

2D-mobiilipelin tasosuunnittelu

Case: KEY-peli

LUT AINEISTOT

Tiivistelmä

Tekijä(t) Toroskainen, Mikko	Julkaisun laji Opinnäytetyö, AMK Sivumäärä 38	Valmistumisaika 2020
Työn nimi 2D-mobiilipelin tasosuunnittelu Case: KEY-peli		
Tutkinto Insinööri (AMK)		
Ohjaavan opettajan nimi, titteli ja organisaatio Ismo Jakonen, lehtori, Tieto- ja viestintäteknikka		
Toimeksiantajan nimi, titteli ja organisaatio LUMA Suomi		
Tiivistelmä Tässä työssä keskityttiin LUMA Suomi hankkeen tilaamaan 2D-mobiilipeliin, joka opettaa koululaisille asioita kivistä, energiasta sekä ympäristöstä. Teoriaosuus koostui siitä, mitä tasosuunnittelu on, sen vaiheita, historiaa ja siihen tarvittavia työkaluja. Case osuudessa käsitellään Key-pelin tasosuunnittelua sekä grafiikan luontia.		
Asiasanat 2D, 3D, sprite, Unity, Photoshop, Illustrator, Asset		

Abstract

Author(s) Toroskainen, Mikko	Type of Publication Thesis, UAS	Published 2020
	Number of Pages 38	
Title of Publication Level design of 2D mobile game Case: KEY-game		
Name of Degree Engineer (UAS)		
Name, title and organization of the supervising teacher Ismo Jakonen, Senior Lecturer, Information and communication technologies		
Name, title and organization of the client LUMA Suomi		
Abstract In this research the focus was on 2D mobile game which was ordered by LUMA Suomi. Game educates things about rocks, energy and environment. Theory portion consists of what level design is, steps of level design, history and level design tools. Case portion consists level design of the key-game and creation of graphics.		
Keywords 2D, 3D, sprite, Unity, Photoshop, Illustrator, Asset		

Sisällys

1	Johdanto.....	1
2	Pelien tasosuunnittelu.....	2
2.1	Vaiheet	2
2.2	Suunnittelu	3
2.3	Tasosuunnittelun historia.....	4
3	Tasosuunnittelun työkalut	9
3.1	Pelimoottorit.....	9
3.1.1	Unity	9
3.1.2	Unreal Engine.....	10
3.1.3	CryEngine.....	11
3.2	Graafiset suunnitteluohjelmat	12
3.2.1	Kaupalliset ohjelmat.....	13
3.2.2	Ilmaiset ohjelmat.....	14
4	2D-pelin graafiset elementit	15
4.1	Spritet.....	15
4.1.1	Käyttöliittymäsprite.....	16
4.1.2	Tason spritet.....	16
4.2	Muut elementit	17
5	Case – Key-peli	19
5.1	Pelin tavoite.....	19
5.2	Tasosuunnittelu	20
5.3	Toteutus	20
5.3.1	Spritejen teko.....	21
5.3.2	Spritejen vienti Unityyn	27
5.3.3	Tasojen luonti	29
6	Yhteenveto	34
	Lähteet	35

1 Johdanto

Nykyään pelillisyyttä käytetään paljon opettamisessa. LUMA Suomi hanke pyrkii kehittämään opetuspelejä koululaisille, jonka vuoksi KEY-peliä lähdettiin kehittämään. Pelillisyyttä on lisätty opetuksessa, koska sen on nähty kiinnostavan ja motivoivan oppilaita.

Tässä työssä keskitytään 2D-mobiilipelin tasosuunnitteluun sekä visuaalisuuteen. Teoriaosuudessa käydään läpi mitä tasosuunnittelu on, sen vaiheita ja suunnittelua sekä tasosuunnittelun historiaa. Työssä tutustutaan, miten 2D-peliin luodaan grafiikkaa sekä miten sitä käytetään tasojen luonnissa. Työssä esitellään myös eri pelimoottoreita sekä graafisia ohjelmia.

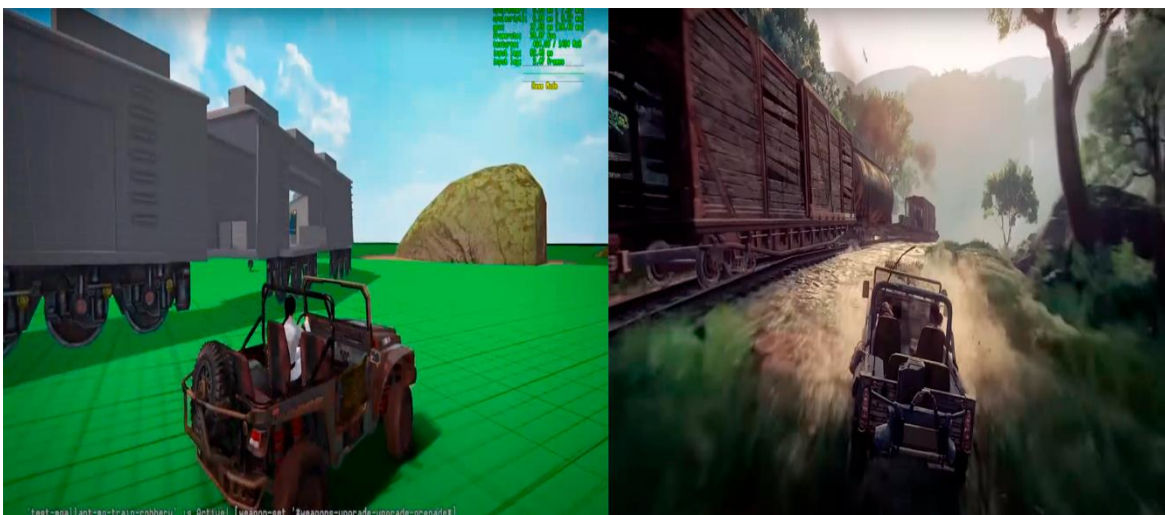
Lopuksi työssä katsotaan koululta tilattua KEY-peliä, eli kivi- energia ja ympäristö peliä. Mikä pelin tavoite ja tarkoitus on, kenelle peli on suunniteltu, mikä oli pelin alkutilanne sekä miten sen tasosuunnittelu ja visuaalisuus toteutetaan. Työssä käydään läpi pelin tasosuunnittelun vaiheita, sekä asettien ja tasojen tekoon käytettyjä tärkeimpiä työkaluja.

2 Pelien tasosuunnittelu

Tasosuunnittelulla tarkoitetaan pelien tasojen, kenttien ja tehtävien suunnittelua ja tekoa. Yleensä suurissa ja keskisuurissa pelistudioissa on erikseen tasosuunnittelija. Indie eli itsenäiset pelin kehittäjät tekevät yleensä pienellä ryhmällä pelejä, jolloin ei ole erikseen tasosuunnittelijaa, vaan he tekevät muita tehtäviä samalla. Tasojen suunnittelussa pitää ottaa huomioon, että tasojen pitää olla erilaisia ja tuoda aina jotain uutta mukanaan, jotta pelaajan mielenkiinto peliin pysyy. Pelin päämekaniikkojen on kuitenkin pysyttävä samana sillä, kun pelaaja on opetettu pelaamaan peliä tietyllä tavalla ja mekaniikoilla, eivät pelaajat halua uusia epäjohdonmukaisia mekaniikkoja tulevan satunnaisesti. (Rouce 2005 s.10) Tasojen pitää sopia pelin tarinaan, jotta kokonaisuudesta tulee järkevä.

2.1 Vaiheet

Hyvä tasosuunnittelu on erittäin tärkeää, kun tehdään peliä, josta pelaaja nauttii. Tasosuunnittelijan on ymmärrettävä kaikkia pelin komponentteja ja kuinka ne sopivat yhteen. (Smith 2008) Tasojen suunnitteluun ei ole mitään tiettyä oikeaa tapaa. (Byrne 2004 s.81) Aluksi tasosuunnittelussa on mietittävä mahdollista pelin tarinaa, jotta tiedetään, minkälaisia tasoja lähdetään tekemään. Pelin lajityyppi vaikuttaa myös tasojen rakenteeseen. Tasoja rakentaessa tehdään yleensä tason tai kentän hahmotelma eli äärirajat, josta tasoa lähdetään tekemään. Tämän jälkeen rakennetaan taso saatavilla olevista peli-aseteista, jotka tässä vaiheessa ovat vain joko vain kuutioita tai sitten puoliksi tehtyjä. Kuvassa 1 näkyy kohta Uncharted The Lost Legacy pelin prototyypistä sekä miltä kohta näyttää valmiissa pelissä. Usein tehdään monta eri versiota suunnitellusta tasosta, jonka jälkeen valitaan paras tai toimivin vaihtoehto. Jos peli sijoittuu oikeaan paikkaan, on hyvä ottaa mallia paikasta käyttäen karttapalveluita. Kun prototyyppi versio tasosta on valmis, voidaan taso tehdä loppuun käyttäen parempia peli-asetteja. (Barclay 2020) Tasoa voidaan koristella lopuksi, jotta se näyttäisi paremmalta. Pelin testausvaiheessa yleensä ilmenee tasossa rakenteellisia virheitä tai heikkouksia, jolloin tasoa joudutaan muuttamaan.



Kuva 1. Prototyypistä lopulliseen peliin (YouTube, 2018 A, YouTube, 2017 A)

2.2 Suunnittelu

Kun lähdetään suunnittelemaan tasoja videopeliin, on otettava huomioon, minkälainen peli on. Jos pelissä on ns. avoin maailma, niin silloin pelaaja pystyy liikkumaan, tutkimaan ja päättämään itse mitä tehtäviä tekee pelimaailmassa vapaasti, jolloin taso on epälineaarinen. (Sefton, 2008) Tasosuunnittelun kannalta avoimen maailman haasteet ovat yksityiskohtien luominen sekä alueiden vaihtelevuus. Kuvassa 2 näkyy Red Dead Redemption pelin kartta, jossa pelaaja pystyy liikkumaan täysin ilman latauksia.



Kuva 2. Avoimen maailman kartta (YouTube, 2010)

Jos pelissä on lineaarinen pelimaailma, niin pelaajalla on tarkat rajat ja maailma on rajoitettu. Tasot lineaarisessa maailmassa ovat yleensä yksityiskohtaisempia, kun avoimessa maailmassa. Tason suunnittelua ja yksityiskohtien asettelu on helpompaa, kun pelaajan

liikkuminen on ennalta arvattavampaa. Avoimen ja lineaarisen pelimaailman yhdistelmä on semi-avoin pelimaailma. Tällöin pelimaailmassa on yksi iso avoin alue kerrallaan, jossa pelaaja voi liikkua vapaasti. Alueita on monta erilaista, mutta näiden välillä ei voi liikkua vapaasti, vaan siirtyessä uuteen alueeseen on pelin ladattava se. Tasosuunnittelussa tämä on helpompi, kun täysin avoin maailma, koska alueita suunniteltaessa alueiden väliset erot voivat olla paljon suurempia, kun eri alueiden rajat eivät ole näkyvillä selkeästi esim. yksi alue voi olla jäätä ja toinen taas kuiva aavikko. Avoimessa maailmassa suurien erojen pitää tulla sulavasti, esim. jos liikutaan talvisesta maisemasta kesäiseen maisemaan, on näiden välillä oltava välivaihe.

3D pelien tasoja suunniteltaessa tasoista pystyy tekemään realistisempia ja liikkumisen kannalta monipuolisempia kuin 2D-peleissä, koska 2D-peleissä liikkuminen on rajattu horisontaaliseen ja vertikaaliseen akseliin, kun taas 3D-peleissä pystyy liikkumaan monipuolisemmin. 2D-peleissä tasojen luonti on nopeampaa ja halvempaa, koska asettien luonti on helpompaa ja ne ovat yksinkertaisempaa. (Eden, 2020)

Jos peli on moninpeli niin tasosuunnittelijan on otettava huomioon kentän tasapainotus eli kentässä ei saa olla paikkoja tai kohtia, jotka luovat liiallisen ylivoiman pelaajalle esimerkiksi, jos tasossa on paikka, johon pääsee vain yhdestä kohtaa ja ensimmäisenä sinne kerennyt pelaaja on voittamaton. Kuitenkin moninpelitasoissa voi olla kohtia, joita hallitsemalla saa etulyöntiaseman. Jos taso ei ole symmetrinen, kuuluu tasossa olla strategia, joka tasapainottaa sen. (Saltzman, 2020)

2.3 Tasosuunnittelun historia

Pelit olivat alussa vain 2D-muodossa, koska niiden tekeminen oli helpompaa grafiikan ja ohjelmoinnin kannalta. Tasot olivat lyhyitä mutta haastavia ja suurin osa peleistä kesti vain muutamia tunteja. Tasot olivat rakenteeltaan yksinkertaisia ja yleensä niissä ei ollut muuta tavoitetta, kun mennä tason aloituspisteestä tason loppupisteeseen. Alussa peleissä oli vain yksi ruutu kerrallaan eli tasot toimivat vain yhden ruudun sisällä. Tänä päivänäkin jotkut pelit käyttävät vain yhtä ruutua.



Kuva 3. Super Mario Bros. (YouTube, 2016 A)

Ensimmäisiä sivuttaisvierittäviä pelejä eli pelejä, joissa kuva liikkuu pelaajan mukana, oli Super Mario Bros. vuonna 1985, joka näkyy kuvassa 3. (Gaming, 2017) Tämä vei tasosuunnittelua eteenpäin, koska nyt tasot ei olleet lukittu vain yhteen ruutuun, vaan yhtä tasoa pystyi jatkamaan horisontaalisesti ja vertikaalisesti ilman uuden tason latausta. Joissakin peleissä ruutu liikkuu itsestään ja pelaajan on pysyttävä sen mukana.



Kuva 4. Metroidvania kartta (YouTube, 2018 B)

1986 julkaistut Metroid ja Castlevania pelit toi tunnetuksi tasosuunnittelutyylin, josta tullut tyyli nimeltään Metroidvania. Siinä yhdistellään elementtejä roguemaisista peleistä ja

roolipeleistä. Tunnetuin elementti metroidvania peleissä on yhdistetty maailma eli pelimaailman eri alueet on yhdistetty toisiinsa ja niihin voi palata aina takaisin. Kuvassa 4 näkyy Castlevania Symphony of the Night pelin kartta, josta näkyy hyvin metroidvania tyylinen taso suunnittelu. Usein pelaajaa houkuteltaan palaamaan vanhoille alueille siten, että ensimmäisellä kerralla alueella tulee vastaan esteitä, joista ei pääse läpi sen hetkisillä kyvyillä tai esineillä. (Giantbomb 2020)



Kuva 5. Battlezone 1980 (YouTube, 2014)

Ensiaskleet 3D peleihin nähtiin Atarin kehittämässä Battlezone kolikkopelissä vuonna 1980, joka näkyy kuvassa 5. (Shahrani 2006 s.1) Tämä oli mullistava uudistus tasosuunnittelussa, kun muut sen ajan pelit olivat 2D-muodossa.



