

Paavo Alavesa

ERIKOISTEHOSTEET VIDEOPELEISSÄ

Partikkelitehosteiden luominen Gerty-peliin

ERIKOISTEHOSTEET VIDEOPELEISSÄ

Opinnäytetyö

Paavo Alavesä
Opinnäytetyö
Kevät 2019
Tietojenkäsittelyn tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittelyn tutkinto-ohjelma, Digitaalinen media

Tekijä(t): Paavo Alavesä

Opinnäytetyön nimi: Erikoistehosteet videopelissä

Työn ohjaaja: Jouni Juntunen

Työn valmistumislukukausi ja -vuosi: 2019

Sivumäärä: sivut + liitteet:
34

Opinnäytetyön tarkoituksena oli luoda erikoistehosteita Spawn Point -osuuskunnan peliin, Gertyyn. Tehosteet luotiin ensin tekemällä tekstuurikartta Adobe Photoshop -kuvankäsittelyohjelmalla, jonka jälkeen ne liitettiin itse peliin käyttämällä Unity3D-pelimoottorin partikkeli järjestelmää. Koska peli on 2D-peli, tein vain 2D-tehosteita. Spawn Point osuuskunnan puolelta ohjaajana toimi Risto Ihalainen.

Korjasin huonot tehosteet sekä tein uusia tehosteita, esimerkiksi räjähdysä, tulta ja sähköä. Tehosteet tehtiin tekemällä animaatioita sekä tekemällä staattisia tehosteita tekstuurikarttojen avulla, jonka jälkeen niitä hiottiin käyttämällä Unityn työkaluja. Tutkin 2D-animaation periaatteita ja erilaisia työskentelytapoja sekä Unityn partikkelityökaluja.

Käytin paljon erilaista internetistä saatavaa ainestoa kuten blogikirjoituksia, arvosteluita, ohjekirjoja sekä YouTube:sta löytyviä oppaita. Projekti saatiin siihen pisteeseen, että pystyimme julkaisemaan pelin Early Access -pelinä.

Asiasanat: erikoistehoste, partikkelitehoste, animaatio, unity, photoshop

ABSTRACT

Oulu University of Applied Sciences
Information technology, Digital media

Author(s): Paavo Alavesa

Title of thesis: Special Effects for a Video Game

Supervisor(s): Jouni Juntunen

Term and year when the thesis was submitted: 2019

Number of pages: 34

The reason of the thesis was to create different kind of effects for Spawn Point's game called Gerty. I created effects using Adobe Photoshop to create a texture sheet and then taking it to Unity3D. Because the game is 2D game, I only did 2D particle effects. My supervisor at Spawn Point was Risto Ihalainen.

I fixed bad looking effects and created new effects like explosions, fire and electricity. I did animations and static effects using texture sheets and after that I tweaked them in Unity using particle effect tools.

I researched different kind of material found in internet, like blogs, reviews, guides and YouTube tutorials. We were able to release the game on Steam as Early Access game.

Keywords: special effect, particle effect, animation, unity, photoshop

SISÄLLYS

1	JOHDANTO	6
1.1	Tausta	6
1.2	Lähtötilanne.....	7
1.3	Tavoitteet ja kehitystehtävät.....	8
1.4	Työkalut.....	9
2	ERIKOISTEHOSTEET PELEISSÄ	11
2.1	Erikoistehosteet.....	11
2.2	2D-animointi	12
2.3	Tekstuurikartta.....	13
2.4	Partikkelitehoste	17
3	ERIKOISTEHOSTEIDEN TOTEUTUS	20
3.1	Peliin tehdyt tehosteet	20
3.2	Partikkelitehosteen työnkulku	23
4	POHDINTA	33
	LÄHTEET.....	34

1 JOHDANTO

Useanlaista peligrafiikkatekniikkaa on käytetty näyttämään pelin sisältöä koko videopelien historian aikana. Laitteiston kehityksen myötä, myös videopelien grafiikat ovat kehittyneet tekstipohjaisista grafiikoista lähes jopa hyvin aidon näköisiin grafiikoihin.

Vaikka lapsuutemme alusta asti olemme ihmetelleet kuinka aidon näköiset grafiikat ovat ja ei se enää paremmaksi voi muuttua, on se joka vuosi yllättänyt vain enemmän. Kun ensimmäinen Tomb Raider -peli julkaistiin Playstation 1 -konsoleille ja kuinka hyvältä Lara Croftin polygoninen keho kaikkineen kulmineen näyttikään, olemme päässeet seuraamaan hahmon graafista kehitystä (katso kuva 1) näiden vuosien aikana, ja onhan se yllättänyt joka kerta.



Kuva 1. Tomb Raider pelin päähahmon graafinen kehitys vuosien aikana

1.1 Tausta

Heti harjoitteluni jälkeen Kuvionilla pääsin Spawn Point -osuuskunnalle tekemään heidän Gerty-peliin uusia erikoistehosteita vanhojen sekä olemattomien tehosteiden tilalle. Harjoittelin ja toteutin tehosteita peliin kuukauden ajan, ennen kuin päätin lähteä tekemään opinnäytetyötäni aiheesta. Pelin julkaisu oli joulukuussa 2018, joten tehosteita tehtiin melko nopealla aikataululla käyttäen Unity-pelimoottoria sekä Adobe Photoshop -kuvankäsittelyohjelmaa.

Itse opinnäytetyön työosan tekeminen alkoi käytännössä heti, vaikka en ollut vielä päättänyt ottavani tätä projektia opinnäytetyöaiheeksi. Opettelin Unityn partikkelityökalujen käyttämistä katsomalla paljon opetusvideoita, kunnes aloitin varsinaisen projektin.

1.2 Lähtötilanne

Gerty-peliä työstää pieni osuuskunta Oulusta, joka koostuu kolmesta työntekijästä: Risto Ihalainen, Ilpo Alatalo ja Mikko Lapinlahti. Ristoa pidetään projektin isänä ja hoitaa projektin koodipuolen. Ilpo toimii projektin tuottajana sekä koodaajana. Hän myös hoitaa videoeditoinnin, blogin sekä sivuston ylläpitämisen. Mikko on yhden miehen bändi, ja tuottaa peliin scifi-musiikkia.

Gerty (katso kuva 2) on neljän pelaajan rogue-lite co-op -peli, joka keskittyy haastavaan taisteluun täysin tuhattavassa maastossa. Tehtäväsi on tutkia kaivosten sarjaa, jotka kuhisevat odottamatonta muukalaisvoimaa. Näissä kaivoksissa tuotetaan tärkeää Juice-resurssia, joten ne täytyy tyhjentää muukalaisista hinnalla millä hyvänsä. Kaivosten välillä toimii teleporttijärjestelmä, jolla pääsee kaivoksen eri tasoihin.



Kuva 2. Gerty-peli

Jokaisella pelin neljästä hahmosta on vain neljä taitoa. Hyökkäys, kaksi erikoisliikettä sekä erikoishyökkäys, jolla on suuri vaikutus, kuinka hahmoluokka toimii. Muilla kuin normaalilla hyökkäyksellä on muutaman sekunnin palautumisaika.

Taitojen ja niiden palautumisajan hallinta on tärkeässä roolissa pelissä, sillä jokainen hahmo käyttäytyy hyvin erillä lailla erilaisten taitojensa takia. Esimerkiksi yhdellä pelin hahmolla on vain väistöpyörähdys, joka muissa peleissä tyypillisesti on monesti kaikilla hahmoilla. Muilla hahmoilla on sitten omat puolustusmekanisminsa. Juice-resurssi toimii valuuttana, jolla voi ostaa erilaisia parannuksia hyökkäyksiin, puolustukseen, varustukseen ja statukseen.

1.3 Tavoitteet ja kehitystehtävät

Projektiin tarvittiin nopealla aikataululla paljon uusia ja erilaisia erikoistehosteita vanhojen sekä olemattomien erikoistehosteiden tilalle. Minulle annettiin pitkä lista tarvittavista ja korvattavista tehosteista, jota sitten päiviteltiin tehosteiden loppuessa. Listaan kuului puolustus-, hyökkäys-, varustus ja statustehostukset sekä erilaisia tehosteita vihollisille. Tehosteilla ei ollut mitään varsinaisia kriteereitä, niiden piti vain sopia pelin graafiseen tyyliin. Pelin graafinen tyyli on kaksiulotteista, jossa on pientä kolmiulotteisuuden vivahdetta, esimerkiksi hahmoissa.

Erilaisia räjähdystehosteita oli paljon, sillä niitä piti tehdä eri hahmoluokille sekä vihollisille. Oli normaaleja räjähdysräjähdys, sähköpurkauksia, ja myrkkyräjähdysräjähdys. Tavoitteena oli tehdä yksi räjähdys päivässä. Toisin kuin räjähdystehosteet, statustehosteet taas pysyvät hahmon päällä niin kauan kuin status kestää. Statustehosteita tuli tehdä myös yksi päivässä. Kun tehoste valmistui, katsoimme sitä yhdessä porukalla ja päätimme, että otamme tehosteen peliin käyttöön.

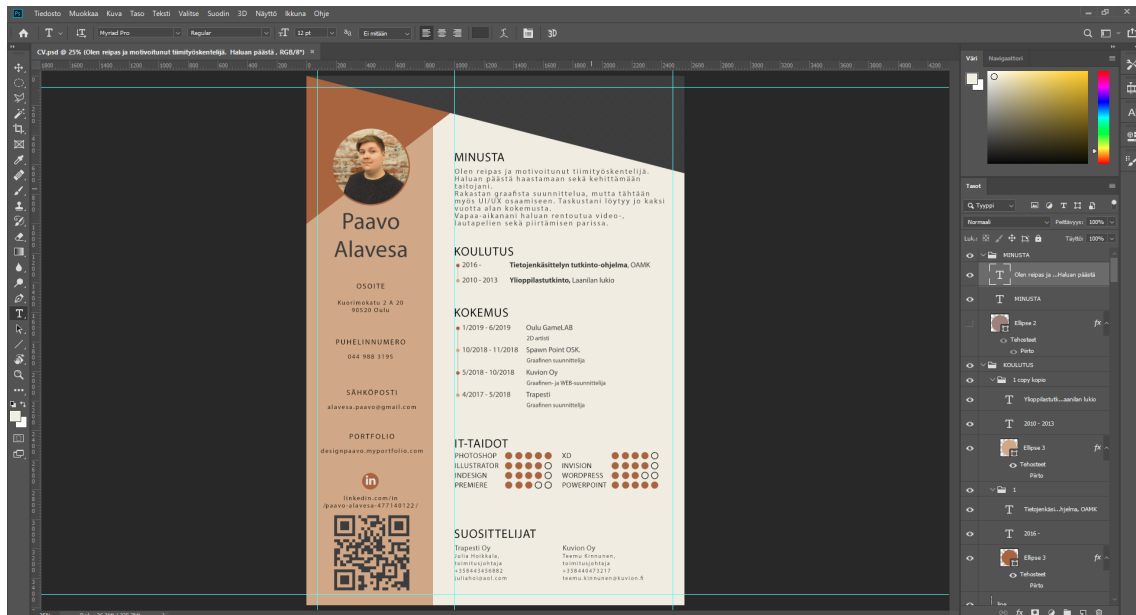
Partikkelitehosteet olivat minulle täysin uusi asia, joten aloitin opettelemalla tehosteiden tekemistä katsomalla erilaisia opastusvideoita YouTubesta. Tämän jälkeen jatkoin tekemällä tehosteita, joita tarvittiin mahdollisimman nopeasti peliin.

