

Niko Kotavuopio

PELIKOLLAASI

PELIKOLLAASI

Niko Kotavuopio
Opinnäytetyö
Kevät 2015
Tietojenkäsittelyn koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma, web-sovelluskehityksen sv

Tekijä: Niko Kotavuopio

Opinnäytetyön nimi: Pelikollaasi

Työn ohjaaja: Matti Viitala

Työn valmistumislukukausi ja -vuosi: Kevät 2015

Sivumäärä: 36

Tämän opinnäytetyön perimmäinen tarkoitus oli luoda pelikokoelman kaltainen projekti. Opinnäytetyöllä tehostetaan omaa tietoutta sekä osaamista pelikehityksestä. Projektiin kuuluu intro-ruutu, valikko sekä pelit. Peleillä havainnollistetaan omaa oppimista ja pelikehittämisen perusteita Unityä käyttämällä. Opinnäytetyö lähti liikkeelle omasta mielenkiinnosta pelikehitykseen ja haluan oppia siitä enemmän.

Opinnäytetyö toteutettiin käyttäen Unity-pelimoottoria, sillä Unityn käyttötaitojen kasvattaminen kuuluu projektin tavoitteisiin. Tavoitteena on myös tuoda ulospäin omaa osaamista projektiin toteutetuilla demopeleillä. Peleissä käytetään Unityn ominaisuuksia sekä pelit koostuvat siten, että Unityn perusominaisuudet käsitellään eri tavoilla.

Opinnäytetyöllä saavutettiin laajempaa tietoutta sekä taitoa pelikollaasin luonnin lisäksi tärkeistä Unityn toiminnoista, joita käytetään pelikehityksessä. Kehitysehdotuksiin kuuluu pelien kontrollien hienosäätö. Muita jatkotoimenpiteitä ovat tarinan luonti pelien toteutuksen ympärille. Lisäksi projektin ollessa näin laaja, voidaan jatkokehittämisellä luoda uusia pelejä. Uusilla peleillä näytettäisiin enemmän puolia Unitystä. Projekti on hyödynnettävissä, kun näytetään omia teoksia ja kokonaiskuvaa muille henkilöille.

Asiasanat: pelinkehitys, pelit, ohjelmointi, Unity

ABSTRACT

Oulu University of Applied Sciences
Business Information Systems, Option of web application development

Author: Niko Kotavuopio

Title of thesis: Pelikollaasi

Supervisor: Matti Viitala

Term and year when the thesis was submitted: Spring 2015 Number of pages: 36

The primary purpose of this thesis was to create project similar to a game collection. The thesis enhanced my knowledge and skills in game development. Project is constructed from three different areas which are screens of intro, game selection and games. These games demonstrate the basics of game development with Unity. The thesis was started out of my own interest towards the game development and will to learn more of it.

The thesis was created by using Unity game engine. Increasing my skills of using Unity belongs into goals of the project. Bringing out my own expertise with the created games is also a part of the goals for this thesis. Games are created with features of unity and are built so that the basic features are handled from different perspectives.

The greater knowledge and skills were achieved with doing this thesis. Important features what are used in game development have strengthened in the process as well. Ideas for improvement includes fine tuning of the controls and creating the story for created games. Making new games as well and putting them into game selection would also be a great target to improve the project even further. New games would show more variety of the Unity features. Benefits from the thesis can be used when the full picture is shown to others.

Keywords: game development, games, programming, Unity

SISÄLLYS

1	JOHDANTO	6
2	TYÖKALUT.....	7
2.1	Unity	7
2.1.1	Unityn käyttöliittymä	7
2.1.2	Unityn ominaisuuksia	8
2.2	MonoDevelop	11
2.3	Adobe Photoshop CS5	11
3	ASSETIT.....	13
3.1	Unity Asset Store.....	13
3.2	BlendSwap	13
3.3	CGTextures	14
3.4	OpenGameArt	14
4	PROJEKTIN ETENEMINEN	16
4.1	Aloitus	16
4.2	Intro	18
4.2.1	Intron suunnittelu	19
4.2.2	Intron toteuttaminen	20
4.3	Pelivalikko	21
4.3.1	Käyttöliittymä ja toiminnallisuus	22
4.3.2	Toteutus	24
4.4	Peli 1 – Village Trouble.....	26
4.4.1	Toteutus.....	26
4.4.2	Tavoite	28
4.5	Peli 2 – Butterfly	28
4.5.1	Toteutus.....	29
4.5.2	Tavoite	30
4.6	Peli 3 – Cubescroller	30
4.6.1	Toteutus.....	31
4.6.2	Tavoite	32
5	YHTEENVETO	34
	LÄHTEET.....	35

1 JOHDANTO

Pelaamiseen ja peleihin liittyen into heräsi jo vuosia sitten. Opiskeluaikanakaan pelit eivät missään välissä päässeet unohtumaan ja hiljalleen kasvoi tunne, että haluan olla pelinkehityksessä mukana. Olin myös huomannut, että uudet pelit eivät herättäneet enää samoja tunteita yhtä vahvasti. Opinnäytetyön aloituksen tultua ajankohtaiseksi olin saanut ideaa opinnäytetyöprojektille Oulu Game Lab -harjoittelujaksolta. Halusin oppia vielä enemmän ohjelmoinnista ja retropelien kehityksestä sekä tuoda osaamistani myös esille. Nämä asiat mielessäni ajattelin ehdottaa pelikokoelmaa opinnäytetyöprojektikseni ja työn aihe hyväksyttiin.

Projektilla kehitetään omia taitoja ja tuoda ilmi osaamista. Projekti olisi verrattavissa klassisiin plug & play -konsoleihin, joissa oli valmiina useita pelejä. Projekti koostuu intro-ruudusta, pelivalikosta ja peleistä. Tällä kokoonpanolla opinnäytetyötä on helppo rajata ja jatkokehittää. On vaikea arvioida, miten paljon pelisuunnitteluun ja -kehittämiseen menee aikaa. Opinnäytetyön suuruutta voidaan rajata pelien määrällä ja sisällöllä.

Opinnäytetyössä käytetään Unity-pelimootoria ja asennuksen mukana tulevaa MonoDevelop-sovellusta. Opinnäytetyössä ei ole painoarvoa grafiikoilla, välttämättömimmät 2D-grafiikat toteutettiin Adobe Photoshop -kuvankäsittelyohjelmalla. Työ esittelee miten pelikehitysosaamista voidaan esittää pelikokoelman muodossa.

2 TYÖKALUT

Kuvataan projektissa käytetyt työkalut, joita voidaan hyödyntää pelinkehitysprosessissa. Työkalut koostuvat yleisimmistä ja useimmin käytetyistä sovelluksista, joita tämän projektin valmistumiseen on käytetty. Sovellukset pääosin jakautuvat kolmeen eri kategoriaan; pelimoottoriin, ohjelmointiin keskittyvästä kehitysympäristöstä sekä muihin työkaluihin. Muut työkalut koostuvat sovelluksista, joilla voidaan toteuttaa visuaalista ilmettä tai muuten parantaa pelaajan kokemusta.

2.1 Unity

Unity on pelikehitykseen luotu pelimoottoriohjelmisto 2D- ja 3D-pelejä varten. Unitysta julkaistiin 3.maaliskuuta 2015 Unity5-ilmaisversion minkä tukemiin yleisimpiin alustoihin kuuluu Windows, Mac, Linux/Steam OS, Unity Webplayer, Android sekä iOS. Tuettuihin alustoihin kuuluu lisäksi lukuisia muita alustoja, kuten esimerkiksi konsolit sekä älytelevisiot. Unity-pelimoottori on kehitetty C/C++ kielellä, mutta logiikka perustuu avoimen lähdekoodin Mono-alustaan. Tuetut ohjelmointikielet ovat C#, JavaScript ja Boo. Näistä kolmesta ohjelmointikielestä huomattavasti vähiten käytetty on Boo. (Unity Technologies 2015a, viitattu 22.3.2015.)

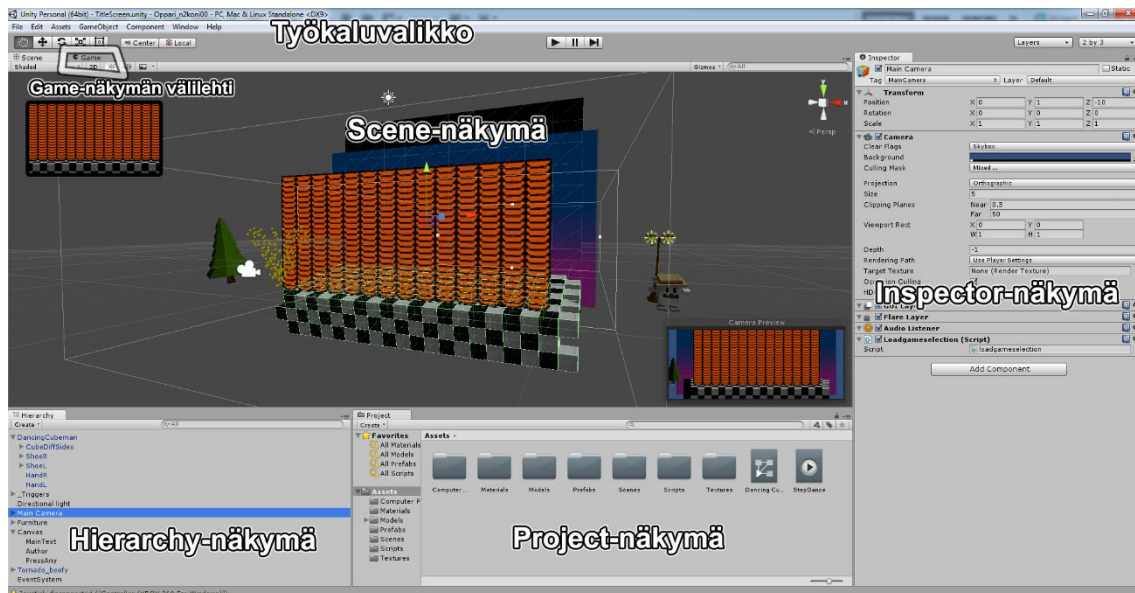
Yleisimpiä pelimoottoreita Unityn lisäksi ovat Unreal Engine ja Source Engine. Kaikilla pelimoottoreilla pystytään tuottamaan hyvää jälkeä, vaikkakin pelimoottorit voivat keskenään poiketa paljon esimerkiksi käyttöliittymän ja toiminnallisuuden suhteen. Pelimoottorit antavat kehittäjille rakentaa pelejä nopeasti sekä tehokkaasti. Pelimoottoreiden käyttöympäristö ja toiminnot helpottavat vaikeiden ja aikaa vievien vaiheiden parissa työskentelyä. (CBS Interactive 2015a, viitattu 02.05.2015.)

