Opinnäytetyö (AMK)

Tieto- ja viestintätekniikka

2020

Aku Lappalainen

HYÖTYPELIEN KÄYTTÖ NUORTEN OPETUSMENETELMÄNÄ

– Case PELIMO

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tieto- ja viestintätekniikka

18.05.2020 | 50 sivua

Aku Lappalainen

HYÖTYPELIEN KÄYTTÖ NUORTEN OPETUSMENETELMÄNÄ

- Case PELIMO

Digitalisaatio ja diginatiivisuus mahdollistavat hyötypelien käytön opetuksessa. Tutkimuksissa on todettu, että hyötypelien käyttö voi usein motivoida nuoria opiskelijoita opiskelemaan perinteisiä opetusmenetelmiä tehokkaammin.

Tämän opinnäytetyön tarkoituksena oli hyötypelin suunnittelun ja teknisen toteutuksen tutkiminen sekä hyötypelin käännösjärjestelmän ja optimoinnin tekninen toteutus PELIMO-projektiin. PELIMO-projektin tavoitteena oli luoda peli 10–12-vuotiaiden monikulttuurisuuskasvatuksen käyttöön. Työssä tarkasteltiin hyötypelien roolia opetuksessa sekä suunnitteluvaiheen merkitystä hyötypelin pedagogisen tavoitteen saavuttamisen kannalta. Tekninen toteutus tuli suunnitella siten, että se soveltuu mahdollisimman hyvin nuorille suunnattuun hyötypeliin.

Peli toteutettiin käyttäen Unity-pelimoottoria ja C# ohjelmointikieltä. Käännösjärjestelmän toteutuksessa käytettiin apuna Google Sheets -palvelua.

Työssä tuotettiin toimiva käännösjärjestelmä pelin valikoille ja minipeleille sekä optimointiin suunnitellut työkalut, jotka täyttivät alkuperäiset vaatimukset. Käännösjärjestelmän ja optimoinnin tarkoituksena oli tehostaa pelin luomaa immersiota ja täten parantaa sen pedagogisia ominaisuuksia. Työn tuloksena syntyneitä työkaluja pystytään hyödyntämään pelin jatkokehityksessä.

ASIASANAT:

hyötypeli, digitalisaatio, optimointi, pelikehitys, lokalisointi

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information and Communication Technology

18.05.2020 | 50 pages

Aku Lappalainen

SERIOUS GAMES AS A PART OF JUVENILE EDUCATION

- Case PELIMO

Digitalization and digital nativity are allowing serious games to gain more footing in education than ever before. The term digital native describes a person who has grown up in the age of digital technology and thus has become familiar with modern digital tools by young age. Studies have shown that the use of serious games in education can often motivate young students to study more effectively than traditional teaching methods.

The goal of the thesis was to carry out research into serious games for children as well as to design and implement localization and optimization tools for the PELIMO project. The goal of the PELIMO project was to create a game to be used in multiculturalism education for 10–12-year-old children as a part of social studies. Furthermore, the tools had to be designed in a way that fits the best when it comes to serious games targeted at children. The commissioner of this thesis was theFirma, a learning environment of Turku University of Applied Sciences.

The game was created using Unity game engine and C# programming language and Google Sheets as localization tools. The theoretical section of the thesis contains research about serious games, localization, and optimization. The findings from the theoretical section were used to combine knowledge from these three subjects to create one working game.

The end results of the thesis were localization and optimization tools that fulfilled the original requirements for supporting the development of a serious game targeted at children. These tools will thus be utilized in the further development of the game project. The practical part of the thesis relies on the knowledge gained from theoretical part of the thesis to make the best decisions available.

KEYWORDS:

serious games, digitalization, optimization, localization, game development

SISÄLTÖ

KÄYTETYT LYHENTEET TAI SANASTO **6**

1 JOHDANTO **7**

2 NUORTEN HYÖTYPELIT **9**

2.1 Oppimisen pelillistäminen 10

2.2 Hyötypelien ja pelillistämisen hyödyt nuorille 11

2.3 Diginatiivisuus oppimisen pelillistämisessä 14

3 NUORTEN HYÖTYPELIN SUUNNITTELU **15**

3.1 Tarinankerronta hyötypeleissä 16

3.2 Hyötypelien suunnittelu nuorille 19

3.3 Lokalisoinnin suunnittelu 22

4 TEKNOLOGIAT HYÖTYPELIEN TOTEUTTAMISESSA **24**

4.1 Nuorten hyötypelien teknologisen toteuttamisen haasteet 24

4.2 Peliteknologiat tarkastelussa 25

4.2.1 Unity3D vastaan Unreal Engine 26

4.2.2 Pelimoottori vastaan selainpohjaisuus 26

4.2.3 Visual novel -lisäosan käyttö Unityssä 28

4.3 Optimointi 29

4.3.1 Optimoinnin hyödyt 29

4.3.2 Optimointi teknisellä tasolla 30

5 ESIMERKKIPROJEKTI PELIMO **32**

5.1 Pelin eteneminen 32

5.2 Pelin lokalisointi 33

5.2.1 Reaaliaikainen käännösjärjestelmä 36

5.3 Pelin optimointi 39

5.3.1 Suoritinkäytön optimointi 39

5.3.2 Muistinkäytön optimointi 42

5.3.3 Tallennustilan optimointi 43

6 LOPUKSI **44**

KUVAT

Kuva 1. Oppimispelien ulottuvuudet (Deterding ym. 2011, 5). 10
Kuva 2. Esimerkki pelillistetystä maantiedon opiskelusta (Seterra 2019). 11
Kuva 3. Flow-tila havainnollistettuna 12
Kuva 4. Esimerkki hahmojen ihmisenkaltaistamisesta (Aku Ankka 2020). 17
Kuva 5. Visuaalisessa novellissa pelaajan valintojen vaikutus ilmaistaan hahmojen ilmeillä sekä kommenteilla (Team Salvato 2017). 19
Kuva 6. Unityn, Unreal Enginen ja selainpohjaisen Phaserin ominaisuuksien vertailu. 28
Kuva 7. Esimerkki Funguksen vuokaaviotyökalusta. 33
Kuva 8. Kuvan seitsemän valintanäkymä pelissä. 33
Kuva 9. Puolipilkkujen käyttö CSV-tiedostossa erottimena. 34
Kuva 10. Mainlocalizationin komentosarja CSV-tiedoston lukemiseen. 35
Kuva 11. Word Id:n määrittely LocalizationElement-komponentille Unityn Inspectorissa. 36
Kuva 12. FetchWords-funktio MainLocalization-tiedostossa. 36
Kuva 13. Käännösfunktioiden logiikka kulkukaaviona, kun FetchWords-funktion parametrina on yksi. 36
Kuva 14. Suoritinkäytön nousu Image-komponentin alfakanavaa muutettaessa. 40
Kuva 15. Suoritinkäyttö kuvien resoluution pienentämisen jälkeen. 40
Kuva 16. Adobe Photoshopin kuvan koon muokkauksen näkymä. 41
Kuva 17. Profilerin näkymä tarkastellessa alkuvalikon muistinkäyttöä. 42
Kuva 18. Muistinkäyttö siirryttäessä alkuvalikon kautta Nikon tarinaan. 42

KÄYTETYT LYHENTEET TAI SANASTO

APK	Android Package Kit, Android-sovellusten tiedostotyyppi.
Canvas	Unityn komponentti käyttöliittymän toteutukseen.
CSV	Comma-separated values, tekstitiedosto, jonka taulukkorakenteen eri kentät on eroteltu toisistaan pilkuilla ja rivinvaihdoilla.
FPS	Frames per second, ruudunpäivitystaajuus.
GB	Gigatavu.
Inspector	Osa Unityn käyttöliittymää, jossa peliobjekteille määritetyt ominaisuudet ja asetukset näkyvät.
kB	Kilotavu.
MB	Megatavu.
PPI	Pixels Per Inch, pikselitiheyden yksikkö
Scene	Yhtä aikaa ladatuista peliobjekteista koostuva tila. Esimerkiksi yksi minipeli.
TSV	Tab-separated values, tekstitiedosto, jonka taulukkorakenteen eri kentät on eroteltu toisistaan tabulaattorilla.

1 JOHDANTO

Tässä opinnäytetyössä tutkitaan, miten suunnitellaan sekä toteutetaan valmis hyötypeli, jonka avulla voidaan pelillistämisen keinoin helpottaa nuorten opiskelua. Opinnäytetyön tarkoituksena on ensin käydä läpi teoriaa hyötypelien suunnittelusta nuorille, sitten hyötypelin käännösjärjestelmän teknistä toteutusta ja lopuksi hyötypelin optimointia teknisellä tasolla. Pelin tekninen toteutus toteutetaan Unityn pelimoottorilla, mutta opinnäytetyö käsittelee myös muita vaihtoehtoja Unitylle. Opinnäytetyön toimeksiantajana toimi theFIRMA, joka on Turun ammattikorkeakoulussa sijaitseva oppimisympäristönä toimiva projektitoimisto. Toimeksiantona oli hyötypelin suunnittelun ja teknisen toteutuksen tutkiminen sekä hyötypelin käännösjärjestelmän ja optimoinnin tekninen toteutus. Toimeksianto kohdistuu PELIMO-peliin, joka on Opetushallituksen rahoittama hanke, jonka tavoitteena on kehittää 10–12-vuotiaille lapsille suunnattu opetuspeli tukemaan monikulttuurisuuskasvatusta kouluissa. (PELIMO 2019.)

Toimeksiannossa tavoitteena on tuottaa peli, jossa saadaan yhdistettyä hyötypelin opetukselliset ominaisuudet ja opetuskäyttöön optimoitu tekninen toteutus toimivaksi kokonaisuudeksi. Pelin mekaniikoissa pyritään intuitiiviseen pelitapaan ja hyödyntämään nuoren sukupolven diginatiivisuutta, jolloin tarinankerronta ja opetukselliset ominaisuudet ovat pelaajalle helpommin ymmärrettävissä.

Työn kolme ensimmäistä lukua käsittelevät teoriaa hyötypeleistä ja niiden suunnittelusta, kun kohdeyleisönä on lapset, sekä tarkastellaan miten kehittäjän näkökulmasta tarinapohjaisen pelin käsikirjoitusten rakenne tulisi toteuttaa. Luku 4 käsittelee työn teknistä toteutusta, jossa käydään läpi mahdollisia teknisiä haasteita, jotka tulee ottaa huomioon, kun kohdeyleisönä on lapset ja peliympäristönä opetustilanne. Luvussa 4 käydään myös läpi pelimoottorin valintaa sekä sivutaan optimointia ja sen teknistä toteutusta. Luvussa 5 käydään läpi PELIMO-projektin vaiheita ja käytännön esimerkkejä siitä, miten opinnäytetyöhön liittyviä aiheita saadaan vietyä konkreettiseen peliin.

Viime aikoina on noussut pinnalle keskustelua varsinkin nuorten poikien oppimisvaikeuksista, joista Opetushallitus on julkaissut raportin ”Poikien oppimishaasteet ja -ratkaisut vuoteen 2025”, jossa käydään läpi mahdollisia syitä poikien oppimisvaikeuksiin. Yhtenä osana isompaa kokonaisuutta Opetushallitus ehdottaa hallitun digipelaamisen tukemista oppimisen menetelmänä. Hallitulla digipelaamisella tarkoitetaan aikuisten puuttumista liialliseen pelaamiseen ja sen ehkäisyyn sekä peliriippuvuuden hoitoon. Raportin

mukaan digipelaamista ei tulisi automaattisesti kategorisoida kielteiseksi ilmiöksi, vaan sitä voisi käyttää opetusmenetelmänä. (Opetushallitus 2017.)

Sen lisäksi, että digipelaamisen on havaittu auttavan oppimisvaikeuksissa, on sillä havaittu olevan myös monia muita hyödyllisiä ominaisuuksia. Yhtenä näistä nuorien pelaajien moraalinen järkeily, jonka on havaittu olevan verrokkiryhmää korkeammalla. Tällä tarkoitetaan sitä, että videopelien pelaaminen vaikuttaa nuoren moraalin kehittymiseen kiihdyttävällä tahdilla. (Williams 2019) Nuoren psyykkisen puolen kehityksen lisäksi digipelaamisen on todettu opettavan passiivisesti esimerkiksi ongelmanratkaisukykyä, kilpailullisuutta ja luovuutta. Monet viihdepelit pitävät sisällään pulmia, joita pelaajan tulee ratkaista edetäkseen pelissä, joka itsessään tukee ongelmanratkaisukykyä. Kilpailullisuus tulee luonnollisesti osana moninpelejä, joissa pelataan muita pelaajia vastaan. Kilpailullisuuden onkin todettu olevan hyödyllistä etenkin niille nuorille, jotka eivät pärjää perinteisissä urheilulajeissa. Luovuuden kehittyminen peleissä näkyy etenkin peleissä, joissa pelaaja pystyy vapaasti luomaan jotain uutta. Esimerkki tällaisesta on Mojangin julkaisema Minecraft, jossa pelaaja pystyy vapaasti suunnittelemaan ja rakentamaan kaikkea mitä mieli tekee. (Mojang 2020) Suoranaisten nuoreen vaikuttavien tekijöiden lisäksi digipelaaminen luo yhteisöllisyyttä nuoren ympärille sekä tarjoaa arkielämää laajemmat mahdollisuudet ystävyyssuhteiden luontiin. (Olson 2020.)

2 NUORTEN HYÖTYPELIT

Hyötypelit (engl. serious games) ovat pelejä, joiden pääasiallisena tarkoituksena on tavallisista viihdepeleistä poiketen keskittyä kehittämään pelaajan tosielämän taitoja käyttämällä pelaamisen tarjoamia ominaisuuksia oppimisen tehostamiseksi, eli niillä tulee olla pedagoginen tavoite. Hyötypelien opetusperiaate perustuu pelaajalle annettavaan palautteeseen, kuten esimerkiksi pelaajan tekemien valintojen pisteytykseen. Hyötypeleissä käytetään paljon samoja elementtejä kuin viihdepeleissä luomaan mukaansatempaavaa oppimisympäristöä kuten mielenkiintoista tarinankerrontaa, pelaajaa palkitsevaa palautejärjestelmää ja pelaajan vuorovaikutuksen merkitystä pelin etenemiseen. (Puolakka 2017.)

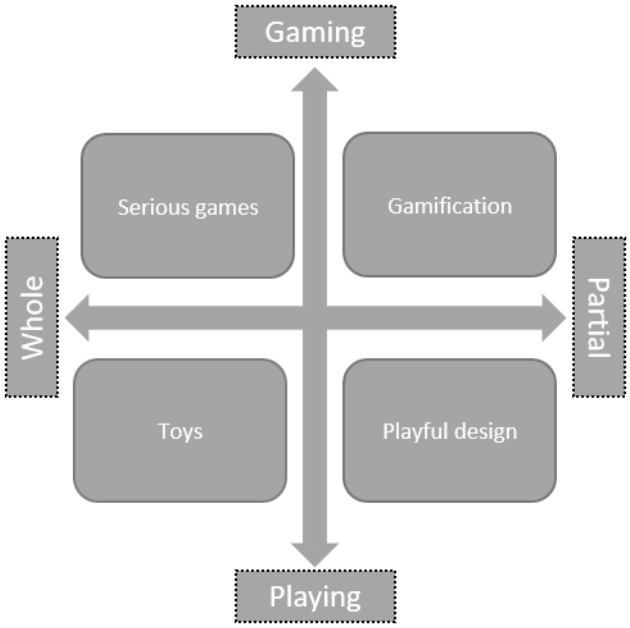
Hyvän hyötypelin tulisi olla mukaansatempaava ja inspiroiva, mutta samaa aikaa pelin tulee pysyä tiettyjen rajojen sisällä, jotta pelin pohjimmainen tarkoitus, oppiminen ei jää taka-alalle. Hyötypelin ja peliominaisuuksia sisältävän opetusmateriaalin erona on se, että pelaajan tulisi kokea etenevänsä valintojensa perusteella. Tällöin pelaaja saadaan tuntemaan, että pelissä on tietyt säännöt ja pelielementit, joita käyttämällä pelissä eteneminen tapahtuu. Pelin ei myöskään tule olla liian vaikea tai helppo, jolloin pelaaja kyllästyy helposti. (Wellesley 2014, 5–7.)

Hyötypelien yleistyminen näkyy Allied Market Reserchin teettämässä tutkimuksessa, jossa vuoden 2016 hyötypelien tuottama liikevaihto oli 2 731 miljoonaa dollaria ja ennustus vuoteen 2023 mennessä liikevaihdoksi on 9 167 miljoonaa dollaria (Sonawane 2019).

Hyötypeleiksi luetaan monesti myös simulaattorit, vaikka ne ovat oma pelilajinsa ja niitä ei tulisi sekoittaa hyötypeleihin. Simulaatioissa pyritään luomaan pelaajalle todenmukainen ympäristö, jossa pelaaja harjoittelee tosielämän tilanteita mahdollisimman realistisella tavalla. Puhtaassa hyötypelissä voidaan asiaa puolestaan lähestyä epärealistisemmalla tarinankerronnalla, kuten asettamalla pelaaja fiktiivisen hahmon asemaan (Design Digitally 2018).

Hyötypelien, pelillistämisen, lelujen ja pelimäisiä ominaisuuksia sisältävien tuotteiden eroa voidaan kuvata käyttämällä nelikenttää (Kuva 1.), jossa pystyakseleina toimivat leikkiminen ja pelaaminen sekä vaaka-akselina kokonaisvaltaisesti peliksi tarkoitettu ja osittain peliksi tarkoitettu. Esimerkiksi hyötypelit eroavat tässä erottelussa pelillistämisestä

(engl. gamification) siten, että pelillistetty tuote ei välttämättä suoraan ole tarkoitettu pelattavaksi vaan sisältää pelimäisiä ominaisuuksia, kun taas hyötypelit on varta vasten suunniteltu pelattavaksi.

