Usporedba "varalice" i simulacije u Blenderu

Knežević, Mihajlo

SVEUČILIŠTE U ZAGREBU
GRAFIČKI FAKULTET

ZAVRŠNI RAD

Mihajlo Knežević

Sveučilište u Zagrebu
Grafički fakultet

Smjer: Dizajn grafičkih proizvoda

ZAVRŠNI RAD

USPOREDBA "VARALICE" I SIMULACIJE U BLENDERU

Mentor:
Izv.prof.dr.sc. Sanja Bjelovučić Kopilović

Student:
Mihajlo Knežević

Zagreb, 2014

Zahvaljujem se svojoj mentorici izv.prof.dr.sc. Sanji Bjelovučić Kopilović na stručnim savjetima i pomoći pri pisanju ovoga rada

SAŽETAK

Kako bi kreirali računalnu 3D grafiku i animaciju u zadanim rokovima potrebno je raspolagati s, relativno, velikim vremenskim i računalnim resursima. Dvije bitne stavke o kojima je već u početku potrebno voditi brigu je omjer brzine renderiranja animacije i kvalitete vizualne prezentacije. Ova dva aspekta danas često znaju biti u obrnuto proporcionalnom odnosu – što, u biti, znači ako želimo brzo renderiranje proizvoda vizualna prezentacija će biti lošija, u suprotnom slučaju ako želimo da nam proizvod bude visoke vizualne kvalitete to će vrijeme renderiranja biti sporije.

Zbog upravo ovog odnosa kvalitete i brzine, možemo se poslužiti i nekim drugačijim pristupima i načinima izrade ovakvih proizvoda. Kako je izvršavanje računalnih simulacija često vrlo zahtjevno po računalne i vremenske resurse možemo se koristiti tehnikom "varalice" koja nam kroz drugačiji pristup stvaranja 3D objekata i animacije omogućava uštedu spomenutih resursa.

U ovom radu obrađena je tema animiranja mehaničkih fenomena pomoću računala putem mehaničkog simulacijskog sustava i tehnike "varalice". Testovima i analizom obrađeni su i uspoređeni različiti načini izrade animacije kroz primjere animiranja tkanine i vode. Glavni fokus je bio na animaciji tkanine, a animiranje vode je dodano kako bi se usporedile drugačije tehnike "varalica". Dokazano je da je u većini slučajeva “varalica” bolje rješenje ako želimo uštedjeti na vremenu renderiranja scene.

KLJUČNE RIJEČI: Blender, računalna animacija, 3D grafika, mehanička računalna simulacija, izrada računalnih vizualnih efekata

SADRŽAJ

1. UVOD

Današnja 3D računalna grafika i animacija postale su dio svakodnevice u svim digitalnim medijima za prikaz multimedijalnog sadržaja. Televizija, računala, mobiteli, igraće konzole, tableti – elektronski uređaji na kojima svakodnevno pregledavamo multimedijske sadržaje i s kojima vršimo interakciju. Industrija računalnih igara i filmova su dva najveća pokretača 3D grafike i danas su te industrije postale jedan od najutjecajnijih faktora na živote svih ljudi koji ih koriste. Ne treba puno niti spominjati koliku zaradu ostvaraju s velikim filmskim projektima ili *blockbuster* računalnim igrama na današnjem globlanom tržištu kojeg je stvorila informacijska tehnologija – televizija, računala, pametni telefoni. Podrška za prikaz 3D grafike na svim tim uređajima postala je standard i naučeni smo očekivati ništa manje. Tako da i nije toliko teško shvatiti potrebu za takvim atraktivnim sadržajima i njihovu važnost u današnjem svijetu.

Iako je današnja računalna snaga svih ovih uređaja, relativno, velika i svakim danom postaje još snažnija – još uvijek postoji taj problem izvršavanja prikaza animacije videa ili interaktive grafike u realnom vremenu.
Problem je i u tome da kako se razvija hardverska snaga računala tako dolazi do razvoja softvera, a time se proporcionalno podiže i kvaliteta prikaza slike. Takav razvoj će se vjerojatno događati tako dugo dok ne postignemo realistični prikaz ljudskog bića i prirode koja ga okružuje u virutalnom svijetu, a sada unazad dvije-tri godine mogli smo svjedočiti pojedinim primjerima koji su dostigli taj cilj.
Naravno, svakodnevno još uvijek koristimo hardver koji ne može podržati takvu realističnu grafiku s generiranjem animacije u realnom vremenu jer nam je to danas još uvijek financijski nepristupačno, a niti ne postoje komercijalni uređaji.
Predviđanje autora ovog rada je da će se ovakav ideal kvalitete grafičkog prikaza dostići kroz narednih pet do deset godina kao standardni prikaz na osobnim računalima.

Ovaj rad se bavi problemom prikaza kvalitete izvođenja 3D grafičkog sadržaja kroz usporedbu simuliranja i zavaravanja grafičkog prikazivanja. Drugim riječima, objašnjeni su načini prikazivanja istih (sličnih) efekata kroz simulaciju i putem "varalice".
Pokušala se razraditi problematika simulacije efekata na sceni i njen utjecaj na računalne resurse. Dan je pregled većine alata potrebnih za izradu simulacije da bi se dobila bolja slika koliko je kompleksna čitava izrada i procesiranje same simulacije. Na primjerima mehaničkog simuliranja vode i tkanine objašnjen je proces kreiranja scene za animaciju tekućine, kao i za tkaninu.

Kao praktični zadatak izrađene su dvije scene vijorenja zastave na vjetru i gibanja valova na moru gdje je u prvoj sceni korištena simulacija tkanine (pomoću sile vjetra) i simulacija mora odnosno oceana, a druga scena je prikazana pomoću "varalica" ili načina stvaranja efekta vijorenja i valova na moru bez da su korišteni simulatori.

Te dvije scene su testirane s obzirom na zauzeće procesorske jedinice i radne memorije, i mjereno je vrijeme renderiranja do konačnog produkta. Dokazana je tvrdnja da se pomoću varalice mogu ostvariti zadovoljavajući rezultati prikazivanja vjerodostojnosti scene i time se uštedjelo na procesorskom vremenu odnosno računalnim resursima.

Slika 1. Usporedba stvarne fotografije i scene iz sustava za izradu igara *CryEngine 2*

Izvor: http://www.g-truc.net/post/0148-4-normal.jpg

Važna napomena: danas se, možda zbog vrtoglavog razvoja tehnologije, premalo pažnje posvećuje terminologiji i multidisciplinarnoj suradnji. U slučaju fizikalnih (konkretno mehaničkih) simulacija znanstvenici koji se bave mehanikom služe se jednom terminologijom, a računalni znanstvenici drugom, ne u svim slučajevima, ali evo primjera: u klasičnoj, inženjerskoj mehanici proučavaju se zasebno deformabilna i nedeformabilna tijela. U računalnim simulacijama postoje termini "*Rigid Body*" i "*Soft Body*" ali oba termina odnose se na deformabilna tijela, samo se radi o nekim razlikama u stupnjevima i vrstama deformabilnosti. U mehanici se "*Rigid Body*" prevodi kao kruto tijelo, a termin "*Soft Body*" koji računalni stručnjaci prevode kao "meko tijelo", u mehanici uopće ne postoji. Kruto tijelo je po mehaničkoj definiciji (i "*Rigid Body*" na engl. također) ono koje se ni na koji način ne može deformirati, dakle ni razbiti. U simulacijama se može, a isto ga zovu "*Rigid Body*". Ova napomena je dodana kao ograda za slučaj eventualnih primjedbi od strane stručnjaka iz područja mehanike, zbog nekih navoda i prijevoda u teorijskom dijelu.

2. RAČUNALNI SUSTAV ZA SIMULACIJU MEHANIKE

U ovom dijelu će biti predstavljen sustav namjenjen simulaciji zakona mehanike uglavnom u računalnoj animaciji filmova i igara.
To je računalni sustav koji simulira mehaničke pojave stvarnog svijeta u domeni računalne grafike, filma i igara. Izraz računalni sustav za simulaciju mehanike se više koristi za softvere koji su namjenjeni simulaciji mehaničkih zakona na preciznijoj, znanstvenijoj razini.
Po tome ih možemo podijeliti u dvije klase:

- izvođenje u realnom vremenu i
- visoko precizni sustav

Sustavi za izvođenje u realnom vremenu se koriste u računalnim igrama i drugim sličnim oblicima računalne interakcije. Kod njih se koriste pojednostavljeni proračuni i manja preciznost simuliacije, a sve to kako bi se zadovoljila potreba za brzim izvođenjem scena odnosno *frame rate*-a (slika po sekundi) da bi igra bila igriva.
Visoko precizni sustavi nalaze primjenu u softverima za precizno simuliranje, recimo za znastvena istraživanja. Isto tako se koriste i za specijalne efekte u filmovima za što realističniji prikaz scene.

2.1. Znanstveni sustavi

Koriste se u raznim granama ljudskih djelatnosti, pa se tako u početku koristilo u vojne svrhe za izračun balističkih projektila artiljerije s obzirom na kut pod kojim se ispaljuju i na količinu baruta. Isto se je uračunavao i utjecaj vjetra na skretanje projektila.
Uporabom jačih kompjutera (superračunala) izvode se neke kompleksnije simulacije kao što je dinamika fluida. U ovom slučaju se česticama dodjeljuju vektori sila koji se kombiniraju i tako se dobiva cirkulacija fluida. Za ove potrebe se razvijeni specijalni procesori – vektor procesori – koji ubrzavaju cijeli proces. ***[16]***

2.2. Sustavi računalnih igara i animacija

Korištenjem simulacije mehanike u igrama i računalnim animacijama postižemo efekte koji se doimaju realnijim u scenama koje prezentiramo gledatelju. U ove sustave implementirani su zakoni mehanike koji približno "dočaravaju" (opisuju) mehaniku iz realnog svijeta.

Ovi sustavi su razvijeni tako da bi simulirali mehaniku, a pri tome zadržali optimalan *frame rate* kako ne bi dolazilo do pretjeranog zauzeća procesorskog vremena što bi se odrazilo na igrivost odnosno neigrivost. Znači, igrivost je stavljena u prvi plan a preciznost simuliranja ipak nije toliko izražena. Mehanički fenomeni koji se prikazuju su obično jednostavnije varijante mehaničkih modela u stvarnosti. Bitno je prikazati neku pojavu kao percepcijski realnu i tu se koriste i neki drugi načini animacije.

Simulacija mehanike u računalnim igrama se sastoji od nekoliko komponenti:

- programskog koda
- detekcije sudara i odgovora na sudar – potrebno za slučajeve kada imamo dva ili više objekta na sceni koji dolaze u interakciju
- simulacija fluida
- animacijski kontrolni sustav
- integracija sredstava

Prve dvije čine samu srž sustava za simuliranje mehanike. Simulacija fluida, animacijski kontrolni sustav i integracija sredstava su dodatne komponente koje se mogu pronaći u modernim sustavima.

Tako imamo i tri obrazca mehaničke simulacije u računalnim igrama:

1. metode kazne (*penalty methods)* – interakcije se događaju po principu materija-aktivacija (meka tijela)
2. metode ograničenja (*constraint based methods*) – zanivaju se na jednadžbama ograničenja kojima se procjenjuju mehanički zakoni

3. metode pobude (*impulse based methods*) – do interakcije dolazi pobuđivanjem (impulzivnim djelovanjem) objekata

Hibridne metode kombiniraju sve gore navedene.

Simuliranje mehanike se bazira na dvije vrste simulacije materije:

1. kruto tijelo – objekti su stavljeni u kategorije s obzirom kako bi se trebali ponašati u interakciji s drugim objektima (zauzimaju manje računalnih resursa)
2. deformabilno tijelo – odnosi se na simulaciju dijelova jednog objekta

U početku, za animiranje likova, koristila se je samo mehanika krutih tijela upravo zbog svojih prednosti brzog i lakšeg računanja. Danas se koriste i deformabilna tijela za animaciju likova u filmovima i igrama, a koriste se i za čestične efekte, fluide, itd.

U nastavku teksta dani su primjeri nekih sustava koji uvelike "opterećuju" računalne resurse i zbog toga je potrebno poznavati kako i gdje se služiti njima.

2.2.1. Sustav čestica

Prilikom kreiranja sustava čestica kao što su oblaci, dim, padaline i dr., koristi se veliki broj 3D koordinata koje se sastoje od točaka, poligona, teksturiranih znakova ili *sprite*-ova. Ovakvi sustavi su jedni od najvećih potrošača računalnih resursa. Zbog hardverskog ograničenja mora se voditi računa o tome kako se modeliraju ovakvi efekti. Tako se rade kompromisi pri kreiranju ovih efekata i dosta je bitna kreativnost samog dizajnera ili modelera kako što vjernije prikazati ove efekete, a opet imati brzinu izvođenja scene.

Slika 2. Primjer čestica na sceni

Izvor: http://img.youtube.com/vi/_5QkPnPDcfc/0.jpg

2.2.2. Ragdoll sustav

Riječ *ragdoll* dolazi od engleskog naziva za igračku (*rag doll* – mekana lutka napravljena od dijelova tkanine). *Ragdoll* sustav se, u računalnim igrama, koristi za simuliranje pokreta kada je neki od likova ubijen. Radi na principu povezanosti kostiju u ljudskom tijelu na način da jedna kost utječe na pomicanje drugih kostiju odnosno cijelog tijela. Mogu se postići i finiji efekti u interakciji s drugim objektima (npr. tekućine), ali tada je zahtjev za računalnom snagom još veći.

2.2.3. Projektili

Simulacija projektila je isto jedna od komponenti koja se koristi u računalnim igrama. Recimo u simulacijama nogometnih utakmica potrebna je simulacija lopte, njene putanje, da bi se dobio što realističniji prikaz. ***[15]***

2.2.4. Rasterećenje centralnog procesora (CPU)

Kako bi se riješili problemi s pretjeranim zauzimanjem procesorskog vremena glavnog procesora razvijene su specijalizirane procesne jedinice koje obavljaju računanje sustava za simulaciju mehanike. Tako imamo rješenje s posebnim mikroprocesorom koji je zadužen samo za izvođenje simulacija u računalnim softverima i rješenje gdje se simulacije mehanike procesiraju preko grafičke procesne jedinice (*GPGPU*).

"Opće procesiranje na grafičkoj procesnoj jedinici"

General Purpose processing on Graphics Processing Unit ili skraćeno *GPGPU* je rješenje koje susrećemo u zadnjih nekoliko godina kod svih većih proizvođača grafičkih kartica (*Nvidia* i *AMD*). Na ovim sustavima podržana je mehanička simulacija za dinamiku krutih tijela. Opcija *GPGPU*-a je podržana putem njihovog sustava *CUDA – Compute Unified Device Architecture*. Ovaj alat omogućuje da se više procesa izvršava paralelno, a ne samo jedan – brzo. *AMD* ima slično rješenje, imenom *Close to Metal – CTM*.

Procesna jedinica za simulaciju mehanike

Postoje i primjeri stvaranja računalne jedinice koja obavlja procesiranje samo mehaničkih simulacija na računalu. *PPU – Physics Processing Unit* je mikroprocesor koji brine za izračunavanje simulacija na računalu. Ideja ove procesne jedinice je da se koristi za izračunavanje simulacijskih algoritama i time se rasterećuje centralni procesor koji se onda koristi u druge svrhe. Konkretnije, *PPU* se koristi za izračun dinamike deformabilnih tijela, krutih tijela, detekcije sudara, analizu ograničenog elementa, simulacije tkanine i kose, prijeloma kod objekata, dinamke fluida, itd. Poznatiji primjer ovakvog sustava je *Ageia*-in *PhysX* čip.

2.2.5. Bullet i Physix - kratak pregled

Bullet

Bullet je računalni sustav za simuliranje mehanike koji se koristi u računalnim igrama i specijalnim efektima filmova. Ovaj sustav podržava detekciju sudara i dinamiku deformabilnih i krutih tijela.

