



Ilari Siltala

# 3D-assetien tuottaminen Quest VR-peliin

Visuaalisen laadun ja optimoinnin tasapainottelu  
VR-peliasseteissa

Metropolia Ammattikorkeakoulu

Muotoilija

3D-animointi ja -visualisointi

Opinnäytetyö

28.4.2025

## Tiivistelmä

Tekijä(t):	Ilari Siltala
Otsikko:	3D-asettien tuottaminen Quest VR-peliin
Sivumäärä:	33 sivua
Aika:	28.4.2024
Tutkinto:	Muotoilija
Tutkinto-ohjelma:	Muotoilun tutkinto-ohjelma
Pääaine:	3D-animointi ja -visualisointi
Ohjaaja(t):	Lehtori Kristian Simolin

---

VR-pelin optimointi on haastavaa, etenkin kun on kyseessä mobiililaitte, kuten Meta Quest 3. Opinnäytetyön tavoitteena on käydä läpi tekniikoita, jotka auttavat 3D-artistia toteuttamaan visionsa laitteen suorituskyvystä riippumatta. Kerron VR-peligrafiikan haasteista ja erityisesti niistä teknisistä esteistä, joita on Quest 3-laitteelle kehittäessä haastavaa ylittää.

Pääpainona on tiivistää 3D-artistille olennainen tieto, jota minulle kertyi tätä peliä tehdessä, ja kertoa, miten hyödynsin sitä peliä varten tekemässäni 3D-grafiikassa. Käytän tekemiäni asetteja esimerkkeinä eri optimointiasioissa, mutta kerro myös tiettyjen asetien luomisprosesseista yksitellen.

Pohdin työssä, miten nykyajan tietokone ja -konsolipelien grafiikat ovat kehittyneet, ja miksi tietyt uudemmat kehitykset - kuten deferred renderöinti - ovat laajalti epäyhteensopivia VR-pelien kanssa. Käyn läpi lyhyesti, miten nykyiset 3D-sisällöntuotantotyökalut tekevät optimoinnista helpompaa kuin aikaisemmin.

Työn loppupuolen käytännön osuudessa esittelen erilaisten asetien työvaiheita ja selitän, mikä niissä oli haastavaa taiteellisesti ja teknisesti.

Asiasanat: 3D-mallinnus, teksturointi, optimointi, suunnittelu, virtuaalitodellisuus

---

Opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

## Abstract

Author(s): Ilari Siltala  
Title: Creating 3D assets for a Quest VR-game  
Number of Pages: 33 pages  
Date: 28 April 2024

Degree: Bachelor of Culture and Arts  
Degree Programme: Design  
Major: 3D Animation and Visualisation  
Instructor(s): Kristian Simolin, Senior lecturer

---

Optimizing a VR game is challenging, especially when it involves a mobile device like the Meta Quest 3.

The goal of this thesis is to explore techniques that help a 3D artist realize their vision, regardless of the device's performance limitations. I discuss the challenges of VR game graphics, focusing particularly on the technical obstacles that are difficult to overcome when developing for the Quest 3.

The primary aim is to condense the essential information I gathered during the game development process for other 3D artists and explain how I applied that knowledge to the 3D assets I created. I use a variety of assets as examples to illustrate different optimization techniques and detail the creation processes of specific assets, covering both the artistic and technical aspects.

I reflect on how graphics in modern PC and console games have evolved, and why certain recent developments—such as deferred rendering—are largely incompatible with VR games. I also briefly review how current 3D content creation tools have made optimization more accessible than before.

In the practical section of the thesis, I elaborate on the workflows of various assets and explain the artistic and technical challenges encountered during their creation.

Keywords: 3D-modeling, texturing, optimization, design, virtual reality

---

This thesis has been checked using Turnitin Originality Check service.

## Sisällys

1	Johdanto	1
2	Keskeisiä asioita VR-peliasettien optimoinnissa	2
2.1	Sanasto	2
2.2	Haasteita 3D-asettien luomisessa Quest-laitteelle	3
2.2.1	Eriyiset haasteet realismissa	4
2.2.2	Renderöintimenetelmät	5
2.2.3	Deferred- ja forward renderöinti	5
2.2.4	Forward-renderöinnin visuaalinen etu	6
2.2.5	Pelin renderöintimenetelmä	6
2.2.6	Nykyajan työkalut tekevät optimoinnista nopeaa	8
3	Optimoinnin työkalut	9
3.1.1	Leivottu valaistus ja valokartta UV-kartat	9
3.1.2	Harkittu UV-kartoitus	10
3.2.1	Polygonien käyttö leivottujen normaalikarttojen kanssa	12
3.2.2	Polygonien käyttö riippumattomana normaalikartoista	14
3.3.1	Tekstuurimaskit	16
3.3.2	Verteksivärimaskit	19
3.4.1	Toistuvat tekstuurit	20
3.4.2	Trim-tekstuurit	21
4	Assetit	21
4.1	Animoitu proppi: Robot Arm	22
4.2	Hahmo: Mecha	25
4.3	Ympäristö: Laboratorio	28
4.4	Interaktiivinen objekti: Hackpanel	30
5	Yhteenveto	31
	Lähteet	32
	Kuvalähteet	33

# 1 Johdanto

Tässä opinnäytetyössä keskityn virtuaalitodellisuuspeleihin optimoitujen 3D-asettien luomiseen. Projektina on Meta Quest 3- ja 2-laitteille kehitetty peli, joka on toteutettu Unreal Engine 5 -pelimoottorilla. Pelin kehittäjänä toimii Vaki Games, ja Meta Platforms tukee projektia teknisen tuen ja testauksen kautta. 3D-artistina olen tuottanut peliin erilaista 3D-sisältöä tekniset vaatimukset huomioiden.

Työskentelyni on sisältänyt erilaisten proppien, hahmojen ja ympäristöjen luomista. Ympäristöjä suunnitellessa olen perehtynyt myös valaistukseen, mikä on mobiililaitteille suunnatuissa peleissä erityisen haastavaa teknisten rajoitteiden vuoksi. Tässä opinnäytetyössä keskityn kuitenkin ensisijaisesti 3D-asettien luomiseen, vaikka valaistuksella ja siihen liittyvillä teknisillä tekijöillä, kuten valokartta UV-kartoilla, onkin tärkeä rooli mallien esittämisessä pelimaailmassa.

Tämä projekti yhdistää kolme minulle mielenkiintoista osa-aluetta: virtuaalitodellisuuden, mobiililaitteet ja optimoinnin. Mobiilipelikehitys on ollut pelialalla tuttu haaste jo pitkään, ja se vaatii jatkuvasti graafisia kompromisseja – nämä samat haasteet pätevät myös Quest-peleihin.

Vaikka Questille kehitetty VR-peli tarjoaa paljon saman kokemuksen kuin muutkin VR-pelit, laitteen suorituskyky asettaa sille merkittäviä rajoituksia. Modernien tietokoneiden ja pelikonsolien teho on kasvanut niin suureksi, että optimoinnin merkitys on osittain vähentynyt. VR-peleissä optimointi on kuitenkin yhä keskeisessä roolissa, ja yhdistettäessä VR mobiililaitteisiin haasteet kasvavat entisestään. Quest-pelien kehityksessä onkin tärkeää hyödyntää leivottua valaistusta, minimoituja polygonimääriä sekä harkittua teksteliheyttä suorituskyvyn varmistamiseksi. VR-peli Questilla on kuitenkin muilla tavoin sama kuin mikä tahansa VR-peli. Siinä on samankaltaiset ohjaimet kuin muissa VR-laseissa, ja kokemus on paljon sama. Ainoastaan Quest-laitteen suorituskyky on rajana.

## 2 Keskeisiä asioita VR-peliasettien optimoinnissa

Käyn tässä luvussa läpi optimointiasioita, jotka ovat tulleet itselleni vastaan tätä peliä tehdessä. Luvussa keskitytään erityisesti 3D-asettien optimointihaasteisiin mobiililaitteille suunnatussa VR-pelissä, sekä teknisiin ja visuaalisiin ratkaisuihin, joilla pyritään säilyttämään korkea suorituskyky ilman suuria kompromisseja grafiikan laadussa.

### 2.1 Sanasto

**Verteksi:** 3D-mallin piste, joka määrittää mallin muodon.

**Polygoni:** 3D-mallin pinta, joka koostuu kolmesta tai useammasta verteksistä.

**UV-kartta:** 2D-koordinaatisto, johon 3D-mallin pinta projisoidaan teksturointia varten.

**Valokartta UV-kartta:** Erillinen UV-kartta, jota käytetään valon ja varjojen leipomiseen mallin pintaan.

**Reunanpehmennys:** Tekniikka, jolla mallin reunat saadaan pehmenettyä.

**Optimointi:** Prosessi, jossa 3D-sisältöä muokataan toimimaan paremmin suorituskyvyn kannalta.

**UV-saareke:** UV-kartalla oleva erillinen osa mallin pinnasta, joka ei ole yhteydessä muihin osiin.

**Normaalikartta:** Tekstuuri, joka luo illuusion yksityiskohdista muuttamatta mallin geometriaa.

**Verteksiväri:** Mallin verteksiin tallennettu väritieto.

## 2.2 Haasteita 3D-asettien luomisessa Quest-laitteelle

Koska Quest 3 on mobiililaitte, sen suorituskyky ei vastaa tietokoneiden tai pelikonsolien tehoa (Justus-Liebig-Universität Gießen i.a.). Siksi mallien polygonimäärien ja tekstuuriresoluutioiden on pysyttävä alhaisina, jotta peli toimii sujuvasti. Virtuaalitodellisuuspeleissä ruudunpäivitysnopeus (FPS) on kriittisempi tekijä kuin missään muussa pelityypissä.

