Mikko Siivinen

Reaaliaikainen 3D-visualisointi selainympäristössä

Muotoilijan johdanto Web3D-sisällön tuottamiseen

Metropolia Ammattikorkeakoulu

Muotoilija

3D-animointi ja -visualisointi

Opinnäytetyö

25.4.2025

Tiivistelmä

Tekijä(t):	Mikko Siivinen
Otsikko:	Reaaliaikainen 3D-visualisointi selainympäristössä
Sivumäärä:	41 sivua + 2 liitettä
Aika:	25.4.2025
Tutkinto:	Muotoilija (AMK)
Tutkinto-ohjelma:	Muotoilun tutkinto-ohjelma
Pääaine:	3D-animointi ja -visualisointi
Ohjaaja(t):	Lehtori Jaro Lehtonen

Tämä opinnäytetyö käsittelee reaaliaikaisen 3D-visualisoinnin suunnittelua ja toteutusta selainympäristössä 3D-mallinnustaitoja omaavan muotoilijan näkökulmasta. Opinnäytetyössä tutkitaan Web3D:n keskeisiä teknologioita sekä laadukkaiden ja tehokkaasti toimivien 3D-mallien luomista selainympäristöön.

Työn tavoitteena on tarjota muotoilijoille ohjeita ja suosituksia jotka auttavat luomaan teknisesti toimivia reaaliaikaisia 3D-visualisointeja verkkoympäristöön. Tutkimusmenetelmänä on olemassaoleviin palveluihin ja teknologian tuottamiin mahdollisuuksiin perehtyminen, selainympäristössä toimivan 3D-mallin rakenteeseen syventyminen sekä sisällön tuottamisen testaaminen.

Opinnäytetyö perehtyy pintapuolisesti taustalla vaikuttaviin teknologioihin ja ohjelmointiin jotka eivät välttämättä suoraan vaikuta muotoilijan tekemiseen, mutta selkeyttävät ja auttavat ymmärtämään reaaliaikaisen 3D-visualisoinnin lainalaisuuksia verkkoympäristössä.

Asiasanat: Web3D, Reaaliaikainen 3D-visualisointi

Abstract

Author(s):	Mikko Siivinen
Title:	Real-time 3D visualization in browsers
Number of Pages:	41 pages + 2 appendices
Date:	25 April 2025
Degree:	Bachelor of Culture and Arts
Degree Programme:	Design
Major:	3D Animation and Visualization
Instructor(s):	Lehtori Jaro Lehtonen

This thesis explores the design and implementation of real-time 3D visualization in a browser environment from a 3D designer's perspective. It examines the key technologies of Web3D as well as the creation of high-quality and efficiently functioning 3D models and visualizations.

The aim of the thesis is to provide designers with guidelines and recommendations in creating technically functional real-time 3D visualizations for the web. The research method involved a technical and visual exploration of existing services, studying the structure of a technically functional Web3D model as well as testing content creation.

The thesis briefly covers underlying technologies and programming that may not directly affect the designer's work but can help clarifying and deepen the understanding of design principles in web environment.

Keywords: Web3D, real-time 3D visualization

This thesis has been checked using Turnitin Originality Check service.

Sisällys

1 Johdanto

Tämän opinnäytetyön tavoitteena on tutkia reaaliaikaista 3D-visualisointia selainympäristössä muotoilijan näkökulmasta. Työ perehtyy yleisellä tasolla Web3D-sisällön taustalla vaikuttaviin teknologioihin ja tekee havaintoja sen ilmiöistä kaupallisessa, kulttuurillisessa, tieteellisessä ja opetuksellisessa käytössä. Opinnäytetyö syventyy erityisesti 3D-mallinnuksia luovan muotoilijan teknisiin ja visuaalisiin työkaluihin. Työssä käydään läpi kuinka 3D-mallit valmistellaan ja renderoidaan selaimessa sekä yleisiä visuaalisia keinoja, joita selainympäristössä olevat reaaliaikaiset 3D-visualisoinnit hyödyntävät.

Tutkimusmenetelmänä käytetään havaintoja reaaliaikaisen 3D-visualisoinnin käytöstä eri verkkosivuilla, asiantuntijalähteitä, työkokemusta ja asiaan tehdyn tutkimustyön tuloksia. Lopuksi tehdään havainnollistava käytännön toteutus, jossa 3D-malli renderoidaan reaaliaikaisesti paikallisesti käynnistetyssä verkkosivussa yksinkertaisilla interaktioilla.

Käytännön osuudessa ei perehdytä itse mallintamiseen, vaan käytetään valmista reaaliaikaiseen renderointiin sopivaa mallia, joka on luotu opinnäytetyössä opittujen menetelmien pohjalta. Ohjelmointiosuus käydään läpi pintapuolisesti eikä syvenny sen taustalla vaikuttaviin teknologioihin, vaan sen on tarkoitus auttaa muotoilijaa saamaan mahdollisimman vähällä ohjelmointimäärällä tehtyä reaaliaikaisen 3D-mallin renderoija selaimeen.

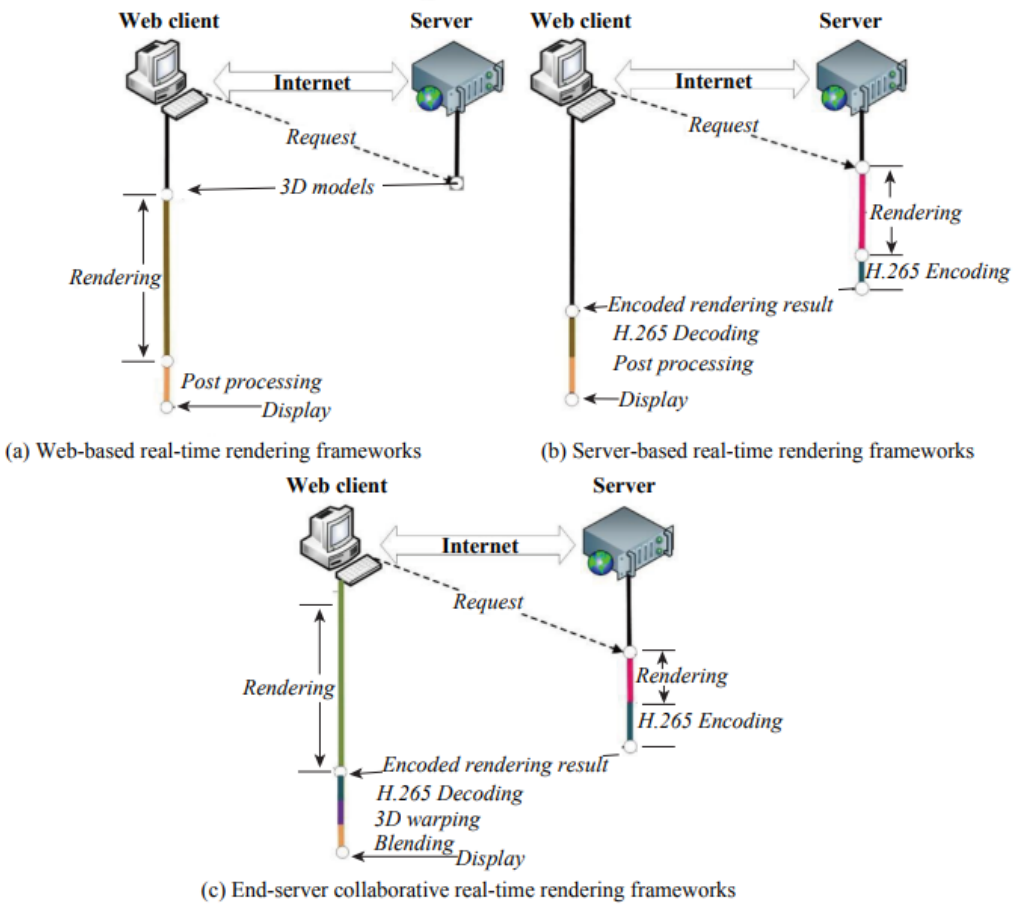
2 Katsaus 3D-visualisointiin selainympäristössä

2.1 Historia ja keskeiset teknologiat

Ensimmäiset selainympäristössä tehdyt reaaliaikaiset 3D-visualisoinnit sijoittuvat 90-luvun alkupuoliskolle. Tuolloin visioitiin alustasta riippumatonta 3D standardia selainympäristöön (Raggett 1994). Aina 2000-luvun jälkipuoliskolle saakka reaaliaikainen 3D-visualisonti vaati selaimeen erikseen ladattavan laajennuksen tai lisäosan, kuten Adobe Flash Playerin. Selaimien lisäosat olivat kuitenkin ongelmallisia tietoturvan, yhteensopivuuden ja ylläpidon kannalta (Linux Code 2024).

Vuodesta 2011 alkaen 3D-grafiikkaa on pystytty esittämään verkkosivuilla ilman lisäosia WebGL (Web Graphics Library) -ohjelmointirajapinnan käyttöönoton myötä. WebGL on kehitetty renderoimaan suorituskykyistä ja interaktiivista 3D- ja 2D-grafiikkaa selaimille ilman lisäosia. Se on ollut tuettuna 2010-luvun puolivälistä saakka yleisimmissä selaimissa kuten Google Chrome, Microsoft Edge, Apple Safari ja Mozilla Firefox. (W3Schools i.a.)

WebGL :n pohjautuva 3D-visualisointi hallitaan verkkosivujen Canvas-elementissä JavaScript -ohjelmointikielen avulla. Itse grafiikan piirtämiseen WebGL käyttää varjostintehosteisiin erikoistunutta OpenGL ES Shading Language (GLSL) -ohjelmointikieltä (Mozilla 2025). WebGL:n avulla pystytään siis esittämään 3D-sisältöä suurimmassa osassa selaimia alustasta riippumatta ja ilman lisäosia, mutta se on matalan tason ohjelmointikieli joka vaatii syvällistä teknistä osaamista ja ohjelmointia jotta 3D-graafikoiden luomia sisältöjä voitaisiin verkkosivuilla esitellä. Tämän vuoksi on tehty erilaisia ohjelmointikirjastoja helpottamaan kehittäjien työtä. Tällainen on esimerkiksi suosittu korkean tason JavaScript-kirjasto Three.js, joka on suunniteltu tuomaan 3D-graafikon kannalta oleellisempia asioita kuten valoja, varjoja, materiaaleja ja tekstuureja selaimeen ilman että niitä ohjelmoidaan itse. Three.js -kirjaston käyttäminen kuitenkin vaatii JavaScript-ohjelmoinnin osaamista, kykyä tuottaa HTML-merkintäkieltä ja selainpuolen kehittämisen yleistä ymmärtämistä. (Three.js 2025.)



Kuva 1. Kuvassa näkyy Web3D-sisällön eri renderointimahdollisuudet (Geng yu, et. al 2023)

Web3D-sisältöjä on tyypillisesti kolmenlaista ja erilaisia tarkoituksia varten. Verkkopohjaiset reaaliaikaisen renderoinnin kehykset toimivat siten, että renderointi tapahtuu suoraan selaimessa käyttäjän laitteella. Tämä on yleisin ja palvelimen kannalta kevyin tapa Web3D-sisältöjen esittämiseen. Selaimien rajoitettu renderointikyky ei kuitenkaan aina saavuta haluttua lopputulosta ja palvelimen puolella laskentaa ei tapahdu ollenkaan, jolloin turvaudutaan osittain tai kokonaan palvelinpuolen laskentaan.

