

Heikki Mäki

PELEISSÄ MALLINTAMISEN  
PERIAATTEET  
Case: Stratogear

Opinnäytetyö  
Tietojenkäsittelyn koulutusohjelma


Joulukuu 2013




**MIKKELIN AMMATTIKORKEAKOULU**

Mikkeli University of Applied Sciences

## KUVAILULEHTI

 <b>MIKKELIN AMMATTIKORKEAKOULU</b> Mikkeli University of Applied Sciences	<b>Opinnäytetyön päivämäärä</b>  3.12.2013
<b>Tekijä(t)</b> Heikki Mäki	<b>Koulutusohjelma ja suuntautuminen</b> <b>Tietojenkäsittelyn koulutusohjelma</b>
<b>Nimeke</b> Peleissä mallintamisen periaatteet	
<b>Tiivistelmä</b>  <p>Opinnäytetyö toteutettiin osana suurempaa projektia. Projektin tavoitteena oli kehittää peli. Peli käyttää alustana PC:tä. Projektin onnistumiseksi pelin tuli koostua vähintään pelattavasta demosta ennen kuin se voitiin katsoa opinnäytetyön osalta suoritetuksi.</p> <p>Projektin koostuu neljästä eri osa-alueesta. Kukin osa-alue muodostaa oman opinnäytetyönsä. Osa-alueet ovat käsikirjoitus, mallintaminen, satunnaisuus ja tekoäly.</p> <p>Tämän opinnäytetyön aiheena on mallintaminen. Tutkimusongelmana käsitellään yleisesti, miten malleja voidaan soveltaa peleihin optimaalisesti ja miten niitä voidaan mallintaa tehokkaammin pelien kannalta. Työssä käsitellään pelien asettamia rajoituksia ja vaatimuksia joista mallintajan on oltava tietoinen jollakin tasolla.</p> <p>Työssä käytetään hyväksi edeltävää käytännönkokemusta ja aihetta käsittelevää mediaa. Työssä on myös hyödynnetty aiheeseen liittyvää kirjallisuutta. On huomioitava, että tietyt tapauskuvailut ovat kuvauksia työn aikana tapahtuneista tapauksista.</p> <p>Työn aikana sovellettiin paljon 3DSMax- ohjelmaa ja Unity-pelimoottoria. Tästä syystä näihin kahteen viitataan työn aikana useaan otteeseen. Työhön sisältyi myös tekstuuriin työstämistä, joten GIMP- kuvankäsittelyohjelma mainitaan myös. Animaatiota ei sovellettu erillisissä ohjelmissa, mutta siitä on työssä oma lukunsa.</p> <p>Työ toteutettiin muiden projektin osa-alueiden rinnalla samanaikaisesti. Toteutus oli projekti luontoinen siinä mielessä, että muitten haarojen kanssa käytiin säännöllistä neuvottelua etenemisestä. Aikataulukin päätettiin myös ryhmässä niin kuin vaadittu sisältö. Ongelmien sattuessa suoritettiin myös ideoiden vaihtoa niiden ratkaisemiseksi.</p> <p>Lopputuloksena päädyttiin pelidemoon, jonka mekaniikat toimivat odotetusti, mutta ennen kaikkea tämän työnsalpa tärkeää on pelin esteetön suoritus ilman häiriöitä. Käytetyt mallit ja tekstuurit soveltuvat tällöin pelin käytettäväksi.</p> <p>Tavoitteena on, että lukija saa tästä työstä jonkinlaisen käsityksen millaista pelinkehitys on. Aloitteleville mallintajille tavoitteena on välittää vähintään muutama hyödyllinen ohjenuora joita voi soveltaa mallintamiseen peleissä.</p>	
<b>Asiasanat (avainsanat)</b>  Mallintaminen, 3D-Mallinnus, kuvantaminen	
<b>Sivumäärä</b> 40	<b>Kieli</b> Suomi
<b>URN</b> URN:NBN:fi:mamk-opinn2013A9145	
<b>Huomautus (huomautukset liitteistä)</b>  	
<b>Ohjaavan opettajan nimi</b> Jukka Selin	<b>Opinnäytetyön toimeksiantaja</b>  

## DESCRIPTION

 <b>MIKKELIN AMMATTIKORKEAKOULU</b> Mikkeli University of Applied Sciences		<b>Date of the bachelor's thesis</b>  3 December 2013
<b>Author(s)</b>  Heikki Mäki	<b>Degree programme and option</b>  Business Information Technology	
<b>Name of the bachelor's thesis</b>  The principles of modelling in games		
<b>Abstract</b>  <p>This bachelor's thesis was done as a part of a larger project. The goal of the project was to create a videogame that used PC as an operating platform. For the project to succeed there should be at least a playable demo of the game before this project could be viewed as completed in terms of this thesis.</p> <p>A general goal of this thesis was to give the reader some insight and perspective on the practical nature of game development. At very least the goal was to provide beginning modellers some helpful guidelines to apply to modelling in games.</p> <p>The project consisted of four separate areas of expertise. Each area in turn formed its own thesis. These areas were scripting, modelling, randomization and artificial intelligence. The subject of this thesis was modelling. The case study surveyed how to apply 3D models to games optimally and how to model 3D-objects for games more efficiently. This thesis also dealt with the limitations and requirements games set and modelers should be conscious of.</p> <p>3DSMax-modelling program and the Unity game engine were in heavy use during this thesis. The GIMP image editor was also heavily utilized to edit textures. Animation was not done in specific programs, but animation was introduced in one chapter in this thesis.</p> <p>The work for this thesis was executed simultaneously with the other parts of the project. This meant that the work was project-oriented, because regular consultation between the different subprojects was required when deciding how to proceed the specific aspects of the project. The contents and schedule were also decided in the larger group. Ideas were exchanged to solve practical problems.</p> <p>The end result of project was a playable demo that functioned as expected. Most importantly the demo functioned fluently with the applied 3D-objects without any interruptions or unintended malfunctions. This meant that optimizing the objects was succesful.</p>		
<b>Subject headings, (keywords)</b>  Modelling, 3D-modelling, imaging		
<b>Pages</b> 40	<b>Language</b> Finnish	<b>URN</b> URN:NBN:fi:mamk-opinn2013A9145
<b>Remarks, notes on appendices</b>		
<b>Tutor</b>  Jukka Selin	<b>Bachelor's thesis assigned by</b>	

# Sisältö

1 JOHDANTO.....	<u>1</u>
2 MALLINTAMISEN TEORIAA.....	<u>2</u>
2.1 Mallintamisen työkalut.....	<u>4</u>
2.2 Mallintamisen sovellukset.....	<u>9</u>
3 MALLIEN EFEKTI.....	<u>13</u>
3.1 Tulkinta.....	<u>14</u>
3.2 Imersio/ Efekti.....	<u>16</u>
3.3 Mallien vaikutus kokonaisuuteen.....	<u>19</u>
4 ANIMOINNISTA.....	<u>23</u>
5 MALLINTAMINEN PELEISSÄ.....	<u>25</u>
5.1 Pelin asettamat rajoitukset.....	<u>27</u>
5.2 Mallien rajoittaminen.....	<u>29</u>
5.3 Mallien rajoittamisen keinot.....	<u>30</u>
5.4 Mallien sovittaminen peleihin.....	<u>33</u>
6 POHDINTA.....	<u>35</u>
7 LÄHTEET.....	<u>39</u>

## 1 JOHDANTO

Tavoitteena on suorittaa projekti, joka lopullisessa muodossaan on vähintään demokelpoinen videopeli. Peli itse koostuu useasta eri opinnäytetyöstä ja tutkimusongelmasta, jotka on puolestaan jaettu niille henkilöille, jotka ovat ottaneet ne vastuulleen. Osa-alueet pitävät sisällään pelimekaniikkaan ja sisällön tuottamiseen liittyviä tutkimusongelmia. Omalta osaltani olen ottanut vastuulleni 3D-grafiikat ja siihen liittyvän sisällön.

Mitä tahansa laajempaa ohjelmaa tehtäessä, joka pitää sisällään samaan aikaan suoritettavia toimintoja, on pidettävä mielessä laskentatehon kokonaismäärä ja sen jakaminen ohjelman toiminnoille. Yleisesti tiedetään, että graafiset ominaisuudet (kuvat, videot, yms) vievät huomattavasti laskentatehoa ja jos näitä elementtejä lisätään osaksi ohjelmaa, on olemassa mahdollisuus, että ne voivat haitata ohjelman suoritusta sujuvasti. Tällöin on tärkeää kyseisten elementtien optimointi. Tällöin varmistetaan, että käytetyt elementit eivät rasita ohjelmaa liikaa ja onnistuvat silti välittämään tarkoituksen mukaisen efektin.

Peleissä käytetään 3D-, animaatio ja graafisia elementtejä jatkuvasti ja jollei noita elementtejä käsitellä sopivalla tavalla, voidaan lopussa huomata, että parhainkaan mittatilaus PC pystyy suorittamaan peliä sujuvasti jos ollenkaan. Siispä on tärkeää pelin kehittämisen alusta lähtien optimoida elementtejä ja jatkaa tätä kehityksen edetessä loppuun asti, oli sitten kyseessä moottoriin itseensä sisältyvä mekanismi tai kevyempi malli.

Tarve optimoida puolestaan synnyttää erilaisia keinoja joilla optimointia voidaan suorittaa. Tämän opinnäytetyön osalta tutkimusongelmana on tutkia ja soveltaa noita keinoja pelin 3D- sisältöön. Tämän tarkoituksena on mahdollistaa pelin sujuvampi suoritus ja laskentatehojen tehokkaampi jakeleminen. Onnistumisen ehtona voidaan pitää sitä, että lopussa peli toimii ilman häiriöitä mallien kanssa. Tämä merkitsee sitä, että mallit eivät ole konfliktissa suoritettujen ohjelmoinnin kanssa eivätkä rasita pelin suoritusta liikaa, jolloin ne veisivät tehoa muilta toiminnoilta.

## 2 MALLINTAMISEN TEORIAA

Nykyään 3D-malleja käytetään suunnittelussa monellakin elämän osa-alueella. Arkkitehtuurin suunnittelusta teknisiin presentaatioihin ja elokuvista peleihin. Mallien tarkoituksena voi olla esimerkiksi testata koneiden osien toimivuutta turvallisesti 3D-ympäristössä. Mallien käyttötarkoituksia on kuitenkin tätä esimerkkiä huomattavasti enemmän. (Hietikko 2007, 23.)

Objekteja voidaan mallintaa useita tarkoituksia varten. Mallinnusta käytetään useilla eri aloilla osana kokonaisuutta. On kyse mallinnuksen suorittajan omasta alasta ja projektista mitä mallinnukseen kuuluu. Tässä luvussa käsitellään tarkemmin siis objektien mallintamista 3D-avaruudessa ja niiden mallien käyttämistä peleissä.

Mallintamisessa on myös kyse uusien ideoiden kokeilemisesta ja soveltamisesta. 3D-mallit ovat osa niin digitaalista taidetta, kuin tiedettäkin jota sovelletaan niiden luomiseen. Tällöin koetellaan rajoja taiteellisesti ja tieteellisesti, jotta saatetaan toteuttaa ideoitu visio. Tämä myös tarkoittaa sitä, että on varauduttava ratkaisemaan prosessin aikana syntyviä ongelmia ilman sen suurempaa avustusta. Mallintaminen ja digitaalinen taide yleensä on suhteellisen tuore ala ja teknologinen ja tieteellinen kehitys vaikuttaa siihen, että tarpeellinen asiantuntemus ei välttämättä ole vielä saatavilla. (Colson 2007, 20.)

Onnistunut peli on, yleisesti ottaen osiensa summa. Tietenkin tekninen toteutus on kriittinen osa mitä tahansa peliohjelmaa. Peliä varten täytyy ottaa käyttöön sille sopiva moottori tai rakentaa se alusta. Objektitkin täytyy saada liikkumaan suunnitelman mukaisesti. On myös saatava toiminnot sijoitettua niille kuuluville alueilla ja varmistettava niiden toimivuus.

Ilman sisältöä tämä on kuitenkin vain, kuten edellä mainittiin, teknistä toteutusta. Ilman varsinaista sovellusta johon voidaan kerätä tuotokset projektin eri osa-alueilta, näillä tuotoksilla ei ole suurempaa kontekstia ja täten pysyvät vain ideoina.

Pelimoottori on tässä vaiheessa oikeastaan ehdoton sillä se mahdollistaa testiympäristön sekä alustan jolle lopullinen peli muodostuu. Käytännössä moottori mahdollistaa kontekstin erilaisille ideoille, joista peli muodostuu. Pelin kehittäminen kannattaa siis aina aloittaa moottorista ja kehitysympäristöstä.

Kehitysympäristö toimii siis ankkurina muille osa-alueille myös mallintamiselle. Tästä johtopäätelmästä päästään sitten varsinaiseen mallintamiseen. Perustasolta katsottuna mallintamisessa on kyse yhtä paljon vaistosta kuin siinä on kyse tekniikasta, ellei enemmänkin. Molemmat ovat opittuja piirteitä ja molemmat eivät parane kuin harjoituksen ja kokemuksen myötä, mutta hyvän vaiston myötä mallintaja oppii muokkamaan mallia tarpeen niin vaatiessa tai lopettamaan muokkauksen kun se ei ole mallin kannalta tarpeellista.

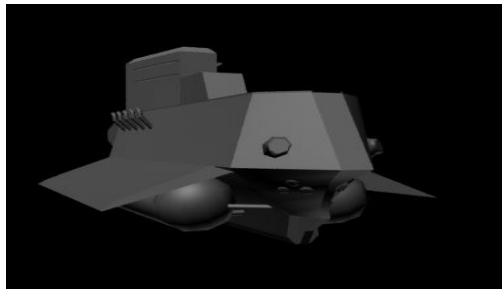
Tekniikka on tärkeä mallien laadun kannalta. Jos henkilöllä on visio tietystä mallista niin on hyvä muistaa, että amatööritasolta lähtiessä tekniset taidot eivät välttämättä tuota haluttua lopputulosta. Täten on syytä varautua siihen, että joutuu tekemään useita vedoksia lopputuloksen saavuttamiseksi. Mitä aikaisemmin tekniikan harjoittelun aloittaa sen paremmaksi se elämän aikana kehittyy. Tässä tietenkin tarkoitetaan tekniikan harjoittelua sillä vaiston karttuminen, kirjoittajan omien kokemusten perusteella, perustuu varsinaisiin työ ja/tai projektikokemuksiin.

Tekniikkaa voi yleensä harjoitella miten vain käytännössä piirustuslehtiöstä Googlen SketchUp- ohjelmaan. Harjoitteluvaihtoehdot tietenkin mukautuvat sen perusteella mikä on kyseessä olevan henkilön vallittu ala. Mitä mallinnukseen 3D-ympäristössä tulee, niin kyky hahmottaa kolmiulotteisia muotoja on avuksi. Piirtotaidosta on suurehko hyöty, sillä mallinnus on perustasolla piirtoa kolmessa ulottuvuudessa. Näistä taidoista on apua alussa, mutta seuraavaksi on aina siirryttävä taitojen omaksumisesta niiden soveltamiseen.

