

Pertti Raami

Nestesimulaatio 3D-artistin käytössä

Vertailussa Blender ja Autodesk Maya

Metropolia Ammattikorkeakoulu

Medianomi (AMK)

Viestintä

Opinnäytetyö

23.2.2016



Tekijä(t) Otsikko	Pertti Raami Nestesimulaatio 3D-artistin käytössä - Vertailussa Blender ja Autodesk Maya
Sivumäärä Aika	33 sivua + 1 liitettä 23.2.2016
Tutkinto	Medianomi (AMK)
Koulutusohjelma	Viestintä
Suuntautumisvaihtoehto	3D-animointi ja -visualisointi
Ohjaaja(t)	Lehtori Jaro Lehtonen

Nestesimulaatiolla, eli numeerisella virtausmallinnuksella tarkoitetaan tietokoneella generoitua virtausmallinnusta. Monissa 3D-ohjelmistoissa on mukana nestesimulaattori ja markkinoilla on useita nestesimulaatioihin keskittyviä ohjelmistoja. Näillä simulaattoreilla voi yleisesti simuloida nesteiden, kaasujen ja kaasun kaltaisten aineiden liikkeitä, joita muuten olisi vaikea mallintaa tai animoida.

Tämän opinnäytetyön tavoitteena oli tutkia neljää simulaatioaikaan vaikuttavaa tekijää, jotka ovat: korkea resoluutio, alhainen viskositeetti, nopea vauhti ja suuri määrittelyalue. Työssä on tutkittu miten Blender-ohjelmisto suoriutuu edeltä mainituista ongelmista verrattuna kaupallisen Autodesk Maya 2014-ohjelmiston Bifrost-simulaattoriin. Vertailu tehtiin toistamalla testit mahdollisimman samankaltaisena Blenderillä sekä Bifrostilla. Lisäksi opinnäytetyössä käydään läpi Blenderin nestesimulaation tarvittavat elementit sekä miten sillä voi tehdä yksinkertaisen nestesimulaation. Läpikäytävät elementit ovat yleisiä kaikissa nestesimulaattoreissa, joten ne ovat myös sovellettavissa Mayan Bifrostissa ja muissa 3D-ohjelmistoissa.

Opinnäytetyön testien puitteissa Blenderin simulaatiot eivät poikenneet simulaatioajoissa paljoa Mayan Bifrostista. Tutkituista ongelmista resoluution suurentaminen vaikutti odotetusti eniten simulaatioaikoihin molemmissa ohjelmistoissa. Maya 2015-version Bifrostissa puuttui mahdollisuus viskositeetin muuttamiseen. Bifrostissa myös osassa testeistä simulaatiot eivät pysyneet eheänä, vaan tulokseksi saadut polygoniverkot rikkoutuivat ja olivat luonnottomia. Testisimulaatioiden perusteella Blender päihittäisi Mayan Bifrostin. Testeissä paljastui, että Maya 2015-version Bifrost-simulaattori on julkaistu keskeneräisenä.

Opinnäytetyössä tehdyt testit ja niiden tulokset voivat auttaa nestesimulaatioista kiinnostuneita tutustumaan siihen, miten simulaatioita tehdään 3D-ohjelmistoilla, sekä mitkä asiat vaikuttavat simulaatioiden visuaalisiin tuloksiin ja simulointiaikoihin. Vaikka tietokoneiden ja ohjelmistojen kehitys vanhentavat testien tulokset nopeasti, niin samat muuttujat vaikuttavat simulaatioaikoihin jatkossakin. On kuitenkin syytä muistaa, että Bifrostissa on paljon ominaisuuksia, jotka ovat testien ulkopuolella ja niitä ei ole Blenderissä, sekä se, että Bifrostin ominaisuuksia on parannettu huomattavasti Mayan 2016-versiossa.

Avainsanat	3D, nestesimulaatio, Blender, Maya, Bifrost

Author(s) Title	Pertti Raami Fluid simulator in use of a 3D artist – Comparing Blender and Autodesk Maya
Number of Pages Date	33 pages + 1 appendices 23 February 2016
Degree	Bachelor of Culture and Arts
Degree Programme	Media
Specialisation option	3D Animation and Visualization
Instructor(s)	Jaro Lehtonen, Senior Lecturer
<p>Fluid simulation in 3D software refers to computational fluid dynamics, which is a way of calculating the flow of streams with computers. Fluid simulators in 3D software provide useful tools for elements, that are manually hard to animate or model, like fluids, gasses and matters which are similar to gasses, for example smoke and dust. Several 3D software have build in fluid simulator but there are also software, which are specially developed for fluid simulations such as Real Flow.</p> <p>The main aim of this thesis was to study how much different Blender's fluid simulator is from Autodesk Maya's Bifrost fluid simulator. Blender is free open source software whereas Autodesk Maya is commercial software that costs approximately 4000 euros. In this study there were four different factors chosen to compare fluid simulators. Tested factors were: high resolution, low viscosity, high speed and large domain. All of those should have considerable effect to increasing simulation time.</p> <p>My second aim was to go through the main elements of Blender's fluid simulation, which are essential to create a simple fluid simulation. The same elements can be found from other fluid simulators.</p> <p>The tests that were carried out did not reveal any significant differences when the simulation times between Blender and Maya were compared. As expected, increasing resolution was the most significant factor in both simulators. Maya 2015-version's Bifrost simulator lacks an attribute to change viscosity, which means that it can simulate only water-like liquids. Bifrost had difficulties in some of the test simulations giving results of broken and unnatural polygon meshes. According to the tests, which were done in the thesis, better results were gained with Blender's fluid simulator. Maya 2015-version's Bifrost simulator had problems with simulations with high resolution and fast moving liquids. Bifrost also lacked some important features, which were needed.</p>	

Keywords	3D, fluid simulation, Blender, Maya, Bifrost

Sisällys

1	Johdanto	1
1.1	Nestesimulaattoreista yleisesti	2
1.2	Nestesimulaatio 3D-animaatiossa ja –visualisoinnissa	3
1.3	Nesteen pinta eritekniikoilla	4
1.3.1	Normal- ja bump-kartta	4
1.3.2	<i>Displacement</i> -kartta ja <i>ocean</i> -simulaattori	5
1.3.3	Nestesimulaatio	7
2	Tutkimusongelma ja menetelmät	8
2.1	Opinnäytetyössä käytettävät nestesimulaattorit	9
2.1.1	Maya Bifrost	9
2.1.2	Blender	9
2.2	Nestesimulaation työvaiheet	10
2.3	Testiskene	12
2.4	Resoluution tarkentaminen	14
2.5	Alhainen viskositeetti	15
2.6	Nopeus	15
2.7	Suuri määrittelyalue	15
3	Tulokset	16
3.1	Resoluution tarkentaminen	16



3.2	Viskositeetin laskeminen	17
3.3	Nopeus	19
3.4	Suuri määrittelyalue	20
4	Tulosten tulkinta	22
4.1	Resoluution tarkentaminen	23
4.2	Viskositeetin laskeminen	27
4.3	Nopeus	28
4.4	Suuri määrittelyalue	29
5	Loppupäätelmät	31
	Lähteet	34
	Liite 1. Termistöä	1



1 Johdanto

Elokuissa voi nähdä miten vesimassat voivat muodostaa ihmishahmon tai vyöryä kymmenien metrien korkuisena seinämänä kaupungin kaduille, tai mainoksissa neste voi lentää spiraalin muotoisena norona pullon ympärillä. On selvää, että edellä mainitut tehosteet ovat liian kalliita tai fyysisesti mahdottomia toteuttaa ilman tietokoneilla tehtyjä erikoistehosteita. Nesteiden massat ja liikkeet ovat niiden monimutkaisuuden takia käytännöllisesti mahdotonta animoida manuaalisesti tietokoneella. Tämän takia niiden tekemiseen käytetään numeerista virtausmallinnusta, eli nestesimulaatioita. (Hodgins & O'Brien 1995.)



Kuva 1 Nestesimulaatioilla voi tehdä efektejä, joita todellisuudessa on lähes mahdotonta toteuttaa. Kuvakaappaus elokuvasta The Day After Tomorrow vuodelta 2004.

Nestesimulaatiolla tarkoitetaan tietokoneella generoitua virtausmallinnusta (Kuzmin 2010; O'Brien & Hodgins 1995). Useassa 3D-ohjelmistoissa on mukana nestesimulaattori joka simuloi nesteen käyttäytymistä (Seymour 2011). Simulaattorit käyttävät tai pohjautuvat osin 1800-luvulta peräisin oleviin Eulerin sekä Navier-Stokesin yhtälöihin, jotka kuvaavat nesteiden käyttäytymistä (Darrigol 2002). Niillä voi yleisesti simuloida nesteiden, kaasujen ja kaasun kaltaisten aineiden, kuten savun, liikkeitä, joita manuaalisesti animaatiolla olisi vaikea toteuttaa uskottavasti (Chrisman 2008).

3D-artisteille tarkoitettuja nestesimulaattoreita on ollut tarjolla ainakin yli 15 vuotta. Esimerkiksi vuonna 2000 julkaistussa Maya 3.0:ssa on jo ollut mukana nestesimulaattori. (Toxik.sk 2015.) Itse en kuitenkaan löytänyt Theseuksen tietokannasta yhtään 3D-artistille suunnattua tutkimusta, jossa käsiteltäisiin nestesimulaattoreita ja niiden toimintaa. Tämän takia haluan käydä opinnäytetyössäni läpi miten rakennetaan toimiva nestesimulaatio Blender-ohjelmistolla. Lisäksi tutkin muutamia simulaattoreiden perusominaisuuksia joiden asetuksilla voidaan vaikuttaa simulaation ulkonäköön ja laskentaan kuluvaan aikaan. Tutkimuksen teen tekemällä sarjan testisimulaatioita Blenderillä, joilla toivon saavani tuloksia miten tutkittavat ominaisuudet vaikuttavat simulaatioihin. Tutkin myös miten kaupallisen Maya-ohjelmiston Bifrost-nestesimulaattori eroaa simulaatioissa verrattuna vapaan lähdekoodin Blenderiin toistamalla testisimulaatiot mahdollisimman samankaltaisina Bifrostissa.

Opinnäytetyössäni tulen käyttämään 3D-ohjelmistojen työkaluista ja termeistä englanninkielistä nimeä, jos niistä ei ole olemassa yleisesti ammattipuheessa käytettävää suomenkielistä versiota. Liitteessä 1 on selitetty tärkeimmät termit ja niiden suomen- ja englanninkieliset muodot.

1.1 Nestesimulaattoreista yleisesti

Tietokoneilla generoitujen virtaussimulaatioiden historia alkaa 1950-luvulla (Chung 2010, 3). Aluksi simulaatiot keskittyivät asevoimien tarkoitukseen laskea esimerkiksi räjähdysten shokkiaaltoja sekä yliaänikoneiden ilmavirtauksia. Ensimmäiset tietokoneella lasketut realistiset kaksiulotteiset virtaussimulaatiot tehtiin 1960-luvun loppupuolella, ja kolmiulotteiset simulaatiot saavutettiin 1980-luvulla. (Zikanov 2010, 4 - 5.)



Kuva 2 Kolmiulotteinen virtausmallinnus polkupyöräilijästä tuulitunnelissa. Kuva: feltracing.com

Elokuvat alkoivat käyttää tietokoneella tehtyjä erikoistehosteita 1970-luvun loppupuolella (Yaeger 2002), ja vuonna 1989 nähtiin ensimmäinen tietokoneella tehty nesteerikoistehoste James Cameronin Syvyys-elokuvassa (The Abyss). Reaalimaailman nestesimulaatiot ovat olleet liian monimutkaisia ja raskaita suoritettaviksi 1990-luvun puoleenväliin asti, eikä niissä ole käytetty mitään fysiikkaan pohjautuvaa nestesimulaatiota. Syvyys-elokuvan vesiolennossa ja muissa varhaisissa digitaalisissa vesitehosteissa on käytetty erilaisia filttäreitä sekä *displacement*- ja *bump*-karttoja, joilla voi harmaasävykuvia käyttäen muokata polygoniverkon pinnanmuotoja. (*Displacement*- ja *bump*-kartoista tarkemmin kappaleessa 1.3 Nesteen pinta eri tekniikoilla.) (Kezsbom 1999; Seymour 2011.)

1990-luvun loppupuolella nestesimulaattoreita alettiin kehittämään 3D-artistille sopivammaksi. Fyysisen tarkkuuden sijasta keskityttiin enemmän mahdollisimman reaaliaikaiseen käyttöliittymään. (Seymour 2011.) Ajatuksena oli, että ainoastaan nesteen uskottava ulkomuoto ja liike ovat tärkeitä, toisin kuin laskentatehoja syövä fyysinen tarkkuus, jota tarvitaan tieteellisiin simulaatioihin (Stam 1999).

Lähiaikoina tullaan näkemään melko varmasti reaaliaikaisia dynaamisia kolmiulotteisia nestesimulaatioita, esimerkiksi Chentanez ja Müller kirjoittavat julkaisussaan Real-Time Eulerian Water Simulation Using a Restricted Tall Cell Grid (2011), miten reaaliaikaisessa grafiikassa pystyttäisiin toteuttamaan suuria vesimassoja. Reaaliaikaisia simulaatioita voidaan käyttää esimerkiksi peleissä tuomaan lisäämään uskottavuutta pelimaailmisiin (Gourlay 2012).