Palvelinpohjaisissa reaaliaikaisen renderoinnin kehyksissä renderointi tapahtuu palvelimella ja kuva suoratoistetaan käyttäjälle. Käyttäjälle esitetään ainoastaan renderoinnin lopputulos ottamatta osaa itse sisällön laskentaan. Tämän tyyppinen kehys mahdollistaa korkeamman tason turvallisuuden 3D-sisällön ollessa palvelimilla eikä käyttäjällä, sekä vähentää käyttäjän puolella tapahtuvaa

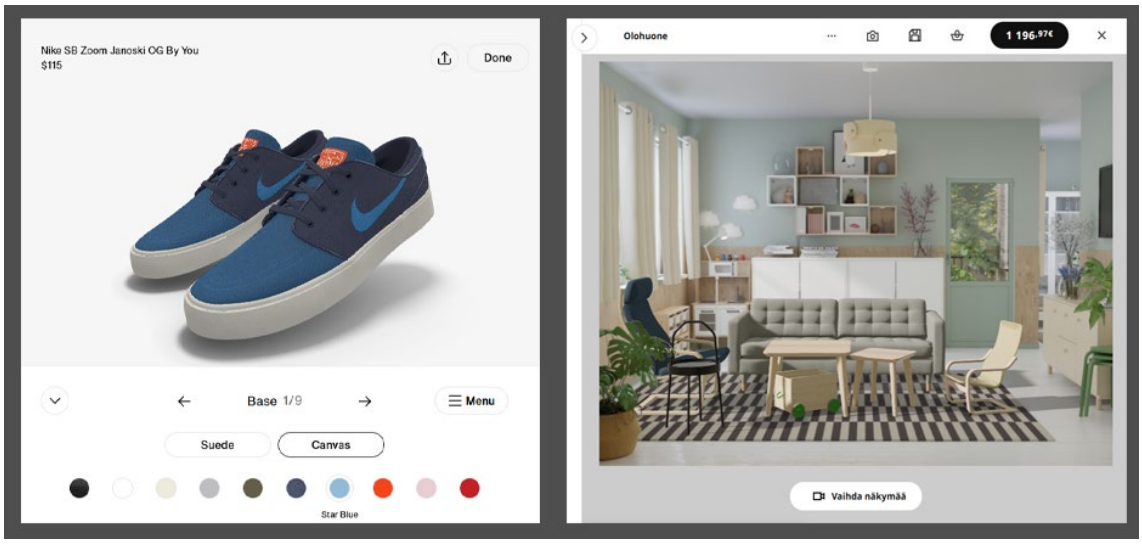
laskentaa, mahdollistaen visuaalisesti näyttävimmän lopputuloksen. Tästä syystä tämäntyyppistä renderointia käytetään esimerkiksi pilvipalveluihin perustuvissa pelipalveluissa.

Kolmas tyypillinen tapa käsitellä 3D-sisältöjä selaimissa on pääte- ja palvelinpuolen yhteistyöhön perustuvat reaaliaikaisen renderoinnin kehykset. Tässä tapauksessa renderointia ja laskentaa jaetaan käyttäjän laitteen ja palvelimen välillä. Sen avulla on mahdollista vähentää ja optimoida käyttäjän ja palvelimen välillä tapahtuvaa laskentaa, ja onkin ideaali ratkaisu varsinkin sisällöille, joissa suorituskyky on erityisen tärkeää tai palvelun käyttö on laaja-alaista. (Geng Yu, et. al 2023.)

2.2 Hyödyntäminen eri aloilla

Reaaliaikaista 3D-visualisointia käytetään selainympäristössä muun muassa kaupallisessa, kulttuurillisessa ja opetuksellisessa toiminnassa.

Tätä teknologiaa hyödyntäviä kaupallisia yrityksiä ovat esimerkiksi Nike ja IKEA. IKEA käyttää tuote-esittelyissään reaaliaikaisia ja interaktiivisia 3D-malleja sekä lisätyn todellisuuden (AR) teknologiaa, jonka avulla käyttäjät voivat sijoittaa virtuaalisia huonekaluja haluamaansa tilaan ennen ostopäätöstä. IKEA Kreativ -palvelussa voidaan suunnittella ja sisustaa tila selaimessa määrittelemällä neliömäärät, huoneen muoto, ikkuna- ja ovipaikat sekä asettamalla huonekaluja tilaan. (IKEA 2023) 3D-artistien rooli tässä on työpaikkailmoituksen perusteella keskittynyt muunmuassa 3D-mallien optimointiin ja hallintaan, fotorealistisen 3D-visualisoinnin automatisointiin ja realismin parantamiseen materiaali- ja renderointitekniikoin (GameJobs 2023).

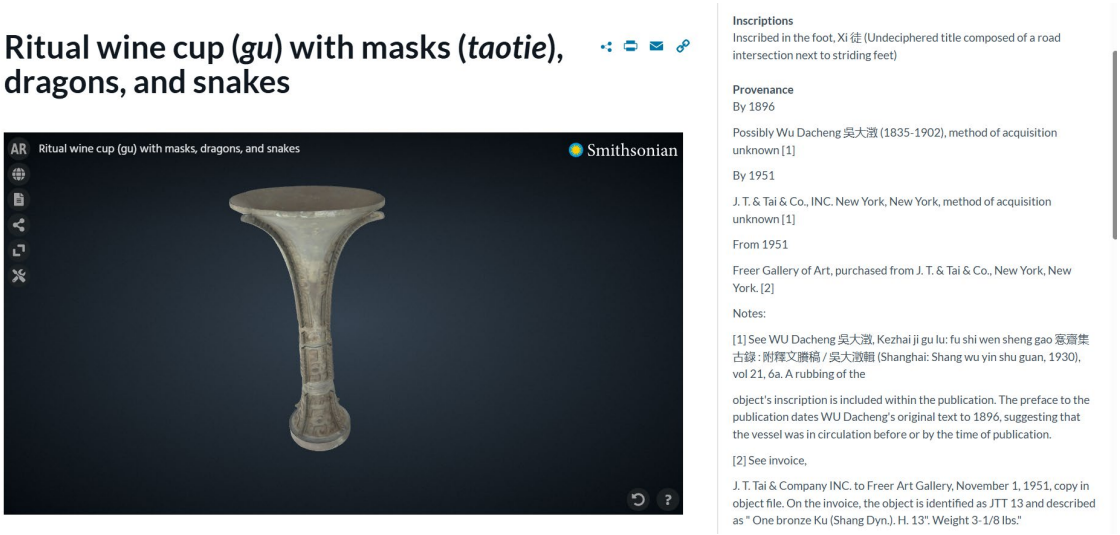


Kuva 2. Kuvakaappaus Nike By You (vas.) ja IKEA Kreativ -verkkosovelluksista

NIKE on luonut selaimeen interaktiivisen palvelun, jossa voi suunnitella kenkien värit ja materiaalit sekä asettaa ne AR-teknologian avulla ihmisestä skannatun mannekiinin jalkaan (Nike 2025a). 3D-artisteilta on vaadittu hyviä 3D-mallinnustaitoja mahdollisimman fotorealististen kenkien tekoon, kykyä työskennellä järjestelmällisesti ja selkeästi jotta mallit on helppo tuoda Niken järjestelmiin sekä teknistä osaamista ja kykyä optimoida mallit reaaliaikaista renderointia varten (Nike 2025b).

Selainpohjaisissa peleissä on hyödynnetty viime vuosina reaaliaikaista 3D-grafiikkaa. Unreal Engine 4 ja Unity -pelimoottorit tukevat WebGL-pohjaisten toteutuksien luomista, sekä pelimoottorimaisia korkean tason web-ohjelmointikirjastoja on tarjolla useita (Naif Mehanna, Walter Rudametkin 2023). Pelinkehittäjille suunnatussa kyselyssä kiinnostus selainpohjaisten toteutuksien tekemiselle on lisääntynyt 16% vuonna 2025, joka on korkein luku vuosikymmeneen (Byshonkov 2023).

Selainpohjaista 3D-visualisointia on hyödynnetty kulttuurialalla esimerkiksi historiallisten 3D-skannattujen esineiden esittämiseen ja tarkastelemiseen sekä virtuaalimuseoissa, joissa käyttäjät voivat liikkua kolmiulotteisessa tilassa vuorovaikutuksellisesti (Web3D Consortium 2023).



Kuva 3. Kuvakaappaus Smithsonian Museumin 3D-katselijasta, jossa esitellään historiallista fotogrammetria -mallia

Internetissä käydyn keskustelun perusteella haasteita tällä hetkellä suurempien pelien kehitykselle on selaimen sopimattomuus tähän tarkoitukseen sekä sen ”väliinputoaminen” laitteiden välillä. Mobiilikäyttäjät ja -kehittäjät todennäköisesti suosivat applikaatioita niiden helppokäyttöisyyden vuoksi. Käyttäjien kannalta on helpompi avata applikaatio puhelimesta kuin navigoida selaimella sivulle, joka vaatisi kirjautumisen ja joskus komponenttien uudelleenlataamisen. Kehittäjien kannalta valmiit alustat tarjoavat laskutusmahdollisuudet eikä heidän tarvitse olla riippuvaisia selaimiin liittyvistä teknisistä rajoitteista ja muutoksista. Pöytäkoneiden käyttäjillä sen sijaan on jo helposti saatavilla huomattavasti suorituskykyisempiä ja näyttävämpiä pelejä esimerkiksi Steam-alustan kautta. (Reddit 2022.)

Opetuksellisessa ja tieteellisessä tarkoituksessa selainpohjaisia 3D-visualisointeja on tehty havainnollistamaan, helpottamaan ja esittämään opetettavaa tai tieteellistä materiaalia. Lapsille on suunnattu erilaisia opetuksellisia pelejä ja sisältöjä kun taas muussa opetuksellisessa ja tieteellisessä käytössä on tehty 3D-grafiikkaa hyödyntäviä katselijoita kolmiulotteisten käsitteiden ja asioiden havainnollistamiseen.

Kuva 4. Kuvakaappaus Interland -pelistä (Google 2025), joka opettaa lapsille verkkoturvasta

Opetuksellisessa ja tieteellisessä käytössä 3D-selaimiin pohjautuvasta reaaliaikaisesta renderoinnista voi olla hyötyä sen helposta saavutettavuudesta, interaktiivisuudesta, opetettavan materiaalin pelillistämisestä sekä vaikeiden asioiden kuten molekyylirakenteiden havainnollistamisesta tai sydämen kolmiulotteisesta rakenteesta. 3D-graafikon rooli on tieteellisten ja opetuksellisten konseptien selkeä visualisointi tutkijoiden opastuksella ja esimerkiksi fotogrammetria-mallien optimointi reaaliaikaista renderointia varten.

2.3 Interaktioita ja tehokeinoja

On useita tapoja esittää 3D-visualisointeja verkkoympäristön renderointipuolella. Tehokeinot toteutetaan pitkälti ohjelmoimalla, mutta malliin voi sisällyttää animaatioita ja muita ominaisuuksia, joita voidaan hyödyntää renderoijassa.

Yksi verkkosisällölle tyypillistä 3D-visualisointia on sivuston vieritykseen sidottu animaatio, jossa animaatiota toistetaan käyttäjän vierityksen mukaan. Animaatio etenee usein käyttäjän vierittäessä sivustoa joko mallissa määriteltyjen leikkeiden mukaan tai ruutu kerrallaan. Vierittäessä käyttäjän taustalla voi vaikka tapahtua tekstiin liittyviä, havainnollistavia 3D-visualisointeja. Tätä varten on luotu esimerkiksi ScrollControls -komponentti React -kehitysympäristöön (Meta 2025) osana React Three Fiber -renderoijan Drei -komponenttikirjastoa (PMND 2025).

