

Henri Lindström & Ari Saaristo

## **Filosofia-aiheisen opetuspelin kehitys lukiokäyttöön**

## **Filosofia-aiheisen opetuspelin kehitys lukiokäyttöön**

Henri Lindström & Ari Saaristo  
Opinnäytetyö  
Kevät 2022  
Tietojenkäsittelyn tutkinto-ohjelma  
Oulun ammattikorkeakoulu

## TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Tradenomi, Tietojenkäsittelyn tutkinto-ohjelma

---

Tekijä(t): Henri Lindström, Ari Saaristo

Opinnäytetyön nimi: Filosofia-aiheisen opetuspelin kehitys lukiokäyttöön

Työn ohjaaja(t): Matti Viitala, Reima Riihimäki

Työn valmistumislukukausi ja -vuosi: kevät 2022

Sivumäärä: 37

---

Tämän opinnäytetyön tavoitteena oli tehdä filosofiapeli Oulun Suomalaisen Yhteiskoulun lukiolle. Peli sai alkunsa TIKO-silta-hankkeen kautta toimeksiantona, jonka jälkeen se jatkui jatkokehityksenä ja opinnäytetyönä.

Käytimme pelin tekemiseen Unity-pelimoottoria, ja ohjelmointikielenä Unityssä käytimme C#-ohjelmointikieltä. Kuvat sekä taustalle että kortteihin löysimme internetistä, ja niiden muokkaamiseen käytimme Adobe Photoshopia ja Illustratoria.

Opinnäytetyössä kerromme tarkemmin hyötypelistä ja niiden tavoitteista, sekä käymme läpi asioita pelin suunnittelusta, työkaluista ja itse toteutuksesta, mutta myös ongelmista, joita kohtasimme rakentaessamme peliä.

Lopputuloksena syntyi "sananselitys"-hyötypeli, jossa käydään läpi lukion filosofiankurssien opetussuunnitelman osioiden erilaisia käsitteitä ja kysymyksiä, joita opiskelijat voivat joko koulussa tai vapaa-ajallaan käydä läpi ja pohtia niitä yhdessä muiden kanssa.

Asiasanat: Hyötypelit, pelisuunnittelu, käyttöliittymät

## ABSTRACT

Oulu University of Applied Sciences  
Degree Programme in Bachelor of Business Administration, Option of Information Technology

---

Author(s): Henri Lindström, Ari Saaristo

Title of thesis: Development of an educational game on philosophy for high school use

Supervisor(s): Matti Viitala, Reima Riihimäki

Term and year when the thesis was submitted: Spring Term 2022

Number of pages: 37

---

The aim of this thesis was to make a philosophy game for a Finnish secondary school in Oulu. We started developing the game in fall 2021 as a project, after which it was further developed as a thesis.

We used the Unity game engine to make the game, and the programming language in Unity was C#. We found the images for both the background and the cards from the internet and used Adobe Photoshop and Illustrator to edit them.

In the thesis we will talk in more detail about serious games and their goals, and we will go through things about designing the game, tools we used and the implementation of the game itself, but also about the problems we encountered while building the game.

The result was a "word explaining" serious game, which goes through different concepts and questions from the curriculum sections of high school philosophy courses, which students can either go through in school, or in their free time, and reflect on them together with others.

Keywords: Serious games, game design, user interfaces

# SISÄLLYS

1	JOHDANTO .....	6
2	HYÖTYPELI .....	7
3	PELIN SUUNNITTELU .....	10
3.1	Pelin perusidea.....	12
3.2	Pelin suunnittelun eteneminen .....	13
4	TYÖVÄLINEET .....	18
4.1	Unity .....	19
4.2	Photoshop ja Illustrator.....	22
5	ADMIN-LIITTYMÄ.....	24
6	TEKNINEN TOTEUTUS .....	26
6.1	Pelin tekninen kehitys.....	26
6.2	WebGL Build .....	31
6.3	Ongelmat teknisessä kehityksessä .....	33
7	POHDINTA .....	34
	LÄHTEET.....	35

# 1 JOHDANTO

Opinnäytetyön tavoitteena oli kehittää peli tukemaan filosofian opiskelua ja opetusta. Peli kehitettiin Oulun Suomalaisen Yhteiskoulun lukiolle (OSYK). Kohderyhmänä toimivat lukion filosofian opiskelijat ja opettaja. Kehitys alkoi tietojenkäsittelyn opintoihin kuuluvana toimeksiannota, joka jatkui jatkokehityksenä ja opinnäytetyönä. Aluksi, toimeksiannon aikana, lähdimme suunnittelemaan peliä lähemmäs muistipeli- ja quiz-tyylistä peliä, jossa olisi kysymykset ja oikeat vastaukset, sekä peli etenisi tasolta tasolle saatujen pisteiden mukaan. Kun opettaja oli saanut uuden opetussuunnitelman mukaiset oppikirjat ja kysymykset laadittua, huomasimme, ettei muistipeli- tai quiz-tyylinen peli olisi kovin relevantti, eikä näin tukisi opetusta aivan halutulla tavalla. Koska filosofiassa käsitteet ja aiheet ovat monesti laajempia ja ymmärrys perustuu pohdinnalle, keskustelulle ja argumentaatiolle, peli muokkaantuikin neljän lukion filosofiankurssin mukaiseksi korttipeliksi, jossa korteissa olisi kysymykset ja käsitteet, joita opiskelijat voisivat pohtia ja keskustella ryhmissä.

Pelin tarkoituksena on auttaa opiskelijoita ja opettajaa tuntityöskentelyssä. Tämäntyylinen hyöty-peli tuo oppimisen myös enemmän nykyaikaan, kun suurimmalta osalta löytyy jonkinlainen älylaite. Opiskelijoiden on helppo käydä käsitteitä ja kysymyksiä läpi ryhmässä, kun riittää, että yhdellä on laite, jolla pääsee internetiin. Peli helpottaa myös siinä mielessä oppimista, kun ei tarvitse käydä käsin kymmeniä eri kortteja läpi, vaan riittää, että nappia painamalla pääsee eteenpäin ja keskustelemaan seuraavasta aiheesta.

Peli toteutettiin web-selaimella pelattavassa muodossa. Tarkoituksena oli myös mahdollistaa pelisisällön päivitys opetussuunnitelman mukaiseksi luomalla opettajalle oikeudet ja ohjeet pelin päivittämiseksi. Kehittämisessä käytettiin Unity-pelimoottoria ja Adoben grafiikka- ja animointityökaluja.

## 2 HYÖTYPELI

Hyötypelit ovat muutakin kuin viihdekäyttöön suunniteltuja pelejä. Kuten viihdepeleissä, myös hyötypeleissä on jokin tavoite, jota yritetään saavuttaa. Tämä tavoite voi olla esimerkiksi jonkin asian oppiminen, terveyden ja kunnon edistäminen tai asenteen muuttaminen.

Hyötypelityypit jaetaan periaatteessa kahden kategorian alle: prosessipainotteisiin ja lopputulos-painotteisiin peleihin. Prosessipainotteisilla peleillä tarkoitetaan lähinnä tutkimus-, päätöksenteko- ja simulaatiopelieitä, joilla ei ole varsinaista pelillistä lopputulosta, vaan oppimisen olisi tarkoitus tapahtua itse peliprosessin tai pelikokemuksen aikana. Lopputulospainotteiset pelit puolestaan tähtäävät uusien taitojen oppimiseen pelissä edistymisen ja pelin sisäisten tavoitteiden tai tasojen suorittamisen kautta. Tähän kategoriaan kuuluvat erilaiset opetus-, motivointi- ja vaikuttamiseen pyrkivät pelit. (Serious Games Guide 2021.)

Simulaatiopelieitä käytetään niin ilmailualalla kuin nykyisessä ajo-opetuksessakin, mutta myös opetuksessa kouluissa ja terveydenhuollossa. Esimerkiksi Stop Disasters! -peli opettaa oppilaille simuloimalla, kuinka katastrofit tai onnettomuudet vaikuttavat ihmisiin ja kuinka toimia kyseisissä tilanteissa. Pulse!! -pelissä puolestaan otetaan hoitajan tai lääkärin rooli ja pyritään hyödyntämään opittuja asioita simuloidussa ympäristössä ilman todellista riskiä. Akili Interactive on kehittänyt jopa ensimmäisen reseptinä ADHD:n hoitoon käytettävän pelin EndeavourRX:n, jonka on todettu auttavan lasta keskittymään ja suoriutumaan useista tehtävistä paremmin. (King, Nico 2021.)

Minecraft Education Editionin avulla voidaan opettaa koululaisille mitä oppiainetta tahansa. Esimerkiksi koodausta, matematiikkaa, kieliä ja taideaineita. Pelin verkkosivulta voi ladata oppiaineisiin liittyviä oppitunteja, opetusohjelmia ja pelimaailmoja, joiden avulla oppiminen tapahtuu, tai osallistua vedettyihin opetusryhmiin. (Minecraft Education Edition 2022.)

Suomi ei ehkä ole mikään suurin maa, mutta korkealaatuinen opetus ja vaikuttava oppimisen taso on saattanut Suomen kuitenkin kärkimaiden joukkoon oppimisen ja opettamisen suhteen, jossa Suomi on jo pitkään pitänyt paikkaansa. Mutta hyötypelien suhteen Suomi näyttää olevan vielä

vähän jäljessä, vaikka jonkin verran pelejä käytetäänkin jo opetuksessa, niin nykyään kun maailma digitalisoituu entistä hurjempaa vauhtia, hyötypelitkin soveltuvat enemmän ja enemmän nykyaikaiseen oppimisympäristöön. (Santti, Heikki & Virtanen, Mari 2018; Stat 2022.)

Vuonna 2018 Suomen vuoden hyötypeliksi valittiin Turku Game Labin ja Humakin kehittämä *Lights on!* -hyötypeli. Pelissä tutustutaan osaan Suomesta ja Virossa löytyviin raunoihin ja paikkoihin, jotka ovat kulttuurihistoriallisesti merkittäviä. Pelin avulla oppii tuntemaan näiden paikkojen historiaa sekä kasvattamaan omaa kulttuuritietämystään, ja lisäksi peli kannustaa pelaajia liikkumaan. Pelissä käytetään lisättyä todellisuutta, jotta pelaaja voi itse kokea historian olemalla vuorovaikutuksessa menneisyyden hahmojen kanssa. (Suominen, Taisto & Luimula, Mika 2019.)

NordicEdu perustettiin vuonna 2011. Se keskittyi kehittämään erilaisia opetus- ja hyötypeliejä ja oli niiden edelläkävijä, kunnes vuonna 2021 luopui hyötypelien kehittämisestä. Mutta sitä ennen heillä kerkesi jo kertyä portfolioon yli 20 peliä, jossa kohtaavat eri kohderyhmien liiketoiminnalliset että opetukselliset tavoitteet. (Business Finland 2016.)

