3D vizualizacija zgrade FERIT-a

Mikić-Vučak, Mario

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Stručni studij informatike

3D vizualizacija zgrade FERIT-a

Završni rad

Mario Mikić-Vučak

Osijek, 2019.

Sadržaj

1. UVOD

U današnje doba 3D vizualizacija je postala neophodan element prilikom planiranja izgradnje arhitektualnih rješenja. Pomoću vizualizacije lako možemo doživjeti kako će se idejno rješenje uklopiti u okoliš te hoće li imati kakve nedostatke ili mane. Cijeli proces počinje od ideje i prijedloga koncepta preko nacrta do prve 3D vizualizacije. Često se upravo zbog pomoći vizualizacije prilikom dizajniranja zgrade, završni izgled zgrade skoro u potpunosti promjeni. Iako su ljudi vizualna bića, ponekad im se jako teško snaći u velikoj količini crta i brojeva gledajući nacrte pa čak i samim 3D dizajnerima koji se s time svakodnevno bave.

3D Vizualizacija može doći do zavidne razine realističnosti u prikazu koncepta ili završnog rješenja zgrade. U većini slučajeva o toj razini realističnosti ovisi količina detalja koja je ponekad nepotrebna pa nam rezultat zgrade izgleda pre-savršeno, ne realno. S obzirom na potrebe dizajnera i klijenta sasvim je dovoljan prikaz zgrade koja nigdje nije prljava, niti ima vizualne efekte korištenja kao što su npr. tragovi guma automobila na prilazima, slijevanje kišnice po fasadi, rast raznih trava uz rub zgrade i slično na što smo navikli u realnom životu. Zbog manjka takvih detalja vizualizacija nam zna izgledati pre-savršeno, te se može odmah raspoznati da je riječ o 3D vizualizaciji. No za potrebe dizajnera i klijenata je to sasvim dovoljno kako bi vidjeli finalni rezultat samog dizajna, njegovog uklapanja u okolinu te općeniti dojam.

Proces same izrade vizualizacije traži poznavanje različitih grana unutar multimedije, fizike, biologije i sl. Kako bi vizualizacija bila što realističnija potrebno je poznavati fiziku svjetlosti, ponašanje različitih svjetlosnih tijela i njihovih fizikalnih obilježja, npr. ponašanje svjetlosti okoline pred zalazak sunca. Sjene su jako bitan faktor a one su dobivene svjetlom. Materijali u samoj vizualizaciji mogu učiniti zgradu lijepom ili ružnom u koliko se ne postave kako treba. Potrebno je poznavati refrakciju stakla, atribute drveta, metala, plastike i ostalih fizikalnih elemenata. Modeliranje same zgrade je dug proces, zahtjeva milimetarsku točnost i razumijevanje nacrta te njegovog opisa.

Cilj ovog rada je upoznavanje s općenitim procesom izrade vizualizacije na primjeru zgrade FERIT-a. Postoji više vrsta vizualizacija i više načina na koji se vizualizacija može izraditi no u ovom radu ćemo vidjeti proces izrade arhitektualne vizualizacije.

1.1 Zadatak završnog rada

U radu prikazati realistični prikaz zgrade u 3D računalnoj vizualizaciji. Interijer i eksterijer ustanove na njenoj geo-lokaciji treba biti što realističniji, to postići korištenjem modrenih alata za 3D grafičko modeliranje. Obilazak zgrade i prikaz njene unutrašnjosti na način "Walktrough", što je uobičajeno u prikazu u virtualnoj stvarnosti.

2. Teorija i korištene tehnologije

U najosnovnijim pojmovima, 3D vizualizacija je dvodimenzionalni prikaz modela računalnog 3D modela koji je dobio svojstva kao što su tekstura, boja i materijal. Svakodnevno viđate 3D prikazivanja, ali većinu vremena vjerojatno niti primijetili. Naravno, postoje očigledne vizualizacije poput igranog filma priče o igračkama. No, vjerojatno biste se iznenadili kad saznate da je većina reklama o proizvodima, stvorena korištenjem 3D vizualizacije prikazivanja. Također, u koliko ste ikada pretraživali npr. dizajn eksterijera ili interijera kuća ili zgrada na internetu, vrlo vjerojatno ste gledali 3D vizualizacije.

2.1. Proces vizualizacije

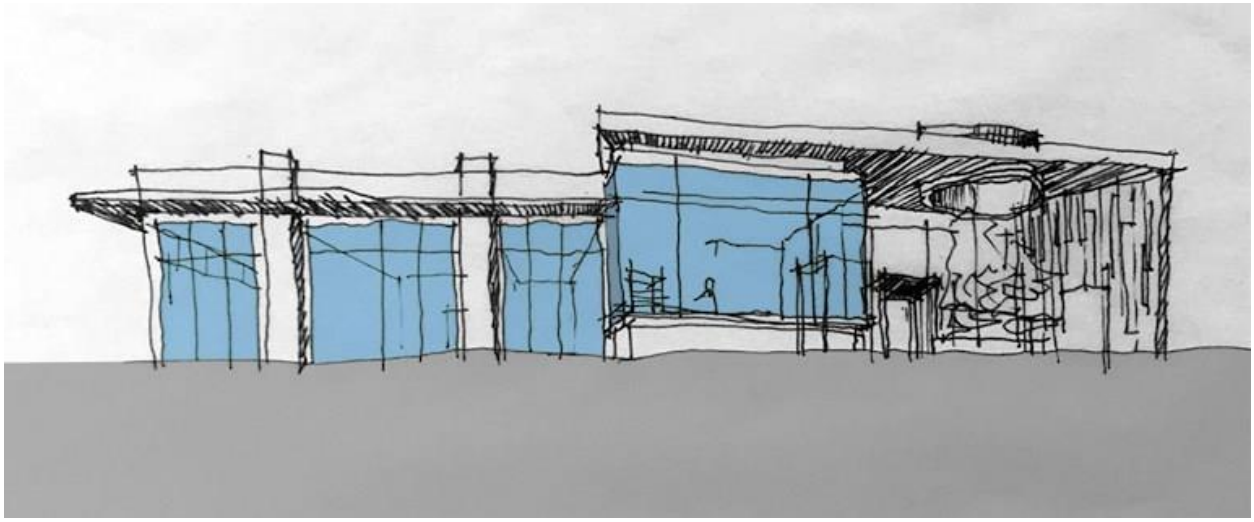
U slijedećim koracima ćemo proći općeniti proces vizualizacije. Proces vizualizacije je kopleksan i večinom ga za svaki pojedini dio radi minimalno jedna osoba pa do cijelog tima. Iako je moguće sve odraditi samostalno, za veće vizualizacije postaje prekompleksno i vremenski zahtjevno. U najkraćim crtama proces vizualizacije bi se mogao prikazati kao:

- Prikupljanje materijala (Fotografiranje, nacrti, usmena predaja, zapisi…)
- Modeliranje (Izrada objekata, zgrada, stolica, prozora…)
- Teksturiranje (Izrada materijala, drvo, metal, list drveta, ljudska koža…)
- Osvjetljenje (Dnevno svjetlo, noćni ambijent, nizajn osvjetljenja zgrade…)
- Postavljanje scene (Dizajn okoliša, teren, satelitske snimke, animacija…)
- Vizualizacija (Postavljanje kamere, globalne opcije, opcije rendera…)

2.1.1. Prikupljanje materijala

Prije nego što počnete s modeliranjem i ostalim koracima prema izradi vizualizacije, potrebno je upoznati se i naći što više opisnih detalja i materijala za vaš projekt. Upoznavanje s projektom je ključan dio za kvalitetnu vizualizaciju i optimiziranu vremensku potrošnju za izradu projekta. Materijali se odnose na slijedeće stavke:

1. Tehnički nacrti – Konceptualno riješenje, završno riješenje, najćešće su u cad formatu ili u drugom vektorskom formatu, može ih se dobiti i kao isprintana verzija ali ona je znanto teža za korištenje te se mora uložiti dodatno vremena za rekonstrukciju vektora iz skeniranog primjera nacrta.
2. 3D objekti – Arhitekti često imaju osobu koja radi 3D objekt iz nacrta što vam dodatno pomaže i skraćuje vrijeme izrade vizualizacije. Najčešći problemi kod takvih objekata su prebacivanje iz formata u format koji vama odgovara prilikom čega se zna izgubiti slojevitost modela, teksture (ako ih ima), događa se triangulacija te se na model mora utrošiti dodatno vrijeme za popravljanje vertexa i poligona. Zna se dogoditi da je 3D model brže izmodelirati iznova (u koliko imate druge materijale kao što su nacrti) nego ga popravljati.
3. Opisni elementi – Opisni elementi su pisani dokumenti o objektu opisnog karaktera. Velikog je značaja kad ne razumijete što je što u nacrtu i koje boje bi trebao biti zid a koje boje bi trebala biti fasada. Potrebno je tražiti što više opisnih elemenata odmah u startu jer ljudi obično misle da se to podrazumjeva.
4. Fotografije – Fotografije primjera sličnih zgrada, interijera, namještaja, boja, materijala i slično su odličan referentni materijal.
5. Skice – Ručno nacrtane skice na bilo kakvom mediju, služe kao dodatni opisni materijal.
6. Usmena predaja – Najčešće korištena u trenucima kad niti jedan materijal ne daje odgovor na vaše pitanje. Pozivi i integracija klijenta u projekt je bitan dio zbog same krajnje izvedbe i smanjenog opsega prepravaka modela, materijala, svjetla i na kraju vizualizacije.