Yksi yleisimpiä kolmiulotteisissa tuote- ja malliesittelyissä olevia interaktioita on mallien pyörittely. Tällaisissa ja muissa interaktiivisissa kokemuksissa on otettava huomioon sen toimivuus eri laitteilla. Yleisesti tuotetta pyöritellään pyyhkäisy-toiminnolla mobiililaitteilla ja hiiren painalluksilla tietokoneilla. Moni tuotekatselija hyödyntää myös kosketusvarjoja (Contact Shadows) tuoden tuotteelle lattiaan kiinnittyneen vaikutelman. Ympäristövalot (Environment map) ja heijastukset asetetaan renderointipuolella, jotta esimerkiksi PBR-materiaalien metallisuus näkyisi. Three.js -koodikirjasto sisältää OrbitControls -lisäosan, joka sisältää tämän toiminnallisuuden (Three.js 2025).

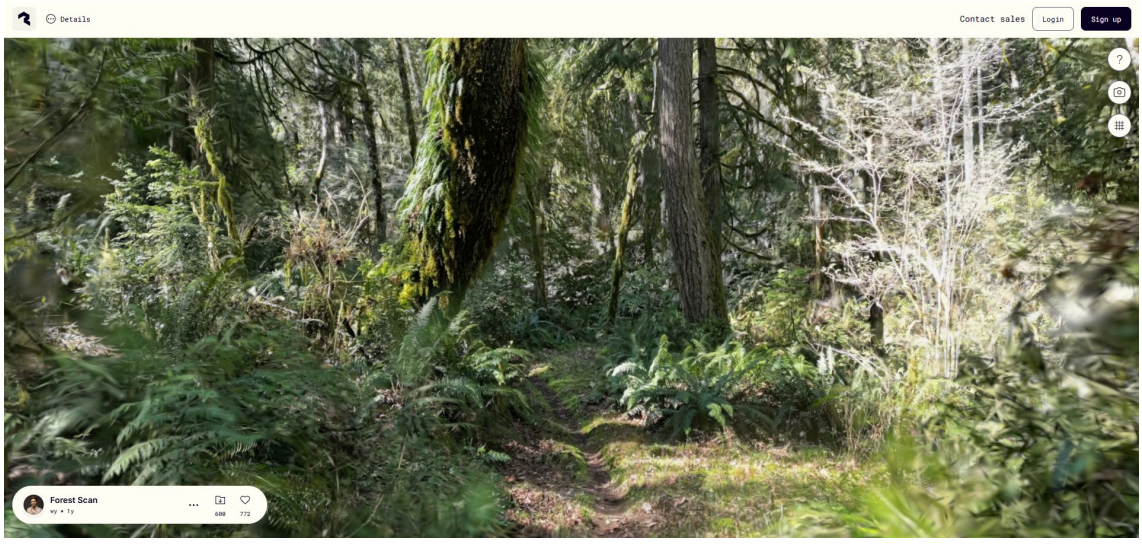
Virtuaalitiloissa liikkuminen yleensä tapahtuu ensimmäisestä persoonasta, jolloin mobiililaitteella käyttäjä voi liikkua koskettamalla lattiaa, kun taas tietokoneella voidaan hyödyntää hiirtä katseluun ja näppäimistöä liikkumiseen kosketustoiminnon lisäksi. On mahdollista myös hyödyntää mobiililaitteen gyroskooppia kameran liikutteluun, jolloin laitetta fyysisesti käänneltämällä voidaan tarkastella tilaa. Esimerkiksi Babylon.js -web-renderointimoottorissa on laaja tuki erilaisille kamera-asetuksille (Babylon.js 2025).

Lisäksi on paljon muita erilaisia ratkaisuja, joissa 3D-visualisoinnin tehokeinoja on hyödynnetty. Niiden toimiminen on 3D-visualisoinnin ja ohjelmoinnin yhteistyöllä saavutettu kokonaisuus, jossa rajana on graafisen ja teknisen osaamisen lisäksi selaimen suorituskyky ja alustariippumaton käytettävyys eri laitteilla.

2.4 Havaintoja kehityssuunnasta

Merkittävimpiä teknisiä edistysaskelia selainpohjaisen 3D-visualisoinnin saralla on WebGPU (Mozilla 2024.), joka on uudentyyppinen grafiikkarajapinta selaimille. Se mahdollistaa suorituskykyisemmän renderoinnin, joka tarjoaa kehittäjille pääsyn laitteen grafiikkaprosessoriin (GPU, Graphics Processing Unit). Esimerkiksi Babylon.js ja Three.js -kirjastot ovat jo alkaneet tukea WebGPU:ta, ja Unity on lisännyt kokeellisen tuen WebGPU-sisällölle. WebGPU on jo saatavilla useimmissa suosituissa selaimissa, mutta sen laajempi käyttöönotto on vielä työn alla.

Gaussian Splatting on myös mielenkiintoinen ja uusi esitystapa kolmiulotteiselle sisällölle. Toisin kuin perinteiset 3D-mallinnusmenetelmät, jotka rakentuvat polygonimalleista ja tekstuureista, Gaussian Splatting käyttää pieniä läpinäkyviä palloja, ”gaussilaisia pilkahduksia”, esittämään tilaa. (Yurkova 2023.)



Kuva 5. Tältä näyttää Gaussian Splatting -metodilla toteutettu 3D-visualisointi Polycam -yrityksen verkkosivuilla (Wyatt 2024).

Selainpohjainen, reaaliaikainen 3D-sisältö on yleistynyt viime vuosina ja sen esittäminen ei enää vaadi erillisiä lisäosia eikä nykyaikaisilta laitteilta aina suurta laskentatehoa. Teknologioiden kuten WebGL:n ja WebGPU:n kehittyessä voidaan olettaa, että 3D-sisällöt tulevat olemaan yhä tavallisempi osa verkkoympäristöä.

3 Selainympäristöön soveltuvan 3D-mallin rakenne

3.1 gLTF-tiedostomuoto

Suurin osa selaimessa reaaliaikaisesti renderoiduista malleista ovat tällä hetkellä standardiksi muodostunutta gLTF -muotoa (Graphics Library Transmission Format), joka on Khronos Groupin kehittämä avoimen lähdekoodin tiedostomuoto (Khronos Group 2025a). gLTF:n tarkoituksena on tarjota mahdollisimman kevyt, suorituskykyinen ja helposti käsiteltävä ratkaisu 3D-sisällön esittämiseen selainympäristössä. gLTF-tiedostomuoto sisältää tärkeimmät ominaisuudet 3D-sisällön renderoimisen kannalta, mukaanlukien meshit, tekstuurit, materiaalit, kamerat, jotkin noodiasetukset ja joint-animaatiot. Tiedosto on mahdollista pakata binäärimuotoon GLB-tiedostomuodossa (Graphics Library Transmission Format Binary File), jolloin kaikki gLTF-tiedoston tieto on pakattu yhteen tiedostoon.

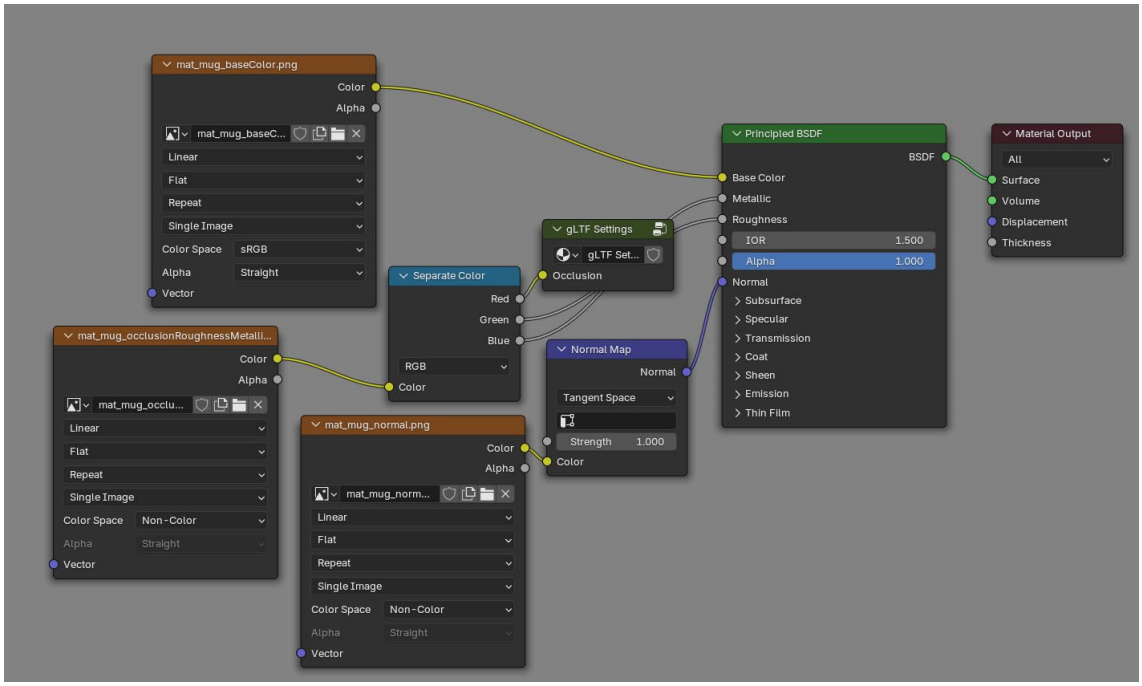
gLTF -tiedostomuoto on hyödyllinen erityisesti silloin, kun projektissa halutaan muokata tai hallita mallin eri osia erillisinä tiedostoina, kuten tekstuureja tai animaatioita. Se on tekstimuotoinen ihmisen luettavissa ja muokattavissa oleva JSON-tiedosto (JavaScript Object Notation), joka tekee siitä hyvän erityisesti kehitysvaiheeseen. gLTF-tiedoston voi avata esimerkiksi tekstieditorilla ja päästä käsiksi 3D-sisällön transform-arvoihin, animaatioihin ja materiaaliasetuksiin. gLTF -tiedosto on siis pohjimmiltaan JSON-tekstiä, joten usein tekstuurit ja mallin binääritiedot, kuten verteksit, normaalit ja UV-koordinaatit, ovat erillisinä tiedostoina (.jpg, .bin jne.). Erilliset tiedostot voidaan upottaa tarvittaessa gLTF-tiedostoon, mutta se vaikuttaa merkittävästi tiedoston kokoon, sillä tiedostot täytyy tässä tapauksessa muuttaa JSON-tekstimuotoon.

Kuva 6. Kuvakaappaus glTF-tiedoston osasta Microsoft Visual Studio Code -ohjelmassa. gLTF-tiedoston 3D-sisältöön pääsee JSON-tekstimuodossa helposti käsiksi esimerkiksi koodieditorilla. Kuvassa näkyy tietoja mallin materiaalista, mesheistä ja kuvatiedostoista.

GLB-tiedostomuoto sen sijaan soveltuu paremmin valmiin mallin jakamiseen tai upottamiseen verkkosivuille, sillä se sisältää kaikki mallin tiedot yhdessä tiedostossa. Tämä nopeuttaa lataamista, vähentää virheitä kuten puuttuvia tekstuureja ja yksinkertaistaa käyttöä erityisesti tapauksissa, joissa mallin sisäisiin osiin ei tarvitse päästä erityisen laaja-alaisesti käsiksi. Koska GLB on binäärimuotoinen, se on myös pienikokoisempi ja nopeampi ladata kuin JSON-muotoinen .gLTF -tiedosto erillisine tekstuurineen ja tiedostoineen. GLB-tiedostomuoto ei ole kuitenkaan destruktiivinen ja samoihin tietoihin pääsee tarvittaessa käsiksi kuin gLTF-formaatissa. Jos tarvitaan erityistä muokkausta tai käsittelyä, on mahdollista muuttaa GLB-tiedosto JSON-muotoon erityisillä työkaluilla. On hyvä ottaa huomioon, että binääritiedostojen kuten GLB :n sisäisiin komponentteihin pääsee yhä käsiksi monessa tapauksessa, joten gLTF-muotoa kannattaa käyttää vain silloin, kun siitä on hyötyä esimerkiksi sisällön dynaamisessa päivittämisessä tai mallista halutaan syystä tai toisesta ihmisen ymmärtämä formaatti jota voidaan muokata käsin. (Mazing 2024.)