Edellä mainittujen taitojen sovellus suoritetaan ensinnä työkalujen valitsemisella. Tällöin kehitetään tekniikkaa edistyneemmällä tasolla, oppimalla mallintamaan erikoistuneimmissa ympäristöissä. Tarjolla olevia työkaluja 3D-mallintamiseen on suuri määrä, mutta tässä opinnäytetyössä viitataan enemmän Autodeskin 3DS-Max ohjelmaan johon sen pääasiallisesta käytöstä mallintamisessa projektin aikana. Työkaluja käsitellään tarkemmin muissa luvuissa.

Tässä kohtaa päästään itse ympäristöön. Selvennetään ensin perus termit 3D-mallintamisessa. Ensinnä termi Mesh. Mesh on englannin kielinen sana, jolla viitataan objektiin 3D-ympäristössä. Meshille on varmaan tarkempi suomenkielinen käännös, mutta yleensä käytetään sanaa mesh, joka on yleisesti hyväksytympi. Tässä opinnäytetyössä käytetään siis mesh-sanaa.

Mesh on, kuten aikaisemmin todettiin, objekti 3D-ympäristössä. Selvennetään vielä, että kyseessä on tietokoneella luotu 3D-ympäristö, epäselvyyksien varalta. Mesh on mallin rajat avaruudessa. Mesh määrittää mallille muodot mitkä muodostavat objektin. Malli on kuitenkin mesh-tilassa vielä tyhjä ja ei näy muutoin kuin aivan perus muodossaan. Mallintaminen alkaa Meshin rakentamisesta, mutta yksinkertaisesti ei voi kuvitella, että mesh itsessään riittää mihinkään tarkoituksen mukaiseen käyttöön. Tällöin tekstuurit otetaan käyttöön. *Kuvassa 1 on esimerkki peliin tulevasta aluksen meshistä ilman tekstuureja*



**KUVA 1 :Raskaan aluksen mesh**

Yksinkertaistettuna tekstuureilla tarkoitetaan sitä, miltä mallin pinta näyttää. Esimerkiksi onko se puuta, metallia vai jotain muuta. Puhutaan siis pintakuvioista joilla päällystetään mesh. Lyhyesti sanottuna tähän löytyy monia työkaluja monille eri taito-aloille, joten niiden valinta ja käyttö riippuu pitkälti tekijästä itsestään. Tekstuurien tarkoituksena on lisätä tarkoituksen mukainen tuntuma meshiin, joka vahvistaa sen fyysisistä ulkonäköä.

Valmiina mallina voidaan pitää mallia joka, koostuu vähintään kahdesta asiasta. Meshistä ja tekstuureista. Tällöin malli on tarkoituksenmukainen niin fyysiseltä olemukseltaan, kuin ulkonäöltäänkin. Tällöin malli on myös käyttövalmis. Käyttövalmiin mallin on aina osoitettava nämä ominaisuudet, koska muutoin ei saavuteta haluttua efektiä.

Mallinnetut objektit ovat tärkeitä pelille siinä missä koodikin. Objektit antavat paikan, jossa soveltaa koodia ja tehdä siitä tarkoituksellista. Kuten todettiin jo aikaisemmin, ilman liitettyjä objekteja koodi on vain koodia. Tämä pätee myös käänteisesti. Ilman ohjelman toteuttavaa koodia, malli on vain malli. Mitä tekijällä on, on hienon näköinen malli ilman mitään sen tarkempaa kontekstia. Muistetaan siis, että pelin kehittäminen on aina monisävyinen kokonaisuus.



## 2.1 Mallintamisen työkalut

Mallintamisen työkaluilla keskitytään tässä opinnäytetyössä juuri 3D-mallintamisen ohjelmiin. Selvennykseksi on todettava, että lukijan ei kannata sekoittaa 3D-mallinnus ohjelmia ja 2D-kuvankäsittelyohjelmia samaan luokkaan. Tosin 2D-kuvankäsittelyllä on roolinsa tekstuurien editoinnissa, mutta tähän palataan myöhemmin.

Teknisesti mallinnus lähtee mallintajan ideoimasta konseptista. Tässä yksissään voi olla vaihteleva määrä vaiheita, riippuen esimerkiksi projektin koosta, tarpeista, tekijöiden visioista, henkilöstön määrästä tai näiden ja muiden mainitsemattomien seikkojen



### **KUVA2: Varhainen konseptiluonnos**

yhdistelmistä. *Kuvassa 2 voidaan nähdä ideoinnin siirtyminen paperille. Tämä yleensä auttaa konseptin kehittämisessä.*

Suuremmissa pelistudioissa on yleensä tapana muodostaa kussakin projektissa konseptitaiteen ympärille oma ryhmänsä. Pienet projektit voivat tyytyä varsin perusteellisiin raaka vedoksiin ennen mallintamista tai tekevät malleista suoraan 3D-vedoksen, jota sitten editoidaan projektin edetessä.

Tässä vaiheessa käytetään vaihtelevan laatuista 2D-konsepteja. Kuten edellä mainittiin ne voivat olla joko suhteellisen korkealaatuisia, tai raakavedoksia. Tärkeintä näiden työkalujen kannalta on auttaa tekijöitä päättämään oikea konsepti projektille, jonka pohjalta voidaan ruveta mallintamaan.

Tarkennetaan tässä vaiheessa, että pelien kehityksessä konseptipiirustukset, eli sketsit ovat enemmän suuntaa-antavia ehdotuksia. Ne eivät ole suoraan sovellettavissa mallintamiseen toisin kuin esimerkiksi Hietikon havainnollistamassa (2007, 27) mekaanisessa suunnittelussa.

Totuuden mukaisesti on varmaan mahdollisuus koodata yksinkertaisia meshejä, mutta ne ovat suhteellisen yksinkertaisia ja eivät ole tarkoitukseen sopivia monimutkaisia malleja ajatellen. Otetaan huomioon vielä se, että meshien ohjelmointi on työlästä, monimutkaista ja tämä työ on aina suoritettava varsinaisen ohjelmoinnin lisäksi. Siispä on hyvä suorittaa mallintaminen erikseen siihen tarkoitettulla ohjelmalla.

Mallintamisessa käytettävien työkalujen määrä on kieltämättä suuri, mutta kaikki ohjelmat eivät välttämättä sovi aina mallintajan tarkoituksiin. Siispä on hyvä aina perehtyä projektin alkuvaiheissa siihen mitä ominaisuuksia malleilta haluaa ja mikä ohjelma parhaiten tarjoaa nuo ominaisuudet. Yksinkertainen periaate, mutta sillä onnistutaan pitämään kiinni omista laatuksenteereistään.

Käyttää mallintaja sitten mitä työkalua tahansa, on hänen kuitenkin syytä perehtyä käyttämäänsä ohjelmistoon myös tekniseltä kannalta. Ohjelmassa voi olla useita eri vaihtoehtoja tiettyjen muotojen tai efektien saavuttamiseen. Hietikko ehdottaa(2007, 39 ), että onkin siis mallintajan vastuulla tutkia mitkä vaihtoehdot toimivat parhaiten tai tehokkaimmin mallin kannalta. Esimerkiksi kaarevan muodon voi soveltaa malliin monellakin tavalla.



**Kuva 3: Kaareva muoto**

*Kuvassa 3 on esimerkiksi venytetty yhtä polygonia "extrude along spline" menetelmällä. Lähestyminen tehokkuusnäkökannalta auttaa sekä mallintajaa, että projektia.*

Googlen Sketch Up on vapaasti saatavilla ja sillä pystytään mallintamaan käytännössä mitä vain, mutta sillä on kuitenkin rajoituksensa. Nämä rajoitukset estävät sitä saavut-

tamasta kilpailijoidensa tasoa mitä tulee ominaisuuksiin. Jos esimerkiksi halutaan mallintaa fotorealistisen näköiset kasvot ja lisätä näihin sopivat tekstuurit valaistus ja animaatiot, niin Sketch Up ei välttämättä täytä kaikkia vaatimuksia ominaisuuksien suhteen.

Voi olla, että yhdessä ohjelmassa ei ole tiettyjä ominaisuuksia joita tarvitaan tietyn efektin luomiseen tai polygoneja ei pysty muokkaamaan halutulla tavalla. Aikaisemman kokemuksen puute voi olla haittana jos jollakin ohjelmalla on korkeampi oppimiskäyrä.

Max on esimerkiksi huomattavasti vaikeaselkoisempi käyttää, kuin Googlen Scetch Up. Onkin suositeltavaa, että perehtyy ensin sen käyttöön ennen kuin mallintaa Maxilla täysipäiväisesti. Ajan varaaminen tuntemattomien ohjelmien opetteluun on kannattavaa, koska tällöin säästetään aikaa myöhempien ongelmien kanssa jotka olisi kokemattomuuttaan tahattomasti aiheuttanut.

On myös otettava huomioon malliin liittyvät rajoitukset. Miten paljon pelimoottori antaa muistia malleille ja niiden efekteille. Miten hyvin mallit voidaan tuoda peliympäristöön, tai ovatko mallinnusohjelma ja moottori edes yhteen sopivia.

Esimerkiksi Unity3D-pelimoottori ja 3DS-Max ovat yhteensopivia vain kun molemmat ovat asennettuna samalle koneelle. Tämä siis tarkoittaa sitä, että edellä mainitussa tapauksessa Maxista pystyy tuomaan Unityyn tiedostoja Maxin omassa .max formaatissa suoraan. Muut koneet joilla ei ole Maxia vaan ainoastaan Unity, eivät voi tulkita .max tiedostoja suoraan, vaan ne on ensin ladattava Maxista Unityyn Maxin omaavalla koneella ja sitten ne ovat tulkittavissa kaikilla koneilla, joissa on pelkästään Unity.

Juuri yhteensopivuussyistä projektissa suosittiin yleisesti .fbx formaattia mallien kohdalla, mutta loppua kohden alkoi ilmetä ongelmia. Kun tuli aika pinnoittaa mallien meshit tekstuureilla, kävi ilmi, että tämä osoittautuikin suhteellisen hankalaksi .fbx formaattisilla tiedostoilla. Siispä lisäselvityksen myötä, tekstuurien kannalta olisi ehkä suotavampaa käyttää .max formaattia ja tuoda valmis tiedosto Unityyn asettina, joka kääntäisi sen automaattisesti.

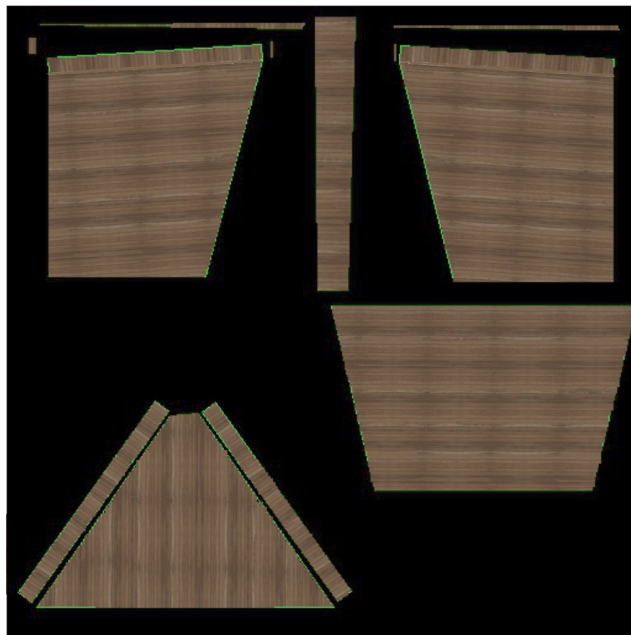
Tästä esimerkistä käy ilmi kuinka vähästä projektin eteneminen voi olla. Kirjoittaja myöntää tehneensä virheen jättäessään selvittämättä tarkemmin Maxin ja Unityn yhteen sopivuutta projektin alussa. Tätä jouduttiin selvittämään projektin aikana muuta-

maan otteeseen, jotta mallien teksturointia voitaisiin jatkaa. Johtopäätöksenä todettakoon, että esitutkimuksen suorittamista käytettävistä työkaluista ei kannata vähätellä.

Meshien mallintamisen jälkeen on vielä suoritettava mallien teksturointi. Luvussa 2 viitataan siihen, että tekstuurien mallintamiseen on olemassa useita keinoja. Tämä on yllättävän toden mukainen toteamus. Korkeamman taitotason omaavat henkilöt voivat tehdä omat tekstuurinsa aina digitaalisista valokuvista lähtien. He ottavat valokuvan, valitsevat halutun pinnan kuvan käsittelyohjelmalla ja editoivat siitä litteän pinnan, joka sopii paremmin pintatekstuuriksi. Tällöin myös saatetaan käyttää useita pintoja toistuvuuden välttämiseksi ja editoidaan pinnat yhteensopiviksi saumojen välttämiseksi.

Tässä opinnäytetyössä tekstuurit tehtiin UVW-map tekniikalla (paria poikkeusta lukuun ottamatta). Tässä mallin mesh avataan tasaiseksi kuvaksi ja renderöidään. Tätä renderöityä kuvaa voidaan sitten käyttää vertauksena kuvan käsittelyohjelmassa sille, miten tekstuuri kuviot sijoitetaan oikein kullekin pinnalle.

Kukin objekti siis avataan siten, että yksittäiset pinnat ovat litteässä tasossa suhteessa toisiinsa ja haluttu pintatekstuuri asetellaan oikeaan asentoon suhteessa pintaan, miten se näkyisi taas 3D-muodossa. Tästä saatu kuva tuodaan puolestaan takaisin 3D-ohjelmaan jolloin tätä käytetään tekstuurikarttana, joka asetellaan tarkoitettuun objektiin. (republicofcode.com, 2013.) *Kuvassa 4 on aluksen keulaan tarkoitettu UVW-kartta.*

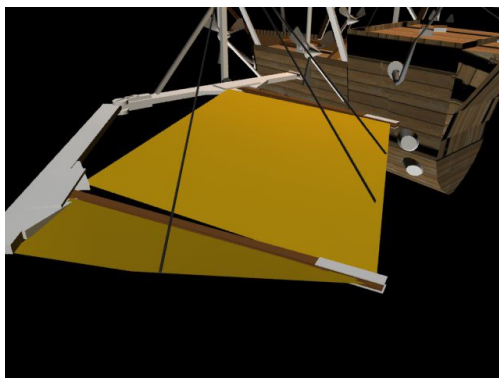


**KUVA 4: UVW-map**

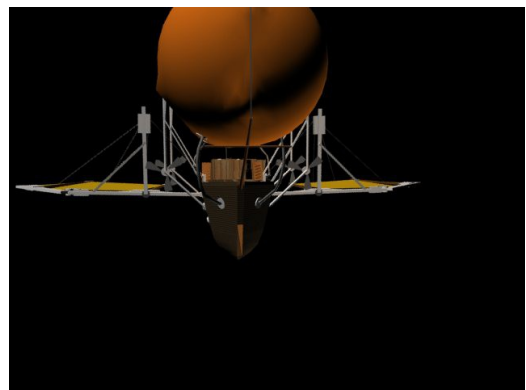
Tekstuurien kuvan käsittelyyn on monia sovelluksia, joten niitä ei tässä opinnäytetyössä listata yksityiskohtaisesti. Suositeltavampi metodi on suorittaa oma vertailunsa neutraalilta pohjalta, jolloin osaa paremmin punnita omat tarpeensa ja rajoituksensa tarjontaa vastaan. Toisin sanottuna oman tutkimuksen tekeminen on kannattavampaa kuin luottaa suoraan toisiin mielipiteisiin. Esimerkiksi tekstuurien muokkaamiseen käytettiin tässä työssä Gimp-ohjelmaa 3DS-Maxin kanssa ja se valittiin siksi, koska se sopi parhaiten budjettiin (ilmainen), aikarajoituksiin ja käyttötaitoon (aikaisempi kokemus Adoben PhotoShopista).