Tietokone- ja konsolipeleissä ruudunpäivitysnopeus voi pudota jopa 30 FPS:ään, mikä heikentää lähinnä tähtäämistä ja ohjauksen tarkkuutta. Joissakin peleissä alhainen ruudunpäivitysnopeus voi jopa luoda elokuvamaisen visuaalisen tyylin, sillä 24 FPS on yleinen elokuvateollisuudessa käytetty standardi. VR-peleissä tällainen efekti ei kuitenkaan toimi.

Virtuaalitodellisuuspeleissä on aina otettava huomioon se, että viivästynyt kuva voi aiheuttaa käyttäjälle pahoinvointia, koska aivot odottavat pään ja käsien liikkeiden olevan yhtä sulavia kuin todellisuudessa (Häkkinen, 2023). Käytännössä ruudunpäivitysnopeuden putoaminen alle 60 FPS:n alkaa jo aiheuttamaan pelaajalle pahoinvointia. Siksi vakaa ja riittävän korkea ruudunpäivitysnopeus on ehdoton prioriteetti VR-peleissä – mitä suurempi, sen parempi. Meta edellytti tähän peliin minimissään 72 FPS:n kauttaaltaan, joka tuntuu miellyttävältä oman pelitestauksen perusteella.

Vaatus korkeaan ruudunpäivitysnopeuteen tekee VR-pelien optimoinnista haastavaa, mutta erityisesti se korostuu Quest -peleissä, joiden suorituskyky vastaa lähinnä tehokkaita älypuhelimia. Siksi resursseja on säästettävä kaikilla mahdollisilla keinoilla.

VR-peleissä optimointia hankaloittaa myös se, että objekteja katsotaan usein hyvin läheltä. Ihanteellisesti asetien tekselitiheyden tulisi olla riittävän korkea, jotta ne näyttävät tarkoilta lähietäisyydeltä. Lisäksi 3D-mallien tulisi sisältää mahdollisimman paljon yksityiskohtia, sillä normaalikartoilla ei voida luoda VR-ympäristössä yhtä vakuuttavaa illuusiota kolmiulotteisuudesta kuin tavallisissa

peleissä. Käytännössä polygoneja on kuitenkin usein pakko vähentää ja normaalikarttoja hyödyntää, etenkin Quest-peleissä.

### 2.2.1 Erityiset haasteet realismissa

Ongelma moninkertaistuu, jos pelin grafiikalla pyritään realismiin. Monet tämän pelin 3D-asseteista tavoittelevat muotoilultaan realismia, vaikka peli sijoittuukin tulevaisuuden cyberpunk-tieteisfiktiomaailmaan. Tämä cyberpunk teema antaa paikoin enemmän vapautta muotoilukompromisseihin ja taiteellisiin ratkaisuihin.

Jokainen videopeli on tarkoituksenmukaisesti laadittu hanke, oli se sitten indie-peli tai AAA-peli. Mikään ei ole vahingollista pelin grafiikassa, etenkin graafisessa suuntauksessa (Hickman Design, i.a.).

Peligrafiikka on yleisesti ottaen täynnä kompromisseja. Tekniset rajoitteet ovat pakottaneet pelinkehittäjät yksinkertaistamaan tiettyjä vaikeasti toteutettavia elementtejä, kuten hiuksia tai kasvustoa. Jokaisen haituvan mallintamisen sijaan käytetään teksturoituja polygoniverkkoja, oli sitten kyseessä ruohikko tai hahmon hiukset. Vaikka nykypäivänä tämä lähestymistapa on korvattu osittain tai kokonaan aidonkaltaisuutta tuottavilla menetelmillä, sama periaate pätee edelleen. Yksityiskohtaisten asioiden edustaminen vähillä yksityiskohdilla on osa impressionistista taidesuuntausta, joka on periaatteiltaan samankaltaista kuin peligrafiikan tuottaminen.

Lähempänä nykypäivää, impressionistinen lähestymistapa on alkanut korvautumaan fotorealismilla. Vanhemmissa peleissä fotorealismiin ei kuitenkaan ollut varaa laitteiden tehojen puutteiden takia, joten jopa realismia tavoittelevat pelit esittivät monimutkaisia asioita yksinkertaistettuina, keskittyen suuriin muotoihin, valaistukseen ja väreihin. Artistien taitavan visuaalisen suunnittelun ansiosta monet vanhatkin pelit, kuten vuonna 2004 julkaistu Half-Life 2, näyttävät siitä huolimatta edelleen vaikuttavilta.

## 2.2.2 Renderöintimenetelmät

Kehitettäessä VR-peliä on tehtävä visuaalisia kompromisseja suorituskyvyn taakamiseksi. VR-laitteella jokainen ruutu on renderöitävä kahdesti – kerran kummallekin silmälle. Tämän vuoksi piirtokutsujen määrää, tekstuurikokoja, tekstuurien määrää sekä polygoni- ja verteksimääriä on pyrittävä pitämään mahdollisimman alhaisina. Sulava pelikokemus on ensisijainen tavoite, sillä se varmistaa, ettei pelaajan eläytyminen rikkoudu.

Eläytymistä tukee myös selkeä ja häiriötön kuva, jossa ei esiinny visuaalisia häiriöitä. Tämä edellyttää korkeaa resoluutiota ja tehokasta reunanpehmennystä, jotka puolestaan kuluttavat laitteen suorituskykyä.

VR-peleissä käytetään pääasiassa kahta renderöintimenetelmää, jotka ovat olleet osa peliteknologiaa jo vuosien ajan: **Forward renderöinti** ja **deferred renderöinti**. Näistä forward renderöinti on vanhempi ja kevyempi vaihtoehto, kun taas deferred rendering on uudempi ja nykyisin yleisemmin käytössä peliteollisuudessa.

## 2.2.3 Deferred- ja forward renderöinti

Forward- ja deferred renderöinti ovat molemmat vaihtoehtoja Quest-pelin kehityksessä, mutta deferred renderöintiä ei suositella laitteen mobiilikoneiston rajoitusten vuoksi. Sitä on kokeiltu joissakin VR-peleissä, mutta se on todettu epäkäytännölliseksi useimmissa käyttötarkoituksissa (Dasch, 2023).

Forward renderöinnissä jokainen objekti käsitellään yksitellen renderöinnin aikana. Objektien ulkonäkö määräytyy valaistuksen ja materiaalien perusteella jo renderöintipassin aikana. Deferred renderöinti sen sijaan mahdollistaa aidomman näköisten grafiikoiden toteuttamisen käyttäen eri menetelmää: Se ei laske objektien valaistusta yksitellen, vaan tallentaa tarvittavat tiedot G-bufferiin ja käsittelee valaistuksen erillisessä renderöintipassissa. Tämä mahdollistaa monimutkaisempien valaistusskenaarioiden toteutuksen verrattuna forward renderöintiin, mutta vaatii enemmän laskentatehoa.

Koska forward renderöinti on yksinkertainen ja kevyin renderöintimenetelmä, se on ollut suosituin vaihtoehto VR-pelien kehityksessä. Quest-laitteilla sitä suositellaan vahvasti – ellei jopa vaadita – suorituskyvyn takaamiseksi. Tulevaisuudessa tämä voi muuttua laitteistojen kehittyessä, mutta Quest 2- ja 3-laitteilla forward renderöinti on edelleen ensisijainen valinta.

#### 2.2.4 Forward-renderöinnin visuaalinen etu

Forward renderöinnin etuna on sen yhteensopivuus multisample anti-aliasing -reunanpehmenysmenetelmän (MSAA) kanssa. MSAA mahdollistaa terävän kuvan ilman visuaalisia artefakteja, mikä on erityisen tärkeää VR-ympäristössä. Deferred renderöinnissä tätä mahdollisuutta ei ole, sillä MSAA ei kykene pehmentämään G-bufferiin tallennetun valaistuksen reunoja.

Deferred renderöinnissä reunanpehmenys on toteutettava post-prosessointimenetelmillä, kuten TAA (temporal anti-aliasing). TAA ja muut post-prosessointiin perustuvat reunanpehmenystekniikat on kuitenkin todettu heikoiksi VR-käytössä, sillä ne voivat aiheuttaa epäterävyyttä ja häiritseviä visuaalisia artefakteja, jotka heikentävät immersiota.

#### 2.2.5 Pelin renderöintimenetelmä

Peliin valikoitui forward renderöinti sen suorituskykyetujen vuoksi. Unreal Engine 5 tukee tätä renderöintimenetelmää hyvin, ja sen editorissa on kattavat työkalut pelinkehitykseen. Forward renderöinti on riittävän kevyt Quest 2- ja 3 VR-laseille, ja olemme pystyneet saavuttamaan sillä peliin sopivan graafisen laadun.