1.2 Nestesimulaatio 3D-animaatiossa ja –visualisoinnissa

Vaikka nykyiset tietokoneet suoriutuvat nopeasti suurista laskusuorituksista, niin nesteiden virtaussimulointi voi olla silti aikaa vievä prosessi. 3D-artistin kannattaa miettiä projektin alussa tarvitaanko neste kolmiulotteisena simulaationa, vai riittääkö partikkeliefekti tai polygoniverkkoa tai sen pintaa muokkaavat kartat kuten *displacement*-, *normal*- ja *bump*-kartat. (Hess 2012, 189 - 195.) Käyn läpi eri karttojen ja dynaamisen simuloinnin ominaisuuksia luvussa 1.3.

Hyvin suunniteltu kohtausta säästää aikaa ja ylimääräisiä testisimulaatiota. Ennen ensimmäistäkään simulaatiota voidaan perehtyä muun muassa siihen, miten virtaukset ja

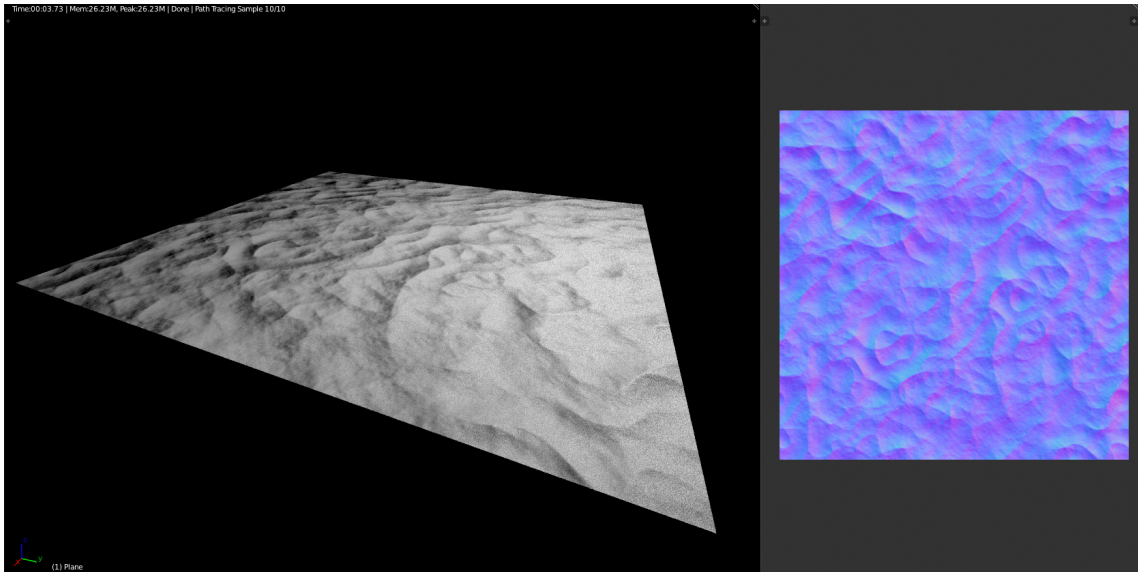
aallot toimivat sekä miten niiden halutaan käyttäytyä kyseessä olevassa skenessä. Tällä tavoin voidaan rakentaa skene, joka toimii jo teoriassa ja vähentää tarvittavien testisimulaatioiden määrää. 3D-visualisoinnissa kannattaa aina kuitenkin simuloida vain näkyvät ja välttämättömimmät alueet. Esimerkiksi meren pohjaan asti ei tarvitse simuloida, jos haluaa aallot iskemään laivan kylkeen. Simuloinnin kanssa kannattaa myös käyttää kevyintä menetelmää, jolla saadaan haluttu lopputulos. Menetelmiä voidaan myös yhdistellä, tällöin vain aktiivinen osa simuloidaan. Esimerkiksi meren pinnasta, alue joka ei ole toiminnan keskipisteenä, voidaan tehdä kevyemmin *displacement*-kartalla. On myös tarpeetonta nostaa resoluutio niin tarkaksi, että simulaattori pystyy simuloimaan pienimmätkin pisarat. Pisarat kannattaa lisätä mieluummin kevyemmällä partikkelitehosteella, jotka hyvin ajoitettuna lisäävät uskottavuutta tehosteseen (Blender 2015).

1.3 Nesteen pinta eriteknikoilla

Tässä kappaleessa käyn läpi kolme eri tekniikkaa, joita voi käyttää neste-elementtien tekemiseen 3D-grafiikassa. Tekniikat ovat *Normal*- ja *bump*-kartat, jotka eivät vaikuta itse geometriaan, sekä geometriaa muuttavat *displacement*-kartat ja nesteen kolmiulotteinen simulointi, eli nestesimulaatio. Näiden tekniikoiden kanssa käytetään useasti myös muita tapoja paremman lopputuloksen saamiseksi, kuten esimerkiksi erilaisia heijastus- ja tekstuurikarttoja sekä partikkeliefektejä (Blender 2015).

1.3.1 Normal- ja bump-kartta

3D-ohjelmistoissa *Normal*- ja *bump* -kartat tekevät polygoniverkon pinnalle illuusion tarkemmasta geometriasta vaikuttamalla siihen miten valot ja varjot piirtyvät sen pintaan. Ne eivät kuitenkaan vaikuta polygoniverkon muotoon millään tavoin, joten ne eivät tarvitse tiheämpää topologiaa tai topologian muuttamista millään tavoin. (Blender 2015.) Koska ne eivät tarvitse tarkempaa topologiaa, niin ne ovat yleisesti kevyempiä ratkaisuja kuin *displacement*-kartan käyttö. Ne ovat hyvä ratkaisu reaaliaikaisessa grafiikassa sekä tilanteissa missä pinta on tarpeeksi kaukana, eikä pinnan silhuetin tarvitse muuttua. Geometriaa muuttamattomien karttojen ongelma on se, että kolmiulotteisuuden illuusio voi helposti rikkoontua kameran ollessa väärässä kulmassa tai liian lähellä kohdetta. (Derakhshani 2012, 299.)

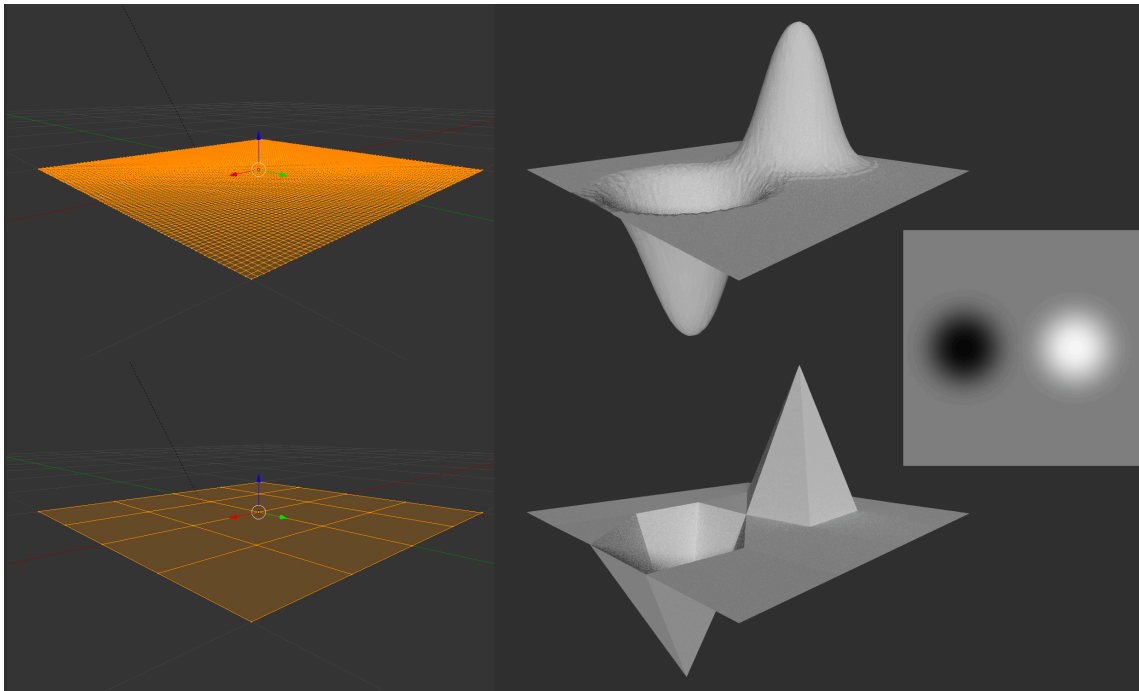


Kuva 3 Kuvakaappaus Blenderistä. Vasemmalla on taso johon on kiinnitetty *normal*-kartta. Oikealla on käytetty *normal*-kartta (kartta: Digitalrune 2014). Tämä ratkaisu toimii, kun alue on kaukana. Pinnan kolmiulotteinen illuusio voi kadota, kun kohdetta katsotaan liian läheltä tai väärästä kulmasta.

1.3.2 *Displacement*-kartta ja *ocean*-simulaattori

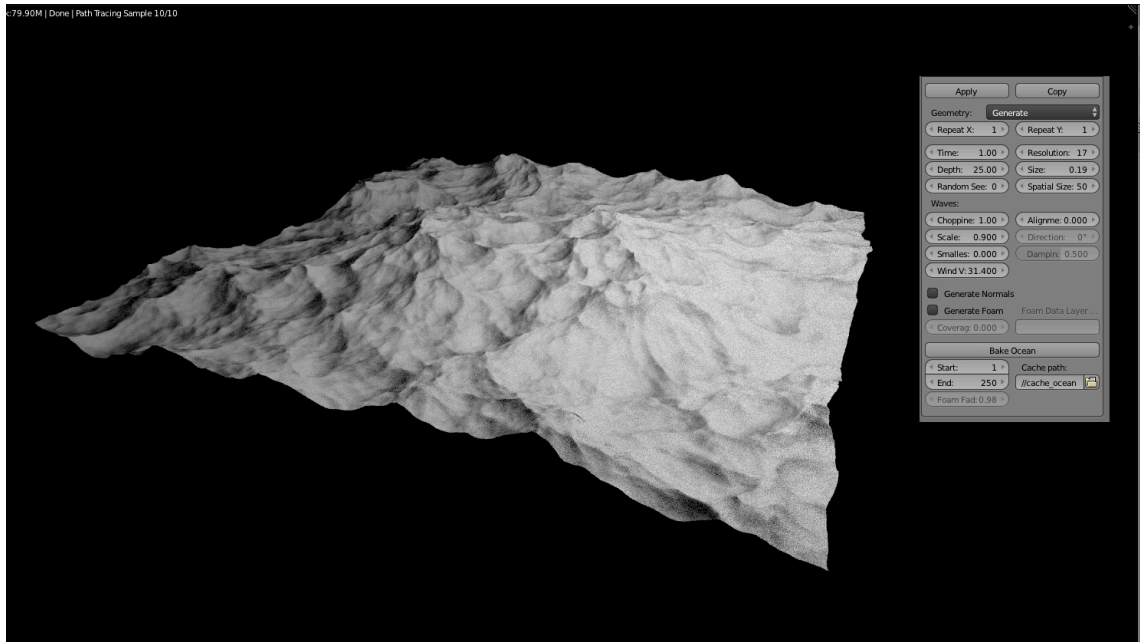
Displacement-kartta on harmaasävykuvakartta joka vaikuttaa polygoniverkon renderöityyn ulkomuotoon. Se siirtää polygoniverkon verteksejä *displacement*-kartan vaikuttavien pikseleiden arvon mukaan sisään- tai ulospäin. Toisin kuin *normal*- ja *bump*-kartat, *displacement*-kartta muuttaa polygoniverkon silhuettia ja sen langettamaa varjoa. Tämän takia *displacement*-kartta toimii, kun halutaan muokatun geometrian peittävän takana olevia asioita, esimerkiksi laiva aaltojen taakse. (Blender 2015.)

Displacement-kartan tarkkuus tulee käytetyn kuvakartan tarkkuudesta ja sen bit-tisyvyydestä sekä polygoniverkon tarkkuudesta. Hyvin toimiakseen se tarvitsee tasaisen topologian, joka voidaan jakaa siististi topologiaa jakavalla modifikaattorilla, kuten *subdivision surfacella* (kuva X). *Displacement*-karttaa käyttäessä polygoniverkon pintaan tulee virheitä jos sen topologia on epätasainen tai se ei ole tarpeeksi tarkka. (Blender 2015.)



Kuva 4 Displacement-karta vaikuttaa siihen, miten polygoniverkko renderöidään. Harmaasävy on oletuksena neutraali ja ei vaikuta lopputulokseen, kun taas tummat ja vaaleat sävyt työntävät verkkejä eri suuntiin. Objektiin polygoniverkon tiheys ja topologian siisteys vaikuttaa siihen, miten displacement-karta toimii.

Blenderissä ja Mayassa on molemmissa omat *ocean*-simulaattorit, jotka käyttävät *displacement*-menetelmää. Nämä ovat nopeita tapoja tehdä laajoja nestepintoja, kuten merta, ja ovat yleisesti helpompi vaihtoehto, kuin manuaalisesti tehty *displacement*-karta. Ocen-simulaattori simuloi kaksiulotteista kuvakarttaa käyttäjän asettamien asetusten mukaan. Asetuksissa on yleisesti muun muassa tuulen suunta ja kovuus sekä aaltojen korkeus. Asetukset tehdään lohkolle, jota voi saumattomasti monistaa syvyys- ja leveysakselin suuntaan, ja näin saadaan tehtyä esimerkiksi meren ulappa. (Blender 2015; Autodesk 2014.)



