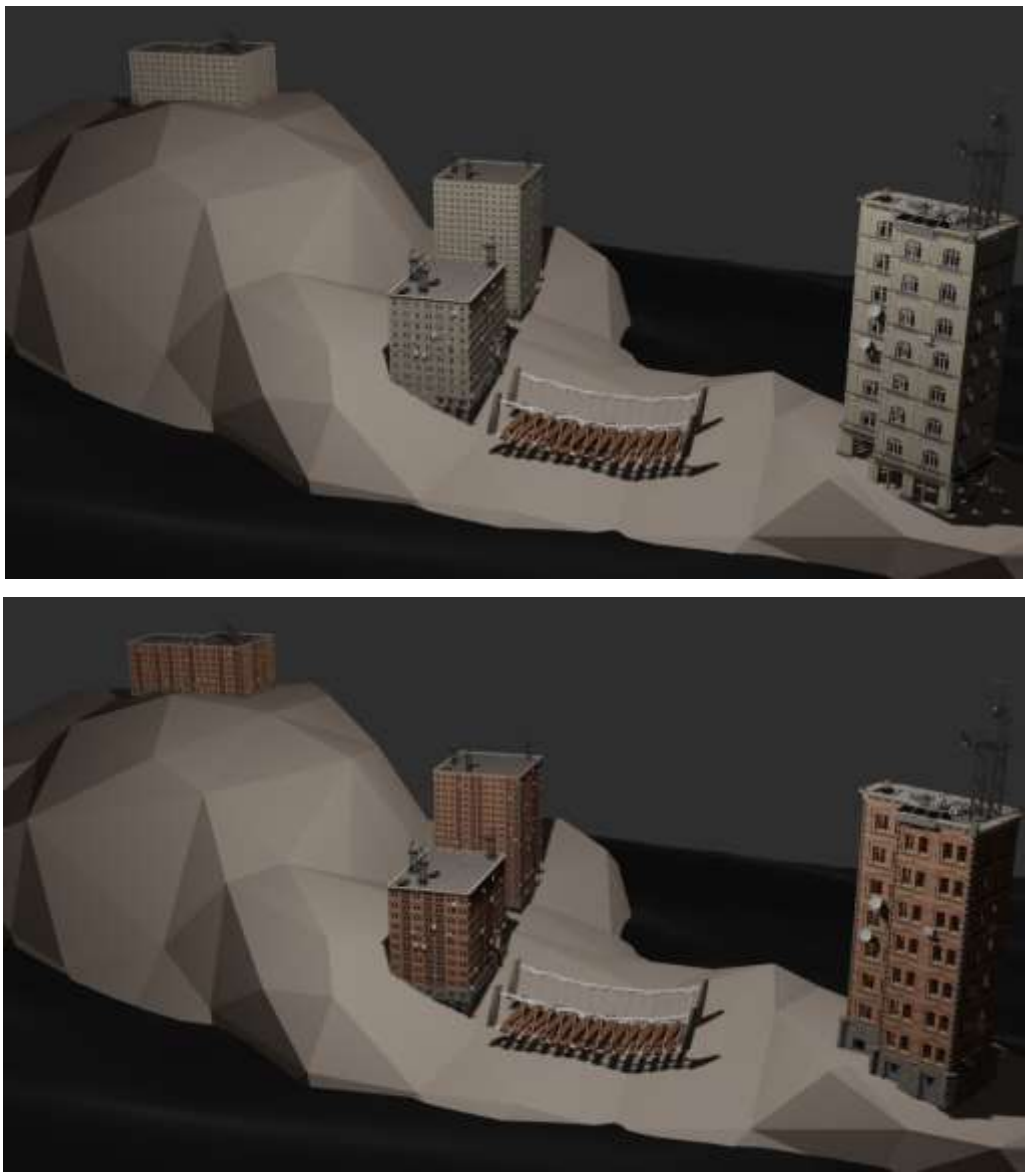


Slika 4.11. *Objekti za generiranje zgrada.*

Slika 4.12. *Raspored elemenata po grupama.*

Aplikacija prilikom generiranja zgrada iskoristiti sve elemente pojedinih grupa te zato nije moguće stvoriti zgrade koje se razlikuju po komponentama, već samo po njihovom rasporedu

