

3D-peliassetin luonnin työvaiheet

Case: Auto peliassetin tekeminen

LAB-ammattikorkeakoulu

Insinööri (AMK), Tieto- ja viestintäteknikka

2023

Jutta Martikainen

Tiivistelmä

Tekijä(t) Jutta Martikainen	Julkaisun laji Opinnäytetyö, AMK Sivumäärä 38	Valmistumisaika 2023
Työn nimi 3D-peliassetin luonnin työvaiheet Case: Auto peliassetin tekeminen		
Tutkinto ja koulutusala Insinööri (AMK), Tieto- ja viestintätekniikka		
Toimeksiantajaorganisaatio (jos opinnäytetyöllä on toimeksiantaja) Henri Koskinen, Toimitusjohtaja ja mediatuottaja, Pixtell Oy		
Tiivistelmä <p>Työn tavoitteena oli tutkia millaisia vaiheita 3D-peliassetin luominen sisältää, sekä miten malli saadaan optimoitua videopeliä varten. Tutkimukseen perehdytään 3D-sovel-luksien näkökulmasta.</p> <p>Toimeksiantajana toimi Pixtell Oy, joka kehittää peliä nimeltä LAST DROP – Legend of Seppo. Tehtävänä oli suunnitella, mallintaa ja teksturoida 3D-asetti, samalla huomioiden videopelien asettamat rajoitteet. Asettia on tarkoitus käyttää videopelissä.</p> <p>Opinnäytetyön on tarkoitus tuoda esille oleellisimmat asiat, mitä peliassettien luomisen työvaiheet pitävät sisällään ja mitä asioita pitää ottaa huomioon, jotta asetti on mahdollisimman toimiva videopelissä. Peliassetin luonnin työvaiheet muodostuvat suunnittelusta, mallintamisesta, teksturoimisesta ja jälkikäsittelystä.</p> <p>Johtopäätöksenä peliassetin luomisprosessissa huomattiin, että eri työvaiheilla on vaikutusta toisiinsa. Hyvällä suunnittelulla helpotetaan asetin luomista.</p>		
Asiasanat peliassetti, 3D-mallintaminen, 3D-malli, teksturointi, tekstuurikartta		

Abstract

Author(s) Jutta Martikainen	Type of Publication Thesis, UAS	Published 2023
	Number of Pages 38	
Title of Publication Work Steps for Creating a 3D Game Asset Case: Making a Car Game Asset		
Degree, Field of Study Bachelor of Engineering, Information and Communications Technology		
Organisation of the client (if the thesis work is commissioned by another party) Henri Koskinen, CEO & Media Producer, Pixtell Oy		
Abstract <p>The goal of the thesis was to investigate the steps involved in creating a 3D game asset and how to make an optimized model for a video game. The research is done from the perspective of 3D applications.</p> <p>Client of the thesis was Pixtell Oy which develops a game called LAST DROP – Legend of Seppo. The task was to design, model and texture a 3D asset while considering the limitations set by video games. The asset is supposed to be used in the video game.</p> <p>The purpose of the thesis is to mention the most essential things what the work steps of creating game assets include and what things must be taken into account so that the asset is as functional as possible in video game. The work steps of creating a game asset consist of planning, modeling, texturing, and post-processing.</p> <p>As a conclusion in the process of creating a game asset it was noticed that different work phases influence each other. Good planning makes it easier to create an asset.</p>		
Keywords game asset, 3D modeling, 3D model, texturing, texture map		

Sisällys

1	Johdanto.....	1
2	3D-mallinnus.....	2
2.1	3D-mallien rakenne.....	2
2.2	3D-mallien ero pelien ja elokuvien välillä	3
2.3	3D-mallinnusohjelmat	4
3	Suunnitteluvaihe	5
3.1	Pelin visuaalinen tyyli ja konseptitaide	5
3.2	Referenssit	6
3.3	3D-mallin käyttötarkoitus.....	6
3.4	Pelialusta	7
4	3D-mallintaminen.....	8
4.1	Polygonien määrä ja sijoittaminen	8
4.2	Level of Detail.....	9
4.3	3D-mallintamisen tekniikat	10
5	Teksturointi	13
5.1	UV-kartoitus.....	13
5.2	Tekstuurikartat.....	14
5.2.1	Värikartta	15
5.2.2	Normaalikartta ja kohoumakartta	16
5.2.3	Kiiltokartta ja karheuskartta.....	17
5.2.4	Läpinäkyvyyskartta	17
5.3	Tekstuurikartan koko	18
5.4	Tekstuurikarttojen tekeminen.....	19
6	Jälkikäsittely	20
6.1	Assetin viimeistely	20
6.2	Rigging ja animaatio	20
6.3	Törmäysobjekti	21
6.4	Tiedostomuoto.....	22
7	CASE: Auto peliassetin tekeminen.....	24
7.1	Toimeksianto ja tavoite	24
7.2	Suunnittelu	24
7.3	Mallinnus	25
7.4	Teksturointi	28
7.5	Jälkikäsittely	32

8	Yhteenveto ja pohdinta	34
	Lähteet	35

Termit

2D	Kaksiulotteinen
3D	Kolmiulotteinen
Alkupiste (origin)	Piste 3D-avaruudessa, jonka sijainti on (0, 0, 0)
Assetti	Elementti, jota käytetään pelin teossa
Reuna (edge)	Viiva kahden verteksin välillä
Face	Polygonin pinta
Konseptitaide	Kuvallinen luonnos tai suunnitelma
Kuvataajuus	Näytölle sekunnissa piirtyvien kuvien määrä
Loop	Verteksien ja reunojen muodostama rengas
Ngon	Polygoni, jossa on enemmän kuin neljä verteksiä yhdistettynä toisiinsa reunoilla
Objekti	Asia tai esine, joka on mallinnettu
Pelimoottori	Sovellus, jolla voidaan rakentaa pelejä, kuten Unreal Engine ja Unity
Polygoni	Vähintään kolme verteksiä yhdistettynä toisiinsa reunoilla
Primitiivi	Usein käytettyjä 3D-rakenteita, jotka tulevat valmiiksi mallinnettuna mallinnussovelluksessa, esimerkiksi kuutio, pallo, ympyrälierö
Renderöinti	Prosessi, jossa tietokone luo 2D- tai 3D-kuvan
RGB	Väriavaruus, joka muodostuu sekoittamalla punaista, vihreää ja sinistä
Tekstuurikartta	2D-kuva, joka asetetaan 3D-mallin pinnalle, jolla vaikutetaan mallin ulkonäköön
Topologia	Kuvastaa miten mallin rakenteen osat kulkevat mallin pinnalla
UV-kartta	3D-mallin rakenteesta luotu 2D-kuva
Verteksi	Piste 3D-avaruudessa

1 Johdanto

3D-mallien luominen pelissä käytettäväksi asettaa omat rajoitteensa verrattuna elokuvaan käytettäväksi. Peliassetin luominen on aikaa vievä prosessi ja isoissa yrityksissä eri työvaiheet on jaettu useammalle ihmiselle. Peliassetin tekemisen eri työvaiheilla on vaikutusta toisiinsa, joten onnistuneen lopputuloksen saamiseksi, on hyvä tietää, mitä eri työvaiheissa tehdään. Videopelialalla käytetään paljon kolmion muotoisia polygoneja, mutta 3D-sovelluksissa polygoneista halutaan nelikulmaita. Nelikulmaiset polygonit toimivat parhaiten 3D-mallia muokatessa ja animoidessa.

Opinnäytetyössä tutkitaan millaisia vaiheita 3D-peliassetin luominen sisältää, sekä millainen 3D-malli on hyvä peleissä käytettäväksi. Peliassetin luominen alkaa suunnittelulla mitä halutaan luoda. 3D-mallintamisessa joudutaan kiinnittämään paljon huomiota mallin topologiaan ja polygonien määrään. Tekstuurikarttojen avulla mallin pintaan saadaan värit ja muita yksityiskohtia. 3D-malli tarvitsee tekstuurikarttojen lisäksi vielä törmäysobjektin ja mahdolliset animaatiot ollakseen valmis peliassetti.

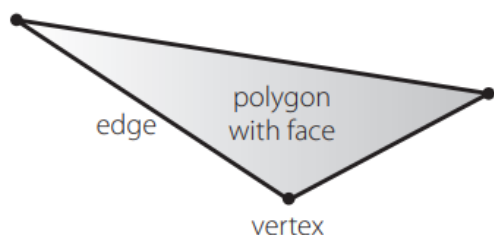
Peliassetin luomisprosessissa käytetään useampia erilaisia sovelluksia. Opinnäytetyö tutkii työn aiheita eniten 3D-sovelluksen näkökulmasta. Työssä ei tutkita pelimoottoreita eikä perehdytä sovelluksissa käytettäviin työkaluihin tai vertailla erilaisten sovelluksien ominaisuuksia.

Opinnäytetyön toimeksiantajana on videontuotantofirma Pixtell Oy, jonka kehitteillä olevaa peliä varten on tarkoitus suunnitella, mallintaa ja teksturoida auto-assetti. Assetin teossa kiinnitetään huomio, miten malli saadaan optimoituksi peliä varten sekä miten mallin polygonimäärä saadaan pidettyä mahdollisimman pienenä. Assetin tiedosto on tarkoitus saada mahdollisimman selkeäksi tarvittavia myöhempiä muokkauksia varten.

2 3D-mallinnus

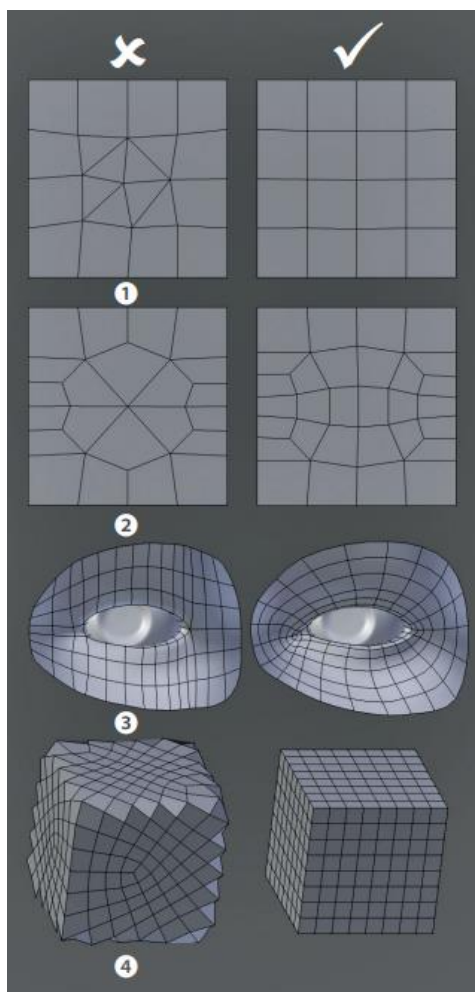
2.1 3D-mallien rakenne

Verteksit (vertex), reunat (edge) ja facet ovat olennaisia 3D-mallin rakenteen osia (Fisher 2012, luku 4). Tietokonegrafiikassa mallin pisteitä kutsutaan vertekseiksi. Verteksien välillä olevia viivoja kutsutaan reunoiksi. Polygoni muodostuu, kun vähintään kolme verteksiä on yhdistettynä toisiinsa reunoilla. Faceksi kutsutaan polygonin sisällä olevaa pintaa. Kuviossa 1 näkyy polygonin rakenne. (Totten 2012, 10.) Nelikulmioisissa polygoneissa on neljä reuna, jotka ovat liitettyä toisiinsa neljällä verteksillä. Ngon on polygoni, jossa on enemmän kuin neljä reuna yhdistettynä toisiinsa vertekseillä. Kolmioiksi kutsutaan polygoneja, joissa on kolme reuna toisissaan kiinni vertekseillä. Mallintaessa pyritään luomaan vain nelikulmaisia polygoneja, mutta kolmioilta ja ngoneilta ei usein pystytä välttymään rakennetta muokatessa. (Pluralsight 2014b.)



Kuvio 1. Polygonin rakenne (Totten 2012, 10)

Topologialla kuvataan miten mallin rakenteen reunat ja facet yhdistyvät toisiinsa ja kulkevat mallin pinnalla. Hyvässä topologiassa vältetään kolmioita, ngoneja ja poleja. Tärkeiden muotojen ympärille luodaan looppeja ja reunat suunnataan muotoa myötäileväksi. Pole on verteksi, jossa kolme, viisi tai enemmän kuin viisi reuna kohtaa. Polet voivat aiheuttaa virheitä rakennetta jakaessa pienimpiin osiin. Lisätyt loopit tekevät mallin rakenteesta helpommin muokattavan ja animoitavan. Esimerkiksi 3D-malleissa silmien ympärille lisätään looppeja. (Simonds 2013, 49–50.) Reunat suuntaamalla mallin ääri viivoja myötäileväksi saavutetaan puhdas ulkonäkö. Hyvä reunojen suuntaus auttaa pitämään mallin tehokkaana ja animaatioita varten rakenteen muovautuvana. (McKinley 2005, 24.) Kuvassa 1 on esimerkki mitä on huono topologia vasemmalla ja oikealla mikä on hyvää topologiaa.



Kuva 1. Huono ja hyvä topologia (Simonds 2013, 51)

2.2 3D-mallien ero pelien ja elokuvien välillä

Peleihin ja elokuvaan mallintaminen ovat samanlaisia tekniikoiltaan, mutta valmistustapa ja rajoitukset ovat erilaisia. Elokuviissa ei oikeastaan ole rajaa, kuinka monta polygonia sallitaan yhdelle mallille, koska elokuvissa tärkeintä on, että malli näyttää ruudulla hyvältä. Peleissä pelimoottori sekä laitteisto, millä peliä pelataan, rajoittavat polygonien määrää. (Pluralsight 2014a.)