Svoju vezu sa drugim aplikacijama ostvaruje putem dodataka (*plugins*), recimo kod *Softimage*-a i *Maya*-e. Integriran je u *Blender*, *LightWave 3D*, *Houdini* i *Cinema 4D*. Importiranje (uvoz) podataka ostvaruje preko *COLLADA*-e (*collaborative design activity* – format za razmjenu datoteka između interaktivnih 3D aplikacija) datotečnog formata. ***[17] [20]***

Bullet se koristi u mnogim aplikacijama za izradu interaktivnog digitalnog sadržaja, a koristi ga i velik broj *game engine*-a (sustava za izradu računalnih igara). Primjeri nekih poznatijih holivudskih filmova i računalnih igara u kojima je korišten *Bullet*:

- filmovi: *Sherlock Holmes* (2009), *Bolt* (2008), *2012* (2009)
- računalne igre: *Toy Story 3: The Video Game*, *Grand Theft Auto IV* i *V, Red Dead Redemption, DiRT* serijal

Slika 3. Korištenje *Bullet*-a u filmu *2012* iz 2009. godine

Izvor: http://www.fxguide.com/wp-content/uploads/2011/01/2012_featured.jpg?0515bf

PhysX

PhysX se koristi u raznim softverima za izradu simulacija digitalne grafike (*Autodesk*-ovi *3ds Max*, *Maya* i *Softimage*, i dr.), ali se uglavnom koristi u računalnim igrama i *game engine*-ima za izradu istih (*Unreal Engine 3*, *Unity*, itd.). Neke od poznatijih *blockbuster* igara u kojima se koristi su:

- *Mirror's Edge*, *Mafia II, Need for Speed: Shift*, *Batman: Arkham City*, *Borderlands 2*

Ovaj sustav podržava dinamiku deformabilnih i krutih tijela, *ragdoll* sustav i kontrole likova, dinamiku vozila, simulaciju volumetrije fluida, čestice i simulaciju tkanine (efekti poput tkanine pod pritiskom i trganja).

PhysX je nastao iz sustava za simulaciju mehanike koji se u početku zvao *NovodeX*. Kompanija koja je proizvela *NovodeX* je prešla pod vlasništvo kompanije *Ageia* koja je kasnije došla u vlasništvo kompanije *Nvidia*.
Još dok je *NovodeX* bio u vlasništvu kompanije *Ageia*, započelo se na razvoju hardvera koji će ubrzati računanje algoritama potrebnih za simulaciju mehaničkih modela na način da se rastereti centralni procesor. Ova tehnologija je nazvana *PhysX PPU* (*physics processing unit*), a softverski dio ovog sustava ("*devkit*" ili *SDK – software development kit*) je preimenovan u *PhysX*. Poslije prelaska pod *Nvidia*-u razvija se hardversko ubrzanje za sustave simulacije na *Nvidia*-inim grafičkim karticama. Kasnije se odustalo od *Ageia*-ine procesne jedinice za simulaciju mehanike. Umjesto zasebne procesne jedinice za simulaciju razvijao se sustav *GPGPU*. Ovaj sustav je omogućio paralelno procesiranje kompleksnih algoritama koje je potrebno izvršiti i pokazao veću učinkovitost nego centralni procesor. ***[18]***

3. BLENDER

Blender je danas vjerojatno najpoznatiji besplatni grafički softver otvorenog koda (*open source*) za 3D modeliranje i animaciju na svijetu. Koristi se za izradu specijalnih vizualnih efekata, animirane filmove, digitalne umjetničke slike, računalne igre, interaktivnu grafiku i 3D modele za printanje. Rad na njemu je započeo 1993. godine i od tada je doživio mnogo unaprjeđenja. Blender v2.5 se često spominje kao verzija koja je promjenila i poboljšala mnoge funkcije, tečnost rada i sučelje programa. Ovaj rad je pisan s obzirom na aktualnu verziju Blender v2.71.

Blender kao paket sadrži sve potrebene alate za izradu filma i interaktivne grafike – alati 3D modeliranja i skuplturiranja (klesanja), UV razmotavanja, teksturiranja, editiranja rasterske grafike, pripreme modela za montažu i animaciju, animiranja likova i objekata na sceni, simulaciju mehaničkih fenomena, rasvjete, usklađivanja pokreta, skriptiranja, sklapanja scene, renderiranja i postprodukcije. ***[2] [14] [19]***

Kada bi pokušali animirati scenu vatre s dimom ili vijorenje odjeće (tkanine) na vjetru na klasičan način u 3D programima za modeliranje i animiranje, ubrzo bi shvatili da je opseg takvog posla prevelik i vremenski predug. Prikaz takve scene vjerojatno ne bi zadovoljio gledatelja koji traži što je moguće realniji prikaz ponašanja tkanine na vjetru koja se valovito, nasumično giba na vjetru. Ili isto tako prikaz pramenova vatre koji gore i iz njih se izdiže volumetrijski dim s milijardama čestica dima. Upravo iz tog razloga, kod animacije ovakvih pojava, koriste se algoritmi simulacije.
Blender, za ovakve pothvate, koristi simulacijski sustav. Imenom *Bullet*, ovaj sustav se čak koristi u holivudskim *blockbuster* filmovima pri izradi specijalnih efekata zgrada koje se urušavaju i simulacije krhotina koje padaju.

3.1. Vrste mehaničkih simulacija u Blenderu

U Blender su ugrađeni mnogi alati za simulaciju prirodnih fenomena. Na taj način imamo mogućnost izrade najrazličitijih animacija koje kao ograničenje imaju samo autorovu (dizajnersku) maštu.

Popis važnijih stanja materije i fenomena koje je moguće simulirati:

- fluid, kiša, plin, dim, prašina, tkanina, kosa (vlas), trava, jato (jato ptica), krzno, pero, mekana tijela (koža, guma, želatina, itd.), čvrsta tijela (kamen, beton, metal, itd.), polje sile (sila, magnet, vjetar, vrtlog, itd.), gravitacija (definirana je kao početna postavka u programu, vrijednosti -9.81 po Z osi)

4. MODIFIKATORI

Modifikatori su operacije kojima automatski izvršavamo različite radnje nad objektima bez da utječemo na njihovu topologiju, tj. djelujemo na njih na nedestruktivan način. Primjenjujemo ih kada želimo izvršiti neku radnju nad objektom za koju bi ručno (manualno, korak po korak) potrošili puno vremena. ***[5]***

Blender, u svojem odjeljku opcija odnosno svojstava, nudi nam panel modifikatora. Panel modifikatora se sastoji od četiri dijela:

- Preinačenje (*Modify*) – ova skupina modifikatora utječe na podatke kao što su vršne točke
- Stvaranje (*Generate*) – konstrukcijski alati za dodavanje novih geometrijskih oblika
- Deformiranje (*Deform*) – imamo utjecaj na promjenu oblika objekta
- Simuliranje (*Simulate*) – pomoću ovih modifikatora aktiviramo simulacije

Slika 4. Modifikatori u Blenderu

Izvor: http://wiki.blender.org/uploads/thumb/7/7a/25-Manual-Modifiers-menu.png/600px-25-Manual-Modifiers-menu.png

4.1. Simulacijska grupa modifikatora

Ovi alati rade u kombinaciji s Blenderovim sustavom za simulaciju mehanike. Pomoću njega stvaramo animacije koje se odnose na najrazličitije prije spomenute efekte koje želimo da se izvršavaju kao simulacija. Verzija Bledner 2.71, u svojoj ponudi tako sadrži:

- Tkanina – uporabom ovog modifikatora mreži objekta (*mesh*) omogućavamo da se objektu dodjele svojstva koja karakteriziraju ponašanje tkanine.
- Sudar – ako želimo da nam nekoliko objekata na sceni bude u međusobnoj interakciji primjenjujemo ovaj modifikator.
- Dinamičko Slikanje – omogućuje jednom objektu slikanje po drugom objektu.
- Eksplodiranje – koristan modifikator kada želimo simulirati efekt eksplozije. Rastvara (rasprskuje) *mesh* objekta na sastavne dijelove.
- Simulacija Fluida – kako i sam naziv sugerira, modifikator koji definira objekt sa svojstvima karaterističnim za fluide.
- Ocean – brza opcija koja nam služi ako trebamo veliku količinu vode u sceni (simulacija vode na površini).
- Čestična Instanca – služi nam za simuliranje čestica pri čemu se koristi oblik *mesh-a* (mreže).
- Sustav čestica – sustav za simulaciju efekata koji se sastoje od velikog broja čestica kao što su dim, vatra, iskre, vatromet, trava, itd.
- Dim – modifikator za stvaranje simulacije realnog ponašnja dima.
- Deformabilno Tijelo – prije modifikatora za tkaninu koristila su se deformabilna tijela za simuliranje efekta tkanine. Ovaj modifikator se koristi i za druge efekte kao što su želatina, jastuk, spužva.

Ukoliko, za vrijeme rada u Blenderu, uključimo simulaciju ili sustav čestica, u većini slučajeva, ovi modifikatori će biti dodani na stog (modifikatorski stog). Njihova jedina uloga je da definiraju mjesto u stogu i na taj način služe kao baza podataka.

5. BLENDEROV SUSTAV ZA MEHANIČKU SIMULACIJU

U ovom dijelu ovoga rada opisani su glavni aspekti i alati pomoću kojih se izvršavaju simulacije u Blenderu. Na ovaj način opisa simulacijskih mogućnosti htjelo se pokazati koliko je razvijen sustav za simuliranje mehaničkih fenomena, a time se htjelo ukazati na potrebu za postojanjem ovakvih alata. Današnji kompjuterski generirani filmovi i interaktivna grafika (računalne igre) koriste sve mogućnosti ovih alata i na taj način oni su postali standard svakog bitnijeg 3D programa za modeliranje i animaciju. Blender, kako ćemo vidjeti, ne zaostaje za "velikim igračima" na ovom području.

Kako je već navedeno, Blender u sebi ima ugrađen sustav koji se brine o što je moguće realnijem simuliranju pojava i interakcije materije u stvarnosti.
Kada u svojstvima odaberemo panel *Physics* sustava, na samoj kartici nam je ponuđeno nekoliko opcija dodijeljivanja načina kojima želimo utjecati na određeni objekt odnosno objekte. ***[3]***
Tako ovdje nalazimo slijedeće opcije koje su po nazivima slične onima u odjeljku simulacijskih modifikatora, a neke su nove (Blender v2.71):

- Polje Sile (*Force Field*)
- Sudar (*Collision*)
- Tkanina (*Cloth*)
- Dinamičko Slikanje (*Dynamic Paint*)
- Deformabilno Tijelo (*Soft Body*)
- Fluid
- Dim (*Smoke*)
- Kruto Tijelo (*Rigid Body*)
- Ograničenje Krutog Tijela (*Rigid Body Constraint*)

Kada želimo postići da neki objekt na sceni bude dio simulacije ovdje odabiremo vrstu mehaničke interakcije odnosno stanje materije.

Primjer

Recimo da želimo balon napunjen vodom ispustiti iz ruke i vidjeti što se događa prilikom kontakta s tlom. U tom slučaju kreirali bi objekt u obliku kugle (zbog jednostavnosti primjera) kojem se dodjeljuje atribut deformabilnog tijela. I bio bi nam potreban još jedan objekt, recimo ravna podloga, kojoj se dodjeli atribut sudara. Kada pokrenemo animaciju dolazi do izvršavanja simulacije. Već u ovom trenutku možemo primjetiti da dolazi do izračunavanja interakcije ova dva tijela. To se očituje u prikazu brojčanih vrijednosti sličica po sekundi (*frames per second - fps*) i vrlo je primjetno sporije izvršavanje simulacije ako imamo slabije računalo koje nije opremljeno sa snažnijim računalnim komponentama za obavljanje ovakvih radnji. Vidljivo je i opterećenje procesorske jedinice (*CPU – central processing unit*) ukoliko pratimo rad procesora na nekom od softvera. Svaki put kada mijenjamo postavke na sceni dolazi do ponovnog izračunavanja simulacije i time ponovnog zauzeća procesorskog vremena.

Svaka od ovih opcija unutar kartice *Physics* sustava posjeduje veliki broj karakteristika i podopcija specifičnih za svaku od kategorija posebno. Na takav način omogućeno nam je kreirati široki spektar simulacija potrebnih za scenu.

5.1. Opis opcija sustava za mehaničke simulacije

Ovdje je prikazan kratak opis svih pojava i stanja materije koje je moguće simulirati unutar Blendera. Kao što se može naslutiti, nakon čitanja ovog opisa, neke od ovih pojava i stanja moguće je dobiti i nekim drugim načinima animiranja, ali opet velik broj efekata koji proizlaze korištenjem ovih alata simuliranja je još uvijek zasad nezamjenjiv jer je opseg posla i vrijeme utrošeno da bi se animirali na klasični način – ogroman i neisplativ.

Neke efekte moguće je dobiti na klasičan način u zadovoljavajućoj kvaliteti kada uzmemo u obzir konačan proizvod i njegove načine prikaza. U takvim slučajevima se možemo koristiti i drugim načinima prikazivanja istog vizualnog efekta ili barem sličnog. To će biti pokazano na primjerima simulacije zastave koja se vijori na vjetru i gibanja valova na moru.

Polje sile

Vrste polja sile:

- Sila – radijalno polje koje je usmjereno prema centru objekta
- Vjetar – usmjerena konstantna sila
- Vrtlog – spiralna slia koja uvrće
- Magnetsko – polje koje ovisi o brzini čestica; simulacija magnetizma stvari
- Harmonijsko – tijelo koje harmonijski titra; harmonijski oscilator
- Naboj – polje koje se temelji na zakonima naboja
- Lennard-Jones – temeljno na Lennard-Jones potencijalu
- Tekstura – ovo polje zasniva se na teksturi
- Vođenje uz pomoć krivulje – ako želimo kreirati polje uzduž objekta koji se savija
- *Boid* – koristi se za simuliranje kretanja jata ptica
- Turbulencija – za stvaranje turbulentnih simulacija u kojima se koristi polje buke (zvuka)
- Vuča – polje koje prigušuje kretnju
- Protjecanje dima – kako i samo ime kaže, koristimo ga za simulaciju kretanja dima u zraku

Svako od ovih polja sile ima još veliki broj posebnih karakteristika koje je moguće kontrolirati kroz njihova zasebna svojstva i na taj način dobiti efekte kakve želimo na sceni. ***[6]***

Iz ovog opisa polja sile vidimo da nam je kroz ove opcije dana mogućnost simuliranja nekih poznatijih fizikalnih fenomena koje susrećemo u svakodnevnom životu. Sa strane animacije i vizualne prezentacije ovih fenomena da se zaključiti da bi se neki ostvarivali lakše od drugih. Ali recimo, kvalitetno animirati scenu u kojoj se krdo bizona ili jato ptica kreće na velikom otvorenom prostoru bilo bi jako teško na klasičan način. Zato u ovakvim slučajevima mehanička simulacija je, zasad, jedino rješenje.