On kannattavaa, että testaa omia tekniikoitaan yksinkertaisimmilla objekteilla, joilla ei ole varsinaisesti mitään tekemistä projektin kanssa. Tämän kaltaisia objekteja luodaan juuri testausta varten. Näillä on hyvä testata tekniikoita joista ei ole varma mutta haluaa kuitenkin sisällyttää ne projektiin. Tällöin vältetään se, että soveltaa malliin tekniikoita jotka eivät toimikaan odotetusti ja sitten hätäillen yritetään saada niitä toimimaan. Lopulta, kun saadaan halutun näköinen tulos, niin se onkin juuri hätäisesti tehdyn näköinen ja huonolaatuinen. Testaamalla tekniikoitaan testimalleihin hiotaan osaamistaan eri tekniikoissa, jolloin vältetään mahdolliset virheet, kun sovelletaan niitä varsinaisiin malleihin.

Todettakoon, että on projektin kannalta hyvä, jos mallintaja pyrkii lähestymään työtään tehokkuuden kannalta. Jos kysytään asiantuntijoilta, vastauksessa painotetaan varmasti tehokkuutta yhtenä lähestymisnäkökantana. Tätä voi soveltaa esimerkiksi malleissa, joissa on useampi osa. Tässä projektissa sovellettiin Top-Down menetelmää, jossa mallin osat kasattiin käsin keskimmäisen osan ympärille. Muita käytettäviä menetelmiä esimerkissä olisi ollut Bottom-Up menetelmä, jossa valmiit osat tuodaan kokoonpanoon tai hybridimenetelmä, joka on yhdistelmä kahdesta edellisestä. (Hietikko 2007, 141.)



**KUVA 6: Erikseen kasattu siipi**



**KUVA 5: Siivet liitettynä runkoon**

## 2.2 Mallintamisen sovellukset

Voidaan ehkä ilman ennakkokäsityksiä olettaa, että mallinnusta käytetään pelien tekemisessä laajasti. Varsinainen sisältö, jonka pelaaja näkee on mallinnuksen aikaansaannosta. Vertauskuvauksellisesti sanottuna: koodi on elämä ja maailma tulee mallinnuksesta.

Tämä ”maailma” puolestaan koostuu monista objekteista, niin kuin maailma yleensä. Esimerkiksi: otetaan vaikka skenaario pelimaailmasta, joka koostuu seuraavasta asetelmasta. Meri. Meressä on saari. Saari toimii kenttänä. Näissä kolmessa asiassa on jo potentiaalia soveltaa monia asioita kolmiulotteisia malleja rakentaessa.

Käsitellään ensin merta. Meri koostuu tosielämässä suolavedestä, mutta peleissä maailman koostumus perustuu suunnittelijan näkemykseen. Tapauksesta riippuen veden koostumus voi olla jotain aivan muuta kuin vettä. Oletetaan tässä tapauksessa meren koostuvan vedestä.

Perustasolla vesi voi olla yksinkertaisesti tekstuuripinta. Tekstuuri on mallintamisessa käytettävä kuvio, joka asetetaan 3D-mallin pintaan. Tekstuurien avulla mallin pinta saadaan koostumaan tietyistä materiaaleista, kuten metallista, nahkasta tai puusta. Kuvio on yleensä monimutkainen ja ei-satunnainen, jotta tekstuuria voidaan käyttää laajemmalla alueella monistamalla kuviota vieriviereen. Tämä metodi on yleisimpiä käytettyjä pinnoitusmenetelmiä. Näyttävyyden takia tekstuureja kuitenkin tehdään myös skannaamalla oikeita pintoja tekstuurinäytteiksi tai niitä mallinnetaan jopa käsin.

Vesi kuitenkin on harvoin liikkumatonta. Veden liikuttamiseen on useita keinoja, tai illuusion luomiseen sen nestetilasta. Mallintaja voi valita pohjavärin tekstuurille, mielellään jotain kontekstiin sopivaa, ja lisätä siihen sopivia efektejä. Esimerkiksi tekstuurikuvio voi koostua taustasta jonka edustalle animoidaan laineita simuloivia juovia. Kuvio sitten levitetään veden pinnalle tekstuuriksi ja etäältä katsottuna voidaan luulla vettä liikkuvaksi.

Lähempää tarkastelua peruskeinoilla valmistetut pinnat eivät tietenkään kestä. On kuitenkin jo pelejä joissa vesi käyttäytyy nestemäisesti läheltä tarkasteltuna. Tähänkin on useampi metodi. Voidaan esimerkiksi lisätä pelimoottoriin mekaniikka veden käyttäytymistä varten, jolloin koodaajat ja mallintajat työskentelevät suhteellisen lähekkäin.

Tällöin tutkia ja soveltaa nestedynamiikkaa halutun ja vaikuttavan tuloksen aikaan saamiseksi.

Toinen esimerkki: voidaan animoida vetenä toimivan pinnan polygoneihin liikerata tai liikeratoja, jotka luovat illuusion aalloista. Nämä aallot sitten monistetaan useaksi kappaleeksi, jotka lopulta muodostavat liikkuvan meren. Tämä on toki työläs ja aikaa vievä vaihtoehto, koska vaaditaan tarkkaa asettelua, että aallot tuntuvat olevan yhtenäistä liikettä eivätkä itsenäisiä kopioituja pintoja. Tällöin animoinninkin tulisi olla tarkkaan suunniteltu ja toteutettu. Tällöin tietenkin voidaan olettaa, että mallintajalla on jonkinlainen peruskäsitys siitä, miten aalto käyttäytyy. Jos ei niin oletetaan, että hän ainakin perehtyy asiaan.

Saari on osa maastoa pelimaailmassa, joten se mallintaminen on aika suoraviivaista. Tällöin mallintaja yleensä käyttää editoria, jolla hän mallintaa muunkin maaston. Kuten mainittiin saari on osa maastoa, joten on kysymys suunnittelusta millainen saaresta tulee. Tietenkin mallintajan on otettava myös huomioon kentän yleinen suunnittelu. Mitä kentässä tapahtuu, minkä toimesta ja miten.

Kentän konteksti määrää sen millainen saari on kysymyksessä. Onko saari arktista tundraa vai trooppista kasvustoa. Tällöin määräytyy mitä ympäristöön oikein laitetaan ja miten ja minne se laitetaan. Näin saadaan aikaiseksi tausta pelin kentälle.

Kenttä puolestaan on rakennettu saarelle. Kentällä on monia yksittäisiä peliobjekteja. Jokainen objekti puolestaan on mallinnettu jollain tavoin. Otetaan vaikka esimerkiksi tausta objekti, joka on tynnyri. Tynnyrille on annettava muoto mallintamalla sen rautalankamalli, eli mesh. Tämä malli puolestaan on päällystettävä sopivalla pinnalla (tekstuurilla) ja muunnettava kehitysympäristölle sopivaan tiedostoformaattiin (esim. .fbx tai .max).

Suhteellisen yksinkertainenkin objekti, kuten tynnyri voidaan puolestaan mallintaa eri tavoilla tai ohjelmilla ja saada aikaan haluttu lopputulos. Objektien monimutkaistuessa on mahdollista, että mallintamisen soveltaminen moninkertaistuu mallia kohden. Tällöin mallintajalta odotetaan useiden osa-alueiden soveltamista.

Esimerkiksi pelihahmo on suhteellisen monimutkainen mallinnettava objekti. Ensin on suoritettava perusmallinnus kuin edellä mainitussa tynnyriesimerkissä. Tämän jälkeen seuraa erikoistuneempia vaiheita.

Pelihahmo ei ole eloton objekti, joten perus meshin mallintamisen jälkeen, mallia pitää tietenkin jalostaa. Kasvojen osalta on tiedettävä esimerkiksi tarkemmat perusmittasuhteet, kuten suun suhde silmiin ja nenän suhde poskipäihin. Tämän kaltaisia tietoja tarvitaan luontevan hahmon luomiseksi. Sama pätee muuhun ruumiiseen. Tällöin voidaan esimerkiksi soveltaa taiteessa tai tieteessä käytettäviä tietoja ruumiin ja kasvojen mittasuhteista ja muokata meshiä tämän perusteella haluttuun muotoon.

Pelihahmon yleensä tarvitsee liikkua, joten tämä edellyttää animointia. Ellei pelaajan hahmo tietenkin sisällä paljon liikkuvia osia joiden kuluu liikkua liiketoiminnon aikana, kuten esimerkiksi kädet ja jalat, voidaan liikkuminen hoitaa ohjelmoinnilla. Kuitenkin tällöin voidaan tarvita jotain animointia liikkuviin osiin, kuten laivan purjeisiin, köysiin tai auton renkaisiin.

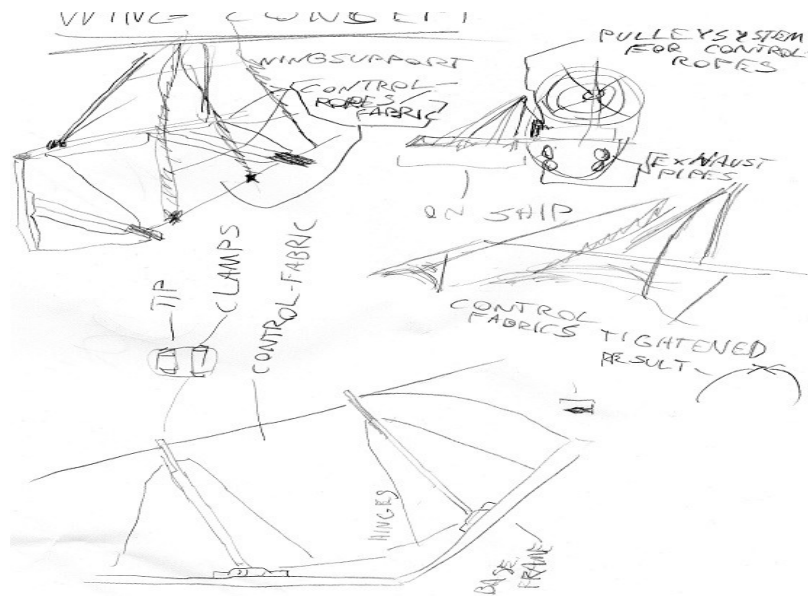
Yleensä uskottavaan pelihahmoon kuuluu useampia liikkuvia osia jotka toimivat yhdessä uskottavan toiminnon luomiseksi. Esimerkiksi kurottava käsi koostuu useasta saman aikaisesta tai ketjutetusta toiminnosta. Jos näitä toimintoja olisi liian vähän, lopputulos näyttäisi mekaaniselta karkealta ja kömpelöltä. Käsi ei kulu robotille, ellei sitä ole tarkoitettu niin joten hahmoissa pyritään aina sulaviin liikeratoihin. Tässä voidaan soveltaa anatomiaa, liikkeen tunnistusta tai liikeratojen tutkimusta riippuen mallintamisen tavoista.

On myös hyvä tietää, että jotkin objektit voivat koostua useista eri mallin komponenteista. Esimerkiksi auto. Autossa on muutoin paikallaan pysyvä runko, mutta liikkuvat pyörät. Auto voi olla myös jousitettu jolloin tämä otetaan huomioon teknisessä mielessä. Tällaiset mallit sisältävät liikkuvia osia. Liikkuvat osat sisältävät animointia, perustuen mallin tekniseen suunnitteluun, joten mallin suunnittelussa kysytään kykyä lähestyä mallia loogisesti, jotta se saadaan toimimaan järkevästi mallin kontekstin ja animoinnin kannalta. Toisin sanoen mallit kysyvät mallintajaltaan teknistä suunnittelukykyä.

Tätä kaikkea voi siis sisältyä yksinkertaiseenkin konseptiin, kuten pelikenttänä toimiva saari merellä, jossa on toimivia objekteja. Perusasetelmasta löytyykin monia mallintamisen sovelluksen vaihtoehtoja.



Tässä ei tarkoiteta sitä, että mallintajalla on oltava kattava teorianäytämys fysiikasta, taidehistoriasta tai, että mallintaja omaa valmiit insinöörin taidot. Niistä ei tietenkään ole mitään haittaa, mutta ainoa ehdoton vaatimus on, että mallintaja tunnustaa faktan, että tosimaailmassa ja fiktiossa objekteilla on kuitenkin konteksti pohjaiset sääntönsä joiden mukaan ne operoivat. Tämän myöntämisen jälkeen mallintajan on omaksuttava tietoa, jolla sovelletaan noita objektiin päteviä sääntöjä. Tätä tietoa puolestaan käytetään mallin rakentamisessa. Lopputuloksena tulisi näin ollen olla uskottava objekti, jonka



**KUVA 7: Aluksen siiven toimintakonsepti**

voidaan mieltää operoivan sääntöjensä mukaisesti. Toisin sanoen uskottava objekti.

Miten uskottava objektista, ympäristöstä tai hahmoista tulee, riippuu siitä miten hyvin malleissa on osattu soveltaa eri aloja, joiden kokonaisuus johtaa haluttuun efektiin. Tietenkään ei edellytetä, että jokaisena kertana, kun lähdetään rakentamaan uutta mallia, että mallintajan tulisi perusteellisesti täydentää tietovarastonsa, mitä tulee mallin jokaiseen alueeseen. On turhaa täydentää jo valmista tietoa pohjaa, mutta jos mallintajalta puuttuu hyvä tietoa pohja, on hyvä täydentää se. Kaikesta huolimatta mallintajan odotetaan kuitenkin käyttävän tietopohjaansa hyväkseen malleissaan.

### 3 MALLIEN EFEKTI

Kuvitellaan skenaario: Meillä on tyhjä 3D-ympäristö. Tähän ympäristöön on suunniteltu vain perus topographia, joka näyttää alueen perusmuodot. Kaikki muu on vielä li-

sättävä. Tähän ympäristöön sijoitetaan malli. Oletetaan, että malli kuuluu yhdelle monille puista joilla malliin on tarkoitus luoda metsäinen ympäristö.

