



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Joona Anttila

3D-mallit web-käyttöliittymässä

3D-mallien käsittely ja niiden hyödyntäminen osana
varastonhallintajärjestelmää

Tietotekniikka
2023

TIIVISTELMÄ

Tekijä	Joona Anttila
Opinnäytetyön nimi	3D-mallit Web-käyttöliittymässä
Vuosi	2023
Kieli	suomi
Sivumäärä	58
Ohjaaja	Kenneth Norrgård

3D-mallien käsittely ja hyödyntäminen tarjoaa monipuolista näkökulmaa ja visuaalisuutta verkkosivulla parantaen käyttökokemuksia. Opinnäytetyön tavoitteena oli tutkia 3D-mallien hyödyntämistä web-käyttöliittymässä erilaisia tekniikoita ja teknologioita hyödyntäen. Tavoitteena oli myös toteuttaa varastomalli 3D-tekniikoita hyödyntäen Pesimal Oy:n hallinnoimaan varastohallintajärjestelmän (engl. Warehouse Management System, WMS) käyttöliittymään. Opinnäytetyössä tutkittiin myös 3D-mallien rakenteita ja toimintaa yleisellä tasolla sekä niiden tuomia haasteita ja mahdollisuuksia. Opinnäytetyö toteutettiin tarkastelemalla eri käyttötarkoituksia, joita verkkosivuilla esiintyy 3D-mallinnuksen käytössä. Käyttötarkoitusten lisäksi tutkittiin tarkemmin tietokonegrafiikan hahmontamista OpenGL- ja WebGL-teknologioilla. Tutkittujen asioiden perusteella mallinnettiin korkeavarastoissa käytettävä kuljetinhissi. Malli tuotiin tiedostosta valmiiseen verkkosivustoon ja sitä muokattiin ja animoitiin Thrjee.js Javascript-kirjastoa hyödyntäen. Mallinnus ja mallin käsittely tuotti omat haasteensa. Mallin koon ja raskauden vuoksi ilmeni renderöintiongelmia, ja mallin kolmiulotteiset ominaisuudet toimivat välillä tosi hitaasti. Mallille oli myös haastavaa löytää oikeaa tiedostomuotoa, joka vastaisi Three.js-kirjaston funktioita mallin tuomiseen verkkosivulle.

Opinnäytetyö osoitti kuitenkin, että käytetyt sovellukset ja tekniikat ovat toimiva kokonaisuus 3D-mallinnukseen verkkosivuilla. 3D-mallinnukseen ja kehyksiin löytyy tarpeeksi ohjeita, joten mallin luominen ja yksinkertaisen 3D-verkkosivun luominen onnistuu myös vasta-alkajalta. Kuitenkin vaikeammat mallintamisen tekniikat, kuten geometriasolmut, vaativat enemmän mallinnustyökaluja.

Avainsanat: 3D-mallinnus, kolmiulotteisuus, tietokonegrafiikka,

web-käyttöliittymä

ABSTRACT

Author Joona Anttila
Title 3D Models in Web User Interfaces
Year 2023
Language Finnish
Pages 58
Name of Supervisor Kenneth Norrgård

Handling and utilizing 3D models provide a versatile perspective and visual appeal to a website, enhancing user experiences. The objective of the thesis was to investigate the utilization of 3D models in a web interface, using various techniques and technologies. The goal was also to implement a warehouse model using 3D techniques in the user interface of the warehouse management system (WMS) managed by Pesmel Oy. The thesis also explored the structures and functionality of 3D models at a general level and the challenges and opportunities they bring. The thesis was conducted by examining different use cases for 3D modeling on websites. In addition to use cases, computer graphics rendering with OpenGL and WebGL technologies was examined in more detail. Based on the research, a conveyor elevator used in high-bay storage was modeled. The model was imported from a file into a ready-made website and was modified and animated using the Three.js JavaScript library. Modeling and handling the model posed their own challenges. Due to the size and complexity of the model, there were rendering issues, and the three-dimensional features of the model sometimes performed very slowly. It was also difficult to find the right file format for the model that would correspond to the functions of the Three.js library for importing the model into the website.

However, the thesis demonstrated that the applications and technologies used are a functional combination for 3D modeling on websites. There are sufficient instructions available for 3D modeling and frameworks, making it possible for even beginners to create a model and a simple 3D website. However, more advanced modeling techniques, such as geometry nodes, require more expertise related to modeling tools.

Keywords: 3D modeling, three-dimensionality, computer graphics, Web User Interface

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KUVA- JA TAULUKKOLUETTELO

LIITELUETTELO

TERMIT JA LYHENTEET

1	JOHDANTO.....	9
2	YHTEISTYÖKUMPPANI	11
2.1	Pesmel Oy	11
2.2	Varastonhallintajärjestelmä.....	11
3	PROJEKTI.....	14
3.1	Tausta.....	14
3.2	Tavoitteet.....	14
4	TEORIATAUSTA	15
4.1	Tuote-esittely	16
4.2	Visualisuuden lisääminen	16
4.3	Taideprojektit ja pelillisuus	19
5	PROJEKTIN LÄHESTYMISTAPA JA TOTEUTUS	21
5.1	2D- ja 3D-grafiikan hahmontaminen	26
5.1.1	OpenGL	26
5.1.2	WebGL.....	27
5.2	Grafiikkakehykset ja -moottorit selaimessa	32
5.2.1	Three.js	32
5.2.2	Babylon.js	36
5.2.3	PlayCanvas	37
6	PROJEKTIN TUOTOKSET	41
6.1	Teknologian valinta.....	41
6.1.1	Vertailukohteet	42
6.1.2	Toiminnallisuus	43

7 YHTEENVETO JA POHDINTA	51
LÄHTEET	53

KUVA- JA TAULUKKOLUETTELO

Kuva 1 Toimitusketju (engl. Supply Chain Management, SCM).	12
Kuva 2 3D-mallien hyödyntämistä Mazdan kotisivuilla.	17
Kuva 3 Esimerkki parallax-efektistä Firewatch videopelin verkkosivustolla.	18
Kuva 4 Interaktiivinen kartta Norjan sademetsärahasen sivustolla.....	19
Kuva 5 Esimerkki normaalikartoituksesta.	22
Kuva 6 NURBS-pinta ja käyrä. Kontrollipisteet on merkitty punaisella.	23
Kuva 7 Leikkaustasojen muodostama alue renderöitynä.	25
Kuva 8 Sivuston rakenne ilman WebGL:ää ja sen kanssa.	28
Kuva 9 OpenGL-renderöintiputki.	30
Kuva 10 Vektori- ja matriisidata.....	32
Kuva 11 Pienen Threejs-sovelluksen rakenne.	33
Kuva 12 Kuution luominen laatikkoprimitiiveillä ja asetetuilla parametreilla.	34
Kuva 13 PlayCanvas editori.	38
Kuva 14 PlayCanvas-äänikomponentti.	39
Kuva 15 Hissimallin rakenne.	44
Kuva 16 Hissin animointi.....	45
Kuva 17 Animointinäkymä Blenderissä.....	45
Kuva 18 Hissimallin rakenne ThreeJs-editorissa.....	46
Kuva 19 Hissimalli verkkosivulla.	47
Esimerkkikoodi 1 Yksinkertainen animaatiokripti PlayCanvasissa.	40
Esimerkkikoodi 2 ThreeJs glTFLoader.....	49
Esimerkkikoodi 3 ThreeJs animaatio.	50

TERMIT JA LYHENTEET

2D	Kaksiulotteinen digitaalinen kuva tai teksti (engl. two dimensional).
3D	Kolmiulotteinen grafiikka, joka on sisäisesti mallinnettu kolmen tilaulottuvuuden suhteen (engl. three dimensional).
B-rep	Rajaesitysmallinnus (engl. boundary representation).
CAD	Tietokoneavusteinen suunnittelu (engl. computer-aided design).
Entiteetti	Yleistermi sille, jonka kielellinen ilmaus voi havaita tai tulkita (engl. Entity).
ERP	Toiminnanohjausjärjestelmä (engl. enterprise resource planning).
Eksportointi	Tiedoston tai datan vieminen esim. jostain sovelluksesta hakemistoon (engl. export).
Frustumi	Kiinteän aineen osa, joka sijaitsee kahden yhdensuuntaisen tason välissä, jotka leikkaavat kiinteää ainetta (engl. Frustum).
.glb	3D-mallien tiedostomuoto, joka sisältää mallin kaikki tiedot (engl. glTF binary).
GLSL	Varjostinkieli visuaalisille tehosteille (engl. OpenGL Shading Language).
glTF	3D-mallien tiedostomuoto (engl. Graphics Language Transmission Format).
Importointi	Tiedoston tai datan tuominen esim. jostain hakemistosta sovellukseen (engl. import).
NURBS	3D-mallinnuksen tapa, jossa käytetään käyriä (engl. Non-uniform rational basis spline).
OpenGL	Ohjelmointirajapinta 2D- ja 3D-grafiikkojen hahmontamiseen (engl. Open Graphics Library).
RFID	Radiotaajuinen etätunnistus (engl. Radio Frequency identification).
SCM	Toimitusketjun hallinta. Tavaravirtojen optimointia ja hallintaa (engl. Supply Chain Management).

Tesselaatio	3D-grafiikassa kappaleen pinnan jakamista pienempiin osiin (engl. Tessellation).
WebGL	JavaScript-ohjelmointirajapinta 2D- ja 3D-grafiikkojen hahmontamiseen verkkosivuilla (engl. Web Graphics Library).
WMS	Varastonhallintajärjestelmä (engl. Warehouse Management System).

1 JOHDANTO

Web-käyttöliittymien kehittäminen ja modernisointi on tärkeässä osassa tulevaisuudessa. 3D-mallien hyödyntäminen ja oikeanlainen käyttäminen voi tarjota tehokkaita työkaluja ja toiminnallisuuksia visualisoinnin parantamiseksi ja kokonaiskuvan hahmottamiseksi.

Tällä hetkellä tietokoneohjelmistojen ja suorituskyvyn jatkuvan kehittymisen ansiosta 3D-ympäristöjen luominen ja käyttäminen on kasvattanut suosiotaan merkittävästi. Nykyään kuka tahansa, jolla on käytössään tietokone ja internetyhteys, kykenee luomaan 3D-objekteja ja esittämään niitä verkkosivuillaan. 3D-mallinnusta käytetään laajasti eri aloilla ja myös yksityishenkilöiden keskuudessa, ja näitä malleja voidaan soveltaa käytännössä mihin tahansa, olipa kyse sitten matkapuhelimista, peleistä tai taideprojekteista.

Opinnäytetyön toimeksiantaja on Pesimal Oy, jonka varastohallintajärjestelmä (engl. Warehouse Management System, WMS) on käytössä maailmanlaajuisesti monella eri yrityksellä. Opinnäytetyön tavoitteena on tutkia eri 3D -tekniikoita ja niiden erilaisia mahdollisuuksia. Tarkoituksena on toteuttaa varastohallintajärjestelmän käyttöliittymään korkeavarastomalli, joka myöhemmässä vaiheessa mallintaisi varastossa tapahtuvan hissien ja lastaustavaran liikkeitä. Tämän käyttöliittymän avulla tavaran seuraaminen on melko vaivatonta ja koko varastonäkymä antaa parempaa ja visuaalisempaa perspektiiviä koko varaston toiminnasta. Tämä sovellus tarjoaa täysin uudenlaisen näkökulman verrattuna aiempaan 2D-ympäristöön.

Opinnäytetyössä tarkastellaan lisäksi 3D-mallien luontia ja tietokonegrafiikan hahmontamista tarkemmin grafiikkakehyksien sekä avoimen grafiikkakirjastojen (engl. Open Graphical Library, OpenGL) ja verkkografiikkakirjastojen (engl. Web Graphical Library, WebGL) kautta.

Työssä käydään lyhyesti läpi 3D-mallinnukseen liittyvää teoriaa, perusteita ja menetelmiä. Työssä vertaillaan myös erilaisia 3D-tekniikoita, joista on tarkoitus saada syvempi ymmärrys siitä, mitä ominaisuuksia nämä eri palvelut tarjoavat. Tavoitteena on löytää parhaiten soveltuvat tekniikat ja työkalut työn optimaalisen tuloksen saavuttamiseksi.

2 YHTEISTYÖKUMPPANI

Luvussa käsitellään toimeksiantajaa, Pesmel Oy:tä, joka on suomalainen konepajayritys. Lisäksi luvussa käsitellään varastohallintajärjestelmää (WMS), johon opinnäytetyössä käsiteltyä 3D-mallinnusta ja sen tekniikoita sovelletaan.

2.1 Pesmel Oy

Pesmel Oy on suomalainen konepajayritys, joka valmistaa automatisoituja materiaalinkäsittely-, lastaus- ja pakkausjärjestelmiä metalli- ja paperiteollisuuden tarpeisiin. Ytimessä on Material Flow How-konsepti, joka asiakkaiden koko tarpeiden kirjon älykkästä asiakaslähtöisestä suunnittelusta aktiiviseen elinkaarihuoltoon ja kunnossapitoon. Älykkäiden pakkaus- ja käsittelyratkaisujen oikea yhdistelmä sekä automaattinen korkeapaikkainen varastointi on räätälöity vastaamaan asiakkaan erityistarpeita ja vaatimuksia.¹

2.2 Varastohallintajärjestelmä

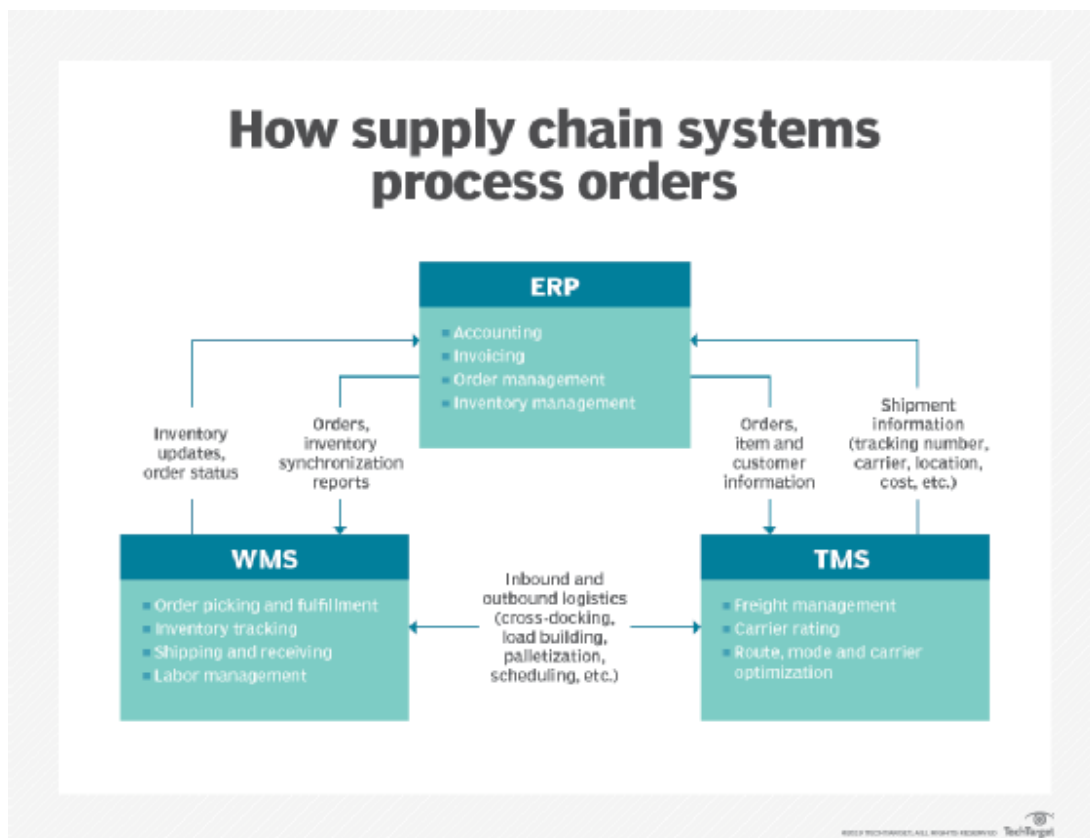
Varastohallintajärjestelmä (engl. Warehouse Management System, WMS) koostuu ohjelmistoista ja prosesseista, joiden avulla organisaatiot voivat hallita ja hallinnoida varastoa materiaalien liikkeessä sisään varastoon ja sieltä ulos.²

Varastot ovat tuotannon ja toimitusketjun toimintojen keskiössä, koska niissä on kaikki prosesseissa käytetty tai tuotettu materiaali raaka-aineista valmiisiin tuotteisiin. WMS:n tarkoituksena on auttaa varmistamaan, että tavarat ja materiaalit liikkuvat varastojen läpi tehokkaimmalla tavalla. WMS käsittelee monia toimintoja, jotka mahdollistavat nämä liikkeet, mukaan lukien varaston seuranta, keräily, vastaanotto ja pakkaaminen.²

¹ Pesmel Oy. 2023. About us.

