

Michael Karlsson

OPETUSPELIN SUUNNITTELU JA TO- TEUTUS ENSIHOITAJAOPISKELI- JOILLE

Opinnäytetyö

Tekniikan ammattikorkeakoulututkinto

Peliohjelmoinnin koulutus

2021



**Kaakkois-Suomen
ammattikorkeakoulu**

Tutkintonimike	Insinööri (AMK)
Tekijä/Tekijät	Michael Karlsson
Työn nimi	Opetuspelin suunnittelu ja toteutus ensihoitajaopiskelijoille
Toimeksiantaja	Kaakkois-Suomen ammattikorkeakoulu
Vuosi	2021
Sivut	38 sivua
Työn ohjaaja(t)	Niina Mässeli

TIIVISTELMÄ

Opinnäytetyön päätavoitteena oli suunnitella ja toteuttaa opetuspelejä ensihoitajaopiskelijoille. Tavoitteena oli, että peli olisi hyödyllinen ja käyttökelpoinen sekä opiskelijoille että tilaajalle. Työn kirjallisen osan tavoitteena on selvittää, millainen peli tarkoitukseen sopii sekä miten sellainen peli voidaan toteuttaa. Työssä kuvataan pelin suunnittelua sekä toteutunutta peliä. Tavoitteena on myös tehdä hyödyllisiä havaintoja ongelmista, joita pelinkehityksen aikana kohdataan. Ajatus opetuspelejä tekemisestä tuli kahdelta ensihoitajaopiskelijalta.

Opetuspeli toteutettiin yhteistyössä kahden ensihoitajaopiskelijan, Sara Hilliahon ja Niina Sormusen kanssa. He toimivat kehitystyössä asiantuntijoina ja vastasivat sisällön oikeaoppisuudesta. Lisäksi he antoivat kehitystyön aikana palautetta pelistä kohderyhmän näkökulmasta. Valmis peli toteutettiin selaimella käytettävänä WebGL-sovelluksena. Pelinkehityksessä työkaluina käytettiin Unity-pelimootoria sekä GitLab-versionhallintapalvelua. Pelinkehitysprosessi dokumentoitiin ja dokumentaatiota hyödynnettiin tutkimusongelman ratkaisuun.

Työn ensisijaisena tuloksena on valmis opetuspelejä. Se vastaa hyvin pelaajien tarpeisiin. Se on toteutettu asiantuntevasti. Toteutuksessa huomioitiin erilaisten opiskelijoiden tarpeet lisäämällä saavutettavuutta sekä huomioimalla erityisesti ei-pelaavien opiskelijoiden tarpeita. Teknisissä ratkaisuissa on huomioitu ensisijaisesti opiskelijaa ja pelinkehityksen sujuvuutta. Saavutettavuuteen liittyvä lainsäädäntö aiheutti pelinkehityksen aikana ongelmia, joita ei osattu odottaa. Osa näistä ongelmista oli liian vaikeita tai työläitä ratkaistavaksi osana tätä opinnäytetyötä.

Johtopäätöksissä todetaan, että opetuspelejä tehdessä saadaan merkittäviä hyötyjä, kun peli toteutetaan alusta asti yhteistyössä aiheen ammattilaisten ja pelin kohderyhmän kanssa. Lisäksi saavutettavuutta käsittelevään lainsäädäntöön kannattaa tutustua koska se koskettaa niin monia toimijoita. Sen esittämistä vaatimuksista on hyödyllistä keskustella tilaajan kanssa heti kuin uutta peliä aletaan suunnittelemaan. Suurimmat ongelmat kohdattiin, kun yritettiin vastata kaikkiin lainsäädännön esittämiin saavutettavuusvaatimuksiin. Muut tekniset- ja suunnitteluun liittyvät ongelmat jäivät työssä vähäisiksi.

Asiasanat: opetusohjelmat, pelisuunnittelu, saavutettavuus

Degree	Bachelor of Engineering
Author (authors)	Michael Karlsson
Thesis title	Design and implementation of an educational game for emergency care students.
Commissioned by	South-Eastern Finland University of Applied Sciences
Time	May 2021
Pages	38 pages
Supervisor	Niina Mässeli

ABSTRACT

The main objective of the thesis was to design and implement an educational game for emergency care students. The goal was to create a game that is useful and convenient to use for students and commissioner alike. The written part of the thesis aimed to investigate what kind of game would suit this need and demonstrate how it could be implemented and to examine problems that were met during the process. The idea of making an educational game came from of two emergency care students.

The game was implemented with the help of two emergency care students: Sara Hilliaho and Niina Sormunen. They acted as both experts in their field and representatives of the games target audience. The game was implemented as a web application using WebGL technology. It was created using the Unity game engine. GitLab service was used to provide version control for the project. The game development process was documented and used to find answers to the research problem.

The main result of this thesis is an educational game. The game meets the needs of the students by providing accurate information. The implementation considered the additional needs some students might have and the needs of non-gamer students. The technical solutions were decided primarily based on the needs of the students and the impact they had on the smoothness of the development process. Legislation relating to accessibility produced unexpected difficulties during the development process. Some of those difficulties were too laborious or difficult to overcome as a part of this thesis.

The main conclusion of this thesis is that there were major advantages in involving the professionals of the subject and the target group of the game at an early stage when implementing an educational game. Legislation relating to accessibility should be familiarized because it applies to so many institutions and organizations. The legal requirements should be discussed with the commissioner immediately when initiating a new game project. The biggest problems in the development process were related to legislation and accessibility. Problems relating to other technical issues or game design were surprisingly minor.

Keywords: accessibility, educational software, game design

SISÄLLYS

1	JOHDANTO.....	6
2	KEHITTÄMISTYÖN TIETOPERUSTA.....	7
2.1	Opinnäytetyön tavoitteet ja tarkoitus.....	7
2.2	Eettisyys ja yhteistyö	7
2.3	Tutkimusmenetelmät	8
2.4	Muita aihetta sivuavia opinnäytetöitä	9
2.5	Tutkimuskysymykset.....	10
3	TEKNOLOGIAT	10
3.1	Git	10
3.2	Unity	12
3.3	WebGL	13
3.4	Opetuspelit ja saavutettavuus.....	14
4	SUUNNITTELU.....	15
4.1	Pelin tavoitteiden tarkastelu	15
4.2	Pelin suunnittelu	16
4.3	Peli-idea.....	17
4.4	Julkaisualustan valinta.....	18
4.5	Työkalujen valinta	19
4.5.1	Versionhallinta	19
4.5.2	Unity WebGL	20
5	TOTEUTUS	22
5.1	Pelimekaniikka ja pelikokemus	22
5.2	Tavoitteet.....	24
5.3	Skenaariot	25
5.4	Tietovisa	26
5.5	Saavutettavuus.....	27
6	TESTAUS	27

6.1	Testausmenetelmät	27
6.2	Toimintavirheet ja poikkeamat	29
7	TULOKSET	30
7.1	Johtopäätökset	33
7.2	Jatkokehitys	34
8	POHDINTA	34
	LÄHTEET	36

1 JOHDANTO

Tämän opinnäytetyön aiheena on opetuspelin suunnittelu ja toteutus ensihoitajaopiskelijoille. Ajatus opetuspelin tekemisestä tuli ensihoitajaopiskelijoilta Sara Hilliaholta sekä Niina Sormuselta. Tarkoituksena oli, että pelissä opetettiin heidän keräämäänsä materiaalia sosiaalisten hätätilanteiden tunnistamisesta ensihoidossa.

Pelien käyttämistä opetukseen voidaan perustella havainnoilla siitä, miten omistautuneesti ja innokkaasti monet pelaajat suhtautuvat peleihin vapaa-ajallaan. Opetuspelin idea on pyrkiä valjastamaan pelin viihteellisyys ja lisätä siihen opetuksellinen elementti. Pelejä käyttämällä opetuksesta voidaan myös tehdä kokemuksellisempaa ja osallistavampaa. Tästä voi olla hyötyä esimerkiksi oppilaalle, jolla on oppimisvaikeuksia perinteisempien opetusmenetelmien kanssa. (Beavis ym. 2017, 51)

Peli toteutettiin selaimella käytettäväksi WebGL-sovellukseksi käyttäen Unity-pelimootoria. Versionhallintaan käytettiin Git-versionhallintajärjestelmää. Opinnäytetyössä käydään läpi tarkemmin edellä mainittuja teknologioita. Lisäksi opinnäytetyössä tutustutaan lakiin digitaalisten palvelujen tarjoamisesta (Laki digitaalisten palvelujen tarjoamisesta 15.03.2019/306), joka asettaa konkreettisia vaatimuksia verkkosisällön saavutettavuudesta viranomaisille sekä julkisoikeudellisille toimijoille.

Opinnäytetyö on toteutettu Kaakkois-Suomen ammattikorkeakoululle eli Xamkille. Xamkilla on kampukset neljällä paikkakunnalla: Kotkassa, Kouvolassa, Mikkelissä ja Savonlinnassa. Vuosina 2019–2020 Xamk tarjosi yhteensä 70 erilaista ammattikorkeakoulu- tai ylempään ammattikorkeakoulututkintoon johtavaa koulutusta ja siellä opiskeli noin 9500 tutkinto-opiskelijaa. (Kaakkois-Suomen ammattikorkeakoulu 2020.)

Ensihoitajan tutkintoon johtavaa koulutusta järjestetään Kaakkois-Suomen ammattikorkeakoulussa Kotkan kampuksella. Nelivuotisen koulutuksen jälkeen opiskelija valmistuu ensihoitotyön asiantuntijaksi ja saa pätevyyden työl-

listyä myös esimerkiksi sairaanhoitajaksi. Ensihoitajan tehtäviin kuuluu kohdata hädässä oleva ihminen ensimmäisenä. Ensihoitaja joutuu työssään haastaviin ja nopeasti muuttuviin tilanteisiin. Tähän opiskelijaa pyritään valmistamaan muun muassa käyttämällä opetuksessa hyväksi simulaatioharjoittelutiloja sekä virtuaaliympäristöjä. (ENSIHOITAJA (AMK) s.a.)

2 KEHITTÄMISTYÖN TIETOPERUSTA

Tässä luvussa käydään läpi opinnäytetyön tietoperusta. Luvun tarkoitus on antaa lukijalle johdantoa yksityiskohtaisempi kuvaus työ alkuasetelmasta, tarkoituksesta sekä tavoitteista.

2.1 Opinnäytetyön tavoitteet ja tarkoitus

Tämä opinnäytetyön tarkoituksena on suunnitella sekä toteuttaa peli, jota ensihoitajaopiskelijat (AMK) voisivat käyttää itseopiskelumateriaalina. Pelin aihealueena on ensihoidossa kohdattavien sosiaalisten hätätilanteiden tunnistaminen. Tavoitteena on että, kehitettävä peli olisi käyttäjille hyödyllinen sekä viihdyttävä. Valmiin pelin tulisi myös olla tilaajan hyödynnettävissä mahdollisimman helposti ja pitkään.

Opinnäytetyön kirjallisen raportin tarkoituksena on selvittää, minkälainen peli vastaa tilaajan ja opiskelijoiden tarpeeseen, ja miten sellaisen pystyy toteuttamaan käyttämällä työhön valittuja teknologioita. Tavoitteeseen on tarkoitus päästä keräämällä tietoa Unity-pelimoottorista ja WebGL-ohjelmointirajapinnasta sekä lisäksi raportoida huomionarvoisia asioita, joita pelinkehityksen aikana kohdataan. Lisäksi opinnäytetyössä esitellään työssä tehty peli sekä avataan pelinkehityksen eri vaiheita ja niiden aikana tehtyjä valintoja. Tavoitteena on, että opinnäytetyö toimii hyödyllisenä dokumenttina ainakin lukijalle, joka haluaa tutustua opetuspelin kehitysprosessiin.

2.2 Eettisyys ja yhteistyö

Koska tämä opinnäytetyö on luonteeltaan toiminnallinen, on sen eettisyyden kannalta merkittävää, että saadaan aikaiseksi käyttökelpoinen tuote. Ensihoidon asiaosaamisessa onnistuminen on opinnäytetyön hyödyllisyyden kannalta

oleellista, koska peliä on tarkoitus käyttää ensihoitajaopiskelijoiden kouluttamiseen. Opinnäytetyön tekemissä olikin tästä syystä mukana kaksi ensihoitajaopiskelijaa, jotka olivat tutustuneet tarkemmin pelin käsittelemään aiheeseen.

