



Karelia-ammattikorkeakoulu
Tradenomi (AMK)

Pelillistäminen ja pelaajakokemus oppimispelin kontekstissa

Inari Pusa

Opinnäytetyö, huhtikuu 2024

www.karelia.fi



OPINNÄYTETYÖ
Huhtikuu 2024
Tietojenkäsittely

Tikkarinne 9
80200 JOENSUU
+358 13 260 600

Tekijä
Inari Pusa

Nimeke
Pelillistäminen ja pelaajakokemus oppimispelin kontekstissa

Toimeksiantaja
Maa- ja metsätalouden koulutusohjelma, Karelia-ammattikorkeakoulu

Tiivistelmä

Tässä opinnäytetyössä tarkastellaan pelien ja oppimisen välistä suhdetta ammattikorkeakouluopiskelijoille suunnatulle Metsätuhot-opintojaksolle toteutetun oppimispelin jatkokehittämiseksi. Pelit ovat monille koukuttavaa vapaa-ajan viihdettä. Opinnäytetyössä selvitetään, miten pelien viihdyttävää ja koukuttavaa luonnetta on pyritty soveltamaan oppimisen motivoijana sekä tutkitaan, mitkä tekijät vaikuttavat pelaajakokemukseen. Työn käytännön tavoitteena oli Karelia-ammattikorkeakoulun Metsätuhot-opintojaksolle keväällä 2023 toteutetun oppimispelin jatkokehittäminen.

Pelin kehityskohteita kartoitettiin pelillistettyä oppimista käsittelevän kirjallisuuden ja peliä pelanneiden opiskelijoiden antaman palautteen pohjalta. Peliohjelmointi toteutettiin Unreal Engine -pelimoottorin blueprint-ohjelmointia sekä C++-kieltä käyttäen. C++-ohjelmointiympäristönä toimi Visual Studio 2022.

Opinnäytetyön tuloksena Metsätuhot-pelin käytettävyyttä sekä oppimis- ja pelaajakokemusta paranneltiin peliohjelmoinnin ja -kehityksen keinoin. Pelaajalle jaetaan aiempaa enemmän tietoa pelin pelaamisesta sekä pelissä edistymisessä oppimisen motivoimiseksi. Käyttöliittymiä selkeytettiin ja pelin sisäistä navigointia helpotettiin. Oppimisen kannalta tärkeän sosiaalisen kanssakäymisen mahdollistamiseksi sekä saatujen opiskelijapalautteiden perusteella seuraava tärkeä jatkokehityksen suunta Metsätuhot-pelille olisi verkkoversion toteuttaminen.

Kieli
suomi

Sivuja 89
Liitteet 1
Liitesivumäärä 1

Asiasanat
pelaajakokemus, käyttöliittymäsuunnittelu, pelillistäminen, peliohjelmointi, Unreal Engine
5



THESIS
April 2024
Degree Programme in Business Information Technology

Tikkarinne 9
80200 JOENSUU
FINLAND
+ 358 13 260 600

Author
Inari Pusa

Title
Gamification and player experience in the context of learning

Commissioned by
Commissioner Karelia University of Applied Sciences

Abstract

This thesis examines the relationship between learning and video games. Video games are an engaging and popular form of entertainment. The research question of this thesis is whether the entertaining and immersive nature of games can be used to motivate a learning experience. A further question contemplates the factors that influence the player experience.

The objective of the practical work of the thesis was to improve the first version of the educational game developed for students of the course Causes of Forest Destructions. The course in question is provided by Karelia University of Applied Sciences. The first version of the game was implemented in the spring of 2023. The focus of the development in this thesis was surveyed based on research of gamification of learning and feedback from students who had played the first version of the game. The game was programmed using the Unreal Engine game development software.

As a result of the thesis, the learning and player experience of the game Causes of Forest Destruction was improved, and the usability was enhanced. Compared to the previous version, more information is provided to the player. This information includes knowledge about how to play the game and the player's progress in the game to motivate them. All user interfaces were redesigned to clarify in-game navigation. Based on student feedback and the importance of social interactions for learning, the next goal should be an online game implementation.

Language
Finnish

Pages 89
Appendices 1
Pages of Appendices 1

Keywords
Player experience, interface design, gamification, game programming, Unreal Engine 5

Sisältö

Lyhenneluettelo ja sanasto	5
1 Johdanto	8
2 Oppimisen pelillistäminen ja pelaajakokemuksen kehittäminen.....	9
2.1 Pelaajakokemus	9
2.2 Käyttöliittymäsuunnittelu	11
2.3 Kontekstina oppiminen	16
2.3.1 Miten ymmärrämme oppimisen.....	17
2.3.2 Digitalisaation tuoma muutos koulutukseen.....	20
2.4 Oppimisen pelillistäminen	22
2.5 Käyttöliittymäsuunnittelun käytänteitä oppimisen tueksi	27
2.6 Saavutettavuus	28
3 Menetelmät	29
3.1 Metsätuhot-pelin kehitysympäristöt ja lähtökohdat	29
3.2 Muut työkalut	31
3.3 Kohderyhmä	31
4 Kehitystyö	32
4.1 Valikot.....	32
4.1.1 UI-komponentit ja kaikkien valikkojen yhteiset kehityskohteet	32
4.1.2 Aloitusinfo	40
4.1.3 Päävalikko ja peliin kirjautuminen	42
4.1.4 Asetukset.....	45
4.1.5 Ohjeistus.....	54
4.1.6 Pysäytys	56
4.1.7 Tekijät	58
4.2 Pelin kulku	60
4.2.1 Kentän käyttöliittymä.....	60
4.2.2 Tenttikysymysten käyttöliittymä	63
4.2.3 Oikeat vastaukset ja edistymisen mittaaminen	66
4.2.4 Kysymysten lisääminen	68
4.2.5 Respawn-toiminto	69
4.2.6 Kentän läpäiseminen	71
4.2.7 Pelin läpäiseminen.....	74
4.3 Vihjekirja	76
4.3.1 Sisältö ja selailu	76
4.3.2 Käyttöliittymä ja komennot.....	78
4.3.3 Kuvien skaalaus.....	79
4.3.4 Vihjemittari	80
4.4 Muut kehityskohteet.....	81
4.4.1 Fontti.....	82
4.4.2 Vihjeiden toistuminen ja vihjeen löytämisestä ilmoittaminen.....	83
4.5 Tekijänoikeudet.....	84
5 Pohdinta.....	84
5.1 Rajoitteet.....	84
5.2 Haasteet	85
5.3 Kehitystyön saavutetut tavoitteet	86
5.4 Pelillistetty tulevaisuus	86
Lähteet.....	87
Liitteet	1

Lyhenneluettelo ja sanasto

Actor

Yksi ja ylin kolmesta Unreal Enginen tarjoamasta Actor-luokasta. Actor-luokkaa voi käyttää monenlaisten olioiden pohjana, esimerkiksi aseiden, rakennusten, tallennuspisteiden tai ajoneuvojen. (Li 2023, luku 5.)

Katso myös: Character, Pawn

Assetti

Pelikehityksen yleistermi pelissä käytetyille rakennuspalikoille. Assetit voivat olla esimerkiksi peligrafiikkaa (2D- tai 3D-taidetta), 3D-malleja, äänileikkeitä, videoita. (Li 2023, luku 12.)

Character

Pawn-luokasta periytyvä alaluokka, jota käytetään pelaajahahmojen luomiseen. Character-luokka vastaanottaa käyttäjän syötteitä ja sinne voidaan sisällyttää pelaajahahmon tarvitsemia arvoja esimerkiksi elämäpisteitä. (Li 2023, luku 5.)

Katso myös: Actor, Pawn

DataTable

DataTable on Unreal Enginen tarjoama struktuuri tiedon säilyttämiseen ja käyttämiseen. DataTable-struktuurin sisälle voi sisällyttää monentyyppistä tietoa kuten muuttujia, listoja muuttujista sekä olio- ja asettiviittauksia. (Epic Games, Inc 2024a.)

GameInstance

GameInstance on hallintaa helpottava Unreal Enginen tarjoama objekti, joka edustaa käynnissä olevan pelin ilmentymää.

GameInstance-luokkaan voidaan määrittää muun muassa pelisessiolle tarpeellisia muuttujia, funktioita ja delegaatteja. (Li 2023, luku 5.)

HUD

HUD on lyhenne sanoista Heads-up Display, ja sillä viitataan peleille tyypilliseen käyttöliittymätyyppiin. HUD-käyttöliittymästä jaetaan pelaajalle tietoa, joka on pelaajalle jatkuvasti oleellista, esimerkiksi elämäpisteiden määrä. (Conçalo, Sherry, Pereira & Fozi 2022, luku 8.)

Konstruktori

Luokan ilmentymän luontiin käytettävä funktio, jossa voidaan alustaa olion attribuutit. Konstruktoria kutsutaan aina ensimmäisenä, kun luokasta luodaan olio. (Weistfeld 2019, luku 3.)

Luokka (olio-ohjelmointi)

Luokka määrittää olion attribuutit ja ominaisuudet olio-ohjelmoinnissa (Weistfeld 2019, luku 1).

Katso myös: olio-ohjelmointi

Node

Termillä node viitataan Unreal Engine -pelimoottorin visuaalisen ohjelmoinnin funktioihin, operaatioihin, tapahtumiin, muuttujakutsuihin ja muihin tapahtumiin, joita halutaan kutsua tai joilla halutaan viitata blueprint-ohjelmoinnissa (Epic Games, Inc. 2024b).

Olio-ohjelmointi (Object-Oriented Concepts)

Ohjelmoinnin konsepti, jolle tyypillisiä piirteitä ovat kapselointi, periytyminen, polymorfismi eli monimuotoisuus ja kompositio. (Weistfeld 2019, luku 1).

Katso myös: periytyminen

Pawn

Actor-luokasta periytyvä alaluokka, jota käytetään tyypillisesti olioihin, joita pelaaja voi kontrolloida, mutta jotka eivät ole itse pelaajan luokka (Character). Esimerkiksi auto tai muu pelaajan ohjaama menopeli on Pawn-luokalle hyvä käyttökohde. (Li 2023, luku 5.)

Katso myös: Character, Pawn

Periytyminen (olio-ohjelmointi)

Periytyminen ohjelmoinnin kontekstissa tarkoittaa kykyä periyttää luokkia toisista luokista. Yläluokkaa, josta periytyvä luokka muodostetaan, kutsutaan vanhempiluokaksi (parent/super class). Siitä periytyvä luokka on alaluokka (child/sub class). Alaluokka perii vanhempansa attribuutit ja funktiot. (Weistfeld 2019, luku 1.)

Katso myös: olio-ohjelmointi, luokka

POV (point of view)

Lyhenne POV tarkoittaa katselukulmaa, josta esimerkiksi peliä katsotaan. Yleisiä peleissä käytettyjä kulmia ovat FP – First Person eli toimijan näkökulma ja TP – Third Person, jossa kamera seuraa pelaajahahmoa (yleensä) selkäpuolelta.

SaveGame

Unreal Enginen tarjoama luokka pelin muuttujien ja tapahtumien tallentamista varten (Epic Games, Inc. 2024c).

ToolTip

Unreal Enginen työkalu, jolla pelaajalle voidaan jakaa lisätietoa ponnahdusikkunan muodossa (Laley 2021).

UserWidget-luokka

Unreal Enginen tarjoama luokkatyyppi käyttöliittymille ja käyttöliittymän osille (Conçalo, Sherry, Pereira & Fozi 2022, luku 8).

1 Johdanto

Tämä opinnäytetyö on jatkoa keväällä 2023 alkaneelle Metsätuhot-projektille. Projektin pitkän tähtäimen tavoitteena on pelillistää Karelian metsäinsinööriopiskelijoiden kurssitarjontaan kuuluva sekä muiden opiskelijoiden vapaasti valittavissa oleva Metsätuhot-opintojakso. Ensimmäisen version Metsätuhot-pelistä toteutti keväällä 2023 kolmannen vuoden tietojenkäsittelyn tradenomiopiskelijaryhmä (Liite 1), tukenaan peliohjelmoinnin lehtori Seppo Nevalainen ja tietojenkäsittelyn lehtori Mika Matveinen. Toimeksiantajana Metsätuhot-projektissa toimii Karelia-ammattikorkeakoulun Maa- ja metsätalouden koulutusohjelma, edustajanaan metsätalouden lehtori Sini Rantanen.

Metsätuhot-pelin ensimmäinen versio otettiin käyttöön opintojaksolla vaihtoehtoisena tapana suorittaa tentti. Loppuvuoden 2023 aikana Metsätuhot-peliä haluttiin lähteä jatkokehittämään ensimmäiseen versioon jääneiden puutteiden ja lehtori Rantasen opiskelijoilta saaman palautteen pohjalta, erityisesti käytettävyyden ja käyttöliittymien osalta.

Tässä opinnäytetyössä tutustutaan ensiksi pelaajakokemukseen, käyttöliittymäsuunnitteluun ja oppimiseen liittyvään teoriaan, joka on toiminut pohjana kehityskohteiden kartoittamisessa ja työskentelyssä. Luvussa 3 tutustutaan tarkemmin itse Metsätuhot-peliin ja kehitystyössä käytettyihin menetelmiin. Luvussa 4 tarkastellaan, kuinka Metsätuhot-pelin käytettävyyttä kehitettiin oppimisen tukemiseksi. Kehitystyö on jaettu kehityskohteittain otsikoihin, joissa kustakin kohteesta kuvataan lähtötilanne eli pelin ensimmäisen version tilanne, kehitysidea ja toteutus sekä mahdolliset havainnot ja ideat jatkokehitykseen. Raportin lopussa tarkastellaan opinnäytetyöprosessia kokonaisuutena pohtien onnistumisia, saavutuksia, puutteita ja rajoitteita, joita prosessin aikana kohdattiin.

2 Oppimisen pelillistäminen ja pelaajakokemuksen kehittäminen

2.1 Pelaajakokemus

Pelaajakokemukseen syvennyttäessä voidaan lähteä aivan alkuruudusta: mitä pelit ovat? Kirjassaan Pelintekijän käsikirja Vuorela (2007, 16) määrittelee vastauksen kysymykseen seuraavasti: ”Pelaaminen on pelaajan näkökulmasta ajanvietteeksi tarkoitettua tuottamatonta toimintaa, jolla on alkutilanne, säännöt ja päämäärä.” Tätä osuvaa tiivistystä, on syytä purkaa ja tarkastella syvemmltä, ymmärtääksemme miksi pelaajakokemus on tärkeä (ellei tärkein) osa pelisuunnittelua.

Pelit ovat siis tuotteita, pelikehittäjien luomuksia kuluttajille eli pelaajille. Pelikehittäjälle haastetta luo pelaajan toiminnan ennakoimattomuus. Pelien kontekstissa tapa, jolla kuluttaja käyttää tuotetta voi olla suhteellisen arvaamatonta ja vaikeasti ennakoitavissa, verrattuna esimerkiksi kirjan lukijaan tai elokuvan katselijaan. (Schell 2019, luku 2.)

Samaa vertausta (elokuvan katselija vrt. pelaaja) esitellessään Inkinen (2008, 9) korostaa pelaajan roolia. Pelaaja ei ole tapahtumien sivustakatselija, vaan aktiivinen toimija sekä luova tekijä pelimaailmassa (Inkinen 2008,9). Inkisen kanssa samassa pelaajakokemusta käsittelevässä kokoomateoksessa Leino, Wirman ja Fernandez (2008, 26) jatkavat pelaajan tärkeyden korostamista itse pelin kannalta. He toteavat, että ilman pelaajaa/pelaajia peli olisi staattinen ja eloton. Koska pelaajat ovat niin ratkaisevassa roolissa pelissä, pelisuunnittelijoiden tulisi pitää pelaajakokemus yhtenä prioriteettina ja jopa päätavoitteenaan suunnittelussaan (Leino ym.). Pelisuunnittelijoiden olisi pyrittävä ymmärtämään pelaajia kokonaisuutena, ihmisinä erilaisine taustoineen, motivaattoreineen ja syineen sekä tavoitteineen, jotka vaikuttavat pelaajakokemukseen, eikä typistää heitä suunnittelussaan vain pelin sisäisten konfliktien ratkojiksi (Leino ym. 2008, 26; Burke 2014, luku 1). Tätä kutsutaan pelaajakeskeiseksi suunnitteluksi (Burke 2014, luku 1).

Schell (2019, luku 2) on samoilla linjoilla Leinon ym. kanssa. Schellin mukaan juuri kokemuksen, ei pelin, tarjoaminen pelaajalle tulisi olla pelikehittäjän

ensisijainen prioriteetti. Hän esittää ajatuksen, että kehittäjälle pelin itsessään tulisi olla vain työkalu, kehys kokemuksen välittämiseen.

Inkinen (2008, 12–13) avaa kokemuksen olemusta. Englannin kielen sana ”experience” kääntyy suomeksi elämykseksi ja kokemukseksi.

Pelaajakokemus sanassa viitataan siis pelaajan kokonaisvaltaiseen elämykseen. Mitkä tekijät vaikuttavat pelaajakokemukseen? (Inkinen (2008, 13) esittelee kuusi tuotteen (kuten pelin) perusedellytystä kokemuksen tarjoamiseen:

- Yksilöllisyys: tuote on tarpeeksi erottuva ja yksilöllinen muista tuotteista, tai jopa uniikki.
- Autenttisuus: tuote on luotettava, eettinen ja aito.
- Tarina: linkittyy vahvasti autenttisuuteen. Tarina muodostaa yhteyden kokemuksen ja todellisen maailman välille, antaen kokemukselle lisäarvoa ja sosiaalisia merkityksiä.
- Moniaistillinen elämys: mitä useampaa aistia tuotteella voi stimuloida, sitä voimakkaampi kokemus syntyy. Visuaalisen näytävyyden lisäksi, esimerkiksi äänet ja musiikki voimistavat kokemusta.
- Kontrasti: vaikka yhteys todelliseen maailman on hyväksi, riittävä kontrasti arjesta ja päivittäin toistuvista rutiineista elävöittää kokemusta.
- Vuorovaikutus: suhde tuotteen käyttäjän, tuotteen tarjoajan tai oppaan sekä muiden tuotetta käyttävien kanssa.

Aiemmat kuusi tuotteen elementtiä kokemuksen perustaksi ovat yleispäteviä, eivät pelkästään videopelien sidottuja. Videopelien rakenteellisia pelikokemukseen vaikuttavia tekijöitä voi tarkastella tarkemmin LeBlancin, Hunicken ja Zubekin (2004, 1–2) luoman MDA-viitekehyksen avulla. MDA-lyhenne tulee sanoista mechanics, dynamics ja aesthetics:

- mekaniikka (mechanics): pelin sisäiset komponentit, data ja algoritmit.
- dynamiikka (dynamics): kuvaa mekaniikan ajonaikaista toteutumista, pelaajan syötteeseen reagoimista ja informaation sekä palautteen antamista pelaajalle.

- estetiikka (aesthetics): haluttujen tunnetilojen nostattamista pelaajassa, hänen ollessaan vuorovaikutuksessa pelimaailman kanssa.

(LeBlanc, Hunicke & Zubek 2004, 1–2.)

Vuorela (2007, 26–30) esittelee muutamia pelien vetovoiman osia, eli pelien koukuttavia piirteitä, jotka tekevät pelistä kokemuksen. Vuorelan mukaan pelikoukkuja ovat eläytyminen, todellisuuspako, haasteet, simuloitu vaara, uhkapeli/sokea sattuma, juoni, tarina, stressin tai aggressioiden purkaminen, kilpailuvietti ja yhteisö.

Yksi pelaajakokemukseen suuresti vaikuttava tekijä, on pelin käyttöliittymä, sillä käyttöliittymä on Schellin (2019, luku 15) mukaisesti linkki pelaajan ja pelin välillä. Käyttöliittymiin ja niiden suunnitteluun tutustumme tarkemmin seuraavassa luvussa.

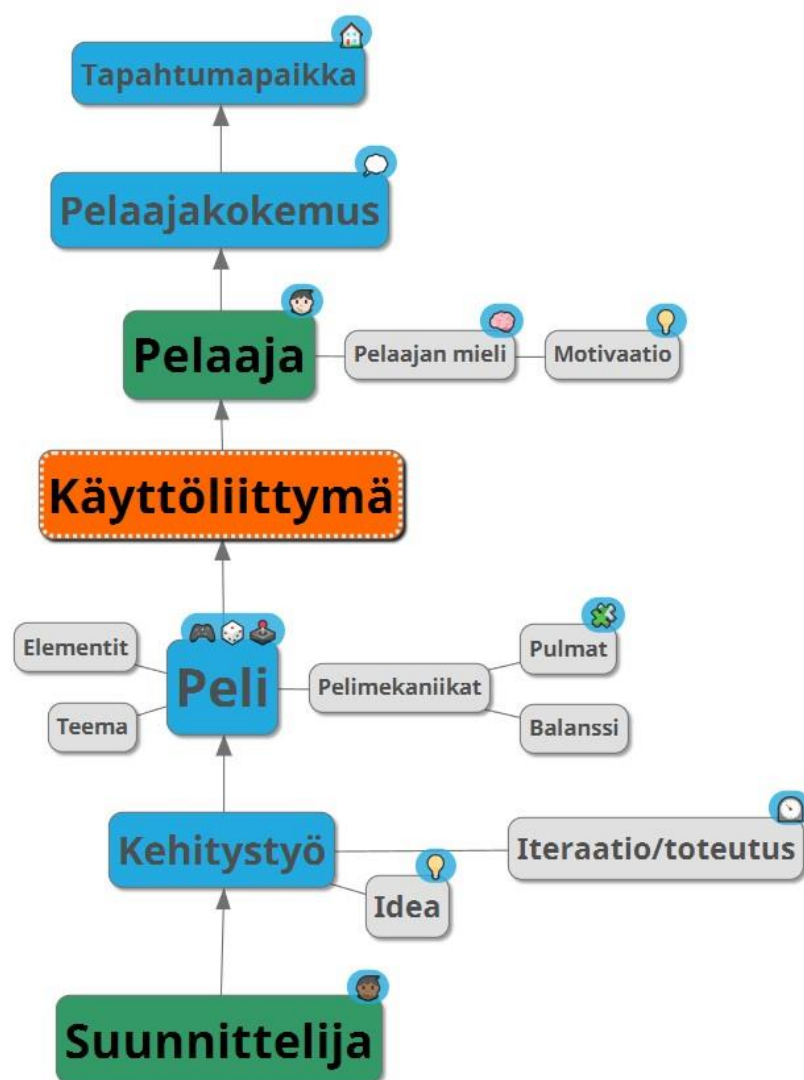
2.2 Käyttöliittymäsuunnittelu

Käyttöliittymällä (interface) viitataan järjestelmään, jolla ihminen operoi tietokonetta. Tietokonetta ei (lähes poikkeuksetta) operoida koskaan suoraan vaan välittyneesti käyttöliittymän kautta. Käyttöliittymän osia ovat sekä sitä kontrolloiva tietokoneohjelma (software) että fyysinen syöttölaite (kuten näppäimistö tai hiiri) käyttäjän antaman syöteen (input) rekisteröimiseksi ja tietokoneen palautteen (output) esittämiseksi. (Oulasvirta 2011, 23.)

Tietokoneen käytössä toistuvaa syöttölaitteiden käyttötapaa kutsutaan vuorovaikutustekniikaksi (Isokoski 2011, 175). Osa vuorovaikutustekniikoista on vakiintuneita, kuten kohteen valitseminen hiirtä liikuttamalla ja hiiren painiketta klikkaamalla. Vaikka ohjelmistot ja laitteistot uusiutuvat jatkuvasti, vuorovaikutustekniikat, joilla niitä käytetään uusiutuvat hitaammin. Isokoski (2011, 176) toteaa vuorovaikutustekniikoiden vakiintuneisuuden käyvän järkeen. Jos uuden ohjelmiston tai laitteen käyttäminen vaatisi aina uuden vuorovaikutustekniikan opettelun, se loisi turhaa kuormitusta käyttäjille, sillä uusia tapoja pitäisi olla opettelemassa toistuvasti muutamien vuosien välein. Tämä ei tarkoita, että vanhoihin tapoihin tulisi tästä syystä kuitenkaan täysin jumiutua. Uusia vuorovaikutustekniikoita tutkitaan ja kehitetään jatkuvasti, sillä

vakiintuneisiin tehtäviinkin voi löytyä tehokkaampia ratkaisuja. (Isokoski 2011, 175–176, 191.)

Tietokonepeleissä, joissa pelaajalla on liikuteltavanaan pelihahmo, on melko vakiintuneet liikituksen vuorovaikutustekniikat. On tyypillistä, että pelaaja liikkuu näppäimistön WASD-painikkeilla (W eteenpäin, A vasemmalle, S taaksepäin ja D oikealle) sekä hiiren liikutuksella (katseen suunta). Tätä vakiintunutta liikutusmekaniikkaa käytetään muun muassa peleissä Apex Legends (2019), PUBG Battlegrounds (2017), Fortnite (2017) ja Minecraft (2011). Videopelien kontekstissa käyttöliittymä on rajapinta (Kuva 1), joka yhdistää pelaajan ja pelin (Schell 2019, luku 15).



Kuva 1. Schellin (2019, luku 15) esitystä käyttöliittymän roolista mukaileva havainnollistava esitys.

Schell (2019, luku 5) esittää, että silmää miellyttävä ulkonäkö, käyttöliittymän sujuvuus ja huomaamattomuus ovat tärkeitä tekijöitä ja laadun mittareita, mutta hänen mukaansa kaikkein tärkein käyttöliittymän ominaisuus on antaa pelaajalle kokemuksestaan kontrollin tunne. Koska käyttöliittymän tehtävänä on pelaajan yhdistäminen pelin maailmaan, on sen merkitys pelaajakokemuksessa merkittävä (Schell 2019, luku 5). Käyttöliittymässä siis välitetään pelaajalle oleellista tietoa pelimaailmasta esimerkiksi mittarein ja osoittimin (Vuorela 2007, 58–59).