3D-elokuvissa animaation jokainen kuva renderöidään etukäteen. Renderöidyt kuvat asetetaan peräkkäin elokuvan animaation luomiseksi. Elokuvien yhden kuvan renderöimiseen voi mennä aikaa tunteja, päiviä tai viikkoja. Pitkien renderöimisaikojen takia elokuvastudiot käyttävät renderöintifarmeja, joissa monilla toisiinsa yhdistetyillä tietokoneilla hoidetaan renderöimistä samanaikaisesti. Videopelien jokaisessa kuvassa pelilaitteen pitää reaaliaikaisesti renderöidä 3D-ympäristö, mallit, valot ja tekstuurit ja vuorovaikutus eri peliobjektien välillä. (Totten 2012, 13.) Reaaliaikainen renderöiminen aiheuttaa peleissä rajoituksia, kuinka yksityiskohtainen 3D-malli voi olla (Becker 2022). Polygoniluku vaikuttaa suuresti

sulavan pelattavuuden saavuttamiseen. Pelikonsoleissa ja -moottoreissa on rajoitettu, kuinka monta polygonia voi olla ruudulla samanaikaisesti. Pienempi polygonimäärä ei tee mallista vähemmän mielenkiintoista. Pelien mallit ovat tehokkaampia elokuvien malleihin verrattuna rajoitetun polygonimäärän takia. (Totten 2012, 13–14.)

2.3 3D-mallinnusohjelmat

Tietokoneiden 3D-mallinnusohjelmilla rakennetaan matemaattisesti 3D-malli. 3D-malleja hyödynnetään monilla erilaisilla toimialoilla kuten videopeleissä, elokuvissa, televisiossa, arkkitehtuurissa, tuotekehityksessä ja lääketieteellisyydessä. 3D-mallinnusohjelmia käytetään esimerkiksi peli- tai elokuvahahmojen luomiseen, tuotteiden suunnittelussa tai rakennuksien sisä- ja ulkopuolien visualisoinnissa. (Autodesk b.) 3D-mallinnussovelluksia on useita ja niiden välillä on eroavaisuuksia, minkälaiseen mallintamiseen ne sopivat parhaiten (All3DP 2023).

3ds Max on ollut Autodeskin omistuksessa vuoden 1990-luvun lopusta lähtien. 90-luvun alkupuolella sovellus oli nimellä Studio 3D. (All3DP 2023.) 3ds Max on ammattikäyttöön tarkoitettu sovellus 3D-mallintamiseen, renderöimiseen ja animaatioiden luomiseen. 3ds Maxin lisenssin hinta vuodessa vaihtelee 2000e ja 3000e välillä, riippuen haluaako maksun suorittaa kerran vuodessa vai joka kuukausi. 3ds Max toimii ainoastaan Windows-käyttöjärjestelmissä. (Autodesk a.)

Blender on ilmainen avoimen lähdekoodin sovellus, joka soveltuu hyvin animoimiseen ja mallintamiseen. Blenderiä käyttävät ihmiset vaihtelevat harrastelijoista, opiskelijoihin ja ammattilaisiin. (All3DP 2023.) Blender on Blender Foundationin vuonna 2002 julkaisema sovellus. Blender tukee koko 3D-mallintamisen prosessia, kuten mallintamista, animaatioita, renderöimistä, videomuokkausta ja pelien luomista. Blender toimii yhtä hyvin Windows-, Mac- ja Linux -käyttöjärjestelmillä. (Blender b.)

Blenderin lisenssi on GNU General Public License (GNU GPL), joka mahdollistaa sovelluksen käyttämisen kaikkeen mitä halutaan. Blenderissä luodut työt ovat työn tekijän omaisuutta. Blender ei aseta sillä luoduille tuotteille kaupallisia rajoituksia ja sitä voidaan käyttää myös opetuksessa. Kaupallisia rajoituksia voi ilmetä muissa sovelluksissa, joissa käytetään kokeilujaksoja tai opiskelijoille suunnattuja versioita. (Blender a.)

3 Suunnitteluvaihe

3.1 Pelin visuaalinen tyyli ja konseptitaide

Visuaalinen ohjenuora takaa yhteneväisen ulkonäön koko pelin läpi. Pelihahmojen ja niiden ympäristön ei pitäisi olla liian erilaisia laadultaan tai visuaaliselta tyyliiltään. Visuaaliset tyylit voidaan jakaa kahteen osaan: realistinen ja tyylitelty. Visuaalinen tyyli voi olla myös jotain näiden kahden väliltä, mikä näkyy kuvassa 2. (Eng 2015, 8.) Konseptitaiteen avulla tuodaan esille pelin visuaalista tyyliä ja saadaan kuva, millainen 3D-malli halutaan luoda (Totten 2012, 4).



Kuva 2. Tyyli realistisesta tyyliteltyyn (Eng 2015, 8)

Konseptitaiteella hahmotetaan pelin ulkonäköä, tyyliä ja tuntumaa, ennen 3D-mallien tuottamista. Konseptitaiteen avulla helposti kommunikoidaan pelin visuaalinen tyyli laajemmalle tuotantotiimille. Konseptitaide auttaa pitämään pelin ympäristön, asujen, aseiden ja muiden asettien ulkonäön yhteneväisinä ja tyylin johdonmukaisena. (Adobe b.) Konseptit voivat olla kynällä piirrettyjä hahmotelmia, veistoksia, 3D-malleja tai muita tapoja, joilla voidaan välittää visuaalinen ulkonäkö toisille ihmisille (Mitchell 2012, 82). Yksinkertaisissakin malleissa pelin ulkonäön hahmottaminen 2D-kuvien avulla auttaa 3D-työn tekemistä. (Totten 2012, 4.)

Konseptitaide on nopea tapa ideoiden kokeiluun ja opastaa mallin teossa (Simonds 2013, 19). Mallintamisen aloittaminen ilman konseptia voi johtaa viikkoja tai kuukausia tehtyyn turhaan työhön, jos idea ei olekaan hyvä. Konseptitaiteella huomataan, onko idea toimiva. Jos konseptitaiteessa nähdään, että idea ei ole hyvä, aikaa on mennyt hukkaan vain tunteja

tai päiviä. Yleisesti on nopeampaa testata ideoitaan piirtämällä, kuin muokkaamalla 3D-mallia. (Gahan 2011, 330.)

3.2 Referenssit

Käsitteellistäessään projektia voidaan käyttää tosielämän asioita referenssinä. Internetti, kirjat, tavarat ympärillä, ja otetut valokuvat ovat hyviä lähteitä referenssien etsimiseen. Fiktiivisissäkin malleissa tosielämän objektien tutkiminen, jotka jakavat jotain yhteistä työn kanssa, auttavat mallin visualisoinnissa. Esimerkiksi lohikäärmettä luodessa, saatetaan etsiä kuvia liskoista ja lepakoista. Liskon suomat ja lepakon siivet ovat piirteitä, joita voidaan hyödyntää lohikäärmeen suunnittelussa. (Simonds 2013, 19–20.)

Kenenkään työtä ei saada suoraan kopioida tekijänoikeuksien rikkomisen välttämiseksi. Mitä tahansa kuvaa tai taideteosta saadaan käyttää inspiraationa. Jos työ on suojattu tekijänoikeudella, pitää tekijänoikeuden haltijalta kysyä lupa työn käyttöön. Jos työ on vähemmän rajoittavan lisenssin alla, kuten Creative Commons, pitää mukautua lisenssin ehtoihin. (Simonds 2013, 20–21.)

3.3 3D-mallin käyttötarkoitus

3D-assetit vaihtelevat merkityksellisyydeltään peleissä. Huomioiden kuinka paljon pelaaja käyttää aikaa assetin katsomiseen määritellään, kuinka paljon yksityiskohtia malli tarvitsee. (Wacom). Arvioidakseen kuinka monta polygonia mallissa kannattaa käyttää riippuu usein seuraavista kolmesta tekijästä:

- Kuinka tärkeä malli on.
- Kuinka kaukana pelaaja on mallista.
- Kuinka paljon pelaaja on vuorovaikutuksessa mallin kanssa.

(Totten, 2012, 14).

Esimerkiksi kaukana sijaitseva laatikko tai rakennus voidaan rakentaa kuudesta polygonista ja niihin lisätyistä tekstuureista haluttujen yksityiskohtien luomiseksi. Polygoneja ei kannata tuhata malleihin, joita pelaaja ei läheltä näe. (Totten 2012, 14.)

Videopelien 3D-mallit voidaan jakaa eri luokkiin, joita ovat hahmot, propit (prop), kulkuneuvot, ympäristöt ja aseet. Hahmot ovat usein kaikista yksityiskohtaisimpia ja monimutkaisimpia 3D-malleja ja niiden tekemiseen menee eniten aikaa verrattuna muihin luokkiin. Propit ovat yleisesti peliobjekteja, jotka eivät ole todellisen maailman eläviä olentoja. Propit ovat usein asioita, joiden kanssa pelaaja on vuorovaikutuksessa tai ne ovat taustalla kullassa ja niitä tarvitaan monissa tapauksissa suuri määrä. Esimerkiksi kranaatit ovat propeja.

Kulkuneuvot muodostuvat usein monista toisiinsa yhdistetyistä osista ja niiden tekeminen vie aikaa. Kokonaista kulkuneuvon rakennetta ei aina tarvitse mallintaa, esimerkiksi monissa peleissä mallinnetaan vain auton ulkopuoli tai alue, jossa pelaaja istuu. Ympäristömalleja voivat olla yksittäiset assetit, esimerkiksi talo tai huone, tai kokonaisia pelin tason karttoja, alueita tai rakennuskomplekseja. Aseiden 3D-mallien usein oletetaan olevan tarkkoja fysiikoiden ja toimintansa suhteen. Pelaajat haluavat aseiden näyttävän hyviltä, joten niihin voidaan käyttää enemmän polygoneja. (3D-Ace 2021.)

3.4 Pelialusta

Pelialustalla tarkoitetaan laitteistoa, jolla peliä pelataan tai tiettyä jakelualustaa. Pelialustoja ovat esimerkiksi tietokoneet, pelikonsolit, puhelimet, VR-lasit ja Facebook. Jokainen pelialusta asettaa omat vaatimuksensa aseteille ja avaa erilaisia mahdollisuuksia teknillisen ja luovan puolen suhteen. Pelialustalla on suuri vaikutus pelin suunnittelussa, koska eri alustoilla on omat suorituskykynsä. (Kramarzewski & De Nucci 2018, 29.)

Tietokoneet ja konsolit sisältävät tehokkaan laitteiston, mikä mahdollistaa paljon yksityiskohtia sisältävien mallien käyttämisen. VR-pelit voivat olla yhtä raskaita kuin tietokonepelit. VR-peleissä huomataan mallien pienet yksityiskohdat, koska malleja katsotaan usein ensimmäisestä persoonasta ja läheltä. Mobiilipelit vaativat paljon optimointia, jotta puhelin ei ylikuumene tai hidastu, vaikka monet nykyaikaiset älypuhelimet ovatkin tehokkaita. Polygonien, asettien ja sisällön määrä on puhelinpeleissä yleisesti vähemmän. Nettisivuilla toimivat pelit eivät normaalisti toimi ilman internetyhteyttä, ja pelin lataaminen vie aikaa. Nettisivuilla olevat pelit vaativat paljon optimointia ja usein käytetään malleja, jotka sisältävät vähän polygoneja sulavan pelattavuuden saavuttamiseksi. (3D-Ace 2021.)

4 3D-mallintaminen

4.1 Polygonien määrä ja sijoittaminen

Suuri määrä polygoneja tekee 3D-mallin rakenteesta tarkemman, mutta monimutkainen paljon polygoneja sisältävä malli on vaikea animoida ja vie paljon tietokoneen resursseja. Jotta malli ei vähentäisi kuvataajuutta, halutaan mallissa käyttää vähemmän polygoneja ja lisätä yksityiskohtaiset tekstuurit. (Mitchell 2012, 126.) Vaikka mallille sallittaisiin suuri polygoniluku, mallintaessa kannattaa aina pyrkiä tehokkuuteen. 3D-malliin ei kannata lisätä polygoneja vain sen takia, että asetettu polygoniraja saavutettaisiin, vaan polygoneilla pitää olla jokin merkitys. Jos tietyllä polygonilla ei ole vaikutusta mallin yksityiskohtien tai rakenteen suhteen, se voidaan poistaa ja lisätä muualle mallin rakenteeseen. (McKinley 2005, 24.)

Alueille, joista 3D-mallin tarvitsee taipua tai liikkua, lisätään polygoneja, jotta animaatioprosessissa ei synny vääristymiä. Lisätty polygonimäärä auttaa myös saavuttamaan animaatiossa realistisemman liikkeen. Esimerkiksi kasvoihin lisätään polygoneja, koska ne käyvät läpi paljon muutoksia animaatioprosessin aikana. (Slick 2020.) Nelikulmiot muovautuvat hyvin animaatiota tehdessä. Kolmiot ja ngonit eivät muovaudu hyvin niiden liikkuessa (Totten 2012, 15). Kolmioita voidaan piilottaa mallin paikkoihin, jotka eivät liiku paljoa. Polet ovat hyödyllisiä esimerkiksi kasvoissa looppien uudelleenohjauksessa, mutta ne kannattaa sijoittaa alueille, jotka eivät liiku paljoa animaatiossa. (Caudron & Nicq 2015, 93.)