Tässä työssä käytetään ensihoitajaopiskelijoiden Sara Hilliahon ja Niina Sormusen tuottamaa materiaalia ja tutkimustyötä sekä heidän osaamistaan omalla alallansa. Asiantuntemuksen lisäksi myös heidän näkemyksiä pelin käyttäjäkokemuksesta huomioidaan koko pelinkehitysprosessin ajan. Näin varmistetaan sitä, että pelin kohdeyleisön käyttäjäkokemus huomioidaan riittävästi jo suunnitteluvaiheessa. Hilliahon ja Sormusen osallisuus valmiiseen tuotteeseen tunnustetaan myös lisäämällä heidät pelin ns. tekijät -näkömään, muiden tekijöiden tavoin. Myöhemmin pelinkehityksen aikana mukaan tiimiimme liittyi graafikon rooliin Noora Jantunen, joka teki pelin lopulliset grafiikat.

Opinnäytetyön aihetta tutkittaessa tutustutaan taustamateriaaliin. Materiaaleiksi valitaan alkuperäisiä ja mahdollisimman uusia lähteitä. Vanhempia asioita kuten esimerkiksi teknologioiden historioita käsiteltäessä myös vanhempia lähteitä voidaan käyttää. Lähteisiin viitataan tekstissä ja opinnäytetyön lopussa on lähdeluettelo.

2.3 Tutkimusmenetelmät

Opinnäytetyössä käytetään soveltavia sekä empiirisiä tutkimusmenetelmiä. Työssä pyritään hyödyntämään aikaisemmissa opinnäytetöissä kerättyjä kokemuksia sekä tietoa, uuden sovelluksen kehittämiseksi. Lisäksi tässä opinnäytetyössä kerätään omia havaintoja opetuspelin kehittämisestä sekä Unity-pelimoottorin WebGL-ominaisuuksien hyödyntämisestä verkkoselainpohjaisen opetuspelin tekoon.

Soveltava tutkimus valittiin tutkimusmenetelmäksi koska työn päätavoite on opetuspelin tekeminen. Työssä keskitytään ensisijaisesti käytännön sovelluksen toteuttamiseen hyödyntäen olemassa olevaa tietoa. Empiiriset tutkimusmenetelmät valittiin, koska työn päätavoitteisiin on tarkoitus vastata hyödyntäen työn aikana kertyneitä kokemuksia pelinkehityksestä.

2.4 Muita aihetta sivuavia opinnäytetöitä

Tätä opinnäytetyötä tehdessä tutustuttiin muutamiin muihin tämän opinnäytetyön kaltaisiin tai tämän työn aihetta sivuaviin opinnäytetöihin. Muita opinnäytetöitä tutkimalla löydettiin tähän työhön tuore ja erilainen näkökulma tarkastella aihetta.

Polvela S. toteutti opinnäytetyön: Mökin esittelysovelluksen toteuttaminen (2020). Opinnäytetyössä valmistui Unity-pelimootorilla toteutettu WebGL-sovellus, kuten tässäkin opinnäytetyössä tulee valmistumaan. Polvela kertoo työssään muun muassa, kuinka sovellusta tehdessä selvisi, että WebGL-sovellus toimii osassa puhelimista, vaikka ei olekaan niissä täysin tuettu. Mobiilissa puhelimen rajattu laskentateho kuitenkin haittasi sovelluksen sujuvaa käyttöä. Työssä päädyttiinkin kehittämään sovellus pääasiassa PC-käyttäjien ehdoilla. Polvela korostaa erityisesti optimoinnin tärkeyttä WebGL-sovellusta tehdessä. Muuten WebGL-teknologia osoittautui työhön hyvin sopivaksi ratkaisuksi.

Solomon F. (2018) toteutti myös opinnäytetyönään opetuspelin käyttäen Unity-pelimootoria. Opinnäytetyössä pelistä tehtiin WebGL-version verkkosivulle sekä PC-laitteelle asennettava natiivi versio. Opinnäytetyössä WebGL-version eduiksi todettiin, että se on helposti käyttäjien saatavilla. Lisäksi peliä ei tarvitse asentaa käyttäjän koneelle ja se toimii erilaisilla laitteistoilla. Haittapuolena WebGL-versiossa nähtiin, että käyttäjän internet-yhteys ja laitteisto vaikuttavat siihen, kuinka sujuvasti peli toimii. Lisäksi testaaminen ja virheiden löytäminen sekä korjaaminen, on hankalampaa kuin PC-versiossa. Myös Solomon korosti työssään optimoinnin tärkeyttä WebGL-sovellusta tehdessä, jotta sovellus toimisi sujuvasti ja luotettavasti.

Kolmanneksi tarkasteltavaksi opinnäytetyöksi valittiin Huopalaisen (2020) opinnäytetyö: Pelin suunnittelu ja toteutus Dreams-kehitysympäristössä. Huopalainen toteutti työnsä hyödyntäen eri teknologioita, kuin mitä tässä opinnäytetyössä käytettiin. Opinnäytetyöt ovat kuitenkin molemmat toiminnallisia opinnäytetöitä, joissa syntyi tuotteena raportin lisäksi myös pelattava peli. Huopalaisen työ valikoitui tarkasteltavaksi, koska siinä avataan merkittävästi enemmän pelin sisällön suunnittelua ja sen vaikutusta käyttäjäkokemukseen kuin

esimerkiksi edellisessä kappaleessa mainitussa Solomonin työssä. Lisäksi Huopalaisen työssä on selkeällä tavalla kuvattu valmistunutta peliä ja sen ominaisuuksia. Työ toimiikin hyvänä esimerkkinä siitä, miten tämän opinnäytetyön tehdessä syntyneitä tuloksia voidaan esittää.

2.5 Tutkimuskysymykset

Tässä opinnäytetyössä suunnitellaan sekä kehitetään opetuspelejä itseopiskelumateriaaliksi. Seuraavat tutkimuskysymykset on johdettu tästä työn alkuasetelmasta sekä työn tavoitteista:

- Millainen peli sopii ensihoitajille (AMK) työkaluksi itseopiskeluun?
- Miten tällainen peli voidaan toteuttaa?
- Minkälaisia haasteita tai ongelmia pelinkehityksen aikana kohdattiin?

Kysymyksiin etsitään vastauksia tutkimalla luotettavia lähteitä. Opinnäytetyön osana tehtävän pelin kehittäminen konkretisoi vastauksia, ja esimerkin muodossa avaa mahdollisuuksia verrata käytäntöä teoriaan.

3 TEKNOLOGIAT

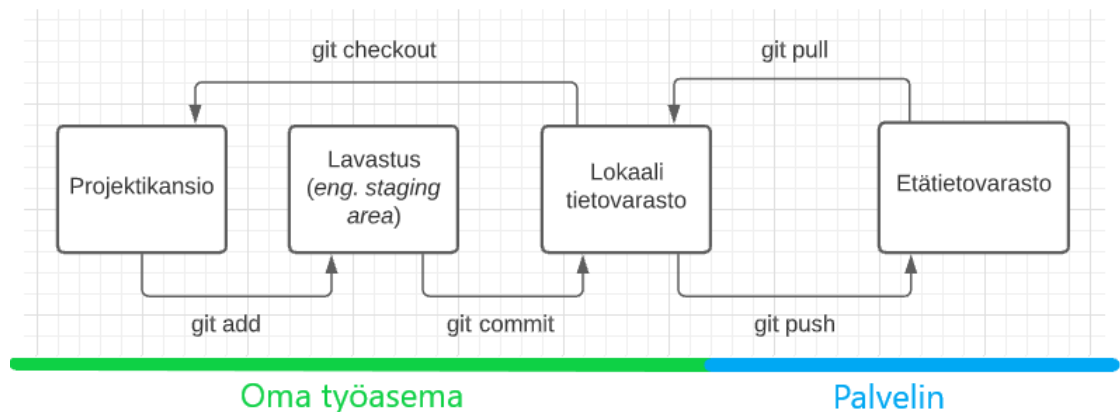
Tässä luvussa käydään läpi tämän opinnäytetyön kannalta keskeiset teknologiat Git-versionhallintajärjestelmä, WebGL-ohjelmointirajapinta sekä Unity-pelimoottori. Lisäksi luvussa käsitellään opetuspelejä ja saavutettavuusvaatimuksia. Myöhemmin luvussa 4 kerrotaan miten ja miksi työssä päädyttiin käyttämään edellä mainittuja teknologioita.

3.1 Git

Git-versionhallintajärjestelmä on Linus Torvaldsin vuonna 2005 kehittämä ilmainen, avoimen lähdekoodin versionhallintajärjestelmä. Git kehitettiin työkaluksi tukemaan Linux-käyttöjärjestelmän kernelin eli ytimen kehittämistä, kun muut sen aikaiset työkalut eivät enää vastanneet kehitystyön tarpeisiin. (Loliger & McCullough 2012, 2) Se on sittemmin kohonnut suosiossaan ja osoit-

tautunut arvokkaaksi työkaluksi monissa muissa pienissä ja suurissa projekteissa. Merkittäviä suuryrityksiä, jotka käyttävät Git-versionhallintajärjestelmää ovat esimerkiksi Google, Facebook sekä Microsoft (Git s.a.).

Git-versionhallintajärjestelmän toimintaperuste on hieman yksinkertaistettuna seuraavanlainen: Aluksi luodaan tietovarasto (engl. repository). Tietovarasto voidaan luoda tyhjänä tai se voidaan luoda olemassa olevan projektin tai asiakirjan pohjalta. Yleensä on käytännöllistä luoda tietovarasto palvelimelle käyttäen jonkin palveluntarjoajan kuten esimerkiksi Github- tai GitLab-palvelun työkaluja ja palvelimia. Tällöin Git-versionhallintajärjestelmä mahdollistaa sen, että usea käyttäjä voi muokata projektia samanaikaisesti. Seuraavaksi omalle työasemalle kloonataan juuri luotu tietovarasto. Myös tietovaraston kloonin pitää sisällään projektin koko versiohistorian eli jokainen ajan tasalla oleva kloonin tietovarastosta toimii varmuuskopiona. (Loliger & McCullough 2012, 19–24.)



Kuva 1: Tärkeimmät komennot ja niiden käyttö Git-versionhallintajärjestelmässä

Kuvassa 1 on esitetty Git-versionhallintajärjestelmän tärkeimmät komennot ja niiden toiminta. Kuvassa projektikansio on hakemisto, jota versionhallintajärjestelmä tarkkailee, ja jossa käyttäjän projekti sijaitsee. Käyttäjän tekemät muutokset eivät siirry itsestään projektikansioista tietovarastoon vaan ne siirretään komentojen avulla. Käyttäjän ei tarvitse siirtää kaikkia tekemiään muutoksia tietovarastoon vaan hän valitsee haluamansa muutokset siirrettäväksi ensin ns. lavastus tilaan git add -komennolla. Valintojen jälkeen kaikki lavastetut muutokset viedään tietovarastoon yhdellä kertaa käyttämällä git commit -komentoa. Samalla käyttäjän on kommentoitava tekemänsä muutokset. Git

push -komennolla lokaalin tietovaraston muutokset kopioidaan ns. etätietovarastoon. Ulkoisella palvelimella sijaitsevan tietovaraston käytöllä huolehditaan projektin varmuuskopioinnista sekä mahdollistetaan sujuva yhteistyö. Jos projektin parissa työskentelee muita ihmisiä, heidän tekemänsä muutokset voidaan hakea omalle laitteelle git pull -komennolla. Git checkout -komennolla voidaan palauttaa aikaisempia versioita projektista projektikansioon tai vaihtaa projektin kehityshaaraa, mikäli projektista on luotu useampia haaroja. (Git 2021)

Git mahdollistaa ja kannustaa projektin jakamista erilaisiin kehityshaaroihin. (Loliger & McCullough 2012, 3) Kehityshaara voidaan irrottaa päähaarasta ja sitä voidaan kehittää omana kokonaisuutenaan. Kehityshaarat mahdollistavat uusien ominaisuuksien kehittämisen ja testaamisen versionhallinnan piirissä, ilman että, päähaaraan tehdään muutoksia. Ne mahdollistavat myös usean erilaisen ominaisuuden samanaikaisen kehittämisen toisistaan riippumattomasti. Omassa haarassa kehitetyt ominaisuudet voidaan yhdistää takaisin päähaaraan, kun ne ovat valmiita. (Loliger & McCullough 2012, 89–2.)