Kuva 5 Kuvakaappaus Blenderin ocean-simulaattorista. Ocean-simulaattorilla saadaan helposti tehtyä animoitua nesteen pintaa ja se on monistettavissa saumattomasti, joten sillä voi tehdä esimerkiksi meren ulapan.

1.3.3 Nestesimulaatio

Nestesimulaatio on näistä menetelmistä hitain ja sitä tulisi käyttää, silloin kun muilla menetelmillä ei pystytäkään saamaan toivottua lopputulosta. Tällaisia tilanteita ovat muun muassa, kun neste täyttää jonkun tilan, nesteen massa hajoaa pisaroiksi tai halutaan nesteen olevan dynaamisesti vuorovaikutuksessa muiden objektien kanssa. (Hess 2012, 189 - 195.) Käyn nestesimulaation tarkemmin läpi kappaleessa 2.2 Nestesimulaation työvaiheet.

2 Tutkimusongelma ja menetelmät

Blenderin dokumentaatioissa (2015) listataan neljä tekijää, jotka ovat ongelmallisia Blenderin Lattice Boltzmann Method -ratkaisijalle (LBM -ratkaisija) ja oletuksena on, että ne hidastavat simulaation laskenta-aikaa. Ongelmat ovat seuraavat:

- Resoluution tarkentaminen
- Viskositeetin vähentäminen
- Simuloitavan nesteen nopeus
- Simuloitavan skenen suurentaminen

Opinnäytetyössäni toistan edellä mainitut ongelmat Blender- ja Maya ohjelmistoilla. Tarkoitukseni on selvittää, kuinka paljon ja miten edellä mainitut tekijät vaikuttavat simulointiin. Toistan kokeet molemmilla ohjelmistoilla mahdollisimman samanlaisina ja lopuksi vertailen näiden kahden simulaattorin ominaisuuksia toisiinsa. Ohjelmistoilla on hiukan erilaiset käyttölogiikat simulaattoreissa, eikä kokeita voi toistaa täysin identtisinä molemmissa ohjelmistoissa. Olettamukseni on kuitenkin se, että testeissä ilmenee ominaisuuksia, joita voi vertailla ohjelmistojen kesken.

Kaikissa kokeissa mitataan seuraavat asiat:

- Simulointiin kuluvan ajan
- Kuinka paljon kolmioita on simuloinnin tuloksena saadussa polygoniverkossa animaation viimeisessä kuvaruudussa simuloinnin loputtua
- Kuinka suureksi cache-tiedosto kasvaa
- Miltä lopputulos vaikuttaa silmämääräisesti tarkasteltuna

Testit suoritan mahdollisimman yksinkertaisessa testiskenessä. Skenen mallinnan Blender-ohjelmistossa ja vien Maya-ohjelmistoon obj-tiedostomuodossa, joka on yleinen tiedostomuoto 3D-objektien siirtämisessä ohjelmistosta toiseen (Sherrod 2008, 600). Mayassa varmistan skenen olevan samassa mittakaavassa Blenderin skeneen verrattuna.

Teen kaikki simulaatiot omalla kotikoneella, joka on Applen iMac (21,5", Mid 2011). Koneen prosessori on 27, GHz Intel Core i5 ja muistia 24 GB.

2.1 Opinnäytetyössä käytettävät nestesimulaattorit

Useimmissa 3D-ohjelmistoissa, joilla voi mallintaa ja animoida, on mukana jonkinlainen nestesimulaattori, tai sellaisen voi ladata ohjelmistoon lisäosana kolmannelta osapuolelta. On myös ohjelmistoja, jotka on suunniteltu varta vasten simulointiin, kuten esimerkiksi Next Limit Technologiesin Realflow. (Seymour 2014.) Tähän opinnäytetyöhön olen valinnut kaksi yleis- 3D-ohjelmistoa, joissa on omat nestesimulaattorit. Ohjelmistot ovat Blender ja Autodeskin Maya. Näitä molempia ohjelmistoja olemme käyttäneet yleisesti koulun projekteissa.

2.1.1 Maya Bifrost

Bifrost julkaistiin Mayan 2015-version mukana keväällä 2014 (digitalartsonline 2014). Bifrostista saatetaan käyttää myös Bifrost-kirjoitusasua. Autodeskin dokumentaatiot käyttävät molempia sekaisin (Autodesk 2014). Bifrostin on tarkoitus olla Autodeskin mukaan helppo ”point and click”-tyylinen nestesimulaattori, jolla tavallinen yleis-3D-artisti pystyy luomaan vakuuttavan näköisiä nestesimulaatioita. Kaiken tämän pitäisi onnistua ilman suurempaa teknistä osaamista. (Cgchannel 2014.)

Bifrost käyttää simulaatioissaan fluid implicit particle -menetelmää (FLIP), joka on alun perin vuodelta 1986, ja sitä on siitä asti kehitelty. Se perustuu The Particle in Cell-menetelmään vuodelta 1963. (Ek, Englesson & Kilby 2011.) Menetelmien syvempi ymmärtäminen ei ole välttämätöntä 3D-artistille, joten en lähde tässä opinnäytetyössä käymään niiden tekniikkaa tarkemmin läpi.

2.1.2 Blender

Blender on ilmainen 3D-ohjelmisto, jonka kehittäminen on alkanut vuonna 1995 animaatio-studio NeoGeon projektina, ja se on nykyään vapaan lähdekoodin ohjelmisto. Blenderiä voi käyttää ilmaiseksi niin omiin kuin kaupallisiin projekteihin ja muun muassa siksi siitä on tullut houkutteleva vaihtoehto freelance-3D-artisteille. (Blender 2015.)

Blenderissä on ollut mukana nestesimulaattori versiosta 2.4 lähtien (Blender 2015). Simulaattori kehitettiin alun perin vuonna 2005 Googlen sponsoroimalla Summer of

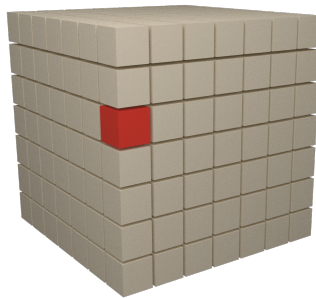
Code-projektina (Mullen 2013, 253). Simulaattori käyttää algoritmina Lattice Boltzmann Method -menetelmää (LBM), joka on yleisten Navier-Stokes- ja Smoothed Particle Hydrodynamics -menetelmien välimaastosta (Blender 2015).

2.2 Nestesimulaation työvaiheet

Seuraavaksi käyn läpi työvaiheet, joilla saadaan yksinkertainen nestesimulaatio Blender-ohjelmistolla. Blender-ohjelmiston ja Bifrost-simulaattorin työvaiheet ovat hiukan erilaisia, mutta niissä molemmissa on käytössä samankaltaiset elementit, joilla saadaan tehtyä yksinkertainen simulaatio.

Nestesimulaatio tarvitsee määrittelyalueen (*domain*), jonka sisällä tapahtuu kaikki mikä liittyy simulaatioon. Määrittelyalueeksi kannattaa valita suorakulmainen särmiö, joka on mitoitettu simulaation mittakaavaan sopivaksi. Määrittelyalueena voi toimia monimutkaisempikin objekti, mutta simuloinnissa se katsotaan aina olevan suorakulmainen särmiö objektin pisimpien mittojen mukaan. Blender-ohjelmistossa simulaation kaikki tärkeimmät asetukset on määrittelyalueeksi asetetussa objektissa ja sille asetettu materiaali toimii simulaation tuloksena syntyvän polygoniverkon materiaalina. (Blender 2015.)

Määrittelyalueen asetuksista tärkein on resoluutio. Resoluutio määrittää simulaation tarkkuuden. Sen arvon voi asettaa erikseen renderointiä ja esikatselua varten, mutta kätevintä on yleensä asettaa esikatselu suoraan renderointitarkkuudeksi. Blenderissä resoluution arvo tarkoittaa kuinka monta vokselia mahtuu määrittelyalueen pitkälle sivulle. Vokselit ovat eräänlaisia kolmiulotteisia pikseleitä ja niillä voi esimerkiksi määrittellä kolmiulotteisen muodon resoluution. (Mullen 2013, 253-254, 269-270.)



Kuva 6 Vokselit ovat kolmiulotteisia pikseleitä, joilla voi määrittellä kolmiulotteisen muodon resoluution.

Blenderissä määrittelyalueen asetuksista suurin osa on itsensä selittäviä, paitsi *time*-asetus. Tämä asetus aiheuttaa jonkun verran hämmennystä, jos sen mekanismia ei tiedä. Siinä asetetaan simulaation aika sekunneissa Blenderin aikajanan ensimmäisen kuvaruudun kohdalla ja aika viimeisen kuvaruudun kohdalla. Eli jos haluaa, että lähteestä on tullut skeneen nestettä ennen ensimmäistä kuvaruutua, niin *start*-arvoksi asetetaan arvo joka on enemmän kuin nolla. Jos simulaatio halutaan alkavan alusta ja sen liikkuvan normaalinopeudella, niin 250 kuvaruudun animaatiossa, joka on 25 kuvaruutua sekunnissa, asetetaan *start*-arvo nolaksi ja *end*-arvoksi asetetaan 10. (Mullen 2013, 269.)

Blenderin aloituskenen kuutio toimii hyvin määrittelyalueena. Objektin määrittäminen määrittelyalueeksi tapahtuu fysiikkavälilehdestä valitsemalla *fluid*, jonka jälkeen *fluidin* tyyppiä valitaan *domain*. (Blender 2015.) Bifrostissa määrittelyaluetta ei tarvitse erikseen asettaa, vaan se tulee automaattisesti skeneen simuloidessa (Autodesk 2014).

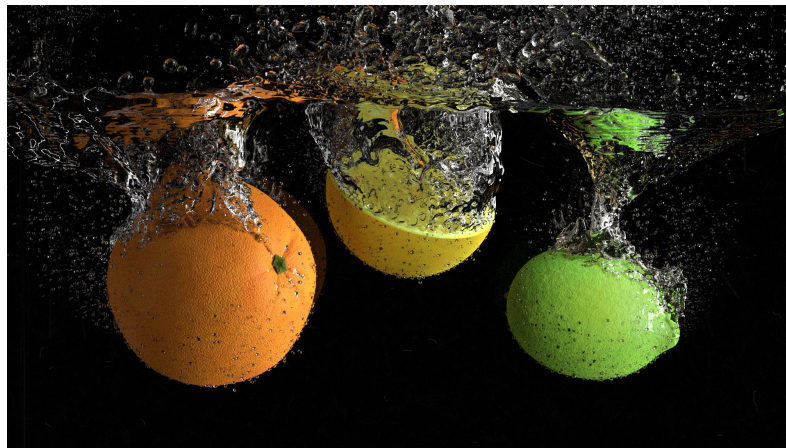
Määrittelyalueen lisäksi simulaatiota varten tarvitaan vielä vähintään nesteen lähde. Lähteitä on kahdenlaisia: *fluid* ja *inflow*. *Fluidiksi* määritelty objekti muuttuu ennen simulaation alkua simuloitavaksi nesteeksi. *Inflow* muuttuu myös samalla tavalla, mutta se tuo simulaation aikana skeneen jatkuvasti lisää nestettä. Molemmille voi asettaa lähtövoiman millä neste lähtee liikkeelle *initial velocity* -asetuksesta. Vauhti asetetaan x, y ja z akseleille, joista neste saa simulaation alussa paineen mihin suuntaan se lähtee liikkeelle. *Inflow* -asetuksella neste suihkuaa jatkuvasti määriteltyyn suuntaan. *Volume initialization* -asetus määrittää muuttuuko nesteeksi objektin tilavuus, ulkopinta vai molemmat. (Blender 2015.)

Nestesimulaatiot tallentuvat tietokoneelle cache-tiedostoina, jotta simulaatio voidaan avata myöhemmin uudestaan. Simulaatiossa jokaisesta kuvaruudusta tallentuu tietokoneelle oma cache-tiedosto, eli yksi simulaatio muodostuu useasta cache-tiedostosta. Nämä tiedostot sisältävät erilaisia tietoja simulaatiosta kyseisen kuvaruudun kohdalta, kuten esimerkiksi polygoniverkon muodon. (Mullen 2013, 256.)

Kun simulaatioon on asetettu määrittelyalue, lähde ja tallennuskohde, niin sen voi simuloida. Blenderin simulaatiossa neste lähtee sille asetetulla lähtövoimalla liikkeelle ja törmää määrittelyalueen reunaan kuin seinään. (Blender 2015.) Bifrostissa määrittelyalueen koko muuttuu simulaation mukaan automaattisesti, joten näillä asetuksilla neste vain valuu esteettä loputtomiin (Autodesk 2014). Karkailevaa nestettä ja muuta ylimää-

räistä nestettä voi poistaa simulaatiosta tekemällä sinne objektin joka toimii ulosvirtauksena asettamalla sille nesteasetuksista tyyppiä *outflow* (Blender 2015).