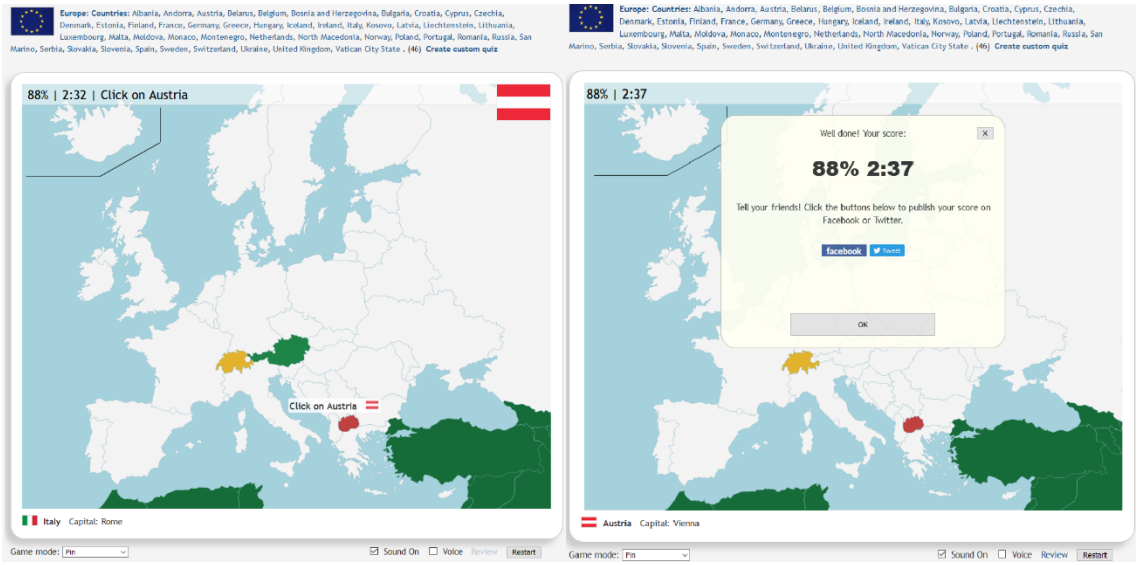


Kuva 1. Oppimispelien ulottuvuudet (Deterding ym. 2011, 5).

2.1 Oppimisen pelillistäminen

Oppimisen pelillistämisellä tarkoitetaan ilmiötä, jossa perinteisen opetuksen tueksi haetaan apua pelimaailman ominaisuuksista. Perinteisessä opetuksessa etenkin nuorilla voidaan havaita hankaluuksia motivaation ja opiskeluun sitoutumisen kanssa, jolloin oppimateriaalien pelillistämisellä voidaan yrittää helpottaa edellä mainittuja opiskelun haasteita. Näitä ominaisuuksia voidaan hakea esimerkiksi välittömällä palautteella pelillistetyssä oppimistilanteessa, jolloin käyttäjä tietää onnistuneensa, sekä saa siitä mielihyvää ja motivaatiota jatkaa oppimista. (Huang & Soman 2013, 7–15.) Kuvassa 2 näkyy esimerkki siitä, miten maantiedon opiskelua voidaan pelillistää. Kuvan 2 pelissä pelaajan tulee tunnistaa Euroopan maat ja klikata niitä. Mikäli pelaaja osuu ensimmäisellä oikeaan, kyseinen maa muuttuu valkoiseksi. Mikäli yrityksiä tulee 2-3 muuttuu maa keltaiseksi. Kolmen yrityksen jälkeen maa alkaa välkkyä punaisena, ja sitä klikattaessa se jää punaiseksi. Maan valinnan jälkeen pelin alareunasta näkee lisätietona maan pääkaupungin, jonka avulla pelaaja voi huomaamattaan oppia myös niitä. Kun kaikki maat on tunnistettu, tulee näytölle palaute pelaajan suoriutumisesta, sekä aika, joka pelaajalta

kului maiden tunnistamiseen. Tässä esimerkissä välitön palaute tulee tapahtumasta, joka seuraa maan klikkaamista, sekä lopullinen palaute pelin päättymisen yhteydessä.



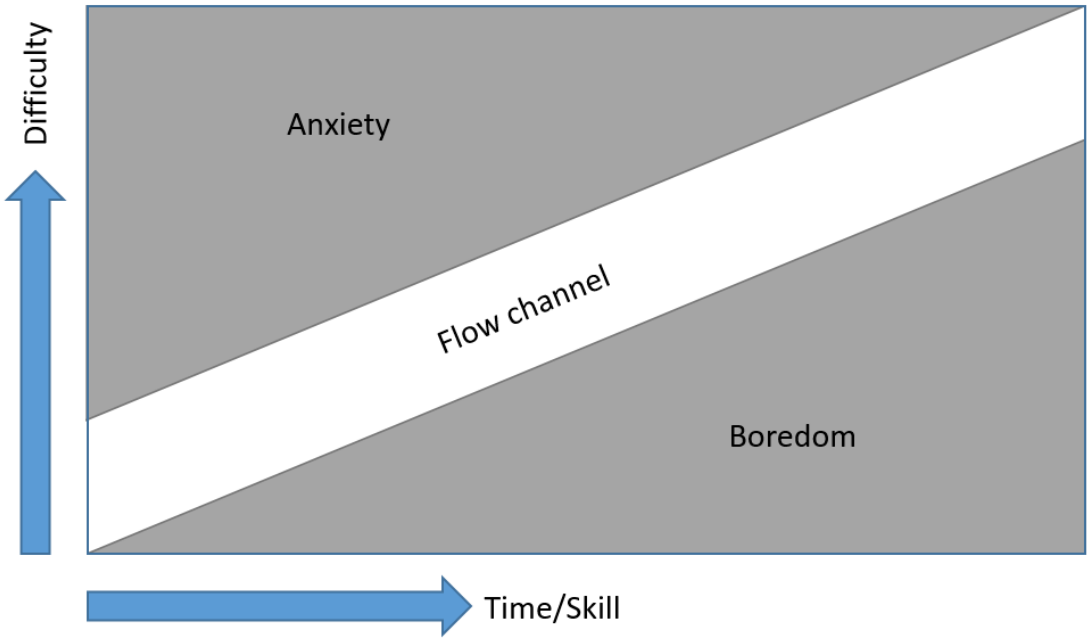
Kuva 2. Esimerkki pelillistetystä maantiedon opiskelusta (Seterra 2019).

Älylaitteiden ja tietokoneiden yleistyminen mahdollistaa kasvavissa määrin digitaalisten hyötypelien käytön opetuksen välineenä. Tilastokeskuksen tutkimuksen mukaan 99 % suomalaisista 10–14-vuotiaista lapsista käyttää aktiivisesti älypuhelinta. Tämä tarkoittaa, että lähes kaikilla on pääsy älylaitteeseen, jota voidaan käyttää opetuksen välineenä lasten opetuksessa. (Tilastokeskus 2017a).

2.2 Hyötypelien ja pelillistämisen hyödyt nuorille

Hyötypelien ja pelillistämisen on todettu stimuloivan aivoja erityisesti nuoremmilla ihmisillä, jolloin pelien opetus saadaan jäämään aivoihin paremmin verrattuna perinteisiin opetusmenetelmiin. Aivojen stimuloinnissa voidaan käyttää apuna myös flow-tilaa, jossa käyttäjän keskittyminen peliin on täydellistä. Flow-tilalla kuvataan psykologista ilmiötä, jossa käyttäjän mieli tai keho on viritetty äärimmilleen saavuttaakseen jotain vaikeaa tai oppimisen arvoista vapaaehtoisesti. Flow-tilassa oppiminen tai fysiologinen suorituskyky on optimilla tasolla. Flow-tilaan päätynyt käyttäjä immersoituu täysin tekemiseen tai oppimiseen. (Cziksentmihalyi 1990, 1–8.) Pelikontekstissa immersiolla tarkoitetaan psykologista eläytymistä, jossa pelaaja uppoaa vahvasti pelin luomaan virtuaalitodellisuuteen. (Sharma 2017) Päästäkseen flow-tilaan tulee opiskeltavan asian haastavuuden

tasapainotella käyttäjän taitotason kanssa, kuten kuvassa 3 tulee ilmi. Pelaamisessa flow-tilalla tähdätään tunteisiin, jotka johtavat peliin sitoutumiseen, uusien asioiden selvittämisen ja oppimisen tehostamiseen. Käyttäjän päästessä flow-tilaan, pystytään sulkemaan ulkomaailman haittatekijät pois ja keskittymään täysin peliin jolloin pystytään hyödyntämään pelaajan koko oppimiskapasiteetti ja tekemään oppimistilanteesta mielekkäämpi. Vaikka flow-tilaa ilmenee myös perinteisesti lukemalla oppimisella, on nuoremman peleistä kiinnostuneen sukupolven saattaminen flow-tilaan helpompaa, kun opetustilanteeseen yhdistetään peli, jonka suunnittelussa tähdätään flow-tilan luomiseen. (Murphy ym. 2015, 16.)



Kuva 3. Flow-tila havainnollistettuna

Pelien käytön opetuksessa on myös huomattu oppilaiden välistä kanssakäymistä liittyen oppimateriaaleihin ja niistä keskusteluun opetustilanteen jälkeen. Syinä tähän on muun muassa pelien luoma innostus oppimateriaalia kohtaan, jolloin oppilaat voivat helpommin käydä keskustelua keskenään opitusta myös opetustilanteen ulkopuolella, sekä hyötypelien tapa antaa välitöntä palautetta pelaajan valinnoista. Tällaisessa tilanteessa omia pelissä tekemiään valintoja voi välittömästi käydä läpi muiden kanssa. (Gamelearn Team 2017.) Käytettäessä pelejä opetuksessa on havaittu, että oppilaat, jotka kokevat oppivansa paremmin käytännönläheisellä tavalla opetettuna omaksuvat opetukset pelin kautta opiskeltuna paremmin kuin tavallisilla opetusmetodeilla opeteltuna. Esimerkiksi matemaattisia ja geometrisia aiheita opiskellessa, kyseisen ryhmän opiskelijoiden on

ollut helpompi hahmottaa opetettava asia pelin muodossa, jossa opiskellaan aihetta käytännön esimerkeillä verrattuna perinteisiin opetusmetodeihin. Muita aihealueita, joissa on havaittu vastaavaa, on muun muassa strategisointi, päättely, kriittinen ajattelu, ongelmaratkaisukyky, luovuus, maantieteelliset taidot, tiede ja kielten opiskelu. (Papanastasiou ym. 2017, 46–53.)

Opetushallitus käynnisti syksyllä 2017 ohjelmakokonaisuuden, jossa tarkastellaan poikien oppimishaasteita ja ratkaisuja niihin vuoteen 2025 mennessä. Selvityksessä käy ilmi poikien opiskelumotivaation puute verrattuna tyttöihin sekä alhaisemmat PISA-testien tulokset. Yhtenä ratkaisuna näihin on ehdotettu opetuksen pelillistämistä, jonka tarkoituksena on poikien motivaation lisääminen oppimiseen kilpailullisuuden ja pelillisyyden avulla. Selvityksessä todettiin myös yhdeksi poikien oppimiseen negatiivisesti vaikuttavaksi tekijäksi peliriippuvuus. Mikäli poikien opetusta häiritsevän peliajankäytön saisi valjastettua opetuskäyttöön käyttämällä hyötypelejä opetusmateriaalin ohessa, saataisiin mahdollisesti nostettua poikien saavutuksia koulutuloksissa lähemmäs tyttöjen tasoa. (Opetushallitus 2019.)

Tilastokeskuksen tutkimuksen mukaan 25 % 10–14-vuotiaista pojista pelaa verkkopelejä päivittäin, kun taas tytöillä sama tulos on vain 1 %. Konsolipeleillä osuuksien suhteet ovat lähes samat, pojilla 18 % ja tytöillä 1 %. Mobiilipelien suosio on muihin pelityyleihin verrattuna tasaisempi, pojilla osuus on 34 % ja tytöillä 19 %, joka luultavammin johtuu älylaitteiden yleistymisestä ja mobiilipelien keveydestä. Vaikka tilastojen mukaan viihdepelaaminen on poikien kohdalla huomattavasti suositumpaa, opetuspelejä tarkastellessa on poikien, sekä tyttöjen tulokset paljon tasaisemmat. Tutkimuksen mukaan sukupuolesta riippumatta 10–14-vuotiaista 31 % pelaa opetuspelejä vähintään kerran kuukaudessa. Opetuspelien sukupuolesta riippumaton suosio johtuu luultavasti siitä, että opetuspelien pelaaminen vapaa-ajalla ei ole niin suosittua kuin viihdepelien vaan niitä pelataan koulussa opetustilanteissa. (Tilastokeskus 2017b.)

Tutkittaessa hyötypelien käyttöä poikien oppimisen tehostamisessa on huomattu merkittävä korrelaatio oppimisessa ja opiskelumotivaation määrässä hyötypelien käytön välillä poikien oppimisessa. Tilburgin yliopistossa tehdyssä tutkimuksessa käytettiin kahta verrokkiryhmää, toisella ryhmällä oppimateriaalina toimi peli ja toisilla saman asiasisällön sisältävä teksti. Ryhmien sukupuolijakaumaan ei kiinnitetty huomiota. Tutkimuksen lopussa testattaville pidettiin koe, jossa piti käyttää tutkimuksen oppimateriaalista opittua tietoa hyväksi. Peliryhmän keskiarvotulos oli huomattavasti parempi kuin ryhmän, jonka oppimateriaali oli tekstimuodossa. Huolimatta yleisestä olettamuksesta, että pelit ovat

motivoivampia kuin teksti, opiskelumotivaatiota tutkittaessa huomattiin, että tekstistä opiskelevat olivat motivoituneempia ja pitivät sitä hyvänä tapana oppia. Seuraavassa testissä ryhmät jaettiin sukupuolien mukaan ja huomattiin, että pojat pitivät pelaamista huomattavasti motivoivampana kuin tekstistä oppimista. Myös verrattaessa poikien oppimista pelistä tekstistä opiskeleviin poikiin, huomattiin, että peleistä opiskelevat saivat huomattavasti paremmat pisteet kuin tekstiryhmän pojat. Tyttöjen ryhmää tutkittaessa ei havaittu merkittävää eroa peli- ja tekstiryhmien välillä. (Stege ym. 2011, 3–7.)

2.3 Diginatiivisuus oppimisen pelillistämisessä

Diginatiivilla tarkoitetaan henkilöä, joka on kasvanut ympäristössä, jossa teknologiat, kuten tietokoneet ja internet ovat olleet jatkuvasti läsnä. Terminä diginatiivisuus nousi pinnalle Marc Prenskyn vuonna 2001 julkaisemassa artikkelissa On the Horizon, jossa Prensky käsitteli opettajien vaikeuksia opettamisessa.

Vaikka diginatiivisuus ei ole vielä itsestäänselvyys kaikkialla, voidaan lähes kaikkia Suomessa peruskoulua käyviä kutsua diginatiiveiksi. Siinä missä ennen nuoret ovat käyttäneet aikaansa esimerkiksi kirjojen lukemiseen, on nykypäivänä digitaalinen media suurelta osin korvannut printtimedian. Prenskyn mukaan yliopistosta valmistuessaan opiskelijat ovat käyttäneet elämästään noin 5 000 tuntia lukemiseen, kun taas vastaava luku videopeleihin käytetyssä ajassa on 10 000 tuntia puhumattakaan television katseluun käytetystä ajasta, joka on 20 000 tuntia. Prenskyn julkaisu Digital Natives, Digital Immigrants on julkaistu vuonna 2001, jolloin digitaalisen median osuus oli vain murto-osa nykyiseen verrattuna. Lukujen voidaankin olettaa nousseen entisestään digitalisaation räjähdysmäisen kasvun ansiosta. (Prensky 2001, 1.)

Nuorten tavat hankkia ja käsitellä tietoa ovat muuttuneet. Siksi myös opetuksen pitäisi pysyä muutoksessa mukana. Prenskyn mukaan muutoksessa ei ole kyse vain siitä, että diginatiivit pitävät enemmän digitaalisesti kuten pelillistetysti, tuodusta informaatiosta, vaan johtuen digitaalisen informaation suuresta kulutuksesta, he pystyvät myös prosessoimaan digitalisoitua tietoa aivan eri tavalla verrattuna aiempiin sukupolviin.

3 NUORTEN HYÖTYPELIN SUUNNITTELU

Pelisuunnittelun kulmakivinä toimivat pelimekaniikka, tarinankerronta ja vuorovaikutus. Pelisuunnittelulla tarkoitetaan prosessia, jossa peliin kehitetään sääntöjä, teemoja ja tarinoita, jotka ohjaavat pelin kulkua. (Mankkinen 2019) Pelimekaniikka keskittyy pelin toiminnallisuuteen, sääntöihin sekä pelissä tapahtuviin toimintoihin. (GameDesigning 2020a) Tarinankerronnan suunnittelussa pyritään luomaan pelille mukaansatempaavaa juonta, joka motivoi pelaajaa jatkamaan pelaamista. Vuorovaikutuksen suunnittelussa keskitytään muokkaamaan pelin ja pelaajan välistä vuorovaikutusta, kuten nappien painalluksia mahdollisimman luonnollisiksi. Johtuen hyötypelien pohjimmaisesta tarkoituksesta eli oppimisesta, tulee kehittäjäryhmän keskittyä pelin suunnittelussa siihen, että pelin ominaisuudet tukevat oppimista. Suurimpana haasteena suunnittelussa onkin pelisuunnittelun ja pelin opetustarkoituksen riittämätön integraatio sekä kehitysryhmän jäsenien väliset erot pelin eri osuuksien ymmärtämisessä. Varsinaisesta pelin teknisestä toteutuksesta vastaavan henkilön ja pelin pedagogisen sisällön laatijan välinen yhteisymmärrys onkin tärkeää hyvän hyötypelin suunnittelussa. Mikäli pelin suunnittelusta vastaavilla henkilöillä ei ole kokemusta siitä mitä pedagogiikka peleissä tarkoittaa ja miten sitä saadaan korostettua pelimekaniikoilla, voi hyötypelin opetustarkoitus jäädä taka-alalle. Myös pelimekaniikat tulee suunnitella sellaisiksi, että niillä saadaan tuettua pelin pedagogista tarkoitusta. Ne eivät kuitenkaan saa olla liian dominoivassa asemassa, jolloin ne voivat alentaa pelin opetuksellista puolta. (Bellotti ym. 2010, 23–26.)