Millä mallilla tahansa on jonkinlainen vaikutus tai efekti yleisöön. Edellä mainitun puun tapauksessa halutun efektin välittäminen voidaan aloittaa jo suunnittelu vaiheessa. Esimerkiksi onko kyseessä arktinen vai trooppinen ympäristö johon puu sijoitetaan. Yleisesti suunniteltu puu voi tuoda väärän efektin, koska se ei sovi pelin konseptiin. Tällöin yleisö voi saada väärän käsityksen ympäristöstä jossa puu sijaitsee. Puuta tai muuta mallia valmistaessa on hyvä perehtyä lähdemateriaaliin, johon malli perustuu, oli kyseinen materiaali sitten tietosanakirjassa, tai suunnittelijan päässä. Onko kyseessä esimerkiksi trooppisen vyöhykkeen puu tai fantasiamaailmassa sijaitseva puulajike. *Kuvassa 8 on näyte texturematesta löydetystä(2013) metallitekstuurista, jonka tarkoituksena on luoda objektille käyttöä kokenut tuntuma.*



**KUVA 8: Käytetty tekstuuripinta**

Malleilla tai ennemminkin niiden efekteillä lisätään pelien miellyttävyyttä ja mukautempavuutta yleisölle. Tällöin asetetaan tunnelmaa, varsinaista kontekstia, fyysisiä rajoja ja ennen kaikkea näkyvää maailmaa, siten kuin se on alun perin suunniteltu. Hyvinkin tehty malli ei toimi jos se ei välitä jonkinlaista efektiä. Tällöin malli jolla ei ole efektiä tai ei välitä tarkoituksen mukaista efektiä, on malli jolla ei ole väliä pelin kannalta. Efektin ei tarvitse olla suuri, joten yksinkertainenkin riittää. Pelaajan katsoessa jotain ja todetessa ”Joo minä tiedän mikä tuo on” on minimaalinen efekti jo saavutettu.

Käänteisesti peliin tuleva malli voidaan pitää käytettävän muistin ja muiden teknisten rajoitusten sisässä pienemmälläkin laadulla ja silti välittää haluttu efekti. Kuten edellä mainittiin, pienikin efekti riittää. Tässä luvussa pyritään määrittämään tarkemmin esteettiset ominaisuudet, jotka tulisi sisällyttää peleihin tuleviin malleihin.

### 3.1 Tulkinta

Tulkinnan kannalta otetaan huomioon miten kukin malli vastaa aiottua konseptia ja myös se, miten yleisö saattaa tulkita mallit. Sanaa ”saattaa” käytetään tässä, koska yleisön reaktiot selviävät yleensä, kun peli on viimeistely. Tämä on yksi suuresti vaikuttava syy miksi yleensä käytetään testiryhmiä ennen julkistamista.

Tulkinnalla tarkoitetaan tässä sitä, että mitä viestejä halutaan välittää mallien välityksellä. Viestien monimutkaisuus riippuu pitkälti pelissä käytettävässä tekniikasta ja objektien ja allien merkityksestä pelissä. Muistin asettamat rajoitukset saattavat vähentää mallien laatua, mutta silloinkin pystytään välittämään peruskonsepteja.

Esimerkiksi pinnat. Oletetaan, että peliympäristössä on symmetrinen kasa palikoita, joilla on tekstuuri. Tekstuuri puolestaan koostuu eri sävyisistä punaruskeista pikseleistä. Näinkin vähällä voidaan välittää viesti, että ympäristössä on tuttu muoto, joka käsitetään seinäksi. Tällä seinällä on myös tunnistettava koostumus. Ihmismieli hakee lähintä vastaavuutta ja suurella todennäköisyydellä päätyy punatiiliseinänsä. Virhemarginaali on tietenkin aina olemassa, mutta hyvin tehty työ jättää sen häviävän pieneksi.

Vastaavasti huonosti tehty työ näkyy ja tuntuu. Jos edellä mainitussa punatiili seinässä ei olisi mitään tunnistettavia muotoja, esimerkiksi laastin luomia saumoja tai mitään saumoja ylipäättänsä. Tällöin pelaaja tunnistaa objektin palikaksi, joka on punainen ruskealla sävyllä. Pelaajalla ei ole mitään käsitystä mistä materiaalista palikka on tehty tai mikä merkitys palikalla oikein on. Siis sen lisäksi, että se on tiellä. Pelaaja ei miellä seinää seinäksi jollei sille perusteta kontekstia efektin välityksellä.

Työkalujen kehittyessä on tullut mahdolliseksi välittää monimutkaisempia viestejä korkealaatuisimmilla malleilla. Grafiikkaprosessorien valmistajat tuovat uusimman piirinsä markkinoille lähes vuoden välein ja uudet renderöinti-, mallinnus- ja animaatiotekniikat kehittyvät niiden rinnalla.

Tekijöiden ei siis tarvitse venyttää yleisön tulkintakykyä, vaan voivat välittää halutut viestit helpommin peliin sisällytettyjen mallien avulla. Nykyään tasoihin voidaan sisällyttää malli rakennuksesta, jolla on omat arkkitehtuuriset viittauksensa. Esimerkiksi onko kyseessä moderni rakennus vai vuosisadan vaihteessa rakennettu talo. Esimerkiksi tessalaation kehittyessä voidaan sisällyttää enemmän polygoneja mesheihin, pienemmällä muistivaatimuksella. ([www.nvidia.com](http://www.nvidia.com), 2010.)

Kehityksen myötä on helpompaa välittää efektejä pelimekaniikkaan, jonka pelaaja kokee varsinaisen pelin aikana. Tämä tarkoittaa sitä, että enää kehittäjien ei tarvitse rajoittaa pelaajan opastamista käyttöliittymään vaan voidaan istuttaa peliin malleja tai efektejä jotka toimivat yhdessä käyttöliittymän kanssa. Tämä ei pelinkehityksen alkuvaiheessa olisi ollut mahdollista, koska yksin kertaisesti teknologia ei sallinut sitä. Nykyään on tosin alettu lisätä peleihin elementtejä käyttöliittymän ulkopuolelle joiden tarkoitus on opastaa pelaajaa. Esimerkiksi maasta ylös ampuva valojuova, joka merkitsee waypoint- tai objective-pistettä on nykyään suosittu kehitysratkaisu.

On myös mahdollista korostaa malleja kentissä. Tämä on suhteellisen yleistä peleissä nykyään. Tähän on monia keinoja. Esimerkiksi mallin sijoitus, valaistus tai tekstuurit. Mallin korostamisella tarkoitetaan sitä, että jokin paikka tai objekti kentässä vetää huomiota puoleensa. Tällöin kyseessä oleva objekti erottuu taustasta. Tällöin pelaaja tulkitsee kyseisen objektin huomion arvoiseksi ja tutkii sitä tarkemmin. Tästä syystä tämä tekniikka sopii paremmin siihen, kun halutaan korostaa tehtävän päämääriä ja tehdä kentästä helpommin ymmärrettävä tai kiinnittää pelaajan huomio johonkin tiettyyn asiaan.

Otetaan esimerkkinä vaikka hitaasti alkava kauhupeli. Tarinan alussa päähahmo vedetään johonkin tapahtumaan mukaan. Hahmo saapuu ensimmäiseen kenttään ja aloittaa tutkimuksensa. Tiedetään, että jokin on vialla, mutta ei mikä. Pelaaja tutkii kenttää hahmon välityksellä ja sitten hän huomaa jotain, joka pistää silmään muusta kokonaisuudesta, sivusilmälläkin nähtynä. Muutoin harmaan puhuvassa huoneessa on seinälle verellä kirjoitettu viesti ”Don't go to the basement!”.

Tietenkin hahmo pelaajan toimesta lähtee kellariin huolimatta selvästä kehotuksesta olla tekemättä niin ja selvästi tässä on tarinankerronnan ja pelinkehityksen kannalta syvempiä psykologisia syitä siihen miksi ja miten, hahmo ja pelaaja saadaan käyttäytymään selvien ohjeiden vastaisesti. Tämä on kuitenkin syvempi aihe jota ei tässä opinnäytetyössä käsitellä. Mikä on tämän työn kannalta merkittävää on viesti itsessään.

Huoneen harmaa kontrasti antaa mahdollisuuden käyttää kirkkaita värejä jonkin korostamiseksi, joka tässä tapauksessa on viesti seinällä. Tumman punainen korostuu muuta huonetta vastaan hyvin. Tällöin kiinnitetään pelaajan huomio tiettyyn objektiin huoneessa.

Seuraavaksi kiinnitetään huomiota viestin materiaaliin. Käytännössä viesti voisi olla kirjoitettu punaisella spraymaalilla ja sillä ei olisi mitään suurempaa väliä, koska haluttu efekti välittyy pelaajalle. Hän ajattelee jo mitä kellarissa on. Se, että viestin pinta on veren kaltaisella tekstuurilla kirjoitettu, auttaa lisäämään alkuperäiseen viestiin ja pelin kontekstiin.

Pelaaja tietää, että jotain on vialla. Hänelle on kerrottu se jo ennen tähän huoneeseen saapumista. Kun hän näkee viestin, jonka kirjoitus materiaalin hän tunnistaa vereksi, hän tietää asioiden olevan vialla ja, että uhkana on myös väkivallan mahdollisuus. Tällöin lisätään pelon tuntua, joka auttaa tällaisen pelin kontekstissa.

Näin siis on välitetty pelaajalle viesti ja efekti korostamalla objektia ympäristössä tietyllä tavalla. Tämä palvelee tunnelmaa, tarinankerrontaa ja pelimekaniikkaa itsessään. Toisin sanoen hyvin suunniteltu ja toteutettu malli auttaa tarkoituksenmukaisessa tulkinna, jolloin pelaajan oma kokemus paranee.

### **3.2 Imersio/ Efekti**

Imersiolla tarkoitetaan sitä, kuinka mukaansatempaavana yleisö pitää koettua sisältöä. Pelaaja kokee siis imersion, kun hän tuntee olevansa mukana pelissä aisti tai tunne tasolla tai kun hänen tunnetasonsa vastaa mitä hänen tulisi tuntea kerronnallisessa tai tulkinta mielessä. Selvennykseksi aisteilla tarkoitetaan tässä kuulo- ja näköaistia ja sitä mitä pelaaja kokee peli ympäristössä.

Imersion saavuttaminen ei ole helpoin asia pelin kehityksessä. Sillä pieniltäkin vaikuttavat asiat voivat vaikuttaa kokemukseen. Pelaajat yleensä suhtautuvat intohimoisesti pelikokemuksiinsa ja niihin liittyviin seikkoihin. Esimerkiksi kaikki seikat, jotka vähentävät pelaajan sulavaa ja intuitiivista toimimista, voidaan kokea haittana ja tarpeettomana esteenä. (Colson 2007, 126.)

Pelaajan imersio voidaan saavuttaa monilla keinoilla. Kaikki näistä keinoista eivät rajoitu mallien laatuun ja tekniikkaan. Imersion saavuttamiseen voidaan käyttää animointia, tekoälyä, pelimekaniikkaa, ääniefektejä, ääninäyttelyä, soundtrackkeja tai käsikirjoitusta. Keinot eivät tietenkään rajoitu tähän ja kunkin osa-alueen kohdalla voi-

daan eritellä pienemmät yksityiskohdat, joita voidaan hyödyntää imersion aikaan saamiseksi.

Edellisessä luvussa mainittiin esimerkki kauhupelistä. Lähdetään siis lisäämään elementtejä huoneeseen josta viesti löytyi. Huoneen tekstuurit ovat hyvä esimerkki. Me tiedämme entuudestaan, että huone on harmaan sävyinen mutta emme juuri muuta. Joten miten voimme käyttää tekstuureja siten, että huoneen tutkimisesta saadaan pelaajan kannalta imerssiivisempi kokemus?

Kontekstin mukaan tiedämme, että jotain on vialla. Siispä voimme lisätä vinkkejä tekstuureihin, jotka viittaavat tähän. Huomioitakoon, että verellä kirjoitettu viesti on suurella todennäköisyydellä itekin tekstuuri, joten tässä tarkoitetaan viestin luomaan efektiin lisäämistä, muuntamalla ympäristöä. Aloitetaan peruspinnosta, kuten katto, lattia ja seinät.

Pintoihin voidaan lisätä lisää verijälkiä jolloin oikeutetaan pelihahmo toteamaan esimerkiksi ”What happened here?” kun hän astuu sisään huoneeseen. Tätä dialogia ei tietenkään tarvitse lisätä sillä pelaaja varmaan ajattelee tätä itekin, jolloin itsestäänselvyyksien toteaminen voi ennemminkin rikkoa imersiota kuin syventää sitä, mutta tämä on enemmänkin suunnittelukysymys.

Pintoihin voidaan lisätä myös suuria kynnen jälkiä, osumavahinkoja, luodin reikiä ja muuta vastaavaa. Kaikki mitä voidaan lisätä riippuu pitkälti pelin konseptista. Käytännössä pinnat voidaan päällystää miten vain konseptin rajoissa. Esimerkiksi mainoskyltti upotettuna tekstuuriin olisi räikeä esimerkki konseptin rikkomisesta, mutta jos mainoskyltille rakennetaan oma mallinsa ja päällystetään kylttitekstuurilla silloin siitä voi tulla objekti, joka sopii konseptiin.

Seuraavaksi tarkastellaan huoneen kaavoitusta tai tarkemmin sanottuna sisältöä. Jos huoneessa on objekteja ne voidaan järjestellä siten, että luodaan vaikutelma siellä tapahtuneesta sekasorosta, jolloin lisätään yleiseen tunnelmaan. Tuolit voivat olla selälään, pöydät kyljellään, lattialla voi olla luodin hylyjä tai hajalleen levitettyjä papereita. Jälleen kerran sisältö määräytyy konseptin mukaan, mutta tällaiset yksityiskohdat lisäävät tunnelmaan ja immerssiivisyyteen.

Tämä tietenkin edellyttää mallintajalta suhteessa lisää työtä yksityiskohtia kohden. Tuoleille ja muille objekteille pitää rakentaa meshit ja niille tekstuurit. Tämä ennen kuin ne voidaan lisätä kenttään. Sitten on vielä päätettävä miten ne vaikuttavat kent-

tään. Ovatko objektit ohjelmoituja objekteja joilla on oma paikkansa kenttäsuunnittelussa ja joilla on omat fysiikkansa ja koodinsa. Vai ovatko ne vain staattisia meshejä jotka ovat vain osa huoneen taustaa. Joka tapauksessa ne kuitenkin vievät muistia, joten on kaiken lisäksi käytettävä harkintaa kuinka paljon yksityiskohtia lisää huoneeseen.

Huoneessa on siis sisältö ja tekstuurit. Mitä muuta voidaan vielä tehdä? Voidaan valaista huone. Huoneessa on tapahtunut jotain. Jotain joka on aiheuttanut vahinkoa huoneelle. Siispä valonlähteet voidaan laittaa välkkymään, kuin kattolamput olisivat vahingoittuneet tapahtumassa. Valon tehoa voidaan myös laskea jolloin saadaan hämärämpi ympäristö, joka puolestaan lisää jännitystä. Esimerkissä on kyse kauhugenrestä jolloin tuo tunnelma on se mitä haetaan.