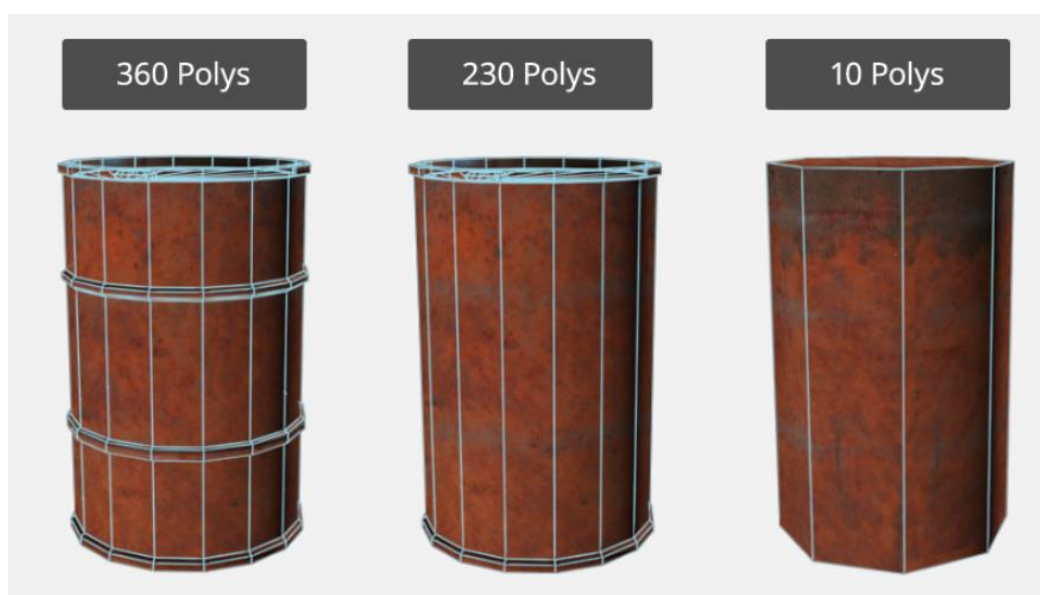
Mallintamista ei kannata aloittaa liian suurella määrällä geometriaa, vaan lisätä sitä aina tarpeen mukaan. Yksinkertaisemman geometrian kanssa työskentely on helpompaa ja helpottaa polygoniluvun pitämistä halutussa määrässä. (McKinley 2005, 24.) 3D-mallin voi rikkoa osiin, jotta yksityiskohtaisemmat osat eivät tee muusta mallin rakenteesta tiheämpää kuin on tarve. Esimerkiksi auton oven runko voi olla geometrialtaan yksinkertainen, mutta ovenkahvassa on avaimenreikä. Jos oven runko ja kahva pidetään yhdessä, auton runkoon pitää lisätä geometriaa tarvittavien kahvan yksityiskohtien saamiseksi. Rungon ja kahvan pitäminen erillään mahdollistaa oven rungon polygoniluvun pysymisen pienenä, samalla kun ovenkahvaan voidaan lisätä yksityiskohtia niin paljon kuin halutaan. (Lampel 2021.) Mallintamisen loppupuolella usein tarvitaan mallin siistimistä, joka on usein mallin muokkaamista, reunojen suuntauksen parantamista ja polygoniluvun hallitsemista. (McKinley 2005, 25.)

Monet pelimoottorit renderöivät polygoneja vain yksipuolisesti optimoinnin takia. Pelimoottorissa polygoni on näkyvä yhdeltä puolelta, mutta on kokonaan läpinäkyvä toiselta puolelta. (Thorn 2017, 58.) Polygonin facen normaalin suunta määrittää, kumpi puoli on näkyvä.

Jokainen face sisältää normaalin, joka on suunnattu kohtisuorasti facen pintaan nähden. (Brito 2009, luku 4)

4.2 Level of Detail

Peleissä käytetään usein level of detail (LOD) -malleja (Pluralsight 2014a). LODit ovat samasta assetista tehtyjä vähemmän polygoneja sisältäviä versioita. LODeissa voi olla myös pienemmät tekstuurikartat. LODin luomisessa pyritään polygonien määrää puolittamaan jokaisen LOD-version kohdalla ilman, että assetin ääriviivat muuttuvat. (Gahan 2011, 9.) Kuvassa 3 nähdään tynnyristä tehdyt eri LODit ja niiden polygonimäärä.



Kuva 3. Tynnyrin LODit (Pluralsight 2014a)

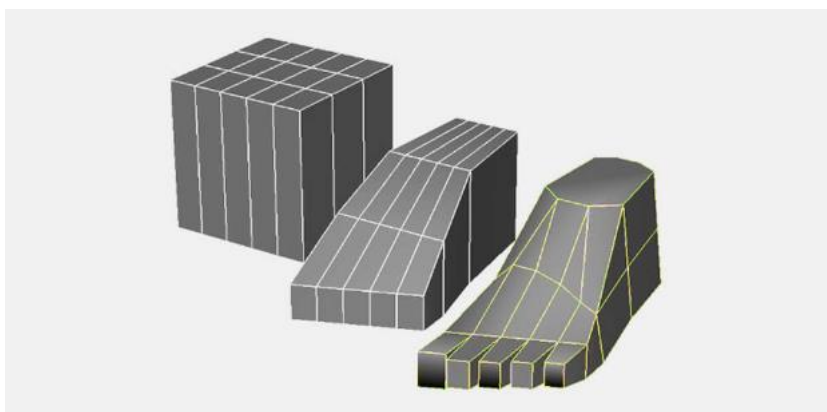
LODit ovat tarpeellisia pelin kuvataajuuden optimoimisessa, kun renderöidään laajoja maitsemia täynnä 3D-malleja. 3D-mallin ollessa lähellä pelaajaa renderöidään LOD, joka sisältää eniten polygoneja. Mallin siirryessä kauemmaksi vaihdetaan LODiin, joka sisältää vähemmän polygoneja. (Gahan 2011, 235.) Peliassetit lähimpänä pelaajaa voivat olla korkealaatuisia, koska pelimoottori ei tarvitse kaikkia näytöllä esiintyviä malleja renderöidä yksityiskohtaisimpina LOD-versioina (Pluralsight 2014a). LOD-ketju voi sisältää usein kahdesta neljään LODia tai enemmänkin, riippuen mallin monimutkaisuudesta ja pelimoottorista. Pelimoottoreissa on rajoituksia, kuinka monta LODia voi olla yhdessä projektissa. (Gahan 2011, 236.)

4.3 3D-mallintamisen tekniikat

3D-mallintamisen tekniikoita on useita, ja jokaisella tekniikalla on omat hyvät ja huonot puolensa (Brito 2009, luku 2). Parhain mallintamisen tekniikka valitaan sen perusteella, millainen 3D-malli halutaan muodoltaan ja yksityiskohdiltaan (Selin).

Box-mallinnus

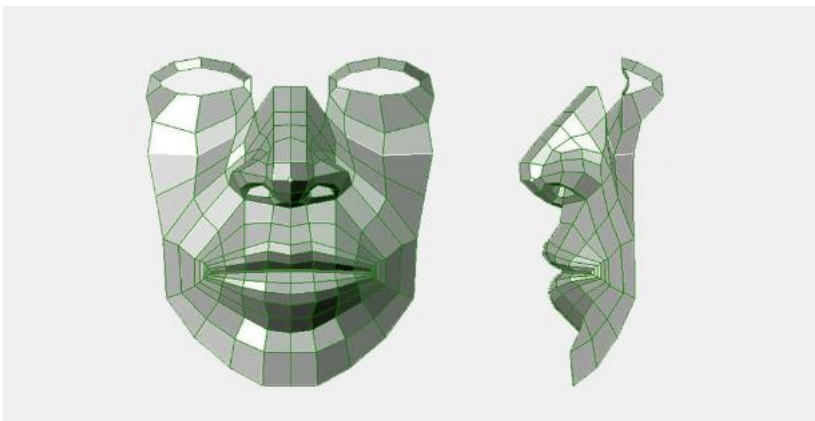
Box-mallinnus (box modeling) aloitetaan primitiivisestä objektista esimerkiksi kuutiosta tai pallosta. Objektin rakenteen muotoa lähdetään muokkaamaan mallintamisen työkaluilla halutun muodon saavuttamiseksi. (Selin.) Kuvassa 4 on esimerkki, miten kuutiota muokkamalla saadaan luotua jalka. Box mallinnus on nopea tapa luoda malleja vähäisillä yksityiskohdilla. Yksityiskohtien saavuttaminen voi olla vaikeaa, koska box-mallinnuksessa valmiiseen topologiaan pitää silloin lisätä looppeja ja uusia reunoja. (Brito 2009, luku 11.) Box-mallinnus toimii hyvin kovapintaisia objekteja luodessa, kuten arkkitehtuureissa tai ihmisten kehittämissä esineissä (Selin).



Kuva 4. Box-mallinnus (Roy 2023)

Polygonimallinnus

Polygonimallinnus (polygon modeling) on lähellä box-mallinnusta. Polygonimallinnus aloitetaan yhdestä reunasta tai muodosta, jossa ei ole syvyyttä. (Selin.) Esimerkiksi malli aloitetaan yksinkertaisesta tasosta ja tason reunoja lähdetään muokkaamaan, jotta saadaan luotua haluttu mallin muoto. Polygonimallinnuksella saavutetaan hyvä yksityiskohtien määrä, koska siinä hallitaan mallin pieniä alueita. (Brito 2009, luku 2.) Polygonimallinnusta käytetään usein kovapintaisten objektien tekemiseen, mutta se soveltuu box-mallinnusta paremmin malleihin, jotka sisältävät luonnonmukaisia piirteitä, kuten patsaat tai rakennuksien koristeet (Selin). Kuvassa 5 on polygonimallinnetut kasvat.



Kuva 5. Polygonimallinnus (Roy 2023)

Digitaalinen veistäminen

Digitaalinen veistäminen (digital sculpting) emuloi fyysisen veistoksen luomista digitaalisten työkalujen avulla (3D-Ace 2017). Box- ja polygonimallinnuksessa vaikutetaan ja nähdään suoraan mallin rakenne, mutta digitaalisessa veistämisessä harvoin mietitään polygonien sijoittumista. (Totten 2012, 106.) Digitaalinen veistäminen on tehokas tapa saavuttaa orgaanisia malleja, koska siinä voidaan keskittyä enemmän mallin muotoon ja miettiä vähemmän topologiaa. (Caudron & Nicq 2015, 55). Digitaalista veistämistä käytetään yleisesti hahmojen, eläinten ja olioiden luomisessa (Selin). Kuvassa 6 on digitaalisesti veistetty olennon pää, joka sisältää paljon yksityiskohtia.



Kuva 6. Digitaalisesti veistetty malli (3D-Ace 2017)

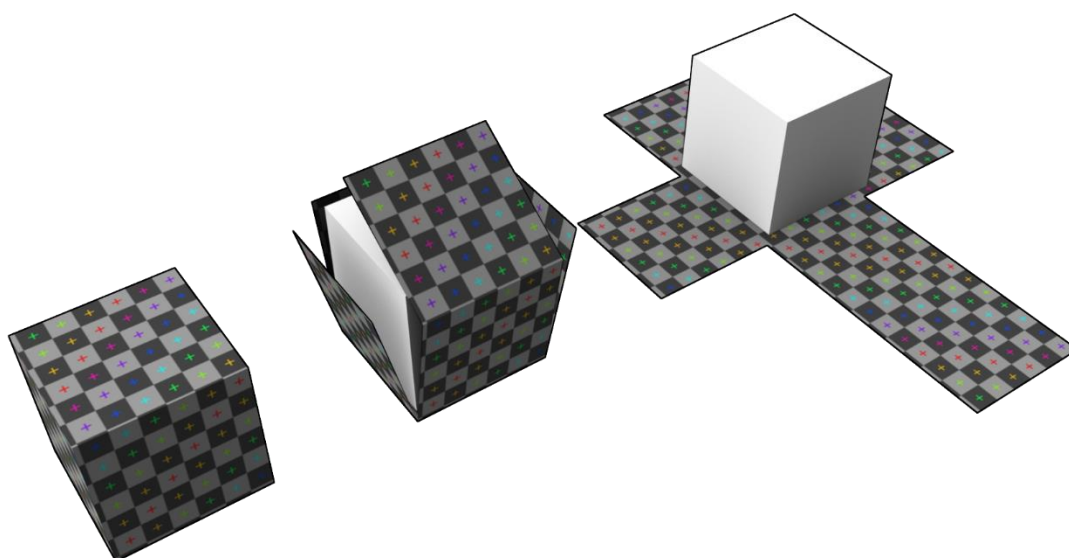
Digitaalinen veistäminen muokkaa mallin geometriaa vaikuttamalla verteksien sijaintiin. Pienien yksityiskohtien saamiseksi 3D-mallin polygoniluvun pitää olla suuri, mikä estää digitaalisesti veistettyjen mallien käyttämistä sellaisinaan peliassetteina. Digitaalisessa

veistämisessä ei usein kiinnitetä huomiota topologiaan, mikä johtaa rakenteeseen, joka ei toimi hyvin animoidessa. (Totten 2012, 106.) Digitaalisesti veistetyille mallille joudutaan tekemään retopologia (retopology), jonka tavoitteena on luoda puhtaampi versio alkuperäisestä veistoksesta hyvällä topologialla ja pienemmällä määrällä polygoneja. (Caudron & Nicq 2015, 55). Digitaalisesti veistetyistä malleista voidaan luoda normaalikartta, joka lisää vähemmän polygoneja sisältävään malliin yksityiskohtien lisäämiseksi. Normaalikartalla saadaan mallin pinnalle luotua yksityiskohtia polygonilukua muuttamatta. (Totten 2012, 106.)

5 Teksturointi

5.1 UV-kartoitus

UV-kartoitus on prosessi, missä 3D-mallin geometria puretaan ja muutetaan 2D-kuvaksi tekstuurikarttojen lisäämistä varten. UV-karttauksen prosessi alkaa saumojen määrittämisellä 3D-malliin. Saumat ovat merkitsevät kohtia, joista mallin rakenne leikataan auki. Tästä syntyvät osat asettuvat litteäksi UV-karttaan. Kuvassa 7 esitetään, miten kuution rakenne saadaan aukaistua litteäksi UV-kartaksi. Polygoneja halutaan pitää yhdessä tekstuurien ja rakenteen perusteella. (Totten 2012, 89.) Samalla tavalla, kun 3D-avaruudessa käytetään X-, Y- ja Z-akseleita kuvaamaan leveyttä, korkeutta ja syvyyttä, UV-kartassa U ja V kuvaavat leveyttä ja korkeutta (McKinley 2005, 86).



Kuva 7. Kuution UV-kartta (Zephyris)

UV-kartan saumat suositaan asettamaan teräviin kulmiin tai alueisiin, joista pelaaja ei niitä näe. Saumat ovat huomaamattomia kohdissa, joissa tekstuuri muuttuu radikaalisti. UV-kartassa tyhjää tilaa jätetään mahdollisimman vähän, jotta pikselien tiheys tekstuurissa saadaan pidettyä mahdollisimman suurena. (Plowman 2016, 60–61, 146.) Jos UV-kartan tilaa ei käytetä tehokkaasti, tarvitsee asetti todennäköisesti suuremmat tekstuurikartat hyödyntämättömän tilan takia (Rafiee). Selkeä UV-kartta on tärkeä, jos tekstuurien tekijä on joku toinen henkilö tiimistä (Plowman 2016, 60). Huono kartoitus voi aiheuttaa selkeitä vääristymiä mallin pinnan ulkonäköön ja tekstuurin venymistä tai litistymistä (Adobe 2022). Vääristymät voidaan tarkistaa helposti asettamalla ruudullinen tekstuuri 3D-malliin, kuten nähdään kuvassa 8, jossa vasemmalla tekstuurissa ei ole huomattavia vääristymiä ja oikealla tekstuuri on venynyt ja litistynyt (Eng 2015, 40–41).