Tähtäsimme myös siihen, että saimme projektin julkaisukuntoon ja voisimme julkaista sen Steamissä Early Access -pelinä joulukuun mennessä. Tämän jälkeen peliin kehitettäisiin uudelleenpe-luuarvoa ja uusia hahmoja.

1.4 Työkalut

Tehosteet tehtiin käyttämällä Unity3D -pelimoottoria sekä Adobe Photoshop -kuvankäsittelyohjelmaa. Aluksi tehtiin tekstuurikartta Photoshopilla, jonka jälkeen tekstuurikartta syötettiin Unity3D-pelimoottoriin ja lisäksi siihen värit sekä muita tehosteita.

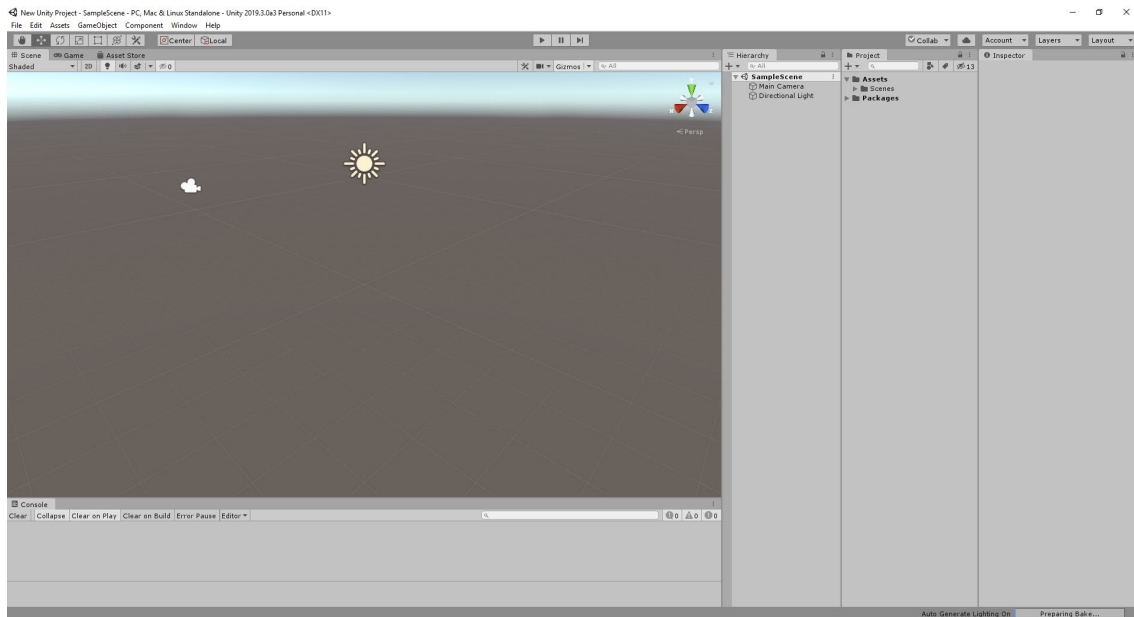
Photoshop on Adoben kehittämä kuvankäsittelyohjelma (katso kuva 3) Windows- ja MacOS-käyttöjärjestelmille, jolla voidaan muokata ja luoda pikseligrafiikkaa. Sillä on myös kaikista intuitiivisin käyttöliittymä verrattuna muihin kuvankäsittelyohjelmiin sekä on suosituin kuvankäsittelyohjelma ammattilaisten keskuudessa. Photoshopin etu muihin kuvankäsittelyohjelmiin on se, että sitä voidaan käyttää graafisen suunnittelun lisäksi taiteen tekemiseen, animoimiseen sekä web-suunnitteluun (Choudhary 2018).



Kuva 3. Adobe Photoshopin käyttöliittymä

Käyttämällä tasopohjaista muokkaustapaa, joka mahdollistaa useamman peittokuvan luomisen ja muokkaamisen, jotka tukevat läpinäkyvyyttä. Tasot voivat toimia myös maskeina tai filtereinä, muokaten taustalla olevia värejä. Voimme myös lisätä varjoja ja muita tehosteita tasoihin (Photoshop Essentials 2016).

Unity3D on maailman suosituin pelimoottori (katso kuva 4). Siinä on todella paljon erilaisia ominaisuuksia ja on tarpeeksi joustava, että sillä voi valmistaa minkä tahansa pelin (Clicklabs 2014).



Kuva 4. Unityn käyttöliittymä

Koska Unity3D:ssä on loistavat monitasoinen yhteensopivuus ominaisuudet, on se suosittu harrastajien sekä AAA-studioiden, eli ison budjetin pelistudioiden keskuudessa. Sillä on tehty esimerkiksi Pokémon Go, Hearthstone sekä Cuphead (Clicklabs 2014).

Ohjelmoijat pitävät ohjelmasta, koska se tukee C#-ohjelmointikieltä sekä sisäänrakennettua Visual Studio integraatiota. Artistit taas pitävät siitä, koska siinä on erinomaiset animaatiotyökalut, jonka takia sillä on yksinkertaista tehdä 3D sekä 2D välianimaatioita (Clicklabs 2014).

Unity3D tarjoaa ilmaisen version kehittäjille pelien julkaisemiseen, kunhan sillä ansaitaan vähemmän kuin 100,000 dollaria. Ne, jotka haluavat maksaa Unitysta, saavat enemmän ominaisuuksia sekä joustavan lisensointisuunnitelman. Premium käyttäjillä on myös pääsy Unityn lähdekoodiin (Clicklabs 2014).

2 ERIKOISTEHOSTEET PELEISSÄ

Erikoistehosteet ovat varmasti jokaiselle tuttuja ainakin jollakin asteella, sillä niitä käytetään peleissä ja elokuvissa luomalla jonkinlainen tehoste, kuten räjähdystehoste. Tehoste voi olla joko oikea räjähdys tai sitten tietokoneella tehty illuusio räjähdyksestä.

Charlie Chaplin on tunnettu juurikin kekseliäistä tehosteistaan, jossa näyttelijä näyttäisi olevan vaarallisissa tilanteissa. Tehosteen tärkein osa onkin kameran asettelu oikeaan paikkaan oikeaan kulmaan ja kuvaamaan erilaisten kuvien vierestä (katso kuva 5).



Kuva 5. Tunnettu Charles Chaplinin tehoste (Chan 2017)

2.1 Erikoistehosteet

Erikoistehosteet voidaan jakaa kolmeen eri kategoriaan, jotka ovat optiset tehosteet, käytännönläheiset tehosteet sekä tietokoneella luodut tehosteet eli CGI (computer-generated imagery). Tehosteet jaetaan yleensä kolmeen eri ryhmään, jotka sisältävät erilaisia tapoja luoda tehosteita erilaisiin käyttötarkoituksiin. Ryhmät ovat optiset, mekaaniset ja tietokoneella luodut tehosteet (Dube 2018).

Optinen tehoste luodaan esimerkiksi muuttamalla kameran iiristä isommaksi tai pienemmäksi. Näin saadaan häivytettyä tai voimistettua valoa kuvan reunoilta, jolloin saadaan häivytyksen tehoste. Jättämällä iiriksen tiettyyn kohtaan voidaan jäljitellä miltä kiikaritähtäin näyttää siitä läpi katsottuna (Film Reference 2018).

Mekaanisilla tehosteilla tarkoitetaan kameran kuvattavaksi tarkoitettuja tehosteita, kuten erilaiset räjähdykset tai tulipalot käyttämällä pyrotekniikkaa sekä tuulettimella luotu tuuli. Mekaanisiin tehosteisiin lasketaan mukaan myös meikkaaminen sekä proteesit (Film110 2018).

Tietokoneella luotuja tehosteita ovat kaikki tehosteet, jotka on tehty käyttämällä tietokonetta. Täten tehosteet ovat laadukkaampia sekä kontrolloidumpia toisin kuin mekaaniset tehosteet (ScienceDaily 2018). Myöskin kustannuksissa säästetään paljon, kun tehoste luodaan tietokoneella. Tyyppillisimpiä CGI-tehosteita ovat erilaiset räjähdykset, rakennelmien hajoamiset sekä futuristiset hologrammit, jotka on lisätty jälkeenpäin videonmuokkauksessa.

2.2 2D-animointi

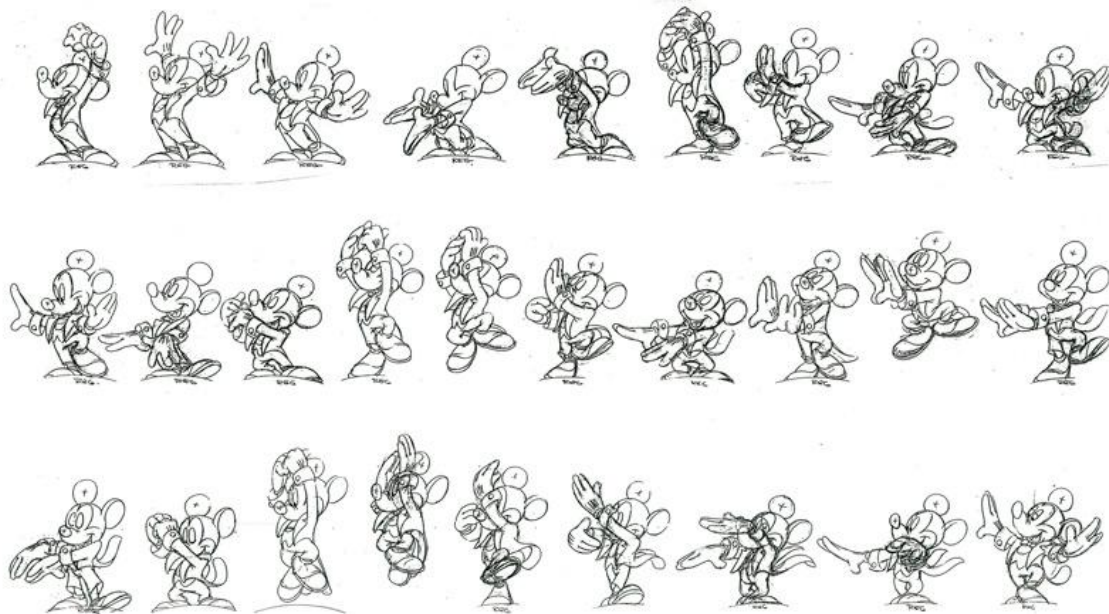
Visuaaliset tehosteet käyttävät animaation perusteita. Periaatteet kuten litistys ja venytys, arkit sekä toissijaiset toiminnot sekä ajoitus ovat tärkeitä. Ne kaikki tekevät visuaalisen tehosteen tuntu-
masta vaikuttavan ja dynaamisen (Tokarev 6.2.2018).

Animaation esi-isänä voidaan pitää yhtä vanhimmasta animaatiosta, joka valmistettiin Jumala Isisille 1600 eaa. 110 pilariin maalattujen Isis-jumala hahmojen asennot muuttuvat jokaisen pilarin kohdalla. Vuonna 1640 Anthonasius Kirchowrin ”Magic Lantern” animaatiossa piirrettiin kuvia erilaisille laselle ja niitä liikuttamalla valon edessä simuloitiin liikettä (Williams 2001, 12-13.).

Ensimmäinen filmille tallennettu animaatio saatiin Thomas Edisonilta vuonna 1896, joka kuvasi James Stuart Blacktonin piirroksia. 1900-luvulla valkokankaille saapuivat perinteisemmällä menetelmällä tehdyt animaatiot, kuten Mikki Hiiri ja Felix Kissa. Nykyään animaatioita on laaja valikoima lapsille ja myöskin aikuisille (Williams 2011, 15, 17, 20).