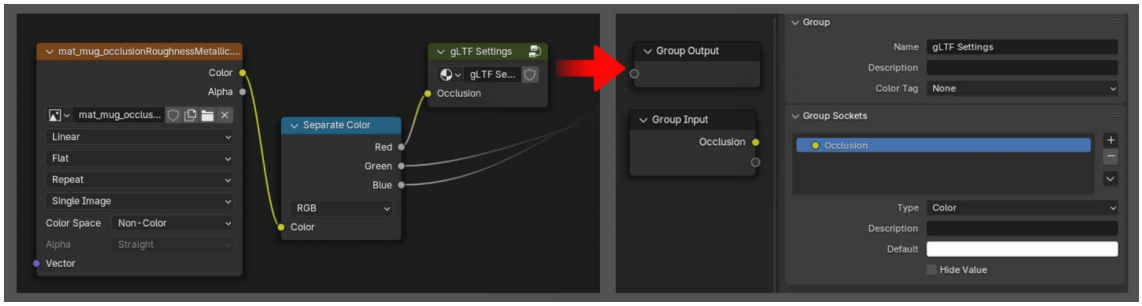
3.2 Materiaalit

gLTF-tiedostomuoto tukee PBR-materiaalien (Physically Based Rendering) vientiä 3D-sisältöihin Metallic/Roughness -työnkulun kautta. Malliin on mahdollista tuoda värikartta (Base Color), metallisuuskartta (Metallic), karkeus (Roughness), Ambient Occlusion -kartta, Normal map -kartta, läpinäkyvyys sekä valokartta (Emissive). Malli tällaisella materiaalilla on vietävissä ulos esimerkiksi Blender -ohjelmasta. Tyypillisesti värikartta on omana tekstuurinaan, joka tulkitaan sRGB-väriavaruudessa. Metallic, Roughness ja Ambient Occlusion -kartat ovat yleensä yhdistetty yhteen tekstuurikarttaan optimointisyistä, sillä ne tarvitsevat toimiakseen vain yhden värikanavan. Tämä on tulkittava väriavaruudessa lineaarisena (raw) tai ei-värillisenä (non-color) tai muuten sen arvot ovat lopputuloksessa väärin. Metallic on tyypillisesti sinisessä värikanavassa, Roughness vihreässä ja Ambient Occlusion punaisessa. Roughness- ja Metallic-kartat voidaan Blenderissä erotella Separate Color -noodin avulla ja yhdistää Principled BSDF-shader noodiin. (Blender Foundation 2025a.)



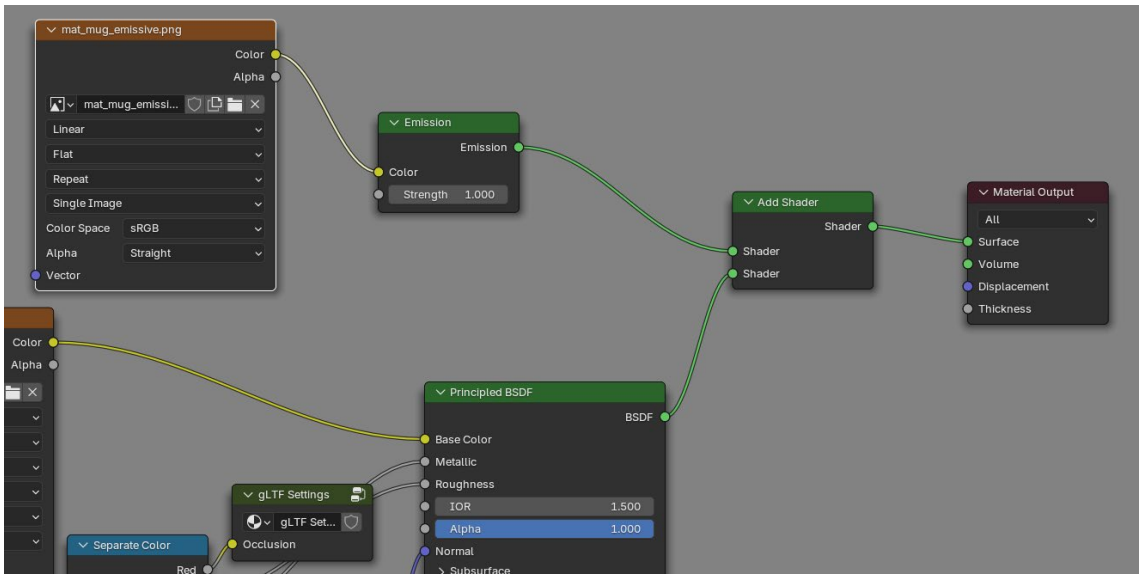
Kuva 7. Node-asetukset PBR-materiaalia sisältävälle mallille, joka on valmisteltu gLTF-exportointia varten.

Tällä hetkellä Blenderissä ei ole esiasetusta Ambient Occlusion -kartan esittämiseen, mutta mikäli glTF exporter -toiminto löytää Occlusion -nimisen sisääntulon (input), se käyttää sitä vientivaiheessa. Tämä onnistuu luomalla tyhjän node -ryhmän (Node Group), josta poistaa sisä- ja ulostulot ja lisää Group Input -noodiin Occlusion -nimisen sisääntulon. Tämä ei näy Blenderin näkymässä, mutta on mukana gLTF-tiedostossa ja näkyy esimerkiksi erilaisissa internetistä löytyvissä glTF -katselijoissa.



Kuva 8. Kuva näyttää Ambient Occlusion -kartan noodiasetukset Blenderissä. glTF Settings -niminen noodi sisältää Occlusion -sisääntulon.

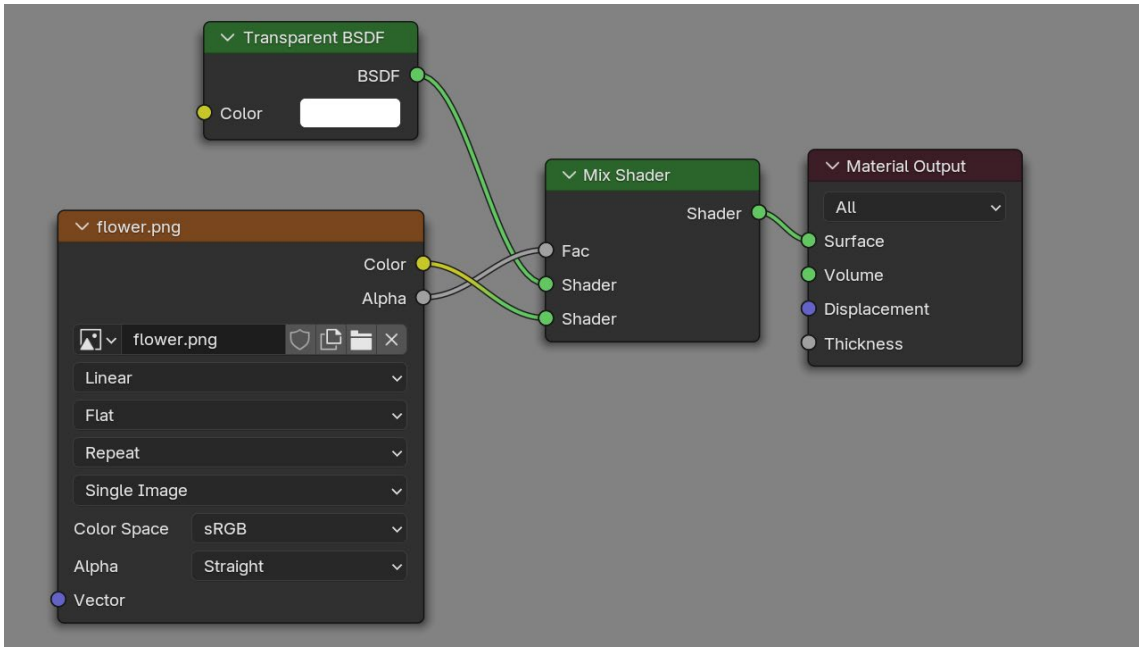
Emissive -karttaan tarvitaan myös mukautettu shader -asetus. Tällöin tarvitaan Principled BSDF -materiaalin rinnalle Emission -noodi, jonka Color -sisääntuloon syötetään emissive -kartta. Emission- ja Principled BSDF-noodit yhdistetään Add Shader -noodissa. (Blender Foundation 2025.)



Kuva 9. Kuvassa Emissive-kartan sisältävän PBR-materiaalin noodiasetukset Blenderissä.

gLTF-muodossa voidaan esittää myös PBR-materiaalin läpinäkyvyyttä (alpha) sekä renderoida materiaalit yksi- tai kaksipuolisesti. Tekstuurikartan alpha-kanava tulee yhdistää Principled BSDF -noodin Alpha -kanavaan ja asettaa materiaalin renderointiasetuksista materiaali Alpha Blended -asetukselle eri asteista läpinäkyvyyttä varten tai Alpha Clip -asetukselle, jolloin jokin tekstuurikartan alue joko on tai ei ole täysin läpinäkyvä. Alpha Blend -asetus saattaa aiheuttaa renderointiongelmia useiden osittaisesti läpinäkyvien materiaalien kanssa, esimerkiksi takana olevat objektit saattavat renderoitua edessä olevien eteen.

Unlit -materiaaleja voidaan viedä gLTF -muotoon sekä läpinäkyvyydellä että ilman. Esimerkiksi Blenderin Background -tyyppinen materiaali toimii hyvin tekstuurikartan esittämiseen, jossa ei ole läpinäkyvyyttä. Tässä tapauksessa tekstuurin värikanava syötetään Background -noodin Color -sisääntuloon. Unlit -materiaali läpinäkyvyydellä voidaan käyttää Mix Shader -noodia, jonka ylempään Shader -sisääntuloon liitetään Transparent BSDF -noodi ja alempaan läpinäkyvyyttä sisältävän kuvatiedoston Color -arvo. Kuvatiedoston alphakanava syötetään Mix Shaderin Fac -sisääntuloon, kunnes lopulta Mix Shader -noodin Shader -ulostulo yhdistetään Material Output -noodin Surface -sisääntuloon.

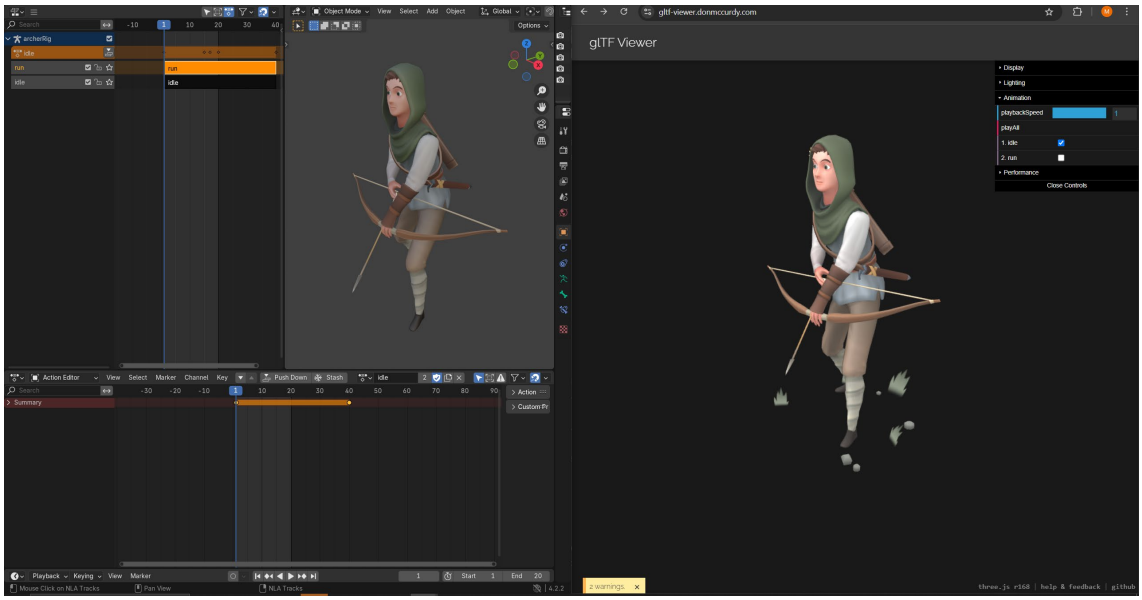


Kuva 10. Noodiasetus Unlit Alpha -materiaalin luontiin

Specular/Glossiness workflown mukaiset PBR -materiaalit ovat myös osittain tuettu Blenderin vientimahdollisuuksissa. glTF-formaatti tukee perinteisten PBR-materiaalien lisäksi myös laajennettuja ominaisuuksia, kuten clearcoat-, transmission- ja sheen-efektejä. Näiden hyödyntäminen edellyttää kuitenkin, että renderointipuolella on erikseen toteutettu tuki laajennuksille.