Slika 2.1.1.1 Primjer ručno nacrtane skice

2.1.2 Modeliranje

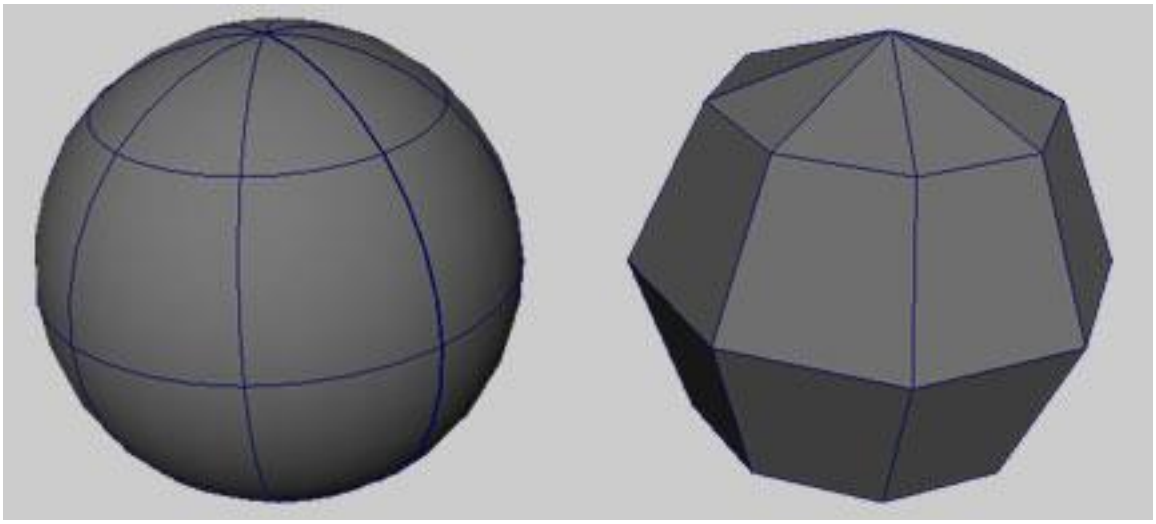
3D modeliranje je proces stvaranja tri-dimenzionalnog modela nekog objekta. Koristeći 3D, moguće je replicirati veličinu, oblik i teksturu stvarnog ili izmišljenog objekta.

Modeliranje se grana na 3 grane:

- Poligonalno
- Modeliranje krivuljama

Mi ćemo koristiti poligonalno modeliranje. Svaki tip modeliranja ima svoje prednosti i mane. Ovisno o tome modelirate li tehničku prirodu ili organsku prirodu izbor će vam biti drugačiji. Najbolje je poznavati obje vrste modeliranja s obzirom da se ovisno o primjeni s jednom i drugom može ubrzati posao i dobiti bolji rezultati. Postoji više tipova modeliranja krivuljama no to nama nije bitno trenutno. Najbitnija razlika između ove dvije glavne metode je u prezentaciju samog 3D objekta. Poligonalno modeliranje se sastoji od točaka koje su međusobno povezane linijama tako čineći plohu. Najmanja ploha *eng.“surface“* se sastoji od 3 točke *eng.“point“* i 3 linije koje se nazivaju *eng. „edge“*. Nakon toga ide *eng. Quad* ploha koja se sastoji od 4 točke i najomiljenija je u poligonalnom svijetu. Razlog tome su alati koji kasnije zaglađuju rubove i/ili od kvadrata rade krug. Dakle, u poligonalnom svijetu, krug se sastoji od mnogo točaka međusobno povezanim linijama. Što više točaka imate, glađi je rub. Modeliranje krivuljama isti taj krug drugačije interpretira. Modeliranje krivuljama koristi matematički opis kružnice, površine su matematički

opisane i nema nazubljenosti kao u poligonalnom svijetu. Krivulje se mogu prebaciti u poligone ali ne i obrnuto.

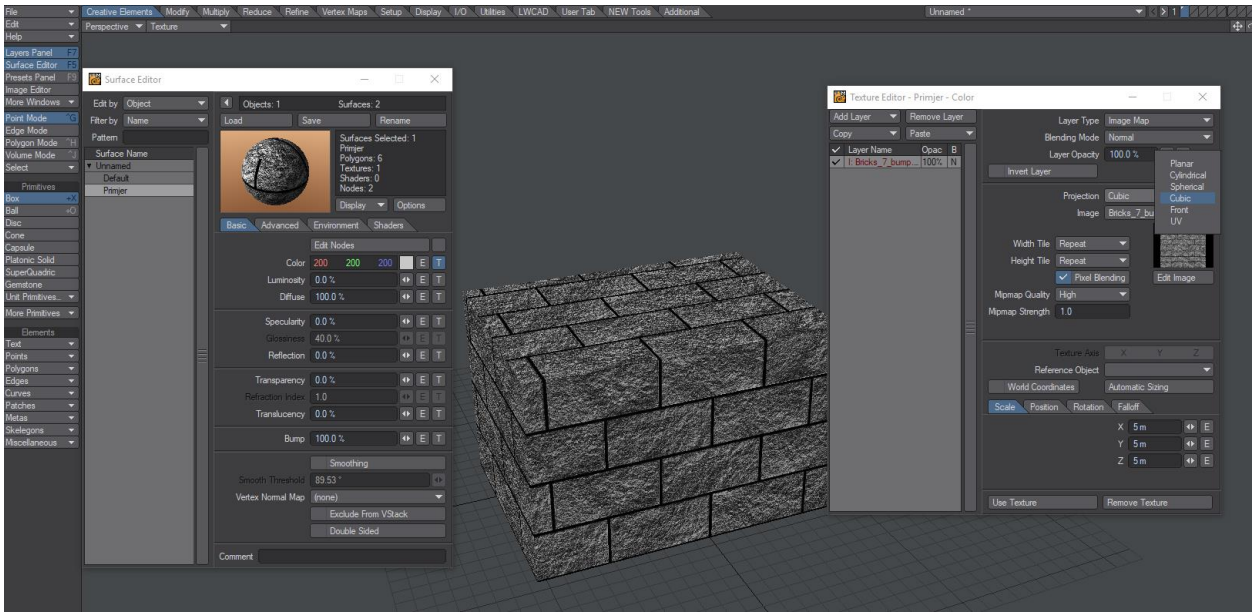


Slika 2.1.2.1 Razlika između modeliranja krivuljama (lijevo) i poligonalnog modeliranja (desno)

2.1.3 Teksturiranje

Kako bi vaš model dobio obilježja stvarnog objekta, potrebno ga je texturirati ili u koliko textura nije potrebna, definirati nekim od materijala (staklo, metal i sl.). Teksture su slike koje se primjenjuju na model. Postoji više načina na koji se to postiže:

- Planarno, Cilindrično, Sferično, Kubno, Pročelje, UV

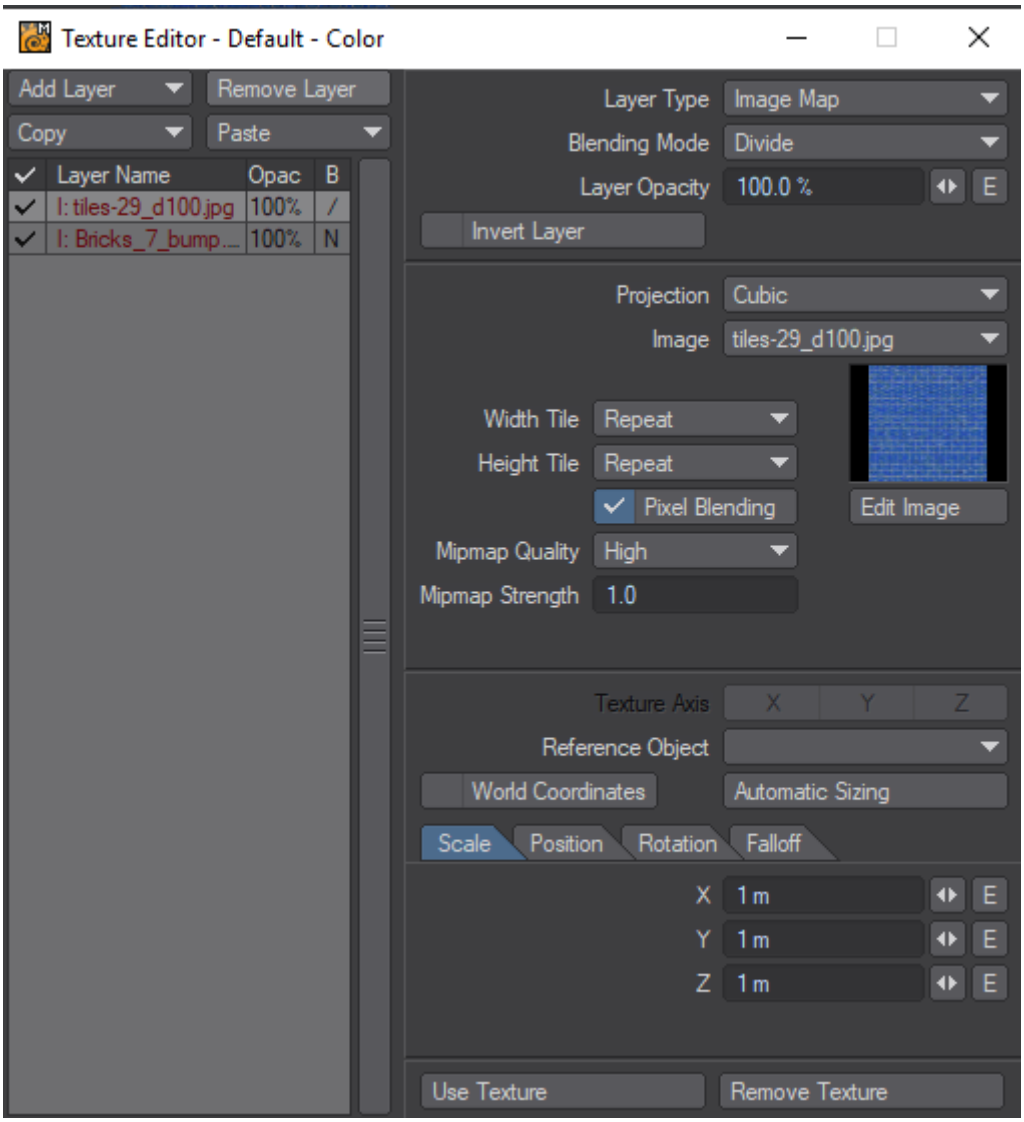


Slika 2.1.3.1 Prikaz kubne projekcije teksture na model.

Tekstura je najčešće referentna fotografija (npr. drvo) koja može biti zapisana u raznim formatima, ovisno o potrebi. Neki od formata zapisa su .jpeg, .png .tiff i sl. Tekstura također može u nekim slučajevima biti video formata, na taj način možemo na poligon staviti video zapis slijevanja vode na prozoru, no to je samo jedna od metoda. Imamo dvije vrste teksturiranja:

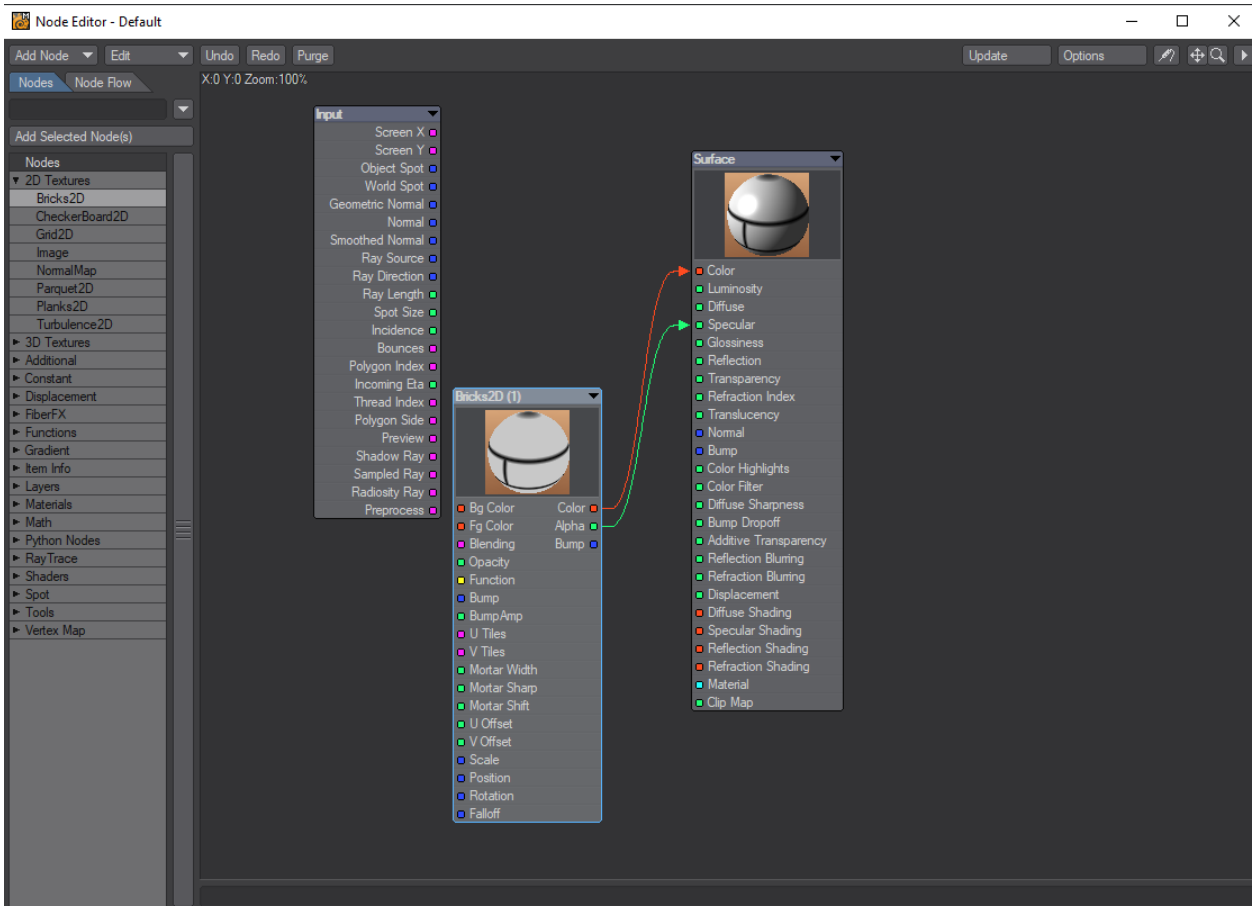
- Slojevito (*eng. Layer*)
- Teksutriranje čvorevima (*eng. Node*)

Slojevito teksturiranje je starija verzija teksturiranja i ima svoje limite zbog kojih je nastalo teksturiranje čvorevima. Slojevito teksturiranje radi na principu slojeva, gdje se najgornja tekstura vidi kao prva i ima interakciju (*eng. Blending*) s prvom ispod nje, neke od interakcija su npr: Additive, Subtractive, Divide, Alpha. Na taj način možemo mješati texture kako bi postigli željene efekte.



Slika 2.1.3.2 Primjer slojevitog teksturiranja s Blending mode: Divide

Teksturiranje čvorovima nam nudi puno više mogućnosti, više nismo ovisni o položaju teksture u slojevitom modelu nego možemo spojiti koju god hoćemo teksturu s kojom god drugom po želji. Također je puno više kanala omogućeno, matematičkih funkcija i sl.