Videopelien animaatiot alkoivat alkuaikoina vain muutaman framen pikselihahmoilla. Nyt kun koneiden tehot ovat nousseet suunnattomasti on voitu tehdä hyvin realistisia grafiikoita sekä animaatioita. Monet indie- ja mobiilikehittäjät käyttävät edelleenkin 2D-animaatioita. Oikeastaan 2D-animaatioit ovat nousseet pinnalle indie-pelien, eli yksittäisten henkilöiden tai pienten ryhmien kehittämien pelien mukana viime aikoina (Njiri 2016).

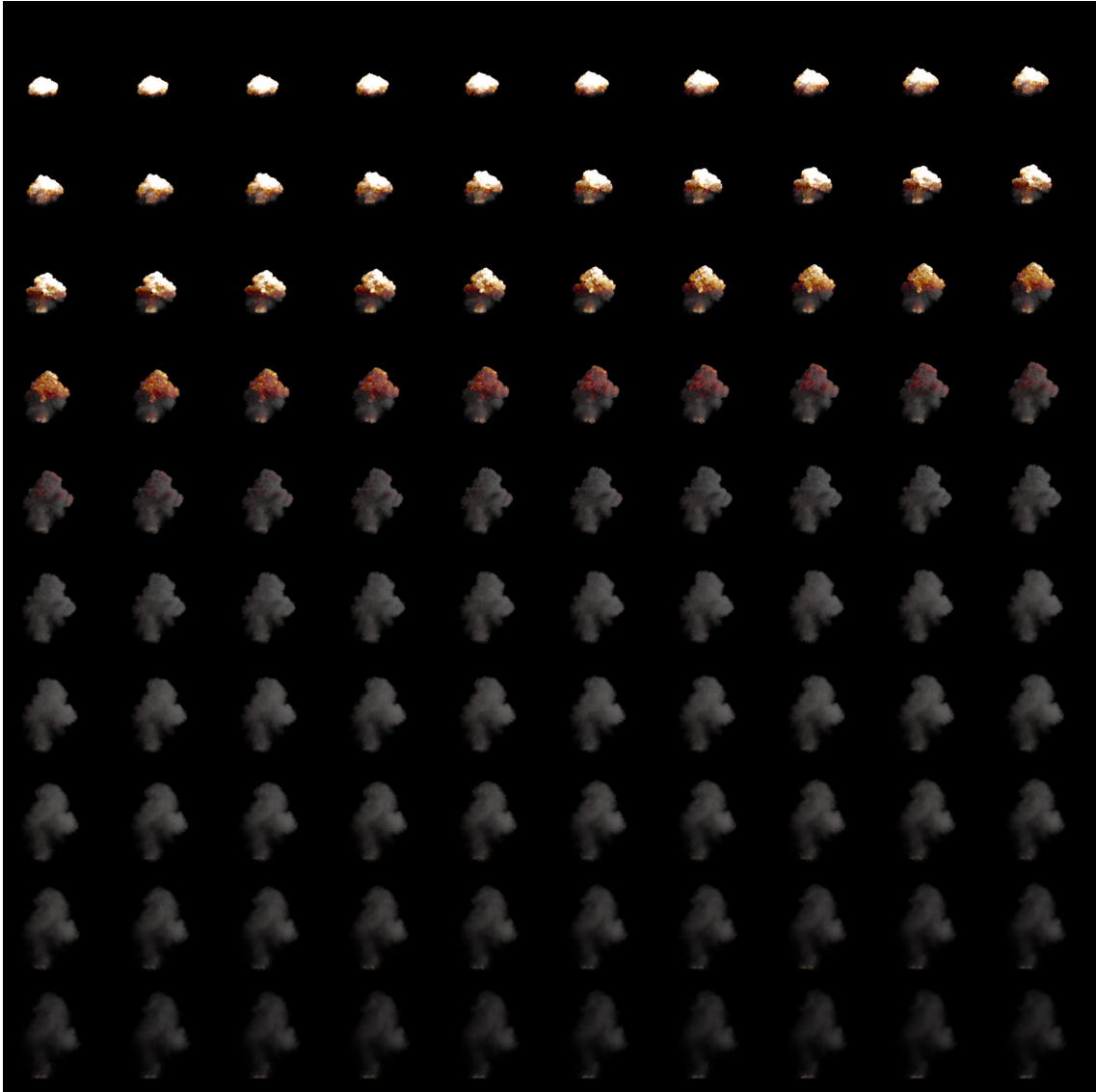
Liikeanimaatio on tekniikka, jossa luodaan liikkeen illuusio tekemällä pieniä muutoksia jokaisen kuvan aikana (katso kuva 6) (Tylor 2019). Eli jos animaatio on 60 framea sekunnissa, vaihtuu kuva sekunnin aikana 60 kertaa. Tällä tekniikalla luodaan myös monet pelien erikoistehosteista.



Kuva 6. Disneyn esimerkki liikeanimaatiosta kuva kovalta (Tylor 2019)

2.3 Tekstuurikartta

Tekstuurikarttoja eli sprite sheetejä on käytetty peleissä jo kauan aikaa. Sprite on yksi graafinen kuva, joka on sisällytetty laajempaan kokonaisuuteen ollakseen osa kohtausta. Lähes kaikki pelitehosteet alkavat sprite sheetin luonnista. Tehosteet käyttävät sprite sheet kokoelmaa, joka koostuu eri spriteistä. Kartan sisältämät spritet voivat myös liikkua vasemmalta oikealle luoden spriteanimaation. Kyseessä on silloin liikkuva teksturi (katso kuva 7) (Code Web 2010).



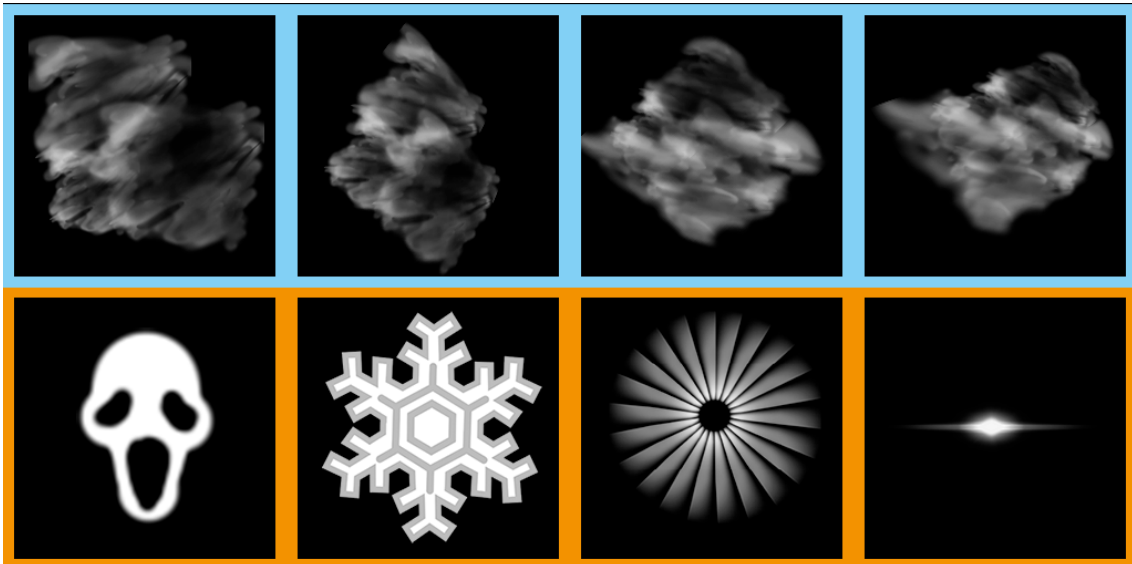
Kuva 7: Sprite sheet räjähdysanimaatiosta (Saraiva 2017)

Spritet ovat suosittu tapa luoda suuria, monimutkaisia kohtauksia sillä voit manipuloida jokaista spriteä erikseen. Tämä mahdollistaa kohtauksen renderöinnin hallinnan (Unity Documentation 2019).

Peleissä monesti onkin kymmeniä, jopa satoja erilaisia spritejä. Jokaisen erillisen spriten lataaminen veisi todella paljon muistia ja tehoa. Siksi käytetään tekstuurikarttoja eli sprite sheetejä.

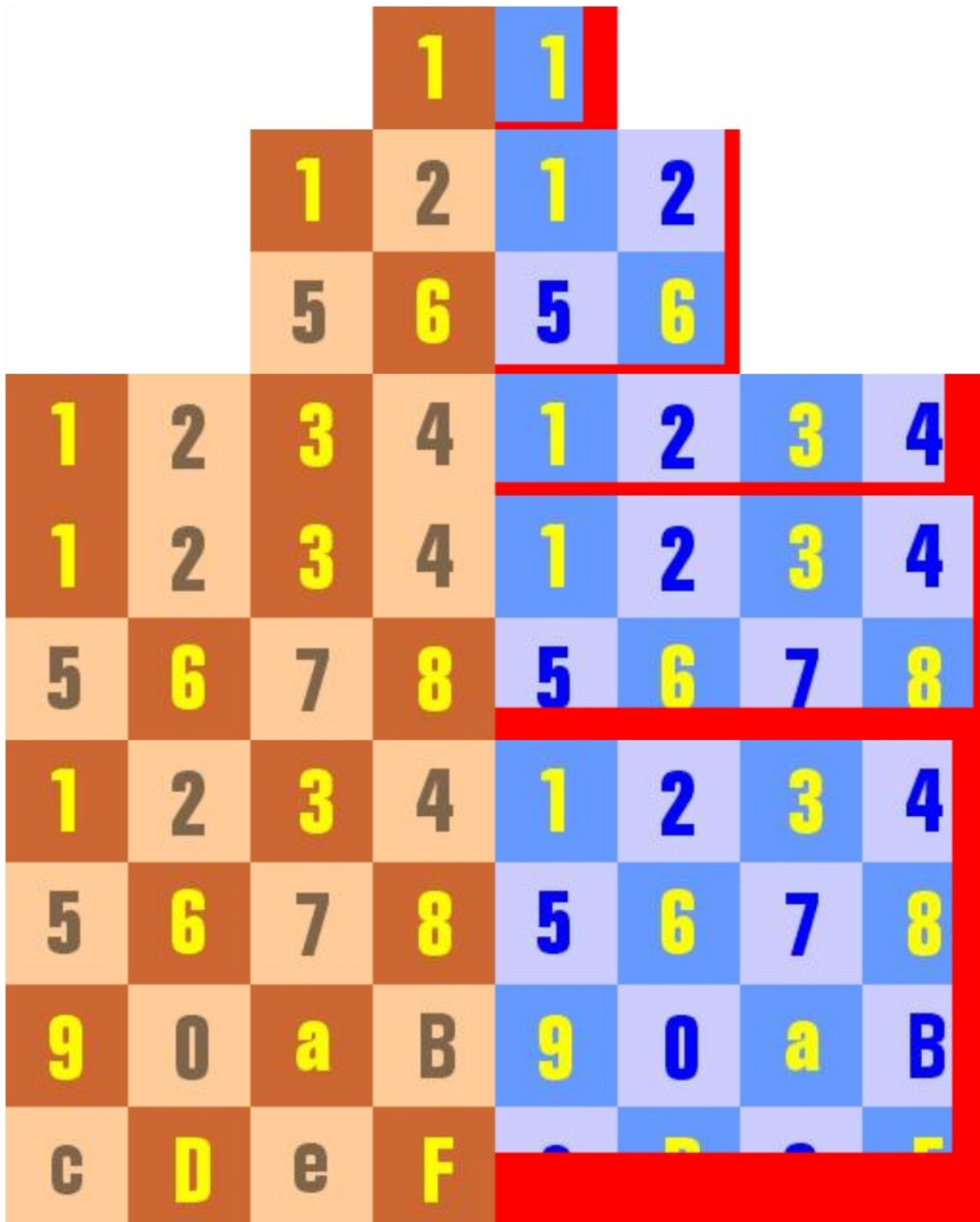
Sprite sheetejä käytetään nopeuttamaan kuvien näyttämistä näytöllä, sillä se on paljon nopeampaa noutaa yksi kuva yhdeltä tekstuurikartalta kuin etsiä useampi kuva ja näyttää ne erikseen (Unity Documentation 2019).