2.1.1 Unityn käyttöliittymä

Unity tarjoaa helppokäyttöisen käyttöliittymän pelin luomista varten. (KUVIO 1) Ensikäyttäjän on hyvä huomioida työkalut yläpalkissa. Työkalut mahdollistavat objektien hallinnan kuten sijainti, rotaatio ja koko. Unity3D-ohjelman käyttöliittymä koostuu näkymistä, joista vakiona näkyvät Hierarchy, Scene, Inspector ja Project. Hierarchy-näkymä kertoo scenen sisällön eli käytännössä objektit, joita on käytetty skenessä. Scene-näkymässä voidaan navigoida ja tarkastella miltä oma

projekti näyttää. Inspector-näkymä toimii portaalina objektien ja joidenkin editorin asetusten ominaisuuksiin. Project-näkymä toimii resurssienhallintana ja mahdollistaa kansiorakenteiden luonnin, jotta esimerkiksi 3D-mallit ja skriptit ovat erillään toisistaan. (Unity Technologies 2015b, viitattu 22.3.2015.)

Unity antaa mahdollisuuden käyttöliittymän kustomointiin ja useasti pelinkehittäjillä on räätälöity käyttöliittymä, käytännössä tämä tarkoittaa muiden näkymien käyttöönottoa. Console- ja Game-näkymät ovat elintärkeitä kehitysprosessissa. Console-näkymä viestii mahdollisista virheistä ja tulostaa varoituksen tarvittaessa. Game-näkymä toimii pelin esikatseluikkunana kameras näkökulmasta ja näyttää miten peli toimii laitteelle käännettynä WYSIWYG -periaatteella (what you see is what you get). (Unity Technologies 2015b, viitattu 22.3.2015.)



KUVIO 1. Unityn käyttöliittymä.

2.1.2 Unityn ominaisuuksia

Tiedostoformaattien tuki Unityssa on laaja, sillä se tukee yleisimpiä kuvankäsittelyohjelmien formaatteja ja kaikkea olennaista mitä pelinkehityksessä käytetään. Laaja tuki formateille mahdollistaa huomattavasti laajemman työskentelyn. Työskenneltäessä voidaan käyttää useita eri

ohjelmistoja ja tämä tuo mahdollisuuksia pelinkehittäjille. Mahdollisuuksiin kuuluu esimerkiksi pysyttäytyminen ilmaisissa ohjelmistoissa. (Unity Technologies 2015c, viitattu 23.3.2015.)

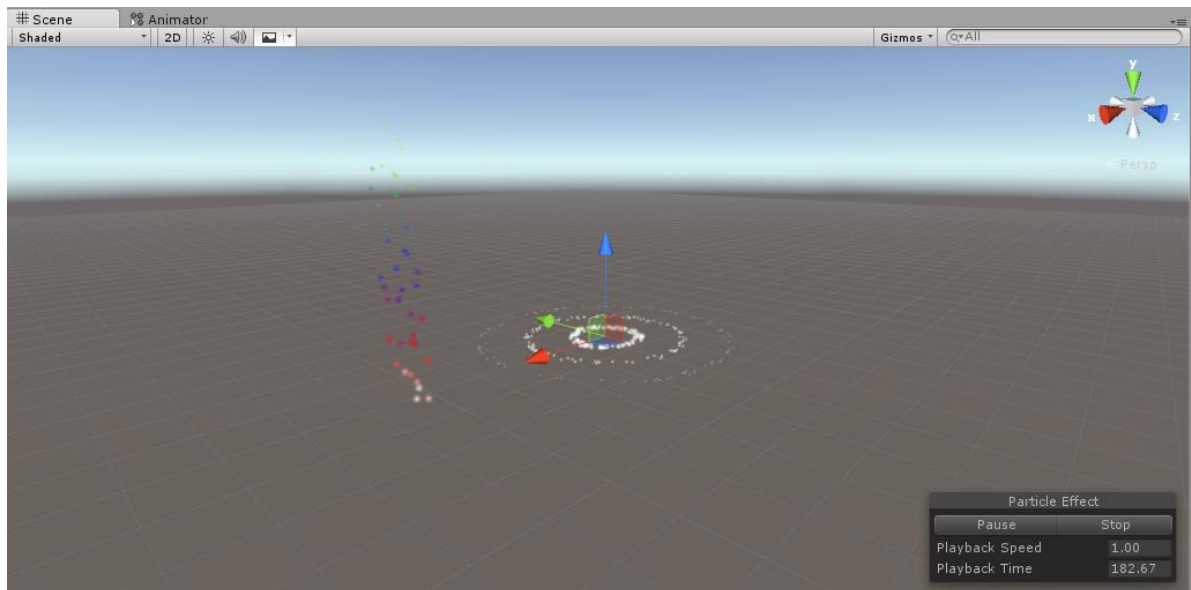
Uusien tiedostojen lisääminen Unityyn tapahtuu vetämällä tiedosto projektinäkömään, mutta tiedostoja voi tuoda myös valikon kautta tai suoraan siirtämällä projektikansioon. Projektinäkömä on hyvä pitää selvänä ja puhtaana, sillä navigointi tiedostojen suhteen helpottuu. Puhdas ja selkeä projektinäkömä nopeuttaa työskentelyä erityisesti ryhmissä. Tällöin kun osa ryhmästä työskentelee oman osa-alueen parissa ja lisää tiedostoja niin löytyvät ne helposti oikean kansiorakenteen avulla. Tiedostoja voi muokata tarvittaessa ja kun projektikansiossa tapahtuu muutoksia Unity pitää itsensä ajan tasalla päivittämällä projektia. (Unity Technologies 2015c, viitattu 23.3.2015.)

Unityssa on tehokas ja rikas animointiin keskittyvä ominaisuus nimeltä Mecanim. Mecanim antaa helppokäyttöisen ja laajan animaatiokokoonpanon kaikille Unityn elementeille. Mecanim tukee muilla sovelluksilla luotuja Unityyn lisättyjä animaatioita ja myös Unityssa toteutetut animaatiot ovat yhteensopivia. Animaation uudelleenohjaus on hyödyllinen osa animoinnissa, sillä jo ennestään olemassaolevaa animaatiota voidaan käyttää uudelleen toiseen objektiin. Animaatioiden esikatselu on mahdollista Unityssa samalla kun animaatiota luodaan. Tämä sallii animaattoreiden työskennellä enemmän itsenäisesti ilman ohjelmoinnin tarvetta. (Unity Technologies 2015d, viitattu 23.3.2015.)

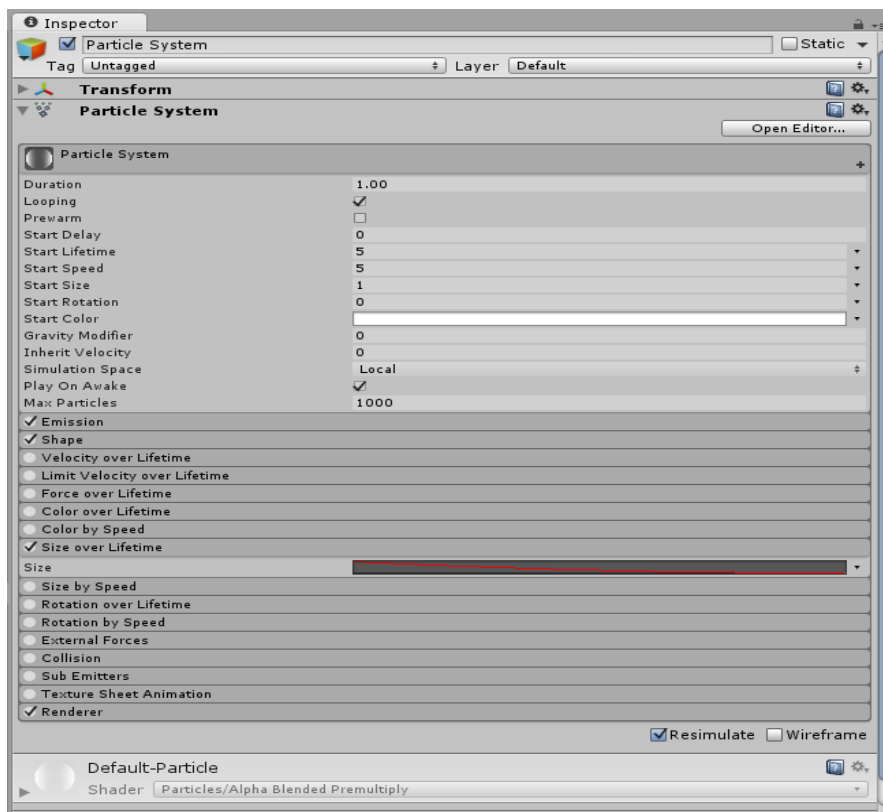
Yksinkertaisimmillaan animaatio voi koostua yhdestä tai kahdesta osasta. Esimerkiksi pelaajalla voi olla erikoisvoima joka koostuu pyörimisestä ja ponnahtelemisesta. Oven avaamiseen ja sulkemiseen riittää yksinkertainen animaatiokontrolleri, mutta ihmishahmon animointi voi vaatia monta kertaa vaativamman animaatiokontrollerin. (Unity Technologies 2015d, viitattu 23.3.2015.)

Efektejä voi toteuttaa käyttäen partikkeleita Unityn Particle System -järjestelmällä. Partikkelit ovat pieniä ja yksinkertaisia kuvia tai polygonien muodostamia kuvioita eli meshejä, joita järjestelmä luo sekä liikuttaa asetusten (KUVIO 3) mukaisesti. Yleensä järjestelmällä luodaan efektejä (KUVIO 2), jotka vaativat monta partikkelia. Räjähdyksen, savuverhojen tai taianomaisten efektien luontiin tarvitaan Particle Systemiä. Particle Systemissä voidaan määrittää partikkelin kuvan lisäksi erilaisia koko- ja väriasetuksia. Partikkelin elinkaari on ennalta määriteltä ja elinkaarta tai partikkelin nopeutta voidaan käyttää hyväksi efektien luonnissa. Esimerkiksi partikkelin väri voi vaihdella elinkaaren- tai nopeuden mukaan. Partikkeleita voidaan myös luoda purkauksissa ja sarjoissa.