Hyötypelin suunnittelu voidaan jakaa viiteen eri malliin, jotka tulisi ottaa huomioon suunnitteluvaiheessa:

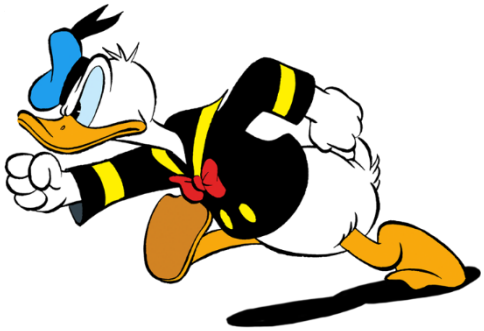
1. integraatiomalli
2. kognitiomalli
3. esillepanomalli
4. sosiaalisen kanssakäymisen ja opetuksen malli
5. sitoutumismalli.

Integraatiomallissa liitetään pelielementit ja oppimistavoitteet saumattomasti yhteen. Kognitiomallissa pyritään aktivoimaan pelaajassa metakognitiivisia ominaisuuksia ja stimuloimaan pelaajan kykyä prosessoida merkityksellistä tietoa pelikokemuksen aikana. Esillepanomallissa tavoitteena on varmistaa, että peliympäristön ja materiaalin prosessointi on tehokasta. Esillepanomallia voidaan tukea esimerkiksi korostamalla pelin

kannalta olennaisia asioita pelissä. Sosiaalisen kanssakäymisen ja opetuksen mallissa pyritään yhdistämään peliin ominaisuuksia, jotka helpottavat esimerkiksi pelaajan pelitavan tulkintaa. Tällaisia ominaisuuksia on esimerkiksi pelin lopussa näytettävät pisteet. Sosiaalisen kanssakäymisen ja opetuksen mallia ei välttämättä näy itse pelissä, vaan sitä voidaan käyttää myös pelitilanteen jälkeen pelisuorituksen arviointiin. Sitoutumismallilla kuvataan tapaa, jolla pelikokemuksesta luodaan motivoiva yhtenäinen paketti, jonka tarkoituksena on saada pelaaja kiintymään peliin ja täten oppimaan paremmin. Näitä malleja käyttämällä saadaan yhdistettyä pelin pedagoginen suunnittelu pelisuunnitteluun, jolloin saadaan hyödynnettyä molempien puolien ominaisuuksia paremmin. (Bellotti ym. 2014, 4.)

3.1 Tarinankerronta hyötypeleissä

Jotta hyötypeleistä saataisiin mielenkiintoisia, on niihin monesti upotettava tarina. Tarinan tyylin ja etenemisen määrittelee kohderyhmä. Esimerkiksi lapsille suunnatussa hyötypelissä voidaan käyttää tarinankerronnassa kertojaa tai antropomorfisia hahmoja. Kertoja on hahmo, jolla on eläimen ruumis, mutta pystyy kuitenkin puhumaan ja ajattelemaan kuin ihminen. Antropomorfisten hahmojen käytöllä pyritään herättämään tunteita pelaajassa, joilla saadaan vahvistettua oppimista tai vaikka pelkästään saamaan lapsi kiinnostuneemmaksi pelistä. (Libell 2004.) Kuvassa 4 on esimerkki tilanteesta, jossa antropomorfiaa käytetään lasten ja nuorten Aku Ankka sarjakuvissa vahvistamaan kohderyhmän kiinnostusta sarjakuvaan ja lisäämään immersiota aiheuttavia tekijöitä. Aikuisille suunnatuissa hyötypeleissä taas voidaan käyttää tosielämään liittyviä arkisia asioita tukemaan tarinankerrontaa. Esimerkiksi kemiaa opettava peli voi käyttää tarinankerronnassa arkisia esimerkkejä, kuten sitä, että tavallinen pöytäsuola on pääasiassa natriumkloridista (NaCl) koostuva suola. Mikäli peli on suunnattu urheilijoille voi yhtenä tarinankerronnan tyylinä toimia, jokin asia mikä sitoo pelaajan urheiluun. Esimerkiksi pelaaja voi edetä jalkapallokentällä kymmenen jaardia jokaista oikeaa vastausta kohden. Joka tapauksessa tarinan tyylin tulisi mukailla kohderyhmälle ennestään tuttuja asioita, joihin on yhdistetty pelin pedagoginen tarkoitus, jolloin mielenkiinto peliin saadaan säilymään ja pelaaja oppimaan samalla kun hän käsittelee jotain jo ennestään tuttua. (Naul 2018, 7–14.)



Kuva 4. Esimerkki hahmojen ihmisenkaltaistamisesta (Aku Ankka 2020).

Tarinankerronnan rooli hyötypeleissä näkyy monella tavalla lopullisen pelin vaikuttavauudessa. Toimivalla tarinankerronnalla tavoitellaan etenkin immersiota, sitoutumista, motivaation lisäämistä sekä tärkeimpänä oppimistulosten kasvattamista. Tarinankerronnallisella immersiolla kuvataan tilaa, jossa pelaaja irtautuu todellisesta elämästä ja kiinnittää kaiken huomionsa peliin, jolloin pelaaja saattaa jopa menettää ajantajunsa ja pelata peliä suunniteltua kauemmin. Tutkimuksessa (Wood ym. 2007, 40–43) kysyttiin 280 henkilöltä, että ovatko he joskus menettäneet ajantajun ja immersoituneet peliin täysin. Vastaajista 99 % vastasi myöntävästi. Suurin yhtenäinen tekijä tähän oli henkilöiden mielestä juoni. Tutkimuksen mukaan testihenkilön halu tietää mitä tulee tapahtumaan pelissä seuraavaksi oli yleinen teema siinä, miten hyvän juonen vaikutus näkyy käyttäjässä. Tarinankerronnalla luodun immersion on havaittu vaikuttavan pelaajan uskomuksiin ja asenteeseen opetusmateriaalia kohtaan tavalla, joka auttaa sisäistämään asian. (Naul 2018, 19.) Sitoutumisella kuvataan käytökseen, tunteisiin tai kognitiivisiin ajattelumenetelmiin pohjautuvaa kykyä pitää käyttäjän keskittyminen aiheessa. Tarinankerronnan merkitys sitoutumisen luomisessa on saada pelaajan mielenkiinto pysymään aiheessa, vaikka aihe itsessään ei olisikaan pelaajan mielestä mielenkiintoinen. Tarinankerronan rooli peleissä on merkittävä, sillä pelaaja menettää mielenkiinnon helpommin peliä kohti, jos siinä ei ole tarinaa. Tutkittaessa kahta kielioppiin perustuvaa peliä, jossa toisessa käytettiin tarinankerronnallista näkökulmaa ja toisessa puhtaasti opetuksellista, havaittiin, että pelaajat kokivat tarinankerronnallisen lähtökohdan omaavan pelin huomattavasti mukaansatempaavammaksi ja pystyivät sitoutumaan sen pelaamiseen. (Hallinen ym. 2009, 4.) Tarinankerronnan vaikutus motivaatioon oppia näkyy esimerkiksi fantasiaan pohjautuvissa tarinoissa. Pelaajien on havaittu kiinnostuvan peleistä, joissa

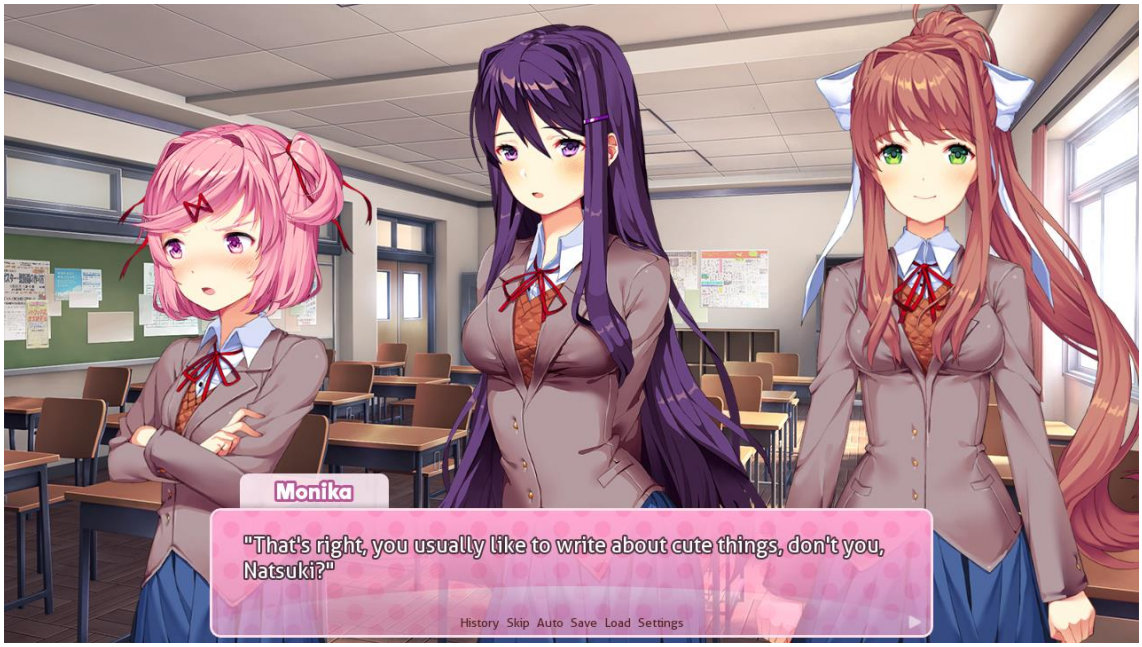
tarinankerronnassa käytetään fantasiaan liittyviä elementtejä, jolloin he saavat lisää motivaatiota jatkaa pelin pelaamista ja peli saa lisää aikaa käyttäjältä. (Naul 2018, 17.)

Onnistuakseen hyötypeleissä tarinankerronta vaatii edellä mainittujen aiheiden lisäksi sen, että tarinankerronnan hyötynäkökulmien perimmäinen tarkoitus eli oppiminen paranee. Monesti tarinankerronta itsessään ei tuo peliin opetuksellisesta näkökulmasta mitään lisää, vaan sillä nimenomaan pyritään siihen, että sen eri vaikutuksien (immersion, sitoutumisen ja motivaation) aiheuttamia hyötyjä pystytään käyttämään apuna opetuksessa. Onnistuneen tarinankerronnan vaikutuksena pelaaja ei enää koe opiskelevansa aihetta vaan pelaa peliä, joka opettaa samalla. Kun pelaaja voi pitää hauskaa ja uppoutua pelimaailmaan, opiskeluun tulee käytettyä huomaamatta enemmän aikaa. Kun taas hyötypelissä, jossa tarinankerronnalla ei ole yhtä suurta roolia, lopahtaa kiinnostus pelaamiseen nopeammin. Joissain tapauksissa tarinankerrontaa voidaan suoraan käyttää opetusmenetelmänä. (Avraamidou & Goedhart 2017.)

Vaikka opetuspeleissä tarinan monesti tulee vain vahvistaa oppimistilannetta luomalla immersiota oppimistilanteen ympärille, voidaan tarinankerrontaa itsessään käyttää opetusvälineenä. Visuaalisissa novelleissa pelaaja kohtaa pelin opetustarkoituksen pelillisesti hyvin yksinkertaisessa ympäristössä. Visuaaliset novellit ovat pelejä, joiden pelaaminen muistuttaa huomattavasti kirjan lukemista. Tarina tuodaan pelaajalle tekstimuodossa, usein kirjoitettuna ensimmäisestä persoonasta päähahmon näkökulmasta. Kirjoista poiketen visuaaliset novellit sisältävät paljon kuvia, joilla peliin luodaan tunnelmaa ja tuodaan pelaajalle tietoa esimerkiksi tarinan tapahtumapaikasta. Kuvat ovat harvoin animoituja, mutta hahmoilla on useita ilmeitä ja asentoja, joita vaihtelemalla saadaan pelaajalle vaikutus siitä, että pelaajan pelissä tekemät valinnat vaikuttavat pelin hahmoihin ja niiden tekemisiin. Visuaalisten novellien tapauksessa opetettavat aiheet tulevat pelaajalle luettavaksi tekstimuodossa ja niitä korostetaan pelin visuaalisilla elementeillä. Näitä ovat hahmojen ilmeiden vaihtelut tai esimerkiksi pelin kuvaillessa kasveja, voidaan kasvin kuva näyttää pelissä samalla kun pelihahmo kertoo faktoja siitä. Pelaaja voi itse vaikuttaa pelissä etenemiseensä yksinkertaisesti klikkaamalla pelin esittämiin kysymyksiin annettuja vaihtoehtoja. (Lebowitz & Klug, 2011, 192–197.) Visuaaliset novellit ovat tästä syystä erinomainen tapa saada tuotua pelaajalle välitöntä palautetta liittyen pelissä pelaajan antamiin vastauksiin esimerkiksi suoralla dialogilla pelin hahmojen ilmeiden ja kommenttien perusteella, kuten kuvassa 5 tulee ilmi. Tällaisella visuaaliseen ja tekstipohjaiseen pohjautuvalla tarinankerronnalla saadaan karsittua peliä vaikeuttavien

pelielementtien tai pelimekaniikkojen määrää, jolloin oppiminen ei jää kiinni siitä osaako pelaaja käyttää laitetta pelaamiseen vai ei. (Pynn 2017, 28.)

Edellä mainittujen opetusta tukevien ominaisuuksien lisäksi on havaittu, että sarjakuvien käyttö lukihäiriöisten opetuksessa on antanut hyvin positiivisia tuloksia. Sarjakuvien käytön opetuksessa on havaittu tukevan lukihäiriöistä kärsivien henkilöiden opiskelua, sillä sarjakuvien tarjoama visuaalinen vaste antaa kontekstin ja purkaa tekstin sisällön visuaaliseen muotoon. Sarjakuvien lailla visuaalisten novellien hyötynä on se, että pelaajalle saadaan annettua paljon tekstimuodossa olevaa informaatiota, jota tuetaan jatkuvasti muuttuvilla visuaalisilla elementeillä. (Redford 2019.) Tällöin pelaaja jaksaa helpommin keskittyä aiheeseen ja pystyy tulkitsemaan tekstiä paremmin verrattuna tavalliseen tekstiseinään.



Kuva 5. Visuaalisessa novellissa pelaajan valintojen vaikutus ilmaistaan hahmojen ilmeillä sekä kommenteilla (Team Salvato 2017).

3.2 Hyötypelien suunnittelu nuorille

Suunnitellessa hyötypelejä ja pelejä yleisesti, kun kyseessä on nuoret ikäryhmät pitää ottaa huomioon ihmisen nopea motorinen sekä psyykkinen kehitys. Siinä missä sama peli voi vedota yhtä hyvin 16- ja 20-vuotiaaseen pelaajaan, on nuoremmilla pelaajilla kyseinen ikähaarukka paljon pienempi. Kaksivuotiaalle kiinnostavat pelimekaniikat ja aiheet ovat monesti jo liian helppoja ja itsestäänselvyyksiä neljävuotiaalle. Tämän takia,

kun nuorille aletaan suunnitella peliä, tulee ikähaarukka rajata selkeästi. Tässä alaluvussa käydään läpi, miten pelin suunnittelu eroaa iän mukaan ikäryhminä 1–2 vuotta, 3–6 vuotta, 6–8 vuotta sekä 9–10 vuotta. (Funday Factory 2017.)

Kun aletaan suunnittelemaan peliä hyvin nuorelle ikäryhmälle, on huomioitava erilaisia lapsen kehitykseen liittyviä asioita, kuten esimerkiksi 1–2-vuotiaiden tapauksessa motoristen taitojen kehittyminen. Tästä syystä pelin suunnittelussa tulee ottaa huomioon se, miten pelin mekaniikat toimiva, jotta ne olisivat helppoja pelaajan hallita. Tällaisia ovat esimerkiksi näytön näpäyttäminen ja asioiden sormella raahaaminen. Johtuen motorisen kehityksen tilasta, pitää myös huomioida se, ettei pelaaja ole välttämättä kykenevä raahaamaan asioita pelissä kovinkaan tarkasti, jolloin virhemarginaalia tulee kasvattaa. Tämä tapahtuu usein esimerkiksi suurentamalla aluetta, johon esine tulee raahata, jolloin tarkkuus ei ole niin tärkeää. Mikäli pääasiallisena pelilaitteena toimii puhelin, tulee myös ottaa huomioon, että pelaajakunta ei välttämättä osaa pitää käsiään pois näytön tieltä pelattaessa, jolloin pelin kannalta tärkeät elementit tulee sijoittaa niin, että ne ovat helposti ulotettavissa. Koska hyvin nuoret pelaajat ovat innokkaita klikkailemaan ja näppäilemään kaikkea, mitä näytöllä näkyy, on interaktiivisten objektien olla helposti erotettavissa esimerkiksi taustagrafiikoista ja muista elementeistä, joista klikkaamalla ei tapahdu mitään. Interaktiiviset osat käsittävät pelin elementit, joiden kanssa vuorovaikuttamalla pelaaja etenee pelissä. Tähän päästään sillä, että pelin kannalta merkittävät elementit ovat esimerkiksi eri sävyisiä kuin graafiseen ilmeeseen liittyvät elementit tai ne liikehtivät näytöllä tavalla, joka erottaa ne muusta. Mikäli pelaaja ei hetkeen ole edennyt pelissä voidaan seuraavaa tarvittavaa tapahtumaa korostaa esimerkiksi tuomalla jotain objektia lähemmäs pelaajaa tai liikuttamalla sitä, jolloin huomio kiinnittyy siihen. Kaiken tarvittavan informaation tulee olla erinäisten visuaalisten vihjeiden takana, koska tämän ikäryhmän pelaajat eivät vielä osaa lukea. Iän vuoksi pelaajien on vaikea keskittyä moneen asiaan yhtä aikaa, ja aika, jonka pelaaja pystyy keskittymään tiettyyn asiaan, on vain sekunteja. Tämän takia pelissä tulisi olla vain tiettyyn pisteeseen saakka tekemistä kerrallaan, jolloin pelaaja ei häkelly liiasta informaatiosta. 1–2-vuotiaille peliä suunnitellessa kannattaa monesti pysyä kaksiulotteisessa maailmassa, sillä näin nuorille kolmiulotteisten asioiden hahmottaminen voi olla vielä hankalaa. (Funday Factory 2017.)