Kuva 6. Spacewar 1962 (YouTube, 2017 B)

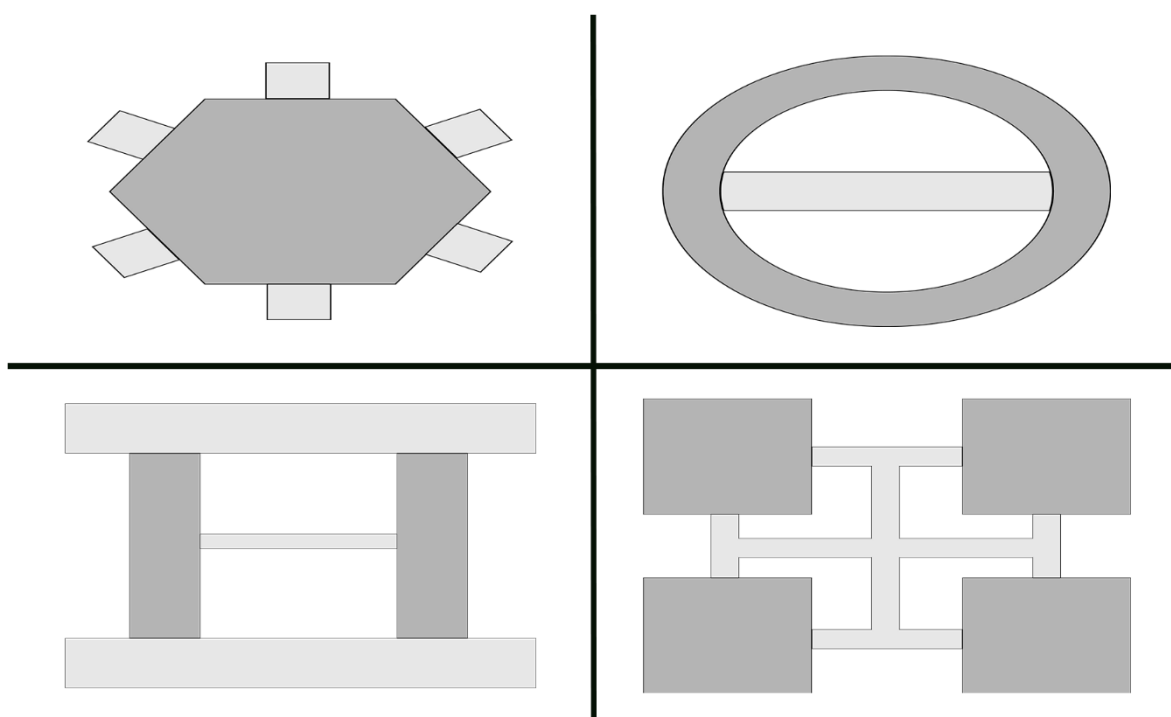
Ensimmäinen videopeli, joka käytti videografiikkaa, oli Massachusetts Institute of Technologyn eli MIT:n opiskelijoiden PDP-1 tietokoneelle kehittämä Spacewar vuonna 1962. (Wolf, 2008, s. xvii) Kuvassa 6 näkyy Spacewarin pelitilanne.



Kuva 7. Crash Bandicoot (YouTube, 2012)

Pelikonsoleiden viidennellä eli 32/64-bittisellä sukupolvella, jossa tunnetuimpia konsoleita olivat Nintendo 64 sekä Sony Playstation 3D-grafiikka yleistyi. 3D-grafiikan ansiosta tasosuunnittelu muuttui täysin, kun nyt käyttöön saatiin z akseli, x ja y akselin lisäksi. 3D maailman ansiosta pelaajahahmo pystyi nyt tutkimaan kenttiä täysin eri tavalla mitä oli

pystynyt 2D-peleissä. Kuvassa 7 näkyy Crash Bandicoot peli, joka oli yksi suosituimmista 3D tasohyppelypeleistä. 3D grafiikan ansiosta pelaaja pystyi nyt liikuttamaan hahmoa eteen, taakse, ylös sekä viistoon. Grafiikoista ja pelimaailmasta pystyi nyt tekemään realistisemman kuin aikaisemmin. Koska konsolit ja tietokoneet olivat tehokkaampia kuin ennen, pystyttiin luomaan isompia ja realistisempia pelimaailmoja. (Fandom.com) Vaikka ensimmäisiä vapaan maailman pelejä oli jo vuonna 1986 Nintendo Entertainment systemille julkaistussa The legen of zelda pelissä niin esimerkiksi Silent Hill ja Final Fantasy VII tasot olivat lähempänä nykyistä vapaan maailman tyyliä. Nykyään peleissä vapaa maailma on lähes saumaton eli pelaajan liikkeessä maailmassa ei ole erikseen alueiden välisiä lataustaukoja ja yleensä rakennuksien sisään menossakaan ei jouduta lataamaan.



Kuva 8. Moninpelitasotyypit

Moninpeleissä tasot eivät ole kokeneet niin suurta muutosta kuin yksinpelit. Aluksi moninpelitasot olivat pieniä ja melkein kaikkien pelien tasot olivat tyyliltään areena, kehä, lineaarinen sekä sijaintipainotteinen. Areenatasoissa on keskikohta, jossa melkein kaikki tapahtuu. Kehätasoissa pelaajien on suunniteltu kulkevan kehää ilman että heidän täytyisi kääntyä takaisin kertaakaan. Lineearisissa tasoissa kaikki perustuu muutamaankin polkuun, joissa pelaajat kohtaavat. Sijaintipainotteisissa tasoissa alueet eroavat toisistaan ja on aina helpposti tunnistettavissa. Joka alueella on omia kohtia, jossa tapahtuu. (Saltzman 2020) Kuvassa 8 näkyy esimerkki jokaisesta tasotyypistä. Ylärivillä on areena, sekä kehätaso ja alarivillä lineaarinen, sekä sijaintipainotteinen taso. Tummallilla merkityt alueet ovat kohtia, jossa pelaajien kohtaamisten on suunniteltu tapahtuvan. Nykymoninpelien kartat ovat isompia ja joissain on myös vapaa maailma.

3 Tasosuunnittelun työkalut

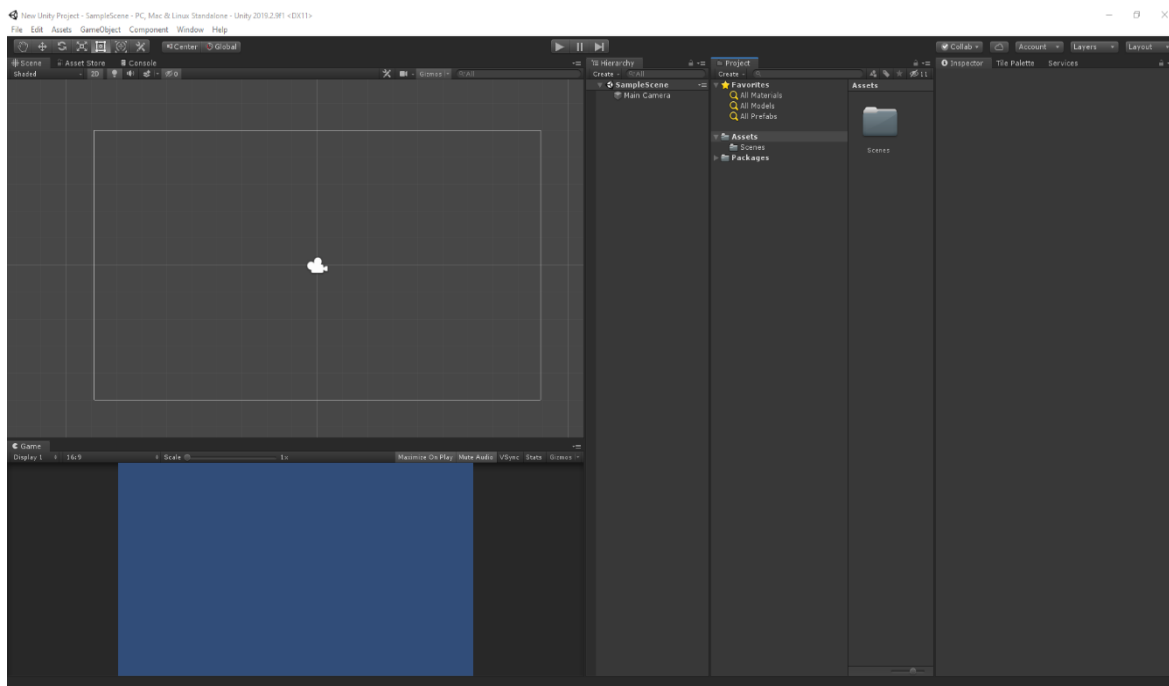
Videopelien tasoja suunniteltaessa ja luodessa tarvitsee monia työkaluja. Kynä ja paperi on hyvä aloitus, mutta tärkeimmät pelejä luodessa ovat pelimoottorit ja graafiset ohjelmat. Graafisissa ohjelmissa luodaan peliin tulevat assetit ja pelimoottoreissa ne sitten asetetaan kohdalleen ja luodaan itse tasot. Työkaluja valittaessa pitää ottaa huomioon mitkä ovat so-pivat parhaiten omaan käyttöön.

3.1 Pelimoottorit

Pelimoottorit ovat kehitysympäristöjä, jotka on pääosin suunniteltu pelin kehittäjille, sekä pelaajille. Pelimoottorit yleensä sisältävät muun muassa grafiikan renderointimoottorin 2D sekä 3D:hen, fysiikkamoottorin, joka simuloi fyysisten esineiden vuorovaikutusta ympäris-töön sekä työkaluja ääniin, ohjelmointiin, lokalisointiin ja tekoälyyn. (Studytonight.com, 2020) Vaikka pelimoottorit sisältävät joitakin valmiiksi tehtyjä spritejä, ääniä sekä 3d malleja, pelinkehittäjät luovat omat peli-asettinsa peleihin. Yleensä kehittäjät käyttävät valmiina olevia pelimoottoreita, joista suosituimmat ovat Unreal sekä Unity. Näistä moottoreista ke-hittäjät joutuvat maksamaan moottorien omistajille. Kehittäjät voivat myös luoda omat moot-torinsa, jolloin ei tarvitse maksaa lisenssiä, kehittäjä tietää paremmin miksi jokin asia toimii tai ei toimi sekä moottorin työkalut on luotu juuri omaa peliä varten, joka tekee pelin kehit-tämisestä sulavampaa. Oman moottorin teko on erittäin aikaa vievää, jonka takia valitaan usein valmis moottori.

3.1.1 Unity

Unity vuonna 2005 julkaistu Unity Technologiesin kehittämä monialustainen pelimoottori. Sillä voidaan luoda 3D, 2D, virtuaalitodellisuus sekä lisätty todellisuus pelejä. Unitystä on ilmainen sekä maksullinen versio. Ilmainen versio on henkilökohtaiseen tai pieniä kehittäjiä varten. Jos yrityksen liikevaihto on yli 100 000\$ on käytettävä maksullista versiota. Unityn mukana tulee myös Asset store kauppapaikka, josta saa sekä maksullisia että ilmaisia as-setteja. Indie eli itsenäiset kehittäjät suosivat unityä sen ilmaisuuden takia.

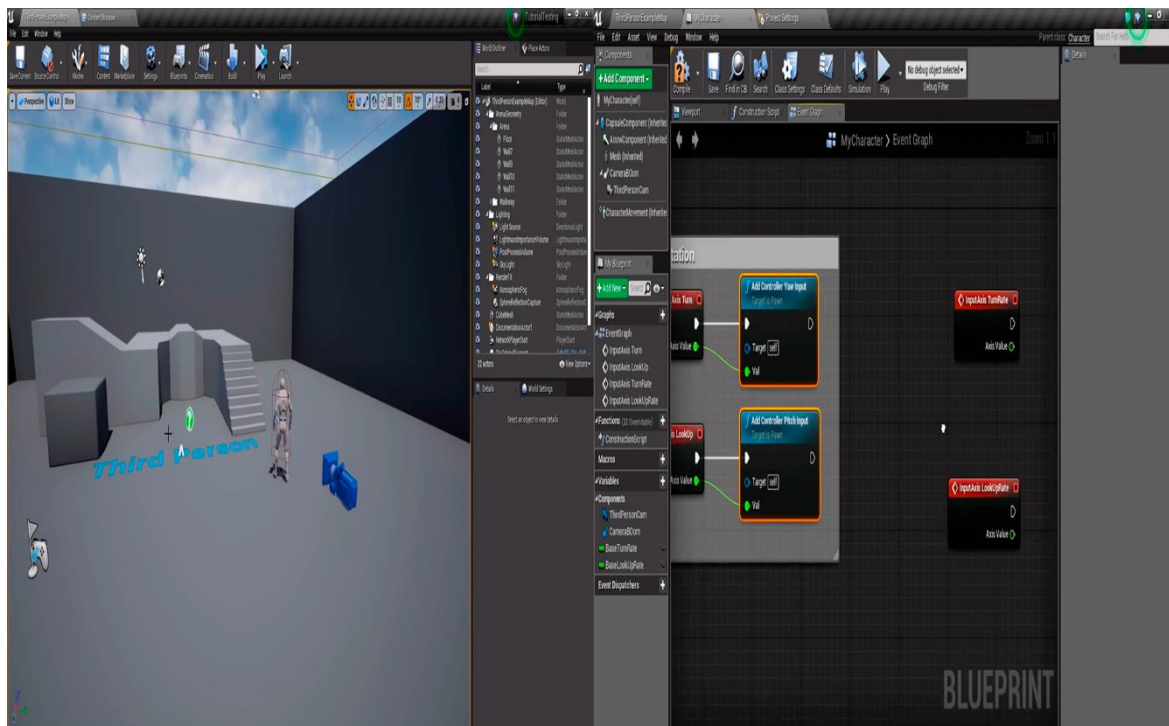


Kuva 9. Unity editor

Unityn sivujen mukaan kehittäjistä yli puolet vuonna 2019 käytti Unityä. Ohjelmointikielistä Unity tukee C# sekä Javascriptiä. (Unity 2020) Kuvassa 9 näkyy Unityn käyttöliittymä. Tunnetuimpia Unity moottoria käyttäviä pelejä ovat muun muassa battle royale henkinen Fall guys: Ultimate knockout, selviytymis henkinen My summer car sekä ensimmäisen persoonan ammutamoninpelejä Escape from tarkov.

3.1.2 Unreal Engine

Unreal Engine on Epic Games:in vuonna 1998 Unreal peliä varten kehittämä pelimoottori. Unreal on yksi suosituimmista pelimoottoreista. Unreal käyttää tukea C++ ohjelmointikieltä, sekä Blueprints nimistä visuaalista, node pohjaista ohjelmointijärjestelmää. Siinä yhdistellään nodeja toisiinsa. Blueprintiä pidetään aloittelijaystävällisempänä kuin c++:saa. Se voi myös muuttua liian sekavaksi, jos nodeja tulee paljon. C++ on joustavampaa ja sillä voi luoda monimutkaisempia mekaniikkoja kuin Blueprintillä. Unrealin mukana tulee kauppa- paikka, josta saa ilmaisia ja maksullisia assetteja.