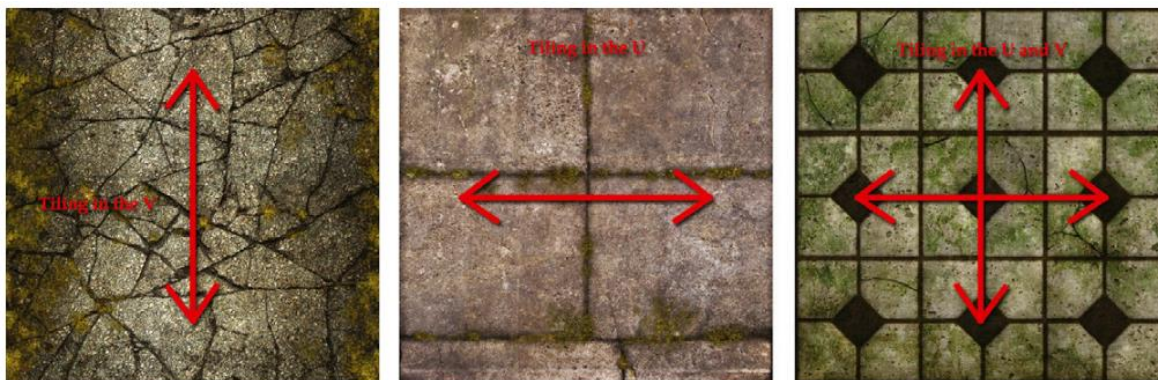
Kuva 8. Tekstuurien vääristymien tarkistaminen (Adobe 2022)

Saumoja pitää olla tarpeeksi, jotta mallin rakenne ei ole epämuodostunut UV-kartassa. Epämuodostumat syntyvät, kun 3D-sovelluksen pitää vääntää ja venyttää mallin geometriaa, jotta rakenne saadaan asetettua litteäksi. Jos käyttää epämuodostunutta UV-karttaa tekstuurien tekemiseen, tulevat tekstuurit näyttämään vääristyneiltä. Saumoja ei suositeta laittamaan liikaakaan, jotta tekstuureissa ei tarvitse maalata liian monen sauman päältä. (Simonds 2013, 112.)

5.2 Tekstuurikartat

Peliasettien rajoitetun polygonimäärän takia tekstuurit ovat peliartisteille tärkeitä työkaluja yksityiskohtien saamiseksi. Tekstuurit ovat 3D-mallin pinnalle asetettuja 2D kuvatiedostoja, jotka muuttavat mallin ulkoista ilmettä. Yksinkertaisimmatkin tekstuurikartat voivat huomattavasti parantaa 3D-mallin ulkonäköä. (Totten 2012, 17, 88.) Tekstuurikarttojen tarkoitus on määrittellä esineen väri, tuntuma ja lisätä yksityiskohtia, joita ei 3D-objektiin mallinneta polygoneilla. (Gahan 2011, 10, 13.) Tekstuurit vaihtelevat yksinkertaisista toistuvista kuoseista ainutlaatuisiin kuviin, jotka on luotu tiettyä 3D-mallia varten. Tekstuureissa on olemassa kaksi pääalajia, tiling tekstuurit ja uniikit tekstuurit. (Adobe a.)

Tiling tekstuureja, jotka ovat saumattomia U- tai V-akselin tai molempien suuntaan, voidaan käyttää uudelleen mallin pinnalla. Esimerkkejä tiling tekstuureista on kuvassa 9. Tiling tekstuureja käytetään yleisimmin peleissä peittämään laajoja alueita tai pintoja, jotka eivät tarvitse uniikkeja yksityiskohtia esimerkiksi tiiliseinässä, maassa tai maisemassa. Hahmoissa ja objekteissa, jotka vaativat erityisiä yksityiskohtia käytetään uniikkeja tekstuureja. (Gahan 2011, 13–14.) Uniikki tekstuuri luodaan tietylle mallille sopivaksi, eikä sitä voida käyttää muissa malleissa uudelleen (Adobe a).



Kuva 9. Tiling tekstuureja (Gahan 2011, 14)

5.2.1 Värikartta

Värikartta (color map tai diffuse map) on yleinen tekstuurikartta, jota käytetään monissa 3D-malleissa. Värikartalla saadaan pinnalle olennaisimmat värit ja yksityiskohdat. Hyvässä värikartassa ei ole varjoja tai korostuksia, vaan valo on jakautunut tasaisesti. (Gahan 2011, 10.)

Värikartta voi olla esimerkiksi pöydän pinnan puutekstuuri tai kokonaisen pelihahmon värit haarniskasta yksityiskohtiin. Kuvassa 10 on vasemmalla hahmolle tehty värikartta ja oikealla nähdään, miltä se näyttää mallin päällä. Värikartan lisäksi 3D-malleissa käytetään usein vähintään kahta muutakin tekstuurikarttaa, kiiltokarttaa ja normaalikarttaa. (Slick 2019.)



Kuva 10. Värikartta (Mitchell 2012, 132)

5.2.2 Normaalikartta ja kohoumakartta

3D-mallin polygonilukua vähennettäessä menetetään huomattava määrä yksityiskohtia (Rafiee). Normaalikartta (normal map) saa mallin pinnan näyttämään kuoppaiselta, ilman että se vaikuttaa mallin siluettiin. Normaalikarttoja luodaan usein mallista, johon on polygoneilla tehty kaikki yksityiskohdat. Normaalikartoilla saavutetaan illuusio korkearesoluutiosta mallista samalla pitäen mallin polygonit mahdollisimman vähäisinä. (Lanier 2015, 260.)

Normaalikartat ovat väriltään RGB-karttoja, eli sekoitus punaista, vihreää ja sinistä. Pelimoottorit ja 3D-sovellukset tulkitsevat normaalikartan väreistä tiedon, miten polygonin facen normaalit suunnataan, luoden illuusion, että yksityiskohdat olisi mallinnettu geometriaan. (Toten 2012, 106.) Kuvasta 11 nähdään miten normaalikartta (oikealla) lisää 3D-mallin pinnalle yksityiskohtia (vasemmalla). Normaalikartta on hyvä pienten yksityiskohtien lisäämiseen, mutta isot yksityiskohdat suositaan mallintamaan (Gahan 2011, 421).



Kuva 11. Normaalikartta (Akenine-Möller ym. 2018, 213)

Kohoumakartta (bump map) vaikuttaa mallin ulkonäköön samalla lailla kuin normaalikartta, eli se luo mallin pinnalle illuusion lisätyistä yksityiskohdista. Kohoumakartta on vanhempi verrattuna normaalikarttaan. Kohoumakartta on väriltään harmaasävyinen, jossa mustat ja tummat alueet ovat syvennyksiä ja valkoiset ja vaaleat alueet tulevat koholle. (Pluralsight 2022.)

5.2.3 Kiiltokartta ja karheuskartta

Kiiltokartta määrittelee, mitkä mallin osat ovat kiiltäviä ja kuinka vahvasti ne kiiltävät (Slick 2019). Kiiltokartta on väriltään mustavalkoinen, jossa mustat alueet ovat mattoja ja valkoiset kiiltäviä. Kiiltokartta on hyödyllinen luodessa materiaaleja, joiden pinnan kiiltävyydessä on vaihtelevuutta. (Totten 2012, 19.) Kiiltokartta on tarpeellinen esimerkiksi pelihahmossa, jonka iho, metallinen vyönsolki ja vaatteiden kankaat eroavat kiiltävyydeltään (Slick 2019).

Karheuskartta ja kiiltokartta ovat toisensa vastakohdat, joten esimerkiksi karheuskartta saadaan muutettua kiiltokartaksi muuttamalla kartan värit käänteisiksi (Price 2023). Kuvassa 12 on esimerkki, miten kiiltokartta saa oven lasiset ja metalliset osat näyttämään kiiltäviltä ja puun pinnan mattaiselta.



Kuva 12. Kiiltokartta (Gahan 2011, 12)

5.2.4 Läpinäkyvyyskartta

Läpinäkyvyyskartta (transparency map tai alpha map) on mustavalkoinen kartta, jossa mustat alueet ovat läpinäkyviä ja valkoiset alueet läpinäkymättömiä (Gahan 2011, 13). Läpinäkyvyyskarttojen avulla voidaan saavuttaa orgaanisia muotoja vaikuttamalla tason läpinäkyvyyteen. Läpinäkyvyyskartan avulla luodaan esimerkiksi lehtiä, kasvustoa, ketjuaitoja, likaisia ikkunoita geometrian pysyessä yksinkertaisena. (Totten 2012, 19.)

Läpinäkyvyyskarttaa käytetään pintoihin, jotka olisivat vaikeita mallintaa tai tietokoneelle liian raskaita kopioida. Esimerkiksi ketjuaidassa läpinäkyvyyskartta on vakuuttava, jos malli ei sijaitse liian etualalla ja läpinäkyvyyskartta mahdollistaa paljon vähäisemmän polygonimäärän. (Slick 2019.) Kuvassa 13 nähdään miten olennon partaa ei olla mallinnettu

yksityiskohtaisesti polygoneilla, vaan läpinäkyvyysskarttaa käytettiin tekemään polygoneista paikoitellen läpinäkyviä.



Kuva 13. Läpinäkyvyysskartta (McKinley 2005, 94)

5.3 Tekstuurikartan koko

Tekstuurikarttoja luodessa suositellaan alussa luomaan tekstuuri suurempana, kuin mitä projektiin tarvittaisiin. Suuriresoluutioiseen tekstuuriin mahtuu enemmän yksityiskohtia ja siitä voidaan luoda erikokoisia pienempiä versioita, joita voidaan hyödyntää, kun luodaan peliä eri pelialustoille. (McKinley 2005, 93.) Tekstuurikarttojen pikselien mittasuhteena suositetaan kahden potenssia. Yleisiä tekstuurikartan kokoja ovat seuraavat (Totten 2012, 147):

- 32x32
- 64x64
- 128x128
- 256x256
- 512x512
- 1024x1024
- 2048x2048
- 4096x4096.

Monet modernit pelimoottorit kykenevät muuntamaan vääränkokoisten tekstuurikarttojen mittasuhteet kahden potenssiin. Oikea tekstuurikarttojen koko varmistaa, että pelimoottori ei kuormitu tekstuurikarttojen kokojen muuntamisesta. (Totten 2012, 147.) Pelikonsoleissa

on erittäin rajoitettu muistin määrä tekstuureja varten verrattuna tietokoneisiin, joten niissä yleisesti suositaan pienempiä tekstuurikarttoja (McKinley 2005, 93).

5.4 Tekstuurikarttojen tekeminen

Tekstuurien tekemiseen on kolme erilaista merkittävää tekniikkaa. Tekstuurikartat voidaan piirtää ja maalaa käsin, oikeasta maailmasta voidaan skannata tai ottaa kuvia materiaaleista ja muuntaa ne tekstuureiksi tai voidaan hyödyntää proseduraalista teksturointia, jossa tietokoneen algoritmit luovat tekstuuriin. Usein tekstuurien teossa käytetään useampaa kuin yhtä tekniikkaa. (Adobe a.)

Valmis UV-kartta viedään grafiikanmuokkausohjelmaan, missä siihen maalataan tekstuurit hyödyntäen piirtotyökaluja tai käytetään valokuvia realistisen lopputuloksen saavuttamiseksi (Plowman 2016, 66). Tekstuurien tekemisen tapaan vaikuttaa, millainen tyyli halutaan. Halutessaan realistista tyyliä käytetään valokuvia, mutta piirrosmaiseen tyyliin maalataan itse tekstuurit. (Eng 2015, 96.)

Tekstuurien luominen piirtämällä tai maalaamalla antaa luovuudelle vapautta. Silloin voidaan lisätä tekstuuriin kuvioita ja muita elementtejä, kuten kulumaa ja naarmuja. Tekstuurien piirtäminen ja maalaaminen alusta lähtien vaatii paljon työtä. Proseduraalinen teksturointi on nopea tapa saada yksityiskohtia ja realismia malliin. Esimerkiksi proseduraalisella teksturoinnilla voidaan tehdä pieniä naarmuja ja haalistunutta väriä mallin kulmiin, riippuen mallin muodosta. Proseduraalisessa teksturoinnissakin on rajansa, kun yritetään kopioida jotain todellisesta maailmasta. Valokuvia muokkaamalla saadaan hyvin todellista maailmaa myötäileviä tekstuureja. (Adobe a.)

6 Jälkikäsitely

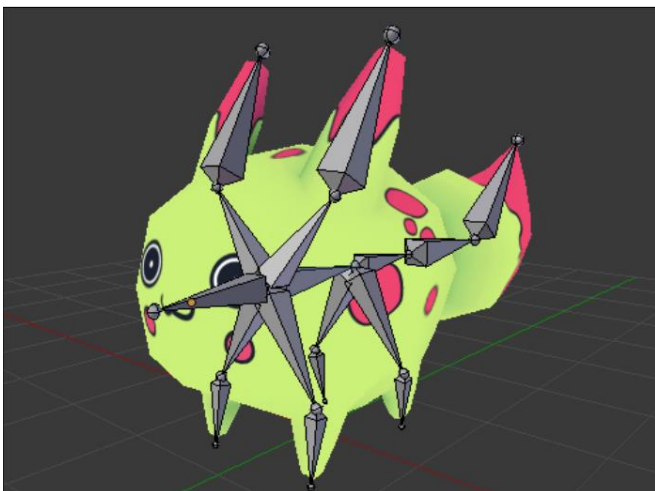
6.1 Assetin viimeistely

Pivot-piste kuvaa mallin paikallista keskikohtaa, joka määrittelee miten muutokset, kuten siirtäminen, kiertäminen ja skaalaaminen tapahtuvat. Peliasseteissa mietitään paras pivot pisteen sijainti mallin käyttötarkoituksen perusteella. Esimerkiksi maan päälle asetettavissa asseteissa halutaan pivot-piste asettaa mallin pohjaan, jotta assetti asettuu maan pinnan yläpuolelle. Pivot-piste asettuu usein 3D-sovelluksen alkupisteen (origin) mukaan, kun malli viedään pelimoottoriin. (Plowman 2016, 73–75.)