² Techtargget. 2020. WMS.

Toimitusketju voi toimia vain niin nopeasti, tarkasti ja tehokkaasti kuin varastoprosessit sen sallivat. WMS:llä on tärkeä rooli toimitusketjun hallinnassa, sillä se hallitsee tilausten toteutusprosesseja raaka-aineiden vastaanottamisesta valmiiden tuotteiden toimittamiseen. Jos raaka-aineita ei vastaanoteta kunnolla tai osia on väärässä varastossa, toimitusketju voi hidastua tai katketa. WMS-järjestelmät ovat kriittisiä näiden prosessien sujuvan toiminnan varmistamisessa seuraamalla varastoa ja varmistamalla, että tavarat varastoidaan ja lajitellaan oikein sekä toimitetaan ja seurataan tarkasti.²



Kuva 1 Toimitusketju (engl. Supply Chain Management, SCM).²

Jokaisella kolmella toimitusketjun (SCM) haaralla on erityinen rooli tilausten hallinnassa ja tietojen jakamisessa kahden muun kanssa. WMS hallitsee tavaran liikkumiseen varaston kautta liittyviä tietoja ja prosesseja.²

Kuva 1 on esitetty kunkin järjestelmän rooli tilausten käsittelyssä ja millaisia tietoja niiden on jaettava keskenään varmistaakseen, että oikea tuote saapuu asiakkaalle ajoissa.²

WMS:n tehtävänä on tukea käyttäjiä hallinnoimaan varaston tai jakelukeskuksen toimitus-, lähetys- ja vastaanottotoimintoja, kuten tavaroiden noutamista hyllyiltä lähettämistä varten tai vastaanotettujen tavaroiden sijoittamista jatkokäsittelyyn. Sen tehtävänä inventoinnissa on seurata viivakoodinlukijoista ja radiotaajuustunnistus (engl. Radio Frequency Identification, RFID) -tunnisteista tulevia varastotietoja ja päivittää ERP-järjestelmän varastohallintamoduuli varmistaakseen, että sillä on uusimmat tiedot. Integrointilinkki synkronoi ERP-järjestelmään ja WMS:ään tallennetut varastotiedot.²

Vaikka WMS on monimutkainen ja kallis toteuttaa ja käyttää, organisaatiot saavat monia etuja, jotka voivat olla merkityksellisempiä kuin monimutkaisuus ja kustannukset.²

WMS:n käyttöönotto voi auttaa organisaatiota vähentämään työvoimakustannuksia ja virheitä tavaroiden keräilyssä, sekä lähettämässä. Se voi auttaa parantamaan varaston tarkkuutta, joustavuutta ja reagointikykyä ja myös asiakaspalvelua. Nykyaikaiset varastohallintajärjestelmät toimivat reaaliaikaisilla tiedoilla, jolloin organisaatio voi hallita uusinta tietoa toiminnoista, kuten tilauksista, lähetyksistä, kuiteista ja tavaroiden liikkeistä.²

3 PROJEKTI

Nykypäivän teollisuus ja digitaalinen maailma eivät tule toimeen ilman 3D-mallinnusta. 3D-suunnittelua käytetään mm. elokuvateollisuudessa, lääketieteessä, arkkitehtuurissa, tekniikassa, videopeleissä, tietokoneavusteisissa ohjelmistoissa (engl. Computer-aided Design, CAD). Alue, joka on edelleen kattamaton ja melko suurilta osin tuntematonkin 3D-alusteisissa ratkaisuissa, on Internet. Suurin osa verkkosivustoista on tasapaksuja, usein 2D-teknologiaa käyttäviä sivustoja, 3D-ominaisuuksilla varustettu verkkosivusto erottuu heti joukosta.³

3.1 Tausta

Pesmel Oy tarjoaa korkeavarasto- ja pakkauslinjajärjestelmiä (WMS) erilaisille tehdasalan yrityksille. Yksi WMS:n tärkeimmistä ominaisuuksista operaattoreiden kannalta on yleisesti varaston toiminnan seuraaminen. Tämä Web-käyttöliittymän näkymä näyttää käyttäjälle varastossa tapahtuvien laitteiden liikkeitä ja tavaran liikkumista varastosta lastaustelakoille ja toisin päin. Tämä näkymä on kuitenkin hieman suppea, sillä 2D-teknologialla on omat rajoitteensa esimerkiksi syvyysominaisuuksien suhteen. Tästä syntyi idea 3D-maailmaan tutustumiseen ja sen mahdolliseen hyödyntämiseen.

3.2 Tavoitteet

Opinnäytetyön yhteydessä toteutetussa projektissa oli tavoitteena suunnitella ja toteuttaa Web-käyttöliittymään näkymä, joka mahdollistaa varastojärjestelmän toiminnallisuuden seuraamisen kolmiulotteisella tasolla. Näkymä toteutettiin osaksi Pesmel Oy:n käyttämää ja hallinnoimaa varastohallintajärjestelmää (WMS). Projektin tavoitteena oli myös tutkia 3D-teknologian tarjoamaa potentiaalia yleisellä tasolla sekä muodostaa parempi käsitys siitä, miten 3D-teknologiaa voidaan hyödyntää Web-ympäristössä yleisesti.

³ 3dwebvision. 2016. 3d-models-on-websites

4 TEORIATAUSTA

Erilaisia ohjelmistoja ja objekteja hyödyntäen voidaan luoda kolmiulotteisia malleja 3D-mallinnuksessa. Tietokoneavusteinen suunnittelu lähti liikkeelle CAD-ohjelmasta, joka kehitettiin 1960-luvun alussa. Ensimmäisiä CAD-malleja tehtiin kaksiulotteisesti, jossa fyysinen kappale piirrettiin vastaavaksi kuvaksi kuin sinikopio.⁴

Kolmiulotteinen mallinnus tuli tutuksi vasta paljon myöhemmin. Suuren vaikutuksen tähän kehitykseen teki B-rep-mallintajien kehittäminen ja yleistyminen vuonna 1974. B-rep-mallinnus, tai Boundary Representation-mallinnus on CAD-sovelluksissa yleisin mallinnustyyppi. Termi tarkoittaa 3D-objektien esittämistä tarkemmin matemaattiselta pohjalta. Tämä esitys määrittää geometriset rajat kiinteään ja ei-kiinteään geometrian välillä. Tämän takia B-rep-malleja hyödynnetäänkin usein konetekniikassa ja tuotteiden suunnittelussa.⁵

Elämme maailmassa, jossa 3D-mallinnustekniikkaa on kaikkialla. 3D-mallinnus on vaikuttanut suuresti yhteiskuntamme kehitykseen videopeleistä elokuvaan ja arkkitehtuuriin. Videopelit ovat tunnetuin esimerkki 3D-mallinnustekniikasta. Kehittäjät luovat virtuaalisia maailmoja grafiikan ja objektien avulla. Arkkitehtuurissa puolestaan malleja käytetään rakennusten suunnitelmien ja suunnittelukonseptien luomiseen, jolloin visualisointia voidaan hyödyntää ennen varsinaista rakentamista. Anatomiaa pystytään myös visualisoimaan ja tästä on suuri apu esimerkiksi lääketieteessä.⁶ Tässä luvussa kerron tarkemmin 3D-mallinnuksen menetelmistä ja tyyleistä.

⁴ 2023. en.wikipedia.org/wiki/History_of_CAD_software.

⁵ [blog.spatial.com. 2019. Main_benefits_of_B-rep_modeling](https://blog.spatial.com/2019/05/main-benefits-of-b-rep-modeling/).

⁶ facequad.com/blogs/post/what-is-3d-modeling. 2023.

4.1 Tuote-esittely

Erottuakseen joukosta brändien valmistajat pyrkivät käyttämään erinäisiä markkinointitemppuja, jotka on tehostettu tarttuvalla kuvalla. Perinteiset staattiset kuvat eivät ole tarpeeksi houkuttelevia nykyajan asiakkaille. 3D-mallit voivat olla hyvä keino esitellä tuotetta. Staattiset kuvat eivät pysty välittämään liikettä samaan tapaan kuin animoidut 3D-mallit. Mallin ollessa kolmiulotteinen, käyttäjä pystyy kääntelemään tuotetta haluamallaan tavalla ja haluamiinsa suuntiin. Tuotteella voi olla erilaisia animointeja, jotka kuvastavat tuotteen toimintaa tai tuote on myös mahdollisesti purettu eri osiin, jolloin käyttäjä pääsee tarkastelemaan rakennetta yksityiskohtaisemmin.⁷

3D-mallinnuksessa saattaa olla myös kustannustehokkaita hyötyjä. Valokuvaajien palkkaaminen ja aikataulujen sopiminen heidän kanssaan tiettyä kuvauspaikkaa ajatellen on mahdollisesti kalliimpaa kuin 3D-mallien hyödyntäminen. 3D-mallit kulkevat kuitenkin usein käsi kädessä kuvauksiin liittyvissä asioissa. Syynä tähän saattaa olla se, että kuluttajat haluavat nähdä oikean alkuperäisen tuotteen, eikä pelkkää tietokoneella luotua mallia. 3D-mallit ovat käytössä usealla eri alalla ja tuotteet voivat olla lähes mitä tahansa arkipäiväisiä tuotteita. Tämän lisäksi ympäristöjä on mahdollista mallintaa, mitä voidaan hyödyntää esimerkiksi asuntonäyttelyissä.⁷

4.2 Visualisuuden lisääminen

Visuaalisuus on tärkeä tekijä mainonnassa ja sen avulla pystyy erottumaan joukosta pienilläkin efekteillä. Nämä perustuvat etenkin asiakkaiden näköhavaintoihin, jotka kiinnittävät paremmin huomiota liikkuvaan kuin staattiseen kuvaan. Verkkosivulle voidaan lisätä jokin aiheeseen liittyvä malli, mutta kyseinen malli ei ole välttämättä

⁷ cgifurniture.com/3d-animation-for-product-design/2023.

mainostettava tuote. **Kuva 2** autoteollisuuden tunnettu merkki Mazda hyödyntää 3D-malleja verkkosivuillaan.



Kuva 2 3D-mallien hyödyntämistä Mazdan kotisivuilla.⁸

⁸ Mazdausa. 2023.

On olemassa verkkosivuja, jotka tuntuvat dynaamisilta. Näkymä tuntuu siltä kuin selaisi useampaa kuin yhtä näkymää kerralla. Tätä toista menetelmää visuaalisuuden lisäämiseksi kutsutaan parallaksi vieritykseksi (engl. parallax scrolling). Tämä lisää sivustolle syvyyttä eri elementtien käsittelyllä. Taustalla oleva elementti voidaan erottaa muista elementeistä ja liikuttaa näitä eri nopeuksilla. Tämä ilmenee käyttäjän selatessa sivua tai hiiren kursorin mukana. Tämä on siis kaksiulotteisen kuvan jakamista useampaan eri kerrokseen ja näin saadaan kolmiulotteinen vaikutelma.⁹ **Kuva 3** Firewatch-pelin verkkosivusto käyttää parallax-efektiä. Se koostuu kuudesta liikkuvasta kerroksesta, jotka luovat syvyyden tunteen.



Kuva 3 Esimerkki parallax-efektistä Firewatch videopelin verkkosivustolla.¹⁰

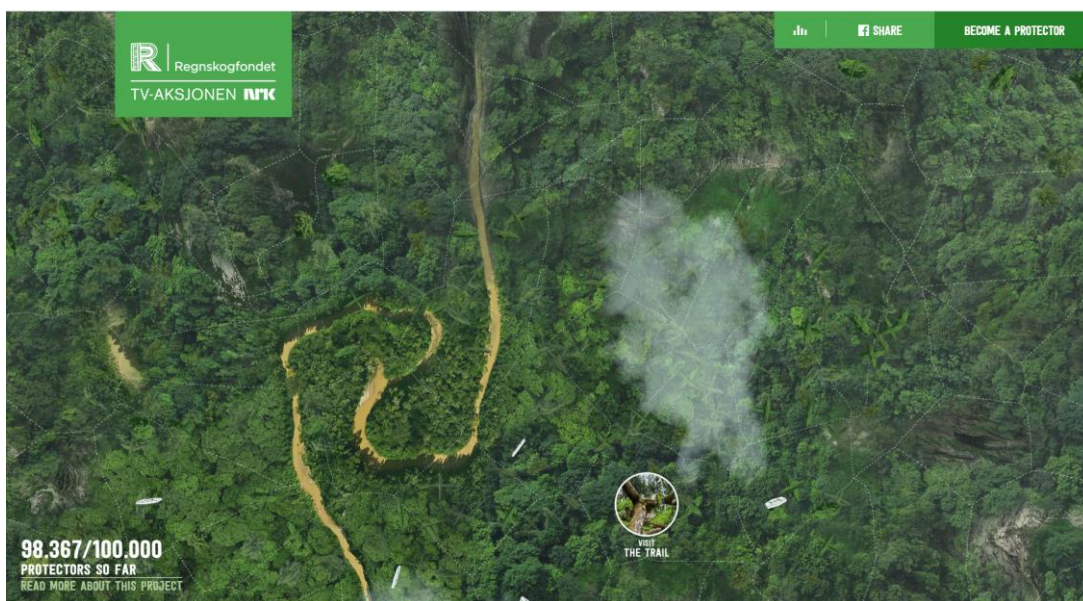
Sivuston luontoelementit liikkuvat vierittäessä sivua alas ja ylös. Efektit voivat olla ajastettuja verkkosivua avatessa tai efektit saattavat aktivoitua vasta jonkin sivulla olevan napin painalluksen jälkeen.