Esimerkkinä NordicEdun kehittämistä peleistä on muun muassa MobiMetsä-peli, joka on kehitetty Suomen partiolaisten ja metsäyhtiö UPM:n kanssa, jossa pelaaja oppii metsätietoutta ja pelaaja pääsee myös itse luontoon liikkumaan. Päättävänä pelissä oli saada lapset ja nuoret oppimaan lisää metsänhoidosta ja eri puulajeista, sekä metsän näkeminen monipuolisena hyvinvoinnin lähteenä. (Nordicedu 2022.)

Toisena esimerkkinä NordicEdun peleistä on tupakan vaaroihin liittyvä minipelikokoelma Fume, joka on kehitetty yhdessä Turun yliopiston hoitotieteen laitoksen kanssa NO! to smoking -projektia varten. Projektin ja pelin tarkoituksena on valistaa tupakkakielteisyyttä ja tutkia sen vaikutusta nuoriin. Lähtökohdaksi oli humoristinen tupakan vaarallisuudesta varoittava peli, jota käytettäisiin oppitunnilla, joka liioittelee tupakan vaarallisuudesta verrattuna terveystiedon tunneilla käytäviin asioihin. Opettaja sitten kertoisi tarkemmin ja vakavammin tupakan oikeista vaaroista. (Nordicedu 2022.)

Kehittelemämme filosofiaoppimispeli ”Kysy Manelta” keskittyy auttamaan opiskelijoita käymään ja ymmärtämään erilaisia filosofisia käsitteitä keskustelemalla niistä muiden opiskelijoiden kanssa ryhmässä. Koska filosofiassa käsitteet ovat laajoja, eikä niissä aina ole vain yhtä oikeaa vastausta, auttaa peli myös opiskelijoita kuuntelemaan muiden pohdintoja ja ymmärtämään asioita



toisten näkökulmasta katsoen. Joten sen lisäksi, että peli auttaa ymmärtämään filosofian käsitteitä syvemmin, hyötty pelistä myös siinä mielessä, että se voi auttaa parantamaan asennetta, koska peliä pelatessa täytyy myös ajatella ja kuunnella muidenkin henkilöiden mielipiteitä eikä vain omiaan, ja se taas myös auttaa opettamaan tiimissä työskentelyä.

### 3 PELIN SUUNNITTELU

Toimeksiantajallamme, joka otti meihin alun perin yhteyttä Oulun Suomalaisen Yhteiskoulun lukion puolelta TIKO-silta-hankkeen kautta, oli jo alun perin ideaa siitä, millainen pelistä voisi tulla. Hänen ideastaan lähdimme sitten kehittämään ideaa eteenpäin, ja alkuperäinen suunnitelma muuttuikin siitä sitä mukaa, kun itse peliä lähti rakentamaan.

Kun aloitimme suunnittelemaan ja kehittämään peliä, päätimme myös pitää palavereita aina parin kolmen viikon välein, jotta pystyimme näyttämään pelin edistymistä ja ideoimaan peliä pidemmälle yhdessä työnantajamme kanssa. Pääsääntöisesti tapasimme OSYK lukion tiloissa, mutta muutamana kertana pidimme tapaamiset etänä, Microsoft Teams -yhteistyöalustan kautta. Tapaamisissa olivat pääsääntöisesti me, toimeksiantajamme sekä meidän opettajamme.

Puhuimme myös yhdessä tapaamisessa pelin käyttöoikeuksista ja tulimme siihen tulokseen, että oikeudet peliin menevät Oulun Suomalaisen Yhteiskoulun lukiolle, jotta he voivat halutessaan tehdä toisenlaisia pelejä esimerkiksi muille oppiaineille, tämän meidän tekemämme pelin pohjalta.

Pixabaysta löysimme kuvia (taustakuvat, Sokrates ja Platon), joita pystyi ilmaiseksi käyttämään, eikä meidän tarvinnut lähteä maksamaan niistä erikseen. Valitsimme Sokrateen ja Platonin peliin, koska he ovat merkittävimmät ja vaikutusvaltaisimmat filosofit historiassa. Sokratesta kuvataan usein länsimaisen filosofian isänä ja hän oli myös itse Platonin opettaja, ja Platon on Sokrateen kuuluisin oppilas. (Wikipedia 2022; Wikipedia 2023.)

Peli alkoi ensin ajatuksena, että siitä tulisi muistipeli tai quiz-peli, jossa ensin etsittäisiin kysymys ja sille sitten vastaus. Mutta opetuksen kannalta päädyimme sellaiseen ajatukseen, että pelissä olisi vain yksi pakka, josta nousee aina kysymys kerrallaan, ja opiskelijat voisivat ryhmässä tunnilla tai tauolla pelailla peliä ja harjoitella eri käsitteitä.

Aluksi pelin grafiikka oli todella yksinkertainen, kaksi filosofia valkoisella taustalla. Pelin kehittyessä lisäsimme siihen vähän väriä ja muita kuvia, jotta siitä tulisi enemmän silmään pistävämpi. (katso kuvio 1.) Muutamaan tapaamiseen saimme mukaan lukion kuvataiteen opettajan, joka antoi palautetta väreistä, fonteista ja grafiikoista ylipäättänsä, minkä vuoksi pelin yleisilme muuttui miellyttävämmän näköiseksi. Esimerkiksi aloitusnäytön kuvassa oli vähän liikaa komponentteja

opettajan mielestä, joten poistimme siitä osan ja lisäksi muutimme väriä, jonka jälkeen kuvasta tuli selkeämpi ja paremman näköinen. Sen lisäksi kurssialueiden pohdintakorttien ja taustakuvan väri oli hänen mielestään hyvä olla yhtenäinen, jotta peli näyttäisi paremmalta. Saimme myös pelistä käyttäjäystävällisemmän, kun aloimme miettimään, että joillakin voi olla esimerkiksi väri-sokeus, niin heitäkin ajatellen mietimme tarkemmin kuvien väritystä ja saavutettavuutta.



KUVIO 1. Kuva aloitusvalikosta muokkaamattomana

Kurssialueiden osat, jotka ovat oikealla puolella, olivat aluksi kaikki oikeassa yläkulmassa, ja kenen ehdotuksestaan laitoimme ne tasaisesti, eli yksi yläkulmaan, toinen keskelle ja kolmas alakulmaan, joten saimme ikoneita isommiksi ja näyttävämmän näköisiksi. Aloitusnäytön ikonit sekä Maunon (Yksi OSYK:in perustaja jäsenistä.) kuvan keskellä ja kaikkien ikoneiden tekstit samalla myös muutimme isommiksi, jotta niistä saa helpommin selvää pienemmälläkin näytöllä.

Kuvataiteen opettajan lisäksi muutamassa tapaamisessa oli mukana lukion tietotekniikan opettaja ja hänen kanssaan keskustelimme enemmän pelin teknisistä asioista. Päätimme tehdä pelin WebGL-buildina, jotta peliä voi pelata verkkoselaimessa, eikä sitä tarvitse erikseen ladata puhelimeen tai tablettiin. Koska peli tehtiin verkkoselaimessa pelattavaksi, niin keskustelimme tietotekniikan opettajan kanssa myös pelin asentamisesta palvelimelle.

Koska meitä oli kaksi ihmistä tekemässä yhdessä tätä peliä, käytimme GitHub-sovellusta, jolloin pystyimme tekemään itsenäisesti muutoksia peliin ja helposti jakamaan ne toistemme kanssa.

GitHubiin luotiin pelille repositorio, joka onnistuu Unity-projektille varsin helposti, ja valikosta voi automaattisesti valita gitignore-tiedoston tekemisen Unitylle. Tätä tiedostoa joutuu muokkaamaan hieman tekstieditorissa tiedostojen oikean polun osoittamiseksi. Gitignore-tiedoston tarkoitus on poistaa turhien tiedostojen jakaminen versionhallinnassa, kuten Unityn log-, temp-, library- tai MemoryCaptures-tiedostoja. GitHub-repositorion luoja antaa tämä jälkeen projektissa mukana olleille oikeudet repositorioon, ja he voivat kloonata tiedoston koneelleen. Muokkaaminen tapahtuu jokaisen omalla koneella itse Unityssä, ja muutokset siirretään GitHubin repositorioon, joko komentoriviltä tai käyttämällä esimerkiksi GitHub Desktop-sovellusta, josta muut projektillaiset voivat ladata päivitykset omalle koneelleen.