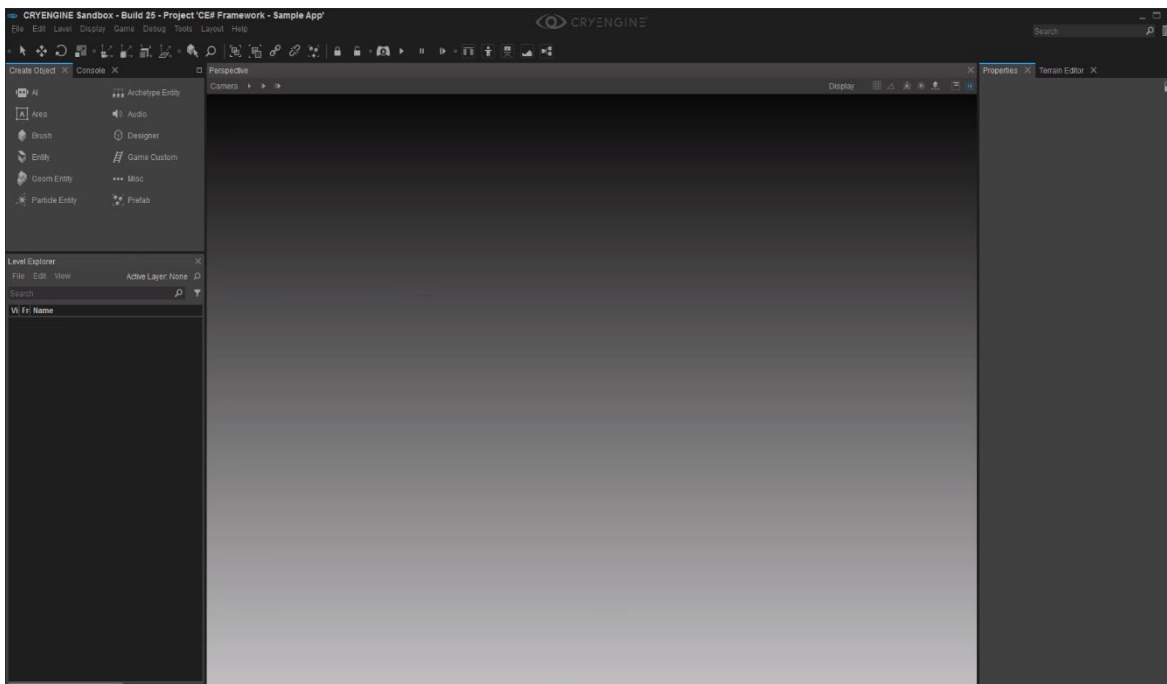


Kuva 10. Unreal Engine (YouTube, 2020, YouTube, 2016 B)

Unreal Engine on ilmainen siihen asti, kunnes liikevaihto ylittää miljoona dollaria. Miljoonan tienattua Epic Gamesille pitää maksaa viiden prosentin rojaltili pelin myynneistä. (Machkovech 2020) Kuvassa 10 vasemmalla näkyy Unrealin käyttöliittymä ja oikealla näkyy node-pohjainen Blueprint ohjelmointijärjestelmä. Tunnetuimpia Unreal Engineä käyttäviä pelejä ovat muun muassa kolmannen persoonan ammutapeli Gears of war, kauhupeli Outlast sekä ajoneuvojalkapallopelejä Rocket League.

3.1.3 CryEngine

CryEngine on vuonna 2002 julkaistu Crytekin kehittämä pelimoottori. Ensimmäinen CryEngineä käyttänyt peli oli vuonna 2004 julkaistu Far Cry. Ohjelmointikielistä CryEngine tukee C++, lua ja C#. CryEnginestä on maksettava Crytekillä viisi prosenttia rojalteja. Ensimmäiset viisi tuhatta dollaria on rojaltilivapaita. (cryengine.com 2020) Myös CryEnginessä on oma kaupapaikka.



Kuva 11. CryEngine (YouTube, 2017 C)

CryEnginen vahvuus on kasvillisuus, koska siinä on itsessään valmiina erittäin yksityiskohtaisia kasvillisuus ja vesi asetteja. (Wilson 2019) Kuvassa 11 näkyy CryEnginen käyttöliittymä. Tunnetuimpia CryEngineä käyttäviä pelejä ovat muun muassa ensimmäisen persoonan ammutapeli Crysis, ensimmäisen persoonan ammutamonipeli Hunt Showdown sekä toimintaroolipeli Kingdom Come: Deliverance.

3.2 Graafiset suunnitteluohjelmat

Tietokonegrafiikassa graafisten ohjelmien avulla voi muokata kuvia tai malleja tietokoneen avulla. Tietokonegrafiikat voidaan jakaa kahteen eri kategoriaan: Bittikarttagrafiikka ja vektorigrafiikka.

Bittikarttagrafiikka on tapa esittää digitaalisessa muodossa tallennettuja kuvia. Bittikarttagrafiikka koostuu pikseleistä. Sitä voidaan muokata kuvankäsittelyohjelmilla. Kuvankäsittelyohjelmien avulla voidaan esimerkiksi muokata pikseleiden kokoa, sijaintia, määrää, väriä sekä kirkkautta. Käytettäviä kuvaformaatteja ovat muun muassa PNG, JPEG ja GIF. (Britannica.com, 2020)

Vektorigrafiikka on koordinaatistoon sidottuihin objekteihin kuten ympyröihin, monikulmioihin ja suoriin perustuvaa grafiikkaa. Objektien muodot sekä ominaisuudet esitetään matemaattisin funktioin sekä koordinaatein. (Chapman 2000 s.86-87) Vektorikuvista ei tule rakeisia suuretaessa, kuten bittikuvista. Bittikuvissa kuvien koko riippuu pikseleiden määrästä, kun taas vektorikuvissa kuvan koko riippuu yksityiskohtien määrästä.

Vektorigrafiikkaa käytetään esim. karttaohjelmissa. Käytettäviä tiedostomuotoja ovat muun muassa SVG, EPS ja PDF.

3.2.1 Kaupalliset ohjelmat

Adobe Photoshop on kuvankäsittelyohjelma, jonka on kehittänyt Adobe Systems. Sen ensimmäinen versio Photoshop 1.0 julkaistiin vuonna 1990. (Shustek 2013) Photoshopista on myös mobiiliin tehty versio normaaleiden tietokoneversioiden lisäksi. Tänä päivänä Photoshop on markkinajohtaja digitaalisten kuvien muokkauksessa, ja siitä on muodostunut alalle standardi. Sen suosiosta kertoo myös se, kun puhekielessäkin käytetään yleisesti sanaa photoshoppaus. Photoshop käyttää kuukausimaksujärjestelmää, mutta tilaus on otettava aina koko vuodeksi kerralla. Se ei ole helpoin ohjelma oppia sen valtavan työkalumäärän takia. Photoshopin maskaus, uudelleen käsittely sekä tasotyökalut ovat parhaita koko markkinoilla. (Bateman 2020)

Luminar on Skylumin kehittämä tekoälyyn perustuva kuvanmuokkausohjelma. Se julkaistiin vuonna 2016. Se suunniteltiin olemaan halvempi versio Adoben Lightroomista. Luminar on tunnettu sen tehokkuudesta, eli sillä saa näyttämään kuvat hyvältä todella nopeasti. Sen pystyy ostamaan kertamaksulla itselleen. (Dempsey 2020) Luminarin saa joko omana sovelluksenaan tai sen saa liitännäisenä Adoben lightroomiin sekä Photoshoppiin. (Skylum.com 2020)

Adobe Illustrator on vektorigrafiikan piirto-ohjelma, jonka on kehittänyt Adobe Systems. Se julkaistiin ensikertaa vuonna 1987. Sillä tehdään yleensä logoja, kirjojen kansia sekä julisteita. Illustratorilla pystyy tekemään grafiikkaa muun muassa elokuvien jälkituotantoon sekä verkkosivuille. Monet käyttävät illustratoria myös taiteen luomiseen. Illustrator käyttää myös Adoben kuukausimaksujärjestelmää, jolloin on tilaus otettava koko vuodeksi. Illustrator on ammattikäyttöön suunniteltu ohjelma.

CorelDraw on vektorigrafiikan piirto-ohjelma, jonka on kehittänyt Corel corporation. Ensimmäinen versio siitä julkaistiin vuonna 1989. (Metrailier 2007) Se tunnetaan Adoben Illustratorin ensisijaisena kilpailijana. CorelDrawin pystyy ostamaan itselleen kertamaksulla. Sitä käytetään enimmäkseen erilaisia dokumentteja korteista mainoksiin. Sen vahvuus on aloittelijaystävällisyys, eli sen oppiminen on helpompaa kuin muissa vektorigrafiikkaohjelmissa. (Admec Multimedia 2014)

3.2.2 Ilmaiset ohjelmat

GIMP on täysin ilmainen avoimen lähdekoodin kuvankäsittelyohjelma. GIMPin ensimmäinen versio julkaistiin vuonna 1996.(gimp.org) Se tunnetaan eräänä parhaimmista ilmaisista kuvankäsittelyohjelmista. Siinä on paljon ominaisuuksia ja sillä voi tehdä lähes kaiken mitä Adoben Photoshopillakin. Vanhoja tietokoneita käyttävät suosivat GIMPiä, koska se vie todella vähän muistia. GIMP on käyttäjäystävällisempi sen muokattavuuden takia.

SVG-edit on ilmainen avoimen lähdekoodin vektorigrafiikkaohjelma. Se julkaistiin vuonna 2009. Se toimii verkkoselaimessa, joten erillistä sovellusta ei tarvitse. Se on hyvä ilmainen vaihtoehto Illustratorille. Se on helppokäyttöinen, siinä on paljon toimintoja ja sillä voi kääntää svg tiedostot kuvatiedostoiksi. (Financesonline.com)

4 2D-pelin graafiset elementit

4.1 Spritet

Spritet ovat kaksiulotteisia kuvia, jotka koostuvat yhdestä tai useammasta kuvasta. joita käytetään erityisesti 2D-peleissä. Sprite Sheetsien avulla voi tehdä animaatioita. Sprite Sheet yhdistää monta spriteä yhdeksi kuva tiedostoksi. Tämä lisää suorituskykyä huomattavasti, koska se vähentää piirrettyjen objektien määrää näytöllä. (Sobolev 2020)



Kuva 12. SpriteSheet (openclipart.com/isaiah658, 2016)

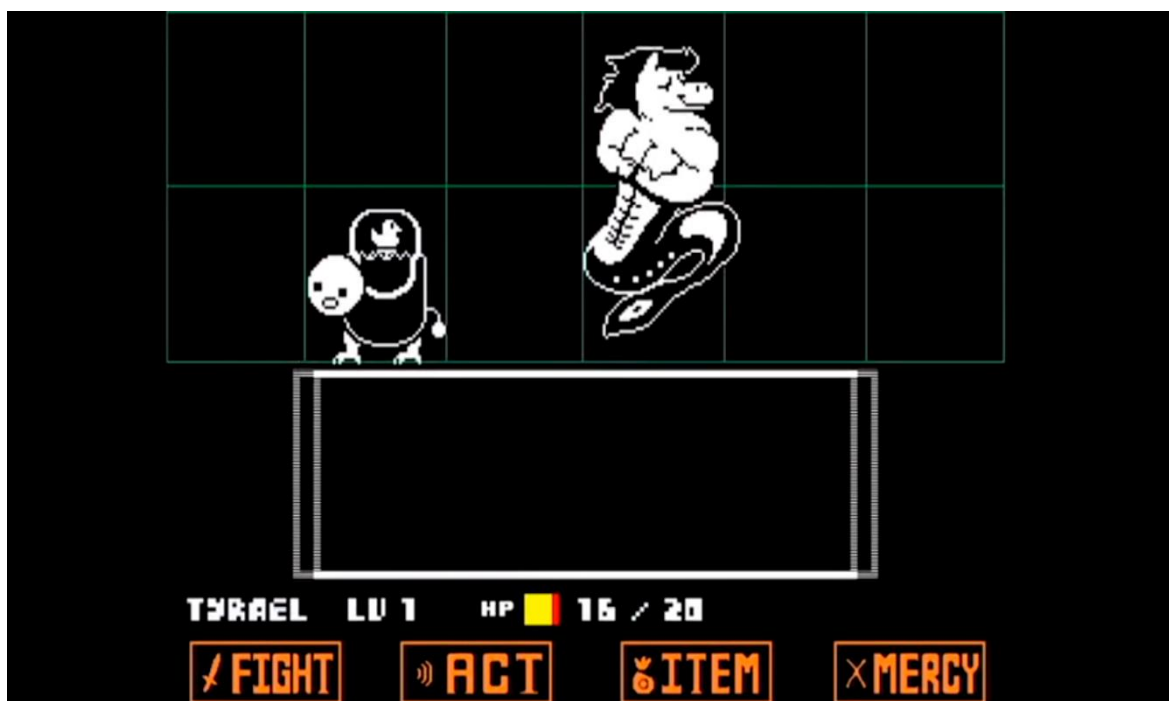
Kuvassa 12 näkyy sprite sheet, jolla hahmon saa esimerkiksi kävelemään. Spriteilla ei ole erillistä valaistusta. Spritejä voi tehdä, joko käyttäen täysin tietokoneelle saatavia sovelluksia tai sitten ne voi piirtää käsin ja sen jälkeen skannata kuva tietokoneelle digitaaliseen muotoon.

1980 luvulla ja vielä 1990 luvullakin, spritet olivat vakio tapa tehdä grafiikkaa videopeleihin. Nykyään spritet peleissä ei ole niin yleisiä, koska suurin osa peleistä käyttää 3D grafiikkaa,

mutta niitä käytetään paljon muihin asioihin esimerkiksi näppäimiin, symboleihin ja muihin käyttöliittymäelementteihin sovelluksissa. (Techterms.com 2012)

4.1.1 Käyttöliittymäsprite

Käyttöliittymä mahdollistaa pelaajan ja pelin välisen vuorovaikutuksen. Täydellistä käyttöliittymää ei voida määritellä, koska kaikilla on siihen oma näkemyksensä. Hyvä käyttöliittymä on yleensä selkeä ja se antaa kaiken tarvittavan informaation pelin aikana. Pelin käyttöliittymän pitäisi olla yhtenäinen eli kaikkien niihin kuuluvien spritejen kuuluu olla samalla tyyllillä tehtyjä, ettei siitä tule liian sekavan näköinen.

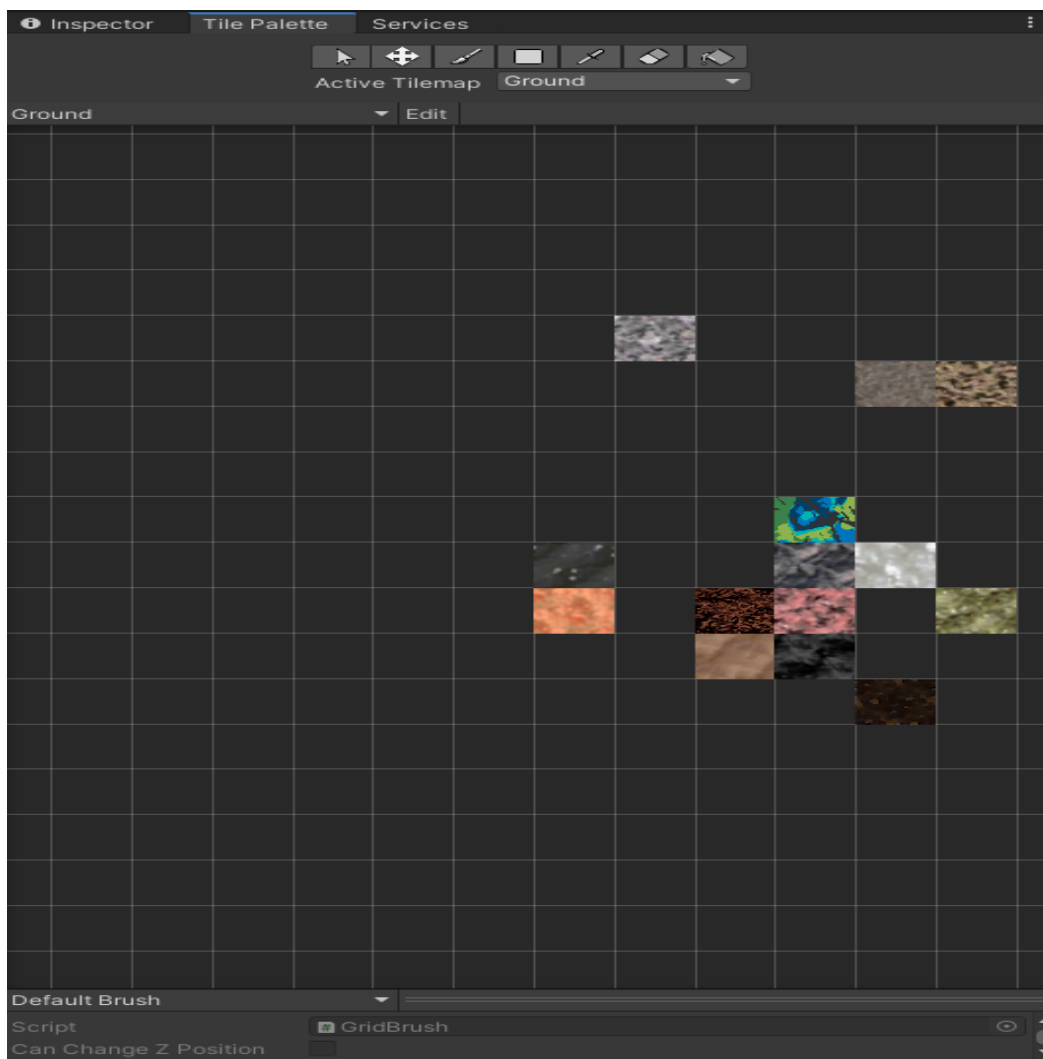


Kuva 13. Selkeä käyttöliittymä (YouTube, 2015)

Kuvassa 13 näkyy Undertale pelin käyttöliittymä, jossa näppäimet ovat tehty samalla tyyllillä, sekä ne ovat selkeät. Pelaaja ymmärtää heti mitä mikäkin näppäin tekee.