Sudar

Ovu simulaciju koristimo kod scena u kojoj se objekti nalaze u situacijama u kojima dolazi do sudara (kontakta). Tkanina, čestice i deformabilna tijela se mogu sudarati s objektima dok *boids* izbjegavaju objekte sudara. ***[7]***

Tkanina

Simulacija tkanine je jedna od kompliciranijih simulacija i zahtjeva dosta optimiziranja da bi se došlo do kvalitete prikaza i brzog izvođenja. Ukoliko se efekti tkanine izvode za računalne igre u kojima je potrebno renderirati scenu u realnom vremenu preporučljivije je koristiti varalicu. ***[5]***

Dinamičko slikanje

Relativno nov alat koji pomaže kod efekata kao što su objekti koji se postepeno zamrzavaju, tragovi koji ostaju u snijegu, kapljice kiše koje čine tlo mokrim, itd.
Radi na način da se određenom objektu dodijeli atribut kista, a drugom (recimo podloga, tlo) atribut platna. Vrlo pojednostavljeno, kistom crtamo po platnu i na taj način možemo stvarati zanimljive efekte. ***[8]***

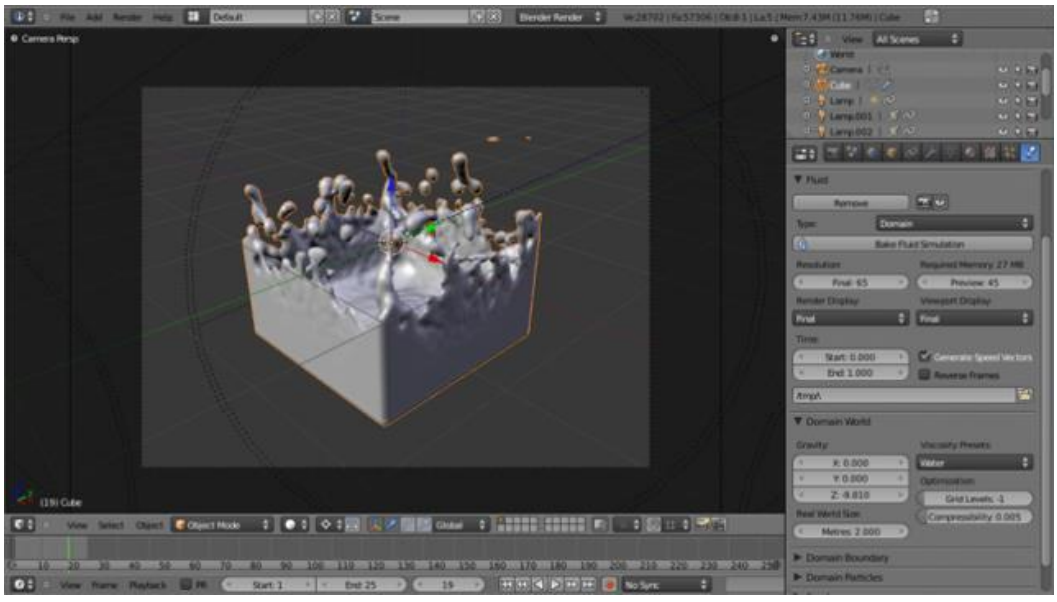
Deformabilno tijelo

Kako je već nešto spomenuto u dijelu o simulacijskim modifikatorima, deformabilno tijelo se koristi prilikom simulacije dosta velikog broja efekata. Uglavnom, to je simulacija nekog tijela koje se deformira u interakciji s ostalim tijelima. ***[9]***

Fluid

Alat koji koristimo za simulaciju fluida. Blender ima zanimljivo riješenu mehaniku ove simulacije gdje u početku moramo definirati dva objekta od kojih jedan predstavlja fluid, a drugi prepreku ili bolje rečeno “posudu za fluid”. “Posuda” predstavlja tzv. simulacijsku domenu koja je u obliku kvadra u kojoj se odvija simulacija fluida. Kada smo to definirali koristimo gumb “*Bake*” (“Ispeci”) da bismo pokrenuli eksportiranje svih postavki i geometrije u simulator. Dolazi do izvršavanja simulacije, generiraju se pregled za svaki *frame* (sliku) i pripadajuća površina *mesh*-a. Ti podaci se spremaju na *hard disk*. Od tamo se odgovarajuće površine fluida učitavaju za prikaz ili renderiranje. ***[10]***

- "Pečenje" (*Bake*) – ovaj proces se koristi kada se nešto treba predizračunati (*pre-computing*) kako bi se kasnije ubrzali neki drugi procesi. Kada se recimo radi renderiranje ispočetka, tada je potrebno dosta vremena da bi se izvršio ovaj proces. Tako opcija "pečenja" omogućuje "zapeći" neke dijelove rendera prije. Time dobivamo puno brže izvršavanje renderiranja jer se, recimo, boje ne moraju svaki puta iznova računati. ***[11]***



Slika 5. Fluid u kockastoj domeni

Izvor: https://cdn.tutsplus.com/cg/uploads/legacy/000_WebRoundups/28_45_Blender_Tuts/Fluid_Sim.jpg

Dim

Slično kako je bila riješena simulacija fluida, tako i dim koristi jedan objekt koji se ponaša kao emiter, a drugi kao domena u kojemu se izvršava simulacija. Dim se rasprostire samo unutar granica domene. Za simulaciju se koristi volumetrij-sko fluidno-bazirani model. Kao rezultat imamo mrežu voksela koju možemo vidjeti u Blenderovom 3D okviru. Simulacija omotava voksele oko čestica. ***[12]***

Kruto tijelo

Alat za simuliranje krutih tijela u Blenderovom sustavu za igre (*game engine*). Postoji i u render sustavu. ***[13]***

Ograničenje krutih tijela

Želimo li prikazati kompicirane dinamične strukture i njihovo lomljenje ovo je alat koji nam to omogućava. Ograničenjima je moguće spajati kruta tijela. ***[13]***

6. "VARALICE"

Varanje u modeliranju i animaciji 3D grafike je poželjna pojava. To je način kako da prikažemo neku scenu a da pri tome imamo normalnu brzinu izvođenja, odnosno da ne dolazi do problema s zastajkivanjem ili usporenošću scene. Vrijeme je faktor koji nam je bitan u cijeloj priči.

Bitno je poznavati sustave s kojima se radi i kako oni utječu na brzinu izvođenja. Dobro je znati alternativna rješenja izvođenja nekog prikaza koji možda i nisu “realno precizne” simulacije, ali nam perceptivno daju isti ili sličan doživljaj kao što nam daje simulacija.

Varalica u ovom radu označava način animiranja neke pojave ili objekta gdje se koriste različita svojstva objekata i njihovi oblici kako bi se postigli efekti pokreta, deformacije ili preobličavanja. Na ovaj način animiranja uvelike uspijevamo očuvati računalne resurse, drugim riječima znatno smanjujemo procesorsko vrijeme i time dobivamo fluidnije izvođenje cijele scene.
Varalicom "zavaravamo" gledatelja koji pecipira neku već viđenu ili doživljenu scenu iz stvarnog života zbog čega on ima bolji dojam ili spoznaju onoga što mu prikazujemo u kadru. Zbog toga smo u mogućnosti koristiti "trikove" kojima prikazujemo, recimo, gibanje tkanine na vjetru gdje nije potrebno apsolutno precizno prikazati svaki detalj mehaničkog gibanja tkanine. Dovoljno je dostići optimalnu razinu prikazivanja koja će zadovoljiti gledatelja gdje će on "povjerovati" da se, u konkretnom primjeru sa tkaninom, radi o sceni taknine koja se vijori na vjetru.

U ovom radu su prikazane dvije varalice od kojih je jedna poslužila za izradu animiranja tkanine na zastavi, a druga varalica je bila zadužena za gibanje valova na moru. Izrada ovih dviju varalica se razlikuje, a time i načini animiranja tih objekata odnosno pojava. Izrada ovih dviju varalica opisana je detaljnije u nastavku gdje se može dobiti bolji uvid u funkcioniranje i princip rada ovakvog načina animacije.

7. TKANINA

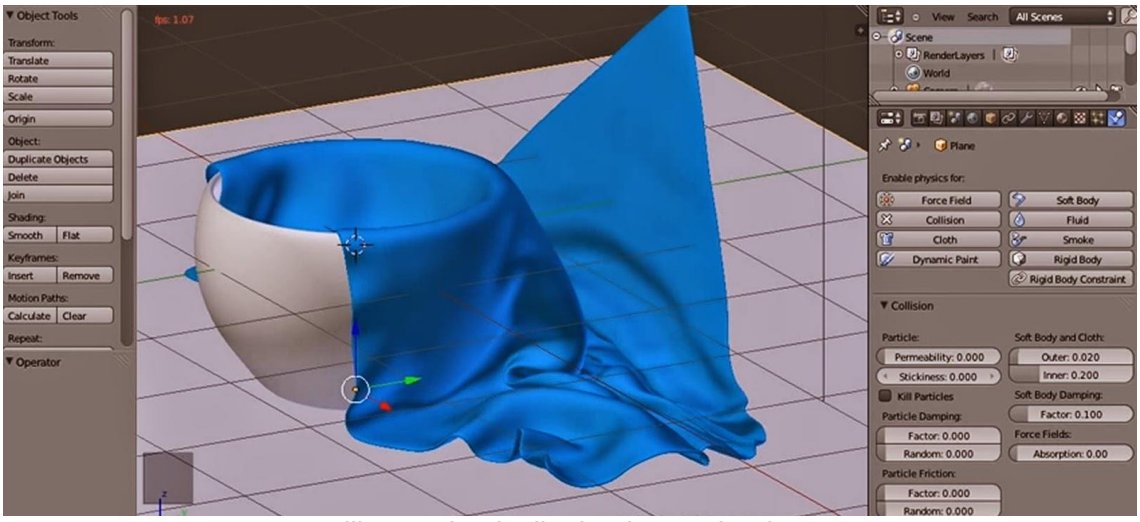
Kako je tkanina u stvarnosti prilično (relativno) jednostavan predmet, tako bi netko mogao pomisliti da njena simulacija unutar 3D programa za modeliranje i animaciju ne zahtjeva veliku vještinu i znanje - i tu se uvelike prevariti. Tkanina, zajedno s čestičnim efektima (fluidi, dim, itd.) spada u područje dinamičnih simulacija. Simulacija je ono što definira kako se tkanina ponaša kada je pod utjecem drugog objekta ili vanjske sile. Simulacija tkanine je jedna od težih stavki u kompjuterskoj grafici zbog strukture materijala i unutarnje i vanjske interakcije strukture (sa samom sobom ili objektima i pojavama iz njene okoline). Ali ukoliko smo u mogućnosti precizno simulirati akcije i reakcije tkanine u našim scenama to će se itekako odraziti na kvalitetu završnog proizvoda.

Moglo bi se reći da se tkanina u nekim slučajevima ponaša kao voda (valovito i slobodno protjecanje materijala), a opet u nekim slučajevima kao objekt čvršće strukture. Kako bi lakše razumjeli složenost (kompleksnost) simuliranja tkanine dobro bi bilo napraviti male pokuse u stvarnom životu. Recimo, pokušati ispustiti majicu iz ruke i promatrati kako se materijal i njegova struktura preklapaju i grupiraju u trenutku kada su u kontaktu s tlom ili kako se tkanina omata oko čvrstih objekata kao što je recimo drvena stolica. Naravno, različite vrste materijala tkanine će se različito ponašati u istim situacijama. Tako u Blenderu postoji nekoliko vrsta preddefiniranih materijala za tkaninu - u verziji Blender 2.71 nalazimo pamuk, traper ili *jeans* tkanina, koža, guma i svila. Objekti tkanine imaju svojstva koja ima omogućavaju da izvršavaju reakciju s obzirom na druge objekte i pojave u sceni. Možemo reći da postoje dvije vrste interakcije tkanine unutar 3D programa za modeliranje i animiranje, a to su interakcija tkanine s objektima (s drugim objektom na sceni ili sama sa sobom) i interakcija s izvanjskim silama kao što je recimo vjetar i gravitacija. ***[1] [4]***

U Blenderu svakom objektu možemo dodijeliti sustav tkanine i kao takvog ga simulirati i animirati po želji. Kod pripreme objekta koji će predstavljati tkaninu bitno je već u početku odlučiti njegovu važnost u sceni odnosno projektu. Time već u startu definiramo da li nam je objekt tkanine bitan ili manje bitan, odnosno da li finalni rezulat mora biti visoke kvalitete ili taj predmet ima sporednu ulogu

pa i nije toliko bitna njegova vizualna prezentacija. Upravo ti parametri utječu na procesorsko vrijeme renderiranja scene. Tako moramo definirati od koliko će se dijelova (poligona) sastojati objekt. To činimo preko opcije dijeljenja na manje dijelove (*Subdivide*). Bit ovog procesa je vrlo jednostavna i logična - ako imamo puno dijeljenja tada dobivamo puno poligona i veću rezoluciju objekta, a time finije deformiranje tkanine. Naravno, negativna strana ovoga je da trošimo više procesorskog vremena za računanje trenutne pozicije pojedinog poligona prilikom simulacije tkanine. U slučaju da nemamo puno dijeljenja (i time malo poligona) dobivamo objekt tkanine male rezolucije i grubih, "vidljivih" poligona prilikom deformacije tog objekta, ali isto tako štedimo računalne resurse i zbog toga se scena može brže odvijati. Odluka je na nama, da već u startu znamo za čega će se koristiti određeni objekt i na taj način imati kontrolu nad scenom koju prezentiramo gledatelju - odnosno da vodimo računa o kvalitetnom, fluidnom izvođenju animacije i kvaliteti vizualne prezentacije - pronaći optimalno rješenje.

Kako bi animirali tkaninu koristimo se simulacijskim modifikatorima koji koriste sustav za simuliranje mehanike unutar Blendera. Efekte simulacije tkanine je moguće dobiti na dva načina. Jedan od njih je pomoću deformabilnih tijela (*Soft Bodies*), a drugi način je pomoću alata specijalizirinog baš za tkaninu.



Slika 6. Simulacija tkanine u Blenderu

Izvor: http://3.bp.blogspot.com/-acAeQuVIXs8/U3rUKFkgWBI/AAAAAAABFqo/GBOSO_qtAMM/s1600/Basic+Cloth+Simulation+in+Blender.jpg

Za animaciju tkanine putem varalice korišten je modifikator premještanja pa je u nastavku opisan ovaj modifikator i način na koji on funkcionira.

7.1. Modifikator premještanja

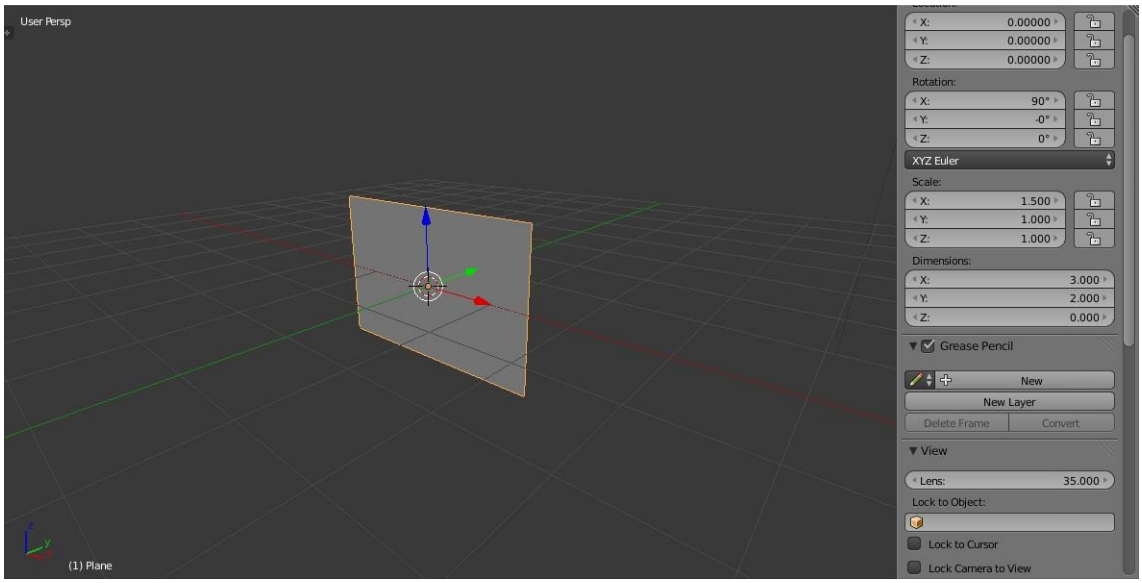
Displace (premjestiti, istisnuti) modifikator je alat koji omogućuje brzu izradu reljefnih oblika. Radi na principu pomicanja vršnih točaka i time deformiranja cijelog *mesh*-a. Premještanje se bazira na teksturi odnosno na jačini kotrasta teksture. Istiskivanje se u prostoru može događati putem odvojenih *RGB* komponenti, određene lokalne osi ili normale vršne točke.

Ovaj modifikator nudi podešavanje opcija gdje se odabire referentna tekstura za premještanje. Zatim je moguće odabrati grupu vršnih točaka za kontrolu utjecaja modifikatora. Opcija srednje razine (*midlevel*) utječe na dio teksture koji se smatra dijelom na koji modifikator nema utjecaj. Koordinate teksture se odnose na sustave koje je potrebno odabrati i prema tome načinu daljnje manipulacije objekta s obzirom na teksturu. Posljednja opcija je opcija snage ili intenziteta kojom kontroliramo jačinu istiskivanja. ***[21]***

7.2. Izrada simulacije za animaciju tkanine

Alat za simuliranje tkanine nije od početka bio dostupan unutar Blendera nego se za ovakve potrebe koristila opcija deformabilnih tijela. Čak i danas, za neke potrebe simulacije tkanine koriste se deformabilna tijela. U ovom radu simulacija tkanine je napravljena s alatom tkanine (*Cloth*).