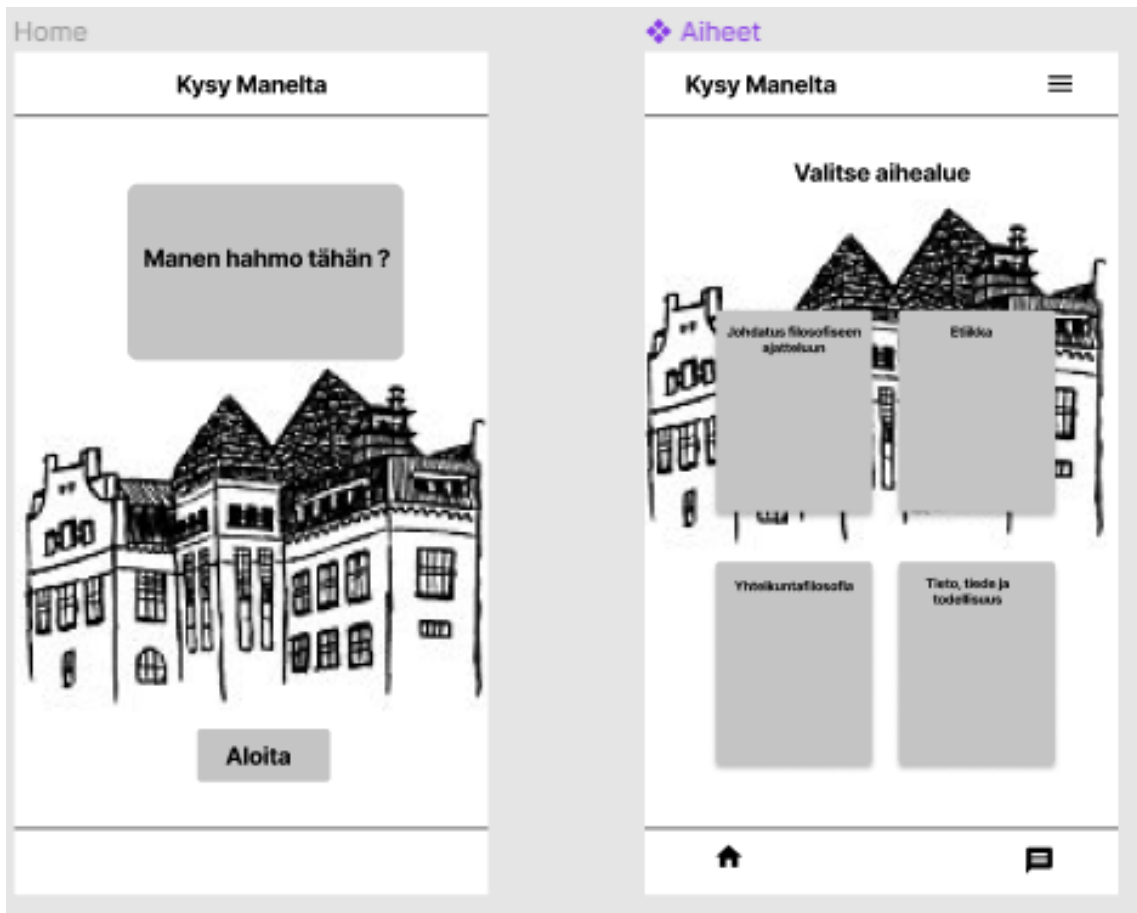
### 3.1 Pelin perusidea

Peli jakaantuu lukion filosofiankurssien opetussuunnitelman mukaisesti neljään eri osioon, joissa on omat alaosiionsa, ja jokaisessa osiossa käydään läpi kyseisen osion aihealueeseen liittyviä kysymyksiä ja eri käsitteitä. Näitä erilaisia käsitteitä tai kysymyksiä löytyy pelistä paljon. Esimerkkejä käsitteistä on muun muassa Sokrates, opiskelijat keskustelevat kuka hän on ja mistä hänet tunnetaan. Toinen käsite voi olla etiikka, ja opiskelijat käyvät läpi, mitä se tarkoittaa.

Jokaiselle aihealueelle on oma skenensä, joihin siirrytään aloituskenen kanvaksella olevan aloitusvalikon ikonia painamalla. Skene on pelin eräänlainen näyttämö tai yleisemmin pelialue. Tässä peli on kuitenkin rakennettu pelialueen päälle luodulle käyttöliittymäelementille, kanvakselle, jonka päällä olevien käyttöliittymäelementtien avulla skenejen välillä liikutaan. Valitulla skenellä on aihealueeseen liittyvät kysymyskortit. Kysymyskorttia painamalla paljastuu kysymys tai pohdinta, joita opiskelijat voivat yhdessä tunnilla käydä läpi. Sen jälkeen, kun kysymys tai pohdinta on käyty läpi, poistuu kortti pakasta, jotta se ei tulisi uudestaan vastaan, kun pakkaa käydään läpi. Sitten jos pelistä poistuu ja tulee takaisin, pakka sekoittuu uudestaan ja siinä on taas kaikki kortit pakassa.

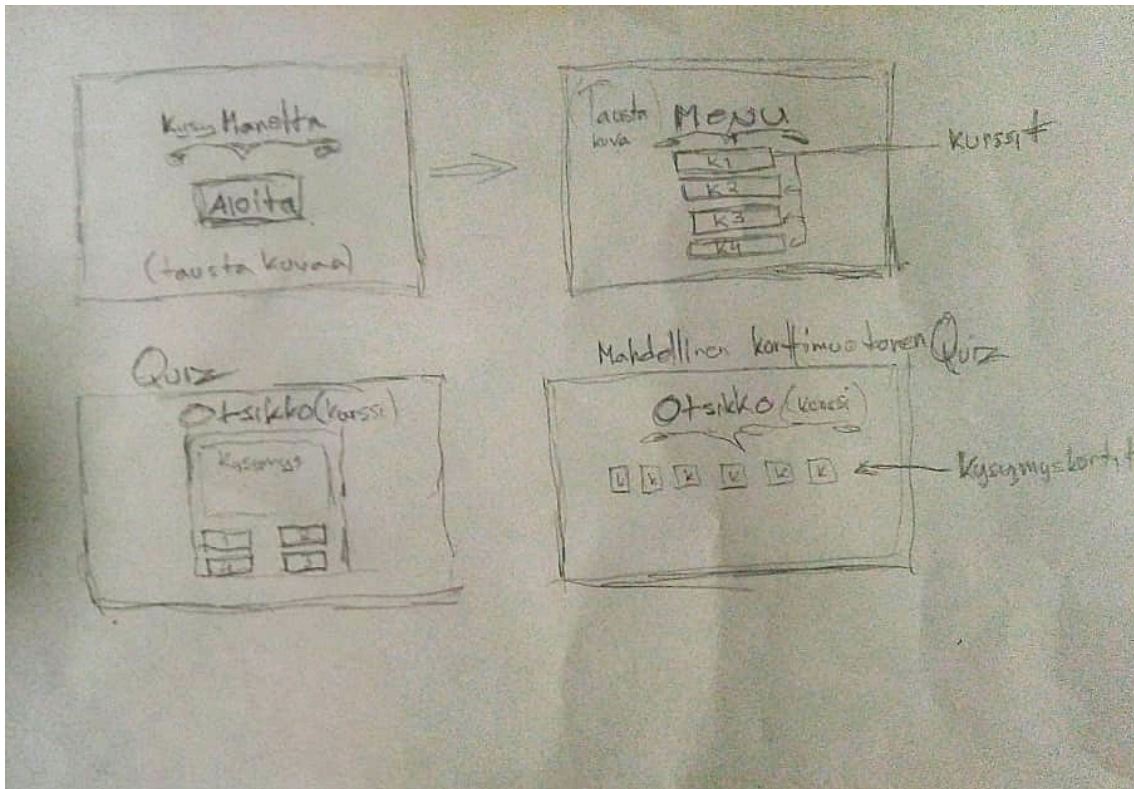
### 3.2 Pelin suunnittelun eteneminen

Pelin suunnittelun alkuvaiheessa täytyi päättää käytettävä kehitysympäristö. Monien vaihtoehtojen joukosta valikoitui kolme: React, Angular ja Unity, joiden välillä mietimme pidempään. Ensimmäinen rautalankaversio tehtiinkin Figma-suunnitteluohjelmalla mobiilinäkymäksi, silmällä pitäen Reactia ja Angularia (kuvio 2).



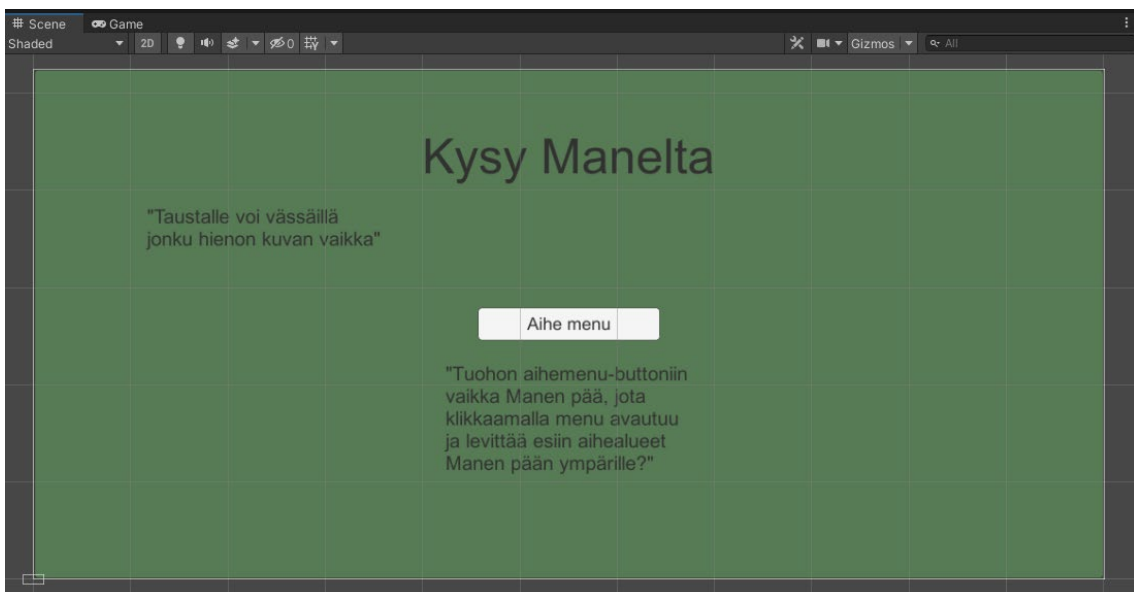
KUVIO 2. Kuva Figmalla suunnitellusta rautalankamallista.

Unity valikoitui lopulta kehitystyökaluksi monipuolisuutensa ja ominaisuuksiensa takia. Aluksi hahmottelua tehtiin hiukan paperille (kuvio 3). Paperilla ja kynällä hahmotteleminen on varsin nopea ja vaivaton tapa luoda juuri esimerkiksi yksinkertaisia UI/UX-rautalankaversioita, joihin voi hahmotella oleelliset asiat. Kuvasta on nähtävissä aloitusruutu, jossa on otsikko ja aloita-nappi, josta päästään kurssimenuun. Kurssin valittuaan käyttäjälle aukeaisi aiheeseen liittyvä joko monivalinta- tai kysymyskorttipohjainen visailu. Pelissä päädyttiin kuitenkin korttipohjaiseen toteutukseen.



KUVIO 3. Kuva alkuvaiheen paperi rautalankamallista

Suunnittelua ja toteutusta pyrittiin jatkamaan suoraan itse Unity-ympäristössä, jossa ideoiden ja toiminnallisuuden kokeileminen onnistuu varsin sujuvasti. (kuvio 4).



KUVIO 4. Kuva ensimmäisestä aloituskenen suunnitelmasta Unity-ympäristössä.