Miellyttävää pelaajakokemusta tavoiteltaessa käyttäjälähtöinen suunnittelu ja käytettävyyden eli pelattavuus ovat osa pelaajan huomioimista (Schell 2019, luku 2). Käyttäjälähtöisessä suunnittelussa ohjelmistokehittäjän tulisi ymmärtää käyttäjän pyrkimykset, tilanteet, tarpeet ja ymmärtämät käsitteet (Wiio 2004, 18). Burke (2014, luku 6) kirjoittaa samaan sävyyn. Pelillistettäessä kokemusta on suunnittelijalla oltava syvä käsitys kohdeyleisönsä, pelaajiensa, tavoitteista (goals) (Burke 2014, luku 6). Hän lisää, että pelaajien tavoitteet eivät aina ole rationaalisia tai helposti tunnistettavia ja pelaajien tavoitteet voivat vaihdella keskenään suuresti.

Käyttäjäystävällisyyttä kartoitettaessa keskitytään usein käyttöliittymään ja sen yksityiskohtien hiomiseen. Käytettävyyden on Wiion (2004, 19–21) mukaan käyttöliittymää laajempi asia. Hänen mukaansa on tietenkin tärkeää, että käyttöliittymän ominaisuudet ovat selkeitä, ymmärrettäviä, nopeita käyttää ja tekstit helposti luettavissa. Jo ennen käyttöliittymien suunnittelua kuitenkin tapahtuu paljon sellaista, mikä voi vaikuttaa hyvän käytettävyyden tavoitteluun. Systemin toiminnallisuutta ja arkkitehtuuria koskevat päätökset sanelevat ne puitteet, joiden sisällä käytettävyyttä tavoitellaan. (Wiio 2004, 23.)

Metsätuhot-peli on toteutettu Windows-käyttöjärjestelmällä varustetulla tietokoneella pelattavaksi videopeliksi, eikä se nykyisessä muodossaan toimi mobiililaitteilla. Peli ei ole yhteydessä verkkoon, vaan se on paikallisesti koneella suoritettava ohjelmisto. Tällaiset rajoitteet ovat hyvä esimerkki systeemitason käytettävyyteen vaikuttavasta asiasta. Tässä opinnäytetyössä ei

kuitenkaan pystytä tai pyritä ratkomaan kaikkia käytettävyyden aspekteja. On kuitenkin perusteltua tuoda esille, ettei käytettävyys ole pelkästään tarkasteltavan ohjelmiston tai sen käyttöliittymän harteilla, vaan siihen vaikuttavat myös itse ohjelman ulkopuoliset seikat (Wiio 2004, 20–21).

Wiio (2004, 29) määrittelee käytettävyyden tarkemmin seuraavasti: käytettävyydeltään hyvä eli käyttäjäystävällinen ohjelmisto on ymmärrettävä, vaivaton, kattava ja esteettisesti miellyttävä. Ymmärrettävän sovelluksen käyttäjälle on helppoa päätellä, kuinka hän pääsee haluamaansa lopputulokseen sovelluksen avulla. Ohjelmiston vaivattomuus mahdollistaa käyttäjälle tehtävistään mahdollisimman yksinkertaisella tavalla suoriutumisen (esimerkiksi ylimääräisiä klikkauksia säästämällä). Kattava sovellus tarjoaa kaikki ne toiminnot ja tiedot, joita käyttäjä tarvitsee hoitaakseen sen tilanteen tai tarpeen, johon sovellus on tarkoitettu. Ja viimeisimpänä, esteettisesti miellyttävä sovellus viestittää käyttäjälle laatua ja osaamista. Pahimmillaan epäesteettisyys voi viedä kaiken huomion käyttäjäkokemuksesta. (Wiio 2004, 29–31.)

Wiio (2004, 31) lisää, että helppo opittavuus ja käytön tehokkuus ovat myös käytettävyyden kehittämisen tavoitteita. Ne kuitenkin hänen mukaansa sisältyvät jo aiemmin mainittuihin. Jos sovellus on ymmärrettävä, se on myös helppo oppia. Tehokkuus vuorostaan on yksi vaivattomuuden ilmenemismuoto. (Wiio 2004, 31.)

Osa ymmärrettävää ja vaivatonta käyttöliittymää on sen antama palaute (Peters 2013, luku 8). Palautteen antamisella on hyvässä käyttöliittymässä, toimivassa pelissä ja tehokkaassa oppimisessa ratkaiseva tekijä (Peters 2013, luku 8). Palautteella ei tässä yhteydessä tarkoiteta pelkästään sanallista palautetta, kuten kehuja tai kritiikkiä. Palautteella käyttöliittymistä ja peleistä puhuttaessa voidaan tarkoittaa esimerkiksi merkkiääntä tai objektin korostamista ääriviivoilla pelaajan liikuttaessaan hiiren objektin kohdalle (Peters 2013, luku 8). Äänien ja visuaalisten palautteiden lisäksi palautetta voidaan myös välittää syöttölaitteiden kautta haptisena palautteena eli tuntoaistilla (Isokoski 2011, 187).

Schell (2019, luku 15) huomio, että pelaaja ja peli ovat palautteen antamisen ja saamisen kautta vuorovaikutuksen silmukassa (loop of interaction). Hän kuvailee vuorovaikutuksen silmukan syntyvän, kun peli antaa tietoa ja palautetta pelaajalle, pelaaja toimii tiedon perusteella ja samalla antaa pelille tietoa omasta toiminnastaan syötteidensä kautta (esimerkiksi pelaaja nostaa maasta peliobjektin), jolloin peli voi antaa uutta pelaajan toimista riippuvaa tietoa pelaajalle (esimerkiksi pelaajan nostama paljastuu aarrekartaksi). Pelin antamilla tiedoilla voi olla voimakas vaikutus siihen mitä pelaaja tekee seuraavaksi. Koko vuorovaikutuksen silmukan kulku voi vaikuttaa siihen, nauttiiko pelaaja pelistä ja ymmärtääkö hän, mitä pelissä tapahtuu. Pelisuunnittelijan on siis tärkeää tiedostaa kuinka paljon ja millaista palautetta missä vaiheessa peliä pelaajalle annetaan. (Schell 2019, luku 15.)

Berg ja Kojo (2011, 155–156) toteavat informaation visualisointia käsittelevässä tekstissään, että tiedon visualisointi, kuten kuvien ja diagrammien käyttö, mahdollistaa suuremman informaatiomäärän välittämisen käyttäjälle nopeammin ja vaivattomammin kuin vaihtoehtoiset esitysmuodot. Perusteluinaan he tuovat esiin, että: ”ihmisaisteista näkö tarjoaa laajimman informaatiokanavan niin aivojen tietoiselle päätöksenteolle kuin intuitiivisemmalle tiedonmuodostukselle”. Visualisoinnit siis helpottavat käyttäjää tarjoamalla tiedon yksinkertaisemmassa muodossa sekä tekevät näköaistin ja käsien kanssa tapahtuvasta yhteistyöstä (hand-eye-coordination) vuorovaikutuksessa käyttöliittymän kanssa intuitiivisempaa. Nämä seikat edistävät ohjelmiston käytettävyyttä. Berg ja Kojo nostavat myös esiin, että hyvässä visualisoinnissa on huomioitu muun muassa kompositio ja ryhmittely sekä käyttäjän huomion tarkoituksellinen ohjaaminen ja katseen kohdentaminen esimerkiksi korostuksilla. Terävän näön kapeuden luomien rajoitteiden tähden oleellisen tiedon löytämistä nopeuttaa tarkoituksenmukainen sijoittelu ja huomion ohjaaminen. (Berg ja Kojo 2011, 155–156, 164–167.)

Yksi käyttöliittymän tärkeistä tavoitteista on myös tiedon jakaminen käyttäjälle (Schell 2019, luku 15). Käyttöliittymäsuunnittelijan on muun muassa pohdittava seuraavia seikkoja: Onko olemassa tietoa, jota pelaaja tarvitsee edetäkseen, mutta jota hän ei saa vain pelimaailmaa katsomalla? Milloin pelaaja tarvitsee

kyseistä tietoa? Tarvitseeko hän sitä koko ajan vai vain välillä? Tarvitseeko hän kyseistä tietoa vain tietyissä tapauksissa? Miten tietoa voidaan jakaa pelaajalle ilman, että se häiritsee pelaajan muuta pelikokemusta? (Schell 2000, luku 15.)

Muun muassa näiden yllä lueteltujen ja muiden samankaltaisten kysymysten pohtiminen voi auttaa ymmärtämään käyttäjän tarpeita. Käyttäjän tarpeiden ymmärtäminen ja niihin vastaaminen ovat käyttöliittymäsuunnittelun ytimessä (Schell 2011, luku 15; Wiio 2004, 18; Peters 2013, luku 1). Metsätuhot-pelin tavoitteina ovat metsätuhoista oppiminen ja oppimisen motivoiminen. Käyttäjän tarpeet ja motiivit opintojakson suorittamiseen voivat vaihdella (lisää sisäisestä ja ulkoisesta motivaatiosta luvussa 2.4). Käyttäjän henkilökohtaisista motiiveista huolimatta, opetuspelin käyttäjä, eli pelaaja, on opiskelija. Jotta opiskelijan tarpeisiin käyttöliittymäsuunnittelun näkökulmasta voidaan vastata, on ymmärrettävä oppimista. Seuraavissa luvuissa tutustutaan oppimiseen, oppimistieteiden historiaan sekä nykYTEknologian tuomiin muutoksiin oppimisessa.

2.3 Kontekstina oppiminen

”Opetuksen kolme tärkeintä osatekijää ovat tavoite, sisältö ja menetelmä. Virtuaaliset välineet ja oppimisympäristöt muuttavat ehkä opetuksen menetelmiä, mutta eivät juurikaan tavoitteita tai sisältöä”. (Mäkitalo & Wallinheimo 2012, luku 2.3.)

Sitä, kuinka me ihmiset opimme ja sisäistämme uutta tietoa, on tutkittu 1800-luvulta saakka (Peters 2013, luku 2). Tätä tieteen alaa kutsutaan oppimistieteiksi (learning sciences). Koska kyseessä on kokonainen tieteenala, tietoa, teorioita ja koulukuntia on valtavasti. Olen pyrkinyt poimimaan ja tiivistämään mielestäni käyttöliittymäsuunnittelun ja pelillistämisen kannalta oleellisimmaksi kokemaani tietoa. Lähteenä vahvasti käyttämässäni Interface Design of Learning -kirjassaan Peters (2013, luku 2) esittelee mielestään tärkeimpinä pitämiään teorioita 1900- ja 2000-luvuilta keskittyen niihin, joita esiintyy eniten yritysten ja oppilaitosten eOppimisen strategioissa (lisää termistä eOppiminen luvussa 2.3.2).

On tärkeää huomioida, että oppiminen ei rajoitu kasvatukseen tai koulutukseen, vaikka tulevissa luvuissa käsitellään oppimista enemmän koulutuksellisessa fokuksessa. Elinikäisellä oppimiselle viitataan tällaiseen läpi elämämme kestävään tietojen ja taitojen kartoitukseen (Peters 2013, luku 2). Koulutuksen ja opetuksen ulkopuoliselle oppimiselle Peters määrittelee termin Informal learning, vapaasti suomennettuna vapaamuotoinen oppiminen. Vapaamuotoisia oppimista on esimerkiksi ystävien tai kollegoiden kesken keskustelussa omaksuttu uusi tieto. (Peters 2013, luku 2.)

2.3.1 Miten ymmärrämme oppimisen

Oppimistieteiden taipale alkoi eläinten käyttäytymisen tutkimisella 1800-luvun lopulla. Behavioristisessa ajattelussa pyrittiin mittaamaan, ennustamaan ja manipuloimaan käyttäytymismalleja. Ulkoisten ärsykkeiden aiheuttama suora toiminta oli tutkimuksen keskiössä, sillä toisin kuin mielen sisäisiä liikkeitä, toimintaa pystytään havaitsemaan. Behaviorismi loi ehdollistumisen käsitteen. Tunnettu esimerkki tästä on Ivan Pavlovin (1849–1936) tutkimuksissaan käyttämät koirat, jotka oppivat, että tietyn kellon soitto tarkoitti aterian saamista. (Peters 2013, luku 2.)

Toinen hyvin varhainen oppimistieteiden alalla tunnistettu oppimiseen vaikuttava ilmiö on vahvistus ja rangaistus (reinforcement and punishment). Oppijaa palkitaan vahvistuksena hyvistä suorituksista ja oikeista vastauksista, kun taas kehnosta suorituksesta ja vääristä vastauksista saa rankaisuna sanktioita. Peters (2013, luku 2) avaa vahvistamisen ja rangaistuksen tarkoituksia kyseisessä yhteydessä. Teorian mukaan on sekä positiivista että negatiivista vahvistamista ja sekä positiivista että negatiivista rankaisemista. Positiivisella ja negatiivisella ei tässä tarkoiteta niiden miellyttävyyttä, vaan sitä lisätäänkö oppimiskokemukseen jotain vai otetaanko jotain pois. (Peters 2013, luku 2.)

Vahvistus ja rankaisu on yleinen strategia opetuksessa edelleen, ja eritoten peleissä. Pelit ja pelillistetty oppiminen suorastaan pursuavat palkintoja.

Positiivisen vahvistamisen keinoina käytetään kunniamerkkejä, pisteitä, tasoja ja tulostauluja. Peters (2013, luku 2) huomauttaa, että vahvistuksen ja rankaisemisen teoria sekä behaviorismi ovat saaneet kritiikkiä siitä, että ne ovat yksinkertaistettuja ja usein vain eläinkokeilla täysin toimiviksi todennettuja ilmiöitä. Ihmisen oppiminen on monitasoisempaa ja syvempää. Kritiikin kohteena on myös ulkoisten palkintojen antama motivaatio. Motivaatio on olennainen osa oppimista, mutta parhaimmillaan se toimii, kun motivaatio on sisäistä. Sisäisellä motivaatiolla tarkoitetaan, että oppija haluaa itse aktiivisesti ymmärtää opiskelemissään kokonaisuuksia. Palkintoihin nojaava motivaatio on ulkoista. Oppija voi motivoitua keräämään palkintoja ja saada suorituksiaan näyttämään hyvältä, kuitenkin oppimatta syvemmin laisinkaan. (Peters 2021, luku 2, luku 7.)

Kognitivismien teoria alkoi kehittyä 1950-luvulla vastauksena behaviorismin saamaan kritiikkiin. Behaviorismissa tärkeänä pidettiin toimintaa ja muita helposti mitattavissa olevia ilmiöitä, kognitivismissa katse käännetään mieleen ja ajatuksiin. Kognitivismien teorian syntyä siivitti uusi nouseva tieteenala tietojenkäsittelytiede (computer science). Kognitivismien perusajatuksena onkin, että ihmisen aivoja tarkastellaan tietokoneen kaltaisena tietoa ja tehtäviä käsittelevänä yksikkönä. (Peters 2013, luku 2.)

Kuten tietokoneella, aivoillamme on rajoitteensa muistissa. Ihmisen työskentelymuistin kapasiteetti on n. 7 (+2) yksikköä (Peters 2013, luku 2; Isokoski 2011, 182). Yksiköiden ollessa yleensä joko akustinen tai visuaalinen muistikuva (Isokoski 2011, 182).

Työskentelymuistimme rajoitteita kuvaamiseen voidaan käyttää käsitettä kognitiivinen kuormitus. Koska ihmisten työskentelymuistilla on rajansa, on hyvä pyrkiä vähentämään ylimääräistä kognitiivista kuormitusta. Huonon suunnittelun tai käytetyn teknologian vuoksi kuormitusta lisääviä tekijöitä voivat olla esimerkiksi vaikeasti luettava teksti tai huonosti kirjoitetut ohjeet. Kognitiiviseen kuormitukseen vaikuttaa myös se, mitä kautta saamme tietoa. Prosessoimme visuaalisesti ja auditiivisesti saamaamme tietoa eri kanavien kautta, ja kykymme

vastaanottaa tietoa yhtäaikaisesti molempien kanavien kautta on rajallista.
(Peters 2013, luku 2.)

Gagnén yhdeksänosainen tapahtumaketju kuvaa kuinka opetuksen tarjoaja voi tukea uuden opitun asian siirtämistä oppijan muistiin.

1. Oppijan huomion saaminen.
2. Tavoitteiden kuvaaminen oppijalle.
3. Aiemmasta opitusta tiedosta muistikuvien nostaminen.
4. Uuden tiedon sisällön esittely.
5. Opastuksen tarjoaminen oppimista varten.
6. Mahdollista ja motivoi harjoittelemaan.
7. Anna palautetta.
8. Arvioi oppijan suoritusta.
9. Auta oppijaa uuden tiedon siirtämisessä osaksi pysyvää muistia ja opin siirtämistä uusiin käytäntöihin.

Peters (2013, luku 2.)

Ihmisen oppimisen ja aivojen toiminnan vertaaminen tietokoneeseen ja koneelliseen tietojenkäsittelyyn vie kysymyksen äärelle, kuinka oppimisen luonne nähdään. Kysymyksessä on kaksi vastakkaista näkökulmaa, objektivismi ja konstruktivismi. Objektivismissa tieto maailmasta nähdään objektiivisena. Oppiessa tieto siirtyy oppijan aivoihin. Opettajan rooli tässä on toimia tiedon välittäjänä. Konstruktivismissa taas oppija rakentaa tiedon maailmasta omissa aivoissaan ja opettajan tehtävänä on oppimisen mahdollistaminen ja helpottaminen. Kuten kognitivismi nousi vastaamaan behaviorismin puutteisiin, nousi konstruktivismi kognitivismin kritiikistä. (Peters 2013, luku 2.)

Termiä oppimistieteet (learning sciences) alettiin käyttää 1980-luvun lopulla. Nykyaikainen käsitys oppimisesta on monisyinen. Oppimistieteet ovat

yhdistelmä muun muassa kasvatopsykologiaa, antropologiaa, sosiologiaa, tietojenkäsittely-, informaatio-, neuro- ja käyttäytymistieteitä. Oppimistieteiden taustalla on sekä kognitiivista että konstruktivistista teoriaa, mutta pääpaino on siinä, että oppiminen tapahtuu kompleksisessä todellisessa maailmassa, joten sitä ei voi tutkia vain kontrolloiduissa laboratorio-olosuhteissa. (Peters 2013, luku 2.)

Kurhila (2011, 272) tiivistää asian näin: ”Yksinkertaisimmillaan oppiminen tarkoittaa yksilön ajattelun tai toiminnan muutosta. Kyseinen muutos voi olla haluttu, ja se voi olla tiedostettu tai tiedostamaton. Opettaminen tarkoittaa tämän muutoksen eli oppimisen edesauttamista. Laadukkainkaan opetus ei silti takaa oppimista, sillä oppiminen vaatii oppijan omaa toimintaa, esimerkiksi ajattelua, lukemista, katselemista tai kuuntelemista. Tähän oppijan toimintaan opetuksella yritetään ja voidaanakin vaikuttaa.”

2.3.2 Digitalisaation tuoma muutos koulutukseen

Digitalisaatio on muuttanut elämäämme niin radikaalisti, että sen tuomia muutoksia on syytä pohtia oppimistamme tarkasteltaessa (Peters 2013, luku 2). Peters (2013) käyttää kirjassaan sanaa eLearning viitattaessa kaikkiin teknologian tukemiin oppimismuotoihin. Termiä e-oppiminen tai eOppiminen käytetään myös suomeksi (Suomen eOppimiskeskus RY 2023). Kurhila (2011, 271) käyttää samasta käsitteestä lyhennettä OOT (oppimis- ja opetusteknologia).

E-oppimisen alle kuuluva online-oppiminen (eng. online learning) kattaa Petersin (2013, luku 3) mukaan kaikki oppimisen muodot, joissa verkkoyhteys on mukana oppimisprosessissa. Hän listaa neljä tyypillistä online-oppimisen muotoa: verkkoyhteisössä oppiminen (online collaborative learning, OCL), etäopetus (online distance education), verkkokurssit eli virtuaalikurssit (online-course ware, OC) ja avoimet massaverkkokurssit (massive open online courses, MOOC).

Mäkitalo ja Wallinheimo (2012, luku 1) kuvaavat oppimista vuorovaikutteisena tapahtumaketjuna. Heidän mukaansa opettajan ja kouluttajan roolina tässä oppimisen tapahtumaketjussa on käyttää asiantuntijuuttaan luodakseen oppimiselle parhaat edellytykset. He kirjoittivat jo kymmenen vuotta sitten julkaistussa teoksessaan, että rajat ”perinteisen” ja verkko-, etä- ja virtuaaliopetuksen välillä ovat sekoittuneet. Välineitä, joilla verkko-, etä- ja virtuaaliopetusta tarjotaan he kutsuvat virtuaalisiksi oppimisympäristöiksi (Mäkitalo ja Wallinheimo 2012, luku 1). Kurhila (2011, 271) käyttää termiä oppimisympäristöt ilman etuliitteitä. Hänen sanoin: ”oppimisympäristöllä tarkoitetaan kaikkia niitä välineitä, tiloja ja osallistujia, jotka ovat mukana opetuksessa, opiskelussa ja oppimisessa”.

Virtuaaliset oppimisympäristöt ja vuorovaikutustilanteet voivat antaa opiskelijalle mahdollisuuden muokata oppimisympäristöään omien mieltymysten tai tarpeiden mukaan (Mäkitalo ja Wallinheimo 2012, luku 1). Etäopetuksen tarjoaminen mahdollistaa myös erilaisten ja erilaisissa elämäntilanteissa olevien ihmisten osallistumisen kursseille tai koulutukseen laajemmin, kun opetus ei ole sidottu aikaan tai paikkaan (Mäkitalo ja Wallinheimo 2012, luku 1). Myöskään opetusvideolta oppiminen ei ole sidottu aikaan tai paikkaan (Peters 2013, luku 8). Videoiden ja animaatioiden käytöstä virtuaaliopetuksessa Peters korostaa muina hyvinä puolina oppijan mahdollisuuden mukauttaa tahtia omiin tarpeisiin. Videota (myös tallennettua verkkoluentoa) voi pysäyttää, kelata, nopeuttaa ja mahdollisesti siihen voi lisätä tekstityksiä. Näiden hyötyjen lisäksi Kurhila (2011, 271–272) nostaa tietoverkkojen ja -varastojen lisäävän oppimisprosessin osapuolien käytettävissä olevien resurssien määrää ja vuorovaikutuksen tiheyttä. Hänkin alleviivaa vuorovaikutuksen tärkeyttä oppimisessa.

Kurhila (2011, 274) kutsuu oppimiseen tarkoitettua ohjelmistoa opetusohjelmaksi. Opetusohjelmat voidaan jaotella harjaannuttamisohjelmiin, perehdyttämisohjelmiin, peleihin ja simulaatioihin. Kurhila kirjoittaa, kuinka opetusohjelmien tuomaa hyötyä opetuksen tehostumiseen on tutkittu ennen ja jälkeen -kokeilla. Hänen mukaansa tutkimus- ja kehitystyössä opetusohjelman on ajateltu itsessään olevan arvokas, vaikkei oppimisen tehostumista

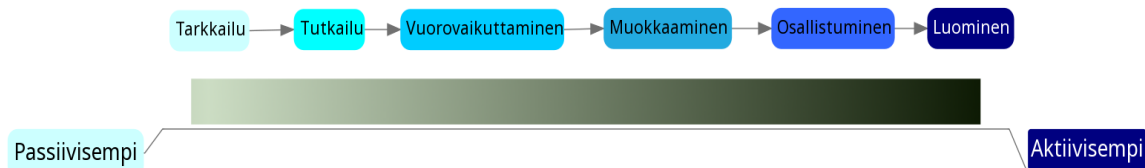
pystyttäisikään todistamaan. Opetusohjelmat voivat helpottaa opetusmateriaalin läpikäymistä ja joissain tapauksissa materiaalin yksilöllistämistä. Muina arvokkaina piirteinä pidetään opetusohjelmien väsymättömyyttä, niiden mahdollisuutta antaa toistuvasti, runsaasti ja tarkasti palautetta, mahdollisuutta opiskella ajasta ja paikasta riippumatta sekä harvinaisten tai vaarallisten tilanteiden harjoittelun tarjoaminen (Kurhila 2011, 275). Myös Peters (2013, luku 2) tuo esiin teknologioiden mahdollistamat realistiset harjoitukset virtuaalisessa maailmassa. Sekä Kurhila että Peters mainitsevat esimerkkeinä virtuaalisista harjoituksista terveydenhuollon opintojen haastavat tilanteet, kuten potilaan diagnosointi (Kurhila 2011, 276) ja kirurgiset operaatiot (Peters 2013, luku 2).

Burke (2014, luku 1) kirjoittaa, että meillä ihmisillä on sisäsyntyinen tarve parantaa itseämme ja elämäämme, mutta useimmiten motivaation puute estää ensimmäiset askeleet kehittymistä kohti. Hänen mukaansa pelillistäminen voi auttaa motivoinnissa matalan kynnyksen perehdytyksen ja positiivisen palautteen avulla. Seuraavassa luvussa tutustumme paremmin pelillistämiseen ja sen mahdollisuuksiin oppimisen tukemisessa.

2.4 Oppimisen pelillistäminen

Pelit, mallinnetut maailmat ja simulaatiot voivat toimia virtuaalisina oppimisympäristöinä (virtual learning environments, VLEs) (Peters 2013, luku 8). Peters esittää kirjassaan *Interface Design for Learning* janan, millaista toiminnallisuutta verkko- ja virtuaalioppimisympäristöt voivat mahdollistaa opiskelijalle (mukaelma janasta Kuva 2). Janalla vasemmalta oikealle siirryttäessä, oppijan rooli muuttuu passiivisemmasta tarkkailijasta yhä aktiivisemmaksi toimijaksi. Esimerkkinä Peters toteaa, että opetusvideo sijoittuisi janalla täysin vasemmalle, kun taas verkkokurssi, jossa vaikuttaminen ja sosiaalinen kanssakäyminen ovat mahdollisia, olisi jo huomattavasti enemmän oikealla (Peters 2013, luku 8). Hänen mukaansa videopelit antavat parhaimmillaan oppijalle motivoivan ja aktiivisen roolin omaan oppimiseensa. Myös Kurhila (2011, 275) kirjoittaa virtuaalisten oppimisympäristöjen ja oppijan vuorovaikutuksen määrästä ja laadusta sekä mahdollisuuksista opiskelijan aktivointiin. Esimerkkeinään hän vertaa e-oppikirjan sivujen verkkaista

kääntelyä (jonka voisi sijoittaa janalla mahdollisesti tutkailun kohdalle) ja nopeatempoista oppimispelejä.