Näin voidaan siis soveltaa mallinnusta imersion aikaan saamiseksi. Tietenkin tähän voidaan lisätä monia muita tekijöitä joilla imersiota syvennetään entisestään. Musiikki raita voidaan asettaa niin, että se soittaa sopivaa musiikkia määrätyllä hetkellä. ”Skriptattuja” tapahtumia voidaan ohjelmoida niihin hetkiin, kun niiden on sopiva tapahtua suunnittelun näkökannalta. Tietenkin äänitys ja hahmojen näyttely vaikuttaa asiaan. Nämä tosin tapahtuvat mallintamisen lisäksi, joten ne ansaitsevat oman käsittelynsä. Vaikka ne ansaitsevat maininnan niin tässä keskitytään kuitenkin, miten mallintamista voidaan käyttää imersion luomiseen.

Imersiota voidaan pitää tärkeänä ominaisuutena pelin menestyksen kannalta. Useiden julkaistujen pelien keskellä pelaaja muistaa paremmin pelit, jotka vetosivat häneen jollakin tasolla. Mitä enemmän peli vetoaa häneen tai mitä useammalla tasolla pelaajaan vedotaan, sitä todennäköisemmin hän pitää pelistä. Tämä on syytä pitää yhtenä periaatteena pelinkehityksen kaikilla osa-alueilla.

Tästä otettakoon esimerkkinä pelien käyttöliittymissä käytettävät grafiikat ja niiden asettelu. Pelin kontekstia voidaan pitää pohjana jolle rakennetaan grafiikat, joilla ei ole suoraa vaikutusta pelaamiseen. Esimerkiksi fantasiapainotteisen pelin grafiikat voivat saada kelttiläisiä viitteitä, scifi pelit käyttävät cyberpunk-tyylisiä grafiikoita ja muun genren peli käyttää kontekstiin sopivia grafiikoita.

Käyttöliittymän asettelun kannalta otetaan huomioon mikä tyyli on parhaiten tasapainossa immersiiivisyyden ja pelattavuuden välillä. Grafiikka voi olla tyyllille uskollinen esimerkiksi vihjeikkunoissa, mutta jos nuo ikkunat käyvät pelaajan tielle haittaamalla

pelaamista, tai muuten pistämällä silmään olemalla liian korostettuja, voi imersio tällöin kärsiä.

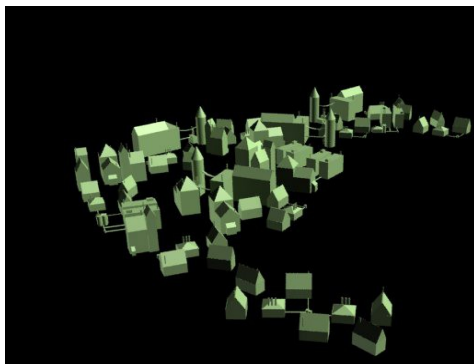
Vastaavasti käyttöliittymän puute suuremmissa kentissä voi haitata pelaamista, hämmentämällä pelaajaa, jolloin tämä ei ole aivan varma mitä hänen olisi tarkoitus tehdä tai minne hänen kuuluisi kulkea. Tästä oiva esimerkki on pelissä käytettävä kompassi. Sitä ei aina tarvita, mutta siihen on voitava välillä viitattava. Siispä se on elementti, joka voidaan tuoda esiin halutessa, mutta pidetään piilossa muulloin immersiiivisyyden säilyttämiseksi.

Pelaajan kokiessa imersio saavutetaan piste, jossa kaikki ”loksahtaa” kohdalleen. Niin taiteellisessa kuin mekaanisessakin mielessä. Tämä on hetki jolloin ääni, kuva, tarina ja ohjelmointi ovat juuri oikeassa mittasuhteessa keskenään aiheuttaen, että pelaaja kokee pelin oikeassa mielentilassa, jolloin hän saa pelistä enemmän irti kokemuksena.

### 3.3 Mallien vaikutus kokonaisuuteen

Mallit vaikuttavat pelien kokonaisuuteen monilla tavoilla. Ne luovat ja visualisoivat kontekstin, luovat immersiiivisyyttä ja antavat pelimekaniikan käytettäviä objekteja. Ne myös voivat vaikuttaa suoritukseen, aiheuttaa yhteensopivuusongelmia ja rikkoa pelin immersiiivisyyden huonolla laadullaan.

Selvitetään ensin muutama termi. Objekteilla tarkoitetaan tässä luvussa pelissä olevia interaktiivisia malleja. Staattisilla malleilla tarkoitetaan ympäristön malleja. Animaatiot ovat suhteellisen selvä käsite, mutta selvyuden vuoksi, niillä tarkoitetaan malleihin sisällettyä animaatiota. Siis lyhyemmin viitaten: objekti, malli ja animaatio.



**KUVA 9: Konteksitssa staattinen malli**



**KUVA 10: Kontekstin mukainen objekti**



Käsitellään ensiksi mallien tarvetta. Miksi peli yleensä tarvitsee malleja? Syy on selvä sillä koodaamalla ei voida rakentaa mallien 3D- meshejä. Mesh on siis varsinainen 3D-malli jota käytetään yhdessä tekstuurien kanssa mallissa. Peli kuitenkin tarvitsee malleja vähintään objekteiksi, joita voidaan käyttää havainnollistamaan pelin toimintoja. Tekstipohjaisissa roolipeleissäkin selostetaan skenaariot joihin pelaaja ottaa osaa, kuvailemalla mitä tapahtuu, missä tapahtuu ja miten tapahtuu. Kuvauksessa sisällytetään myös se mitä tapahtumaan kuuluu. Eli onko paikalla objekteja, miten monta ja mitä objekteja. Tämä on yksi perusteellisimmista syistä miksi peliprojekti tarvitsee malleja.

Pelien kehityksessä keskitytään nykyään tarkemmin median itsensä muuntamisessa näkymättömäksi yleisölle. Eli mallintajien ja animaattoreiden tavoitteena on luoda synteettinen maailma, joka säilyy omien sääntöjensä mukaisesti ja sallii yleisön katsoa tuota maailmaa median läpi. Colson viittaa tähän (2007, 34.) tilana, jossa yleisö ei ole tietoinen muusta kuin luodusta maailmasta ja juonen kerronnasta.

Tämän saavuttamiseksi pyritään kiinnittämään huomiota ilmiöihin ja ominaisuuksiin, joita ihmiset eivät luonnostaan ajattele yleensä. Erilaisten sisältöjen ominaisuudet, luonnollinen liike ja miten objektit käyttäytyvät painovoiman johdosta. Nämä ovat vain pieni esimerkkien joukko asioista, jotka on otettava huomioon, tulkittava digitaalisessa ympäristössä ja lopuksi mallinnettava, jotta saadaan aikaiseksi ekvivalentti rakenne. (Colson 2007, 34.)

Edellä mainittiin pelin objektit. Objektin tarkempi määrittäminen on se malli, joka on jollain lailla interaktiivinen tai sillä on jonkinlainen vaikutus tai osa pelin mekaniikassa. Esimerkiksi NPC-hahmoja, käytettäviä tavaroita, asusteita, ajoneuvoja, aseita, vihollisia ja muuta vastaavaa. Nämä kaikki tarvitsevat omat mallinsa. Tämä on itsestään selvää, koska ei voi korvata oven mallia esimerkiksi humanoidi vihollisella. Tulos: pelissä oven virkaa toimittavat samat viholliset, joita vastaan pelaaja taisteli hetki sitten. Hieinan äärimmäinen esimerkki, joten otetaan toinen esimerkki tästä projektista.

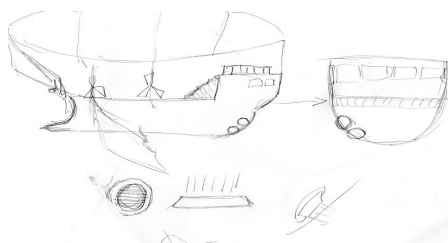
Varhain projektissa päätettiin, että pelissä olisi faktioita joihin pelaaja voisi liittyä. Piraatit, military tai commerce. Tässä kuitenkin tulee vastaan edellä mainittu ongelma. Tässä vaiheessa projektia ensimmäinen aluksen mesh oli jo valmis, mutta tämä oli tarkoitettu puolueettomalle pelaajalle. Siis tätä samaa mallia ei voitaisi käyttää ainoana aluksena kaikille faktioille. Tämä oli itsestään selvyyttä sillä se näkyisi negatiivisesti lopputuloksessa ja muutenkin se näyttäisi hölmöltä ja rikkoisi kaiken imersion jos kai-

killa näkyisi tismalleen sama alus. Tämä korjautui helposti. Lopputuloksena projektista löytyy mallit ainakin rahtialukselle, perusalukselle, military-alukselle ja nopealle alukselle.

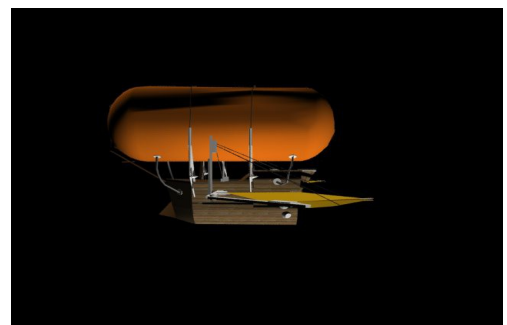
Pelin konteksti määrittää tietenkin ympäristön tarpeet. Näiden tarpeiden mukaan tehdään mallit kuhunkin ympäristöön ja kenttään. Tämä pätee niin edellä mainittuihin objekteihin, kuin staattisiin malleihinkin. Tehtyjen mallien uskollisuus asetetulle kontekstille vaikuttaa puolestaan kentän lopulliseen ulkoasuun.

Tämä on suhteellisen suoraviivainen käsite. Tässä projektissa pyritään luomaan laiva peli. Pelissä on siis laivoja. Laiva on kone, jonka tarkoitus on liikkua paikkojen välillä vapaasti. Siispä tarvitaan iso kenttä laivoille liikuttavaksi.

Sitten päätettiin, että peli toteutettaisiin steampunk-genressä ja, että laivat olisivatkin ilmalaivoja. Siispä kenttä muuttuikin nyt nopeasti. Laivat liikkuvatkin nyt ilmassa ja niiden liikkuminen ei enää rajoitukaan veteen. Tämä tietenkin muuttaa laivojen suunnittelua. Miten ne pysyvät ilmassa? Mitä työntövoimaa ne käyttävät? Miten maailma oikein toimii?



**KUVA12: Uudistetun kontekstin alusluonnos**



**KUVA 11: Lopputulos**

Tämä näkyy laivojen suunnittelussa. Esimerkiksi siinä, miten niistä löytyy zeppeliinejä, höyrykäyttöisiä suihkumoottoreita, lepakkomaisia kangas-siipiä ja muita genrelle tyypillisiä ilmalaivan varusteita. Konteksti määräsi miltä laivat näyttäisivät ja toimisivat.

Maailman kontekstissa alukset tarvitsevat jonkinlaisen sataman. Suhteellisen suoraviivainen käsite tässäkin kontekstissa. Lisätään muokattavia kaupunki malleja kenttään joihin projektin ohjelmointipuoli lisää mekaniikan, kun se on suunniteltu. Sitten joku projektissa keksii, että ”Minulla on idea. Lisätään leijuvia saaria joissa on oma kau-

punkiyhteisönsä!”. Tietenkään kun kuulee toimivan idean, sitä ei tietenkään voi jättää pois kokeilematta, joten peliä varten piti suunnitella leijuva saaren konsepti.

Projektissa käytiin lyhyt neuvottelu mitkä ovat maailman säännöt leijuva saaria kohtaan. Eli millä periaatteella saari pysyy ilmassa ja mitä varten. Sitten sille rakennettiin oma meshinsä omista osistaan valmiiksi teksturointia varten. Pelissä on siis leijuva saari, joka myös toimii satamana.

Lopputuloksena on ympäristö, joka noudattaa sille asetettuja rajoja. Kyseessä olevat rajat tässä tapauksessa ovat pelin genre ja sen mekaniikka. Laivojen suunnittelu estää niiden käyttämisen missään muualla kuin ilmassa johon ne on tarkoitettu ja niiden suunnittelu pysyy myös uskollisena genrelle. Tällöin pelistä ei löydy jotain, joka ei sinne kuulu. Tässä tapauksessa tähtialuksia tai purjelaivoja, kuten alkuperäisidean mukaan olisi pelissä ollut kyse.

Nyt mallit myös vaikuttavat kokonaisuuteen halutulla tavalla. Ne auttavat rakentamaan maailman, joka on suunniteltu noudattamaan tiettyjä periaatteita. Ne auttavat yhtenäisen kokonaiskuvan rakentamisessa ja ne on toteutettu tähän maailmaan sopivalla tavalla. Mallien tyyllinen toteutus auttaa siinä, että mallit sopivat niille asetettuun kontekstiin. Tässä tapauksessa steam-punk ilmalaivat sopivat steam-punk universumiin.

Mallien toteutus ei kuitenkaan rajoitu pelkkään tyyliin. Tämä on tärkeä yksityiskohta muistaa. Tyyli ja konteksti ovat ohjenuoria joiden mukaan mallia tehdään. Ne eivät ole ehdottomia määräyksiä vaan suunta johon mallintajan on vietävä malliaan. Mallintajan on itse sovellettava konseptia, joka on hänelle annettu.

Vastakohtana edelliselle esimerkille tarkastellaan sitä mitä ei pidä tehdä kun pyritään ehjään kokonaisuuteen. Ensinnä tiedä pelisi konsepti. Tämä on usein toistettu sana tässä opin näytetyössä syystä. On ilmiselvää, että jos sinulla ei ole mitään aavistusta mitä ohjenuoria sinun tulisi noudattaa mallintaessa niin sinun ei kannata edes aloittaa ennen kuin on jokin käsitys mistä koko pelissä on kyse. On totta, että yksi peliprojektin aloitettava kipinä on taideluonnos, mutta tuolloinkin tekijöillä on jonkinlainen konsepti valmiina. Ilman sitä tuotos on väistämättä sekavaa ja tuntuu satunnaiselta, vaikka pystyisikin sijoittamaan ne projektiin muokkaamalla malleja ja tekstuureita. Alkuperäiset vaikutteet silti paistavat läpi ja tällöin oma visio hukkuu sekavuuteen.

Muista kommunikointi. Tämä on sääntö, joka on peliprojekteissa ehdoton. Ellet ollut mukana muodostamassa alkuperäisideaa on todennäköistä, että saat mallien toimeksiantot muilta projektin jäseniltä. Projektin jäsenten välinen kommunikointi pitää mallintajan tietoisena asioista, auttaa ideoiden jakamisessa ja ennen kaikkea se välittää mallintajalle konseptin. On itsestään selvää siis, miten tärkeä osa kommunikaatio on mallintajan työtä projektissa, vaikka sillä ei ole suoraa yhteyttä työhön itseensä. Kommunikaatio tiedottaa mallintajalle prioriteetit kullakin hetkellä. Milloin tarvitaan tekstuurit tiettyyn malliin ja milloin tietty mesh tarvitaan viimeistään. Kommunikaatio tiedottaa mallintajalle myös onko jokin ominaisuus leikattu pelistä ja säästää mallintajalta potentiaalisesti työtunteja, jotka muutoin olisi haaskattu malliin, joka ei kuitenkaan päädy peliin.