Lisäksi käytössä on täysin staattinen, leivottu valaistus. Tämä on erittäin tehokas tapa toteuttaa valaistus VR-ympäristössä, sillä reaaliaikainen dynaamisten varjojen laskeminen on suorituskykyä kuormittavaa. Ambient occlusion -efekti voisi parantaa varjostuksen laatua ilman dynaamisia valoja, mutta jopa kevyin vaihtoehto, Screen-Space Ambient Occlusion (SSAO), todettiin liian raskaaksi

tähän peliin. Tästä syystä pelissä on pitäydytty pelkästään leivotussa valaistuksessa, jota tukevat cubemap-heijastukset. Tämä yhdistelmä tuo peliin sopivaa realismin tuntua, ilman että suorituskyky kärsii kohtuuttomasti. Kuvassa 1 näkyy leivottu valaistus pelimaailmassa.



*Kuva 1. Esimerkki pelialueesta, jonka teemana on romuttamo. (Siltala 2025).*

Tekniset kompromissit tuovat mukanaan myös taiteellisia rajoituksia, mutta tämä on yleinen haaste kaikissa Quest-peleissä. Mobiililaitteiden suorituskyky asettaa omat rajoituksensa, minkä vuoksi niille kehitetyt pelit muistuttavat usein vanhempia pelejä tekniseltä toteutukseltaan. Tämä ei kuitenkaan tarkoita, että lopputulos näyttäisi huonolta – monet vanhat pelit ovat yllättävän visuaalisesti vakuuttavia vielä tänäkin päivänä. Niiden kehittäjät osasivat hyödyntää saata-villa olevat työkalut ja resurssit tehokkaasti, optimoida pelit kohdealustoille ja luoda vaikuttavan visuaalisen ilmeen taitavalla mallinnuksella ja teksturoinnilla. Samojen periaatteiden soveltaminen on edelleen tärkeää ja voi auttaa saavutta-maan laadukkaan lopputuloksen, vaikka tekniset rajoitukset asettavatkin omat haasteensa.

Vaikka nämä optimointimenetelmät eivät ole kadonneet mihinkään, peliteolli-suuden käytännöt muuttuvat jatkuvasti laitteiston kehittyessä. Leivottu valaistus

oli aiemmin lähes itsestäänselvyys kaikille pelialalla työskenteleville 3D-artisteille, kunnes dynaaminen valaistus saavutti tason, jolla se mahdollisti aidonkaltaisen valaistuksen myös kuluttajatason tietokoneilla ja konsoleilla.

### 2.2.6 Nykyajan työkalut tekevät optimoinnista nopeaa

Kun peligrafiikan standardit kehittyvät ja uudet 3D-artistit omaksuvat moderneja työskentelytapoja, vanhojen grafiikkateknologioiden hyödyntäminen ei ole enää itsestään selvää. Nykyisin dynaaminen valaistus ja korkeat polygonimäärät ovat vallitsevia käytäntöjä, ja monet optimointiin liittyvät periaatteet ovat jääneet vähemmälle huomiolle. Kuitenkin silloin, kun resurssit ovat rajalliset—kuten mobiili-VR-ympäristöissä—on välttämätöntä palata kevyempiin ratkaisuihin.

Vanhempien teknisten menetelmien hyödyntäminen ei kuitenkaan tarkoita huonompaa visuaalista lopputulosta, vaan päinvastoin voi johtaa optimoituun ja harkittuun grafiikkaan. Aiemmin kehitetyt tekniikat, kuten valojen leipominen ja UV-tilan tehokas käyttö, mahdollistavat edelleen laadukkaan visuaalisen ilmeen, kun niitä sovelletaan oikein. Näiden taitojen hallinta vaatii usein oman aikansa, sillä tietoa ei aina löydy helposti. Tiettyjä menetelmiä joudutaan etsimään vanhoista tutoriaaleista, analysoimalla aikakauden pelejä tai takaisinmallintamalla niiden ratkaisuja.

Toisaalta modernit työkalut voivat nopeuttaa klassisemman 3D-peligrafiikan työvaiheita. Esimerkiksi valokarttojen UV-karttojen ja LOD-mallien luominen, jotka aiemmin olivat täysin käsityötä, voidaan nyt osittain automatisoida. Unreal Engine 5 tarjoaa työkalut näiden tehtävien nopeuttamiseen, vaikka manuaalinen hienosäätö on edelleen usein tarpeen parhaan lopputuloksen saavuttamiseksi.

Tämän pelin kehityksessä kaikki edellä mainitut optimointikeinot ovat olleet välttämättömiä. Koska peli käyttää leivottua valaistusta, valokartta UV-kartat ovat olennainen osa lähes jokaisen mallin tuotantoa, hahmoja lukuun ottamatta. Polygonimäärät on pidettävä kurissa, mikä edellyttää huolellista mallintamista ja

UV-kartoitusta. Lisäksi mallien suunnittelussa hyödynnetään rakenteellisia toistoja, mikä mahdollistaa UV-saarekkeiden pinoamisen ja säästää arvokasta tekstuuritilaa.

Tämä lähestymistapa ei ole vain tekninen pakko, vaan myös tietoinen valinta luoda tehokkaasti toimivaa, mutta samalla visuaalisesti uskottavaa peligrafiikkaa rajoitetuilla resursseilla.

### 3 Optimoinnin työkalut

Kerron tässä luvussa työskentelymenetelmistä, jotka helpottavat optimointia ja helpottavat asettien toteuttamista hyvällä visuaalisella laadulla. On olemassa lukuisia menetelmiä optimoida asetteja VR-peliin, mutta käyn läpi tärkeimmät niistä, joita käytin tätä peliä tehdessä.

#### 3.1.1 Leivottu valaistus ja valokartta UV-kartat

Pelin jokaiselle staattiselle objektille luodaan valokartta UV-kartat. Kaikissa mal-leissa on perus-UV-kanava, UV0, joka vastaa tekstuurien, kuten perusvärin ja normaalikartan, sijoittamisesta mallin pinnalle. Tämän lisäksi on UV1-kanava, joka Unreal Engine 5:ssä on oletuksena varattu valokartoille. UV-kanavia voidaan käyttää monenlaisiin tarkoituksiin, kuten maskaamiseen, mutta tässä vaiheessa keskityn erityisesti UV1-kanavan käyttöön valokartoituksessa.

Valokartta UV-kartat eroavat usein tavallisista UV-kartoista, ja niiden tulisi usein olla erilaiset, jotta valokartta näyttää parhaimmalta. Koska valokartta-UV kartoittaa valaistuksen mallin polygoneihin, sen UV-saarekkeiden ei tulisi koskaan olla päällekkäisiä. Saumoja tulisi olla mahdollisimman vähän, sillä alhaisemmalla resoluutiolla valokartan UV-saarekkeiden väliin voi ilmetä vuotoa. Tämä vuoto tarkoittaa, että varjostetun UV-saarekkeen osat voivat peittää valaistusta toisesta UV-saarekkeesta, mikä heikentää valokartan visuaalista laatua. Vältettävissä olevien vuotojen minimoimiseksi on tärkeää vähentää saumoja ja varmistaa, että UV-saarekkeiden väliin jää riittävästi tyhjää tilaa pikselitasolla (Epic Games,

2023). Kuvassa 2 on esimerkki manuaalisesti tehdystä UV-kartasta jota käytetään pelkästään valokartoille.



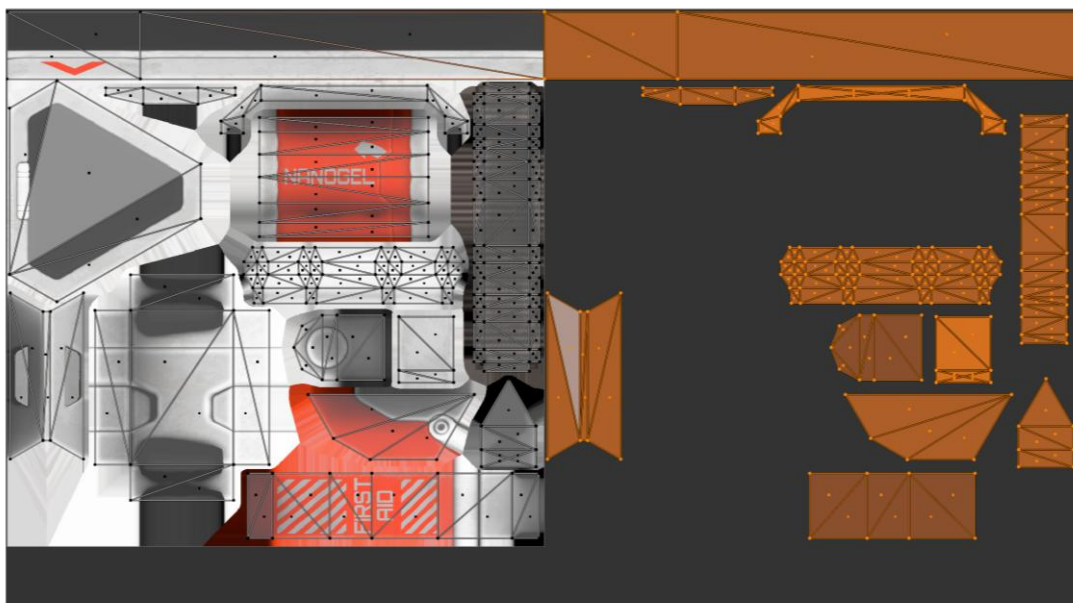
*Kuva 2. 3D-malli ja sen valokartta UV-kartta. (Siltala 2025).*

### 3.1.2 Harkittu UV-kartoitus

Tekstuuriresoluutiot ovat yleensä alhaisia Quest 3 -peleissä, sillä laite ei tue kovin graafisesti raskaita pelejä. Tässä pelissä useimmissa 3D-aseteissa on maksimissaan 1024x1024 pikselin tekstuurit. Pienemmissä objekteissa on vielä pienemmät tekstuuriresoluutiot, ja joissakin tärkeissä ja suurissa objekteissa käytetään 2048x2048 pikselin tekstureja.