Suuressa peliprojektissa mallin skaala ja orientaatio ovat tärkeitä asioita hallita projektin alkuvaiheesta asti. Jos pelimoottorissa mallien koot eivät ole oikeassa suhteessa toisiinsa, mallien uudelleenskaalaaminen ja orientaation muuttaminen voi viedä paljon aikaa. (McKinley 2005, 26.) Monet pelimoottorit käyttävät mittana metrijärjestelmää. 3D-sovellukseen kannattaa asettaa mitat myös metreiksi, jotta mallien skaala on helpompi määrittää oikeaksi 3D-sovelluksen ja pelimoottorin välillä. (Thorn 2017, 220.)

6.2 Rigging ja animaatio

Valmiit 3D-mallit hahmoista, autoista ja muista objekteista tarvitsevat rigin (rig), jotta mallit saadaan liikkumaan peleissä tai muissa animaatioissa (Mitchell 2012, 128). Rigging on prosessi, jossa rakennetaan luuranko (armature) ja se linkitetään mallin rakenteeseen. Kuvassa 14 nähdään esimerkki miltä mallin luuranko näyttää Blenderissä. Animaatiot luodaan liikuttelemalla mallin sisällä olevaa luurankoa. Luuranko muodostuu useammasta luusta (bones). Luut voidaan määrittää vaikuttamaan 3D-mallin rakenteen tiettyjen verteksien kanssa, jolloin verteksit seuraavat luiden liikettä. Tyypillisessä pelianimaatiossa rigatulle mallille luodaan jaksollisia animaatioita, joita voidaan aktivoida pelimoottorin koodissa. Koodi ohjaa, milloin mallin animaatio esiintyy ja miten malli liikkuu animaation aikana. (Toten 2012, 158, 194.)

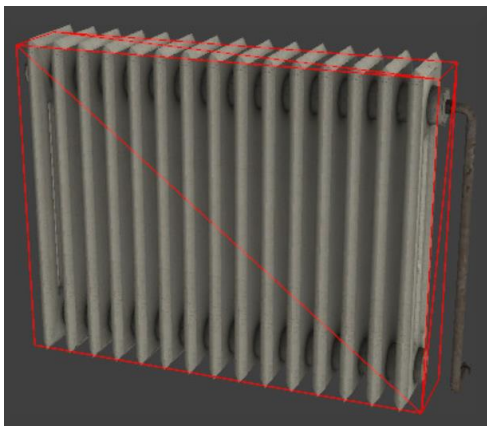


Kuva 14. Olion luuranko Blenderissä (Eng 2015, 69)

Rigging-prosessissa pelejä varten on rajoitteensa verrattuna elokuvia varten. Elokuviissa-luurangot voivat olla monimutkaisia ja sisältää enemmän osia, kuin peleihin luodut luurangot. Luurangot sisältävät paljon tietoa ja siksi vievät paljon muistia. Luurangot kertovat pelimoottorille ja mallinnussovellukselle, miten mallin rakenteen kuuluu muovautua ja miten luut vaikuttavat toisiinsa. Luissa on vanhempia (parent) ja lapsia (child), jotka ovat liitettynä toisiinsa. Lapsiluut seuraavat vanhempiluun muutoksia (sijaintia, kiertoa ja skaalaa) ja muita vanhemman attribuutteja. Luiden liikkeisiin voidaan myös asettaa rajoituksia. Liian monimutkainen luuranko hidastaa videopelin kuvataajuutta. (Totten 2012, 158, 160–161.)

6.3 Törmäysobjekti

Törmäysobjekti (collision) varmistaa, että pelaaja ja muut peliobjektit törmäävät oikein toisiinsa ja estää pelaajaa kävelemästä assettien läpi (Plowman 2016, 79, 82). Törmäysobjekti mahdollistaa vuorovaikutuksen assetin ja pelaajan välillä. Esimerkiksi pelaaja ei voi nostaa assettia, josta uupuu törmäysobjekti, vaan pelaajan käsi menee assetin läpi. (Epic Developer Community.) Törmäysobjekti on näkymätön ja sen rakenne pidetään yksinkertaisena verrattuna 3D-malliin (Valve Developer Community 2022). Kuvassa 15 lämpöpatterille on tehty suorakulmaisen särmiön muotoinen törmäysobjekti.



Kuva 15. Törmäysobjekti (Valve Developer Community 2022)

Näkymättömän törmäysobjektin ei tarvitse olla yhteneväinen mallin muodon kanssa, vaan muotoa karkeasti myötäilevä. Yksinkertainen törmäysobjekti on hyötysuhteeltaan parempi ja ero tarkan ja yksinkertaisen törmäysobjektin välillä on huomaamaton peleissä. Yksinkertaisimmat törmäysobjektit ovat muodoltaan primitiivejä ja useampia primitiivejä voidaan käyttää yhden törmäysobjektin luomiseen. (Unity Documentation 2023.) Liian monimutkaiset törmäysobjektit tekevät pelistä raskaan, mikä voi johtaa pelin kaatumiseen (Huston 2020).

6.4 Tiedostomuoto

3D-tiedostomuotoja on olemassa useita erilaisia. Neljä olennaisimpia asiaa, mitä 3D-tiedostot voivat pitää sisällään on tiedot mallin geometriasta, pinnan tekstuurista, 3D-ympäristön yksityiskohdista ja animaatiosta. Tiedostomuotojen välillä on vaihtelevuutta, mitä tietoja ne pitävät sisällään. Tiedostomuodot voidaan jakaa kahteen luokkaan, yksityisomistukselliset ja neutraalit tiedostomuodot. (Kelly & Chakravorty 2022.)

Yksityisomistukselliset tiedostomuodot on suunniteltu ja optimoitu toimimaan tietyssä sovelluksessa. Esimerkiksi Blenderissä blend-tiedosto ja AutoCADin DWG-tiedosto ovat yksityisomistuksellisia. Yksityisomistuksellisen tiedostomuodon huono puoli on, että se toimii vain siinä sovelluksessa, mitä sen luomiseen on käytetty. Neutraalit tiedostomuodot on suunniteltu toimimaan sovelluksesta riippumatta. Neutraalien tiedostomuotojen avulla vältytään yhteensopivuusongelmilta, joita esiintyy yksityisomistuksellisten tiedostomuotojen kanssa. Oikea tiedostomuoto valitaan riippuen, missä sovelluksissa sitä halutaan käyttää ja mitä tietoja halutaan siirtää. (Kelly & Chakravorty 2022.)

FBX on suosittu tiedostomuoto peliasettien viennissä pelimoottoriin. Monet suosittu 3D-sovellukset tukevat FBX-tiedostoja. FBX-tiedosto säilyttää hyvin mallin geometrian, materiaalit ja materiaalien asetukset. FBX pystyy myös siirtämään animaation tiedot. FBX on

patentoitu tiedostomuoto ja se on Autodeskin omistama. (Plowman 2016, 75.) FBX:n on kehittänyt Kaydara ja Autodesk osti sen vuonna 2006. (Kelly & Chakravorty 2022.)

7 CASE: Auto peliasetin tekeminen

7.1 Toimeksianto ja tavoite

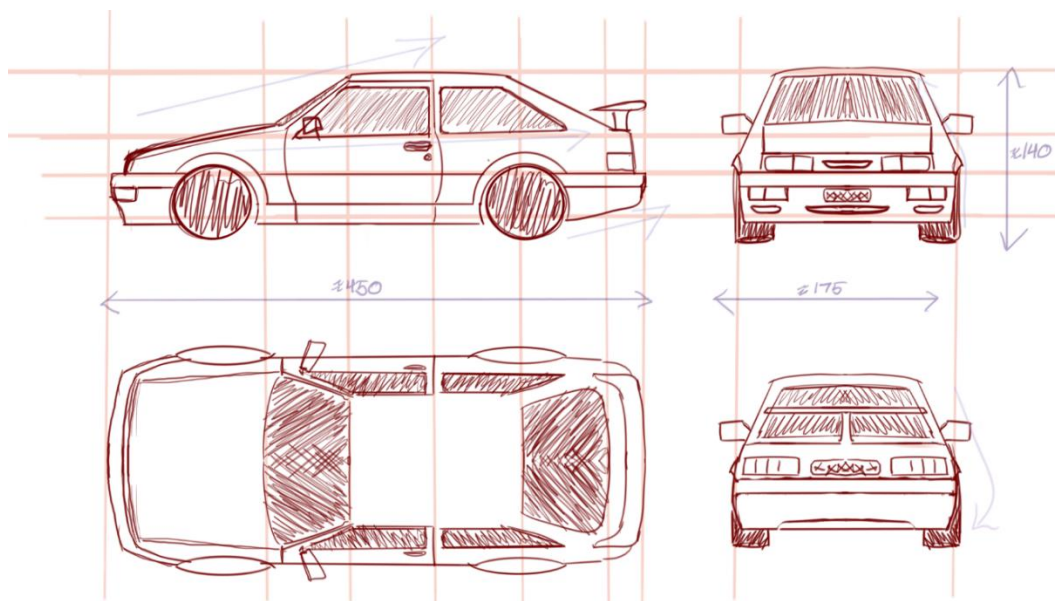
Työn toimeksiantajana toimii vuonna 2020 perustettu videotuotantoyritys Pixtell Oy, jolla on kehitteillä peli nimeltään LAST DROP – Legend of Seppo. Peli sijoittuu 1990-luvun Jyväskylään ja on visuaaliselta tyylieltään realistinen. Tehtävänä on suunnitella ja mallintaa 1900-luvun lopun teeman mukainen auto ja teksturoida se. Autoa tullaan käyttämään pelissä koristeena ja NPC:n (non-playable character) sekä pelaajan ajamana. Tavoitteena on luoda malli, jossa otetaan huomioon 3D-mallin optimointi peliä varten.

Mallinnusprosessissa kiinnitetään huomio mallin topologiaan, polygonien määrään, miten mallia tullaan käyttämään pelissä, sekä miten myöhemmin lisättävät animaatiot vaikuttavat rakenteeseen. Tekstuurit luodaan mahdollisimman optimoiduiksi. Työssä ei lisätä auton pintaan yksityiskohtia, kuten naarmuja tai ruostetta, sillä ne lisätään Unreal Engineessä. Rigejä, animaatiota, ja törmäysobjektia ei tarvitse tätä työtä varten tehdä. Ne tullaan lisäämään assettiin myöhemmin, koska ne ovat kuitenkin tarpeellisia pelin kannalta. Unreal Engineessä myös hyödynnetään automaattista LODien luomista, joten assetista ei ole tarve manuaalisesti tehdä LODeja.

7.2 Suunnittelu

Työnantaja antoi ohjeistuksen, millainen autoassetti pitää luoda peliä varten toiminnaltaan ja tyylieltään. Auton pitää olla 1980–1990-luvun mukainen, sulava muodoltaan, ei laatikko-mainen, viistoperäinen ja kolmiovinen. Työ aloitetaan etsimällä netistä mahdollisimman paljon referenssikuvia autoista, joka vastaa vaadittuja piirteitä. Netistä löydettyjen kuvien avulla luodaan konseptikuva piirtosovelluksella SketchBook. Konseptikuvasta tulee esiin auton ulkonäkö ulkopuolelta, ilman mitään tarkkoja yksityiskohtia.

Konseptikuvaa luodessa huomioidaan, ettei yhtä autoa voida kopioida suoraan sellaiseenaan peliin, jotta tekijänoikeuksia ei rikota. Siksi inspiraatiota otetaan useammasta eri autosta ja muokataan mittasuhteita. Kuva toimii myöhemmin ohjenuorana, kun aloitetaan mallinnusta. Kuvassa 16 nähdään auton konseptikuvaan piirretyt eri kuvakulmat. Yksityiskohtia aloitetaan lisäämään vasta, kun auton rakenne ja muoto on saatu tehtyä ja hyväksyttyä toimeksiantajalla.

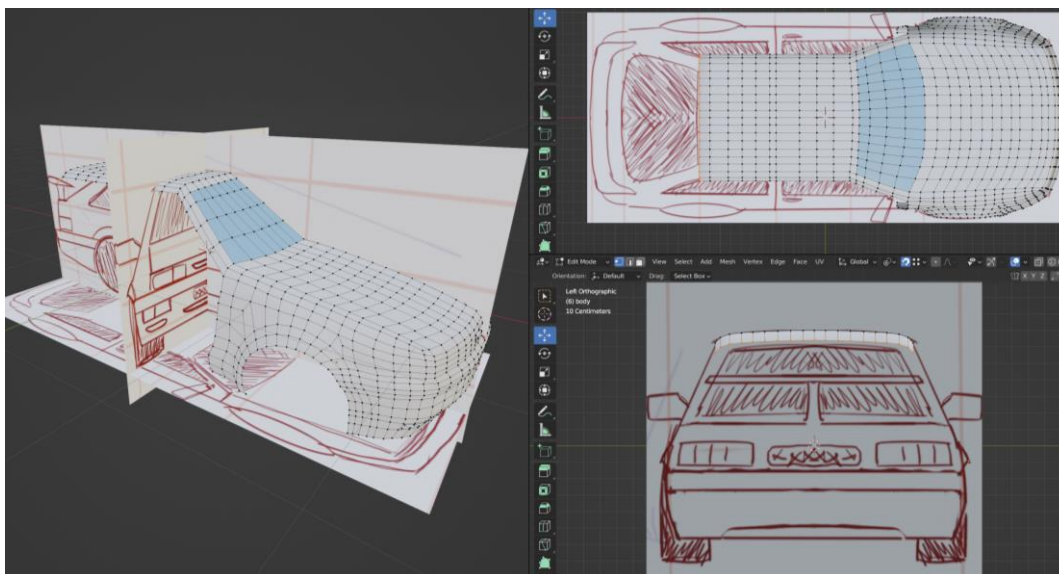


Kuva 16. Auton konseptikuva

Ennen mallintamisen aloittamista, pitää tietää miten autoa tullaan käyttämään pelissä, koska sillä on merkitystä 3D-mallin rakenteen suhteen. Pelaaja pystyy näkemään auton läheltä, joten polygoneja pitää olla tarpeeksi ja topologian siisti. Auton ovet pitää pystyä avaamaan, joten niiden pitää olla erillisinä osina animaatioita varten. Pelaaja pystyy itse ajamaan autoa, ja kameraa voi vaihdella pelissä kolmannelta persoonasta ensimmäiseen persoonaan, joten auto pitää olla mallinnettu ja teksturoitu ulkopuolen lisäksi sisältäkin. Et-sitään 90-luvun autojen sisältä kuvia referensseiksi, joiden pohjalta voidaan luoda auton sisäpuoli.