Kun luodaan animaatiota sprite sheetin avulla, otetaan jokainen kuva vasemmalta oikealle ja luodaan illuusio liikkeestä kuva kuvalta. Staattisen spriten ulkonäkö ei muutu vaan kaikki liike tapahtuu tehoseen lähettäjän eli emitterin liikuttamisella (katso kuva 8). Emitteri voidaan asettaa pelaajan päälle, jolloin se seuraa pelaajaa (Unity Documentation 2019).



Kuva 8. Sprite sheet, jossa on sinisten laatikoiden sisällä animoituja sekä oranssien laatikoiden sisällä staattisia elementtejä

Kuten normaalit tekstuurikartat, sprite sheettien on oltava "Power of Two" eli kahden potenssi säännön mukaisia (katso kuva 9). Syy sääntöön on, että pelimoottorit on koodattu toimimaan niin, jotta pelimoottori voi prosessoida tietoa tehokkaammin (KatsBits 2019).



Kuva 9. Vasemmalla olevia (64x64, 128x128, 64x256, 128x256, 256x256) suositellaan käytettävän ja oikealla olevia (46 x 60, 120 x 123, 235 x 57, 250 x 111, 239 x 216) ei kannata käyttää (KatsBits 2019)

2.4 Partikkelitehoste

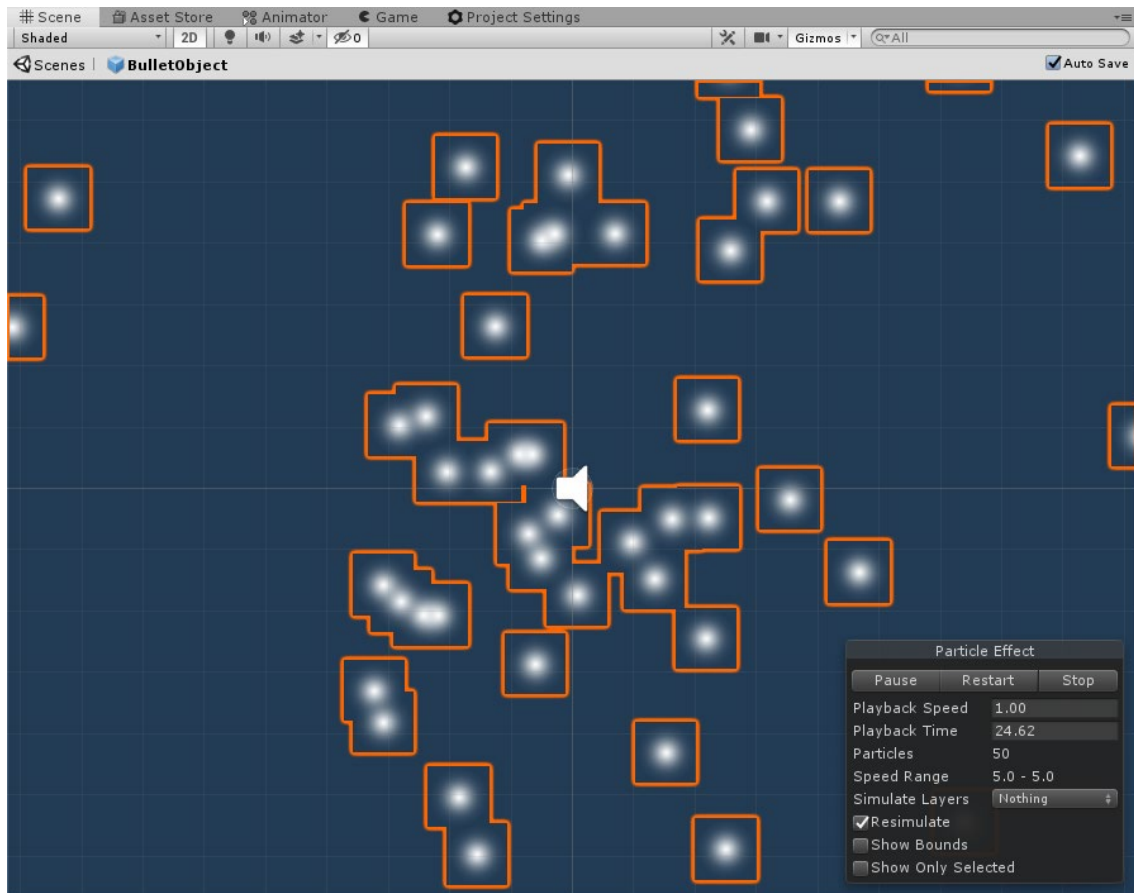
Pelien erikoistehosteet, eli partikkelitehosteet käyttävät samoja periaatteita kuin elokuvatehosteet. Partikkelitehosteet demonstroivat erilaisia ympäristö- sekä elementtitehosteita, joiden tarkoituksena on luoda realistisia ja utopistisia visuaalisia tehosteita käyttäen sprite-sheettiä eli kokoelmaa pieniä kuvia yhdessä suuressa kuvassa tai meshiä eli kolmiulotteista mallia (katso kuva 10). (Unreal Engine 2018).



Kuva 10. Lumitehoste, vesitehoste sekä tulitehoste, jotka on luotu käyttämällä partikkelitehosteita (Unreal Engine 2018)

Jokainen partikkeli esittää pientä osaa nestettä tai epämääräistä kokonaisuutta ja tehosteen kaikki partikkelit luovat vaikutuksen kokonaisuudesta. Jos tarkoituksena on tehdä laskentatehoa vähän vievä peli, niin vaikeammin hallittavilla partikkeleilla tehtyjä tehosteita parempi vaihtoehto on tehdä sprite-sheet -animaatio käyttämällä 2D-kuvia (Unity 2018).

Partikkeli tehoste on pienien objektien tai spritejen, partikkeleiden, simulaatio tietyn ajan aikana (katso kuvio 11). Näillä partikkeleilla on tyypillisesti paikka ja nopeus, joihin vaikuttaa erilaiset voimat, kuten painovoima. Joissakin toteutuksissa voi olla sisäisiä partikkelitehosteita ja monia muita hienoja tehosteita (Unity Documentation 2019).



Kuva 11. Unityn oletuspartikkelit (Day 2019)

Jokaisella partikkelilla on ennalta määritetty elinikä, tyypillisesti vain muutama sekunti, jonka aikana se voi muuttua useamman kerran. Sen elinikä alkaa, kun se on generoitu tai lähetetty ulos sen partikkelijärjestelmästä eli emitteristä. Järjestelmä lähettää partikkeleita satunnaisissa asennoissa ympyrän, kartion, suorakulmion tai pallonpuoliskon muotoisen alueen sisällä, tai sitten mielivaltaisessa kolmiulotteisessa alueessa eli meshissä. Partikkeli elää niin kauan, kunnes siltä loppuu aika, jonka jälkeen se poistuu järjestelmästä. Järjestelmän lähetystaso osoittaa karkeasti, kuinka monta partikkelia emitter lähetetään sekunnissa (Unity Documentation 2019).

Lähetys ja elinikä asetukset määrittävät järjestelmän käyttäytymiseen, mutta yksittäiset partikkelit saattavat vaihtua ajan kanssa. Jokaisella on nopeus vektori, joka partikkelin suunnan ja matkan

jokaisella kuvanpäivityksellä. Nopeutta voidaan vaihtaa käyttämällä voimaa sekä painovoimaa, jotka on syötetty järjestelmään tai jos partikkeleihin vaikuttaa alustan tuulialue. Partikkelin väriä, kokoa ja kiertoa voidaan sen eliniän aikana vaihtaa tai sitten sen tietyn nopeuden aikana. Väri määrittää sen alfa (läpinäkyvyys) komponentin, joten partikkeliä voidaan himmentää vähitellen kokonaan pois sekä takaisin (Unity Documentation 2019).

Partikkeli dynamiikkoja käytettynä yhdessä, voidaan käyttää monenlaisen nesteen simulointiin vakuuttavasti. Esimerkiksi vesiputousta voidaan simuloida käyttämällä ohutta lähtömuotoa ja päästämällä vesipartikkelit putoamaan painovoiman avulla samalla nopeutuen. Tulesta johtuva savu yleensä nousee, laajenee ja ajan myötä hälvenee, joten järjestelmä tulisi käyttää painovoiman vastaista voimaa savupartikkeleihin ja laajentaa sekä häivyttää niitä niiden elinaikana (Unity Documentation 2019).

3 ERIKOISTEHOSTEIDEN TOTEUTUS

Kuvittele kohtaus pelissä, jossa raketti lentää ilman läpi jättäen savuvanan ilmaan. Yhtäkkiä raketti kuitenkin räjähtää lennättäen osia ja kipinöitä joka puolelle. Räjähtäneen raketin osat osuvat olentoon, jonka veri ja kehonosat lentävät kaikkialle jättäen veritahroja. Jokainen osa kohtauksessa on osa hyvää partikkelijärjestelmää. Savua, kipinöitä ja verta tehdään rutiininomaisesti käyttämällä nykyajan partikkelijärjestelmää.

Ymmärtääkseen nämä tehosteet, täytyy rakentaa partikkelisysteemi, joka on nopea, joustava ja muokattavissa. Onneksi nämä partikkelisysteemit löytyvät suurimmasta osasta pelimoottoreista ja partikkelitehosteet ovat helposti muokattavissa ilman isompaa koodin kirjoittamista.

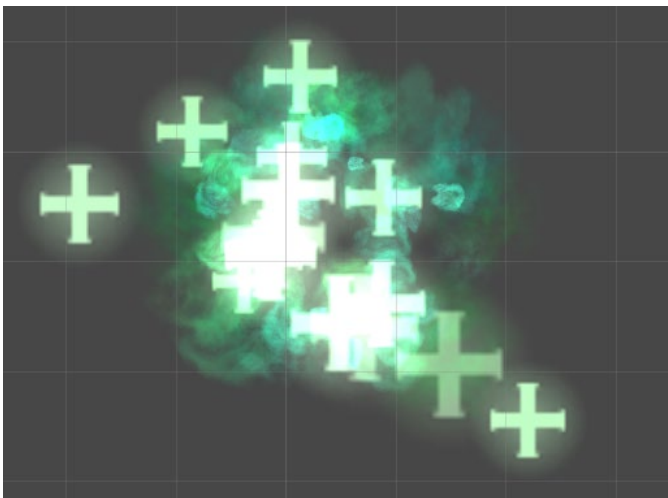
3.1 Peliin tehdyt tehosteet

Minulle annetulla listalla oli paljon tehosteita (katso kuva 12), mutta minua suositeltiin aloittamaan puolustustehosteiden tekeminen, sillä ne olivat yksinkertaisimpia. Listaan oltiin listattu puolustus-, hyökkäys-, varustus ja statustehostuksiin.