Monimutkaisempia simulaatioita varten tarvitaan pintoja, joihin neste voi osua sekä voimia, jotka muokkaavat nesteen käyttäytymistä simulaatiossa. Objekteja voidaan asettaa esteiksi määrittämällä niille fysiikkavälilehdeltä nesteasetukset ja valitsemalla tyyppiä *obstacle*. Jos nesteen halutaan yrittävän muovautuvan jonkun geometrian mukaan, esimerkiksi ihmishahmoksi, niin objektin voi määrittää nesteasetuksista kontrolliobjektiksi. Kontrolliobjekti joko työntää nestettä itsestään pois päin, tai vetää nestettä itseensä täyttäen omaa muotoaan. (Blender 2015.)



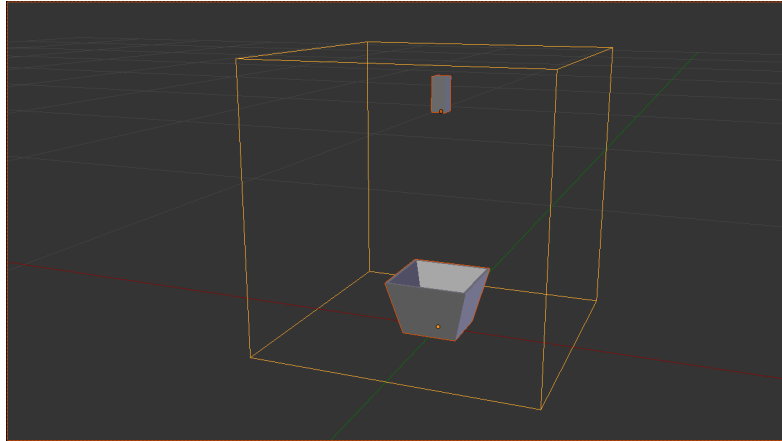
Kuva 7 Hedelmät ovat määriteltä esteiksi Blenderin nestesimulaatiossa. Kuva: www.blenderguru.com

Aikaisempien optioiden lisäksi Blenderin nestevalikossa on mahdollista määrittää objekti nestesimulaation partikkeliobjektiksi. Partikkeleita on kolme erilaista: *drops* irtoavat simulaatiosta roiskeiden lailla, *floats* seuraavat simulaation pinnalla ja *tracer* jotka seuraavat simulaatiota jäljessä. *Drops* ja *floats* -partikkeleilla voidaan tuoda lisää yksityiskohtia simulaation roiskeisiin ja pintaan nostamatta itse simulaation resoluutiota, kun taas *tracer*-partikkeleilla voidaan tehdä esimerkiksi simulaation jälkeen jäävää sumua. Partikkeliobjekteille pitää osoittaa oma cache-kansio jonne simulaatio tallentuu. (Mullen 2013, 271-276.)

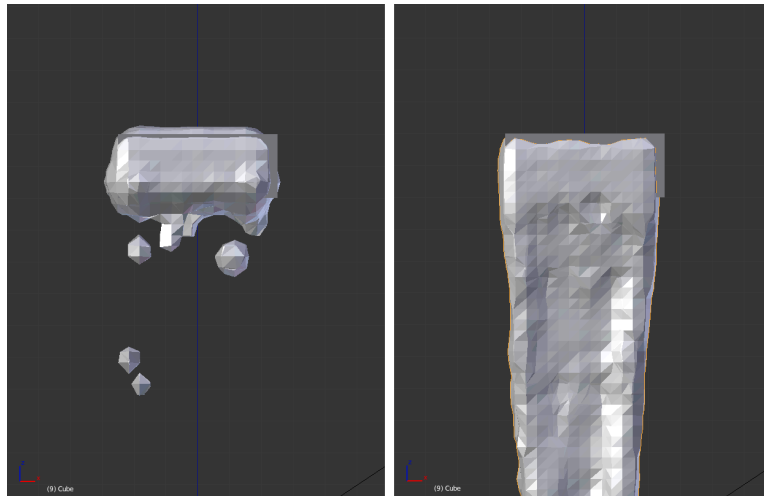
2.3 Testiskene

Testiskenessä on nollapisteessä kulmikas kulho-objekti jonka sivut ovat 15 cm ja korkeus 10 cm. Emitterinä toimii 2,5 x 2,5 x 6 cm suorakulmainen särmiö, jonka alaosa on

40 cm korkeudella skenen nollapisteestä. Nämä kaikki ovat kuution sisällä, jonka sivut ovat 50 cm kanttiinsa. (Kuva X.) Kuutio on asetettu määrittelyalueeksi Blenderissä. Mayassa kuutio on asetettu esteeksi, jotta simulaatio pysyy samalla alueella kuin Blenderin simulaatioissa. Mayassa määrittelyaluetta ei erikseen aseteta, vaan se tulee automaattisesti, kun skeneen asetetaan Bifrost-simulaatio (Autodesk 2014).



Kuva 8 Testiskene Blenderissä



Kuva 9 Vasemmalla neste jää tihkumaan ilman lähtövoiman asettamista. Oikealla on toivottu tulos.

Emitteristä lähtevän nesteen ominaisuudeksi on valittu esiasetuksista vesi molemmissa ohjelmistoissa. Blenderissä neste jää tihkumaan emitteristä hiljalleen, jos sille ei aseta mitään lähtöarvoa (kuva X). Asetin molemmissa ohjelmistoissa nesteille arvot, joilla ne lähtevät emitteristä suoraan alaspäin. Mayassa emitterillä ei ole oletuksena mitään asetusta, jolla vaikutetaan nesteen lähtösuuntaan tai -nopeuteen, joten lisäsin emitterin Bifrostissa myös kiihdyttimeksi. Bifrostissa kiihdyttimeksi asetettu objekti toimii nesteisiin vaikuttavana voimana, jolla voi vaikuttaa, millaisella teholla, alueella ja mihin suun-

taan nestettä pakotetaan. Blenderissä asetin emitterin lähtöarvoksi 3,2 metriä sekunnissa ja Bifrostissa kiihdyttimen vaikuttajan arvoksi 6. Arvot sain tekemällä testisimulaatioita ja yrittämällä saada paineen vaikuttamaan suurin piirtein samalta molemmissa ohjelmistoissa. Säädin myös molemmissa ohjelmistoissa simulaatioiden aika-attribuutteja niin, että neste näyttäisi silmämääräisesti liikkuvan samalla nopeudella ja osuvan oikeassa kuvaruudussa kulhon pohjaan. Kuvaruutu, jossa simulaatio osuu kulhoon, vaihtelee ruudulla tai parilla eri simulaatiokerroilla sekä resoluution muuttuessa. Osumisruudun kohdistin Mayan ja Blenderin välillä keskitarhoilla resoluutiolla – Blenderissä resoluutiolla 150 ja Mayassa 0,5. Osumisajan tärkeyden perustelen sillä, että olen omissa simulaatioissani huomannut simuloinnin hidastuvan huomattavasti nesteen törmätessä johonkin esteeseen ja muuttaessaan suuntaa, kun taas vapaasti valuva neste simuloituu suhteellisen nopeasti.

2.4 Resoluution tarkentaminen

Resoluutio on yksi tärkeimmistä parametreista joka vaikuttaa simulaation laskenta-aikaan sekä, millainen visuaalinen lopputulos tulee olemaan (Mullen 2013, 269). Tässä testissä tutkin, kuinka paljon simulaatioaikaan vaikuttaa resoluution tarkentaminen.

Blenderin resoluutio on sama kuin kuinka monta vokselia mahtuu määrittelyalueen pisinimmälle sivulle. Vokselit ovat eräänlaisia kolmiulotteisia pikseleitä, ja niillä voi määrittellä esimerkiksi kolmiulotteisen tilan resoluution. (Mullen 2013, 269-270.) Bifrostissa resoluutiota kontrolloidaan *master voxel size* -attribuutilla. Attribuutin arvo on Mayan asetuksista asetetun skenen skaalayksikön mukaan, eli jos skaalayksikkö on senttimetri, niin simulaatio tapahtuu senttimetrin tarkkuudella *master voxel size* arvolla 1. (Autodesk 2014.)

Oletuksenani on, että simulaatioaika nousee nopeampaa suhteessa resoluution tarkentamiseen. Tarkkuuden kasvaessa tietokoneelle pitäisi tulla enemmän laskettavaa, mikä hidastaa prosessia. Oletan simulaation tarvittavan ajan jossain vaiheessa nousevan huomattavasti molemmissa ohjelmistoissa. Kaupallisessa Mayassa mahdollisesti myöhemmin, tai ei ollenkaan opinnäytetyössäni tehtävien kokeiden mittakaavassa.

2.5 Alhainen viskositeetti

Viskositeetti on nesteiden tunnusomainen ominaisuus, joka määrittää, millainen neste on kyseessä. Se muokkaa nesteen käyttäytymistä, kuten kuinka paljon neste vastustaa sitä muovaavia voimia, kuinka helposti se roiskuu tai miten se virtaa. Esimerkiksi vedellä on alhainen viskositeetti: se on ohutta, virtaa nopeasti ja pinta rikkoutuu helposti. Korkean viskositeetin aineet ovat jähmeitä, ne eivät roisku yhtä helposti ja ne pysyvät helpommin muodossaan. Viskositeetti ei ole kuitenkaan ainut parametri, joka vaikuttaa simuloitun nesteen olemukseen, esimerkiksi myös resoluutio ja liukkaus muuttavat simuloitun nesteen käyttäytymistä. (Dutt, Ghosh, Prasad, Rani & Viswanath 2007, 1; Mullen 2013, 263.)

Oletuksenani on, että viskositeetin laskeminen vaikuttaa laskenta-aikaan tasaisesti. Uskon myös laskennan olevan hitaampaa, kun massat liikkuvat nopeammin ja hajoavat helpommin. Ongelmat ovat mitä luultavammin samankaltaisia kuin nopean vauhdin kanssa.

2.6 Nopeus

Blenderin käyttämä Lattice Boltzmann -menetelmän pitäisi olla suhteellisen nopea ja tarkka simuloimaan nesteiden ja kaasujen virtauksia hitailla nopeuksilla. Menetelmällä on kuitenkin artikkelien mukaan luontaisia vaikeuksia simuloida suuria nopeuksia. (Chen & Doolen 1998, 335; Blender 2015.)

Oletan, että kun simuloitun nesteen nopeus nousee tarpeeksi suureksi, niin simulaation polygoniverkkoon syntyy näkyviä epäloogisia repeämiä tai muita artefakteja. Uskon simulointiajan kasvavan tasaisessa suhteessa nopeuteen.

2.7 Suuri määrittelyalue

Määrittelyalue on suorakulmainen särmiö, joka jakaantuu vokseleihin ja jonka alueella kaikki simulaatio tapahtuu. Alueen ulkopuolella ei tapahdu mitään simulaatiota, joten määrittelyalueen pitää olla yhtä suuri kuin mihin kauimmaisten pisaroiden halutaan lentävän. (Mullen 2013, 254)

Toistaakseen simulaatioita yhtä tarkasti suuri määrittelyalue tarvitsee tarkemman resoluution kuin pienempi määrittelyalue, joka on rajattu tarkemmin. Suuremmassa määrittelyalueessa laskentatarkkuus on levittäytynyt suuremmalle alueelle, eikä sen tarkkuus mahdollisesti riitä tuottamaan mitään näkyvää tulosta pienen alueen simulaatiossa. Tämän takia simulaatioiden laskenta-ajat saattavat venyä epäkäytännöllisen pitkiksi, jos yksityiskohtaiset simulaatiot levittäytyvät isolle alalle. Simuloitava alue kannattaa miettiä ja rajata tarkasti, jotta simulointi ei turhaan laske alueita, jotka eivät vaikuta lopputulokseen. (Hess 2012, 195.)

Oletan simulaatioaikojen nousevan määrittelyalueen suuretessa hiukan enemmän kuin pelkän resoluution nostamisella, jos simulaation kaikki muut elementit ovat samassa suhteessa simulaatioon.

3 Tulokset

3.1 Resoluution tarkentaminen

Taulukossa X on testisimulaatioiden resoluutiot ja saadut simulaatioajat. Blender-ohjelmistossa aloitin simulaatiot resoluution arvosta 12, minkä jälkeen nostin sen arvoon 25. Tämän jälkeen kaksinkertaistin resoluution arvon jokaista simulaatiokertaa varten arvoon 400 asti. Bifrostissa karkein resoluutio oli 1,6, jota lähdin puolittamaan arvoon 0,05 asti. Muita attribuutteja en muuttanut simulointien välillä.

Taulukko 1. Taulukkoon merkitty seuraavat: käytetty resoluutio, simulointiin kulunut aika, kolmioiden määrä simulaation viimeisessä kuvaruudussa sekä simulaation cache-tiedostojen yhteen laskettu koko megatavuissa.

Blender – resoluution tarkentaminen				
Järjestys-numero	Resoluutio	Simulointiaika	Kolmioiden määrä	Cachen koko MB
1	12	-	-	-
2	25	00:00:02	580	1,3
3	50	00:00:08	3280	3,6
4	100	00:01:41	21808	16,1
5	200	00:22:37	205248	118,4
6	400	07:40:30	2075632	1170
Bifrost – resoluution tarkentaminen				
Järjestys-numero	Resoluutio	Simulointiaika	Kolmioiden määrä	Cachen koko MB
1	1,6	00:00:26	30424	151
2	0,8	00:00:59	107128	366
3	0,4	00:03:13	440048	1140
4	0,2	00:07:36	631276	2780
5	0,1	00:39:48	582872	12410
6	0,05	06:26:26	2156572	68330

3.2 Viskositeetin laskeminen

Mayan 2015-versiossa Bifrostista puuttuu kokonaan viskositeetin säätömahdollisuus, ja sillä voi simuloida ainoastaan veden kaltaisia nesteitä. Viskositeetin säätö on mukana Maya 2016-versiossa joka ilmestyy keväällä 2015 (Seymour 2014). Vaikka Bifrostilla sai hyytelömäisempää ja sitkoisempaa jälkeä laskemalla resoluutiota, niin siitä puuttuu esimerkiksi ominaisuus säilyttää muotoa, jota tarvitaan, jos esimerkiksi halutaan simuloida suklaan kasaantumista keksin päälle. Blenderissä viskositeetin muokkaaminen on ollut mukana alusta asti (Blender 2015).