Meta suosittelee Quest-peleissä käytettävän mieluummin pienempää määrää suurempia tekstureja monen pienen tekstuurin sijaan. Tämä suositus voidaan toteuttaa muun muassa pakkaamalla tekstureja atlaksiin, käyttämällä trim-tekstureja ja hyödyntämällä saumattomia tekstureja ympäristöjen teksturoimisessa. Erityisesti atlasoinnista on hyötyä, kun kentässä on useita pieniä objekteja lähekkäin. Jos nämä objektit jakavat saman tekstuuriatlasten ja materiaalin, ne voidaan yhdistää, jolloin säästetään piirtokutsuja. Piirtokutsujen minimointi on tärkeää suorituskyvyn kannalta.

Omissa 3D-aseteissani pidän 1024x1024 pikselin resoluutiota lähes absoluuttisena rajana, mutta tietyissä tapauksissa suurempikin tekstuurikoko saattaa olla välttämätön. Tärkeintä on kuitenkin riittävä tekselitiheys. Usein huomaan, että suuremman mallin yksityiskohtaisuus on riittävä 1024x1024 pikselin tekstuureilla, kunhan varaan riittävästi tekselitiheyttä erityisesti pelaajan kameran lähellä oleviin kohtiin. Yksi keino optimoida tätä on UV-saarekkeiden pinoaminen päällekkäin. Tämä onnistuu, kun suunnittelen mallin symmetrisesti ja käytän toistuvia elementtejä. Esimerkiksi auton pyörien UV-saarekkeet voivat olla pinottuina päällekkäin, sillä niiden muodot ovat identtiset. Toistuvat elementit ja pinotut UV-saarekkeet näkyvät kuvassa 3 esitetyssä assetissa.



Kuva 3. 3D-malli, jossa on pinottuja UV-saarekkeita. (Siltala 2025).

Suorin tapa hyödyntää rajallista UV-tilaa on asettaa eri UV-saarekkeille eri tekselitiheydet. Esimerkiksi ensimmäisen persoonan ammuskelupeliin suunnattua tuliassetta UV-kartoittaessa on helppo päätellä, mihin osiin mallista kannattaa varata eniten yksityiskohtia. Jos kyseessä on pistooli, luistin ja erityisesti sen takaosan UV-saarekkeisiin kannattaa asettaa korkea tekselitiheys, kuten 2048 px/m, koska ne ovat lähellä pelaajan kameraa. Vähemmän näkyville osille, kuten lippaaseen, voi puolestaan valita esimerkiksi 1024 px/m tai 512 px/m tekselitiheyden.

Tätä kannattaa alkaa miettiä jo high- ja low poly -malleja tehtäessä. High poly -mallin yksityiskohdat saattavat jäädä liian pieniksi erottuakseen normaalikartasta, jos niiden UV-saarekkeissa on alhaiset tekselitiheydet. Low poly -mallia suunniteltaessa on myös hyvä miettiä, mihin saumat sijoitetaan. Jos sauma menee kahden eri tekselitiheyden rajalle, ero saattaa olla näkyvä. On suositeltavaa välttää liian suuria hyppyjä tekselitiheyksissä yhden mallin sisällä. Jos UV-tilan puute kuitenkin vaatii kompromisseja, alhaisimmat tekselitiheydet kannattaa varata UV-saarekkeisiin, jotka eivät ole helposti näkyvissä.

Ei ole aina välttämätöntä pyrkiä esimerkkitekstuuriin, kuten 2048 px/m. Tiheys voi vaihdella pelin ja kohdealustan mukaan, ja jos itse tekstuuri on pieni, tekselitiheys rajoittuu sen mukaan. Nykyään monissa peleissä käytetään hyvin korkeita tekselitiheyksiä, koska tietokoneiden ja konsolien suorituskyvyt sekä näyttöjen resoluutiot ovat parantuneet. Vaikka tekselitiheys ja tekstuuriresoluutio eivät ole sama asia, ne kulkevat käsi kädessä. Jos kohdealustan tekniset rajoitteet pitävät tekstuurien resoluutiot alhaisina, myös tekselitiheydet kärsivät, ja tämä on tullut minulla vastaan myös tätä peliä kehittäessä.

### 3.2.1 Polygonien käyttö leivottujen normaalikarttojen kanssa

Tyypillisesti peleissä käytetään low poly -malleissa leivottuja normaalikarttoja yksityiskohtien elävöittämiseen. VR-peleissä normaalikartat eivät kuitenkaan ole yhtä tehokkaita kolmiulotteisen geometrian illuusion luomisessa. Tavallisessa

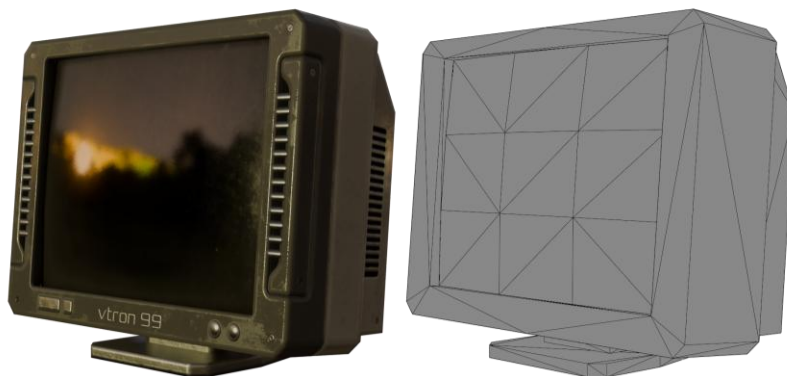
pelissä normaalikartta voi tuoda jopa pelkistettyyn polygoniverkkoon yksityiskohtia, jotka saavat assetin näyttämään paljon monimutkaisemmalta kuin se oikeasti on. Sama illuusio ei kuitenkaan toimi yhtä hyvin VR-peleissä, sillä VR-laseissa on stereoskooppinen syvyys. Normaalikartan yksityiskohdat näyttävät litteiltä, kun objektia tarkastellaan eri kulmista VR-laseilla.

Normaalikarttoja voi kuitenkin edelleen käyttää lähes samalla tavalla VR-peleissä kuin perinteisissä peleissä. Quest-peleissä niitä on usein pakko käyttää, koska polygonirajat tulevat nopeasti vastaan, jos yrität mallintaa jokaisen yksityiskohdan realistisesti. Kannattaa varata tärkeimpiin malleihin, kuten hero-asetteihin, enemmän polygoneja, jotta saat tarvittavat yksityiskohdat aikaiseksi ilman, että olet täysin riippuvainen normaalikartoista. Ympäristön rakenteet ja suurissa määrissä esiintyvät propit voivat hyödyntää normaalikartoitusta enemmän, koska pelaaja ei usein tutki niitä kovin läheltä.

Low poly -mallien ja normaalikarttojen käyttöä lähestytään yleensä sillä periaatteella, että ensisijaisesti assetin siluetin tulee olla mahdollisimman hyvä. Valikoituihin silueteissa näkyviin reunoihin lisätään viisteet, jotta saadaan epäluonnollisen terävät reunat häivytettyä. Muissa osissa viisteet jätetään pois, erityisesti jos ne ovat siluetin sisällä. Joitakin mallin osia ei edes sisällytetä low poly -malliin muuta kuin sileällä polygonipinnalla, ja niiden yksityiskohtien edustaminen jää normaalikartan vastuulle.

Useimmissa VR-peleissä noudatetaan samankaltaista lähestymistapaa polygonien käytössä, mutta suurempi osa assetin komponenteista toteutetaan kolmiulotteisina, ja vain erittäin pienet yksityiskohdat jäävät normaalikartan varaan. Modernit pelialustat kykenevät käsittelemään miljoonia polygoneja yhtä aikaa, joten se ei ole ongelma. Quest 3:ssa rajat ovat kuitenkin enemmänkin satojen tuhansien polygonien luokkaa, joten monet yksityiskohdat jäävät silti normaalikartan hoidettaviksi, erityisesti kun on kyse kauempaa katsottavasta 3D-asettistä. Oman pelitestauksen perusteella normaalikartat ovat oikeastaan yllättävän

tehokkaita VR-peleissä, mutta niitä on käytettävä varovaisemmin tavallisiin peleihin verrattuna. Kuvassa 4 näkyvä assetti saa suurimman osan yksityiskohdistaan normaalikartasta, mutta se näyttää pelissä sopivalta taustaobjektina.