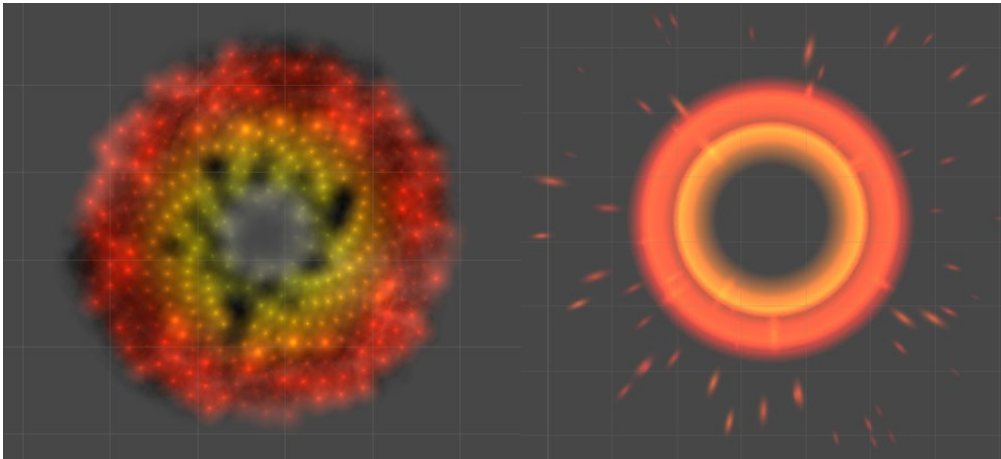
Kuva 12. Erilaisia tekemiäni partikkelitehosteita pelistä

Kaikki alkoi parantavien taitojen luomisella, jotka kuuluivat puolustustehosteisiin. Kokeilin erilaisia yhdistelmiä ja katsomalla mikä toimisi parhaiten. Koska kyseessä on parantava taito, värin ja muodon valinta oli helppoa, sillä ne luovat illuusion parantavasta voimasta ristin sekä vihreän värin avulla (katso kuva 13). Aluksi loin savutehosteen, jossa emitteri luo sekunnin välein uuden satunnaisiin suuntiin suuntaavan savupartikkelirykelmän, jotka pyörivät hitaasti luoden savun illuusion. Tämän saman toistin ristospritellä. Tässä vaiheessa kokeilin enemmänkin, miten partikkelitehosteasetukset toimivat.



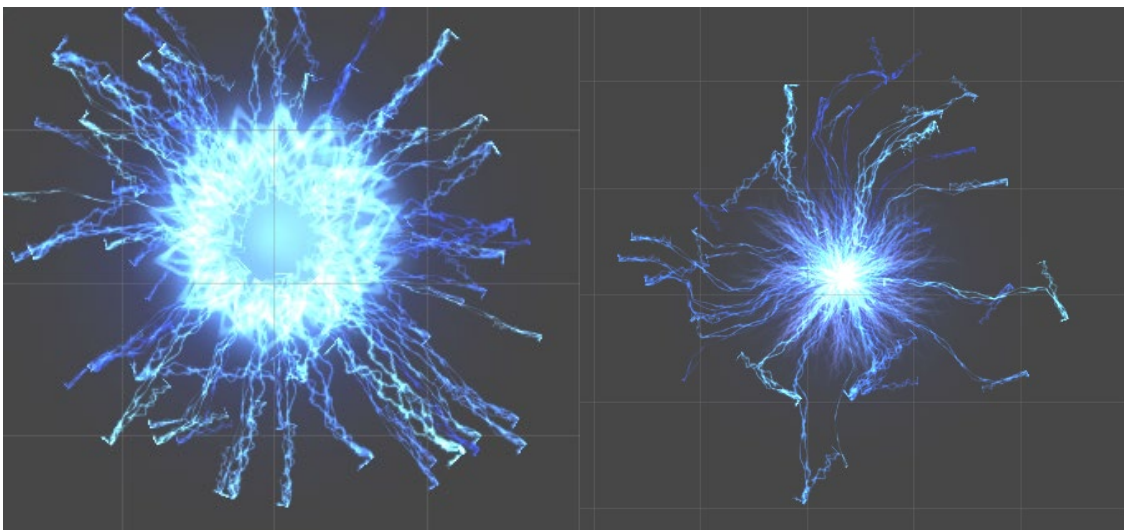
Kuva 13. Healing Cloud on nimensä mukainen taito eli parantava pilvi

Kun aloin luottamaan osaamiseeni, minua suositeltiin jatkamaan erilaisten räjähdysten ja sähkötehosteiden parissa. Räjähdyksessä koostui kolmesta eri vaiheesta. On itse räjähdys, sitten paineaalto, jota seuraa savu tai kipinät (katso kuva 14). Tekemissäni räjähdyksissä emitteri luo ensin usean punaisen ja oranssin värisen pienen ympyrän muotoisen spriten, jotka laajenevat ja pysähtyvät saavuttaessaan tietyn matkan. Tämän jälkeen emitteri luo suurelle alueelle savupartikkeleita ja kipinöitä.



Kuva 14. Erilaisia räjähdystehosteita.

Sähkötehosteissa kokeilin Trail Mesh - ja Noise Module -ominaisuutta. Trail Meshillä tarkoitetaan vauhtiviivoja. Se siis luo partikkelin perään vauhtiviivan, kun partikkeli liikkuu. Vauhtiviiva generoidaan siis kahden pisteen välille, josta otetaan annetulla aikajaksolla näytteitä annetulle tiheydellä. Noise moduuli luo taas partikkeleihin turbulenssia (katso kuva 15). Muuten tehosteen luonti muistuttaa pitkälti räjähdystehostetta.



Kuva 15. Electro Wave ja Tesla Coil -taito, jossa Trail Mesh ja Noise module

3.2 Partikkelitehosteen työnkulku

Koska työnkulku on hyvin samanlainen jokaisessa tehosteessa, ei ole järkeä selostaa jokaisen erillisen tehosteen työnkulkua. Tässä käyn läpi yhden tehosteen työnkulun.

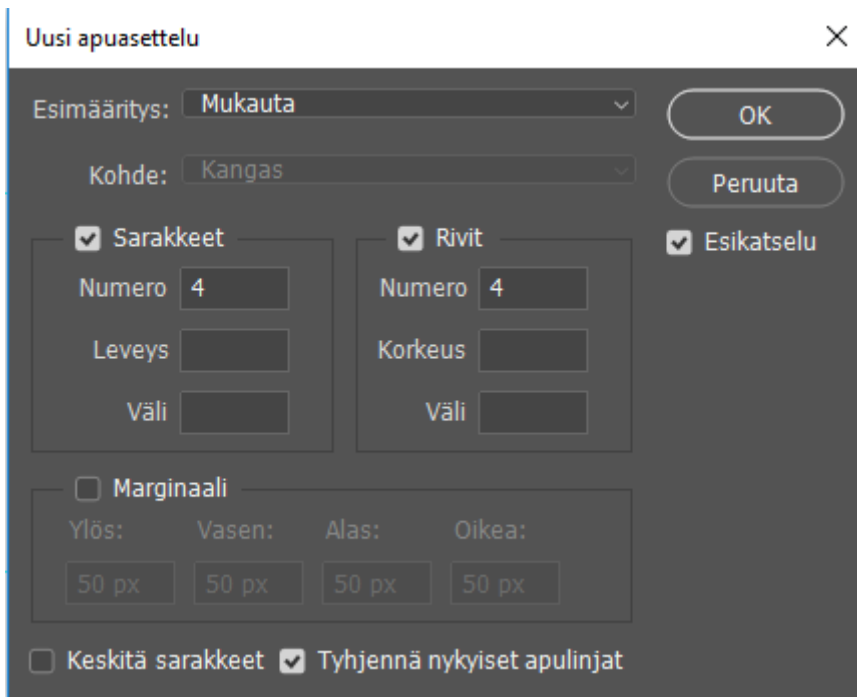
Kaikki alkoi siitä, että tein sprite sheetin partikkelitehosteelle käyttämällä kuvankäsittelyohjelmaa. Itselläni on Adobe Photoshop, joka kävi mainiosti työn tekemiseen.

Aluksi tein uuden PSD-tiedoston nimellä Fire Particle. Se on mitoiltaan 512 x 512 pikseliä ja tarkkuus 300. Käytin myös RGB-väritilaa, joka näyttää värit parhaiten tietokoneen ruudulla (katso kuva 16).



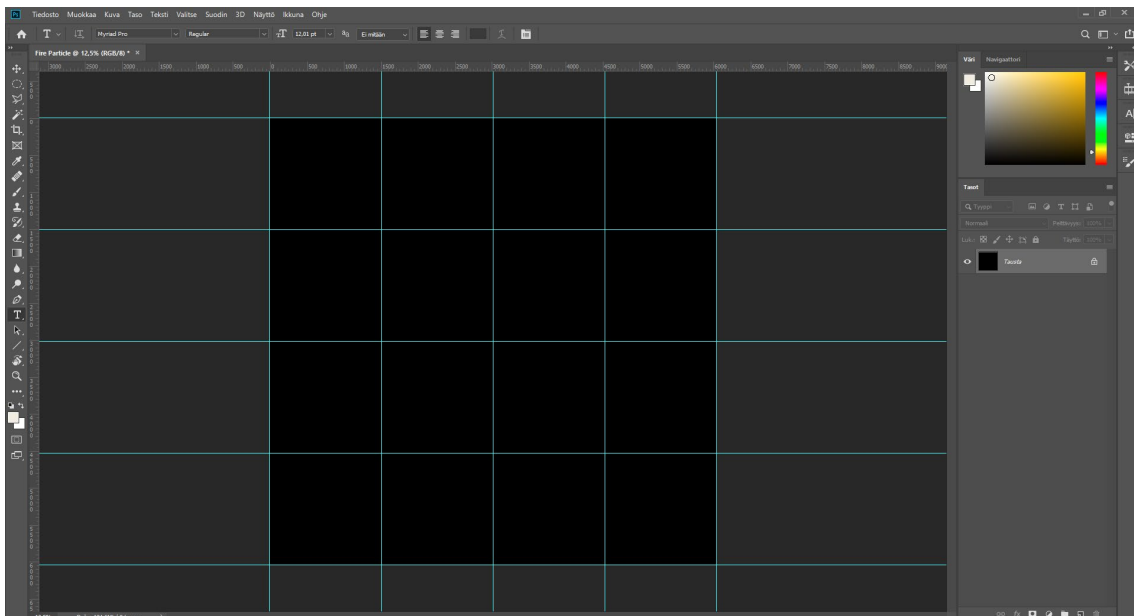
Kuva 16. Kuva Adobe Photoshopin tiedostonluonnista

Seuraavaksi jaoin piirtoalueen 16 osaan käyttämällä apuviivoja. Helpoiten se onnistui käyttämällä apuasettelua (katso kuva 17).