3.2 Unity

Unity-pelimoottori on alun perin pelinkehitykseen luotu pelimoottori. Pelimoottori on ohjelmistokehys, joka sisältää valmiita työkaluja esimerkiksi fysiikan mallintamiseen ja grafiikan piirtämiseen näytölle (Kuorikoski 2018, 236). Valmiiksi olemassa olevien työkalujen käyttäminen nopeuttaa huomattavasti pelinkehitystä. Unity-pelimoottoria on kehitetty ja brändätty myös tuotteena, jota voidaan käyttää työkaluna myös muunlaisten medioiden luomiseen. Sitä on käytetty muun muassa visualisoimaan arkkitehtien suunnitelmia hyödyntäen lisättyä todellisuutta sekä jopa elokuvantekoon. (Welcome to Unity s.a.)

Unity-pelimoottori on rojaltivapaa ja ilmainen yksityiselle henkilölle, jos sen avulla hankitut tulot alittavat 100 000 \$ vuodessa. Sillä voi kehittää sovelluksia yli kahdellekymmenelle eri alustalle puhelimista pelikonsoleihin. (Unity Personal s.a.)

3.3 WebGL

WebGL on API eli ohjelmointirajapinta, jota käytetään tuottamaan interaktiivista 3D-grafiikkaa verkkoselaimessa. Se perustuu OpenGL ES 2.0 -teknologiaan, jota käytetään tavallisesti esimerkiksi C/C++ -kontekstissa, kun taas WebGL-ohjelmointirajapintaa käytetään JavaScript- ja HTML-kontekstissa. (Anyuru 2012, 1) WebGL-ohjelmistorajapinta on sisäänrakennettu moniin nykyaikaisiin verkkoselaimiin ja on hyödynnettävissä ilman että, käyttäjän tarvitsee lisätä liitännäisiä selaimensa. WebGL-ohjelmistorajapintaa hyödyntämällä, selaimella oleva sovellus pystyy hyödyntämään käyttäjän laitteen näytönohjainta. Tämä antaa lisää laskentatehoa selaimen käyttöön, ja mahdollistaa esimerkiksi pelin, tai kolmiulotteisen karttasovelluksen käytön selaimella. WebGL-ohjelmistorajapintaa käytetään HTML5 canvas -elementin kautta. Canvas-elementti yksinään ei mahdollista interaktiivista 3D-grafiikkaa. (Anyuru 2012, 24)

WebGL on rojaltivapaa avoimen standardin teknologia, jota pystyy hyödyntämään sitä tukevissa verkkoselaimissa. Sitä hallinnoi ja kehittää Khronos Group. Khronos Group on yli 150 yrityksestä koostuva yhteenliittymä, jonka tehtävä on kehittää rojaltivapaita standardeja mahdollistamaan erilaisten systeemien yhteensopivuuksia. Mukana yhteenliittymässä ovat selaintenkehittäjistä muun muassa Apple, Google, Mozilla, Opera sekä Microsoft. Lisäksi mukana on monia tietokoneen osia ja komponentteja valmistavia yrityksiä. Näin ollen WebGL-ohjelmointirajapinta onkin tällä hetkellä tuettu suurimmassa osassa uusista selaimista. (WebGL Overview s.a.)

WebGL 1.0 on julkaistu jo maaliskuussa 2011 (Khronos Releases... s.a.). Sen seuraajaksi on kaavailtu WebGPU-standardia. Se on WebGL-standardin tavoin selaintenkehittäjien ja tietokoneenkomponenttiensuostajien yhteinen projekti. Se on suunniteltu muun muassa suorittamaan vähemmän kutsuja tietokoneen prosessorille ja hyödyntämään paremmin näytönohjaimien rakennetta toimintaperiaatteessaan. Valmistuessaan WebGPU-standardin kaavailaan mahdollistavan kunnianhimoisempia ja sujuvammin toimivia applikaatioita. Toistaiseksi se on kuitenkin vielä kehitysvaiheessa. (Wallez 2020.) Selainkehittäjät tulevat jatkossa käyttämään enemmän resursseja WebGPU-

standardin kehittämiseen ja siirtymään WebGL-standardin tukemisessa ylläpitovaiheeseen. Khronos Group -yhteenliittymässä uskotaan WebGL-tekniikan olevan kuitenkin täällä pysyäkseen ja näiden kahden tekniikan kehittyvän rintarinnan (Khronos Group. 2020). Tämä vaikuttaa mielestäni turvalliselta olettamukselta, koska uuden tekniikan täysi hyödyntäminen vaatii tässä tapauksessa selaintenvalmistajien kehitystyön lisäksi käyttäjien laitekannan uudistumista (Wallez 2020).

3.4 Opetuspelit ja saavutettavuus

Opetuspelien tarkoituksena on opettaa pelaajalle tiettyä, ennalta määritettyä aineistoa, ja motivoida häntä opiskelemaan (Manninen 2007, 22). Pelien käyttämistä opetukseen voidaan perustella havainnoilla siitä, miten omistautuneesti ja innokkaasti monet pelaajat suhtautuvat peleihin vapaa-ajallaan. Opetuspelin idea on pyrkiä valjastamaan pelin viihteellisyys ja lisätä siihen opetuskellinen elementti. Pelejä käyttämällä opetuksesta voidaan myös tehdä kokemuksellisempaa ja osallistavampaa. Tästä voi olla hyötyä esimerkiksi oppilaalle, jolla on oppimisvaikeuksia perinteisempien opetusmenetelmien kanssa. (Beavis ym. 2017, 51.)

Viranomaisten ja muiden julkisoikeudellisten laitosten tarjoamaa verkkosisältöä ohjaavat Suomessa useat lait. Laki digitaalisten palvelujen tarjoamisesta (Laki digitaalisten palvelujen tarjoamisesta 15.03.2019/306) asettaa konkreettisia vaatimuksia verkkosisällön saavutettavuudesta. Sen yhtenä tarkoituksena on antaa erilaisille käyttäjille yhtäläiset mahdollisuudet hyödyntää digitaalisia palveluita. Lain 12. §:ssä todetaan, että lakia valvoo Etelä-Suomen aluehallintovirasto ja että sen tehtävänä on valvonnan lisäksi, antaa ohjausta säädettyjen vaatimusten noudattamisesta. Saavutettavuusvaatimukset määritellään Euroopan unionissa ja niiden suomenkieliset viralliset käännökset ovat lain mukaan saatavilla osoitteesta www.saavutettavuusvaatimukset.fi (Laki digitaalisten palvelujen tarjoamisesta 15.03.2019/306, 7. §).

Laki digitaalisten palvelujen tarjoamisesta (Laki digitaalisten palvelujen tarjoamisesta 15.03.2019/306) edellyttää WCAG-ohjeistuksen uusimman version kriteerien noudattamista, poikkeuksena AAA-tason kriteerit sekä kriteerit koskien suoria verkkolähteyksiä. Noudatettavia kriteerejä on 49 ja ne on jaoteltu

neljän otsikon alle: havaittavuus, hallittavuus, ymmärrettävyys sekä toimintavarmuus (WCAG 2.1: lain vaatimukset s.a.).

Laki digitaalisten palvelujen tarjoamisesta (Laki digitaalisten palvelujen tarjoamisesta 15.03.2019/306) koskee kaikkia lain piiriin kuuluvien toimijoiden verkkosivustoja ja mobiilisovelluksia. Lain kannalta ei ole merkitystä, onko palvelu toimijan sisäisessä verkossa vai ei. Myöskään palvelun kohderyhmä ei vaikuta lain soveltamiseen. Opetussisältö saa poiketa vaatimuksista siinä tapauksessa, jos sitä käytetään rajatussa ryhmässä ja vain tilapäisesti. (Mitä palveluja ja sisältöjä laki koskee? s.a.)

4 SUUNNITTELU

Opinnäytetyössä valmistuvalle pelille oli tilaajanpuolelta asetettu tavoitteeksi, että peli tulisi olemaan opetuskäyttöön tarkoitettu, sitä tulisi olla mahdollisimman kätevä käyttää ja sen tulisi olla saatavilla kaikille opiskelijoille. Tässä luvussa kuvataan, miten pelin suunnitteluprosessi eteni, ja miten siinä huomioitiin pelille asetetut tavoitteet sekä vaatimukset.

4.1 Pelin tavoitteiden tarkastelu

Pelin suunnittelu aloitettiin tarkastelemalla materiaalia, jota pelissä on tarkoitus opettaa. Pelin aiheena on sosiaalisten hätätilanteiden tunnistaminen sekä niiden omien ilmoitusvelvollisuuksien hallitseminen. Opetettava asia jakautuu viiteen pääkohtaan, jotka liittyvät erilaisiin sosiaalisiin hätätilanteisiin (lapsiin, asuinoloihin, turvallisuuteen, päihteisiin, sekä henkiseen traumaan liittyvät). Opetuspelinä tämän pelin tulisi opettaa ja kannustaa pelaaja perehtymään aiheeseen interaktiivisella tavalla (Manninen 2007, 22). Ideoimalla yhdessä materiaalin tekijöiden kanssa tunnistettiin seuraavanlaisia tavoitteita, jotka vaikuttavat pelisuunnitteluun:

- Pelaajan tulisi oppia tunnistamaan erilaiset sosiaaliset hätätilanteet
- Pelaajaa halutaan kannustaa tekemään hätätilanteiden tunnistamisen kannalta oleellisia havaintoja
- Pelaajan tulisi oppia, kenelle hän on ilmoitusvelvollinen eri tilanteissa

Pelinkehityksen edetessä tunnistettiin lisäksi, että pelin tulisi sopia eritasoisille pelaajille. Erityisesti havahduttiin siihen, että kaikilla opiskelijoilla ei välttämättä ole lainkaan kokemusta pelaamisesta mutta myös heidän tulee pystyä yhtä lailla hyötymään pelistä. Pelisuunnittelussa ei siis voida täysin tukeutua erilaisiin peleissä vakiintuneisiin käytäntöihin ja olettaa pelaajan ymmärtävän miten hänen tulee toimia. Aikaisempaan listalle tavoitteista lisättiin:

- Opiskelijoiden tulisi pystyä hyötymään pelin koko sisällöstä, riippumatta heidän taidoistaan tai aikaisemmista pelikokemuksista

Asetettujen tavoitteiden lisäksi huomioitiin, että suunnitelma ei myöskään saisi olla liian kunnianhimoisen, jotta peli valmistuisi ajallaan. Suunnittelun alkuvaiheessa ei vielä ymmärretty ottaa huomioon lainsäädännön vaatimuksia saavutettavuudelle, vaikkakin saavutettavuutta huomioitiin projektissa alusta asti muista syistä.

4.2 Pelin suunnittelu

Pelille asetettujen tavoitteiden jälkeen siirryttiin suunnittelemaan millainen peli vastaisi niihin. Suunnittelun tarkoituksena on luoda pelisuunnitelma, jonka perusteella peliä ryhdytään toteuttamaan (Manninen 2007, 29).