Partikkeli voi luoda tietyssä vaiheessa elinkaartaan toisia partikkeleita, niin vaihtoehdot mahdollisille toteutuksille ovat lukemattomat. (Unity Technologies 2015e, viitattu 27.3.2015.)



KUVIO 2. Particle System –järjestelmän efektejä.



KUVIO 3. Particle System –järjestelmän säätömahdollisuuksia

Unitylla voi toteuttaa mobiilisovelluksia, mutta Unityn Game-näkymä ei ole riittävä kun simuloinnissa tulee tarve gyroskoopin tai kosketuskontrollien käytölle. Mobiililaitteille on saatavilla Unity Remote -sovellus, jolla voidaan simuloida kehityksessä olevaa projektia suoraan lennosta. Unity Remote vähentää tarvetta tehdä asennuspaketteja testausta varten. Alustariippumattomissa peleissä tämä on hyvä huomioida jo kehitysvaiheessa. (Unity Technologies 2015f, viitattu 23.3.2015.)

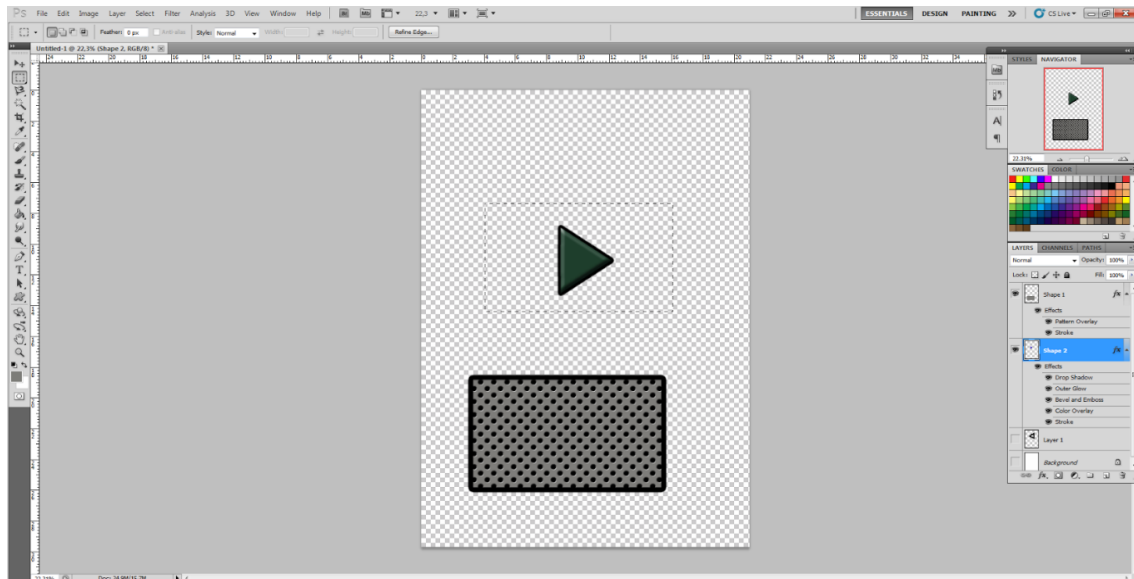
2.2 MonoDevelop

MonoDevelop on avoimen lähdekoodin kehitysympäristö, mikä lisää tekstieditoryöskentelyyn hyödyllisiä lisäominaisuuksia, kuten debuggaus- ja projektinhallintatyökälut. Asennettaessa Unitya voi valita haluaako asentaa mukana tulevan Unitylle kustomoidun MonoDevelop-ohjelmiston. MonoDevelopissa on samankaltaisia ominaisuuksia ja toimintoja, kuten NetBeans- ja Microsoft Visual Studio -ohjelmissä. Tunnetuimpiin toimintoihin kuuluu graafisen käyttöliittymän lisäksi myös koodintäydennys. (MonoDevelop Project 2015, viitattu 24.3.2015.)

MonoDevelop kehitysympäristö auttaa pelinkehittäjää käytännössä luomaan skriptejä, mitkä tuovat projektiin eloa ja toimintoja. Tuki yleisimmille käyttöjärjestelmille löytyy, eli MonoDevelop-sovellusta voidaan käyttää yhteensopivien käyttöjärjestelmien kanssa. Yleisimpiin käyttöjärjestelmiin kuuluu Windows, OS X ja Linux. MonoDevelop tukee lukuisia lisäosia, mikä tekee kehitysympäristöstä enemmän joustavan. (MonoDevelop Project 2015, viitattu 24.3.2015.)

2.3 Adobe Photoshop CS5

Adobe Photoshop (KUVIO 4) on Adobe Systemsin kehittämä kuvankäsittelyohjelma. Adobe Photoshop on yksi käytetyimmistä kuvankäsittelyohjelmista ja on saatavilla Windows- ja Mac OS-käyttöjärjestelmille. Photoshopilla voi muokata ja yhdistellä kuvia ja lopuksi tallentaa ne haluttuun tiedostoformaattiin. (Adobe Systems 2015, viitattu 05.04.2015.)



KUVIO 4. Photoshopin käyttöliittymä.

Photoshopin oma tiedostformaatti on .psd (Photoshop Document), joka säilyttää Photoshopissa tehdyt muutokset ja mahdollistaa helpon ja nopeiden muutoksien teon. Adobe Photoshop -ohjelmaan voi avata useita välilehtiä eri kuville ja työskennellä monen kuvan parissa samaan aikaan. Ohjelmalla pystyy myös luoda kuvia ja animaatioita, vaikkakin Photoshop on enemmän tunnettu kuvankäsittelyohjelmana. (Wikimedia Foundation 2015a, viitattu 04.04.2015.)

3 ASSETIT

Assetit ovat digitaalista dataa ja tiedostoja. Pelikehitykseen liittyen assetit on luotu helpottamaan ja nopeuttamaan työn edistymistä. Assetit voivat koostua Unity editoriin luoduista lisäominaisuuksista ja helppokäyttöisemmistä työkaluista, skripteistä tai 3D-malleista. Assetit voivat olla kokoelmia ja sisältää kaikkia edellämainittuja asioita. Kokoelmat yleensä keskittyvät rakennussarjoiksi, joilla saa valmiin projektipohjan. Unityssa asset-paketteja voidaan tuoda projektiin Asset-valikon alta. Projektitiedostoista voi luoda paketin myöhempää käyttöä varten, jolloin tiedostojen ulosvienti on helppoa. Unityyn yhteensopivia asetteja saa lukuisilta jakelulähteiltä. (Wikimedia Foundation 2015b, viitattu 05.04.2015.)

3.1 Unity Asset Store

Unity Asset Store on Unityn pääjakelukanava asettien hankinnan sekä hallinnan suhteen. Asset Storessa on paljon asetteja julkaisijoilta, joiden nettisivuilta saa dokumentaation asettien käyttöön ja chat-pohjaista apua tarvittaessa. Assetit ovat Unity Asset Storessa lajiteltu selvästi kategorioittain. Complete Projects -kategorian alta löytyy projektipohjia erityyppisiin peleihin. Projektipohjat auttavat Unityn oppimisessa kun voi tarkastella näiden toteutusta. Unity Asset Store säilyttää käyttäjän assetit, mutta asettien saatavuuden loputtua käyttäjälle jää paikallinen kopio asettista. (Unity Technologies 2015g, viitattu 05.04.2015.)

Unity Asset Store valvoo jakelukanavaan julkaistuja asetteja. Laatuvaatimukseen kuuluu hyvin luodut tekstuurit, 3D-mallit, skriptit ja yleisesti hyvä suunnittelu. Aseteilta vaaditaan myös kokoa, sillä yhden kolmiulotteisen objektin lisääminen Asset Storeen ei ole hyväksyttävää, ellei siinä ole jotain erikoista. Asettien luontiin pitää olla käytetty enemmän kuin pari klikkausta, eli hutaisten tehdyt assetit eivät jakelukanavalta löydy. (Unity Technologies 2015h, viitattu 05.04.2015.)

3.2 BlendSwap

Blend Swap on intohimoisten 3D-artistien yhteisö, jotka jakavat töitään Creative Commons -lisenssin mukaisesti. Creative Commons -lisenssijärjestelmässä on neljä päälisenssiä. Lisenssöinnin lähtökohtana on, että teoksesta voidaan luoda kopioita sekä esittää ja näyttää teosta

julkisesti. Lisäksi teosta voidaan levittää ja välittää eteenpäin sekä tehdä muutoksia teokseen tarvittaessa. Näitä perusoikeuksia rajoitetaan ehdoilla, jotka tulevat esille teokselle määritetystä lisenssistä. (Wikimedia Foundation 2015c, viitattu 03.05.2015.)

Sivusto keskittyy Blenderissä (ilmainen 3D-mallinnustyökalu) luotujen 3D-mallien jakamiseen. Sivustolla voi toivoa mallia ja artisteilla on mahdollisuus toteuttaa malleja kysynnän mukaan. Blend Swap tarjoaa mahdollisuuden kasvattamaan mainetta ja parantamaan omia taitoja. Creative Commons -lisenssöinti varmistaa, että artistit eivät jää mainitsematta asiassa mihin heidän töitään käytetään. Blend Swap -yhteisön jäsenet voivat itse valita millaisen lisenssin alle työnsä laittavat. Artisteja voi myös palkata sivuston kautta ja tämä auttaa monia saamaan harrastuspohjaista työtä. (Blend Swap 2015, viitattu 05.04.2015.)

3.3 CGTextures

CGTextures on yksi tunnetuimmista sivustoista tekstuurien hankintaan. 3D-objektit tarvitsevat teksturointia tuomaan esille perspektiiviä ja persoonallisuutta. CGTextures-sivusto tarjoaa tutoriaaleja laajan tekstuuri valikoiman ohella. Tutoriaaleista löytyy ohjeita, miten luodaan saumattomia tekstuureita tai vinkkejä liittyen Photoshop-ohjelman käyttöön. (CGTextures.com 2015, viitattu 05.04.2015.)