4.1.2 Tason spritet

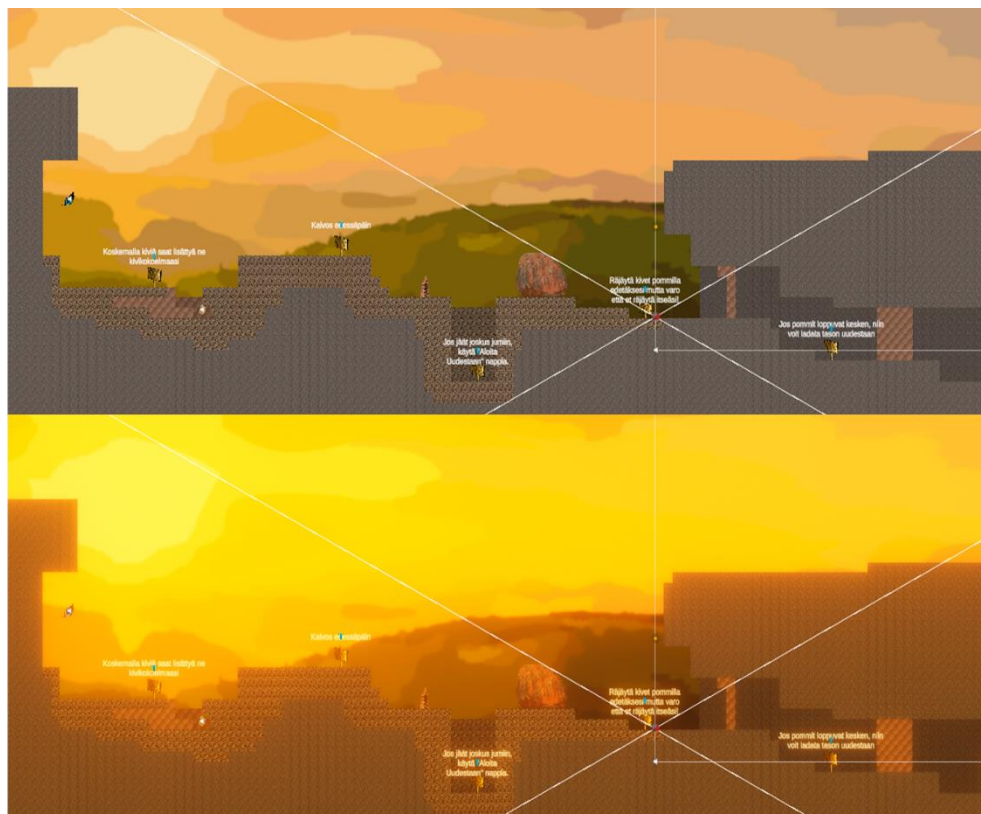
Unityssä 2D-pelin tasojen tekoon käytetään yleensä Tilemap työkalua, koska sillä on helppoa luoda tasoja ja sillä voi luoda välittömästi prototyyppejä, joita pääsee testaamaan heti editorissa. Tämä nopeuttaa pelin kehitystä sillä aikaa ei kulu niin paljoa tasojen testaamiseen. Tilemap koostuu level gridistä ja muista komponenteista, jotka toimivat yhdessä. Järjestelmällä pystyy maalata tiiliä gridiin ja hallita niiden toimintaa. (learn.unity.com 2020) Kuvassa 14 näkyy Tilepalette, jossa tilemappeja käytetään.



Kuva 14. Tilemap

4.2 Muut elementit

2D-peleihin pystyy spritejen lisäksi tehdä peliin ulkonäköä parantavia asioita käyttäen partikkelijärjestelmiä tai jälkikäsitteilytyökaluja. Partikkelijärjestelmä simuloi ja renderöi montaa pientä kuvaa tai mallia. Jokainen partikkeli järjestelmässä esittää yksittäistä graafista elementtiä efektissä. Partikkelijärjestelmät ovat hyödyllisiä, kun halutaan luoda tulta, savua tai nesteitä, koska spriteilla on vaikeaa kuvata tällaisia objekteja. (docs.unity3d.com 2020)



Kuva 15. Jälkikäsittely

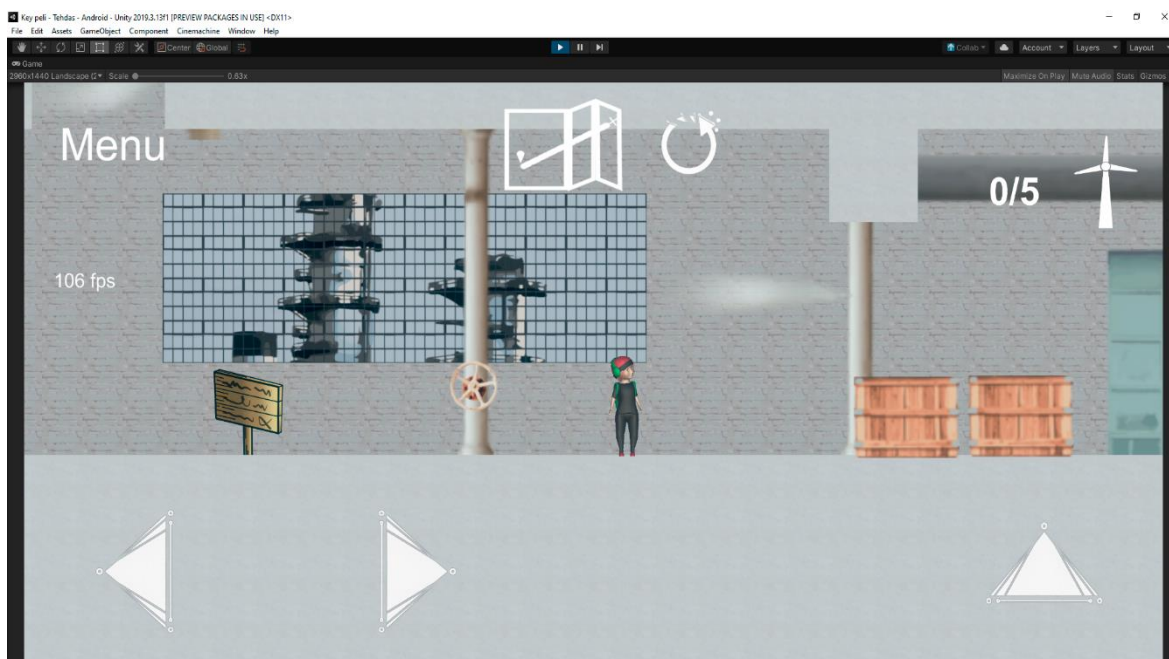
Jälkikäsittely on prosessi, jossa asetetaan filttareita ja efektejä peliin. Jälkikäsittelytyökaluja käytetään sen takia, että se parantaa pelin ulkoasua. Jälkikäsittelyllä voi muuttaa pelin ulkonäköä todella paljon. Jälkikäsittelytyökaluilla saa varjoista ja pinnoista realistisemman näköisiä. (Cirne 2019) Valoisuuden säädöillä saa peliin täysin eri tunnelman. Kuvassa 14 näkyy, miten jälkikäsittelytyökaluilla on luotu auringonlaskun tunnelma peliin.

5 Case – Key-peli

5.1 Pelin tavoite

KEY-peli eli Kivi, Energia ja Ympäristö peli on LUMA Suomen hanke, jonka tarkoitus on lisätä koululaisten tietämystä kivistä energiasta sekä ympäristöstä. Pelissä saa tietoa tiettyistä kivistä, sekä mihin niitä käytetään. Siinä myös kerätään roskaa sekä palkitaan oikein kierrätyksestä. Pelissä on myös videoita, jotka antavat tärkeää tietoa energiaan liittyvistä asioista. Asioiden oppiminen varmistetaan pelin aikana tulevilla kyselyillä.

Peliin oli tarkoitus tehdä viisi eri tasoa, joista valmiiksi saatiin neljä. Peliä oli kehitetty aikaisemmin toisen ryhmän toimesta. Peliin oli tehty kaksi kaivostasoa, joista kumpaakaan ei loppujen lopuksi käytetty. Valmiina oli myös pelaajahahmo, sekä musiikit.



Kuva 16. Pelaajahahmo

Peliä kehitettiin 2D-ympäristössä, mutta pelaajahahmo oli 3D-malli, joka näkyy kuvan 16 keskellä. Pelin teossa käytettiin Unityä, koska pelin ensimmäiset versiot olivat tehty sillä, sekä siitä oli jo aikaisempaa kokemusta. Pelin kehityksessä käytettiin Unityn Collaborate toimintoa, jonka avulla pystyimme käyttämään samaa projektitiedostoa. Spritejen luontiin käytettiin Adobe Photoshopia sekä Illustratoria, koska niitä on aikaisemmin jo opittu käyttämään sekä ne ovat helppoja ja monipuolisia.

5.2 Tasosuunnittelu

Tasojen suunnittelu aloitettiin miettimällä minkä tyyllisiä tasoja luotaisiin. Päätimme luoda pelien tasoista metroidvania tyyllisiä. Tasoissa on paljon hyppimistä, sekä kahdessa tasossa on oikoteitä. Tasojen ideana on, että tasoissa on tietty keskikohta, josta lähtee reittejä eri suuntiin.

Tasoissa on kerättäviä esineitä, joilla suoritetaan teemaan sopiva tehtävä. Kaivostasossa tehtävänä on kerätä kiviä. Mökkitasossa tarkoituksena oli kerätä kasetteja, joilla pystyi katsomaan videoita televisioista. Pelissä käytetyt videot saatiin STEK:iltä. Metsätasossa kerätään roskia, jotka pitää kierrättää oikealla tavalla. Tehdastasossa kerätään tuulimyllyn osia, jonka jälkeen tyylimylly koottiin ja tällä tavalla saatiin koneet toimiviksi, joilla pystyi rakentamaan kaivos tasosta kerätyillä kivillä esineitä.

Joka tason ulkoasu on erilainen, sekä tasot tuovat mukanaan uusia mekaniikkoja, jotta pelaajan mielenkiinto peliin pysyisi. Tasot suunniteltiin myös haastaviksi, jotta pelaaja ei voisi pelata peliä liian nopeasti läpi. Haasteellisuuden tarkoituksena on myös, että pelaaja ei kylästyisi peliin niin nopeasti.

Pelin ensimmäinen taso eli kaivostaso suunniteltiin toimimaan pelaajalle perusmekaniikkojen perehdyttäjänä. Kaivo on myös pelin ainut täysin lineaarinen taso. Pelin muihin tasoihin suunniteltiin keskikohta, johon pääsee tasojen joka alueesta. Joka alueessa on omat haastavat kohdat, joiden jälkeen on keräiltävä esine. Kun pelaaja epäonnistuu esimerkiksi tippumalla tai räjäyttämällä itsensä pommilla, joutuu pelaaja aloittamaan tason alusta, sekä keräämään esineet uudestaan.

Peliin tehtiin tasojen pääsyä vaikeuttavia esineitä, joihin osuessaan pelaaja joutuu aloittamaan tason alusta. Tappavat esineet suunniteltiin tasojen teemaan sopiviksi.

Peliin lisättiin oikoteitä, joiden tarkoituksena on tehdä tasoista helpommin läpi päästäviä. Oikotiet sijoitettiin alueiden alkuun ja loppuun. Kun pelaaja pääsee alueen läpi, hän pystyy avata oikotien, jonka jälkeen, jos epäonnistuttuaan joutuu aloittamaan tason alusta, ei hänen tarvitse enää mennä aluetta uudestaan, vaan pystyy kulkea alueen päähän oikotien kautta. Oikoteitä on pelin kahdessa viimeisessä tasossa.

5.3 Toteutus

Pelissä on neljä tasoa, sekä kaksi kyselytasoa. Tasoissa on jokaisessa oma teema sekä teemaan sopivat spritet. Osa spriteistä on tehty valokuvia jäljittelemällä ja osa täysin käsin. Tason esineet ja koristeet tehtiin spriteiksi ja tasot rakennettiin Unityssä tile paletta

käyttäen, koska sillä on helppoa ja nopeaa luoda tasoja. Tasoja viimeisteltiin käyttäen Unityn post-process työkaluja.