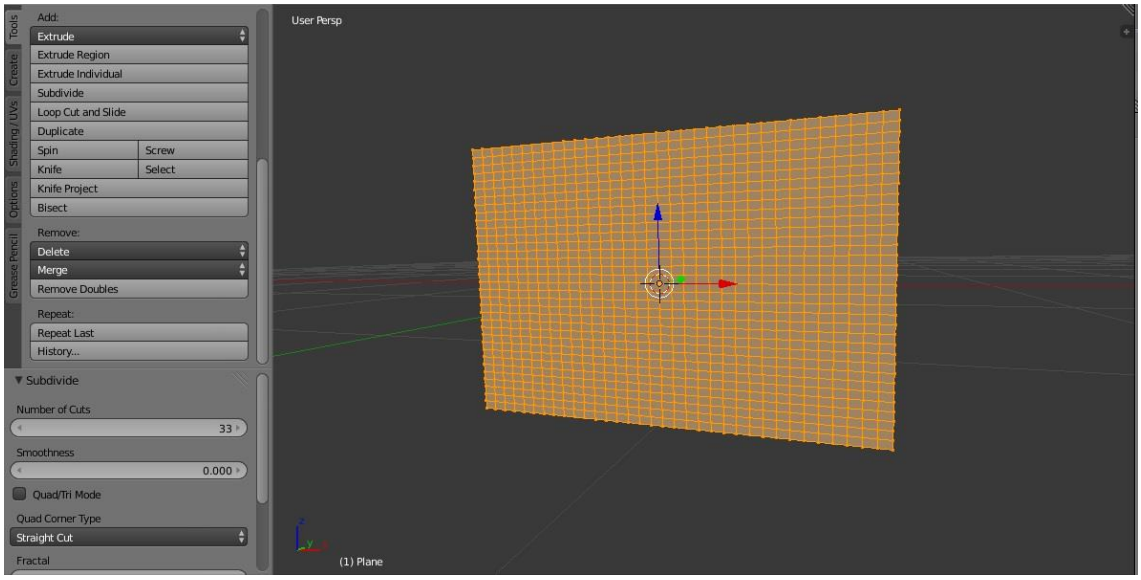
Kako bi kreirali tkaninu, na scenu prvo dodajemo mrežu plohe (*plane*). To činimo na način da na tipkovnici pritisnemo *Shift+A* i odaberemo *Mesh* → *Plane*. Zatim pomoću *Scale* alata ili unosom brojčanih vrijednosti u *Dimensions* okviru (*Tools* izbornik) određujemo veličinu buduće zastave. Dimenzija zastave po X vrijednosti je 3 metra, a po Y vrijednosti 2 metra. Nakon toga je zastava zarotirana za 90 stupnjeva po X osi (R → X → 90).



Slika 7. Određivanje veličine i rotacija

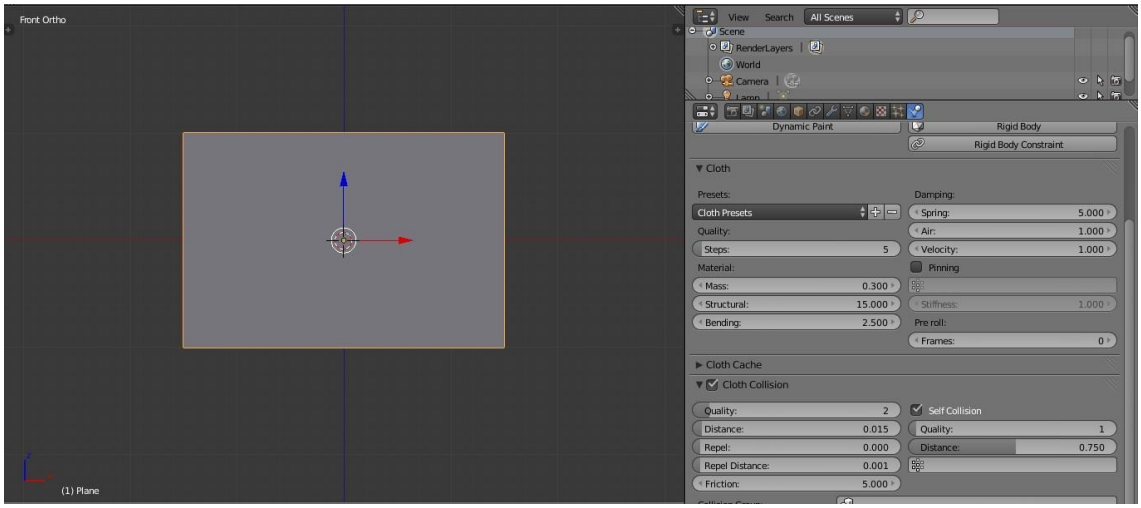
Potom zastavu dijelimo na dovoljno dijelova kako bi zadovoljili kvalitetu prikaza tkanine odnosno da se u animaciji ne primjećuju pravokutni oblici *mesh*-a. Ako je potrebno možemo još utjecati na “glatkoću” putem *Smooth* naredbe koja se nalazi na kartici *Tools*. Dijeljenje se obavlja u *edit* modu (Tab tipka) pomoću naredbe *Subdivide* koja se nalazi u *Tools* kartici. *Subdivide* naredba je pritisnuta jednom i u okviru *Number of Cuts* upisana je vrijednost 33, time je ploha

podjeljna u 1156 poligona. Vidljiva je promjena na mreži plohe. Na kraju je jedan put primijenjena naredba *Smooth*.



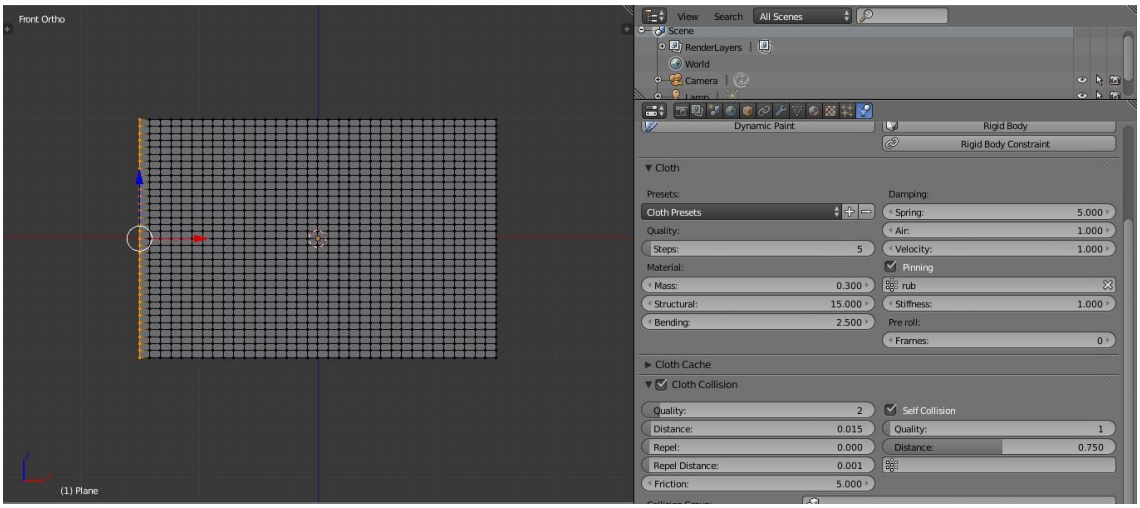
Slika 8. Dijeljenje plohe pomoću *Subdivide* naredbe

Nakon toga dodjeljujemo sustav za simulaciju tkanine (izbornik *Physics*) u kojemu namještamo sve potrebne stavke kako bi dobili što vjerniji prikaz tkanine koja se vijori na vjetru. Stavka *Mass* je namještena na vrijednost 0.3, *Bending* ima vrijednost 2.5, a *Spring* 5. Aktivirana je opcija *Self Collision* u kartici *Cloth Collision* kako tkanina ne bi prolazila kroz samu sebe. Još je namještena gravitacija u kartici *Cloth Field Weights* na vrijednost 0.3.



Slika 9. Definiranje *Cloth* sustava

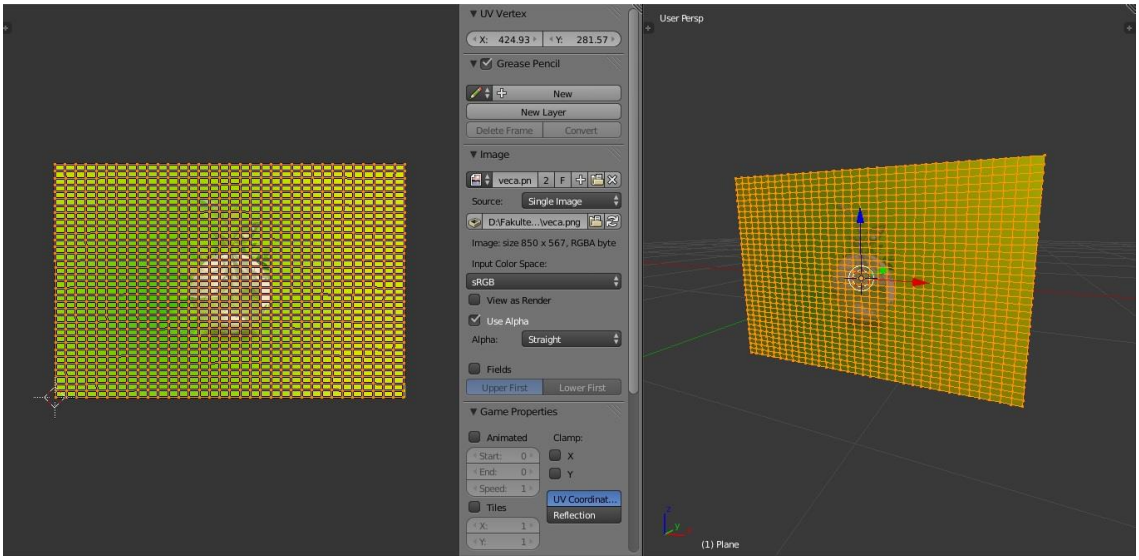
Potom je potrebno označiti vrhove na plohi koji će predstavljati dio zastave koji je pričvršćen na jarbol ili pritku. To možemo učiniti na nekoliko načina – autor ovog rada je to učinio u *Edit* modu gdje je odabrao vrhove (*vertices*) na lijevom rubu plohe i njih označio kao grupu. To činimo tako da odaberemo objekt tkanine, zatim pritisnemo tipku *Tab* koja omogućava editiranje objekta. Nakon toga selektiramo lijevu stranu vršnih točaka na objektu (selektiranje je lakše ako se radi u *Front Ortho* pogledu). U *Object data* izborniku, u kartici *Vertex Groups* stvorena je grupa odabranih vršnih točaka i dodijeljeno joj je smisleno ime. Ovo nam koristi za povezivanje s karakteristikama mehaničkih svojstava tkanine.
To činimo preko opcije *Pinning* (pričvršćivanje) koju je potrebno aktivirati u kartici *Cloth* (*Physics* izbornik) i ispod nje odabrati ime grupe vrhova koje smo odabrali kao rub koji je pričvršćen na jarbol – na ovaj način osiguravamo da se tkanina na tom dijelu ne pomiče.



Slika 10. Definiranje nepomičnog dijela tkanine

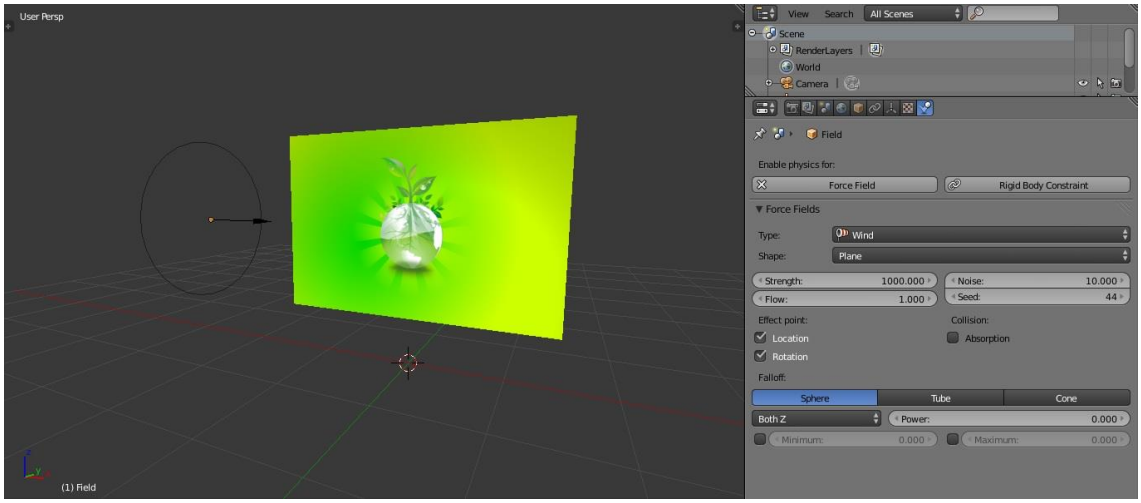
U izborniku *Material* plohi je dodijeljen materijal (izabrana boja - *Diffuse*, maknuto svojstvo odražavanja okoline – *Mirror*). Potom je na plohu stavljena tekstura zastave. Tekstura je napravljena u softveru za vektorsku grafiku *Inkscape*. Ilustracija je preuzeta s *web* stranice *Pixabay.* Njoj je pridružen gradijent koji čini pozadinu zastave. Spremljena je kao PNG datotečni format. Ovakva datoteka je omogućila teksturiranje preko Blenderovog alata *UV Mapping* gdje je moguće precizno definirati poziciju teksture.

Prvo je u izborniku *Texture* dodana nova tekstura. Zatim je učitana PNG datoteka s tvrdog diska na računalu koja je ranije kreirana. Nakon toga su vraćene vrijednosti objekta u *Scale* okviru na vrijednosti 1, da ne bi došlo do problema u prikazivanju. To je učinjeno na način da je izabran objekt i na tipkovnici pritisnuto *Ctrl+A* i zatim odabrana *Scale* opcija. Nakon toga bilo je potrebno prebaciti se u *UV Editing Screen layout* gdje je UV mapirana tekstura na objekt. Uz označen objekt trebalo se prebaciti u *Edit* mod i označiti sve vršne točke na zastavi. Pritiskom na tipku “U” otvara se izbornik u kojem je odabrana *Unwrap* opcija. To stvara mrežu na lijevoj strani *UV Editing layout*-a kojom precizno namještamo poziciju teksture. Zatim je u *UV Image Editor*-u odabrana ista datoteka koja je odabrana ranije kao tekstura. Nakon toga je bilo potrebno precizno postaviti mrežu preko slike teksture pomoću alata *Scale* i *Grab*.



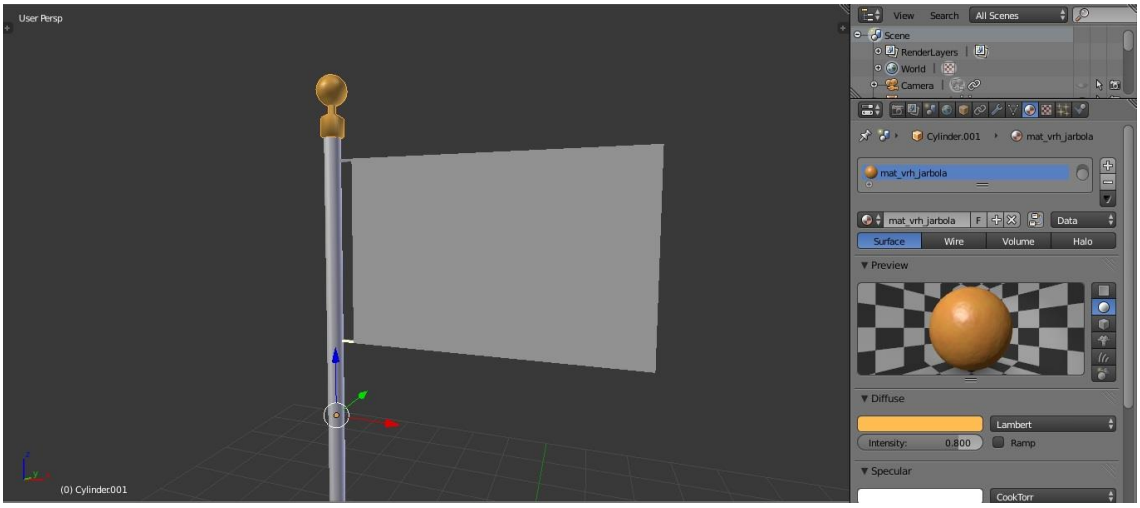
Slika 11. UV mapiranje

Kako bi dobili animaciju simulacije tkanine potrebno je bilo dodati i simulator vjetra. To je učinjeno tako da je na scenu dodan objekt vjetra koji radi putem *Physics* sustava. To činimo tako da odaberemo *Add → Force Field → Wind*. Podešene su opcije jačine (*Strength*) na vrijednost 1000, buke ili šuma (*Noise*) vjetra na 10 i *Seed* na 96. Nakon toga je ovaj objekt pozicioniran lijevo od zastave, a izvor vjetra zarotiran u pravcu širenja zastave.