⁹ <https://www.sketch.com/blog/what-is-a-parallax-effect/>. 2022.

¹⁰ Firewatchgame. 2023.

4.3 Taideprojektit ja pelillisyyt

Taideprojekteihin on olemassa sovelluksia, joita mallinnuksen avulla voi toteuttaa. Taideprojektit eroavat visuaalisuuden lisäämisestä siten, että taideprojekteissa mallinnus on vielä tärkeämmässä ja suuremmassa osassa kuin itse sivuston tarkoitusperä. Näissä voi olla mukana myös henkilökohtaisia projekteja, mutta myös eri järjestöjen yhteiskunnallisia ongelmia käsitteleviä asioita. **Kuva 4** on Norja sademetsärahas-ton sivusto. Sivustolla esittää interaktiivisella kartalla sademetsän luontoa ja siellä olevia asukkaita ja eläimiä.



Kuva 4 Interaktiivinen kartta Norjan sademetsärahas-ton sivustolla.¹¹

¹¹ Interaktiivinen_kartta_Norjan_sademetsärahas-to. 2023.

Tällainen interaktiivinen kartta on hyvä pelillisyyden esimerkki. Se on vain pienemmässä osassa kokonaisuutta tuomassa interaktiivisuutta. Kartan alue on renderöity 3D-malliksi, ja käyttäjä hiirtä klikkailemalla tutkia kartan eri osia. On olemassa myös sivustoja, jossa pelillisuus on koko sivuston idea.

Selkeästi suurempien ja sisällöltään massiivisten pelien laittaminen selaimeen on toistaiseksi haastavaa. Esimerkiksi RuneScape, joka on yksi suosituimmista selaimen kautta pelattava verkkoroolipeli. Kyseinen peli on toiminut Java-liitännäisen avulla ennen, mutta nykyisin suosituimmat selaimet ovat lopettaneet kyseisten liitännäisten tukemisen. Näitä Java-ohjelmia eli Java-sovelmia kutsutaan Java Appletiksi. Tämä oli suurin syy sille, että Javasta haluttiin eroon ja sen tilalla alettiin käyttämään HTML5-merkintäkieltä ja WebGL-ohjelmointirajapintaa.^{12 13 14}

¹² Costlow, Erik. 2021. The end of Applets.

¹³ Shankland, Stephen. 2013. RuneScape Java dumb for HTML 5.

¹⁴ Runefest's big reveals. 2014.

5 PROJEKTIN LÄHESTYMISTAPA JA TOTEUTUS

Internetissä on saatavilla useita eri ohjelmia ja työkaluja 3D-mallien luomiseen ja käsittelyyn. Maya ja 3ds Max kuuluvat muun muassa tähän joukkoon ja ne ovat Autodeskin kehittämiä. Nämä ovat suosittuja parhaimmissa elokuvatuotannoissa ja videopeleissä. Siksi ne ovatkin aloittelijoille ja harrastelijoille kalliita ja epäkäytännöllisiä.^{15 16} Malleja on kuitenkin täysin mahdollista luoda myös ilmaisilla ohjelmilla. Suosituimpiin kuuluva tai jopa suosituin ohjelma on Blender. Se on avoimen lähdekoodin ohjelma. Vaikka sen ominaisuudet ovatkin Mayaa jäljessä, niin silti se on erittäin käytännöllinen 3D-mallien luomiseen ja käsittelyyn.¹⁷

3D-malleja on mahdollista luoda myös pelkässä verkkoselaimessa. Spline on selaimessa toimiva ilmainen 3D-mallinnustyökalu. Sitä on helpompi käyttää verrattuna täysin varusteltuihin 3D-sovelluksiin ja mitään ulkoisia kirjastoja ei tarvitse käyttää mallin luomisessa verkkosivulle. Tästä syystä Spline on käytännöllinen yksinkertaiseen 3D-mallinnukseen. Haastavammissa projekteissa Splinen ominaisuudet eivät välttämättä riitä.¹⁸

3D-mallinnuksessa viitataan usein monikulmioverkkoon, josta käytetään myös nimitystä polygonimallit. Polygonimallit ovat kokoelma pisteitä, reunoja ja pintoja, joita käytetään määrittämään 3D-objektin muoto ja ääriviivat. Se on vanhin geometrian esitysmuoto, jota käytetään tietokonegrafiikassa objektien luomiseen 3D-avaruudessa. Mallit koostuvat tahoista (engl. face), jotka puolestaan koostuvat reunoista (engl. edge), ja reunat muodostuvat vertekspisteistä (engl. vertex, monik. vertices). Polygonit esiintyvät usein pelkistetympinä malleina, kolmioina tai neliöinä. Mallin on sitä tarkempi, mitä enemmän polygoneja mallissa esiintyy, mutta samalla se voi olla raskaampi käsitellä toiminnallisuuksien kannalta. Alla olevassa **Kuva 5** on esimerkki normaalikartoituksesta (engl. normal mapping). Sitä käytetään

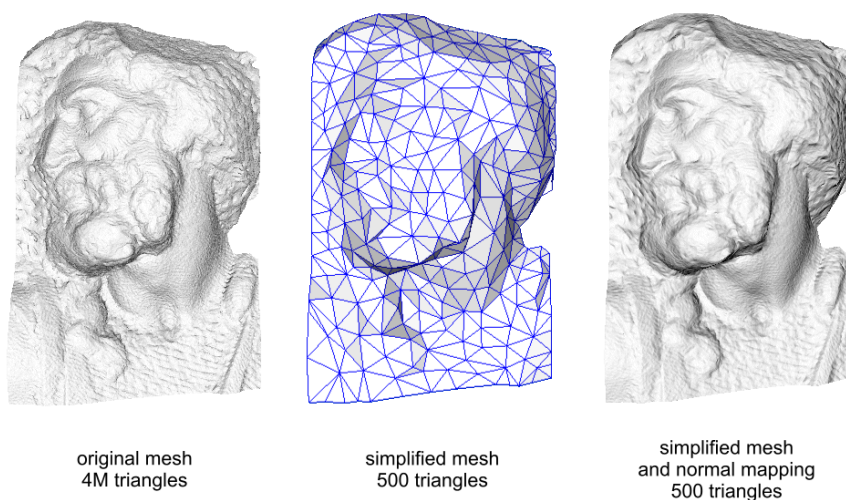
¹⁵ Autodesk Maya. 2023.

¹⁶ Autodesk 3ds Max. 2023.

¹⁷ Blender. 2023.

¹⁸ Spline. 2023.

mallinnuksessa apuna ja se auttaa saamaan mallille lisää yksityiskohtia ilman suurta polygonien määrän lisäystä. Normaalikartoituksessa tallennetaan mallin suunta, ei syvyys.^{19 20}



Kuva 5 Esimerkki normaalikartoituksesta.²¹

Kuvassa vasemmalla on alkuperäinen malli, jossa polygonikolmioiden määrä on neljä miljoonaa. Keskellä on yksinkertaistettu versio, jossa on vain 500 kolmiota ja oikealla näemme saman yksinkertaistetun version, johon on lisätty normaalikartoitus. Patsaat näyttävät vasemmalla ja oikealla lähes identtisiltä, vaikka polygonien määrä onkin vain murto-osa alkuperäisestä.

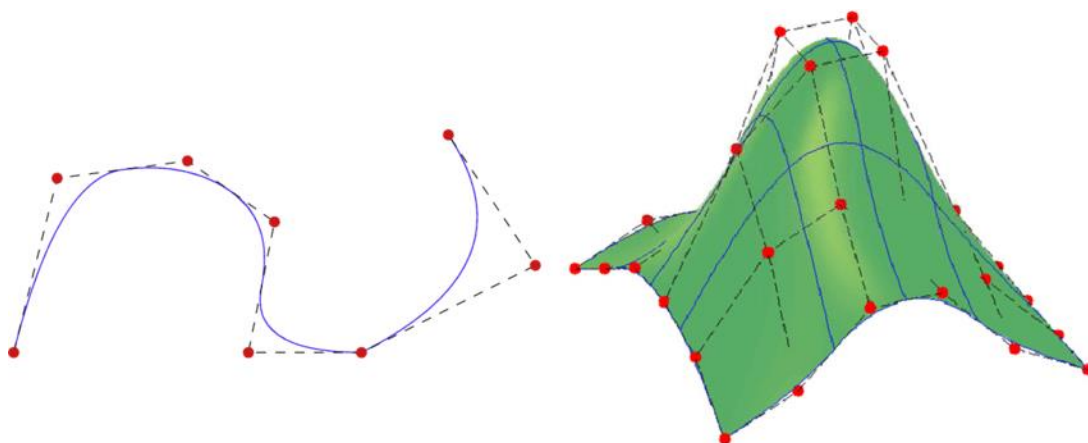
Toinen tapa 3D-mallien esittämiseen on NURBS (engl. Non-uniform rational basis spline) -mallinnus. Se on matemaattinen malli, jota käytetään käyrien ja pintojen luomiseen ja esittämiseen tietokonegrafiikassa. Käytännössä käyrä on monien pienten viivojen yhdistelmä, ja kun niitä on riittävästi, se näyttää silmälle kuin yhtenäiseltä käyrältä. NURBS-käyrät perustuvat matemaattisesti laskettuihin

¹⁹ Introduction to polygons. 2023.

²⁰ Normal mapping. 2023

²¹ Different maps in PBR textures. Normal mapping. 2023.

ohjauspisteisiin, jotka määrittävät käyrän muodon. **Kuva 6** on punaisella merkitty kontrollipisteet. Vasemmalla kuvassa on vähemmän kontrollipisteitä sisältävä käyrä ja oikealla useasta pisteestä luotu NURBS-pinta.²²



Kuva 6 NURBS-pinta ja käyrä. Kontrollipisteet on merkitty punaisella.²³

Adaptiivinen renderöinti erottaa NURBS-mallinnuksen ja polygonimallit toisistaan. Tämä tarkoittaa, että mallin pikselien määrä voi muuttua sen koon muuttuessa ja tällä tavoin adaptiivinen renderöinti mahdollistaa tarkan mallinnuksen riippumatta koosta. Tämä on samankaltainen periaate kuin esimerkiksi tietokoneen teksteissä, joissa tekstin koko suurenee zoomatessa ja pysyy silti terävänä. Polygonimalleilla on ennalta määrätty määrä pisteitä, reunoja ja tahoja, kun taas NURBS-mallissa pinnat muokkautuvat automaattisesti polygonien sijaan. Tätä pinnan jakamista pienempiin primitiiveihin kutsutaan tesselaatioksi.²⁴

Usein malleilla on myös määritelty materiaali, joka kuvastaa geometrian pinnan ominaisuuksia, kuten väriä, heijastavuutta ja pintakuviointia. Pintakuviointi voi

²² NURBS-mallinnus. 2023.

²³ NURBS-curve example. 2023.

²⁴ Interactive rendering of NURBS surfaces. 2017.

muodostua esimerkiksi toistuvista Kuvaista ja sitä hyödynnetään antamaan mallille syvyyttä tai luomaan vaikutelmaa materiaalista. Materiaalin ominaisuudet määräytyvät yleensä valitun varjostimen (engl. shader) perusteella.²⁵

Valaistus vaikuttaa merkittävästi siihen, miten materiaalit näkyvät ja miten ne halutaan tuoda esitettäväksi. Valaistus perustuu verteksien normaaleihin, ja niiden perusteella lasketaan varjostus, joka vaikuttaa pikseleiden valoisuuteen ja tummuuteen. Hahmontamisessa 2D- tai 3D-geometria muunnetaan matemaattisesti pikseleiksi, jotka esitetään näytöllä. Valaistus on olennainen osa mallin realismia, sillä ilman valoa malli voi näyttää täysin pimeältä, ja huonolla valaistuksella ilman varjoja ja syvyyttä malli saattaa näyttää tasaiselta ja kaksiulotteiselta.²⁶

Kameralla on tärkeä rooli mallin tarkastelussa, sillä se määrittää, mitä osia mallista näytetään ruudulla. Vaikka malli on olemassa matemaattisena tiedonkappaleena, se ei näy ilman kameraa. Kamera luo myös perspektiivin ja parallaksin. Parallaksilla tarkoitetaan sitä, että kauempana olevat kohteet näkyvät pienempinä, ja tätä efektiä voidaan käyttää syvyyden vaikutelman luomiseen. Toisaalta ortografinen perspektiivi säilyttää kohteiden kokosuhteen riippumatta niiden sijainnista, mikä voi olla hyödyllistä tietyissä tilanteissa.²⁷

Kameran polttoväli viittaa objektiivin pituuteen, joka vaikuttaa siihen, kuinka laaja kuva kameralla voidaan ottaa. Suuremmilla polttoväleillä saadaan tiukempi kuva, mutta samalla ne voivat aiheuttaa kalansilmäefektin, jossa kuva vääristyy reunoilla.²⁸

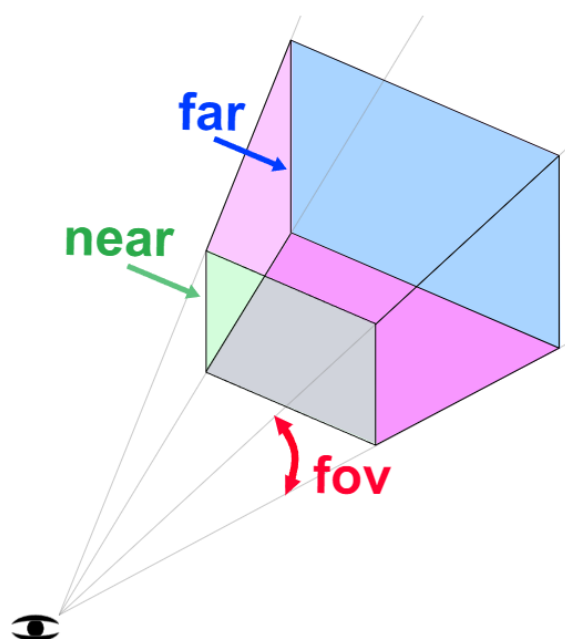
²⁵ Meshes, Materials, Shaders and Textures. 2019.

²⁶ What is 3D-lighting and how is it used in animation. 2022.

²⁷ What is the function of a camera in a 3D-modeling software. 2023.

²⁸ Autodesk. Focal length. 2023.