Blender käyttää kinemaattista viskositeettia ja sen voi säätää määrittelyalueen ”*Domain World*”-paneelin asetuksista. Asetuksissa on valmiina kolme esiasetusta: hunaja, vesi ja öljy. Esiasetuksien lisäksi viskoosisuuden voi asettaa manuaalisesti. Tämä tapahtuu esiasetusten alapuolelta, jossa base-kenttään asetetaan kantaluku ja eksponent-kenttään luku 0 - 10 väliltä, joka toimii kymmenkertaisena negatiivisena eksponenttina kantaluvulle. Esimerkiksi oletuksena olevan veden kantaluku on 1 ja eksponentti on 6, jolloin saadaan $1 \cdot 10^{-6} = 0.000001$. Mitä pienempi luku on, niin sitä pienempi viskositeetti sillä on. Valmiita arvoja muun tyyppisille nesteille löytyy keskustelufoorumeilta ja Blenderin omista dokumentaatioista. (Blender 2015.)

Blenderissä tein simulaatiot muuttamalla eksponentin arvoa kahdella aloittaen arvosta 0 ja päättyen maksimiin 10. Kantaluku on kaikissa simuloinneissa 1. Simuloinneissa resoluutio on 200, muita arvoja tai asetuksia en muuttanut. Testin tulokset on luettavissa taulukosta X.

Taulukko 2. Taulukossa on Blenderillä saadut tulokset, kun nestesimulaation viskositeetti-arvon exponent-lukua nostetaan jokaista simulaatio kertaa varten kahdella.

Blender – viskositeetin laskeminen					
Järjestys-numero	Resoluutio	Viskositeetti	Simulointiaika	Kolmioiden määrä	Cachen koko MB
1	200	0	00:11:18	50548	41,8
2	200	2	00:10:52	47874	37,5
3	200	4	00:18:31	140204	86,5
4	200	6	00:22:37	205248	118,4
5	200	8	00:23:45	202870	116,5
6	200	10	00:23:25	223818	119,3

3.3 Nopeus

Testiskenessä neste virtasi esteettä suoraan alaspäin, ja aikaisempien testien kulho oli korvattu objektilla, joka poistaa simulaatiosta siihen osuvat nesteet. Simulaatioista nestemassoja poistava objekti asetetaan Blenderissä *fluid*-asetuksista *outflow*-tyyppiseksi (Blender 2015), ja Mayassa se lisätään Bifrostiin valitsemalla *kill plane*-objekti Bifrostin asetuksista (Autodesk 2014).

Taulukko 3. Taulukossa on Blenderillä ja Mayalla saadut tulokset kun lähteestä lähtevän nestemassan nopeutta on kaksinkertaistettu jokaista simulaatiokertaa varten, sekä tilanne jossa lähtönopeus on 0.

Blender – nopeus					
Järjestysnumero	Resoluutio	Nopeus	Simulointiaika	Kolmioiden määrä	Cachen koko MB
1	100	0	00:00:43	1060	2,2
2	100	0,8	00:00:46	2844	4,1
3	100	1,6	00:00:55	3384	4,9
4	100	3,2	00:01:13	3820	5,4
5	100	6,4	00:02:12	3858	5,8
6	100	12,8	00:03:41	3886	5,9
Bifrost – nopeus					
Järjestysnumero	Resoluutio	Nopeus	Simulointiaika	Kolmioiden määrä	Cachen koko MB
1	0,4	0	00:00:33	20104	196,3
2	0,4	1,5	00:00:38	22556	221,9
3	0,4	3	00:00:42	23876	245,8
4	0,4	6	00:00:44	25016	257,7
5	0,4	12	00:00:47	25724	273,1
6	0,4	24	00:00:50	21168	284,9

3.4 Suuri määrittelyalue

Blender ja Bifrostin erilaisten resoluution määrittelytapojen takia ohjelmistoja on vaikea verrata toisiinsa määrittelyalueen kasvamisen suhteen. Blenderissä resoluutio on määrittelyalueen pisimmän sivun pituus jaettuna resoluution arvolla. Tämän vuoksi määrittelyalueen kasvaessa resoluutio ei muutu, vaan yksittäinen vokseli kasvaa skenen mukana. (Mullen 2013, 270). Bifrostissa resoluutio on vokselin koko Maya-skenen maailman skaalaysikössä, joten skenen suurentaminen nostaa määrittelyalueen resoluutiota, eli yksittäinen vokseli pysyy samankokoisena (Autodesk 2014). Tämän takia päädyin tekemään testisimulaatiot molemmissa simulaattoreissa kahteen kertaan. Ensimmäisellä kerralla suurentamalla koko skenen kokoa koskematta resoluutioon. Toisella simulaatiolla muutan resoluutiota koon muukaan. Skaalasin simulaation muita elementtejä samassa mittasuhteessa, koska Bifrostissa määrittelyalue tulee automaattisesti simulaatioon, eikä määrittelyalueen kokoa voi manuaalisesti suurentaa. Tämän takia pelkän määrittelyalueen suurentaminen ei onnistunut. Lisäksi suhteutin nesteen lähtönopeuden simuloitavan skenen kokoon, jotta neste osuisi suunnilleen samassa kuvaruudussa esteeseen.

Taulukko 4. Taulukossa on Blenderillä saadut tulokset kun määrittelyalueen kokoa on kaksinkertaistettu jokaista simulaatiota varten. Määrittelyalueen suurentaminen on tehty kaksi kertaa. Ensimmäisellä kerralla resoluutio on kaikissa testeissä sama. Toisella kerralla resoluutio on kaksinkertaistettu määrittelyalueen mukana. Kaikissa testeissä nesteen lähtönopeus on suhteutettuna määrittelyalueen kokoon, jotta se osuisi esteeseen samassa kuvaruudussa.

Blender – suuri määrittelyalue					
Järjestysnumero	Resoluutio	Simuloitavan alueen koko	Simulointiaika	Kolmioiden määrä	Cachen koko MB
1	100	0.125	00:02:04	26896	14,8
2	100	0,25	00:01:43	24620	14
3	100	0,5	00:01:28	22436	16,8
4	100	1	00:01:30	26464	19,2
5	100	2	00:01:36	27408	22,9
Blender – suuri määrittelyalue, muuttuva resoluutio					
Järjestysnumero	Resoluutio	Simuloitavan alueen koko	Simulointiaika	Kolmioiden määrä	Cachen koko MB
1	25	0.125	00:00:02	904	1,6
2	50	0,25	00:00:08	3104	3,1
3	100	0,5	00:01:28	22436	16,8
4	200	1	00:22:25	217470	129,7
5	400	2	07:21:30	2170640	1270

Taulukko 5. Taulukossa on Mayalla saadut tulokset kun määrittelyalueen kokoa on kaksinkertaistettu jokaista simulaatiota varten. Määrittelyalueen suurentaminen on tehty kaksi kertaa. Ensimmäisellä kerralla resoluutio on kaikissa testeissä sama. Toisella kerralla resoluutio on puolitetty määrittelyalueen mukaan. Kaikissa testeissä nesteen lähtönopeus on suhteutettuna määrittelyalueen kokoon, jotta se osuisi esteeseen samassa kuvaruudussa.

Bifrost – suuri määrittelyalue					
Järjestysnumero	Resoluutio	Simuloitavan alueen koko	Simulointiaika	Kolmioiden määrä	Cachen koko MB
1	0,2	0.125	00:02:05	86724	297,8
2	0,2	0,25	00:02:27	362784	1200
3	0,2	0,5	00:07:36	631276	2780
4	0,2	1	00:40:57	745952	12760
5	0,2	2	04:08:31	3052612	39110
Bifrost – suuri määrittelyalue, muuttuva resoluutio					
Järjestysnumero	Resoluutio	Simuloitavan alueen koko	Simulointiaika	Kolmioiden määrä	Cachen koko MB
1	0,05	0.125	00:08:04	131556	2550
2	0,1	0,25	00:07:15	147416	2480
3	0,2	0,5	00:07:36	631276	2780
4	0,4	1	00:09:23	1562536	3490
5	0,8	2	00:15:31	2649904	4790

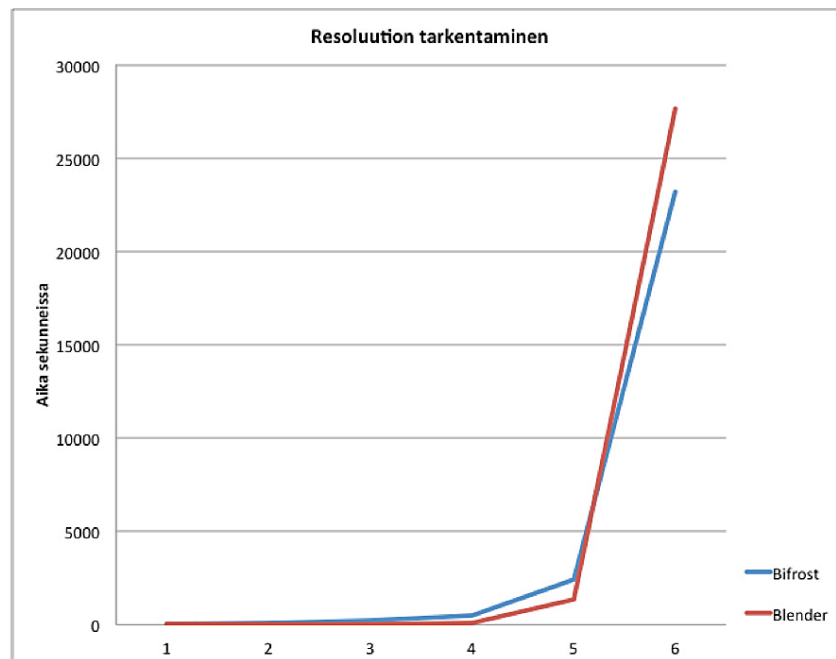
4 Tulosten tulkinta

Molemmissa ohjelmistoissa kaikki neljästä tutkitusta ongelmatekijästä hidasti simulaatioaikoja. Resoluution nostaminen vaikutti odotetusti eniten simulaatioihin. Kolmen sekunnin testisimulaatioiden laskenta moninkertaistui resoluution kaksinkertaistuessa. Resoluution nostaminen vaikutti myös simulaation ulkonäköön ja käyttäytymiseen kaikista eniten. Viskositeetti ja nopeuden nostaminen hidasti tasaisesti simulaatioaikoja, ja Bifrostilla nopeuden nostaminen hajotti myös simulaation eheyden. Viskositeetin muut-

taminen puuttui kokonaan Bifrostista. Blenderissä viskositeetin muuttaminen toimi oletusti. Määrittelyalueen suurentamisesta ei tullut selkeitä tuloksia. Tämä saattoi johtua testin useista muuttujista. Määrittelyaluetestissä tarkempia tuloksia olisi voinut saada erilaisella testiskenellä, tai jos tuloksia olisi kerätty jollain muulla tarkemmalla menetelmällä.

4.1 Resoluution tarkentaminen

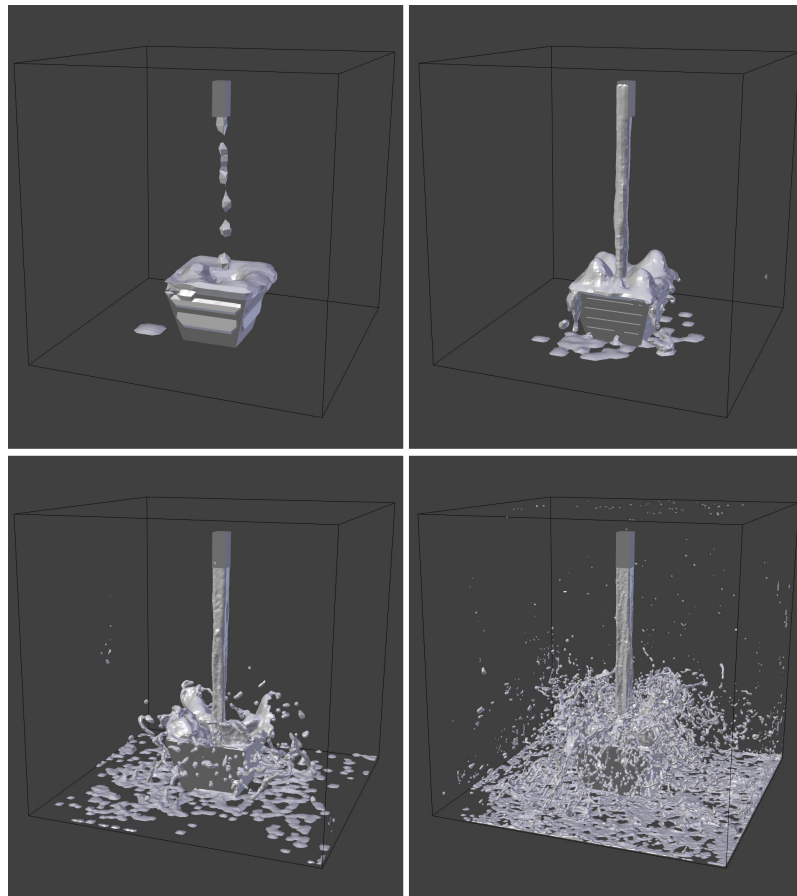
Molemmissa ohjelmistoissa resoluution kaksinkertaistaminen moninkertaisti odotetusti simulaatioaikaa. Blenderin ja Bifrostin simulaatioajat eivät kuitenkaan ole suoraan verrannollisia toisiinsa, koska simulaatioita ei voida toteuttaa täysin yhtenäisesti eri ohjelmistoilla. Kuitenkin kun vertaillaan molempien simulaatioaikoja samassa taulukossa, niin saadaan mielenkiintoinen tulos. Molemmat käyrät ovat melkein identtisiä, kun resoluutiota kaksinkertaistetaan (Kaavio X).