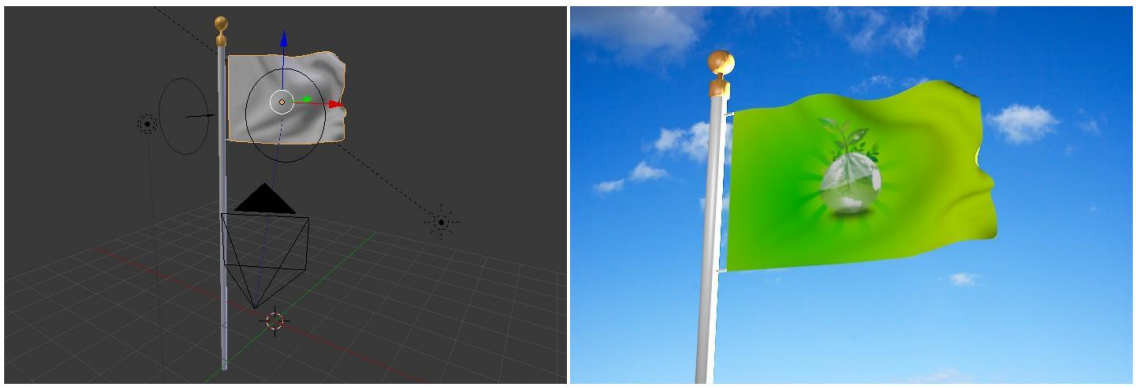
Slika 12. Dodavanje izvora vjetra

Na sceni je dodatno kreiran jarbol ili pritka za zastavu s dvije spojnice pomoću alata ekstruzije (E tipka na tipkovnici). Jarbol je napravljen iz jednog cilindra koji se sastoji od 32 segmenta. Dodan mu je materijal i siva nijansa boje. Postavke detaljnosti sfere nisu mijenjane već su preuzete početne vrijednosti kada dodajemo objekt sfere na scenu, isto mu je dodan materijal. Objekti su potom spojeni da bi sačinjavali jedan objekt. Nakon toga da bi se obojila “kapu” jarbola u “bakrenu” nijansu potrebno je označiti vršne točke sfere i “vrata” cilindra te njih odvojiti pomoću alata *Separate* (P). Potom je “kapi” dodan novi materijal s pripadajućom bojom. Nakon toga još je dodijeljena proceduralna tekstura vrste “oblak” da bi se dobio hrapvi izgled “kape”. U *Influence* izborniku aktivirana je boja i odabrana slična nijansa “bakrene”, i još je aktivirana opcija *Normal* (vrijednost 1) u dijelu *Geometry*. Još su na scenu dodane i dvije spojnice (spajaju zastavu i jarbol), a oblikovane su iz geometrijskog tijela kocke i dodan im je materijal.



Slika 13. Izrada jarbola i spojnica

U scenu je još dodana slika plavog neba s oblacima (preuzeta s *web* stranice *Pixabay*) koja predstavlja pozadinu scene. To je učinjeno putem izbornika *Texture* gdje je odabrana opcija *Show world textures* i u njoj je odabrana fotografija neba koja se nalazila na tvrdom disku računala. Aktivirane su opcije u *Influence* kartici za *Horizon*, *Zenith Up* i *Zenith Down*. Zatim su još u izborniku *World* i kartici *World* aktivirane stavke *Paper Sky* i *Blend Sky*. Svi objekti su pozicionirani tako da čine smislenu cjelinu koja prikazuje scenu vijorenja zastave na vjetru koja se nalazi na vrhu jarbola s nebom u pozadini.



Slika 14. Izrada simulacije tkanine i konačni kadar

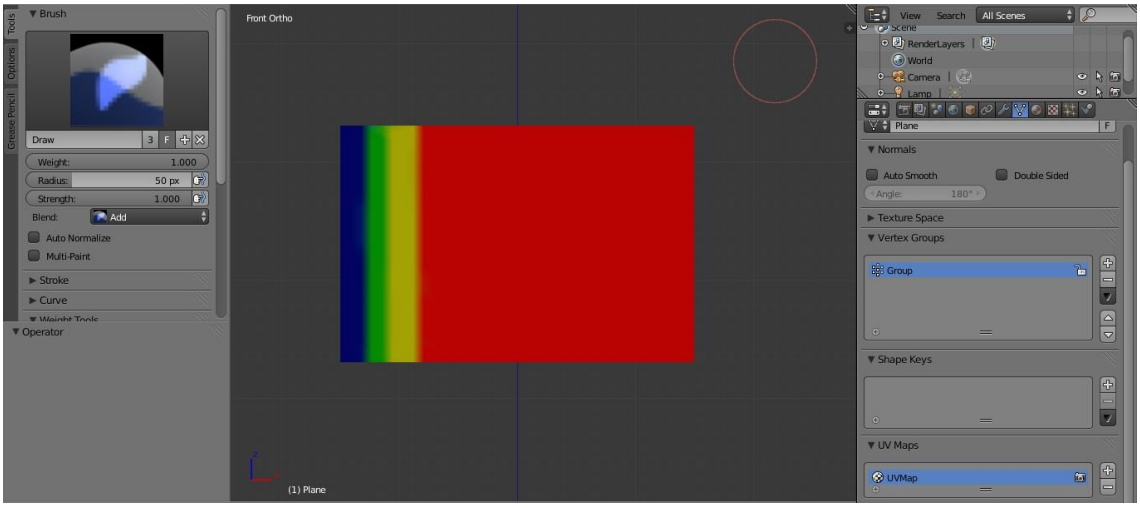
Na sceni se još nalaze dodatni objekti kao što je kamera povezana s praznim objektom (vrste “krug”) i dva izvora svijetla (vrste “točka” i “sunce”). Animiranje scene je napravljeno kroz 25 slika u sekundi, odnosno animiranje je imalo

sveukupno 501 sliku (zbog početka na nultom *frame*-u, a završetka na petstotom). To je dalo animaciju u trajanju od 20 sekundi. Za sveukupnu izradu ove animacije autoru ovog rada bilo je potrebno oko 15-ak minuta.

7.3. Izrada "varalice" za animaciju tkanine

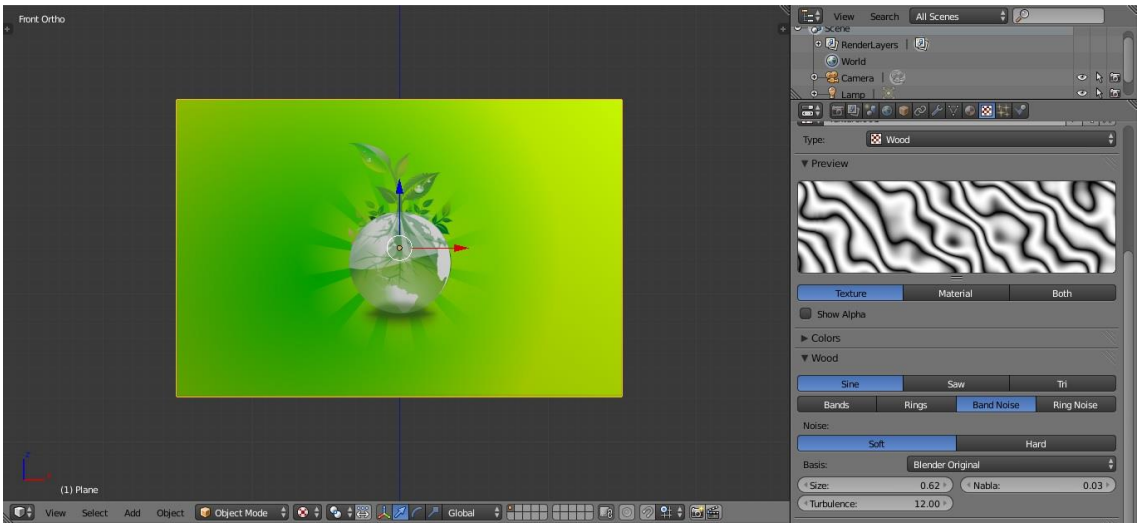
Kod izrade zastave za varalicu pazilo se da se zadrže iste dimenzije i detaljnost (dimenzije: 3 m * 2 m, 1156 poligona). Isto kao i kod izrade simulacije model zastave je kreiran iz plohe (*plane*) koju je podijeljena na isti način i s istim brojem poligona i primjenjena je naredba *Smooth*. Potom je dodan materijal i tekstura odnosno UV mapiranje na indentičan način koji smo koristili kod izrade simulacije tkanine.

Kako je potrebno prikazati da se tkanina ne pomiče na strani na kojoj je pričvršćena za jarbol tako moramo definirati područja da bi Blender znao koji je dio pomičan, a koji nije – odnosno koliko se pojedini dio zastave pomiče (ovo ima utjecaj na određivanje jačine *Displace* modifikatora). To je izvedeno preko "težine" područja plohe. Nakon kreiranja grupe vrhova (*vertex group*), uz pomoć alata *Weight Paint* (slikanje težine) označena su područja na zastavi. Plava boja ima vrijednost 0, a crvena 1. Boje između koristimo kako bi dobili finije prijelaze. Ovo je učinjeno tako da je označen objekt tkanine i u izborniku *Vertex Groups* kreirana nova grupa. Tada se treba prebaciti u *Weight Paint* mod i pomoću *Brush* alata definirati zone na zastavi.



Slika 15. *Weight Paint*

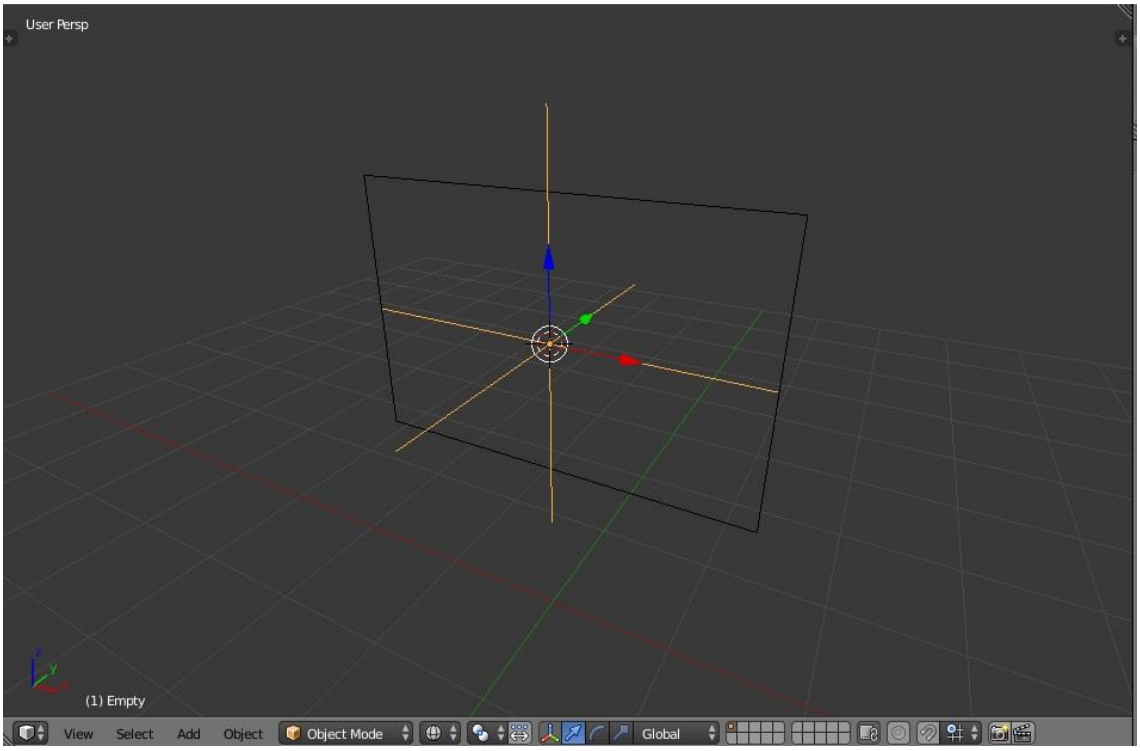
Prije same primjene *Displace* modifikatora kreirana je nova tekstura u drugom kanalu koja nam služi upravo za animiranje zastave. Korištena je "drvena" vrsta teksture koja svojim oblicima daje dojam gibanja valova. U kartici *Wood* izabran je *Band Noise*, *Size* je promjenjen na vrijednost 0.62, *Nabla* na 0.03 i *Turbulence* na 12. Odmah je postavljena kao neaktivna jer nam ne služi kao standardna tekstura.



Slika 16. Dodjeljivanje *Wood* teksture u drugom kanalu

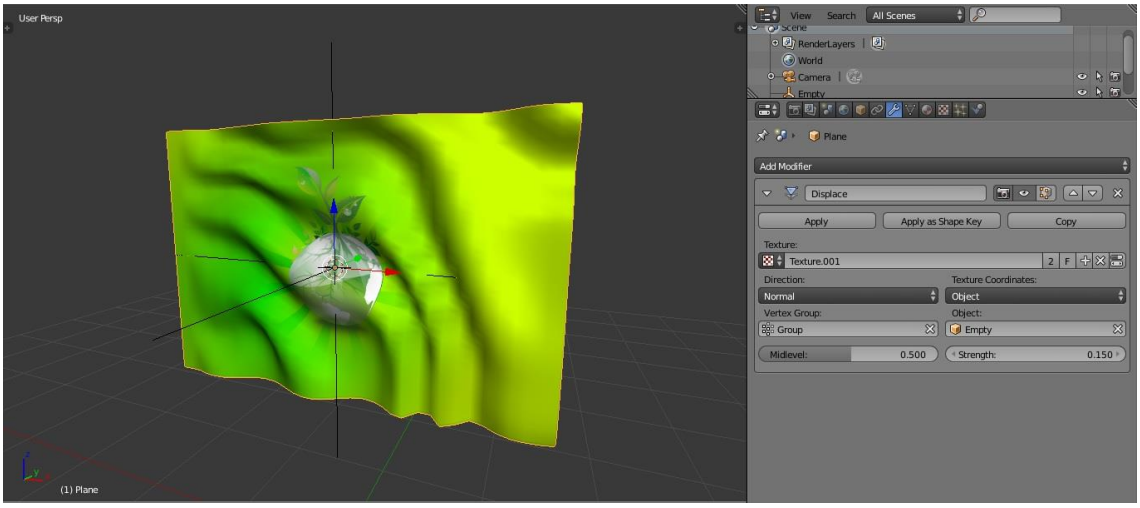
Kako bi stavili ovu teksturu u pokret kreiran je novi “prazni” objekt (povezujemo ga s *Displace* modifikatorom) koji svojim pomicanjem animira tkaninu. *Empty* je

dodan na scenu tako da je pritisnuto *Shift+A* i odabrano *Empty* → *Plain Axes*. Prije toga je određeno da kursor bude u sredini zastave kako bi se ondje smjestio novi *Empty.* Zatim je još skaliran da horizontalne linije, po X osi, budu na rubovima zastave.