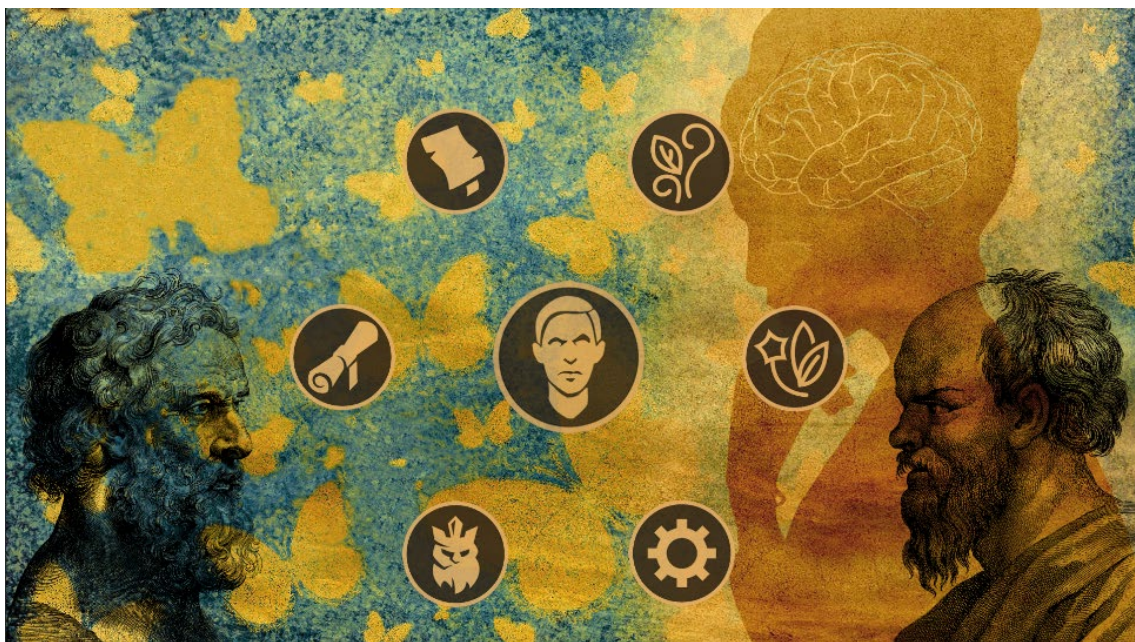


Ensimmäisessä toimeksiantajalle esitellyssä versiossa oli valkoisella taustalla kaksi filosofia ja pääikoni valikkopainikkeena. Pään kuvaa klikkaamalla aukeaa sen ympärille kurssien ikonit, joita klikkaamalla pelissä liikutaan aihealueiden välillä (kuvio 5).

Idea lähdettiin hiukan viemään eteenpäin lisäämällä tausta ja väriä. Taustaa muokattiin Photoshopissa siten, että poistettiin vasemmalta ylimääräisenä kuvassa olevat kello ja tiimalasi sekä säädettiin hiukan värejä kirkkaammaksi ja poistettiin hiukan vihertävyyttä (kuvio 6).



KUVIO 5. Kuva ensimmäisestä toimeksiantajalle esitellystä aloitusskenestä.



KUVIO 6. Kuva aloitusvalikosta muokattuna.

Ensimmäisen kurssin näkyvässä olivat aluksi kortit ilman taustakuvaa mustina laatikkoina ja ilman rajattua pelialuetta. Tätä muokattiin lisäämällä läpinäkyvä pelialue ja vaihtamalla kortteihin kurssialueeseen sopiva taustakuva, otsikkofontti ja värimaisema, joka oli kuvataiteenopettajan mukaan sopiva teemaan (kuvio 7).



KUVIO 7. Kuva ensimmäisen kurssin skenestä.

Tästä lähdimme kehittämään peliä lähemmäs korttipakkamaista toteutusta, jossa kortit olisivat pakassa päällekkäin. Kysymyskortit sekoitetaan ja ne tulevat päällekkäin satunnaisjärjestyksessä. Päälimmäistä korttia painamalla paljastuu keskusteltava aihe, jonka jälkeen kortti poistuu pakasta ja pelaaja voi valita uuden kortin. Halutessaan pelaaja voi myös sekoittaa pakan uudelleen shuffle- napista, jolloin kaikki kortit tulevat peliin mukaan satunnaisjärjestyksessä (kuvio 8).





KUVIO 8. Ensimmäisen kurssin aloitusnäkyssä kysymyskortti auki.

## 4 TYÖVÄLINEET

Kun suunnittelimme peliä, meidän piti päättää, millä pelimoottorilla lähdemme sitä rakentamaan, joten etsimme netistä erilaisia suosittuja pelimoottoreita. Eri sivuja selatessamme, suosituimpiin ja käytetyimpiin pelimoottoreihin kuuluivat muun muassa Unity, Unreal Engine, Godot, CryEngine ja tietty monia muitakin.

Unreal Engine on hyvä huippuluokan grafiikan luomiseen, ja pelimoottorina se on suorituskykyisempi kuin monet muut pelimoottorit, mutta se on enemmän luotu isompiin projekteihin ja projekteihin, joissa työskennellään tiimissä. Se on ennemminkin luotu 3D-pelien tekemiseen, eikä niinkään yksinkertaisten pelien tekemiseen, vaikka nekin onnistuvat sillä. Esimerkkejä Unreal Enginellä tehdyistä peleistä ovat: *Soulcalibur VI*, *Gears 5* sekä *Dargon Quest XI*. (Schardon, Lindsay 2022.)

Godot-pelimoottori on ilmainen ja sillä voi rakentaa sekä 2- että 3D-pelejä, joten se sopii hyvin erilaisten pelien tekemiseen. Huono puoli tässä on se, että Godot käyttää omaa henkilökohtaista kieltä koodaamiseen, nimeltään GDScript, ja se ei välttämättä ole kaikkien mieleen, kun joutuu opettelemaan taas yhden koodikielen, jota käytetään vain Godotissa. Esimerkkejä Godotilla tehdyistä peleistä ovat: *ProtoCorgi*, *Deep Sixed* sekä *Gun-Toting Cats*. (Schardon, Lindsay 2022.)

CryEngine pelimoottorina on enemmän kokeneimmille pelintekijöille, eikä ihan aloittelijoille tehty. Sillä voi luoda näyttäviä ympäristöjä ja grafiikoita peleihin. Esimerkkejä CryEnginellä luoduista peleistä ovat: *Crysis*, *Far Cry* sekä *Kingdom Comes: Deliverance* (Natvidad, Sid 2022; Wirtz, Bryan 2023.)

Oikean pelimoottorin valinta on tärkeää pelin rakentamiselle, ja siihen liittyy monta eri tekijää, esimerkiksi se millaista peliä rakentaa, tuleeko peli 2D:nä vai 3D:nä, onko aiempaa kokemusta pelin tekemisestä ja millaisia funktioita peliin sisältyy. Näiden tekijöiden kannalta meidän täytyi vertailla näitä eri pelimoottoreita, jotta saimme valittua meille sopivimman, ja näistä päädyimme valitsemaan Unityn.

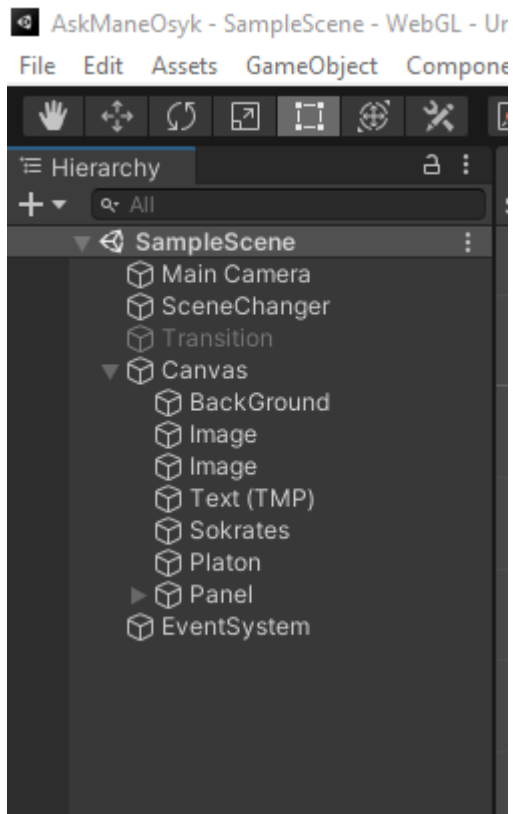
## 4.1 Unity

Unity on Unity Technologiesin 2D- ja 3D-pelien luomiseen tarkoitettu alustariippumaton pelimoottori (Wikipedia 2023). Pelimoottorin voi ladata ilmaiseksi Unityn sivuilta, mutta siitä on olemassa ilmaisversion lisäksi myös Plus-versio, joka maksaa 399 dollaria vuodessa, sekä Pro-versio, joka maksaa 1 800 dollaria vuodessa. Teollisuus- ja yritysversiot löytyvät myös erikseen maksaen 2 520 dollaria tai 4 000 dollaria vuodelta. Tämän pelin toteuttamiseen olemme käyttäneet ilmaisversiota. (Unity 2022).

Unity on hyvä pelimoottori aloittelijalle, Unityä voi käyttää ilmaiseksi niin kauan, kunnes tulot ovat alle 100 000, tämän jälkeen täytyy vaihtaa johonkin maksulliseen versioon. Käyttö on nopeaa ja pelin tekeminen onnistuu myös ilman koodaustakin tai vain vähäisellä koodauksella. Ohjelmointikielenä toimii C#. Unityn käyttöön löytyy heiltä itseltään ohjevideo, sekä esimerkiksi Youtubesta löytyy valtava määrä erilaisia opetusvideoita Unityn käytöstä sekä pelien tekemisestä, joten käytön opetteleminen ja aloittaminen ei ole mikään ongelma. Unityltä löytyy myös oma asset store, jossa on tuhansia erilaisia paketteja, joita voi hyödyntää pelin rakentamisessa, osa näistä paketeista on ilmaisia ja osa taas maksullisia.

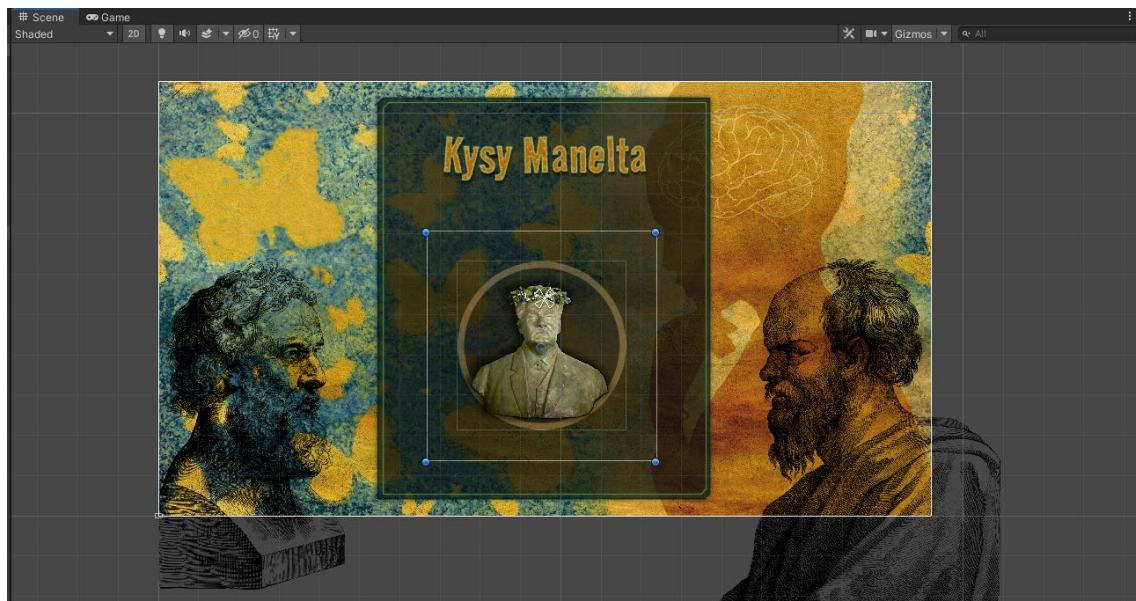
Unityn Asset Storesta hyödynsimme ilmaisia Clean Vector Icons- ja LeanTween-paketteja. Pelin kortteihin luotiin animaatiota LeanTween-paketin avulla. Animaatio toteutettiin C#-koodikielellä ja liittämällä se peliobjektiin.