Kameran näkökenttä määrittää sen alueen, jolla se tarkastelee asioita. Tämä alue, jota kutsutaan myös frustumiksi, määritellään leikkaustasoilla. Leikkaustasojen korkeus riippuu näkökentästä, kun taas leveyteen vaikuttavat näkökenttä ja kuvasuhde. **Kuva 7** näkyy, miten leikkaustasoista muodostuu kartion muotoinen alue. Kamera hahmontaa vain tässä alueessa olevat kohteet.²⁹



Kuva 7 Leikkaustasojen muodostama alue renderöitynä.²⁸

Leikkaustasoja on kaksi. Yksi on aina lähempänä ja toinen puolestaan kauempana. Kun kyseessä on pienempiä esineitä, lähempänä oleva taso täytyy olla lähempänä kameraa kuin isompien esineiden kanssa. Kaukana oleva leikkaustaso tekisi esineestä olemattoman ja sitä kautta erittäin vaikean havaita.²⁸

²⁹ ThreeJS Fundamentals. 2023.

5.1 2D- ja 3D-grafiikan hahmontaminen

5.1.1 OpenGL

OpenGL on ohjelmointirajapinta, tunnetaan myös nimellä API (engl. Application Programming Interface), joka on suunniteltu tietokonegrafiikan hahmontamista varten. Hahmontamisessa on pääasiassa kaksi lähestymistapaa. Toisena vaihtoehtona on ohjelmiston hahmontaminen, joka hyödyntää tietokoneen keskusyksikköä, eli CPU:ta. Tällä tavoin ohjelmisto ottaa kaikki hahmontamiseen liittyvät osatekijät, kuten valaistuksen, varjot ja pintaKuvat, ja muuttaa ne kuvatiedostoiksi. Tämä prosessi on kuitenkin hidasta, ja yhden ruudun hahmontaminen tällä tavoin voi kestää useita tunteja. Lopputulos voi kuitenkin olla erittäin realistinen, ja tästä syystä tätä tekniikkaa käytetään esimerkiksi elokuvatuotannoissa.³⁰ Hahmontaminen voi myös tapahtua reaaliaikaisesti, jolloin hahmontaminen suoritetaan useimmiten tietokoneen grafiikkasuorittimella, joka tunnetaan nimellä GPU (engl. Graphics Processing Unit). Koska reaaliaikainen hahmontaminen edellyttää suurta laskentatehoa, ovat kehittyneet ohjelmointirajapinnat, kuten OpenGL, suunniteltu käsittelemään nopeasti muuttuvaa grafiikkadataa ja suorittamaan hahmontamisen laskennallisesti tehokkaasti.³¹ 3D-mallinnus saavutetaan lisäämällä perinteiseen kaksiulotteiseen x- ja y-akselin koordinaatistoon syvyysakseli z. 3D-tietokonegrafiikka pyrkii luomaan illuusion syvyydestä, vaikka se esittää tuloksensa litteällä tietokoneen näytöllä. OpenGL ja vastaavat ohjelmointirajapinnat hyödyntävät matematiikkaa, kuten trigonometriaa ja matriisin manipulointia, saavuttaakseen tämän illuusion.

On kuitenkin tärkeä huomata, että syvän matematiikan täydellinen ymmärtäminen ei ole välttämätöntä tietokonegrafiikan luomiseen. Monet grafiikkakehityksen työkalut ja ohjelmistot tarjoavat abstraktiotason toimintoja, jotka helpottavat 3D-grafiikan luomista ilman syvää matemaattista osaamista. Grafiikkaohjelmoijien ja -suunnittelijoiden tiimit voivat myös jakaa vastuun, jolloin tiimin jäsenet voivat

³⁰ OpenGL. 2022. Getting started.

³¹ en.wikipedia.org. OpenGL. 2023.

keskittyä omaan osa-alueeseensa. Kokonaisuudessaan, vaikka matematiikka on tärkeä osa 3D-tietokonegrafiikkaa, ei ole välttämätöntä olla syvällistä matemaattikkoa grafiikan luomiseen, sillä saatavilla on monia työkaluja ja resursseja, jotka helpottavat grafiikan kehittämistä ja mahdollistavat yhteistyön monenlaisten osaajien kanssa.³²

5.1.2 WebGL

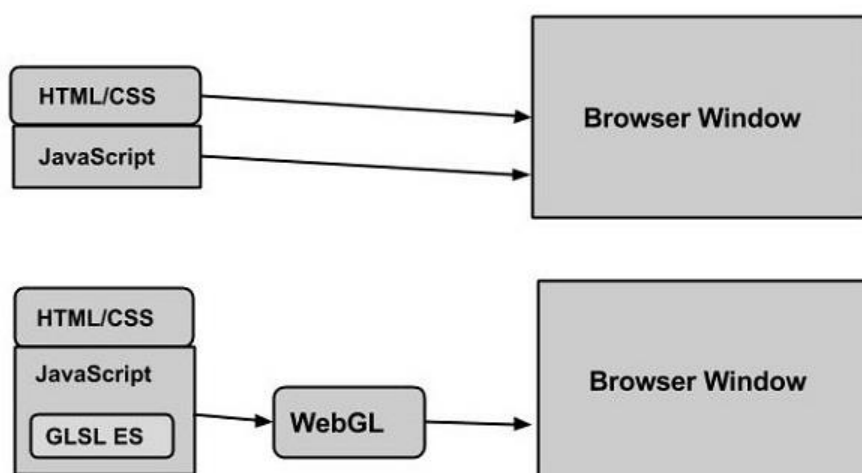
WebGL (engl. Web Graphics Library) on JavaScript-ohjelmointirajapinta, joka mahdollistaa tietokonegrafiikan renderöinnin suoraan verkkosivuilla. Tämä ohjelmointirajapinta perustuu OpenGL ES -rajapintaan (Open Graphics Library for Embedded Systems), joka on suunniteltu sulautetuille järjestelmille. WebGL on siis rakenteeltaan hyvin samankaltainen OpenGL ES:n kanssa. WebGL:n avulla verkkosivujen kehittäjät voivat luoda interaktiivisia ja graafisesti vaikuttavia 3D-mallinnuksia ja -visualisointeja suoraan verkkoselaimeen ilman tarvetta erillisille liitännäisille tai laajennuksille. Tämä tekee 3D-grafiikan ja visuaalisten tehosteiden integroimisesta verkkosivuille helpompaa ja saavutettavampaa laajalle yleisölle. WebGL on tullut suosituksi erityisesti pelikehittäjien, tieteellisten visualisointien tekijöiden ja muiden grafiikan ammattilaisten keskuudessa, jotka haluavat tuoda 3D-grafiikan verkkoympäristöön.³³

WebGL käyttää varjostinpohjaista lähestymistapaa, joka koostuu JavaScript-koodista sekä varjostinkoodista (engl. shader). Shader, jota kutsutaan myös varjostimeksi, on tietokoneohjelma, joka ohjelmoi grafiikan visuaalisia tehosteita. Varjostinkoodi kirjoitetaan GLSL-varjostuskielellä (OpenGL ES Shading Language), ja se vastaa WebGL-piirtotoimintojen toteuttamisesta. **Kuva 8** verrataan tavallisen verkkosivun ja WebGL:ää käyttävän verkkosivun rakennetta. Ylempänä kuvassa on

³² Learn OpenGL. 2023. Coordinate systems.

³³ WebGL. 2022. Main

tavallinen verkkosivu, joka ei käytä WebGL:ää, ja alempana on WebGL:ää käyttävä verkkosivu, joka sisältää myös shader-kielen GLSL-koodia. WebGL:ssä shaderit ovat tehokas tapa hallita graafista ulostuloa ja saavuttaa monimutkaisia visuaalisia efektejä, kuten varjostuksia, tekstuuriä käsittelyä ja valaistusta. GLSL on erityinen ohjelmointikieli, joka on suunniteltu nimenomaan varjostimien kirjoittamiseen, ja se mahdollistaa monipuoliset graafiset manipulaatiot WebGL-sovelluksissa. Shaderit toimivat yhteistyössä JavaScript-koodin kanssa, joka ohjaa WebGL-sovelluksen yleistä toimintaa ja liitännäisiä, jotka saattavat olla tarpeen monimutkaisten grafiikkatehtävien suorittamiseksi.³⁴



Kuva 8 Sivuston rakenne ilman WebGL:ää ja sen kanssa.³⁵

GLSL-varjostinkieli kirjoitetaan usein JavaScript-ympäristössä WebGL-sovelluksissa. Tämä yhdistää grafiikkivarjostimen ja sovelluskoodin saman ohjelmointiympäristön sisään. Kehittäjä voi siten luoda graafisia tehosteita ja määrittää visuaalisen ulostulon suoraan JavaScript-sovelluksen sisällä ilman tarvetta erillisille tiedostoille, mikä yksinkertaistaa koodin hallintaa, koska kaikki tarvittavat tiedostot pysyvät samassa projektissa.³⁴

³⁴ khronos.org/webgl/wiki/getting started. 2011.

³⁵ WebGL overview. Kuva. 2023.

WebGL tukee sekä verteksivarjostimia (vertex shaders) että fragmenttivarjostimia (fragment shaders), ja nämä ovat tärkeitä osia grafiikkaprosessissa.³⁶

Verteksivarjostimet käsittelevät pisteitä, joita kutsutaan vertekseiksi. Verteksi on tietokonegrafiikassa piste, jolla on määritellyt koordinaatit (x, y, z) ja mahdollisesti muita ominaisuuksia, kuten väri, tekstuurikoordinaatit ja heijastavuustiedot. Verteksivarjostimet ottavat nämä verteksitiedot ja voivat suorittaa niihin erilaisia matemaattisia operaatioita. Niitä käytetään muokkaamaan verteksien sijainteja ja lisäämään niihin erilaisia visuaalisia tehosteita, kuten värinmuutoksia, varjostuksia ja muodonmuutoksia. Verteksivarjostimet ovat tärkeitä 3D-mallin muokkaamisessa ennen sen piirtämistä näytölle.³⁶

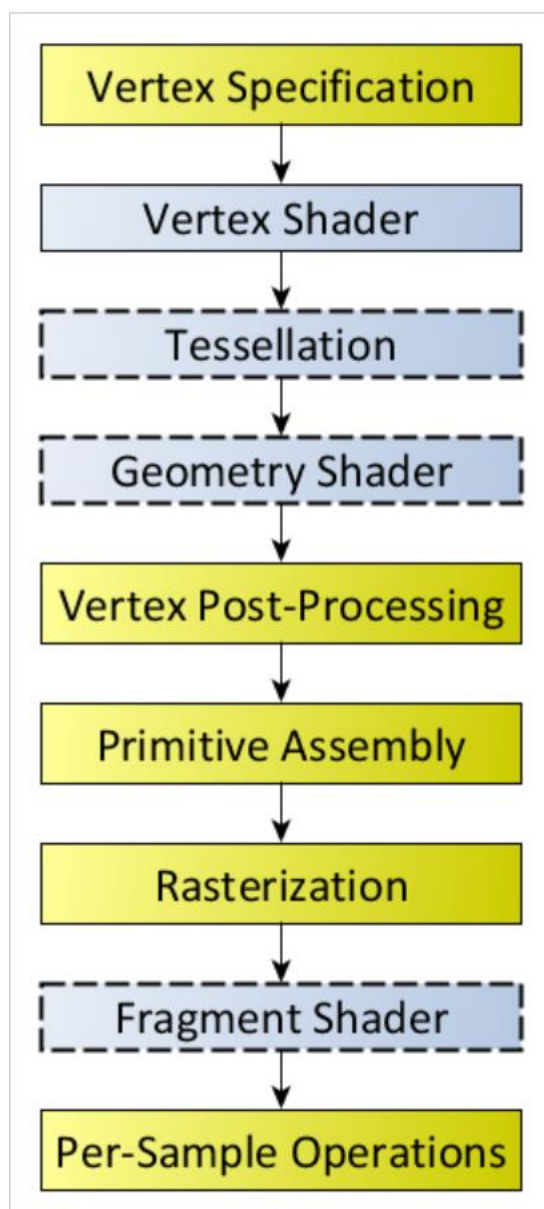
Fragmenttivarjostimet puolestaan käsittelevät pikseleitä, jotka syntyvät siitä, kun piirretty kolmio peittää osan näytöstä. Fragmentti on pieni osa kolmiota, joka lopulta näkyy näytöllä, ja fragmenttivarjostimet määrittävät, miten nämä pikselit tulevat näyttöön. Ne voivat suorittaa erilaisia laskutoimituksia, kuten värin määrittämistä, tekstuurien soveltamista ja valaistuksen laskemista. Fragmenttivarjostimet ovat tärkeitä värien ja visuaalisten tehosteiden lisäämisessä näytölle piirrettyihin kolmioihin.³⁶

Yhdistämällä verteksivarjostimet ja fragmenttivarjostimet voit saavuttaa monimutkaisia ja vaikuttavia visuaalisia tuloksia WebGL-sovelluksissa. Näiden varjostimien avulla voit muokata ja manipuloida 3D-malleja sekä luoda upeita graafisia tehosteita suoraan verkkosivullasi.³⁶

Hahmontaminen toteutetaan grafiikkaputken avulla (engl. Rendering pipeline). **Kuva 9** esitetään grafiikkaputken kaavio, jossa aluksi syötetään verteksit varjostimeen, ja varjostin tuottaa tuloksena olevat verteksit (engl. Output vertex). Verteksitietoa voidaan valinnaisesti käsitellä tesselaation avulla, ja primitiivit voivat myös läpikäydä geometriavarjostimen muokkauksen. Primitiivit ovat yksinkertaisia

³⁶ khronos.org/webgl. 2023.

geometriaan liittyviä malleja, kuten kolmioita ja viivoja. Geometriavarjostin voi käsitellä primitiivien vertekspisteitä ja muokata primitiivejä esimerkiksi niiden poistamiseksi tai muuttamiseksi, kuten muuttamalla kolmioita pisteiksi. Geometriavarjostin siis käsittelee saapuvat primitiivit ja voi palauttaa nollasta useampaan lähtöprimitiiviin.³⁷



Kuva 9 OpenGL-renderöintiputki.³⁶

³⁷ Rendering pipeline overview. 2022. Khronos Group.

Rasterointi merkitsee primitiivien muuntamista fragmenteiksi. Fragmentit ovat joukkoja tiloja, joita hyödynnetään pikselin lopullisen datan määrittämiseen. Fragmenttivarjostin laskee jokaiselle fragmentille, joka on syntynyt rasteroinnin tuloksena, värin ja syvyyden. Viimeisessä vaiheessa, joka tunnetaan nimellä "Per-Sample Operations", fragmentteja käsitellään ja niistä saatu data tallennetaan eri puskureihin.³⁸

GLSL-varjostimet vaativat pääohjelman (main function), josta sovellus käynnistyy. JavaScript-koodissa puolestaan joka rivi suoritetaan kerrallaan peräkkäin. Toisaalta toinen ero JavaScriptiin on se, että GLSL-varjostimissa funktion paluuarvo määritellään sen tyyppin mukaan. Jos funktio ei palauta arvoa, käytetään avainsanaa "void". GLSL ei tue esimerkiksi merkkijonodatatyyppejä, vaan sallitut datatyypit ovat ainoastaan kokonaisluku (integer), liukuluku (float) ja looginen (boolean).³⁷

GLSL-datan voi esittää vektori- tai matriisimuodossa. **Kuva 10** näytetään esimerkit vektorimuotoisesta ja matriisimuotoisesta datasta. Vasemmalla kuvassa on esimerkki vektorista, joka järjestää tiedon listaksi. Vektoreita käytetään esimerkiksi koordinaattien ja väridatan esittämiseen. Matriisi puolestaan järjestää tiedon taulukoksi, ja niitä käytetään esimerkiksi kuvausmatriiseihin (engl. Transform matrix), jotka mahdollistavat kappaleiden koon, sijainnin ja orientaation muokkaamisen.³⁹

³⁸ OpenGL/wiki/Fragment Shader. 2020.