i dimenzijama. Takvo proširenje aplikacije zahtijevalo bi dupliciranje setova geometrijskih čvorova i određivanje koji će upravljati kojom grupom. Stoga se raznolikost u ovoj aplikaciji ostvaruje ručnom izmjenom modela po grupama na način da se izostave elementi koji neće biti korišteni. Slikom 4.13. prikazan je isti kadar s različitim kombinacijama korištenih elemenata.



Slika 4.13. *Građevine generirane korištenjem različitih elemenata.*

Prilikom izmjene elemenata ili kreiranja novih, potrebno je pratiti upute u dokumentaciji [11]. Primjenom modifikatora *building* na podatke učitane dodatkom OSM, generiraju se zgrade na predviđenim mjestima. Na slikama 4.14., 4.15. i 4.16. prikazan je rezultat povezivanja *Blender* OSM-a s aplikacijom iz više uglova.

Slika 4.14. *Grad popunjen proceduralno generiranim zgradama.*

Slika 4.15. *Grad popunjen proceduralno generiranim zgradama.*

Slika 4.16. *Ulica popunjena proceduralno generiranim zgradama.*

Prilikom testiranja aplikacije, optimalan je rezultat postignut pri radu s jednostavnim zgradama s minimalnim brojem oštrih kutova i bez nepravilnih ili zaobljenih oblika. Aplikacija teško prepoznaje i postavlja rubne elemente pa u takvim slučajevima može doći do neželjenih deformacija. Također, gustoća zidnih i krovnih elemenata osjetljiva je na ekstremne dimenzije zgrada pa je poželjno ukloniti ili urediti zgrade prevelikih ili premalih dimenzija kako bi takvi elementi bili pravilno postavljeni.

Za prikazivanje slika, u *Blenderu* su iznad grada postavljena 2 točkasta izvora svjetla (engl. *Point light source*) te je za generiranje slike korišten *Eevee rendering engine* [12].

5. ZAKLJUČAK

U ovom radu, kreirana je aplikacija koja pruža korisnicima proceduralno generiranje različitih građevina te prilagodbu njihovih svojstava korištenjem geometrijskih čvorova u *Blenderu*. Utvrđeni su problemi i ograničenja modeliranja takvih objekata bez korištenja ove tehnologije i ponuđeno je nekoliko programa koji nude stvaranje programskog rješenja na sličan način.

Detaljno je objašnjen način rada navedene tehnologije te je pruženo korisničko sučelje unutar aplikacije za parametriziranu izmjenu svojstava generiranih objekata. Pravilan rad aplikacije osiguran je korištenjem gotovih priključaka unutar programa. Definirani su pojmovi potrebni za shvaćanje i pravilno korištenje aplikacije kao i za njenu izmjenu i prilagodbu prema potrebama različitim potencijalnim korisnicima.

Pregledom postojećeg programskog rješenja, prilagođen je njegov način rada za potrebe ove aplikacije te je omogućena izgradnja različitih vrsta objekata na isti način. Korišteni priključci kao i njihovi autori navedeni su u popisu literature.

Ova aplikacija može se koristiti u područjima koja zahtijevaju efikasno stvaranje modela različitih građevina, a neka od njih su arhitektura, razvoj video igara ili filmska industrija.

LITERATURA

[1] X. Ren, L. Jiang, X, Tang, W. Liu, 3D Wireframe Modeling and Viewpoint Estimation for Multi-Class Objects Combining Deep Neural Network and Deformable Model Matching, MDPI, 2019, dostupno na: https://www.mdpi.com/2076-3417/9/10/1975 [24.6.2023.]