Slika 17. Dodavanje *Empty* objekta na scenu (*Wireframe Shading*)

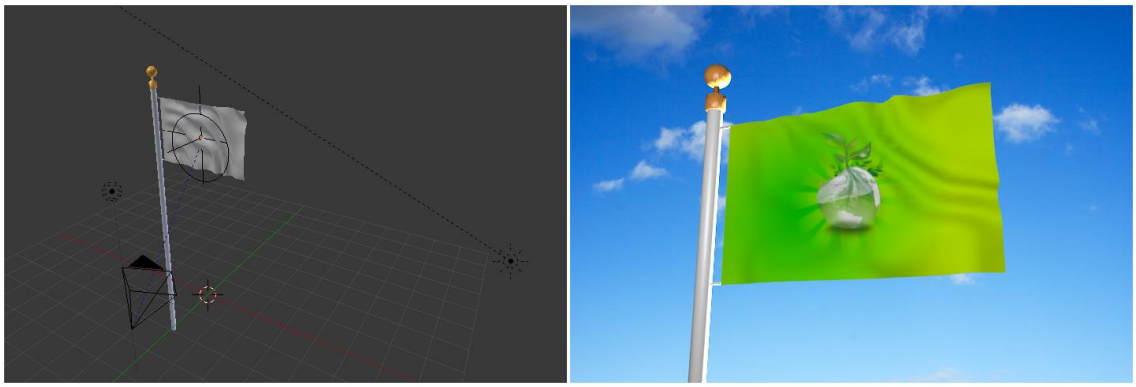
Nakon toga je primijenjen *Displace* modifikator na objekt zastave i podešeni su svi parametri potrebni da bi dobili efekt valovitog gibanja tkanine. U opcijama modifikatora bilo je potrebno postaviti teksturu drveta u okviru *Texture,* zatim je povezana težina vrhova točaka u *Vertex Group* okviru (koristi se grupa vršnih točaka koju je kreirana preko *Weight Paint* alata), koordinate teksture su prebačene na "objekt" opciju (da bi se objektni koordinatni sustav koristio za koordinate teksture), odabran je “prazni” objekt u *Object* okviru i promijenjena je vrijednost *Strength* na vrijednost 0.15.



Slika 18. Postavke u *Displace* modifikatoru

Objekt jarbola i poveznica je preuzet iz scene za simulacije zastave i ostali su nepromijenjeni. Isto tako imamo i dva izvora svjetlosti, kameru i “prazan” objekt za kardiranje scene. Kao i u sceni simulacije korištena je ista slika za pozadinu neba s oblacima.

Animacija je napravljena od 501 *frame*-a, a traje 20 sekundi – 25 *fps* (slika po sekundi). Animiranje pomoću varalice je napravljeno tako da je uz označeni *Empty* označen nulti *frame* na vremenskoj traci i dodan mu *keyframe* (ključna slika) lokacije (I → *Location*). Nakon toga označen je *frame* 500 i *Empty* je pomaknut za određenu udaljenost (udaljenost s obzirom na vrijeme određuje koliko će se brzo valovi gibati na tkanini). Uzeta je optimalna udaljenost da bi se dobio realističan prikaz brzine vijorenja zastave. Nakon toga postavljen je završni *keyframe* za animaciju scene. Autoru je trebalo isto oko 15-ak minuta da napravi varalicu.



Slika 19. Izrada varalice tkanine i konačni kadar

8. FLUIDI

Simulacijom fluida je moguće dobiti različite efekte i to ne samo tekućina već i dima, ekplozija i sličnih fenomena. Fluidi su materija koja se razlijeva po okolini, tako je i sustav koji simulira fluide napravljen na principu tog stanja. Mogli bi taj sustav podijeliti u dvije faze:

1. faza u kojoj fluid dolazi u okolinu
2. faza njihove interakcije, odnosno razlijevanje ili razvijanje fluida po okolini

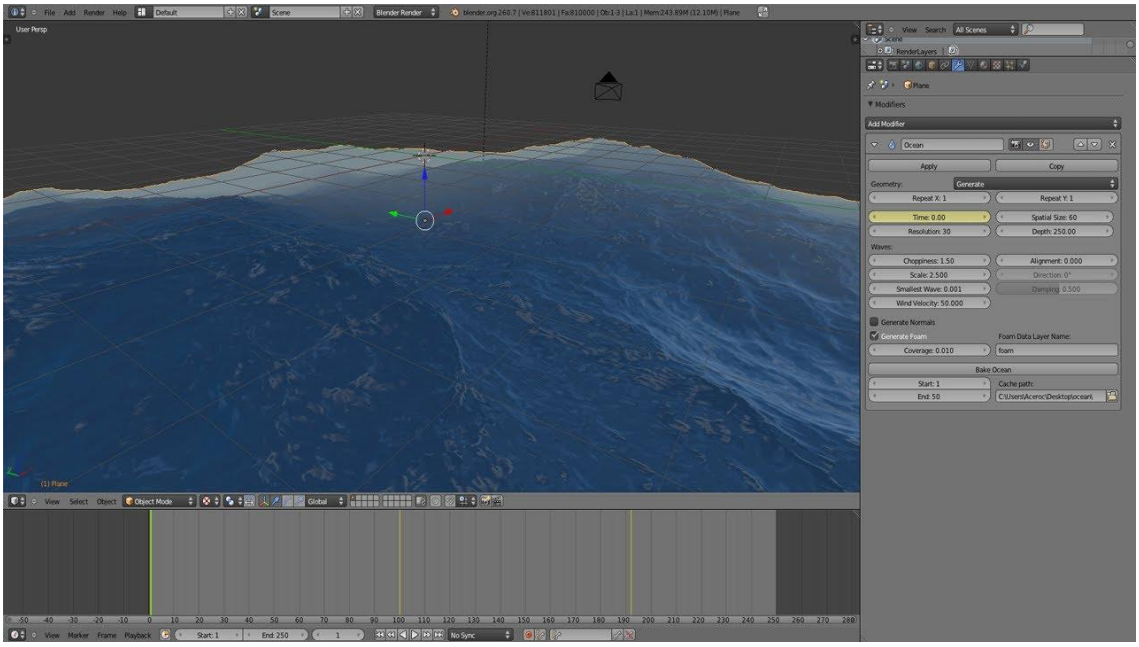
Simulacije fluida u području računalne grafike variraju od onih visoko kompleksnih koje se koriste kod izrade znanstvenih studija i specijalnih efekata za filmove, i onih jednostavnijih koje se odvijaju u realnom vremenu gdje se koriste čestični sustavi (računalne igre). ***[22]***

U Blenderu se, kako je već i prije spomenuto, koriste dva objekta za simulaciju fluida. Jedan predstavlja samo materiju fluida, a drugi je domena u kojoj se fluid razvija. Moguće je pomoću parametara odrediti gravitaciju i viskoznost s obzirom na domenu. Rezolucija fluida je jedna od najvažnijih stavki koje treba u početku uzeti u obzir i odrediti ju u vezi s onim što želimo dobiti. Rezolucija nam, u biti, određuje koliko će nam detaljna biti simulacija fluida. ***[23]***

Sustav za simulaciju fluida sadrži mnogo parametara s kojima je moguće postići najrazličitije efekte i prilično detaljno "ponašanje" fluida. Ovaj sustav se zbog svoje kompleksnosti jako oslanja na računalne resurse (radna memorija i procesor) i zbog toga nije bio dobar izbor za prikaz scene koja je bila zamišljena kao zadatak u ovom radu.
U ovom radu cilj je bio opisati i prikazati jednu jednostavniju simulaciju mora i laganog mreškanja valova. Blender ima poseban alat za izradu simulacije mora odnosno oceana – tzv. Ocean modifikator.

8.1. Modifikator oceana

Modifikatorom oceana simuliraju se i generiraju valovi odnosno deformacije na površini oceana. Moguće je simulirati efekte pjene koje stvaraju valovi.
Princip rada simulatora se zasniva na *FFT* metodama gdje se generiraju dvodimenzionalne koordinatne mreže simulacijskih informacija. Mogu se generirati tri tipa podataka, a to su: premještanje, normale i dodatni podaci koji se koriste za izračun vrhova valova. Kada je simulacija gotova mape se koriste da bi se premještala (istiskivala) odnosno deformirala geometrijska površina u trodimenzionalnom prostoru. Putem ovog modifikatora se također dodaje kanal boja vršnih točaka za efekt pjene na valovima.
Ponuđene su mnoge opcije kako bi se dobio veliki raspon mogućnosti kreiranja ovakvih i sličnih efekata. Moguće je utjecati na geometriju putem automatskog generiranja površine i proizvoljnog utjecaja. Dodavati nove, istovjetne površine onoj prvoj koju smo stvorili. Pomoću opcije vremena utječemo na brzinu animiranja valova. Rezolucijom kontroliramo omjer detaljnosti i brzine izvođenja simulacije oceana. Omogućeno je povećavati i smanjivati veličinu površine i dubinu ispod površine. Kontrola uzburkanosti oceana tj. oštrina valova se vrši putem opcije *choppiness*. Dalje je moguće namjestiti obujam valova, pravac valova s obzirom na vjetar, stupanj pod kojim su valovi poravnati, brzinu vjetra, itd.

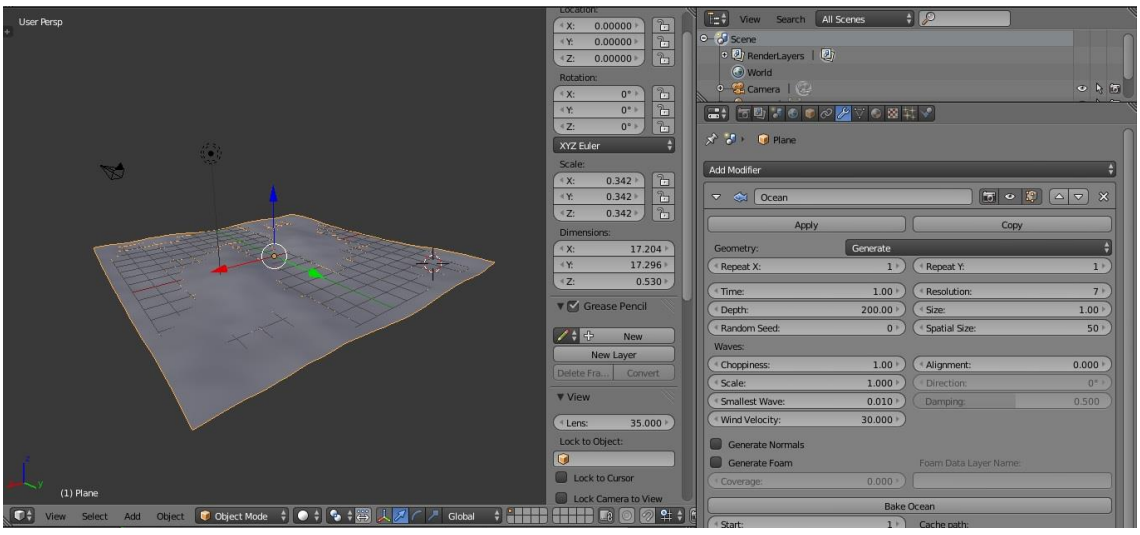


Slika 20. Simulacija oceana u Blenderu

Izvor: https://i.ytimg.com/vi/hZ8_ddfDxUE/maxresdefault.jpg

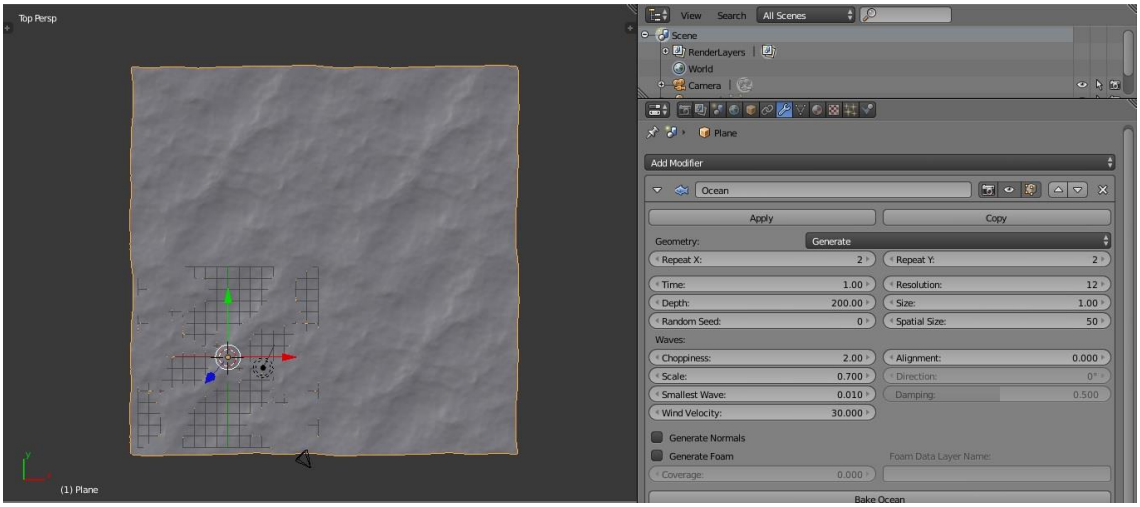
8.2. Izrada simulacije za animaciju mora

Na scenu je dodan *Plane mesh* i odmah je primjenjen modifikator oceana. Jednostavno je selektirana ploha i u izborniku *Modifiers* odabran *Ocean*. Nakon toga je skalirana površinu mora na, otprilike, 17 metara po X i Y dimenziji.



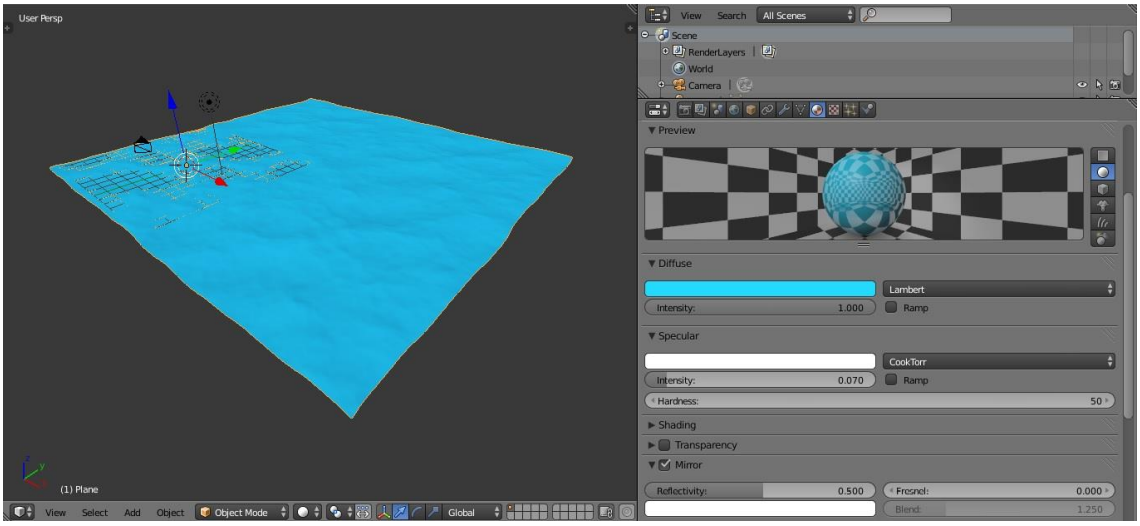
Slika 21. Modifikator oceana

Nakon toga su podešeni parametri u modifikatoru oceana gdje su bitnije postavke bile automatsko generiranje geometrije (*Geometry → Generate*), ponavljanje geometrije po X i Y osi 2 puta (ovime je dobivena veća površina mora; otprilike 34 m * 34 m), rezolucija je postavljena na vrijednost 12, *choppiness* je postavljen na vrijednost 2 kako bi se dobila veća oštrina valova, i još se dodatno s opcijom skaliranja istaknuća utjecalo na vidljivost valova. Druge opcije su ostale nepromijenjene odnosno ostavljene su početne postavke u verziji Blendera u kojoj se radilo.