³⁹ Vector and matrix. 2023.

$$\begin{pmatrix} 1 & 5 & 3 \\ 4 & 9 & 2 \\ 5 & 6 & 7 \end{pmatrix} \quad \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Kuva 10 Vektori- ja matriisidata.³⁹

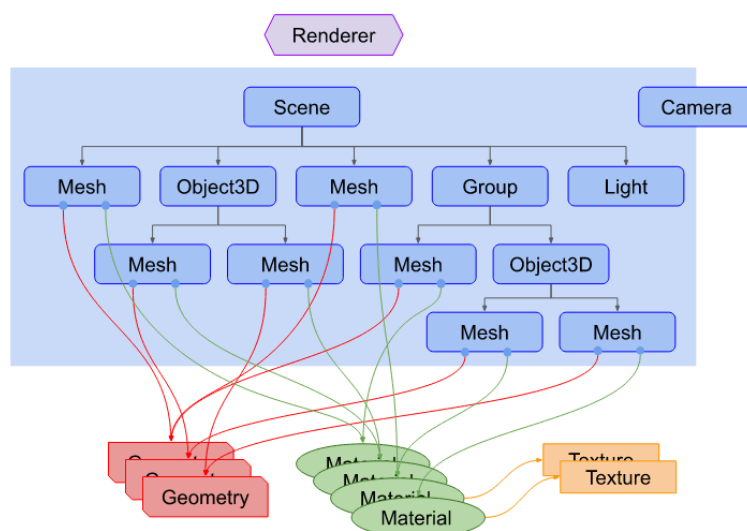
5.2 Grafiikkakehykset ja -moottorit selaimessa

Pelkistetty 3D-mallinnus ei enää edellytä huippuluokan tietokonetta kehittäjältä, vaan kuka tahansa voi luoda WebGL-sovelluksen pelkkää tekstieditoria ja selainta käyttäen. Kuitenkin suora WebGL-ohjelmointi pelkällä JavaScript-koodilla voi olla monimutkaista ja altista virheille. Tämän vuoksi usein käytetään grafiikkakehystä (graphics framework) helpottamaan tätä prosessia. Koska erilaisia grafiikkakehyksiä on useita, tarkastelen tässä kolmea niistä lähemmin.

5.2.1 Three.js

Three.js on JavaScript-kirjasto, joka on suunniteltu helpottamaan ja nopeuttamaan WebGL-sisällön luomista. Kirjastot ovat kokoelmia resursseja, kuten valmiita koodinpätkiä ja toiminnallisuuksia, jotka auttavat kehittäjiä luomaan interaktiivisia 3D-grafiikkaa sisältäviä sovelluksia. Three.js-sovellus rakentuu kohtaukseen (engl. scene). Kohtaus toimii rakenteena, joka toimii säiliönä eri objekteille. Kohtaus sijaitsee sovelluksen juuressa ja määrittää, mitä asioita sovellus hahmontaa tai esittää. **Kuva 11** esitetään pienen Three.js-sovelluksen rakenne. Sovellus koostuu

erilaisista mesh-objekteista, kameroista ja valoista, jotka yhdessä muodostavat 3D-maailman ja sen visuaalisen sisällön.⁴⁰



Kuva 11 Pienen Threejs-sovelluksen rakenne.⁴¹

Mesh-objektit koostuvat geometriasta ja materiaalista. Geometria viittaa hahmoteltavaan objektiin, kuten esimerkiksi matkapuhelimeen tai autoon, joka yleensä luodaan 3D-mallinnusohjelmassa. Tämän jälkeen malli tallennetaan tiedostoon ja ladataan verkkosivustolle. Three.js tarjoaa myös valmiita primitiivejä yksinkertaisille objekteille, kuten kuutioille ja palloille, joita voi muokata eri parametrien avulla. **Kuva 12** on esimerkki kuution luomisesta Three.js-koodissa.⁴⁰ Alla myös koodiesimerkki kuution luomisesta Three.js-kirjastolla:

⁴⁰ ThreeJS manual. 2023.

⁴¹ Structure of ThreeJS app. 2023.

```
// Luodaan kuution geometria

var geometry = new THREE.BoxGeometry(1, 1, 1);

// Määritellään kuution materiaali

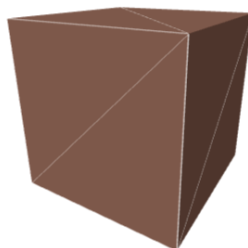
var material = new THREE.MeshBasicMaterial({ color: 0x00ff00 });

// Luodaan mesh-objekti, joka yhdistää geometrian ja materiaalin

var cube = new THREE.Mesh(geometry, material);

// Lisätään kuutio kohtaukseen

scene.add(cube);
```



```
const width = 8; 
const height = 8; 
const depth = 8; 
const geometry = new THREE.BoxGeometry( width, height, depth );
```

Kuva 12 Kuution luominen laatikkoprimitiiveillä ja asetetuilla parametreilla. ⁴²

⁴² ThreeJS_manual_primitives. 2023.

Materiaali yleensä saapuu mallinnusohjelman kanssa saman tiedoston mukana. PintaKuvainti on kuva, joka ladataan tiedostosta ja asetetaan geometrian ympärille. Yhtä materiaalia ja tekstuuria voidaan käyttää useissa eri geometrisissa objekteissa.⁴¹

Three.js tarjoaa useita erilaisia kameroita, jotka mahdollistavat erilaisten kuvakulmien ja näkökulmien luomisen 3D-sovelluksissa. Näistä kameroista yksi on aiemmin mainittu ortografinen kamera eli **OrthographicCamera**, joka pitää kuvattavan kohteen samankokoisena riippumatta kameran etäisyydestä. Tämä tekee siitä sopivan erityisesti tekniseen ja arkkitehtoniseen visualisointiin.⁴¹

Kuitenkin yleisin ja eniten ihmisen silmää imitoiva kamera on **PerspectiveCamera**. Tämä kamera luo perspektiivin vaikutelman, joka tekee 3D-kohteista näyttävän realistisia. Kameraa luodessa sille voidaan määrittää erilaisia parametreja. Näihin parametreihin kuuluvat näkökenttä (engl. field of view), kuvasuhde (engl. aspect ratio) sekä near- ja far-etäisyydet (engl. near and far planes).⁴¹

3D-mallien visualisointiin tarvitaan hahmontaja, joka vastaa siitä, miten kohtaus näytetään ja miten mallit piirretään kameran näkökulmasta. Nykyisin yleisesti käytetään WebGLRendereria. Aikaisemmin oli muita vaihtoehtoja, kuten CanvasRenderer, mutta niissä esiintyi puutteita ja suorituskykyongelmia. Vuonna 2018 Three.js-kehittäjä Ricardo Cabello ilmoitti sosiaalisessa mediassa, että CanvasRenderer poistetaan kirjastosta, koska WebGL on laajasti tuettu kaikkialla.⁴²

Kohtauksen visualisointi vaatii koodissa piirto- tai animointisilmukan. Tämä merkitsee, että kohtaus renderöidään näytölle joka kerta, kun näyttö päivittyy.⁴³ Päivitysten lukumäärä riippuu näytön virkistystaajuudesta, joka useimmissa näytöissä on 60 hertsiä (engl. hertz, Hz). Koska virkistystaajuus vaikuttaa siihen, kuinka sulavasti asiat näkyvät näytöllä, pelinäytöissä tämä luku voi olla jopa 500 Hz.

⁴³ ThreeJS. 2023. Creating a scene.

5.2.2 Babylon.Js

Babylon.Js:n alkuperä on Microsoftin luoma, avoimen lähdekoodin grafiikkamoottori. Babylon.Js:llä on ollut merkittäviä sovelluksia, kuten Niken "Nike By You" -kenkiensuunnittelusovellus, joka antaa käyttäjille mahdollisuuden muokata kenkien materiaaleja ja värejä sekä suunnitella omia kenkiään. Myös vähittäiskauppa Target on hyödyntänyt Babylon.Js:ää kehittäessään 3D-sisustussovelluksen, joka mahdollistaa käyttäjien huoneiden koon määrittämisen ja niiden sisustamisen asettamalla tuotteita 3D-tilassa.^{44 45}

Tutkin kahta yksinkertaista sovellusta, jotka on luotu Three.js- ja Babylon-kehysalustoilla. Molemmat sovellukset sisälsivät laatikon, kameran ja valon. Sovellusten välillä oli havaittavissa joitain eroja. Molempien kehysten avulla voitiin simuloida valoa ympäristöstä mallille. Three.js:ssä tätä valoa kutsutaan HemisphereLightiksi, kun taas Babylonissa se on HemisphericLight. Valaistuksen toteutus eroaa kuitenkin näiden kahden kehysten välillä. Babylon.Js:ssä valon luomisen yhteydessä määritellään valon nimi, heijastumissuunta ja kohta, johon valo liittyy, ja sen jälkeen sille voidaan asettaa väri erikseen. Toisaalta Three.js:ssä valolle voidaan suoraan määrittää taivaan väri, maan väri ja valon teho ennen kuin se lisätään kohtaukseen. Vaikka nämä kaksi grafiikkakirjastoa tarjoavat samankaltaisia toiminnallisuuksia, niiden käyttöön liittyvät yksityiskohdat voivat vaihdella, ja joissain tapauksissa ero voi näyttäytyä lähinnä nimityksissä. Esimerkiksi Three.js-sovelluksissa käytetään "renderer"-objektia, kun taas Babylon.Js:ssä vastaavan tehtävän hoitaa "BABYLON.Engine".

⁴⁴ Irwin, Emma. 2021. Microsoft Open-Source success story—Babylon.

⁴⁵ Babylon.Js kotisivut. 2023.

Vaikka Three.js ja Babylon.js eroavatkin toisistaan joissain toteutusyksityiskohdissa, niiden sovellusten perusrakenne pysyy melko samankaltaisena. Three.js saattaa vaatia enemmän koodirivejä tietyissä tilanteissa verrattuna Babyloniin, mutta se saattaa tuntua yksinkertaisemmalta, riippuen käyttötarkoituksesta. Kuitenkin on tärkeää huomata, että Babylon voi olla parempi valinta monimutkaisempiin sovelluksiin, erityisesti sen laajennusjärjestelmän ja fysiikkamoottoreiden käytön mahdollisuuksien ansiosta. Babylonin vahvuudet voivat tulla esiin, kun tarvitaan lisäominaisuuksia ja monimutkaisempaa toiminnallisuutta, kuten simulaatioita ja vuorovaikutteisia 3D-ympäristöjä. Lopullinen valinta riippuu sovelluksesi tarpeista ja siitä, mikä kirjasto sopii parhaiten projektisi tavoitteisiin.⁴⁶

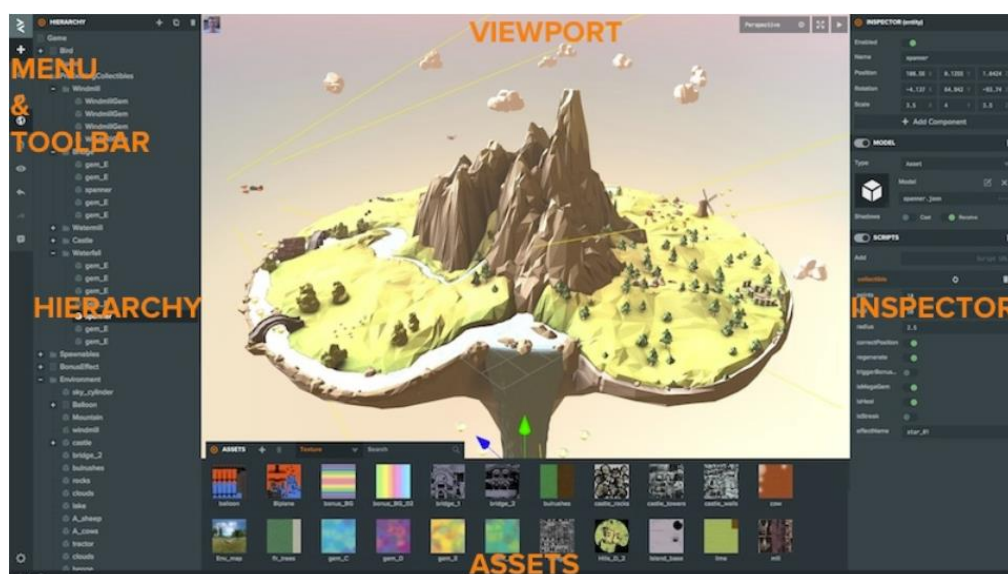
5.2.3 PlayCanvas

PlayCanvas on toinen suosittu avoimen lähdekoodin 3D-moottori, joka julkaistiin vuonna 2014 ja on erityisesti suunniteltu pelien kehittämiseen. PlayCanvas eroaa Babylonista ja Three.js:stä siinä, että se tarjoaa tehokkaan integroidun muokkausohjelman. **Kuva 13** näkyvä muokkausohjelma mahdollistaa uusien objektien luomisen ja muokkaamisen ilman koodin kirjoittamisen. Tämä tekee PlayCanvasista houkuttelevan vaihtoehdon erityisesti niille, jotka haluavat luoda pelejä ja 3D-sovelluksia visuaalisella tavalla ilman syvällistä ohjelmointiosaamista.⁴⁷⁴⁸

⁴⁶ Using a physics engine. Babylon.js. 2023.

⁴⁷ Eastcott, Will & Nyman Robert. 2014. PlayCanvas Goes Open Source.

⁴⁸ Editor. PlayCanvas. 2023.