Kuva 2. Mukaelma oppijan aktiivisuutta kuvaavasta janasta (Peters 2013, luku 8).

Pelillistämässä (gamification) on kyse ihmisten motivoinnista ja tavoitteiden saavuttamisen tukemisessa pelimekaanisin ja kokemuskeskeisen suunnittelun keinoin (Burke 2014). Pelimekaaniset keinot voivat olla esimerkiksi luvussa 2.1 mainittuja Vuorelan (2007, 26–30) esittämiä pelikoukkuja. Burke (2014) korostaa, ettei pelillistämässä ole kyse vain tehtävien ja toimintojen esittämisestä videopeliltä näyttävässä muodossa, vaan kyseessä on motivoinnin työkalu.

Sekä Kurhila (2011, 280), Peters (2013, luku 7), Schell (2019, luku 11) että Burke (2014, luku 7) toteavat, että parhaimmillaan pelit ovat erittäin motivoivia. Motivaatio on tärkeää oppimisen kannalta (Burke 2014, luku 4). Mutta ymmärtääksemme kuinka motivoida oppijaa, on ymmärrettävä mistä oppijan motivaatio kumpuaa. Sisäisellä motivaatiolla tarkoitetaan itsestä kumpuavaa halua ja kiinnostusta tehdä jotain vapaaehtoisesti, koska se tuntuu tärkeältä, mielekkäältä tai kiinnostavalta (Peters 2013, luku 7, Schell 2019, luku 11). Ulkoinen motivaatio taas tulee itsen ulkopuolelta, kuten siitä, että asian tekemisestä maksetaan tai sen tekemiseen on velvollisuus (Peters 2013, luku 7, Schell 2019, luku 11). On kuitenkin huomioitava, että usein sisäinen ja ulkoinen motivaatio sekoittuvat toisiinsa (Schell 2019, luku 11).

Peters jatkaa (2013, luku 7), että motivaatioiden luonteen lisäksi tärkeää on itsemääräämisoikeus. Hän esittelee, että oppimisen kannalta sisäistä motivaatiota on yleisesti pidetty ihanteellisempänä kuin ulkoista motivaatiota. Sisäinen motivaatio yhdistetään hyviin oppimistuloksiin, osaamisen tunteeseen, sinnikkyuteen ja luovuuteen. Petersin mukaan totuus on, että on olemassa tylsiä, epämieluisia ja puuduttavia tehtäviä, joihin on vaikea motivoida ketään

sisäisesti. Siksi itsemääräämisoikeudella on merkitystä. Peters käyttää esimerkkinään lääkäriopiskelijaa, jonka on opetettava tenttiä varten ulkoa lista farmaseuttisia ainesosia. Ulkoa opetteluun motivaatio on ulkoista ja tehtävä väsyttävää, mutta sen tekemistä auttaa opiskelijan itse valittu urasuunta ja -tavoitteet, jotka kumpuavat sisäisestä motivaatiosta. Toisena esimerkkinä itsemääräämisoikeudesta Peters mainitsee, että puuduttavassakin tavaroiden lajittelutehtävässä lajittelija voi motivoitua, jos hän saa itse valita, kuinka hän sen tekee. (Peters 2013, luku 7.)

Kurhilan (2011, 280) mukaan pelien käyttö opetuksessa on kiehtonut opetusteknologioiden tutkijoita juuri siitä näkökulmasta, että peleihin yhdistetään korkea sisäinen motivaatio, joka haluttaisiin valjastaa oppimiseen. Hän kuitenkin jatkaa, että nämä pelien motivoivat elementit ovat osoittautuneet vaikeasti hyödynnettäviksi opetuskäytössä (Kurhila 2011, 280).

Burke (2014, luku 4) avaa tärkeitä keinoja oppimisen pelillistämiseen:

- Päämäärän määrittely. Lopullisen päämäärän ja siihen johtavien välietappien esittämisellä on ratkaiseva vaikutus oppimiskokemuksen onnistumiseen.
- Kokonaisuuden pilkkominen palasiksi ja välietapeiksi. Pelaajalle ei tule kaataa kaikkea tietoa kerralla. Pienet askeleet ja kokemus edistymisestä auttavat motivoimaan, esimerkiksi eri tasolta toiselle siirtyminen tai kunniamerkin saavuttaminen pelissä ovat tällaisia motivoinnin lähteitä.
- Tietoaukkojen kartoittaminen. Oppiminen on prosessi, jossa uusi tieto yhdistetään jo aiemmin tiedettyyn asiaan. Oppimisen kannalta on tärkeää ymmärtää kokonaisuus, joten oppimisentarjoajan on pidettävä huolta, ettei annettuun tietoon jää aukkoja.
- Teorian yhdistäminen koukuttavaan toistoon. Pelillistämisessä voidaan käyttää keinona silmukkamaista toistoon perustuvaa iteraatiota. Annetaan pelaajalle opastusta, haaste ja palautetta, kuinka pelaajan yritykset suorittaa haaste sujuu. Kun haaste on suoritettu, alkaa silmukka alusta uuden tiedon ja haasteen avulla.

- Mahdollisuus kommunikoida, tehdä ryhmätyötä, vertaisoppia ja keskustella mentoreiden kanssa. Oppimisen sosiaalisten aspektien huomioiminen hyödyttää oppimisen pelillistämistä.
- Onnistumisten palkitseminen. Oppijan saavutusten tunnustaminen ja virstanpylväiden saavuttamisen juhlistaminen on tärkeää oppijan motivaation ylläpitämiseksi.

(Burke 2014, luku 4.)

Peters (2013) esittää Burken kanssa osittain samoja näkemyksiä, mutta hän lisää oppimisen pelillistämisen keinoihin vielä seuraavat seikat:

- Turhien tehtävien minimointi ja automaatio. Opiskelijan ei pitäisi joutua käyttämään liiaksi aikaansa pelissä tehtäviin, jotka eivät edistä oppimista (Peters 2013, luku 11).
- Erilaisten formaattien ja medioiden käyttö, eli tietoa jaetaan monipuolisemmin kuin esimerkiksi vain tekstimuodossa (Peters 2013, luku 11).
- Erilaisten oppilaiden tunnistaminen. Opiskelijoiden erilaiset taustat, kulttuurit, haasteet ja motivaation syyt tai sen puutteet ovat kaikki huomionarvoisia seikkoja (Peters 2013, luku 11).
- Sosiaalinen kanssakäymisen ja ryhmätoimintaan osallistumisen on todettu lisäävän tuottavuutta, tyytyväisyyttä, parantavan oppimistulokisa ja vähentävän opiskelun kesken jättämistä (Peters 2013, luku 6).
- Prosessoimme visuaalisesti ja auditiivisesti saamaamme tietoa eri kanavien kautta, ja kykymme vastaanottaa tietoa yhtäaikaaisesti molempien kanavien kautta on rajallista (Peters 2013, luku 2).
- Kun oppijan tarvitsee kuvailla opittua omin sanoin, on hänen yhdistettävä opittu tieto aiempaan osaamiseensa ja todellisen maailman konteksteihin verrattuna ulkoa opettelemiseen ja ulkoa opitun teorian toistamiseen (Peters 2013, luku 2).
- Todellisen maailman kontekstin merkitys ja oppimisen sijoittaminen siihen autenttiseen tilanteeseen, johon se liittyy, pelkän teorian sijasta (situated learning) (Peters 2013, luku 2).

- Pitkien kirjoitettujen ohjeistusten välttäminen. Esimerkiksi kontekstia ja visualisointeja yhdistämällä voidaan käyttäjälle jakaa tietoa pelistä (kuten ohjausmekaniikoista) tekstiohjeiden sijasta. (Peters 2013, luku 10.)
- Minimalistinen suunnittelu ja ylimääräisyyksien välttäminen (Peters 2013, luku 10).
- Oppijan edistymisen visualisointi (Peters 2013, luku 10).

Kurhila esittelee pelin elementtejä, jotka ovat tärkeitä onnistuneelle opetuspelille:

- Kiinnostava tarina, jossa on draaman kaari. Yksinkertainenkin tarinallisuus voi riittää.
- Alhainen oppimiskynnys, mutta vaiheittain etenevä vaikeustaso. Pelin on tarjottava pelaajalle haastetta, jotta pelaaja joutuu analysoimaan omaa pelistrategiaansa.
- Käyttöliittymän intuitiivisuus. Helposti omaksuttavat vuorovaikutustekniikat sekä tarvittava visuaalinen ja auditiivinen informaatio.
- Johdonmukainen pelimaailma, jossa on omat sääntönsä eivätkä säännöt muutu perustelematta kesken pelin. Pelaaja pystyy vaikuttamaan pelimaailmaan.
- Toiminnasta saatavan palautteen pitää olla selvää, vaikka se olisikin epäsuoraa palautetta.

(Kurhila 2011, 280 mukaan Dumbleton 2007.)

Kurhila (2011, 281) huomauttaa, että oppimispelit ja pelien käyttö opetuksessa eivät ole osoittautuneet kaikille sopivaksi oppimistavaksi. Kaikki eivät pidä peleistä, eivätkä myöskään motivoitu pelaamaan. Kurhilaan mukaan ei siksi ole realistista, että pelejä voisi kattavasti hyödyntää oppimisessa edes motivaation lisääjänä.

2.5 Käyttöliittymäsuunnittelun käytänteitä oppimisen tueksi

Uuden oppiminen voi olla hidasta, työlästä ja kuormittavaa (Peters 2013, luku 4). Käyttöliittymäsuunnittelijan on pidettävä huolta, että käyttöliittymä itsessään ei lisää opiskelijan kognitiivista kuormitusta (Peters 2013, luku 4; Kurhila 2011, 271–271). Opiskelijan huomion kiinnittymistä liiaksi oppimisen kannalta epäolennaisiin asioihin kuten työvälineen (oppimisympäristön) ominaisuuksiin tulisi pyrkiä välttämään (Kurhila 2011, 272).

Käyttöliittymän ei tulisi olla liian monimutkainen pelin alussa. Käyttöliittymän monipuolisuutta on hyvä esitellä käyttäjälle asteittain. Oppijalle esitellään lisää vaihtoehtoja, rajoja ja sääntöjä sitä mukaa, kun oppijalle on annettu aikaa ja mahdollisuus tutustua rauhassa sen hetkiseen tilanteeseen. Liian monimutkaiselta tuntuva käyttöliittymä heti alkuun esiteltynä kuormittaa käyttäjää, ja voi pahimmillaan tuottaa ahdistusta jopa niin, ettei oppija tahdo käyttää ohjelmistoa. (Peters 2013, luku 8.)

Sijoittelu on tärkeä osa kaikkea käyttöliittymäsuunnittelua. Peters (2013, luku 10) esittelee hyvän sijoittelun edellytyksiä oppimisen näkökulmasta. Hänen mukaansa on huomioitava toisiinsa liittyvien tekstien ja visualisoitujen materiaalien (kuten kuvat ja diagrammit) lähekkäisyys. Tutkimusten mukaan tekstin ja havainnollistavan kuvan lähekkäin sijoittelu parantaa oppimista. Lisäksi tehtävissä, joissa annetaan käyttäjälle lyhyttä sanallista palautetta esimerkiksi monivalintatehtävän yhteydessä jaettu oikea vastaus, tulisi palautteen olla valitun vastauksen perässä, ei kaikkien vastausvaihtoehtojen alla. (Peters 2013, luku 10.)

Peters (2013, luku 8) esittää, että palautteen antaminen on erityisen tärkeää oppimisympäristöissä. Tapauksissa, joissa tehtävään tai kysymykseen on oikea ja väärä vastausvaihtoehto, ei riitä, että käyttäjälle ilmoitetaan, oliko hänen vastauksensa oikein vai väärin. Hänelle on lisäksi todistettava perusteluja, miksi vastaus oli oikein tai väärin. Palautetta on pyrittävä antamaan jatkuvasti ja monissa muodoissa. Jatkuvan palautteen lisäksi olisi pyrittävä antamaan palautetta yhteenvetoina, esimerkiksi välitappien ja koko suorituksen lopuksi.

Kuten on jo todettu (kappaleessa 2.2) palautteen jakamiseksi käyttäjälle on monia keinoja. Peters tuo esiin sen, että pelit ja virtuaaliympäristöt mahdollistavat seurausten näyttämisen ja toimivat hyödyllisinä palautteen antamisen kanavina. Pelissä esimerkiksi pieleen mennyt kemiallinen sekoitus voidaan räjäyttää turvallisesti, mutta opettavaisesti. (Peters 2013, luku 10.)

Sekä Schellin (2019, luku 15), Petersin (2013, luku 8) että Mäkitalon ja Wallinheimon (2012, luku 1) mukaan käyttöliittymän muokattavuus ja kontrollin antaminen käyttäjälle omasta oppimis- ja/tai pelaajakokemuksestaan ovat hyvin suunnitellun ohjelmiston tunnusmerkkejä. Käyttöliittymän muokattavuus tukee myös saavutettavuuden periaatteita, joista tarkemmin seuraavassa luvussa.

2.6 Saavutettavuus

Vaikka tämän opinnäytetyön keskiössä tai tavoitteena ei suoranaisesti ole saavutettavuuden arviointi, on saavutettavuutta kuitenkin syytä tarkastella ohjelmiston käytettävyyden ja oppimisen konteksteissa. Suomessa on voimassa yhdenvertaisuuslaki (1325/2014), johon on kirjattu opiskelijan oikeus kohtuullisiin mukautuksiin opetuksessa (15. §).

Pesonen ja Nieminen (2021, 72) kuvailevat saavutettavaa opetusta opetuksiksi, ”joka tarjoaa jokaiselle opiskelijalle hänen tarvitsemansa yksilölliset järjestelyt”. Jotta opetus voisi olla oikeasti inklusiivista, tulisi oppimisen haasteita ymmärtää niin medikaalisen eli lääketieteellisen mallin kuin myös sosiaalisen mallin näkökulmista. Medikaalisessa mallissa keskitytään opiskelijan diagnosoitaviin piirteisiin, joiden aiheuttamiin tarpeisiin tulisi oppilaitoksen tarjota yksilöllisiä tarpeita. (Pesonen ja Nieminen 2021, 72.)

Sosiaalisessa mallissa oppimisen haasteet käännetään opetuksen rakenteisiin. Yksilöitä ei pitäisi pyrkiä muuttamaan ja auttamaan sopeutumaan ylhäältäpäin annettuun normiin. Oppimisen ympäristöt ja toteutukset tulisi suunnitella lähtökohtaisesti niin inklusiivisesti, ettei yksilöllisille järjestelyille ole tarvetta. Toisin sanoen, tällöin opetus olisi hyvin saavutettavaa. (Pesonen ja Nieminen 2021, 66–67.)

Saavutettavuudella siis viitataan digipalveluiden, verkkosivujen ja mobiilisovelluksien esteettömyyteen. Saavutettavuudessa on huomioitava tekninen saavutettavuus, helppokäyttöisyys ja ymmärrettävyys. Näin voidaan varmistaa, että mahdollisimman monet erilaiset ihmiset pystyvät käyttämään palvelua tai tuotetta. Saavutettavuus lisää yhdenvertaisuutta.

Tekninen toteutus tukee muun muassa erilaisia päätelaitteita ja avustavia teknologioita, kuten puheohjausta ja ruudunlukuohjelmia. Helppokäyttöinen digipalvelu on helppo hahmottaa, mahdollisten sivujen nimien ja symbolien tulisi olla kuvaavia ja selkeitä sekä haluttu toiminta on vaivatonta suorittaa.

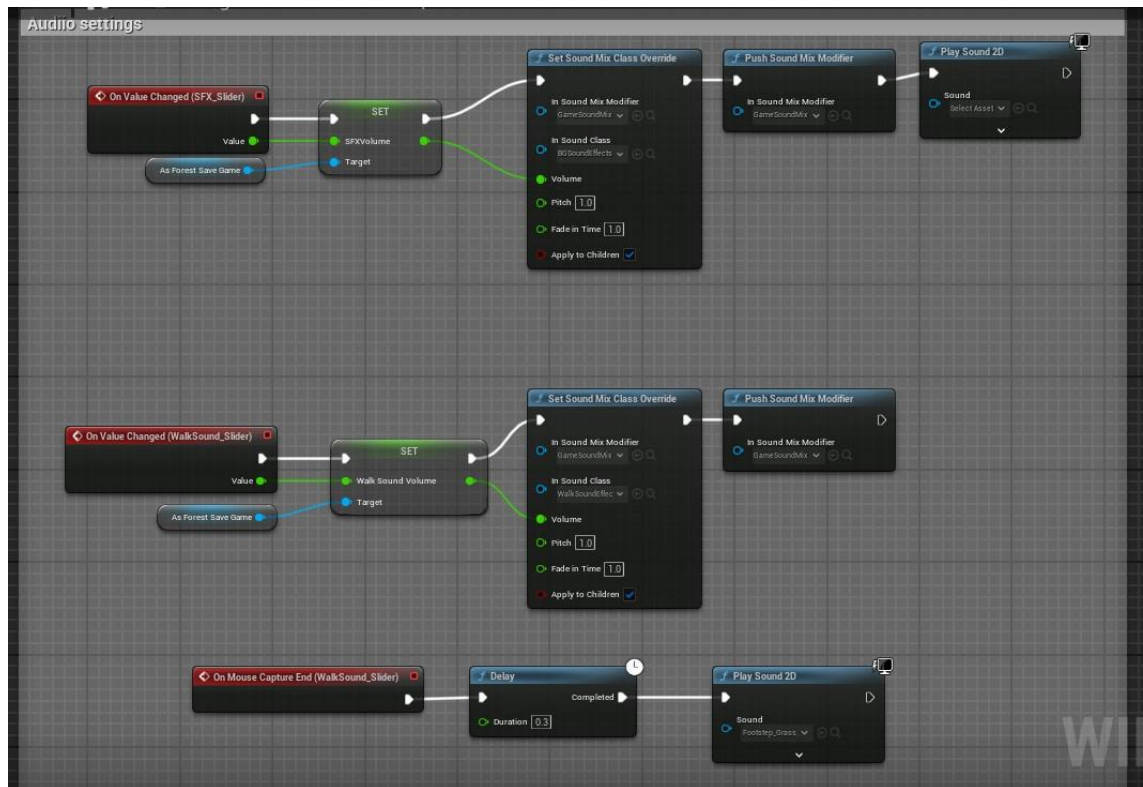
Ymmärrettävyyteen liittyviä seikkoja on esimerkiksi selkeän, ymmärrettävän kielen käyttö ja selkokielen optio. Sisältöä tarjotaan monikanavaisesti, eli useilla eri tavoilla, kuvilla, äänenä, videoina ja tekstinä. Tässä raportissa käsiteltiin aiemmin käytettävyyttä (kohta 2.3 Käyttöliittymäsuunnittelu alla). Nämä kaksi risteävät useassa kohtaa. Saavutettavuutta kuten käytettävyyttä on syytä suunnitella käyttäjälähtöisesti. Käyttäjän tilanteiden, tarpeiden ja mahdollisten rajoitteiden sekä haasteiden huomioiminen on osa hyvää suunnittelua. Suunnittele kaikille -periaate on nimensä mukaisesti suunnittelua, jonka tarkoituksena on huomioida mahdollisimman laajasti erilaiset käyttäjät alusta asti. (Etelä-Suomen Aluehallintovirasto, 2023.)

3 Menetelmät

3.1 Metsätuhot-pelin kehitysympäristöt ja lähtökohdat

Metsätuhot-peliprojektin tavoitteena on pelillistä Karelia-ammattikorkeakoulun tarjoama Metsätuhot-opintojakso. Pelillistämistä on Metsätuhot-projektissa lähdetty kehittämään Unreal Engine 5 (lyh. UE5) -pelimoottorilla. Unreal Engine -ohjelmistosta on käytössä versio 5.1.1. Kyseinen pelimoottori ja versio valikoituivat opinnäytetyöhön sen vuoksi, että näillä on kehitetty Metsätuhot-peliä jo keväällä 2023. Pelimoottorilla viitataan kehitysympäristöön eli tietokonesovellukseen, joka on kehitetty erityisesti pelien tuottamiseen. Pelimoottori, kuten Unreal Engine 5, tarjoaa käyttäjälleen valmiiksi luotuja moduuleja, tiedostoja ja tukea, jolloin pelikehittäjän ei tarvitse ohjelmoida kaikkea alusta alkaen itse. Pelikehittäjä pystyy sekä luomaan täysin uusia että muokkaamaan pelimoottorin valmiiksi tarjoamia ominaisuuksia projektissaan.

UE5 tukee kahta eri ohjelmointitapaa, Blueprint-ohjelmointia (Kuva 3) ja C++-kieltä (Kuva 4). (Conçalo, Sherry, Pereira & Fozi 2022, luku 1.)



Kuva 3. Esimerkki Blueprint-ohjelmoinnista.

C++-ohjelmointiympäristönä projektissa toimii Visual Studio 2022. Unreal Engine on suunniteltu toimimaan sulavasti juuri Visual Studion kanssa (Epic Game, Inc 2024d), jonka vuoksi tässäkin projektissa käytetään Visual Studiota. Visuaalisen skriptauskielen lisäksi Blueprint-termiä käytetään Unreal Engine 5:ssä myös pelimoottorin valmiiksi tarjoamista Blueprint-luokista (Blueprint class) ja Blueprint-ominaisuuksista (Blueprint asset) (Conçalo ym. 2022, luku 1).

```

// Summary
void UForestDestructionGameInstance::SaveGame()
{
    // Create the save game object
    UForestSaveGame* ForestSaveGame = Cast<UForestSaveGame>(UGameplayStatics::CreateSaveGameObject(UForestSaveGame::StaticClass()));
    // Save variables
    ForestSaveGame->MapName = UGameInstance::GetWorld()->GetName();
    ForestSaveGame->DateStart = DateStart;
    ForestSaveGame->PlayerName = PlayerName;
    ForestSaveGame->StudentId = StudentId;
    // Save the responses by interaction by map into
    // the map container using structures.
    // First get the map names
    TArray<FString> MapNames;
    ResponsesByInteractionsByMap.GetKeys(MapNames);
    // Then go through each map
    for (FString MapName : MapNames) {
        // Get the interaction as names from the map
    }
}

```

Kuva 4. Esimerkki C++-ohjelmoinnista.

Metsätuhot-pelin kyseiselle opintojaksolle ilmoittautuneet opiskelijat saavat ladattavakseen omalle tai oppilaitoksen tietokoneelle kurssin Moodlesta.

Moodle on todellisen elämän opetustilanteita ja yhteisöllistä tiedonrakentelua jäljittelevä oppimisympäristö (Mäkitalo & Wallinheimo 2012, 2.1). Metsätuhotkurssin opiskelijat saavat lisäksi Moodlen kautta käyttöohjeet pelin lataamiseen ja pelaamiseen. Metsätuhot-peli liittyy siis opintojaksoon ja on ladattavissa nonstop-verkkokurssilta, eli se on nykyisessä muodossaan oma paikallinen ohjelmistonsa eikä Metsätuhot-peli ole yhteydessä verkkoon. Metsätuhot-pelin voisi mielestäni näillä perusteilla määritellä pelillistetyn oppimisen lisäksi virtuaalioppimiseksi.

3.2 Muut työkalut

Versionhallinnan ylläpitoon käytin Git:iä Azure DevOpsin projektinhallintaympäristössä. Nämä valikoituivat työkaluiksi, koska nämä olivat käytössä myös keväällä 2023. Pystyin käyttämään jo olemassa olevaa etäversionhallintaa Azure DevOpsista käsin kloonaamalla projektin sieltä paikalliseksi versioksi omalle tietokoneelleni. Itse projektinhallintaan käytin ilmaista projektinhallintatyökalua Trelloa. En käyttänyt Azure DevOpsin projektinhallinnan työkaluja muun kuin versionhallinnan osalta kustannussyistä. Tästä olisi koitunut kustannuksia, jotka voitiin välttää Trellon käyttämisellä. Kuvankäsittelyyn ja UI-komponenttien piirtämiseen valitsin GIMP 2.10.24 - kuvankäsittelyohjelman. Tähänkin perusteluina olivat kustannussyt, sillä GIMP on ilmaisohjelma. Opinnäytetyön aikana yhteydenpitoa toimeksiantajaan ja opinnäytetyöohjaajaan pidettiin sähköpostitse ja Teams-tapaamisissa. Teams kuuluu Microsoft 365 -ohjelmistoihin, joita henkilöstö ja opiskelijat käyttävät Karelia-ammattikorkeakoulussa.

Kehityskohteita kartoitimme yhdessä toimeksiantajan edustajan Sini Rantasen kanssa, joka on Metsätuhot-kurssista vastaava lehtori. Sain Rantaselta myös arvokasta palautetta peliä pelanneilta opiskelijoilta.

3.3 Kohderyhmä

Metsätuhot-pelin kohderyhmä koostuu ammattikorkeakouluopiskelijoista, jotka ovat osallistuneet suomenkieliselle etäopintojaksolle. Esimerkiksi taustoiltaan ja iältään kohderyhmäläiset vaihtelevat. Kuitenkin heillä voidaan olettaa olevan jonkin verran teknologista osaamista, sillä opintojakso suoritetaan ja sille

ilmoittaudutaan verkossa. Tältä pohjalta ei kuitenkaan voida olettaa, että heillä olisi kokemusta videopeleistä.