Peliä ei suunniteltu heti alusta loppuun asti vaan hyödynnettiin toistuvaa suunnittelu-kehitys-testaussykliä (Kuva 2):



Kuva 2: Suunnittelu-kehitys-testaussyklissä testauksen tulos vaikuttaa suunnitelmaan (Manninen 2007, 71, muokattu)

Suunnittelu-kehitys-testaus -syklissä suunnittelu aloitetaan alustavalla suunnittelulla. Alustava suunnitelma toteutetaan, testataan ja arvioidaan. Arvioinnin perusteella suunnitelmaa muutetaan. Uusi suunnitelma toteutetaan ja sykliä

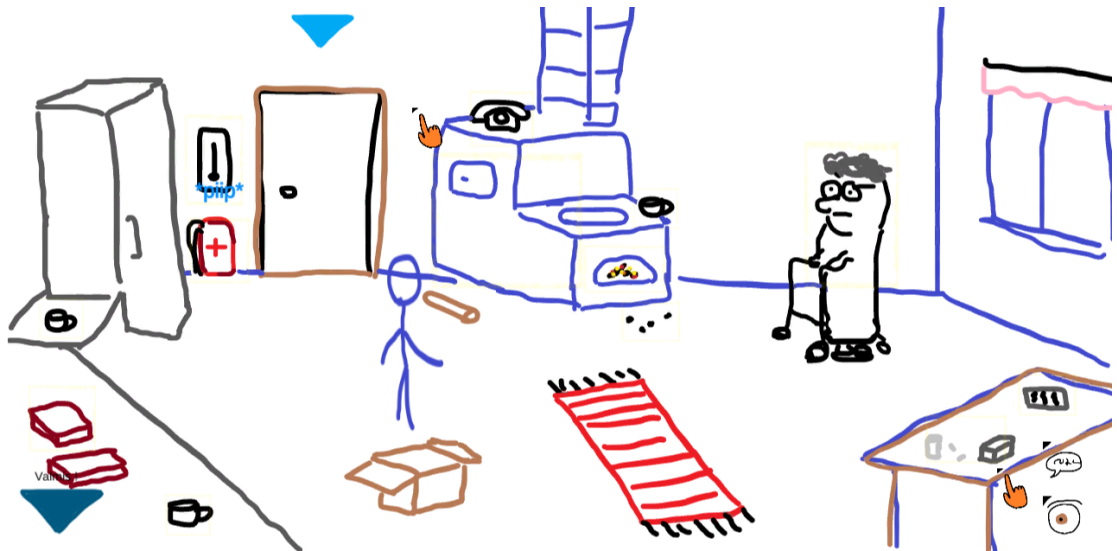
käydään läpi, kunnes toteutukseen ollaan tyytyväisiä. Yllä esitetty suunnittelu-kehitys-testaussykli poikkeaa niin sanotusta yritys–erehdys-menetelmästä siten, että syklissä toteutetaan vain suunniteltuja asioita. (Manninen 2007, 30) Suunnittelun kaikissa vaiheissa hyödynnettiin sitä, että pelin tekoon osallistuvat ensihoidon asiantuntijat edustivat myös pelin kohderyhmää. Näin päästiin tarkastelemaan paremmin pelaajan näkökulmaa suunnitteluun nähden.

4.3 Peli-idea

Tässä luvussa kuvataan alkuperäinen peli-idea, jota opinnäytetyössä lähdettiin toteuttamaan. Testauksen ja arvioinnin myötä idea eli prosessin aikana ja varsinainen toteutunut peli poikkeaa tarkoituksella siitä jonkin verran. Toteutunut peli kuvataan luvussa 5.

Pelissä pelaaja ohjaa ensihoitotehtävällä olevaa ensihoitajaa. Pelihahmo ja -maailma olisivat 2-ulotteisia. Peli on rauhallinen, kevyt ongelmanratkonta-salapoliiisipeli. Pelaaja joutuu yksittäisiin skenaarioihin. Skenaariot ovat hälytystehtäviä ja pelaaja saa tilanteesta jonkinlaiset alkutiedot. Pelaajan on hahmoa ohjaamalla tutkittava ympäristöä ja tehtävä havaintoja, joiden perusteella lopuksi päätellä keille hän on ilmoitusvelvollinen. Pelaaja voi liikkua, katsoa, koskea tai puhua pelin muille esineille tai hahmoille. Pelaaja voi muuttaa kursorinsa joko silmäksi, puhekuplaksi tai kädeksi. Tämä kertoo pelaajalle, mitä tutkimustapaa hän on käyttämässä. Epäoleellisiakin asioita tutkimalla pelaaja voi silti löytää jotain hauskaa tai kiinnostavaa. Kun pelaaja kokee olevansa valmis, hän joutuu antamaan arvionsa potilaan tilanteessa. Lopuksi pelaajan annetaan tietää, miten hyvin hän selvisi.

Skenaarioita pyritään saada tehtyä ainakin kolme. Lisäksi peli resurssien riittävyyden salliessa peli voi sisältää myös aiheeseen liittyvän tietovisan. Suunnittelusta pelinäköymästä tehtiin havainnekuva, jota käytettiin myös pelin ensimmäisessä testiversiossa (kuva 3). Havainnekuva on pelin ensimmäisestä skenaariosta. Skenaariossa potilaana on vanhempi naishenkilö, joka asuu yksin sotkuisessa asunnossa.



Kuva 3: Pelisuunnittelussa käytetty havainnekuva peli-ideasta. Pelaajan hahmoa kuvaa sininen tikku-ukko. Oikean alareunan sormea, silmää ja puhekuplaa klikkaamalla kursori muuttuu vastaavaksi kuvaksi ja pelaa pystyy ns. "katsomaan" tai "koskemaan" esineitä tai "puhumaan" ihmisille. Sinisiä nuolia klikkaamalla siirytään toiseen huoneeseen. Puhuttaessa alareunaan ilmestyisi vaihtoehtoja, joista pelaaja voi valita sanomisensa.

4.4 Julkaisualustan valinta

Yksi ensimmäisistä päätöksistä peliä suunniteltaessa oli, mille julkaisualustalle peli luodaan. Asetettuihin tavoitteisiin katsottiin parhaiten vastaavan joko mobiililla tai selaimella toimiva sovellus.

Mobiilisovellusta puoltaa erityisesti valmiin sovelluksen helppokäyttöisyys. Puhelimelle ladattava sovellus kulkee käyttäjän mukana eikä sen käynnistämiseksi tarvitse kuin klikata ikonia puhelimen näytöllä. Lisäksi 96 % suomalaisista 16–74-vuotiaista omistaa älypuhelimien. Osuus on opiskelijoiden keskuudessa todennäköisesti vielä suurempi. Kannettava tietokone on käytettävissä vain noin 80 prosentilla omassa kotitaloudessaan ja pöytätietokone vain noin joka toisella. (DNA 2020.)

Myös selainsovelluksen kehittämiseksi löydettiin perusteluja. Ensimmäkin verkkoselaimelle kehittäminen ja testaaminen on monin tavoin yksinkertaisempaa. Mobiilille kehitettäessä on otettava huomioon laitteiden eri käyttöjärjestelmät, erilaiset sovellusten jakamiseen käytettävät kanavat sekä sovelluksen skaalaminen hyvin erikokoisille näytöille. Kunnollinen testaaminen vaatii myös sitä, että käytettävissä on useita, erilaisia laitteita. (Väyrynen 2020.) Toiseksi Kaakkois-Suomen ammattikorkeakoululla on käytössä niin sanottu BYOD-toi-

mintamalli (Bring your own device), eli opiskelussa käytetään omia tietokoneita. Käytännössä kaikilla opiskelijoilla pitäisi siis olla käytettävissään vähintään oppilaitoksen suositteleman laitteen tasoinen oma tietokone. (BYOD – BRING YOUR OWN DEVICE s.a.) Edellä mainitut asiat ottaen huomioon, päädyttiin toteuttamaan sovellus selaimelle.

4.5 Työkalujen valinta

Pelin vaatimusten selvittämisen sekä alustan valinnan jälkeen valittiin työkalut, jotka parhaiten tukisivat asetettuja tavoitteita. Nämä työkalut olivat Unity-pelimoottori sekä GitLab-versionhallintapalvelu.

4.5.1 Versionhallinta

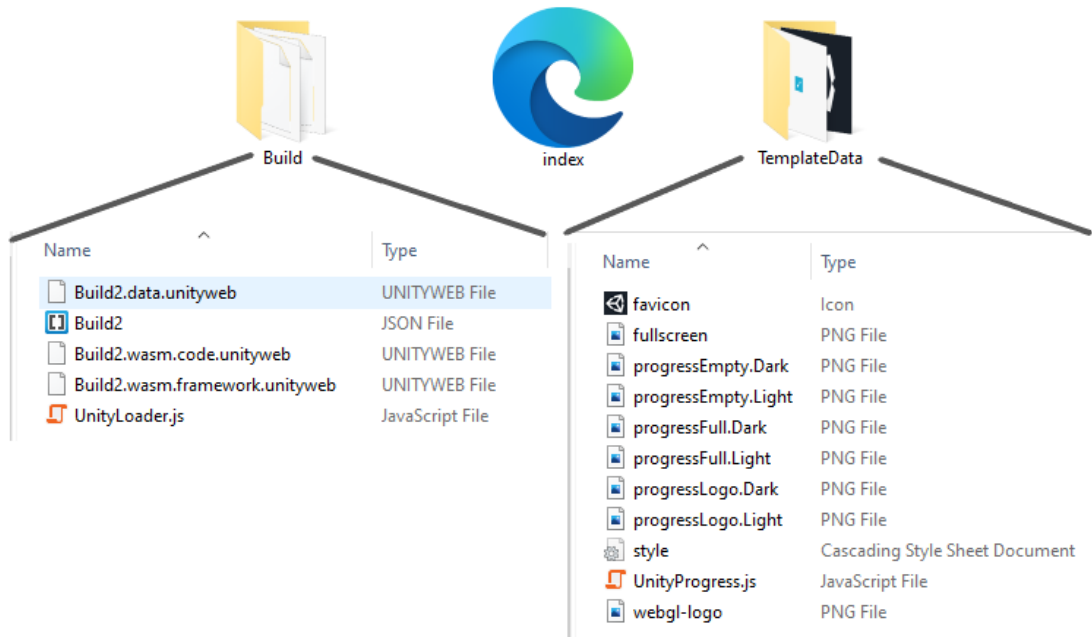
Opinnäytetyön versionhallintaan valittiin GitLab-verkkopalvelu. GitLab on Git-versionhallintaa hyödyntävä kehitysalusta. Muita suosittuja samankaltaisia palveluita ovat esimerkiksi Github sekä Bitbucket. Edellä mainituista vaihtoehdoista GitLab-palveluun päädyttiin, koska toisin kuin esimerkiksi Github, GitLab-palvelu sallii ohjelmistokehityksen suljetulla lähdekoodilla ilman lisämaksua (eli omaa lähdekoodia ei tarvitse pitää julkisena) sekä sen helppokäyttöisyyden vuoksi. (GitLab s.a.)

Versionhallinta on erityisen tärkeää, kun projektin parissa työskentelee useampia ihmisiä, mutta myös yksin työskennellessä versionhallinta on hyödyllinen työkalu. Versionhallinnalla saavutettiin tässä projektissa kaksi merkittävää etua. Ensimmäiseksi sen käyttö mahdollisti vaivattoman palaamisen edelliseen tai vielä aikaisempaan versioon projektissa, silloin kun nykyiseen versioon oli tehty ei toivottuja muutoksia. Tällaisia muutoksia saattoi joskus tulla tehtyä huomaamattaan tai sitten niin, että päätetäänkin toteuttaa jokin ominaisuus, eri tavalla kuin aluksi suunniteltiin. Kun versionhallinta toteutetaan käyttäen GitLab- tai sitä vastaavaa palvelua, saadaan myös samalla toteutettua projektin säännöllinen varmuuskopiointi. Säännöllisellä versionhallinnan käytöllä varmistetaan, että projektista on aina olemassa tuore varmuuskopio GitLab-palvelun palvelimella siltä varalta, jos työkoneella oleva versio tai itse työkone, tuhoutuu tai vaurioituu. (Maisala 2019.)

4.5.2 Unity WebGL

WebGL-sovellusten tekeminen Unity-pelimootorilla on ollut mahdollista maaliskuusta 2015, siitä lähtien kun Unity 5.0 julkaistiin (Echterhoff 2015). Opinäytetyössä tehty peli luotiin Unity-pelimootorilla ja siitä tehtiin selaimella toimiva WebGL-verkkosovellus käyttäen pelimootorin työkaluja. Peli ohjelmoitiin C#-ohjelmointikielellä. Unity-pelimootorin työkaluilla pelistä tehtiin selaimella toimiva, muun muassa HTML5-, JavaScript- ja WebAssembly-verkkostandardeja hyödyntävä, JavaScript ohjelman. Prosessin jälkeen ohjelma voidaan laittaa verkkosivulle, jonka kautta sitä voidaan käyttää WebGL-tekniologiaa tukevalla selaimella. (Unity Technologies 2021b.)

Valmis sovellus sisältää index-tiedoston, template-kansion sekä build-kansion. Template-kansio sisältää pelin latausruutuun liittyvät kuvat sekä tyylitiedoston. Lisäksi se sisältää JavaScript-tiedoston, jolla ohjataan esimerkiksi latauksen edistymistä osoittavaa latausviivaa. Jos latausruutua halutaan muuttaa, voidaan luoda oma WebGL-pohja, jonka voi ottaa käyttöön Unity-ohjelman projektiasetuksista. (Unity Technologies 2021b.) Build-kansio sisältää varsinaisen pelin sekä sen suorittamiseen tarvittavia JavaScript- ja WebAssembly-tiedostoja, kirjastoja sekä lisäosia. Lisäksi kansio sisältää json-tiedoston, josta ilmenee oleellisia tietoja kuten pelin nimi, Unity-versio, jolla se on luotu sekä eri tiedostojen url-osoitteita. Myös tämän tiedoston ohjelmoijan kannalta oleellisia asioita kannattaa muokata Unity-ohjelman projektiasetuksista (Unity Technologies 2021a). Sovelluksen kansiorakenne on kuvattu kuvassa 4.