3–6-vuotiailla motoriset taidot ovat jo tarpeeksi kehittyneet jopa vähän haastavampiinkin pelimekaniikkoihin. Myös kyseisen ryhmän psyykkinen kehitys on edennyt tasolle, joka mahdollistaa enemmän itsenäistä ajattelua muiden matkimisen sijaan. Kun käsitellään esikouluikäisiä pelaajina, yhdeksi suurimmista haasteista nousee esiin kyseisen

ikäryhmän lyhyt huomiojänne. Tämä näkyy pelaamisessa siten, että mikäli jonkin tavoitteen, esimerkiksi kentän läpäisy kestää liian kauan tai uuden mekaniikan opettelu vie aikaa pelaamiselta, menettää pelaaja kiinnostuksensa peliin ja siirtyy tekemään jotain muuta. Pelin pitäisi palkita pelaaja usein, että näin ei tapahtuisi. Tatiana Lyng Pedersen mukaan esikouluikäisille pelejä suunnitellessa pelien käyttöliittymän tulisikin olla niin intuitiivinen, että pelin pelaaminen ei vaadi erillistä ohjeistusta. (Funday Factory 2017.)

6–8-vuotiaita tarkastellessa iän tuoma kehitys aiheuttaa muun muassa sen, että pelaajat ovat keskittyneempiä saamaan jo aloitetun tehtävän tehdyksi ja siirtyvät vasta sen jälkeen eteenpäin. Tästä syystä pelissä etenemisen havainnointi on tärkeää, jotta pelaaja saa onnistumisen tunteen ja kokee etenevänsä pelissä. Tällaisia mekaniikkoja ovat muun muassa hahmon tason kasvaminen sekä uusien asioiden avaaminen. Onnistumisen tunnetta voidaan tässä ikäryhmässä korostaa sillä, että autetaan pelaajaa selkeästi tunnistamaan, että hän on oppinut jotain uutta pelin myötä. Esimerkiksi valintojen antaminen pelaajalle, mitkä vaikuttavat pelin kulkuun, antavat pelaajalle onnistumisen ja oppimisen tunteen, kun he vastaavat oikein. Onnistumisesta seuraavien palkintojen, kuten saavutuksien tai mitalien saaminen ja niiden jakaminen kavereiden tai vanhempien kanssa lisää entisestään motivaatiota jatkaa pelaamista. Aiemmin käytyihin ikäryhmiin poiketen 6–8-vuotiaat pitävät peleissä siitä, että voivat itse olla vastuussa pelissä tapahtuvista asioista. Tämän takia pelin tulisi antaa selvät ohjeet pelaajalle siitä, mitä pelaajan tulisi tehdä ja mihin se johtaa. (Funday Factory 2017.)

Aikaisemmin läpikäytyjen ikäryhmien pelit ovat pääasiassa hetken ajankuluksi suunniteltuja. 9–10-vuotiaille pelejä suunnitellessa voidaan alkaa jo miettimään keinoja, joilla pelistä saadaan mielenkiintoisempi pidempiäkin pelijaksoja huomioiden. Kyseisen ikäryhmän pelien suunnittelussa voidaankin käyttää hyvin mentaliteettia, että pelin oppimisen tulisi olla helppoa. Jos pelissä haluaa kuitenkin olla todella hyvä, tulee sen vaatia enemmän pelaajalta. Kuten aiemmissakin ikäryhmissä, palautteen merkitys on huomattava pelaajan motivaation säilyttämisen kannalta. Poiketen aikaisemmista ikäryhmistä, 9–10-vuotiaille voidaan alkaa suunnittelemaan ”hiekkalaatikko” tyylisiä pelejä, joissa vain pelaajan mielikuvitus on rajana sille mitä pelissä tapahtuu. Hiekkalaatikkopelien onkin tarkoitus avartaa lapsen ajatusmaailmaa ja antaa heidän itse pelata peliä niin kuin he itse haluavat, unohtamatta pelin pedagogista tarkoitusta. (Funday Factory 2017.)

3.3 Lokalisoinnin suunnittelu

Lokalisoinnilla tarkoitetaan tuotteen, julkaisun tai ohjelmiston sovittamista eri maiden kulttuuriin. Lokalisointiin kuuluu esimerkiksi tuotteen kääntäminen eri kielille, maantieteellisten viittausten käytön tarkastelu, sukupuoliroolien tarkastelu sekä valuutan vaihtaminen. (Rouse 2015) Etenkin hyötypeleissä, joissa immersion merkitys on suuri pelin opetuksen kannalta, tulee lokalisointi ottaa huomioon suunnitteluvaiheessa, sekä siihen tulee panostaa, mikäli hyötypelin halutaan toimivan muuallakin, kuin pelin pääasiallisessa kohdemaassa.

Hyötypeliä lokalisoidessa on lokalisoijien tunnettava peli ja sen pelimekaaniikat hyvin. Tällöin pelin käännökseen ja muihin lokalisointiin liittyviin muutoksiin saadaan tehtyä järkevä strategia, jonka avulla lokalisointi tapahtuu niin, että pelin immersio ei kärsi lokalisoinnin seurauksena. Hyötypelien lokalisointi voikin tietyissä tapauksissa olla paljon monimutkaisempi prosessi, kuin vain suora käännös olemassa olevasta pelistä. (Lachat-Leal 2016, 2.)

Vaikka tekstin kääntäminen kielestä toiselle on vain osa lokalisointiprosessia, on myös sen aiheuttamia haasteita pohdittava laajemmalla näkökulmalla. Esimerkiksi käännettäessä pelejä saksan kielelle huomataan, että kieli käyttää kirjaimia, joita ei välttämättä pelin suunnitteluvaiheessa valitusta fonttikirjastosta löydy. Tämä ilmiö korostuu entisestään siirryttäessä kieliin, jotka käyttävät latinalaisen kirjainjärjestelmän sijaan muita kirjainjärjestelmiä, kuten esimerkiksi kyrillistä kirjainjärjestelmää. (Yoccoz 2020.) Fonttien vaihtamisen lisäksi lokalisoinnissa tulee ottaa huomioon eri kielten sanojen pituudet. Esimerkiksi saksan kieli on tunnettu pitkistä sanoistaan, ja tällöin pitää ottaa huomioon, miten pelin käyttöliittymä taipuu pidempien sanojen ympärille. (Duncan 2016) Lisähaastetta lokalisoinnin suunnitteluun tuo, mikäli pelin kääntämistä suunnitellaan arabialaiselle kirjaimistolle. Arabialaista kirjaimistoa käytetään muun muassa arabian ja persian kielten kirjoittamiseen. Arabialaisen kirjaimiston merkittävänä poikkeamana latinalaiseen kirjaimistoon kirjainten lisäksi on lukusuunta, joka latinalaisesta poiketen on oikealta vasemmalle. Tällöin käyttöliittymä on suunniteltava niin, että se soveltuu myös kielille, joita luetaan oikealta vasemmalle. Tämä tarkoittaa tekstin suunnan vaihtamisen lisäksi sitä, että käyttöliittymän lukusuunta on oikealta vasemmalle. (Zaiets 2020.)

Tärkeänä osana lokalisointia, etenkin hyötypeleissä on kohdeyleisön kulttuurin mukauttaminen peliin (engl. culturalization). Kulttuurillisia muuttujia ovat esimerkiksi erilainen

historia, nykyisyys, uskonto, etnisyys ja geopoliittiset tekijät. Mikäli näitä tekijöitä ei oteta huomioon, pahimmassa tapauksessa peli voidaan jopa estää tietyissä maissa hallituksen toimesta. (Texeira 2017.) Esimerkiksi Kiinaan vietävästä pelistä tulee poistaa pääkallot, veri ja uskonnollinen symboliikka. (Huang 2019; Liszewski 2019.)

4 TEKNOLOGIAT HYÖTYPELIEN TOTEUTTAMISESSA

4.1 Nuorten hyötypelien teknologisen toteuttamisen haasteet

Kuten opinnäytetyössä on aiemmin tullut ilmi, perustuu hyötypelien pedagoginen vaikutus vahvasti pelin luomaan immersioon ja yksinkertaisuuteen. Tällöin pelistä voidaan monesti karsia ylimääräiset elementit pois, jotka voisivat häiritä oppimista. Yksinkertaistettujen pelimekaniikkojen ja elementtien tekeminen helpottavat huomattavasti pelin kehittämistä teknologisesta näkökulmasta tarkasteltuna. Immersion luominen vaatii peliltä kuitenkin teknisiä ominaisuuksia, joihin tulee kehitysvaiheessa panostaa. Pienikin virhe pelissä saattaa katkaista pelaajan ajatuksenjuoksun. Tällaisia virheitä voivat olla esimerkiksi graafiset virheet eli ”glitchit” tai pahimmassa tapauksessa jopa koko pelin jumittavat tai edistymisen estävät virheet eli ”bugit”.

Älylaitteiden yleistymisen seurauksena mobiililaitteilla pelattavien hyötypelien määrä on kasvanut ja oppilaat voivatkin käyttää omia mobiililaitteitaan oppimistilanteen yhteydessä. Teknologisesti tämä asettaa monia haasteita. Pelkästään Android-käyttöjärjestelmää käyttäviä laitemalleja on ollut vuonna 2015 kymmeniätuhansia (Tung 2015). Lisäksi on lukuisia muita käyttöjärjestelmiä yleisempänä, Applen iOS. Näiden eri mallien käyttöjärjestelmäerojen lisäksi laitteissa on lukuisia muita eri ominaisuuksia, kuten muistin koko, kameroiden resoluutio ja määrä sekä hyötypelikehityksen kannalta suurin haaste, erikokoiset näytöt. Suurimpana ongelmana hyötypelien teknologisessa toteuttamisessa onkin saada peli näyttämään samalta kaikilla eri kuvasuhteilla sekä pikselitiheyksillä. Tämä tarkoittaa sitä, että jo pelin suunnitteluvaiheessa tulee päättää, miten peliobjektien piirtäminen halutaan pelissä toteuttaa, jotta kaikki pelaajat saisivat saman vaikutuksen pelistä riippumatta laitteesta. Esimerkiksi tablettitietokoneiden yleisin kuvasuhde on 4:3, kun taas mobiililaitteilla yleisin kuvasuhde on 16:9. Kuvasuhteella tarkoitetaan laitteen näytön korkeutta suhteessa sen leveyteen. Pikselitiheydellä puolestaan viitataan siihen, kuinka monta fyysistä pikseliä on tietyllä alueella. Useimmiten pikselitiheydestä käytetään yksikköä PPI (engl. pixels per inch) eli montako pikseliä mahtuu näytössä yhdelle neliötuumalle eli 6,45 neliösenttimetrille. (Screensiz 2020.)

Mikäli hyötypeli on tarkoitettu käyttäjien omilla mobiililaitteilla käytettäväksi, pitää toteutusvaiheessa ottaa myös huomioon eri laitteiden rajoitettu tallennuskapasiteetti. Vaikka tallennustilan kasvu on ollut nopeaa 2010-luvulla ja täten useimmissa laitteissa on

riittävästi tilaa isollekin pelille, tulee ottaa huomioon haluaako käyttäjä ladata suuria tiedostoja omalle laitteelleen viemään tilaa. Suuren pelin lataaminen kestää kauemmin kuin pienen ja suuren pelin lataaminen ei ole yhtä houkuttelevaa, kuin pienen pelin, jota pääsee nopeasti pelaamaan latauksen aloitettua. On havaittu, että sovelluksen koon kasvaessa jokaista kuutta megatavua kohden, asennusten määrä on laskenut prosentin. Syitä tähän saattaa olla latauksen kesto, internetyhteyden heikkous ja paljonko lataus vie liittymästä dataa ja täten maksaa. (Tolomei 2017.) Mobiililaitteiden haasteena on myös rajoitettu laskentateho. Uusimpien älypuhelimien prosessorien laskentateho riittää jopa vaativampaankin kolmiulotteiseen hahmontamiseen eli objektien luomiseen mallista. Kuitenkin hyötypeleissä, jotka suunnataan alustavasti opetuskäyttöön, on otettava huomioon, että käyttäjäkunnan puhelimien laskentateho saattaa vaihdella suuresti, koska puhelimia ei ole varta vasten ostettu pelaamistarkoituksessa. Mikäli pelin halutaan toimivan viiveettä myös heikompitehoisilla puhelimilla, on toteutuksessa otettava huomioon pelin optimointi, jolla voidaan parantaa pelin suorituskykyä. Opinnäytetyön kohdassa optimointi käydään läpi tarkemmin, miten pelin kokoa sekä pelin vaatimaa laskentatehoa voidaan rajoittaa tarvittaessa.

4.2 Peliteknologiat tarkastelussa

Peliteknologioilla tarkoitetaan teknologioita, jotka luovat raamit pelikehitystä varten pelikehittäjille. Näitä ovat esimerkiksi pelimoottorit, ohjelmistokehykset ja näihin luodut lisäosat, jotka helpottavat pelikehittäjien työtä. Pelimoottorilla tarkoitetaan kehitystyökaluna käytettävää ohjelmistoa tai ohjelmistokehystä (engl. framework), jonka tarkoituksena on helpottaa pelikehitystä tuomalla kehittäjälle tietyt pelien vaatimat perustoiminnallisuudet, kuten fysiikkamoottorin, grafiikan hahmontamisen, animaatioiden hallinnan sekä käyttäjän syötteen tulkinnan. (Enger 2013.) Ohjelmistokehys ei itsessään ole pelimoottori vaan alusta, joka pelimoottorin tavoin sisältää ominaisuuksia, jotka helpottavat kehittämistä, mutta eivät ole suoraan suunniteltu pelikehitystä ajatellen (TechTerms 2013). Tässä alaluvussa tarkastellaan suosituimpien pelimoottoreiden, Unityn ja Unreal Enginen, tarjoamia ominaisuuksia visuaalisten novellien tekoon (GameDesigning 2020b). Pelimoottorien ominaisuuksien lisäksi alaluvussa käydään läpi, mitä hyötyä selainpohjaisuudesta olisi hyötypelin toteuttamisessa verrattuna aiemmin mainittuihin pelimoottoreihin.

4.2.1 Unity3D vastaan Unreal Engine

Unity3D on Unity Technologiesin kehittämä kehitystyökalu, jolla pystytään luomaan järjestämäriippumattomia (engl. cross-platform) pelejä ja sovelluksia (Unity Technologies 2020a). Unreal Engine on Epic Gamesin kehittämä työkalu pelikehitykseen, teollisuuden visualisointiin ja simulaatioiden toteuttamiseen (Epic Games 2020a).

Vaikka Unity ja Unreal Engine molemmat sopivat erinomaisesti pelinkehitykseen, on niissä kuitenkin huomattavasti eroja. Vaikka molemmat työkalut ovat ilmaisia käyttää, on molempien työkalujen käytössä oma maksunsa, jos pelin tuotot ylittävät tietyn rajan. Jos Unityllä tehty peli tuottaa vuodessa yli 200 000 dollaria, on kehittäjän maksettava 150 dollaria Unity Technologiesille Pro -lisenssistä (Unity Technologies 2020b). Unreal Enginen kohdalla, Epic Games vaatii viiden prosentin rojalteja kehittäjältä, jos pelin kvartaalituotot ylittävät 3 000 dollaria (Epic Games 2020b). 2D-ominaisuuksien kohdalla, molemmat Unity sekä Unreal Engine tarjoavat tuen 2D-komponenteille. Vaikka Unreal Enginessä on Paper2D-järjestelmä, joka tukee 2D-pelien toteuttamista, on Epic Games lopettanut kyseisen järjestelmän tukemisen. Puuttuvan tuen lisäksi järjestelmä on melko raskas etenkin mobiililaitteilla. (Reddit 2018.) Kehittäessä mobiilipeliä tulee pelin olla hyvin optimoitu ja kevyt suorittaa. Unityn tarjoamat ominaisuudet pelin optimointiin ovat huomattavasti paremmat, kuin Unreal Enginen ja tällöin se soveltuu paremmin mobiilikehitykseen.

Visual novel -tyylistä peliä suunnitellessa tulee ottaa huomioon pelimoottorin tarjoamat valmiit kehitystyökalut. Unityn Asset Store tarjoaa useita työkaluja visuaalisten novellien kehittämiseen (Unity Technologies 2020c). Unreal Enginen kohdalla työkaluja voidaan asentaa Marketplacesta (Epic Games 2020c). Unreal Enginen kohdalla valmiiden työkalujen määrä on huomattavasti rajoitetumpi verrattuna Unityn Asset Storen tarjoamiin työkaluihin. Lisäksi Marketplacen tarjoamat lisäosat ovat usein maksullisia, kun taas Asset Storesta on saatavilla useita ilmaisia työkaluja eri käyttötarkoituksiin.

4.2.2 Pelimoottori vastaan selainpohjaisuus

Selainpohjaisessa pelissä peli käyttää alustanaan käyttäjän laitteen selainta ja hakee vaadittavan datan internetistä pelin aloittamisen tai pelaamisen yhteydessä. Selainpohjaiset pelit eivät välttämättä vaadi käyttäjää lataamaan mitään omalle laitteelleen, jolloin

vaadittava tallennustila on hyvin pieni. Selainpohjaisten pelien kehityksessä voidaan myös käyttää siihen tarkoitettuja pelimoottoreita. Selainpohjaisista pelimoottoreista käytetään usein ’pelimoottori’ nimityksen sijaan nimitystä ’ohjelmistokehys’, sillä ne monesti antavat kehittäjälle käyttöön vain kehyksen, jonka avulla pelikehitys nopeutuu ja helpottuu. Selainpohjaisten pelien rakentamiseen suunnitellut ohjelmistokehykset voivat olla samankaltaisia tavalliseen työpöytäkäyttöön suunniteltujen pelimoottereiden kanssa, mutta ovat monesti huomattavasti alkeellisempia ja sisältävät vain murto-osan ominaisuuksista joita esimerkiksi Unity tarjoaa työpöytä ja mobiilipelien rakentamisessa.