Kuva 13 PlayCanvas editori. ⁴⁷

PlayCanvas-sovelluksissa on samat peruskomponentit kuin aiemmin mainituissa kehysissä. Pelissä on lähes poikkeuksetta kamera, valo ja 3D-malleja. PlayCanvasissa 3D-malli on osa laajempaa kokonaisuutta, jota kutsutaan entiteeteiksi (engl. Entity). Entiteetti rakentuu komponenteista (engl. Component), joiden toimintaa määrittelevät komponenttijärjestelmät (engl. Component system).⁴⁹

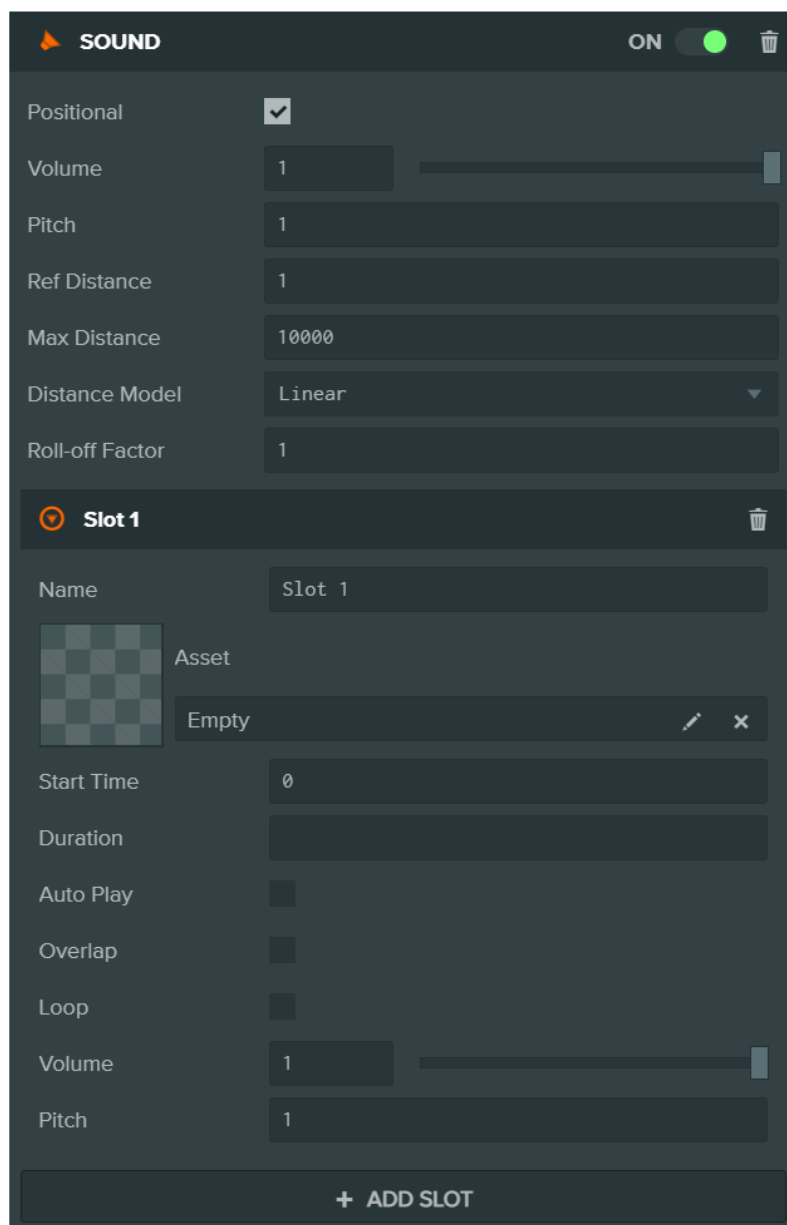
PlayCanvasin omassa esimerkissä yksinkertainen entiteetti rakentuu kolmesta osasta: äänikomponentista, mallikomponentista ja skriptikomponentista. Mallikomponentti kuvaa 3D-mallia tai -primitiiviä, ja se piirretään entiteetin sijaintiin, kun mallikomponentti otetaan käyttöön. Tässä esimerkissä käytetään jalkapalloa mallina. ⁵⁰

Äänikomponentti mahdollistaa äänitiedostojen toistamisen. **Kuva 14** näytetään yksinkertainen äänikomponentti, joka on sijoitettu yhteen kohtaan. Jokaiselle tarvittavalle äänitiedostolle lisätään paikka entiteetille, ja kehittäjä voi määrittää editorissa esimerkiksi aloitusajan, keston ja äänitiedoston toistolistan.

⁴⁹ Entity. PlayCanvas. 2023.

⁵⁰ Making a simple game – part 1. PlayCanvas. 2023.

Esimerkkisovelluksessa jalkapallolle määritetään yksi sijainti äänelle, joka toistetaan aina kun pallo pomppii.⁴⁹



Kuva 14 PlayCanvas-äänikomponentti.⁵¹

⁵¹ PlayCanvas. Äänikomponentti. 2023.

Skriptit ovat kooditiedostoja, jotka ohjaavat sovelluksen toimintoja. Vaikka PlayCanvasin editori auttaa vähentämään koodin kirjoittamisen tarvetta, kaikissa interaktiivisissa sovelluksissa tarvitaan silti vähintään yksi skripti. Jalkapallopelissä skriptit voivat esimerkiksi hallita pallon pyörimistä. **Esimerkkikoodi 1** PlayCanvasin 20 käyttöoppaassa annetaan esimerkki esineen pyörittämisestä käyttämällä rotate-metodia. Esinettä pyöritetään kymmenen astetta joka sekunti.⁴⁹

```
var Rotate = pc.createScript("rotate");

Rotate.prototype.update = function (dt) {
    this.entity.rotate(0, 10*dt, 0);
};
```

Esimerkkikoodi 1 Yksinkertainen animaatioskripti PlayCanvasissa.⁵²

⁵² Scripting. PlayCanvas. 2023.

6 PROJEKTIN TUOTOKSET

Opinnäytetyön tavoitteena on tuoda tiedostona oleva 3D-malli web-käyttöliittymään ja animoida sen liikkeitä vastaamaan oikean laitteen liikkeitä. Tarkoituksena on siis käyttää mallinnettua varastohissiä, joita käytetään korkeavarastolinjoilla kuljettamaan erilaisia keloja. Tätä varastohissiä ja sen liikkeitä animoidaan sivuttais- ja pystysuunnassa ja myös hississä olevaa haarukkaa, jonka tehtävänä on työntää tavara varastokanavaan tai ottaa se sieltä pois. Esittelen myös samalla yleisimpiä tekniikoita 3D-mallien käsittelyyn ja koodirakenteen puolelta tärkeimpiä osia työn toteutuksen kannalta. Tämä on ensimmäinen kerta, kun itse olen 3D-mallien kanssa tekemisissä, joten työ antaa myös näkökulmaa siihen, kuinka 3D-tekniikoita onnistuu käsittelemään, vaikka siihen ei olisikaan mitään aiempaa kokemusta.

6.1 Teknologian valinta

Mallinnussovellukseksi valitsin Blenderin. Se on ilmainen, joten se soveltuu erittäin hyvin opinnäytetyön kaltaiseen projektiin. Blender on myös helppokäyttöinen, koska siihen löytyy paljon ohjeita tekstimuodossa ja myös opastusvideoita, joista aloittelijatkin ymmärtävät helposti sovelluksen toimintaperiaatteet. Sovelluksen ilmaisuudesta huolimatta se pystyy monien eri maksullisten sovellusten tavoin, ammattimaiseen mallinnukseen ja hahmontamiseen.

Opinnäytetyössäni käytössäni on korkeavarastoiden varastohissin 3D-malli 3D-tiedostona. Kyseinen hissi on melko vaikea muoto 3D-malliksi, sillä se koostuu neljästä erilaisesta osasta, joita täytyy käsitellä erikseen vähän kuin omana objektinaan. Hissi sisältää myös monia erilaisia yksityiskohtia, jotka tekevät mallista raskaamman käsitellä renderöitäessä. Mallinnuksen tukena käytän erilaisia YouTube-videoita, jossa esiteltiin ja ohjeistettiin Blenderin toimintaa. Käytän apuna myös Blenderin omilta sivuilta löytyvää manuaalia. Mallinnuksen tavoitteena on

tehdä mallista kevyempi, poistamalla siitä turhia yksityiskohtia, jotta malli olisi sulavampi verkkosivulla esitettäessä.

Ohjelmointiratkaisut toteutetaan grafiikkakehys ThreeJs:ää hyödyntäen. Tutkiessa eri 3D-mallinnukseen liittyviä kehyksiä, ThreeJs tuntui yksinkertaisimmalta ja parhaiten soveltuvalta verkkosivulla toimimista ajatellen. Ohjeita ja tutoriaaleja on myös saatavilla riittävästi ja niiden avulla on helppoa tutustua kehyksen käyttöön ja toiminnallisuuksiin. Blenderillä hahmontamisen lisäksi luon hissille animaatioita, jotka toteutetaan ThreeJs-kehystä hyödyntäen.

6.1.1 Vertailukohteet

Käsittelen opinnäytetyössäni kolmea erilaista grafiikkakehystä, joita voi käyttää 3D-mallien käsittelemiseen. BabylonJs, ThreeJs ja PlayCanvas ovat tehokkaita 3D-grafiikkakirjastoja, joita käytetään usein web-sovellusten ja pelien kehittämiseen.

BabylonJs on tarkoitettu pääosin pelien kehittämiseen ja se tarjoaa monipuolisia ominaisuuksia, kuten fysiikkamoottorin ja erinomaisen materiaalijärjestelmän. BabylonJs:llä on vahva yhteisö ja laaja dokumentaatio, mikä tekee siitä helpon aloittelijoille ja kokeneille kehittäjille.

ThreeJs on erittäin joustava ja antaa kehittäjille suuren vapauden luoda omia ratkaisujaan. Se ei ole rajoittunut mihinkään tiettyyn sovellusalueeseen ja soveltuu hyvin grafiikan ja visualisoinnin luomiseen. ThreeJs:llä on myös vahva yhteisö ja hyödyllinen dokumentaatio.

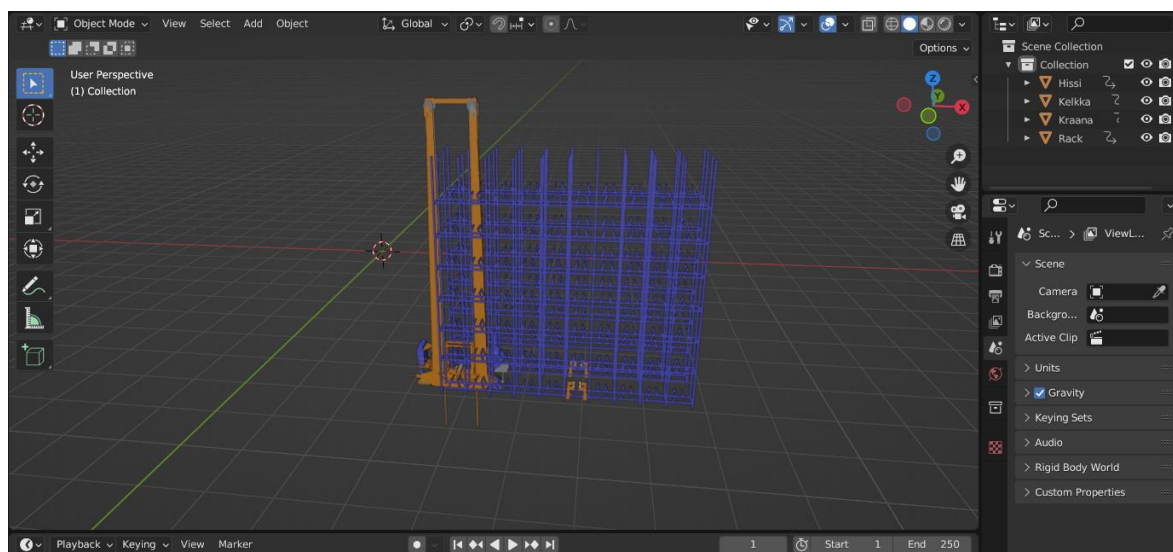
PlayCanvas on ainutlaatuinen, koska se tarjoaa pilvipohjaisen kehitysympäristön 3D-web-pelien luomiseen. Se on erityisen voimakas monipelien ja mobiilipelien kehittämisessä. PlayCanvasilla on pienempi yhteisö verrattuna edellä mainittuihin vaihtoehtoihin, mutta se tarjoaa silti riittävästi tukea ja dokumentaatiota.

Käyttötarkoituksen vuoksi päädyn käyttämään ThreeJs:ää, sillä se antaa vapauden räätälöidä grafiikkaa ja se soveltuu parhaiten web-kehityksessä. BabylonJs ja PlayCanvas ovat taas parempia pelien kehittämiseen.

6.1.2 Toiminnallisuus

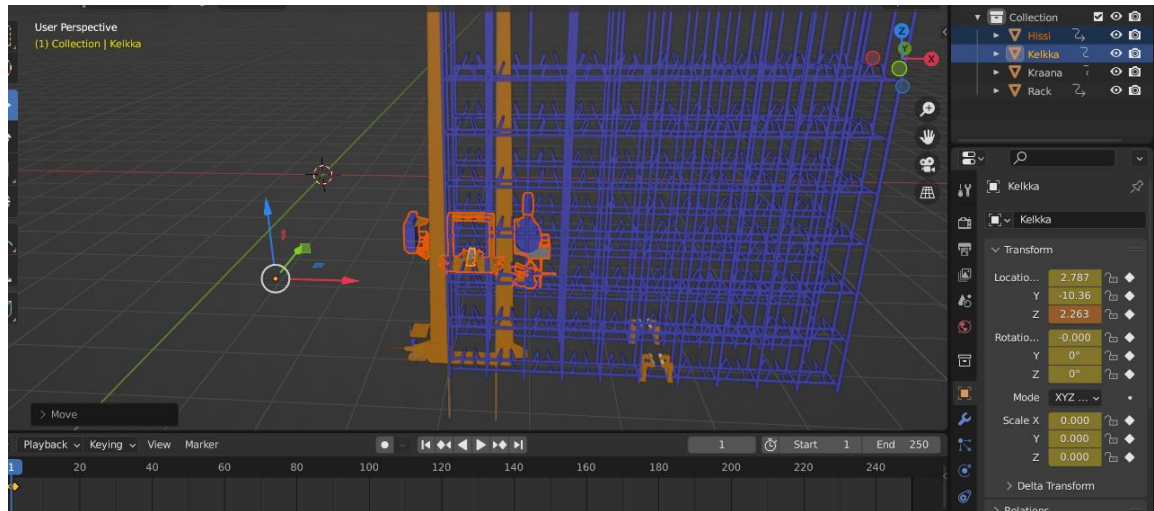
Projektini alkaa erilaisten hissimallien käsittelyllä Blenderissä. Käsittelen noin kymmenkuntaa valmista hissimallia, jotka saan työpaikallani työskentelevältä visuaalisten komponenttien suunnittelijalta. Oikean mallin hyödyntäminen vaatii jonkin verran aikaa, sillä ThreeJs suosii vain tietynlaisia tiedostomuotoja, joten mallit piti viedä (engl. export) Blenderistä oikeanlaisessa tiedostomuodossa ulos. Vaikeuksia tuottaa myös se, että mallit ovat tiedostokooltaan erikokoisia, mikä vaikeuttaa heti hankaluuksia mallin toimivuudessa. Mitä isompi tiedostokoko, sitä kauemmin Blenderillä kestää avata kyseinen tiedosto, puhumattakaan sen käsittelystä. Tämä aiheuttaa sovelluksen kaatumisen muutamia kertoja.