Kuva 4: Unity-pelimoottorin luoman WebGL-sovelluksen kansiorakenne. Projektiasetukset sekä Unity-pelimoottorin versio vaikuttavat jonkin verran Build- ja TemplateData-kansioiden sisältöihin.

Index-tiedosto on html-tiedosto, jonka ajamalla peli käynnistyy. Unity-pelimoottorin tekemän WebGL-projektin index-tiedosto voi näyttää esimerkiksi kuvan 5 kaltaiselta.

```

1 |<!DOCTYPE html>
2 |<html lang="en-us">
3 |  <head>
4 |    <meta charset="utf-8">
5 |    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
6 |    <title>Unity WebGL Player | ensiapupeli</title>
7 |    <link rel="shortcut icon" href="TemplateData/favicon.ico">
8 |    <link rel="stylesheet" href="TemplateData/style.css">
9 |    <script src="TemplateData/UnityProgress.js"></script>
10 |   <script src="Build/UnityLoader.js"></script>
11 |   <script>
12 |     var unityInstance = UnityLoader.instantiate("unityContainer",
13 |       "Build/Buildit.json", {onProgress: UnityProgress});
14 |   </script>
15 | </head>
16 | <body>
17 |   <div class="webgl-content">
18 |     <div id="unityContainer" style="width: 960px; height: 640px"></div>
19 |   </div>
20 | </body>
21 | </html>

```

Kuva 5: Unity-pelimoottorin luoma pelin index-tiedosto. Index-tiedosto on lyhyt html-tiedosto ja se luodaan automaattisesti projektiasetusten perusteella.

Index-tiedostosta nähdään, että valmis WebGL-sovellus on yksin muuten tyhjällä verkkosivulla. Tiedostoa muokkaamalla verkkosivulle voidaan lisätä muita elementtejä.

Unity-pelimoottorin käyttämistä pelin toteuttamiseen tuki useampi asia. Merkittävin näistä on se, että Unity-ohjelmisto ja sen käyttämä C#-ohjelmointikieli ovat kirjoittajalle tuttuja. Pelinkehityksessä jää siten enemmän aikaa suunnitteluun ja toteuttamiseen. Aikaisempi kokemus myös oletettavasti vähentäisi ennalta-arvaamattomien ongelmien määrää, täten helpottaen aikataulussa pysymistä. Tämä mahdollisti hieman kunnianhimoisemman pelin suunnittelemisen. Lisäksi Unityllä on kattava dokumentaatio sekä aktiivinen käyttäjäkunta, jotka molemmat helpottavat erilaisista ongelmatilanteista selviytymistä. Pelin tekoon käytettiin Unity-versiota 2019.4.17f1, joka on julkaistu 22.12.2020.

5 TOTEUTUS

Opinnäytetyössä toteutettiin tavoitteiden mukaisesti opetuspelejä ensihoitajaopiskelijoiden itseopiskelumateriaaliksi. Tässä luvussa kuvataan toteutunut peli. Luvussa kuvataan pelin yleiskuvaus, pelimekaniikka, pelikokemus sekä pelaajan tavoitteet. Sen jälkeen kuvataan hieman, miten peli on toteutettu, sekä esitellään peliin sisäänrakennettu tietovisa. Lopuksi kerrotaan, miten saavutettavuutta on huomioitu valmiissa pelissä.

Pelissä pelaaja ohjaa ensihoitajaa ensihoitotehtävällä. Pelaajan tarkoituksena on tehdä havainnot potilaan elämäntilanteesta ja havaita erilaisia sosiaalisia hätätilanteita kuten esimerkiksi potilaan kyvyttömyys huolehtia itsestään, taikka vaaralliset asuinolot. Pelissä on kolme erilaista skenaariota, joihin pelaaja joutuu. Jokaisen skenaarion lopussa pelaajalta kysytään minkälaista apua potilas pelaajan mielestä tarvitsisi.

5.1 Pelimekaniikka ja pelikokemus

Pelimekaniikka on joukko sääntöjä, joista pelin pelattavuus muodostuu. Se koostuu pelin kontroleista eli ohjauksesta, sen sisäisistä syy- ja seuraussuhteista sekä pelin pelaajalle antamasta palautteesta. Pelimekaniikka käsittää pelin säännöt, jotka pyritään opettamaan pelaajalle, sekä pelimaailman sään-

töjä, joita ei aina ole edes tarkoitus kertoa pelaajalle. Pelimekaniikka määrittelee, miten peliä voidaan pelata ja kuinka haasteellinen peli on. (Kuorikoski 2018, 240.)

Pelin ohjaimena toimii hiiri. Pelaaja ohjaa hahmoaan klikkaamalla hiirellään. Ruudun oikeassa alareunassa on symboleita, joita klikkaamalla pelaaja pystyy muuttamaan kursorinsa kädeksi, suurennuslasiksi tai puhekuplaksi. Erilaisia kursoreita käyttämällä pelaaja pystyy koskemaan, katsomaan ja puhumaan pelissä oleville esineille ja ihmisille. Puhekuplalla toista hahmoa klikkaamalla aukeaa keskusteluikkuna, josta pelaaja voi valita haluamansa vaihtoehdon 1–4 keskusteluvaihtoehdosta kerrallaan. Tietty keskusteluvaihtoehdon valitseminen tai tiettyjen esineiden tutkiminen avaa uusia vaihtoehtoja pelaajalle. Kuvasessa 6 on esitetty, miten pelihahmo toimii esimerkkitilanteessa, eri hiiren kursorivaihtoehtoja käytettäessä.



Jos pelaaja on huomannut häkämittarin hälyttävän, niin pelaaja auttaa Marjatan ulos



Pelihahmo sanoo, että näkee Marjatan kädessä ison mustelman



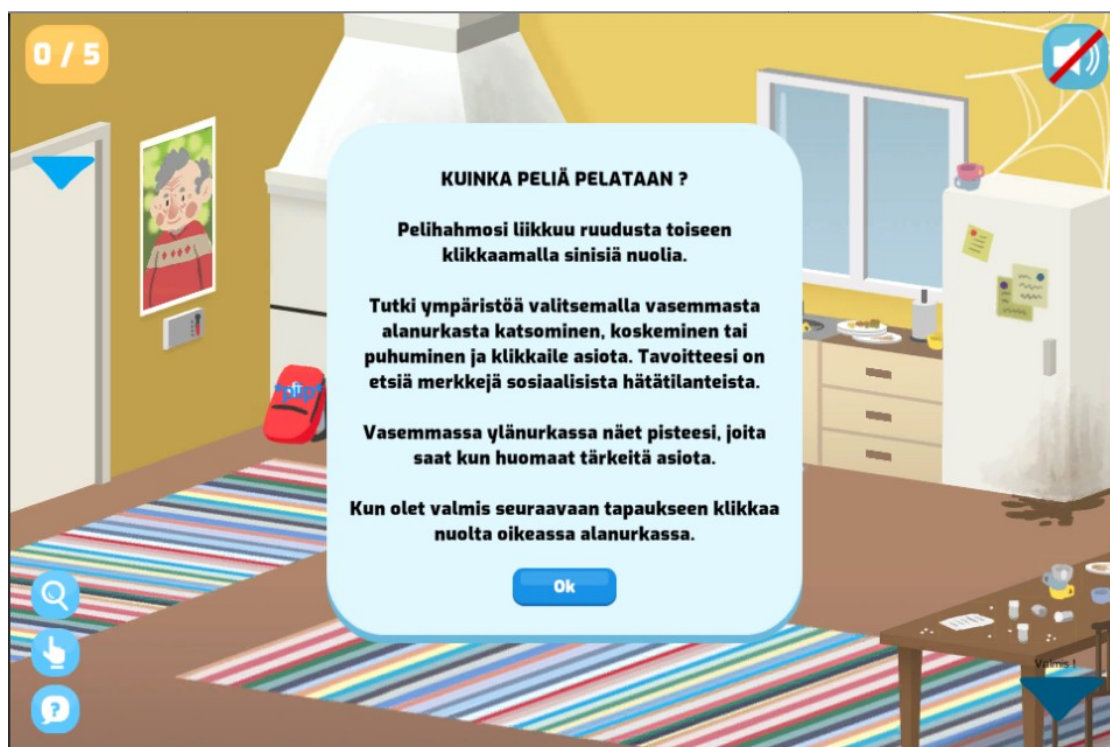
Keskustelu ikkuna aukeaa ja pelaaja voi valita mitä haluaa sanoa tai kysyä Marjatalta

Kuva 6: Pelihahmon toiminta eri kursorivaihtoehdoilla Marjatta-nimistä potilasta klikattaessa. Sormi-kursorilla klikattaessa pelihahmo auttaa Marjatan ulos häkäisestä talosta. Suurennuslasia käyttämällä pelihahmo katsoo tarkemmin potilasta ja huomaa, että potilaalla on suuri mustelma kädessä. Puhekupla-kursorilla klikkaamalla aukea keskusteluikkuna.

Peli on tarkoituksella vaikeusasteeltaan helppo, jottei peli vaikeutensa takia estäisi ketään pelaamasta. Peli on pyritty pitämään ulkoasultaan siistinä sekä yksinkertaisena, jotta se olisi myös vähemmän pelaaville helposti lähestyttävä. Peliin on lisätty jonkin verran huumoria ja yritetty näin tehdä myös epäoleellisten asioiden tutkimisesta mielenkiintoista. Pelaajan hahmo esimerkiksi vastustaa voimakkaasti kehotusta koskettaa yhdessä tapauksessa olevaa suurta hämähäkinseittiä. Tutkimisen ja löytämisen riemu yhdistettynä pelin lyhyeen keston toivottavasti estää kokeneempia pelaajia tylsistymästä peliin.

5.2 Tavoitteet

Pelaajan tavoitteena on löytää vihjeitä, joista voisi olettaa, että potilashahmoilla eivät ole kaikki asiat kunnossa. Toisinaan tutkimalla tilannetta pelaajalle aukeaa uusia etenemissuuntia. Esimerkiksi ensimmäisessä skenaariossa lämpömittaria katsomalla pelaaja voi huomata, että huoneessa on kylmä. Tämän jälkeen pelaajalle aukeaa uusi keskustelu vaihtoehto potilaan kanssa, jossa pelaaja voi kysyä potilaalta paleleeko potilas asunnossaan. Pelin tavoitteet ja kontrollit kerrotaan pelaajalle pelin alussa tutoriaali-ikkunassa. (Kuva 7.)



Kuva 7: Pelin tutoriaali-ikkuna. Tutoriaali-ikkuna näytetään jokaisen tehtävän alussa. Tutoriaali-ikkunassa kerrotaan pelaajalle pelin kontrollit sekä pelaajan tavoite.

Pelaaja saa pisteitä löytäessään vihjeitä. Pelinäköymässä näkyy pelaajan keräämät pisteet sekä maksimi pisteet, jotta pelaaja tietää, milloin hän etenee pelissä ja milloin hän on löytänyt kaikki oleelliset vihjeet. Pelin lopussa pelaaja valitsee vihjeiden perusteella kenelle tai keille viranomaisille hänen tulisi ilmoittaa potilaan tilanteesta.

5.3 Skenaariot

Pelissä on kolme erilaista skenaariota eli tapausta joihin pelaaja päätyy. Tapaukset pelataan ennalta määrättyssä järjestyksessä. Tapaukset ovat itsenäisiä kokonaisuuksia, eli pelaajan teoilla yhdessä tapauksessa ei ole vaikutusta muihin tapauksiin.