Unityn käyttöliittymänäkymä koostuu kuudesta pääikkunasta: hierarkia-, skene-, peli-, projekti-, konsoli ja ominaisuusikkunoista. Hierarkiaikkunassa näkyvät kyseisen auki olevan skenen kaikki skenellä olevat peliobjektit ja niiden liitokset toisiinsa (kuvio 9).



KUVIO 9. Hierarkiaikkuna.

Skeneikkunassa näkyvät hierarkiassa olevat peliobjektit ja miten ne jäsenyivät kyseiselle skeneelle. Skene-näkymässä tapahtuu myös koko peliskenen rakentaminen ja peliobjektien muokkaaminen (kuvio 10).



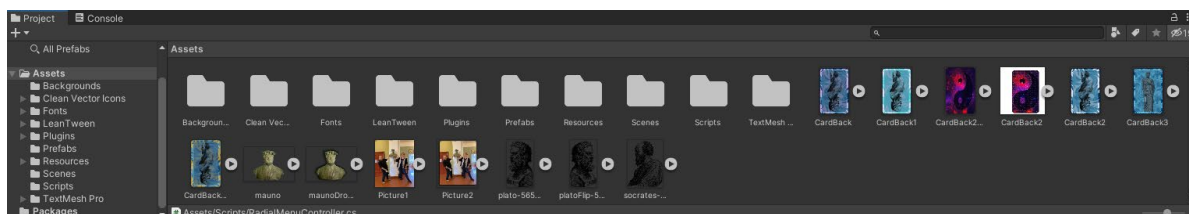
KUVIO 10. Skene-näkymä.

Pelinäkymässä ei itse editointia pysty tekemään, vaan tässä näkymässä voidaan kokeilla, miltä peli näyttää skenellä olevan kameran kautta ja miten peli toimii (kuvio11).



KUVIO 11. Peli-näkymä.

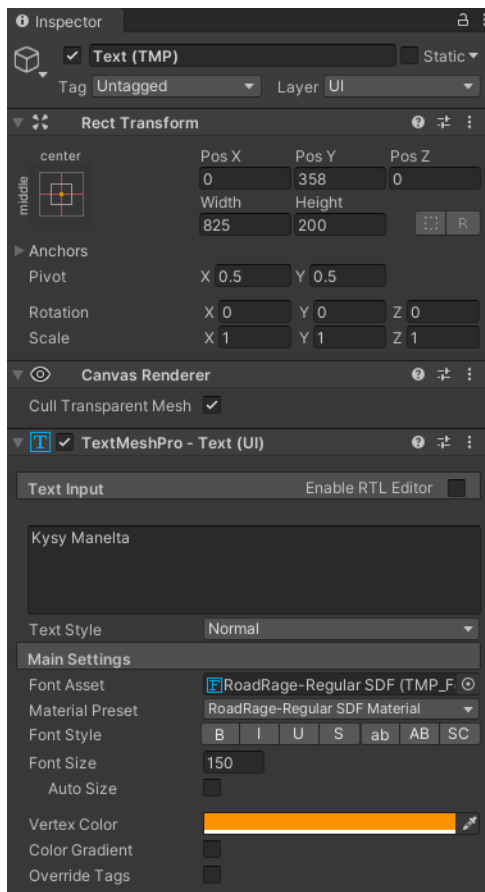
Projekti-ikkunassa näkyvät kaikki pelin rakennuspalikat tai tiedostot, Assetit. Täältä löytyvät myös pelin kaikki skenet ja peliin tuodut assetit. Konsoli-ikkunaan tulostuvat virhe-, varoitus- ja muut editorin luomat ilmoitukset, esimerkiksi skripteissä olevat käänösvirheet. Konsoli-ikkunassa näkyvät myös skripteissä hyödynnetyt Debug.Log()-viestit (kuvio 12).



KUVIO 12. Projekti-ikkuna ja välilehtenä konsoli-ikkuna.

Ominaisuusikkunassa näkyvät valitun peliobjektin ominaisuudet ja asetukset, joita voidaan täältä myös editoida (kuvio13).





KUVIO 13. Ominaisuus -ikkuna.

## 4.2 Photoshop ja Illustrator

Pelin grafiikkaa tehtiin Adoben ohjelmistoilla, pääasiallisesti Photoshopilla ja Illustratorilla. Pelin kortit on suunniteltu Illustratoria käyttäen ja ikoneita ja kuvia on muokattu Photoshopilla. Unityssä on mahdollista käyttää 2D PSD Importer-pakettia, jolloin Photoshopin omia PSD-tiedostoja voidaan tallentaa ja viedä Unityyn PSB-muodossa. Tämä mahdollistaa häviöttömän siirron, ja mukana siirtyvät kaikki kuvan tasot tai layerit ja suurimmat osat kuvan ominaisuuksista, mitkä olivat läsnä Photoshop-ympäristössäkin. Tuotaessa PSB-kuva Unityyn, tekee Unity siitä suoraan prefabin hierarkiaikkunaan, jossa näkyvät kaikki kuvan tasot tai layerit omina lapsiobjekteina, joita on helppo muokata ja käsitellä. Prefab on eräänlainen sapluuna, johon voidaan varastoida peliobjektin kaikki ominaisuudet, lapsiobjektit ja komponentit ja tallentaa se asettina. Prefabia voi sitten helposti käyttää uudestaan. Tämä nopeuttaa huomattavasti isojen, monikerroksisten asettejen

käsittelyä ja työkulkua. Prefab asset nimitystä käytetään kaikista pelin rakennuspalikoista tai tiedostoista, joita voidaan tuoda Unityn ulkopuolelta tai rakentaa toisilla ohjelmilla kuten Photoshopilla tai Illustratorilla. Tässä kuitenkin kortit ja kuvakkeet olivat varsin pieniä ja yksinkertaisia, niin ne tallennettiin myös PNG-formaatissa itse Illustratorissa (kuvio 14) ja Photoshopissa (kuvio 15), ja tuotiin sitten vasta vedä ja pudotatoiminnolla, hiirellä vetäen aseteiksi. (Oriz, Eduardo & Zachariah, Shanti & Stock, Kristin 2021; Unity 2022.)



KUVIO 14. Kuvakaappaus pelikortin takaosasta.



KUVIO 15. Kuvankaappaus ikoneista.

## 5 ADMIN-LIITTYMÄ

Admin-käyttöliittymän tekeminen oli tärkeä ja tarpeellinen osa pelin kehittämisessä, koska kun kehitimme peliä, lisäsimme peliin kysymykset ja käsitteet, jotka saimme toimeksiantajaltamme. Juuri oli kuitenkin tulossa uudet opetus suunnitelman mukaiset oppikirjat, joissa on uusia käsitteitä, niin emme voineet lisätä kaikkia sinne kerralla. Joten päätimme tehdä admin-käyttöliittymän toimeksiantajallemme, jotta hän voi itse lisätä uudet käsitteet sitä mukaa, kunhan uudet oppikirjat ovat julkistettu. Admin-liittymän tarkoitus oli myös lisätä mahdollisuus päivittää ja vaihtaa uudet kysymykset vanhojen tilalle niin haluttaessa.

Mietimme erilaisia vaihtoehtoja, esimerkiksi CSV, JSON ja XML. Pääasiassa etsimme Unityn Asset Storesta, jos sieltä löytyisi jotain, mikä olisi mahdollisimman yksinkertainen tapa, mitä voisi käyttää tiedon tallentamiseen peliin. Tämä oli tärkeää, jotta saataisiin tehtyä käsitteiden lisääminen mahdollisimman helpoksi toimeksiantajallemme, ilman turhaa vääntöä, että ei tarvitsisi ladata ylimääräisiä ohjelmia ja sitten opettaa niiden käyttöä hänelle. Katsoimme myös YouTubeista erilaisia videoita, mitä muut pelintekijät ovat käyttäneet heidän applikaatioissaan. Myöskin GitHubista etsimme, josko sieltä löytyisi valmiiksi muiden tekemiä tiedostoja ladattavaksi ja käytettäväksi meille.

Päädyimme lopulta valitsemaan CSV-tallennusmuodon, koska se tuntui toimeksiantajaa ajatellen käytännöllisimmältä. Valinta tuntui myös helpoimmalta siksi, että pelissä ei ole kortteihin haettavia oikeita vastauksia, vaan vain pelkät kysymykset tai käsitteet, joita opiskelijat tunnilla pohtivat ja joista keskustelevat. CSV on tiedostomuoto, jolla pystytään tallentamaan taulukkomuotoista tietoa tekstitiedostoksi. CSV-lyhenne tulee sanoista comma-separated values, joka nimensä mukaisesti erottaa taulukkorakenteen kentät toisistaan pilkuilla ja rivinvaihdolla. CSV-tiedostot toteutettiin siten, että jokaiselle kurssialueelle tehtiin yksi oma CSV-tiedosto, johon kysymykset olivat jaettu kolumneihin kurssialueen aiheiden mukaan ja josta koodilla haettiin riviltä aihealueen korttiin oikea kysymys. Esimerkkinä kuvakaappaus ensimmäisen kurssin CSV-tiedostosta, joka jakautuu kolmeen osa-alueeseen, jotka näkyvät kuviossa omissa kolumneissaan (kuvio 16.)