Sivuston tekstuureita voi ladata ja käyttää ilmaiseksi, mutta niitä ei saa myydä eikä jakaa uudelleenkäyttöön. Sivustoa voi käyttää ilmaisilla tunnuksilla, mutta käyttäjä voi ostaa jäsenyyden. Jäsenillä on oikeus ladata enemmän tekstuureja päivittäin ja jäsenet voivat ladata korkeammalla resoluutiolla kuin ilmaiskäyttäjät. Ilmaiskäyttäjät voivat ladata 640x640-resoluution kuvia kun jäsenet voivat ladata suurimmalla saatavilla olevalla resoluutiolla. (CGTextures.com 2015, viitattu 05.04.2015.)

3.4 OpenGameArt

OpenGameArt www-sivu tarjoaa korkealuokkaista ilmaisilisenssillä olevaa taidetta pelinkehittäjille. Käyttäjät voivat lisätä sisältöä sivulle ja valita lisenssin työnsä. Sivusto on pitkälle kuten Blend Swap, mutta OpenGameArt-sivusto ei rajoitu 3D-malleihin. (OpenGameArt 2015, viitattu 05.04.2015.)

Sivuston tarjoamat kategoriat pelienkehitykseen ovat 2D ja 3D -materiaali, ääniefektit, tekstuurit, konseptitaide, pelimusiikki, dokumentit sekä tutoriaalit. OpenGameArt-sivustolla on aktiivinen foorumi ja chat-toiminto käyttäjille. Animaatioita sivulta löytyy 2D- ja 3D-objekteille. Sisältöä voi myös selata eri filtteröinnillä, esimerkiksi listaus viikon suosituimmista tai uusimmista materiaaleista. (OpenGameArt 2015, viitattu 05.04.2015.)

4 PROJEKTIN ETENEMINEN

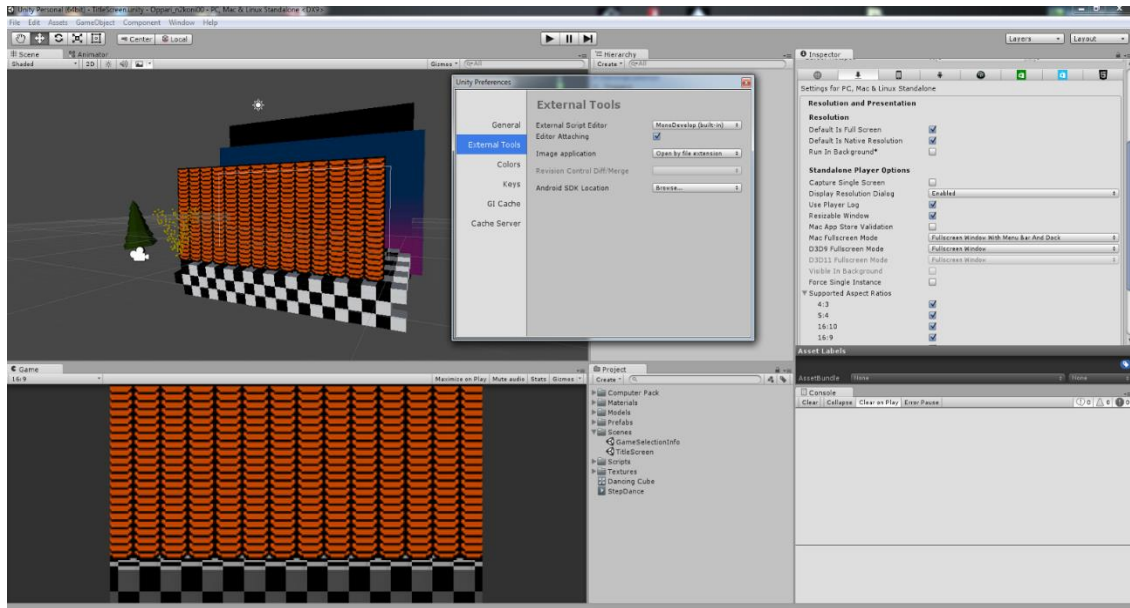
Käsitellään projektin etenemistä aloituksesta lähtien projektin valmistumiseen saakka. Luvussa käydään läpi projektin alkuvaihe, suunnittelutyö sekä sen toteuttaminen. Projekti koostuu kolmesta pääosasta, jotka ovat intro-ruutu, pelivalikko ja luodut pelit. Projektin edetessä huomioidaan myös omaa oppimista pelikehityksen vaiheista.

Valmistuneen projektin käyttö tulee kohdistumaan käyttäjälle, jolla on halua tarkastella työtäni. Käyttäjän määritelmään ei kuulu mikä tahansa pelaaja, vaan projekti toimii näyttönä siitä, mitä Unitylla olen toteuttanut. Käyttäjä toimin itse, mikäli näillä tuotoksilla haluan tuoda esille tiettyä ominaisuutta. Oletettavasti valmistunutta projektia tulee käyttämään myös muut henkilöt.

4.1 Aloitus

Opinnäytetyötä hyväksyttäessä valittiin aihe, mihin on motivaatiota ja halua oppia enemmän. Projekti alkoi selaamalla verkosta retropelejä ja tarkastellen eri pelien toteutuksia sekä tyyliuuntauksia. Videopalvelu YouTubesta löytyvillä opetusvideoilla Unityn yleiskäytöstä ja ohjelmoinnista saatiin hyvä alkukertaus. Saadakseen pelikehityksessä aikaan jotain vaikuttavaa ei tarvitse rahoitusta ja tavoitteeksi on hyvä ottaa oma oppiminen. Tällöin on tärkeää arvostaa omaa työnjälkeä. Ennen opinnäytetyön aloittamista Unityn ominaisuudet tarvitsivat kertausta, ja kertausta tapahtuu projektin aikanakin.

Projektia aloittaessa säädettiin asetukset ja näkymät (KUVIO 5) parhaaksi katsotulla tavalla, jotta työskentely projektin parissa olisi mahdollisimman luontevaa. Näkymiä voidaan vaihdella tarpeen mukaan ja aloituksessa kyseinen näkymä mahdollistaa monen asian samanaikaisen tarkastelun. Uutta projektia luodessa voidaan valita valmiita asetteja avuksi. Opinnäytetyössä käytetään ilmaisia ja ostettuja asetteja, mutta niiden käyttöönotto tapahtuu tarpeen mukaan. Projektin pohja oli valmis työn aloittamiseen, mutta seuraavan vaiheen ja yleisen ilmeen suunnittelulle oli tarvetta. Valmiiseen pohjaan alettiin nopean suunnittelun jälkeen toteuttamaan projektin seuraavaa vaihetta, mikä oli intro-ruutu.



KUVIO 5. Kustomoitu työskentelytila ja Unityn asetuksia.

Projektissa suunnitteluosuus kokonaisuudessaan on mittava. Suunnitteluvaiheissa suunniteltiin tarvittava osuus kerrallaan, eikä koko työtä kerralla loppuun. Suunnitteluosuudet menivät osaluonteittain. Ensimmäinen suunnittelutyötä tarvittiin intron toteuttamiseen, sen jälkeen pelivalikko ja lopulta jokainen peli suunniteltiin nopeilla piirroksilla. Opinnäytetyön rajaaminenkin saatiin ratkaistua, sillä pelikonsepteja voi tuottaa useita tai rajata muutamaaan. Ensimmäinen tavoite oli saada vähintään yksi pelidemon runko toteutettua.

Visuaalinen ilme on tärkeää peleissä ja peliprojekteissa. Projektissa luodaan visuaalinen ilme Unityn valmiita primitiivejä käyttäen. Väritystä ja teksturointia toteuttaessa käytetään verkosta ilmaisilisenssillä löytyvää materiaalia. Osa tekstuureista kumminkin toteutetaan alussa listatuilla kehitysokaluilla. Projektin visuaalista ilmettä voidaan kehittää myöhemmin, tärkeintä on saada toimiva runko valmiiksi.

4.2 Intro

Videopeleissä on yleensä aina aloitusruutu (title screen) ennen kuin pelaaja päästetään syvemmälle pelin lumoihin. Pelaajalle tämä yleensä näytetään julkaisijan ja kehittäjien logojen jälkeen. Aloitusruudussa näytetään pelaajalle pelin nimi ja näkymä toimiikin pienen esittelyn kaltaisena pelaajan perspektiivissä. On mahdollista että, aloitusruutu näyttää pelaajalle jo hieman pelimekaniikoita. Tästä on esimerkkinä Super Mario Bros 3:n aloitusruutu, missä pelaajahahmot liikkuvat itsestään ja hoitelevat jo pelissä ilmeneviä vihollisia.

Intro-ruutu ei havainnollista projektin varsinaista sisältöä vaan enemmän toimii esittelevänä elementtinä käyttäjälle. Introsta saa selville projektin nimen ja huomaa graafista tyyliä, mitä on saavutettu Unityn primitiivien lisäksi ilmaisia 3D-malleja käyttäen. Intro-ruutu toimii portaalina seuraavaan vaiheeseen eli pelin valintaan. (KUVIO 6) Käyttäjä painaa Intro-ruudussa mitä tahansa näppäintä jatkaakseen pelivalikkoon. Intro-ruudussa olevien alkuanimaatioiden aikana ei voi siirtyä pelivalikkoon, vaan on odotettava tekstien ilmentymistä.

Intro-ruudun yleisilme koostuu alustavasti Super Mario Bros 3 -pelin alkuruudun tapaisesti, mutta poikkeavasti sekä rajallisesti. Yleisilme palauttaa mieleen vanhan 8-bittisen NES-konsolin (Nintendo Entertainment System) pelien ilmeen ja luo jo käsityksen siitä, etteivät pelikonseptit tule olemaan liian suuria vaan pieniä näyttekappaleita. Intro-ruutu sekoittaa 2D- ja 3D-ominaisuuksia luoden kuvan tyylistä mikä ei ole samanlaista ja itseään toistavaa. Tyyllillä halutaan myös antaa kuvaa teatterimaisesta ympäristöstä ilman yleisöä. Tällöin käyttäjä tuntee kuuluvansa yleisöön ja kokee että pelit näytetään juuri hänelle. Teatterimainen avaus projektille on osuva, sillä se kuvaa oma-aloitteisuutta sekä tarjoaa käyttäjälle mahdollisuuden syventyä projektiin odottamalla odottamatonta.



KUVIO 6. Intro-ruudun avaus.