Skenaariot pyrittiin toteuttamaan teknisesti sellaisella tavalla, että uuden tapauksen lisääminen peliin pelinkehitysvaiheessa, tuottaisi mahdollisimman vähän lisätyötä. Toteutuneessa ratkaisussa uuden tapauksen lisääminen tehtiin niin, että vanha tapaus kopioitiin pelimoottorissa ja muokattiin uudeksi. Uuden tapauksen lisääminen ei täten vaikuta lainkaan vanhoihin tapauksiin, myös kaikki uusi koodi kirjoitettiin irrallaan valmiista tapauksista. Tapaukset olivat toteutukseltaan hyvin samankaltaisia. Ohjelmoinnin näkökulmasta, merkittävimmät erot tapausten välillä olivat pelihahmon reaktiot eri esineisiin sekä pelihahmon ja potilaan väliset keskustelut.

Uutta tapausta luodessa, sen vaatima uusi koodi, kirjoitettiin yhteen uuteen luokkaan. Luokka on olio-ohjelmoinnissa olion kuvaus. Uudet luokat perivät Story-nimisen luokan eli niiden tuli sisältää siinä kuvatut funktiot (Kuva 8).

```

5 public abstract class Story
6 {
7     // setup story
8     public abstract void StartStory();
9
10    // input what was clicked using which mouse cursor, return comment for player
11    public abstract string ClickedWith(string _object, string _mouseMode);
12
13    // return conversation lines to gameManager
14    public abstract string[] GetConversationLines(int identity, string line);
15
16    // return max score for current story
17    public abstract int GetScoreMax();
18
19    // return end screen comment for answer, input wheather answer was right
20    public abstract string GetEndScreenComment(bool wasRight);
21
22    // returns next story
23    public abstract Story NextStory();
24
25    //Returns corresponding sceneName
26    public abstract string GetSceneName();
27
28
29 }
```

Kuva 8: Story-nimisen luokan määrittely. Luokka on abstrakti eli siitä ei voi tehdä oliota mutta toinen luokka voi periä sen rakenteen.

Jokaista skenaariota kohden toteutettiin yksi uusi Story-luokan perivä luokka. Menetelmän hienous on se, että jokaisesta Story-luokan perivästä uudesta luokasta löytyy Story-luokassa kuvatut funktiot, mutta funktioiden ei tarvitse olla toteutukseltaan samanlaisia keskenään. Täten muualla koodissa voidaan näitä luokkia kohdella keskenään vaihtokelpoisina.

5.4 Tietovisa

Tarinavetoisen pelin lisäksi pelin päävalikosta pääsee pelaamaan tietovisaa. Tietovisassa pelaaja voi kerrata osaamistaan sekä tutustua aiheisiin, jotka jäivät tarinapohjaisessa pelissä pienemmälle huomiolle. Tietovisassa on monivalintakysymyksiä, joihin pelaaja valitsee oikean tai oikeat vastaukset. Peli laskee pelaajan saamat pisteet ja antaa pelaajalle myös vastausta selventävää kirjallista palautetta pelaajalle. Kuvassa 9 on kuvakaappaus tietovisasta.



Kuva 9: Kuvakaappaus tietovisasta. Kuvassa pelaaja on valinnut kaksi alimmaista vastausta, jota ilmaistaan pienillä nuolilla vastausvaihtoehtojen reunoilla ja lukinnut vastauksensa. Peli on tarkastanut vastaukset ja värjännyt oikeat vastaukset vihreällä ja väärät punaisella.

Tietovisan ulkoasua suunniteltaessa panostettiin erityisesti ulkoasun selkeyteen. Esimerkiksi väriin vastauksiin lisättiin pelinkehityksen aikana x-tekstuuri, jotta vastaukset erottuvat värin lisäksi myös muodoltaan.

5.5 Saavutettavuus

Pelin saavutettavuutta huomioitiin pelissä erilaisin keinoin. Aikaisemmin kuvattiin jo, miten kokemattomat pelaajat on huomioitu toteutuksessa. Pelin saavutettavuutta arvioitiin ja suunniteltiin myös muista näkökulmista. Tässä apuna käytettiin WCAG-standardin version 2.1 virallista suomennosta (ks. W3C 2018).

Pelissä hyödynnettiin WCAG-standardin vaatimuksia tekstien- ja taustojen värien kontrastisuhteelle (ks. W3C 2018, § 1.4.3). Lisäksi pelissä ei esitetty mitään informaatiota pelkän värin muodossa vaan käytettiin sen lisäksi myös tekstiä tai muotoa. Pelin taustamusiikki on oletuksella mykistettynä koska peli käynnistyy selaimessa automaattisesti. Standardissa olisi riittänyt se, että pelillä on mykistysnappi (ks. W3C 2018, § 1.4.2). Tämä koettiin kuitenkin vielä käyttäjäystävällisemmäksi vaihtoehdoksi. Kaikkia saavutettavuusvaatimuksia ei kuitenkaan pystytty toteuttamaan.

6 TESTAUS

Kaikki ohjelmointi edellyttää testausta, jotta valmiin tuotteen laatua voidaan hallita. Tässä luvussa käydään läpi, miten peliä on testattu ja millaisia toimintavirheitä testauksissa havaittiin.

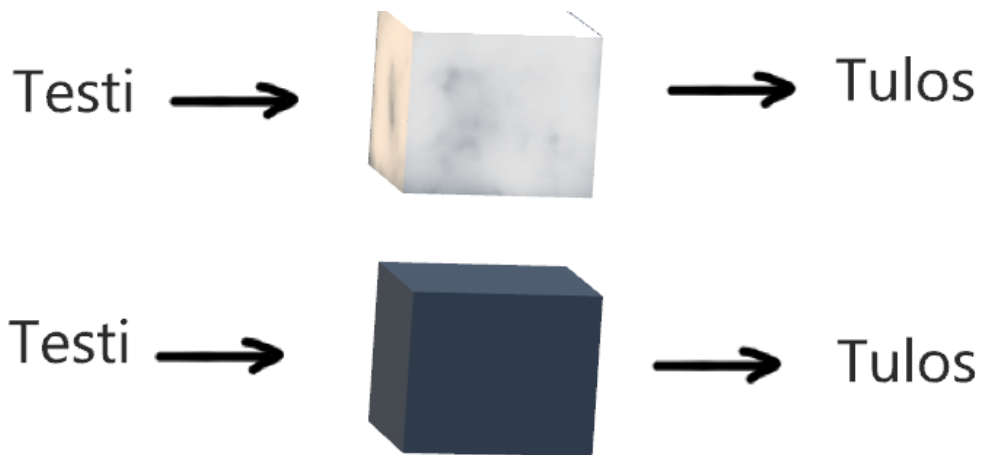
6.1 Testausmenetelmät

Peliä testattiin pääasiassa kolmella erilaisella tavalla. Nopein ja yksinkertaisin tapa oli testata peliä Unity-pelimoottorin sisällä Play-moodissa. Play-moodi on Unity-pelimoottorin sisäänrakennettu testitila. Play-moodi soveltui hyvin koodin kirjoitusvirheiden taikka ajatusvirheiden takia syntyneiden toimintavirheiden löytämiseen ja perustoimintojen testaamiseen. Tällaiset projektissa kohdatut toimintavirheet olivat jokseenkin turhan triviaaleja, jotta niistä kertominen tässä hyödyttäisi lukijaa. Tätä testaustapaa käytettiin jokaisen uuden ominaisuuden kehittämisen ja lisäämisen yhteydessä.

Toinen testaamistapa, jolla peliä testattiin, vaati pelin viemisen pois Unity-ympäristöstä ja sen muuttamisen WebGL-sovellukseksi. Tämän jälkeen sovellus

ladattiin verkkosivulle ja sitä päästiin testaamaan menemällä selaimella verkkosivun osoitteeseen. Prosessi vie huomattavasti enemmän aikaa mutta peliä päästään testaamaan, sillä alustalla, jolla sitä lopulta tullaan myös pelaamaan. Verkkosivulla testattaessa löytyi toimintavirheitä, jotka liittyivät erityisesti verkkojulkaisuun.

Kaksi ensimmäistä testaustapaa olivat niin sanottua white box -testausta. Niimellä tarkoitetaan sitä, että testaaja tietää kuinka testauksen kohde on rakennettu (Kuva 10). Testatessa voidaan tutkia yksittäisiä ominaisuuksia yksityiskohtaisesti, tarkkailemalla toimivatko koodissa olevat raja- ja poikkeustapaukset toivotusti. Se soveltuu erityisesti uusien ominaisuuksien testaamiseen sekä toimintavirheiden löytämiseen. (Testbytes 2019b)



Kuva 10: White box -testaus ja black box -testaus. White box -testauksen nimellä korostetaan sitä, että testaaja tietää, miten testauskohde toimii. Black box -testauksessa testaaja ei "näe" laatikon sisään, eli testaaja ei tunne ohjelman koodia.

Kolmas pelin testauksessa käytetty testaustapa oli black box -testausta. Black box -testauksessa testaajat eivät tunne ohjelman koodia eikä heidän tarvitse osata ohjelmoida. Tämä testaustapa soveltuu white box -testausta paremmin pelin käytettävyyden ja kokonaisuuden testaamiseen. (Testbytes 2019a)

Kolmannessa testaustavassa hyödynnettiin sitä, että projektissa oli mukana kaksi ensihoitajaopiskelijaa, jotka vastasivat pelin asiasisällöstä. He edustivat samalla myös varsin hyvin pelin kohderyhmää, vaikka pelin aihe olikin heille jo luonnollisestikin tuttu. Pyytämällä heitä kokeilemaan verkkosivulle ladattuja

pelin prototyyppejä saatiin hyvin hyödyllistä palautetta liittyen sekä pelin asiasältöön, että pelin pelattavuuteen. Peliä ei valitettavasti ehditty testata kokonaan projektin ulkopuolisilla opiskelijoilla aikataulujen takia.

6.2 Toimintavirheet ja poikkeamat

Testauksen yhteydessä havaittiin erilaisia toimintavirheitä. Suurin osa virheistä oli pieniä ja niiden korjaaminen oli yksinkertaista. Seuraavaksi käsitellään virheistä muutamaa, jotka olivat muita merkittävämpiä. Merkittäviä toimintavirheitä olivat kursorin muuttamisen toimimattomuus sekä WebGL-sovelluksen aiheuttama näppäinansa.

Pelissä hiiren kursoria pystyy muuttamaan silmäksi, kädeksi ja puhekuplaksi. Muuttuva kursori on pelimekaniikan kannalta oleellista. Ominaisuutta testattaessa WebGL-sovelluksessa huomattiin, että kursori ei muutu, vaan se pysyi tavallisen nuolena. Pelimoottorin valmis metodi ei toiminut WebGL-versiossa. Selaimella toimivan WebGL-version kursoria muuttamiseen oltaisiin voitu käyttää JavaScript-liitännästä. Sellaisen tekeminen ja testaaminen olisi ollut kuitenkin tarpeettoman työlästä. Ongelma päädyttiin lopulta kiertämään. Kun kursoria muutetaan pelissä, todellisuudessa kursori piilotetaan ja näkymätöntä kursoria seuraamaan asetetaan kuva, joksi kursori oli tarkoitus muuttaa.

Pelin saavutettavuutta arvioidessa huomattiin merkittävä poikkeama. Unity-pelimoottorilla tehty WebGL-sovellus aiheutti niin sanotun näppäimistöansan. Näppäimistöansalla tarkoitetaan tässä sitä, että käyttäjä pystyy siirtämään kohdistuksen sovellukseen pelkkää näppäimistöä käyttämällä, mutta ei siirtämään kohdistusta pois käyttäen vain näppäimistöä. Ongelma on merkittävä, koska myös ei-saavutettavilta teknologioilta edellytetään häiriöttömyyttä (ks. W3C 2018, 5.2.5). Näppäimistöansa johtui siitä, että oletusasetuksena kohdistettu sovellus kaappasi kaikki näppäinpainallukset. Ongelma oli tässä tapauksessa helppo korjata. Koska pelissä käytettiin vain hiiren klikkauksia käyttäjän syötteenä, voitiin oletus poistaa käytöstä. Muutoksen jälkeen näppäimistöansa poistui ja käyttäjä pystyi taas navigoimaan verkkosivulla käyttäen esimerkiksi nuolinäppäimiä ja tabulaattoria.