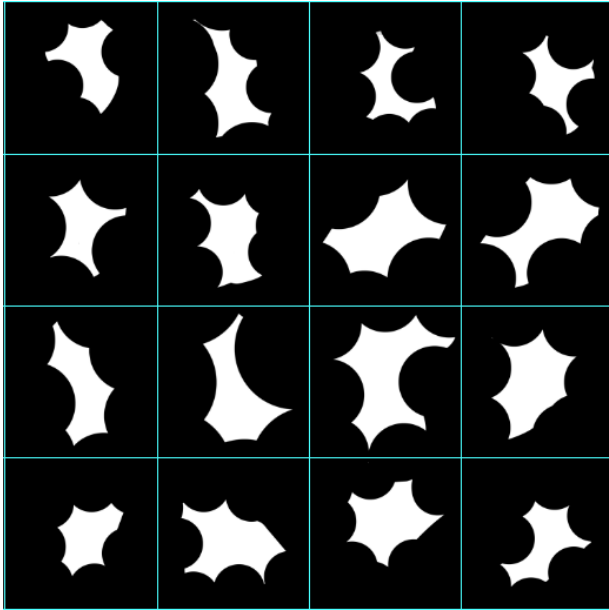
Kuva 17. Kuva apuasettelusta, jossa voi määrittää apuviivojen määrän sekä niiden välissä olevan tilan määrän

Koska piirsin spritet valkoisella, on piirtopohjan oltava musta (katso kuva 18). Tähän käytin pikanäppäinyhdistelmää ctrl + i.



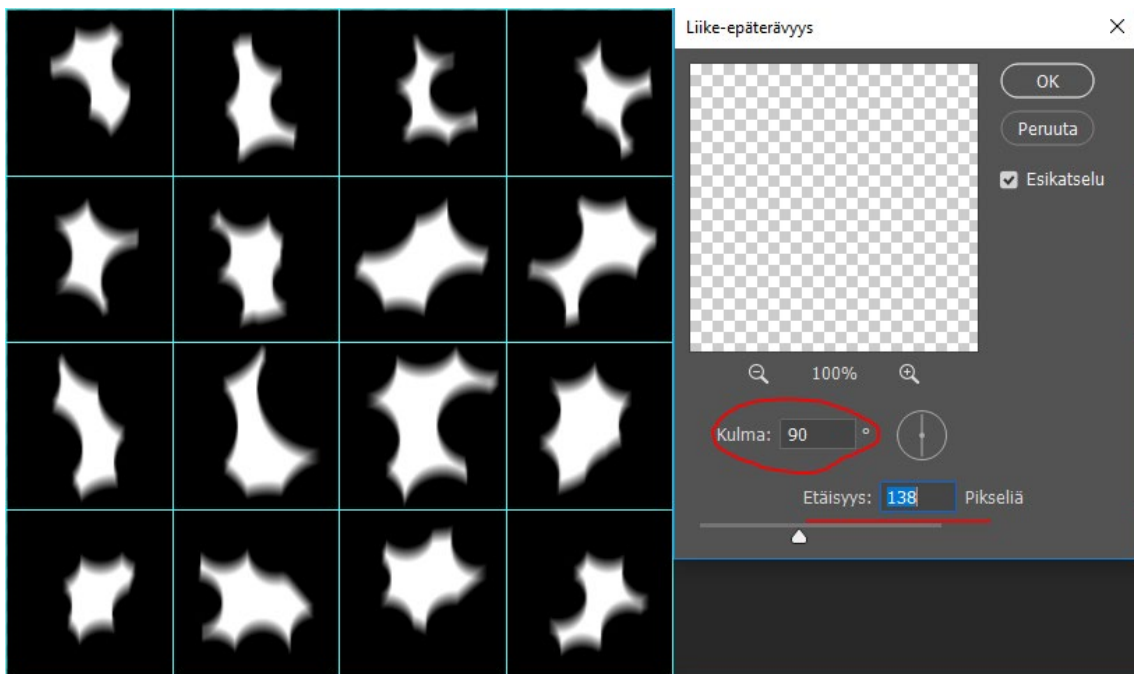
Kuva 18 Jokaisen laatikon sisälle on helppo maalata spritet

Tein uuden tason käyttämällä pikanäppäinyhdistelmää shift + ctrl + n. Seuraavaksi aloin piirtämään uudelle läpinäkyvälle tasolle spritejä käyttämällä pensseli- ja pyyhekumityökalua (katso kuva 19).



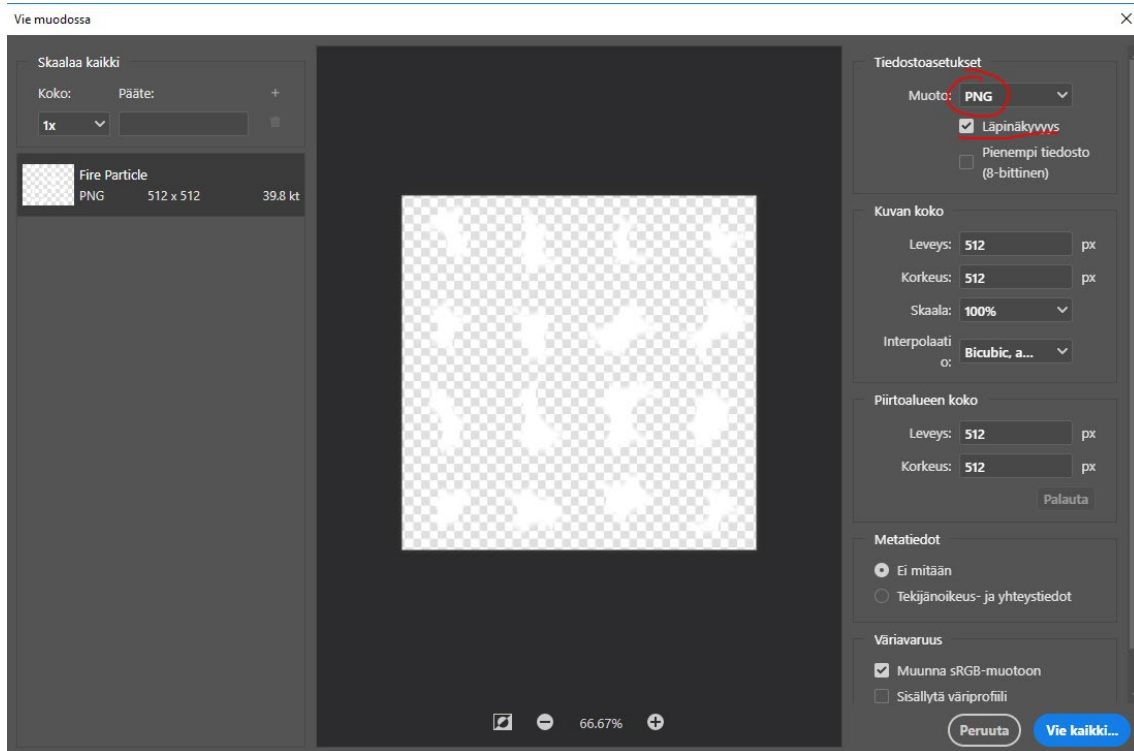
Kuva 19. Sprite sheet

Kun sain valmiiksi 16 erilaista spriteä, en halunnut niissä olevan kovin tarkat reunat sillä ne liikkuvat tehosteessa ylöspäin. Lisäsin spriteihin hieman epäterävyyttä käyttämällä liike-epäterävyys -suodatinta. Asetin kulman 90 asteeseen ja laitoin etäisyydeksi 138 pikseliä, että ne näyttävät liikkuvan ylös tai alas (katso kuva 20).



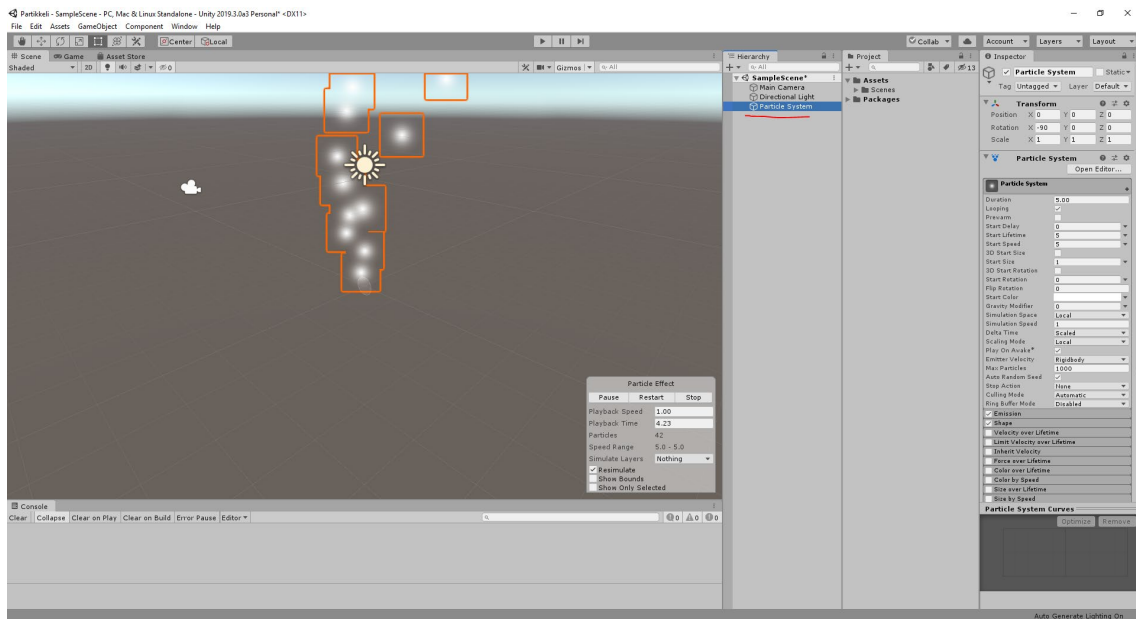
Kuva 20. Liike-epäterävyys -suodatin

Sprite sheetin pitää olla läpinäkyvä, joten otin vain muistan taustan pois näkyvistä painamalla tason vieressä olevaa silmäikonia. Apuviivat eivät näy tallennetussa kuvassa automaattisesti. Tallensin kuvan läpinäkyvänä PNG-tiedostona, ettei taustalle tule mitään taustaväriä (katso kuva 21). Se haittaisi tehosteen luomisessa.



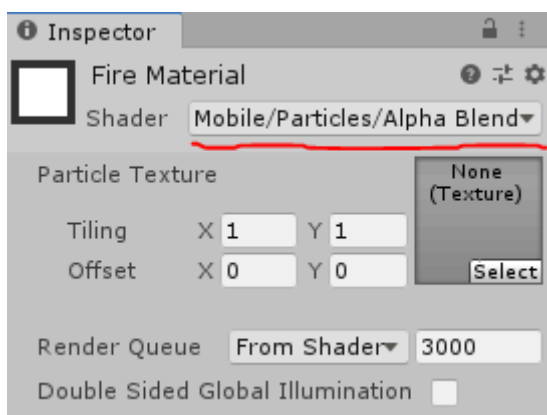
Kuva 21. Sprite sheetin tallentaminen

Nyt kun minulla oli sprite sheet valmiina, loin uuden tiedoston Unity3D -ohjelmalla. Tein "Parent" nimisen tehosteen alle tulee itse tehosteen kaikki komponentit kuten Fire yms. (katso kuva 22).



