Opinnäytetyö (AMK)

Tietotekniikka

NTIETS13P

2017

Heikki Wendelin

3D-SISÄLLÖN TUOTTAMINEN LAITURIN SUUNNITTELUSOVELLUKSEEN

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma

Ohjaaja: yliopettaja Mika Luimula, dos.

2017 | 37 sivua

Heikki Wendelin

3D-SISÄLLÖN TUOTTAMINEN LAITURIN SUUNNITTELUSOVELLUKSEEN

Tutkimuksen tavoitteena oli suunnitella käyttöliittymä myyntikonfiguraattorina toimivalle verkkosovellukselle, ja tuottaa sovellukseen vaadittava 3D-sisältö. Sovelluksen tuli toimia alustana, jolla potentiaaliset asiakkaat voisivat selata laiturinvalmistajien valikoimaa ja tutustua erilaisiin laituriyhdistelmiin sovelluksen realistisessa saaristomaisemassa. Opinnäytetyön toimeksiantajana toimi Virtuaalisen kone- ja prosessisuunnittelun yritys Premode Oy.

Työssä tutkittiin kehitystyön tueksi tavanomaisen 3D-peligrafiikan kehitysprosessin vaiheita. Lisäksi perusteltiin, miksi Premode päätti hyödyntää työssä käytettyjä ohjelmistoja. Tämä suoritettiin vertailemalla suosituimpia pelimoottoreita ja 3D-mallinnuohjelmistoja. Työ käsittelee seuraavia aiheita: käyttöliittymän suunnittelu, 3D-materiaalin käsittely mallinnus-ohjelmistossa ja pelimoottori Unityn työkalut.

Tuloksena projektista saatiin Unity pelimoottoriin tuodut ja valaisukartoitetut 3D-mallit. Mallinnettu ympäristö tuotiin eloon Unityssä kehitetyllä valaistuksella ja saaristoympäristöä imitoivilla efekteillä. Sovellus oli tässä vaiheessa projektia valmis ohjelmoitavaksi ja se siirtyi toisen henkilön viimeisteltäväksi.

ASIASANAT:

3D-mallinnus, Pelimoottori

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree Programme in Information Technology

supervisor: Principal Lecturer Mika Luimula, Adj.Prof.

2017 | 37 pages

Heikki Wendelin

CREATING 3D-CONTENT FOR A DOCK DESIGNING APPLICATION

The subject of this thesis is the production of 3D- graphics for a web application. This thesis will also expand to cover the planning of a user interface, modelling in 3Dsoftware and exploring the tools in Unity 5 game engine. The thesis was written for Premode Oy, a company specializing in virtual prototyping. The graphical work was commissioned to be the foundation for an upcoming product.

This thesis first covers the theory of 3D modelling and the concept of game engines. Then it follows the development process from designing the user interface and 3D modelling in 3D Studio Max modelling software, to embellishing the results in Unity 5 game engine.

The finished product is an application intended for designing docks. The finished application helps demonstrating what the real-life counterparts of the designed docks would look like. The application was made to run on the webpages of dock builders.

KEYWORDS:

3D-modelling, game engine

SISÄLTÖ

KÄYTETYT LYHENTEET JA SANASTO

Blueprint	Pelimoottori Unreal Enginestä löytyvä visuaalinen ohjelmointikieli.
Kuvataajuus	Näytölle sekunnissa piirrettyjen kuvien määrä.
NURBS	Non-Uniformal Rational B-Spline on yksi kolmesta käytetyimmästä mallinnuksen muodosta.
Polygonibudjetti	Näkyvissä olevien polygonien enimmäismäärä, kuvataajuuden pysyessä hyväksyttävänä.
Renderöinti	Kuvan luominen mallista tietokoneohjelman avulla. Malli on datasta muodostuva ohjelmallinen kuvaus usein kolmeulotteisesta objektista.
Shader	Shader on ohjelma, jolla määrätään, miten tietokone renderöi kunkin pikselin. Shadereita käytetään useimmiten säätelemään valaistusta ja varjostusta.
Tekseli	Pikseli tekstuurikartassa.

1 JOHDANTO

Tämän opinnäytetyön aiheena oli 3D-sisällön tuottaminen verkkosovellukseen. Projektin ja opinnäytetyön toimeksiantajana toimi Premode Oy. Idea opinnäytetyöhön syntyi toimeksiantajan suunnitelmasta kehittää sovellus laiturien demonstroimiseen verkkosovelluksessa. Sovelluksen oli tarkoitus toimia myyntikonfiguraattorina eli alustana, jolla potentiaaliset asiakkaat voisivat selata laiturinvalmistajien valikoimaa ja tutustua erilaisiin laituriyhdistelmiin sovelluksen realistisessa saaristomaisemassa. Premode oli jo suorittanut kehitysideasta taustatutkimuksen, jonka tuloksena todettiin, ettei vastaavanlaista laiturinsuunnittelusovellusta ole vielä markkinoilla.

Suomessa on 20–30 laiturinvalmistajaa, joilla voisi olla kyseisenlaiselle sovellukselle kiinnostusta. Sovellus oli toimeksiantajalla vasta idean tasolla, joten opinnäytetyössäni suunnittelin ja toteutin projektin niin pitkälle, kuin sovelluksen suunnittelijana ja graafikkona oli mahdollista. Toisin sanoen, käytiin siis läpi vaiheet 3D-mallintamisesta mallien kokoonpanoon ja valottamiseen pelimoottori Unityssä. Kun sovelluksen sisältöä oli tarpeeksi, se myytiin promootioversiona Kuhalaiturit Oy:lle. Laiturityypit ja laiturien mitat on sovelluksen ensimmäistä versiota varten näin ollen otettu kyseisen yrityksen tietojen pohjalta.

Työssä tutkittiin kehitystyön tueksi tavanomaisen 3D-peligrafiikan kehitysprosessin vaiheita. Mielenkiintoisimpiin vaiheisiin kuuluivat tarkasti mallinnettujen 3D-mallien heijastaminen tekstuurina pienemmällä geometrialla mallinnetuille objekteille, UVW-kartoitus ja normal-kartan tekstuuriksi renderöinti (MDN Web Docs 2017.). Lisäksi työssä perusteltiin Premode Oy:n ohjelmistovalintoja, suorittamalla pelimoottorien- ja 3D-mallinnusohjelmistojen vertailu. Vertailu suoritettiin kolmen pelinkehityksessä suosituimman 3D-malliinnusohjelmiston ja pelimoottorin hyötyjä ja haittoja.

2 3D-MALLINNUS

3D-mallinnus on mallinnusohjelman käyttämistä fyysisiä objekteja vastaavien kolmeulotteisten mallien luomiseen. 3D-mallinnusta käytetään useilla eri aloilla. Elokuva-, televisio-, videopeli-, arkkitehtuuri-, rakennus-, tiede- ja lääketiedeteollisuus käyttävät kaikki 3D-malleja erilaisiin graafisiin käyttötarkoituksiin.

3D-mallinnusohjelmalla voidaan tuottaa malleja eri työkalujen ja lähestymistapojen avulla. Käytetyimpiä geometrian muokkaustapoja ovat polygonimallinnus, NURBS-mallinnus (Non-Uniformal Rational B-Spline) ja jakavamallinnus (subdivision modelling) (Autodesk 2017a.)

Polygonimallinnus tapahtuu muokkaamalla polygoneja ja niiden elementtejä. Polygonit ovat suorasivuisia kolmeulotteisten pisteiden (verteksi) ja niitä yhdistävien suorien viivojen (reunojen) määräämiä tasoja. Polygonitasot ovat laajasti sovellettavissa, ja ne ovat suosituin tapa kehittää animaatioefektejä elokuva- ja peliteollisuuden ja 3D-nettikehityksen parissa.

NURBS-mallinnus perustuu kurvien muokkaamiseen. Mallien muodot määräytyvät matemaattisten kaavojen mukaan. NURBS on hyödyllisin, kun mallinnetaan monimutkaisia kurveja vaativia sileäpintaisia kappaleita, kuten viinilasi tai jalkapallo. (Seraphinacorazza 2012.)

Jakavamallinnus ottaa piirteitä sekä polygoni- että NURBS-mallinnuksesta. NURBS-mallinnuksen tavoin, jakavassa mallinnuksessa voidaan luoda sileitä orgaanisia kappaleita varsin harvoja verteksejä käyttäen. Karkeat 3D-mallit saavat pehmeät muodot, jakamalla itseään polygonitasolla. Polygonimallinnuksen tavoin, jakavamallinnus mahdollistaa yksityiskohtien pursottamisen mallinnuksen aikana. (Zorin, D. 2006.)

2.1 Suosituimmat 3D-mallinnusohjelmistot

2.1.1 3D Studio Max

Työssä käytetty Autodesk 3D Studio Max on Autodeskin kehittämä ohjelmisto, joka on tunnetuimpia mallintamiseen, animointiin ja teksturointiin käytettyjä 3D-mallinnusohjelmapaketteja. Ohjelmistoa käyttävät erityisesti pelinkehittäjät ja mainos- ja arkkitehtuuristudiot.