On siis syytä olla oma-aloitteinen mitä tulee kommunikaatioon. Vaikka kommunikaatio olisi projektissa järjestetty hyvin, oma-aloitteisuus lisää kommunikaation tehokkuutta entisestään. Sitä kannattaa siis pitää yllä.

Tunne lähdemateriaalisi. Ei tietenkään edellytetä, että mallintaja on yleinen trivian ja maailman asiantuntija. Jos hän ei kuitenkaan tiedä mistä puhutaan, kun projektin palaverissa esitellään hänelle konsepti, jossa käsitellään cyber-punk hahmoa, kannattaa hänen täydentää tietokantojaan eri genreistä. Tällöin vältetään operoimasta täysin pimeässä sen suhteen mitä projekti tarvitsee ja saavutetaan yhtenäisempi kokonaisuus helpommalla.

#### **4 ANIMOINNISTA**

Yksinkertaisesti sanottuna, animointi on liikeratojen sovittamista malleihin, jotta saadaan aikaan tarkoituksenmukainen liike. Animaatiota ei projektin aikana tarvittu paljoakaan, mutta se on kuitenkin merkittävä osa mallintamista, etenkin peleissä, joten se vaatii oman käsittelynsä. Tässä luvussa käsitellään pinta puolisesti animaatiota, sen teoriaa ja sen mahdollisia sovelluksia.

Peleissä näkee liikettä kaikkialla. Kaikki tämä liike on animaatiota määritelmän mukaisesti. Kaikki liikkeet eivät ole välttämättä toteutettu samoilla välineillä tai metodeilla, mutta ne ovat objektien liikkeitä peliympäristössä, joten ne lasketaan animaatioksi. Määritellään seuraavaksi animaatio vähän tarkemmin.

Animaatio on käytännössä elottoman objektin liikuttamista tavalla, joka sopii objektin omaan kontekstiin. Esimerkiksi propelli on esine, joka pyörii akselinsa ympäri tosielämässä. Kun sen näkee missä tahansa virtuaaliympäristössä, oletetaan sen käyttäytyvän samalla tavalla kuin sen tosielämän vastine. Eli tarkoituksenmukainen animaatio laittaisi sen pyörimään akselinsa ympäri.

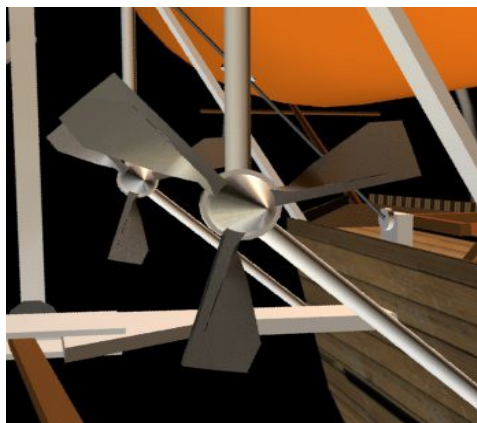
Animaatio itsessään ei toimi ilman kontekstia. Animaatio ilman kontekstia näyttää oudolta ja jos animaatiolla ei ole mitään rajoituksia niin se myös näyttää luonnottomalta. Rajoituksilla tarkoitetaan tässä pelin omia sääntöjä. Pelin fysiikka määrää esimerkiksi sen, mihin objekteihin vaikuttaa painovoima ja mihin ei.

Esimerkiksi, jos objekti löydetään ympäristössä leijumassa puolen metrin korkeudessa maasta. Tämä on pelin sääntöjen rikkomista. Toinen esimerkki on hahmojen satunnainen teleporttaaminen tai levitoiminen. Kaikki edellä mainitut tapaukset ovat useimmiten bugeja eli virheitä ohjelmoinnissa. Oletuksena on tietenkin, että kukaan pelin kehittäjä ei tietoisesti halua rikkoa omia sääntöjään.

Joka tapauksessa animaatioon pätevät samat säännöt kuin muihinkin mallintamisen osa-alueisiin, mitä tulee kontekstiin. Käytännössä animaatio on oma alalajinsa, mutta se sisältyy kuitenkin mallintamiseen. Tällöin samoja periaatteita voidaan soveltaa mallintamisesta animaatioon.

Animaatioon pätevät myös samat muistin ja laitteiston asettamat rajoitukset, kuin malleihinkin. Mallien meshit ja tekstuurit vievät oman osansa saatavilla olevasta muistista, mutta animaatio kuluttaa vielä oman osansa erikseen. Ottaen huomioon mikä määrä animaatiota peleissä nykyään on, niin voidaan päätellä, ettei se ole aivan pieni osa muistia, jonka animaatio vie.

Animaatio on myös tärkeä osa peliä itseään. Animaatioksi voidaan laskea myös objektien liikuttaminen ohjelmoinnin avulla. Esimerkiksi projektin pelissä alusten propelleille on asetettu koodiin komento kiertää propellien meshiä oman akselinsa ympäri. Tällöin on saatu aikaan propellien liikkuminen koodilla animoimatta tätä toimintoa erikseen.



**KUVA 13: Propellin mesh**

Pelien osalta kaikki liike mitä pelaaja havaitsee on animaatiota. Tähän sisältyvät niin pelaajan suorittamat toiminnot ja liikkeet, hahmojen reaktiot, ajoneuvojen liikkeet ja ympäristön tapahtumat. Animaatio on pelin kannalta tärkeämpää kuin mallien meshien polygonien määrä tai tekstuurien koko, joten animaatioilla tulisi olla prioriteetti jos editoinnille on tarvetta. Mieluummin pienennetään tekstuureja ja polygonien määrää, kuin poistetaan tai muutetaan toimivaa animaatiota, joka sopii pelin kontekstiin ja tyyliin.

Animaatiolla on siis tärkeä rooli, kun puhutaan siitä, mitä sijoitetaan lopputuotteeseen. On kuitenkin syytä muistaa myös tämän vastapaino. Fakta on, että animaation laatu riippuu yksin kertaistesti mallin laadusta, jolle se tehdään. Tämäkin on taas tasapainoilua laadun ja rajoitusten välillä.

Projektin aikana animaatio tapahtui suurimmaksi osaksi koodilla. Tästä syystä malleihin sovellettujen animaatioiden osuus jäi suhteellisen pieneksi. Animaatio kuitenkin ansaitsee oman käsittelynsä tässä opinnäytetyössä vaikkakin tämä on suhteellisen pintapuolinen käsittely. Animaatiota ei voi eikä saa vähätellä, sillä se on onnistuneen pelin kannalta välttämätöntä.

## **5 MALLINTAMINEN PELEISSÄ**

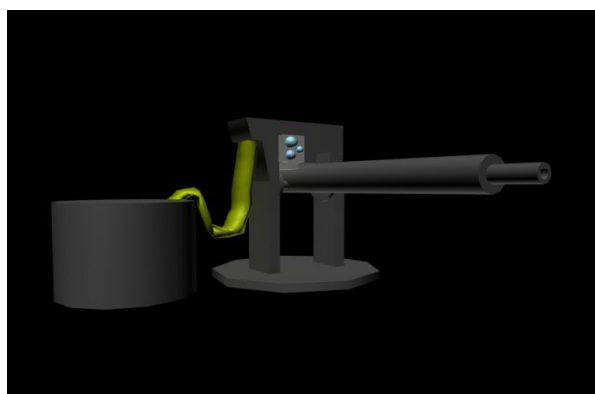
Mallintaminen peliprojektissa on oma lukunsa. On harvinainen poikkeus, jos on osana peliprojektia ja saa tai edes pystyy tuottamaan sisältöä ilman vuorovaikutusta. Erittäin usein pelin idea on ryhmässä suunniteltu idea tai joku muu, kuin mallinnuksesta vastaava, on vastuussa pelin alkuperäisideasta. Lähtökohtana voidaan myös olettaa, että

toteutettava peli on ryhmäprojekti. Pelin kehitys ei koskaan tapahdu tyhjiössä ja tämä on huomioitava omassa roolissaan projektissa, oli sitten rooli mikä tahansa.

On siis otettava huomioon toteutuksen aikana, että mallintajalla on tiettyjä velvollisuuksia alkuperäisideaa ja sen tuottajaa kohtaan. Kaikki omat ideat eivät välttämättä palvele kokonaisuutta tai pysy uskollisena alkuperäisidealle. Esimerkiksi objektien mallintamisessa on hyvä ylläpitää säännöllistä vuorovaikutusta muiden projektinhaarojen kanssa mallin kehityksen aikana.

Pelin kehitysprojekti on oikeastaan suhteellisen strukturoitu ympäristö. Aikataulu voi olla joustava, mutta tietyt vaiheet on kuitenkin hoidettava ennen kuin siirrytään tehtävälisessä eteenpäin. Sama pätee mallintamiseen kuin koodaamiseenkin. Täytyy ensin olla konsepti, josta lähdetään liikkeelle.

Ensin tulee ideointi. Projektin johto, tai haarojen johtajat, tai muut asiaankuuluvat tahot päättävät mitä tarvitaan. Objekti itsessään on tapauskohtainen. Se voi olla tykki, pelihahmo, ajoneuvo tms. Peruskonseptin jälkeen luonnostellaan kuvailun perusteella alustava objekti. Tässä vaiheessa vuorovaikutus on tärkeää sillä niin sanottu tyhjiössä mallinnettu objekti ei välttämättä vastaa sovittua konseptia. Palautteen pohjalta voidaan valita tarkoituksenmukainen luonnos, jonka perusteella voidaan mallintaa lopullista mallia. *Kuvassa 14 näkyvässä tykissä käytettiin tätä periaatetta.*



**KUVA 14: Alukseen asennettavan tykin mesh**

Muutoksia voi tapahtua projektin aikana, joten niihin on syytä varautua. Tiettyjä objekteja saatetaan leikata lopullisesta tuotteesta, konsepteja saatetaan muuttaa, projektihenkilöstössä saattaa tapahtua muutoksia tai budjetti voi yhtäkkiä muuttua parempaan tai huonompaan. Tämä on yksi syy miksi objektin mallinnuksessa on hyvä olla vuorovaikutuksessa muun projektihenkilöstön kanssa. Voidaan pitää faktana, että mallintajan rakentaessa objektiaan, hän alkaa menettää perspektiiviään jo noin kahden

tunnin jälkeen. Hän väsyy, ajatukset harhailevat tai hän voi myös olla vähemmän kriittisempi omaa työtään kohtaan. Tätä voi tapahtua esimerkiksi silloin kun on asetettu määräaika tavoite ja projektin kaikki haarat voivat tiedostamatta työskennellä liian kovaa pyrkiessään tuottamaan sisältöä ennen aikarajaa. Tällöin väistämättä menettää osan omasta näkemyksestään. Tapahtuu virheitä ja käytetään oikoteitä suhteessa enemmän kuin muulloin.

Tällöin tuore perspektiivi on avuksi. Palautteen avulla voidaan muokata mallia sopivampaan muotoon. Aiemmin vaietetut ideat voivat kantaa hedelmää kyseessä olevassa kontekstissa ja ne voivat päätyä peliin. Vuorovaikutuksen avulla voidaan myös määrittää toimivatko jotkin ideat pelin kannalta jo kehitys vaiheessa, vai leikataanko kyseiset ideat pelistä pois. Tämä viimeiseksi mainittu pätee sekä ohjelmointiin, että mallintamiseen.

## **5.1 Pelin asettamat rajoitukset**

Missä tahansa pelissä on nykyajan teknologia tason asettamat tekniset rajoitukset. Tämä on totta myös tässä projektissa. Yleensä tekniset rajoitukset riippuvat siitä, mille alustalle lähdetään peliä tekemään, mutta joskus tieteenkin peliin saatetaan sisällyttää ominaisuuksia, jotka eivät ole yleisiä standardeja. Tällä tarkoitetaan sitä, että sisältääkö peli ominaisuuden, jota enemmistöllä kohdeyleisöstä ei ole. Tämä on kuitenkin suhteellisen harvinaista ja tätä ei tapahtunut tässä projektissa, joten tätä ei liiemmin käsitellä tässä opinnäytetyössä.

Mallintamisen kannalta on yleensä kysymys pelille valitulla alustalla saatavilla olevasta muistista. Muistia käytetään mallien osalta renderointiin ja animaatioon. Pelissä olevassa mallissa on useita renderöitäviä osia. Mallin mesh, sen pintatekstuurit, jotka puolestaan voivat koostua useimmista päällekkäisistä kartoista, animaatiot ja valaistus. Kukin osan kuluttaa saatavilla olevaa muistia jonkin verran. Mitä suurempi ja yksityiskohtaisempi malli sen enemmän muistia kuluu.

Siispä hyvän mallintajan on harjoitettava malliensa editointia, mitä tulee peliprojekteihin. Tätä on harjoitettu alusta pitäen mallintaessa tämän projektin aikana. Kun tätä toimintaa harjoitetaan oikein, syntyneet mallit täten eivät vaadi ylettömiä määriä muistia



lopun pelin kustannuksella. Tällöin ne voidaan sisällyttää peliin huolehtimatta suoritusasteen laskemisesta.

On kuitenkin huomioitava, että meidän projektimme on suunniteltu alusta lähtien käytettäväksi PC:llä. Alustana PC, etenkin nykyään, on suhteellisen kykenevä suorittamaan pelejä perustasollakin. Puhtaan suoritusasteen kannalta PC voittaa nykyiset konsolit ja mobiililaitteet. Tästä syystä PC on ehkä helpoin alusta mallien optimoinnille. Optimointia on kuitenkin harjoitettava, koska on pidettävä mielessä kohdeyleisönsä enemmistö. Kaikki PC:t eivät ole lainkaan optimoitu pelaamiseen ja tavoitteena on varmistaa, että pystyt levittämään peliäsi mahdollisimman laajalle.

Konsolit ja mobiililaitteet ovat puolestaan rajoitettuja ympäristöjä, joilla on asetetut laitteistot, jotka eivät muutu kyseisen laitteen eliniän aikana. Ohjelmistot voivat tietenkin muuttua, mutta tämä ei vaikuta muistin saatavuuteen. Muistin määrä pysyy laiteessa itsessään samana, lukuun ottamatta uusia teknisiä julkaisuja tietenkin. Tällä tarkoitetaan muistia lisääviä moduuleita tai suorituskykyä lisääviä lisälaitteita, mutta PC:n lisäksi tällaiset laitteet ovat harvinaisia muilla alustoilla.

Jos olisimme rakentaneet pelin esimerkiksi XBOX360:lle, olisi ollut suurempi tarve pitää muistin kulutusta silmällä kehityksen aikana, koska 360:lla on huomattavasti rajoituneempi laitteisto kuin pc:llä. On huomattavasti todennäköisempää kuluttaa muisti loppuun rajoitetuilla alustoilla, mikä on itsestäänselvyys. Editoinnilla on tässä suhteessa tärkeämpi merkitys.

Muistin kanssa tasapainoillessa on otettava huomioon myös se mitä muuta peliin tulee. Peliin tulevat toiminnot voivat myös määrittää sen miten paljon muistia on saatavilla. Monimutkaisemmat toiminnot selvästi kuluttavat enemmän muistia ja niiden lisääntyvä määrä lisää muistin kulutusta.