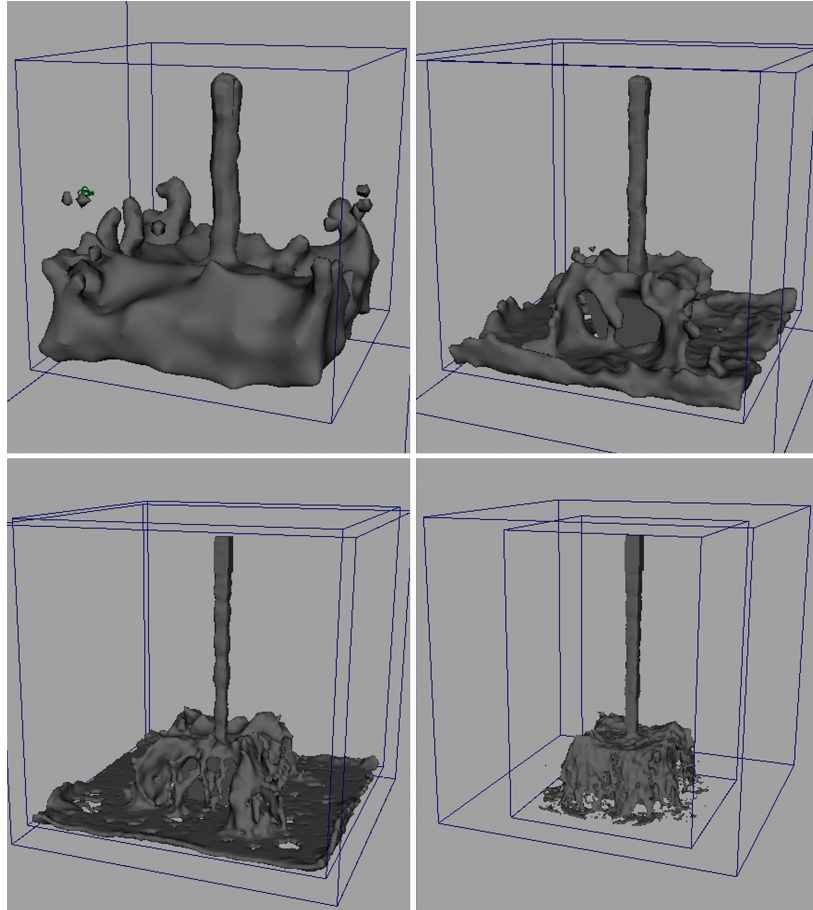
Kaavio 1. Blenderin ja Bifrostin simulaatioista saadut ajat, kun resoluutiota on kaksinkertaistettu jokaista simulaatiota varten. Pystysarakkeessa on simulointiin kulunut aika sekunneissa ja vaakasarakeessa simulaation järjestysnumero. Järjestysnumeron resoluutio löytyy taulukosta X. Simulointiin kulunutta aikaa ei voi suoraan verrata ohjelmistojen välillä, mutta kaaviosta voi nähdä, että molemmissa ohjelmistoissa aika lisääntyy lähes samassa suhteessa toisiinsa nähden.

Testissä kolmen sekunnin simulaation cache-tiedostot vei Bifrostilla tilaa koneen kova-levyltä 68,3 gigatavua resoluution tarkkuudella 0,05, kun taas Blenderillä testin suurin cache-tiedosto oli 1,2 gigatavua resoluutiolla 400. Omissa kokeiluissani noin kolmen sekunnin simulaatiot Bifrostilla ovat saattaneet viedä yli 150 gigatavua tilaa. Tämän takia Bifrostin simulaatiot Maya 2015-versiolla ovat epäkäytännöllisen suuria siirrettäväksi koneelta toiselle tai arkistoitavaksi myöhempää käyttöä varten.

Oli yllättävää, kuinka vaikeaa oli ennustaa pienellä resoluutiolla tehdyistä simulaatiosta korkearesoluutioisen simulaation tulosta. Molemmissa ohjelmistoilla pienellä resoluutiolla neste vaikuttaa olevan jähmeää eikä roisku tai levittäydy niin suurelle alueelle, kuten suurella resoluutiolla (kuva X). Tämä vaikeuttaa esimerkiksi määrittelyalueen koon suunnittelemista niin, että pisarat eivät osu sen reunoihin.

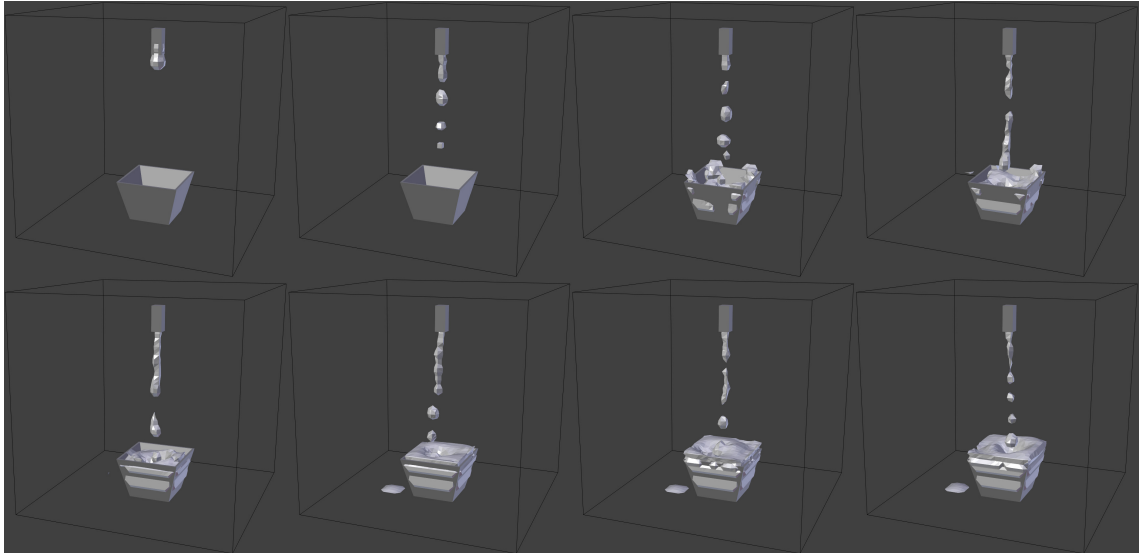


Kuva 10 Blenderissä simulaatioiden viimeinen kuvaruutu. Resoluutio vasemmalta oikealle: 50, 100, 200 ja 400. Näin pienistä kuvista voi jo nähdä, kuinka nesteen paine vaikuttaa erilaiselta resoluution muuttuessa.



Kuva 11 Mayassa simulaatioiden viimeinen kuvaruutu. Master voxel size vasemmalta oikealle: 1.6, 0.8, 0.4, 0.2. Simuloitavan nesteen massa ja tilavuus tuntuvat pienentyvän resoluution noustessa.

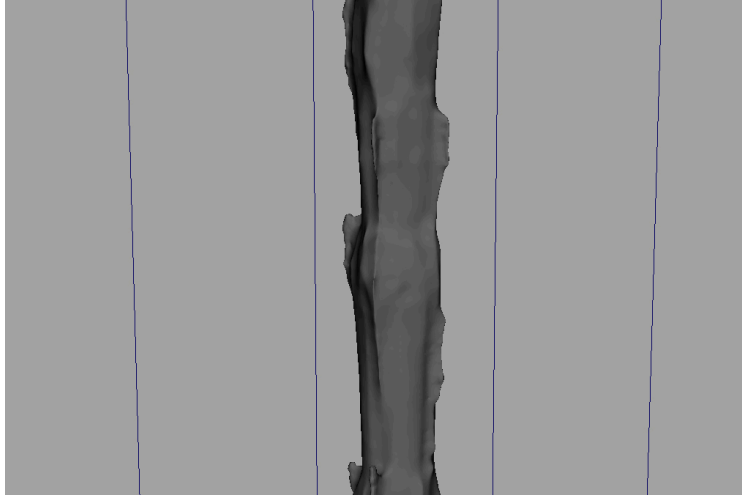
Blenderissä ei saanut alle 75:n resoluutiolla mitään käyttökelpoista tulosta. Karkeilla resoluutiolla polygoniverkon osia saattoi hävitä, ilmestyä tai turvota odottamattomalla tavalla (kuva X). Testiskenessä resoluutiolla 75 sai ensimmäisen karkean simulaation, jolla pystyi jotenkuten hahmottamaan, miten neste tulee käyttäytymään. Karkeimmissa resoluutioissa Bifrostin simulointi pysyi eheänä. Se tuotti paksun oloista nestettä, joka kuitenkin ilmestyi oletetusti emitteristä, eikä hävinnyt tai turvonnut samalla tavalla, kuten Blenderissä alhaisen resoluution simulaatiot.



Kuva 12 Kuvakaappaus Blenderin simulaatiosta resoluutiolla 50. Resoluutio on liian alhainen, jotta simulaatio olisi eheä ja toimisi oletetusti. Vasemmalla ylhäällä ensimmäinen kuvaruutu ja seuraavat kuvaruudut ovat kymmenen välein: 11, 21, 31...

Blenderissä alhaisilla resoluutioilla pitää olla tarkkana, ettei nesteen lähde ole liian lähellä määrittelyalueen reunaa. Jos neste leikkaa määrittelyalueen reunaa, simulaatio keskeytyy heti alkuun ilman selventävää virheviestiä. Alhaisella resoluutiolla nesteen tilavuus on suurempi kuin korkeilla resoluutiolla, ja näin ollen se saattaa leikata määrittelyalueen reunaa, vaikka tarkemmalla resoluutiolla neste mahtuisikin määrittelyalueelle. Tässä kokeessa Blenderillä ei pystynyt simuloimaan testiskeneä resoluutiolla 12, koska neste oli simuloitessa liian suuri määrittelyalueelle.

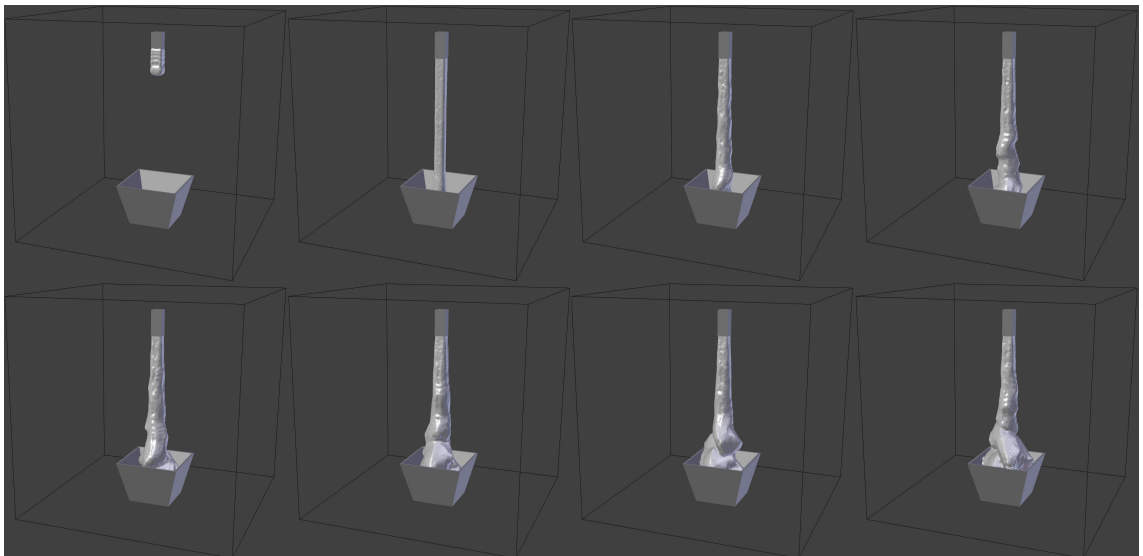
Bifrostissa oli yllättävää, että liian korkealla resoluutiolla simulaation tulos muuttui epäuskottavaksi, pinta näytti rikkonaiselta ja kulmikkaalta (kuva X). Testeissä tämän kaltainen tulos tuli *Master Voxel Size*-arvolla 0,05. Tällä arvolla Mayan työskentelyikkuna muuttui simulaation jälkeen niin hitaaksi, että sillä oli vaikea tehdä enään mitään.



Kuva 13 Lähikuva Bifrostin simulaatiosta *Master Voxel Size*-arvolla 0,05. Simulaatio muuttui rikkonaiseksi ja teräväreunaiseksi, joka ei ollut toivottu lopputulos.