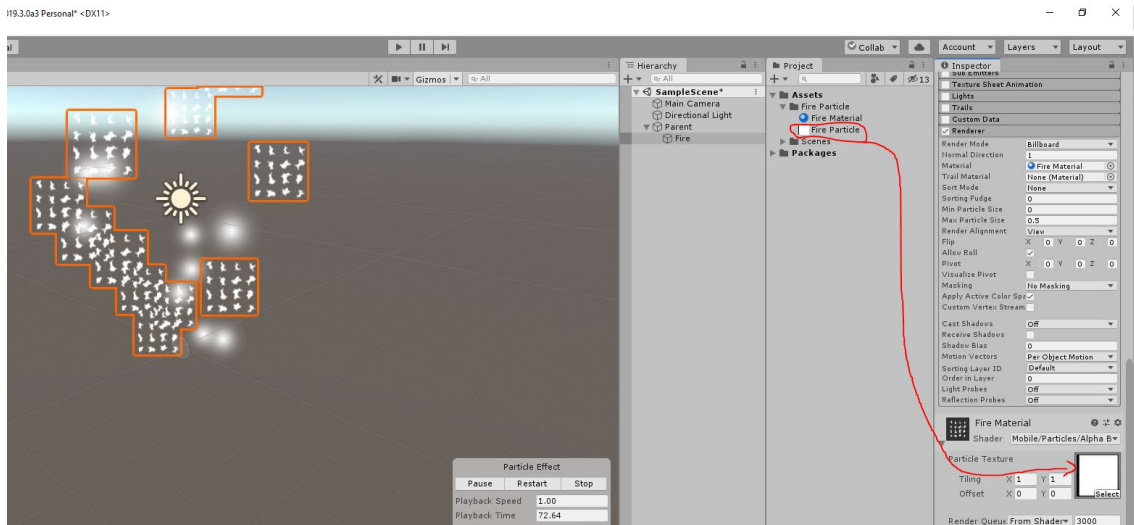
Kuva 22. Unityn oletus partikkelitehoste

Jotta tiedostot pidettäisiin järjestyksessä, kannatta jokaiselle elementille tehdä oma kansionsa. Tein tehosteelleni kansion, jonne tein materiaalin. Materiaalin shaderiksi laitoin Alpha Blend, että voin muokata sen väriä ja läpinäkyvyyttä (katso kuva 23).



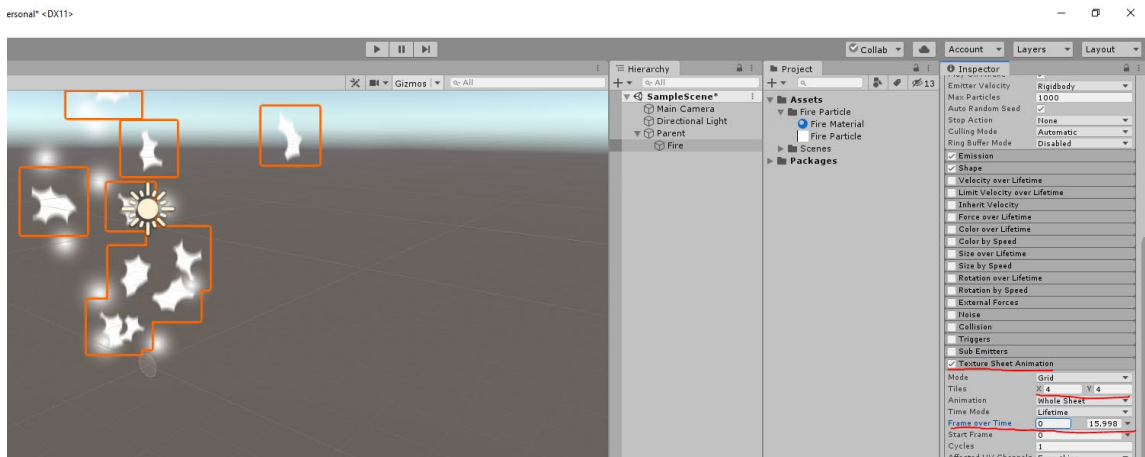
Kuva 23. Materiaalin asetukset

Kun tyhjän materiaalin laittaa render-kohtaan, renderöi järjestelmä vain valkoisen neliön. Materiaalin tekstuuriksi laitoin tekemäni sprite sheetin (katso kuva 24).



Kuva 24. Sprite sheet asetetaan materiaalin tekstuuriksi

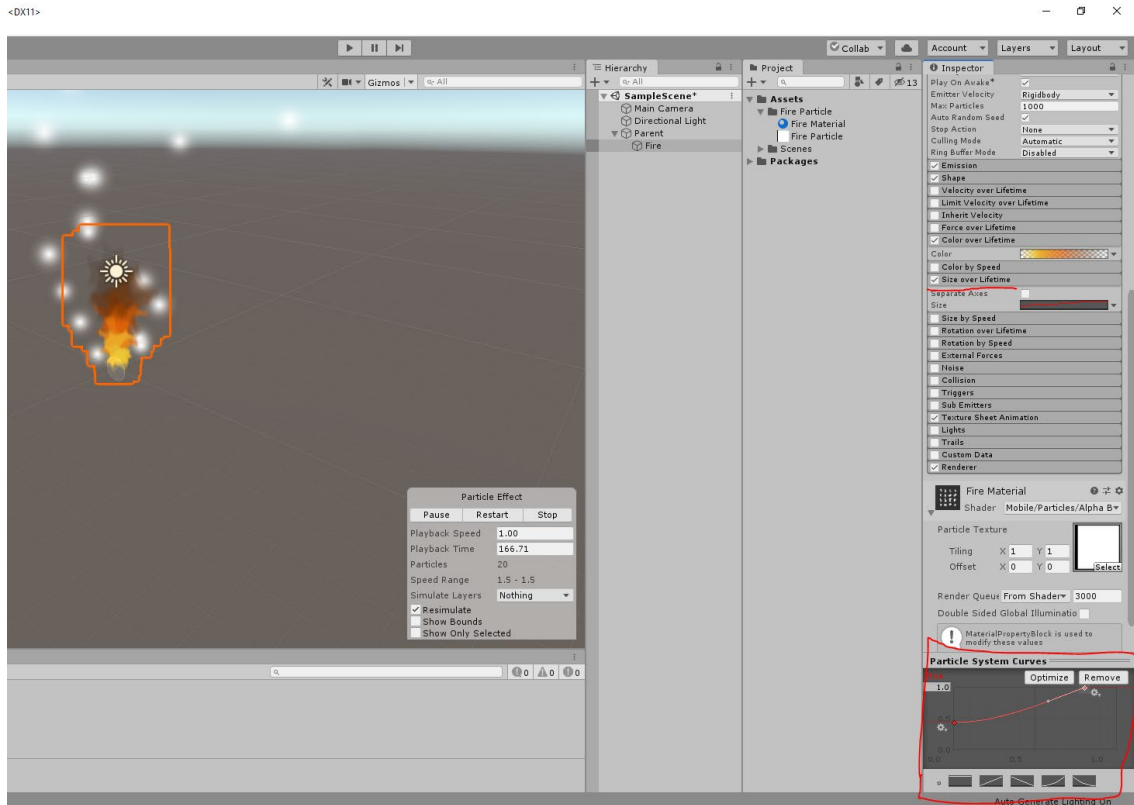
Ongelmana seuraavaksi oli, että partikkelit olivat nyt koko sprite sheet, eikä yksi sprite sheetin kuvista (katso kuva 24). Tämän sain korjattua laittamalla Texture Sheet Animation -toiminnon päälle. Ja koska spritejä on sprite sheetillä 16, laitetaan tiles-kohtaan X ja Y arvoiksi neljä. Nyt partikkelit vaihtuivat, mutta kaikki vaihtuivat yhtä aikaa. Tämä korjattiin laittamalla Frame over Time -kohtaan, että se vaihtaa spritejä satunnaisesti 0 ja 16 väliltä (katso kuva 25).



Kuva 25. Texture Sheet Animation

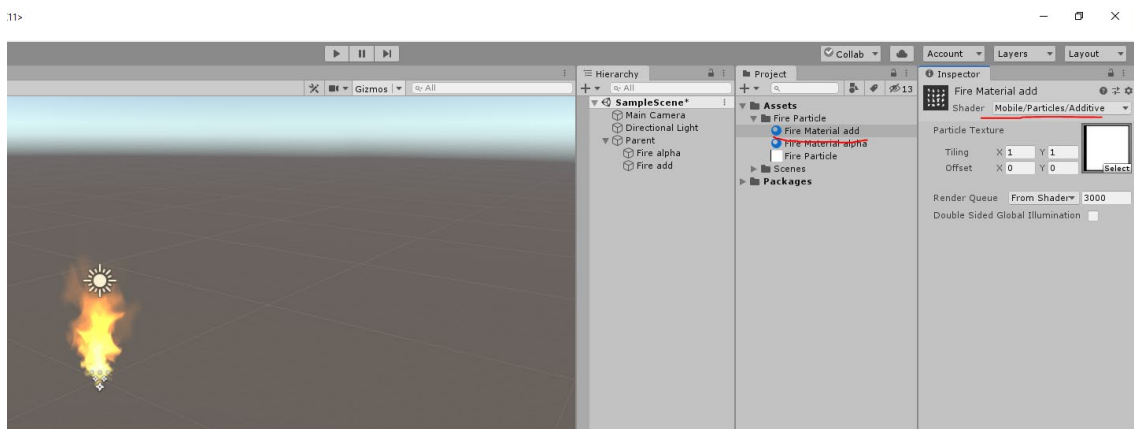
Seuraavaksi säädin tehoston aloituskokoa ja määritin partikkelin koon suurenemaan koko partikkelin eliniän ajan. Aloituskoko määrittää sen, että minkä kokoisena partikkeli ilmestyy kohtaukseen. Suurensin myös partikkelin elinikää, ettei se häviä heti. Tämän jälkeen muutin emitterin muotoa, että sain partikkelit nousemaan tulen lailla. Ja koska tein sprite sheetin valkoisella, on sen väri helppo vaihtaa unityn partikkelitehosteasetuksista. Laitoin värin ensin alkamaan läpinäkyvänä, joka

luo fade-tehosteen tulelle (katso kuva 26). Tämän jälkeen väri alkaa kirkkaan keltaisena ja käy elinikänsä aikana vielä oranssin, punaisen sekä mustan, jonka jälkeen se muuttuu läpinäkymättömäksi.



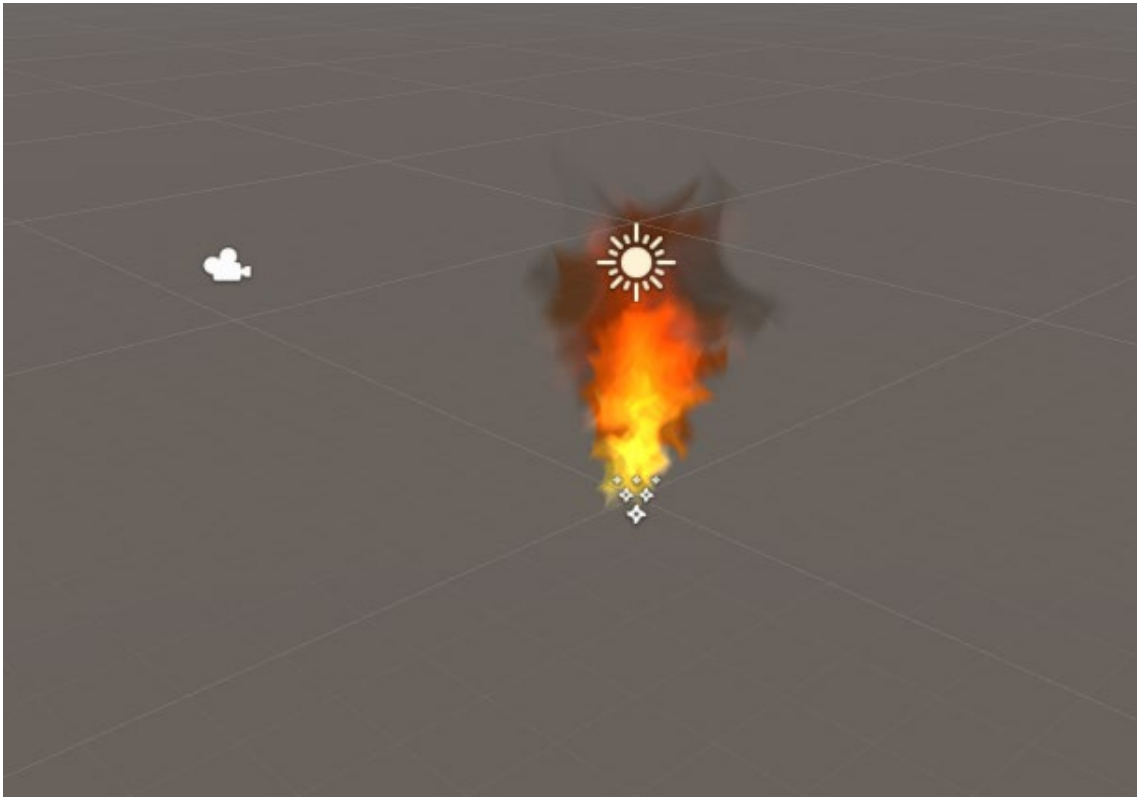
Kuva 26. Käyttämällä käyrää sain tehosteen koon alkamaan pienenä ja päättymään suurena

Kun tuli näytti hyvältä, tein vielä niille lisä tehosteen. Lisätehoste oli käytännössä kopio tulitehosteesta, vain hieman isompi ja läpinäkyvämpi. Käytin kuitenkin lisätehosteen materiaaliin Additive shaderia, että sain siitä kirkkaamman näköisen (katso kuva 27).