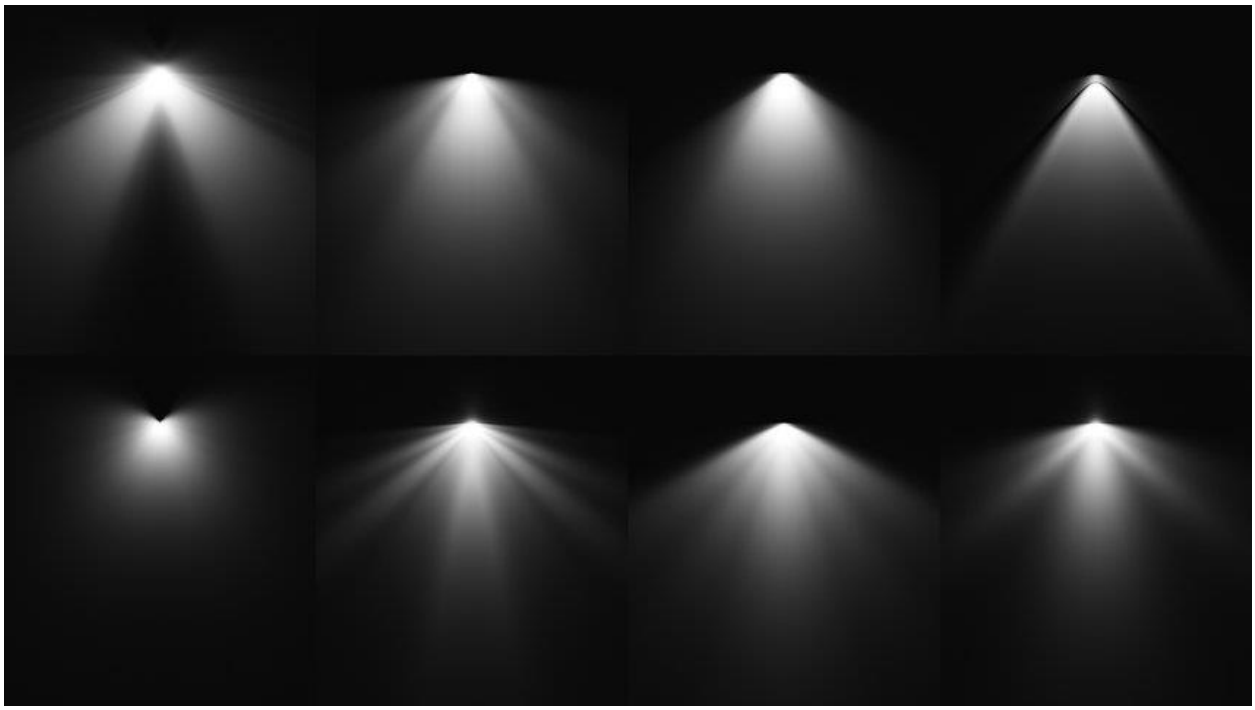
Slika 2.1.3.3. Primjer teksturiranja čvorovima. Surface čvor je završni čvor koji će se vidjeti na modelu.

Teksturiranje ima još jednu stavku a to su vrste teksturnih mapa. Na model se mogu staviti različite teksture na različite kanale. Tako za prikaz npr. lista hrasta koristimo kanal boje. No da bi list bio izrezan poput lista a ne kvadratičan prikaz slike, koristimo kanal prozirnosti i u njega učitavamo siluetu tog lista u crno bijeloj boji. Dalje, list nam je sada izrezan ali djeluje ravno, ne realno. Tu utječemo dodavajući teksturu na kanal koji upravlja s razinama „izbočenosti“ i dobijemo dojam da su žilice na listu izbočene. Nadalje možemo dodavati teksture za refleksiju, dodatnu prozirnost i slično.

2.1.4 Osvjetljenje

Osvjetljenje daje dojam vizualizaciji kako ne bi djelovala „flat“. Igrom svjetla i sjene postižemo dinamičnost i kontrast vizualizacije koji je bitan čimbenik u ostvarivanju osjećaja prostora i dubine. Glavno i najbitnije osvjetljenje je „Global illumination“ koji se konfigurira na

samom početku rada sa svjetlima. Tu se postavlja trenutno doba „dana“, mogu se postaviti i opcije položaja sunca, intenziteta, vremena (zimsko, ljetno), boje (kelvin) i sl. Nakon toga se dodaju ostala svjetla: Sferično, Prostorno, Udaljeno i sl. To su generalna svjetla koja možemo naći u našem programu u kojem radimo. No za pravi doživljaj svjetla koristi se fotometrijsko svjetlo. Fotometrijsko svjetlo je virtualna replika pravog svjetla. Za određenu lampu se može naći fotometrijski zapis u obliku svjetlo.ies. To svjetlo će identično reproducirati osvjetljenje kao što prava fizička lampa osvjetljava prostor. Na njemu možemo mijenjati boju i intenzitet.



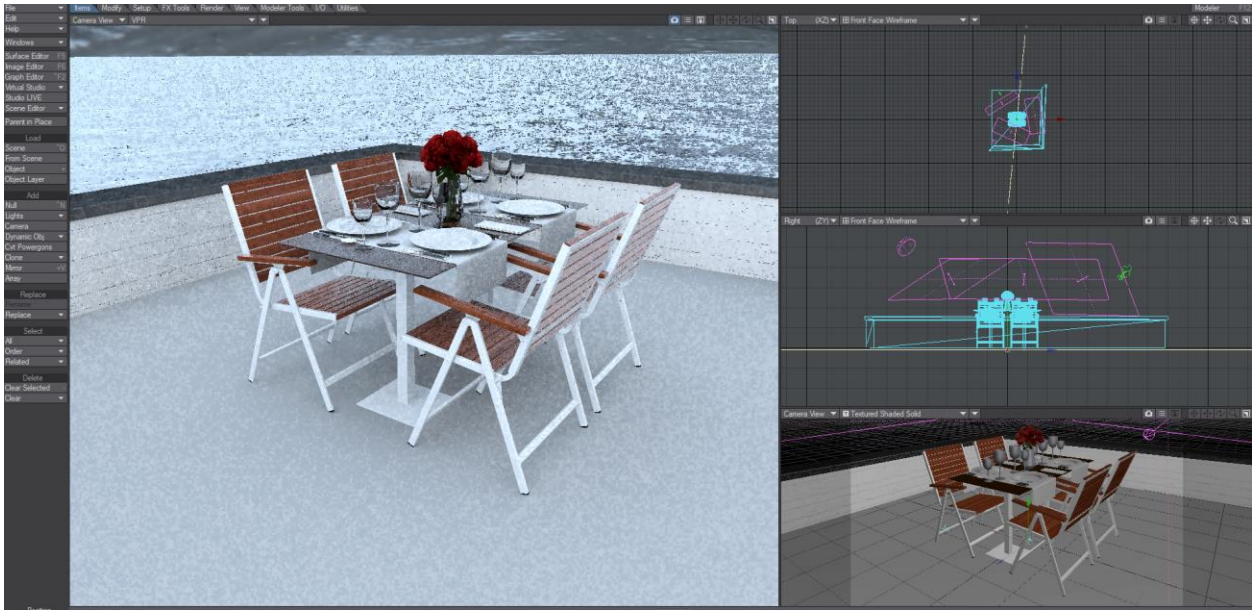
Slika 2.1.4.1. Primjer fotometričnih svjetala.

Svako svjetlo ima još jedan bitan element, sjenu. Na svjetlima a i na samom globalnom osvjetljenju možemo utjecati na sjene. Sjene mogu biti oštre, meke, raspršene, svjetlije, tamnije, sve ovisi o našoj potrebi. Standardno će nam udaljeno svjetlo dati oštre sjene, ono je velikog intenziteta i na velikoj udaljenosti, slično je suncu. Prostorno svjetlo će nam dati raspršene sjene, ono može biti neke veličine n x m (metara) i predstavlja globalno osvjetljenje ili ulazak svjetla kroz prozor. Koristeći više različitih svjetala postižemo uvjerljiviju vizualizaciju.

Svjetla funkcioniraju kao matematički opis pravog svjetla, fotona. Možemo postaviti broj čestica koje će se emitirati, odbijati i na kraju stvoriti sliku. Kamera se ponaša kao naše oko koje prikuplja fotone. Sa svjetlima i količinom emitirajućih čestica treba biti oprezan jer su proračuni jako zahtjevni, pogotovo ako se broj odbijanja stavi sa recimo 2 na 10. Što je više čestica i odbijanja to je bolji rezultat, no u nekim slučajevima s 100 čestica i 3 odbijanja dobijemo isti rezultat kao i s 1000 čestica i 10 odbijanja.

2.1.5. Postavljanje scene

Kod 3D vizualizacije, scena je element koji sadrži sve modele, animacije, svjetla i teksture. Dio koji se u sceni najviše postavlja su kutovi kamere, instanciranje objekata poput trave i drveća, teksture okoliša i slično. Scena je predzadnji korak do vizualizacije i već se u njoj može pomoću VPR-a pregledati i uštimati scena do zadnjeg detalja. U ovom koraku vidimo kako će izgledati vizualizacija, svjetla, teksture i slično. VPR (Viewport Preview Renderer) je nativni interaktivni renderer.