7.3 Mallinnus

Käytetään työntekoon Blenderiä. Ennen mallintamisen aloittamista asetetaan Blenderin mittaksi metrijärjestelmä, koska Unreal Engineissä käytetään metrejä mittakaavana. Yhteinen mittakaava heti alusta lähtien poistaa tarpeen skaalata mallia myöhemmin. Mallintamisen aloittamista varten asetetaan auton konseptikuva 3D-sovellukseen. Valitaan auton ulkokuoren mallintamistekniikaksi polygonimallinnus, koska silloin keskitytään suoraan polygonien asetteluun ja topologiaan. Ulkokuorta mallintaessa otetaan huomioon, että ovet pitää saada myöhemmin animoitua, joten asetetaan reunat ovien muotoja myötäileviksi, jotta ne voidaan irrottaa ulkokuoresta. Kuvassa 17 nähdään konseptikuvan käyttö Blenderissä ja auton rungon mallintamista. Parannellaan 3D-mallin topologiaa mallintamisen aikana.

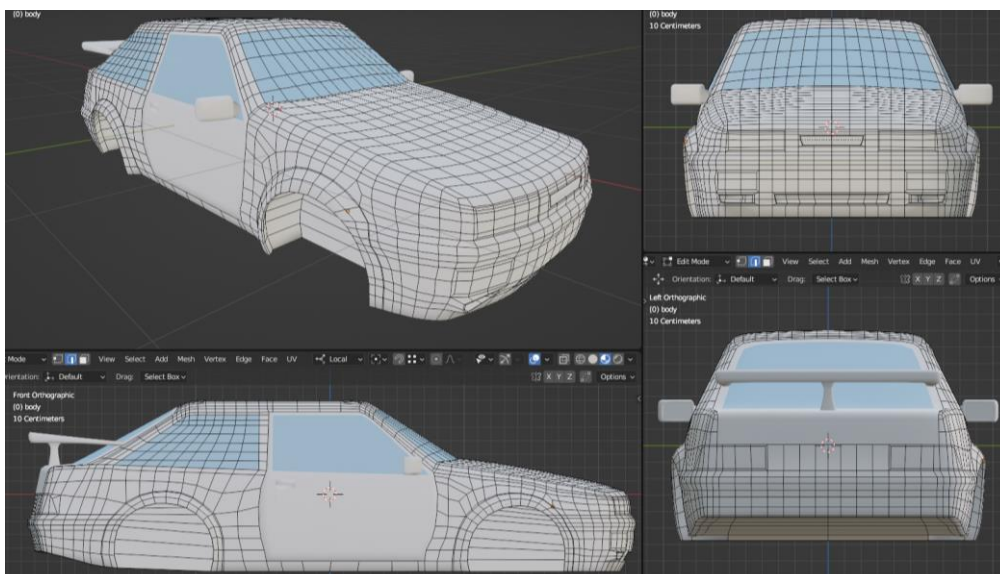


Kuva 17. Rungon polygonimallinnus

Mallintamisprosessin aikana luodaan tiedostosta tallennettuja kopioita, jotta virheiden satuttaessa voidaan palata aikaisempaan versioon. 3D-sovelluksessa luodaan objekteista myös kopioita ennen suurien muokkauksien tekemistä, jotta tilanteissa, joissa ollaan tyytymättömiä lopputulokseen, voidaan palata malliin, johon ei olla tehty muutoksia.

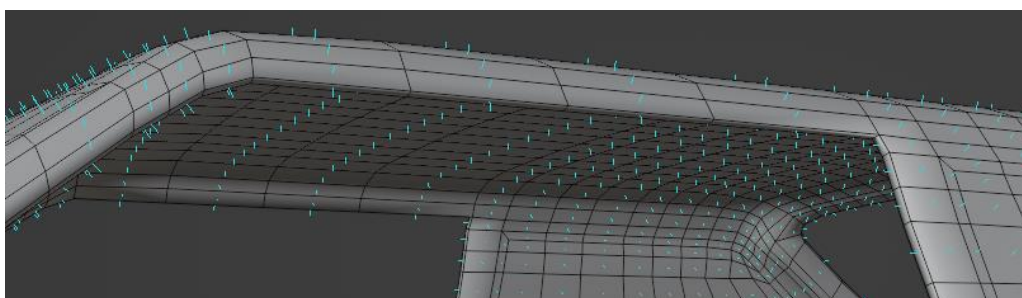
Mallin rungon muodon ollessa valmis, voidaan ovet irrottaa rungon rakenteesta omiksi objekteikseen. Kun ovet ja runko ovat irrallaan toisistaan pitää muistaa, että jos ovien reunoilla sijaitsevia verteksien sijaintia lähdetään muokkaamaan, syntyy helposti eri objektien välille rakoja. Pyritään pitämään ovien ääri viivoja vastaavat verteksit yhteneväisinä sijainniltaan rungossa ja ovissa.

Auton ulkopuolelta uupuu vielä spoileri ja sivupeilit. Mallinnetaan spoileri ja sivupeilit omina objekteinaan, jotta niiden mallintamista varten ei tarvitse rungon rakenteeseen lisätä polygoneja. Spoileri ja sivupeilit voidaan mallintaa box-mallinnuksella, koska ne ovat yksinkertaisia muodoiltaan. Aloitetaan spoilerin ja sivupeilien tekeminen kuutiosta, joita lähdetään muokkaamaan Blenderin työkaluilla. Auton rungon keulaan voidaan mallintaa yksityiskohdat. Kuvassa 18 on auton ulkopuoli kokonaan mallinnettuna ja siitä nähdään myös mallin reunojen suuntautuminen.



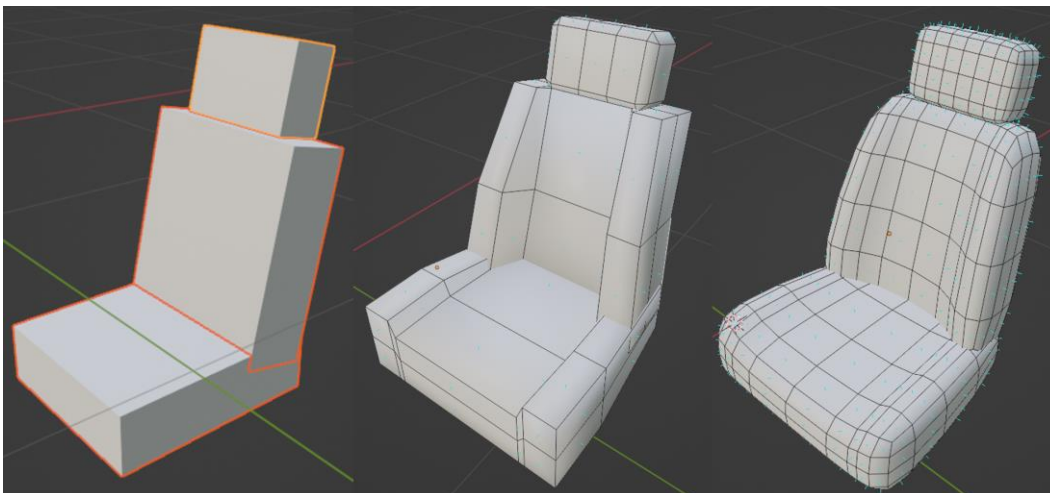
Kuva 18. Auton ulkopuoli mallinnettuna

Unreal Engine:ssä käytetään polygonien yksipuolista renderöimistä, joten auton sisäpuolelle tehdään polygonikerros, jonka normaalit osoittavat sisäänpäin. Kuvassa 19 nähdään normaalien suuntautuminen sinisillä viivoilla auton sisäpuolta varten luodussa kerroksessa.



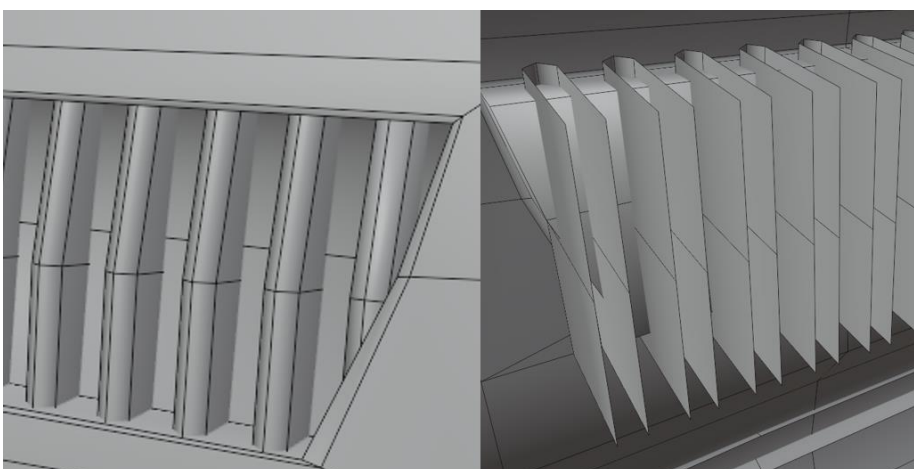
Kuva 19. Normaalien suuntautumista

Monien auton sisäpuolen osien mallintamisessa voidaan käyttää box-mallinnusta. Auton penkit, tuulettimet, polkimet, taustapeili, ratti saadaan luotua aloittamalla primitiivisestä muodosta. Esimerkiksi kuvassa 20 on vaiheet, miten etupenkki saadaan luotua lisäämällä rakenteeseen looppeja, siirtämällä polygoneja ja jakamalla polygonit pienempiin osiin (subdivision).



Kuva 20. Tuolin box-mallintaminen

Polygonimäärän pienentämiseksi voidaan poistaa polygoneja alueilta, joita ei pelaaja pysty näkemään. Esimerkiksi tuolia mallintaessa polygoneja jää piiloon, koska selkänoja ja istuinosa ovat osittain sisäkkäin. Kuvassa 21 on esimerkki, miten auton keulassa olevista yksityiskohdista, on auton rungon sisäpuolelle jäävät kokonaiset facet poistettu. Reunoja ja looppeja voidaan poistaa tai yhdistää toisiinsa alueilla, joissa ne eivät ole merkitseviä mallin ääriviivojen tai saumojen sijoittamisen suhteen.

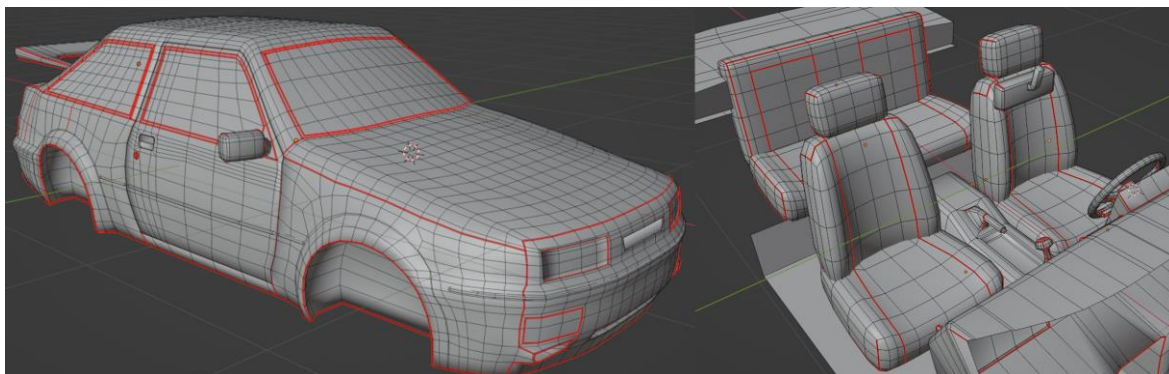


Kuva 21. Polygonimäärän vähentäminen

7.4 Teksturointi

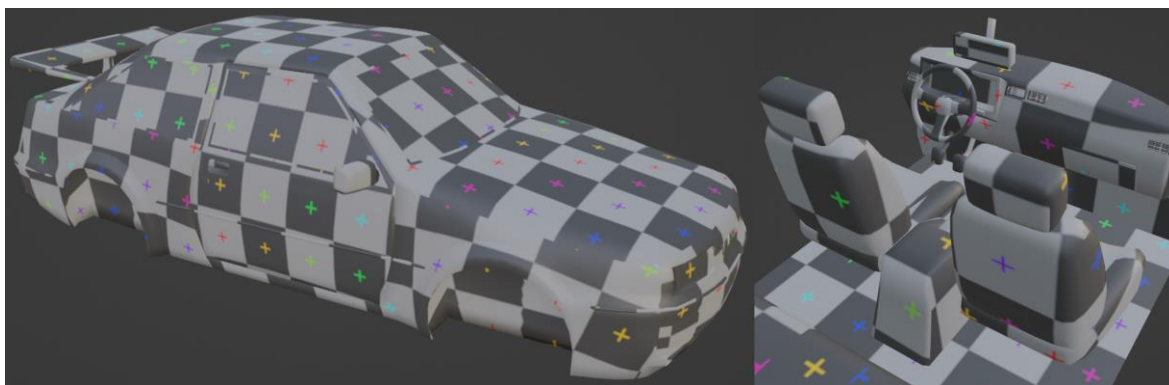
Työnantajan ohje UV-karttojen määrään on kaksi. Auton maalin väri halutaan omaksi UV-kartakseen, jotta pelimoottorissa voidaan myöhemmin satunnaistaa saman auton useat mallit eri värisiksi. Saumat asetetaan auton maalin värin reunoille, jotta värialue saadaan erotettua omaksi UV-kartakseen. Asetetaan saumoja myös alueille, joissa tekstuurien ulkonäkö tulee selkeästi eroamaan toisistaan. Pyritään piilottamaan saumoja objektien alle,

joista pelaaja ei voi niitä nähdä. Kuvassa 22 on malliin lisätyt saumat merkattu punaisilla viivoilla.