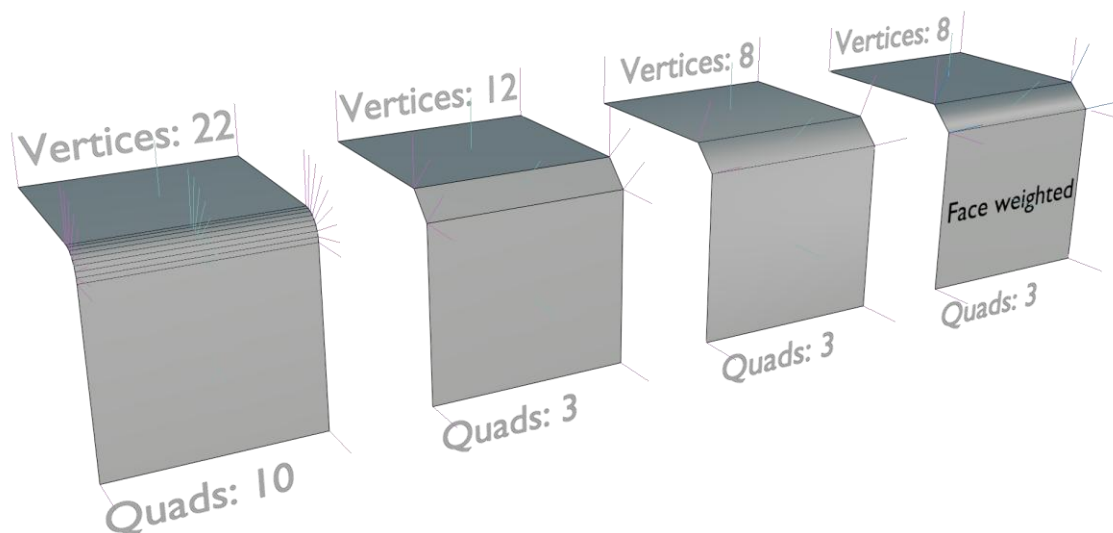
*Kuva 4. 3D-malli näytöstä, jossa on leivottu normal-kartta. (Siltala 2025).*

### 3.2.2 Polygonien käyttö riippumattomana normaalikartoista

Jokaiselle assetille ei kannata leipoa uniikkeja normaalikarttoja, sillä se tekisi pelistä erittäin raskaan ja veisi kohtuuttomasti aikaa. Hero-aseteille – kuten hahmoille ja heidän käyttämilleen esineille – on kuitenkin suositeltavaa leipoa normaalikartat ja tehdä assettikohtaiset tekstuurit. Ympäristöissä ja suuremmissa propeissa sen sijaan kannattaa hyödyntää toistuvia tekstuureja, decaleja ja trim-tekstuureja. Näiden tekniikoiden yhdistäminen verteksiväreihin ja ID-karttoihin maskaamisessa - sekä tehokas polygonien ja verteksien käyttö - mahdollistavat monenlaisten ympäristöjen luomisen.

Tärkeintä on saada malli näyttämään hyvältä. Paikoin on haastavaa tehdä low- tai medium poly -mallista visuaalisesti miellyttävä ilman high poly -mallista leivottuja normal-karttoja. Ympäristön malleissa tämä on kuitenkin usein välttämätöntä, mutta siihen on olemassa keinoja. Jos esimerkiksi halutaan lisätä silmillä nähtävää pyöreyttä reunoihin, viisteet toimivat hyvin. Viisteet voivat kuitenkin aiheuttaa mallin normaaleissa vääristymiä, jos niitä ei tue tukilooppeilla. Tukiloopit tekevät mallista raskaamman, joten joskus viisteet jätetään kokonaan pois reunoista, jolloin reunoista tulee epäluonnollisen terävät.

Tukiloopit voi kuitenkin jättää pois käyttämällä *face weighted normals* -menetelmää. Tyypillisessä mallinnuksessa 3D-ohjelmat tuottavat jokaiseen verteksiin keskiarvoistetun normaalin, joka perustuu verteksiä ympäröiviin polygoneihin. Alhaisilla polygonimäärillä tämä voi johtaa varjostusongelmiin, sillä esimerkiksi suuren, litteän pinnan reunalla sijaitseva viiste voi tehdä litteästä pinnasta kaarevan näköisen. Kuva 5 osoittaa, että *face weighted normals* tarjoaa vaihtoehdoisen tavan luoda verteksille normaalin, joka perustuu suurimman pinnan normaaliin. Tämä tarkoittaa, että litteän pinnan reunalla oleva viiste ei vaikuta suuren pinnan normaaliin ollenkaan, jolloin varjostus pysyy luonnollisena ja malli näyttää puhtaalta.



Kuva 5. Neljä esimerkkimallia, joissa on erilaisia viisteitä. (Siltala 2025).

Tuloksena on viiste, joka on polygonimäärältään alhainen eikä vaadi tukilooppeja. Ilman *face weighted normals* -menetelmää tämän viisteen toteuttaminen ilman visuaalisia vääristymiä olisi vaatinut ylimääräisiä verteksejä ja polygoneja, tehden mallista raskaamman.

Alien Isolation ja Cyberpunk 2077 ovat esimerkkejä peleistä, joissa *face weighted normals* -menetelmää on käytetty tehokkaasti. Näissä peleissä se erottuu selkeästi, sillä se tukee pelien taiteellista suuntaa, jossa 1980-luvun kulmikas

retrofuturismi on vahvasti esillä. Vaikka tätä menetelmää käytetään monissa peleissä, sen vaikutus näkyy erityisesti näissä kahdessa pelissä. Itsekin olen hyödyntänyt sitä tämän pelin aseteissa lisäyksityiskohtien tuomiseksi ympäristömalleihin, mutta aina harkiten. Joissain tapauksissa jätän reunat täysin viisteetömmiksi polygonimäärän vähentämiseksi. On kuitenkin tärkeää huomata, että vaikka reunassa ei olisi viisteitä, verteksimäärä pysyy samana pelimoottorissa, sillä jokaisessa terävässä reunassa verteksit on jaettu kahtia. Kuvassa 6 on esimerkki tekemästani suurikokoisesta propista, joka jakaa tekstuurinsa monien muiden assettien kanssa ja käyttää suhteellisen paljon viisteitä.



*Kuva 6. 3D-malli tehdaskoneesta, jossa on suuri määrä viisteitä. (Siltala 2025).*

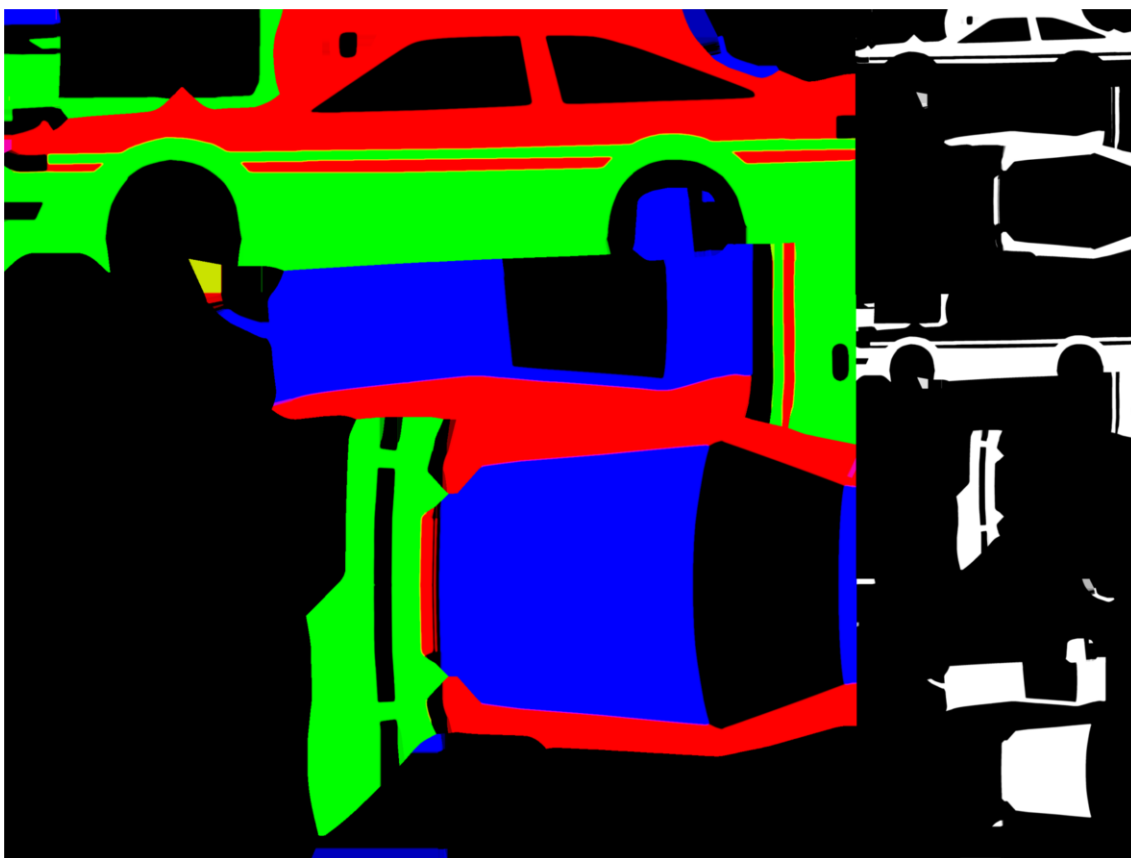
### 3.3.1 Tekstuurimaskit

Maskit ovat polygonien käytön rinnalla olennainen osa optimointia, ja maskausmenetelmiä on enemmän kuin yksi. Valitsen itse maskausmenetelmän riippuen assetin toteutusmenetelmästä. Jos käytän mallissa uniikkeja tekstuureja ja lei-

vottuja normaalikarttoja, suosin tekstuurimaskausta. Maskien avulla on mahdollista rajata tekstuurin tai värin alueita materiaalissa tarkasti. Maskin avulla voidaan esimerkiksi määrittää, missä kohtaa materiaalia näkyy likaa tai reunakulumaa. Yksinkertaisimmillaan maskilla voi esimerkiksi määrittää, minkä värinen jokin alue materiaalissa on.