4 Kehitystyö

Seuraavissa luvuissa käsittelen Metsätuhot-peliin tekemääni kehitystyötä. Olen jaotellut kehityskohteita alaotsikkokokonaisuuksiin. Tapauksissa, joissa alaotsikon alla on useita tarkasteltavia kehityskohteita, olen jakanut nämä alaotsikoiden alaotsikoihin. Käsittelen kehityskohteita tekstissä toistuvalla kaavalla Lähtötilanne, Kehitysidea, Toteutus, Havainnot ja Jatkokehitys. Otsikon Lähtötilanne alle kuvaan Metsätuhot-pelin ensimmäisen version tilannetta. Kehitysidea-otsikon alla vastaan kysymykseen ”kuinka lähtötilannetta voisi kehittää parempaan suuntaan?” sekä pyrin perustelevaan miksi. Mihin käyttäjän tarpeisiin tai toiveisiin kehitysidealla pyritään vastaamaan? Toteutus-otsikon alle kuvaan, millaiseen lopputulokseen päädyin. Toteutuneen tuloksen ohella haluan nostaa toteutuksen aikana esiin nousseita ongelmia, pohdintoja ja oppimiani asioita otsikon Havainnot alle. Jatkokehitys-otsikon alle avaan ajatuksiani siitä, miten mielestäni kyseistä kohdetta voisi edelleen jatkokehittää. Mikäli Jatkokehitys-otsikointi puuttuu, se tarkoittaa, ettei konkreettisia jatkokehityskohteita noussut esiin. Kehitystyö ei ole raportoitu kronologisesti vaan olen pyrkinyt ryhmittelemään erillisiä kehityskohteita kokonaisuuksiksi.

4.1 Valikot

Tämän otsikkokokonaisuuden alla käsittelen Metsätuhot-pelin valikoita (menu) ja yleisesti pelin käyttöliittymiä koskevia kehityskohteita. Valikoiden vuorovaikutustekniikkana toimii hiiren liikkeet ja klikkaukset, joka Isokosken (2011, 176) mukaisesti on vakiintunut tapa kohteiden valitsemiseen ohjelmistoissa.

4.1.1 UI-komponentit ja kaikkien valikkojen yhteiset kehityskohteet

Tässä otsikkoryhmässä käyn läpi toteuttamiani UI-komponentteja eli käyttöliittymän (user interface) rakennuspalikoita, joita voidaan käyttää yhteisesti kaikissa pelin käyttöliittymissä.

4.1.1.1 Painikkeet

Lähtötilanne

Pelin valikoiden käyttöliittymissä olevat painikkeet ovat tekstimuodossa. Niistä puuttuu kokonaan visuaalinen ilme, jonka vuoksi pelaajalle voi olla epäselvää, mitkä elementit ovat painikkeita, mitkä vain tekstiä. Painikkeet eivät reagoi pelaajan viedessä hiiren osoittimen painikkeen päälle. Painikkeet eivät reagoi (visuaalisesti) painallukseen. Kaikki painikkeet ovat yksittäisinä objekteinaan, eivätkä ole sidoksissa toisiinsa.

Kehitysidea

Default eli peruspainikkeille oman blueprint-luokan toteuttaminen. Kyseiselle painikeluokan painikkeelle visuaalisen ilmeen määrittely, havainnollistamaan käyttäjälle mitkä elementit ovat painikkeita. Painikkeiden tulisi olla isoja ja erottuvia, jotta niiden klikkaaminen on helppoa (Peters 2013, luku 9). Painikkeiden tulisi olla rajattuja ja tarpeeksi erillään toisistaan vääriin kohteeseen klikkaamisten minimoimiseksi (Peters 2013, luku 9).

Painikkeeseen hiiren liikkeisiin ("mouse hover over" -efekti) ja hiiren painallukseen reagoivien visuaalisten muutosten toteuttaminen. Kun pelissä käytetyt painikkeet ovat saman luokan ilmentymiä, niitä kaikkia voidaan muokata yläluokasta käsin. Esimerkiksi mikäli painikkeiden visuaalisessa esityksessä jotain halutaan muokata, yläluokkaa muokkaamalla saadaan muutos kaikkiin sen lapsiin. Jos painikkeet eivät ole sidoksissa toisiinsa muokkaukset on tehtävä yksitellen.

Toteutus

Tein painikkeelle blueprint-luokan, joka periytyy Unreal Enginen Button-luokasta. Määritin painikkeelle normaalin ulkoasun lisäksi ulkoasut miltä painike näyttää, kun hiiren osoitin liikutetaan siihen päälle sekä miltä painike näyttää sitä painettaessa. Kävin läpi pelin käyttöliittymien luokat, ja vaihdoin painikkeet luomani painikeluokan ilmentymiksi. Käytin painikkeen ulkoasuun omistamaani

ostettua 2D-painikeasettia, jota myös itse muokkasin GIMP-kuvankäsittelyohjelmalla.

Havainnot

Unreal Engine tarjoaa omien komponenttiluokkien luomiseen erinomaisesti tukea. Oman blueprint-luokan luominen on helppoa ja periessään Button-luokan luotu luokka sisältää kaikki Button-luokan ominaisuudet, joita voi muokata mieleisekseen. Tällaiset komponenttien luokat kannattaisi kuitenkin luoda heti projektin alkuun, ja ottaa ne käyttöön heti. Vanhojen painikkeiden vaihtaminen oman toteuttamani painikeluokan ilmentymiksi ei ollut vaikeaa, mutta jokseenkin työlästä. Painikkeet, joihin viitattiin blueprint-ohjelmoinnissa, vaativat viittausten korjaamista painikkeen luokan vaihduttua.

4.1.1.2 Tekstiobjektit

Lähtötilanne

Pelissä käytetyt tekstielementit eivät ole sidoksissa toisiinsa. Tämä nousee kehityskohdaksi kehittäjän, ei käyttäjän, näkökulmasta.

Kehitysidea

Mikäli tekstiobjekteihin halutaan tehdä muutoksia, esimerkiksi vaihtaa fonttia, jokainen tekstielementti olisi käytävä läpi yksitellen. Olio-ohjelmoinnin periaatteiden vuoksi niistä olisi hyvä luoda oma luokkansa. Luokka on malli, jonka pohjalta oliot (tässä tapauksessa tekstiobjektit) luodaan (Weisfeld, 2019 luku 1). Koska pelissä tarvitaan useampaa toisistaan eroavaa tekstityyppiä (ainakin otsikot, painikkeet, infotekstit), näille jokaiselle luodaan oma luokka.

Toteutus

Loin tekstejä varten kolme erillistä luokkaa, jotka eroavat toisistaan fontin koolla, kirjainvälillä ja fontin ulkoreunan paksuudella. Kaikkiin tekstiluokkiin valitsin fontin väriksi valkoisen tekstin tummanruskealla reunuksella, jotta teksti erottuisi kontrastin avulla paremmin. Kävin läpi pelin käyttöliittymiä ja vaihdoin yksi

kerrallaan aiemmin käytössä olleet tekstielementit joihinkin kolmesta luomastani tekstiluokasta käyttökohteen mukaan.

4.1.1.3 Tiedon lisääminen ja infoikkunat

Lähtötilanne

Käyttäjä saa pelissä hyvin vähän tietoa käyttöliittymistä. Pelissä ei ole käytössä infoikkunoita (Tool Tip) tai muutakaan lisätietoa jakavaa käytäntöä. Sanaa infoikkuna käytän tarkoittamaan hiiren liikkeisiin perustuvaa hiiren osoittimen viereen aukeavaa lisätietoikkunaa. Infoikkuna aukeaa, kun käyttäjä vie hiiren osoittimen elementin päälle, joka on määritetty sisältämään tällaista lisäinformaatiota (Kuva 5).



Kuva 5. Valkoisella taustalla oleva teksti on infoikkuna, joka on auennut hiiren osoittimen siirryttyä kohteen päälle.

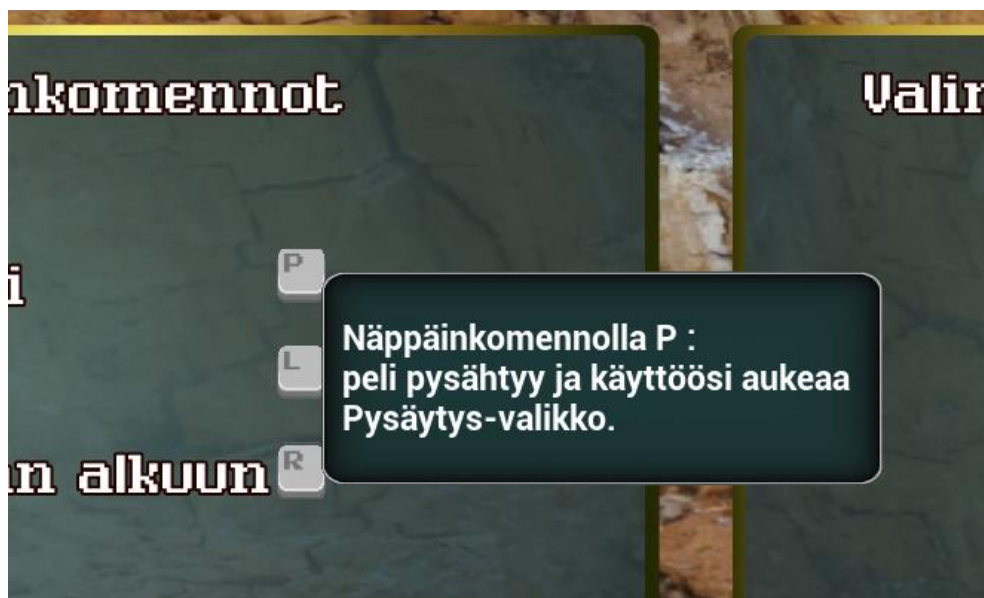
Kehitysidea

Infoikkuna widget-luokan luominen, jota voidaan käyttää kaikissa halutuissa blueprint-luokissa. Unreal Engine tarjoaa jo valmiiksi Tool tip -ominaisuuden (Kuva 5), mutta sen tekstikoko on melko pieni eikä se sovi mielestäni visuaalisesti Metsätuhot-peliin.

Hiiren liikkeisiin perustuva lisätiedon tai -ominaisuuksien tarjoaminen käyttäjälle on yleinen tapa, jota muun muassa Google käyttää (Peters 2013, luku 5). Peters tarkentaa, että tämä auttaa pitämään käyttöliittymän yksinkertaisena ja siistimpänä, kun lisätoimintoja tai -tietoa tarjotaan vasta käyttäjän sitä kaivatessa. Kuten otsikon 2.2 Käyttöliittymäsuunnittelu alla on todettu, yksi käyttöliittymän tärkeistä tavoitteista on tiedon jakaminen (Schell 2019, luku 15). Infoikkunoilla pyritään takaamaan käyttäjälle kaikki tarvittava tieto, jota hän voi tarvita, mutta jota hän ei tarvitse koko ajan.

Toteutus

Loin kaksi erikokoista UserWidget-tyyppistä blueprint-luokkaa, jotka sisältävät taustapaneelin ja tekstiobjektin. Unreal Engineissä Widget-luokkia voi käyttää muiden luokkien sisällä. ToolTip-tyyppisen widgetin voi sijoittaa esimerkiksi tiettyyn painikkeeseen tai muuhun UI-komponenttiin, jolloin infoikkuna aukeaa käyttäjän viedessä hiiren infoikkunan sisältävän objektin päälle (Kuva 6). Infoikkunan sisältöä voi halutessaan määrittää ehtojen mukaisesti (tästä esimerkki kohdassa 4.1.2 Aloitusinfo) tai kuten useimmissa tämän opinnäytetyön tapauksissa sisältö voi pysyä aina samana eli muuttumattomana.



Kuva 6. Uuden infoikkunan tyyli.

Jatkokehitys

Voisi olla perusteltua toteuttaa asetuksiin infoikkunoita koskeva asetus, josta ne voi kytkeä pois käytöstä. Jotkut käyttäjät voivat kokea infoikkunat häiritsevinä, varsinkin mitä tutummaksi peli tulee, sitä tarpeettomammaksi niiden tarjoama lisäinformaatio käy. Jäin myös pohtimaan, olisiko hyvä toteuttaa infoikkunoihin jokin ”ponnahdusraja”. Toteutettu logiikka pitäisi kirjaa siitä, kuinka monesti käyttäjälle on kyseinen infoikkuna avautunut. Raja-arvon ylittyttyä infoikkuna ei enää aukeaisi.

Pohdin myös, että haluaisin toteuttaa jokaiseen käyttöliittymään erillisen infopainikkeen, joka antaisi lisää tietoa kyseisestä käyttöliittymästä ja sen toiminnoista. Tämä idea tuli minulle kesken kehitystyön, eikä aikaa sen toteuttamiselle jäänyt.

4.1.1.4 Käyttöliittymien visuaalinen viimeistely

Lähtötilanne

Pelin eri näkymistä puuttuu peleille tyypillinen yhtenäinen visuaalinen ilme toistuvine väreineen, teemoineen tai elementteineen. Tekstit, painikkeet ja muut

UI-komponentit ovat käyttöliittymissä yleensä lisättynä taustakuvan päälle ilman rajoituksia tai visuaalista sijoittelua.

Kehitysidea

Käyttöliittymäelementeille oman taustapaneelin toteutus, jolla UI-komponentteja voi ryhmitellä ja ne voidaan selkeästi erottaa taustasta. Tarpeen mukaan muiden mahdollisten visuaalisten UI-komponenttien luominen, joilla ei ole toiminnallisuutta, vaan niiden tarkoituksena on yhtenäistää pelin yleisilmettä ja tuoda viimeistellympää vaikutelmaa.

Taustapaneelin toteuttaminen auttaa ryhmittelemään UI-elementtejä käyttöliittymässä ja auttaa käyttäjän huomion ohjaamisessa niihin. Sijoittelu, ryhmittely ja huomion ohjaaminen nopeuttavat ja selkeyttävät ohjelmiston käyttöä (Berg ja Koko, 167). Wiion (2004, 230) mukaan sovelluksen on tarjottava käyttäjälleen yhtenäiset ja johdonmukaiset käyttö- ja esitystavat. Näkisin, että yhtenäinen ilme pelissä on osa johdonmukaista esitystapaa.

Toteutus

Tein GIMP-ohjelmalla reunustetun läpikuultavan paneelin, jota voi käyttää niin taustana painikkeille kuin muillekin UI-komponenteille. Tarkoituksena oli rajata selkeästi olennaiset objektit kokonaisuuksiksi, joihin käyttäjän huomio kiinnittyy. Toin projektiin otsikoiden taustaksi assetin, jonka olin ostanut samalta tekijältä kuin painikeluokassa käyttämäni alkuperäisen assetin.

Suurimman osan pelin valikoista kohdalla tein kopiot Metsätuhot-pelin jo olemassa olevista valikoista. Muokkasin näitä kopioituja versioita visuaalisesti yhteensopiviksi taustapaneelin, otsikkokuvan, painikkeiden, taustakuvien ja fonttien osalta (kuten tulen mainitsemaan kunkin valikon kehityksestä kertoessani). Haluan tässä tuoda ilmi, että tehtyäni visuaaliset muokkaukset esittelin ne toimeksiantajan edustajalle Sini Rantaselle. Hyväksyin uudet visuaaliset ilmeet hänellä varmistaakseni, että ne palvelevat tarkoitustaan pelissä. En mainitse tätä vaihetta jatkossa kutakin kehityskohdetta käsitellessäni toiston välttämiseksi. Hyväksynnän saamisen jälkeen poistin pelin

alkuperäiset widget-luokat ja muokkasin viitaukset kohdistumaan näihin muokattuihin valikoihin.

Havainnot

Jälkiviisaana ihmettelen, miksi en tehnyt taustakehyksestä omaa blueprint-luokkaa kuten tein painikkeista ja tekstiobjekteista. Olisi loogista, että myös tästä UI-elementistä olisi oma luokkansa. Mikäli taustakehystä halutaan tulevaisuudessa muokata, olisi parempi, mikäli kaikki kehysobjektit olisivat saman yläluokan ilmentymiä.

Jatkokehitys

Taustakehyksestä voisi tehdä oman widget-luokan samoilla perusteilla kuin painikkeiden ja tekstiobjektien luokkatoteutukset tehtiin.

4.1.1.5 Siirtymät ja animaatiot

Lähtötilanne

Animaatioita käytetään käyttöliittymissä, mutta vähäisesti. Siirtymät käyttöliittymästä toiseen tapahtuvat pääasiassa välittömästi, joitain poikkeuksia lukuun ottamatta.

Kehitysidea

Animaatioiden lisääminen käyttöliittymiin ja niiden välillä siirtymisiin. Tässä yhteydessä animaatioista puhumisella tarkoitetaan hyvin yksinkertaisia animaatioita, kuten ruudun asteittainen tummentuminen siirtymien välillä tai UI-komponentin liikkumista tai koon vaihtelua.

Pienetkin animaatiot elävöittävät pelimaailmaa. Schell (2019, luku 13) toteaa, että ihmiset nauttivat kauniista ja kiinnostavista asioista. Hänen mukaansa tästä syystä peleissä käytetään usein musiikkia ja animaatioita pelaajan palkitsemisen yhteydessä. Schell ei tarkenna tarkoittaako hän tässä kohtaa tarinallisia animaatioita vai käyttöliittymäobjektien animaatioita. Mutta

käyttöliittymä- ja taustaelementtien animaatioita käytetään monissa peleissä kentän suorituksen yhteydessä, esimerkiksi Super Mario World 2: Yoshi's Island (1995), Darkest Dungeon (2016) ja Dead by Daylight (2016).

Toteutus

Toteutin muutamia yksittäisiä animaatioita, joista mainitaan kohdissa 4.2.5 Respawn-toiminto, 4.2.6 Kentän läpäiseminen ja 4.2.7 Pelin läpäiseminen. Käytin toteutuksissa apuna Matt Asplandin (2021a) tutoriaalivideota.

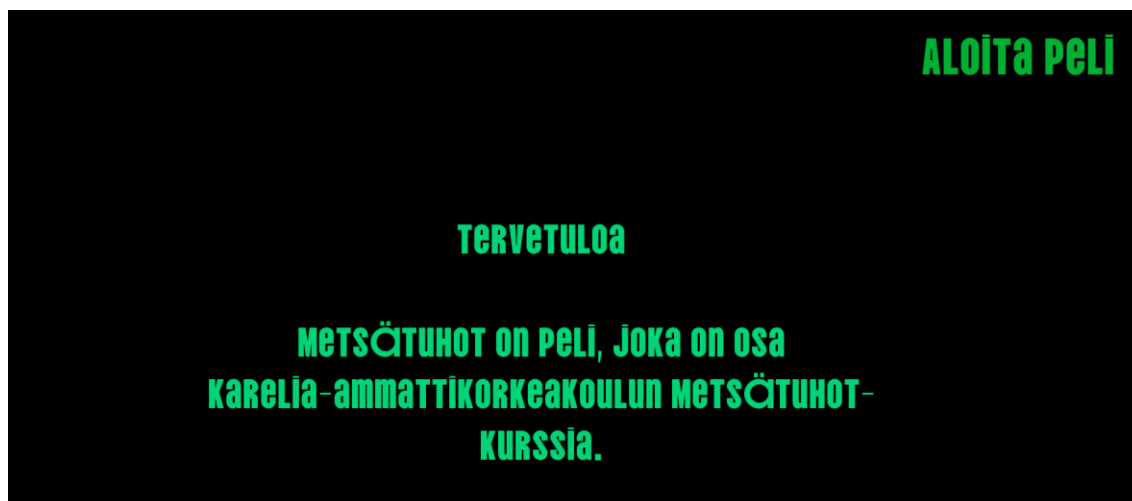
Havainnot

Olisin halunnut toteuttaa enemmänkin pieniä käyttöliittymäanimaatioita, mutta niiden tekeminen yksitellen ei mahtunut aikatauluuni. Yksittäisiä animaatioita tehdessäni aloin pohtia onko animaatioista mahdollista tehdä yläluokkaa. Character-luokan animaatioille on oma blueprint-luokkansa ja näissä on kyse huomattavasti monimutkaisemmista animaatioista ja animaation käsittelystä. Animaatiot olivat prioriteettilistalla niin alhaalla, ettei aikaa asian opiskeluun löytynyt tämän projektin puitteissa.

4.1.2 Aloitusinfo

Lähtötilanne

Käyttäjän käynnistäessä pelin ensimmäisenä avautuu pelin aloitusinfo. Käyttöliittymässä rullaa teksti, jonka tarkoituksena on informoida pelaajaa pelin kulusta (Kuva 7). Käyttöliittymän tausta on musta ja siinä on yksi painike: "Aloita peli". Painikkeella käyttäjä voi ohittaa alkuinfon ja siirtyä päävalikkoon. Aloitusinfo tuntuu mielestäni vajaalta. Siinä ei esimerkiksi kuvailla montako pistettä pelaaja tarvitsee edetäkseen kentissä tai kuinka kentästä siirrytään seuraavaan.



Kuva 7. Pelin aloittamisen lähtötilanne.

Kehitysidea

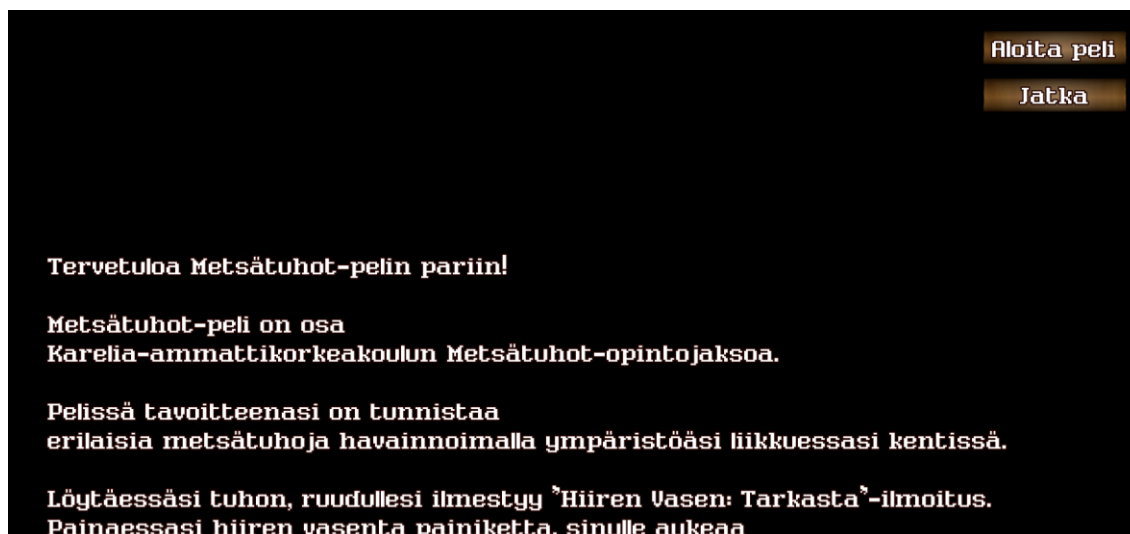
Vaikka Peters (2013, luku 10) ei suosittele käyttämään pitkiä tekstimuotoisia ohjeita, on mielestäni käytettävyyden ja saavutettavuuden kannalta perusteltua sisällyttää peliin ohjeet pelin kulusta. Varsinkin, kun ohjeet voi halutessaan ohittaa. Kun ajatellaan kohderyhmää, ei voida olettaa, että kaikilla Metsätuhot-pelin pelaajista olisi kokemusta videopeleistä. Uskon, että joillekin pelit voivat olla täysin oman mukavuusalueen ulkopuolella, jolloin aloitusinfo voi auttaa käyttäjää pääsemään sisään peliin.

Koska pelissä ei ole tutoriaalienttä, olisi mielestäni relevanttia lisätä aloitusinfoon tietoja, kuten tarvittavien pisteiden määrä, kentästä toiseen siirtyminen ja kenttien määrä. Lisäksi saavutettavuuden ”Suunnittele kaikille”-periaatteen mukaisesti (esitely osiossa 2.6 Saavutettavuus) kokisin, että liikkuvan tekstin lukeminen voi olla joillekin käyttäjille haastavaa. Käyttöliittymään voisi tällä perusteella lisätä tekstin vierityksen pysäyttävän painikkeen.

Toteutus

Lisäsin päävalikkoon siirron tekevän painikkeen alle liikkuvan tekstin pysäyttävän painikkeen (Kuva 8). Toteutin painikkeeseen logiikan, joka painettaessa pysäytyksen lisäksi vaihtaa painikkeeseen tekstin ”Jatka”.

Painettaessa painiketta tässä tilassa teksti liikkuu jälleen ja painikkeen tekstiksi vaihtuu ”Pysäytä”. Lisäsin tekstiin tiedot pisteiden ja kenttien määrästä sekä kuinka kentästä toiseen siirtyminen tapahtuu.



Kuva 8. Pelin aloitus kehitystyön jälkeen.

Jatkokehitys

Tutoriaalienten toteuttaminen palvelisi kuormittavasta tekstistä luopumista ja peleille tyypillistä käytännön kautta oppimista. Tutoriaalitasossa voisi kentissä ja pelissä etenemisen esittää asteittain, jolloin kaikkea uutta tietoa ei tarvitsisi sisäistää kerralla. Asteittainen tutoriaaliperehdytys ja minitutoriaalit ovat Petersin (2013, luku 4) mukaan hyviä keinoja perehdyttää käyttäjää minimaalisella kuormituksella.