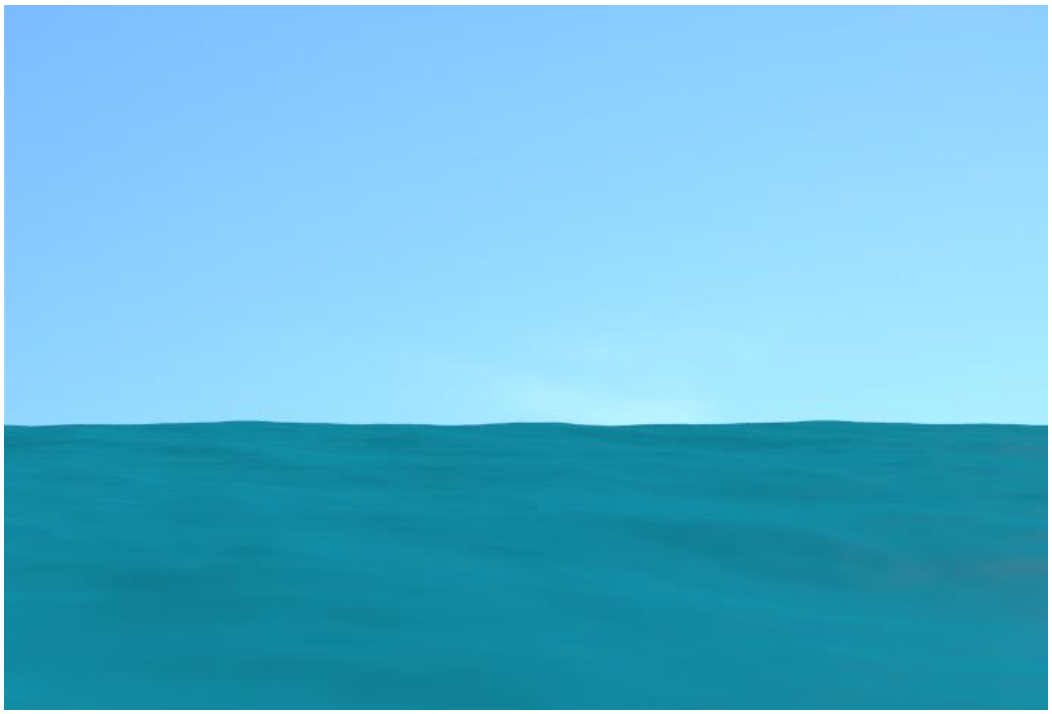
Slika 22. Postavke modifikatora *Ocean*

Slijedeće je bilo potrebno postaviti materijal na objekt mora. U izborniku *Material* objektu je izabrana nijansa plave boje u *Diffuse* okviru i intenzitet je stavljen na 1, u *Specular* okviru namješten je intenzitet svjetline na vrijednost 0.07 i još je uključena opcija ogledala (*Mirror*) u kojoj je vrijednost refleksije 0.5.



Slika 23. Materijal mora

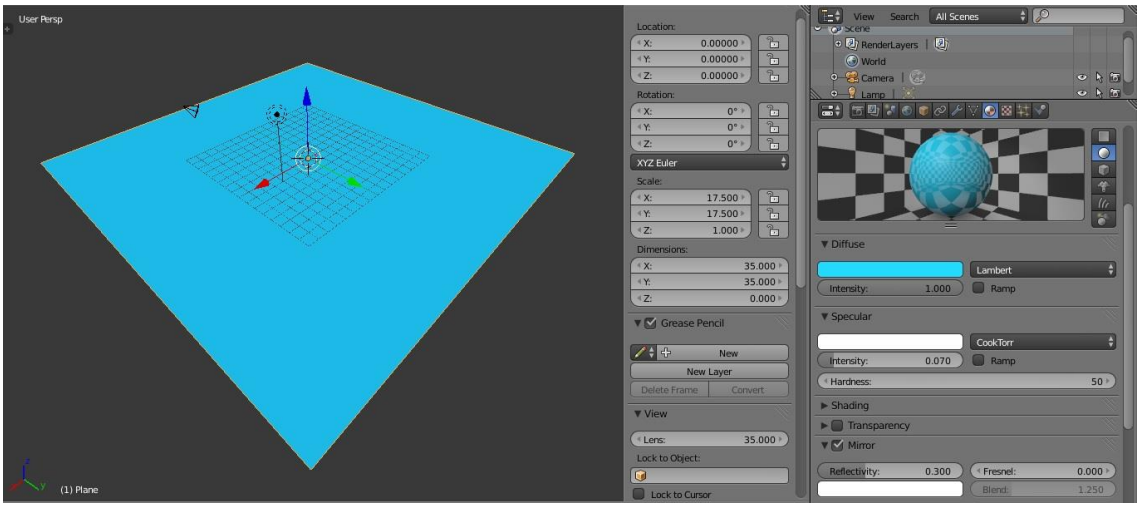
Na kraju je na scenu dodan izvor svijetlosti vrste sunce. I kao pozadinska slika postavljena je fotografija plavog neba koja ima utjecaj na refleksiju u objektu oceana. Fotografija je preuzeta s *web* stranice *Pixabay*. Na sceni se još nalaze kamera i “prazni” objekt koji služe za kadriranje scene. Animacija se sastoji od sveukupno 501 slike s *frame rate*-om od 25 slika po sekundi. Ukupno trajanje animacije je 20 sekundi. Cijeli proces izrade je trajao oko 5 minuta.



Slika 24. Konačni rezultat

8.3. Izrada "varalice" za animaciju mora

Za izradu “varalice” koristio se objekt plohe kojem su određene dimenzije na 35 m po X i 35 m po Y vrijednosti. Ploha predstavlja površinu mora. Nakon toga, u izborniku materijala, na objekt je primjenjen materijal nijanse plave boje u *Diffuse* kartici, intenzitet mu je namješten na vrijednost 1. Odbljesak svjetla u *Specular* kartici je postavljen na vrijednost 0.07. Uključeno je svojstvo ogledala i ondje je podešena vrijednost refleksije na 0.3.



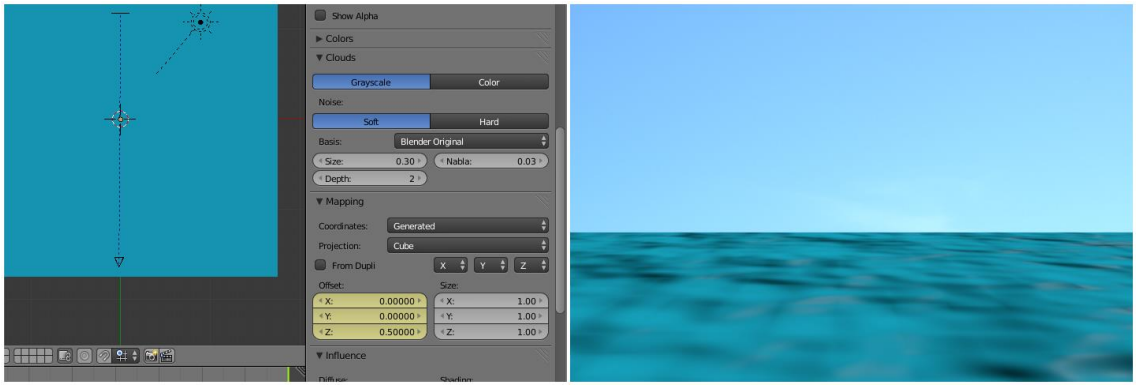
Slika 25. Ploha koja predstavlja more

Nakon toga primijenjena je proceduralna tekstura “oblaka” u izborniku *Texture*. U kartici *Clouds* određena je vrijednost veličine buke ili šuma na 0.3. Za ostale postavke preuzete su početne vrijednosti.

Osim objekta koji predstavlja more, na sceni su još bili jedan objekt izvora svijetlosti vrste “sunce” i kamera i prazni objekt koji su poslužili za namještanje kadra. U pozadinu scene postavljena je ista fotografija plavog neba koje utječe na refleksiju objekta mora (istu fotografiju smo koristili i u simulaciji).

Animiranje se kao i u svim primjerima dosada izvršavalo u 25 slika pod sekundi sa sveukupno 501 slikom. Varalica se animirala na način da se na nultom *frame*-u postavio *keyframe* dok nam je kursor miša bio postavljen na *Offset* vri-

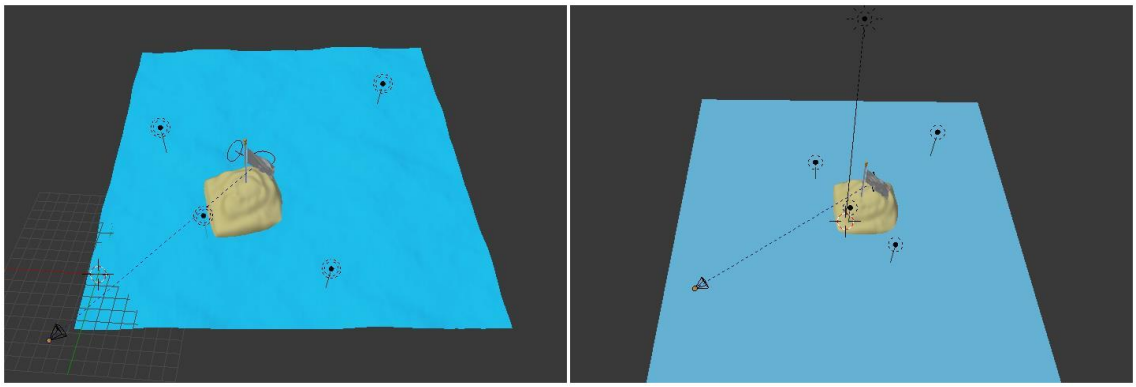
jednosti Z u kartici *Mapping* koja se nalazi unutar izbornika Texture. Nakon toga na vremenskoj traci odabran je *frame* 500 i ponovno ponovljen postupak u *Mapping* kartici s time da je prije toga promijenjena vrijednost na 0.5 i nakon toga dodan završni *keyframe*. Ovom promjenom vrijednosti smo utjecali na fino podešavanje teksture i tako smo kroz vrijeme animacije dobili efekt pomicanja po površini. Za izradu ove animacije bilo je potrebno oko 5 minuta.



Slika 26. Postavke za teksturu, animaciju i konačni rezultat

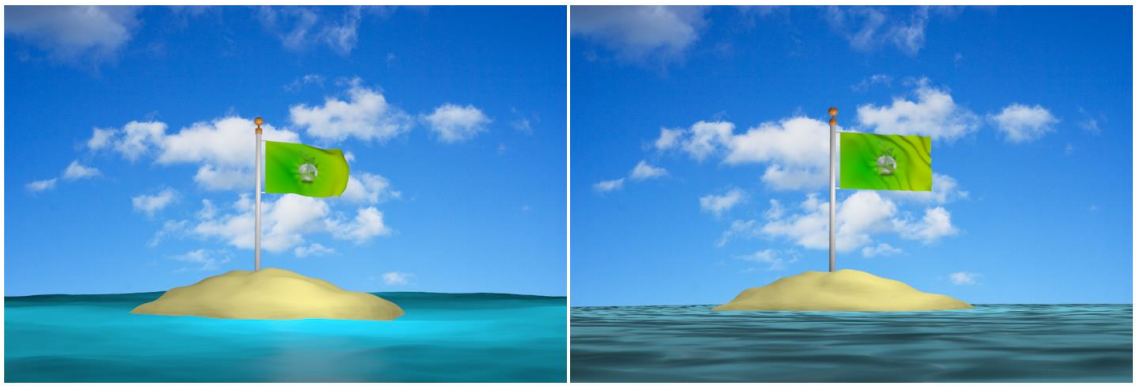
8.4. Izrada kombiniranih scena za simulaciju i "varalicu"

U kombiniranim scenama korišteni su svi "glavni" objekti koje su kreirani u početku za pojedinačne scene. Jedino su neke stavke mijenjane kako bi se postiglo da dvije scene što više "nalikuju" jedna na drugu. To se odnosi prvenstveno na osvijetljenje scena – korišten je isti broj "točkastog" izvora svijetlosti, ali neke opcije kao što je intenzitet svijetlosti i pozicioniranje izvora na sceni se razlikuju. U sceni "varalice" postoji jedan dodatni izvor svijetlosti vrste "sunce". Isto tako postoje razlike i kod "glavnih" objekata u kombiniranoj i pojedinačnoj sceni s manjim preinakama, recimo, vrijednosti refleksije mora, buke ili šuma proceduralne teksture, itd. Znači, ovo je učinjeno da bi scene bile sličnije.



Slika 27. Izrada kombiniranih scena

U obje scene je dodan jedan novi objekt koji nije mijenjan. To je objekt koji predstavlja pješčani otok. Kreiran je putem *Plane mesh*-a gdje su putem alata selekcije i alata *Proportional Editing* u *Edit* modu kreirane nasumične neravnine (uz pomoć osnovne *Grab* naredbe) kako bi se dobio što prirodniji prikaz otoka. Otok je veličine, otprilike 9.5 metara po X i Y osi i 3 metra po Z osi. U svakoj od scena je rotiran i postavljen kako bi se bolje uklopio u kadar.



Slika 28. Konačni rezultati

9. TESTOVI I ANALIZA

Računalno na kojem su se provodili testovi je prijenosno računalo opremljeno za uredsko poslovanje koje služi za obavljanje većine standardnih, svakodnevnih aktivnosti i rješavanja samostalnih studentskih zadataka.

Specifikacije računala:
Intel Pentium B950, 2.10 GHz (procesor)
4096 MB RAM (radna memorija)
SATA 500 GB HDD (tvrdi disk)
Windows 7 Home Premium 64 bit (operativni sustav)
Intel HD Graphics (grafički sustav)

Kako je autor ovog rada u vrijeme pisanja rada bio onemogućen koristiti jače, produkcijsko računalo tako se moralo paziti na kompleksnost scena koje će se testirati. Kako bi se testovi proveli u prihvatljivom vremenu tako je pažnja posvećena da se na sceni nalazi relativno manji broj objekata koji i sami nisu sačinjeni od većeg broja poligona i detalja kao što se visoko kvalitetne teksture. Isto su se tako pokušale optimizirati postavke preciznosti izvođenja fizikalnih simulacija, kao i nepretjerivanje s naprednijim osvijetljenjem, refleksijama, česticama, dinamičnim sjenama i sličnim efektima.

Za renderiranje svih scena koristio se Blenderov interni render (*Blender Render*). U svim scenama korištene su iste postavke za izlaznu (*output*) video datoteku. Postavke su slijedeće:

- dimenzije rezolucije: 720 px * 486 px
- anti-aliasing: 8
- broj slika (*frame*-ova): 501
- broj slika po sekundi: 25
- video datotečni format: MPEG
- RGB sustav boja
- ostale postavke u sustavu rendera nisu mijenjane, odnosno preuzete su početne Blenderove vrijednosti

Svako renderiranje je praćeno i putem Windowsovog sustava za praćenje računalnih resursa (*Resource Monitor*). Praćene su komponente centralnog procesora (*CPU*) i radne memorije (*RAM*) u kojoj je praćena *private* sekcija memorije. Moglo se primijetiti da je u svim testovima centralni procesor radio punim postotkom iskoristivosti (100%, u čemu je, u prosjeku, 83.1% zauzimao Blender – vrijednosti između 78% i 89%). Vrijednosti su tokom renderiranja konstantno varirale.
Kod nadzora radne memorije uočeni su podjednaki rezultati za simulaciju i varalicu, ali može se reći da je za simulaciju ipak u svim testovima iskorištavan veći kapacietet memorije – posebno u scenama koje su sadržavale veći broj objekata i osvijetljenja. Vrijednosti su tokom renderiranja konstantno varirale.
Vrijednosti *CPU*-a i *RAM*-a su opažane otprilike jedne minute nakon početka renderiranja. Njih treba uzeti s rezervom jer su vrijednosti tokom cijelog procesa oscilirale vjerojatno pod utjecajem i drugih procesa potrebnih za rad samog operativnog sustava. Ovi rezultati služe informativno i ne predstavljaju mjerene vrijednosti. Ovo je samo potvrdilo teoriju zahtjevnosti sustava renderiranja za procesorskim i memorijskim resursima. Kako testno računalo nije bilo opremljeno hardverskom grafičkom procesnom jedinicom, nisu se mogli provesti testovi renderiranja preko Blenderovog *GPU* sustava renderiranja.

Bitno je napomenuti da su scene izrađivane tako da budu što sličnije jedna drugoj i time sve postavke nisu iste za simulaciju i varalicu. To je najviše vidljivo u osvijetljenosti scena. Utjecaj na brzinu renderiranja ima i kompleksnost kadra tj. količina objekata koji se vide, strane koje su vidljive na sceni, detaljnost tekstura, itd.