7 TULOKSET

Opinnäytetyössä ensisijaisena tuloksena valmistui peli, jota opiskelija pystyy käyttämään itseopiskelumateriaalina. Pelissä pelaaja kohtaa ensihoidon asiantuntijoiden suunnittelemaa tilanteita, joissa ilmenee sosiaalisia hätätilanteita. Hätätilanteet ovat sellaisia, joita pelaaja saattaa kohdata tulevaisuudessa työelämässä. Pelin avulla opiskelija voi rauhassa, ilman kiirettä, tutustua pelin tilanteisiin. Peli kannustaa opiskelijaa aktiivisesti tutustumaan potilaan tilanteeseen hälytystehtävän yhteydessä ja opettaa häntä lain määrittämästä ilmoitusvelvollisuudesta. Lisäksi pelissä on tietovisaosio, jota pelaaja voi käyttää tietojensa testaamiseen sekä tutustua aiheen niihin puoliin, joita pelin kolmessa skenaarioissa ei käsitelty. Näiltä osin tulokset vastaavat siis erinomaisesti tavoitteisiin.

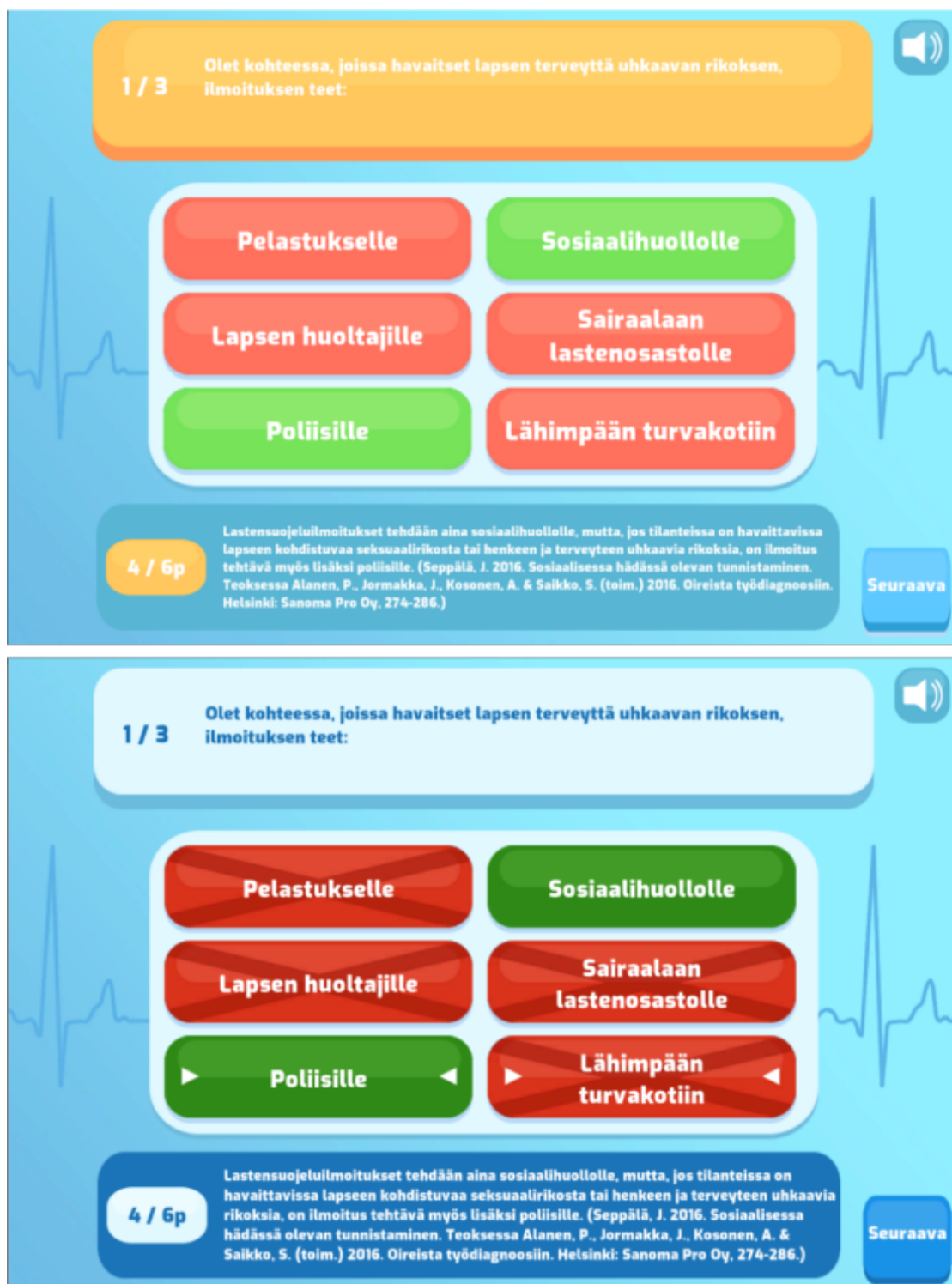
Peli kehitettiin yhteistyössä kahden ensihoitajaopiskelijan, Hilliahon ja Sormusen kanssa. Heidän osallistumisensa pelinkehitykseen asiantuntijaroolissa, koettiin lähes välttämättömäksi pelin onnistumisen kannalta. Tämä johtui etenkin siitä, että opetettavat asiat olivat korkeakoulutasoisia. Opetuspelin aiheen hallitsemisen välttämättömyys ei luonnollisestikaan ollut kovin yllättävää, taikka tuloksena merkittävä. Asiantuntijoiden lisäksi ensihoitajaopiskelijat edustivat kuitenkin myös pelin kohderyhmää. Pelin kohderyhmä otettiin siis mukaan pelinkehitykseen aivan alusta asti. Lisäksi tässä tapauksessa, he edustivat myös opiskelijoita, joilla ei juurikaan ole kokemusta pelaamisesta. Pelinkehityksen aikana koettiin erittäin hyödylliseksi palaute, jota Hilliaholta ja Sormuselta saatiin ei-pelaavina, kohderyhmän edustajina. Palauteen ansiosta pelin aloittelijaystävällisyyttä haittaavia ominaisuuksia löydettiin ja korjattiin.

Pelin suunnittelussa käytettiin suunnittelu-kehitys-testaussykliä, josta johtuen suunnitelmat muuttuivat sen myötä, kun niitä testattiin käytännössä. Suunnitelmia myös dokumentoitiin varsin kevyesti koska pelin tekoon ei osallistunut muita ohjelmoijia. Tämä ei kuitenkaan tuottanut ongelmia ennen kuin tiimin koko kasvoi neljään Noora Jantusen liittyessä graafikon rooliin. Neljän hengen tiimin ohjaaminen vaati selvästi enemmän aikaa, jotta jokainen tiesi mitä heiltä odotettiin, eikä kukaan tekisi turhaa työtä. Tässä onnistuttiin pitämällä enemmän yhteisiä palavereja ja varaamalla hieman enemmän aikaa projektin johtamiseen.

Pelissä on rauhallinen tunnelma, eikä pelaajalta vaadita nopeutta taikka erityistä taitoa pelin läpäisemiseksi. Pelin ulkoasu on selkeä, ja siinä on yksinkertainen käyttöliittymä. Peli suunniteltiin toimivaksi sellaisella laitteistolla, jota oppilaitos edellytti muutenkin opiskelijoiden vähintään omistavan. Pelinkehityksessä huomioitiin se, että kaikilla pelaajilla ei välttämättä ole juurikaan kokemusta pelaamisesta. Peli sisälsi ohjeita pelin pelaamiseen. Edellä mainitut asiat kasvattivat pelin saavutettavuutta ja mahdollistavat sen, että useampi opiskelija pystyy hyötymään pelistä.

Peli on toteutettu WebGL-verkkosovelluksena, joten tilaajan on helppo jakaa peli opiskelijoiden käytettäväksi laittamalla peli verkkosivulle. WebGL-tekniikka on laajasti tuettu eikä sen tuen oleteta katoavan lähiaikoina. Peli on täten tekniseltä toteutukseltaan käyttökelpoinen myös tulevaisuudessa. Toteutuksessa huomioitiin myös tilaajan verkkosisällön saavutettavuutta säätelevää lainsäädäntöä. Peli täyttää lain edellytyksen häiriöttömyydestä (ks. W3C 2018, § 5.2.5) mutta se ei ole saavutettava WCAG 2.1 -standardin tarkoittamalla tavalla. Saavutettavuus olisi käytännössä edellyttänyt, että sokeakin pelaaja olisi pystynyt ainakin käyttämään peliä, esimerkiksi ruudunlukuohjelman avulla. Sellaisen pelin suunnittelu ja toteutus olisi todennäköisesti ollut kuitenkin liian suuri työ opinnäytetyöksi. Lisäksi saavutettavuusvaatimuksista ei oltu tietoisia, kun peliä alettiin suunnittelemaan eikä tilaajakaan osannut silloin niitä vaatia. Tilaajan pitäisi kuitenkin, lain tulkinnasta riippuen, pystyä hyödyntämään peliä, kun pelin tarjoamalle sisällölle tarjotaan saavutettava vaihtoehto.

Kuvassa 11 on verrattu kahta versiota pelin tietovisasta. Siinä nähdään, miten aikaisemmassa kehitysvaiheessa oleva versio tietovisasta poikkeaa lopullisesta versiosta, jossa saavutettavuus on huomioitu WCAG 2.1 -standardia hyödyntämällä.



Kuva 11: Tietovia pelin eri kehitysvaiheissa. Ylhäällä näkyy pelin aikaisempi versio ja alhaalla lopullinen versio. Lopullisessa versiossa saavutettavuutta on lisätty kiinnittämällä huomiota tekstien ja taustojen väliseen kontrastisuhteeseen sekä lisäämällä punaisen värin lisäksi x-muoto ilmaisemaan väärää vastausta. Lisäksi nuolilla ilmaistaan pelaajan valitsemia vastausvaihtoehtoja.

Tietovisan versioita vertailemalla nähdään miten saavutettavuuden lisäämisestä hyötyvät myös pelaajat, joilla ei ole aisteihin liittyviä rajoitteita. Tekstit

ovat samankokoisia molemmissa versioissa mutta lopullisessa versiossa ne erottuvat paremmin taustasta.

7.1 Johtopäätökset

Työssä toteutettiin työn tavoitteiden mukaisesti opetuspelejä. Tavoitteena oli myös selvittää, millainen peli sopii ensihoitajille työkaluksi itseopiskeluun, ja miten sellainen peli voidaan toteuttaa. Lisäksi tarkoituksena oli havaita ja raportoida pelinkehityksen aikana kohdattuja ongelmia. Kysymyksiin oli tarkoitus vastata hyödyntämällä pelinkehityksessä tehtyjä havaintoja.

Ensihoitajakoulutus on ammattikorkeakoulutason koulutus, joten opetuspeliksi soveltuvalta peliltä vaaditaan vankkaa asiantuntemusta. Sen suunnittelussa tulee ottaa huomioon, että kaikilla opiskelijoilla ei ole kokemusta pelaamisesta ja suunnitella peli kokemattomimpien ehdoilla. Pelin on oltava riittävän saavutettava sekä opiskelijan että oppilaitoksen näkökulmasta koska saavutettavuus sekä mahdollistaa pelaamisen että parantaa pelikokemusta. Oppilaitoksen näkökulmaan sisältyy myös lainsäädännöllinen puoli, joka on huomioitava. Opetuspelin hyödyntämiä teknologioita valitessa pitää huomioida, että teknologiat ovat käytettävissä nyt ja tulevaisuudessa.

Tämä työ toteutettiin yhteistyössä ensihoidonopiskelijoiden kanssa. He olivat sekä aiheen asiantuntijoita, että kohderyhmän ei-pelaavia edustajia. Näin peli toteutettiin asiantuntijoiden avulla, sitä suunniteltiin kohderyhmän kanssa ja sitä testattiin kohderyhmällä heti alusta asti. Yhteistyöstä ensihoidonopiskelijoiden kanssa olikin merkittävää hyötyä koko pelinkehityksen aikana. Opetuspelejä tulisi tästä syystä suunnitella ja testata yhdessä kohderyhmän kanssa.

Pelinkehityksen suurin haaste oli lain edellyttämien saavutettavuusvaatimusten täyttäminen, joihin ei pystytty täysin pelisuunnitelman tavoitteiden mukaisesti vastaamaan. Tästä syystä on tärkeää, että varsinkin saavutettavuusvaatimuksia tarkastellaan heti aluksi, kun lähdetään toteuttamaan tällaista palvelua julkisoikeudelliselle toimijalle.