	A	B	C
1	osa1	osa2	osa3
2	Filosofian klassinen määritelmä?	Tiedon suhde kontekstiin?	Mitä on metafysiikka?
3	Filosofian suhde tieteeseen?	Luotettavuus?	Mitä tarkoittaa materialismi?
4	Filosofinen ajattelu?	Dogmatismi?	Mitä tarkoittaa idealismi ja sen muodoista objektiivinen ja subjektiivinen idealismi?
5	Sammakko kaivossa perspektiivi?	Relativismi?	Mitä tarkoittavat dualismia ja Platonin ideaoppi?
6	Matkan teon ja perille pääsyn suhde?	Absoluuttinen?	Millainen on Popperin kolme maailman malli?
7	Teoreettinen filosofia?	Postmoderni?	Miten vapaaseen tahtoon liittyvät determinismi ja indeterminismi?
8	Käytännöllinen filosofia?	Skeptisismi?	Millaisia ratkaisuja yleiskäsitteiden ongelmaan ovat käsiterealismi ja nominalismi?
9	Metafysiikka?	Paradoksi?	Viekö metafysiikka ja laajemminkin filosofia harhasta todellisuuteen?
10	Tietoteoria?	Rajatieto?	Valitsetko lukion jälkeisen jatko-opiskelupaikan vapaasti?
11	Logiikka?	Kriittinen realismi?	Kuka oli Victor Sucksdorff ja keitä olivat Ruusu-Ristolaiset?
12	Etiikka?	Tiede?	

KUVIO 17. Ensimmäisen kurssialueen kysymyksiä csv tiedostona.

## 6 TEKNINEN TOTEUTUS

### 6.1 Pelin tekninen kehitys

Projekti tehtiin Unity-pelimootorilla luomalla aluksi 2D-projekti landscape-moodissa. Tällöin Unity luo automaattisesti aloitusnäkyvän eli skenen vaakasuuntaan. Seuraavaksi tehtiin UI-elementtinä kanvas, jolle pelin aloitusruutu rakennettiin. Pelin nimi tekstielementtinä on TextMeshPro ja kirjaisimeksi valikoitui RoadRage-Regular. TextMeshPro on normaalia tekstielementtiä edistyneempi elementti, joka mahdollistaa useiden tehosteiden käytön sekä laajemmat muokaus- ja asettelu-mahdollisuudet. Keskellä oleva Mauno Rosendahlin patsaskuva toimii UI-elementissä eräänlaisena valikkonappina, jota painamalla aukeaa kehäksi Mauno-napin ympärille filosofiankurssien ikonit nappi-elementteinä. Tämä toiminnallisuus luotiin C#-scriptillä Visual Studiossa. Maunon patsaskuva on otettu kameralla ja sitä on muokattu Photoshopilla korjaamalla väriarvoja ja lisäämällä varjostus- ja ääriiviivaefektit. Kurssi-ikoneita on muokattu Photoshopilla lisäämällä varjostusefektia ja kurssien tunnisteet kirjoitustyökalulla. Ikoneihin on myös lisätty animaatiota LeanTween-liitännäisellä ja skenen vaihto toiminnallisuus C#-skriptillä (kuvio 18).



KUVIO 18. Kuva aloitus skenestä, jossa Maunon kuvaa painettu ja kurssivaihtoehdot auki.

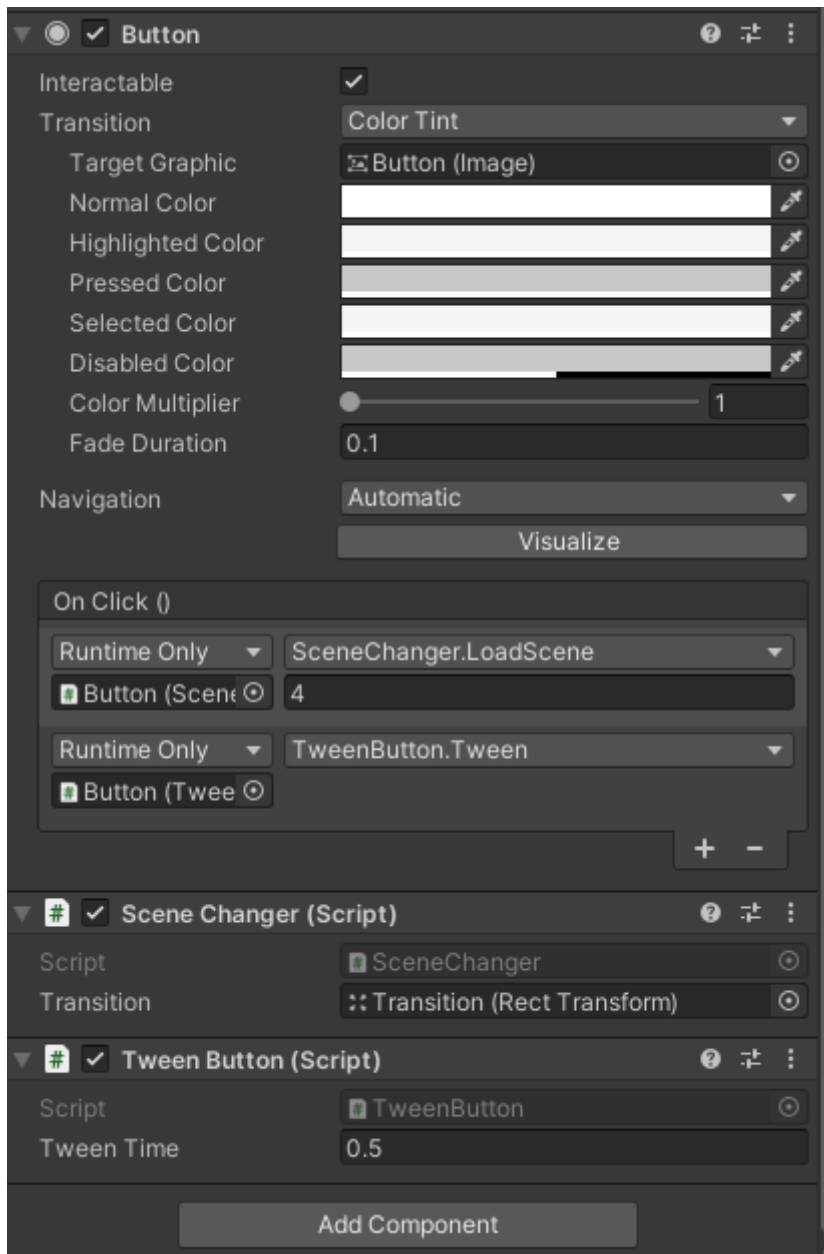
Ikonikehästä painamalla kurssin ikonia päästään filosofiankurssin kysymyksiin, ja kurssin aihepiirin kysymysskenejen välillä liikutaan pelialueen oikealla puolella olevien nappien kautta painamalla. Liikuttaessa eri kurssiskenejen välillä pyrittiin skenenvaihtoa tekemään pehmeämmäksi luomalla pieni skenen häivytysvaihe siirtymän väliin. Tämä toteutettiin luomalla SceneChanger-skripti ja hyödyntämällä siinä LeanTween liitännäistä Unityn omien animaatiomahdollisuuksien sijaan. SceneChanger komponenttiin, ominaisuusikkunaan, laitettiin skeneen luotu kanvaksen kokoinen kuva-objekti nimeltään transition, joka määriteltiin mustaksi. LeanTween liitännäisellä pysytään sitten koodissa määrittelemään kuvan värisyvyysarvoja ja niiden välistä siirtymäaika.

Alla kuviossa (kuvio 19) olevalla ohjelmakoodilla toteutettiin myös näkymän vaihto yhdistämällä kyseinen ohjelmakoodi kurssien painikkeiden OnClick()-metodiin Unityn ominaisuusikkunassa ja määrittelemällä haluttu näkymä rakennusasetuksissa olevalla näkymän järjestysnumerolla.

```
6
7 public class SceneChanger : MonoBehaviour
8 {
9     [SerializeField] RectTransform Transition;
10
11     public void Start()
12     {
13         Transition.gameObject.SetActive(true);
14
15         LeanTween.alpha(Transition, 1, 0);
16         LeanTween.alpha(Transition, 0, 0.5f).setOnComplete(() =>
17         {
18             Transition.gameObject.SetActive(false);
19         });
20     }
21
22     public void LoadScene(int i)
23     {
24
25         Transition.gameObject.SetActive(true);
26         LeanTween.alpha(Transition, 0, 0);
27         LeanTween.alpha(Transition, 1, 0.8f).setOnComplete(() =>
28         {
29             SceneManager.LoadScene(i);
30         });
31     }
32 }
33
34
35
```

KUVIO 19. Kuvakaappaus SceneChanger-skriptistä, jolla luotiin skenen vaihto ja transition efekti.

Alla olevassa kuviossa (kuvio 20.) näkyvät ominaisuusikkunassa myös OnClick()-metodin lisäksi napilla olevat SceneChanger- ja TweenButtonkomponentit, joista ensimmäinen toteuttaa skene-siirtymän häivytysefektiä ja toinen napeilla olevaa animaatiota, jolle pystyy ominaisuusikkunassa määrittämään ajan, Tween Timen.



KUVIO 20. Kuva Unityn ominaisuus -ikkunasta. Kuvassa nappi-elementti.

TweenButton-skriptissä (kuvio 24.) on käytetty IPointerEnter- ja IPointerExitHandleriä, jotka tunnistavat hiiren liikkeen GameObjectin päälle tai pois, ja kummallakin tapahtumalla on tehty

LeanTween-pluginiä hyväksikäyttäen napin animaatio. Hiiren liikkeessa napin päälle se suurenee ja hiiren liikkeessa napin päältä pois se palautuu normaalin kokoiseksi.

```
TweenButton.cs
Miscellaneous Files
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.EventSystems;

public class TweenButton : MonoBehaviour, IPointerEnterHandler, IPointerExitHandler
{
    public float tweenTime;

    public void Tween()
    {
        LeanTween.cancel(gameObject);
        transform.localScale = Vector3.one;
        LeanTween.scale(gameObject, Vector3.one * 1.5f, tweenTime).setEaseOutQuad();
    }

    public void OnPointerEnter(PointerEventData eventData)
    {
        transform.localScale = Vector3.one;
        LeanTween.scale(gameObject, Vector3.one * 1.5f, tweenTime).setEaseOutQuad();
    }

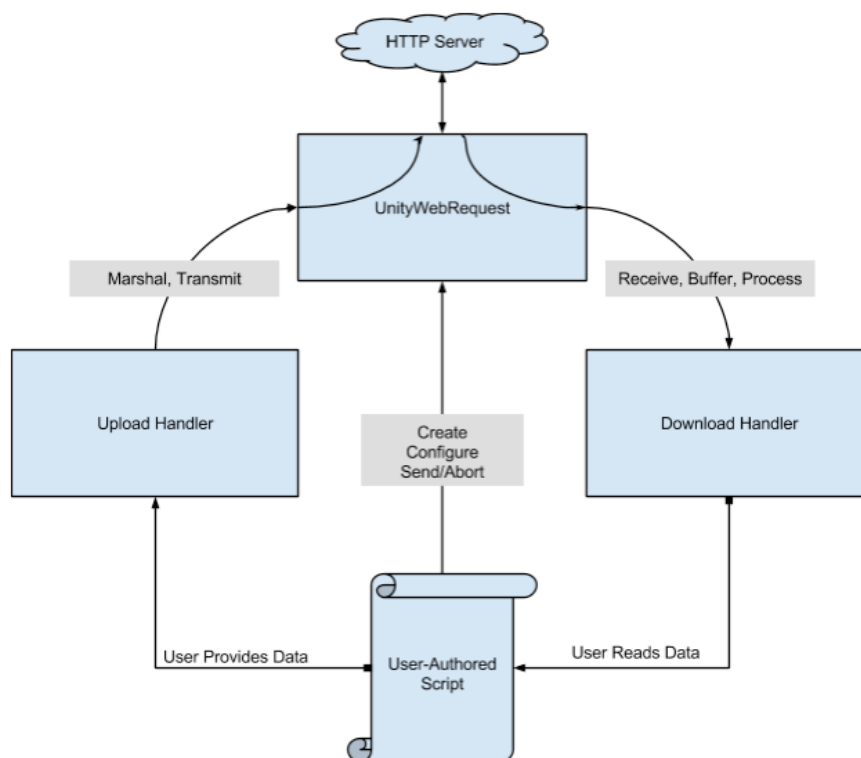
    public void OnPointerExit(PointerEventData eventData)
    {
        transform.localScale = Vector3.one;
        LeanTween.scale(gameObject, Vector3.one, tweenTime).setEaseOutQuad();
    }
}
```

KUVIO 21. TweenButton-skripti, jolla saadaan nappeihin pientä animaatiota.