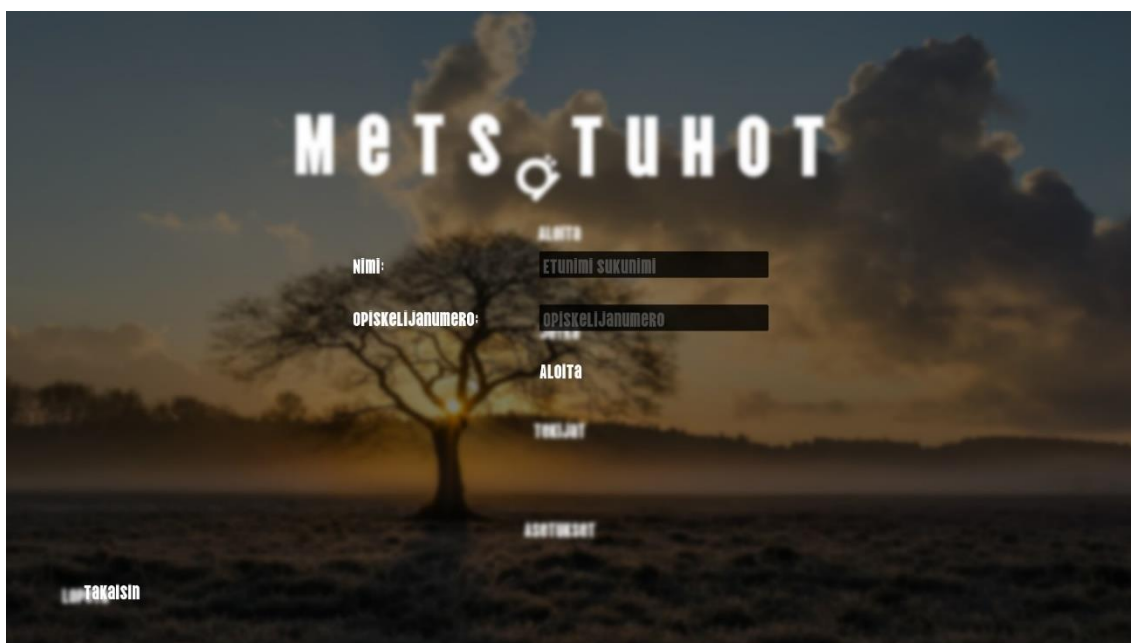
4.1.3 Päävalikko ja peliin kirjautuminen

Lähtötilanne

Päävalikosta pelaaja pääsee aloittamaan pelin, jatkamaan peliä ja lopettamaan pelin sekä tarkastelemaan pelin tekijöiden tietoja tai asetuksia (Kuva 9). Painaessaan Aloita-painiketta aukeaa pelaajalle kirjautumisikkuna (Kuva 10). Pelaajan tulee täyttää tekstilaatikoihin nimensä ja opiskelijanumeronsa ennen kuin hän voi aloittaa pelaamisen. Aloita-painike ei päästä pelaajaa aloittamaan, jos molemmat tai toinen tekstilaatikoista on jätetty tyhjäksi tai jos kohtaan opiskelijanumero on laitettu aakkosia.



Kuva 9. Päävalikon lähtötilanne.



Kuva 10. Peliin kirjautumisen lähtötilanne.

Kehitysidea

Ulkoasun muokkaaminen yhtenäiseksi peliin toteutetuilla UI-komponenteilla ja graafisten elementtien - kuten otsikoinnin, taustakuvan ja rajauksen - toteuttaminen sekä taustakuvan vaihtaminen. Päävalikkoon ohjeistuksiin vievän painikkeen lisääminen.

Kirjautumisikkunan logiikkaan tarkemmat tarkastelut annetuista syötteistä. Eli niin, ettei kohtaan nimi voi laittaa numeroita tai erikoismerkkejä eikä kohtaan opiskelijanumero voi laittaa erikoismerkkejä. Opiskelijanumero-tekstilaatikkoon on entuudestaan toteutettu aakkosten hyväksyminen.

Käyttäjän syötteet tekstilaatikkoon voivat tulla muussakin kuin toivotussa muodossa. Jotta virheellisten käyttäjäsyötteiden aiheuttamia ongelmia voidaan ehkäistä, on tyypillistä ohjelmoida syötteen sopivuutta tarkkailevaa logiikkaa (Whittaker 2009, luku 3).

Toteutus

Lisäsin päävalikkoon UI-elementtejä ja vaihdoin jo olemassa olevat objektit toteuttamieni UI-komponenttien ilmentymiksi. Lisäsin painikkeen ohjeistuksiin (katso luku 4.1.5) (Kuva 11) sekä myöhemmässä kehitysvaiheessa Tulokset-painikkeen (katso luku 4.2.7). Muokkasin kirjautumisikkunaa vaihtamalla jälleen komponentit omiin toteutuksiini ja lisäämällä taustapaneelin (Kuva 12). Etsin luokan ja kohdan, jossa käyttäjän syötteet tarkastetaan. Funktiokutsu oli toteutettu rajapinnalla ja funktion määrittely C++-kielellä. Lisäsin nimen ja opiskelijanumeron tarkastelun funktioihin ehdot, jotka eivät hyväksy erikoismerkkejä sekä nimeen ehdon, joka ei hyväksy numeroita.



Kuva 11. Päävalikko nyt.



Kuva 12. Kirjautumisikkuna kehitystyön jälkeen.

Havainnot

Testatessa huomasin, että käyttäjä voi edelleen painaa kirjautumisnäkyvän taustalla näkyvän päävalikon painikkeita. Tämä ei ole tarkoituksellista, korjasin käyttöliittymän logiikkaa niin, ettei taustalla näkyviä painikkeita voi painaa.

Huomasin jälkikäteen, että olisin voinut sijoitella käyttöliittymän komponentit paremmin. Takaisin-painike hukkuu helposti taustaan. Sen olisi voinut siirtää taustapaneelin sisälle. Kun Takaisin-painikkeen on kaikesta muusta erillään ja muun ryhmittelyn ulkopuolella, se voi jäädä täysin huomaamatta. Tämä on pieni mutta huomion arvoinen yksityiskohta, sillä Petersin (2013, luku 10) mukaan on hyvän suunnittelun mukaista tarjota opiskelijalle aina mahdollisuus peruuttaa.

4.1.4 Asetukset

Tutustutaan seuraavaksi pelin asetusten käyttöliittymän kehityskohteisiin. Asetuksissa oli käyttöliittymistä runsaimmin kehitettävää, joten olen jakanut asetuksia käsittelevät kehityskohteet kolmeen alaotsikkoon.

4.1.4.1 Grafiikka-asetukset ja asetusten käyttöliittymä

Lähtötilanne

Käyttäjä ei näe mitkä asetukset hänellä on valittuina. Käyttäjälle on epäselvää, onko hänen painalluksillaan vaikutusta. Grafiikka-asetukset on toteutettu painikkeista, mutta painikkeet ovat vain tekstimuotoisia. Ne eivät anna käyttäjälle juurikaan informaatiota. Niistä puuttuu visuaalinen ulkoasu sekä hiiren liikkeisiin ja painalluksiin reagoiminen. Käyttöliittymässä ei ole taustakuvaa, eikä muutakaan peligrafiikkaa (Kuva 13).



Kuva 13. Asetusten käyttöliittymän lähtötilanne.

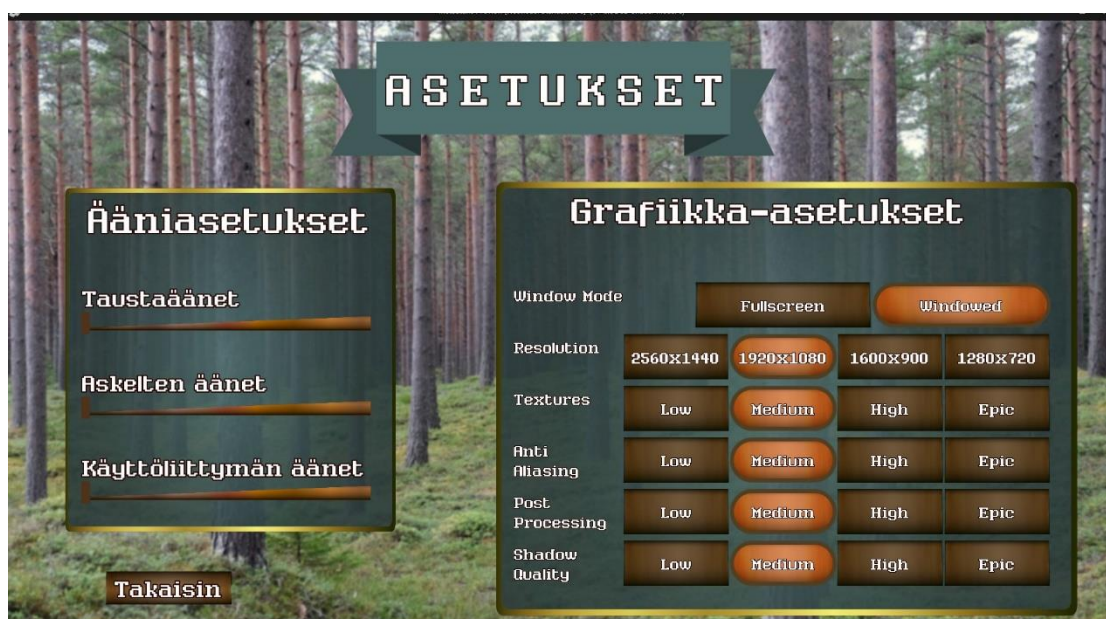
Kehitysidea

Painikkeiden vaihtaminen uuden painikeluokan ilmentymiksi. Taustakuvan lisääminen ja graafisen ilmeen muokkaaminen peliin sopivaksi. Asetusten visuaalisen esittämisen parantaminen, jotta käyttäjä näkee, mitkä asetukset hänellä on tällä hetkellä valittuina. Valittujen asetusten näyttäminen on tiedon sekä palautteen antamista käyttäjälle. Käyttäjä voi nähdä, että hänen valinnoillaan on vaikutusta. Schellin (2019, luku 15) mukaan yksi käyttöliittymän tärkeistä tavoitteista on tiedon jakaminen. Palautteen antaminen on osa ymmärrettävää ja vaivatonta käyttöliittymää (Peters 2013, luku 8).

Painikkeiden logiikan muokkaaminen niin, että valittu painike on poistettu käytöstä eli disabled. Tällöin painikkeen ulkoasu muuttuu ja käyttäjälle erottuu, mikä asetuksesta on valittuna. Kunkin asetuskokonaisuuden painikkeita varten painikkeet läpikäyvä ja valitun painikkeen disabloivan funktion toteuttaminen.

Toteutus

Vaihdoin kaikki asetusten tekstielementit ja painikkeet ilmentymiksi tekemistäni UI-komponenttiluokista, jotka esittelin kohdassa 4.1.1. Kävin läpi asetusten logiikan ja muokkasin sitä luodakseni asetuksiin näkyville käyttäjän nykyiset asetukset (Kuva 14).



Kuva 14. Asetusten käyttöliittymä, jossa on yhdistettynä grafiikka- ja ääniasetukset.

Painikkeiden ulkoasua muokkaavia funktioita kutsutaan käyttäjän klikatessa painikkeita. Jotta oikeat asetukset näkyisivät myös ensimmäisellä kerralla käyttäjän avatessa asetukset, toteutin myös funktion, jota kutsutaan asetusten blueprintin konstruktorissa. Kyseisessä funktiossa haetaan Unreal Enginen tarjoamalla GetGameUserSettings-solmulla pelin nykyiset asetukset ja päivitetään painikkeet asetusten mukaisesti. Poistin asetuksesta siellä aiemmin olleen View Distance -asetuksen, jolla määritetään kuvakulman etäisyyttä. Metsätuhot-pelissä kuvakulma ja sen etäisyys ovat pelaajahahmoon määritettyjä vakioarvoja.

Havainnot

Ensimmäisen version asetuksista toteutin kehitysidea otsikon alla kuvatulla tavalla. Eli asetuksen valitseminen poisti käytöstä, eli disabloi, painetun asetuksen painikkeen. Painikkeen ulkoasu muuttui tällöin haaleaksi, eikä painiketta voinut enää painaa. Tämän toteutuksen tein sen helppouden tähden. Painikkeen ulkoasua ei tarvinnut vaihtaa sillä Unreal Engine tarjoaa automaattisesti painikkeille ulkoasun painikkeen ollessa pois käytöstä (disabled). Saadessani opinnäytetyöohjaajaltani palautetta osana opinnäytetyöprosessia, hän nosti esiin tämän ratkaisun puutteellisuuden. Toteutus ei tukenut hyvän suunnittelun periaatteita siltä osin, että se ennemminkin näytti käyttäjälle vain asetukset, jotka hänellä eivät olleet valittuna. Käyttäjälle oleellista tietoa on se, mitkä asetukset hänellä on valittuina, joten niiden korostaminen olisi perustellumpaa kuin niiden himmentäminen (Kuva 15). Tästä syystä päädyin kehitysideasta poikkeavaan ratkaisuun.



Kuva 15. Valitut asetukset näkyvät hyvin himmeällä.

Pohdin, olisiko syytä muuttaa asetuskokonaisuuksien nimet ja asetusvaihtoehdot suomenkielisiksi. Tämä olisi tuntuisi luonnolliselta, kun kaikki muu Metsätuhot-pelissä on suomenkielistä. Koin, että suomenkieliset nimet saattaisivat kuitenkin olla epäselvempiä kuin englanninkieliset vastineensa, sillä

kaikille termeille ei ole vakiintuneita suomenkielisiä vastineita. Pohdin myös, ovatko asetusten termit ylipäättään liian epäselviä käyttäjälle. Jälleen jälkiviisaana, käyttäjälle tiedon jakamisen nimissä, olisi ollut hyvä toteuttaa asetuksiinkin infoikkunoita (tool tip). Infoikkunoissa asetusten merkitystä olisi voinut avata käyttäjälle. Pohdin lisäksi, että käyttäjälle olisi mielekästä näyttää hänen valintojensa seuraukset selkeämmin. Window mode- ja resoluutiovalinnoissa valintojen seuraukset ovat ilmeiset peliruudun koon ja resoluution muuttuessa, mutta muissa asetuksissa käyttäjän on vaikeaa nähdä, onko valinnoilla vaikutusta. Tämän voisi toteuttaa niin, että asetusten käyttöliittymässä olisi aidosti peligrafiikkaa nähtävillä (esimerkiksi puun ja pensaiden muodossa), ja grafiikan muutokset todella vaihtuisivat käyttäjälle esitettynä.

Jatkokehitys

Kuten yllä esitin, mielestäni käyttäjä saa liian vähän tietoa grafiikka-asetuksista. Infoikkunoilla tai grafiikan esittämällä asetusten käyttöliittymässä voisi selkeyttää asetusten toimintaa käyttäjälle.

4.1.4.2 Ääniasetukset

Lähtötilanne

Ääniasetuksia ei ole. Kentissä on taustaääniä (esimerkiksi linnunlaulua ja hyttysen ininää) sekä pelaajan itse ”aiheuttamia” ääniä eli askelten ääniä, mutta niiden voimakkuuteen käyttäjä voi vaikuttaa ainoastaan oman laitteensa volyyymia säätämällä.

Kehitysidea

Ääniasetusten lisääminen, niin että taustaäänillä ja askeläänillä on omat asetuksensa sekä ääniasetusten tallentaminen. On käyttäjäystävällistä antaa käyttäjälle mahdollisuus muuttaa äänten voimakkuutta tai halutessaan sulkea äänet kokonaan pois, mikäli ne eivät ole täysin välttämättömiä oppimisen kannalta. Metsätuhot-pelissä ei ole taustamusiikkia, mutta kaikissa kentissä on taustaääniä. Koska pelissä pyritään simuloimaan tilannetta, jossa pelaaja liikkuu

metsässä, on mielestäni perusteltua käyttää näitä taustääniä, kuten lintujen laulua. Mutta on hyvä, että käyttäjällä on mahdollisuus säätää äänet pois, kuten itikan ininä, joka käy pitemmän päälle hyvin raskaaksi ja varmasti häiritsee keskittymistä. Metsätuhot-pelin nykyisessä monivalintatentti muodossa ei ole pitkiä lukupätkiä tai erittäin tarkkaa keskittymistä vaativia tehtäviä. Mikäli näitä myöhemmin toteutetaan, olisi hyvä, että peli sulkisi automaattisesti taustäänet pois tällaisten tehtävien aikana. Kuten on jo useaan otteeseen todettu, kontrollin antaminen käyttäjälle on kriittisessä roolissa käytettävyyden sekä pelaaja- ja oppimiskokemuksen kannalta. Pelaajakokemuksessa kontrollin antamisen tärkeyttä korostaa Schell (2019, luku 5). Oppimiskokemuksen näkökulmasta samaa asiaa puoltavat Mäkitalo ja Wallinheimo (2012, luku 1).

Saavutettavuutta ajatellen on myös otettava huomioon, että jotkut käyttäjät voivat haluta tai tarvita voimakkaampaa volyyymia. Kuten Etelä-Suomen Aluehallintoviraston (2023) saavutettavuutta koskevilla verkkosivuilla todetaan käyttäjän tilanteiden, tarpeiden ja mahdollisten rajoitteiden sekä haasteiden huomioiminen kuuluu saavutettavaan suunnitteluun.

Toteutus

Ääniasetusten luomisen aloitin käymällä läpi pelissä käytettävät ääniasetit. Tein tarvittavat blueprint-luokat ääniasetusten käsittelyyn (Sound Class Mix ja kullekin ääniryhmälle oman Sound Class -luokkansa). Loin Slider-luokasta periytyvän luokan volyymin liukusäätimille, johon määritin säätimen ulkoasun. Lisäsin asetuksiin taustapaneelilla rajattuna kaksi kyseisen tyyppistä liukusäädintä otsikoineen. Lisäsin pelin tallennuksesta huolehtivaan blueprint-luokkaan liukulukumuuttujat taustäänille, kävelyäänille ja käyttöliittymän äänille, joille määritin perusarvoksi 0.5. Vaikka projektissa ei tässä vaiheessa ollut käyttöliittymä-ääniä käytössä, lisäsin kolmannen volyyhimuuttujan tulevaisuuden varalle. Määritin liukusäätimiin logiikan, joka liukusäätimen arvoa vaihdettaessa vaihtaa kyseisen ääniryhmän volyymit. Käytin ääniasetusten toteuttamiseen nimimerkkien DankSoss (2023) ja Matt Aspland (2021b) tutoriaalivideoita.

Tässä vaiheessa testaillessani havahtui siihen, että käyttäjä ei kuule kävelyäänien muutosta asetusten käyttöliittymässä. Taustäänien muutokset olivat havaittavissa, sillä asetusten taustalla kuuluu metsän ääniä, kuten lintujen laulua. Lisäsin askelten äänet määrittelevään slider-komponenttiin On Mouse Capture End -ominaisuuden. Eli käyttäjän vapauttaessa sliderin hiirellään pienen viiveen jälkeen soitetaan yksittäinen askelääni. Toteutin lopulta samaan tapaan liukuvolyymisäätimen myös käyttöliittymän äänitehosteille.

Havainnot

Koska pelissä on ollut useampi kehittäjä, peliin on lisännyt ääniä useampi henkilö keväällä 2023. Kaikki ääniasetit eivät löydy samasta paikasta projektin sisältökansioista. Jotta kaikki pelissä soivat äänet olisivat kytköksissä ääniasetuksiin, niiden täytyy olla luokiteltuna oikeaan ääniluokkaan. Kun etsin ääniasetteja määritelläkseni jokaiselle ääniluokan, yritin järjestellä projektin sisäisiä kansioita selkeämmin. Tämä on jälkikäteen haastavaa, sillä viittaukset oikeisiin tiedostoihin saattavat rikkoutua kansiopolkua vaihdettaessa Unreal Engine -projektin sisällä. Luovuin järjestelystä huomattessani, että olin luonut joistain ääniaseteista tuplia eli vain pahentanut sekasortoa.

Kuten kehitysideasta käy ilmi, en aluksi edes suunnitellut tekeväni liukusäätimille omaa komponentti-blueprintia. Mutta tehtyäni painikkeille ja tekstiobjekteille aiemmin esittelemäni luokat, olin todennut omien UI-komponenttiluokkien tekemisen ja käyttämisen hyvin käteväksi.

4.1.4.3 Asetusten tallentaminen

Lähtötilanne

Koska käyttäjälle ei ole näkyvissä, mitkä asetukset hänellä on valittuina, ei ole selvää, tallentuvatko valitut asetukset.

Kehitysidea

Grafiikka- ja ääniasetusten tallentaminen ja käyttäjälle tiedon jakaminen asetusten tallentamisesta. Asetusten tallentaminen lienee niin vakiintunut tapa, ettei sen tärkeyttä käyttämissäni lähteissä mainita. On kuitenkin helppo ymmärtää, miksi asetusten tallentaminen on oleellista. Jos haluamansa asetukset pitäisi joka kerta valita uudelleen ja uudelleen, olisi se pelaajan ajan ja hermojen haaskausta. Asetusten tallentaminen edistää käytettävyyttä, sillä vaivattomuus on yksi käytettävyyden piirteistä (Wiio 2004, 29–31). Myös Petersin (2013, luku 11) mukaan turhien tehtävien ja vaiheiden minimointi on käyttäjän huomioimista.

Asetusten tallentamisen toteuttaminen Unreal Enginen tarjoamien asetusten hakemisen (Get Game User Settings) ja tallentamisen (Save Settings) funktioiden sekä Unreal Enginen tarjoaman SaveGame-luokan avulla. Tallentamisen logiikan sijoittaminen käyttöliittymän Takaisin-painikkeeseen (Kuva 14). Minulla ei ollut suunnitelmaa, kuinka jakaisin tiedon tallentamisesta ennen kehitystyön aloittamista.

Toteutus

Pelin ensimmäisen version asetusten käyttöliittymän luokassa oli jo toteutettu logiikkaa pelin asetusten tallentamista varten, mutten ollut täysin varma sen toimivuudesta, koska kuten aiemmin on mainittu, käyttäjälle ei näkynyt mitkä asetukset hänellä on käytössään. Luokan konstruktorissa tarkastellaan, onko pelaajan käynnissä olevalla pelillä jo tietyssä muistipaikassa olemassa oleva SaveGame-objekti. Tarkastelu tehdään muistipaikan (slot) nimellä Does Save Game Exist -nodella. Mikäli muistissa ei ole tallennusta kyseisellä nimellä ForestSaveGame-objekti luodaan (Create Save Game Object -nodella ja otetaan talteen muuttujareferenssiksi. Viimeisenä kutsutaan toteuttamaani Load Sliders -funktiota. Jos olemassa oleva tallennus löydetään, sekin otetaan talteen muuttujaan referenssiksi ja Load Sliders -funktiota kutsutaan myös täällä.

Load Sliders -funktio ottaa syötteenä aiemmin luodun viittauksen SaveGame-objektiin. Viittauksen avulla haetaan volyymeja vastaavat liukuluvut ja asetetaan liikusäätimiin kuhunkin oma arvonsa tallennuksesta löytyvien volyymien perusteella.

Takaisin-painikkeen painallukseen reagoivassa funktiokutsussa oli ensimmäisen version jäljiltä jo tarvittavia nodeja pelin tallentamiseen. Näiden asetusten lisäksi tässä tapauksessa oli tarvetta tallentaa volyymien liukulukuarvot. Tätä varten tein SaveSliders-funktion, jossa kustakin liikusäätimestä haetaan arvo ja se asetetaan ForestSaveGame-luokan vastaaviin muuttujiin. Näiden arvojen pitäisi jo valmiiksi pitää paikkaansa On Value Changed -kutsun kautta, mutta tein tämän varotoimenä. SaveSliders -funktion jälkeen kutsutaan Unreal Enginen Save Game to Slot -funktioita, jolle annetaan syötteenä referenssi ForestSaveGame-objektista ja muistipaikan nimi.

Viimeisimpänä jo ensimmäisessä versioissa toteutetulla tarkastelulla vertaillaan, onko peli pysäytystilassa vai ei. Tämä on tarpeellista, sillä on relevanttia tietää palatako Takaisin-painikkeella päävalikkoon vai pysäytysvalikkoon. Jos peli on pysäytetty, asetuksiin on tultu pysäytysvalikon kautta. Jos peli ei ole pysäytetty, on valikkoon tultu päävalikon kautta. Käytin asetusten lataamiseen ja tallentamiseen apunani tutoriaalivideota (DankSoss 2023).

Havainnot

Pohdin, toteuttaisinko erikseen ”Tallenna asetukset”-painikkeen. Kuitenkin, koska Metsätuhot-pelin on (ainakin tässä versiossa) asetuksiltaan melko yksinkertainen, koin, että erillisestä painikkeesta koituisi käyttäjälle vain turhia klikkauksia, joiden välttäminen on Wiion (2004, 30) mukaan käytettävyyden huomioimista. Yksinkertaisella tarkoitan sitä, että kaikki asetuksen mahtuvat yhteen näkymään. Jos tulevaisuudessa peliin toteutettaisiin muokattavat kontrolliasetukset, ja asetusnäkyviä olisi useampi, kokisin ”Tallenna peli” -painikkeen relevantiksi.

4.1.5 Ohjeistus

Lähtötilanne

Kun opiskelija lataa Metsätuhot-pelin kurssin Moodlesta, hän saa samalla kirjallisen ohjeistuksen peliin. Pelin alussa pyörii myös tekstimuotoinen ohjeistus (tarkempi kuvaus otsikon 4.1.2. alla). Pelin sisällä ei ole muuta ohjeistusta pelin kulusta tai komennoista.

Kehitysidea

Uuden käyttöliittymän toteuttaminen ohjeistuksille. Kaikkien pelin näppäinkomentojen esittelemisen havainnollistavien kuvien kera. Kaikkien pelin hiirikomentojen esittelemisen havainnollistavien kuvien ja hiiren liikutusta kuvaavien animaatioiden kera. Kaikkiin ohjeistuksiin myös tekstiselitys sekä lisätiedon antaminen infoikkunoilla. Ohjeistusvalikon sijoittelun ja visuaalisesti yhtenevän ilmeen toteuttaminen peliin luotujen UI-komponenttiluokkien avulla.