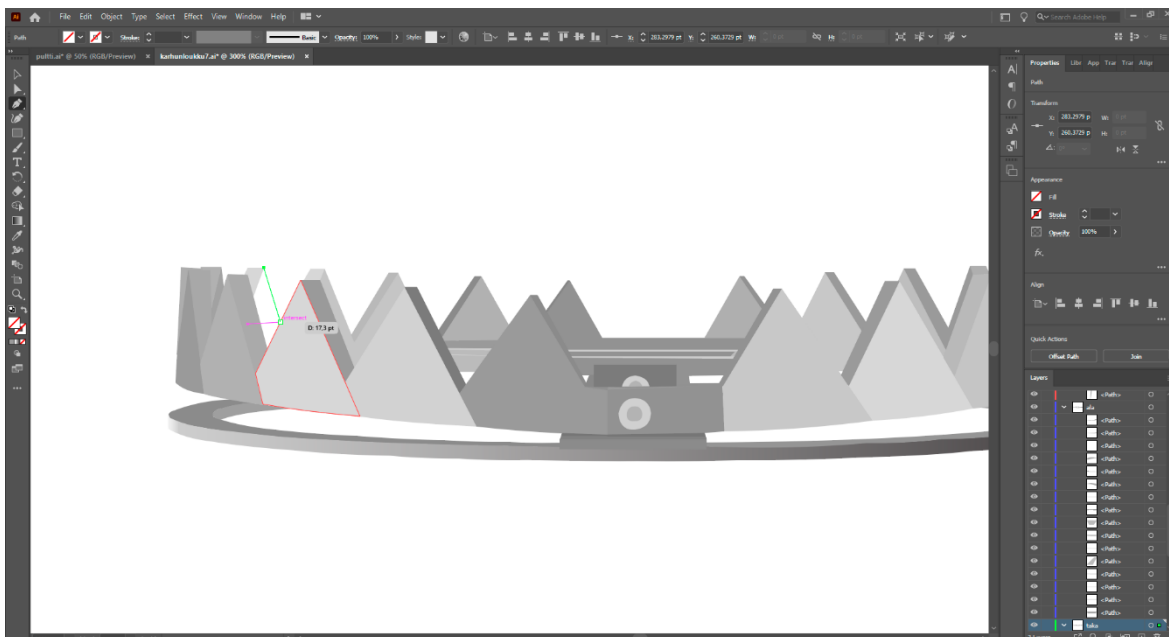
Kuva 17. Spritejä tasosta

Kuvassa 15 näkyy pelin jokaisesta tasosta olevia spritejä. Ylemmän rivin vasemmassa lohossa näkyy kaivostason savikivitiili sekä keräiltävä kvartsikivi. Ylemmän rivin oikeassa lohossa on mökkitason tv, josta pelaaja näkee videoita kerättyään kasetteja, sekä tuhottava ja tippuva pahlilaatikkotiili. Alarivin vasemmassa lohossa on metsätason maatiili sekä kerättävä ja kierrätettävä sanomalehtiroska. Alarivin oikeassa lohossa näkyy tehdastason kone, jolla pystyy valmistamaan korun, sekä tehdastason taustatiili.

5.3.1 Spritejen teko

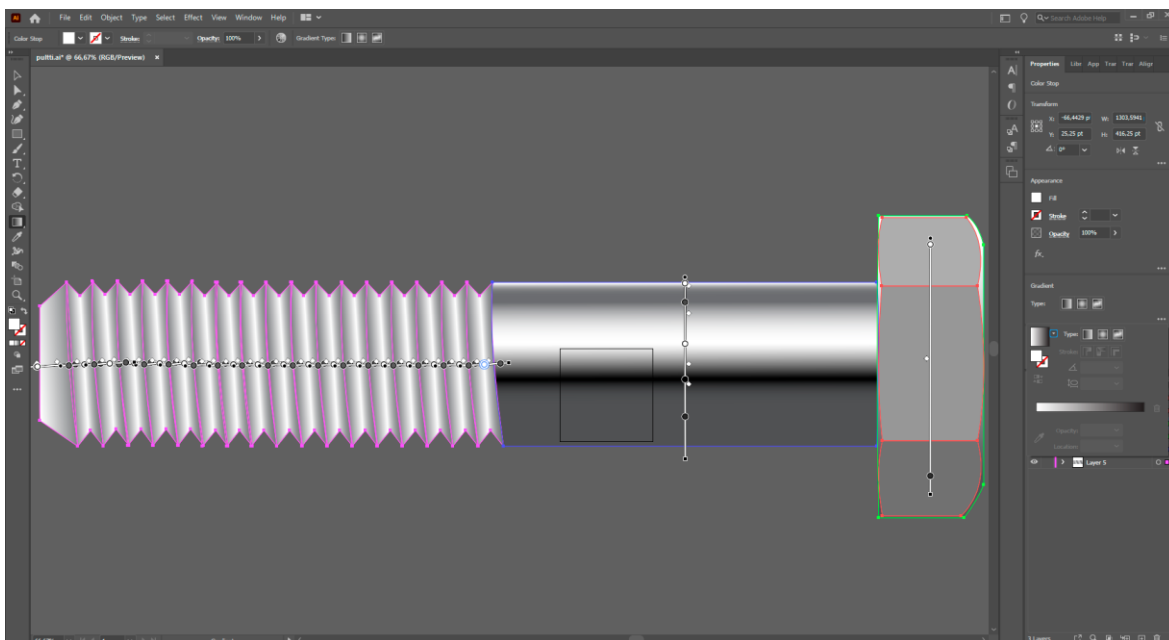
Spritet tehtiin Adobe Illustratoria, sekä Photoshoppia käyttäen. Käytettävät, kerättävät esineet ja osa koriste-esineistä tehtiin täysin manuaalisesti Illustratorin pen toolia käyttämällä. Pelissä joka tasossa oli oma teema, joten piti miettiä mitkä esineet sopivat teemaan.

Spritejen teko aloitetaan usein tyhjästä artboardista, johon lähdetään tekemään kuvioita. Jos haluaa ottaa mallia kuvasta illustratoriin voi myös asettaa kuvan, ja sen jälkeen voi lähteä tekemään kuvan mukaan käyttämällä esimerkiksi pen toolia.



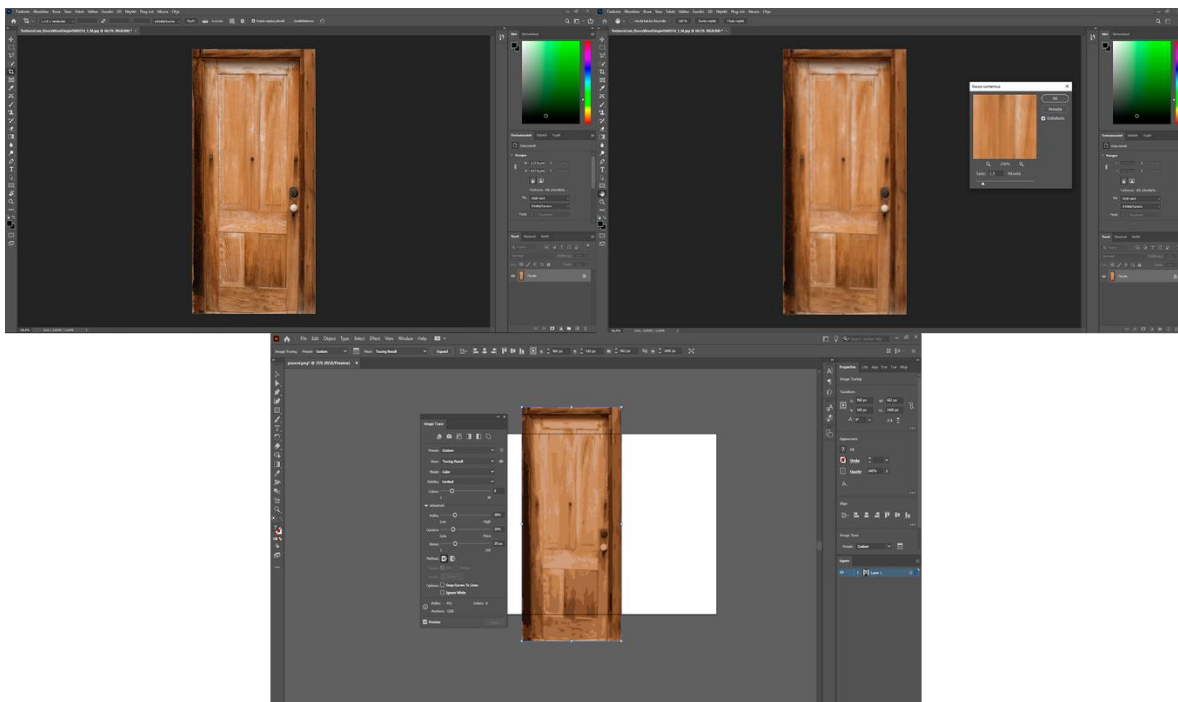
Kuva 18. Pentool

Eniten käytetty työkalu spritejen teossa oli pen tool, jolla pystyy piirtämään viivoja sekä käyriä. Kuvassa 16 on metsätason karhunrauta, joka on tehty täysin pen toolilla piirtämällä. Sillä luodaan muotoja luomalla kuvioita anchor pisteillä. Kun kuvio on luotu sille voi valita haluamansa värin.



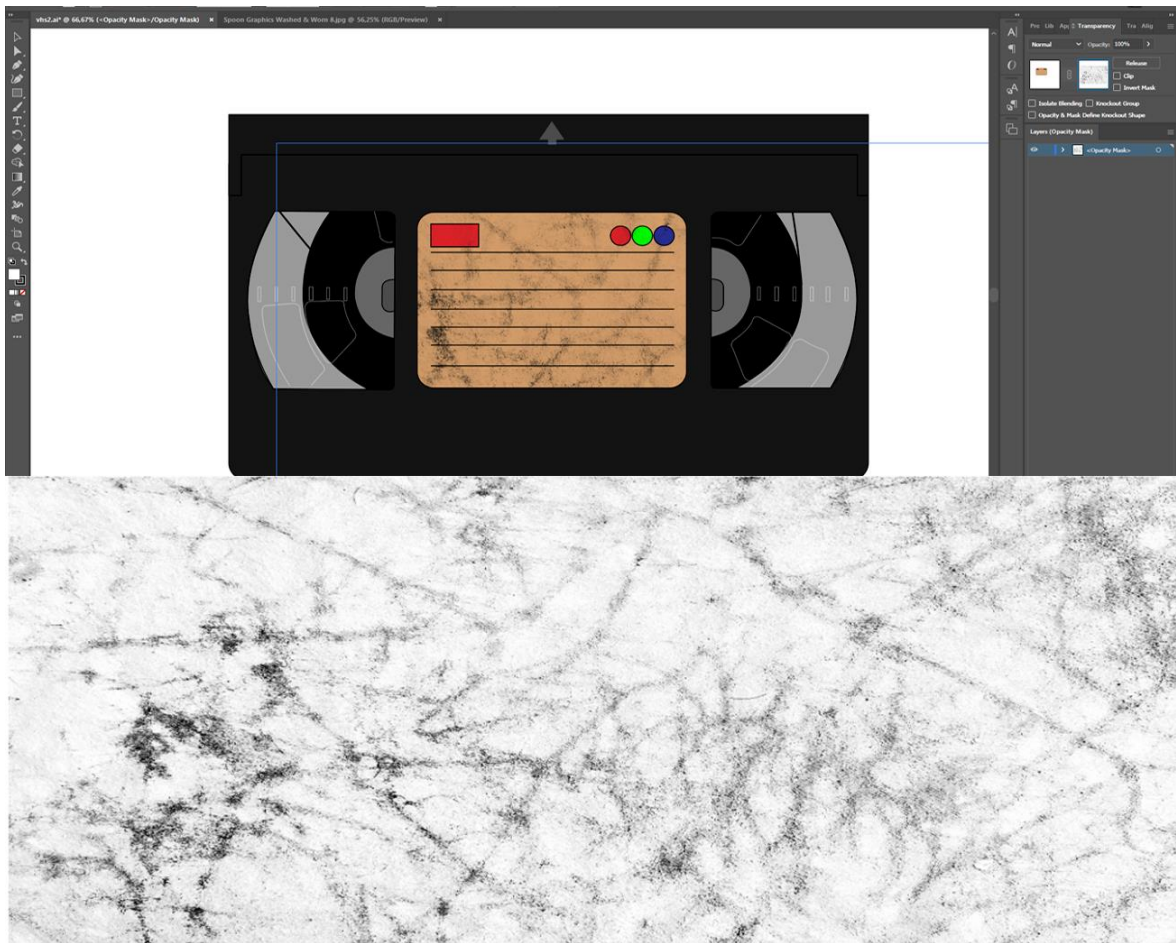
Kuva 19. Gradient tool

Yksi spritejen teossa käytetyimmistä työkaluista oli gradient työkalu, jolla vektoreihin saa luotua helposti esimerkiksi hienon heijastuksen. Kuvassa 17 on yksi metsätason kerättävistä ja kierrätettävistä roskista, johon on käytetty gradient työkalua luomaan pulttiin kiiltoa.



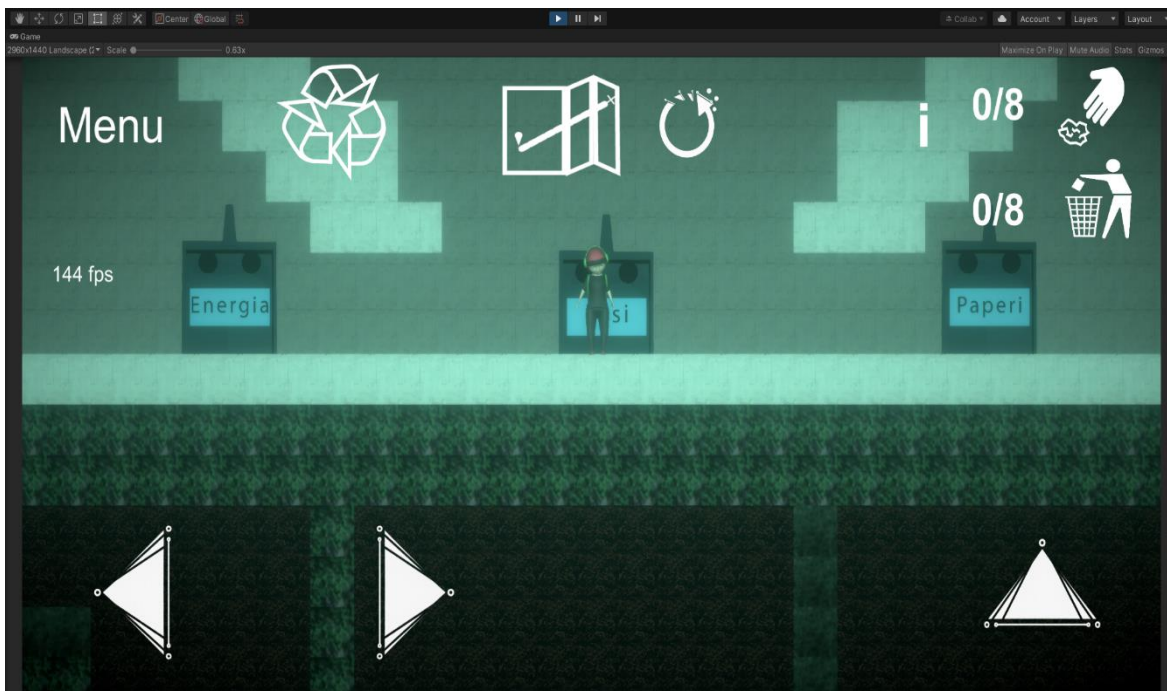
Kuva 20. Image Trace

Sarjakuvamaisia spritejä voi tehdä myös suoraan valokuvista käyttäen image trace työkalua, joka muuntaa bittikarttagrafiikan vektorigrafiikaksi. Kuvassa 18 ylärivillä näkyy, miten valokuva sumennetaan photoshopissa käyttäen gauss sumennus työkalua. Sumennus tehtiin sen takia, että image tracea tehdessä kuvasta tulisi sarjakuvamaisempi. Kuvassa 4 alhaalla näkyy image trace työkalun käyttö, jossa voi säätää esimerkiksi paljon eri värejä kuvaan haluaa sekä paljon kulmia ja polkuja haluaa. Image trace jättää kuviin valkoiset reunat, jonka takia ne poistettiin photoshopissa joko pyyhi työkalulla tai sivellin työkalulla maalamalla väriä päälle. Reunat on myös mahdollista poistaa valitsemalla valkoinen alue taikasauvatyökalulla, mutta tällöin reunasta ei tule tasainen.