Selainpelien kehitykseen luotujen ohjelmistokehysten etuna Unityn kaltaiseen pelimoottoriin löytyy niiden alkeellisuudesta. Koska kyseiset ohjelmistokehykset pitävät sisällään usein huomattavasti vähemmän ominaisuuksia verrattuna monipuolisiin pelimoottoreihin, on niiden käyttäminen usein kevyempää ja vaatii vähemmän resursseja päätelaitteelta. Vaikka selainpohjaisen pelin käyttöönotto ei vaadi erillistä pelin lataamista sovelluskaupasta, vaatii se silti verkkoyhteyden, jotta sivustolle, jolla peli sijaitsee, päästään päätelaitteella. Käyttäjä hyötyy tästä siten, ettei joudu lataamaan koko peliä päätelaitteeseensa, jolloin se ei myöskään vie tallennustilaa. Haittana selainpohjaisuudessa ilmenee se, että verkkoyhteys tarvitaan aina, kun peliä halutaan pelata. Tästä syystä pelin kehitysvaiheessa tulee ottaa selvää, mitkä tekijät vaikuttavat kyseisen pelin julkaisuympäristössä ja niiden perusteella tehdä valintoja pelimoottorilla tuotetun ”standalone applicationin” ja selainpohjaisuuden välillä. Standalone applicationilla tarkoitetaan peliä, joka ei vaadi muuta kuin itsensä toimiakseen. Esimerkkejä tällaisista pelimoottoreista ovat esimerkiksi Phaser ja PhysicsJS. (Phaser 2020) (PhysicsJS 2020.) Phaser on Photon Stormin kehittämä ohjelmistokehys, joka on tehty 2D pelien luomiseen selainympäristössä.

	UNITY	UNREAL ENGINE	PHASER
ILMAINEN	Kyllä	Kyllä	Kyllä
2D OMINAISUUDET	Kattavat	Paper2D ei ole enää aktiivisesti tuettu	Kattavat
MOBIILITUKI	Optimoitavissa mobiililaitteille	Pelit vaativat paljon resursseja laitteelta.	Optimoitavissa mobiililaitteille
VISUAL NOVEL TYÖKALUT	Runsaasti ilmaisia työkaluja Asset Storesta	Muutamia työkaluja marketplacessa, suurin osa maksullisia	Muutamia työkaluja
DATAN KÄYTTÖ	Ladatessa	Ladatessa	Avatessa

Kuva 6. Unityn, Unreal Enginen ja selainpohjaisen Phaserin ominaisuuksien vertailu.

Pelimoottoreiden selvänä hyötynä selainpohjaisuuteen verrattuna onkin niiden sisältämät perustoiminnallisuudet, jotka on suunniteltu ajatellen pelikehitystä. Kun peli toteutetaan Unityllä, saadaan tuotettua niin sanottu standalone application tyylinen peli. Pelin asennustiedoston ladattua pelin ei tarvitse olla yhteydessä enää internettiin tai saada dataa mistään muualta kuin asennuksen mukana tulleista tiedostoista. Selainpohjaiset pelit taas tukeutuvat siihen, että pelaaja on jatkuvasti yhteydessä internettiin omalla laitteellaan.

4.2.3 Visual novel -lisäosan käyttö Unityssä

Koska Unity tarjoaa laajan valikoiman lisäosia, joita voi ladata Asset Storesta, on monessa tapauksessa viisasta tarkastella, mikäli kehityksessä oleva projekti pystyy hyödyntämään jotain niistä. Sen sijaan, että kaikki ominaisuudet kehitettäisiin itse alusta alkaen, voidaan projektin kannalta tärkeimpiä osia etsiä Asset Storesta. Tämä johtuu siitä, että vaikka lisäosat olisivatkin maksullisia, monesti niiden tarjoamien ominaisuuksien luominen alusta alkaen Unityssä veisi enemmän resursseja verrattuna lisäosan hintaan. Ilmaisia visuaalisten novellien tekoon tarkoitettuja lisäosia tarkastellessa Unityn Asset Storessa nousee kaksi lisäosaa muiden yli: Visual Novel Toolkit Free (Sol-tribe 2014) ja

Fungus (Fungus Games 2020). Vertaillessa näitä kahta työkalua, havaittiin, että Visual Novel Toolkit Free on päivitetty viimeksi vuonna 2014, kun taas Funguksen kehitys on hyvinkin aktiivista ja opinnäytetyön kirjoitushetkellä Funguksen viimeisin päivitys on suoritettu 9. tammikuuta. Tästä syystä tarkastelussa päädyttiin Funguksen käyttöön opinnäytetyön esimerkkiprojektissa. Funguksen valinta pohjautuu Juha Aarnion tekemään tutkimukseen aiheesta. (Aarnio 2019, 13-17)

Fungus tarjoaa laajan valikoiman visuaalisissa novelleissa käytettäviä ominaisuuksia ja työkaluja, kuten graafisen vuokaaviotyökalun pelin etenemisen suunnitteluun ja toteuttamiseen. Funguksen omat ominaisuudet minimoivat kehittäjien ohjelmoinnin tarpeen, sillä kaikki oleelliset visuaalisen novellin kehitykseen liittyvät ominaisuudet löytyvät helposti vuokaaviotyökalun asetuksista.

4.3 Optimointi

Tämän alaluvun tarkoituksena on avata sitä, mitä optimointi on ja miten sitä voidaan hyödyntää Unityllä toteutetussa pelissä.

Optimoinnilla tarkoitetaan pelin tai ohjelmiston muokkaamista muotoon, joka vie mahdollisimman vähän laskentatehoa laitteesta (Khumansinh 2015). Ohjelmistokehityksen yhteydessä optimoinnista puhuttaessa tarkoitetaan monesti komentosarjojen optimointia, kun taas pelikehityksessä grafiikan hahmontamisen, fysiikkaelementtien ja tallennustilan optimoinnin merkitys kasvaa.

Usein eniten optimointia vaativat pelin graafiset osat, jotka vaativat tehoja laitteen grafiikkasuorittimelta (engl. graphics processing unit) ja laitteen keskussuorittimelta (engl. central processing unit). Koska grafiikkasuorittimen ja keskussuorittimen optimointi poikkeaa toisistaan merkittävästi, tulee grafiikan optimoinnissa selvittää, mitkä ominaisuudet vievät laskentatehoa laitteelta. Monesti keskussuorittimen optimoinnilla saatetaan tarkoittaa sitä, että keskussuorittimen työtä annetaan grafiikkasuorittimen tehtäväksi ja toisinpäin. (Unity Technologies 2020d.)

4.3.1 Optimoinnin hyödyt

Käyttäjän näkökulmasta optimoitu peli erottuu monesti sillä, että pelin suorituskyky paranee. Tällä tarkoitetaan, että pelin yleinen FPS (engl. frames per second) eli

ruudunpäivitystaajuus paranee etenkin laskentatehollisesti heikommilla laitteilla. Nykypeleissä pyritään tavoittelemaan 60FPS rajapyykkiä, joka on useimpien näyttöjen virkistystaajuus. Nopeatempoisissa ensimmäisen persoonan ammuntapeleissä on usein tarpeen saada pelin ruudunpäivitystaajuudeksi yli 144, joka on uusimpien pelinäyttöjen virkistystaajuus. Mikäli ruudunpäivitystaajuus tippuu alle 30, alkavat pelissä animaatiot sekä liike vaikuttamaan pätkiviltä ja pelaaminen alkaa tuntua tökkivältä. (Klappenbach 2020.)

Etenkin mobiilipelien maailmassa, jossa laitteiden tallennustila on hyvin rajallinen ja puhelinliittymien datakatot saattavat karkottaa pelaajat isojen pelien luota, on tallennustilan optimointi hyvin olennaista. Nykypeleissä on alkanut muodostua tavaksi, että pelien koko saattaa olla satoja gigatavuja. Tämä johtuu siitä, että pelit sisältävät yhä enemmän korkealaatuisia tekstuureja. (Walton 2019.) Mikäli pelaajan pitää valita lataako hän 50MB pelin vai vastaavan pelin, jonka vaatima tallennustila on 20MB, on pienempi peli huomattavasti houkuttelevammassa asemassa.

Hyvin optimoitu peli toimii paremmin myös heikommilla laitteilla, vie vähemmän tallennustilaa ja kuluttaa vähemmän akkua mobiililaitteilla käytettäessä. Näistä syistä optimointiin tulisi varata pelinkehitysprosessissa oma aikansa, sillä optimoidun ja optimoimattoman pelin erot saattavat olla merkittäviä alkaen jo siitä miten houkuttavalta ne näyttävät käyttäjälle lataustilanteessa.

4.3.2 Optimointi teknisellä tasolla

Tärkeimpänä työkaluna Unityssä pelin optimoimisen suhteen toimii Profiler-työkalu. Työkalun avulla pystytään helposti ja selkeästi erottelemaan mitkä pelin elementit vievät laskentatehoa laitteelta. Kuvassa 15 näkyy esimerkki Unityn Profiler -työkalusta, jolla tarkastellaan suoritinkäyttöä pelin alkuvalikon taustakuvan alfakanavan muutoksen aikana. (Unity Technologies 2020e.) Alfakanava (engl. Alpha channel) eli läpinäkyvyyskanava määrittelee värin tai kuvan läpinäkyvyyden (Techopedia 2012).

Keskussuorittimen käytön optimointi Unityssä keskittyy usein datan käsittelyyn. Tärkeintä keskussuorittimen käytön optimoinnin kannalta on käyttää harkiten Unityn Update -funktiota, joka on funktio, jonka pelimoottori suorittaa jokaisen näytön päivityksen yhteydessä. Update-funktion käytön rajoittamisen lisäksi tulisi tarkastella muita funktioita tai elementtejä, joita kutsutaan tai joiden ominaisuudet muuttuvat jatkuvasti. Tarkastelussa

tulee kiinnittää huomiota, onko mahdollista päivittää niitä harvemmin. Esimerkkejä näistä ovat esimerkiksi pelin tallentaminen tai tekoälyn reitinhaku. Grafiikkasuorittimen käytön optimointi liittyy enemmän pelin graafiseen optimointiin. Tärkeimpiä elementtejä grafiikkasuorittimen optimoinnissa ovat päälle piirtämisen (engl. overdraw) välttäminen sekä valaistuksen optimointi. Päälle piirtämisellä tarkoitetaan tapahtumaa, jossa sama pikseli piirretään useasi ruudunpäivityksen yhteydessä. Tämä tapahtuu, kun useampi objekti piirretään toistensa päälle. Valaistuksen optimoinnissa pyritään siihen, että mahdollisimman suuri osa valoista sekä varjoista olisivat staattisia, eli pelaaja ei pysty esimerkiksi omalla liikkeellään vaikuttamaan niihin. (Unity Technologies 2020d.)

Tallennustilaa optimoidessa etenkin mobiilipelien kohdalla voidaan monesti laskea tekstuureiden resoluutiota hyvinkin paljon johtuen mobiililaitteiden näyttöjen pienestä koosta ja suuresta pikselitiheydestä, jolloin matalampiresoluutioiset kuvat eivät laske pelin graafista ulkoasua merkittävästi. Mitä pienempiä pelin spritet ovat, sitä kevyempiä ne ovat pelimoottorille hahmontaa käyttäjän nähtäväksi. Tämän lisäksi spritejen resoluutio ja määrä on yksi suurimmista tallennustilan kuluttajista. (Unity Technologies 2020f.) Kuvassa 16 huomataan miten kuvan resoluution tiputtaminen pienentää huomattavasti kuvan vaatimaa tallennustilaa, joka on suoraan yhteydessä valmiin pelin vaatimaan tallennustilaan.

5 ESIMERKKIPROJEKTI PELIMO

PELIMO-peli on Opetushallituksen rahoittama hanke. PELIMOn tavoitteena on kehittää 10–12-vuotiaille lapsille suunnattu opetuspeli tukemaan monikulttuurisuuskasvatusta kouluissa. (PELIMO 2019.)

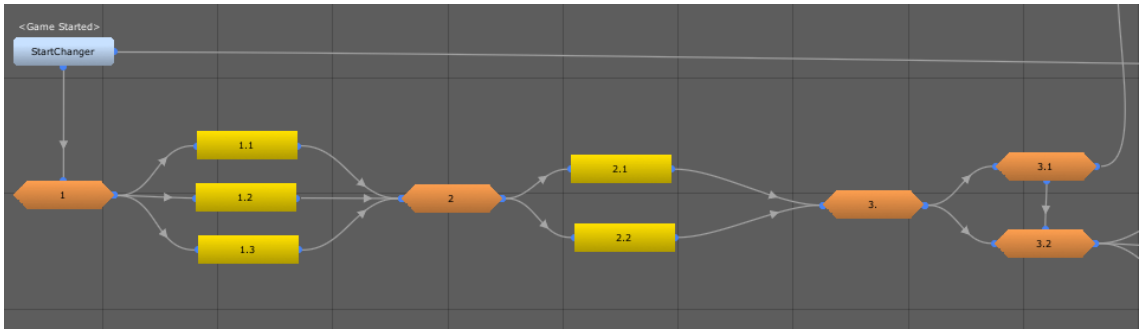
Esimerkkiprojektissa toteutetaan peli Unity-pelimoottorilla, käyttäen apuna Fungus -lisäosaa, joka helpottaa tarinankerronnan rakentamista peliin. Pelin mekaniikoissa pyritään intuitiiviseen pelitapaan ja hyödyntämään nuoren sukupolven diginatiivisuutta, jolloin tarinankerronta ja opetukselliset ominaisuudet saadaan helpommin pelaajan ymmärrettäväksi.

5.1 Pelin eteneminen

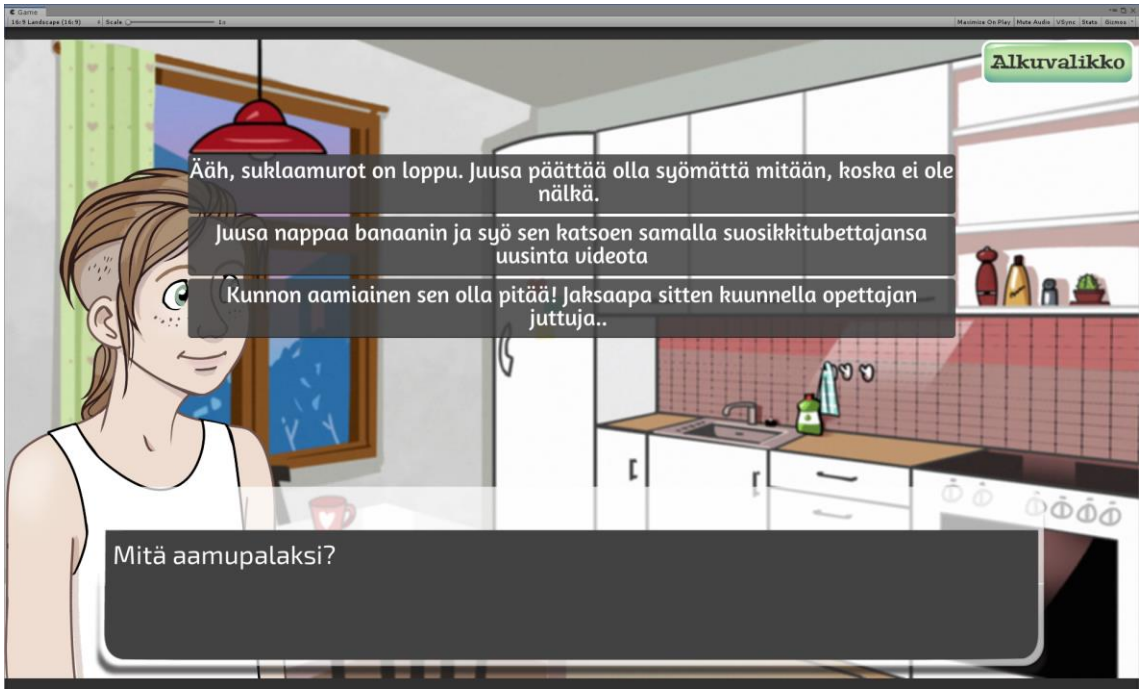
Peli tukeutuu vahvasti Fungus -lisäosaan, joka sisältää ominaisuuksia, jotka helpottavat visuaalisen novellin rakentamista Unity-pelimoottorilla. Visuaalisen novellin rakentaminen Fungusta käyttäen tapahtuu Funguksen vuokaaviotyökalulla, josta on esitetty esimerkki kuvassa 7. Pelin eteneminen määrittyy pelaajan päätöksien perusteella. Kuten kuvassa 8 näkyy, voi pelaaja kohdassa yksi valita kolmen eri vaihtoehdon välillä ja perustuen valittuun vaihtoehtoon, etenee peli sen mukaan. Tietyistä valinnoista peli pitää myös kirjaa pisteiden muodossa, jotka vaikuttavat lopulliseen palautteeseen.

Visuaalisen novellin lisäksi PELIMO sisältää useita minipelejä, jotka on toteutettu Funguksen ulkopuolella. Varsinainen visuaalinen novelli sekä minipelit saadaan yhdistettyä Unityn SceneManager -luokan avulla. Siirtymää minipelien ja visuaalisen novellin välillä pyritään pehmittämään animaatioilla ja häivytyksillä, jolloin pelissä saadaan tietty immersio pysymään myös minipelien välillä. Pelaajan suoritukset minipeleissä vaikuttavat tarinan lopputulokseen siten, että mikäli minipelin lopputulos on positiivinen niin

pelaaja saa tietyn määrän pisteitä varsinaiseen tarinaan. Mikäli lopputulos on negatiivinen, menettää pelaaja tietyn määrän pisteitä.