Pelin sisällä tulisi olla ohjeistus pelin komennoista. Peters (2013, luku 10) suosittelee välttämään pitkiä kirjoitettuja ohjeistuksia ja miettimään mieluummin, kuinka tiedot voisi välittää käyttäjälle näköhavainnoilla ja kontekstilla. Kontekstia ja visualisointeja voi yhdistää esittelemään käyttäjälle muun muassa pelin kontrollointimekaniikkoja. Tästä syystä toteuttava ohjeistus voisi hyödyntää paljon kuvia, pieniä animaatioita ja kontekstia, mieluummin kuin pelkkää tekstiä. Kuitenkin olisi Wiion (2004, 20–21) esittelemän hyvän käytettävyyden kattavuutta koskevan kriteerin mukaista mielestäni yhdistää aiemmin lueteltuihin esitystapoihin tekstiä. Wiion mukaan: ”Kattava sovellus tarjoaa kaikki ne toiminnot ja tiedot, joita käyttäjä tarvitsee hoitaakseen sen tilanteen tai tarpeen, johon sovellus on tarkoitettu”.

Toteutus

Lisäsin pää- ja pysäytysvalikkoihin painikkeet, joista pääsee tarkastelemaan näppäin- ja hiirikomentojen ohjeistusta (Kuvat 11 ja 18). Ohjeistus on ensisijaisesti toteutettu ajatellen käyttäjiä, joilla ei ole juurikaan kokemusta videopeleistä. Kokeneemmille videopelaajille näppäimistön WASD-liikkuminen

ja hiirellä katseen kääntäminen ovat tuttuja liikutusmekanismeja, kuten on esitetty raportin luvussa 2.2. Ryhmittelin ja sijoitin ohjeistukset neljään reunustettuun kokonaisuuteen (Kuva 15). Toteutin hiirtä koskeviin ohjeistuksiin pienet animaatiot kuvaamaan hiiren liikutusta ja painikkeiden käyttöä lisätäkseen ohjeiden ymmärrettävyyttä. Näppäinkomentoihin etsin tekijänoikeuksiltaan vapaasti käytettävissä olevat näppäimistön painikkeita vastaavat kuvat. Kuvilla ja animaatioilla pelaaja saa visuaalista ohjetukea tekstin ohelle (Kuva 16). Lisäsin ohjeisiin myös infoikkunat. Käyttäjä saa kaikista ohjeistuksista lisätietoa viedessään hiiren osoittimen kunkin ohjeistuksen päälle. Pysin ratkaisuun, jossa tekstiä ei ole liikaa (infotulvan välttämiseksi), mutta kuitenkin tarpeeksi niin, ettei epäselvyyksiä komentoista jäisi. Pysin toteuttamaan ratkaisun, jossa olisi huomioitu sekä kokeneet että kokemattomimmat käyttäjät. Halusin, että videopelisiin ja tietokoneen käyttöön tottunut pelaaja saa nopealla vilkaisulla tarvitsemansa tiedot, mutta kokemattomampi käyttäjä saa halutessaan lisäksi sanallista ohjeistusta.



Kuva 16. Ohjeistukset pelin komentoihin.

Havainnot

Mielestäni ohjeistuksiin olisi hyvä lisätä vielä enemmän tapauskohtausta tietoa pelin kulusta, esimerkiksi kysymyksiin vastaamisesta. Vaikka kysymyksiin vastaaminen on entuudestaan tietokoneohjelmistoja käyttäneelle pelaajalle

tuttua, ei se välttämättä ole yksiselitteistä varsinkaan kysymyksissä, joissa voi valita useamman vastausvaihtoehdon. En kuitenkaan keksinyt, kuinka esittelisin ohjeita tällaisiin tapauskohtaisiin pelitilanteisiin. Kokisin itse opastusvideot parhaaksi ohjeistukseksi. Ehkä tulevaisuudessa Metsätuhot-opintojaksolle voitaisiin lisätä ohjeistusvideoita pelin pelaamisesta.

Jatkokehitys

Pelaajalla tulisi olla mahdollisuus muokata pelin kontrollointimekaniikkoja, jotta hänellä olisi mahdollisuus vaikuttaa enemmän pelaajakokemukseensa. Ohjeistuksen otsikoinnin voisi vaihtaa Kontrolliasetuksiksi, josta käsin käyttäjä voisi halutessaan muokata täysin painikekomentojen painikkeet. Tämä tuli kehitysideaksi liian myöhään, jotta olisin voinut lisätä sen osaksi tätä opinnäytetyötä.

Myös pelin ensimmäiseen kenttään voisi toteuttaa tutoriaalimaisen aloituksen, jossa kaikki tarvittavat näppäin- ja hiirikomennot esitellään pelaajalle pelimaailmassa sitä mukaan, kun hän niitä kentässä tarvitsee. Petersin (2013, luku 10) mukaan tämä videopeleissä usein käytetty ohjeiden ja sääntöjen esittelyn tapa vähentää kognitiivista kuormaa ja nostattaa motivaatiota. Mieleeni tuli myös, että vaihtoehtoisesti pelin kulusta voisi tehdä ohjevideon, joka lisättäisiin Metsätuhot-kurssialueelle.

4.1.6 Pysäytys

Lähtötilanne

Pelin voi pysäyttää P-näppäimellä. Pysäytettäessä pelaajalle aukeaa pysäytysvalikko, josta pelaaja pääsee jatkamaan peliä tai siirtymään joko asetuksiin, päävalikkoon tai lopettamaan pelin (Kuva 17).



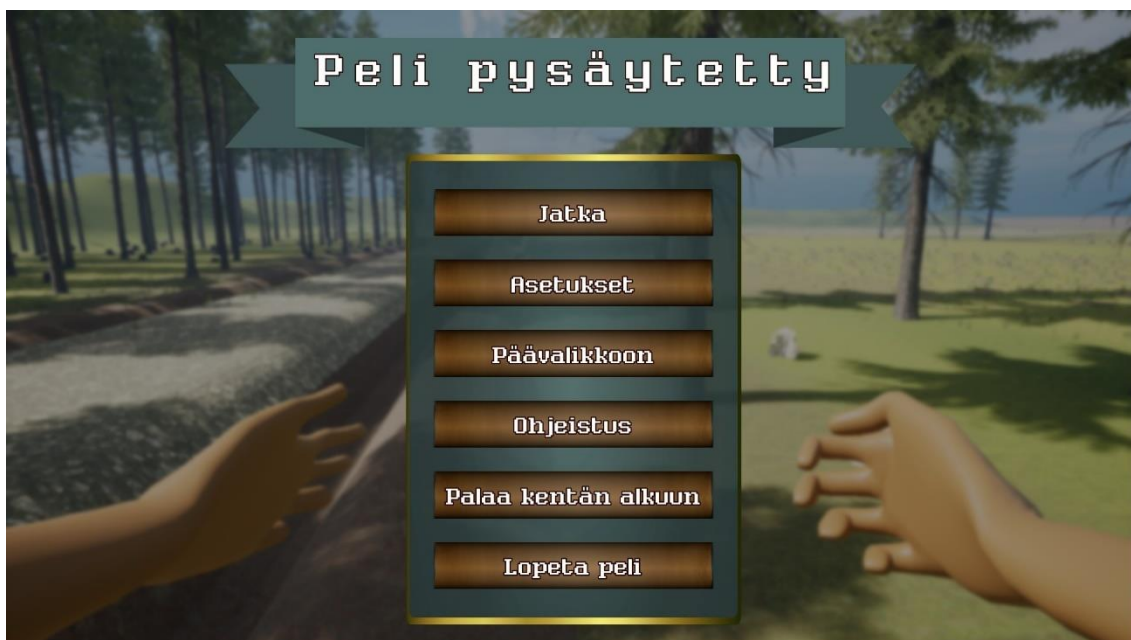
Kuva 17. Pysäytysvalikon lähtötilanne.

Kehitysidea

Pysäytysvalikon ilmeen päivittäminen pelille yhtenäiseen ulkoasuun vaihtamalla painikkeiden ja tekstiobjektien luokat. Ohjeistuksiin siirtävän painikkeen lisääminen, jotta käyttäjän ei tarvitse kulkea päävalikon kautta. Turhat välivaiheet voivat laskea oppijan motivaatiota (Peters 2013, luku 11).

Toteutus

Päivitin pysäytysvalikkoa hyvin suoraviivaisesti kehitysideassa kuvatulla tavalla. Päädyin kuitenkin myöhemmin lisäämään pysäytysvalikkoon painikkeet myös respawn-toiminnolle (kohta 4.2.5) ja kenttäsiirtymälle (kohta 4.2.6) (Kuva 18).



Kuva 18. Pysäytysvalikko kehitystyön jälkeen.

Jatkokehitys

Testatessani peliä mietin, olisiko tarpeen lisätä ”Lopeta peli” -painikkeesta painettaessa aukeava ponnistusikkuna, jossa varmistettaisiin pelaajan halu lopettaa pelaaminen. Tällä voitaisiin minimoida vahinkoklikkauksien aiheuttamaa turhautumista. Tämä olisi myös linjassa muissa kohdissa tekemiäni toteutusten kanssa sekä Petersin (2013, luku 10) esittämän ajatuksen kanssa, että käyttäjälle tulisi aina tarjota peruutuksen mahdollisuus.

4.1.7 Tekijät

Lähtötilanne

Metsätuhot-pelin tekijöiden nimet löytyvät Tekijät-näkymästä (Kuva 19).



Kuva 19. Tekijöiden esittelyn lähtötilanne.

Kehitysidea

Ulkoasun muokkaaminen yhtenäiseksi peliin toteutetuilla UI-komponenteilla eli painikkeiden vaihtaminen ja graafisten elementtien kuten otsikoinnin ja rajauksen toteuttaminen sekä taustakuvan vaihtaminen. Tekijöissä mainitaan ainoastaan opiskelijat, jotka ovat tehneet kehitystyötä Metsätuhot-pelissä. Koska pelissä on muidenkin henkilöiden tekemiä asetteja, heidät tulee lisätä tekijöiden esittelyyn.

Toteutus

Lisäsin tekijöihin otsikot ”Kiitokset” ja ”Muut tekijät”. Lisäsin kiitoksiin Karelia-ammattikorkeakoulun lehtorit, jotka ovat olleet osallisina Metsätuhot-projektissa. ”Muut tekijät” -otsikon alle listasin henkilöitä ja tahoja, joiden asetteja on käytetty pelissä. Rajasin kaikki tekstit taustaelementillä, lisäsin taustakuvan ja muokkasin ulkoasua jo muista valikoista tutuksi tulleella tavalla (Kuva 20).



Kuva 20. Tekijöiden esittelyn käyttöliittymä nyt.

4.2 Pelin kulku

Seuraavissa luvuissa käsittelen kehityskohteita ja aspekteja, jotka liittyvät pelissä etenemiseen, pelikentissä liikkumiseen ja pelin läpäisemiseen.

4.2.1 Kentän käyttöliittymä

Lähtötilanne

Kentissä pelaajalle ei käytetä Heads-Up Display -käyttöliittymää eli HUDia lähes ollenkaan. Ainoastaan tilanteissa, joissa pelaaja löytää tuhon, poimii vihjeen tai pyrkii kulkemaan portista, hänelle jaetaan tekstimuotoista tietoa kentän käyttöliittymässä.

Kehitysidea

Pelillistämisen tavoitteena on käyttäjän motivoiminen (Burke 2014, luku 4). Pienet askeleet, kokemus edistymisestä ja onnistumisten palkitseminen ovat esimerkkejä keinoista, joita pelillistämisessä käytetään motivointiin (Burke 2014). Näillä perusteilla se, että pelaajalle ei näytetä edes tietoa hänen omista pisteistään, tuntuu epämotivoivalta ja pelillistämisen tarkoituksen vastaiselta. Kehitysideana on siis pelaajalle edistymisen eli pisteiden kertymisen esittäminen edistymistä kuvastavalla mittarilla. Kuten Vuorelan (2007, 58)

Pelintekijän käsikirjassa mainitaan mittarit ovat tapa välittää oleellista tietoa pelimaailmasta pelaajalle. Unreal Engine:ssä mittareita varten on valmis käyttöliittymäkomponentti: Progress Bar. Se on hyvä työkalu tiedonvälittämiseksi käyttäjälle tilanteissa, jossa esitetyllä tiedolla on minimi- ja maksimimäärät kuten tässä tilanteessa pelaajan pisteet.

HUD on silti hyvä pitää mahdollisimman minimalistisena. Petersin (2013, luku 2) mukaan käyttöliittymä itsessään ei saisi viedä liikaa huomiota ja täten tuottaa ylimääräistä kognitiivista kuormitusta. Minimalistinen HUD sopii myös mielestäni Metsätuhot-pelin tavoitteisiin. Pelissä pyritään simuloimaan todellisen maailman tilannetta, jossa opiskelija liikkuu metsässä ja etsii merkkejä metsätuhojen aiheuttajista.

Toteutus

Lisäsin jo olemassa olevaan HUD-blueprintiin Unreal Enginen Progress Bar -tyyppisen UI-komponentin, jolle määritin itse GIMP-ohjelmistolla tekemäni ulkoasun. Käytin edistyksen esittämiseen apunani ForestDestructionGameInstance-luokassa jo valmiiksi pisteenlaskua varten toteutettua pelaajan pisteiden määrän nykyisessä kentässä laskevaa ja palauttavaa funktioita sekä kenttien maksimipisteet määrittävää muuttujaa. Näistä saadaan laskettua ja esitettyä pelaajan pistemäärän prosentuaalinen arvo.

Lisäsin HUDin blueprintiin Boolean muuttujan, joka kertoo, onko pelaaja parhaillaan vastaamassa kysymykseen. Tämä totuusarvo määrittää näytetäänkö pistemittaria pelaajalle. Totuusarvon muutokset tapahtuvat pelaajan Character-luokassa. Lisäsin lehtori Rantasen palautteen pohjalta vielä pisteet näkymään numeroina mittarin päällä, sekä selkeyttävän "Pisteet:"-otsikoinnin (Kuva 21).



Kuva 21. Pelin HUD pistelaskurimittarilla.

Havainnot

En ollut kehitysideassa ja omassa suunnitteluvaiheessa ottanut huomioon, että mittaria ei sovi näyttää pelaajalle hänen vastatessaan kysymyksiin. Testatessa havahduin siihen, että mittari reagoi oikeisiin vastauksiin välittömästi eli suoraan pelaajan raxsittaessa oikean vastausvaihtoehdon, jo ennen kuin hän sulkee kysymysikkunan. Tenttikysymysnäkyssä oikean vastausvaihtoehdon saattoi siis löytää vain klikkaillen vastausvaihtoehtoja ja katsoen missä kohtaa mittari nousee. Tallainen virhe rikkoo koko pelin idean.

Rantaselta saamani palaute auttoi toteutuksen kehityksessä. Hänen huomionsa pisteiden näyttämisestä numeroina ja otsikoinnin lisääminen paransivat toteutuksen ymmärrettävyyttä.

Jatkokehitys

Teknisen toteutuksen näkökulmasta voisi olla parempi ratkaisu kutsua pelaajan pisteet mittaria varten vain esimerkiksi pelaajan sulkiessa kysymysikkunan eikä jatkuvana reaaliaikaisena funktiokutsuna, jonka nyt toteutin.

4.2.2 Tenttikysymysten käyttöliittymä

Lähtötilanne

Tenttikysymyksissä käyttäjälle esitetään monivalintakysymys. Osassa kysymyksissä on kysymykseen liittyvä kuva, osassa pelkkä tekstimuotoinen kysymys. Osaan kysymyksistä voi valita useamman kuin yhden oikean vastausvaihtoehdon. Mikäli kysymykseen on mahdollista valita useampi vastausvaihtoehto, täytyy vastaukset vahvistaa vastaukset tallentavan painikkeen kautta. Vastaukset vahvistava painike on käytössä vain kysymyksissä, joissa voi valita useamman vastausvaihtoehdon. Jos vastausvaihtoehtoja on paljon, eivätkä ne kaikki mahdu näkymään yhtäaikaaisesti, voi käyttäjä rullata vastauksia hiiren keskirullalla tai vastausvaihtoehtojen oikealle puolelle ilmestyvästä vierityspainikkeesta. Kysymyskäyttöliittymän painikkeet ovat Unreal Enginen tarjoamia peruspainikkeita (Kuva 22).



Kuva 22. Kysymysten käyttöliittymän lähtötilanne.

Kehitysidea

Käyttäjälle tiedon lisääminen käyttöliittymästä. Painikkeisiin infoikkunoiden lisääminen, jossa niiden toiminnallisuus on selitetty (eli sulje ikkuna ja vahvista

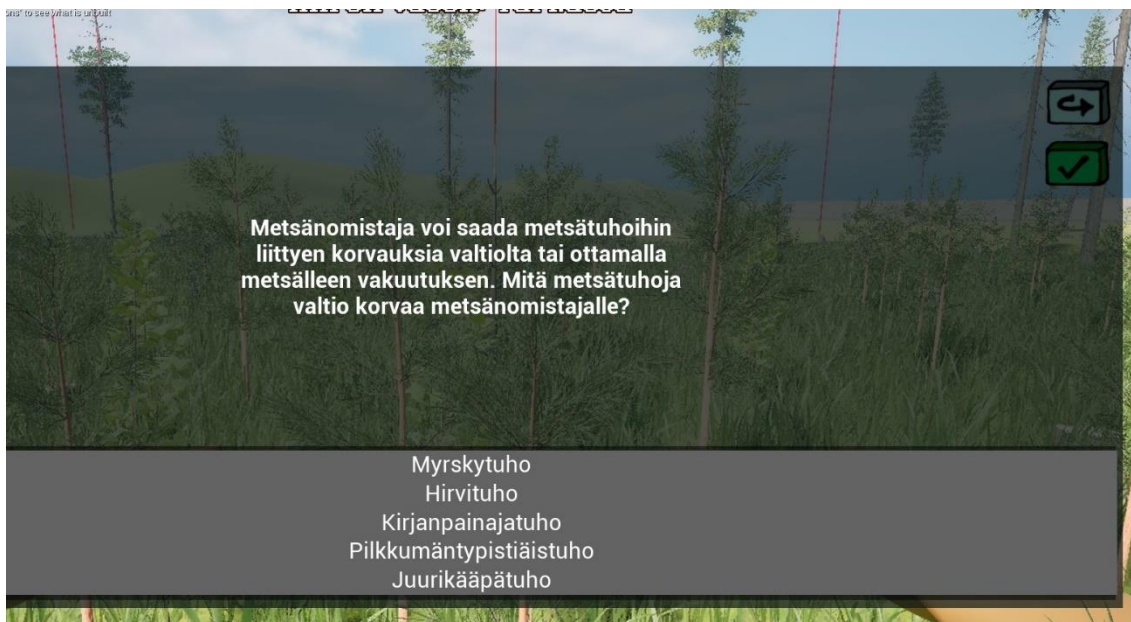
vastaus). Käyttäjälle ei välttämättä ole selvää, että vastausvaihtoehtoja on enemmän kuin ruudulla näkyy (Kuva 22). Tämän korjaamiseksi voisi joko vastausvaihtoehtoja vähentää tai vastausvaihtoehdot esittävää tilaa suurentaa, jotta kaikki vastausvaihtoehdot näkyvät kerralla. Vaihtoehtoisesti asian voisi ratkaista jakamalla käyttäjälle visuaalisesti tai sanallisesti tietoa, että vastausvaihtoehtoja saa lisää vierittämällä. Visuaalisesti tämä voisi tarkoittaa vierityspalkin korostamista ja sanallisesti infoikkunan lisäämistä. Painikkeisiin persoonallisemman ulkonäön vaihtaminen, jotta saadaan viimeistellympi vaikutelma.

Näillä korjausliikkeillä parannettaisiin kysymysten käyttöliittymän käytettävyyttä usealla Wiion (2004, 28–30) esittelemällä osa-alueella: ymmärrettävyys, vaivattomuus ja esteettinen miellyttävyys. Ymmärrettävässä käyttöliittymässä käyttäjän on helppo päätellä, kuinka hän pääsee haluamaansa lopputulokseen. Ensimmäisen version painikkeet eivät mielestäni ole intuitiivisesti selkeitä käyttää. Käyttäjä ei esimerkiksi välttämättä ymmärrä, että hänen on painettava painiketta OK (Kuva 22) vahvistaakseen vastauksen. Infoikkunoilla tämä tieto voidaan jakaa käyttäjälle ja ymmärrettävyys paranee. Jos vastausvaihtoehtoja on paljon, ne kaikki eivät näy käyttäjälle kerralla. Käyttäjältä voi jäädä helposti huomaamatta, että vastausvaihtoehdot jatkuvat alla. Hän saattaisi esimerkiksi tietää oikean vastauksen tenttikysymykseen, mutta jos hän ei näe oikeaa vastausta vastausvaihtoehtojen joukossa (koska hän ei tiedä, että vastauksia on enemmän kuin ruudulla), hän kokisi tilanteen varmasti hämmentäväksi ja saattaisi valita väärän vastausvaihtoehdon. Esteettinen miellyttävyys on osittain mielipidekysymys, joten painikkeiden vaihtaminen ei välttämättä kaikkien mielestä johda yksiselitteisesti miellyttävämpään lopputulokseen. Mielestäni se kuitenkin vie viimeistellymmän näköiseen lopputulokseen, kun pelissä ei käytetä default-painikkeita.

Toteutus

Muokkasin käyttöliittymän värimaailmaa. Vaihdoin painikkeiden ulkoasun GIMP-ohjelmistolla piirrettyihin painikkeisiin, joilla on esitysmuodot normaalitilan lisäksi hiiren liikkeisiin reagoimiseen ja painallukseen. Lisäsin painikkeisiin infoikkunat ja niihin painikkeiden toiminnallisuutta avaavat tekstit ”sulje ikkuna” ja ”vahvista

vastaus”. Siirsin vastaukset vahvistavan painikkeen kysymysten sivusta poistumispainikkeen alle käyttöliittymän ylänurkkaan (Kuva 23). Käyttäjän informoimiseksi vastausvaihtoehtojen määräästä toteutin myös infoikkunalla (Kuva 24). Toteutin tämän niin, että infoikkuna on näkyvässä silloin, kun vastausvaihtoehtoja on enemmän kuin näkyville mahtuu.



Kuva 23. Vaihdetut painikkeet.



Kuva 24. Kysymysten käyttöliittymän muutoksia, kuten ponnahtava infoikkuna kun vastauksia on lisää alla.

Havainnot

Alkuun mietimme toimeksiantajan kanssa yhdessä, että ehkä paras vaihtoehto näkymän selkeyttämiseksi olisi vastausvaihtoehtojen karsiminen. Tämä olisi kuitenkin johtanut siihen, että kaikki monta oikeaa vastausta sisältävät kysymykset olisivat poistuneet. Tämän tyyppisissä kysymyksissä vastaukset ovat yleensä pidempiä lauseita, eikä niitä mahtuisi näkymään kuin muutama kerrallaan. Kun olin luopunut ajatuksesta poistaa vastausvaihtoehtoja, ajattelin korostaa vierityspalkin ulkonäköä niin, että se erottuisi tai suorastaan hyppäisi silmään. Tein useampia erilaisia kokeiluja, mutta ne olivat mielestäni toinen toistaan huonompia ja totesin ”silmiin hyppäävän” -komponentin vain häiritseväksi tekijäksi.

Jatkokehitys

Jäin pohtimaan, olisiko syytä ottaa vastaukset vahvistava painike käyttöön kaikissa kysymyksissä, ei pelkästään kysymyksissä, joissa on useampi mahdollinen oikea vastaus. Tämä olisi siinä mielessä parempi, että se vakiinnuttaisi kysymysten toiminnan, kun kysymysten toimintalogiikka ei vaihtelisi. Toisaalta, jos kysymyksiin, joissa on vain yksi oikea vastausvaihtoehto, lisättäisiin vastauksen vahvistaminen pakolliseksi, loisi se yhden ”ylimääräisen” välivaiheen käyttäjälle. Ylimääräisiä tylsiä ja pakollisia vaihteita tulisi välttää (Schell 2019, luku 13).

4.2.3 Oikeat vastaukset ja edistymisen mittaaminen

Lähtötilanne

Pelaaja ei saa tietoa menikö hänen vastauksensa oikein vai väärin. Pelaajalle ei ole helposti saatavilla hänen edistymisensä kentässä eli kerryttämiensä pisteiden määrä tenttikysymyksistä. Ainoa tapa saada tietoa omasta edistymisestään on kulkea kentässä olevalle siirtymäportille. Yrittäessään kulkea portista ilman että hänellä on tarvittava pistemäärä, pelaaja saa ilmoituksen, että hänellä ei ole tarpeeksi pisteistä (Kuva 25). Jos pelaajalla on tarpeeksi pisteitä, hän siirtyy portista seuraavaan kenttään. Silloinkaan kun

pelaajan pisteet riittäisivät portista kulkemiseen, hän ei saa tietoa siitä muutoin, kuin yrittämällä kulkea portista.



Kuva 25. Näkymä, kun pelaaja koettaa kulkea portista liian vähillä pisteillä.

Kehitysidea

Tässä raportissa jo useaan kertaan käsitelty fakta on, että palautteen antamisen ja tiedon jakamisen pelaajalle on todettu parantavan pelaajakokemusta (Schell 2019, luku 15). Olisi täten perusteltua, että pelaajan tulisi saada reaaliaikaista tietoa vastauksistaan ja edistymisestään.

Toteutus

Toteutin edistymisen mittaamista varten kohdassa 4.2.1 kuvatun mittarin.

Havainnot

Kehityskohde ristesi kentän käyttöliittymän kehityksen kanssa. Halusin kuitenkin käsitellä tätä erillisenä otsikkona, sillä tässä käsiteltiin myös kysymysten käyttöliittymään liittyviä asioita.

Olisi perusteltua näyttää pelaajalle tenttikysymyksissä menikö hänen vastauksensa oikein vai väärin, eikä vain esittää edistymistä mittarilla. Oppimisen kannalta paras vaihtoehto olisi, että monivalintakysymykseen vastaamisen jälkeen pelaaja saisi vastausten oikeellisuuden lisäksi nähdä perusteluja vastauksiinsa (Peters 2013, luku 10). Tämä olisi vaatinut niin suuria muutoksia kysymysten käyttöliittymään sekä kysymysten ja vastausten logiikkaan, etten lähtenyt sitä tässä vaiheessa toteuttamaan.