3DS Max on tunnettu helppokäyttöisistä mallinnustyökaluistaan ja on tästä syystä hyvä ohjelma uusille käyttäjille. 3DS Maxin erityispiirteisiin kuuluvat sen erinomaiset valotus- ja renderöintitoiminnot. Se on erityisen suosittu arkkitehtuuri puolella.

3DS Maxin käyttö perustuu kuukausi- tai vuosimaksuihin. Kuukausittainen hinta 3DS Max 2017 versiolle on 200 euroa. Lisenssejä saa myös yhdeksi, kahdeksi tai kolmeksi vuodeksi. Ohjelma on tarkoitettu ammattilaiskäyttöön, joten se on melko kallis vaihtoehto yksittäisille käyttäjille tai pienille firmoille. Tutustumiskäyttöön ohjelmasta on myös ilmainen 30:n päivän kokeiluversio, sekä opiskelijoille kolmen vuoden opiskelijalisenssi. (Autodesk 2017b.)

2.1.2 Maya LT

Maya on toinen laajalti käytetty Autodeskin kehittämä applikaatio. Maya LT on kevennetty versio Mayan standardiapplikaatiosta. Maya LT on tarkoitettu nimenomaan pelinkehitykseen, joten se sisältää vain pelinkehitykseen olennaiset työkalut. Maya on kiivennyt 3D Studio Maxin tasolle mallintamisen suhteen, mutta sen työnkulku eroaa Maxista, ja siihen saattaa olla hieman haastavampi perehtyä. Maya LT:n paras puoli on epäilemättä sen hinta. Maya LT maksaa vain 36 euroa kuukaudessa tai 300 euroa vuodessa. (Pluralsight 2015.)

2.1.3 Blender

Blender on Blender organisaation kehittämä avoimen lähdekoodin mallinnusohjelmisto. Blenderin työnkulun on kuvailtu olevan ensin haastavaa ja myöhemmin todella intuitiivista. Blender on erinomainen valinta, jos sen käyttötarkoituksena on nimenomaan 3D-peligrafiikan kehitys. Blenderillä on laaja käyttäjäkunta, johon voi tukeutua. Ohjelmisto perustuu avoimeen lähdekoodiin, joten edistynyt käyttäjä voi muokata ohjelmaa mielensä mukaiseksi. Blender on täysin ilmainen 3D-mallinnusohjelmisto. (Pluralsight 2015.)

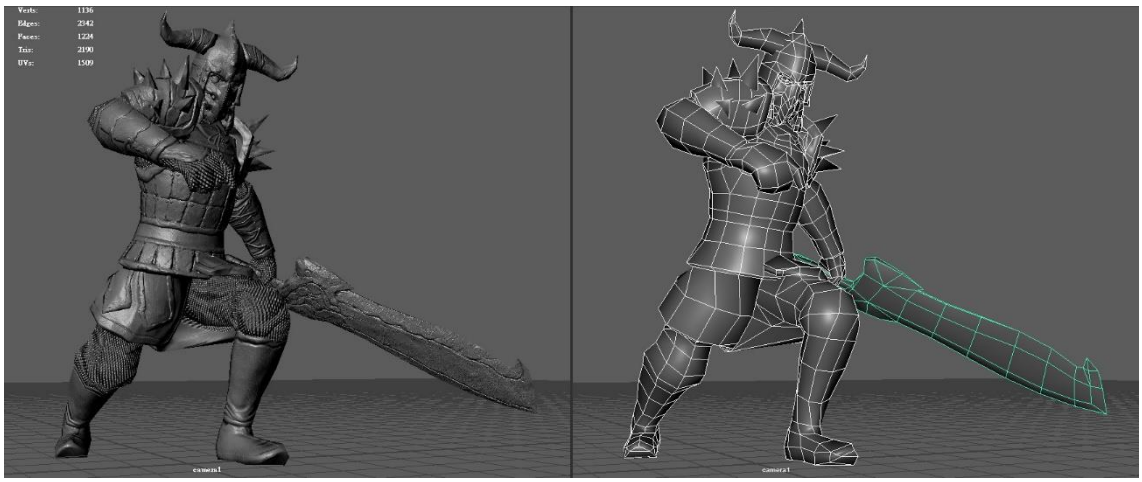
Taulukko 1. 3D-mallinnusohjelmien vertailu.

	3D Studio Max	Maya LT	Blender
Hinta	200 euroa kuukaudessa. Lisenssejä on myös kahdeksi ja kolmeksi vuodeksi.	36 euroa kuukaudessa tai 300 euroa vuodessa.	Ilmainen
Riippumattomuus	Suljettu lähdekoodi	Suljettu lähdekoodi	Avoin lähdekoodi
Helppokäyttöisyys	Tunnettu helppokäyttöisistä mallinnustyökaluistaan ja on tästä syystä hyvä ohjelma uusille käyttäjille.	Työnkulku eroaa Maxista ja siihen saattaa olla hieman haastavampi perehtyä.	Työnkulun on kuvailtu olevan ensin haastavaa ja myöhemmin todella intuitiivista.

3DS Max, kuten taulukossa 1 näkyy, on ylivoimaisesti näistä 3D-mallinnusohjelmistoista hintavin. Tästä huolimatta on se edelleen suurempien yritysten eniten suosima mallinnusohjelmisto. 3DS Maxin helpon käytettävyyden ja ammattimaisen lopputuloksen suhde, sekä sen tuttu brändi, toimivat vielä vahvoina valttikortteina. Premode omisti kertamaksullisen vuoden 2013 version 3DS Maxista. Tästä syystä huolimatta yrityksen pienestä koosta, mallintaminen suoritettiin kyseisellä mallinnusohjelmistolla.

2.2 Pelien 3D-grafiikan työnkulku

Ennen mallintamisen aloittamista on hyvä hakea referenssikuvia mallinnettavasta kohteesta. Myös omat luonnokset toimivat hyvin. Jokaista 3D-objektia varten tulee mallintaa kaksi versiota: Low poly-mallinnos ja high poly-mallinnos. (Kuva 1.)



Kuva 1. High poly ja low poly. (Suen, P. 2015.)

2.2.1 Low poly ja high poly -mallinnus

High poly on yksityiskohtainen versio 3D-objektista, eli sen polygonien määrä on korkeampi. Low poly-versiota ei mallinneta yhtä tarkasti, ja siinä on vähemmän polygoneja.

Renderöintitapa ja käytetty pelialusta määräävät pelin polygonibudjetin (Wikipedia 2017.). Toivotun polygonibudjetin ja kuvataajuuden (Robet, T. 2013.) saavuttamiseksi, low poly-mallinnoksia käytetään lähinnä tietokonepeleissä ja muissa ohjelmistoissa, joissa manipuloidaan 3D-objekteja reaaliajassa.

Animoidut elokuvat ja tietokoneella renderöidyt kuvat käyttävät high poly-malleja. Niillä on korkeampi polygonibudjetti, koska renderöintiä ei tarvitse suorittaa korkeampaa

kuvataajuutta vaativassa reaaliajassa. Lisäksi animoitujen elokuvien kehityksessä hyödynnetään laajaa tietokoneiden verkostoa, joten käytetyt prosessointitehot ovat pelinkehitystä paremmat. (AUGI 2014; Wikipedia 2017.)

2.2.2 Teksturointi ja UVW-kartoitus 3DS Maxissa

UVW-kartoituksella tarkoitetaan teksturointiprosessia, missä 2D-tekstuuri kartoitetaan 3D-objektille. 3DS Maxissa tämä suoritetaan UVW Unwrap-työkalulla. UVW Unwrapissä 3D-objektin polygoniverkko avataan kaksiulotteiseksi tasoksi, jolle tekstuuri voidaan asetella tai maalata. Asetteluprosessin helpottamiseksi on UVW-editorissa mahdollista valita käytetty tekstuuri polygoniverkon taustakuvaksi.

Tekstuurit on mahdollista maalata suoraan 3DS Maxissa, mutta yleisesti tekstuurit luodaan parempaa jälkeä tuottavilla Adobe Photoshopin kaltaisilla, ulkoisilla kuvankäsittelyohjelmilla. Pelinkehityksen tärkeimpiä tekstuurityyppejä ovat diffuse-, specular- ja normal-kartat.

Diffuse-kartta on kaikkein yleisin teksturointimenetelmä. Siinä ympäröidään 3D-objekti kuvalla, joka säilyttää samalla alkuperäisten pikseliensä värit. Skannattuja kuvia tai digitaalikameralla otettuja kuvia voidaan käyttää diffuse-karttoina tuomaan 3D-mallille fotorealismia. (Reallusion 2017.)

Bump-kartat käyttävät kuvan harmaanväriskaalan arvoja luomaan variaatioita 3D-objektin pinnan varjostuksessa. Bump-kartalla voidaan lisätä 3D-malliin yksityiskohtaisuutta, kasvattamatta objektin polygonien määrää. (Reallusion 2017.)

Normal-kartta on bump-kartoituksen yleisimmin käytetty muoto. Normal-karttojen tavanomaisin käyttötapa on huomattavasti parantaa low poly-mallin yksityiskohtaisuutta luomalla normal-kartta vastaavasta high poly-mallista. (Reallusion 2017.)