Tekstuurimaski on mustavalkoinen kuva, jossa valkoiset kohdat edustavat näkyviä alueita ja mustat kohdat edustavat piilotettuja alueita. Esimerkiksi, jos haluat luoda halkeaman betoniseinään, maskissa halkeama on valkoinen ja muu osa maskista on musta. Jos halkeama halutaan pehmeämmäksi reunoilta, maskissa voi olla myös harmaan sävyjä halkeaman reunojen ympärillä. Näiden harmaiden sävyjen alueet näkyvät osittain, mikä luo pehmeän siirtymän maskin ja materiaalin välillä.

Tämä on hyödyllistä erityisesti silloin, kun halutaan luoda luonnollisia tai liukuvia muutoksia materiaalin pinnassa, kuten kulumista tai likaa, ilman teräviä reunoja. Maskien käyttö on erityisen tärkeää luodessa yksityiskohtaisia ja realistisia materiaaleja, sillä ne mahdollistavat hallitun ja tarkasti säädettävän visuaalisen vaikutelman ilman, että koko materiaaliin tarvitsisi tehdä merkittäviä muutoksia.



*Kuva 7. Kolme maskia pakattu R, G ja B värikanaville. (Siltala 2025).*

Maskit ovat yleensä greyscale-tekstuureja, mutta ne voidaan myös pakata useammaksi maskiksi yhteen tekstuuritiedostoon. Tässä tapauksessa jokaiselle värikanavalle määritellään oma maski. Kuvassa 7 on esimerkki tästä.

ID-maskausta käytetään usein tekstuurimuistin ja piirtokutsujen säästämiseksi. Maskaus itsessään on monille digitaalisen sisällön luojille tuttu käsite, ja ID-kartta toimii samankaltaisesti. ID-maskissa useampi maski on pakattu kuvatiedoston värikanaville.

ID-maski mahdollistaa useiden väri variaatioiden luomisen yhdestä materiaalista ilman tarvetta lisätä ylimääräisiä diffuusikarttoja, kuten kuvassa 8 näkyy. Yhdistettynä trim-tekstuureihin ja saumattomiin tekstuureihin, ID-maski on tehokas

tapa tuoda malliin sekä väri variaatiota että yksityiskohtia ilman suurta suorituskykyvaikutusta.



*Kuva 8. ID-maskilla väritetty auto. (Siltala 2025).*

### 3.3.2 Verteksivärimaskit

Verteksiväriä voi käyttää myös maskina. Suosin verteksivärimaskeja useimpien suuremmissa malleissa, jotka käyttävät toistuvia tekstuureja – kuten arkkitehtuurissa. Verteksivärejä on helppo maalata suoraan Unreal Editorissa, mahdollistaen esimerkiksi mudan maalaamisen lattiaan juuri niihin kohtiin, joissa se vaikuttaa tilanteen kannalta sopivalta. Kuten tekstuurimaskissa, verteksivärimaskissa valkoinen alue on näkyvä ja musta alue on näkymätön. Verteksivärimaskin etuna on tekstuurimuistin säästäminen, mutta sen muoto määräytyy verteksien mukaan. Ainoa tapa lisätä resoluutiota tällaiseen maskiin on lisätä verteksimäärää, mikä voi kuitenkin heikentää suorituskykyä.

Yksi yleisimmistä verteksivärimaskauksen käyttötarkoituksista on kahden tai useamman materiaalin välisen siirtymän toteuttaminen maastossa, kuten hiekan ja ruohikon välillä. Maastossa, jossa on usein epätasaisuuksia, käytetään suhteellisen tiheää polygoniverkkoa, joka soveltuu hyvin verteksivärimaskaamiseen. Verteksivärimaskin visuaalista laatua voi parantaa sekoittamalla siihen

saumattoman maskitekstuurin, joka rikkoo verteksivärimaskin mustan ja valkoisen sävyn välisen siirtymän, tehden siitä luonnollisemman näköisen. Kuvassa 9 on edellä mainitulla tavalla toteutettu siirtymä asfaltin ja katukivien välillä.



*Kuva 9. Kahden materiaalin välinen siirtymä 3D-ympäristössä. (Siltala 2025).*

Olen tässä pelissä käyttänyt verteksivärimaskausta pääosin mallien värittämiseen. Se on suorituskyvyn kannalta edullinen tapa luoda malliin visuaalista kiinnostavuutta, sillä vaikka materiaalin pohjatekstuuri on valkoinen tai harmaa, toistuvaan käyttöön suunniteltu trimsheet tai saumaton tekstuuri, verteksivärimaskin neljä värikanavaa voivat tuoda siihen monta eri väriä. Lisäksi olen käyttänyt sitä likaisemmissa ympäristöissä hiekan ja muiden kerrosten lisäämiseen.

### 3.4.1 Toistuvat tekstuurit

Toistuvat tekstuurit ovat hyödyllisiä suurten assettien teksturoinnissa, koska ne mahdollistavat yksityiskohtien lisäämisen ilman, että assetin tekstuurikoko kasvaa kohtuuttoman suureksi. Tekstuuri toistuu jatkuvasti mallin pinnalla. Tämä on erityisen hyödyllistä, kun halutaan peittää suuri alue yhdellä tekstuurilla, kuten esimerkiksi maastossa, suurissa seinäpinnoissa tai ympäristöissä, joissa ei tarvita ainutlaatuista yksityiskohtaisuutta jokaiselle osalle.

Saumattomat tekstuurit ovat yleisimpiä toistuvia tekstuureja. Ne ovat suunniteltu siten, että tekstuurilla ei ole näkyviä reunoja, jolloin se voi toistua saumattomasti kaikissa suunnissa. Tämä takaa sen, että tekstuuri näyttää jatkuvalta, vaikka sitä käytetään laajoilla alueilla. Tällaiset tekstuurit ovat erityisen tehokkaita esimerkiksi lattioiden, seinien ja taustaelementtien teksturoimisessa.

### 3.4.2 Trim-tekstuurit

Trim-tekstuuri on yhdellä akselilla toistuva tekstuuri, jota voidaan käyttää monenlaisissa tarkoituksissa. Yleisimmin sitä hyödynnetään ympäristöjen teksturoimiseen, mutta sen käyttö ulottuu myös suuriin laitteisiin tai kulkuneuvoihin, kuten laivoihin tai avaruusaluksiin. Trim-tekstuurit otetaan usein käyttöön silloin, kun asettiin tarvitaan korkeaa tekselitiheyttä, mutta tekstuurien määrää ei voida lisätä kohtuuttomasti. Ne tarjoavat tavan luoda yksityiskohtia ja tekstuuriefektejä ilman, että resurssien käyttö ylittää suorituskyvyn rajoja.

## 4 Assetit

Tässä luvussa esittelen valikoituja esimerkkejä peliin tekemästäni 3D-grafiikasta. Jokaisella esimerkillä oli omat tekniset ja luovat haasteensa, mikä teki niiden toteutusprosessista jollain tapaa ainutlaatuisen.

#### 4.1 Animoitu proppi: Robot Arm



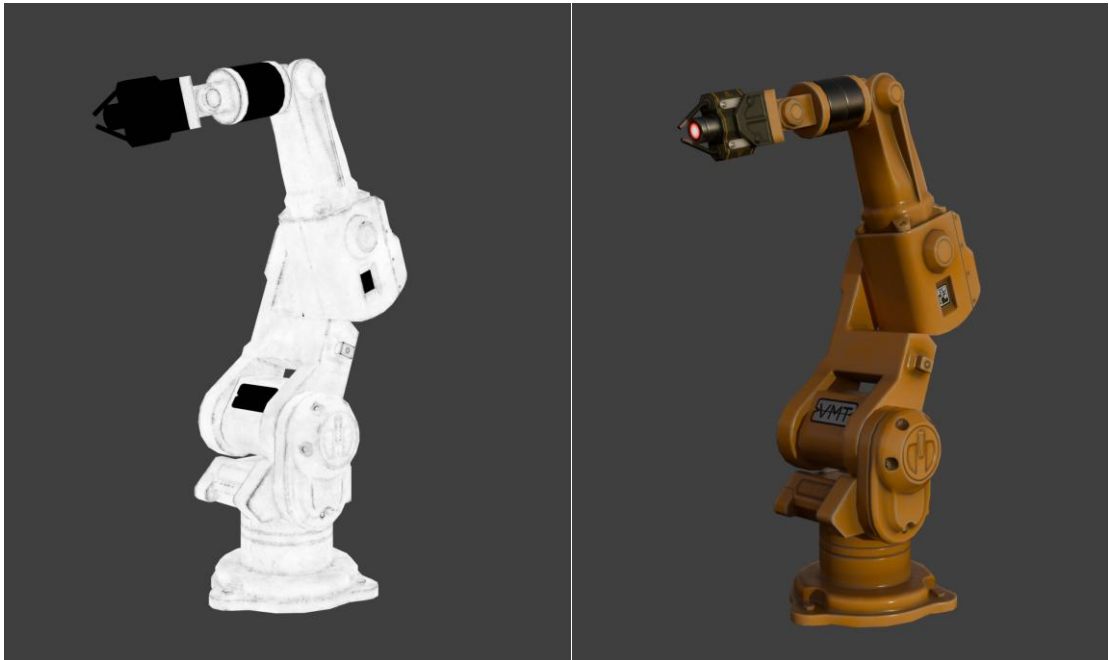
*Kuva 10. Mekaaninen käsi, joka toimii asetehaassa. (Siltala 2025).*

Olen tehnyt peliin lukuisia asetteja hyödyntäen opinnäytetyössä käsittelemiäni menetelmiä. Yksi näistä on robottikäsi, joka on liikkuva proppi. Se on korkeudeltaan noin neljä metriä ja suhteellisen kapea. 1024x1024 pikselin tekstuureilla sain asettiin hyväksyttävän tekselitiheyden, joka on osasta riippuen 512 px/m tai 256 px/m. Otin myös huomioon sen, että assetti saattaa näkyä 512x512 pikselin tekstuureilla Quest 2 -laitteella, joten pidin huolen, että assetti näyttää puolitettullakin tekselitiheydellä hyväksyttävältä.