Jatkokehitys

Kysymysten käyttöliittymän ja logiikan muuttaminen yllä kuvatulla tavalla.

4.2.4 Kysymysten lisääminen

Lähtötilanne

Samat kysymykset toistuvat pelissä melko usein.

Kehitysidea

Mielestäni olisi mielekäästä, ettei kysymyksiin, joihin on jo vastannut oikein, tarvitsisi vastata uudelleen pelin aikana. Pelistä puuttuu logiikka, joka pitäisi kirjata kysymyksistä kenttien välillä. Samassa kentässä sama kysymys toistuu vain, jos siihen on vastannut jo väärin. Koska kysymyssysteemin täysi muutos ei ole realistinen tässä vaiheessa, kysymyksiä lisäämällä voitaisiin toisteisuutta vähentää yksinkertaisella tavalla.

Toteutus

Lisäsin peliin kymmenen Sini Rantasen minulle toimittamaa tenttikysymystä. Kysymykset ovat säilöttyinä DataTable-tyyppiseen kysymystaulukkoon. Jotta tenttikysymykset voivat tulla pelimaailmassa vastaan, oli minun kunkin kohdalla tarkistettava, löytyvätkö niiden tiedot tuholaistaulukosta (DataTable). Tuholaistaulukossa kunkin tuhonaiheuttajan kohdalla on määritelty, millaisiin puihin ja puun elinkaaren vaiheisiin kyseinen tuho voi iskeä ja miltä tämä tuho

näyttää pelimaailmassa. Taulukossa on myös määritelty, mitkä tenttikysymykset koskevat tätä tuhoa.

Havainnot

Kaikkein parhaaksi kokisin toteutuksen, jossa kussakin kentässä olisi täysin omat kysymyksensä. Tämä vaatisi kenttien, tuhojen syntymisen ja kysymysten logiikan totaalisen muutoksen, johon en kokenut näissä puitteissa ja tällä aikataululla soolokehittäjänä pystyväni.

4.2.5 Respawn-toiminto

Lähtötilanne

Opiskelijapalautteista Metsätuhot-pelistä oli noussut esille pelihahmon liikkumiseen liittyviä ongelmia, tai tarkemmin liikkumisen jumiutumiseen. Pelihahmon on joissain kentissä ja kohdissa mahdollista joutua tilanteeseen, jossa pelaaja jää jumiin. Metsätuhot-pelissä pelaajan liikkuminen rajoittuu kävelemiseen (WASD-näppäimet) ja katseen kääntämiseen (hiiren liikutus).

Kehitysidea

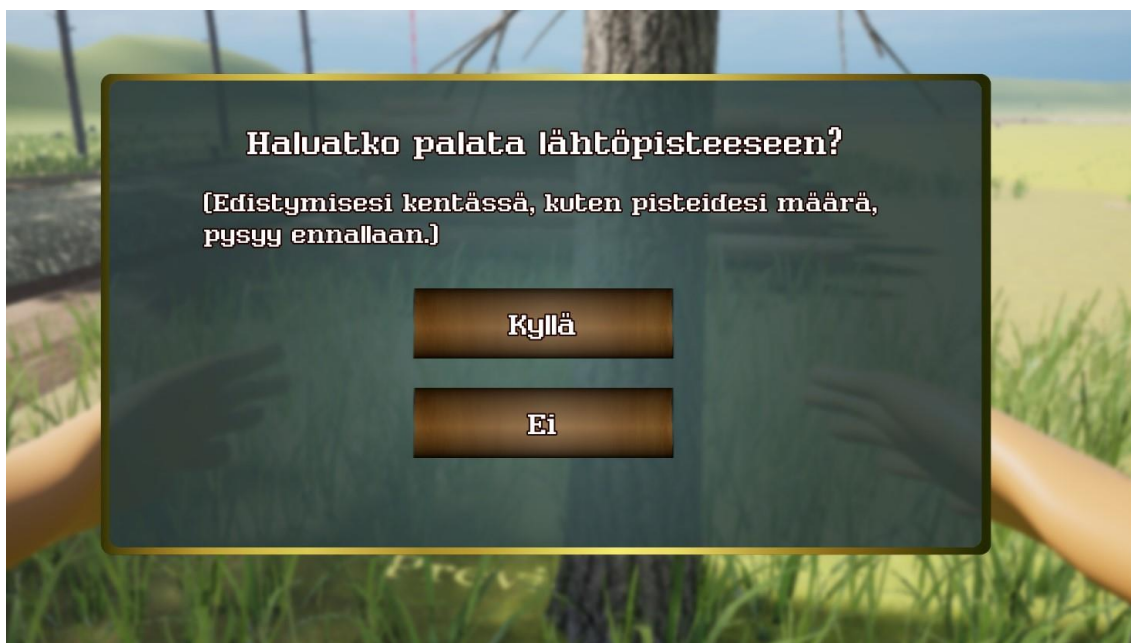
Pelaajakokemuksen parantamiseksi pelaajalle on toteutettava keinoja päästä jatkamaan peliä tilanteissa, joissa pelihahmo jää ”fyysisesti” jumiin. Tällä toiminnolla varmistetaan, että pelaaja pystyy läpäisemään Metsätuhot-pelin. Läpäisy ei saisi jäädä kiinni huolimattomasta pelisuunnittelusta. Jumiin jäämisen mahdollisuus sopii mielestäni täyttämään Vuorelan (2007, 70) selkeän vian kriteerit. Selkeäksi viaksi Vuorela määrittää testattaessa havaitun pelaamista suoraan haittaava vian, joka olisi ohjelmallisesti korjattavissa.

Monille First Person POV -peleille on tyypillistä, että pelaaja voi kävelemisen lisäksi juosta, hyppiä ja kulkea kyykyssä. Hypyllä voisi mahdollisesta jumiutumisesta päästä pois, mutta hyppääminen pelissä todennäköisesti mahdollistaisi uusia jumiin joutumisia ja pahimmillaan kentän rajojen ulkopuolelle (out of bounds) pääsyä. Kenttiä ei ole suunniteltu eikä toteutettu pelaajan hyppäämistä ajatellen. Ratkaisuksi ongelmaan voisi toimia Respawn-

toiminnon toteuttaminen pelihahmolle. Spawn-termiä käytetään peleissä objektien syntyisestä pelimaailmaan. Respawn eli uudelleen syntyminen tarkoittaa, että uudelleen syntyvä peliohjekti katoaa tai tuhoutuu pelimaailmassa syntyäkseen uudelleen.

Toteutus

Toteutin ratkaisun, jossa pelaaja voi R-näppäintä painaessaan tai pysäytysvalikon kautta (Kuva 18) valita palata kentän alkupisteeseen. Ennen siirtymistä peli pysähtyy ja pelaajalle aukeaa ikkuna, jossa varmistetaan pelaajalta hänen halunsa siirtyä alkupisteeseen (Kuva 26). Pelaajan valitessa "Kyllä" ruutu pimenee hetkeksi ja hän siirtyy kentässä pisteeseen, josta hän kyseisessä kentässä aloitti. Valitessaan "Ei" pelaajan peli jatkuu pisteestä, jossa hän painoi R-näppäintä. Pelaaja voi käyttää respawn-toimintoa loputtomasti, oli hän jumissa tai ei.



Kuva 26. Käyttöliittymällä varmistetaan, että pelaaja todella haluaa siirtyä lähtöpisteeseen.

Jatkokehitys

Respawn-toiminto palauttaa pelaajan alkupisteeseen, mutta hänen katseensa suunta määräytyy sen mukaan, mihin hän katsoi painaessaan R tai avatessaan

pysäytysvalikon. Toteutus tuntuisi viimeistellymmältä, mikäli pelaajan katseen suunta olisi aina sama kuin kentän alussa. Tämän voisi toteuttaa määrittelemällä CharacterForest-luokalle spawnLocation-muuttujan lisäksi Rotation-tyyppisen muuttujan, johon talletettaisiin pelaajan katseen suunta.

4.2.6 Kentän läpäiseminen

Lähtötilanne

Kun pelaaja on vastannut oikein tarpeeksi moneen kysymykseen ja saanut tarpeeksi pisteitä, hänen tulee kävellä kentän vaihdon mahdollistavalle portille. Pelaajalle ei anneta tietoa hänen edistymisestään muulloin kuin hänen yrittäessään kulkea portista. Jos pelaajan pisteet eivät riitä tai hän ei ole vastannut tarpeeksi moneen kysymykseen, hän saa siitä tiedon tässä vaiheessa.

Kehitysidea

Olisi pelaajan kannalta motivoivampaa, jos hän saisi tietoa edistymisestään reaaliaikaisesti kuten on käsitelty luvuissa 4.2.1 ja 4.2.3. Metsätuhot-pelin ensimmäistä versiota pelanneilta opiskelijoilta saadussa palautteessa tuli ilmi portille kulkemisen kokeminen turhauttavaksi. Peters (2013, luku 11) toteaaakin, että opiskelijalle ei pitäisi sysätä turhia tehtäviä tai välivaiheita. Portille kulkeminen on mielestäni ja opiskelijapalautteiden perusteella turha tehtävä. Portin kautta kentästä toiseen siirtymisen sijaan pelaajalle voitaisiin antaa mahdollisuus siirtyä seuraavaan kenttään heti, kun hänen pisteensä siihen riittävät ja poistaa kokonaan tarve vastata tarpeeksi moneen kysymykseen kappalemääräisesti.

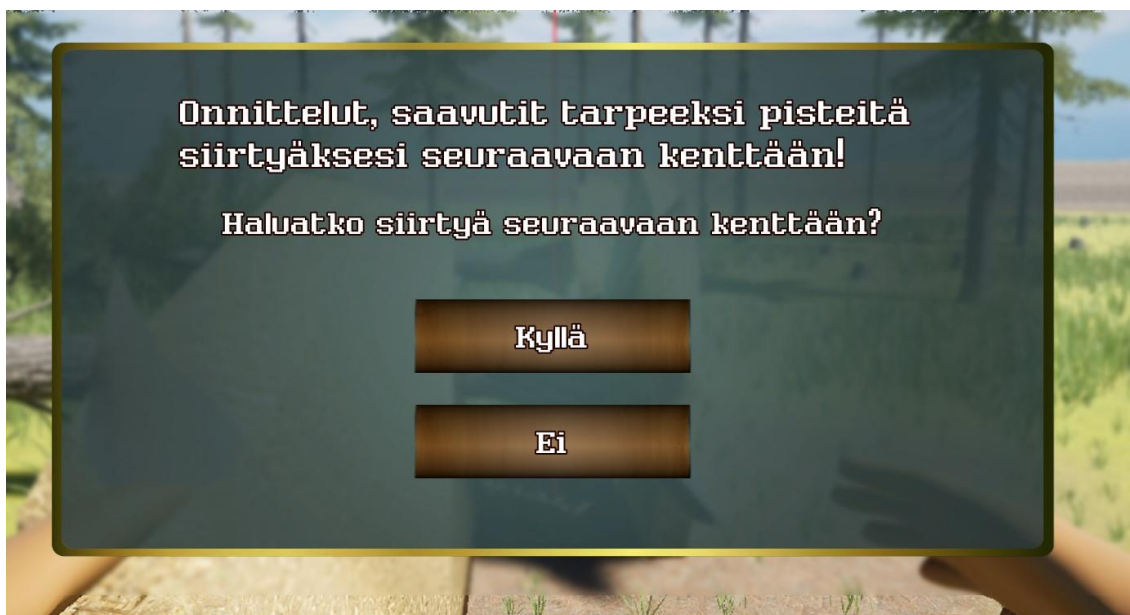
Toteutus

Toteutin pisteiden tarkkailun, joka tarkastaa pelaajan pisteet hänen sulkiessaan kysymysikkunan. Eli aina kun pelaaja sulkee kysymysikkunan, funktiokutsu tarkastaa pelaajan pisteiden määrän. Jos pisteitä on tarpeeksi, lyhyen odotuksen jälkeen pelaajalle avautuu animoitu siirtymäikkuna. Tein tämän lyhyen odotuksen ja animaation, jotta siirtymä ei tuntuisi äkilliseltä

töksäytykseltä. Auenneessa siirtymäikkunassa pelaaja saa näkyviinsä onnittelet saaduista pisteistään ja kysymyksen siitä, haluaako hän siirtyä seuraavaan kenttään. Pelaajan painaessa Kyllä-painiketta hän siirtyy välianimaation, eli ruudun asteittaisen tummentumisen, jälkeen seuraavan kentän alkuun. Mikäli pelaaja ei jostain syystä halua vielä siirtyä seuraavaan kenttään, hänelle tiedotetaan infoikkunan kautta, että hän voi siirtyä seuraavaan kenttään pysäytysvalikon kautta. Pysäytysvalikkoon lisäsin painikkeen (Kuva 27), joka avaa uudelleen tämän saman kenttien välisen siirtymäikkunan (Kuva 28). Pysäytysvalikossa siirtymäikkunan avaava painike on näkyvillä vain, kun pelaaja on saanut kyseisessä kentässä tarpeeksi pisteitä.



Kuva 27. Alhaalla oikealla on nähtävissä painike, josta pääsee seuraavaan kenttään.



Kuva 28. Kenttien siirtymisen varmistamisen käyttöliittymä.

Aiemmin kenttien määrittely oli sidoksissa portteihin. Seuraava kenttä oli kenttäsuunnittelijan määriteltävä kentän sisällä olevaan porttiobjektiin. Koska poistin portit kokonaan käytöstä ja kentistä, oli kenttien järjestys määriteltävä muualle. Lisäsin GameInstancen blueprint-toteutukseen listan kenttien halutusta järjestyksestä. Toteutin samaan blueprintiin funktion, jota kutsutaan pelaajan halutessa siirtyä seuraavaan kenttään. Funktiossa nykyisen kentän nimeä verrataan kenttien nimien listaan silmukassa. Jos ja kun nykyistä kentän nimeä vastaava muuttuja on löydetty, palautetaan listalta seuraavan kentän nimi. Tämä palautuksena saatu seuraavan kentän nimi syötetään funktiolle, joka avaa kentän nimen perusteella. Peli tallennetaan ja asteittainen tummentumisanimatio toistetaan ennen seuraavan kentän avaamista.

Jatkokehitys

Koska palautteen saaminen ja oman edistymisen näkeminen on oppijalle hyödyllistä (Peters 2013, luku 10), voisi kenttien väliin lisätä kenttäkoosteen. Tässä koosteessa voisi pelaajalle näyttää, kuinka moneen kysymykseen hän vastasi prosentuaalisesti oikein. Tulevissa kentissä voisi kenttäkoosteessa verrata aiemman kentän vastausprosenttia.

4.2.7 Pelin läpäiseminen

Lähtötilanne

Läpäistyään neljä kenttää pelaaja läpäisee pelin. Hänelle aukeaa esiin näkymä, josta hän voi kopioida suoritukselleen generoidun tunnistekoodin. Tällä tunnisteella pelaaja voi todistaa läpäisseensä pelin. Pelaajalle aukeavasta käyttöliittymästä hän myös näkee suorituksensa aloitus- ja lopetusajankohdan sekä saamansa arvosanan (0–5). Metsätuhot-pelin ensimmäisen version pelanneelta opiskelijalta saadun palautteen mukaan opiskelija oli saanut pistemääräkseen yli 100 %, mutta saanut siitä huolimatta arvosanaksi hylätyn. Pelaaja ei voi avata pelin läpäisyn todistavaa käyttöliittymää uudelleen, jos hän poistuu siitä.

Kehitysidea

Ensimmäisen version suunnittelussa ajateltiin, että mikäli pelaaja ei vastaa prosentuaalisesti tarpeeksi oikein hänen suorituksensa hylätään. Hylätyn tuloksen saadessaan pelaajan olisi pelattava peli uudelleen oppiakseen käsitellyt asiat paremmin. Tämä ratkaisu yhdistettynä siihen, ettei pelaaja näe pelatessa vastasiko hän oikein, on pelaajalle äärimmäisen turhauttava ja helposti lannistava. Lisäksi monivalintakysymyksistä koostuva visailu on jo valmiiksi kömpelö ratkaisu oppimisen kannalta (Peters 2013, luku 8). Voidaan pohtia, oppiiko pelaaja uutta, vaikka pelikertoja olisi enemmän. Oppimisen lisäämiseksi peliin olisi hyvä lisätä sisältöä ja motivaattoreita, ei turhaa haasteellisuutta. Päätimme toimeksiantajan edustajan metsätalouden lehtori Sini Rantasen kanssa, että arvosanaosuus suorituksesta poistetaan. Pelin läpäisy vaatii oikeiden vastausten antamista eli läpäistäkseen pelin pelaajan on täytynyt vastata tarpeeksi moneen kysymykseen oikein.

Jos pelaaja poistuu pelin läpäisyn näkymästä, hänellä ei ole keinoa palata siihen. Eli mikäli opiskelija on unohtanut kopioida oman läpäisy tunnisteensa, hänen ei ole mahdollista saada sitä enää, eikä hän voi tarkastella omaa suoritustaan. Tämä sotii pelaajalle kontrollin ja palautteen antamisen tärkeyttä vastaan. Tämän ehkäisemiseksi päävalikkoon tulisi lisätä painike, josta tulokset ja suoritusnäkyman saa avattua uudelleen.

Toteutus

Tutustuin pisteenlaskun ja arvosanan antavaan logiikkaan C++-koodissa. Vaihdoin käyttöliittymän tekstielementit ja painikkeen toteuttamiini UI-komponenttien luokkiin (nämä luokat on kuvailtu kohdassa 4.1.1). Ohitin koodissa arvosanan laskevan koodin ja piilotin käyttöliittymästä arvosanan esittämiseen käytetyt tekstiobjektit. En poistanut näitä mahdollista tulevaa jatkokehitystä varten.

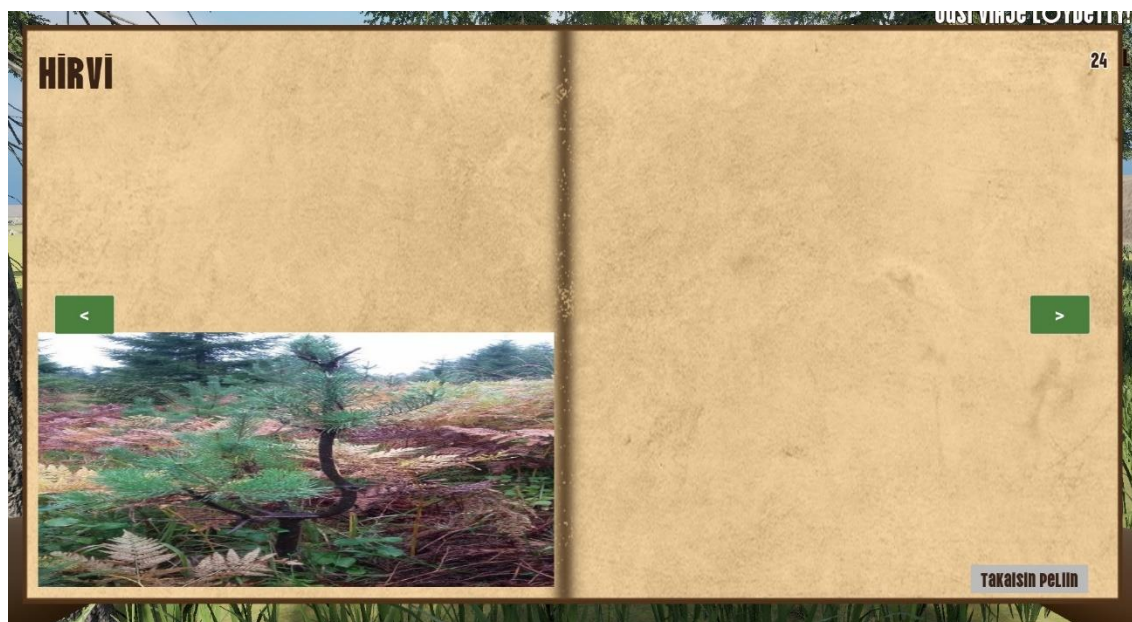
Jatkokehitys

Toteutusta voisi jalostaa niin, että pelaajalla olisi aina mahdollisuus tarkastella omaa etenemistään kyseisen käyttöliittymän kautta. Suoritustunniste tulisi näkyviin vasta pelaajan läpäistessä pelin. Pelaajalle oman suorituksen visuaalinen esittäminen on yksi motivoinnin työkalu (Burke 2014, luku 4). Edistymisen näyttävään käyttöliittymään voisi lisätä tietoa esimerkiksi kerättyjen oikeiden vastausten prosentuaalisesta määrästä suhteessa vastattuihin kysymyksiin, kuten ensimmäisessä versiossa. Sillä erotuksella, että se olisi vain pelaajalle annettavaa tietoa, eikä määrittäisi suorituksen arvoa.

Tähän näkymään voisi lisätä myös muita motivoivia kunniamerkkejä saavutuksista. Kunniamerkkien käyttö on peleille tyypillinen motivaation keino (Burke 2014, luku 4). Peleissä on myös tavanomaista antaa pelaajalle haasteita, jotka eivät liity ”todellisiin” tavoitteisiin vaan niiden tarkoituksena on tuoda lisähaastetta tai viihdykettä. Tällaisista haasteista pelaaja myös usein palkitaan kunniamerkillä tai -maininnalla. Englanniksi näitä kunniamerkkejä kutsutaan usein sanoilla achievement, badge tai trophy riippuen pelistä ja pelialustasta. Kunniamerkkejä käytetään kannustimina jo aiemmin tässä raportissa mainituissakin peleissä kuten Darkest Dungeon (2016), Fortnite (2017), ja PUBG Battlegrounds (2018). Tuoreempia esimerkkejä kunniamerkkejä käyttävistä ison profiilin peleistä ovat muun muassa Baldur's Gate 3 (2023), Diablo IV (2023) ja Alan Wake 2 (2023). Metsätuhot-pelissä näitä saavutuskunniamerkkejä voisivat olla esimerkiksi saavutus siitä, että on kerännyt yli 50 % vihjeistä tai saavutus kentissä kävelyistä välimatkoista.

4.3 Vihjekirja

Metsätuhot-pelissä pelikenttiin generoituu vihjeitä, joita pelaaja voi kerätä kävelemällä niiden lähelle. Keräämiään vihjeitä pelaaja voi tarkastella lokikirjastaan (kuva 29), jonka hän voi avata L-näppäinkomennolla. Vihjeet liittyvät pelissä (huom. ei kentässä, vaan yleisesti koko pelissä) esiintyviin metsätuhoihin. Kullekin pelissä käsitellylle tuholle on mahdollista löytää neljä vihjettä per tuho. Kirjaan kerätyt vihjeet pysyvät kirjassa kenttien siirtymien välillä. Peliin generoituu enemmän vihjeitä, kuin mitä löydettäviä vihjeitä todellisuudessa on. Pelaaja voi löytää saman vihjeen kahdesti. Vihjeet ovat joko teksti- tai kuvamuotoisia. Vihjekirja toteutettiin Metsätuhot-peliin tuomaan tenttikysymysten ohella lisä toiminnallisuutta sekä lisäämään pelaajan motivaatiota liikkua kentissä.



Kuva 29. Pelaaja voi kerätä vihjeitä kentistä. Vihjeet tallentuvat lokikirjaan.

4.3.1 Sisältö ja selailu

Lähtötilanne

Kirja avautuu aukeamalta, jolla kirja oli viimeksi pelaajan sulkiessa sen (ensimmäisellä kerralla kirja aukeaa ensimmäiseltä aukeamalta). Kaikkien löydettävissä olevien vihjeiden otsikot eli tuhonaiheuttajat ovat nähtävissä

kirjassa, vaikkei kyseiseen tuhoon olisi löytynyt vihjettä. Toisin sanoen kirjassa on paljon tyhjiä aukeamia, jos vihjeitä on poimittu vain vähän.

Kehitysidea

Kirjan aukeamisen toteuttaminen viimeisimmäksi poimitun vihjeen kohdalta selkeyttäisi kirjan käyttöä. Pelaajalle annettavan palautteen ja tiedon välittämisen kannalta on ristiriitaista, että pelaajalle ei ole selvää, minkä vihjeen hän poimi viimeksi. Pelaajalle voi olla hämmentävää ja ainakin epämotivoivaa avata kirja, kun löydetty vihje täytyy itse etsiä selailemalla. Kirjassa ei tarvitsisi näyttää tyhjiä sivuja. Sivujen selailu olisi nopeampaa ja vähemmän turhauttavaa, mikäli vain poimitujen vihjeiden sivut ovat näkyvissä. Tämä vähentää turhan selailun tarvetta.

Näillä korjausliikkeillä parannettaisiin kirjan käyttöliittymää Wiion (2014, 19–21) esittelemien hyvän käytettävyyden kriteereiden — selkeyttämisen, ymmärrettävyyden ja käytön nopeuden — osalta. Turhien klikkausten (eli tässä tapauksessa tyhjiä sivujen turhan selailun) poistaminen parantaa myös käyttöliittymän vaivattomuutta (Wiio 2004, 29). Viimeisimmän vihjeen kohdalta avaaminen ja tyhjiä sivujen selailun minimoiminen palvelevat käyttäjät tarpeita paremmin. Käyttöliittymäsuunnittelussa olisi huomioitava käyttäjän tarpeet ja toiveet (Burke 2014, luku 6, Wiio 2004, 18).

Vihjeen poimimisen yhteyteen logiikka (funktiot/muuttujat), joka ottaa talteen vihjeet poimimisjärjestyksessä listaan. Kirjan luokassa selauslogiikan muuttaminen tähän listaan pohjautuvaksi. Kirjan avaamisen yhteydessä kirjan käyttöliittymän päivitysfunktion kutsuminen viimeisimpänä poimitun vihjeen tuhonaiheuttajan mukaan.