Oikeanlaisen mallin löytyessä, sitä on suhteellisen helppoa ja vaivatonta käsitellä Blenderin avulla. **Kuva 15** huomataan hissimallin rakenne Blender-sovelluksessa. Malli sijaitsee yhdessä tiedostossa, mutta tiedoston sisällä se on pilkottu neljään osaan, joihin kuuluu hissiosa, kraana, joka on hissin runko-osa, kelkka, joka on pieni liikkuva kisko hissien keskiosassa ja räkki, joka vastaa varastokanavaa, jossa kuljetettavat tavarat sijaitsevat ja joihin ne varastoidaan. Kyseinen malli soveltuu parhaiten käyttötarkoitukseen, sillä malli on kaikinpuolin kevyt ja helppo käsitellä ja se on tiedostomuodoltaan fbx., joka on suhteellisen vaivatonta muuntaa ThreeJs:n tukemaan glb. muotoon.

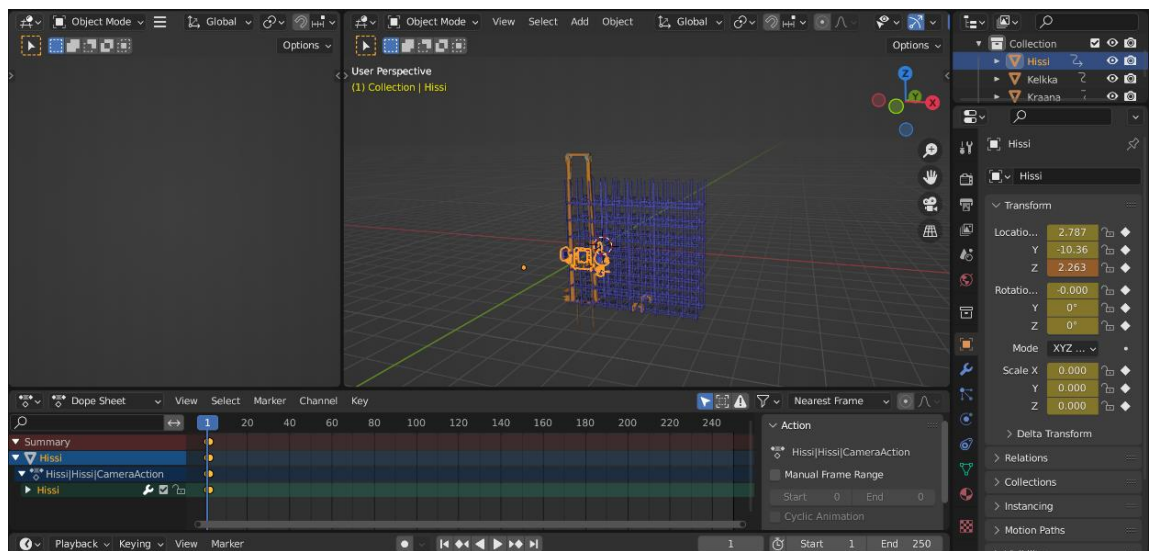


Kuva 15 Hissimallin rakenne.

Ennen tiedoston muuntamista hahmonnan mallille ja mallin eri osille erilaisia animaatioita, joita on mahdollista tehdä Blenderin valmiiden ominaisuuksien ansiosta. **Kuva 16** esittää animaation tekemistä, jossa kohteeksi on valittuna hissi ja kelkkaosa, koska kelkka on osa hissiä, niin sitä täytyy liikuttaa hissien liikkeen mukana. Muuten kelkka jäisi alas lähtöpisteeseen, kun hissiosa liikkuu. Osille luodaan ylös-alas-suuntainen liike, niin, että haluamat osat täytyy olla valittuna, jonka jälkeen osia liikutellaan kuvassa näkyvällä horisontaali- ja vertikaaliakselilla. Ennen kuin osia liikutellaan, niille täytyy asettaa alkupiste. Osien liikuttelun jälkeen niille asetetaan lopetuspiste ja aika, kuinka kauan liikkeen halutaan kestävän. Tästä muodostuu animaatio. Animaatioita voi tarkastella luomisen jälkeen Blenderin animaationäkymässä, joka on nähtävissä **Kuva 17**. Kuvassa näkyy osille luodut animaatiot, joille voidaan antaa eri animaatioita kuvastavat nimet. Mallinnan ensin hissiosalle ja kelkalle ylösalaisen liikkeen, minkä jälkeen mallinnan kelkalle syvyysliikkeen, joka liikkuu kanavaan ja sieltä takaisin. Lopuksi mallinnan hissien rungolle sivuttaissuuntaisen liikkeen.

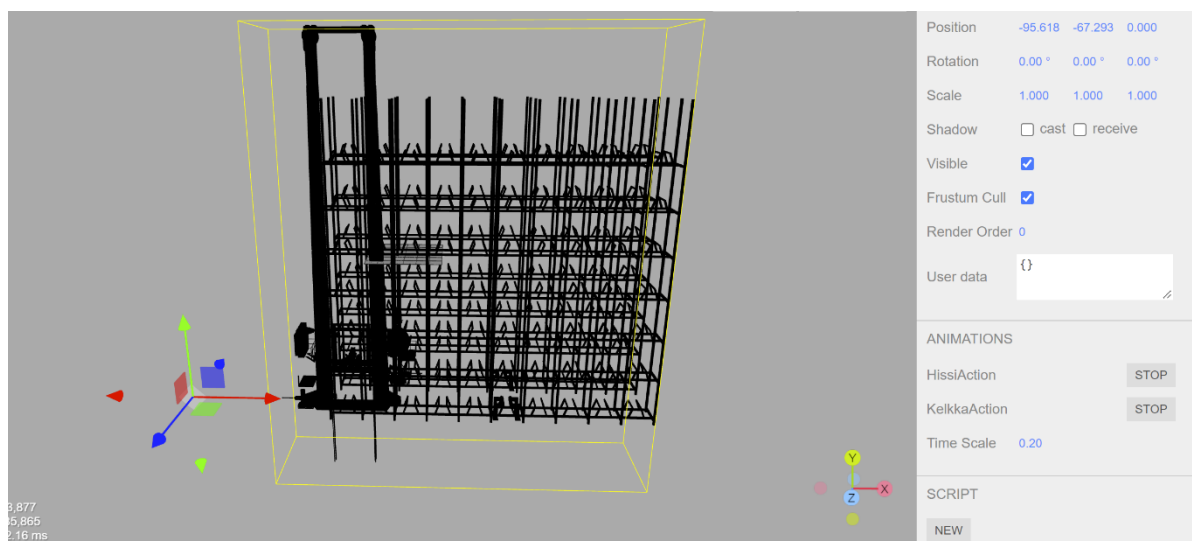


Kuva 16 Hissin animointi.



Kuva 17 Animointinäkymä Blenderissä.

Mallinnuksen jälkeen mallia on helppo tarkastella verkosta löytyvän ThreeJs-editorin avulla kuten alla olevasta **Kuva 18** näkyy.

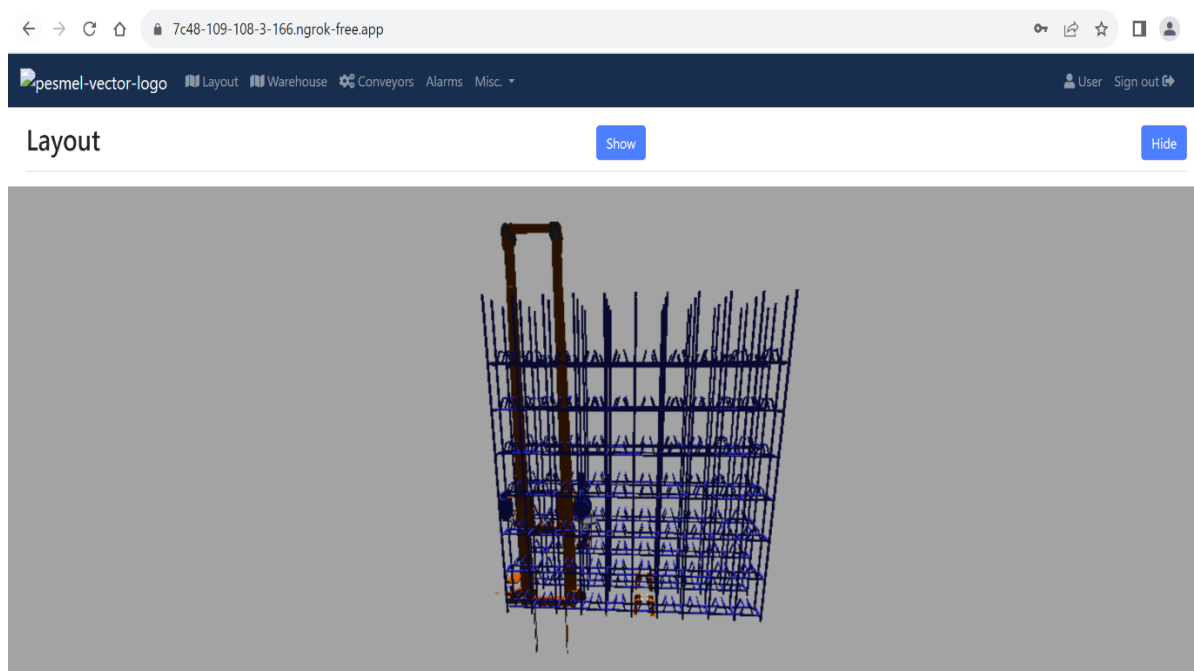


Kuva 18 Hissimallin rakenne ThreeJs-editorissa.⁵³

Mallin saaminen ja esittäminen näkymässä tapahtuu joko tuomalla (engl. import) se tiedostoista tai raahamalla ja tiputtamalla se suoraan editorin näkymään. ThreeJs-editorilla mallia voi katselmoida pyörittämällä sitä eri suuntiin ja liikuttelemalla sitä ylös ja alas tai eri suuntiin. Jos malli on tiedostossa jaettuna eri osiin, niitä osia pystyy liikuttelemaan tässä näkymässä erikseen. Kuvan oikeasta alakulmasta on havaittavissa "HissiAction" ja "KelkkaAction", joille on kaksi painiketta, jotka ovat aktiivisena joko play- tai stop-tilassa. Nämä ovat Blenderissä luotuja animaatioita, joita on helppo testata ThreeJs-editorilla. Tämän avulla on helppoa ja nopeaa kokeilla animaatioiden toimivuutta ennen varsinaista mallin käyttöä verkkosivuilla.

⁵³ ThreeJs-editori. Kuva 2023.

Hissimalli on nyt valmis vietäväksi verkkosivulle. Kuten aiemmin totesin, malleja on yksinkertaisin käsitellä glb. tiedostomuodossa. ThreeJs sisältää myös parhaat ominaisuudet kyseisten tiedostomuotojen käsittelyyn. Glb tarkoittaa glTF binary tiedostomuotoa, joka mahdollistaa mallin, sen tietojen ja tekstuurien siirtämisen samassa tiedostossa. glTF on englanniksi Graphics Language Transmission Format, jonka on kehittänyt OpenGL kehittäjä Khronos Group.⁵⁴ **Kuva 19** näkyy hissimalli Pesmelin käyttämään Ui-ympäristöön importattuna. Layout-sivu on alkuun kokonaan tyhjä, mutta show-painiketta painamalla käyttäjä saa hetken kuluttua kyseisen mallin näkymään. Hide-painiketta painamalla mallin voi puolestaan piilottaa.



Kuva 19 Hissimalli verkkosivulla.

⁵⁴ ThreeJs/glTF. 2023.

Malli tuodaan sivulle ThreeJs:n omaa .glTF-kirjastoa hyödyntäen. Mallin tuominen tyhjälle verkkosivulle vaatisi HTML-canvas-elementin, jotta voidaan määritellä mille alueelle malli halutaan tuoda. Tässä tapauksessa, kun malli tuodaan valmiille verkkosivulle, elementtiä ei vaadita. JavaScript-koodin puolella mallille luodaan hahmontaja, kamera ja uusi kohta (engl. new scene). WebGLRenderer toimii hahmontajana ja PerspectiveCamera kamerana. Mallin hahmontamiseksi tarvitaan hahmontamissilmukka. Käytän render-metodia määrittääkseni, mitä kohtausta halutaan piirtää ja mitä kameraa käytetään. Silmukassa käytän requestAnimationFrame-metodia varmistaakseni, että hahmontaminen tapahtuu jokaisen näytön päivityksen yhteydessä.

Ensimmäisillä testauskerroilla mallia oli vaikea saada näkyviin erinäisten ongelmien takia, jotka oli kuitenkin helppo selvittää ja ratkaista consolin avulla. Yleisimmät ongelmat johtuivat siitä, että .glb-mallin tiedostopolun rakenne oli väärä tai malli oli liian raskasrakenteinen .glTF-kirjaston ladattavaksi.

Testausten jälkeen malli tuotiin verkkosivulle onnistuneesti. Kameran näkökenttä ja sen erilaiset leikkaustasot vaativat hienosäätöä, jotta malli näkyisi paremmin. Nämä ominaisuudet ovat kuitenkin usein ohjelmoijan päätettävissä, että millä tavalla hän haluaa mallin esille. Asetan myös mallille zoomausmahdollisuuden OrbitControls-kirjaston avulla. Tämän avulla mallia voi loitontaa kauemmas tai zoomata lähemmäs hiiren rullaa vierittämällä.

Alla olevassa **Esimerkkikoodi 2** näkyy otanta koodirakenteesta, jonka funktio on ladata hissimallista luotu .glb-tiedosto hyödyntäen glTFLoader-nimistä kirjastoa. Esimerkkikoodissa glTFLoader-objekti alustetaan muuttujaan, jonka jälkeen sille asetetaan oikea tiedostosijainti, josta kyseinen malli löytyy .glb-muodossa. Mallille voidaan asettaa halutessaan paikkapisteet, jotka määrävät mallille tietyn kohdan, kun se lisätään kohtaukseen. Normaalisti tämä ei ole tarpeellista, jos kyseessä on vain yksi malli. Tässä kuitenkin on neljä erilaista objektiä liittyen hissien rakenteeseen, niin paikan määrittäminen on välttämätöntä. Jos tätä ei olisi, niin hissien runko saattaisi sijaita eri paikassa kuin esimerkiksi varsinainen hissiosa.

```
const ambientLight = new THREE.AmbientLight(0x333333);
scene.add(ambientLight);

const directionalLight = new THREE.DirectionalLight(0xFFFFFF, 0.8);
scene.add(directionalLight);

var loader = new GLTFLoader()
loader.load('assets/kehikko.glb', function (gltf) {
  kehikko = gltf.scene
  kehikko.position.setX(83.5)
  kehikko.position.setY(-10)
  kehikko.position.setZ(-30)
  scene.add(kehikko)
})

loader.load('assets/kraana.glb', function (gltf) {
  kraana = gltf.scene
  scene.add(kraana);
})

loader.load('assets/hissi.glb', function (gltf) {
  hissi = gltf.scene
  scene.add(hissi);
})

loader.load('assets/kelkka.glb', function (gltf) {
  kelkka = gltf.scene
  scene.add(kelkka);
})
```

Esimerkkikoodi 2 ThreeJs glTFLoader.