Esimerkiksi suuressa kentässä on suuri määrä NPC-hahmoja. Kaikki hahmot suorittavat niille määrättyä toimintaa samanaikaisesti. Tällöin kullakin hahmolla on oma ohjelmansa, jonka tehtävä täytyy suorittaa jossakin välissä. Tällöin kukin ohjelma käyttää muistia tietojen varastoimiseen ja niihin viittaamiseen. Lasketaan tähän vielä kaikki se mitä muuta ympäristössä tapahtuu.

Lisätään tähän vielä pohjamekaniikka, joka tarvitaan ympäristön kasassa pitämiseen ja muodostamiseen. Lisätään tähän vielä mallit, valaistustiedot, animaatiot ja tekstuurit. Yhtäkkiä muistia onkin kulunut jo jonkin verran. Tietenkin muistin kulutuksen määrä

on suhteessa projektin skaalaan, mutta tietoisuutta saatavilla olevista resursseista ei voi vähätellä.

## 5.2 Mallien rajoittaminen

Mallien rajoittamisella tarkoitetaan tässä opinnäytetyössä malleja joita pyritään editoimaan siten, että mallit eivät vie muistia pelin toiminnan kannalta kriittisille sovelluksilta. Tämän seikan johdosta voidaan aina havaita tiettyjä laatueroja malleissa, joita käytetään esimerkiksi CGI-elokuvissa ja peleissä käytettävien välillä. CGI:ssä ei ole muita elementtejä, jotka kilpailevat samoista resursseista. Peleissä puolestaan muistin saatavuus on otettava huomioon.

Tämä laatueroon takia muiden formaattien mallit näyttävät aina sanalla sanoen paremmilta. Tämä on huomattavissa animaatioissa, pintatekstuureissa, valaistuksessa tai meshien yksityiskohtaisuudessa. Kaikkien näiden elementtien renderöinti vie pelissä muistia ja riippuen muistin määrästä, se voi olla pois sovellukselta, joka vaikuttaa pelin vakauteen. Johtopäätöksenä voidaan olettaa, että huonosti editoidut mallit voivat kaataa pelin, sitä suorittaessa.

Mallintajan vastuulla on oltava varsinaisen mallintamisen lisäksi se miten hän varmistaa mallien soveltuvuuden peliin mekaanisessa mielessä. Tämä edellyttää mallien tarkastelua kriittisessä ja käytännöllisessä mielessä. Tällöin ei voi keskittyä pelkästään mallien esteettisyyteen ja taiteelliseen näkemykseen. On muistettava, että mallit ovat osa käytännöllistä kokonaisuutta, johon kuuluu monia osa-alueita, joista vastaavat monet eri henkilöt. Tietty nöyryys on hyväksi tässä mielessä.

Käytännöllinen kannattaa olla jo projektin alusta asti sillä se säästää paljon aikaa editoinnin kannalta. Pelien parissa työskentelevä mallintaja voi ja tulee saamaan käyttöönsä uusia ominaisuuksia, joilla rakentaa malleja ohjelmistojen ja teknologian kehityessä. Tuoreet mallintajat puolestaan löytävät oppimisen yhteydessä ominaisuuksia, joista he eivät aikaisemmin tieneet. Molemmat ryhmät voivat tuntea vetoa käyttää näitä malleissaan heti tilaisuuden tullen. Tämä ei sinällään ole uutta, eikä lainkaan paha asia. Uusia asioita kannattaa kokeilla kun ne oikeasti palvelevat kokonaisuutta.

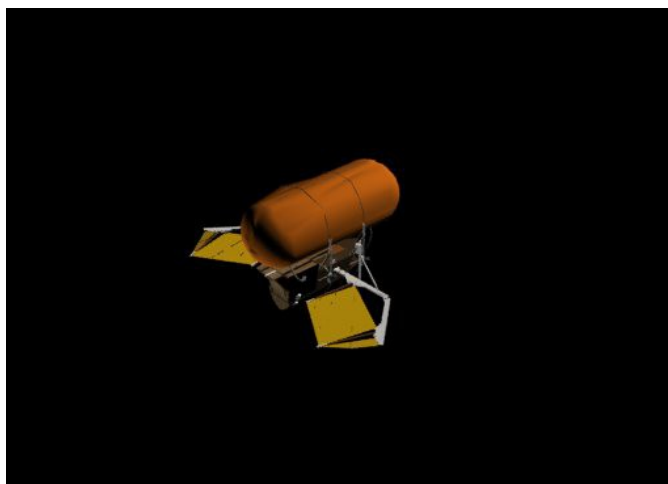
Tärkeä asia, joka on syytä muistaa, on se, että uutuudenviehätys on pidettävä kurissa ja toissa sijaisena impulssina. Vastakkaisessa tapauksessa, jos antautuu uutuuden viehätysten valtaan, voivat laitteisto- ja muistirajoitukset tulla vastaan ja sitten on joko keksittävä mistä muistia saadaan lisää, miten kierretään rajoitukset tai editoitava malliejaan, joka on loppuvaiheessa työläämpi ja aikaa vievä prosessi.

Yksinkertaistetusti sanoen voidaan todeta että käytännöllisempi näkökulma on pelien kanssa tärkeämpää kuin muissa medioissa. Pelien rakentaminen on jatkuvaa tasapainoilua laadun ja toimivuuden välillä. Mallintajan on osattava siis sijoittaa omat odotuksensa ryhmän ja projektin vaatimuksiin myös teknisessä kuin taiteellisessakin mielessä. Kun tämän pitää mielessä, voidaan tätä soveltaa projektin aikana useilla tavoilla jolloin säästää itseltään paljon vaivaa lopussa.

### **5.3 Mallien rajoittamisen keinot**

Projektin konsepti määrittää tietenkin sen mitä mallinnetaan, mutta projekti ei toteudu, jos mallien paino pelkästään aiheuttaa koko sovelluksen kaatumisen. Yleinen järjestyksen käyttö auttaa alusta asti pitkälle projektin aikana. Se onko tarpeen sisällyttää joitain osia malleihin muistin kustannuksella, tarvitaanko valaistusta tuohon kohtaan tai kuinka tarkkoja tekstuurien on oltava tässä kontekstissa. Tällaisten seikkojen pohdinta auttaa muistin säästämässä jolloin jää paremmin tilaa sovittaa mekaaninen ja taiteellinen puoli paremmin yhteen kokonaisuudeksi.

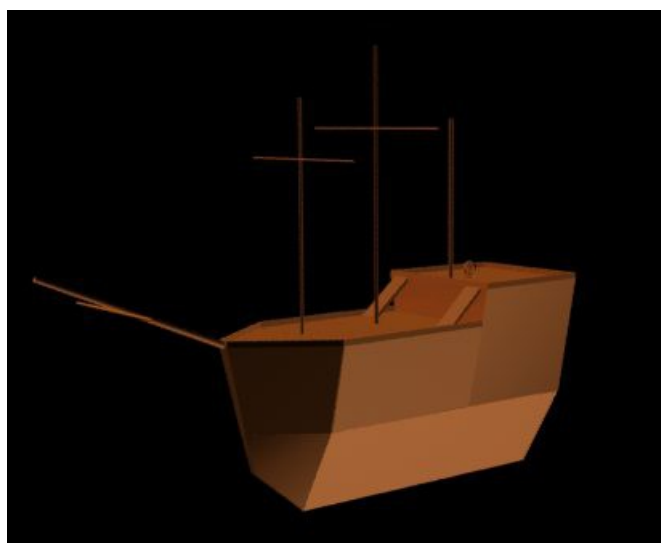
Tätä periaatetta voitiin soveltaa ja sovellettiinkin tämän opinnäytetyön aikana. Prosessi alkoi konseptin hahmottamisesta. Tarkoituksena oli rakentaa laivoilla käytävä reaaliaikainen strategia tyylinen peli johon on sekoitettu roolipelimäisiä mekanismeja. Tietona oli alussa, että peli ikkuna olisi reaaliaikaisten strategiapelien tyylinen. Tällöin oli selvää, että kaikkien yksityiskohtien ei tarvitsisi olla huipputarkkoja, koska pelaaja ei kuitenkaan näkisi kaikkea tuosta näkökulmasta. Tätä periaatetta voitiin käyttää hyväksi myös mallien mallintamisessa, koska tällöin voidaan yhä tehdä näyttäviä malleja pienemällä polygonien määrällä.



**KUVA 15: Esimerkki pelin perspektiivistä**

Mallien rajoittamisessa ei ole pelkästään kyse mallien koosta saatavilla olevan muistin suhteen, vaan on huomioitava myös projektin muut muuttujat. Mikä on projektin deadline? Millaiset tekstuurit on suunniteltu malleille? Miten monimutkaisia malleista tulee? Muitakin seikkoja ilmenee tilannekohtaisesti, mutta kaikki tällaiset seikat vaikuttavat malleihin ja myös mallintamisen tehokkuuteen.

Ensimmäisen laivan runkoa oli jo alettu mallintamaan, kun projektin konsepti päätettiin vaihtaa 1700- luvun aikaisista laivoista steampunk-tyyliseen maailmaan ja ilmalai-voihin. Koska runko oli vain alustava hahmotelma ja sopi uuteen tyyliin, siitä tarvitsi muuttaa suhteessa vain vähän ominaisuuksia. Muussa tapauksessa olisi jouduttu mallintamaan uusi mesh alusta asti.



**KUVA 16: Varhainen laivan mesh**

On huomioitava, että ominaisuuksia muutettiin ja vain konseptiin sopimattomat osat poistettiin. Esimerkiksi mastot jotka olivat muutoin tiellä. Poistettujen mastojen tilalle

asetettiin myöhemmin zeppeliinikaasupallo. Perusrungon lievää pyöristystä lukuun ottamatta, runko säilyi pitkälti kuitenkin samassa muodossa, joka auttoi rajoittamaan meshissä käytettyjen polygonien määrää. Tässä säästyivät myös projektin kannalta aikaa, koska ei tarvinnut mallintaa uutta perushahmotelmaa alusta. Tässä tapauksessa käytössä oli runko, jonka ympärille lähdettiin kasamaan varsinaista mallia.

Yksi seikka, joka vaikuttaa mallien rajoittamiseen, on pelaajan perspektiivi. Tällä tietenkin tarkoitetaan sitä, miten pelaaja näkee peliympäristön käytännössä. Jos pelaajan näkökulma toteutetaan ensimmäisessä persoonassa, on tällöin suositeltavaa, että kiinnitetään tarkempaa huomiota läheltä nähtyihin yksityiskohtiin.

Tämä ei tietenkään ollut ongelma pelinkehityksessä aikaisemmin, koska teknologia rajoitukset merkitsivät sitä, että mallien yksityiskohdissa pystyttiin vain menemään niin pitkälle kuin pystyttiin renderöimään sen ajan teknologialla. Nykyajan teknologia pystyy huomattavasti enempään, jolloin on edellä mainitussa esimerkkitapauksessa määriteltävä yksityiskohtien taso niin, että mallit eivät vie kaikkea tilaa. Nykyään pystytään renderöimään huomattavan fotorealistisia malleja, mutta pelin moottori ei välttämättä suoriudu kuormasta.

Tässä opinnäytetyössä puolestaan käytetään kolmannen persoonan kuvakulmaa, joka on kiinteässä kulmassa yläviistoon katsoen pelinympäristössä. Tätä kuvakulmaa yleisimmin käytetään strategiapeleissä. Tässä perspektiivissä ei ole samanlaista ongelmaa kuin ensimmäisessä persoonassa. Mallien yksityiskohtaisuus ei välttämättä kiinnitä pelaajan huomiota, koska mallit ovat suhteessa kauempana pelaajasta. Samalla voidaan kiinnittää enemmän huomiota millaisia meshejä ja tekstuureja halutaan malleiksi ja malleihin.

Tällöin ei kuitenkaan ole vapauduttu siitä, että mallien yksityiskohtaisuutta pitää säännöstellä. Tekstuureilla on edelleen kokonsa ja mesheissä polygonien määränsä. Kun on vähemmän rajoituksia, niin on todennäköistä, että kokee alustavasti innostusta mallintaa mahdollisimman tarkasti. Tämä on virhe, sillä piakkoin ollaan tilanteessa jossa ei ole muuta vaihtoehtoa, kuin käydä kaikki mallit läpi ja editoida ne niin, että peli suoriutuu niistä.

Mallintaessa tähän opinnäytetyöhön, jokaista mallia lähdettiin mallintamaan periaatteella, että miten vähällä onnistutaan luomaan haluttu efekti. Tällöin oli tietenkin konsultoitava muun projektiryhmän kanssa säännöllisesti siitä mitä malleihin haluttiin li-

sää, mitä vähemmän ja mihin suuntaan oltiin mallia lähdössä kehittämään. Tällä prosessilla pyrittiin nimenomaan rajoittamaan mallien meshien polygonien määrää.

Mallejamme myös testattiin säännöllisesti pelimoottorissa varmistaaksemme niiden ulkoasut ja toimivuus. Tällöin myös varmistui, joka kerralla, miten peli itse suoriutui malleista. Toistaiseksi ei ole ilmennyt suoritusason laskemista, joten peli suoriutuu malleista ja on onnistuttu luomaan malleja, jotka vastaavat alkuperäistä konseptia.

Mallien polygonit ovat tietenkin vain yksi mahdollinen keino säästää muistin kulutuksessa. Muita osa-alueita voidaan myös säätää siten, että optimoidaan muistin kulutus. Esimerkiksi säädetään tekstuurien kokoa siten, että pinnassa näkyy mahdollisimman pieni määrä pikseleitä joita tarvitsee renderöidä.

Mallien ja tekstuurien koon rajoittaminen eivät ole tietenkään ainoita keinoja muistin säästämiseksi. Jos näin olisi tästä seuraisi se, että pelin koon kasvaessa laatu huononiisi. Tämä ei kuitenkaan pidä paikkaansa. On useita pelejä joissa on suhteessa suuri määrä sisältöä jotka onnistuvat säilyttämään hyvän laatutason. Ennen kaikkea laatu tuntuu paranevan vuodesta toiseen.

Projektiin osallistujien yhteistyö on tässä kriittistä. Tässä sovelletaan nimittäin monimutkaisempia metodeja, joita pelaaja ei tule koskaan näkemään. Englannin kielessä tälle on termi ”Smoke and mirrors”. Tällä viitataan taikurin esitykseen ja siihen mitä kulissee tapahtuu esityksen aikana, mitä yleisö ei koskaan näe.

Esimerkiksi koostetaan maailma erillisistä alueista. Sitten renderöidään sisältö alueelle jolla pelaaja on sillä hetkellä. Tällöin ei tarvitse renderöidä turhaan alueita joita pelaaja ei kuitenkaan näe ja ne voidaan jättää taustalle odottamaan vuoroaan ja säästää muistia näin.

Maailma voidaan myös rakentaa siten, että näkyvä maailma on ”olemassa” tietyille etäisyydelle pelaajasta kaikkiin suuntiin. Tällöin objektit, jotka ylittävät näköetäisyyden, suorittavat edelleen ohjelmointiaan, mutta niiden renderöinti odottaa siihen asti kunnes pelaaja on määrätyn etäisyyden päässä.