2.2.3 Normal-kartan tekstuuriksi renderöinti 3DS Maxissa (Baking)

Normal-kartan tekstuuriksi renderöinnillä tarkoitetaan prosessia, jossa luodaan tekstuurikartta high poly-mallista, ja annetaan kyseinen tekstuuri low poly-mallille. Tähän viitataan usein myös baking-prosessina.

High poly-malli tulee asetella siten, että se peittää kokonaan low poly-mallin. Mallien tulee pysyä mahdollisimman tarkasti toistensa päällä. Tämän jälkeen low poly-mallille tulee antaa Projection-modifikaattori. Projection-modifikaattorin asetuksista Reference Geometry kohdan alta valitaan mallinnettu high poly-malli. Aktivoimalla Projection-asetuksista Shaded-valintaruudun, tulee harmaanvihreällä värillä näkyviin low poly- ja high poly-mallien päällekkäisyydet. Tämän saa korjattua skaalaamalla low poly-mallia. Mallia skaalataan nostamalla Shaded valintaruudun alla olevaa Push Amount arvoa.

Tämän jälkeen siirrytään muokkaamaan RenderToTexture-toiminnon asetuksia. Ensin valitaan low poly-malli ja avataan RenderToTexture (pikanäppäin 0 tai Rendering > RenderToTexure). Tärkeintä on katsoa, että Projection Mapping-asetuksissa on valittuna Projection. Lisäksi on määriteltävä, minne lopputuloksena toivotut kartat tallennetaan.

2.2.4 Tiedostoformaatin valinta

On kaksi tapaa siirtää tiedostoja 3D-mallinnusohjelmistosta pelimoottoriin: yleiset 3D-formaatit ja alkuperäisformaatit. Nämä formaattityypit vaikuttavat rajusti siihen, miten niissä tallennettuja 3D-malleja voidaan hyödyntää.

Alkuperäisformaatit, kuten .max ja .blend, Blender ja 3D Studio Max, antavat kehittäjälle mahdollisuuden muokata mallia suoraan pelimoottorissa. Sovellus tulee toimimaan kuitenkin vain koneilla, joille on asennettu lisensoitu kopio käytetystä mallinnusohjelmistosta. (Unity3D 2017a.)

Yleiset 3D-formaatit, kuten .fbx tai .obj, antavat kehittäjälle mahdollisuuden muokata tiedostoa laajassa valikoimassa eri mallinnusohjelmistoja. Yleistä 3D-formaattia käyttämällä on mahdollista tallentaa vain tarvittu osa koko 3D-mallia. Tästä syystä tähän muotoon tallennetut tiedostot vievät usein alkuperäisformaatissa tallennettuja tiedostoja vähemmän tilaa. Yleinen 3D-formaatti on lisäksi ainoa tapa käyttää vähemmän käytetyissä 3D-mallinnusohjelmistoissa kehitettyjä objekteja, joiden omia tiedostoformaatteja pelimoottorit eivät välttämättä suoraan tue. (Unity3D 2017a.)

Yleisen 3D-formaatin heikkous on, ettei siinä tallennettuja tiedostoja ole mahdollista suoraan muokata pelimoottorissa. Tiedosto on uudelleen muunnettava yleiseen 3D-formaattiin käytetyssä mallinnusohjelmistossa, minkä jälkeen vanhan 3D-mallin voi korvata uudella tiedostolla. (Unity3D 2017a.)

3 PELIMOOTTORI

Pelimoottori on videopelien luontiin ja kehitykseen tarkoitettu ohjelmisto. Pelimoottorit tarjoavat uudelleenkäytettäviä komponentteja ja apuvälineitä, joita muokkaamalla pelin saa herätettyä eloon. Lataaminen, esillepano, animoidut mallinnukset, törmäyksen tunnistus esineiden välillä, fysiikka, syöte, graafinen käyttöliittymä ja jopa osa pelin käyttämästä tekoälystä löytyy osana komponentteja, jotka muodostavat pelimoottorin. Pelimoottoreiden tärkein hyöty on kuitenkin mahdollisuus sovittaa peli helposti monelle eri pelialustalle konsoleista tietokoneisiin.

3.1 Suosituimmat pelimoottorit

3.1.1 Unity 5

Unity 5 pelimoottori on Unity Technologiesin kehittämä pelimoottori. Unityn ilmaisversio on monipuolisin, täysin ilmainen pelimoottori markkinoilla. Unity tukee lähes kaikkia pelialustoja, ja sen käyttäjäkunta on suuri. Unityn sisäisessä verkkokaupassa, Unity Asset Storessa, on tarjolla runsaasti ilmaisia 3D-malleja ja muita resursseja. Unity on aloittelijaystävällinen pelimoottori, mutta sen työkalut ovat varsin rajalliset.

3.1.2 Unreal Engine 4

Unreal Engine 4 on Epic Games Oy:n kehittämä huippusuosittu pelimoottori. Unreal Enginen viimeisin versio sallii ennen liian raskaina pidettyjen graafisen ominaisuuksien hyödyntämistä mobiilialustoilla. Globaalivalotus, bump-kartat, heijastukset ja reaaliajassa suoritetut animaatiot kuuluvat kaikki kyseisiin ominaisuuksiin. Unreal Enginellä on mahdollista luoda graafisesti erittäin vaikuttavaa jälkeä verratessa sen suurimpaan kilpailijaan Unity 5:een. Unreal Enginen suurimpana heikkoutena voidaan pitää sen jyrkkää oppimiskäyrää. Pelimoottori on ilmainen, mutta Epic Games ottaa palkkionaan 5% sillä tuotettujen pelien ja projektien tuotoista. (Matan, A. 2017.)

UE4 sisältää uniikin graafisesti esitetyn koodikielen, jonka nimi on Blueprint (Epic Games 2017.). Blueprint perustuu koodielementtien yhdistelyyn visuaalisesti. Tämä antaa koko kehitystiimille peruskäsityksen siitä, mitä koodissa tapahtuu.

3.1.3 GameMaker Studio 2

Game Maker on YoYo Games yrityksen kehittämä pelimoottori. Game Makerin käyttö ei vaadi aikaisempaa koodauskokemusta. Tämän vuoksi kehittäjät voivat nopeuttaa pelinkehitysprosessia. Huomattava heikkous tällä pelimoottorilla on, että sen tarjoamat kehitystyökalut ovat varsin heikot. Game Makerin lisenssit ovat kertamaksullisia ja vaihtelevat 100:sta 400:aan euroa. (Aspis, M. 2017.)

Taulukko 2. Pelimoottorien vertailu.

	Unity 5	Unreal Engine 4	Game Maker Studio 2
Hinta	Perusversio on ilmainen.	Pelimoottori on ilmainen, mutta Epic Games ottaa palkkionaan 5% sillä tuotettujen pelien ja projektien tuotoista.	Lisenssit ovat kertamaksullisia ja vaihtelevat 100:sta 400:aan euroa.
Riippumattomuus	Avoimen lähdekoodin lisenssistä voi neuvotella Unity Technologies yrityksen kanssa.	Avoin lähdekoodi	Suljettu lähdekoodi
Graafiset kyvyt	Unity ei yllä Unreal Enginen tasolle graaffisessa kilpailussa. Yksinkertaisuutensa vuoksi se on kuitenkin edelleen suositeltu pelimoottori mobiilialustoille.	UE4 on esitellyistä pelimoottoreista tehokkain.	GameMaker on kilpailukykyinen mobiili- ja 2D-pelimoottori. 3D-toimintojen ja valmiina ladattavan sisällön puute kuitenkin heikentävät sen suosiota.
Helppokäyttöisyys	Unity on 3D-pelimoottoriksi yksinkertainen oppia.	UE4:n uniikki koodikieli Blueprint on visuaalisen perustansa vuoksi helppo koodikieli alkajalle. Yleisemmällä tasolla UE4 on näistä pelimoottoreista haastavin oppia.	GameMaker ei vaadi lainkaan koodausta. Pääasiallisesti 2D-pelimoottorina GameMaker on esitellyistä pelimoottoreista aloittelijaystävällisin.

Unity 5 oli Premode Oy:n käyttämä pelimoottori. Tämä pelimoottori oli siis ainoa järkevä valinta tähän projektiin, jotta taattiin jatkossa sovelluksen helppo ylläpidettävyys. Premode suosii Unityä, koska sen ilmaisversio kattaa kaikki yrityksen tarpeet. Myös Unityn lyömätön, julkaisualustojen määrä, paransi sen asemaa verrattuna muihin Pelimoottoreihin. Loppupelissä todellinen syy Unityn valintaan oli kuitenkin ohjelmien, talukossa 2 näkyvät, hintaerot.

3.2 Ohjesääntöjä pelien optimoimiseen

Pelien optimointiin vaikuttavat useat tekijät. Graafikon osalta optimoinnissa kannattaa erityisesti huomioida polygonibudjetti ja käytettyjen materiaalien ja luiden (bones) määrä. Unityssä on hyvä pitää mielessä myös käytettyjen Skinned Mesh Rendererien määrä.