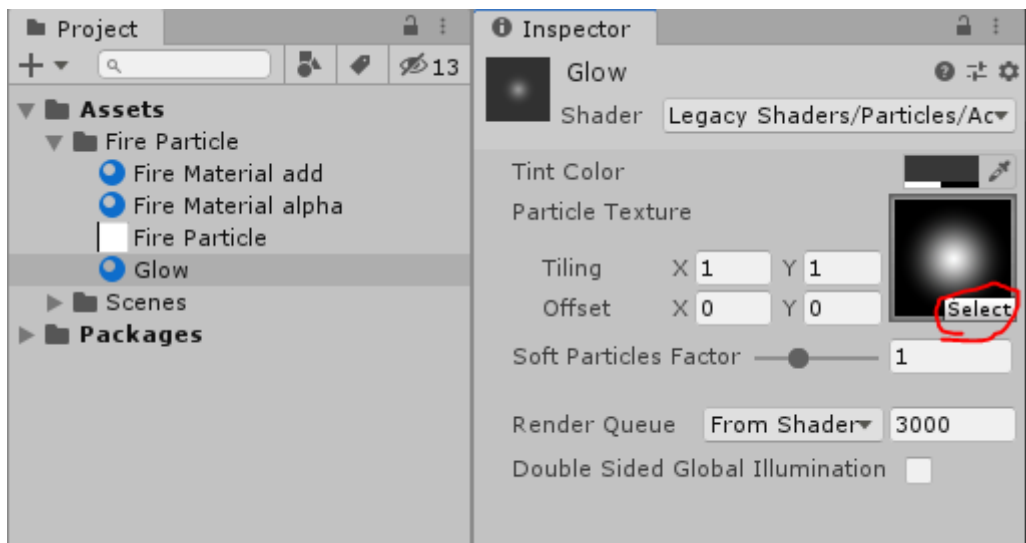
Kuva 27. Kirkas liekki

Himmensin hieman lisätehosteen väriä, että sain savun näyttämään läpinäkyvämmältä. Itse tulitehoste oli nyt valmis (katso kuva 28).



Kuva 28. Tulitehoste

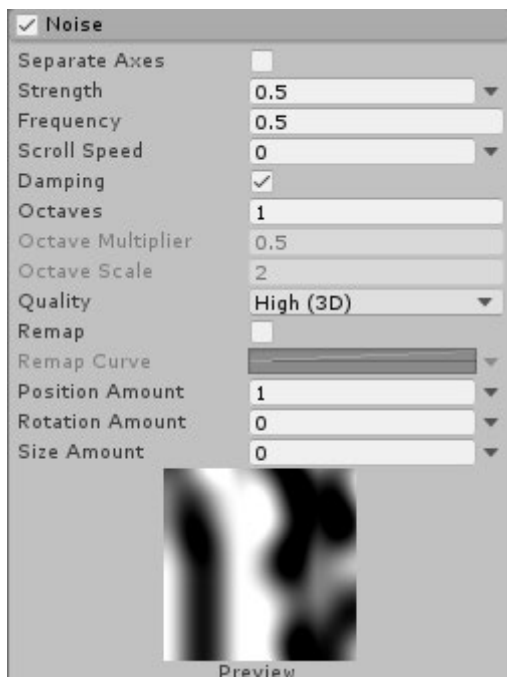
Halusin tehosteeseen hieman hohtoa, että se näyttäisi hieman järkevämmältä. Tein sille oman partikkelitehosteen ja materiaalin. Materiaaliin käytin Unityn Default-Particle -tekstuuria, joka luo vain valokeilan (katso kuva 29).



Kuva 29. Unityn Default-Particle -tekstuurin

Tein valokeilasta kirkkaan ja hyvin läpinäkyvän tehoston käyttämällä samoja menetelmiä, mitä käytin tulen tekemiseen. Tuli näytti nyt hyvältä ja oli aika lisätä siihen kipinöitä. Tein kipinätehosteelle taas oman partikkelitehosteen ja materiaalin. Materiaalin tekstuuriksi laitoin aikaisemmin tu-
leen käyttämäni sprite sheetin. Asetin emitterin lähettämään paljon pieniä tekstuureita jatkuvalla syötöllä.

Koska kipinät vaan kulkivat suoraan ylöspäin, oli hyvä kokeilla Unityn Noise moduulia, joka lisää partikkeleihin turbulenssia. Näin luodaan illuusio, että kipinät kulkisivat savun mukana ympäriinsä (katso kuva 30).



Kuva 30. Noise module

Hioin tehostetta vielä jonkin verran, että sain kipinäpartikkelit näyttämään paremmilta (katso kuva 31). Tämän jälkeen koko tehoste oli valmis ja olin siihen tyytyväinen.



Kuva 31. Valmis tulitehoste

4 POHDINTA

Olin todella kiinnostunut aiheesta, jota Spawn Point osuuskunta minulle ehdotti opinnäytetyötäni varten. Olen tyytyväinen opinnäytetyöhön, sillä partikkelitehosteet tehosteet olivat minulle täysin uusi asia. Pääsin harjoittelemaan ja työstämään tehosteita mahtavan työporukan ja työympäristön parissa.

Projektin tehosteet olivat suhteellisen haastavia suunnitella, sillä minulla ei ollut aikaisempaa kokemusta niistä. Hyvällä listauksella ja tiimityöskentelyllä niistä saatiin peliin sopivia. Parastahan tässä oli se, että voin laittaa tehosteet portfoliooni. Tunnen oppineeni paljon pelialasta ja pelinkehityksestä. Pelin tehosteita tehdessä kävi selville, että miten yhteen pieneen peliin voikaan mahtua niin monta erilaista tehostetta. Kaikkia tehosteita ei onneksi tarvinnut tehdä ihan alusta asti, sillä pystyin uudelleenkäyttämään ja muokkaamaan jo valmiita tehosteita.

Opittuja asioita oli Unity3D -pelimoottorin partikkelijärjestelmän osaaminen sekä erilaisten tekstuurien, kuten animoitujen sekä staattisten tehosteiden tekeminen. Vaikka sainkin aikaan hyviä tehosteita, mielestäni osa tehosteista kuitenkin jäi niin sanotusti hieman karkeiksi reunojen ympäriltä kokemattomuuteni takia ja niitä olisi voinut vielä hioa jonkin verran.

Projekti saatiin siihen pisteeseen, että peli julkaistiin Steam-palvelussa Early Access -pelinä. Peli on saanut hyvän vastaanoton sen ostaneilta.

Tämä projekti on ollut minulle mahtava kokemus, josta olen oppinut paljon erikoistehosteiden tekemisestä ja sen työnkulusta. Eniten opin partikkelitehosteiden suunnittelusta ja toteuttamisesta sekä niiden asettamisesta peliin oikeille paikoille. Minulle tarjottiin mahdollisuutta jatkaa projektissa ja jos siitä palkkaa saa, niin todellakin jatkan. Aion aktiivisesti seurata pelialan tarjoamia työmahdollisuuksia sekä uusia mahdollisuuksia oppia. Olen todella kiitollinen tästä mahdollisuudesta kehittyä alalla, jossa minulla on vielä paljon kasvettavaa.

Koska peli on vielä Early Access -pelinä, joudutaan siihen vielä tekemään paljon töitä, kuten uusia vihollisia, tehosteita ja hahmoluokkia. Tavoitteena on saada peli kokonaan valmiiksi ja ehkä jopa ansaita sillä hieman.

LÄHTEET

Chan, C. 2017. How The Special Effects Of Silent Films Were Done. Viitattu 23.9.2019, <https://www.gizmodo.com.au/2017/01/how-the-special-effects-of-silent-films-were-done/>.

Choudhary, A. 2018. Why is Adobe Photoshop the best photography software? Viitattu 22.9.2019, <https://www.lcca.org.uk/blog/education/why-is-adobe-photoshop-the-best-photography-software/>.

Clicklabs. 8 Reasons Why Unity Game Engine is the Best. Viitattu 22.9.2019, <http://click-labs.com/8-reasons-unity-game-engine-best/>.

Dube, R. 2018. What Is CGI Animation? Viitattu 22.9.2019, <https://www.makeuseof.com/tag/technology-explained-what-is-cgi-animation/>.

Film reference, Special effects. Viitattu 11.12.2018, <http://www.filmreference.com/encyclopedia/Romantic-Comedy-Yugoslavia/Special-Effects-OPTICAL-EFFECTS.html>.

Film110, Special Effects. Viitattu 11.12.2018, <http://film110.pbworks.com/w/page/12610294/Special%20Effects>.

KatsBits, Making better textures for games, 'power of two' and proper image dimensions. Viitattu 30.6.2019, <http://www.katsbits.com/tutorials/textures/make-better-textures-correct-size-and-power-of-two.php>.

Njiri, M. 2016. The History Of Video Game Animation (1958-2011). Viitattu 22.9.2019, <https://techinfographics.com/the-history-of-video-game-animation-1958-2011/>.

Photoshop Essentials, Understanding Layers In Photoshop. Viitattu 22.9.2019, <https://www.photoshopessentials.com/basics/understanding-photoshop-layers/>.

Saraiva, C. 2017. Explosion + sprite sheet. Viitattu 22.9.2019, <https://www.artstation.com/artwork/8VKKQ>.

ScienceDaily, Computer-generated imagery. Viitattu 11.12.2018, https://www.sciencedaily.com/terms/computer-generated_imagery.htm.

Torkev, K. 2017. VFX Production for AAA Video Games. Viitattu 29.6.2019, <https://80.lv/articles/the-secrets-of-vfx-production-for-doom/>.

Unity, What is a Particle System? Viitattu 11.12.2018, <https://docs.unity3d.com/Manual/PartSysWhatIs.html>.

Unreal Engine, Particle Effects. Viitattu 11.12.2018, <https://docs.unrealengine.com/en-us/Resources/Showcases/Effects>.

Williams, R. 2001. The Animator's Survival Kit. Lontoo: Faber and Faber Limited.