Kuva 7. Esimerkki Funguksen vuokaaviotyökalusta.



Kuva 8. Kuvan seitsemän valintanäkymä pelissä.

5.2 Pelin lokalisointi

PELIMO on suunnattu suomalaisille 10–12-vuotiaille lapsille, ja kielet, joita pelin tulee tukea, ovat suomi ja ruotsi. Koska peli on lähtökohtaisesti suunnattu vain suomalaisille, ei pelin suunnittelussa tai toteutuksessa ole tarpeen kiinnittää huomiota kulttuurillisiin eroihin eri maiden välillä. Pelin lokalisoinnissa on kuitenkin otettava huomioon eri alueiden tavat ja lasten ajattelumallit, koska Suomi on maantieteellisesti suhteellisen laaja

maa. Tällöin ei voida käyttää pelkästään esimerkiksi pääkaupunkiseudun lapsille tuttua terminologiaa vaan on pitäydyttävä enemmän yleisemmin ymmärrettävässä kielen tyylissä.

Varsinaisen visuaalisen novellin kääntämisestä vastaa Funguksen oma lokalisointi komentosarja, jonka toiminta perustuu CSV (comma-separated values) -tiedoston lukemiseen ja sen sisällön asettamiseen pelin dialogeihin sekä valintanäkymiin. Vaikka Fungus vastaa visuaalisen novellin kääntämisestä, niin suomeksi, kuin ruotsiksikin oli peliin luotava käännösjärjestelmä myös minipelejä sekä valikkoja varten. Minipelien sekä valikkojen kääntämiseen luotu käännösjärjestelmä käyttää melkein samaa lähestymistapaa kääntämiseen, kuin Funguksen oma käännösjärjestelmä, jotta kääntäjän ei tarvitse käyttää useita erilaisia ohjelmia käännösten teossa. CSV-tiedostoja voidaan muokata yleisimmillä tekstinkäsittelyyn soveltuvilla ohjelmilla kuten Notepadilla. CSV-tiedostossa samalla rivillä olevia sanoja voidaan erotella erilaisilla erottimilla, kuten pilkuilla, puolipisteillä ja sarkaimella. Johtuen siitä, että kenttien arvot voivat joissain tapauksissa sisältää myös pilkkuja, on tässä tapauksessa päädytty käyttämään puolipisteitä arvojen erottamiseen toisistaan. Kuvassa 9 näkyy CSV-tiedosto avattuna Visual Studio Codessa. Kuitenkin, jotta tiedoston rakennetta olisi mahdollisimman helppo ymmärtää, voidaan tiedosto avata taulukko-ohjelmalla, kuten Microsoft Excelillä. Kuten kuvassa 9 näkyy, muokkaa Excel CSV-tiedoston helppolukuiseksi, jolloin kääntäjän työ helpottuu, sillä tekstielementtien tunnisteet ovat helposti erotettavissa sanan eri versioista eri kielillä.

```
1  ID;Finnish;Swedish
2  1;Aloita;Börja
3  2;Asetukset;Inställningar
4  3;Lopeta;Utfart
5  4;Neljä;Fyra
6  5;Viisi;Fem
7  6;Kuusi;Sex
8
```

	1	2	3	4
1	ID	Finnish	Swedish	
2	1	Aloita	Börja	
3	2	Asetukset	Inställningar	
4	3	Lopeta	Utfart	
5	4	Neljä	Fyra	
6	5	Viisi	Fem	
7	6	Kuusi	Sex	

Kuva 9. Puolipilkkujen käyttö CSV-tiedostossa erottimena. Sarakkeessa yksi on sanan tunniste, sarakkeessa kaksi on sanan suomenkielinen versio ja sarakkeessa kolme sanan ruotsinkielinen versio.

Kuten edellä mainittiin, tapahtuu minipelien kääntäminen erillään muun pelin kääntämisestä. Minipelien käännösjärjestelmää suunnitellessa tuli ottaa huomioon se, miten saadaan luotua mahdollisimman tehokas käännösjärjestelmä kaikkien näkökulmasta. Jotta riskejä mahdollisista sanojen joutumisista vääriin paikkoihin pelin sisällä voitaisiin

vähentää, tuli käännöksiä varten valitun tiedostomuodon olla kääntäjän näkökulmasta helppolukuista, mutta myös suoraan tulkittavissa pelin komentosarjojen sisällä. Kuten kuvassa 9 näkyy, on CSV-tiedostomuoto Excelin kautta avattuna hyvin selkeä tulkita kääntäjän päässä ja sen käsittely Unityssä sellaisenaan on myös helppoa.

Unityn sisällä minipelien käännösjärjestelmä koostuu kahdesta eri komentosarjatiedostosta, MainLocalization.cs ja LocalizationElement.cs tiedostoista. Mainlocalization sisältää CSV-tiedoston lukemiseen liittyvät komentosarjat. Kuvassa 10 näkyy, kuinka Mainocalization tiedostossa `Start()` funktion sisällä luetaan CSV-tiedoston sisältö, jonka jälkeen sen sisältö pilkotaan riveittäin tekstijonotaulukkoon. Esimerkiksi kuvan 9 CSV-tiedoston `splitLine[0]` olisi ID;Finnish;Swedish.

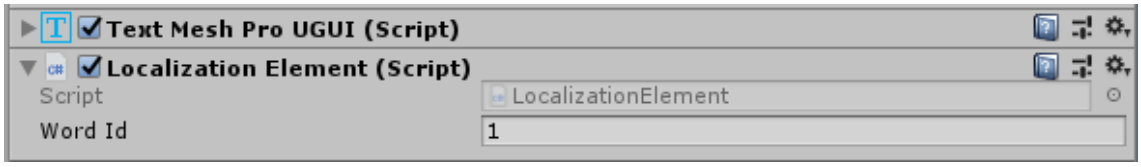
```
void Start()
{
    string csvFileContent = csvFile.text.ToString(); //Reads the contents of the csv file
    string[] fLines = Regex.Split(csvFileContent, "\n"); //Splits the contents of the csv file into different strings after each new line

    for (int i = 0; i < fLines.Length; i++) //Saves the contents of split lines into ArrayList()
    {
        string valueLine = fLines[i];
        individualStrings.Add(valueLine);
    }
    loaded = true; //Sets loaded to trues after adding all strings to the list. This allows LocalizationElement to set words on correct places
}
```

Kuva 10. Mainlocalizationin komentosarja CSV-tiedoston lukemiseen.

Unityn sisällä tekstin hahmontamisesta vastaa TextMeshProUGUI-komponentti, ja jokaisella peliobjektilla, joka sisältää kyseisen komponentin on myös oma LocalizationElement komponentti. Jokaiselle LocalizationElement-komponentille on määrätty oma tunniste, jonka perusteella pelin käynnistyessä komentosarja hakee MainLocalizationista tunnistetta vastaavan sanan. Kuten kuvassa 11 näkyy, on Unityn inspectorissa LocalizationElementin tunniste nähtävissä ja muokattavissa helposti. Inspector on osa Unityn käyttöliittymää, jossa peliobjekteille määritetyt ominaisuudet ja asetukset näkyvät. Tämä mahdollistaa sen, että uusien tekstielementtien lisäämineen peliin on vaivatonta, sillä kehittäjän ei tarvitse muokata koodia uusien tekstielementtien kohdalla. Ennen sanan palautusta kuvan 11 komentosarja tarkistaa vielä mikä on tällä hetkellä kielivalintana. Kuten kuvassa 9 näkyy, on suomenkielinen sana sarakkeessa kaksi ja ruotsinkielinen sana sarakkeessa kolme. Esimerkiksi, mikäli pelin käynnistyessä CSV-tiedostona käytettäisiin kuvan 9 CSV-tiedostoa, palauttaisi kuvan 11 LocalizationElement saman peliobjektin komponenttiin TextMeshProUGUI sanan ”Aloita”, mikäli `languageId` olisi yksi ja sanan ”Börja”, jos `languageId` olisi kaksi. Kuvassa 14 näkyy kyseinen tapahtumasarja graafisesti esitettynä kulkukaavion muodossa. Käännösjärjestelmä on suunniteltu siten, että käyttäjä pystyy pelin alkuvalikon kautta vaihtamaan pelin kieltä aina ilman, että

peliä edes tulee käynnistää uudelleen. Tämä mahdollistaa esimerkiksi uuden kielen opiskelun tai vahvistamisen niin haluttaessa.

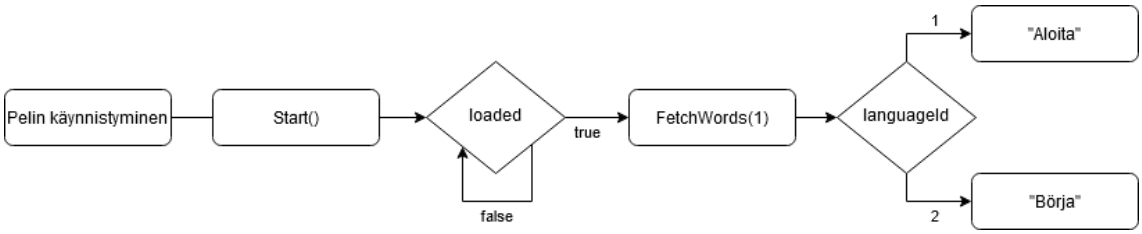


Kuva 11. Word Id:n määrittely LocalizationElement-komponentille Unityn Inspectorissa.

Vaikka teoriassa käännösjärjestelmän toiminnan pitäisi olla aukotonta, on TextMeshProUGUI-komponenteille annettu oletusarvona kyseisen tekstielementin sanan suomenkielinen versio siltä varalta, että käännösjärjestelmän toiminnassa ilmenee virheitä. Testilaitteilla testattaessa käännökset tapahtuivat myös heikommilla laitteilla välittömästi ja käyttäjän huomaamatta.

```
public string FetchWords(int wordId) //Splits rows into words and returns them to LocalizationElement.cs
{
    string[] requiredValue = Regex.Split ((individualStrings[wordId].ToString()),";");
    return requiredValue[languageId];
}
```

Kuva 12. FetchWords-funktio MainLocalization-tiedostossa.



Kuva 13. Käännösfunktioiden logiikka kulkukaaviona, kun FetchWords-funktion parametrina on yksi.

5.2.1 Reaaliaikainen käännösjärjestelmä

Miten sitten käytännössä tätä prosessia pystyttäisiin helpottamaan kääntäjän kannalta? Jotta kääntäjä ei joutuisi useaan otteeseen pelaamaan peliä läpi perehtyäkseen pelin juoneen, voidaan käännöksien lisääminen toteuttaa reaaliaikaisesti. Tällä tarkoitetaan sitä, että kääntäjä voi peliä pelatessaan kääntää kyseistä peliä edetessään siinä. Näin voidaan varmistua siitä, että kääntäjä ei joudu erikseen käyttämään aikaa pelin pelaamiseen, jonka jälkeen aloittaisi kääntämisen. Kyseisestä prosessista on hyötyä monella eri

saralla. Merkittävimpänä hyötynä on kääntämisen vaatima aika, joka pienenee huomattavasti pelikertojen vähenemisen myötä. Tämä korreloi suoraan projektin kustannuksiin vähentämällä kääntämiseen käytettävien resurssien määrää. Toisena merkittävänä hyötynä on kääntäjän kyky nähdä miten käännetyt tekstit sopivat osaksi käyttöliittymää. Kuten aiemmin mainittu, voi joidenkin kielien kohdalla sanojen pituudessa tulla haasteita. Mikäli kääntäjä pystyy reaaliaikaisesti seuraamaan käännösten ulkoasua pelin sisällä, voi hän tehdä kompromisseja, mikäli käännökset venyvät liian pitkiksi. Tällöin voidaan käyttää vastaavia lauseita, jotka sopivat paremmin käyttöliittymään. Tällaisesta tilanteessa, jossa kääntäjä voi itse tulkita käännösten toimivuutta käännösvaiheen aikana on hyötyä myös kehitystiimin jäsenille. Poiketen normaalista tavasta, jossa käännökset lähetetään kehitystiimille, jotka joutuvat kokoamaan uuden version pelistä, voidaan pitäytyä samassa pelin versiossa pidempään. Lisäksi kun poistetaan tarve viestiä muutoksista kehitystiimin ja kääntäjien välillä, säästetään huomattavasti aikaa, joka olisi ennen mennyt esimerkiksi sähköpostiviestien vaihtamiseen.

Jotta reaaliaikainen käännösjärjestelmä pystyttäisiin toteuttamaan, vaatii se toimiakseen seuraavat asiat:

1. Verkkoyhteyden omaava laite, jolla voidaan muokata käännöstiedostoja.
2. Tietokanta tai palvelu, jonne käännöstiedostot voidaan tallentaa ja josta ne voidaan noutaa.
3. Testilaite, jolle käännettävä peli on asennettu sekä toimiva verkkoyhteys.

Tässä työssä käytin käännöstiedostojen muokkaamiseen tietokonetta ja älypuhelinta. Käännöstiedostojen tallennuspaikaksi valikoitui Googlen tarjoama Google Sheets. Tämä johtuu siitä, että Google Sheets tarjoaa laajasti tunnetuksi tulleita ominaisuuksia Microsoft Excelistä ja sen käyttö on helppoa sekä ilmaista. Google Sheetsin avulla tuotettu ratkaisu ei myöskään tarvitse erillistä tietokantaa, josta tieto haetaan. Käännösjärjestelmän toimivuutta testattiin Unityn sisällä, Windows tietokoneelle kootussa valmiissa pelissä sekä Android laitteelle kootussa pelissä.

Reaaliaikainen käännösjärjestelmä pohjautuu siihen, että kääntäjä pysty edellä mainittua taulukkorakennetta hyödyntäen kirjaamaan käännöksiä Google Sheets -palveluun, josta peli hakee ne reaaliaikaisesti. Kyseisen järjestelmän haittapuolena on se, että se vaatii jatkuvan verkkoyhteyden päätelaitteen ja Google Sheets -palvelun välillä. Tämä aiheuttaa sen, että mikäli kääntäjällä ei ole jostain syystä mahdollisuutta toimivaan verkkoyhteyteen, ei kyseistä käännösjärjestelmää voida käyttää luotettavasti. Tästä syystä edellä

mainittu käännösjärjestelmä on pelin julkaisuversiota ajatellen parempi vaihtoehto. Tulee ottaa huomioon, että kyseinen käännösjärjestelmä on suunniteltu vain kehitysvaihetta varten johtuen sen verkkoyhteysvaatimuksesta. Tämä johtuu siitä, että nuorten hyötypeleissä pitää välttää ylimääräisten muuttujien, tässä tapauksessa verkkoyhteyden käyttöä, kuten työn teoriaosuudessa käy ilmi.

Tekniseltä toteutukselta reaaliaikainen käännösjärjestelmä on hyvin samankaltainen verrattuna edeltäjäänsä. Ainoana erona on, että käännöstiedostoa ei haeta laitteen lokaaleista tiedostoista, vaan Google Sheets -palvelusta. Google mahdollistaa automatisoidun tiedostojen lataamisen Sheets -palvelusta hyvin yksinkertaisesti. Google Sheets osoiterivit koostuvat kolmesta osasta. Tässä työssä olemme kiinnostuneita kahdesta viimeisestä osasta, dokumentin tunnisteesta ja parametri osasta. Dokumentin tunnisteen perusteella tiedetään mitä dokumenttia halutaan käsitellä ja parametriosa määrittelee sen käsittelytavan. Esimerkiksi jaettaessa osoiterivi `"https://docs.google.com/spreadsheets/d/12345678/export?format=tsv"` osiin, saadaan seuraavanlaiset osat:

1. `https://docs.google.com/spreadsheets/d/`
2. `12345678`
3. `/export?format=tsv`

Esimerkin tapauksessa haetaan dokumentista, jonka tunniste on 12345678 CSV-tiedostoa muistuttava tiedosto. Parametriosa määrittelee, että halutaan hakea sisältö kyseisestä osoitteesta (export) tiedostomuodossa (format) TSV (tab-separated values).

Käännösjärjestelmän suunnittelun yhteydessä päädyttiin puolipilkkujen käyttöön rivierottimina, mutta Google Sheets ei tue kyseistä tapaa käsitellä CSV-tiedostoja. Tämän takia joudutaan vaihtamaan rivierotin puolipilkusta tabulaattoriksi. Teknisesti tämä ei aiheuta ongelmia, sillä käännösten ei lähtökohtaisesti kuuluisi pitää sisällään rivinvaihtoja. Kuitenkin komentosarjan puolella joudutaan rivierottimet käsittelemään tabulaattoreina puolipilkkujen sijaan.

Kun tarvittavat muutokset käännösjärjestelmään on tehty, eli vaihdettu käännöstiedoston sijainti Google Sheets osoitteeksi oikeilla parametreilla, hakee peli aina käynnistyessään uudet käännökset Google Sheets -palvelusta. Koska tämä ei kuitenkaan reaaliaikaisesti päivitä käännöksiä, täytyy muutoksia tekevää funktiota kutsua pelin sisällä

säännöllisesti. Tämän voi toteuttaa luomalla säännöllisin väliajoin kutsuttavan funktion, joka vastaa tekstitiedoston päivittämisestä.