Skinned Mesh Renderer on luuanimaatioiden (bone animations) renderöintiin kehitetty Unityn komponentti. Sillä voi väännellä 3D-objektien muotoja etukäteen määrättyjen animaatioiden mukaisesti. Skinned Mesh Renderer on hyödyllinen animoidessa hahmoja ja objekteja, joiden nivelet taipuvat (Toisin kuin esimerkiksi koneet, joiden nivelet ovat saranamaisempia). Kunkin hahmon animoinnissa tulisi käyttää vain yhtä Skinned Mesh Renderiä. Kahden Skinned Mesh Rendererin käyttö karkeasti kaksinkertaistaisi renderöintiin kuluvan ajan.

Pelin polygonibudjetti riippuu kohdealustasta. Mobiililaitteilla jokaisella objektilla kannattaa pyrkiä 300:n ja 1500:n polygonin välimaastoon. Pöytäkoneilla objektien polygonien määrät kannattaa pitää 1500:n ja 4000:n polygonin välissä.

Polygonien määrä kannattaa pitää alhaisempana, jos pelissä on saman aikaisesti näkyvissä paljon hahmoja. Yksittäisille objekteille riittää lähes aina maksimissaan 3 materiaalia. Luita tyypillisissä tietokonepeleissä on 15–60. Kolmekymmentä luuta on suositeltu maksimimäärä mobiilialustoille. (Unity3D 2017b.)

3.3 Valaisukartoitus (Unity 5)

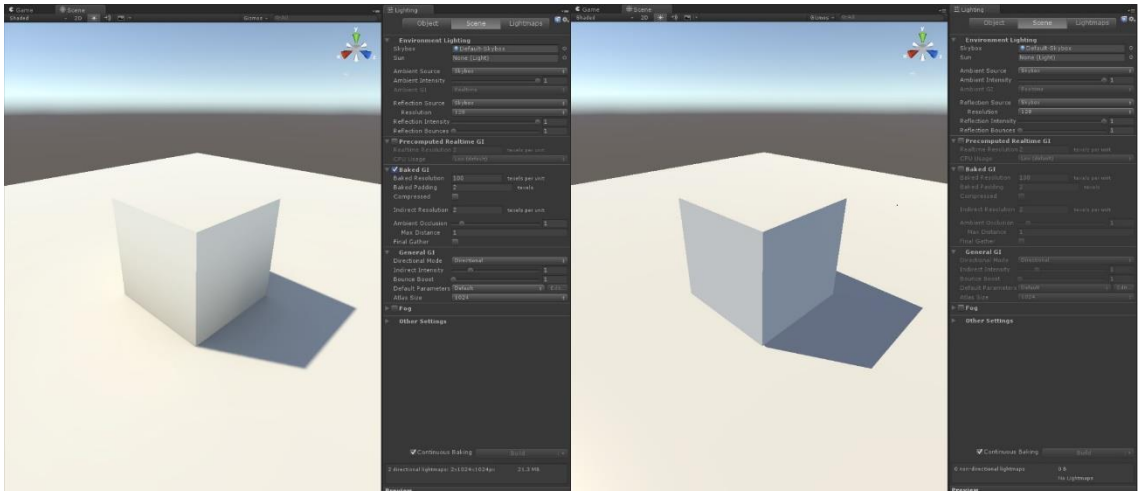
Valaisukartoituksella tarkoitetaan prosessia, jossa reaaliaikavalo korvataan etukäteen lasketulla tekstureihin upotetulla valotuksella. Valaisukartoitettavien objektien on oltava staattisia, koska valaisukartoitus ei tapahdu reaaliajassa. Valaisukartoitus mahdollistaa laajan valotuksen suhteellisen pienellä prosessointiteholla.

Valaisukartoitus Unityssä vaatii staattisia objekteja, jotka voidaan valokartoittaa. Valot, jotka on päätetty leipoa tekstuureihin, on myös määriteltävä ennen valokartoituksen aloittamista. Tärkein asia valaisukarttoja muokatessa on valaisukartoituksen resoluution tasapainottaminen suorituskyvyn kanssa. Valaisukartoituksen resoluutioon vaikuttavia Lighting-parametrejä (Kuva 3.) ovat Scale In Lightmap, Atlas Size ja luonnollisesti itse resoluutio asetus: Baked Resolution. (Tino 2015.)

Scale in Lightmap määrää, paljonko tilaa kukin objekti vie valotuskartalta. Tekselitiheyden (Mochochi, B. & Lahiri, K. & Cadambi, S. 2006.) esikatselutoiminnolla saa näkyviin valaisukartan resoluution.

Atlas Size määrää valaisukartan koon pikseleissä. Atlas Size ja Lightmap Resolution sijaitsevat Lighting > Scene valikon alla.

Lopullisessa valaisukartassa kannatta huomioida myös resoluutioparametrien alta löytyvä Ambient Occlusion-liukusäädin (Kuva 2.). Ambient Occlusion on ilmiö, jossa esineiden rajamaasto pimenee, kun esineiden pinnat lähestyvät toisiaan. Tämä antaa esineille massan vaikutelman. (Sassybot, Tino. 2015.)



Kuva 2. Vasemmalla kuvassa käytössä Ambient Occlusion. (Tino 2015.)

Kuva 3. Valaisukartan parametrejä. (Tino 2015.)

Progressive Lightmapper on Unity 5:n uusin erittäin hyödyllinen valaisukartoitus ominaisuus. Progressive Lightmapper mahdollistaa lopullisen valaisukartan raa'an version esikatselun. Tämä nopeuttaa valaisukartan iterointia, ja vähentää tarvetta jatkuvasti säädellä valaisukartan resoluutioasetuksia.

3.4 Ympäristövalaistus

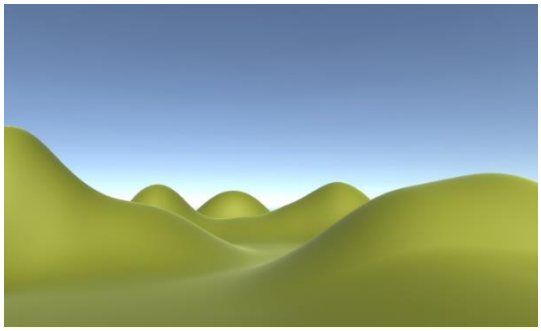
Unity 5:ssä saa helposti kelvollisen ulkovalaistuksen lisäämällä maisemaan Directional Light- valon. Tämä simuloi jo hyväksyttävästi auringon valoa. Todellisuudessa auringon valon pitää kuitenkin läpäistä ilmakehä, pilviä ja muita esteitä, joten tässä ulkovalaistuksessa on vielä parantamisen varaa.

Ympäristövalaistus (Ambient light) on tärkeä huomioida valaistusta luodessa. Ympäristövalaistus jätetään usein peleissä ja demoissa Unityn oletusharmaaseen väriin, mikä on huomattava menetetty mahdollisuus. Ympäristövalaistus on integraalinen osa tunnelman ja vuorokaudenajan luomisessa. Esimerkiksi, jos asetetaan päivävalaistuksen lähdevalo kellertäväksi ja ympäristövalaistus sinertäväksi, saadaan aikaan realistinen ulkoilma- peliympäristö. Öisiä peliympäristöjä luodessa ympäristövalaistukseksi on hyvä valita väri mustan ja tummansinisen väliltä. Sininen tässä tapauksessa simuloi hyvin kuunvaloa, kun taas musta ympäristövalaistus luo synkimmän vaikutelman ja toimii hyvin kauhutunnelmaa tavoitellessa. Mustaa ympäristövalaistusta kannattaa käyttää myös sisätilojen peliympäristöjä luodessa. Tämä antaa keinotekoisille- ja ikkunoista tulevalle valoille enemmän valtaa tunnelman luomisessa. (Runevision 2011.)

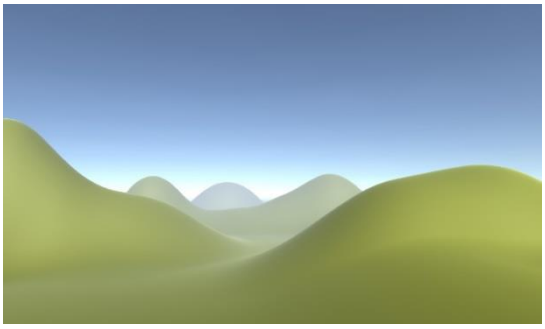
3.5 Sumuasetukset (fog)

Sumuasetus on helppo ja nopea tapa tuoda ulkoilmatiloihin enemmän tunnelmaa. Sumuasetusta ei yllättäen käytetä vain sumun luomiseen, vaan sillä voi myös tuoda avaruuden vaikutelmaa säätämällä asetukset todella alhaiseksi. Asetusta on siis hyvä käyttää aina, kun peliympäristössä on näkyvissä kuvien 4–6 tavoin kaukaisia kappaleita. Sumun väriasetusta on myös hyvä säätää peliympäristön valotuksen mukaisesti. Kuvassa 3 on käytetty vaaleansinertävää sumua, mutta vaikkapa auringonlaskussa on hyvä käyttää oranssin sävyjä. (Runevision 2011.)