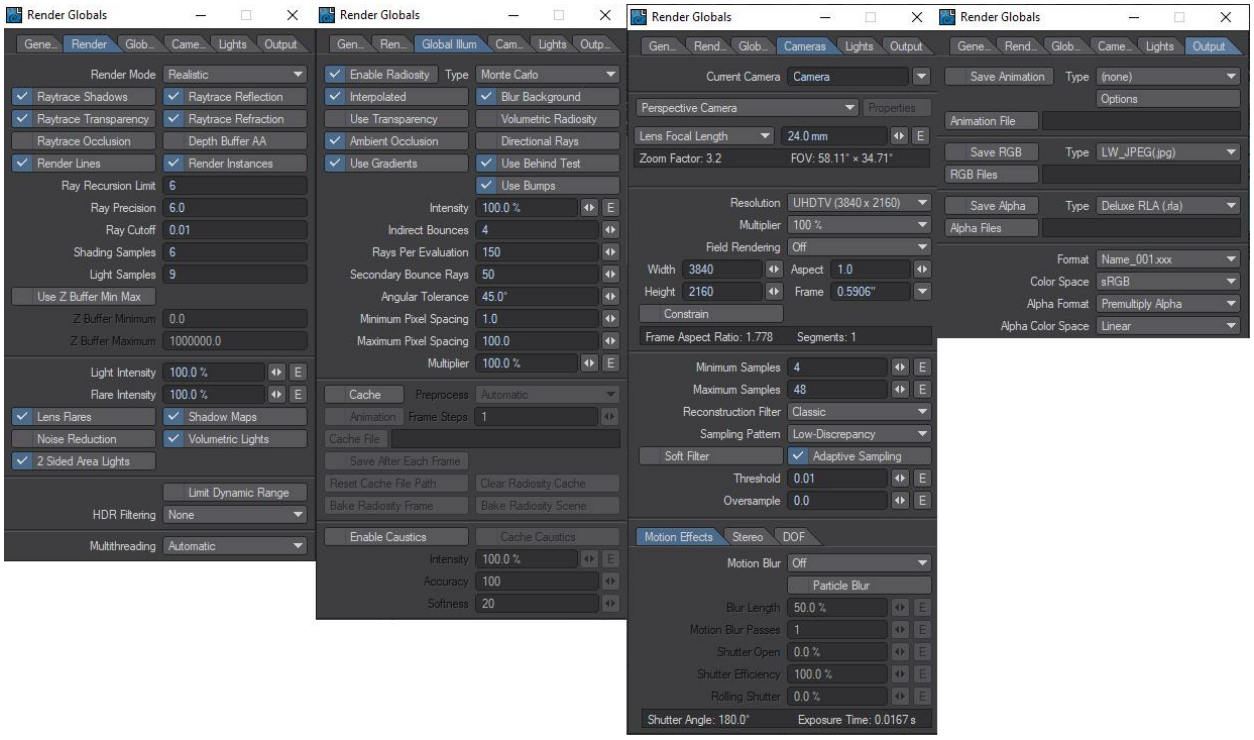


Slika 2.1.5.1 Prikaz scene pomoću VPR-a.

2.1.6 Render

Nakon što je postavljena scena i svi elementi te smo sigurni da želimo započeti proces generiranja završne vizualizacije dolazimo do postavki samog renderiranja. U procesu renderiranja nas generalno zanimaju postavke veličine vizualizacije (klasično UHD). Što veća rezolucija, bolja kvaliteta no puno sporiji render. Na manjoj rezoluciji gledamo da bude antialiasing veći (uklanjanje nazubljenosti na rubovima). Rendering ima puno dodatnih opcija koje utječu na završnu vizualizaciju, no s obzirom da su kompleksne nećemo ih se doticati u

ovom radu. Kroz dolje opisanu realizaciju na pravom primjeru ću staviti neke od opcija koje utječu na rezultat vizualizacije.



Slika 2.1.6.1. Primjer nekih od opcija prilikom postavljanja završnog rendera.

2.2. Korištene tehnologije

Tehnologije korištene u ovom radu se provode kroz programe Lightwave, Photoshop i Insta 360 Sticher. Kombinacija ovih programa nam omogućuje stvaranje realistične vizualizacije na način da pomoću Photoshopa možemo stvoriti, mijenjati i uređivati teksture, napraviti post-processing na završnoj vizualizaciji. Pomoću Inste 360 stichera stvaramo equirectangular sliku koja predstavlja okoliš (enviroment map) oko same zgrade. U Lightwaveu modeliramo, texturiramo, dodajemo svjetla, pravimo scenu i na kraju izvozimo vizualizaciju koju ćemo potom u Photoshopu dodatno odraditi.

2.2.1. Lightwave

NewTek LightWave 3D cjelovito je riješenje za modeliranje, renderiranje i animiranje. Lightwave se uvelike koristi u televizijskoj i filmskoj produkciji, razvoju videoigara, grafike za tisak i dizajnu. LightWave je dizajniran za korištenje jednog umjetnika, malog tima ili velikog

projekta. LightWave 3D kombinira vrhunski render s moćnim, intuitivnim alatima za modeliranje i animaciju.

LightWave se neprimjetno uklapa u velike multi-softverske cjevovode - sa svojim moćnim alatima za razmjenu koji uključuju FBX, ZBrush GoZ, Collada, Unity Game Engine Support i Autodesk Geometry Cache. Za razliku od ostalih programskih paketa, LightWave nudi umjetnicima i studijima cjelovito cjelovito rješenje odmah izvan okvira.

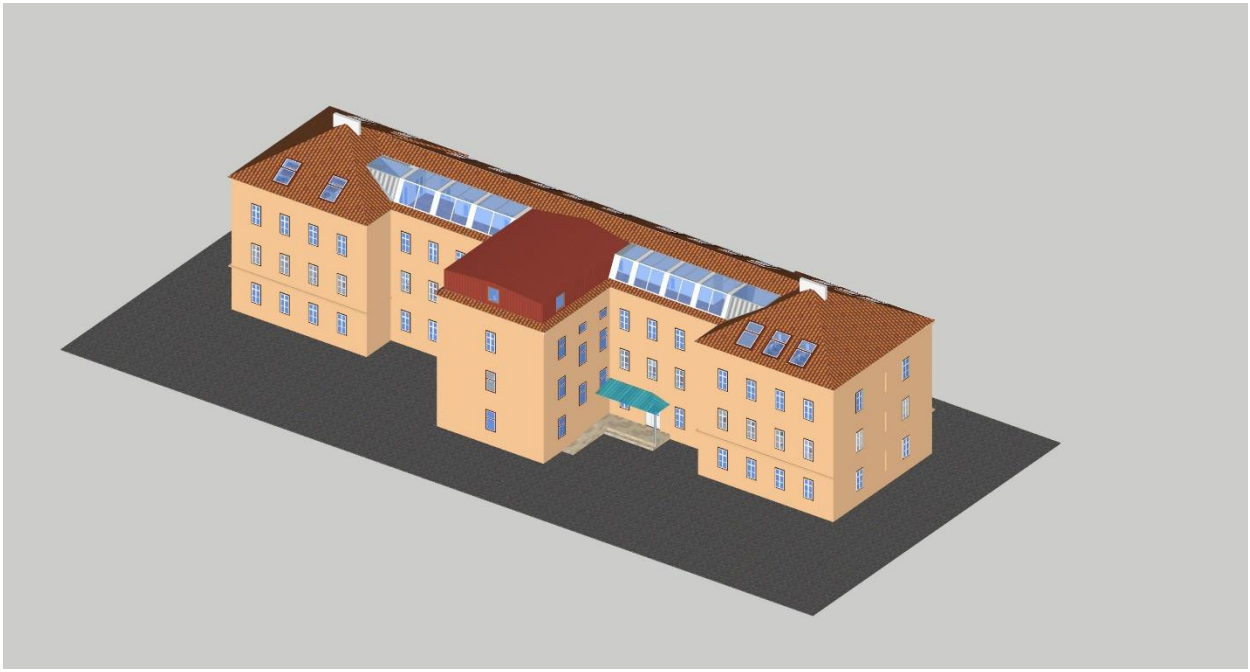
- Robusno poligonalno i podzemno modeliranje površine.
- Sjajni alati za teksturu PBR, s moćnim namjenskim sjenilima.
- Snažni alati za animaciju i opremu.
- Volumetrijski i dinamički sustavi s efektnim efektima.
- Ugrađen je dokazano nagrađivani motor Global Illumination od renomirane proizvodnje.