Sen polygonimäärä on hieman yli 2000 kolmiota, riippuen päätyosaan asennetusta työkalusta. Työkaluja on kolme erilaista: Puristin, hitsauskone ja laserleikkuri. Nämä työkalut ovat sisällytetty samaan tekstuuriin, ja työkalut koostuvat

osittain jaetuista komponenteista. Päädyin tähän ratkaisuun UV-tilan säästämiseksi, jotta tekselitiheys olisi mahdollisimman korkea. Kuvassa 10 näkyy käsi varustettuna puristimella.

Assetti tulee käyttöön asetehaaseen, jossa se kokoaa robotteja liukuhihnalla. Robottikädestä esiintyy muutama eri variaatio, joilla on eri työkalut ja väritykset. Kädet on koottu erillisistä osista blueprinteissa ja animoitu käyttäen level sequenceja, joilla liukuhihnoilla liikkuvat robotitkin on animoitu. Värivariaatiot on toteutettu ID-kartalla. Kuvassa 11 asettissa on oranssi väritys.



*Kuva 11. ID-kartalla väritetty mekaaninen käsi. (Siltala 2025).*

Aloitin asettin tekemisen tuomalla blockout-version mallinnusohjelmaan. Blockout-versio asettista oli sopivan kokoinen kenttään ja mittasuhteiltaan uskottava, joten erityisiä haasteita ei ilmennyt asettin visuaalisen ilmeen keksimisen kanssa. Se oli paljolti tarkoitettu edustamaan nykyaikaista tehdasrobotia, joten pystyin katsomaan kuvia oikeista robottikäsisistä, tekemään luonnoksen niiden perusteella ja ryhtyä sen jälkeen mallintamaan sitä.

Tein ensiksi kuvassa 12 esitetyn high poly mallin, jonka jälkeen tein low poly mallin, UV-kartat ja tekstuurit. Tässä tapauksessa vein keskeneräisen high poly mallin pelimoottoriin testikäyttöön vähennetyllä polygonimäärällä, ennen kuin lopullinen low poly malli oli valmis ja teksturoitu. Tässä vaiheessa robottikäsi oli artikuloitu vain yhdestä kohdasta, ja level sequence -animaatiot olivat vielä hyvin pelkistettyjä.



*Kuva 12. Mekaanisen käden high poly -malli. (Siltala 2025).*

Kun sain assetin tekstuureineen valmiiksi, vein sen erillisinä osina UE5-editoriin ja kokosin sen blueprintiksi. En toteuttanut tätä assettia skeletal meshinä, koska se olisi ollut turhan raskasta laitteelle. Sen sijaan animoin blueprinttiä level sequence editor -työkalun avulla, joka toimi lopulta erittäin hyvin tässä käyttötarkoituksessa. Kuvassa 13 näkyy robottikädet animoituina peliympäristössä.

Tässä assetissa oli erityisiä haasteita, joita ei staattisessa propissa ilmene yhtä merkittävästi. Pelissä ei käytetä dynaamisia valoja, dynaamisia varjoja tai Ambient Occlusion -varjostusta, joten oli pieni riski, että assetti saattaa näyttää litteältä pelissä. Pyrin estämään tätä lisäämällä pohjavärikarttaan leivotun ambient occlusion -kartan ja cavity-kartan multiply-sekoitustilalla ja alhaisella opacity-asetuksella. Tämä auttaa luomaan hillityn varjostuksen illuusion liikkuvassa objektissa, joka on vain staattisilla valoilla valaistu.



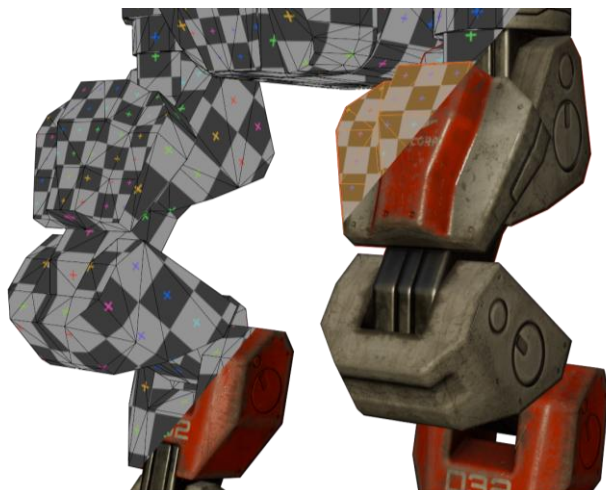
*Kuva 13. Mekaaniset kädet tehdasympäristössä. (Siltala 2025).*

## 4.2 Hahmo: Mecha



*Kuva 14. Mecha edestä ja takaa. (Siltala 2025).*

Proppien lisäksi olen luonut peliin hahmoja ja hahmojen varusteita. Kuvassa 14 esitetty mecha, eli taistelurobotti, on korkeudeltaan noin kolme metriä ja leveydeltään kaksi metriä. Tekselitiheys on monissa UV-saarekkeissa 256 px/m, mutta joissakin yksityiskohtaisuutta vaativissa saarekkeissa se on korkeampi. Tekstuuriresoluutio on 1024 pikseliä. Kuvassa 15 näkyy tekselitiheyseroja.

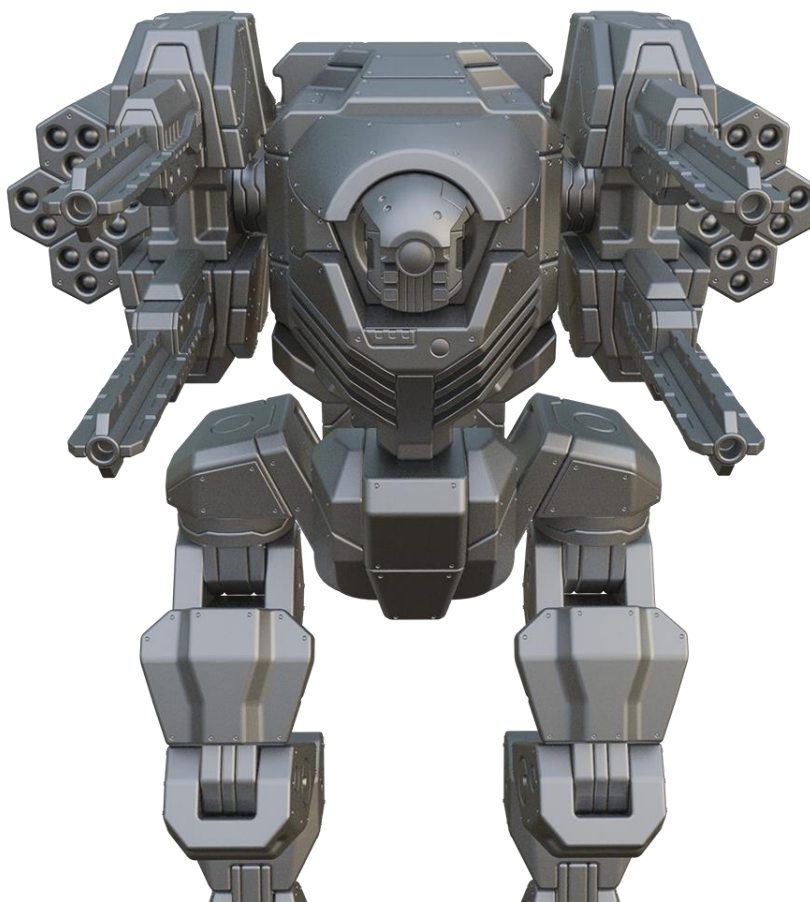


*Kuva 15. Mechan jalan osien tekselitiheyksien eroja. (Siltala 2025).*

Mecha on yksi pelissä esiintyvistä robottivihollisista. Siitä on olemassa tavallinen variaatio ja suurempi boss fight variaatio. Käytin tässä assetissa symmetriaa ja toistuvia elementtejä niissä kohdissa, joissa se oli visuaalisesti sopivaa ja optimoinnin kannalta järkevää. Tämä auttoi säästämään UV-tilaa, saavuttaen mahdollisimman suuren tekselitiheyden. Tästä huolimatta tekselitiheys on suhteellisen alhainen, joten en yrittänyt lisätä high poly -malliin liian pieniä yksityiskohtia, enkä myöskään lisännyt liian pieniä tekstejä tai symboleja teksturointivaiheessa. Varasin kuitenkin strategisesti tiettyihin kohtiin tarpeeksi tekselitiheyttä, jotta pienemmät yksityiskohdat näyttäisivät mahdollisimman hyviltä.