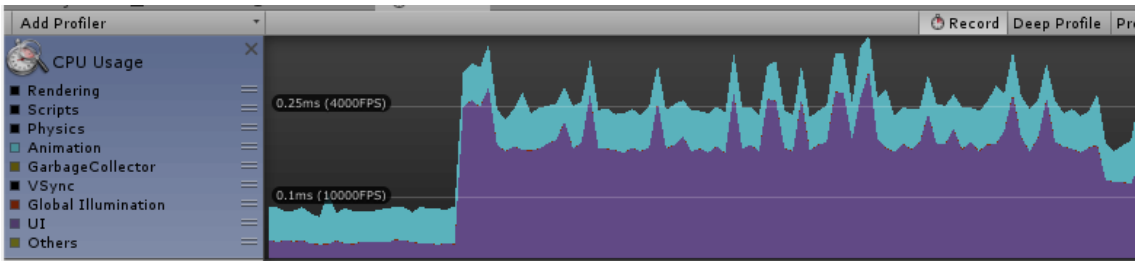
5.3 Pelin optimointi

Tässä alaluvussa käydään läpi, miten PELIMOn optimointi on toteutettu. Optimoinnissa keskityttiin kolmeen eri osa-alueeseen, suoritinkäytön, muistin sekä tallennustilan optimointiin. Pelin optimointiin käytettiin Unityn omaa Profiler-työkalua, jonka avulla pystyy paikallistamaan skene (engl. scene) kohtaisesti mitkä komponentit vievät eniten laskentatehoa tai muistia ja täten voitiin tehdä tarvittavia toimenpiteitä, jotta peli saataisiin kevyemmäksi. Skeneillä tarkoitetaan Unityn sisällä olevia yksittäisiä näyttämöitä, jotka sisältävät osan pelistä. PELIMOn tapauksessa jokainen visuaalisen novellin tarina ja minipeli ovat oma skenensä.

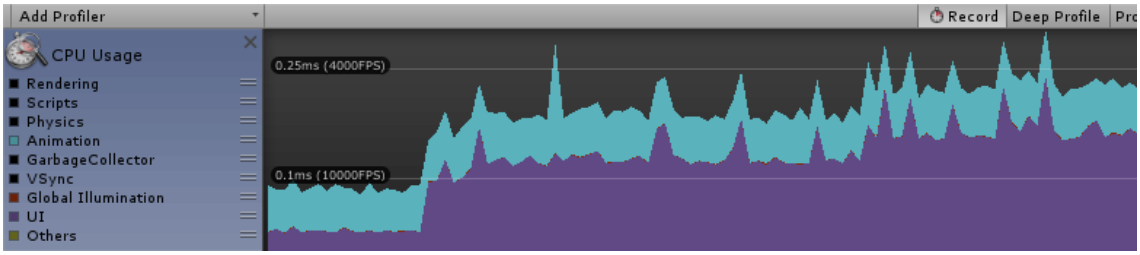
5.3.1 Suoritinkäytön optimointi

Koska PELIMO on kaksiulotteinen peli eikä sisällä fysiikkaelementtejä, on peli itsessään suhteellisen kevyt suoritettava myös vähemmän laskentatehoa omaavilla laitteilla. Vaikka peli ei yleisesti ottaen vaadi juuri lainkaan laskentatehoa, on kuitenkin huomioitava tietyt pelin ominaisuuden, jotka saattavat vaatia yllättävänkin paljon tehoa. Pelin alkuvalikon taustan liike on toteutettu animoimalla Canvakselle piirrettyjä Image-komponentteja ja jossa kuvan vaihtuminen on toteutettu laskemalla nykyisen kuvan alfakanavaa sekä nostamalla seuraavan kuvan alfakanavaa. Koska Image-komponenttien alfakanavan vaihtaminen vaatii suhteessa muuhun kohtuullisen paljon tehoja, huomataan kuvassa 14 miten käyttöliittymän (UI) vaatima suoritinteho kasvaa aina alfakanavia muutettaessa. Kuten kuvassa 14 näkyy graafin vasemmassa reunassa, ei hahmontamisen taajuuden heikkeneminen 0,1ms:sta 0,25ms:iin vaikuta juurikaan tehokkaan suorittimen omaavalla tietokoneella pelatessa. Kuitenkin etenkin vanhemman sukupolven mobiililaitteilla äkillinen vaaditun tehon kasvu saattaa vaikuttaa yllättävänkin paljon. Heikoimmilla laitteilla muutos saattaa aiheuttaa jopa näkyvää pätkimistä animaatiossa. Vaadittua laskentatehoa pienentää hieman se, että kuvien alfakanavia ei muuteta muissa kuin sitä vaativissa kohdissa. Toinen vaihtoehto on kuvien resoluution pienentäminen. Kuvassa 15 voidaan nähdä miltä alfakanavan muutos näyttää, kun kuvien kokoa pudotettiin. Kuvien alfakanavan muuttamisen sijaan voitaisiin myös käyttää erillistä kuvakokoelmaa

(engl. sprite sheet), joka sisältää versioita samasta kuvasta eri alfakanavalla. Näiden kuvien läpikäyminen nopeasti luo lähes samanlaisen efektin, joka on saavutettavissa Unityn animaatioilla alfakanavan muokkaamisessa. Tämä kuitenkin vaatisi lisää tallennustilaa peliltä ja jotta saavutettu animaatio ei vaikuttaisi tökkivältä tulisi kuvia olla suuri määrä. Vaikka Unityn dokumentaatioissa ”power of two” -säännön todetaan vaikuttavan laskentatehoon merkittävästi, ei PELIMOn tapauksessa havaittu merkittävää muutosta suoritinkäytössä kyseistä menetelmää käyttäen. Power of two tarkoittaa sitä, että kuvien korkeus ja leveys ovat luvun kaksi potensseja.



Kuva 14. Suoritinkäytön nousu Image-komponentin alfakanavaa muutettaessa.



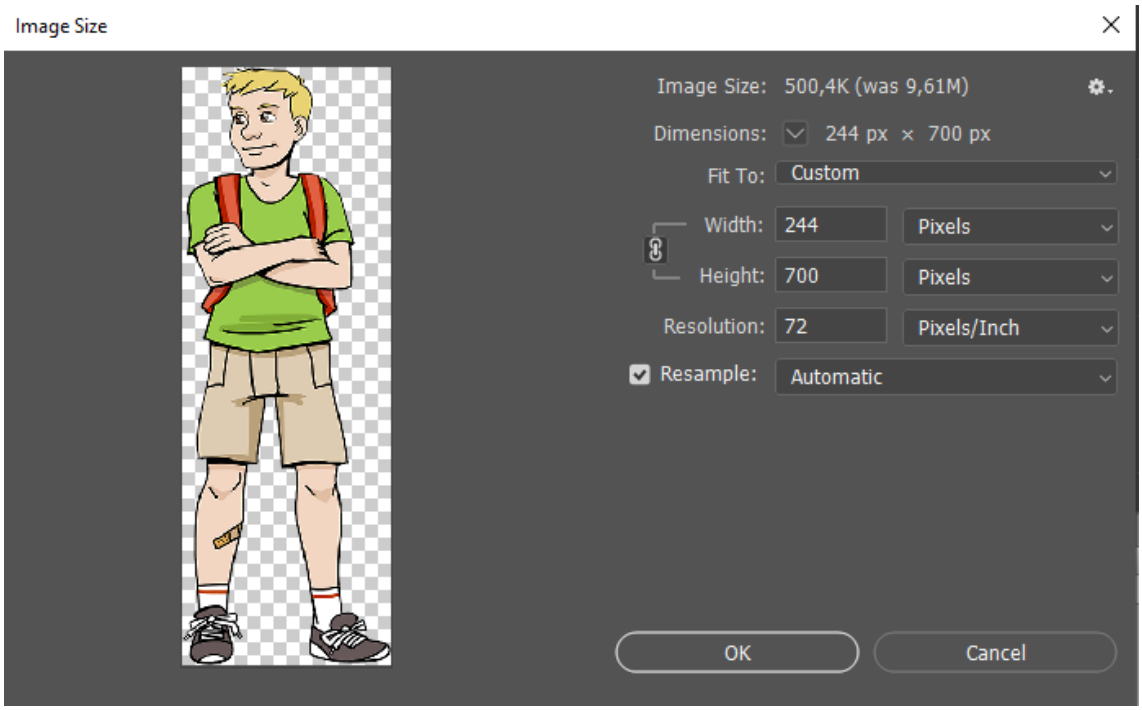
Kuva 15. Suoritinkäyttö kuvien resoluution pienentämisen jälkeen.

Esimerkiksi alkuvalikon hahmokuvien alkuperäinen koko oli 1083x3103 pikseliä (leveys kertaa korkeus) ja muokkauksen jälkeiseksi kooksi tuli 244x700 pikseliä. Kuvan 16 yläreunan ”Image Size:” kohdassa näkyy, miten kuvan koko pienenee huomattavasti kuvan resoluution pienentämisen seurauksena. Tehokkain ratkaisu olisi käyttää taustakuvien kohdalla SpriteRenderer-komponentteja, joiden tarkoituksena on piirtää 2D-kuvia ilman raskasta Canvas-komponenttia, joka yleensä on vastuussa Unityllä tuotettujen pelien käyttöliittymän hahmontamisesta. (Unity Technologies 2020g) (Unity Technologies, 2020h). Lisäksi niiden animoiminen onkin kevyempää. Koska peli pitää kuitenkin saada skaalautumaan eri kokoisille näytöille, on Canvaksen käyttäminen teknisesti helpompi ratkaisu. Kuvien koon pienentämisen lisäksi Unityn sisällä voidaan vaihtaa kuvien

tuontiasetuksia, jolloin pystytään kompressoimaan kuvia. Kuvien kompressoiminen vaikuttaa tallennustilan pienentymisen lisäksi myös suoritinkäytön ja muistinkäytön paranemiseen.

PELIMOn minipelit sekä alkuvalikko pohjautuvat vahvasti Unityn Canvas-komponenttien käyttöön. Canvas-komponentit mahdollistavat helpon skaalautumisen eri laitteille. Johtuen Unityn tavasta hahmontaa elementit Canvakselle, voi sen käyttö ajoittain viedä runsaasti resursseja laitteelta. Esimerkki tällaisesta on se, mikäli Canvas-komponentille määritellään runsaasti ”child objekteja” eli objekteja, jotka hierarkiassa kuuluvat Canvakseen. Tältä pystytään välttymään jakamalla pelinäkymä useaan eri Canvakseen. Yhteen Canvakseen tulee tällöin sisällyttää elementit, jotka yleensä käytössä näkyvät samanaikaisesti käyttäjälle. Tällaisia ovat esimerkiksi eri valikoiden sijoittaminen omille Canvaksille.

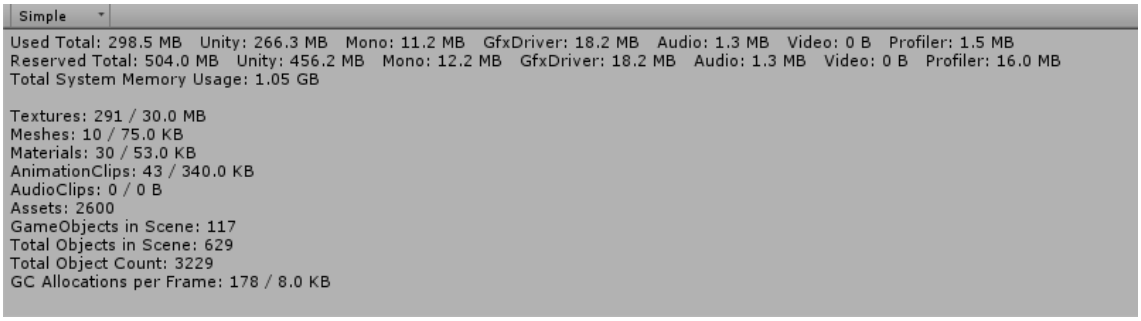
Komentosarjojen viemää laskentatehoa tutkiessa todettiin, että niiden viemät resurssit olivat lähes olemattomat, eivätkä täten vaatineet lisätoimenpiteitä.



Kuva 16. Adobe Photoshopin kuvan koon muokkauksen näkymä.

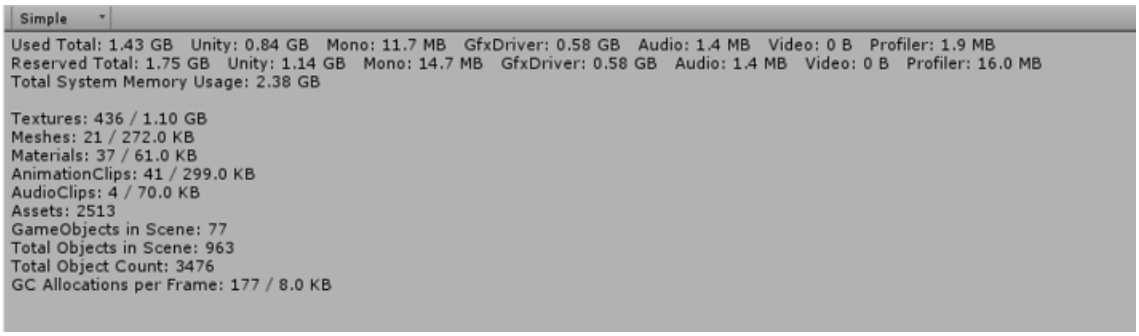
5.3.2 Muistinkäytön optimointi

Jo pelin testausvaiheessa havaittiin, että tietyillä laitteilla skenejen välillä siirryttäessä peli saattoi kaatua itsestään. Skenen vaihdosta tarkastellessa Profilerilla havaittiin, että muistinkäyttö nousi huomattavasti, kun siirryttiin skenestä toiseen. Kuten kuvassa 17 tulee ilmi, koostuu alkuvalikossa pelin muistinkäyttö suurelta osin tekstuureista, jotka vievät 30 MB skenen käyttämästä muistista.



Kuva 17. Profilerin näkymä tarkastellessa alkuvalikon muistinkäyttöä.

Siirryttäessä pelissä alkuvalikosta Nikon huoneeseen ja siitä itse Nikon tarinaan, huomattiin tekstuurien vievän jo 1.10 GB muistista. Syynä tähän on käytettyjen tekstuurien suuri koko. Suurin osa pelin kuvista oli resoluutioltaan hyvin suuria ja täten veivät paljon muistia. Ratkaisuna tähän oli pienentää kuvien resoluutiota sekä asettaa kuvien tuontiasetuksista maksimikoko pienemmäksi, jolloin kuvien tarkkuus heikkeni, mutta myös niiden vaatima muistinkäyttö pieneni huomattavasti. Koska pelin pääasiallinen peliväline on mobiililaite, jonka näytön koko on hyvin rajallinen, pystyttiin kuvien resoluutiota pudottamaan melko pieneksi ilman, että kuvanlaatu silminnähtävästi kärsii. Kuvien resoluution ja tuontiasetusten muuttaminen pudotti muistinkäytön 1.1 gigatavusta noin 227 megatavuun.



Kuva 18. Muistinkäyttö siirryttäessä alkuvalikon kautta Nikon tarinaan.

Kyseisten toimenpiteiden jälkeen testilaitteilla ei enää havaittu kaatumisia ja ruudunpäivitystaajuuden keskiarvo kasvoi etenkin heikommilla laitteilla, joten voidaan todeta, että muistinkäytön optimoinnilla oli suuri merkitys pelin pelattavuuden kannalta.

5.3.3 Tallennustilan optimointi

Pelin vaatiman tallennustilan koko on suoraan riippuvainen siitä miten paljon peli sisältää grafiikkaa eli tässä tapauksessa 2D-kuvia. Itse skenejen sisällä luodut peliobjektit eivät itsessään vie juurikaan tallennustilaa edes pelin kehitysvaiheessa. Jotta pelin viemää tallennustilaa saadaan pienennettyä pitää siis joko vähentää olemassa olevien kuvien määrää ja uusiokäyttää samoja kuvia eri tilanteissa, joihin ne voisivat sopia, tai yksinkertaisesti pienentää kuvien resoluutiota. Koska PELIMO on hyvin graafinen peli, jossa tausta sekä hahmokuvien ja niiden eri ilmeiden vaikutus on merkittävä pelin pedagogisen tavoitteen kannalta, on kuvien uusiokäyttämisen sijaan pienennettävä kuvien resoluutiota. Yhtenä ratkaisuna itse asennettavan sovelluksen tallennustilan optimointiin olisi rakentaa peli niin, että latausvaiheessa peli ei sisällä muuta kuin alkuvalikon spritet. Tarinoihin edetessä peli lataisi internetistä kyseiseen tarinaan vaadittavat kuvat ja tarinan päätyttyä poistaisi ne puhelimen muistista. Tämä ratkaisu kuitenkin vaatii enemmän dataliikennettä ja, kun kyseessä on peruskoulua käyvät nuoret, on parempi ratkaisu ladata koko peli kuvineen kerralla, vaikka se tapahtuu tallennustilan kustannuksella. Tällöin varmistutaan myös siitä, että laitteella on tarpeeksi tallennustilaa koko pelille.

Ennen tallennustilan optimoinnin aloittamista, pelin Android-version asennustiedoston eli APK:n (Android package kit) koko oli 101 864 kB. Kun kuvien resoluution pienentäminen oli suoritettu, pienentyi asennustiedoston koko 47 579 kB.

6 LOPUKSI

Työn tarkoituksena oli tarkastella ominaisuuksia, joita hyvän hyötypelin tulee sisältää, ja rakentaa niiden ympärille toimiva kokonaisuus kohderyhmä huomioiden. Työn tekninen osuus toteutettiin käyttäen Unity-pelimoottoria ja Fungusta. Työn lopputuloksena saatiin kolme tarinaa sisältävä visuaalinen novelli. Jokainen tarina sisältää minipelejä, joiden käännösjärjestelmän toteutus oli osa opinnäytetyön teknistä toteutusta. Käännösjärjestelmän lisäksi osana opinnäytetyön teknistä toteutusta oli pelin optimointi kohderyhmän tarpeisiin sopivaksi. Työn lopuksi lukija saa käsityksen siitä mitä ovat nuorten hyötypelit, miten niitä voidaan käyttää opiskelussa ja mitä tulee ottaa huomioon, kun kohderyhmänä on nuoret. Lisäksi työn tekninen osuus vastaa kysymyksiin siitä, miten immersiota ja pedagogisuutta pystytään tukemaan erilaisilla teknologisilla ratkaisuilla.