3.3 Animaatiot

gLTF muotoon voidaan sisällyttää objektien lokaatio-, rotaatio- ja skaalausanimaatioita (transform), luurankoanimaatioita (skeletal animation) ja Shape keys -animaatioita, joita käytetään esimerkiksi hahmojen ilmeisiin. Animaatiot tallennetaan gLTF -tiedostoon leikkeinä (clip). Blenderissä käytetään NLA Tracks -ominaisuutta erottelemaan animaatioleikkeet toisistaan ”Push Down” -toiminnon avulla.



Kuva 11. Vasemmalla näkyy malli kahdella NLA Track -animaatiolla Blenderissä ja oikealla sama malli vietynä nimettyine animaatioineen Don McCurdyn gLTF -katselijaan (McCurdy 2024).

Materiaalin ominaisuuksien animointia, kuten värejä tai läpinäkyvyyttä, ei voida viedä Blenderistä gLTF-muotoon, vaan sen tyyppiset animaatiot ohjelmoidaan. Toistaakseen animaatioita Web3D-ympäristössä täytyy ne määritellä

renderoijassa. Animaatioita käsitellään selainpuolella leikkeinä, joita voidaan toistaa tai pysäyttää, nopeuttaa tai hidastaa, sulauttaa toisiinsa (blend) tai toistaa tietystä kohdasta. Animaatioleikkeiden (Animation Clip) keyframejen aika- ja transform-arvoja voidaan myös tarvittaessa muokata. (Three.js 2025.)

3.4 Valot ja kamera

glTF -tiedostoon pystytään sisällyttämään tietoa valaistuksesta ja kamerasta (Khronos Group 2025a), jota voidaan hyödyntää myöhemmin. Tiedostomuoto mahdollistaa tällä hetkellä kolmen tyyppisten valojen viennin: Directional Light, Point Light ja Spot Light. Näillä voidaan määritellä esimerkiksi mallin perusvalaistus. Usein valaistus kuitenkin tehdään selainpuolella, jossa valojen ominaisuuksia voidaan hallita vapaammin. Valojen viennissä kannattaa myös huomioida, että kaikki katselijat eivät tue glTF:n valo-ominaisuuksia oletusarvoisesti. Renderoijaan on mahdollista tuoda myös beikattu valaistus tai tehdä itse beikkaus renderoijassa, mikäli mallissa on valaisua varten tehty UV-kartta.

glTF-formaatti tukee kameratietojen vientiä, ja tiedostoon voidaan tallentaa joko Perspective- tai Orthographic-kamera. Tämä voi olla hyödyllisiä esimerkiksi valmiiksi määritellyn katselukulman tallentamiseen tai tietyn sommittelman säilyttämiseen. Kuten valaistuksenkin kohdalla, käytännössä kameran toiminnallisuus määritetään usein renderoijassa ohjelmallisesti. Silloin voidaan hyödyntää esimerkiksi käyttäjän ohjattavissa olevaa kameraa, kuten OrbitControls-komponenttia (PMND 2025), jossa mallia voidaan selaimessa pyöritellä.

3.5 Optimointi ja nimeäminen

Mallien optimointi on suositeltavaa olla pitkälti samalla tasolla mobiililaitteille suunnattujen sisältöjen kanssa. Sen lisäksi että verkkosivuilla vieraillaan useimmiten mobiililaitteilla (StatCounter 2025), selaimilla on rajallinen muisti ja muita rajoituksia. On hyvä ottaa erityisesti tekstuurikoissa ja polygonimäärissä

huomioon sivustojen latausajat. Myös esimerkiksi Draw call -kutsujen määrä 3D-sisällössä on hyvä pitää mahdollisimman vähäisenä. Googlen kehittämä avoimen lähdekoodin DRACO-pakkausmenetelmä (Google 2025) mahdollistaa geometrian pakkaamisen yleensä ilman havaittavaa laadun heikkenemistä. Blender tukee DRACO-pakkausta suoraan vientivaiheessa, mutta selaimessa se vaatii erillisen purkutyökalun. DRACO-pakkaus on yksi vaihtoehto, kun pullonkaulana on geometrian suuri määrä, mutta tekstuurikokoihin se ei vaikuta. Tekstuurikokoja voi pienentää Khronos Groupin kehittämällä KTX2-pakkausformaatilla (Khronos Group 2025b), joka mahdollistaa korkealaatuisten tekstuurien käytön pienemmässä tiedostokoossa, mutta tämä vaatii myös erillisen tuen renderoijalta.

Hyvä nimeämiskäytäntö niin mallin sisäisessä rakenteessa (meshit, materiaalit, animaatiot yms.) kuin tiedostonimissä helpottaa projektin hallintaa, kun malli siirtyy muotoilijalta kehittäjälle tai tuotantoon. Yhtenäinen nimeämiskäytäntö parantaa luettavuutta ja nopeuttaa tiedostojen ja mallin eri osien löytymistä. Muotoilijan on hyvä ottaa huomioon, että ohjelmointipuolella mallin navigointi ei ole yhtä helppoa kuin mallinnusohjelmissa eikä visuaalista palautetta ole tekstin lisäksi välttämättä lainkaan, vaikka monet kehitysympäristöihin ladattavat lisäosat tässä auttavatkin. Tiedostonimissä tulee välttää erikoismerkkejä ja välilyöntejä. Yleisiä tapoja nimeämiseen on alaviivan käyttö (esim. material_clay) ja camelCase-käytäntö, jossa tarkentava määritelmä alkaa isolla kirjaimella (esim. materialClay). Tärkeintä on, että nimeämiskäytäntö on yhtenäinen ja helppo ymmärtää. Lisäksi kansiorakenteen jakaminen selkeisiin alikansioihin auttaa pitämään projektin siistinä ja vähentää sekaannuksia.

3.6 Mallin vienti ja testaus

gLTF-formaattiin voi viedä poikkeuksellisen paljon erilaista tietoa mallista, joten on hyvä valmistella malli kehitysympäristöä varten siistimällä malli turhista objekteista, poistamalla ylimääräiset transform-arvot ja rajaamalla vietäväksi vain tarvitut elementit ja tiedot.

Blenderin renderoija ei näytä gLTF-malleja aina oikein eikä havaitse niiden mahdollisia selainpuolen ongelmia. Malleja voi testata erilaisissa gLTF-katselijoissa ennen toteutuksissa käyttämistä havaitaakseen virheet ja mallin oikeanlainen näkyminen. Tällaisia ovat esimerkiksi Don McCurdyn gLTF viewer (McCurdy 2022) ja Khronos Groupin glTF Sample Viewer (Khronos Group 2025c). Sen lisäksi että mallin näkee katselijassa oikein, on hyvä tarkistaa tuottaako se virhesanomia, varoituksia tai muita ilmoituksia ennen kuin sen vie kehitysympäristöön.

Kuva 12. Kuvassa näkyy malli jolla on yksi varoitus ja sen kuvaus Khronos Groupin gLTF -katselijassa (Khronos Group 2025c).

4 Käytännön toteutus: 3D-malli selaimeen

Seuraavassa osiossa tehdään yksinkertainen käytännön toteutus reaaliaikaisesta 3D-visualisoinnista selainympäristössä. Staattinen 3D-malli esitetään tietokoneen selaimessa interaktioin ja valaisuin. Visualisointi tapahtuu paikallisesti joten sen tarkastelu on mahdollista vain käytettävällä tietokoneella http://localhost/ -osoitteessa. Halutessaan sivuston voi avata lähiverkkoon, jolloin sitä pääsee tarkastelemaan samassa verkossa olevilla muilla laitteilla. Sivuston voi myös julkaista verkkoon, mutta tämä esimerkki keskittyy paikalliseen esitykseen. Kehitysympäristönä toimii komponenttipohjainen ReactJS -koodikieli (Meta 2025) sen laajojen esiasetuksien sekä vähäisen koodimäärän vuoksi. Tarkoituksena on saada muotoilijan näkökulmasta lyhyin askelin reaaliaikaista 3D-visualisointia selaimeen sekä havainnollistamaan sen taustalla vaikuttavia teknologioita. Käyttöjärjestelmänä toimii Windows 10. On tärkeää, että asennukset ovat ajan tasalla eikä vanhoja versioita käytetä. Jokaisesta ohjelmasta löytyy LTS (Long Term Support) -versio, joka suositellaan asennettavaksi esimerkkiä varten.

4.1 Vaaditut ohjelmistot ja asennukset

Esimerkkityö vaatii käyttöjärjestelmään asennettavaksi Blender (Blender Foundation 2025b) ja Microsoft Visual Studio Code (Microsoft 2025) -ohjelmat sekä Node.js -ajoympäristön (OpenJS Foundation 2025).

Blender -mallinnusohjelmassa on oletusarvoisesti gLTF- ja GLB -viemismahdollisuus, sekä laaja tuki näiden valmistelulle. Toteutuksessa on käytetty versiota 4.2.2 LTS.

Microsoft Visual Studio Code on suosittu kehitysympäristö ohjelmointiin. Paikallinen verkkosivusto luodaan ja esitetään sen avulla. Visual Studio Code tarjoaa oletuksena riittävät toiminnot React- ja JavaScript-kehitykseen ilman lisäosia. Toteutuksessa on käytetty versiota 1.99.3.

Node.js on ajoympäristö JavaScript-kehitykselle. Suurin osa verkkosivuista käyttää tavalla tai toisella JavaScript -ohjelmointikieltä ja selaimessa tapahtuva reaaliaikainen 3D-visualisointi vaatii sen käyttöä. Suorittaakseen JavaScript -koodia selaimen ulkopuolella on kuitenkin asennettava tietokoneelle tätä tukeva ajoympäristö Node.js. Sen mukana asentuu automaattisesti npm (Node Package Manager), jonka avulla voidaan ladata kehitysympäristöön erilaisia koodikirjastoja.

Windowsin komentorivistä (Command Prompt) voi tarkistaa, onko Node.js ja npm asennettu komennoilla ”node -v” ja ”npm -v”. Esimerkissä on Node.js v. 22.14.0 ja npm versio 10.9.2.

4.2 3D-mallin käsittely

Tässä esimerkissä käytetään opinnäytetyön muissa vaiheissa esiintyvää 3D-mallinnusta kahvimukista. Se on mallinnettu Blenderissä realismia tavoittelevaksi ja käyttää PBR-materiaalia.