Vaikuttavissa elokuvakohtauksissa, tai kuten yleensä käytetään termiä ”cutscene”, voidaan puolestaan käyttää rajattua renderöintiä. Kameralla rajataan kohtaus siten, että kamera ei ylitä tiettyjä rajoja ja renderöidään kohtaus noiden rajojen sisältä. Tällöin saadaan renderöityä näyttävä kohtaus pelaajan tietämättä, että kohtaus on erillinen tila ja kaikki kameran ulkopuolella on tyhjää tilaa.

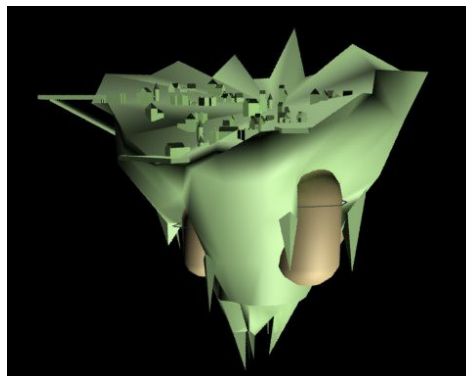
Enimmäkseen se riippuu kehitysalustasta millaisia keinoja on saatavilla mallien rajoittamiseen. Taitava projektihenkilöstö voi keksiä omia keinojaan rajoittaa malleja, mutta ennen kaikkea on oltava oma-aloitteinen. On olemassa huomattava määrä yleisesti käytettyjä keinoja joita voi käyttää keskenään saavuttaakseen parhaan lopputuloksen. Ne eivät vain ole välttämättä itsestään selviä vaihtoehtoja ja tutkimus on edellytys niiden löytämiselle.

Mallit eivät ole puolestaan ainoita joissa voidaan harjoittaa säästämistä. Sama pätee esimerkiksi ääniin ja valaistukseen. Pelin kehittämisessä on aina useampia ominaisuuksia, joita voidaan karsia muistin säästämiseksi. Se mitä ne alueet ovat, riippuu pitkälti projektista ja sen tarpeista.

#### 5.4 Mallien sovittaminen peleihin

Mallien valmistuttua, ne on vielä saatava liitettyä peliin. Tämä on fakta, jota ei voida kiertää mitenkään. Tässä vaiheessa koettujen ongelmien määrä vaihtelee sen perusteella, kuinka kokenutta projektihenkilöstö on käytettyjen työkalujen ja ympäristöjen suhteen tai mitenkä eri ohjelmat sopivat toistensa kanssa yhteen. Lyhyesti sanottuna tässä vaiheessa ongelmien paljouteen vaikuttavat siis inhimilliset ja tekniset muuttujat.

Esimerkiksi pelissämme käytettävä leijuva saari koki ongelmia, jotka voidaan laskea inhimillisen erehdyksen syyksi. Saaren mesh ladattiin asettina Unity-moottoriin. Saarta liikuteltaessa huomattiin kuitenkin, että saaren kyljessä oli huomattavia reikiä. Pienen testaamisen jälkeen todettiin myöhemmin, että reiät johtuivat päällekkäisistä polygoneista meshissä itsessään. Tämän jälkeen reiät korjattiin asettelemalla polygonit uudestaan. Saaren kylki näkyy nykyään paremmin.



**KUVA 17: Leijuva saari**

Inhimillinen erehdys ja huolimattomuus olivat syynä tähän virheeseen. Siispä on suotavaa olla tietoinen omista rajoituksistaan mallintaessa. Monimutkaiset mallit eivät ole pieniä projekteja ja tietyt virheet ovat erittäin todennäköisiä mitä enemmän työtä asetetaan yksilölle. Negatiivisesti vaikuttavien inhimillisten muuttujien minimoimiseksi, laadunvalvonta ja testaus ovat välttämättömiä. Vaikka työskentelisi yksin, joka oli tämän projektin aikana todellisuus, on pidettävä avoin keskustelu yllä muiden haarojen kanssa. Tällöin varmistetaan parempi tietoisuus projektissa esiintyvistä ongelmista ja siitä seikasta liittyvätkö ne omaan haaraan.

Projektin mallintajana, kirjoittaja oli ainoa jolla oli käytössään 3DS-Max-ohjelma. Muut haarat käyttivät yksinomaan Unity-pelimoottoria. Tällöin Unityn ja Maxin välinen yhteen sopivuus oli alustava ongelma. Unity ei nimittäin pysty lukemaan Maxin omaa tiedostoformaattia suoraan, ellei Unityä käytävällä päätteellä ole asennettu Maxia.

Tämä ratkesi helposti käyttämällä yksinkertaisesti toista yhteensopivaa formaattia. Tässä tapauksessa käytimme .fbx formaattia. Tämä kuitenkin luo omat vaikeutensa mallien sovittamiselle, sillä viemällä mallit .fbx formaatissa on otettava ensiksi huomioon niiden virheetön kääntäminen haluttuun formaattiin. Virheitä voi esiintyä tässä vaiheessa useitakin ja näiden virheiden takia Unity ei välttämättä näytä malleja halutulla tavalla. On syytä huomauttaa, että nämäkin virheet voivat suurimmaksi osaksi johtua käyttäjän virheestä joko mallintamisessa tai kääntämisen aikana.

Projektin aikana koetut vikatilat olivat selvästi enemmistöltään käyttövirheitä. Ohjelmissa itsessään on varmasti moniakin vikoja, jotka voivat vaikuttaa projektin etenemiseen. Max kaatui kirjoittajalta sammuttaessa moneenkin otteeseen, mutta koska se ei vaikuttanut projektiin millään lailla tätä ei voida laskea.

Johtopäätöksenä voidaan siis vetää se seikka, että useimmat vikatiloista mallien suhteen projektin aikana johtuivat käyttäjien kokemattomuudesta ja virheellisestä suorituksesta. Ongelmat ratkesivat lopulta syvemmän tutkimisen ja testaamisen johdosta. Jos mallintajalla on siis ongelmia mallien sijoittamisessa peliin, loogisinta on käydä omat toimenpiteensä läpi tarkemmin. Tällöin varmistetaan inhimillinen tekijä tai poistetaan se ongelman syynä. Koska se on enemmistössä ongelmien syynä, on se hyvä ensimmäisenä asiana tarkistaa.



## 6 POHDINTA

Projektin aikana ajateltiin säännöllisesti eroja varhaisen pelinkehityksen ja nykytilanteen välillä. Pelien kehitys on selvästi muuttunut virtaviivaisemmaksi. Mieltien millainen työ oli aikoinaan saada kaksi palkkia ja pikselistä koostuva pallo liikkumaan ruudulla tarkoituksen mukaisesti ja kutsua sitä nimellä PONG.

Ennen kaikkea kehitys ympäristöt ovat muuntuneet siten, että ne sallivat yhä useampien ihmisten työstää ideoitaan konkreettiseen muotoon. Tämän projektin alussa pohdittiin koodin kirjoittamista Microsoftin VisualStudio-ympäristössä. Piakkoin kuitenkin tulimme tietoisiksi Unity-moottorista ja vaihdoimme siihen. Pienen opetteluksen jälkeen kaikki projektissa tunsivat käytetyn ympäristön piirteet ja alkoivat sopeuttaa lähestymistään kehittämisesä tätä ympäristöä silmällä pitäen.

Ympäristön kehitys ei kuitenkaan ole takuu perspektiivin kehityksestä. Pelin kehityksen kannalta tämä on tärkeämpää, koska ilman näkemystä ja perspektiiviä tuotettu sisältö on merkityksetöntä. Tämän takia on kriittisen tärkeää, että jokaista seuraavaa pelikehittelijöiden sukupolvea kohden varmistetaan, että heillä on jokin idea eikä erillinen pala sisältöä tai jokin hieno mekanismi.

1990-luvun puolessa välissä kuluttajateknologia saavutti pisteen, jossa oli mahdollista käsitellä monimutkaisimpia kuvia, renderöidä 3D-malleja, muokata videota ja lisätä efektejä. Viimeisen kahdeksantoista vuoden aikana sukupolvi, joka on lapsesta asti elänyt videopelien ja muun interaktiivisen materiaalin keskellä, on aikuistumassa. Tämä puolestaan merkitsee kokonaista sukupolvea, jolla on pitkään kehittynyt näkemys pelien nykytilanteesta ja erilaisia visioita niiden tulevasta suunnasta. (Jana & Tribe 2006, 10.)

Tämä sukupolvi on merkittävä siinä mielessä, että tämä on sukupolvi, joka on nähnyt pelien kehityskaaren kokonaan niiden varhaisesta yleistymisestä aina nykymuotoon asti. Tämä antaa perspektiivin siitä mitä ne olivat, miten ne muuttuivat ja miten ne voivat muuttua. Seuraavat sukupolvet saavuttavat tämän perspektiivin ajallaan, mutta toistaiseksi he näkevät vain mitä pelit ovat nyt.

Pelien kehitys on ollut tasaisesti kiihtyvää niiden kotikäytössä yleistymisen jälkeen. Tämä voidaan havaita esimerkiksi miten Nintendo on vienyt Mariota eteenpäin. 8-bit-tisestä 2D-hahmosta 3D-hahmoon alle kahdessa vuosikymmenessä. Pelaajat, jotka

lapsuusvuosinaan pelasivat Mariot olohuoneissaan, löytävät vanhat Mario-pelit nykyään museoista ja retrokeräilypaikoista.

Pelien teknologian kehitys on jatkuvaa vieläkin, mutta on jo saavutettu piste jossa se ei riitä hyvän pelin tekemiseen. Nykyään pelit vaativat syvyyttä jollakin tasolla, oli kyseessä sitten tarinankerronta tai moninpelissä käytettävät mekaniikat. Merkit viittaavat, kirjoittajan mielestä kehityksen jatkumiseen tämän osalta.

Osa tästä syvyyden tarpeesta liittyy siihen, että ensimmäisiä pelejä pelanneet eivät koskaan lopettaneet pelaamista. Pelaajien aikuistuesssa tarve aikuistuneemmalle sisällölle kasvoi myös. Tämän johdosta pelit kehittyivät nopeasti ulos roolistaan leikkikaluina ja perustivat itselleen oman mediahaaransa.

Tämä uusi mediahaara puolestaan jatkaa kehitystään edelleen. Se ei ole vielä täysin muotoutunut ja usein kuullaan puhuttavan viime aikoina, että pelit mediana kärsivät pienoisesta identiteetti kriisistä. Tämä pitää kirjoittajan mielestä paikkansa tällä hetkellä, mutta tilanne ei ole kuitenkaan pysyvä.

Mallinnus peleissä on osa tätä kehitystä. Peleissä mallinnukseen käytettävien työkalujen käyttö on kokenut ja on kokemassa muutosta. Yleiskäyttöön suunniteltujen työkalujen käyttö tuskin koskaan jää pois, sillä ne ovat helpommin saatavilla ja käytettävissä niin harrastelija kuin ammattilaistasollakin. Ne mahdollistavat pienempien pelien kehittämisen ja avustavat pelien ympärille kehittyneiden yhteisöjen modien tekemisessä ja julkaisussa.

Toisaalla kehitellään jatkuvasti paremmin moottoreiden kanssa yhteensopivia sovelluksia, jotka mahdollistavat enemmän kunnianhimoisempien ominaisuuksien lisäämisen peleihin. Joka tapauksessa, molempien tyyppisten sovellusten kehitys jatkuu kirjoittajan mielestä toistaiseksi tasaisena.

Tulevaisuutta arvioitaessa on luultavaa, että yksi alue, joka kokee kehitystä, on käyttäjävällisyys, mitä tulee mallinnusohjelmiin. Projektin aikana käytetyssä 3DSMaxissa on esimerkiksi paljon ominaisuuksia joiden käyttö ei ole helppoa. Tuotteen valmistajat tulevat ehkä muokkaamaan tuotteitaan suuremman yleisön toiveesta, mutta tämä on vain kirjoittajan arvio tulevaisuudesta.

Pelin kehitys ei ole välttämättä helppo projekti, mutta kynnyks siihen ryhtymiseen on helpompi ylittää kuin ennen. Vähän yli viisi vuotta sitten olisi ollut hankala olla osana projektia ilman kattavaa ennakkotietoa. Tässä projektissa oli suhteellisen helppoa

omaksua työkalut ja ympäristöt suhteellisen lyhyessä ajassa. Tämä kertoo kuinka virtaviivaista pelinkehitys on nykytilassaan, mutta on edelleen muistettava, ettei se tee siitä helppoa projektia.

Tätä kirjoittaessa pelimme on saavuttamassa demo vaihettaan ja silti siinä ei ole läheskään kaikkea sisältöä mitä se tarvitsee. Tästä esimerkkinä tekstuurien liittäminen. Puhumattakaan siitä, mitä me haluaisimme vielä lisätä siihen. Kuitenkin peli on vakaa mallien kanssa, koodi toimii muutamaa häiriötä lukuun ottamatta (Tätä kirjoittaessa virheenkorjaus on vielä menossa) ja eri sisällöt muodostavat kokonaisuuden.

Ongelmien ratkettua on aina vielä jäljellä raaka työ. Toistetaan toimintoja, kunnes saadaan haluttu sisältö tai ominaisuus valmiiksi. Pelin kehityksessä on kyse ongelmien ratkaisemisesta, mutta se koostuu myös työstä, joka voi edetä ongelmien ratkettua. Jos päättää ruveta pelinkehitykseen, on siis syytä varautua ongelmien ratkaisuun, jotta voi tehdä työtä hyvän idean eteen.

Tulevaisuudessa todennäköisesti keskitytään juuri vanhojen ongelmien ratkaisemiseen, jotta saadaan enemmän työtä aikaan. Tämä tietenkin tarkoittaa sitä, että ideoidaan uusia ominaisuuksia, joita ei ole ennen kokeiltu. Tämä puolestaan aloittaa syklin alusta, mutta tämä on kestävän kehityksen merkki. Eikä se nykypäivänä osoita hidastumisen merkkejä.

## 7 LÄHTEET

Colson, Richard 2007. The Fundamentals of Digital Art. Switzerland: AVA Publishing.

Hietikko, Esa 2007. Autodesk Inventor. Helsinki: Readme.fi.

Jana, Reena & Tribe, Mark 2006. New Media Art. Germany: Taschen.

Nvidia 2010. DirectX 11 Tessellation. Artikkelin yrityksen www-sivuilla. <http://www.nvidia.com/object/tessellation.html> Päivitetty 30.10.2010. Luettu 5.11.2013.

Scott, Bill 2013. Ilmaisten käyttökuvien sivusto. <http://texturemate.com/content/free-texture-metal-2010091212> Päivitetty 30.10.2013. Luettu 1.11.2013.

Texturing In 3D Studio Max (Using UVW and Unwrap UVW Maps). 2011. www-dokumentti. [http://www.republicofcode.com/tutorials/3ds/texture\\_stealth/](http://www.republicofcode.com/tutorials/3ds/texture_stealth/) Päivitetty 1.6.2011. Luettu 29.10.2013.