Pelkästään tällaisella rakenteella malli näkyisi mustana, jos kohtaukseen ei ole lisätty valoja. Kohtauksessani käytän kahdenlaista valaistusta. Ensimmäinen valaistus, jonka lisään kohtaukseen on AmbientLight. Tämä valotyyppi valaisee kattavasti kaikki kohtauksessa olevat objektit yhtä tasaisesti. Tätä ei voida kuitenkaan käyttää varjojen luomiseen, koska sillä ei ole suuntaa. Väri, jota käytän AmbientLightin kanssa, on harmaa. Toinen valotyyppi on DirectionalLight, joka suunnataan osoittamaan suoraan hissimalliin. Yleisin asetus tälle valotyypille on valkoinen väri ja voimakkuus 0,8, joten valitsin nämä arvot myös omaan työhöni.

Aluksi hissimallilla oli kaksi animaatiota, jotka oli tehty Blenderin avulla. Ongelmaksi muodostui hissien liikkeen monipuolisuus ja niiden kaikkien animointi. Lopputulema oli se, että Blender olisi liian epäkäytännöllinen sellaisiin animointeihin. Ratkaisu löytyi ThreeJs:n kirjastosta löytyvästä Tween-funktiosta. Se

ei kuitenkaan löydy suoraan ThreeJs:n kirjastosta, vaan se täytyy ladata ensin npm installerin avulla, jonka jälkeen sen voi lisätä kirjastoon ja ottaa käyttöön import-metodin avulla. Tween on animointityökalu, jota voi käyttää esimerkiksi 3D-mallien animoimiseen. Mallille täytyy määrittää aloitus- ja lopetuspiste, mutta animaation aikaiset välipisteet tweening-moottori määrittää itse. Tällä pystyy suhteellisen helposti animoimaan vertikaalisia ja horisontaalisia liikkeitä. Näiden lisäksi olen animoinut hissin kelkkaosalle syvyysuuntaista liikettä.

Kuva 19 on lopullinen valmis hissimalli renderöitynä verkkosivulle.

```
function randomMove () {

  var coordinatesX = [0, 12.5, 25.5, 38, 50.5, 63, 75.5, 88.5, 101, 113.5];
  var randomX = Math.floor(Math.random() * coordinatesX.length);
  var coordinatesY = [0, 15, 31, 47, 63, 80, 98];
  var randomY = Math.floor(Math.random() * coordinatesY.length);

  const tween1 = new TWEEN.Tween({x: 0, y: 0})
  .to({x: coordinatesX[randomX], y: coordinatesY[randomY]}, 5000)
  .onUpdate((coords) => {
    kraana.position.x = coords.x;
    hissi.position.x = coords.x;
    hissi.position.y = coords.y;
    kelkka.position.x = coords.x;
    kelkka.position.y = coords.y;
  });

  tween1.delay(3000);

  const tween3 = new TWEEN.Tween({z:0})
  .to({z:10}, 3000)
  .onUpdate((coords) => {
    kelkka.position.z = coords.z;
  });
  tween3.delay(1000);

  const tween4 = new TWEEN.Tween({z: 10})
  .to({z:0}, 4000)
  .onComplete (() => {
    randomMove()
  })
  .onUpdate((coords) => {
    kelkka.position.z = coords.z;
  });
}
```

Esimerkkikoodi 3 ThreeJs animaatio.

7 YHTEENVETO JA POHDINTA

Tämä opinnäytetyö käsitteli 3D-mallinnusta ja sen laajoja sovellusmahdollisuuksia eri aloilla. 3D-mallinnus on kehittynyt merkittävästi viime vuosikymmenten aikana, ja se tarjoaa monipuolisia työkaluja ja mahdollisuuksia niin teollisuudessa kuin viiheessäkin. Tämän opinnäytetyön tarkoituksena oli antaa kattava yleiskatsaus 3D-mallinnuksen perusteisiin, sen käyttötarkoituksiin ja tulevaisuuden potentiaaliin. Opinnäytetyössä muokattiin valmista hissimallia Blenderillä, jotta sen pystyisi esittämään verkkosivulla ThreeJs grafiikkakehyksen avulla. Yhtenä tutkimuksen kohteena oli tutustua kolmeen erilaiseen 3D-mallinnuksen työkaluihin ja löytää niistä sopivin verkkoympäristöön käytettäväksi. Mallin käsittely oli ennako-odotuksiin nähden kohtalaisen yksinkertaista. Opetusvideoita ja dokumentteja oli paljon ja ne olivat helposti saatavilla. Erilaisia näkökulmia on kuitenkin olemassa, koska paljon riippuu esimerkiksi siitä, millaisen mallin haluaa ylipäänsä mallintaa ja mitä eri ominaisuuksia sille haluaa asettaa.

Mallinnuksessa esiintyi kuitenkin lukuisia haasteita. Vaikeuksia ilmeni erityisesti silloin, kun toiminto oli monimutkaisempi, esimerkiksi johtuen useista varjostin- tai geometriamoduuleista. Joissakin tilanteissa toiminnon suorittaminen edellytti tiettyä tekniikkaa.

Myös 3D-objektin esittämisessä ilmeni ongelmia. Aluksi sivusto näytti vain tyhjän ruudun. Kuitenkin selaimen konsolin kautta voitiin päätellä, että ongelma ei liittynyt pelkästään mallin näkymättömyyteen, vaan koko Three.js-sovellus ei toiminut oikein. Mallin näkyvyysongelmat liittyivät pääosin siihen, että mallin tiedostopolku oli kirjoitettu ohjelmakoodiin väärin tai malli oli muokattu liian raskasrakenteiseksi Blenderissä, jolloin glTFLoader ei kyennyt sitä renderöimään verkkosivulle. Ongelma oli suhteellisen helppo selvittää ja ratkaista verkkosivuilla olevan inspect-elementin avulla ja sitä kautta konsolia tutkimalla. Malli saatiin lopulta näkyviin pienillä muokkauksilla.

Sovelluksella on hyviä jatkokehitysmahdollisuuksia Pesmel Oy:n varastohallintajärjestelmää ajatellen. Tämän avulla voitaisiin luopua yksinkertaisemmasta 2D-näkymästä ja muuttaa se kolmiulotteiseen suuntaan. Kolmiulotteinen rakenne antaa paljon yksityiskohtaisempaa tietoa laitteiden toimintaperiaatteista ja materiaalivirrasta myös asiakkaille. Tämä opinnäytetyö toimii myös tietopankkina kaikille, jotka ovat kiinnostuneita 3D-mallinnuksesta ja sen monipuolisista mahdollisuuksista. Opinnäytetyö tarjoaa yleiskatsauksen aiheeseen ja innostaa lukijaa tutkimaan 3D-mallinnuksen maailmaa syvemmin.

LÄHTEET

3dwebvision. 2016. Viitattu 26.6.2023

<https://3dwebvision.com/about-3d-models-on-websites/>

Autodesk. Maya. 2023. Viitattu 1.8.2023.

<https://www.autodesk.com/products/maya/overview?term=1-YEAR&tab=subscription&plc=MAYA>

Autodesk. 3ds Max. 2023. Viitattu 1.8.2023.

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

Autodesk. Focal length. 2023. Viitattu 1.8.2023.

<https://www.autodesk.com/support/technical/article/caas/sfdcarticles/sfdcarticle/How-to-create-a-camera-that-is-close-to-human-view-in-3ds-Max.html>

Babylon.Js kotisivut. 2023. Viitattu 14.8.2023.

<https://www.babylonjs.com/>

Blender mallinnustyökalun kotisivut. 2023. Blender. Viitattu 1.8.2023.

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

Costlow, Erik. 2021. The end of Applets. Viitattu 19.7.2023.

<https://www.infoq.com/news/2021/03/end-of-applets/>

3D-animation for product design. 2023. Viitattu 11.9.2023

<https://cgifurniture.com/3d-animation-for-product-design/>

Eastcott, Will & Nyman Robert. 2014. PlayCanvas Goes Open Source. Viitattu 14.8.2023.

<https://hacks.mozilla.org/2014/06/playcanvas-goes-open-source/>

Editor. PlayCanvas. 2023. Viitattu 14.8.2023.

<https://developer.playcanvas.com/en/user-manual/designer/>

Editor. Kuva. PlayCanvas. Viitattu 14.8.2023.

<https://developer.playcanvas.com/en/user-manual/designer/>

Entity. PlayCanvas. 2023. Viitattu 14.8.2023.

<https://developer.playcanvas.com/en/api/pc.Entity.html>

en.wikipedia.org. OpenGL. 2023. Viitattu 14.8.2023.

<https://en.wikipedia.org/wiki/OpenGL>

en.wikipedia.org. WebGL. 2023. Viitattu 14.8.2023.

<https://en.wikipedia.org/wiki/WebGL>

<https://facequad.com/blogs/post/what-is-3d-modeling>

facesquad/what-is-3d-modeling. 2023. Viitattu 11.9.2023.

Irwin, Emma. 2021. Microsoft Open-Source success story—Babylon. Viitattu 14.8.2023.

<https://cloudblogs.microsoft.com/opensource/2021/02/22/microsoft-open-source-success-story-babylon/>

Introduction to polygons. 2023. Viitattu 1.8.2023.

https://help.autodesk.com/view/MAYALT/2015/ENU/?guid=Polygons_overview_Introduction_to_polygons

Interactive rendering of NURBS surfaces. PDF. 2017. Viitattu 1.8.2023

<https://core.ac.uk/download/pdf/80522586.pdf>

khronos.org/webgl/wiki/getting started. 2011. Viitattu 14.8.2023.

https://www.khronos.org/webgl/wiki/Getting_Started

khronos.org/webgl. 2023. Viitattu 14.8.2023.

<https://www.khronos.org/webgl/>

Learn OpenGL/Getting started/Coordinate systems. 2023. Viitattu 14.8.2023

<https://learnopengl.com/Getting-started/Coordinate-Systems>

Meshes, Materials, Shaders and Textures. 2019. Viitattu 1.8.2023

<https://docs.unity3d.com/ru/2019.4/Manual/Shaders.html>

Making a simple game – part 1. PlayCanvas. 2023. Viitattu 14.8.2023.

<https://developer.playcanvas.com/en/tutorials/keepyup-part-one/#model-component>

Normal mapping. 2023. Viitattu 1.8.2023.

https://en.wikipedia.org/wiki/Normal_mapping

Normaalikartoitus. Kuva. 2022. Viitattu 1.8.2023.

<https://fi.wikipedia.org/wiki/Normaalikartoitus>

NURBS-mallinnus. 2023. Viitattu 1.8.2023

<https://www.3ds.com/store/cad/nurbs-modeling>

OpenGL/wiki/Getting started. 2022. Viitattu 14.8.2023.

https://www.khronos.org/opengl/wiki/Getting_Started

OpenGL/wiki/Fragment Shader. 2020. Viitattu 14.8.2023

https://www.khronos.org/opengl/wiki/Fragment_Shader

Pesmel/industry/metal-industry. 2023. Viitattu 12.6.2023

<https://pesmel.com/industry/metal-industry/>

PlayCanvas. Äänikomponentti. Kuva. 2023. Viitattu 14.8.2023.

<https://developer.playcanvas.com/en/user-manual/packs/components/sound/>

Ruudunkaappaus. Mazdausa. 2023. Viitattu 18.7.2023.

<https://www.mazdausa.com/>

Ruudunkaappaus. Firewatchgame. 2023. Viitattu 19.7.2023.

<https://www.firewatchgame.com/>

Ruudunkaappaus. Interaktiivinen kartta. Norjan sademetsärahassto. 2023.

Viitattu 19.7.2023.

<https://rainforest.arkivert.no/#kart>

Ruudunkaappaus. Techtarger/WMS. 2020. Viitattu 12.6.2023

<https://www.techtarget.com/searcherp/definition/warehouse-management-system-WMS>

Runefest's big reveals. 2014. Viitattu 19.7.2023.

<https://secure.runescape.com/m=forum/forums?254,255,410,65496148&showuser=ArcaneFox>

Rendering pipeline overview. 2022. Khronos Group. Viitattu 14.8.2023

https://www.khronos.org/opengl/wiki/Rendering_Pipeline_Overview

Sketch/what-is-parallax-effect. 2023. Viitattu 19.7.2023.

<https://www.sketch.com/blog/what-is-a-parallax-effect/>

Shankland, Stephen. 2013. It's about time RuneScape dumps Java for HTML5.

Viitattu 19.7.2023.

<https://www.cnet.com/tech/services-and-software/its-about-time-runescape-dumps-java-for-html5/>

Structure of ThreeJS app. Kuva. 2023. Viitattu 14.8.2023.

<https://dev.to/Oxfloyd/create-an-interactive-3d-portfolio-website-that-stands-out-to-employers-47gc>

Spline mallinnustyökalun kotisivut. 2023. Spline. Viitattu 1.8.2023.

<https://spline.design/>

Scripting. PlayCanvas. 2023. Viitattu 14.8.2023.

<https://developer.playcanvas.com/en/user-manual/scripting/>

Techtarget/WMS. 2020. Viitattu 12.6.2023

<https://www.techtarget.com/searcherp/definition/warehouse-management-system-WMS>

Techtarget/RFID. 2021. Viitattu 12.6.2023

<https://www.techtarget.com/iotagenda/definition/RFID-radio-frequency-identification>

ThreeJS Fundamentals. 2023. Viitattu 1.8.2023.

<https://threejs.org/manual/#en/fundamentals>

ThreeJS manual. 2023. Viitattu 14.8.2023.

<https://threejs.org/manual/>

ThreeJS. Primitives. 2023. Viitattu 14.8.2023.

<https://threejs.org/manual/#en/primitives>

ThreeJS. Creating a scene. 2023. Viitattu 14.8.2023

<https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene>

Using a physics engine. Babylon.Js. 2023. Viitattu 14.8.2023.

<https://doc.babylonjs.com/features/featuresDeepDive/physics/usingPhysicsEngine>

Vector and matrix. Kuva. 2023. Viitattu 14.8.2023.

<https://www.geeksforgeeks.org/r-matrices/>

What is 3D-lighting and how is it used in animation. 2022. Viitattu 1.8.2023

<https://darvideo.tv/dictionary/3d-lighting/>

What is the function of a camera in a 3D-modeling software. 2023. Viitattu 1.8.2023

<https://www.quora.com/What-is-the-function-of-a-camera-in-a-3D-modeling-software>

WebGL/Main_page. 2022. Viitattu 14.8.2023

https://www.khronos.org/webgl/wiki/Main_Page

WebGL overview. Kuva. 2023. Viitattu 14.8.2023

https://www.tutorialspoint.com/webgl/webgl_overview.htm

ThreeJs-editori. Kuva 2023. Viitattu 30.9.2023

<https://threejs.org/editor/>

ThreeJs/glTF. 2023. Viitattu 3.10.2023

<https://threejs.org/docs/#examples/en/loaders/GLTFLoader>