4.2.1 Intron suunnittelu

Pääasia mikä piti tuoda esille ennen pelivalikon suunnittelua ja toteutusta, oli projektin nimen lisäksi alustavia tietoja projektista. Yksinään projektin nimeä varten ei introlla ollut suurta merkitystä, mutta sillä haluttiin tuoda esille Unityn ominaisuuksia jo ennen peliosuutta ja toteuttaa visuaalista ilmettä. Introssa haluttiin myös tuoda esille ajastamista, vuorovaikuttamista ja objektien manipulointia. Näitä varten Intro-ruudun (KUVIO 7) toteutus sai enemmän painoarvoa.

Halusin vapaat kädet toteuttaa introa ilman, että valitsemani tyyli rajaisi mahdollisuuksia ja siitä tuli idea teatterin kaltaisesta aloituksesta. Aloituksessa näyttämö avataan punaisilla pikseligrafiikalla toteutetuilla verhoilla. Aloitus muistuttaa Super Mario Bros 3 -pelin aloitusruutua, luoden nostalgista tunnelmaa. Esiintymislavan suunnittelussa ilmeni, ettei yksi asia muistuttamaan Super Mario -pelisarjan nostalgisuudesta ollut tarpeeksi. Väriyksellä korostettiin retromaisia piirteitä ja toteutuksesta tuli samantapainen, mutta kolmiulotteisena. Näyttämöllä oli avainelementit aloitusruudulle ja seuraavana oli vuorossa osuus näyttämön esitelmästä. Kuutioista koostuva tornado kulkee näyttämön ohi, puhaltuen lavasteita paikoilleen. Lavasteina ovat puut, katuvalot sekä steppaava kuutiomies. Näiden tapahtumasarjojen jälkeen näyttämö alkoi täyttyä liikaa ja

tuodaan esille projektin nimi. Nimen esiintulon jälkeen annetaan käyttäjälle mahdollisuus siirtyä pelivalikkoon.

4.2.2 Intron toteuttaminen

Kuvankäsittelyohjelmalla luotiin avauselementtinä toimivat punaiset verhot. Teatterilava rakentui Unityn 3D-ominaisuuksilla eri materiaaleja käyttäen. Lavasteet ovat 3D-objekteja ilmaislisenssillä. Ruudun peittävä punainen verho nousee ajastimen avulla. Ajastuksen hoitaa Unityn Time -luokka ja luokan alta löytyvä staattinen `timeSinceLevelLoad`-muuttuja. Toteutuksessa tarkastetaan onko `timeSinceLevelLoad`:n arvo suurempi kuin `delayTime`. Arvon kasvaessa `delayTime` muuttujaa korkeammaksi aloitetaan verhojen nostotoimenpide toimintojen `transform.position` ja `Vector3.Lerp` avulla.

Kuutioista rakentuva tornado aloittaa rakentumisen heti scenen latauduttua, mutta odottaa varmistusta siitä että verhot ovat ylhäällä. Varmistaminen tapahtuu boolean-muuttujaa käyttämällä ja skriptit ovat vuorovaikutuksessa keskenään. Tornadossa pyörivät kuutiot eivät tuhoudu `Destroy()`-funktiolla vaan kuutioita uudelleenkäytetään tehokkaasti. Käytännössä kuutiosta poistetaan renderöinti ajalta, kun kuutio tekee matkaa lähtöpisteeseen. Tämä luo efektin loputtomasta tornadosta ja säästää objektien luonnista ja tuhoamisesta aiheutuvia kuluja. Tornado toimii myös aktivointina seuraaville tapahtumille, kuten lavasteiden ilmaantumiselle sekä steppaavan kuutiomiehen saapumiselle.

Introssa tutustuttiin myös hieman Mecanimiin, jota käytettiin steppauksen luonnissa. Tanssiaskleet ovat hyvin yksinkertaiset mukaansatempaavan rytmien saattamana. Lavasteita liikuttaa erillinen skripti, missä myös määritellään koordinaateilla lavasteiden uusi sijainti. Liikkeeseen on käytetty `Time.deltaTime` -toimintoa ja tarpeen mukaan kertolaskulla nopeutettu tai hidastettu liikettä. Taustalla erottuva tähtitaivas, ja sinne ilmaantuvat tähdet on luotu käyttäen kuvien vaihtuvalla alfa-arvolla, mikä on kuvissa käytetty läpinäkyvyyskanava. Tähdet ovat taivaalla kokoajan mutta käyttäjälle ne ilmaantuvat tyhjästä vähitellen eri tahdilla.



KUVIO 7. Intro -ruutu.

4.3 Pelivalikko

Yleisimmin useasta vaihtoehdosta koostuva valikko liitetään hahmon valintaan. Tällöin valikon tehtävänä on luoda kuva monipuolisesta tarjonnasta ja tuoda esille hahmon ominaisuuksia. Esiintyviä ominaisuuksia yleensä ovat hahmon kyvyt, attribuutit ja ulkonäkö. Tiivistettäessä voidaan sanoa että valikon ydintehtävä on tuoda esille tietoa tulevasta. Projektin pelivalikko ei liity päätöksentekoon vaikuttaen yhteen peliin vaan valikossa voi selailia eri pelejä. (CBS Interactive 2015b, viitattu 01.04.2015.)

Unityn perusominaisuudet mahdollistavat käyttöliittymän toteuttamisen nopeasti ja selkeästi. Käyttöliittymä luodaan canvas-elementille. Canvasta voidaan siirrellä ja se voidaan renderöidä 3D-maailmaan, jolloin sitä voidaan käsitellä kuin tavallista objektia. Canvas on mahdollista myös renderöidä ruudun tilaan, jolloin canvas kiinnittyy kameraan ollen kokoaikaisesti esillä. Unityn käyttöliittymätyökalut ovat helppokäyttöisiä ja nopeuttavat käyttöliittymän luontia. Käyttöliittymätyökaluihin kuuluu Panel, Button, Text, Image, Scrollbar, Toggle, Input, Canvas ja EventSystem. Jokaiselle työkalulle voidaan määrittää tapahtuma ja ehto milloin tapahtuma käynnistyy. Esimerkiksi uutta painiketta luodessa määritellään mitä tapahtuu kun nappia painetaan.

Tapahtumia voi olla useita eli samaan painikkeeseen voidaan määritellä tapahtuma kursorin ollessa painikkeen päällä. (The Game Contriver 2015, viitattu 01.04.2015.)

4.3.1 Käyttöliittymä ja toiminnallisuus

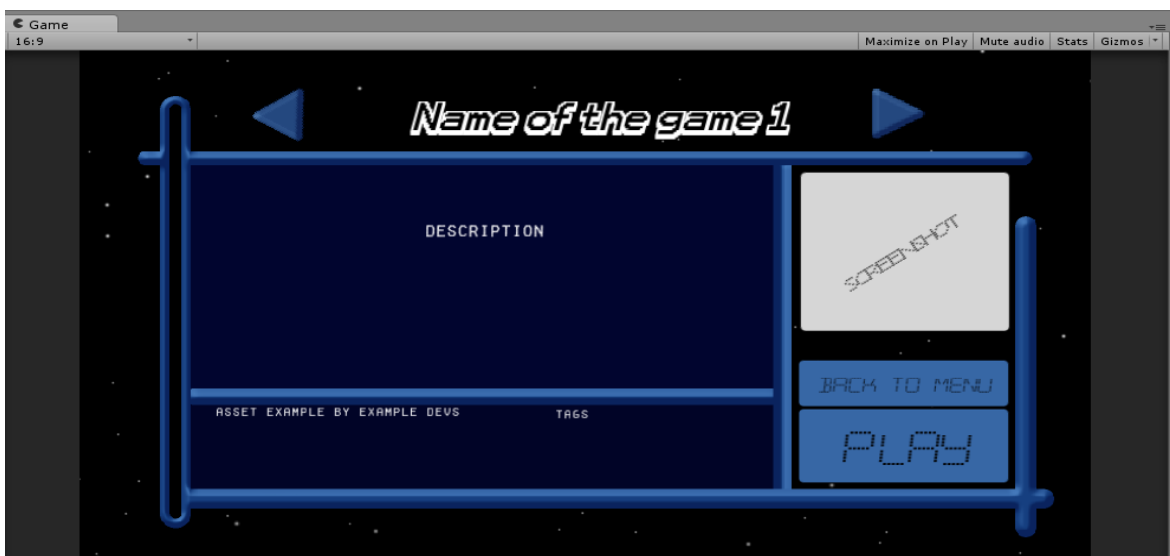
Pelivalikoiden toteutuksessa pidetään mielessä käyttötarkoitus ja tuetaan valikon ydintoimintoa, oli ydintoiminnassa kyse hahmon- tai pelienvainnasta. Valikoissa tärkeä elementti on selkeys ja ettei käyttäjän silmiä laiteta liian ahtaalle. Valikoissa voi tuoda esille monta elementtiä kerralla, useimmiten peleissä onkin valikot liitetty yhteen. Valikoiden integrointi on riippuvainen pelin toteutuksesta ja käyttökokemuksesta. Projektin pelivalikko luotiin selkeäksi ja käyttäjän tarvittaessa lisätietoja voidaan siirtyä info-ruutuun.

Suunnitellessa käyttöliittymää yksi tärkeimmistä ominaisuuksista on valikon symmetrisyys, sillä symmetrisyys ei rasita käyttäjän silmiä. Valikko suunniteltiin muuten symmetriseksi, mutta rajapintaa rikkoen osa valikossa käytetyssä taustasta luotiin tuomaan epäsymmetrisyyttä. Epäsymmetrisyyden avulla kiinnittää käyttäjä enemmän huomiota valikkoon. Epäsymmetrisyys tuo käyttäjälle myös hieman syvyyttä sekä täytettä valikkoon. Näin valikko näyttää yhtäaikaan tilavalta, muttei kuitenkaan tyhjältä. Käyttöliittymään suunnitellaan ensin alueet ja niiden sijainnit sen perusteella mitä tietoa käyttäjälle halutaan tuoda. Sijainteihin sekä alueiden korostukseen vaikuttaa millainen painoarvo alueen viestillä käyttäjälle on. Suunnittelun seuraavassa vaiheessa tuli esille teema sekä tyyli. Tyyliä haettiin selvää toiminnallisuutta retromaisella toteutuksella. Teemaksi soveltui avaruusaiheinen valikko. (KUVIO 8)



KUVIO 8. Pelivalikon käyttöliittymä.