Samaa yleisilmettä ja otsikkofonttia on pyritty käyttämään jokaisen kurssin skeneissä taustakuvia, korttien väritystä ja teemaa lukuun ottamatta. Kortti on luotu Canvas-elementille Create empty-toiminnolla pelobjektiksi, jonka alla on lapsielementteinä kortin etu- ja takakuva sekä kysymysteksti TextMeshPro-elementtinä. Kortin takakuva on luotu Adobe Illustratorin työkaluilla, johon on lisätty filosofien vektorigrafiikkana olevat kuvat. Kortit ovat Unityn Grid Layout Group-elementin sisällä, jolla voidaan määrätä korttien asemointi canvakselle. Grid Layout Group-komponentti koostuu soluista. Solujen kokoa, määrää ja asemointia säädetään ominaisuusikkunasta elementin rajatulla alueella. Kysymyskortit tehtiin aluksi ScriptableObjecteiksi resurssikansioon ja laitettiin valmiiksi canvakselle. ScriptableObject on eräänlainen data-säiliö, johon voidaan helposti säilöä isoja määriä tietoa asettina, ja tarkoituksena on pienentää projektin muistin käyttöä vähentämällä tarpeetonta tiedon toistoa. Korteissa on etu- ja takakuva sekä TextMeshPro-elementtinä kysymystekstit, jotka kirjoitettiin valmiiksi Unityssä opettajalta saatujen kysymysten mukaan. Tätä

lähdettiin muokkaamaan siten, että opettaja pystyisi muokkaamaan ja lisäämään kysymystekstejä tarpeen mukaan. Kortit luodaan kysymystekstien määrän mukaan kanvakselle pelin alkaessa. Korttien kysymysten tallennusmuodoksi valikoitui CSV-tiedostomuoto. Kunkin kurssin kysymykset ovat omina CSV-tiedostoina, joissa kurssin aihealueet ovat omilla kolumneissaan. Näistä koodilla haetaan kortteihin oikeat kysymykset omilta riveiltään. Kysymyskortit tuodaan kanvakselle satunnaisessa järjestyksessä korttipakkana, josta painamalla päällimmäistä korttia aukeaa kysymys ja tämän jälkeen kortti poistetaan pakasta. Pakan saa sekoitettua shuffle-napista, jolloin poistuneet kysymyskortit ovat pelissä uudelleen mukana.

Lopulta peliä muokattiin niin, että opettajan tekemät CSV-tiedostot ovat palvelimella valmiina, mistä ne haetaan suoraan UnityWebRequest-luokkaa käyttäen ja luodaan kysymyskortit kysymyksistä kanvakselle satunnaisjärjestyksessä. Kysymykset tallentuvat pelaajan omalle laitteelle paikallisesti. Opettajan on helppo muokata CSV-tiedostoa palvelimelta, ja peliä ladattaessa uudestaan ovat kysymykset päivittyneet peliin. Alla kaavio UnityWebRequestin toiminnasta (kuvio 22.) (Unity 2022.)

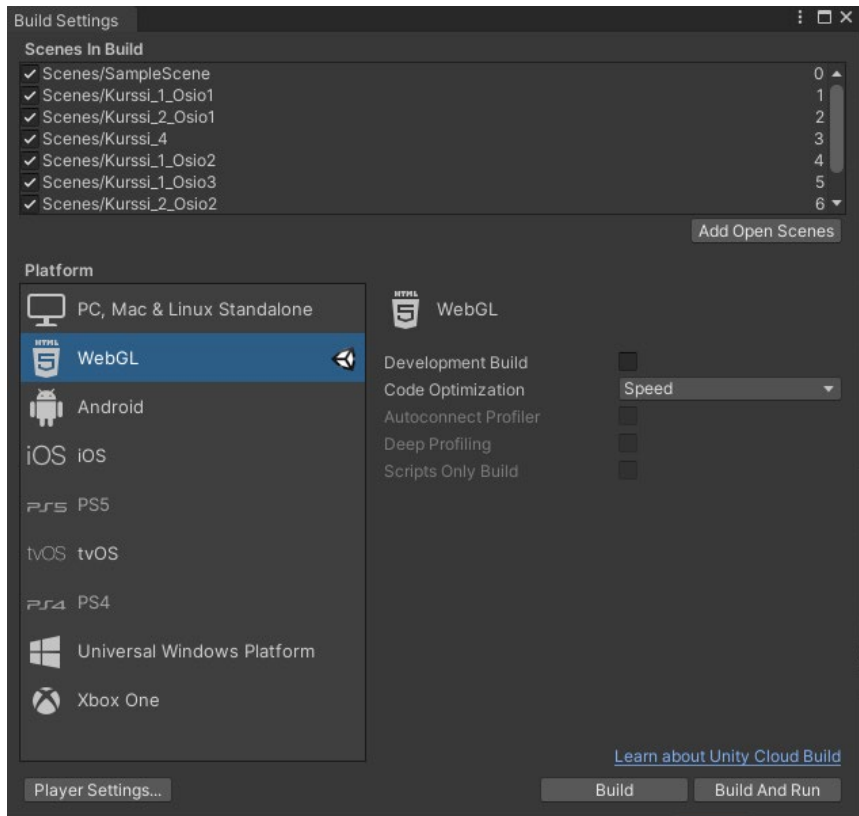


KUVIO 22. UnityWebRequest.

UnityWebRequestin tarkoituksena on mahdollistaa kommunikaatio pelin ja verkkopalvelimien kanssa HTTP-protokollaa, RESTful APIa ja pääasiallisesti metodeja GET, POST ja PUT käyttäen. UploadHandler-objekti hoitaa tiedon siirron palvelimelle ja DownloadHandler-objekti palvelimelta tiedon latauksen ja prosessoinnin. UnityWebRequest toimii useimmilla alustoilla, kuten WebGL, Universal Windows Platform ja iOS- sekä Android mobiilialustoilla. (Unity 2022.)

## 6.2 WebGL Build

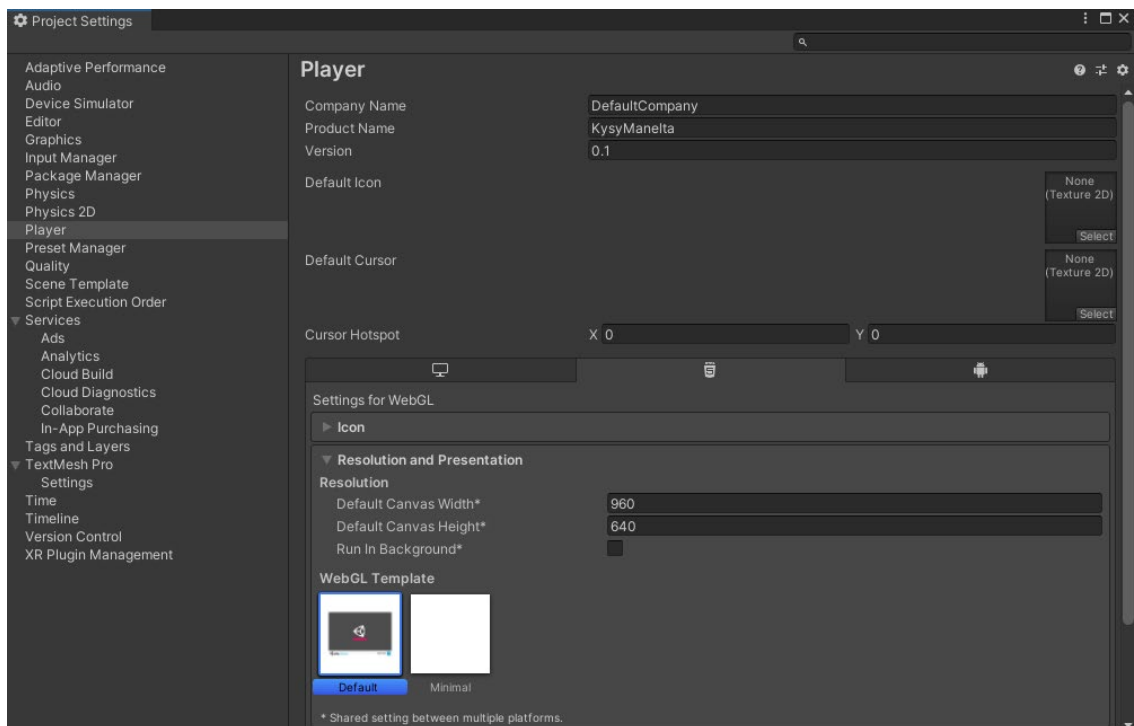
Pelin WebGL-rakennus on tehty Unityssä varsin yksinkertaiseksi. Ensin valitaan File-valikosta Build Settings, jossa varmistetaan, että kaikki halutut näkymät ovat mukana. Seuraavaksi valitaan WebGL-vaihtoehto (kuvio 23.).