Kuva 4. Ilmaperspektiivi.



Kuva 5. Ei sumuasetuksia.



Kuva 6. Sumuasetukset käytössä.

4 KÄYTTÖLIITTYMÄ

4.1 Käyttöliittymän määritelmä

Käyttöliittymä eli UI (User Interface) mahdollistaa käyttäjän ja laitteiston tai ohjelmiston välisen kommunikaation. Ohjelmiston käyttöliittymä, tai toiselta nimeltään graafinen käyttöliittymä, voi esimerkiksi sisältää ikkunoita, ikoneita, valikoita ja nappuloita, joiden avulla käyttäjä voi olla vuorovaikutuksessa ohjelmiston kanssa. Laitteistojen käyttöliittymänä taas saattaa toimia kauko-ohjain, peliohjain, hiiri tai näppäimistö. Suurin osa nykypäivän käyttöjärjestelmistä suunnitellaan käyttämään laitteiston ja ohjelmiston yhdistelmää.

Tietokone on tyypillinen esimerkki käyttöliittymästä, joka käyttää sekä laitetta että ohjelmistoa. Ohjauslaitteen käyttöjärjestelmä koostuisi tässä tapauksessa näppäimistöstä, hiirestä, näytöstä ja ohjelmiston näytölle piirtämästä graafisesta käyttöliittymästä.

4.2 Graafinen käyttöliittymä

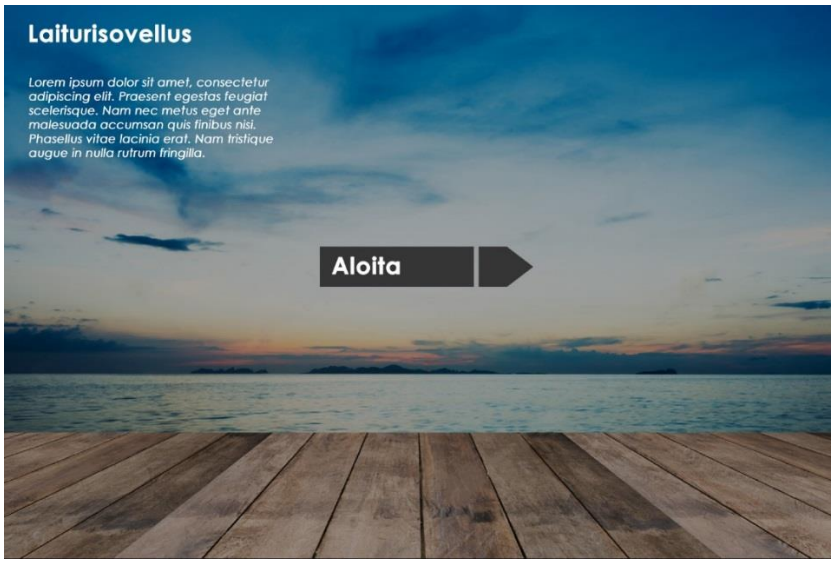
Graafinen käyttöliittymä eli GUI (Graphical User Interface) on yksi käyttöliittymän muoto. Termi otettiin käyttöön erottamaan graafinen käyttöliittymä tekstipohjaisista käyttöliittymistä. Ennen graafisia käyttöliittymiä, kommunikoitiin tietokoneiden kanssa tekstipohjaisten komentotyökalujen avulla. Komentotyökalujen sujuva käyttö vaati syötettävien komentojen ulkoa opettelua ja tietokoneiden palautteen tulkitsemiseen tarvittiin vahvaa osaamista. (Computer Hope 2017.)

5 GRAAFISEN KÄYTTÖLIITTYMÄN SUUNNITTELUOSUUS

Käyttöliittymän suunnittelu lähti harkitsemalla, mitä laiturin valmistuksessa olennaisia elementtejä applikaatioon tulisi liittää. Oli myös otettava huomioon, miten käyttöliittymä tulisi jakaa verkkosivun ja sovelluksen välillä. Käyttöliittymä päädyttiin pääasiallisesti toteuttamaan sovelluksen sisällä vapaampien kustomointimahdollisuuksien vuoksi.

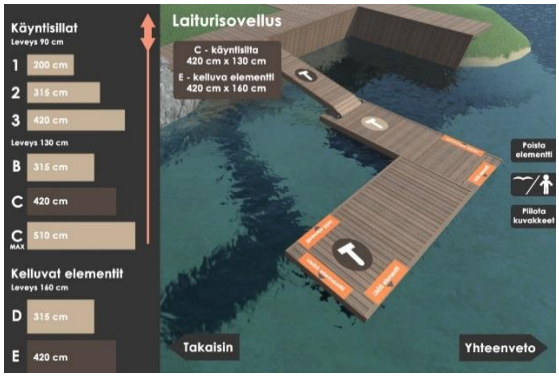
Laiturisovellukselle suunniteltiin etusivu, rannan valintasivu, Laitureiden muokkaus- ja yhteenvetosivut. (Kuva 7; Kuva 8; Kuva 9; Kuva 10; Kuva 11.)

Etusivun tehtävänä oli kertoa tietoja sovelluksesta. Rannan valintasivulle tuli kuvia sovellukseen mallinnettuja rantoja vastaavista todellisista rannoista. Rannat nimettiin kuvien päälle, ja koska käyttöliittymä haluttiin pitää mahdollisimman helposti navigoitavana, ei käyttäjälle annettu takaisin painikkeen ja ranta valintojen lisäksi muita vaihtoehtoja. Etusivulle ja rantavalikolle haettiin yhteinen taustakuva. Laadukkaimpia tekijänoikeusvapaita valokuvia saa helpoiten ostamalla Shutterstock nimisestä lisensoidun sisällön kaupankäyntiympäristöstä. (Shutterstock 2017.) Rantavalikon taustakuvaan lisättiin gaussian blur-filtteri, jotta käyttäjän huomio kiinnittyisi paremmin rantavalikkoihin.



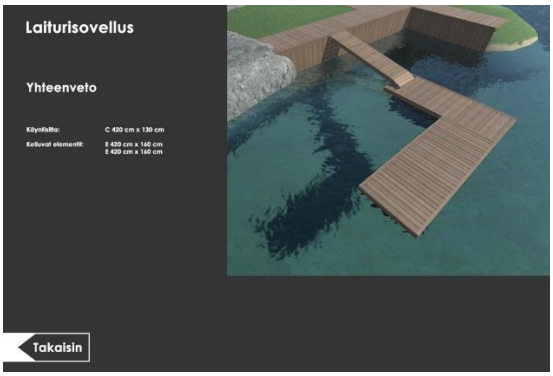
Kuva 7. Laiturisovelluksen etusivu.

Kuva 8. Rannan valintasivu.



Kuva 9. Laitureiden muokkaus-sivu.

Kuva 10. Käyttöliittymä on piilotettu.



Kuva 11. Yhteenvetosivu.

6 3D-MALLINNUSOSUUS JA SIINÄ HYÖDYNNETYT TYÖKALUT

Toimin projektin ainoana 3D-graafikkona. Projektin mallinnus osuudessa käytettiin Autodeskin 3DS Maxia.

6.1 Saaren mallinnus

Saaren (Kuva 12.) muotoilu aloitettiin litistämällä pallo sopiviin mittasuhteisiin. Pallolle annettiin varsin maltillisesti polygoneja, koska esimerkiksi hiekkaranta osio ei vaatinut niitä paljon. Pallosta poistettiin alapuoli, ja sen reunoja pursotettiin kaartuvasti ulospäin luomaan loivenevan meren pohjan. Tämän jälkeen kivikko-osion polygonimäärää kasvatettiin puolittamalla kivikoksi mallinnettavan alueen polygoneja muutamaan kertaan. Saaren keskiosan ruohikolle ja kivikko- ja hiekkarannoille annettiin materiaalitunnukset myöhempää teksturointia varten.