Mallin materiaalilla on viisi tekstuuria: Base Color, Metallic, Roughness, Normal Map ja Ambient Occlusion. Ne yhdistyvät PBR-materiaaleja tukevassa Principled BSDF -noodissa.

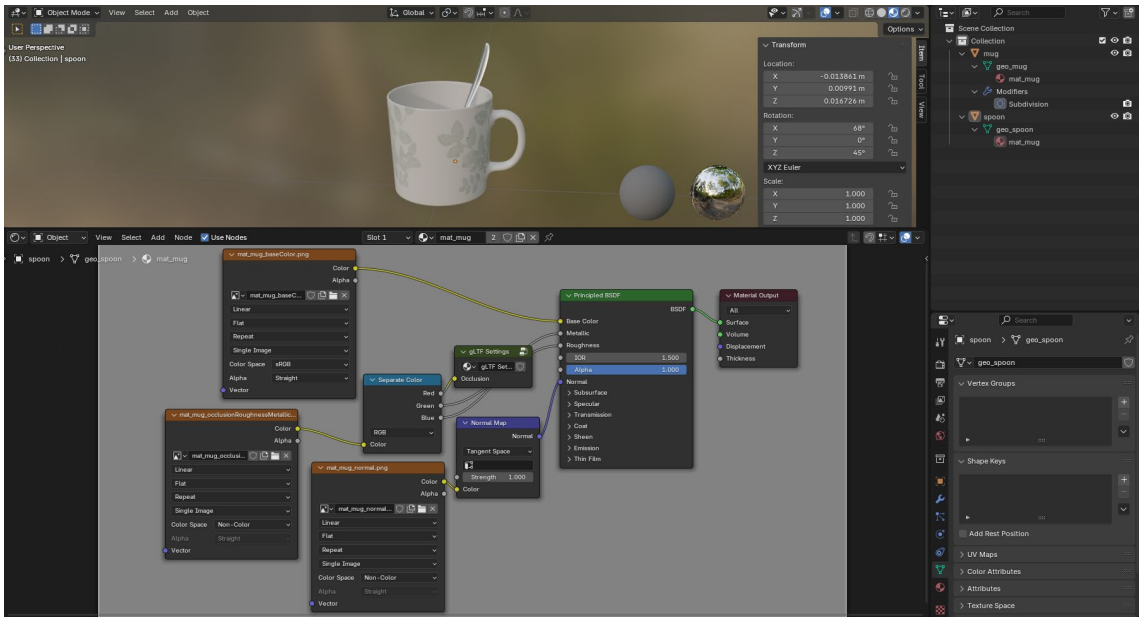
Base Color on omana tekstuurikarttanaan. Se tulkitaan sRGB-väriavaruudessa ja yhdistetään Principled BSDF -noodin Base Color -sisääntuloon.

Ambient Occlusion, Metallic ja Roughness -kartat ovat yhdistetty yhteen kuvatiedostoon, jonka punaisessa värikanavassa on Ambient Occlusion, vihreässä Roughness ja sinisessä Metallic. Tämä kuva tulkitaan ei-värillisessä (Non-Color) väriavaruudessa. Separate Color -noodi erottelee värikanavat omikseen, ja Metallic ja Roughness yhdistetään Principled BSDF:n vastaaviin sisääntuloihin. Ambient Occlusion yhdistetään Node Group -noodiin, joka on nimetty gLTF Settings ja jossa on Occlusion -sisääntulo.

Normal map tulkitaan myös ei-värillisessä väriavaruudessa ja se yhdistetään Normal Map -noodin kautta materiaalin Normal -sisääntuloon.

Malli koostuu kahdesta objektista: keraamisesta mukista ja metallisesta lusikasta. Mukiosan pohja on koordinaatiston nollapisteessä ja sen transform-arvot ovat oletusarvoissa. Lusikalla on lokaatio- ja rotaatio-arvoja. Molempien objektien transform-arvot ovat lopullisessa GLB-tiedostossa luettavissa ja muokattavissa.

Blenderin mittausjärjestelmän mukaan mukin korkeus ja syvyys on noin 8 senttimetriä. Jos malli on liian suuri tai pieni se voi olla näkymättä selaimen renderoijassa, joten mittasuhde-asetukset on hyvä olla kunnossa. Objektit ovat nimetty mug ja spoon, niiden meshit geo_ -etuliitteellä ja materiaali jonka objektit jakavat mat_ -etuliitteellä. Objekteilla on Subdivision Modifier -asetus, jonka voi halutessaan olla viemättä GLB-tiedostoon.

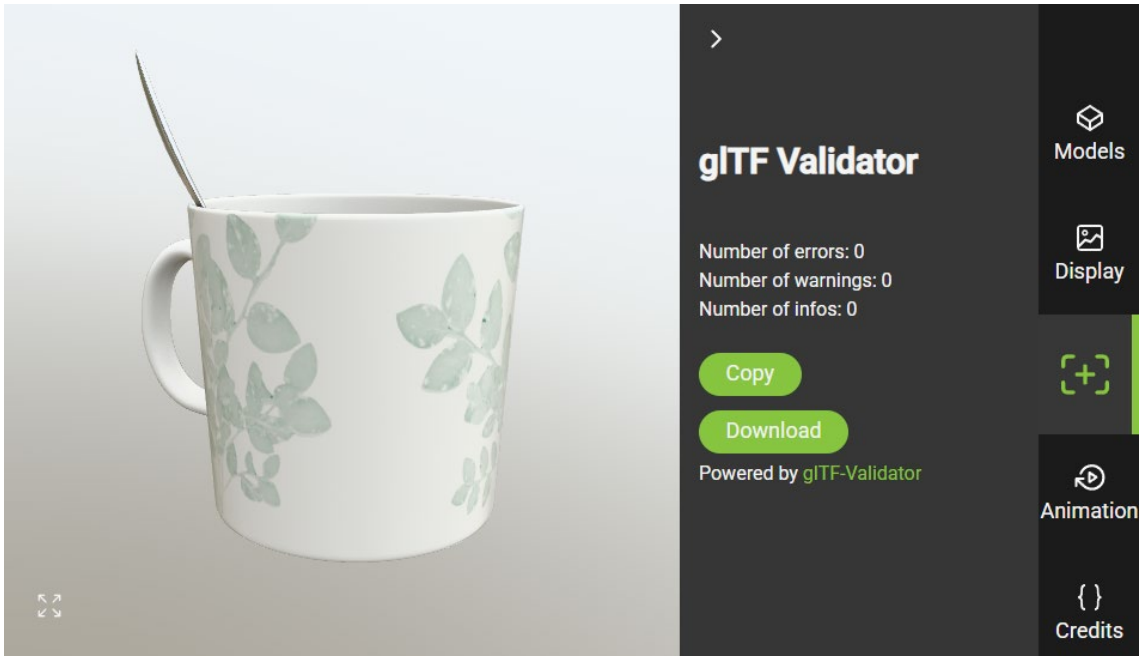


Kuva 13. Kuvassa näkyy materiaaliasetukset, objektihierarkia ja mukin visualisointi ennen sen viemistä .GLB -muotoon

Malli viedään .GLB-muodossa (File -> Export) kokonaisuudessaan ja siihen tarvitaan asetuksissa materiaalin, UV-kartan, tangenttien ja normaalien tiedot. Mikäli haluaa Subdivision Modifier -asetuksen viennissä mukana, on valittava

Apply Modifiers -ruutu Mesh -välilehden alta. Malli olisi mahdollista viedä kompressoidussa muodossa, mutta renderoijaa ei muokata tässä tapauksessa tukemaan ylimääräistä kompressointia.

Vienti tuottaa binäärimuotoisen .GLB -tiedoston, joka testataan Khronos Groupin gLTF Sample Viewer -katselijassa. Malli ladataan gLTF-katselijaan ja tarkastetaan, näyttääkö se oikealta ja tuottaako se virheitä, varoituksia tai muuta tietoa mallista. Mikäli ilmoituksia ei tule, voidaan olettaa että mallissa ei ole virheitä tai erityistä mainittavaa ja sen pitäisi toimia renderoijassa.



Kuva 14. Malli näyttää gLTF -katselijassa oikealta eikä tuota virheitä, varoituksia tai ilmoituksia, joten sen voi viedä kehitysympäristöön.

4.3 Verkkosivun ohjelmointi

Tässä osiossa ohjelmoidaan ja käydään pintapuolisesti läpi, mitä koodissa tapahtuu. Tavoitteena on saada Blenderistä viety .GLB -malli renderoituneena selaimelle. Verkkosivun pohjana toimii Vite (Vite Documentation 2024), joka luo React-applikaation pohjan. Ohjelmointivaiheessa käytetään komentoriviä ja muokataan sekä lisätään koodia Visual Studio Codessa (VSCode).

Ensimmäiseksi VSCodessa avataan projektikansio (Open folder). Tämän jälkeen avataan komentorivi (Terminal -> New terminal), jotta voidaan asentaa React-applikaation pohja ja tarvittavat koodikirjastot.

Varmistetaan komentoriviltä, että kansio on oikea johon asennus tehdään (esim. D:\Mikko\Web3D). Komentoriville kirjoitetaan:

npm create vite@latest

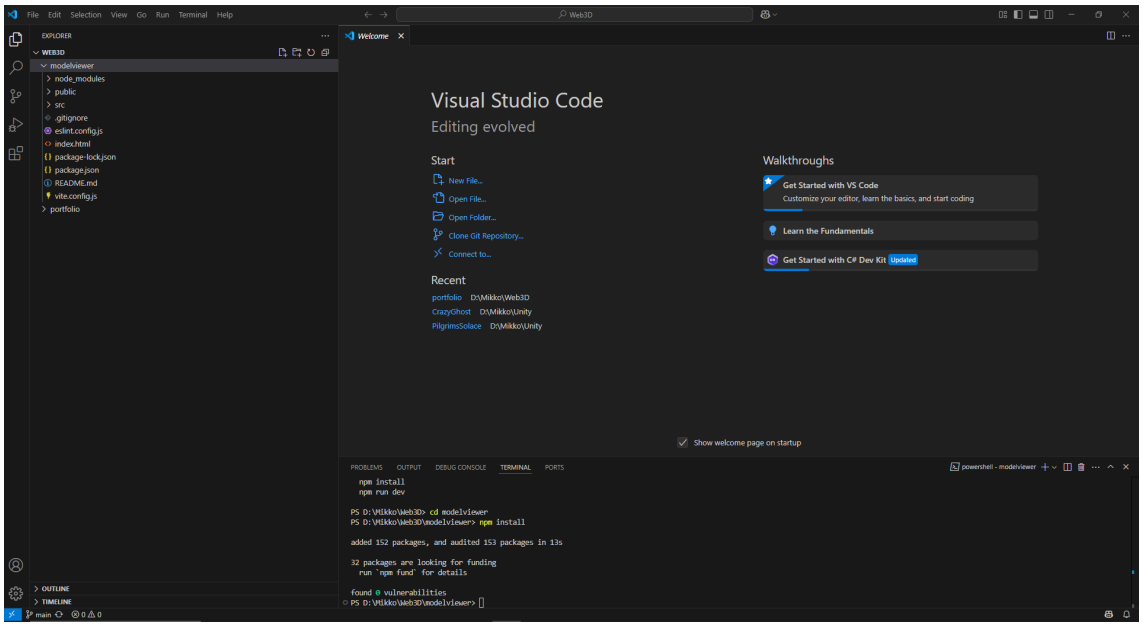
Tämä komento ensiksi kysyy projektin nimeä ja luo sille kansion aktiiviseen kansiorakenteeseen (esim. D:\Mikko\Web3D\). Nimeksi annetaan tässä esimerkissä ”modelviewer”. Seuraavaksi komentorivillä kysytään kehysympäristöä (framework). Nuolinäppäimillä liikkumalla valitaan kohta React ja painetaan Enter. Kolmantena kysytään projektin variant -tyypistä, ja tässä kohtaa valitaan JavaScript.