4.2 Viskositeetin laskeminen

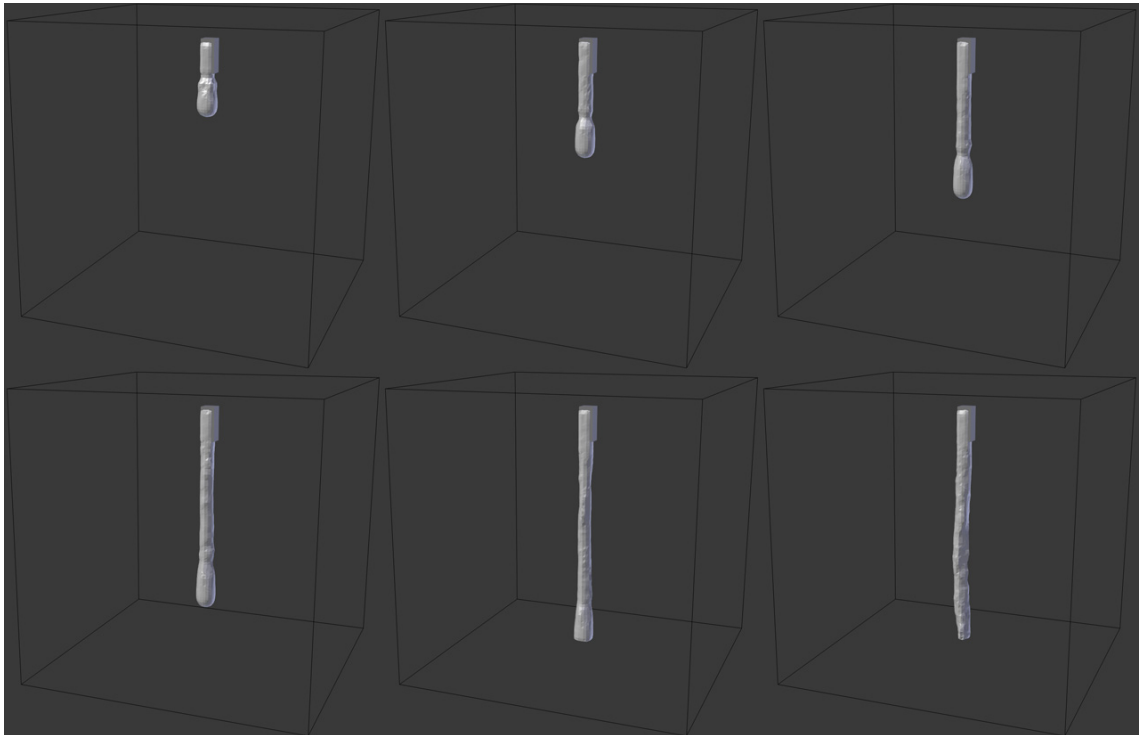
Testeissä Blenderin simulointiajat olivat oletettuja siinä mielessä, että viskositeetin laskeminen nosti pääsääntöisesti simulointiaikoja. Yllättävää oli, että molemmissa ääripäissä oli poikkeus. Eksponentin nostaminen kahdella antoi hivenen nopeamman simulointiajan kaikista juoksevimmalla ja sitkaimmalla asetuksella. Yleisesti huomionarvoista on, että kun viskositeetti laskee sen verran, että nesteen pinta ei pysy yhtenäisenä ja alkaa rikkoutua roiskeiksi, niin simulointiaika nousee merkittävästi.



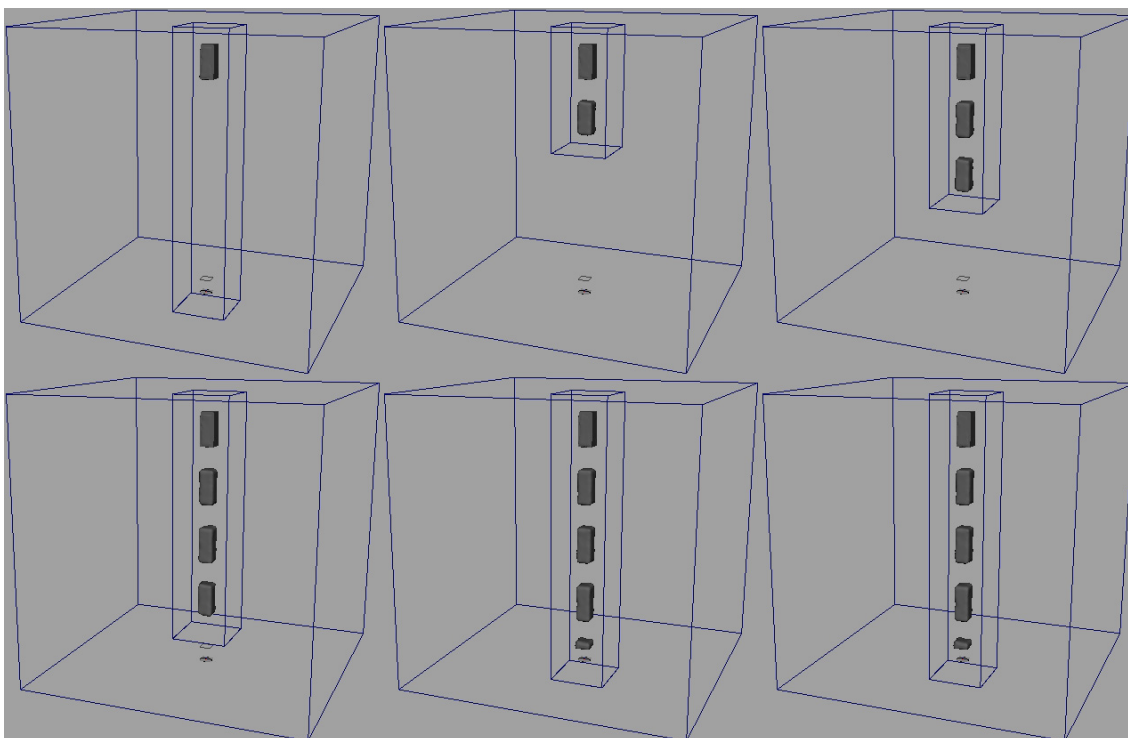
Kuva 14 Alhaisella viskositeetillä neste on paksua ja kasautuu itsensä päälle. Kuvasarjassa kuva-kaappaus Blenderin simulaatiosta viskositeetin eksponentilla 0. Vasemmalla ylhäällä ensimmäinen kuvaruutu ja seuraavat kuvaruudut ovat kymmenen välein: 11, 21, 31...

4.3 Nopeus

Nopeus vaikutti simulaatioaikaan yllättävän paljon. Varsinkin Blenderissä simulaatioajat kasvoivat odotettua enemmän nopeuden noustessa. Bifrost ei pystynyt simuloimaan nopeasti liikkuvaa nestettä, vaan nestevana muuttui kulmikkaaksi ja toistuvaksi, kunnes hajosi laatikkomaisiksi paloiksi (kuva X). Tämän vuoksi Bifrostissa simulaatioajat eivät kasvaneet samalla tavalla kuin Blenderissä. Blender piti massan yhtenäisenä suurissa nopeuksissakin. Bifrostin hajoavia simulaatioita olisi mahdollisesti voinut korjata säätämällä simulaation *Transport*- ja *Time Stepping*-attribuutteja, jotka tarkentavat simulaation laskentaa kuvaruutujen välillä, mutta nostavat simulointiaikaa (Autodesk 2014). En kuitenkaan lähtenyt selvittämään näitä attribuutteja tämän opinnäytetyön puitteissa.



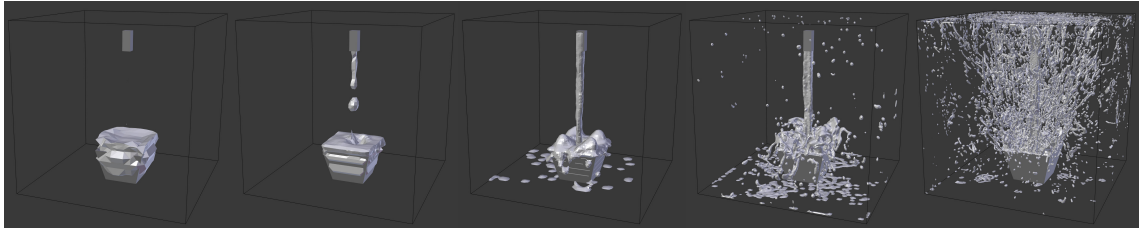
Kuva 15 Blenderissä simulaatio pysyy eheänä vauhdin kasvaessa. Vasemmalta ylhäältä viisi ensimmäistä kuvaruutua. Oikealla alhaalla on viimeinen kuvaruutu.



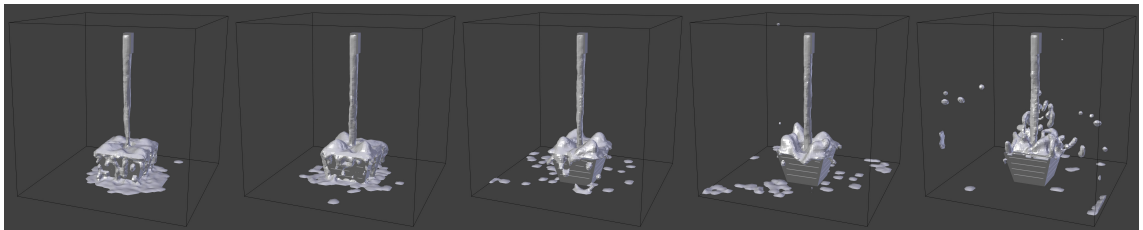
Kuva 16 Nopeissa vauhteissa Bifrostin simulaatio hajosi kulmikkaiksi laatikoiksi.

4.4 Suuri määrittelyalue

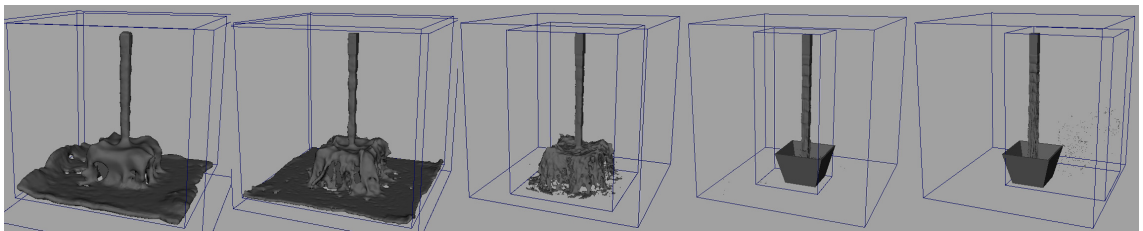
Määrittely alueen suurentamisesta saadut tulokset eivät olleet samalla tavalla selkeitä, kuten muiden testien tulokset. Kokeissa oli liikaa muuttujia, kuten nesteen nopeus ja resoluution suhde simuloitavan skenen kokoon. Jos nesteen lähtönopeutta ei muuttanut skeneä suurentaessa, niin se osui esteeseen myöhemmässä kuvaruudussa ja suurimmissa skeneissä neste ei ehtinyt osua esteeseen ollenkaan. Muuttaessa nesteen nopeutta skenen koon mukaan, niin neste osui suurin piirtein samassa kuvaruudussa esteeseen, mutta paine oli kovempi. Nesteen kovemman paineen takia roiskeita oli enemmän (kuvat X ja X) ja ne saattoivat vaikuttaa pidentämällä simulaatioaikoja. Roiskumista lukuun ottamatta skenen suurentaminen, niin että resoluutio kasvaa, antoi samankaltaisen tuloksen, kuin pelkkä resoluution suurentaminen. Simuloidessa suuria määrittelyalueita Bifrostin simulaatio muuttui myös käyttökelttomaksi kuten resoluution tarkentamistestissä (kuva X).



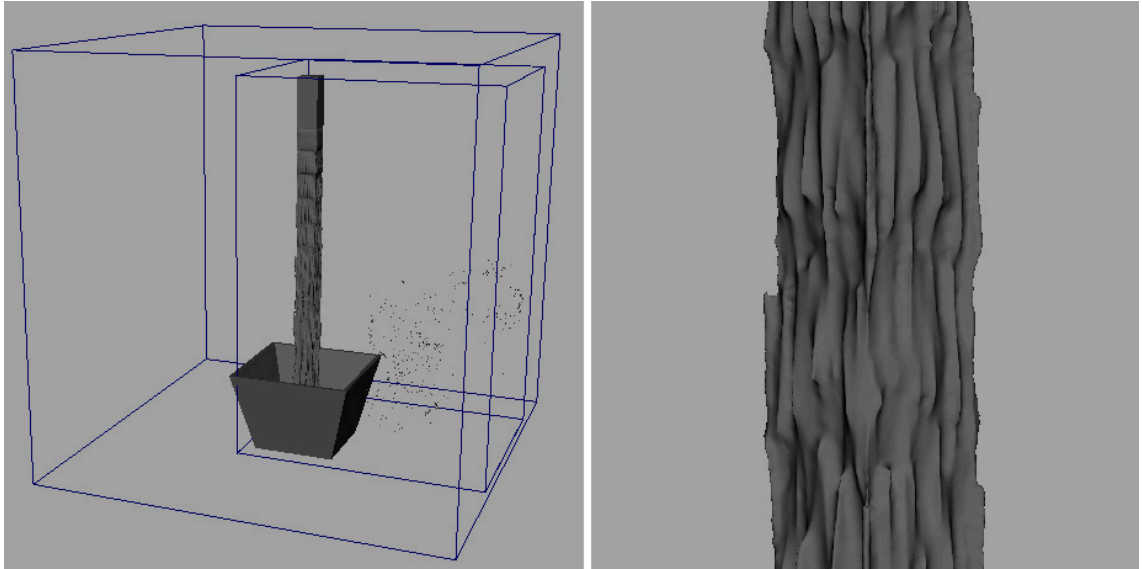
Kuva 17 Blenderissä skenen skaalaaminen ja resoluution muuttaminen vaikutti lähes samalla tavalla, kuin pelkkä resoluution muuttaminen. Nopeuden suhteuttaminen skenen kokoon vaikutti roiskeiden määrään. Jos nopeutta ei muuttanut, niin neste ei osunut esteeseen samassa kuvavuudessa.