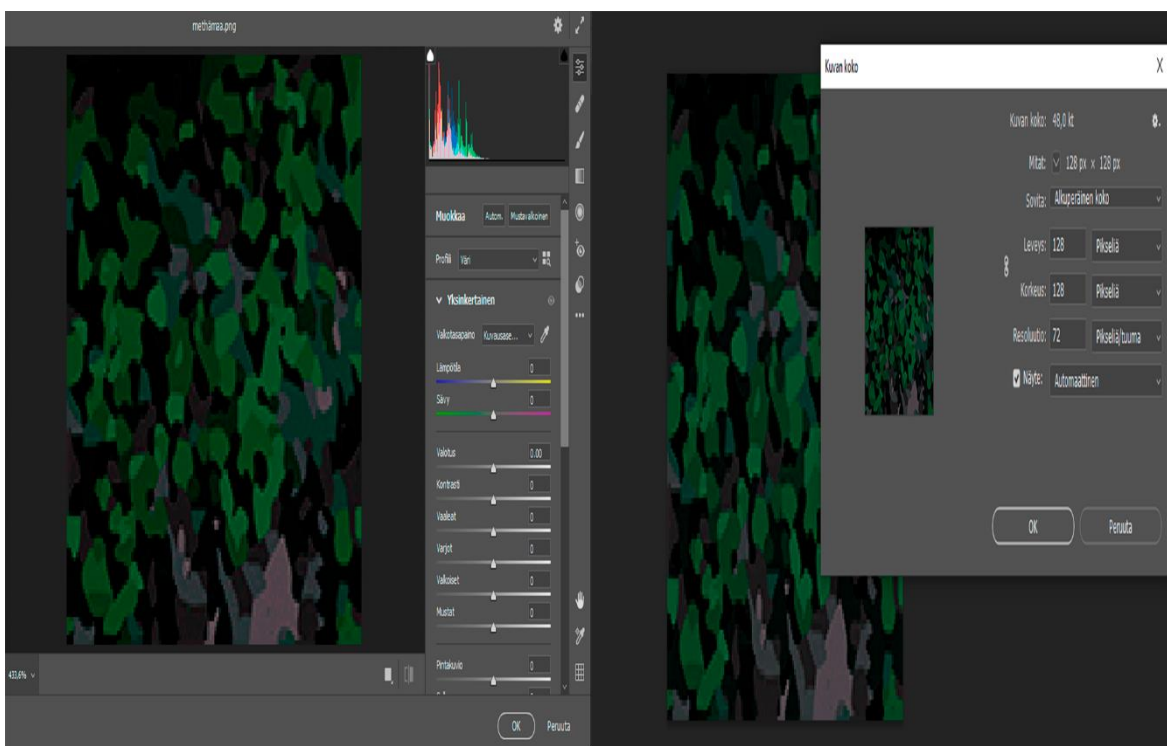
Kuva 21. Tekstuurin laitto

Illustratorissa pystyy laittamaan myös tekstureja kuviin. Kuvassa 19 olevan kasetin keskellä on käytetty kuvan alaosassa näkyvää tekstuuria, joka laittaa kasetin näyttämään kulluneelta. Illustratorissa tekstureita kuviin saa transparency työkalussa luomalla maskin. Haluttu tekstuurikuva kopioidaan maskiksi. Tekstuurikuva on usein suurempi kuin kohtaan teksturi laitetaan, tällöin teksturi siirretään kopiointin jälkeen haluttuun kohtaan.



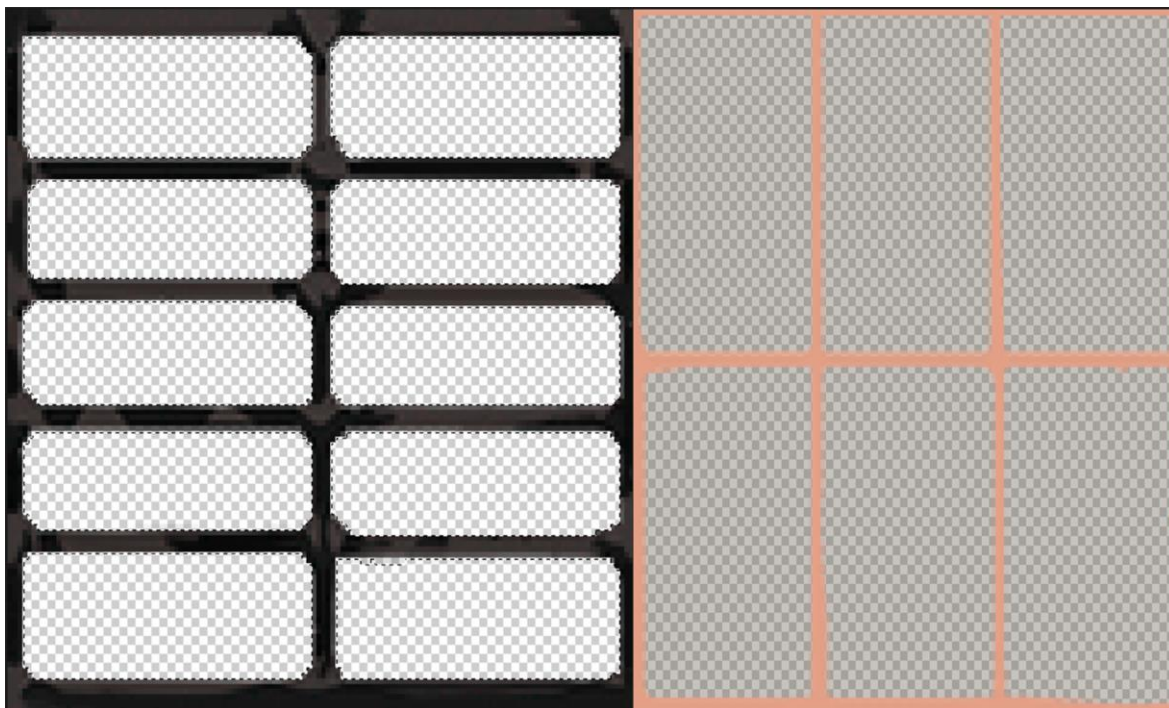
Kuva 22. Käyttöliittymäspritet

Käyttöliittymäspritejä tehdessä pitää huomioida, että koko käyttöliittymä on samalla tyylillä tehty. Pelin käyttöliittymien napit pitää olla tarpeeksi isoja, että niitä olisi helppo painaa pelitilanteessa. Pelissä kaikki on valkoisella tehtyä. Kuvassa 20 näkyy ylempänä pelin metsätason koko käyttöliittymä. Käyttöliittymä luotiin Illustratorin pen toolia käyttäen.



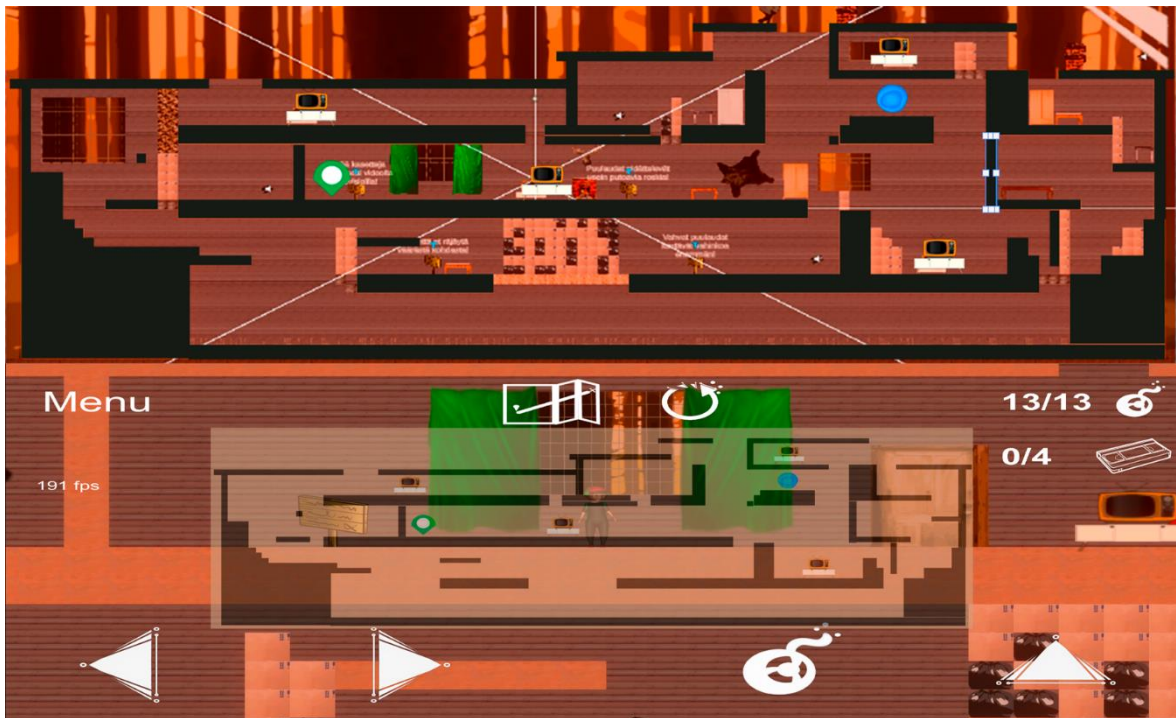
Kuva 23. Tiilien viimeistely

Tason tiilien tekoon käytettiin tason teemaan sopivia valokuvia, joista Photoshopissa rajattiin pieniä kohtia käyttäen crop työkalua. Rajauksen jälkeen kuva vietiin Illustratoriin, jossa kuva tyylliteltiin vastaamaan pelin muuta grafiikkaa. Kuvassa 21 vasemmalla näkyy Camera Raw-suodin, jota käytettiin, kun tyyllittely oli valmis, ja kuva tuotiin takaisin Photoshopiin, jossa kuvan värejä muunnettiin tason estetiikkaan sopiviksi käyttäen Camera Raw-suodinta. Kuvassa 7 oikealla puolella kuvan koon muokkaus, jota käytettiin, koska Unityssä pelin Level gridin tiilikoko oli 128 x 128 oli kuvan koko muutettava Photoshopissa siihen kokoon.



Kuva 24. Läpinäkyvät tiilet

Pelissä olevat ikkuna ja ritilä tiilet viimeisteltiin Illustratorin tyyllittelyn jälkeen Photoshopissa. Kuvassa 22 vasemmalla näkyy, miten ritilä tiilestä on poistettu kohtia läpinäkyvyyden luomiseksi. Kuvassa 22 oikealla taas on ikkunatiili, joka on muokattu lasimaiseksi vähentämällä kohtien peittävyttä.

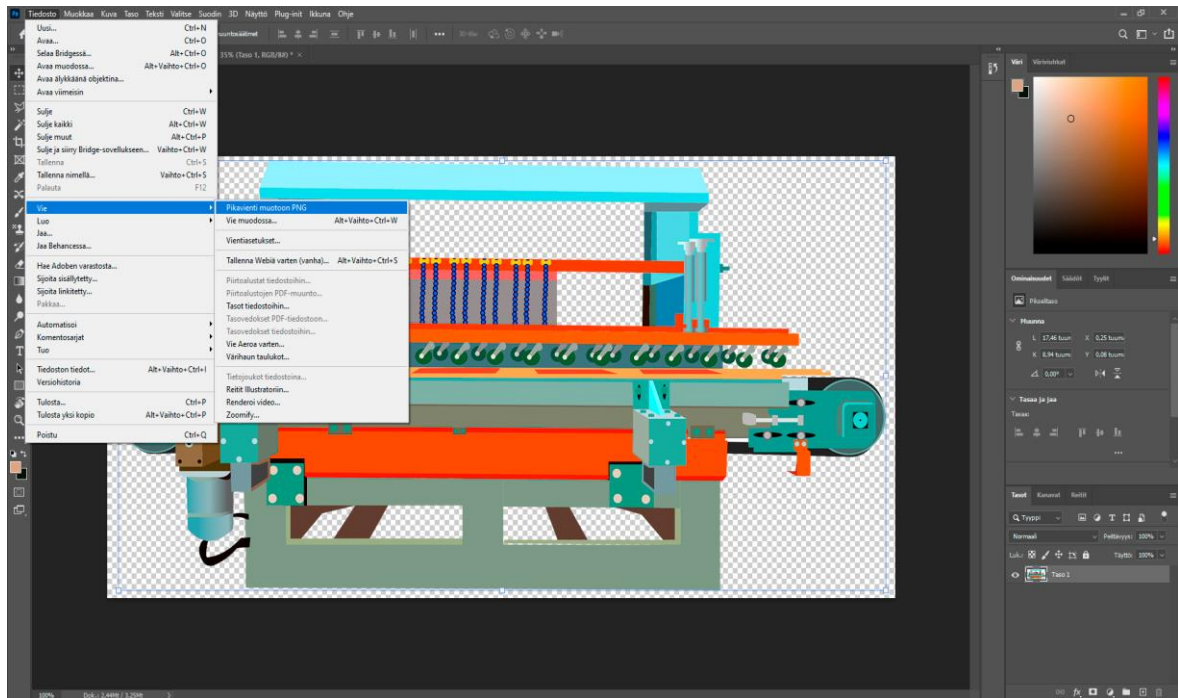


Kuva 25. Kartan luonti

Pelin jokaisessa tasossa on mahdollisuus katsoa karttaa. Kartan luomiseen käytettiin Unity Editorista otettua kuvakaappausta, jonka pohjalta tehtiin tason rajat käyttäen Photoshopin suorakulmiotyökalua. Kuvassa 23 ylhäällä näkyy kartan luontivaihe ja kuvassa alhaalla näkyy, miten kartta näkyy KEY-pelissä. Kartasta selviää tason aloituskohta, seuraavaan tasoon vievän portaalin sijainti, sekä tärkeimmät tason suorittamiseen liittyvät kohdat ja esineet. Kartan läpinäkyvyyttä lisättiin myös, jotta pelaaja näkisi itsensä.

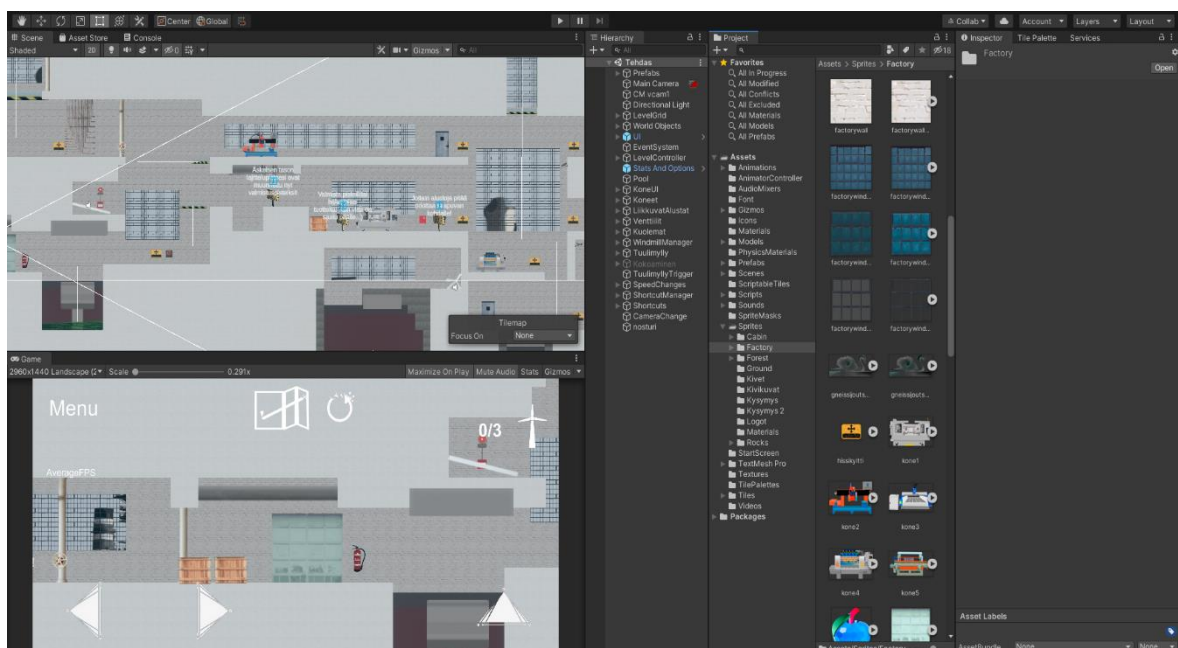
5.3.2 Spritejen vienti Unityyn

Kun spritet saatiin valmiiksi, ne piti viedä Unityyn, jossa niitä päästiin käyttämään. Spriteihin käytetyt kuvatiedostot tehtiin niin pieneksi kuin mahdollista, jotta ne veisivät mahdollisimman vähän tallennustilaa.