Tämän jälkeen komentorivillä pitäisi näkyä seuraavaksi ajettavat komennot:

cd modelviewer -komento ohjaa aktiiviseksi kansioksi juuri luodun projektikansion. Mikäli projektin nimi on jokin muu, korvataan modelviewer asianmukaisella kansion nimellä. Tänne asennetaan koodikirjastot ja täällä ajetaan loput komennoista (npm install ja npm run dev).

npm install -komento lataa ja asentaa tarvittavat koodikirjastot ja Viten React-applikaation pohjan.

npm run dev -komento käynnistää verkkosivun selaimeen paikallisesti https://localhost/ -osoitteeseen. Tämä komento pitää erikseen pysäyttää painamalla terminaalissa **CTRL+C** -näppäimiä samanaikaisesti. Kun tämän komennon ajaa tässä vaiheessa, pitäisi näkyä selaimessa localhost -osoitteessa Viten React-applikaatiopohjan oletussivu.



Kuva 15. Näkymä Visual Studio Codesta React-applikaation pohjan lataamisen jälkeen.

Mikäli terminaalissa ilmenee virheilmoituksia, on syytä tarkistaa että Node.js -ajoympäristöstä ja Visual Studio Codesta on ladattu uusin versio. Joitakin varoituksia tai huomioita saattaa npm install -komennon jälkeen näkyä joistakin lukuisista koodikirjastoista, mutta nämä eivät pitäisi vaikuttaa sisällön paikalliseen käynnistämiseen.

Projektikansio koostuu kolmesta kansiosta, joista node_modules kansioon on npm install -komennon myötä ladattu tarvittavat koodikirjastot. public -kansiossa on julkiset kuvat, mallit ja muut, jota lähdekoodi hyödyntää. src -kansiossa on lähdekoodi ja siellä tapahtuu itse ohjelmointi.

Seuraavaksi asennetaan tarvittavat 3D-kirjastot Viten React-applikaation pohjaan seuraavalla komennolla:

npm install three @react-three/fiber @react-three/drei

Tämä komento asentaa projektiin kolme kirjastoa. ThreeJS-kirjasto (three) on 3D renderoijan ydin, joka mahdollistaa 3D-visualisoinnin selaimessa. React Three Fiberin (@react-three/fiber) avulla ThreeJS -kirjastoa voidaan käyttää

React-applikaatiossa. React Three Drei (@react-three/drei) sisältää paljon valmiita 3D-komponentteja, kuten valoja ja kontrolleja.

Seuraavaksi siirretään aikaisemmin tehty .GLB tiedosto projektin public -kansioon. VSCoden Explorer -näkymä näyttää kovalevyllä olevan kansion sisällön, joten mallin voi viedä joko käyttöjärjestelmän puolella oikeaan kansioon tai tiputtaa Explorer -näkymän public -kansioon. Tämän jälkeen navigoidaan App.jsx -tiedostoon ja korvataan kaikki sen sisältö seuraavalla koodilla:

```
import React from 'react'
import { Canvas } from '@react-three/fiber'
import {OrbitControls, useGLTF, Environment, ContactShadows } from '@re-
act-three/drei'

function Model() {
  const gltf = useGLTF('/coffeemug.glb')
  return <primitive object={gltf.scene} />
}

export default function App() {
  return (
     <Canvas camera={{ position: [0, 0.15, 0.3], fov: 60 }} style={{ height:
'100vh', width: '100vw' }}>
      <ambientLight intensity={1}/>
      <directionalLight position={[2, 2, 2]} intensity={1}/>
      <Environment preset="city"/>
      <Model/>
      <ContactShadows scale={1} opacity={0.4}/>
      <OrbitControls/>
    </Canvas>
  )
}
```

4.3.1 Lyhyt selitys koodin osista

Koodin alussa tuodaan App.jsx -tiedostolle käytettäväksi tarvittavat koodikirjastot import -komennolla. React -kirjasto tarvitaan React-applikaation suorittamista varten. React Three Fiber -kirjastosta tarvitaan Canvas -elementti 3D-sisällön esittämiseen. Drei -kirjastosta käytetään OrbitControls, useGLTF, ContactShadows ja Environment -komponentteja. OrbitControls mahdollistaa mallin pyörittelyn, useGLTF mallin lataamisen, Environment tuo mallille ympäristövalaistuksen ja ContactShadows kontaktivarjot.

function Model() -kohdassa luodaan uusi React-komponentti nimeltä Model, joka voidaan sijoittaa Canvas -elementin sisäpuolelle samalla tavalla kuin mikä tahansa muu komponentti.

const gltf = useGLTF('/coffeemug.glb') -kohta käyttää Drei-kirjaston useGLTF-koukkua (hook), joka lataa mallin annetusta polusta. Malli tulee olla public-kansiossa, jotta selain löytää sen.

return <primitive object={gltf.scene} /> -kohdassa gltf.scene viittaa itse 3D-objektin mallihierarkiaan, ja voi sisältää useita asioita kuten kameroita, valoja tai animaatioita. <primitive> -elementti kertoo React Three Fiberille, että kyseessä on 3D-objekti. Tämä tieto palautetaan (return) funktion sisältä, jolloin malli voidaan näyttää osana <Canvas> -elementtiä.

export default function App() -kohdassa suoritetaan selaimelle näkyvä koodi. **<Canvas/>** vie selaimeen HTML Canvas-elementin, jolle piirretään 3D-visualisointi. Canvas sisältää muun muassa renderoivan kameran tietoja ja itse Canvas-elementtiä voi tyylitellä **style** -osiossa CSS-kielellä. Tässä tapauksessa kameraa on tuotu lähemmäksi ja Canvas-elementin kooksi on määritelty 100% selainikkunan korkeudesta (vh, viewport height) ja leveydestä (vw, viewport width).

<ambientLight/> on Drei -kirjastosta tuotu nimensä mukainen ambient -valo. Tämän kirkkautta voi muuttaa vaihtamalla **intensity** -arvoa.

<directionalLight/> on React Three Fiber -kirjaston suuntavalo. **position** -arvo määrittää valon suunnan X-, Y- ja Z-akselilla.

<Environment/> on Drei -kirjaston ympäristövalaisu. Tähän voi ladata joko oman environment map -kuvan tai käyttää komponenttiin sisällytettyjä asiasetuksia. Tässä tapauksessa käytetään city -esiasetusta, joka on HDRI-kuva kaupunkinäkymästä. Se valaisee mallin ja tuo heijastukset metallisuuteen.

<Model/> -kohdassa tuodaan aikaisemmin määritelty funktio **Model()** Canvas -elementin sisälle.

<ContactShadows/> tuo Drei-kirjastosta kontaktivarjot mallille. Tämän avulla mallin alle piirretään varjot ja sen arvoja voi muutella haluamallaan tavalla.

<OrbitControls/> tuo interaktiot malliin ja on myös osa Drei-kirjastoa. Tietokoneella mallia voi pyöritellä painamalla hiiren vasenta näppäintä, liikutella kameraa oikealla näppäimellä ja vierittämällä mennä kauemmas tai lähemmäs mallia.

Drei -kirjastosta löytyy myös useita muita komponentteja 3D-visualisoinnin avuksi. Näitä löytyy Drei-kirjaston virallisesta dokumentaatiosta (Drei.docs 2025).

Halutessaan voi muokata kansiorakenteesta löytyvän index.css -muotoilutiedostossa verkkosivun taustaväriä vaaleammaksi, jotta kontaktivarjot näkyvät selkeämmin. CSS on muotoilukieli, jolla vaikutetaan sivuston väreihin ja käyttöliittymän näkymään. Viten React-applikaation pohjan index.css tiedoston :root -osion background-color -arvon HEX-väriarvo (esim. #242424) muutetaan valkoiseksi korvaamalla se #ffffff -arvolla.

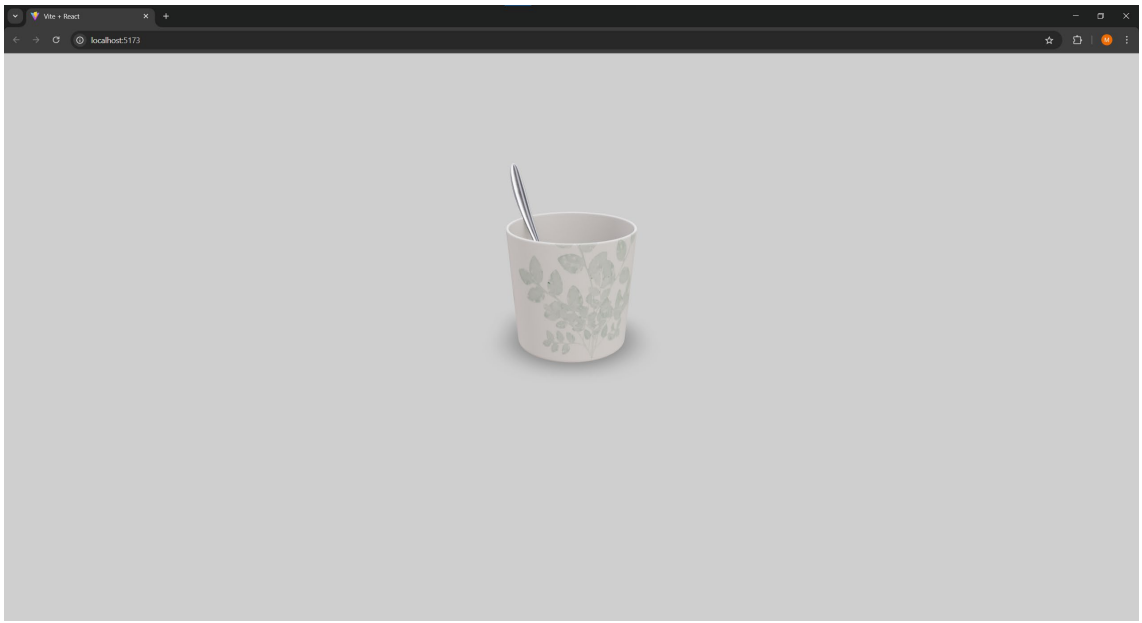
index.css -tiedoston vastaava kohta voidaan korvata seuraavalla koodilla:

```
:root {
  font-family: system-ui, Avenir, Helvetica, Arial, sans-serif;
  line-height: 1.5;
  font-weight: 400;

  color-scheme: light dark;
  color: rgba(255, 255, 255, 0.87);
  background-color: #ffffff; /* tämä HEX-väriarvo vaihdetaan valkoiseksi */

  font-synthesis: none;
  text-rendering: optimizeLegibility;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
```

4.4 Valmis verkkosivu

Nyt esimerkkiprojekti on valmis testattavaksi.

Terminaaliin kirjoitetaan **npm run dev** ja se avaa paikallisen verkkosivun https://localhost/ -osoitteessa. Sen voi sulkea painamalla **CTRL+C** Visual Studio Coden terminaalissa.



Kuva 16. Malli on onnistuneesti renderoitu PBR-materiaaleineen selaimessa.

Kun malli on valmisteltu ja tarvittava ohjelmointi tehty, voidaan nähdä 3D-malli renderoituna selaimessa. Sen asetuksia ja lisäominaisuuksia on helppo muokata React Three Drei -koodikirjaston valmiiden komponenttien avulla. Sivuston voi avata lähiverkkoon muiden laitteiden testattavaksi ajamalla terminaalissa komennon **npm run dev --host**. Tällöin sivustolla voi vierailla laitteella joka on samassa verkossa, esimerkiksi puhelimella. Halutessaan projektia voi jatkokehittää ja tehdä julkiseksi nettiin. Tällaisia hosting-palveluita on tarjolla ilmaisia, kuten Vercel (Vercel 2025).