Valikon käyttöliittymä tuo selkeyttä suurilla napeilla ja nappeja korostetaan cursorin ollessa nappien päällä. Valikossa käytetty fontti tuo persoonallisuutta sci-fi genreltä. Käyttäjä saa valikosta ensisilmäyksellä kaiken olennaisen tiedon toimivuuden kannalta. Pelien nimet ovat keskitetty ja laidoille erottuu selvästi Info- ja Play-painikkeet. Info-ruudulla (KUVIO 9) tuodaan esille pelin nimi, kuvakaappaus pelistä ja kuvaus pelin sisällöstä. Käyttäjä voi laukaista pelin käynnistyksen myös Info-ruudulta tai navigoida otsikkoa ympäröivillä nuolilla seuraavaan peliin. (Leap Motion 2014, viitattu 27.03.2015.)



KUVIO 9. Pelivalikon info -ruutu.

4.3.2 Toteutus

Toteutuksessa käytettiin Unityn editoria luomaan pohja käyttöliittymälle. Ohjelmoiden luotiin metodeja eri tapahtumia varten. Metodeja kutsutaan eri tilanteissa, kuten Info-nappia painettaessa kutsutaan gameselect-skriptin ShowInfo()-metodia. Metodi mahdollistaa myös navigoinnin Info-ruudussa painamalla nuoli-painikkeita. Kokonaisuudessaan pelivalikko koostuu kahdesta 3D-maailmaan renderöidystä canvaksesta. Ensimmäinen canvas sisältää pelivalinnan ja toinen canvas sisältää pelin lisätiedot. Siirryttäessä toiseen ruutuun, annetaan kameralle suoraan uudet koordinaatit. Liikkuva tausta luotiin Unityn particle system -elementtiä luomaan tähtimäisiä partikkeleita liikkumaan alhaalta ylös. Partikkelit katoavat lentäessään ulos kameran kuvasta ja samaan aikaan luodaan uusia.

Pelivalikon yksi tärkeimmistä työvälineistä on Unityn Slider -objekti. Slideria voidaan liikuttaa hiirellä tai nuolinäppäimillä, jolloin sliderin arvo muuttuu. Arvo määrittää, mikä peli on valittuna ja arvon vaihtuessa vaihtuu myös pelivalinta. Sliderin arvon mukaan määritellään myös Info-ruudun tekstit. Tekstin kohtiin otsikko, kuvaus, assetit ja tagit tulevat julkisesta string-listasta jokaiseen kohtaan erikseen. Kuvakaappaukset toimivat samalla periaatteella eli kuvat poimitaan listasta. Toteutuksessa käytettiin myös nappeja, paneeleita, tekstiä ja kuvia. Napeissa olevat korostukset tehtiin Color Tint -transitiolla. Color Tint antaa määrittää napille normaalin värin lisäksi korostuksessa käytettävän värin niinkuin värin mikä esiintyy painettaessa. Värejä valitessa myös alfa-kanava on käytössä, millä voi muuttaa kuvaa tai väriä näkymättömämmäksi. Paneelit luovat tilan teksteille, kuten Game Selection -ruudussa peleille on valittu keskellä sijaitseva tila. Info-ruudussa on useita paneeleita kun on useita sijainteja teksteille. Paneelit antavat teksteille taustavärin tuoden tekstejä erottuvammiksi. Ruutujen otsikoissa ei ole käytetty paneelia, vaan tekstikenttää. Otsikot erottuvat erilaisen fontin ansiosta.

Ohjelmoinnin osa-alueelta laitettiin toimintoja Unityn käyttöliittymätyökaluihin. Sliderin arvon muutosta tarkkaillaan skriptissä ja toimintoja tehdään arvon mukaan. Slider-objektiin saa OnValueChanged-tapahtuman, mihin voi lisätä metodin. Kyseinen metodi suoritetaan aina kun sliderin arvo muuttuu, sillä OnValueChanged tarkkailee arvoa automaattisesti. Havainnollistamaan mikä peli on valittuna, luotiin objekti minkä Y-akselin arvoksi määritellään uusi arvo. Y-akselin arvoksi tulee games-listan objektin arvo. Käytännössä valintaobjektilla on aina sama korkeusarvo kuin pelin nimellä. Kaikille käyttöliittymä-objekteille voi laittaa event trigger -komponentin. Event

triggerin kautta voidaan objektille lisätä uusi funktio. Lopputuloksena on valmistunut info-ruutu (KUVIO 10) sekä pelivalikko. (KUVIO 11)



KUVIO 10. Valmistunut info-ruutu.



KUVIO 11. Valmistunut pelivalikko.

4.4 Peli 1 – Village Trouble

Village Trouble (KUVIO 12) demonstroi yleistä 2D RPG-pelin liikkumista, lyömistä sekä vuorovaikutusta npc-hahmojen kanssa. Pelissä ei ole taistelu-osuutta, huolimatta siitä että pelaaja voi suorittaa miekkaiskun ja kyläläiset pyytävät pelaajaa pelastamaan kylää hirviöiden vallasta. Village Troublen idea lähti Legend of the Zelda ja Pokemon-peleissä käytetystä liikkumistyylistä.

Village Trouble havainnollistaa Unityn 2D-ominaisuuksia käyttämällä hyväksi Unityn uusia käyttöliittymätyökaluja sekä 2D-pelikehitykselle suunnattuja toimintoja. Graafisen ilmeen pelille luovat käytetyt spritet, jotka antavat syvyyttä ja perspektiiviä käyttäjälle. Syvyyttä käytännössä 2D-pelissä ei ole sillä, spritet ovat leveys- ja korkeusakseleista koostuvia kuvia.



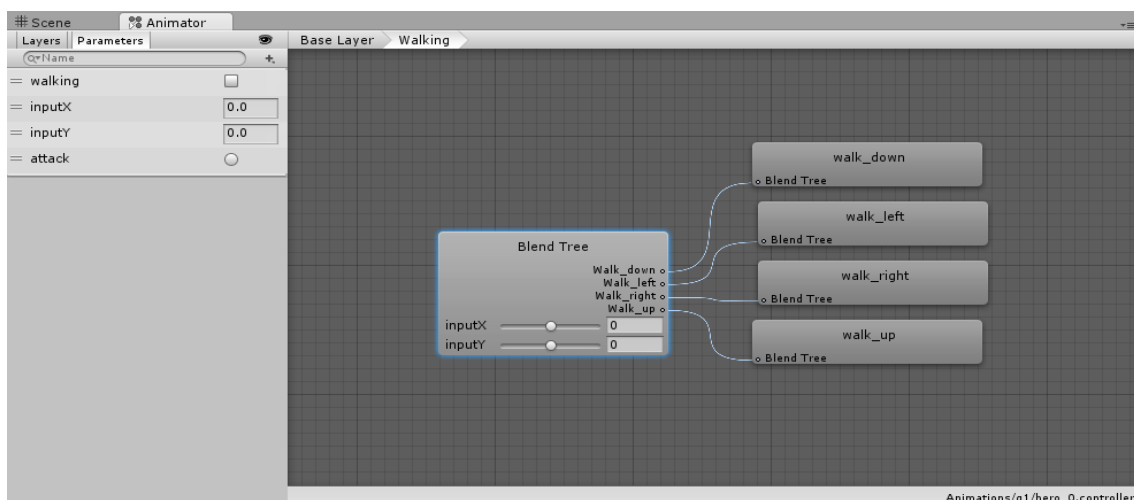
KUVIO 12. Village Trouble -peli.

4.4.1 Toteutus

Village Trouble toteutettiin Unityn 2D-kokoonpanolla. Kokoonpanossa kameran projektiio ei ole perspektiivinen kuin 3D-peleissä vaan kamerasta tehtiin ortografinen. Ortografinen kamera näyttää objektit kaksiulotteisena. Liikkuminen on sallittua ainoastaan neljään eri pääilmansuuntaan. Pelaajan liikkuessa yhteen suuntaan lukitaan liikkuminen jäljelle jääviin suuntiin, kunnes hahmolle annetaan komento vaihtaa suuntaa. Liikkumisen toteutuksessa käytettiin Unityn Vector2-luokkaa, millä voidaan työskennellä 2D-ulottuvuuden vektoreilla ja pisteillä.

Pelaajahahmolle luotiin kaksi kykyä. Pelaajan kyky jutella npc-hahmoille luotiin OnTriggerEnter2D ja OnTriggerExit2D metodeja hyväksikäyttäen. Käytännössä vuorovaikutus toimii, kun pelaaja on npc-hahmon alueen sisällä ja valitsee jutella hahmolle. Keskustelutapahtuman ollessa aktiivinen ruudulle tulostuu tekstipaneeli ilmaisemaan mitä kylän väellä on sanottavana. Keskustelu loppuu kun pelaaja liikkuu ulos npc-hahmon alueelta ja tekstipaneeli katoaa välittömästi. Pelaajan toinen kyky on iskeä miekalla pelaajan kasvojen osoittamaan suuntaan. Miekalla lyönti käy painamalla vasenta shift-näppäintä, jolloin pelaajan sprite vaihtuu väliaikaisesti spriteksi, missä pelaajan miekanisku on havaittavissa. Tapahtuman aikana pelaajan miekka välittää tiedon siitä, osuiko pelaaja hirviöön vai ei.

Animointi oli monivaiheinen sillä, ensin pelaajahahmon spritet leikattiin spritesheet-kuvasta, mikä sisälsi kaikki pelaajahahmoon liittyvät spritet. Seuraavaksi spriteistä luotiin animaatiot ja animaatioita käsiteltiin Mecanim-työkalulla. Pelaajahahmolle annettiin animaatiokontrolleri, mikä sisältää tiedot animaattorista sekä hahmoon liitetystä animaatioista. Animaattoriin lisättiin parametrit walking, attack, inputX ja inputY. Parametrit mahdollistavat vaikuttamisen animaatiokontrolleriin ja sen sisältämiin tietoihin suoraan skriptistä. Pelaajahahmon liikkuessa horisontaalisesta tai vertikaalisesti saa inputX- ja inputY-parametrit arvon, mikä määrittää mitä animaatiota käytetään. Animaation valintaan käytettiin blend tree -ominaisuutta eli käytännössä pelaajan liikkuessa otetaan animaattorin walking-tilasta parametrien mukainen animaatio. Blend tree tarkastelee parametrejä, kun inputX-parametri saa arvon 0 ja inputY-parametri arvon 1, niin suoritetaan yläsuuntaan liikkumista havainnollistava animaatio. (KUVIO 13)



KUVIO 13. Pelaajan liikkumiseen vaikuttava blend tree ja parametrit.