KUVIO 23. Build Settings -ikkuna

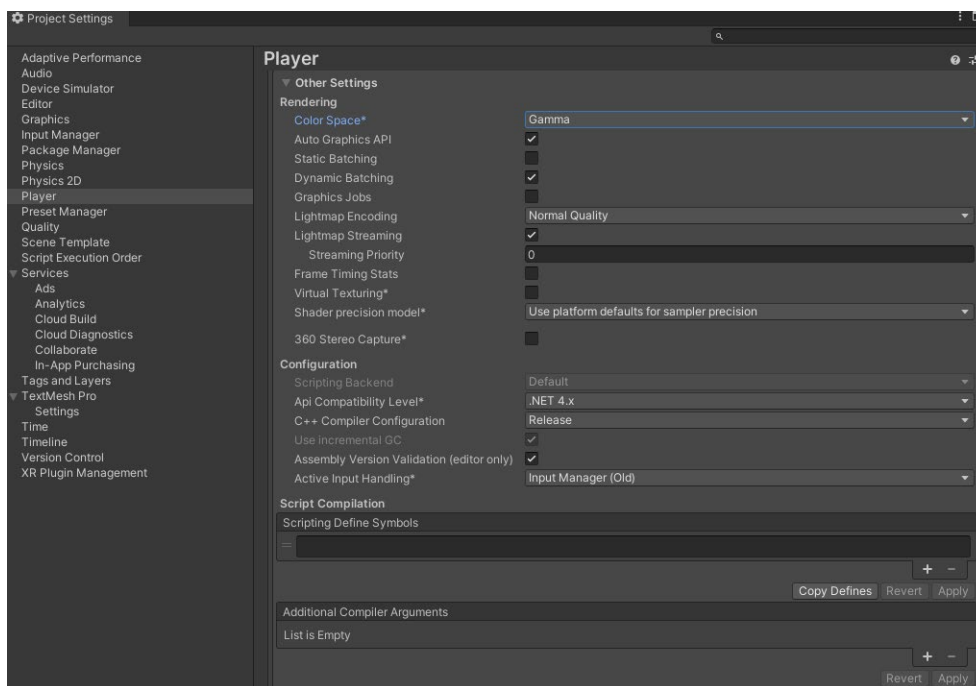
Tämän jälkeen voidaan haluttaessa muuttaa pelaaja-asetuksia valitsemalla Player Settings, jossa voidaan määrittää tekijän ja tuotteen nimi sekä rakennuksen versio. Resolution and Presentation-asetuksista voidaan määrittää pelin ikkunan aloituskoko (kuvio 24.).





KUVIO 24. WebGL Player Settings.

Valikosta löytyy toki muitakin asetuksia, mutta pelin rakentamista voidaan kokeilla yksinkertaisimmillaan jo oletusasetuksilla, jotka tuntuvat olevan Unityssä aika hyvin kohdillaan (kuvio 28).



KUVIO 25. Lisäasetuksia



Tämän jälkeen voidaankin kokeilla, onnistuuko pelin rakentaminen painamalla Build-nappia. Jolloin Unity kysyy, mihin rakennettu peli ja sen tiedostot tallennetaan. Olisikin suositeltavaa luoda ihan oma kansionsa rakennettavalle pelille, jotta kaikki tiedostot päätyvät varmasti oikeaan paikkaan.

### **6.3 Ongelmat teknisessä kehityksessä**

Ensimmäisen testiversion vieminen palvelimelle onnistui, mutta Chrome-selaimella aloitusnäytön toiminnallisuus ei ollut aivan halutunlainen. Klikattaessa Mauno-nappia levittäytyivät kurssien ikonit yli pelialueen. Firefox- ja Edge-selaimissa vastaavaa ongelmaa ei esiintynyt.

Ongelmaan ei auttanut koodin muuttaminen siten, että pienennettäisiin ikoneiden etäisyyttä Mauno-napista. Chrome-selain ei reagoinut koodin muuttamiseen millään lailla. Lokaalisti testatuna node.js serverillä ongelmaa ei ilmennyt Chrome-selaimellakaan, mutta Xampin kautta ongelma oli läsnä.

## 7 POHDINTA

Molemmilla oli alussa muisteltavaa ja opiskeltavaa Unity-pelimoottorin käytöstä, ja vaikka se oli aikaa vievää, niin pääsimme molemmat suhteellisen nopeasti enemmän tai vähemmän jyvälle sen käytöstä, sekä pääsimme itse pelin suunnittelun ja tekemisen pariin. Koska opinnäytetyö jatkuu jatkokehityksenä syksyllä 2021 alkaneesta toimeksiannosta, olimme jo opinnäytetyötä aloittaessa kerenneet tutustumaan aiheisiin, jota tässä käsitelimme, sekä opettelemaan opinnäytetyössä mainittavista työkaluista.

Kahdestaan työskentely opinnäytetyön parissa onnistui, koska meillä oli hyvät työkalut yhdessä työskentelemiseen, muun muassa Microsoft Teams ja GitHub. Pidimme yhteyttä pääasiallisesti puhelimitse viestein, mutta myös Microsoft Teams yhteistyöalustan kautta soitellen. Sen lisäksi että meillä oli palavereita parin viikon välein toimeksiantajamme ja ohjaajiemme kanssa, pidimme erikseen näitä meidän omia palavereitamme noin viikon tai kahden välein, riippuen siitä, miten olimme edenneet ja mitä olimme suunnitelleet tekevämme seuraavaan palaveriin mennessä.

## LÄHTEET

Brackeys 2017. How to use GitHub with Unity. Hakupäivä 30.11.2021. [How to use GitHub with Unity](#)

Business Finland 2016. Ajankohtaista Varsinais-Suomesta: Suomalaiset hyötypelit tulevat. Hakupäivä 7..2.2022. <https://www.epressi.com/tiedotteet/media/ajankohtaista-varsinais-suomesta-suomalaiset-hyotypelit-tulevat.html>

Dealessandri, Marie 2020. What is the best game engine: is Unity right for you? Hakupäivä 10.1.2022. <https://www.gamesindustry.biz/articles/2020-01-16-what-is-the-best-game-engine-is-unity-the-right-game-engine-for-you>

Kean, Kadeisha 2022. The best 5 game engines for beginners in video game development. Hakupäivä 8.5.2022. <https://www.makeuseof.com/best-free-game-engine-for-beginners/>

King, Nico 2021. Serious Games Guide: Everything you need to know in 2021. Hakupäivä 14.12.2021. <https://www.chaostheorygames.com/blog/serious-games-guide-everything-you-need-to-know-in-2021>

Minecraft education edition n.d. Hakupäivä 10.1.2022. <https://education.minecraft.net/fi-fi/homepage>

Natvidad, Sid 2022. 10 Best games made with CryEngine. Hakupäivä 22.5.2022. <https://game-rant.com/best-games-made-with-cryengine/>

Nordicedu 2022. Hyödyllisiä pelikokemuksia ja innostavaa oppimista. Hakupäivä 7.2.2022. <https://nordicedu.com/>

Nordicedu 2022. Minipelikokoelma tupakoimattomuuden kannattavuudesta. Hakupäivä 7.2.2022. <https://nordicedu.com/portfolio/fume>

Nordicedu 2022. MobiMetsä – Mobiilipeli partiolaisille. Hakupäivä 7.2.2022.

<https://nordicedu.com/portfolio/mobimetsa>

Oriz, Eduardo & Zachariah, Shanti & Stock, Kristin 2021. How to speed up 2D art workflows with 2D PSD Importer. Hakupäivä 22.5.2022. <https://blog.unity.com/technology/how-to-speed-up-2d-art-workflows-with-2d-psd-importer>

Placzek, Mark 2021. Improving Accessibility in Unity Games – Part 1. Hakupäivä 30.11.2021. <https://www.raywenderlich.com/5783444-improving-accessibility-in-unity-games-part-1>

Preforce 2020. What are the most popular game engines? Hakupäivä 8.5.2022. <https://www.preforce.com/blog/vcs/most-popular-game-engines>

Santti, Heikki & Virtanen, Mari 2018. Hyötypelit opetuksen uudistajina? Hakupäivä 7.2.2022. <https://blogit.metropolia.fi/hiilta-ja-timanttia/2018/11/19/hyotypelit-opetuksen-uudistajina/>

Schardon, Lindsay 2022. Best game engines for 2022 – Which should you use? Hakupäivä 8.5.2022. <https://gamedevacademy.org/best-game-engines/>

Sibony, Joseph 2021. Top 7 gaming engines you should consider for 2022. Hakupäivä 8.5.2022. <https://www.incredibuild.com/blog/top-7-gaming-engines-you-should-consider-for-2020>

Stat n.d. Suomi maailman kärjessä. Hakupäivä 7.2.2022. <https://www.stat.fi/tup/satavuotias-suomi/suomi-maailman-karjessa.html>

Suominen, Taisto & Luimula, Mika 2019. Vuoden 2018 hyötypeli on Turku Game Labin ja Humakin Lights On! Hakupäivä 30.11.2021. <https://www.turkuamk.fi/fi/ajankohtaista/2137/vuoden-2018-hyotypeli-turku-game-labin-ja-humakin-lights/>

Unity 2018. Prefabs. Hakupäivä 30.11.2021. <https://docs.unity3d.com/2020.3/Documentation/Manual/Prefabs.html>

Unity 2021. Unity Learn. Hakupäivä 30.11.2021. <https://learn.unity.com/>

Unity 2021. ScriptableObject. Hakupäivä 30.11.2021. <https://docs.unity3d.com/2020.3/Documentation/Manual/class-ScriptableObject.html>

Unity 2021. Scripting. Hakupäivä 30.11.2021. <https://docs.unity3d.com/2020.3/Documentation/Manual/ScriptingSection.html>

Unity 2021. Unity UI: Unity User Interface. Hakupäivä 30.11.2021. <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/index.html>

Unity 2022. Plans and pricing, start creating with Unity. Hakupäivä 10.1.2022. <https://store.unity.com/>

Unity 2022. PSD Importer Overview. Hakupäivä 23.5.2022. <https://docs.unity3d.com/Packages/com.unity.2d.psdimporter@7.0/manual/index.html>

Unity 2022. UnityWebRequest. Hakupäivä 8.5.2022. <https://docs.huihoo.com/unity/5.4/Documentation/en/Manual/UnityWebRequest.html>

Wikipedia 2023. Unity (game engine) Hakupäivä 30.11.2021. [https://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))

Wikipedia 2022. CSV. Hakupäivä 30.5.2022. <https://fi.wikipedia.org/wiki/CSV>

Wikipedia 2022. Sokrates. Hakupäivä 27.5.2022. <https://fi.wikipedia.org/wiki/Sokrates>

Wikipedia 2023. Platon. Hakupäivä 27.5.2022. <https://fi.wikipedia.org/wiki/Platon>

Wirtz, Bryan 2023. Unreal vs CryEngine: Which engine should you use? Pros, Cons, and the final verdict. Hakupäivä 8.5.2022. <https://www.gamedesigning.org/engines/cry-vs-unreal/>