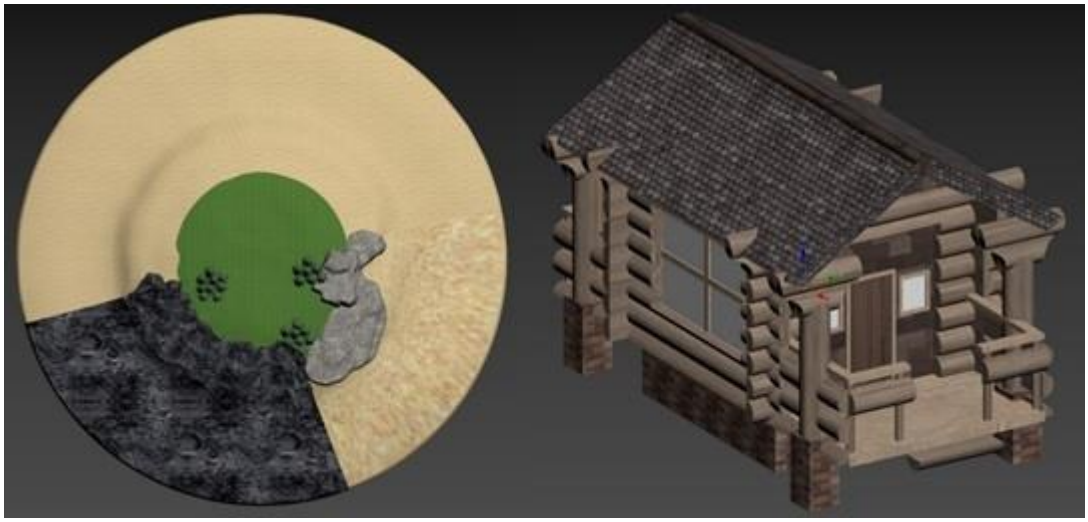
Kivikkorannan pohja muodostui valitsemalla sille varattu alue polygonivalintana ja lisäämällä noise-modifikaattori. Noise-modifikaattorilla saatiin vaivatta luotua kivikkomaista sattumanvaraiselta näyttävää maastoa. Illuusiota täydennettiin ripottelemalla rannalle sinne tänne isompia kiviä. Isompien kivien kanssa mallintaminen aloitettiin laatikkostandardimuodosta, josta sitten pursutettiin polygoneja haluttuun muotoon. Kiviä pyöristettiin tämän toimenpiteen jälkeen polygonien reunoja manuaalisesti muokkaamalla. Viimesilauksena kalliorannalle lisättiin meshsmooth-modifikaattori muotojen pehmentämiseksi. Isompia kiviä käytettiin myös piilottamaan epätasaisuudet erilaisten rantojen saumakodissa.

Kallioranta muokattiin erillisestä objektista, sillä sen geometria erosi huomattavasti saaren tasaisista muodoista. Kalliorannan mallintamisessa suoritettiin samat työvaiheet, kuin kivikkorannan kanssa. Eroina oli noise-modifikaattorin pois jättäminen ja mallintamisen suorittaminen suuremmilla polygoneilla.

6.2 Mökin mallinnus

Mökin (Kuva 12.) muodot saatiin pääasiallisesti sylinterin puolikkaita kopioimalla seinien hirsiksi. Kunkin seinän muodostavat sylinterin puolikkaat kiinnitettiin toisiinsa niin, että niiden tasaiset puolet muodostivat yhden polygonin. Tämä tehtiin, jotta sisäseinien teksturointi onnistuisi ja vältyttäisiin turhalta geometrialta. Mökin ulkoseinään leikattiin tasainen alue, joka jaettiin kolmeen osaan ovea ja ikkunoita varten. Nämä kolme aluetta sisennettiin, ja näin muodostuneet sisimmät polygonit liikutettiin ja skaalattiin toivottuihin muotoihin. Asetellut polygonit pursotettiin oven- ja ikkunoiden karmeiksi.

Mökin mallinnuksen nopeuttamiseksi mökistä viimeisteltiin vain neljännes, mitä kopioitiin ja heijasteltiin luomaan täydellinen kokonaisuus. Kivijalat ja katto mallinnettiin erillisinä objekteina. 3D-mallinnusosuudessa eniten käytettyjä työkaluja olivat laatikkomallintamisen perustyökalut.



Kuva 12. Saari ja mökki.

6.3 Maaston yksityiskohdat

Aluskasvillisuuden mallintamiseksi malliin tuotiin läpinäkyvä ruohikkotekstuuri materiaaliksi yksipolygoniselle tasolle. Tasoa kopioitiin ja käänneltiin eri kulmiin luomaan

sattumanvaraiselta vaikuttava ruohikkotilkku. Tilkun geometria yhdistettiin yhdeksi objektiksi, jotta sen muokkaaminen ja kopiointi Unityssä toimisi vaivatta.

Kuvan 13 kaislikkoa varten mallinnettiin yksi kaisla, jota kopioitiin kaislikoksi. Kaikki kaislan osat mallinnettiin omina objekteinaan standardimuodoista. Kaislikon teksturointi oli luovempi prosessi. Tekstuurit haettiin ilmaisia tekstuureja jakavalta ulkoiselta verkkosivulta (Texturer 2017.). Sivu oli kehitetty 3D-artisteille, suunnittelijoille, verkkosuunnittelijoille ja animaattoreille. Kaislikon vihreät osat käyttävät banaaninlehtitekstuuria ja vaalean osan tekstuuri on otettu vehnäpellon ilmakuvasta.

Kuva 13. Kaislikko.

6.4 Laiturien mallinnus

Sovelluksessa käytettävien laiturien mallinnus oli projektin eniten tarkkuutta vaativa osuus. Laiturien mittasuhteiden tuli olla Kuha-Laiturit Oy:n laiturielementtien mukaiset ja ponttonit tuli asetella kuvan 14 tavoin elementtien alle. Kuvassa 15 näkyvät kaikki mallinnetut laiturielementit.

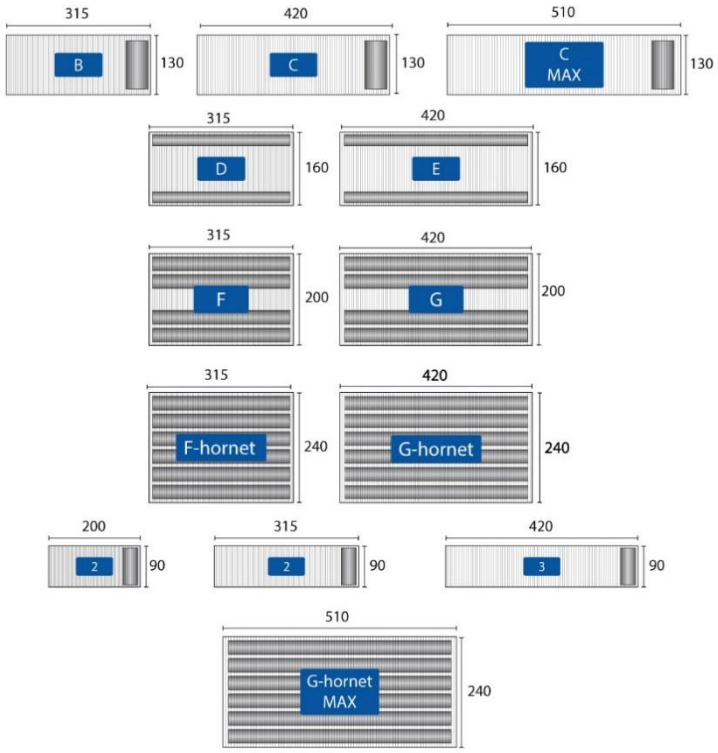
Laiturit kiinnitettiin toisiinsa kuvan 16 kääntyvien metalliliitosten avulla. Liitoksia oli kolmea tyyppiä eri tilanteita varten. Suoraa liitosta käytettiin yhdistämään kaksi saman levyistä laiturielementtiä. Eri levyisiä elementtejä yhdistettäessä käytettiin 90 asteen liitosta. Laudalla vahvistettu liitos oli kalliokiinnitystä varten.

Keskikokoiset, suhteellisen kevyet laiturit ankkuroitiin kuvan 18 mukaisesti putkiankkuroinnila. Kuvassa näkyviin laitureihin on kiinnitetty neljä putkiankkuria. Vain kahta ankkuria käytetään riippuen siitä, miten päin laiturin palat asetetaan, suhteessa edelliseen palaan. Putkiankkurointi suoritettiin eri taivoin riippuen ankkurointi asennosta suhteessa edeltävään laiturin osaan. Tämä ankkurointi sopii noin 2-3 metriä syviin rantoihin. Putkiankkuroinnissa laiturin runkoon kiinnitetään kasettilenkit, joiden läpi ankkuriputket juntataan pohjaan. Avattava kasettilenkki helpottaa putkien irrottamista, jos laituri tarvitsee nostaa talveksi rannalle. Putkiankkurointi ei sovi kalliopohjiin.