Kuva 26. PNG:n teko

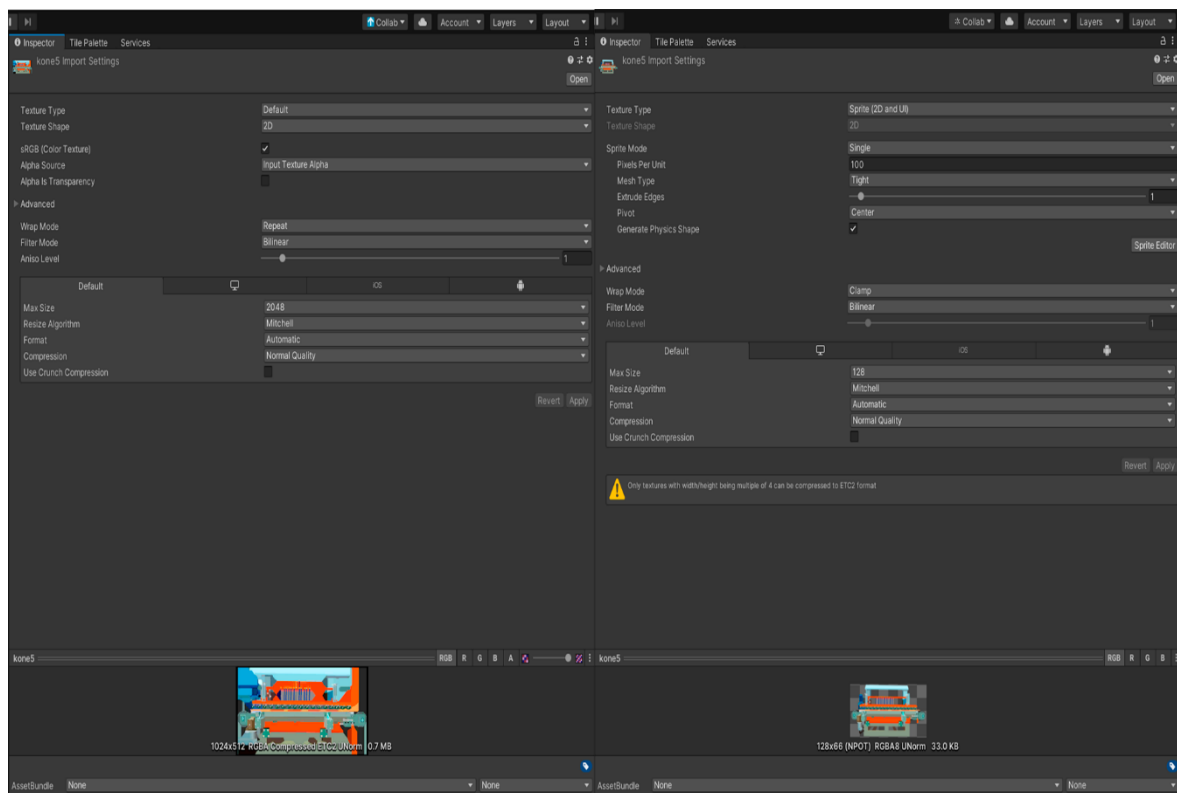
Kaikki spriteihin käytetyt kuvatiedostot tehtiin PNG:ksi. PNG:itä käytettiin, koska ne ovat hyvälaatuisia ja PNG:t tukevat alfakanavia, jotka mahdollistaa läpinäkyvyyden kuvissa. Kuvassa 24 näkyy Photoshopin pikavienti muotoon PNG, joka on helppo ja nopea tapa saada kuva PNG muotoon.



Kuva 27. Spritejen siirto Unityyn

Unityssä spriteille tehtiin oma kansio, jonka sisälle tehtiin omat kansiot joka tason spriteille. Tällä tavoin oli helppo löytää haluamansa spritet. Myös tiilille tehtiin kansionsa. Kuvassa 25

näky Unityssä avoinna oleva tehdas tasolle tehty sprite kansio, johon siirrettiin valmiina olevat spritet raahaamalla. Jos spritejä tarvitsi muuttaa jälkikäteen, se onnistui helposti korvaamalla tiedosto pelin kansiossa. Selkeä kansiorakenne helpotti oikean paikan löytämistä.

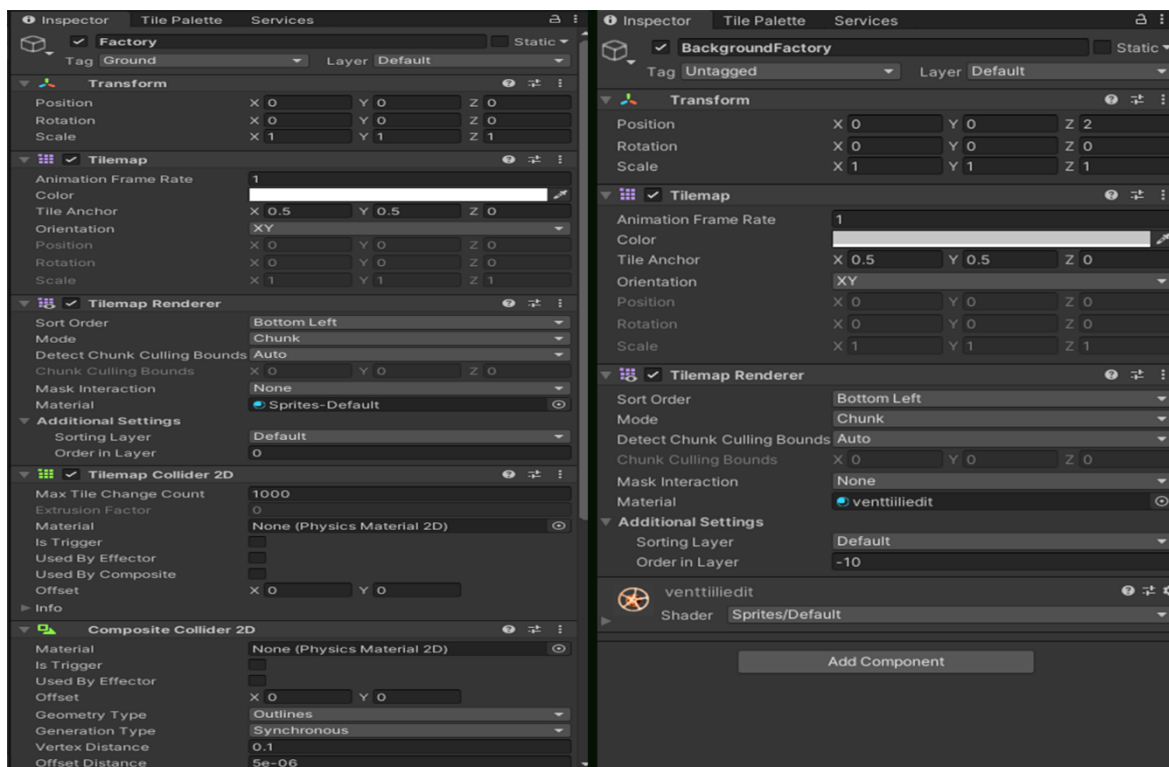


Kuva 28. Kuvan muuttaminen spriteksi

Kun kuva oli siirretty oikeaan kansioon, tämän jälkeen kuvan tekstuurityyppi oli muutettava spriteksi Unityn Inspectorissa. Tämä tehdään sen takia, että default textuuria ei voi käyttää 2D-tasossa. Kuvien maksimikokoja oli myös pienennettävä, jotta pelin suorituskyky parani. Maksimikokoa säädettiin jokaisessa spritessä erikseen, koska jotkin spritet näyttivät liian huonolaatuisilta, jos kokoa säädettiin liian pieneksi. Tiilien kohdalla piti myös muuttaa pixels per unit kokoon 128, koska se oli level gridin koko. Kuvassa 26 näkyy Unityn Inspector, jossa kuvien tekstuurityyppien ja kokojen säätäminen onnistui.

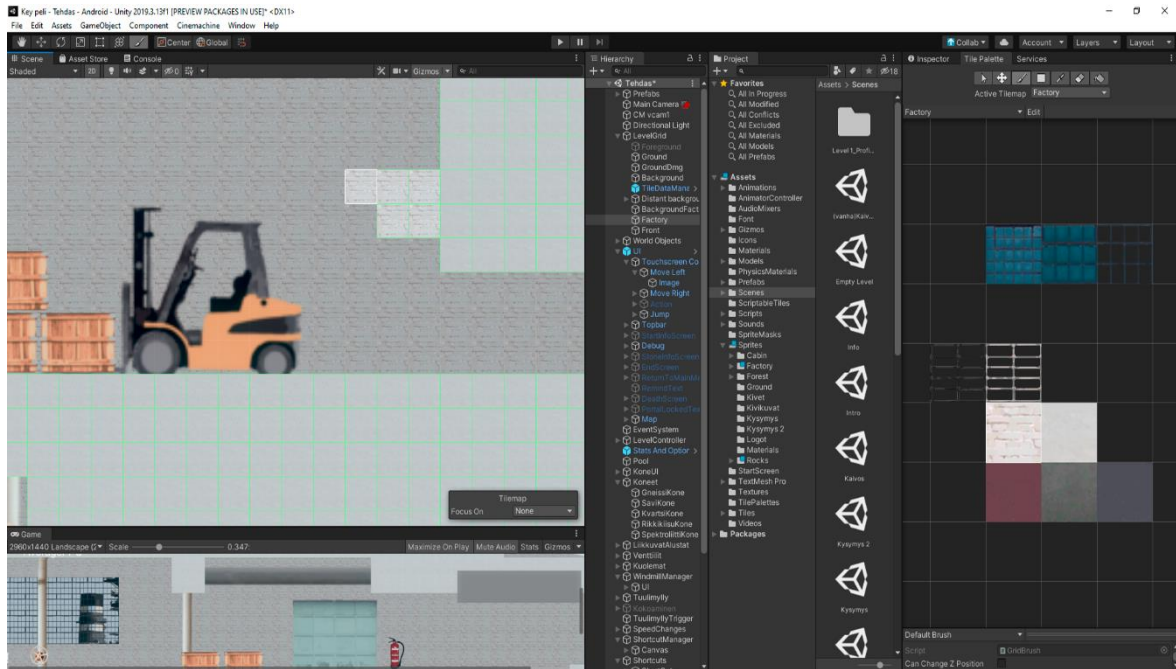
5.3.3 Tasojen luonti

Tasot luotiin käyttäen Unityn Tile Paletta, jonka työkalujen avulla pystyy maalaamaan tile-mappeja. Tile Paletten avulla tasoja pystyi helposti testaamaan ja muuttelamaan jos jokin ei toiminut suunnitellusti. Pelin jokaisessa tasossa on oma Tile Palette, johon laitettiin siihen tasoon sopivat tiilet.



Kuva 29. Tilemappien säätö

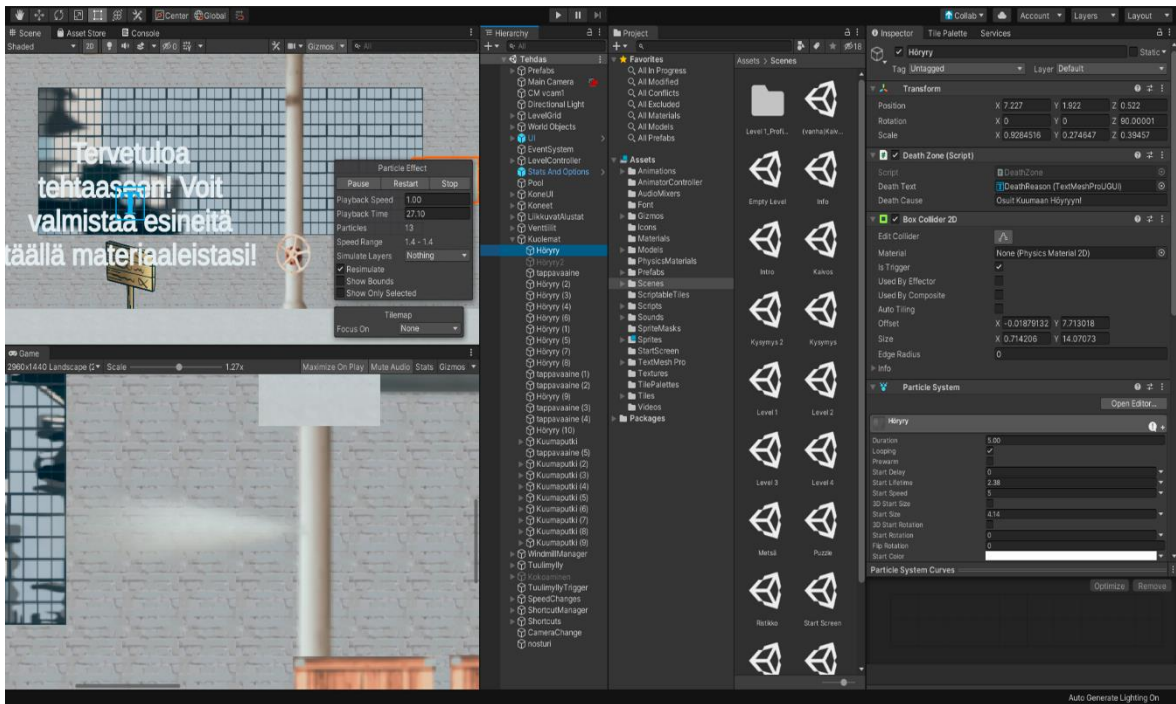
Joka tasossa käytettiin kahta eri tilemappia, joista toinen oli tason pää tilemap, eli se, jonka päällä pelaaja kulki. Toinen tilemap toimi taas tason taustana tummempana. Tausta tilemap luotiin vaihtamalla taso taaemmaksi. Tämä onnistui säätämällä Order in Layer kohtaa inspectorissa pienemmäksi. Jotta tilemapin tiilet saatiin näyttämään siltä, että ne olisivat taustalla, säädimme niiden värejä tummemmiksi inspectorin värisäätö kohdasta. Pää tilemapeissa oli myös törmäystarkastelu. Kuvassa 27 näkyvät säädöt pää sekä tausta tilemapeissa.