Työn teoriaosuudessa kerrottiin, että mikäli pelin halutaan olevan mielenkiintoinen pelattava ja samalla opetuksellinen, on pelin sisällön ja pelimekaniikkojen peilattava kohderyhmän tieto- ja taitotasoa niin fyysisellä kuin psyykkisellä tasolla. Vaikka pelillistetyt opetusmetodit eivät pysty korvaamaan kaikilla osa-alueilla perinteisiä metodeja, havaittiin, että matemaattisia ja geometrisia aiheita opiskellessaan nuoret pystyivät hahmottamaan opetettavan asian pelinä paremmin verrattuna perinteisiin opetusmetodeihin. Nuorten opetuksessa hyötypelien käytön selvänä etuna havaittiin flow-tilan käytön mahdollisuudet. Johtuen diginatiivisuudesta ja nuorten kiinnostuksesta peleihin verrattuna perinteisiin opetusmenetelmiin pystytään pelaaja saattamaan helpommin flow-tilaan, joka edesauttaa oppimisessa. Flow-tilan käytön merkitys nouseekin teoriaosuudessa ylitse muiden, kun puhutaan hyötypelaamisen hyödyistä verrattuna perinteiseen opiskeluun.

Työssä ilmeni, että flow-tilan ylläpitämisen kannalta juonen ja pelin sisällön lisäksi teknisen toteutuksen merkitys nousee pintaan etenkin, kun kohderyhmänä on nuoret ja peliympäristönä opetustilanne. Näiden havaintojen valossa pystyttiin asettamaan tavoitteet pelin optimoinnin suhteen. Havaintojen perusteella pelin optimointi suoritettiin niin, että peli toimii vaivattomasti myös vanhemmilla laitteilla sekä laitteilla, jotka ovat alemmasta hintaluokasta. Optimoinnissa pyrittiin siihen, että peli toimisi tasaisella ruudunpäivitystaajuudella kaikilla yleisimmin käytössä olevilla laitteilla. Tämä mahdollistaa hyötypelien käytön opetuksessa myös silloin kun resursseista on pulaa ja joudutaan käyttämään alemman hintaluokan laitteita. Optimoinnista saatuja tuloksia analysoitiin käyttämällä Unityn Profiler -työkalua ja vertaamalla siitä saatuja tuloksia toisiinsa, sekä arvoihin, joita

vastaavanlaisissa peleissä on saatu. Optimoinnin lopputuloksena saatiin pelin vaatimaa tallennustilaa pienennettyä huomattavasti ja sen laitteelta vaatimia resursseja pienennettyä niin, että pelin pelaaminen on mahdollista kaikilla nykyaikaisilla laitteilla.

Teknisen toteutuksen toisena osana oli minipelien käännösjärjestelmän toteuttaminen. Teoriaosuudessa havaittiin, että lokalisoinnin merkitys hyötypeleissä on merkittävä, kun puhutaan pelin opetuksellisista ominaisuuksista. Tämän seurauksena minipeleille toteutettiin helposti käytettävä käännösjärjestelmä, jonka käyttäminen ei vaadi kääntäjältä ohjelmointi- tai tietoteknisiä taitoja. Käännösjärjestelmän tekninen toteutustapa mahdollistaa myös korjausten ja muutosten tekemisen helposti.

Työn valossa jatkokehityksessä nousee esille testaaminen ja sen tarve. Vaikka tekninen toteutus on tehty niin, että pelin pitäisi teoriassa toimia useimmilla laitteilla, on testattujen laitteiden määrä kuitenkin kohtuu pieni. Pelin testaaminen sen oikeassa käyttötarkoituksessa, eli opetustilanteessa useilla eri laitteilla vahvistaisi teoriaa ja saattaisi tuoda ilmi asioita, joita ei pystytty huomioimaan suunnittelu ja kehitysvaiheessa.

Työn lopullista tulosta arvioitaessa pystytään toteamaan työn täyttävän kaikki toimeksiannossa vaaditut kohdat. Työssä saatiin teoriaosuudessa selvitettyä, mitä opetuspelin toteutuksessa pitää ottaa huomioon, ja sen perusteella pystyttiin luomaan toimiva tekninen kokonaisuus, joka tukeutuu teoriaosuudessa saatuun tietoon hyötypeleistä ja nuorten digitaalisesta opiskelusta. Sen lisäksi että työssä tehtyä selvitystä pystyttiin käyttämään hyväksi PELIMO -hankkeessa, soveltuu se myös muihin lapsille suunnattuihin hyötypeleihin.

LÄHTEET

Aarnio, J. 2019. OPETUSPELIN SUUNNITTELU JA TEKNINEN TOTEUTUS. Viitattu 18.5.2020. https://www.theseus.fi/bitstream/handle/10024/226774/Juha_Aarnio.pdf?sequence=2&isAllowed=y

Aku Ankka 2020. Aku Ankka. Viitattu 25.2.2020. https://www.akuankka.fi/hahmo/1/aku-ankka

Avraamidou, L.; Goedhart. M. 2017. Tell me a Story: the use of narrative as a learning tool for natural selection. Viitattu 6.2.2020. https://www.tandfonline.com/doi/full/10.1080/09523987.2017.1324361.

Bellotti, F.; Berta, R.; De Gloria, A. 2010. Designing Effective Serious Games: Opportunities and Challenges for Research. Viitattu 23.10.2019. https://www.researchgate.net/publication/49588460_Designing_Effective_Serious_Games_Opportunities_and_Challenges_for_Research.

Bellotti, F.; Berta, R.; De Gloria, A.; Ott, M.; Arnab, S.; de Freitas, S.; Kiili, K. 2014. Designing Serious Games for education: from Pedagogical principles to Game Mechanisms. Viitattu 23.10.2019. https://hal.archives-ouvertes.fr/hal-00985800/document.

Cziksentmihalyi, M. 1990. Flow – The Psychology of optimal experience. Viitattu 16.10.2019. https://www.researchgate.net/profile/Mihaly_Csikszentmihalyi/publication/224927532_Flow_The_Psychology_of_Optimal_Experience/links/55ad6c2f08aed614b097b39e.pdf.

Design Digitally 2017. Gamification, Serious Games, and Simulations: What's the Difference? Viitattu 26.2.2020. https://www.designingdigitally.com/blog/2018/10/gamification-serious-games-and-simulations-what%E2%80%99s-difference.

Deterding, S.; Dixon, D.; Khaled, R. Nacke, L. 2011. From Game Design Elements to Gamefulness: Defining Gamification. Viitattu 11.10.2019. https://www.researchgate.net/publication/230854710_From_Game_Design_Elements_to_Gamefulness_Defining_Gamification.

Duncan, B. 2016. Why the German language has so many great words. Viitattu 6.2.2020. https://theconversation.com/why-the-german-language-has-so-many-great-words-55554.

Enger, M. 2013. Game Engines: How do they work? Viitattu 11.2.2020. https://www.giantbomb.com/profile/michaelenger/blog/game-engines-how-do-they-work/101529/.

Epic Games 2020a. Unreal Engine. Viitattu 13.3.2020. https://www.unrealengine.com/en-US/.

Epic Games 2020b. Unreal Engine commercial game deployment guidelines. Viitattu 13.3.2020. https://www.unrealengine.com/en-US/release.

Epic Games 2020c. Marketplace. Viitattu 13.3.2020. https://www.unrealengine.com/marketplace/en-US/store.

Funday Factory 2017. Game Design for Kids. Viitattu 07.12.2019. https://fundayfactory.com/media/147699/age-appropriate-game-design-for-children.pdf.

Fungus Games 2020. Fungus. Viitattu 13.3.2020. https://assetstore.unity.com/packages/templates/systems/fungus-34184.

GameDesigning 2020a. Beginner's Guide to Game Mechanics. Viitattu 3.4.2020. https://www.gamedesigning.org/learn/basic-game-mechanics/.

GameDesigning 2020b. The Top 10 Video Game Engines. Viitattu 6.3.2020. https://www.game-designing.org/career/video-game-engines/.

Gamelearn Team 2017. Eight examples that explain all you need to know about serious games and game-based learning. Viitattu 17.10.2019. https://www.game-learn.com/all-you-need-to-know-serious-games-game-based-learning-examples/.

Google 2020. Google Sheets. Viitattu 22.4.2020. https://www.google.com/sheets/about/.

Hallinen, N.; Walker, E.; Wylie, R.; Ogan, A.; Jones, C. 2009. I Was Playing When I Learned: A Narrative Game for French Aspectual Distinctions. Viitattu 30.10.2019. http://chrisharrison.net/amyogan/files/NHallinen_FrenchVerbGame.pdf.

Huang, W. & Soman, D. 2013. A Practitioner's Guide To Gamification Of Education. Viitattu 16.10.2019. https://inside.rotman.utoronto.ca/behaviouraleconomicsinaction/files/2013/09/GuideGamificationEducationDec2013.pdf.

Huang, Z. 2019. China's new rules on video games: no blood, dead bodies, or mahjong. Viitattu 7.4.2020. https://www.inkstonenews.com/tech/chinas-new-rules-video-games-no-blood-dead-bodies-or-mahjong/article/3007275.

Khumansinh, J. 2015. Game Assets (Graphics) Optimization. Viitattu 2.3.2020. http://www.theappguruz.com/blog/game-assets-graphics-optimization.

Klappenbach, M. 2020. Understanding and Optimizing Video Game Frame Rates. Viitattu 3.3.2020. https://www.lifewire.com/optimizing-video-game-frame-rates-811784.

Lachat-Leal, C. 2016. Serious games localisation. Playability and translation strategies. Viitattu 11.2.2020. https://www.researchgate.net/publication/336835481_Serious_games_localisation_Playability_and_translation_strategies%20.

Lebowitz, J.; Klug. C. 2011. Interactive Storytelling for Video Games: A Player-Centered Approach to Creating Memorable Characters and Stories. Viitattu 17.4.2020. https://epdf.pub/interactive-storytelling-for-video-games-a-player-centered-approach-to-creating-.html.

Libell, M. 2004. The pros and cons of anthropomorphism. Viitattu 27.10.2019. https://lup.lub.lu.se/search/publication/1037629

Liszewski, A 2019. China's New Video Game Rules Officially Ban Blood, Corpses, Mahjong, and Poker. Viitattu 7.4.2020. https://gizmodo.com/chinas-new-video-game-rules-officially-ban-blood-corps-1834221957.

Mankkinen, J. 2019. Pelisuunnittelijoille tehtiin kirja: "Ilotulitusraketit räjähtelivät päässäni ja sain hillittömän inspiraation". Viitattu 3.4.2020. https://yle.fi/uutiset/3-10989839.

Mojang. 2020. Minecraft. Viitattu 18.5.2020. https://www.minecraft.net/en-us/.

Murphy, C.; Chertoff, D.; Guerrero, M.; Moffitt, K. 2015. Design Better Games: Flow, Motivation, and Fun in Learning Games. Viitattu 16.10.2019. https://www.researchgate.net/publication/266052900_Design_Better_Games_Flow_Motivation_and_Fun_in_Learning_Games.

Naul, E. 2018. Why Story Matters: A Review of Narrative in Serious Games. Viitattu 27.10.2019. https://repositories.lib.utexas.edu/handle/2152/68898%20.

Opetushallitus 2017. Poikien oppimishaasteet ja -ratkaisut vuoteen 2025. Viitattu 10.10.2019 Saatavilla sähköisesti osoitteessa https://www.oph.fi/sites/default/files/documents/poikien-oppimishaasteet-ja-ratkaisut-vuoteen-2025.pdf .

Olson, C. 2020 9 Benefits of Video Games for Your Child. Viitattu 18.5.2020 https://www.parents.com/kids/development/benefits-of-video-games/.

Phaser 2020. Viitattu 25.2.2020. https://phaser.io/.

Papanastasiou, G.; Athanasios, D.; Skianis, C. 2017. Serious Games in Preschool and Primary Education: Benefits And Impacts on Curriculum Course Syllabus. Viitattu 21.10.2019. https://www.researchgate.net/publication/313115425_Serious_Games_in_Preschool_and_Primary_Education_Benefits_And_Impacts_on_Curriculum_Course_Syllabus.

PELIMO 2019. Pelillistäminen mahdollisuutena monikulttuurisuuskasvatuksessa. Viitattu 17.2.2020. Saatavilla sähköisesti osoitteessa https://pelimo.turkuamk.fi/.

PhysicsJS 2020. Viitattu 25.2.2020. http://wellcaffeinated.net/PhysicsJS/.

Prensky, M. 2001. Digital Natives, Digital Immigrants. Viitattu 28.1.2020. http://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf.

Puolakka, M 2017. Mitä hyötypelit ovat? NordicEdu. Viitattu 26.2.2020. https://nordicedu.com/blogi/mita-hyotypelit-ovat.

Pynn, I. 2017. School Has a Bad Storyline: Gamification in Educational Environments. Viitattu 6.4.2020. https://stars.library.ucf.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=6652&context=etd.

Reddit 2018. WHY exactly is Paper2D not recommended? Viitattu 13.3.2020. https://www.reddit.com/r/unrealengine/comments/7kwt2a/why_exactly_is_paper2d_not_recommended/.

Redford, K. 2019. Graphic Novels Welcome Everyone Into the Reading Conversation. Viitattu 6.4.2020. http://dyslexia.yale.edu/resources/tools-technology/suggested-reading/graphic-novels/.

Rouse, M. 2015. localization. Viitattu 18.5.2020. https://searchcio.techtarget.com/definition/localization.

Screensiz 2020. Screen sizes. Viitattu 11.2.2020. http://screensiz.es/.

Sonawane, K. 2019. Serious Games Market Expected to Reach $9,167 Million by 2023. Viitattu 10.10.2019 https://www.alliedmarketresearch.com/press-release/serious-games-market.html.

Stege, L.; Lankveld, G.; Spronck, P. 2011. Serious Games in Education. Viitattu 14.10.2019. https://pdfs.semanticscholar.org/2d7f/20b98562caa2e27952c2319976bb2112b285.pdf%.

Sharma, H. 2017. Learning a language through immersion. Viitattu 24.3.2020. https://www.worddive.com/blog/learning-a-language-through-immersion/.

Seterra 2019. Europe: Countries - Map Quiz Game. Viitattu 25.10.2019. https://online.seterra.com/en/vgp/3007

Sol-tribe 2014. Visual Novel Toolkit Free. Viitattu 13.3.2020. https://assetstore.unity.com/packages/templates/visual-novel-toolkit-free-9416.

Team Salvato 2017. Doki Doki Literature Club! Viitattu 26.2.2020. https://teamsalvato.itch.io/ddlc.

Techopedia 2012. Alpha Channel. Viitattu 21.4.2020. https://www.techopedia.com/definition/1945/alpha-channel.

TechTerms 2013. Framework Definition. Viitattu 17.4.2020. https://techterms.com/definition/framework.

Texeira, A. 2017. Video Game Culturalization: Definition and Best Practices (IGDA LocSIG). Viitattu 6.2.2020. https://www.at-it-translator.com/video-game-culturalization-definition-and-best-practices-igda-locsig/.

Tilastokeskus 2017a. Media equipment use , those aged 10 or over (2002 / 2017) % by Background variable, Year, Sex, Data and Response. Viitattu 11.10.2019. http://pxnet2.stat.fi/PXWeb/pxweb/en/StatFin/StatFin__eli__vpa__Tieto_ja_viestintatekniikka/statfin_vpa_pxt_185.px/table/tableViewLayout1/.

Tilastokeskus 2017b. Pelaamisen useus. Viitattu 14.10.2019. http://pxnet2.stat.fi/PXWeb/pxweb/fi/StatFin/search/?searchquery=pelaamisen+useus.

Tolomei, S. 2017. Shrinking APKs, growing installs. Viitattu 17.4.2020. https://medium.com/googleplaydev/shrinking-apks-growing-installs-5d3fcba23ce2.

Tung, L. 2015. Android fragmentation: There are now 24,000 devices from 1,300 brands. Viitattu 11.2.2020. https://www.zdnet.com/article/android-fragmentation-there-are-now-24000-devices-from-1300-brands/.

Unity Technologies 2020a. Unity. Viitattu 13.3.2020. https://unity.com/.

Unity Technologies 2020b. Compare Plans. Viitattu 23.3.2020. https://store.unity.com/compare-plans.

Unity Technologies 2020c. Asset Store. Viitattu 13.3.2020. https://assetstore.unity.com/.

Unity Technologies 2020d. Optimizing graphics performance. Viitattu 2.3.2020. https://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html.

Unity Technologies 2020e. Profiler overview. Viitattu 3.3.2020. https://docs.unity3d.com/Manual/Profiler.html.

Unity Technologies 2020f. Reducing the file size of your build. Viitattu 3.3.2020. https://docs.unity3d.com/Manual/ReducingFilesize.html.

Unity Technologies 2020g. SpriteRenderer. Viitattu 21.4.2020. https://docs.unity3d.com/ScriptReference/SpriteRenderer.html.

Unity Technologies 2020h. Canvas. Viitattu 21.4.2020. https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/UICanvas.html.

Walton, J. 2019. Why are game install sizes getting so big? Viitattu 3.3.2020. https://www.pcgamer.com/why-are-game-install-sizes-getting-so-big/.

Wellesley, R. 2014. What Makes Serious Games Effective? 5 Questions to Ask When Evaluating Serious Games in the Workplace. Viitattu 1.2.2020. https://www.researchgate.net/publication/262798193_What_Makes_Serious_Games_Effective_5_Questions_to_Ask_When_Evaluating_Serious_Games_in_the_Workplace.

Williams, R. Young people who play video games have higher moral reasoning skills. Viitattu 18.5.2020. https://inews.co.uk/news/technology/young-people-who-play-video-games-have-higher-moral-reasoning-141532

Wood, R.; Griffiths, M.; Parke, A. 2007. Experiences of Time Loss among Videogame Players: An Empirical Study. Viitattu 15.12.2019. https://pdfs.semanticscholar.org/7f2e/6179bd615f41d51845857e2dce55abba33df.pdf.

Yoccoz, D. 2020. How to Localize Indie Games for Non-Latin Alphabets. Viitattu 6.2.2020. https://www.gamasutra.com/blogs/DamienYoccoz/20200304/359036/How_to_Localize_Indie_Games_for_NonLatin_Alphabets.php.

Zaiets, S. 2020. Game Localization To Arabic - Why And How? Viitattu 6.2.2020. https://www.localizedirect.com/posts/arabic-game-localization.