5 Yhteenveto

Opinnäytetyön ensimmäisessä osiossa käytiin läpi yleinen katsaus reaaliaikaisesta 3D-visualisoinnin taustalla vaikuttavista teknologioista, käyttötapauksista ja kehityssuunnasta. Lyhyt historiaosuus liitettiin siksi, koska selaimessa renderoitu 3D-sisältö ei ole erityisen tunnettu aihe. Käyttötapauksia käytiin läpi havainnollistamaan selainympäristön mahdollisuuksia 3D-visualisoinnille. Kehittyssuunnassa tarkasteltiin Web3D-visualisoinnin kehityksen nykytilaa, työn alla olevaa standardia (WebGPU) ja mahdollisia tulevaisuudennäkymiä eri renderointitekniikoin. Tämä osio antoi minulle laaja-alaisemman näkymän Web3D-sisällön eri ilmiöistä ja mahdollisuuksista. Kehitystyötä tutkiessa ymmärsin enemmän tulevista standardeista sekä erilaisista mahdollisista kehityssuunnista.

Toinen osa on keskittynyt 3D-mallinnusta osaavan muotoilijan työkaluihin, rooliin ja teknisesti oikeanlaisten mallien tuottamiseen Web3D-sisällössä. Oletuksena oli, että lukija ymmärtää entuudestaan 3D-visualisoinnin laajalti, mutta ilman tietämystä reaaliaikaisten mallien luomisesta selainympäristöön. Aiheiden ja asioiden valinnan taustalla oli työelämän tilanteita 3D-graafikkona, jossa Web3D-sisältöä luotiin tiiviissä yhteistyössä ohjelmoijien kanssa. Tähän osioon pyrin liittämään oleellista tietoa oikeanlaisen Web3D-mallin tuottamiseen ja toimittamiseen. Tämä osa oli minulle pitkälti opitun kertausta ja voisi toimia eräänlaisena työkalupakkina mihin palata esimerkiksi jonkun tietyn noodiverkon unohtuessa.

Kolmannessa osiossa tehtiin käytännön toteutus verkkosivusta, jossa pyörii 3D-malli. Ohjeistus laadittiin muotoilijaa silmällä pitäen ja koodin määrä pidettiin vähäisenä tuloksia saadakseen. Tarkoituksena oli ensinnäkin antaa muotoilijalle käytettävä pohja 3D-mallien renderointiin selaimessa, mutta myös havainnollistamaan taustalla vaikuttavia teknologioita ja web-ohjelmoinnin roolia renderoinnissa. Lopputulos on käyttökelpoinen, mutta ei käy läpi monia web-ohjelmoinnin kannalta kriittisiä seikkoja, joten se on tarkoitettu ensisijaisesti havainnollistamistarkoitukseen.

Kolmas osio toimi minulle muistinvirkistyksenä ja voisi auttaa tilanteessa jossa 3D-visualisointia pitäisi tuottaa itsenäisesti verkkosivuille.

Opinnäytetyön teoriaosuuden esimerkeissä ja toteutusvaiheessa käytettiin samaa 3D-mallia kahvikupista yksinkertaistamaan ja yhdistämään sekä opittua että käytäntöä. Opinnäytetyössä tehtyä verkkosivua voi jatkokehittää julkaistuksi verkkosivuksi, erilaisia ohjelmoinnin tehokeinoja voidaan tutkia asianomaisista dokumentaatioista lisää ja teoriaosuudessa opittuja asioita voidaan hyödyntää Web3D-visualisoinnissa.

Kokonaisuudessaan opinnäytetyö antoi minulle laajemman näkemyksen Web3D-visualisoinnin hyödyntämisestä eri tarkoituksiin ja sen kehityksen nykytilasta. Työ voi toimia henkilökohtaisena työkalupakkina tai johdantona muotoilijoille, jotka haluavat ymmärtää tai tehdä Web3D-sisältöä.

Lähteet

Babylon.js 2025. Camera Introduction. Dokumentaatio. https://doc.babylonjs.com/features/featuresDeepDive/cameras/camera_introduction/

Blender Foundation 2025a. gLTF 2.0. Blender Manual. Verkkosivu. https://docs.blender.org/manual/en/2.80/addons/io_scene_gltf2.html

Blender Foundation 2025b. Ohjelmisto. https://www.blender.org/

Byshonkov, Dmitriy 2025. GDC: The State of the Game Industry in 2025. Verkkosivu. https://gamedevreports.substack.com/p/gdc-the-state-of-the-game-industry

GameJobs 2023. 3D Technical Artist. Työpaikkailmoitus. https://gamejobs.co/3D-Technical-Artist-IKEA-Kreativ-Mixed-Reality-at-Geomagical-Labs

Geng Yu, Chang Liu, Ting Fang, Jinyuan Jia, Enming Lin, Yiqiang He, Siyuan Fu, Long Wang, Lei Wei, Qingyu Huang 2023. A survey of real-time rendering on Web3D application. Verkkosivu. https://www.sciencedirect.com/science/article/pii/S2096579622000419

Google 2025. Draco 3D data compression. Verkkosivu. https://google.github.io/draco/

IKEA 2023. IKEA launches new AI-powered, digital experience empowering customers to create lifelike room designs. Verkkosivu. https://www.ikea.com/us/en/newsroom/corporate-news/ikea-launches-new-ai-powered-digital-experience-empowering-customers-to-create-lifelike-room-designs-pub58c94890/

Khronos Group 2025a. glTF: The 3D Asset Delivery Format. Verkkosivu. https://www.khronos.org/gltf/

Khronos Group 2025b. KTX Specifications. Verkkosivu. https://github.khronos.org/KTX-Specification/ (viitattu 20.4.2025)

Khronos Group 2025c. glTF Sample Viewer. Verkkosivu. https://github.khronos.org/glTF-Sample-Viewer-Release/ (viitattu 20.4.2025)

Linux Code 2024. A Blast from the Past – How We Got Here. Verkkosivu. https://thelinuxcode.com/how-to-unleash-the-power-of-webgl-shaders-in-webassembly/

McCurdy, Don 2024. glTF Viewer. Verkkosivu. https://gltf-viewer.donmccurdy.com/ (viitattu 20.4.2025)

Mazing 2024. GLB Vs GLTF. Verkkosivu. https://mazingxr.com/en/glb-vs-gltf/

Meta 2025. React Documentation. Verkkosivu. https://react.dev/

Microsoft 2025. Visual Studio Code. Ohjelmisto. https://code.visualstudio.com/

Mozilla 2025. WebGL: 2D and 3D graphics for the web. Verkkosivu. (viitattu 20.4.2025) https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API

Naif Mehanna, Walter Rudametkin 2023. Caught in the Game: On the History and Evolution of Web browser Gaming. Verkkosivu. https://arxiv.org/pdf/2304.14791

Nike 2025a. Nike By You. Verkkosivu. https://www.nike.com/nike-by-you

Nike 2025b. Senior 3D Modeler. Työpaikkailmoitus. https://careers.nike.com/senior-3d-modeler/job/R-52499 (viitattu 15.4.2025)

OpenJS Foundation 2025. Run JavaScript Everywhere. Verkkosivu. https://nodejs.org/en

PMND 2025. Drei documentation. Verkkosivusto. https://drei.docs.pmnd.rs/

Raggett, Dave 1994. Extending WWW to support Platform Independent Virtual Reality. Verkkosivu. https://www.w3.org/People/Raggett/vrml/vrml.html

Reddit 2022. Verkkosivu. https://www.reddit.com/r/gamedev/comments/umg3ir/why_not_browserbased_3d_games_widely_developed/

Schwarz, Martin 2023. Ikea brings products to Google Search as AR models. Verkkosivu. https://mixed-news.com/en/ikea-products-ar-models-google-search/

Stack Overflow 2024. Developer Survey 2024. Verkkosivu. https://survey.stackoverflow.co/2024/

StatCounter 2025. Desktop vs Mobile Market Share Worldwide. Verkkosivu. https://gs.statcounter.com/platform-market-share/desktop-mobile/worldwide/

Three.js 2025. Three.js Manual. Verkkosivu. https://threejs.org/manual/

W3Schools i.a. WebGL Tutorial. Verkkosivu. https://www.w3schools.com/graphics/webgl_intro.asp

Web3D Consortium 2023. Cultural and Natural Heritage. Verkkosivu. https://www.web3d.org/working-groups/heritage

Yurkova, Kate. 2023. A Comprehensive Overview of Gaussian Splatting. Medium. Verkkosivu. https://medium.com/data-science/a-comprehensive-overview-of-gaussian-splatting-e7d570081362

Kuvalähteet

Kuva 1. Geng Yu, Chang Liu, Ting Fang, Jinyuan Jia, Enming Lin, Yiqiang He, Siyuan Fu, Long Wang, Lei Wei, Qingyu Huang 2023. A survey of real-time rendering on Web3D application. Kaaviokuva. https://www.sciencedirect.com/science/article/pii/S2096579622000419

Kuva 2. Kuvakooste. Kuvat vasemmalta oikealle.

- Nike 2025. Nike By You. Verkkosivu. https://www.nike.com/fi/nike-by-you (viitattu 20.4.2025)

- IKEA 2023. IKEA Kreativ. Verkkosivu. https://www.ikea.com/fi/fi/home-design/ (viitattu 20.4.2025)

Kuva 3. Ritual wine cup (gu) with masks (taotie), dragons, and snakes. Smithsonian Museum 2022. Verkkosivu. https://3d.si.edu/object/3d/ritual-wine-cup-gu-masks-taotie-dragons-and-snakes:77942e15-a113-4e4a-a83e-d881844bdf2e (viitattu 20.4.2025)

Kuva 4. Google 2025. Interland. https://beinternetawesome.withgoogle.com/en_us/interland/ (viitattu 20.4.2025)

Kuva 5. Roy, Wyatt. 3D-skannaus. Polycam. 2024. https://poly.cam/capture/370ef5b0-b01f-40aa-bf95-250cb1a64cb2 (viitattu 20.4.2025)

Kuva 6. Microsoft Corporation. 2025. Visual Studio Code. Ohjelmisto. https://code.visualstudio.com/

Kuva 7. Blender Foundation. 2025. Blender, versio 4.2.2. Ohjelmisto. https://www.blender.org/

Kuva 8. Blender Foundation. 2025. Blender, versio 4.2.2. Ohjelmisto. https://www.blender.org/

Kuva 9. Blender Foundation. 2025. Blender, versio 4.2.2. Ohjelmisto. https://www.blender.org/

Kuva 10. Blender Foundation. 2025. Blender, versio 4.2.2. Ohjelmisto. https://www.blender.org/

Kuva 11. Blender Foundation. 2025. Blender, versio 4.2.2. Ohjelmisto. https://www.blender.org/

Kuva 12. Khronos Group 2025. glTF Sample Viewer. Verkkosivu. https://github.khronos.org/glTF-Sample-Viewer-Release/ (viitattu 20.4.2025)

Kuva 13. Blender Foundation. 2025. Blender, versio 4.2.2. Ohjelmisto. https://www.blender.org/

Kuva 14. Khronos Group. 2025. glTF Sample Viewer. Verkkosivu. https://github.khronos.org/glTF-Sample-Viewer-Release/ (viitattu 20.4.2025)

Kuva 15. Microsoft Corporation. 2025. Visual Studio Code. Ohjelmisto. https://code.visualstudio.com/

Kuva 16. Siivinen, Mikko. 2025. Paikallinen verkkosivu.

Liitteet

Liite 1. Web3D-sisältöä varten luotu 3D-malli .GLB -muodossa ja sen projektitiedostot

Liite 2. Toteutusvaiheessa tehdyn projektin tiedostot.