Izvršeno je deset animacija koje su uspoređene kroz pet testova. Mjereno vrijeme renderiranja pojedine scene i izračunato je prosječno renderiranje jedne slike. Prvo je testirana animacija tkanine, zatim fluida i potom kombinirana scena tkanine i fluida. U četvrtom i petom testu analizirane su ponovno scene fluida i kombinirane scene ali s dodanom tehnikom *Ray Tracing* prilikom renderiranja.

- *Ray Tracing* označava tehniku renderiranja scene pomoću zraka svijetlosti koja simulira neke naprednije vizualne efekte kako dolazi u kontakt s raznim okolnim objektima na sceni (refleksija, lom svijetlosti, raspršivanje, i dr.). Koristi se za dobivanje realističnih scena i time zahtjeva dosta računalnih resursa. ***[24]***

9.1. Test 1. Usporedba scene zastave

Simulacija – ZASTAVA

Renderiranje je trajalo sveukupno 10 minuta i 25 sekundi. Jedna slika je u prosjeku renderirana 1.25 sekundi. Prosječna iskorištenost *CPU*-a je 85%, a *RAM*-a 197 KB.

Varalica – ZASTAVA

Sveukupno renderiranje je 9 minuta i 51 sekunda. Prosječno vrijeme koje je bilo potrebno da se jedna slika izrenderira je 1.182 sekunde. Prosječna iskorištenost *CPU*-a je 82%, a *RAM*-a 170 KB.

Tablica 1. Usporedba renderiranja scene sa zastavom

Renderiranje	Simulacija	Varalica
Sveukupno	10 min i 25 sek	9 min i 51 sek
Prosjek jedne slike	1.25 sek	1.182 sek

- U ovom testu je dokazana tvrdnja da je varalica bolji izbor ako želimo uštedjeti na vremenu renderiranja.

9.2. Test 2. Usporedba scene mora

Simulacija – MORE

Renderiranje je proteklo za 17 minuta i 23 sekunde. U prosjeku se jedna slika renderirala 2.086 sekundi. Centralni procesor, u prosjeku, je radio na 89%, a *RAM* na 230 KB.

Varalica – MORE

Za 11 minuta i 15 sekundi završeno je renderiranje scene. Prosječno vrijeme renderiranja jedne slike je 1.35 sekundi. Opažanje prosječnog rada centralnog procesora je 84%, radne memorije 175 KB.

Tablica 2. Usporedba renderiranja scene mora

Renderiranje	Simulacija	Varalica
Sveukupno	17 min i 23 sek	11 min i 15 sek
Prosjek jedne slike	2.086 sek	1.35 sek

- U testu renderiranja mora dokazano je da se varalicom štedi procesorsko vrijeme.

9.3. Test 3. Usporedba scene zastave i mora (kombinirano)

U slijedećim testovima korištene su kompleksnije scene u kojima su kombinirane prijašnje jednostavnije scene. Važno je za napomenuti da u ovim scenema postoji novi objekt (poligonalni *mesh*) koji predstavlja pješčani otok i isti je korišten u svim kombiniranim scenama. Osim otoka dodano je još nekoliko izvora svijetlosti. Mijenjane neke postavke izvora svijetlosti za pojedine scene kako bi se na što bolji način osvijetlila scena. Postavke koje su mijenjane su *energy* (količina svijetla koju izvor emitira), *falloff distance* (udaljenost na kojoj je vrijednost svijetlosti na pola od originalnog intenziteta), pozicije pojedinih izvora i kut upada svijetla.

Simulacija – KOMBINIRANO

Vrijeme koje je bilo potrebno za renderiranje ove scene je 25 minuta i 24 sekunde. Za jednu sliku, u prosjeku, je bilo potrebno 3.048 sekundi. Centralni procesor je u prosjeku radio na 78% od ukupne iskorištenosti, radna memorija na 314 KB.

Varalica – KOMBINIRANO

Renderiranje je završeno za 18 minuta i 12 sekundi. Prosječno vrijeme renderiranja jedne slike je 2.184 sekunde. Prosječna skoristivost *CPU*-a je bila na 80%, a radna memorija je pokazivala vrijednost 155 KB.

Tablica 3. Usporedba renderiranja kombinirane scene

Renderiranje	Simulacija	Varalica
Sveukupno	25 min i 24 sek	18 min i 12 sek
Prosjek jedne slike	3.048 sek	2.184 sek

- I u ovom testu smo dokazali da varalicom štedimo računalne resurse.

9.4. Test 4. Usporedba scene mora s RAY TRACING tehnikom

Simulacija – MORE s *RAY TRACING*-om

Za renderiranje ovakve scene bilo je potrebno 54 minute i 19 sekundi. Jedna slika u prosjeku se renderirala 6.518 sekundi. Centralni procesor je u prosjeku radio na 85%, a *RAM* je koristio 285 KB kapaciteta.

Varalica – MORE s *RAY TRACING*-om

Ovaj test je prošao pomalo neočekivano. Rezultati renderiranja su pokazali vrijeme od 75 minuta i 20 sekundi. To znači da je u prosjeku za jednu sliku trebalo 9.04 sekunde. Prosječan rad *CPU*-a je pokazivao 87%, *RAM* 75 KB. Ovaj test pokazuje da je ova tehnika varalice ipak zahtjevnija po računalne resurse i da treba pripaziti kada se radi scena s *Ray Tracing* tehnikom koja je inače poznata

kao tehnika koja zahtjeva mnogo računalne snage.
Autor ovog rada pretpostavlja da je ovdje došlo do duljeg renderiranje scene, s obzirom na sličnu scenu u simulaciji, zbog drugačijeg pristupa animiranju varalicom koji nije niti zamišljen kao dobra opcija animiranja u kombinaciji s *Ray Tracing* tehnikom. Vjerojatno se sustav renderiranja ovdje susreo s *Ray Trace*-anom teksturom koja je u pokretu što usporilo čitav proces.
Naknadnim testiranjem je utvrđeno da vrijeme renderiranja znatno ovisi o veličini površine koju zauzima objekt mora u kadru.

Tablica 4. Usporedba renderiranja scene mora s *RAY TRACING*-om

Renderiranje	Simulacija	Varalica
Sveukupno	54 min i 19 sek	75 min i 20 sek
Prosjek jedne slike	6.518 sek	9.04 sek

- Ovaj test nam pokazuje da varalica nije uvijek brže rješenje i da treba biti oprezan s korištenjem *Ray Tracing* tehnike.

9.5. Test 5. Usporedba kombinirane scene s RAY TRACING tehnikom

Simulacija – KOMBINIRANO s *RAY TRACING*-om
Renderiranje je trajalo 64 minute. Prosječno vrijeme renderiranja jedne slike je 7.68 sekundi. Prosječan rad procesora je bio na 80%, a radna memorija je pokazivala 340 KB.

Varalica – KOMBINIRANO s *RAY TRACING*-om
Ovaj test je pokazao opet iznenađenje (upravo zbog testa Varalica – MORE s *RAY TRACING*-om). Ovdje smo dobili rezultat kako je varalica opet bolji izbor ako želimo brže renderiranu scenu.
Autor ovog rada pretpostavlja da se to dogodilo zbog manjih preinaka u postavkama osvijetljenja i drugačijeg kadriranja scene gdje površina mora ne

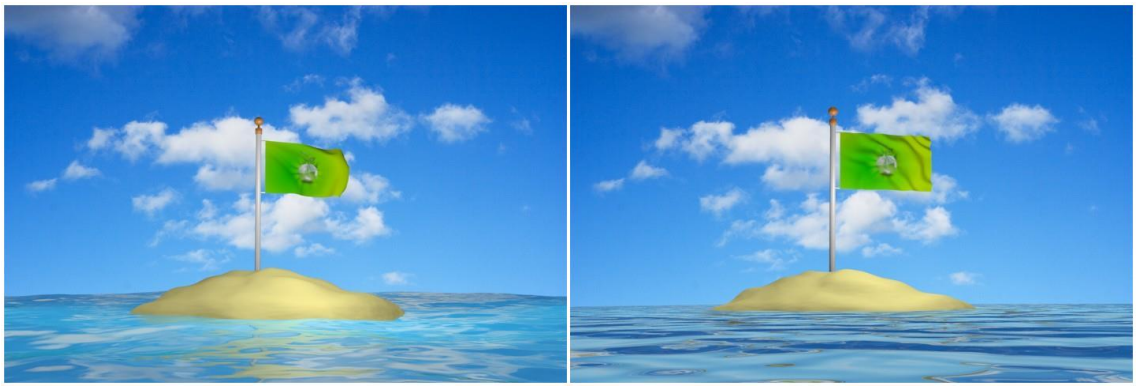
zauzima veliki udio na sceni. Naknadnim testiranjem renderiranja jedne slike je utvrđeno da veličina površine mora koja se pojavljuje u sceni znatno utječe na vrijeme renderiranja.

Sveukupno vrijeme renderiranja je 33 minute i 27 sekundi. Za jednu sliku u prosjeku je utrošeno 4.014 sekunde. *CPU* je u prosjeku pokazivao 81%, a *RAM* vrijednost od 203 KB.

Tablica 5. Usporedba renderiranja kombinirane scene s *RAY TRACING*-om

Renderiranje	Simulacija	Varalica
Sveukupno	64 min	33 min i 27 sek
Prosjek jedne slike	7.68 sek	4.014 sek

- Ovaj test pokazuje da je varalica bolji izbor ako želimo brže renderiranje.



Slika 29. Usporedba kombiniranih scena s *Ray Tracing* tehnikom

10. ZAKLJUČAK

Varalice nam u nekim slučajevima daju prednost nad simulacijama gdje se na brz i fleksibilan način može jednostavnije doći do rješenja. Kako je vidljivo iz rezultata testova, varalicom možemo uštedjeti na vremenu renderiranja scene. Bitno je već u početku imati viziju konačnog proizvoda i na taj način organizirati cjelokupan proces rada s obzirom na kvalitetu vizualne izvedbe animacije i vremenskih rokova.

Pri odabiru varalice potrebno je voditi dosta brige o načinu njene izrade i koliko će nam vjerno animirati neku pojavu ili objekt. Kako je u *Testu 5 - Varalica – KOMBINIRANO s RAY TRACING-om* došlo do neočekivanog rezultata gdje se pokazalo da je varalica lošije rješenje nego simulacija, tako bi već u početku trebalo imati saznanja o pojedinoj varalici i kakve rezultate možemo očekivati.

Ako naš projekt na kojem radimo ne zahtijeva visoko precizne simulirane fizikalne animacije, varalica će nam vjerojatno poslužiti kao alat uštede procesorskog vremena. Ovakvi projekti mogu se odnosti prvenstveno na računalne igre gdje je potrebno renderirati virtualnu okolinu u realnom vremenu, interaktivnu 3D grafiku apstraktnog tipa i filmove "veselijeg i zabavnijeg" sadržaja gdje nije toliko bitno zadovoljiti imitiranje pojava iz stvarnosti. U računalnim igrama, igrač često puta nema vremena stajati i promatrati da li se neka pojava fizikalno precizno animirala i dovoljno ga je stimulirati s jednostavnijom izvedbom scene koja će u potpunosti zadovoljiti potrebu za vjerodostojnošću prikaza u tom trenutku. Slično bi mogli reći i za ostale oblike prikazivanja grafike, ukoliko gledateljev glavni fokus nije na sceni koja ima sporednu ulogu, tada se na njoj može koristiti varalica, odnosno načini animiranja kako bi se takva scena mogla izvoditi a da ne zauzima mnogo računalnih resursa.

Simulacija nam daje preciznije i realističnije animiranje u scenama i kod nekih animacija je nezamjenjiva, ali sve to na trošak računalnih resursa i vremena renderiranja. Ako su nam potrebni precizni prikazi animacije u svijetu 3D grafike

onda bi bilo dobro imati dovoljno vremena za izradu takvih simuliranih projekata kao i dovoljno računalne snage kako bi cijeli proces prošao u granicama prihvatljivosti.

S obzirom na konstantni razvoj grafičke računalne tehnologije, autor ovoga rada vjeruje da se u bliskoj budućnosti neće smanjiti potreba za izradom varalica odnosno načina kojim bi se uštedjeli računalni resursi. Mišljenja je da kako se razvija računalna tehnologija, tako se javlja i veća potreba za kompleksnijim i realističnijim prikazom 3D animacije pogotovo u interaktivnoj grafici koja se renderira u realnom vremenu. Zbog toga je bitno dobro poznavati tehnologiju, što nam ona omogućava i koja su njena ograničenja. Tek na taj način možemo maksimalno optimizirati omjer kvalitete prikaza računalne grafike i brzine njenog izvođenja. Kreativnost je u nama ... :)

11. LITERATURA:

1. *Kelly L. Murdock* (2009). *3ds Max® 2010 Bible, Wiley Publishing, Indianapolis*
2. *James Cronister* (2011). *Blender Basics - Classroom Tutorial Book - 4th Edition*, dostupno na: http://www.cdschools.org/site/default.aspx?PageID=1, 30. srpanj, 2014.
3. *** http://wiki.blender.org/ - *Blender Wiki, Introduction to Physics Simulation*, 30. srpanj, 2014.
4. *** http://wiki.blender.org/ - *Blender Wiki, Cloth Simulation*, 30. srpanj, 2014.
5. *** http://wiki.blender.org/ - *Blender Wiki, Modifiers*, 31. srpanj, 2014.
6. *** http://wiki.blender.org/ - *Blender Wiki, Force Fields*, 01. kolovoz, 2014.
7. *** http://wiki.blender.org/ - *Blender Wiki, Collisions*, 01. kolovoz, 2014.
8. *** http://wiki.blender.org/ - *Blender Wiki, Dynamic Paint*, 01. kolovoz, 2014.
9. *** http://wiki.blender.org/ - *Blender Wiki, Soft Bodies*, 01. kolovoz, 2014.
10. *** http://wiki.blender.org/ - *Blender Wiki, Fluid Simulation*, 01. kolovoz, 2014.
11. *** http://wiki.blender.org/ - *Blender Wiki, Render Baking*, 02. kolovoz, 2014.
12. *** http://wiki.blender.org/ - *Blender Wiki, Smoke Simulation*, 01. kolovoz, 2014.
13. *** http://wiki.blender.org/ - *Blender Wiki, Rigid Body Simulation*, 01. kolovoz, 2014.
14. *** http://en.wikipedia.org/wiki/Main_Page - *Wikipedia, 3D modeling*, 04. kolovoz, 2014.
15. *** http://en.wikipedia.org/wiki/Main_Page - *Wikipedia, Game physics*, 05. kolovoz, 2014.
16. *** http://en.wikipedia.org/wiki/Main_Page - *Wikipedia, Physics engine*, 05. kolovoz, 2014.
17. *** http://en.wikipedia.org/wiki/Main_Page - *Wikipedia, Bullet (software)*, 07. kolovoz, 2014.

18. *** http://en.wikipedia.org/wiki/Main_Page - *Wikipedia, PhysX*, 07. kolovoz, 2014.
19. *** http://wiki.blender.org/ - *Blender Wiki, What is Blender?*, 08. kolovoz, 2014.
20. *** http://en.wikipedia.org/wiki/Main_Page - *Wikipedia, COLLADA*, 09. kolovoz, 2014.
21. *** http://wiki.blender.org/ - *Blender Wiki, Displace Modifier*, 16. kolovoz, 2014.
22. *** http://en.wikipedia.org/wiki/Main_Page - *Wikipedia, Fluid simulation*, 16. kolovoz, 2014.
23. *** http://wiki.blender.org/ - *Blender Wiki, Fluid Simulation*, 16. kolovoz, 2014.
24. *** http://en.wikipedia.org/wiki/Main_Page - *Wikipedia, Ray tracing (graphics)*, 20. kolovoz, 2014.

12. SLIKE

- ilustracija korištena u scenama s zastavom: http://pixabay.com/en/earth-globe-plant-eco-green-158806/ - 14. kolovoz, 2014.
- pozadinska slika korištena u scenama simualcija i varalica zastavi i kombiniranih scena (odnosi se i na dio s *Ray Tracing* tehnikom): http://pixabay.com/en/sky-blue-cloud-clouds-15198/ - 14. kolovoz, 2014.
- pozadinska slika korištena u scenama simulacija i varalica mora (odnosi se i na dio s *Ray Tracing* tehnikom): http://pixabay.com/en/blue-sky-clouds-290314/ - 22. kolovoz, 2014.