2.2.2. Photoshop

Adobe Photoshop je rasterski grafički urednik koji je razvio i objavio Adobe Inc. za Windows i macOS. Izvorno su je stvorili 1988. Thomas i John Knoll. Od tada je ovaj softver postao industrijski standard ne samo u uređivanju rasterske grafike, već i u digitalnoj umjetnosti u cjelini. Naziv softvera tako je postao generički zaštitni znak, što dovodi do njegove upotrebe kao glagola (npr. "Za fotografiranje slike", "fotošopiranje" i "natjecanje u fotošopu", iako Adobe obeshrabruje takvu upotrebu. Photoshop može uređivati i sastavljati rasterske slike u više slojeva i podržava maske, alfa kompoziciju i nekoliko modela u boji, uključujući RGB, CMYK, CIELAB, spot boju i duotone. Photoshop koristi svoje vlastite formate datoteka PSD i PSB za podršku ovih značajki. Uz rastersku grafiku, ovaj softver ima ograničene mogućnosti uređivanja ili prikazivanja teksta i vektorske grafike (posebno kroz isječak staza za potonju), kao i 3D grafiku i video. Skup značajki može se proširiti dodacima; programi razvijeni i distribuirani neovisno o Photoshopu koji se pokreću unutar njega i nude nove ili poboljšane značajke.

3. Realizacija

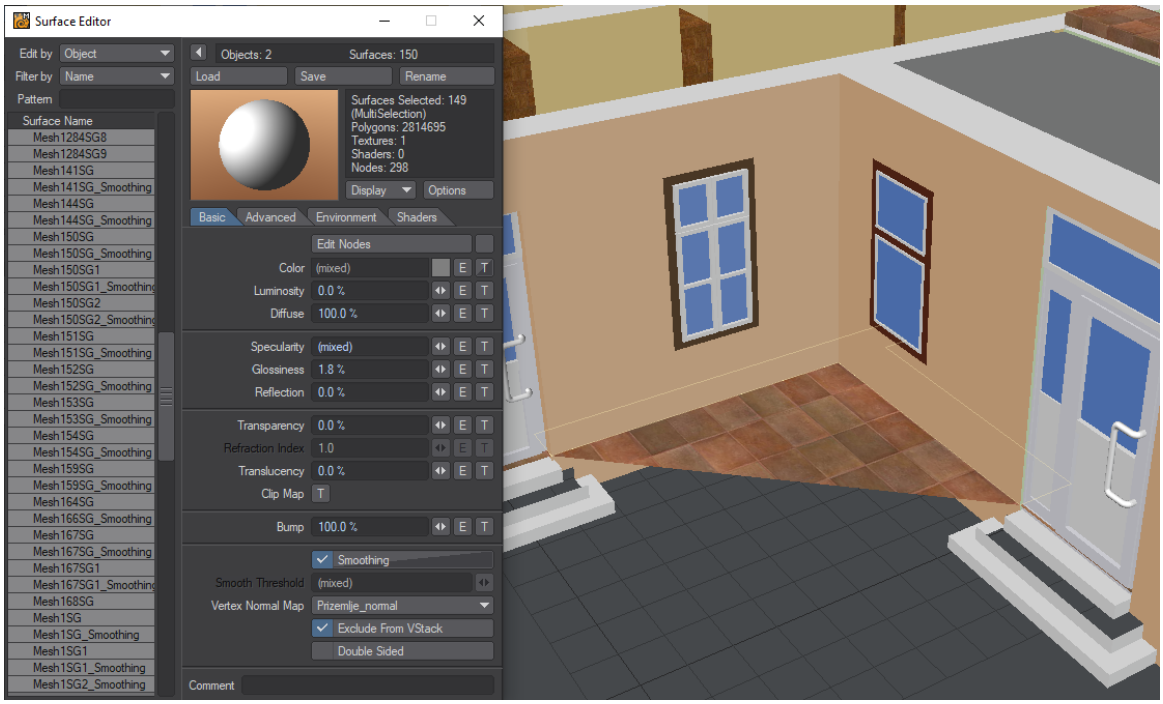
Izrada vizualizacije zgrade FERIT-a se bazira na već odrađenom jednostavnom modelu zgrade u programu SketchUP od Google-a. Program SketchUP je napravljen za brzinske 3D skečeve kako bi mogli prikazati 3D modele bez puno vremena ulaganja u modeliranje i gdje nam nije bitna realističnost nego općenito model. Prva stvar koju moramo napraviti je izbaciti zgradu kao objekt iz programa SketchUP u .fbx formatu kako bi zadržali teksture (ako ih ima i ako su uporabljive). Nakon toga, uvozimo model u Lightwave i započinjemo proces reparacije modela, modeliranja dodatnih elemenata, teksturiranja, postavljanja svjetala, postavljanja scene, renderiranja i na kraju post-procesinga kako bi dobili završnu vizualizaciju.



Slika 3.1. Izgled referentnog modela u programu SketchUP.

3.1. Prerada referentnog modela

Korištenje referentnog modela rađenog u drugom programu ima svoje pozitivne i negativne strane. Pozitivno je što ne moramo koristiti nacrte, samim time manje vremena trošimo na izradu realizacije. Negativan dio je stvaranje velike količine nepotrebnih poligona. Raspad modela, često se znaju poligoni krivo triangulirati (triangulacija je pretvaranje svih poligona s više od 3 točke u poligone s 3 točke). Također, prilikom izvoza programi često znaju dati generička imena površinama kao što se vidi u primjeru ispod: Mesh1284SG8 a trebalo bi biti recimo: Prozor.

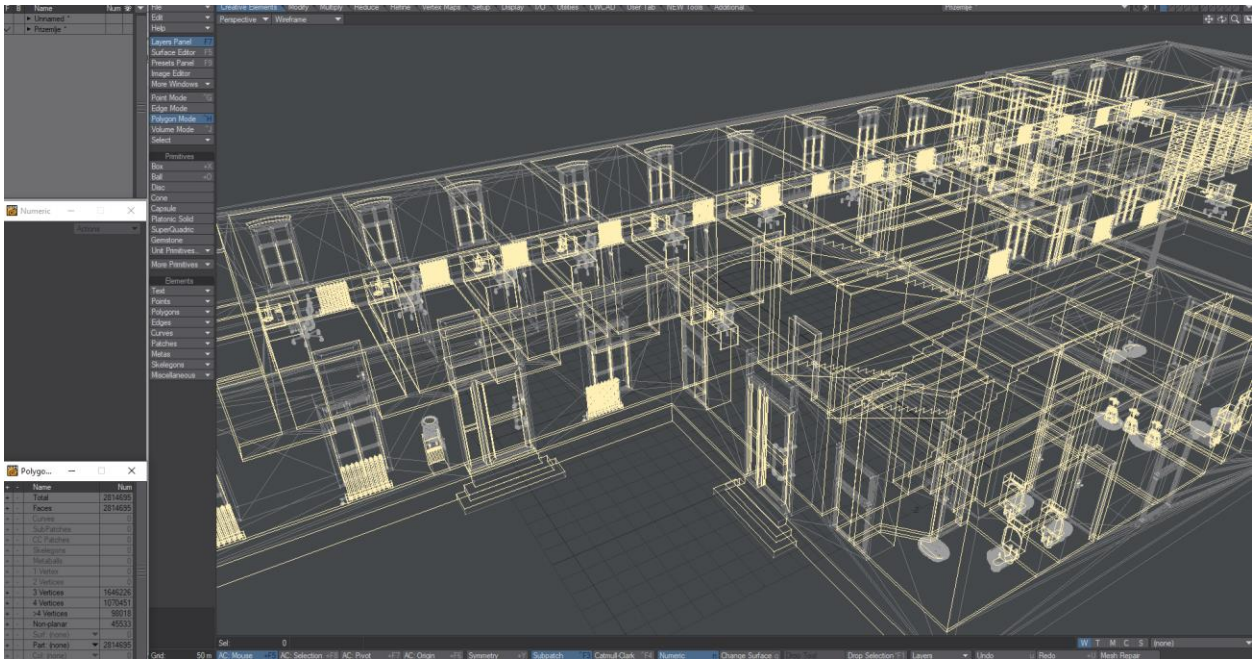


3.1.1. Primjeri pogrešaka prilkom izvoza modela.

U trenutnom izvozu je prikazan samo sloj „Prizemlje“ i već sadrži više od 2 miliona poligona. Stolice i stolovi su „zaljepljeni“ u modelu, na taj način je nemoguće raditi zbog prevelike potrošnje memorije i procesorske snage. Za sve što radimo program će svaki puta očitavati 2 miliona poligona. Slijedeći korak je optimizacija svakog sloja:

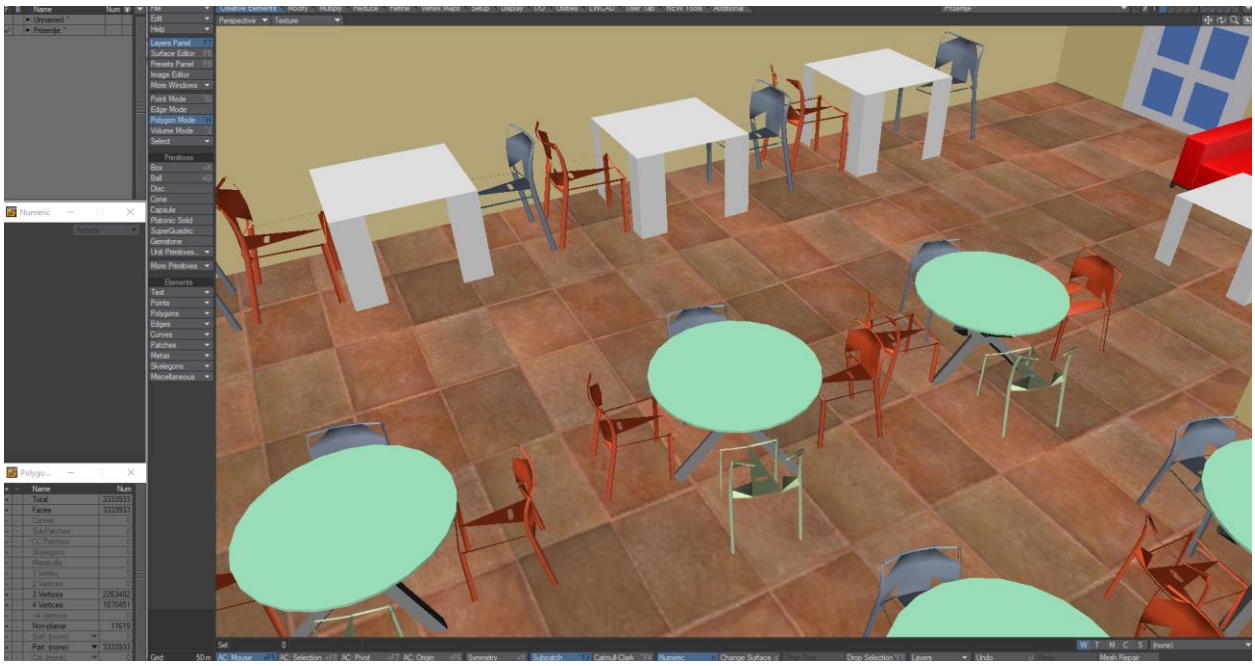
1. Podrum
2. Prizemlje
3. Prvi kat
4. Drugi kat
5. Potkrovlje
6. Krov
7. Okoliš

Optimizacija se sastoji od uklanjanja svih stolica i stolova, ostavljamo samo po jednu instancu istoga. Pripremamo poligone za kasnije instanciranje stolova i stolica prilikom renderiranja. Ovo nam uvelike olakšava sami rad na modelu. Zatim eliminiramo višak poligona, ponovno modeliramo određene elemente za koje je vremenski zahtjevnija optimizacija. Svim površinama moramo dati smislena imena kako bi kasnije znali dodijeliti teksture smisleno. Ako ostavimo površine s ovim generičkim nazivima, nećemo znati kasnije što je staklo a što drvo.



Slika 3.1.2. Prikaz poligona s više od 3 točke.

U slici 3.1.2. se može vidjeti selekcija svih poligona s više od 3 točke. Koristiti ćemo alat koji se zove „Triangulate“ kako bi sve te poligone sveli na 3 točke. Na taj način izbjegavamo probleme koji znaju nastati prilikom rendera a vide se na slici 3.1.1. gdje je pod izašao izvan zgrade.



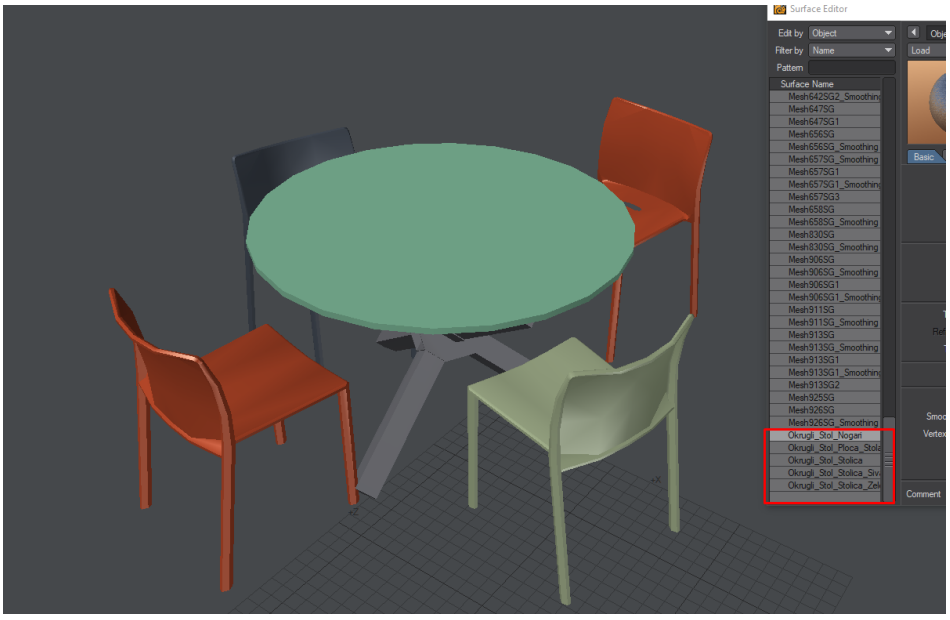
Slika 3.1.3. Prikaz problematike izvrnutih poligona. Poligoni koji fale su tu, ali su okrenuti u krivom smijeru.

Slika 3.1.3. Prikazuje učionicu koja ima više stolova i stolica. Stolovi i stolice su problematične zbog izvrnutih poligona. Ovdje nam dobro dođe instanciranje zbog toga što ćemo

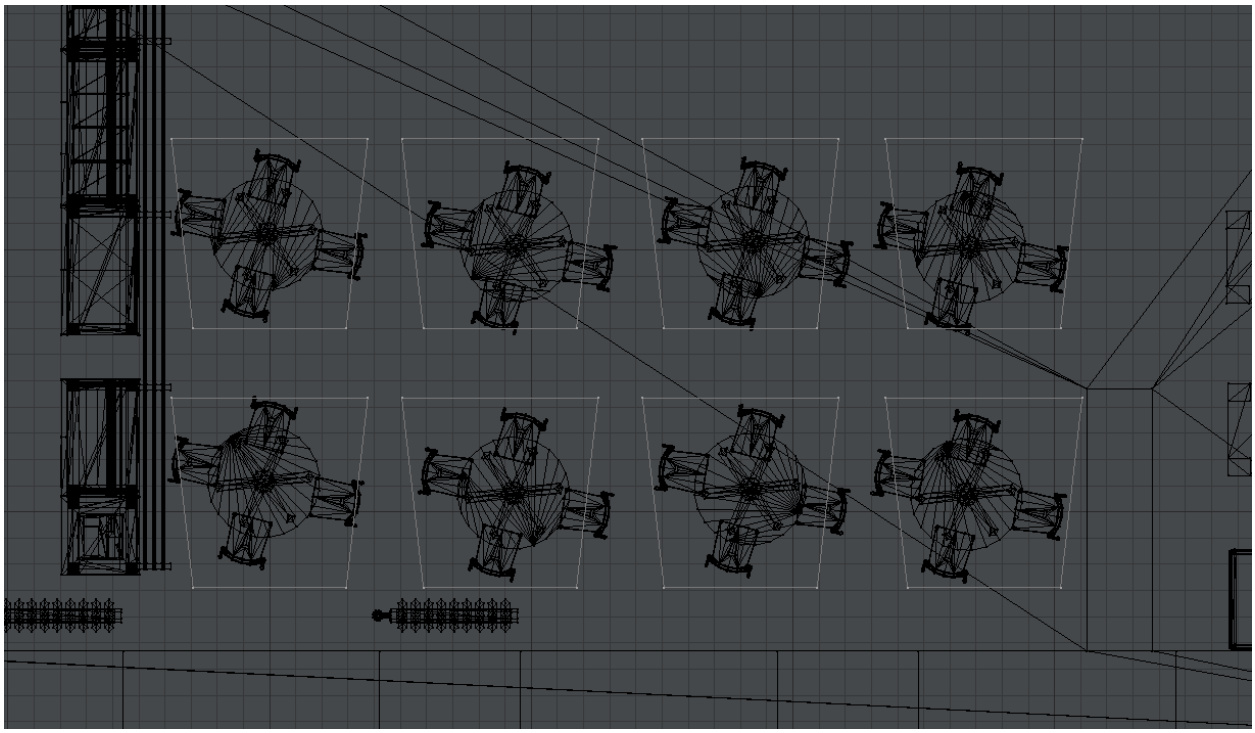
uzeti samo jedan stol i stolicu, njih ćemo popraviti, okrenuti poligone, izmodlirati ako što fali, dodjeliti materijal i na kraju snimiti kao poseban objekt. Kada se render bude izvršavao, program će sam populirati učionicu na točkama gdje odredimo da stoje određene stolice. Možemo točno reći koji stol i koja stolica da populiraju prostoriju i koliko komada, kako će biti okrenuti i koje veličine. Kako budemo snimali modele, trebamo prvo postaviti LightWave standardni direktorij koji izgleda ovako:

- Naziv Projekta (3D Vizualizacija FERIT)
    - backup
    - cache
    - images
    - lights
    - objects
    - renders
    - scenes

Mape su konstruirane tako da svaka ima svoje datoteke. U ovom slučaju redom mapa backup sadrži backupe projekta, cache sadrži predmemoriju koju možemo stvoriti tokom renderiranja, snimimo predmemoriju cijele scene npr. proračuna svjetla kako bi slijedeći render iz drugog kuta samo učitao već izrenderiranu svjetlost. Ovakav princip nam omogućava veliku uštedu vremena jer cache se često računa satima. Images mapa sadrži teksture, light sadrži fotometrična svjetla, objects sadrži modele, renders sadrži rednere i scenes sadrži scene.



Slika 3.1.4. Prikaz uređenog stola i stolice, izdvojenih u poseban sloj i uredno imenovanih površina.



Slika 3.1.5. Prikaz poligona (bijeli kvadrati) koji će kasnije biti korišteni za instanciranje stolova.

1. ZAKLJUČAK

7. LITERATURA

Knjige:

[1] J. Lockhart, Modern PHP: New Features and Good Practices, O'Reilly Media Inc., Sebastopol, 2015.

[2] L. Welling, L. Thomson, PHP and MySQL Web Development, Sams Publishing, Indiannapolis, 2005.

[3] M. Stauffer, Laravel: Up and Running: A Framework for Building Modern PHP Apps, O'Reilly Media Inc., Sebastopol, 2015.

[4] A. Ravulavaru, Learning Ionic - Build Hybrid Mobile Applications with HTML5, Pact Publishing, Birmigham, 2015.

[5] D. Rees, Laravel: Code Bright, Leanpub, 2014.

[6] K. Shotts, Mastering PhoneGap Mobile Application Development, PACKT Publishing, Birmigham, 2016

Članci:

[1] I. Malavolta, S. Ruberto, T. Soru, V. Terragni, Hybrid Mobile Apps in the Google Play Store: An Exploratory Investigation, Mobile Software Engineering and Systems (MOBILESoft), 2015 2nd ACM International Conference, 16. - 17. svibanj 2015.

[2] S. Aghaee, C. Pautasso, Mashup development with HTML5, Mashups '09/'10 3rd and 4th International Workshop on Web APIs and Services Mashups, Ayia Napa, Cipar — 01. prosinac 2010.

[3] N. Serrano, J. Hernantes, G. Gallardo, Mobile Web Apps, IEEE Software, vol. 30., 03. rujan 2013.

[4] Kako programirati za Android ili iOS, Bug, broj 252, studeni 2013.

Internetski izvori:

[1] MVC Xerox Parc – Internet http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html (01.06.2016)

[2] Models-Views-Controller, Trygve Reenskaug http://heim.ifi.uio.no/~trygver/1979/mvc-2/1979-12-MVC.pdf (01.06.2016)

[3] MVC Originals http://heim.ifi.uio.no/~trygver/2007/MVC_Originals.pdf (01.06.2016)

[4] JavaScript – JSP Pages http://www.kirkdorffer.com/jspspecs/jsp092.html (01.06.2016)

[5] Structure of Laravel Applications http://laravelbook.com/laravel-architecture/ (01.06.2016.)

[6] Structure od Ionic App http://mcgivery.com/structure-of-an-ionic-app/ (01.06.2016.)

[7] Controllers in Ionic Framework http://mcgivery.com/controllers-ionicangular/ (01.06.2016.)

[8] Ionic preface http://ionicframework.com/docs/guide/preface.html (01.06.2016.)

SAŽETAK

Ključne riječi

U svojim počecima, Internet i mobilne aplikacije su služile samo za prezentaciju informacija. Razvoj tehnologije i programskih jezika je došao do točke da se u relativno kratkom vremenskom roku može napraviti aplikacija koja se oslanja na udaljeni servis te radi na mobitelima i tabletima. Tako da oglasnik ponude i potražnje, umjesto samo u stvarnom svijetu, možemo imati i na internetu. Postoji nekoliko različitih načina izvedbe. U svojoj srži, aplikacija koja se nalazi na mobitelu i pokreće se je tzv. klijent dio, a mrežna stranica je poslužiteljska strana sustava. Cijeli sustav je koncipiran tako da postoji jednostrana komunikacija, i klijent šalje zahtjev za podacima na poslužiteljski dio. Poslužitelj vraća oblikovane podatke, i oni se prikazuju unutar aplikacije. Obje strane se sastoje od sličnog principa – imamo središnju funkcionalnost koja je kontroler, i koja dolazni zahtjev usmjerava prema dohvatu, obradi i slanju ili prikazu podataka. Princip rada se zove MVC obrazac. Poslužiteljska strana šalje podatke, a u klijentskom dijelu se ti podaci samo prikazuju. Poslužiteljska strana još ima i dodanu mogućnost da može i prikazati podatke. Konačni proizvod je sinergija klijentske i poslužiteljske aplikacije, koje koristeći moderne tehnologije i načine prezentacije i obrade podataka, čine jedan cjelovit proizvod.

Ključne riječi: oglasnik, hibridna aplikacija, MVC obrazac, poslužitelj, klijent

ABSTRACT

Keywords

Developing classified application for supply and demand of products

In their early years, Internet and mobile applications were capable only for presenting various data. During years, development of technology, and programming languages offered us to develop application in short time-span. In other words, we can develop not only classifieds in papers, we can have them in digital world too. There are several ways of accomplishing that task. In its core, there is mobile application, which is client side, and there is website which represents server-side of whole system. Idea behind this application is that there is one-way communication, meaning that client sends requests to the server. Server responds with formatted data, and they are shown inside application. Both sides consists of similar arhitecture – there is main functionality which is controller for collecting, and transfering requests. Controller communicates to database, and collects, edits, and shows data. This arhitecture is called MVC. Server application can send and show data, but client side can only show data. Final product is synergy between client and server

side. By using modern technology and modern ways of presenting and processing data, they are one, whole product.

Keywords: classified, hybrid application, MVC paradigm, server, client

ŽIVOTOPIS

PRILOZI

Popis slika:

Popis korištenih oznaka i kratica

engl.	Engleski
DBMS	Sustav za upravljanje bazama podataka (engl. Database Management System)
OOP	Objektno orijentirano programiranje (engl. Object Oriented Programming)
PHP	U početku Personal Home Page, kasnije PHP: Hypertext Preprocessor
MVC	engl. Model-View-Controller
PDO	PHP objekti podataka (engl. PHP Data Objects)
AJAX	Asinhroni JavaScript i XML (engl. Asynchronous JavaScript and XML)
SQL	Strukturirani upitni jezik (engl. Structured Query Langauge)
CRUD	Funkcionalnosti izrade, čitanja, ažuriranja i brisanja
HTTP	engl. HyperText Transfer Protocol
HTML	Prezentacijski jezik za izradu web stranica (engl. HyperText Markup Language)
CSS	Kaskadne stranice stilova (engl. Cascading Style Sheets)
XML	Jezik za proširivo označavanje podataka (engl. Extensible Markup Language)
URL	jedinstveni lokator resursa (engl. Uniform Resource Locator)
GUI	engl. Graphical User Interface
API	engl. Application Programming Interface