Kuva 30. Tile Palette

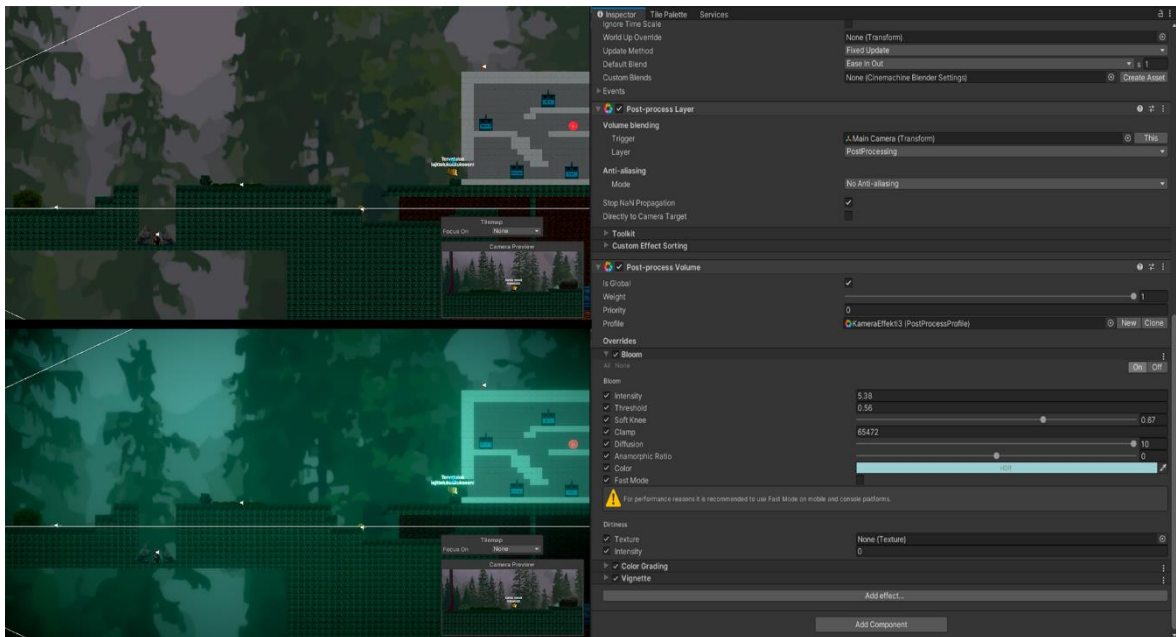
Tile Paletessa on 7 eri työkaluja, joista käytetyimmät olivat. maalaustyökalu, laatikkotäyttötyökalu, sekä pyyhekumi työkalu. Maalaustyökalun avulla tile paletesta valittua tiiltä pystyy maalamaan aktiivisena olevaan tilemappiin. Laatikkotäyttötyökalulla pystyy aktiiviseen tilemappiin maalaamaan isompaa aluetta kerralla. Pyyhekumityökalulla pystyy poistamaan tiiliä tilemapeista. Kuvassa 28 näkyy oikealla Tile palette, josta valitaan haluttu työkalu, sekä tiili, jolla lähdetään tasoja maalaamaan. Kuvassa vasemmalla näkyy kuva tehdastasosta, josta näkyy sävyero pää sekä tausta tilemapin välillä. Tausta tilemappiin maalatut tiilet ovat tummempia kuin pää tilemappiin maalatut tiilet.

Pelin esineet eivät käytä tilepalettea, vaan ne ovat normaaleja spritejä, jotka on aseteltu tasoon. Myös esineiden värejä muutettiin tarvittaessa inspectorista. Spritejen värejä jouduttiin säätämään tummemmiksi metsätasossa, koska metsätason teema oli sateinen metsä, jolloin spritejenkin kuului olla tummia. Spritejä tehtiin kahden laisia, niitä, joihin laitettiin törmäystarkastelu, eli pelaaja pystyi seisomaan ja hyppimään niiden päällä, sekä taustalle kuuluvia koriste spritejä, joiden kanssa pelaaja ei pystynyt tekemään mitään. Koristespritejen tarkoituksena oli tehdä tasoista kiinnostavamman näköisiä.



Kuva 31. Partikkelijärjestelmä

Tehdastason yhdeksi pelaajan tappavaksi esineeksi tehtiin putkista vuotavaa kuumaa höyryä, joka toteutettiin partikkelijärjestelmää käyttämällä. Kuvassa 29 näkyy höyry, johon osuessaan pelaaja joutuu aloittamaan tason alusta. Partikkeliefektin materiaalin kokoa, kestoja, väriä ja paljon muuta pystyy säätämään inspectorissa, joka näkyy kuvan oikealla puolella.



Kuva 32. Jälkikäsitteilytyökalut

Kun tasot oli saatu halutun näköisiksi rakenteeltaan, sekä esineiltään pystyi tasoista tehdä paremman näköisiä käyttäen Unityn jälkikäsittelytyökaluja. Metsätasoon oli alun perin tarkoituksena tehdä vesisadetta partikkeli efektiä käyttämällä, mutta siitä jouduttiin luopumaan suorituskykyä parantaaksemme. Tasoon saatiin kuitenkin usvainen ja sumuinen tunnelma jälkikäsittelytyökalujen avulla. Kuvan 30 vasemmalla puolella näkyy ero, kun jälkikäsittelytyökaluja on käytetty. Ylemmässä kuvassa jälkikäsittelyä ei ole tehty ja alemmassa on. Kuvan oikealla näkyy inspector, josta jälkikäsittelytyökaluja säädetään. Eniten säädimme hehkeyttä, värikorjausta, sekä vinjettiä.

6 Yhteenveto

Työssä selviteltiin mitä tasosuunnittelu on, mitkä ovat siihen liittyviä tärkeitä asioita, sekä tasosuunnittelun historiaa. Tärkeimpänä opittuna asiana on se, ettei tasosuunnittelulle ole oikeaa tapaa, sillä jokaisella on omat tapansa ja pelaajat pitävät eri tyyleistä.

Tavoitteena oli tehdä 2D-mobiilipeli, joka koostuisi viidestä eri tasosta. Joka tasossa piti olla oma teemansa. Tasojen teemojen piti liittyä kiviin, energiaan sekä ympäristöön. Pelin oli tarkoitettu peliksi, jota pelaamalla opiskelijat oppisivat asioita pelin teemoihin liittyvistä asioista.

Pelistä saatiin valmiiksi testaus versio, mutta siitä jäi puuttumaan viimeinen taso kokonaan. Viimeinen taso oli suunniteltu olevan kotitaso, joka olisi ollut lyhyt kertaus jokaisesta tasosta, jonka lopussa olisi kerättyjä, sekä valmistettuja esineitä laitettu kodin hyllylle. Peliä testanneet antoivat palautetta, joka oli pääosin hyvää. Testaajat sanoivat peliä koukuttavaksi. Peli tehtiin android puhelimille, mutta palautteessa haluttiin myös ios versiota, sillä peliä ei voitaisi käyttää opetustarkoitukseen, jos kaikki eivät voisi peliä pelata. Peliin jäi joidakin ongelmia, joita emme saaneet ratkaistua, sillä aika loppui kesken.

Peliä kehittäessä eniten ongelmia tuotti, uusien tasojen keksiminen, sekä pelin optimointi.

Jos peliä haluttaisiin jatkokehittää, siihen voisi lisätä puuttuneen viimeisen tason, optimoida peliä sekä korjata virheitä, joita peliin jäi.

Lähteet

Admec Multimedia. 2014 Differences Between Adobe Illustrator and CorelDraw Viitattu 3.11.2020

Saatavissa <https://www.admecindia.co.in/graphic-design/differences-between-adobe-illustrator-and-coreldraw/>

Barclay, M. 2020. Blocktober 2020. Viitattu 23.10.2020

Saatavissa <http://www.mikebarclay.co.uk/category/level-design/>

Bateman, A. 2020 17 Best Photo Editing Software for Photographers (2020 Ultimate Guide) Viitattu 3.11.2020

Saatavissa <https://enviragallery.com/best-photo-editing-software-for-photographers/>

Britannica 2020 Raster graphics Viitattu 2.11.2020

Saatavissa [Raster graphics | computer science | Britannica](https://www.britannica.com/technology/raster-graphics)

Byrne, E. 2004. Game Level Design. Viitattu 22.11.2020

Chapman, N, J. 2000 Digital Multimedia Viitattu 2.11.2020

Cirne, V. 2019 Everything you need to know about Post-processing in Unity Viitattu 17.11.2020

Saatavissa <https://medium.com/@vinicius.cirne/everything-you-need-to-know-about-post-processing-in-unity-72b09534ac38>

CryEngine 2020 Licensing Viitattu 1.11.2020

Saatavissa <https://www.cryengine.com/support/view/licensing>

Dempsey, J. 2020 Luminar 4 Review – AI Features, Sky Enhancement, and Smooth Editing Viitattu 3.11.2020

Saatavissa <https://www.photoworkout.com/luminar-review/>

docs.unity3d.com 2020 Particle systems Viitattu 16.11.2020

Saatavissa <https://docs.unity3d.com/Manual/ParticleSystems.html>

Eden, M. 2020 2D Vs 3D Games: Differences, Benefits And Costs Viitattu 25.10.2020

Saatavissa <https://meliorgames.com/game-development/2d-vs-3d-games-differences-benefits-and-costs/>

Fandom.com. Fifth generation of video game consoles Viitattu 29.10.2020

Saatavissa https://ultimatepopculture.fandom.com/wiki/Fifth_generation_of_video_game_consoles#cite_note-Fifth-3

Financesonline.com SVG-edit Review Viitattu 3.11.2020

Saatavissa <https://reviews.financesonline.com/p/svg-edit/>

Gaming, N. 2017 The evolution of platform games in 9 steps Viitattu 28.10.2020

Saatavissa <https://www.redbull.com/in-en/evolution-of-platformers>

Giantbomb. 2020 Metroidvania. Viitattu 28.10.2020

Saatavissa <https://www.giantbomb.com/metroidvania/3015-2440/>

Gimp.org prehistory Viitattu 3.11.2020

Saatavissa <https://www.gimp.org/about/prehistory.html>

learn.unity.com 2020 Introduction to Tilemaps Viitattu 15.11.2020

Saatavissa <https://learn.unity.com/tutorial/introduction-to-tilemaps#>

Machkovech, S. 2020 Unreal Engine is now royalty-free until a game makes a whopping \$1 million Viitattu 1.11.2020

Saatavissa <https://arstechnica.com/gaming/2020/05/unreal-engine-is-now-royalty-free-until-a-game-makes-a-whopping-1-million/>

Metrailler, G. 2007 Coreldraw history Viitattu 3.11.2020

Saatavissa <https://community.coreldraw.com/share/b/gerard/posts/a-little-history-of-coreldraw-graphics-suite>

Openclipart.org, isaiah658. 2016 Retro Character Sprite Sheet Viitattu 28.11.2020

Saatavissa [Retro Character Sprite Sheet - Openclipart](https://openclipart.org/isaiah658/2016/Retro-Character-Sprite-Sheet)

Rouce, A. 2005. Game Design: Theory & Practice Second Edition. Viitattu 20.10.2020

Saltzman, M. 2020. Secrets of the Sages: Level Design. Viitattu 29.10.2020

Saatavissa https://www.gamasutra.com/view/feature/131767/secrets_of_the_sages_level_design.php?print=1

Sefton, J. 2008 The roots of open-world games Viitattu 25.10.2020

Saatavissa <https://www.gamesradar.com/the-roots-of-open-world-games/>

Shahrani, S. 2006 Educational Feature: A History and Analysis of Level Design in 3D Computer Games - Pt. 1. Viitattu 26.10.2020

Saatavissa https://www.gamasutra.com/view/feature/131083/educational_feature_a_history_and_.php

Shustek, L. 2013 ADOBE PHOTOSHOP SOURCE CODE Viitattu 3.11.2020

Saatavissa <https://computerhistory.org/blog/adobe-photoshop-source-code/?key=adobe-photoshop-source-code>

Skylum.com 2020 Luminar 4 Viitattu 3.11.2020

Saatavissa https://skylum.com/luminar?utm_source=google&utm_medium=cpc&utm_campaign=Luminar_brand_search_ww_en&utm_term=exact&gclid=Cj0KCQjw28T8BRDbARIsAEOMBcx5oRxyFeIR1IC-2nsWZi72hReLq47Wt7rn0RGUNhruVMXtEZdlbIEaArAzEALw_wcB

Smith, G. 2008. A framework for analysis of 2D platformer levels. Viitattu 22.10.2020

Saatavissa https://www.researchgate.net/publication/229039146_A_framework_for_analysis_of_2D_platformer_levels

Sobolev, J. 2020 What Are Sprites And How They Work In Games? Viitattu 5.11.2020

Saatavissa <https://gamingshift.com/sprites-in-games/>

Studytonight. 2020. Game Engine and History of Game Development Viitattu 2.11.2020

Saatavissa <https://www.studytonight.com/3d-game-engineering-with-unity/game-engine#>

Techterm.com 2012 Sprite Definition Viitattu 5.11.2020

Saatavissa <https://techterms.com/definition/sprite>

Unity.com. 2020 Viitattu 1.11.2020

Saatavissa <https://store.unity.com/products/unity-pro>

Wilson, L. 2019 Godot, Unity, Unreal Engine, CryEngine? Which Game Engine Should I Choose? Viitattu 1.11.2020

Saatavissa <https://medium.com/@thelukaswils/godot-unity-unreal-engine-cryengine-which-game-engine-should-i-choose-553f8ff7999f>

Wolf, M. 2008. The Video Game Explosion. Viitattu 26.10.2020

YouTube, 2010. Treasure Hunter Challenges - Red Dead Redemption Viitattu 28.11.2020
Saatavissa <https://youtu.be/d10SjygB4gM>

YouTube, 2012. Crash Bandicoot: Part 1 [PS1 Classic] Viitattu 28.11.2020
Saatavissa <https://youtu.be/ioSaPJHxSkU>

YouTube, 2014. Arcade Game: Battle Zone (1980 Atari) [Re-Uploaded] Viitattu
28.11.2020
Saatavissa <https://youtu.be/oBSAaMH5hiw>

YouTube, 2015. Undertale - Full Pacifist Playthrough - No Commentary Viitattu
28.11.2020
Saatavissa <https://youtu.be/vEVR2FUB3ic>

YouTube, 2016 A. Super Mario Bros (NES) Level 1-1 Viitattu 28.11.2020
Saatavissa <https://youtu.be/-avspZlbOWU>

YouTube, 2016 B. Unreal Engine 4 - Adding Basic Movement (With controller support) Viitattu 28.11.2020
Saatavissa https://youtu.be/YA1r_gwS9yg

YouTube, 2017 A. UNCHARTED: The Lost Legacy ENDING · Chapter 9: End of the Line
(100% Collectibles) Viitattu 28.11.2020
Saatavissa <https://youtu.be/Xvsg8J7Fy3Q>

YouTube, 2017 B. Lyle Bickley explains the PDP-1 (and we play the original Spacewar!)
Viitattu 28.11.2020
Saatavissa <https://youtu.be/1EWQYAfMYw>

YouTube, 2017 C. Cryengine 5 Getting Started : Install & First Project Viitattu 28.11.2020
Saatavissa <https://youtu.be/aubfy1dUjCE>

YouTube, 2018 A. Uncharted: The Lost Legacy — Train Level Early Prototype Viitattu
28.11.2020
Saatavissa <https://youtu.be/5Cv30vCgPso>

YouTube, 2018 B. How to Get Map Legend Trophy Guide - 200.6% Map - Castlevania
Symphony of the Night – Requiem Viitattu 28.11.2020
Saatavissa https://youtu.be/_wzv2PP4j2I

YouTube, 2020 Unreal Engine 4 Beginner Tutorial: Getting Started Viitattu 28.11.2020
Saatavissa <https://youtu.be/qeKDRqWtGV0>