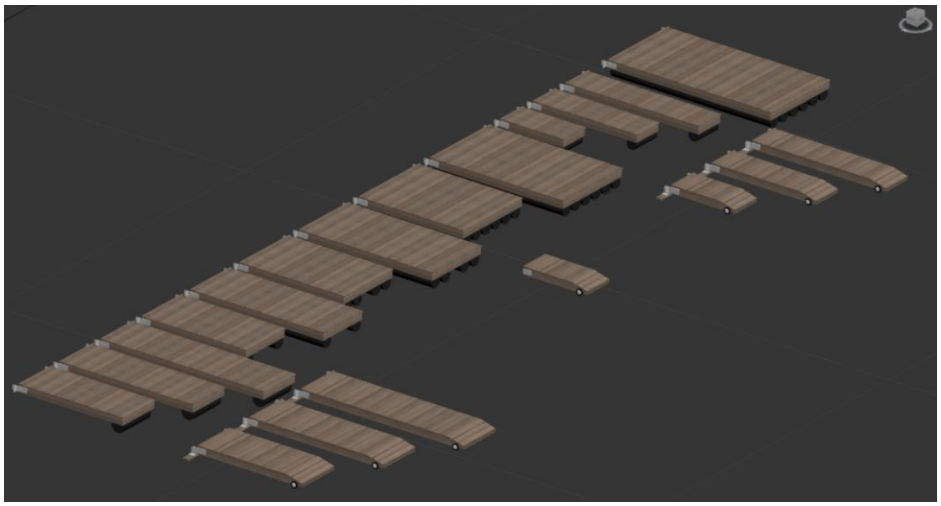
Kuvassa 17 esitettyä ankkurointitapaa kutsutaan painoankkuroinniksi. Painoankkuroinnissa laiturin ulkopäästä vedetään kaksi 10 metrin kettinkiä pohjapainoihin ristiin 45 asteen kulmassa. Ketjut kulkevat pohjaa pitkin eivätkä häiritse uimista tai rantautumista. Käytettyjen pohjapainojen koko riippuu aina laiturin koosta. Ketjuja varten mallinnettin vain yksi metallilenkki ja painot. Metallilenkille annettiin Unityssä cylinder collider, rigidbody ja hingejoint -modifikaattorit ja sitä monistettiin suoraksi kettingiksi. Mainittujen modifikaattorien avulla linkit saatiin käyttämään painovoimaa ja kiinnitäytymään toisiinsa kettingin tavoin. Kettingin ensimmäiselle linkille aktivoitiin rigidbodyn, kinetic niminen ominaisuus, jotta ketjut pysyisivät kiinni laitureissa. (Lip-lap laituri 2017)

Painoankkurointia varten harkittiin ankkureiden valmiiksi asettelua, jotta vältyttäisiin ylimääräiseltä tehojen käyttämiseltä. Rannan pohjan korkeuserojen vuoksi päädyttiin kuitenkin käyttämään raskaampaa vaihtoehtoa. Tätä ratkaisua olisi mahdollista säätää myöhemmin, jos se aiheuttaisi liikaa rasitusta.

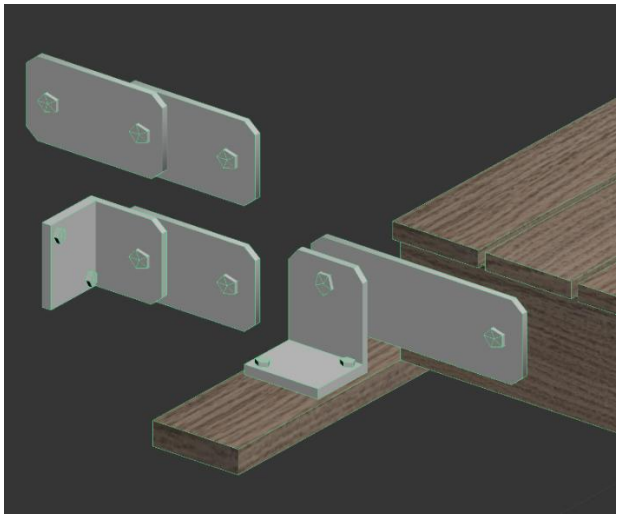
Laitureista oli alun perin tarkoitus jättää alapuoli mallintamatta. Kun kävi selväksi, että kuvakulmaa tulisi saada liikuttaa myös veden alle, oli laiturit mallinnettava kokonaisuudessaan.



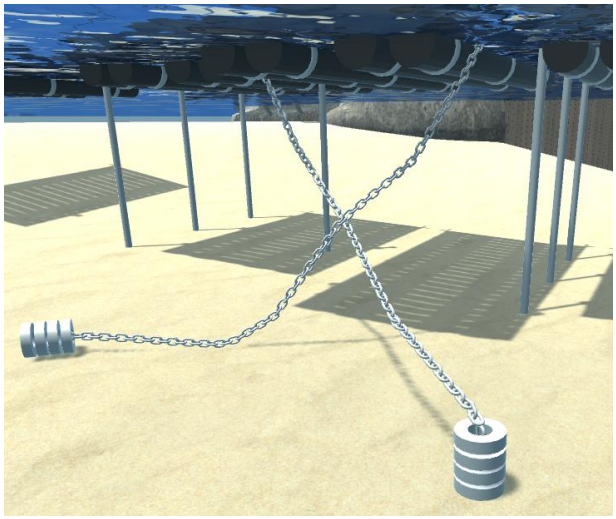
Kuva 14. Laiturielementit. (Kuhalaiturit 2017.)



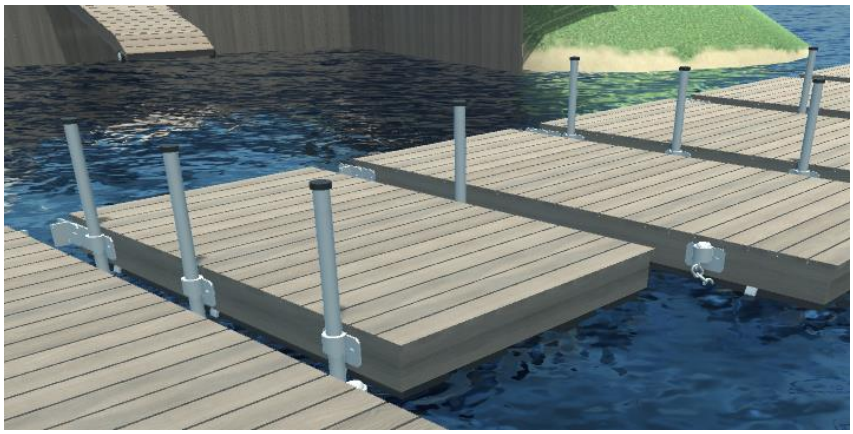
Kuva 15. Mallinnetut laiturielementit.



Kuva 16. Laitureiden kiinnikkeet.



Kuva 17. Painoankkurointi.



Kuva 18. Putkiankkurointi.

7 UNITY 5 PELIMOOTTORIOSUUS

Projektin pelimoottoriosuudessa käytettiin Unityn ilmaisversiota. Graafikkona rooliini kuului vielä animointi, valaisukartoitus ja efektien luominen Unityssä. Nämä vaiheet suoritettiin, kun oli ensin tuotu ja aseteltu mallit Unityssä. Ohjelmointiosuuden suorittaminen tulisi tapahtumaan viimeisenä, ja sen suoritti toinen henkilö.

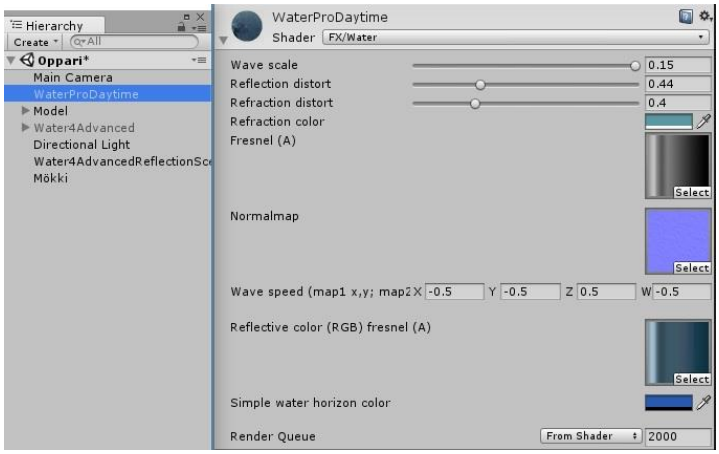
7.1 Meri

Työn saaristoteeman vuoksi aidon näköinen meri oli tunnelman luomiseen äärimmäisen tärkeä. Unityn valmiista shadereista löytyi muutamia eri vaihtoehtoja esittämään merta. Näyttävimpiä vaihtoehtoja olivat WaterProDaytime- ja Water4Advanced shaderit (Shehata, O. 2015.).

Water4Advanced shaderilla oli enemmän kustomointivaihtoehtoja. Säätämällä Wave Amplitude-parametriä kyseisen shaderin asetuksista, saatiin oletuksena ollut raju aallokko muutettua sopivan tyyneksi mereksi. Lisäksi käyttämällä Bump Direction-asetusta saatiin myös aaltojen suunta muutettua toivotuksi.

Kustomointiominaisuuksistaan huolimatta, projektin merta valittiin esittämään WaterProDaytime shaderi. WaterProDaytime todettiin helppokäyttöisemmäksi ja jo oletusasetuksillaan Water4Advanced shaderia aidomman näköiseksi. Aaltojen skaala (Wave Scale-asetus) jätettiin projektin tässä vaiheessa maksimille. Myöhemmin mahdollisesti ilmenevien teho-ongelmien ilmetessä skaalaa pystyisi helposti pienentämään.

Kuva 19. WaterProDaytime.



Kuva 20. WaterProDaytime käytetyt asetukset.

7.2 Aurinko ja taivas

Paras tapa imitoida aurinkoa Unity 5:ssä on ladata ilmainen Lens Flare -paketti Asset Storesta. Asettamalla Lens Flare-efektille kellertävä väri saadaan aikaan varsin uskottava aurinko. Vähiten tehoja vaativa tapa saada taivas näyttäväksi oli vaihtaa oletus sky map toivotun näköistä taivasta vastaavaksi.