4.4.2 Tavoite

Varsinainen pelin tavoite on havainnollistaa Unitysta löytyvää Mecanim-työkalua kaksiulotteisessa pelissä. Liikkuminen, miekanisku ja hirviöissä tapahtuva pieni värinämäinen liike luotiin Mecanimia käyttäen. Tällä peli sai eloa ja pelin kylä ei ollut aavemaisen autio. Mecanimin käyttö havainnollistaa animaatioiden käyttöä pelissä. Kaksiulotteisessa pelissä otettiin huomioon syvyys-akselin puuttuminen pelin mekaniikoissa, mutta kaksiulotteisen pelin kehitysvaiheessa unityeditorissa käytettiin hyväksi kaikkia akseleita. Z-akselilla asetettiin spritet eri syvyytasoille. Syvyytasoilla eroteltiin spritet toisistaan, jotta työskentelystä animoitujen objektien kanssa tuli loogisempaa.

Mecanimin avulla voidaan peliin lisätä animaatioita pelistä riippumatta. 2D-peleissä animaatiot tosin luonnollisesti ovat pienempimuotoisia kuin verrattaessa 3D-peleihin. Village Troublen kaltaisessa pelissä animaatioilla pystyttäisiin toteuttamaan esimerkiksi taikoja. Mahdollisuuksia laajentamiselle ja mekaniikkojen parantelulle on, mutta pelillä havainnollistetaan myös keskustelua kyläläisten kanssa ja 2D-maailman rakennusta unityeditorilla.

4.5 Peli 2 – Butterfly

Butterfly-peli (KUVIO 14) havainnollistaa 3D-pelimaailmaa ja siihen liittyviä asioita, kuten valaistus ja varjot sekä 3D-objektit. Alussa pelin oli tarkoitus koostua vain vapaalennosta 3D-ympäristössä, mutta suunnitelmat muuttuivat, kun peli tarvitsi tavoitteen ja mekaanisia elementtejä. Pelissä näkyvät partikkeli-efektit korostavat pelaajalle tärkeitä objekteja.

Pelissä käytetään painovoimaa, nopeutta sekä korkeutta pelaajahahmona toimivan perhosen ohjaamiseen. Näitä ominaisuuksia havainnollistetaan jatkuvasti perhosen ollessa lennossa, sillä perhosen ilmalentoon pelaajalta vaaditaan varovaisia toimenpiteitä. Perhosen siivissä pyörii animaatio aina kun perhonen ei koske maahan tai objekteihin. Siipien animaatiolla demonstroidaan Mecanimin käyttöä 3D-pelin objektissa, jolloin syvyyskin on huomattavissa.



KUVIO 14. Butterfly-pelin aloitus.

4.5.1 Toteutus

Butterfly-pelissä nousee ilmaan puunkannon päältä lentoon ja aletaan keräilemään partikkeliefekteillä korostettuja renkaita. Renkaiden keräämisen jälkeen täytyy suorittaa vielä laskeutuminen sienitalon vierestä löytyvälle kukalle. 3D-maailman toteutuksessa käytettiin 3D-objekteja Unity Asset Storesta löytyvästä Toon Level Kit -assetista, minkä tekijänä on The Hedgehog Team.

Pelin käyttöliittymästä pelaaja voi tarkastella perhosen nopeutta tai korkeutta, mitkä auttavat keräämään renkaita. Käyttöliittymästä voi katsoa myös paljon aikaa on tehtävän suorittamiseen mennyt tai montako rengasta puuttuu kahdeksan renkaan tavoitteesta. Käyttöliittymä toteutettiin paneeleilla sekä tekstiobjekteilla. Paneelit antavat tekstiobjekteille korostusta, jotta olisivat ne helposti luettavissa. Kerättävät renkaat toimivat OnTriggerEnter2D-metodin avulla. Renkailla tarkastetaan pelaajan kerättyjä ja keräämättömiä renkaita. Renkaat tuhoutuvat Destroy()-funktiolla heti pelaajan koskettua renkaihin, jolloin pelaajalle lisätään yksi kerätty rengas ja käyttöliittymässä tekstit päivittyvät vastaamaan tilannetta.

Perhonen on kokoajan painovoiman vaikutuksen alaisena. Pelaaja voi painaa välilyönti-näppäintä antaakseen perhoselle nostetta kumoamaan painovoimaa. Kannolta nousun jälkeen perhonen

liikkuu kokoajan hiljaa eteenpäin ja pelaajalla on mahdollisuus nopeuttaa eteenpäinliikettä W-näppäintä painamalla. Kääntyminen onnistuu A- ja D-näppäimillä.

4.5.2 Tavoite

Pelin tärkeimpiä tavoitteita on tuoda esille 3D-pelimaailmaa ja renkaiden keräämistä sekä erilaisten voimien vaikutusta peliobjektiin. Perhoseen vaikuttava painovoima ja pelaajan antama noste luovat peliin perhosen lentotapaa muistuttavan liikesarjan. Perhosen lentotyylinä on hieman epäsäännöllinen liike Y-akselilla. Käytännössä perhonen näyttää tippuvan hiljaa maata kohden, mutta välilyönti-näppäimellä toteutettavasta nosteesta johtuen perhonen liikkuu korkeammalle. Pelin kolmiulotteinen pelimaailma luotiin metsämaisen rentouttavaksi ja perhoselle soveltuvaksi ympäristöksi. Renkaiden kerääminen toimii pelin tarkoituksena ja kerääminen voi joissain tapauksissa olla haastavaakin. Renkaiden keräämisen mahdollistaa skriptien vuorovaikuttaminen, eli skriptit tarkastelevat toistensa muuttujia ja toimivat näiden mukaan.

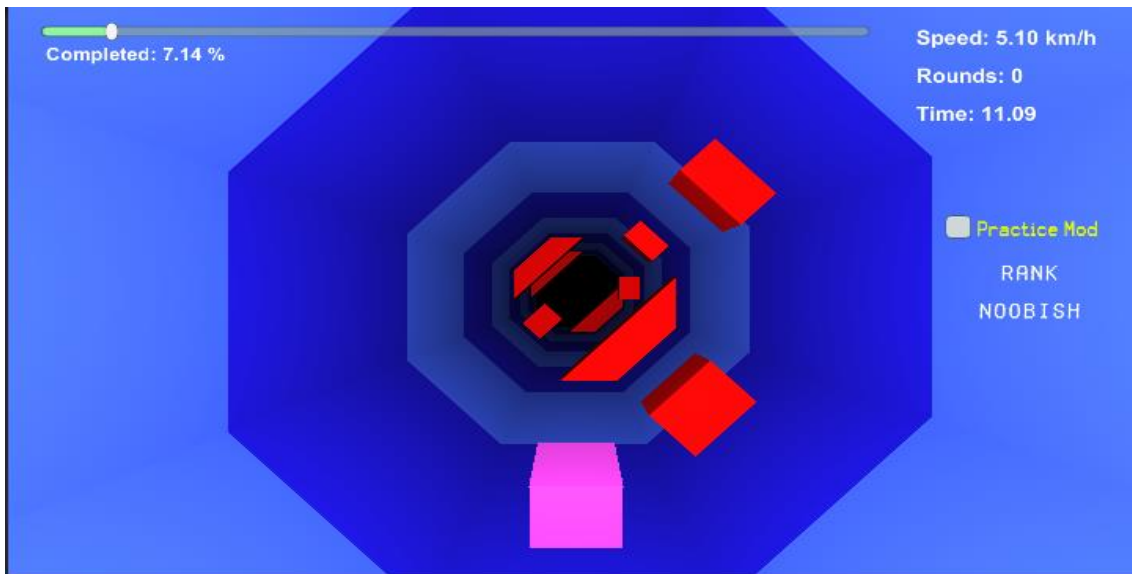
Pelistä voidaan havaita, että perhosen lepattelemiseen on käytetty Mecanimia ja siipien lepatteluun käytettyä animaatiota toistetaan aina perhosen ollessa ilmassa. Pelin edetessä on käytetty partikkeliefektejä tuomaan korostusta peliobjekteille ja näin ollen helpottamaan pelaajan navigointia oikeaan suuntaan. Partikkeleita käytetään keräiltävissä renkaissa sekä lopuksi kukassa mille yritetään laskeutua. Käyttöliittymä on pelissä tapahtuvista asioista ajan tasalla ja sillä tuodaan esille esimerkiksi kuinka kauan tehtävään menee aikaa. Tehtävänä on nousta kannolta, kerätä renkaat ja laskeutua kukalle. Käyttöliittymä on pelaajalle iso apuväline tehtävän suorittamiseen.

4.6 Peli 3 – Cubescroller

Cubescroller-pelissä (KUVIO 15) käsitellään pelaajan reaktiokykyä jatkuvasti nopeutuvaan kuutioon. Peli siis koostuu putkista, mitkä on täytetty esteillä ja pelaajan kuutio yritetään saada mahdollisemman pitkälle. Nopeutuvaa kuutiota ei varsinaisesti ohjailta vaan pelaaja voi käänellä putkea, minkä mukana kääntyy myös esteinä toimivat peliobjektit.

Kuutio pyörii nopeampaa pelin edetessä, luoden vaikutuksen siitä että pelaajan kuutio aiheuttaisi etenemisen. Visuaalista vaikutusta vahvistaa myös värit ja niiden asettelu. Automaattinen nopeutuminen ei tapahdu eksponentiaalisesti, mikä aiheuttaisi pelin nopeutuvan mahdottomaksi,

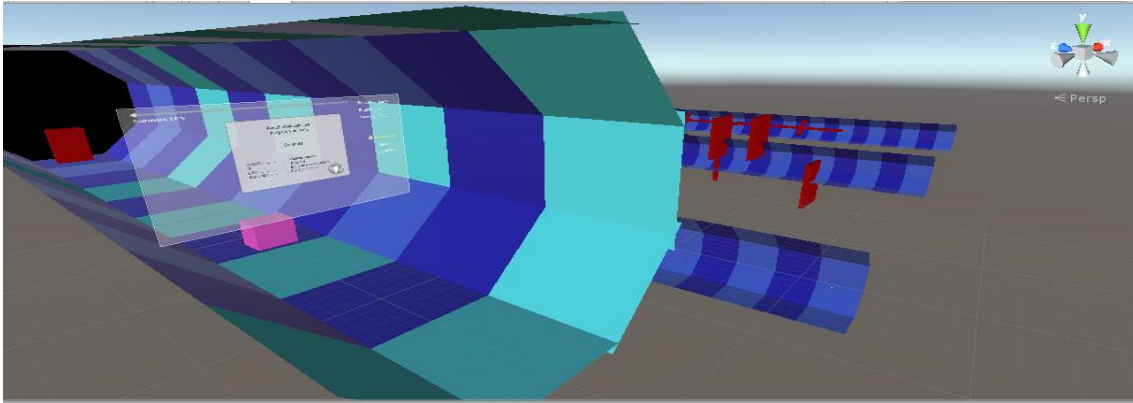
vaan nopeutuminen lasketaan pelissä kuluneen ajan mukaan. Pelin alkuvaiheessa nopeutuminen on nopeaa, mutta hidastuu kokoajan mitä pitemmälle pelissä pääsee.