7.2 Jatkokehitys

Toteutunut peli on verkkosovellus ja sitä käytetään verkkosivun kautta. Unity-pelimootorin tuottama sivu on tyhjä, lukuun ottamatta tehtyä peliä. Kehitystä kannattaakin mielestäni jatkaa kehittämällä tätä verkkosivua. Sivusta kannattaa tehdä ulkoasultaan ja ominaisuuksiltaan yhtenevän tilaajan verkkosivuston kanssa. Lisäksi lainsäädännön kannalta olisi tärkeää, että sivulla tarjottaisiin saavutettava vaihtoehto pelin informatiiviselle sisällölle.

Pelin kehittämistä voidaan myös edelleen jatkaa. Helppona jatkokehitys-ideana, peliin voidaan tehdä uusia kenttiä, sekä lisätä kysymyksiä tietovisaan. On kuitenkin huolehdittava, ettei pelistä tule niin suurta ja raskasta, että pelattavuus kärsii. Mahdollisuutta tehdä pelistä puhelimella toimiva mobiilisovellus, voisi myös olla mielenkiintoista tutkia. Pelinkehitystä voidaan jatkaa myös lisäämällä pelin saavutettavuutta. Peliin voisi esimerkiksi toteuttaa kontrollit näppäimistölle ja antaa pelaajan valita miten hän haluaa ohjata peliä. Saavutettavuutta lisäisi myös se, että pelaaja pystyisi vaikuttamaan pelin tekstien koon ja pelin käyttämiin väreihin.

8 POHDINTA

Tämän opinnäytetyön tuloksena valmistunut peli on monella tapaa onnistunut. Peli on toteutukseltaan selkeä ja yksinkertainen niin ulkoasultaan kuin sisäisesti koodipuoleltaan. Peli vastaa hyvin pelaajan tarpeisiin ja huomioi erilaiset pelaajat. Pelinkehityksessä pysyttiin hyvin aikataulussa ja peli toteutettiin suunnitellussa laajuudessaan. Odotuksena oli, että kehityksen aikana kohdattaisiin enemmän teknisiä haasteita. Unity-pelimootorilla tehty WebGL-sovellus oli kuitenkin teknisesti helpompi toteuttaa kuin ensin ajateltiin. Myöskin pelin rakenne osattiin valita niin, että ohjelmointityöstä selviydettiin sujuvasti.

Yhteistyö Hilliahon ja Sormusen kanssa koettiin hyödylliseksi ja mielenkiintoiseksi. Peliin tehtiin monia muutoksia heidän antamansa palautteen pohjalta ja peli olisi näyttänyt hyvin erilaiselta ilman heidän panostaan. Pelin kehittämisen yhteistyössä kohderyhmän kanssa antoi myös varmuutta projektin oikeasta suunnasta. Opetuspeliä suunniteltaessa voitaisiinkin huomioda enem-

män kohderyhmät kohderyhmien sisällä, esimerkiksi tässä tapauksessa, ensihoitajaopikelijoista ei-pelaavat opiskelijat. Näin päästään paremmin käsiksi olennaisimpaan palautteeseen.

Tilaa antoi pelin toteuttamiseen varsin vapaat kädet. Pelin suunnittelussa pyydettiin ensisijaisesti huomioimaan opiskelijan näkökulmaa. Tästä päädyttiin kuitenkin tilanteeseen, jossa huomataan kesken pelinkehityksen, että pelin peliä koskee lainsäädäntö, jota ei oltu suunnittelun alussa lainkaan tietoisia. Saavutettavuusvaatimusten asettamien haasteisiin pyrittiin vastaamaan täysin siinä kuitenkaan onnistumatta. Vaatimukset koettiin raskaiksi ja jopa häiriöksi, eihän esimerkiksi sokea henkilö voisi toimia ensihoitajana. Toisaalta taas saavutettavuuden lisääminen tuntui parantavan kaikkien pelikokemusta. Myöskin lainsäädäntöä ja WCAG-standardia oli vaikea tulkita, joten tässä vaiheessa ei ole täyttä varmuutta siitä, kuinka hyvin Kaakkois-Suomen ammattikorkeakoulu pystyy peliä hyödyntämään. Olisikin ollut kaikkien etu, että kaikki olisivat olleet heti tietoisia näistä vaatimuksista ja niiden noudattamista olisi suunniteltu yhdessä. Digipalvelulaki on toisaalta astunut voimaan vasta 2019, eli sen jälkeen, kun aloitin tietotekniikan opintoni, enkä ennen tätä opinnäytetyötä ollut tutustunut siihen. Se koskettaa viranomaisten lisäksi suurta joukkoa julkisia palveluita tuottavia tahoja. Voisikin olla perusteltua sisällyttää saavutettavuutta ja sitä koskevaa lainsäädäntöä opiskeluun.

Kaiken kaikkiaan opinnäytetyö oli kuitenkin mielenkiintoinen ja palkitseva projekti. Hyödyllisen pelin kehittäminen oli motivoivaa, ja pelinkehityksen aikana opittiin paljon uusia asioita.

LÄHTEET

Anyuru, A. 2012. Professional WebGL Programming: Developing 3D Graphics for the Web. Hoboken: John Wiley & Sons.

Beavis, C., Dezuanni, M. & O'Mara, J. 2017. Serious Play: Literacy, Learning and Digital Games. Milton: Taylor and Francis.

BYOD – BRING YOUR OWN DEVICE s.a. Kaakkois-Suomen ammattikorkeakoulu. WWW-dokumentti. Saatavissa: <https://www.xamk.fi/byod-bring-device/> [Viitattu 29.4.2021].

DNA. 2020. Digitaaliset elämäntavat. PDF-dokumentti. Saatavissa: <https://www.sttinfo.fi/data/attachments/00520/fc19227b-ef08-452e-9c33-0ed70294f355.pdf> [Viitattu 16.1.2021].

Echterhoff, J. 2015. WebGL Roadmap. WWW-dokumentti. Unity Forums. Saatavissa: <https://forum.unity.com/threads/webgl-roadmap.334408/> [Viitattu 27.4.2021].

ENSIHOITAJA (AMK) s.a. Kaakkois-Suomen ammattikorkeakoulu. WWW-dokumentti. Saatavissa: <https://www.xamk.fi/koulutukset/ensihoitaja-amk/> [Viitattu 9.6.2021].

Git. 2021. Complete list of all commands. WWW-dokumentti. Päivitetty 15.3.2021. Saatavissa: https://git-scm.com/docs/git#_git_commands [Viitattu 9.5.2021].

Git s.a. About. WWW-dokumentti. Saatavissa: <https://git-scm.com/about> [Viitattu 8.4.2021].

GitLab s.a. Why is this page called 'Is it any good?'. WWW-dokumentti. Saatavissa: <https://about.gitlab.com/is-it-any-good/#gitlab-is-the-most-used-devops-tool> [Viitattu 7.4.2021].

Huopalainen, N. 2020. Pelin suunnittelu ja toteutus Dreams-kehitysympäristössä. Kaakkois-Suomen ammattikorkeakoulu. Tieto- ja viestintäteknikan koulutusohjelma. Opinnäyte. PDF-dokumentti. Saatavissa: <http://urn.fi/URN:NBN:fi:amk-2020101221272> [Viitattu 27.4.2021]

Kaakkois-Suomen ammattikorkeakoulu. 2020. Xamkin avainluvut 2019-2020. PDF-tiedosto. Saatavissa: https://www.xamk.fi/wp-content/uploads/2020/09/xamkin-avainluvut-2019_2020_.pdf [Viitattu 19.4.2021].

Khronos Group. 2020. WebGL Working Group Updates, WebGL Meetup, November 2020. PDF-tiedosto. Saatavissa: <https://www.khronos.org/assets/uploads/developers/library/2020-virtual-webgl-meetup/01%20WebGL%20WG%20Updates.pdf> [Viitattu 27.4.2021].

Khronos Releases Final WebGL 1.0 Specification s.a. Khronos Group. WWW-dokumentti. Saatavissa: <https://www.khronos.org/news/press/khronos-releases-final-webgl-1.0-specification> [Viitattu 27.4.2021].

Kuorikoski, J. 2018. Pelitaiteen manifesti. Helsinki: Gaudeamus.

Laki digitaalisten palvelujen tarjoamisesta 15.03.2019/306.

Loliger, J. & McCullough, M. 2012. Version Control with Git, 2nd Edition. California: O'Reilly Media, Inc.

Maisala, S. 2019. Versionhallinta on tutkimuksen välttämätön työkalu – Gitlab tukee tutkijan työtä Helsingin yliopistossa. *Think Open Digest* 2 (1). Verkkolehti. Saatavissa: <https://blogs.helsinki.fi/thinkopen/versionhallinta-on-valttamaton-tyokalu-tutkimukselle/> [Viitattu 7.4.2021].

Manninen, T. 2007. Pelisuunnittelijan käsikirja, Ideasta eteenpäin. Oulu: Rajalla.

Mitä palveluja ja sisältöjä laki koskee? s.a. Etelä-Suomen aluehallintovirasto. WWW-dokumentti. Saatavissa: www.saavutettavuusvaatimukset.fi/digipalvelu-lain-vaatimukset/mita-palveluja-ja-sisaltoja-laki-koskee [Viitattu 11.5.2021].

Polvela, S. 2020. Mökin esittelysovelluksen toteuttaminen. Lapin AMK. Tieto- ja viestintätekniiikan koulutusohjelma. Opinnäyte. PDF-dokumentti. Saatavissa: <http://urn.fi/URN:NBN:fi:amk-2020052814916> [Viitattu 27.4.2021].

Solomon, F. 2018. EDUCATIONAL GAME FOR FORESTRY : Creating Unity 3D Game. Lapin AMK. Tieto- ja viestintätekniiikan koulutusohjelma. Opinnäyte. PDF-dokumentti. Saatavissa: <http://urn.fi/URN:NBN:fi:amk-201802152497> [Viitattu 27.4.2021].

Testbytes. 2019a. Black Box Testing Techniques with Examples. WWW-dokumentti. Saatavissa: <https://www.testbytes.net/blog/black-box-testing/> [Viitattu 6.5.2021].

Testbytes. 2019b. What is White Box Testing? Techniques, Examples and Types. WWW-dokumentti. Saatavissa: <https://www.testbytes.net/blog/white-box-testing/> [Viitattu 6.5.2021].

Unity Personal s.a. Unity Technologies. WWW-dokumentti Saatavilla: <https://store.unity.com/products/unity-personal> [Viitattu 3.5.2021].

Unity Technologies. 2021a. Building and running a WebGL project. WWW-dokumentti. Saatavilla: <https://docs.unity3d.com/Manual/webgl-building.html> [Viitattu 3.5.2021].

Unity Technologies. 2021b. Getting started with WebGL development. WWW-dokumentti. Saatavilla: <https://docs.unity3d.com/Manual/webgl-gettingstarted.html> [Viitattu 3.5.2021].

Unity Technologies. 2021c. Using WebGL templates. WWW-dokumentti. Saatavilla: <https://docs.unity3d.com/Manual/webgl-templates.html> [Viitattu 3.5.2021].

Väyrynen, H. 2020. Monialustaisen mobiilisovelluksen kehittäminen ja julkaisu. Kaakkois-Suomen ammattikorkeakoulu. Tieto- ja viestintätekniikan koulutusohjelma. Opinnäyte. PDF-dokumentti Saatavissa: <http://urn.fi/URN:NBN:fi:amk-2020060216121> [Viitattu 27.4.2021].

W3C. 2018. Verkkosisällön saavutettavuusohjeet (WCAG) 2.1. WWW-dokumentti. Saatavissa: www.w3.org/Translations/WCAG21-fi/ [Viitattu 12.5.2021].

Wallez, C. 2020. WebGPU: Next-generation 3D graphics on the web (DevFest 2019). Google Developer Groups. Youtube. Videoleike. Julkaistu 3.3.2020. Saatavissa: https://www.youtube.com/watch?v=EhWvqaRDz5s&t=830s&ab_channel=GoogleDeveloperGroups [Viitattu 18.4.2021].

WCAG 2.1: lain vaatimukset s.a. Etelä-Suomen aluehallintovirasto. WWW-dokumentti. Saatavissa: www.saavutettavuusvaatimukset.fi/digipalvelulain-vaatimukset/wcag-2-1/ [Viitattu 11.5.2021].

WebGL Overview s.a. Khronos Group. WWW-dokumentti. Saatavissa: <https://www.khronos.org/webgl/> [Viitattu 14.4.2021].

Welcome to Unity s.a. Unity Technologies. WWW-dokumentti. Saatavilla: <https://unity.com/our-company> [Viitattu 3.5.2021].