Kuva 22. Saumat

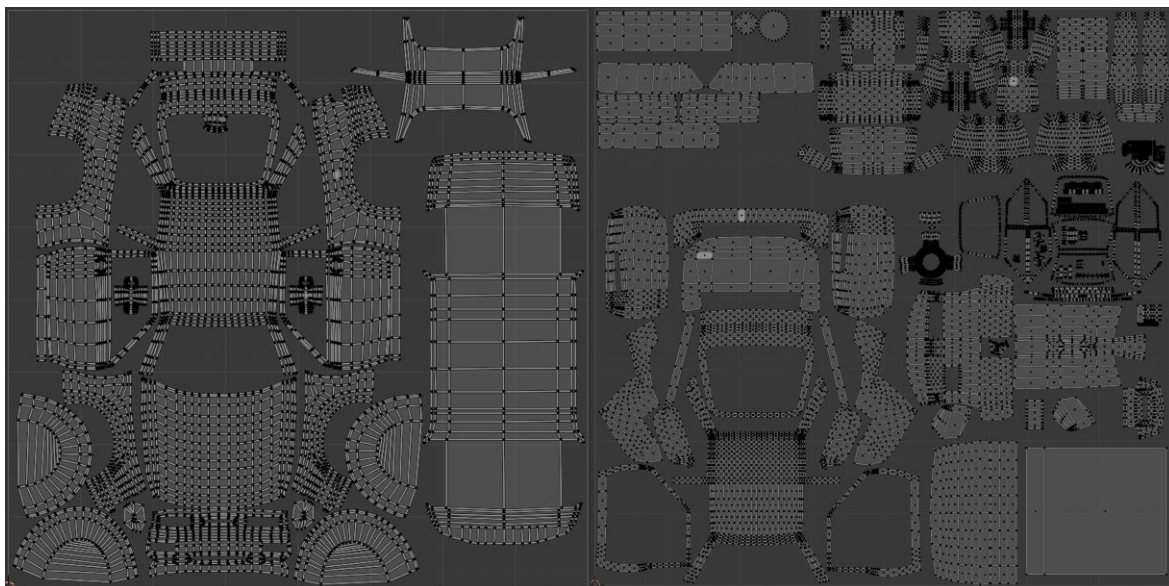
Tarkistetaan että tärkeillä ja näkyvillä alueilla ei synny UV-karttaan epämuodostumia asettamalla mallille ruudukkotekstuuri. Jos venymiä tai litistymiä nähdään pitää saumojen sijaintia muokata tai lisätä saumojen määrää. Kuvassa 23 nähdään, miten ruudukkotekstuuri asettuu suurimmaksi osaksi ilman venymistä mallin pinnalle. Pientä epämuodostumaa teksturiin syntyy lokasuojien alle runkoon, mutta se ei haittaa, sillä se alue tulee jäämään renkaiden taakse piiloon. Kun ollaan tyytyväisiä, miten tekstuurit asettuvat mallin pinnalla voidaan aloittaa UV-kartan muokkaaminen.



Kuva 23. Ruudukkotekstuuri

Blender luo UV-kartan osat määriteltyjen saumojen perusteella. Karttaan syntyvät osat ovat satunnaisessa järjestyksessä. Järjestellään UV-kartan osat uudelleen, jotta tekstuurikarttojen tekeminen on helpompaa. Maalin UV-karttaa varten suurin osa osista asetellaan sen perusteella missä ne sijaitsevat 3D-mallissa. Maalin UV-kartasta voidaan hahmottaa auton oikea ja vasen kylki toisistaan. Auton toinen UV-kartta on monimutkaisempi ja osia ryhmitellään halutun ulkonäön perusteella. Esimerkiksi ikkunoiden tiivisteet pidetään lähellä toisiaan, koska ne ovat mustia ja mattaisia. Tuoleissa osat, jotka ovat samanlaisia kankaaltaan, pidetään samansuuntaisina, ettei kankaan kuviointi vaihda suuntaa. Alueet, jotka tulevat

tarvitsemaan tarkkoja yksityiskohtia skaalataan isommiksi UV-karttaan, jolloin nämä alueet saavat suuremman pikselitiheyden. Suurempi pikselitiheys mahdollistaa selkeiden ja terävien yksityiskohtien saavuttamisen. Esimerkiksi auton mittaristo skaalattiin isommaksi, jotta siihen voidaan kirjoittaa numeroita. Kuvassa 24 on vasemmalla auton maalipintaa varten luotu UV-kartta ja oikealla UV-kartta, joka sisältää kaikki muut auton 3D-mallin pinnat. Tallennetaan valmiit UV-kartat tekstuurikarttojen luomista varten.



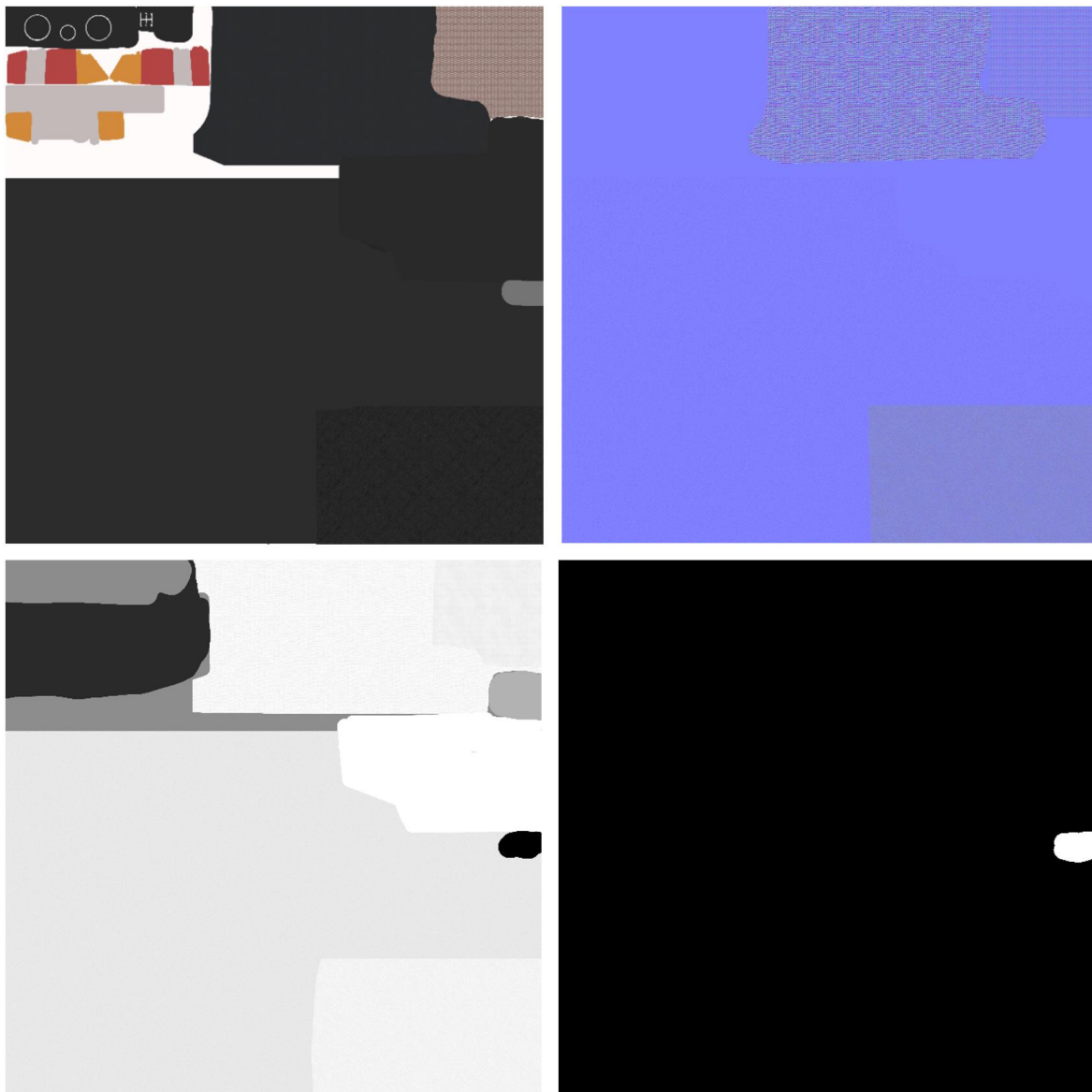
Kuva 24. UV-kartat

Viedään UV-kartat kuvanmuokkausohjelmaan GIMP. Koska autoa tullaan ajamaan sisältäpäin ensimmäisestä persoonasta, pitää tekstuurikarttojen olla tarpeeksi suuria, jotta auton kojelaudan ja muiden nappien merkit ovat selkeitä. Voidaan tehdä tekstuurikartat koossa 4096x4096, ja tarvittaessa myöhemmin pienentää tekstuurikarttojen kokoa. Tekstuurikarttoja on helpompi lähteä pienentämään kuin suurentamaan.

Tekstuurikartat voidaan tehdä osittain piirtämällä ja hyödynnetään valmiita tekstuureita tekijänoikeudet huomioiden. Yleisesti ei haluta käyttää mitään valmista, jonka tekijänoikeudet kieltävät muokkaamisen tai vaativat alkuperäisen tekijän mainitsemista. Tarvittavia tekstuurikarttoja miettiessä otetaan huomioon mitä tekstuurikarttoja Unreal Engine tukee. Mietitään mitä ominaisuuksia ja eroavaisuuksia auton ulkonäkö tarvitsee. Autoon halutaan esimerkiksi kankaisuutta tuoleihin ja keinoahkaisuutta kojelautaan.

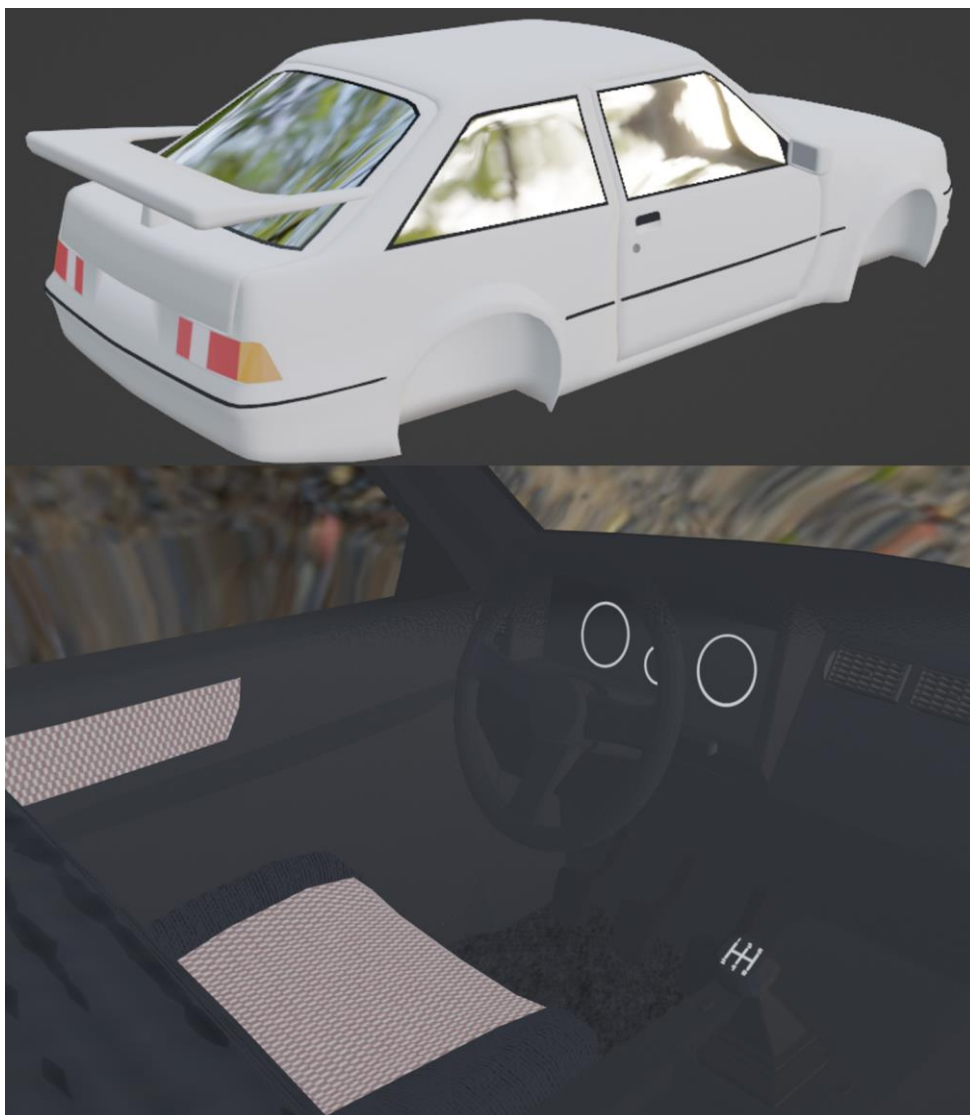
Aloitetaan tekstuurikartat tekemällä värikartta, jotta saadaan pintaan lisättyä halutut kuviot ja väri. Keinoahka ja kankaat eivät ole tasaisia pinnoiltaan, vaan niissä on usein pieniä kohoumia ja syvennyksiä, jotka voidaan saavuttaa normaalikartalla. Kankaan ja tekonahkan pinnan kiiltävyyden välillä on selkeä ero, jota varten tarvitaan karheuskartta. Auton lukkopesiä ja peilejä varten tarvitaan metallisuuskartta. Kuvassa 25 on auton toista UV-karttaa

mukaillen tehty värikartta, normaalikartta, karheuskartta ja kiiltokartta. Auton maalin värin tekstuurikartat tulevat olemaan yksinkertaisia, koska maali tulee olemaan kaikkialta samanlainen väritään, kiilloltaan ja metallisuudeltaan. Koska maalipinta ei sisällä tarkkoja yksityiskohtia, voidaan sen tekstuurikartat tehdä pienemmässä koossa.