KUVIO 15. Cubescroller-peli alkuvaiheessa.

4.6.1 Toteutus

Peli on rakennettu Unityn primitiivejä käyttäen. Valmiiden peliobjektien avulla luotiin neljä eri tunnelia. Pelin päätoimintaperiaate on saattaa kuutio mahdollisimman pitkälle eli varsinaista loppua ei ole, sillä peli päättyy kun kuutio törmää esteisiin. Peli lähtee liikkeelle alkutunnelista, mikä suunniteltiin esteiden kannalta helpoksi. Aloitusputken päässä on portaali, mistä pelaaja siirretään sattumalla valitun putken alkuun. Sattumalla valitaan yksi kolmesta uniikista putkesta, joista jokainen putki on esteiltään erilainen ja tuo pelaajalle haastetta. (KUVIO 16) Haasteellisemman putken päässä on portaali aloitusputken alkuun. Käytännössä alkuputken tarkoituksena on siis luoda helppo aloitus pelaajalle ja toimia uuden kierroksen alkua havainnollistavana. Alkuputken aikana pelaaja voi jo valmistautua tulevaan haasteputken sisältöön uudella nopeudella.



KUVIO 16. Aloitusputki (vas.) ja haasteellisemmat putket.

Pelaaja ei voi liikuttaa kuutiota millään tavalla vaan pelaajan ohjattava on putki, missä kuutio on liikkeellä. Putkea pelaaja voi käänellä hieman sivusuuntiin A- ja D-näppäimillä. Pelin nopeutuessa tulee tarve äkkinäisimmille liikkeille, joihin pelaajalle annetaan ratkaisu kääntää putki ylösalaisin W- ja S-näppäimillä. Putken liikuttaminen tuntuu alussa haastavalta mutta sisäistäessä kontrollit, kuution alkumatka helpottuu huomattavasti. Kuution nopeus lisääntyy alussa haastavasti, mutta pelissä voidaan laittaa päälle Practice Mode, jolloin kuutio ei voi törmätä esteisiin ja peli jatkuu niin pitkälle kun ominaisuus otetaan pois päältä.

Täydeksi kierrokseksi lasketaan aloitusputken ja yhden haasteputken sisältämä matka. Pelin käyttöliittymässä oleva suorituspalkki viestii pelaajalle kuinka paljon kyseistä kierrosta on suoritettu. Suorituspalkki toimii triggereiden avulla siten, että kuution ohittaessa triggerin lisäään suorituspalkkiin tilannetta vastaava arvo. Suorituspalkki toteuttiin käyttöliittymätyökalujen slider-objektilla. Pelin kuluessa päivitetään myös pelaajalle uusi arvonimi hienosta suoritumisesta. Arvonimi annetaan ajan mukaan kuinka pitkään pelaaja on selvinnyt loppumattomissa putkissa ja näytetään pelaajalle käyttöliittymässä.

4.6.2 Tavoite

Pelissä tuodaan esille primitiiveillä toteutettua kolmiulotteista ympäristöä ja yksi tärkeimpänä demonstroinnin kohteena on yksinkertaisen visuaalisen toteutuksen vaikutelma. Pelimekaniikoilla tavoitellaan nopeita reaktioita ja mekaniikkojen ollessa yksinkertaiset pelin oppimiskäyrä on helppo omaksua. Esteiden muokkaamisella saataisiin nykyistä laajempi, sillä esteet voisivat esimerkiksi

olla liikkuvia tai muotoa muuttavia. Cubescroller-peli havainnollistaa, että valmiiden primitiivien käytöllä yhdistettynä pelin mekaanisiin piirteisiin voidaan saavuttaa hyvä lopputulos.

Pelimekaniikkojen yhteistyö toiminnollisuuden kanssa on yksi tärkeimpiä tavoitteita, mitä peli tuo käyttäjälle. Kuution liikkuminen portaalien avulla tuo peliin moniulotteisuutta ja huomattavaa syvyyttä, sillä portaalin toisella puolella pelaajaa voisi odottaa toisen putken sijasta esimerkiksi avoin maailma tai aivan erilaiset mekaniikat. Visuaalinen tyyli sekä efektit tuovat pelin syvyyttä ja nopeatempoisuutta esille tukien pelissä käytettävää liikkumismekaniikkaa. Kuution rotaatiovahdin lisääntyminen viestii pelaajalle vahdin jatkuvaa nopeutumista.

5 YHTEENVETO

Opinnäytetyön tarkoituksena toteuttiin runko pelikokoelmalle ja kokoelmaan pelejä. Vähäinen suunnittelu vaikeutti toteutusta, sillä hyvän suunnittelun tärkeys korostui opinnäytetyön myöhemmissä kohdissa. Suunnitteluvaiheissa ongelmina oli esimerkiksi, kun en ollut täysin päättänyt kaikkia ominaisuuksia. Suunnittelu toteutettiin nopeasti paperille mikä johti siihen, että suunnitelmat olivat liian sekavat myöhempää käyttöä varten ja suunnittelussa syntyi epäselvyyksiä toiminnallisuuteen liittyen. Ratkaisu tähän olisi ollut luoda kattava suunnitteludokumentti, josta asiat olisi käynyt kohta kohdalta ilmi. Suunnittelussa olisi ollut hyvä myös huomioida millaisia skriptejä tarvitaan toiminnallisuuteen, sillä toteutusvaiheessa tuli osa käyttämättömiä toiminnallisuuksia.

Sisällön toteuttaminen onnistui odotusten mukaisesti. Kokonaisuudesta tuli toimiva ja käytännöllinen, mutta graafisesti hieman kankea. Opinnäytetyön tarkoituksena ei ollut graafinen tuottaminen, mutta kuvankäsittelyohjelman käytöstä tuli enemmän tietoutta. Suurin osa graafiikoista ladattiin internetistä, mikä rajoitti työn ulkoasua hyvin pitkälle. Toiminnallinen sisältö on hyvin tiivistetty. Materiaalin lisääminen ja poistaminen on helppoa myöhempää käyttöä ajatellen. Lopputulokseen olen tyytyväinen, sillä työtä tehdessä tuli etsittyä paljon tietoa käytetyistä työkaluista ja kehitysprosessikin selveni. Yksi tärkeimpänä opittuna asiana on suunnittelun perusteellinen toteuttaminen.

Jatkokehityksiä projektille olisi ulkoasun päivityksen lisäksi pelien kontrollien säätämistä. Peleihin voitaisiin myös lisätä enemmän pelattavaa sisältöä tai luoda tarinaa. Lisäksi pelit voisivat liittyä toisiinsa, jolloin loppukäyttäjälle voitaisiin luoda mahdollisuus tarkastella tarinaa eri perspektiiveistä. Jokaiselle pelille voitaisiin toteuttaa alkuruutu, mistä olisi mahdollisuus asetusten konfigurointiin.

LÄHTEET

Adobe Systems 2015. Viitattu 05.04.2015, <http://www.adobe.com/products/photoshop.htm>.

Blend Swap 2015. Blend Swap. Viitattu 05.04.2015, <http://www.blendswap.com/>.

CBS Interactive 2015a. Game Engines: How do they work? Viitattu 02.05.2015, <http://www.giantbomb.com/profile/michaelenger/blog/game-engines-how-do-they-work/101529/>.

CBS Interactive 2015b. Character Select Screen. Viitattu 01.04.2015, <http://www.giantbomb.com/character-select-screen/3015-2125/>.

CGTextures.com 2015. CGTextures. Viitattu 05.04.2015, <http://www.cgtextures.com/>.

Leap Motion 2014. Menu Design Guidelines. Viitattu 27.03.2015, https://developer.leapmotion.com/documentation/csharp/practices/Leap_Menu_Design_Guidelines.html.

MonoDevelop Project 2015. Cross platform IDE for C#, F# and more. Viitattu 24.3.2015, <http://www.monodevelop.com/>.

OpenGameArt 2015. OpenGameArt. Viitattu 05.04.2015, <http://opengameart.org/>.

The Game Contriver 2015. Create A Dynamic/Responsive Menu. Viitattu 01.04.2015, <http://www.thegamecontriver.com/2014/08/create-a-menu-in-unity-4-6.html>.

Unity Technologies 2015a. Unity5. Viitattu 22.3.2015, <http://unity3d.com/>.

Unity Technologies 2015b. Learning the Interface. Viitattu 22.3.2015, <http://docs.unity3d.com/Manual/LearningtheInterface.html>.

Unity Technologies 2015c. Importing Assets. Viitattu 23.3.2015,
<http://docs.unity3d.com/Manual/ImportingAssets.html>.

Unity Technologies 2015d. Animation Overview. Viitattu 23.3.2015,
<http://docs.unity3d.com/Manual/AnimationOverview.html>.

Unity Technologies 2015e. Particle Systems. Viitattu 27.3.2015,
<http://docs.unity3d.com/Manual/ParticleSystems.html>.

Unity Technologies 2015f. A Feature-Rich and Highly Flexible Editor. Viitattu 23.3.2015,
<http://unity3d.com/unity/editor>.

Unity Technologies 2015g. Asset Store. Viitattu 05.04.2015,
<http://blogs.unity3d.com/category/asset-store/>.

Unity Technologies 2015h. Submission Guidelines. Viitattu 03.05.2015, <https://unity3d.com/asset-store/sell-assets/submission-guidelines>.

Wikimedia Foundation 2015a. Adobe Photoshop. Viitattu 04.04.2015,
http://en.wikipedia.org/wiki/Adobe_Photoshop.

Wikimedia Foundation 2015b. Digital Asset. Viitattu 05.04.2015,
http://en.wikipedia.org/wiki/Digital_asset.

Wikimedia Foundation 2015c. Creative Commons. Viitattu 03.05.2015,
http://en.wikipedia.org/wiki/Creative_Commons.