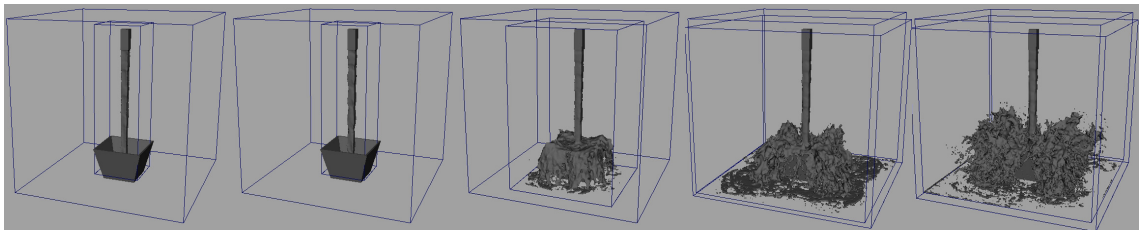
Kuva 18 Kun Blenderissä skaalaa skenen kokoa muuttamatta resoluutiota, niin simulaatiossa ainoastaan roiskeiden määrä kasvaa, koska nesteen nopeutta on jouduttu nostamaan.



Kuva 19 Bifrostissa määrittelyalueen resoluutio muuttuu oletuksena, jos simuloitavan skenen kokoa skaalataan. Tämä johtuu siitä, että *master voxel size*-attribuutti määrittää vokselin koon suhteutettuna skenen skaalaan. Bifrostissa skenen suurentaminen, muuttamatta *master voxel size*-attribuuttia, vaikutti samalla tavalla, kuin resoluution muuttaminen.



Kuva 20 Bifrostilla simuloitavan alueen kasvattaminen sai aikaan käyttökelvottoman tuloksen, kun skenen koko oli kaksi metriä kanttiinsa.



Kuva 21 Bifrostissa roiskeiden määrä kasvoi selkeästi, kun nopeus kaksinkertaistettiin skenen kanssa ja resoluutio pysyi samassa suhteessa skenen kokoon nähden.

5 Loppupäätelmät

Olen tietoinen, että opinnäytetyön testit eivät olleet tieteellisiä. Eikä se ollut testien tarkoituksaan, vaan niiden tehtävänä on olla suuntaa antavia ohjeita itselleni ja muille nestesimulaatioista kiinnostuneille joilla ei ole aikaisempaa kokemusta nestesimulaatioiden tekemisestä. Bifrostin osalta tulokset ovat vanhentuneita jo tämän loppupäätelmän kirjoittamisen aikana, koska Autodesk julkaisi Maya 2016-version keväällä 2015. On kuitenkin oletettavissa, että samat muuttujat vaikuttavat kaikissa tulevilla versioisakin, joten niitä voi soveltaa jatkossakin. Myös muillakin ohjelmistoilla pätee samat muuttujat, joten tietoja voi myös soveltaa muissakin ohjelmistoissakin kuin Blenderissä tai Mayassa.

Blenderin nestesimulaattori yllätti itseni positiivisesti opinnäytetyötä tehdessäni. Jos tekijällä on nestesimulaatioiden perusteet hallussa, niin sillä voi tehdä monipuolisesti pienen mittakaavan simulaatioita. Mutta vaikka opinnäytetyön testien puitteissa Blenderin nestesimulaattori pystyi samaan, tai parempaan tulokseen, kuin Mayan Bifrost, niin Bifrostissa on kuitenkin ominaisuuksia jotka puuttuvat Blenderistä. Näitä ominaisuuksia ovat muun muassa simulaatioista saatavat erilaiset värikartat, joilla voi jälkikäteen muokata nesteen eri elementtejä.

Itse petyin kuinka opinnäytetyössä käytetyn Autodeskin Maya 2015-version Bifrost oli monissa määrin keskeneräinen simulaattori. Seuraaviin versioihin on kuitenkin tulossa lisää ominaisuuksia ja huomattavia parannuksia (Seymour, Mike 2014). Maya 2016-version Bifrostissa on mukana muun muassa viskositeetin säätömahdollisuus, mahdollisuus tallentaa cache-tiedostoja pakatussa muodossa sekä savujen ja kaasujen kaltaisten aineiden simulointi (Autodesk 2015). Opinnäytetyötä tehdessä varsinkin Bifrostin cache-tiedostojen koot aiheutti ongelmia niiden viedessä yli 300 gigabittia kovalevytilaa. Päivitysten myötä Bifrost vaikuttaisi kehittyvän nopeasti omaan luokkaansa simulaattorina Blenderiin nähden.

Autodeskien julkaistessa niin keskeneräisenä Bifrost lähes neljän tuhannen euron ohjelmistossa, niin se kuitenkin muistutti, että kuluttajan pitää olla tarkkaavainen ostaessaan ohjelmistoja. Kannattaa ottaa selvää mitä ominaisuuksia ostettavilta ohjelmistolta vaatii ja täyttääkö ohjelmisto kyseiset vaatimukset. Varsinkin täysin uusien ominaisuuksien, kuten Bifrostin kohdalla, kehitys saattaa olla kesken ja virheet korvataan vasta seuraavaan täyteen versioon. Tämän kaltaiset ongelmat muuttuvat jonkin verran, kun Autodesk lopettaa pysyvien lisenssien myynnin tammikuussa 2016 ja myy jatkossa ainoastaan määräaikaista lisenssejä (Norris 2015).

On hienoa, että nestesimulaatioiden tekeminen on tullut nykyään mahdolliseksi suuremmalle määrälle 3D-artisteja tietokoneiden nopeuduttua ja 3D-ohjelmistojen kehittyessä. Simulaatioiden tekeminen on kuitenkin silti vielä vaativaa ja hidasta. Varsinkin suurten simulaatioiden kohdalla kannattaa miettiä vaihtoehtoisia ratkaisuja, tai mahdollisuutta pilkkoa simulaatio pienempiin osiin, joita on helpompi kontrolloida ja nopeampi simuloida uudelleen. Bifrostin myötä Autodeskin Mayasta on muodostumassa enemmän "kaikki yhdessä paketissa"-3D-ohjelmisto. Uskon 3D-generalistejen tekevän jatkossa enemmän simulaatioita Bifrostilla, mutta siitä ei ole syrjäyttämään varta vasten simulaatioihin kehitettyjä ohjelmistoja kuten Real Flow.

Lopuksi itselleni suurin hyöty oli opinnäytetyötä varten etsitty tieto, sekä se, että opin tekemään Blenderillä sekä Mayan Bifrostilla nestesimulaatioita. Samojen tietojen pohjalta on ollut helpompaa oppia uusien simulaattoreiden käyttöä ja tietoja on voinut soveltaa myös savusimulaatioihin, joissa periaatteet ovat hyvin samanlaisia.

Lähteet

Autodesk 2014. Autodesk Maya 2015 Help.

<<http://help.autodesk.com/view/MAYAUL/2015/ENU/>> (luettu 1.9.2015)

Autodesk 2015. Autodesk Maya 2016 Help.

<<http://help.autodesk.com/view/MAYAUL/2016/ENU/>> (luettu 8.9.2015)

Blender 2015. Doc:20.6/Manual/Physics/Fluid/Technical details. Wiki.blender.org.

<http://wiki.blender.org/index.php/Doc:2.6/Manual/Physics/Fluid/Technical_details> (luettu 22.1.2015)

Cgchannel 2014. Marcus Nordenstam discusses Bifrost. Cgchannel.com

<<http://www.cgchannel.com/2014/03/bifrosts-marcus-nordenstam-on-next-gen-fluid-sims/>>(luettu 22.1.2015)

Chen, Shiyi; Doolen, Gary D. 1998. Lattice Boltzmann Method For Fluid Flows.

<<http://venus.usc.edu/PAPERS/MultiScaleMechanics/annurev.fluid.30.1.329.pdf>> Annual Reviews Inc.

Chentanez, Nuttapong; Müller, Matthias 2011. Real-Time Eulerian Water Simulation

Using a Restricted Tall Cell Grid. <<http://www.cs.ucr.edu/~gupta/teaching/260-15/tallCells.pdf>> SIGGRAPH. NVIDIA PhysX Research.

Chrisman, Cameron 2008. A Comparison Of Grid-Based Techniques For Navier-

Stokes Fluid Simulation In Computer Graphics. San Diego. University Of California

Chung, T.J. 2010. Computational Fluid Dynamics. New York. Cambridge University Press.

Darrigol, Olivier 2002. Stability And Instability In Nineteenth-Century Fluid Mechanics.

Société Mathématique De France

Derakhshani, Dariush 2012. Introducing Autodesk Maya 2013. Sybex.

Dutt, Nidamarty; Ghosh, Tushar; Prasad, Dasika; Rani, Kalipatnapu; Viswanath, Dabir

2007. Viscosity of Liquids: Theory, Estimation, Experiment, and Data. Dordrecht. Springer

Ek, Joel; Englesson, Dan; Kilby, Joakim 2011. Fluid Simulation Using Implicit Particles: Advanced Game Programming.

Gad-El-Hak, Mohamed 1998. Fluid Mechanics from the Beginning to the Third Millen-

nium. Notre Dame. University of Notre Dame.

Gourlay, Michael J. 2012. Fluid Simulation for Video Games (part 1). Intel.

<<https://software.intel.com/en-us/articles/fluid-simulation-for-video-games-part-1/>> (luettu 4.9.2015)

Hess, D. Ronald 2012. Animating with Blender: How to Create Short Animations from Start to Finish. Burlington. Focal Press.

Kezsbom, Ian 1999. Visual Effects in Film and Television. New York. Binghamton University

Kuzmin, Dmitri 2010. A Guide to Numerical Methods for Transport Equations. Erlangen-NürBerg. Friedrich-Alexander-Universität.

Mullen, Tony 2013. Mastering Blender - Second Edition. Indianapolis. John Wiley & Sons, Inc.

Norris, Michael 2015. Autodesk Licensing Changes - What You Need to Know. SoftwareONE. <blog.softwareone.com/autodesk-licensing-changes-what-you-need-to-know> (luettu 11.9.2015)

O'Brien, James F.; Hodgins, Jessica K. 1995. Dynamic Simulation of Splashing Fluids. Atlanta. Georgia Institute of Technology

Realflow 2015. RF2014IntroGuide. PFD-dokumentti.

Seymour, Mike 2011. The Science of Fluid Sims. FX Guide <<http://www.fxguide.com/featured/the-science-of-fluid-sims/>> (luettu 22.1.2015)

Seymour, Mike 2014. Bifrost: Exclusive first in-depth look. FX Guide. www.fxguide.com/featured/bifrost-exclusive-first-in-depth-look/ (luettu 11.9.2015)

Sherrod, Allen 2008. Game Graphics Programming. Boston. Course Technology / Cengage Learning

Stam, Joe 1999. Stable Fluids. SIGGRAPH Annual Conference Series. <<http://www.autodeskresearch.com/publications/stablefluid>>

Toxik.sk 2015. Maya History. <<http://www.toxik.sk/maya-startup-window-history/>> (luettu 8.5.2015)

Yaeger, Larry 2002. A Brief, Early History of Computer Graphics in Film <http://shinyverse.org/larryy/cgi.html> (luettu 15.4.2015)

Zikanov, Oleg 2010. Essential Computational Fluid Dynamics. New Jersey. John Wiley & Sons. Inc.

Liite 1. Termistöä

Seuraavassa on selitetty lyhyesti keskeisiä termejä joita olen käyttänyt opinnäytetyössäni. Mukana on englannin- ja suomenkielinen sana ja alleviivattuna on opinnäytetyössä käytetty versio.

Scene – Skene, paikka, tapahtumapaikka. 3D-ohjelmistossa virtuaalinen 3D työskentelytila jossa esimerkiksi mallinnetaan.

Domain – Määrittelyalue. On suorakulmainen särmiö, jonka sisällä simulaatio tapahtuu.

Bake – Leipoa. Tämän opinnäytetyön yhteydessä leipomisella tarkoitetaan komentoa, jolla simulaatio lasketaan välimuistiin (katso cache - välimuisti).

Cache – Välimuisti. Väliaikainen tila kovalevyllä jonne esimerkiksi simulaatiot tallentuvat. Cachen voi tallentaa jolloin se voidaan ladata skeneen jälkikäteen niin, että simulaatiota ei tarvitse laskea uudelleen.

Emitter – Lähetin, säteilijä. Blenderissä ”inflow”. Lähde josta neste ilmestyy skeneen.

Obstacle – Este. Nestesimulaatiossa objekteja voi määrittää esteiksi joihin simulaatio törmää.

Voxel - Vokseli. Kolmiulotteinen pikseli.