Aloitin assetin toteuttamisen piirtämällä muutaman luonnoksen. Sen jälkeen mallinsin pelkistetyn prototyypin, jossa oli sopivat mittasuhteet ja toimivat mekanismit jaloille ja tykeille. Tein tämän prototyypin pohjalta kuvassa 16 näkyvän high poly -mallin. Varsinaisessa mallinnusvaiheessa lisäsin mechalle sopivan

aseistuksen koostuen raidetykeistä ja räjähdelaukaisinpattereista.



*Kuva 16. Mechan high poly -malli. (Siltala 2025).*

High poly -mallin suunnittelu ja toteutus oli mielenkiintoinen prosessi. Peliin oli tehty jo aikaisemmin robottivihollisia, jotka ovat pelimaailmassa saman korporaation valmistamia. Koska tämäkin mecha on myös saman korporaation tuote, pyrin tekemään siitä tyyliltään samankaltaisen, mutta kuitenkin tunnistettavan ja ainutlaatuisen omana hahmonaan. Lisäksi halusin tehdä sen jaloista mekaanisesti toimivat ja koko robotista mittasuhteiltaan vankan ja tasapainoisen näköisen. Ylävartalon aseistus, ammuksien ja generaattori vaikuttavat painavilta, joten halusin tehdä mechan lantiosta ja jaloista suhteellisen jyrkät.

Pyrimme myös ottamaan suunnittelussa huomioon sen, että Quest 3-pelihahmoissa ei voi olla kovin korkeita polygonimääriä. Saamani tietojen mukaan polygonimäärät voivat riippua projektista, tilanteesta ja muista erilaisista seikoista, mutta hyvä määrä on kuitenkin siinä 7000 kolmion paikkeilla. Tässä mechassa on

6468 kolmiota. Pyrin suunnittelemaan mallin alusta asti alhainen polygonimäärä mielessä, mutta en koe sen rajoittaneen minua luovasti kovinkaan paljoa, sillä kulmikkua ja suuria muotoja on useimmiten muutenkin luvassa, kun kyseessä on Cyberpunk-genren tuotanto.

### 4.3 Ympäristö: Laboratorio



*Kuva 17. Laboratorion generaattoritila. (Siltala 2025).*

Yksi tähän peliin tekemistäni ympäristöistä on laboratorio. Ympäristöstä oli alustava versio tehtynä, mutta se vaati yhtenäisen visuaalisen ilmeen. Tein modulaarisen kitin, joka sisältää seinä- lattia- ja kattopaloja, sekä erilaisia välikappaleita. Teksturoin seinät trim-tekstuureilla ja lattiat ja katot saumattomilla tekstuureilla, joita näkyy kuvassa 17. Toin lisäyksityiskohtia seiniin käyttämällä decal-tekstuureja. Käytin verteksivärimaskeja lisätäkseni seiniin värejä.

Kuten muissakin tekemissäni ympäristöissä tässä projektissa, olin vastuussa laboratorion valaistuksesta. Valaistus on tässä projektissa ollut hieman haastavaa, mutta tähän ympäristöön päästyäni olin jo aika pitkälti selvittänyt forward rendering -valaistuksen oikut. Iteroiminen yleensä tuottaa hyvän tuloksen ennen pitkää, mutta valojen leipominen vie siinä iteraatioiden välissä oman aikansa.

Usein pitää vain aloittaa asettamalla valoja tilaan, leipoa valot ja sen jälkeen säätää valojen voimakkuuksia tai kokeilla uusia sijainteja valoille. Kuvassa 18

esiintyvän käytävän valaistus ei ole lopullinen, mutta olen sitä tehdessä leiponut valot useampaan kertaan uusiksi. Heijastusten kirkkautta joutuu usein säätämään käsin, ja heijastuksenkaappareita on asteltava varovasti. Joskus täytyy myös säätää kimpoavan valon voimakkuutta, jos tilan pinnat eivät heijasta tarpeeksi paljon valoa.



*Kuva 18. Laboratorion käytävä. (Siltala 2025).*

#### 4.4 Interaktiivinen objekti: Hackpanel



*Kuva 19. Hackpanel. (Siltala 2025).*

Tämä assetti sisältää minipelin, jossa pelaaja ottaa VR-ohjaimilla kiinni kahvoista ja suorittaa niitä manipuloimalla vaaditun objektiivin. Assetti on suurehko, yli kaksi metriä korkea, ja pelaaja näkee sen hyvin läheltä. Suurin haaste oli ylläpitää sopiva yksityiskohtaisuuden taso, vaikka tekstuurit ovatkin vain 1024x1024 pikseliä. Resoluutio on muuten sopiva tämänkokoiseen assettiin, mutta koska sitä tarkastellaan läheltä, minun oli varattava holografista näyttöä lähimpänä oleville osille suhteellisen paljon tekselitiheyttä. Lisäksi päädyin muotoilemaan assetin niin, että siinä on mahdollisimman monta toistuvaa elementtiä, jotka ovat UV-kartassa päällekkäin. Tämä antoi minulle vielä lisää tekselitiheyttä koneen muihinkin osiin.

Tästä objektista oli alun perin olemassa pelkistetty malli, joka oli koottu erilaisista propeista. Kokonaisuus oli tosin visuaalisesti epäyhtenäinen ja tekniseltä toteutukseltaan epäoptimaalinen. Assetin lopullinen ulkoasu sai alkunsa karkeista konsepteista, jonka jälkeen aloin mallintamaan ja samalla lisäämään elementtejä, joita alkuperäisissä konsepteissa ei ollut. Saatuaani assetin low poly -

mallin sellaiseen kuntoon, että osien sijainnit eivät muutu, lisäksi sen pelimootoriin, jotta ohjelmoija pääsi aloittamaan asettiin liittyvän minipelin lopullisen version toteutuksen.

## 5 Yhteenveto

Quest 3 -pelin grafiikkaa suunniteltaessa on tehtävä kompromisseja suorituskyvyn vuoksi, mutta nämä eivät automaattisesti estä halutun visuaalisen tyylin saavuttamista. Kun pelimaailman mittakaava ja tyyli rajataan jo varhain, teknisiä ongelmia ilmenee vähemmän. Esimerkiksi kenttä kannattaa pitää kohtuullisen kokoisena ja teemaltaan selkeänä, jotta yksityiskohtia voidaan lisätä tiheämmin ilman suorituskykyongelmia.

Asettien suunnittelussa optimointi on keskeistä: muotoilun yksinkertaistaminen sekä alhaiset tekstuuriresoluutiot ja polygonimäärät ovat käytännössä välttämättömiä. Myös UV-tilan tehokas hyödyntäminen ja valokartta UV-karttojen valmistelu on syytä huomioida jo mallinnusvaiheessa. Staattinen valaistus ja tekstuurien uudelleenkäyttö ovat konkreettisia keinoja parantaa suorituskykyä ilman suuria visuaalisia kompromisseja.

Lisäksi esimerkiksi LOD-tasojen hyödyntäminen ja mahdollisimman kevyiden varjostimien käyttö voivat vaikuttaa merkittävästi lopputuloksen toimivuuteen. Kun kaikki osa-alueet tukevat toisiaan alusta asti, teknisiä rajoitteita tuskin edes huomaa peliä kehittäessä.

## Lähteet

Epic Games 2023. Unwrapping UVs for Lightmaps. Verkkosivu. [https://dev.epicgames.com/documentation/en-us/unreal-engine/unwrapping-uv-for-lightmaps?application\\_version=4.27](https://dev.epicgames.com/documentation/en-us/unreal-engine/unwrapping-uv-for-lightmaps?application_version=4.27) (viitattu 22.4.2025).

Artola, Sofia 2024. The art behind gameplay: A journey through video game visual styles. Verkkosivu. <https://hickmandesign.co.uk/blog/history/art-behind-gameplay-journey-video-game-visual-styles> (viitattu 22.4.2025).

Justus-Liebig-Universität Gießen. (i.a.). Data sheet Meta Quest 3. PDF. [https://www.uni-giessen.de/de/studium/lehre/projekte/nidit/goals/quest3/datasheet\\_quest-3.pdf](https://www.uni-giessen.de/de/studium/lehre/projekte/nidit/goals/quest3/datasheet_quest-3.pdf) (viitattu 22.4.2025)

Trevor Dasch 2023. PC rendering techniques to avoid when developing for mobile VR. Verkkosivu. <https://developers.meta.com/horizon/blog/pc-rendering-techniques-to-avoid-when-developing-for-mobile-vr/> (viitattu 22.4.2025)

Jukka Häkkinen 2017. Virtuaalitodellisuus voi aiheuttaa pahoinvointia – näin estät sen. Verkkosivu. <https://yle.fi/aihe/a/20-191520> (viitattu 22.4.2025).

## **Kuvalähteet**

Kuva 1. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 2. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 3. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 4. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 5. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 6. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 7. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 8. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 9. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 10. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 11. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 12. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 13. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 14. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 15. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 16. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 17. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 18. Siltala, Ilari 2025. Kuva omasta 3D-mallista.

Kuva 19. Siltala, Ilari 2025. Kuva omasta 3D-mallista.