Työssä oli otettava huomioon, että sovelluksen tulisi toimia mobiilialustoilla, joten käytetyt menetelmät rajoitettiin vähiten tehoja vieviin. Näyttävintä jälkeä ulkoilmatiloja kehittäessä olisi saatu hyödyntämällä god rays/sun shaft -toimintoa. God rays aiheuttaa valon osittaista hajontaa sen törmätessä läpikuultaviin kappaleisiin. Kyseistä työkalua olisi voitu hyödyntää, jos pilvet olisi mallinnettu objekteina ja puihin olisi kiinnitetty enemmän huomiota. Tällöin valonsäteet olisivat voineet vaikuttaa ympäristöön. Kaikki tämä olisi vaatinut huomattavasti enemmän tehoa käytetyltä alustalta, joten vaihtoehto hylättiin aikaisessa vaiheessa kehitysprosessia.

Kuva 21. Lens Flaren avulla tehty aurinko.

Kuva 22. Esimerkki God rays -efektistä. (unity3D 2017.)

7.3 Viimeistely

Tutkimusten mukaan ihmiset tekevät ostopäätöksiä tukeutuen enemmän tunteisiin kuin mainosten varsinaiseen sisältöön. Tästä syystä saarelle ripoteltiin viimesilauksena kuvia ihmisistä nauttimassa kesäpäivästä. Lisäksi saarelle mallinnettiin vielä muutamia tunteisiin vetoavia, kesästä muistuttavia esineitä, kuten vesileluja, grilli ja lokkeja taivaalla. (Kuva 23.)

Kuva 23. Viimesilauksia.

8 JOHTOPÄÄTÖKSET

Opinnäytetyössä pyrittiin tutkimaan käytetyn mallinnusohjelmiston, 3ds Maxin toimintoja ja pelien 3D-grafiikan työnkulkua kyseisessä mallinnusohjelmistossa. Työssä tutustuttiin myös useisiin Unity pelimoottorin olennaisiin prosesseihin, kuten valaisukartoitukseen ja ulkoilmatilojen realistiseen valaisuun.

Lopputuloksena työstä saatiin tavoiteltu 3D-sisältö. 3D-mallit tuotiin pelimoottoriin ja aseteltiin valmiiksi ohjelmoitavaksi ja implementoitavaksi verkkosovellukseen. Projektin valaistus laitettiin kuntoon, ja asetelmaan lisättiin realistinen meri. Oletus sky map vaihdettiin toivotunlaista taivasta vastaavaan. Maasto valaisukartoitettiin, mutta laiturit jätettiin reaaliaikaisen valon varaan, koska niitä tuli voida käännellä useaan eri kulmaan.

Lopputuloksena saatua merta olisi voitu vielä parannella, jos olisi paremmin hallittu shaderien käyttö, valon säteiden luominen ja muut olennaiset efektit. Unity Asset Storesta löytyisi tähän tarkoitukseen paketteja, mutta ne jätettiin toistaiseksi hankkimatta. Työssä vältettiin käyttämästä high poly-mallien heijastamista low poly-mallien tekstuureiksi, sillä pääasiallisesti tuotteessa pyrittiin mainostamaan laitureita, jotka eivät kulmikkaan geometriansa vuoksi vaatineet kyseistä tekniikkaa. Jälkeenpäin tarkasteltuna, mallinnetusta maastosta ja kasvillisuudesta olisi kuitenkin kannattanut tehdä myös tarkemmat mallit. Näiden avulla olisi voitu tehdä erikseen versiot mobiilialustoille ja tietokoneille. Erilliset versiot olisivat mahdollistaneet God Rays-komponentin tyylisten raskaampien efektien käyttämisen projektissa.

Opinnäytetyötä tehdessäni opin yllättävän paljon 3D-peligrafiikan kehityksestä. Kokoamalla koko prosessin yhteen työhön, sain kattavamman käsityksen kaikista työssä käytetyistä ohjelmista ja niiden hyödyllisimmistä toiminnoista. 3D-sisällön tuottamisprosessia ymmärtämällä sain paremman käsityksen kunkin vaiheen vaatimuksista ja rajoituksista. Tämä tulee helpottamaan tulevissa projekteissa optimaalisen lopputuloksen saavuttamisessa minimiajassa.

Jatkokehitettäväksi sovellukseen jäi laiturielementtien muokkaamista. Jokainen laiturinvalmistaja tarvitsisi räätälöidyt laiturielementit. Luonnollinen seuraava askel esittelysovelluksen kehityksessä olisi asiakkaiden rakentamien laiturien myynti suoraan sovelluksessa. Vaatimukset tämän tyyppiseen verkkokauppasovellukseen olisivat kuitenkin jo huomattavasti nykyistä vaativammat.

LÄHTEET

AUGI 2014. 3D-peligrafiikan työnkulku. Viitattu 21.11.2017.
https://www.augi.com/articles/detail/game-assets-workflow

Autodesk. 2017a. 3D-mallintaminen. Viitattu 9.11.2017.
https://www.autodesk.com.au/solutions/3d-modeling-software

Autodesk 2017b. Ohjelman lataus. Viitattu 15.11.2017. http://www.autodesk.eu/store

Epic Games 2017. Blueprint. Viitattu 10.12.2017.
www.epicgames.com > Unreal Engine > Documentation > Blueprints Visual Scripting

Computer Hope 2017. Graafinen käyttöliittymä. 2.10.2017.
https://www.computerhope.com/jargon/g/gui.htm

Robet, T. 2013. Kuvataajuus. Viitattu 11.12.2017.
http://aframe.com/blog/2013/07/a-beginners-guide-to-frame-rates/

MDN Web Docs 2017. Renderöinti. Viitattu 11.12.2017.
https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_on_the_web/Basic_theory

Mochochi, B. & Lahiri, K. & Cadambi, S. 2006. Power analysis of mobile 3D graphics. Viitattu 11.12.2017. https://dl.acm.org/citation.cfm?id=1131617

Kuhalaiturit 2017. Viitattu 5.12.2017.
http://www.laiturit.fi/elementit/

Lip-lap laituri 2017. Ankkurointi. Viitattu 1.12.2017.
https://www.lip-lap.fi/laiturit/ankkurointi/muoviponttonilaiturin-ankkurointi

Aspis, M. 2017. Pelimoottorit. Viitattu 29.11.2017.
http://www.discoversdk.com/blog/6-top-game-engines-in-2017

Shehata, O. 2015. Shader. Viitattu 11.12.2017.
https://gamedevelopment.tutsplus.com/tutorials/a-beginners-guide-to-coding-graphics-shaders--cms-23313

Suen, P. 2015. Viitattu 5.12.2017.
https://www.artstation.com/artwork/1w8QG

Pluralsight 2015. 3D-mallinnusohjelmistot. Viitattu 29.11.2017.
https://www.pluralsight.com/blog/film-games/3ds-max-maya-lt-blender-3d-software-choose-asset-creation

Reallusion 2017. Karttatyypit. Viitattu 3.12.2017.
https://www.reallusion.com/iclone/help/iclone3/15_multiple_channel_texture_mapping/types_of_maps.htm

Runevision 2011. Unity valaistus tutoriaali. Viitattu 17.11.2017.
http://answers.unity3d.com/questions/40674/creating-mood-and-atmosphere-with-lighting-in-unit.html

Seraphinacorazza 2012. NURBS. Viitattu 10.12.2017.
https://seraphinacorazza.wordpress.com/2012/12/28/modeling-techniques-differences-between-nurbs-and-polygon-modelling-new-york-city-rooftop-3d-architecture-project-research

Shutterstock 2017. Kuvien jakelusivusto. Viitattu 25.11.2017.
https://www.shutterstock.com/fi/

Sonofatrus 2012. Kivien mallinnus tutoriaali. Viitattu 5.11.2017.
https://www.youtube.com/watch?v=EAsGCtOhkQo

Zorin, D. 2006. Modeling with Multiresolution Subdivision Surfaces. Deepdyve. Viitattu 10.12.2017.
https://www.deepdyve.com/lp/association-for-computing-machinery/modeling-with-multiresolution-subdivision-surfaces-HJ4CkWQTzF

Texturer 2017. Viitattu 20.11.2017 http://texturer.com

Tino 2015. Valokartoitus Unityssä. Viitattu 12.11.2017.
https://sassybot.com/blog/lightmapping-in-unity-5/

Unity3D 2017a. Importointi. Viitattu 17.10.2017.
https://docs.unity3d.com/Manual/HOWTO-importObject.html

Unity3D 2017b. Optimoitu suorituskyky. Viitattu 17.10.2017.

https://docs.unity3d.com/Manual/ModelingOptimizedCharacters.html

Unity3D. 2017. Sun Shafts. Viitattu 5.12.2017.
https://docs.unity3d.com/550/Documentation/Manual/script-SunShafts.html

Unity3D 2010. Lens Flare. Viitattu 3.11.2017.
https://www.assetstore.unity3d.com/en/#!/content/5

Wikipedia 2017. Pienellä geometrialla mallintaminen. Viitattu 22.11.2017.
https://en.wikipedia.org/wiki/Low_poly