Kuva 25. Tekstuurikarttoja

Liitetään tekstuurikartat 3D-malliin ja tarkistetaan että ne asettuvat oikein ja yksityiskohdat tulevat näkyviin. Mallin pinnan näyttäessä rakeiselta tarvitsisi polygonit suuremman alueen UV-kartasta itselleen pikselitiheyden kasvattamiseksi tai suuremman tekstuurikarttojen koon. Kuvassa 26 esitetään miltä tekstuurikartat saivat mallin näyttämään Blenderissä.



Kuva 26. Teksturoitu malli

7.5 Jälkikäsittely

Tarkistetaan että polygonien normaalit osoittavat oikeaan suuntaan, etteivät ne jää läpinäkyviksi pelimoottorin facejen yksipuolisen renderöinnin takia. Katsotaan, että auton skaala ja suuntautuminen on oikein. Blenderissä mallin pivot-piste asetuu 3D-avaruuden alkupisteen perusteella, joten siirretään malli keskelle X- ja Y-akseleita ja nostetaan Z-akselilla rakenne alkupisteen yläpuolelle. Yhdistetään objekteja toisiinsa, jotta 3D-tiedostossa toisistaan irrallaan olevia kappaleita ei ole liikaa. Esimerkiksi kojelautaa varten mallinnetut napit voidaan kaikki yhdistää toisiinsa. 3D-sovelluksissa pystyy myöhemmin uudelleen irrottamaan mallin osia toisistaan, jos on tarve.

Varmistetaan että assetti ja assetin osat on nimetty oikein työnantajan ohjeen mukaan. Selkeys tiedostojen nimissä on tärkeää, koska peli tulee sisältämään todella paljon erilaisia

malleja ja tekstuurikarttoja. Oikea nimeäminen mahdollistaa helpon muokkaamisen jälkikäteen, kun peliassetin osat ovat helposti löydettävissä. Esimerkiksi jos mallin osat olisivat nimeltään kuutio1, kuutio2, kuutio3, eivät ne anna mitään tietoa, mutta kun osat ovat nimeltään ovi, penkki, runko, nähdään mikä osa on mikäkin. Poistetaan tiedostosta turhat objektit ja tekstuurit, joita on voinut syntyä luomisprosessissa, jotta tiedoston koko saadaan minimoitua. Jos jokaisesta työstä säilytetään kaikki mitä on syntynyt luomisprosessien aikana, tarvittavan tallennustilan määrä kasvaa huomattavasti varsinkin isommissa projekteissa, jotka sisältävät tuhansia malleja ja niiden tekstuurikarttoja.

Säilytetään assetti Blenderin omana blend-tiedostomuotona, koska tämän pelin 3D-mallintajista suurin osa käyttää Blenderiä, joten tiedosto on valmiiksi optimoitu Blenderiin, jos siihen tehdään muutoksia. Tallennetaan assetti myös FBX-tiedostona, koska se pystyy säilyttämään tiedon 3D-mallista ja tekstuureista ja se toimii hyvin useiden eri sovelluksien välillä. Assetin tallentaminen neutraalina FBX-muotona varmistaa, että tiedostoa voi tarvittaessa muokata joku toinen ihminen ilman ongelmia, joita voi syntyä, kun yksityisomistuksellista tiedostomuotoa lähdetään siirtämään eri sovellukseen. FBX-tiedostomuodolla voidaan viedä assetti Unreal Engineen ja tarkistetaan mallin ulkonäkö. Pelimoottorissa varmistetaan, että tekstuurit ovat toimivat ja mikään polygoni ei ole läpinäkyvä väärinpäin olevan normaalin suunnan takia.

8 Yhteenveto ja pohdinta

Autoassetti saatiin suunniteltua, mallinnettua ja teksturoitua, mutta paremman lopputuloksen saavuttamiseksi yksityiskohtia pitää mallintaa vielä lisää, kuten radio ja muita kojelaudan nappeja. Näiden asioiden mallintaminen ei kuitenkaan aiheuta UV-kartan kokonaan uudelleentekemistä, koska UV-kartassa on tilaa tarvittavien yksityiskohtien polygoneille. Tarvittavien mallinnettavien osien koot ovat myös pieniä ja ne eivät vaadi tarkkoja yksityiskohtia. Ainoastaan radion näyttö tulee sisältämään tarkempia yksityiskohtia ja vaatii isomman tilan UV-kartasta. Tekstuurikartat vaativat pientä muokkausta, jotta ulkonäöstä saadaan vielä realistisempi ja siistimpi. Aikaisempi kokemattomuus GIMP-sovelluksen käytöstä hankaloitti tekstuurikarttojen tekemistä. Blender oli ennestään tuttu sovellus käyttää.

Toimivan peliasetin luominen on monimutkainen prosessi ja vaatii paljon tietoa ja osaamista. Työvaiheilla on vaikutusta toisiinsa ja videopelien asettamat rajoitteet aseteille on oleellista tietää ennen töiden tekemistä.

Suunnitteluvaiheessa käytetään luovuutta ja tiedonhakua ja hyvä suunnittelu auttaa myöhemmissä vaiheissa, kuten mallintamisessa ja teksturoimisessa. Mallintaessa tavoitellaan hyvää topologiaa, jotta mallin muokattavuus ja animaatioiden tekeminen on mahdollista. Polygonien määrä pyritään pitämään mahdollisimman pienenä, koska liian raskas malli alkaa hidastamaan pelien kuvataajuutta. Useampaa erilaista tekstuurikarttaa hyödyntämällä saadaan mallin pintaan paljon yksityiskohtia ja vaihtelevuutta ulkonäöllisesti. Törmäysobjekti on oleellinen osa peliasettia, koska se estää pelaajaa kävelemästä mallin läpi. Oikea tallennuksen tiedostomuoto mahdollistaa asetin sisältämien tietojen siirtämisen eri sovelluksien välillä.

Jatkotutkimuksia voitaisiin tehdä sovelluksien ominaisuuksista ja työkaluista ja miten asetin luomisprosessia saataisiin nopeutettua. Monimutkaisien asettien luomisen työvaiheet vievät erittäin paljon aikaa, koska objekteja tarvitsee mallintaa paljon, tekstuurikartat sisältävät useampia osia ja tekstuurikarttoihin tarvitsee paljon erilaisia yksityiskohtia. Jatkotutkimuksen mahdollisuus on myös perehtyminen yksityiskohtaisemmin animaatioprosessiin ja miten voidaan saavuttaa realistisia animaatioita.

Lähteet

3D-Ace. 2017. 3D Modeling Techniques in Games. Viitattu 2.5.2023. Saatavissa

<https://3d-ace.com/blog/3d-modeling-techniques-in-games/>

3D-Ace. 2021. 3D Modelling for Video Games: How to Create Beautiful Assets. Viitattu

28.4.2023. Saatavissa <https://3d-ace.com/blog/3d-modeling-for-video-games-how-to-create-beautiful-assets/>

Adobe. 2022. 3D texture editing. Viitattu 11.4.2023. Saatavissa

<https://helpx.adobe.com/photoshop/using/3d-texture-editing.html>

Adobe. a. 3D texturing solution with Adobe Substance 3D. Viitattu 23.3.2023. Saatavissa

<https://www.adobe.com/products/substance3d/discover/3d-texturing.html>

Adobe. b. An Introduction to Video Game Concept Art. Viitattu 26.3.2023. Saatavissa

<https://www.adobe.com/uk/creativecloud/illustration/discover/video-game-concept-art.html>

Akenine-Möller, T., Haines, E., Hoffman, N., Pesce, A., Iwanicki, M. & Hillaire, S. 2018.

Real Time Rendering Fourth Edition. E-kirja. CRC Press. LAB Primo.

All3DP. 2023. Top 10: The Best 3D Modeling Software (Some Are Free). Viitattu

12.5.2023. Saatavissa <https://all3dp.com/1/best-free-3d-modeling-software-3d-cad-3d-design-software/>

Autodesk. a. Autodesk 3ds Max: Create massive worlds and high-quality designs. Viitattu

12.5.2023. Saatavissa <https://www.autodesk.fi/products/3ds-max/overview?term=1-YEAR&tab=subscription>

Autodesk. b. 3D Modeling Software. Viitattu 12.5.2023. Saatavissa

<https://www.autodesk.com/solutions/3d-modeling-software>

Becker, L. 2022. The Many Ways to Get Started with Making and Selling Game Assets.

Viitattu 13.3.2023. Saatavissa <https://thegamehers.com/blog/getting-started-making-digital-game-assets>

Blender. a. Blender is Free Software. Viitattu 12.5.2023. Saatavissa

<https://www.blender.org/about/license/>

Blender. b. The Freedom to Create. Viitattu 12.5.2023. Saatavissa

<https://www.blender.org/about/>

Brito, A. 2009. Blender 3D 2.49 Incredible Machines. E-kirja. Packt Publishing. LAB

Primo.

- Caudron, R. & Nicq, P. 2015. Blender 3D By Example. E-kirja. Packt Publishing. LAB Primo.
- Eng, L. Z. 2015. Building a Game with Unity and Blender. E-kirja. Packt Publishing. LAB Primo.
- Epic Developer Community. Glossary for Unreal Editor for Fortnite. Viitattu 29.4.2023. Saatavissa <https://dev.epicgames.com/documentation/en-us/uefn/unreal-editor-for-fortnite-glossary#collision>
- Fisher, G. 2012. Blender 3D Basics. E-kirja. Packt Publishing. LAB Primo.
- Gahan, A. 2011. 3ds Max Modeling for Games. E-kirja. Elsevier Inc. LAB Primo.
- Huston, S. 2020. Box Colliders Are Simply Magical. Medium. Viitattu 29.4.2023. Saatavissa <https://medium.com/@notaproblem/box-colliders-are-simply-magical-6a146fb3ba49>
- Kelly, R. & Chakravorty, D. 2022. The Top 10 3D File Formats of 2022. All3DP. Viitattu 9.5.2023. Saatavissa <https://all3dp.com/2/most-common-3d-file-formats-model/>
- Kramarzewski, A & De Nucci, E. 2018. Practical Game Design. E-kirja. Packt Publishing. LAB Primo.
- Lampel, J. 2021. 8 Tips for Clean Topology in Blender (Updated for 2021). CG Cookie. Viitattu 1.2.2023. Saatavissa <https://cgcookie.com/posts/guide-to-clean-topology>
- Lanier, L. 2015. Advanced Maya Texturing and Lightning. E-kirja. John Wiley & Sons. LAB Primo.
- McKinley, M. 2005. The Game Artist's Guide to Maya. E-Kirja. John Wiley & Sons. LAB Primo.
- Mitchell, B. L. 2012. Game Design Essentials. E-kirja. John Wiley & Sons. LAB Primo.
- Plowman, J. 2016. 3D Game Design with Unreal Engine 4 and Blender. E-kirja. Packt Publishing. LAB Primo.
- Pluralsight. 2022. Differences between Displacement, Bump and Normal Maps. Viitattu 2.5.2023. Saatavissa <https://www.pluralsight.com/blog/film-games/bump-normal-and-displacement-maps>
- Pluralsight. 2014a. What's the Difference? A Comparison of Modeling for Games and Modeling for Movies. Viitattu 23.1.2023. Saatavissa <https://www.pluralsight.com/blog/film->

games/whats-the-difference-a-comparison-of-modeling-for-games-and-modeling-for-movies

Pluralsight. 2014b. Why Are Ngons and Triangles so Bad? Viitattu 23.3.2023. Saatavissa <https://www.pluralsight.com/blog/film-games/ngons-triangles-bad>

Price, A. 2023. What are the different texture maps for? Poliigon. Viitattu 25.4.2023. Saatavissa <https://help.poliigon.com/en/articles/1712652-what-are-the-different-texture-maps-for>

Rafiee, A. What Is a Game Ready Asset and How to Have One: A Comprehensive Guide. Dream Farm Studios. Viitattu 13.3.2023. Saatavissa <https://dreamfarmstudios.com/blog/game-ready-asset/>

Roy, S. 2023. 3D Modeling: An Overview of Various Techniques. ThePro3DStudio. Viitattu 28.4.2023. Saatavissa <https://professional3dservices.com/blog/3d-modeling-techniques.html>

Selin, E. 10 Different types of 3D modeling techniques. Artistic Render. Viitattu 24.3.2023. Saatavissa <https://artisticrender.com/10-different-types-of-3d-modeling-techniques/>

Simonds, B. 2013. Blender Master Class. E-kirja. William Pollock. LAB Primo.

Slick, J. 2020. The Definition of Topology and Its Purpose in 3D Animation. Lifewire. Viitattu 3.2.2023. Saatavissa <https://www.lifewire.com/topology-in-3d-animation-2181>

Slick, J. 2019. Surfacing 101: The Basics of Texture Mapping. Lifewire. Viitattu 10.3.2023. Saatavissa <https://www.lifewire.com/texture-mapping-1956>

Thorn, A. 2017. How to Cheat in Blender 2.7x. E-kirja. Taylor & Francis Group. LAB Primo.

Totten, C. 2012. Game Character Creation with Blender and Unity. E-kirja. Indianapolis: John Wiley & Sons. LAB Primo.

Unity Documentation. 2023. Introduction to collision. Viitattu 29.4.2023. Saatavissa <https://docs.unity3d.com/Manual/CollidersOverview.html>

Valve Developer Community. 2022. Collision Mesh. Viitattu 29.4.2023. Saatavissa https://developer.valvesoftware.com/wiki/Collision_mesh

Wacom. Improve your 3D workflow with these simple tips. Viitattu 31.1.2023. Saatavissa <https://www.wacom.com/en-us/discover/3d-game/improve-your-3d-workflow>

Zephyris. 2008. File:Cube Representative UV Unwrapping.png. Wikimedia Commons.

Viitattu 7.4.2023. Saatavissa <https://commons.wikimedia.org/w/index.php?curid=7202834>