Toteutus

Vaikka olin itse toteuttanut pelin ensimmäisen version vihjeiden ja kirjan toiminnallisuuden, minulla meni aikaa sen toteutukseen uudelleen tutustumiseen. Muutin logiikkaa kehitysideassa kuvatulla tavalla.

Jatkokehitys

Vaikka kirja aukeaa uusimman vihjeen kohdalla, voisi kaikki uudet vihjeet merkata esimerkiksi huutomerkillä tai vastaavalla korostuksella. Tämä voisi selkeyttää selailua, mikäli pelaaja ei avaa kirjaa jokaisen nostamansa vihjeen kohdalla. Korostus voisi myös lisätä saavutuksen tunnetta, jos kirjan yläkulmassa näkyisi uusien vihjeiden lukumäärä. Vihjeet pysyisivät korostettuina, kunnes pelaaja on nähnyt ne eli avannut kyseisen vihjeen aukeaman.

4.3.2 Käyttöliittymä ja komennot

Lähtötilanne

Kirjassa on käytössä Unreal Enginen oletuspainikkeet. Kirja avautuu ja sulkeutuu L-painiketta painettaessa näppäimistöllä. Peli ei pysähdy kirjan avaamisen ajaksi.

Kehitysidea

Painikkeiden vaihtaminen pelin ilmeeseen sopivampiin ja persoonallisempiin painikkeisiin. Pelin pysäyttäminen kirjan avaamisen ajaksi, jotta pelaaja ei voi kävellä kirjan käyttöliittymän ollessa auki. Pelin pysäyttäminen estää pelaajaa liikkumasta kentässä samalla, kun kirja on auki. Tällä estetään se, ettei pelaaja esimerkiksi vahingossa yritä avata tenttikysymystä kirjan ollessa auki sekä se, ettei pelaaja voi kerätä uusia vihjeitä kirjaa parhaillaan tarkastellessaan.

Toteutus

Toin peliin piirtämäni nuolisymbolin muotoisen painiketyylin, jossa on ilme mouse hover over -efektille (Kuva 30). Lisäsin kirjan avaamisen yhteyteen pelin pysäyttämisen. Koska pelin pysäytys estää käyttäjän näppäinkomentojen syöttämisen, ei kirjaa enää voi sulkea L-painikkeella. Tämän vuoksi korostin kirjan sulkevaa painiketta, josta käyttäjä pääsee takaisin kenttään ja saa taas vapauden liikkua.



Kuva 30. Kirja vaihdetuilla painikkeilla, kuvien asettelulla ja vihjemittareilla.

4.3.3 Kuvien skaalaus

Lähtötilanne

Kuvamuotoiset vihjeet skaalautuvat niille varattuihin objekteihin riippumatta kuvan todellisesta kuvasuhteesta (Kuva 29). Tämän vuoksi osa kuvista on venytettyjä ja epäselviä.

Kehitysidea

Kuvien skaalautuminen on tärkeää, sillä niissä esitetään yksityiskohtia metsätuhoista. On oppimisen kannalta ristiriitaista, että vihjekirjan kuvat ovat vaikeasti tarkasteltavia. Kuvien skaalautumisen ratkaisemiseksi kuvat voisivat kuvaobjektin sijaan olla sijoitettuna painikkeisiin. Pelaajan painaessa painiketta avautuu kuva todellisessa koossaan näytölle, jolloin pelaaja pääsee tarkastelemaan yksityiskohtia. Pelaajalle olisi hyvä myös indikoida, että pienet sivuilla olevat kuvat ovat painikkeita. Isoksi avautuvan kuvan hyöty jää tyhjäksi, mikäli pelaajalle ei ole selvää, että hän voi klikkaamalla saada kuvan suuremmaksi. Lisäämällä esimerkiksi suurennuslasisymbolin kuvan viereen pelaajalle voisi visuaalisesti viestittää, että kuvaa voi tutkia lähemmin.

Toteutus

Vaihdoin kuvat painikkeiksi, jotka avautuvat pienen animaation saattelemana todellisessa koossaan tummennetulla taustalla kirjan keskelle.

Suurennuslasisymbolilla kuvan klikkaamiseen kannustaminen jäi toteuttamatta, sillä en löytänyt hyvää ja tekijänoikeuksiltaan vapaasti käytettävää kuvaketta.

Ratkaisin tilanteen lisäämällä ”Klikkaa suurentaaksesi kuvaa” -tekstin sisältävät infoikkunat.

Jatkokehitys

Infoikkunoiden korvaaminen visuaalisella vihjeellä tekstivihjeen sijasta.

Ylimääräisen tekstin minimoimiseksi infoikkunat voisi korvata esimerkiksi juuri suurennuslasikuvakkeella tai hiiren osoittimen animoinnilla, kun hiiren osoitin viedään painikemuotoisen kuvan päälle kirjassa.

4.3.4 Vihjemittari

Lähtötilanne

Pelaaja näkee kaikki metsätuhojen otsikot, ja hän voi päätellä tästä vihjeiden määrän. Pelaaja ei kuitenkaan saa mistään tietoa, montako vihjettä hän on löytänyt, muutoin kuin itse vihjeensä laskemalla.

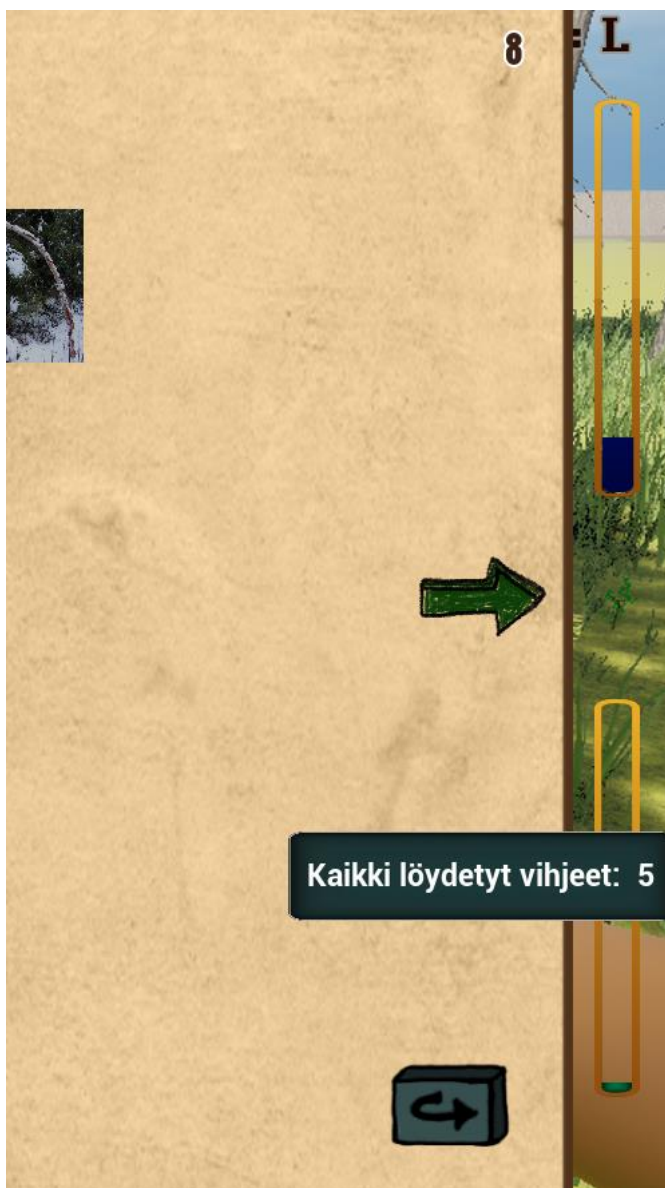
Kehitysidea

Mittarit ovat keino pelaajan motivoimiseksi pelimekaanisin keinoin (Vuorela 2007, 58). Mittarin täytyminen näyttää pelaajalle konkreettisesti hänen edistyksensä. Tämän vuoksi kirjan yhteyteen olisi perusteltua lisätä vihjeiden määrää kuvaava mittari, josta pelaaja näkee keräämiensä vihjeiden määrän prosentuaalisesti.

Toteutus

Lisäsin kirjaan mittarin, josta pelaaja näkee keräämiensä vihjeiden määrän suhteessa pelissä yhteensä kaiken kaikkiaan oleviin vihjeisiin. Tässä edistys näkyy vähäisenä, sillä vihjeitä mahtuu neljään kenttään niin monta. Tästä syystä

päädyin lisäämään myös kenttäkohtaisen mittarin, joka näyttää montako prosenttia kunkin kentän vihjeistä pelaaja on kerännyt. Lisäsin mittareihin infoikkunat, jotka näyttävät kerättyjen vihjeiden lukumäärät (Kuva 31), sillä se tarkoittaa käyttäjälle, mitä mittarit pitävät sisällään.



Kuva 31. Infoikkuna kertoo löydettyjen vihjeiden lukumäärän pelaajan viedessä hiiren mittarin päälle.

4.4 Muut kehityskohteet

Kehitystyön aikana ilmeni paljon pieniä tekniseen toteuttamiseen liittyviä parannuksia. Ei ole suotavaa luoda kesken kehitystyön uusia kehityskohteita, mutta opiskelijana kokemuksen puutteen takia on lähes väistämätöntä, että

uusien kehityskohteita voi ilmetä. Esittelen tämän otsikon alla tällaisista ne, jotka ovat pelaajan näkökulmasta huomattavissa.

4.4.1 Fontti

Lähtötilanne

Pelin valikoissa käytetty fontti ei tue ä- ja ö-kirjaimia. Ä- ja ö-kirjaimia voidaan käyttää, mutta ne eivät vastaa a- ja o-kirjaimia. Fontissa isot ja pienet alkukirjaimet ovat samankokoisia, ja niitä käytetään sekaisin. Esimerkiksi iso E-kirjain on kooltaan isompi, mutta silti e. L-kirjain on ison L-kirjaimen näköinen myös pienellä kirjoitetussa tekstissä (Kuva 7). Tämä hankaloittaa tekstin lukemista, kuten myös se, että teksti on yleisesti käytössä valkoisena fonttina ilman ääriivvoja. Vaaleilla taustoilla fonttia voi olla haasteellista lukea. Tenttikysymyksissä on käytössä erilainen fontti.

Kehitysidea

Fontin vaihtaminen selkeämmin luettavaan ja ä- sekä ö-kirjaimia tukevaan fonttimalliin.

Toteutus

Toin projektiin uuden fontin, jonka määritin fontiksi kaikkiin luomiini tekstiluokkiin (Kuva 8).

Havainnot

Fontin vaihtamista nopeutti suuresti se, että tekstiobjektit olivat nyt sidoksissa toisiinsa. Minun tarvitsi vain vaihtaa käytetyt fontit luomistani UI-komponenttien blueprint-luokista. Koska tekstiobjektien olivat, näiden luokkien ilmentymiä, niiden fontit vaihtuivat yläluokan mukaan.

4.4.2 Vihjeiden toistuminen ja vihjeen löytämisestä ilmoittaminen

Lähtötilanne

Testatessa huomasin, että vihjeiden logiikka on virheellinen. Sama vihje voi tulla seuraavassa kentässä uudelleen, varsinkin kun kenttiin syntyy liian monta vihjettä, jolloin osa vihjeistä on väistämättä tyhjiä. Eli jo löydetyistä vihjeistä ei ole toimivaa ”kirjanpitoa”. Myös ”Vihje löydetty! Avaa kirja painamalla L” -teksti, joka ilmestyy aina pelaajan kerätessä vihjeen muutamaksi sekunniksi (Kuva 32), jumiutuu pelaajan ruutuun, jos pelaaja kerää toisen vihjeen ennen kuin ensimmäisen vihjeen laukaisema ponnahdusteksti on kadonnut.



Kuva 32. Oikeassa yläkulmassa on nähtävissä pelaajalle tuleva vihjeen löytämisen ilmoitus.

Kehitysidea

Vihjesysteemin muuttaminen niin, etteivät samat vihjeet voi toistua sekä vihjeiden määrän vähentäminen kentissä. ”Vihje löydetty” -tekstiin tarkastelun lisääminen sen suhteen, onko teksti jo näkyvässä ennen tekstin lisäämistä ruudulle. Jos teksti on jo ruudulla, sitä ei lisätä toistamiseen.

Toteutus

Vähensin kenttään generoitavien vihjeiden määrää neljäsosaan aiemmasta. Lisäsin vihjeiden generointiin toimivan tarkastelun, joka estää antamasta pelaajalle vihjettä, jonka pelaaja on jo saanut. Lisäsin ajastettuun "Vihje löydetty" -tekstiin kehitysidean mukaisen rajoitteen, jolloin tekstiä ei avata uudelleen, jos se on jo näkyvissä.

4.5 Tekijänoikeudet

Lisäsin Metsätuhot-peliin itse luomieni asettien lisäksi omistamiani asetteja, joiden tekijät ovat antaneet tuotteiden ostaneelle oikeuden muokata ja käyttää niitä omissa tai kaupallisissa projekteissaan. Niiden eteenpäin myyminen on kiellettyä.

Metsätuhot-peliin on ensimmäisen version kehittämisen aikana ostettu vastaavalla lisenssillä maaston, puiden ja kasvien 3D-mallit, tekstuurit, materiaalit sekä ääniasetteja. Mahdollisessa jatkokehityksessä on huomioitava, että näitä kaikkia saa käyttää vapaasti vain Metsätuhot-pelissä. Kehittäjillä ei ole oikeutta ottaa niitä käyttöön muihin projekteihinsa.

5 Pohdinta

Tässä opinnäytetyön viimeisessä luvussa pohdin opinnäytetyön tuloksia. Pohdin onnistumistani tietoperustaan pohjautuvassa kehitystyössä kokonaisuutena sekä esitän työssä ilmenneitä haasteita, rajoitteita ja ajatuksia tulevaisuuteen.

5.1 Rajoitteet

Opinnäytetyön tavoitteena oli kehittää Metsätuhot-pelin pelaajakokemusta, käytettävyyttä ja käyttöliittymiä oppimisen pelillistämisestä ammennetusta tiedosta käsin. Minun oli kuitenkin jo etukäteen huomioitava ja hyväksyttävä, etten pysty kaikkia haasteita ratkaisemaan. Esimerkiksi pelin kulkuun minulla ei ole yksittäisenä kehittäjänä rajallisena aikana suuresti mahdollisuuksia vaikuttaa. Peli on nykyisessä muodossaan monivalintatentti. Oppimisen

kannalta visailutyypiset ratkaisut eivät ole ideaaleja (Peters 2013, luku 8). Ne eivät tuo opittavaa asiaa todellisen elämän kontekstiin (Peters 2013), vaikka pelissä on pyritty todellisen metsän ympäristöä simuloimaan. Visailut motivoivat vain ulkoisesti, ei sisäisesti (Peters 2013). Eli pelaajan motivaatio keskittyy oikean palautteen saamiseksi vastauksesta, ei todelliseen oppimiseen ja asian sisäistämiseen. Ei ollut realistista lähteä muuttamaan koko pelin sisältöä tässä vaiheessa. Ehkä Metsätuhot-peliä voidaan ajan saatossa kasvattaa vastaamaan paremmin oppimisen pelillistämisen haasteisiin myös sisältönsä puolesta.

Toinen suuri pelikehitykseen liittyvä rajoite, joka minun oli hyväksyttävä ja suljettava ulos kehitystyöstäni oli se, ettei Metstätuhot-peli ole verkkopeli. Tämä sulkee pois pelin mahdollisuudet sosiaaliseen kanssakäymiseen muiden opiskelijoiden kanssa, joka tutkitusti on tärkeää oppimisen kannalta (Peters 2013, luku 4). Metsätuhot-pelistä verkkopelin toteuttaminen olisi pelin pelattavuuden kannalta äärimmäisen tärkeää. Verkkopelin tekeminen tuo mukanaan niin suuria haasteita, ettei siihen tässä opinnäytetyössä lähdetty.

5.2 Haasteet

Edellä esitettyjen rajoitteiden lisäksi haasteekseni nousi opinnäytetyöprosessissa aiherajaukseni laajuus erityisesti tietoperustan kartoittamisen kohdalla. Löysin runsaasti lähdemateriaaleja ja aiherajaukseni tuntui opinnäytetyötä tehdessäni liian laajalta. Näin opinnäytetyötä viimeistellessäni pohdin, että minun olisi pitänyt tehdä aiherajaus tiukemmaksi. Myös kehitystyössä minua haastoi liiallisuus. Minun oli vaikeaa rajata kehityskohteita, sillä halusin edistää niitä kaikkia ja olisin halunnut tehdä entistäkin enemmän. Näissä haasteissa on positiivinen pohjavire, onhan kyse siitä, että innostuin tutkimastani aiheesta sekä koen suurta paloa pelikehitykseen. Minua jäi myös vaivaamaan se, etten kokenut esitelleeni juuri lainkaan pelien heikkouksia ja haasteita opetuskäytössä.

Eräs haasteistani oli testauksen puute. Testasin peliä ja ominaisuuksia kehitystyön ohessa jatkuvasti, mutta käytettävyyttä kehitettäessä olisi erittäin

perusteltua testauttaa kehityskohdetta testipelaajilla. Onneksi tukenani oli Metsätuhot-pelin ensimmäisen version pelaajilta saatu palaute.

5.3 Kehitystyön saavutetut tavoitteet

Koen, että kehitystyölläni Metsätuhot-pelin parissa on merkitystä pelin käytettävyyden kannalta, joten koen saavuttaneeni käytettävyyden ja pelattavuuden kehittämiseen tähtäävät tavoitteeni. Oppimisen pelillistämiseksi olisi Metsätuhot-pelissä vielä paljon kehitettävää, sillä oppiminen on laaja ja monitahoinen prosessi. Pelillistämisen ytimessä on oppijan motivointi. Pyrin pitämään sen mielessäni kehitystyön ajan ja koen, että sain pieniä edistysaskelia sen eteen aikaiseksi. Tiedon jakaminen pelaajalle hänen etenemisestään ja vaikutuksistaan pelimaailmaan on todettu motivoivaksi, joten infoikkunoiden ja mittareiden lisäämisellä pyrin tukemaan sitä. Opin itse tämän oppimispelin kehityksen parissa valtavasti.

5.4 Pelillistetty tulevaisuus

Uskon, että pelillistämistä tullaan näkemään koulutuksessa enenevässä määrin. Epäilen kuitenkin, että kehitys pelillistettyyn oppimiseen tapahtuu hitaasti. Ehkä tulemme näkemään yksittäisiä kursseja pelien muodossa tai kursseja, joiden osana on peli vaihtoehtoisena suoritustapana, kuten Metsätuhot-opintojaksolla. Jo yksittäisen ja suhteellisen yksinkertaisen pelin toteuttaminen vaatii runsaasti työtä. Ainakaan toistaiseksi pelit eivät myöskään voi itsenäisesti mukautua ja kehittyä, vaan ne ja niiden tarjoama tieto vaatisi ylläpitoa. Muun muassa nämä seikat hidastavat pelillistetyn oppimisen käyttöönottoa oppilaitoksissa ja muussa opetuksessa. Jään mielenkiinnolla odottamaan, mitä tulevaisuus tuo tämänkin tiimoilta tullessaan.

Lähteet

- Alan Wake 2. 2023. Remedy Entertainment.
- Apex Legends. 2019. Electronic Arts.
- Baldur's Gate 3. 2023. Larian Studios.
- Berg, M. & Kojo, I. 2011. Informaation visualisointi. Teoksessa Oulasvirta, A. (toim.). Ihmisen ja tietokoneen vuorovaikutus. Helsinki: Gaudeamus Helsinki University Press, Oy Yliopistokustannus, HYY yhtymä.
- Burke, B. 2014. Gamify: How Gamification Motivates People to Do Extraordinary Things. Bibliomotion, Inc: Brookline. O'Reilly for Higher Education.
- Conçalo, M., Sherry, D., Pereira, D. ja Fozi, H. 2022. Elevating Game Experiences with Unreal Enigne 5 – Second Edition. Packt Publishing Ltd: Birmingham. O'Reilly for Higher Education.
- DankSoss. 2023. Change Volume Settings UE5 Tutorial Save and Load pt 1. YouTube: <https://www.youtube.com/watch?v=aurLIF-G8kk> 15.9.2023
- Darkest Dungeon. 2016. Red Hook Studios.
- Dead by Daylight. 2016. Behaviour Interactive.
- Diablo IV. 2023. Blizzard Entertainment, Inc.
- Epic Games, Inc. 2024a. Unreal Engine 5.3 Documentation: Data Driven Gameplay Events. <https://docs.unrealengine.com/4.27/en-US/InteractiveExperiences/DataDriven/> 29.1.2024
- Epic Games, Inc. 2024b. Unreal Engine 4.26 Documentation: Nodes. <https://docs.unrealengine.com/4.26/en-US/ProgrammingAndScripting/Blueprints/UserGuide/Nodes/> 29.01.2024
- Epic Games, Inc. 2024c. Unreal Engine 4.26 Documentation: Saving and Loading Your Game. <https://docs.unrealengine.com/4.26/en-US/InteractiveExperiences/SaveGame/> 29.01.2024
- Epic Games, Inc. 2024d. Unreal Engine 5.3 Documentation: Setting Up Visual Studio. <https://docs.unrealengine.com/5.3/en-US/setting-up-visual-studio-development-environment-for-cplusplus-projects-in-unreal-engine/> 01.02.2024

Etelä-Suomen Aluehallintovirasto. Yleistä saavutettavuudesta.

<https://www.saavutettavuusvaatimukset.fi/yleista-saavutettavuudesta/> 3.10.2023

Fortnite. 2017. Epic Games.

Inkinen, S. 2008. Quo vadis, homo ludens?. Teoksessa Leino, O. Wirman, H. & Fernandez, A. (toim.). Extending experiences – Structure, analysis, and design of computer game player experience. Rovaniemi: Lapland University Press.

Isokoski, P. 2011. Syöttölaitteet. Teoksessa Oulasvirta, A. (toim.). Ihmisen ja tietokoneen vuorovaikutus. Helsinki: Gaudeamus University Press, Oy Yliopistokustannus, HYY yhtymä.

Kurhila, J. 2011. Opetus- ja oppimisteknologiat. Teoksessa Oulasvirta, A. (toim.). Ihmisen ja tietokoneen vuorovaikutus. Helsinki: Gaudeamus University Press, Oy, Yliopistokustannus, HYY yhtymä.

Laley, R. 2021. Unreal Engine 4 Tutorial – Tooltips. YuoTube.

<https://www.youtube.com/watch?v=zLRMDtKHyZA> 29.01.2024

Leblanc, M., Hunicke, R. ja Zubek, R. 2004. MDA: A Formal Approach to Game Design and Game Research. Game Design and Tuning Workshop at the Game Developers Conference, San Jose 2001-2004.

https://www.researchgate.net/publication/228884866_MDA_A_Formal_Approach_to_Game_Design_and_Game_Research
07.09.2023

Li, Z. 2023. Unreal Engine 5 Game Development with C++ Scripting. Birmingham-Mumbai: Packt Publishing. O'Reilly for Higher Education.

Leino, O. Wirman, H. & Fernandez, A. 2008. Extending experiences – Structure, analysis, and design of computer game player experience. Rovaniemi: Lapland University Press.

Matt Aspland. 2021a. Widget Animations | Fade, Slide, And Pop Examples – Unreal Engine 4 Tutorial. Youtube:

<https://www.youtube.com/watch?v=p2YwlpY0we4&t=470s>
23.10.2023

- Matt Aspland. 2021b. Change Master Volume Audio On Slider – Unreal Engine 4 Tutorial. YouTube:
<https://www.youtube.com/watch?v=VSy4ybZQK3I> 15.9.2023
- Mäkitalo, E. ja Wallinheimo, K. 2012. Virtuaaliset ympäristöt – Innostava oppiminen, tehokas koulutus. Alma Talent: Helsinki. Pirkanmaan Kirjastot.
- Oulasvirta, A. 2011. Ihmisen ja tietokoneen vuorovaikutus. Helsinki: Gaudeamus Helsinki University Press, Oy Yliopistukustannus, HYY yhtymä.
- Peters, D. 2013. Interface Design for Learning: Design Strategies for Learning Experiences. New Riders. O'Reilly for Higher Education.
- Pesonen, H. & Nieminen, J. 2021. Huomioi oppimisen esteet: inklusiivinen opetus korkeakoulutuksessa. Jyväskylä: PS-kustannus.
- PUBG Battlegrounds. 2017. KRAFTON, Inc.
- Schell, J. 2019. The Art of Game Design, 3rd Edition. A K Peters/CRC Press: Boca Raton. O'Reilly for Higher Education. 08.09.2023
- Suomen eOppimiskeskus RY. 2023. Kestävä kehitys kohtasi virtuaaliodellisuuden: kaksi peliä jakoivat oppimiskilpailun voiton. <https://eoppimiskeskus.fi/kestava-kehitys-kohtasi-virtuaaliodellisuuden-kaksi-pelia-jakoivat-oppimiskilpailun-voiton/> 08.09.2023
- Super Mario World 2: Yoshi's Island. 1995. Nintendo.
- Vuorela, V. 2007. Pelintekijän käsikirja. Helsinki: BTJ Kustannus.
- Weisfeld, M. 2019. Objec-Oriented Thought Process, 5th Edition. Addison-Wesley Professional. O'Reilly for Higher Education.
- Whittaker, J. 2009. Exploratory Software Testing. Addison-Wesley Professional. O'Reilly for Higher Education.
- Wiio, A. 2004. Käyttäjystävällisen sovelluksen suunnittelu. Helsinki: Edita Publishing Oy.

Liitteet

Metsätuhot-pelin ensimmäisen version kehittäjät:

Kehittäjät:

Alexi R.

Inari P.

Jani R.

Jari K.

Jasu K.

Marko M.

Matias A.

Niko R.

Onni F.

Toni T.

Tuukka J.

Veera T.

Ville A.

Ohjaajat:

Matveinen Mika

Nevalainen Seppo

Toimeksiantaja:

Rantanen Sini