[2] J.P. Guimaraes, Cinema 4D Xpresso Nodes, Scribd, 2017, dostupno na: https://www.scribd.com/document/352262123/Cinema-4D-Xpresso-Nodes [30.6.2023.]

[3] T. Šantić, Programski jezik Python, 2017.

[4] Preuzeto s https://www.google.com [24.6.2023.]

[5] V. Skala, Procedural modeling in theory and practice, ResearchGate, 2010, dostupno na: https://www.researchgate.net/publication/45539668_Procedural_modeling_in_theory_and_practice [30.6.2023]

[6] Preuzeto s https://www.sidefx.com/products/houdini [24.6.2023.]

[7] J. Lampel, Procedural Modeling with Blender's Geometry Nodes: A workshop on taking advantage of the Geometry Nodes feature in Blender for procedural modeling, ACM Digital Library, 2022, dostupno na: https://www.mdpi.com/2076-3417/9/10/1975 [30.6.2023.]

[8] Prochitecture, Blender-OSM: OpenStreetMap and Terrain for Blender, Gumroad, 2020, dostupno na: https://prochitecture.gumroad.com/l/blender-osm [24.6.2023.]

[9] P. Oliva, Buildify 1.0, Gumroad, 2022, dostupno na: https://paveloliva.gumroad.com/l/buildify [24.6.2023.]

[10] T. Bergsman, W. Azim, Quixel bridge, Quixel 2011, dostupno na: https://quixel.com/bridge [24.6.2023.]

[11] P. Oliva, Buildify_1.0, Google Docs, 2022, dostupno na: https://docs.google.com/document/d/1xOR41G5D2-gxJGu3MyR5Oi-2VJvoz7js11W7Cit-G-Q/edit#heading=h.flj86qbo0n04 [26.8. 2023.]

[12] Shading, Lighting, and Rendering with Blender’s EEVEE, Github, 2018, dostupno na: https://github.com/PacktPublishing/Shading-Lighting-and-Rendering-with-Blenders-EEVEE [27.8.2023.]

SAŽETAK

Cilj ovog rada bio je napraviti *Blender* aplikaciju koja omogućuje proceduralno generiranje građevina koristeći geometrijske čvorove. Za stvaranje programskog rješenja korišten je priključak *Open Street Map* (OSM) i *Quixel bridge* aplikacija za uvoz 3D modela iz internetske baze podataka *Megascans.* Za proceduralno generiranje zgrada korištena je *Blender* aplikacija *Buildify* te je po uzoru na nju ugrađena funkcionalnost proceduralnog generiranja mostova. U korisničko sučelje izloženi su parametri koji mijenjaju svojstva generiranih objekata kako bi se olakšala njihova izmjena, odnosno kako bi bila moguća bez otvaranja i promjene postavki geometrijskih čvorova. Generiranje gradova omogućeno je integracijom OSM-a. Pri testiranju aplikacije potrebno je pravilno postaviti dimenzije objekata koji se koriste kako bi se postigli željeni rezultati. Aplikaciju je moguće proširiti dodavanjem podrške za rad sa zgradama kompleksnih oblika.

Ključne riječi: *Blender*, geometrijski čvorovi, građevina, proceduralno generiranje

ABSTRACT

The goal of this project was to create a Blender application which allows procedural generation of buildings using geometry nodes. Open Street Map (OSM) was a plug-in used in creation of this application, as well as the Quixel bridge application used for importing 3D models from a network library called Megascans. A blender application Buildify was used to enable procedural generation of buildings and, by its example, the function to procedurally generate bridges was added. Parameters which alter the properties of buildings are presented in the user interface in order to make their customization easier, precisely, making it possible without having to open the geometry nodes setup. City rendering was enabled by the use of the OSM plug-in. While testing this application, it's necessary to set the proper dimensions of objects used in order to ensure good results. It is possible to extend the application by adding support for working with buildings of complex shapes.

Key words: Blender, building, geometry nodes, procedural generation