

Opinnäytetyö (AMK)

Insinööri (AMK), Tieto- ja viestintäteknikka

Pelitekniikka

2016

Markus Norrgran

2D-ANIMAATIOOTEKNIikka PELEISSÄ

Markus Norrgran

2D-ANIMAATIOTEKNIikka PELEISSÄ

Opinnäytetyön projektiosuus pohjautuu kahteen projektiin: Turku Game Labin Sanalankaan ja Ikoniac Oy:n Galactic Echoes -pelin introanimaatioon. Sanalanka on osa Tekesin rahoittamaa Fast WOW -projektia, jonka lopputuloksena on tuotteita tutkimus- ja yritysikäyttöön. Tuotteena Sanalanka on logopedinen työkalu lasten puheterapiaan. Opinnäytetyön kirjoittaja on toteuttanut Sanalangan animaatiot ja ison osan graafisesta sisällöstä sekä modifioinut käsikirjoitusta peliyhteensopivaan muotoon. Ikoniac Oy:n Galactic Echoesin introanimaatio on kokonaan opinnäytetyön kirjoittajan toteuttama. Ongelmana oli se, että renderoitujen animaatioiden koko ei ole mobiiliyhteensopiva. Oli etsittävä keinoja ja työkaluja, joilla minuutin mittaisesta animaatiosta sai kevyemmän niin, että tuotannon laatu säilyisi erinomaisena.

Toteutuksessa käytettiin Esoteric Softwaren Spine -animaatiotyökalua. Animaatioiden grafiikat toteutettiin Adobe Illustratorilla ja Photoshopilla. Animaatiotyössä sovellettiin luuteknikoita, sprite swapia, skinnausta ja muuttujia, jotka ovat vuorovaikuttavia pelimoottorin kanssa. Molemmissa projekteissa pelimoottorina oli Unity.

Ongelmana oli mobiilituotteen rajalliset tilaresurssit, jotka vaativat erilaista lähestymissuuntaa kuin renderoitu animaatio tai kuvasekvenssianimaatio. Luuanimaatio, joka renderoitui pelin käydessä, vastasi ongelmaan hyvin. Tekniikka on vähän käytetty muutamaa sekuntia pidempien välianimaatioiden toteutuksessa, vaikka se optimoi animaation tilankäyttöä erinomaisesti verrattuna huonolaatuiseenkin renderoituun animaatioon.

Teoriaosiossa käsitellään laadukkaan 2D-pelianimaation toteutusta ja laadukkaan jäljen tuottamiseen tarkoitettuja keinoja niin pelisuunnittelun kuin grafiikan toteutuksenkin kannalta. Lähdemateriaalina on modernia aineistoa pääasiassa viiden viime vuoden ajalta. Opinnäytetyön teoriaosuudessa käsitellään niin luuanimaatiota kuin kuvasekvenssianimaatiotakin.

ASIASANAT:

Pelianimaatio, peliala, peligrafiikka

Markus Norrgran

2D ANIMATION TECHNOLOGY IN GAMES

The thesis is based on two projects. First one is TGLs Sanalanka, and the second one is intro animation for Ikoniacs Galactic Echoes. Sanalanka is part of Fast WOW project, which is funded by Tekes. Results from the Fast WOW projects are research applications and pilot software projects for companies. Sanalanka is logopedic therapy tool for children with difficulties to learn their first language. Writer of the thesis has executed animations for Sanalanka, and major part of graphical content. Intro animation for Galactic Echoes is also animated by writer of this thesis.

In both cases, the problem was the size of animations. Both applications are for mobile platforms, so the size of rendered animations were too large. Better ways to animate long cut scenes for mobile platforms, were searched. The priority was, that quality of animations would be good with selected technology.

Esoteric Software's Spine were the main animation tool for both of the projects. Graphics were made with Adobe Illustrator and Adobe Photoshop. Animations technique, which were used, was bone animations with sprite swap, skinning and mesh warp. In both projects, the game engine was Unity.

The problem were mobile projects with limited space resources that requires different approach than rendered animations or image sequence animations. Runtime, which repeated bone animation within the game engine, responded to the problem very well.

In theory chapters the topic is quality 2D animations in modern game development.

KEYWORDS:

Game Animation, Games, Game Graphics

SISÄLTÖ

1 JOHDANTO	1
2 2D-PELIANIMAATIOIDEN TUOTANTOMETODIT	2
2.1 Esituotanto	3
2.2 Luuanimaatio	3
2.3 Kuvasekvenssianimaatio	5
3 PELIANIMAATION TOTEUTUS	9
3.1 Hahmoanimaation toteutus ja tehokeinot	10
3.2 Liikkeen tehokeinot Mariel Cartwrightin ja 12 Principles of Animationin mukaan	10
3.3 Partikkelipilvet	12
3.4 Aktiivisten elementtien ja objektien animaatiot	13
3.5 Passiivisten objektien ja taustan animaatiot	17
3.6 Pelien välianimaatiot	19
4 SPINE-ANIMAATIO PELIMOOTTORISSA	23
4.1 Polygon	23
4.2 Event	24
4.3 Skinnaus (skin)	24
4.4 Animaatioiden toimitus pelimoottoriin	24
5 PROJEKTIT SANALANKA JA GALACTIC ECHOES	26
5.1 Galactic Echoes -introanimaatio	26
5.2 Sanalangan peli- ja mikroanimaatiot sekä introanimaatio	31
5.3 Projektien tulokset	36
6 LOPPUPÄÄTELMÄ	37
LÄHTEET	38

KUVIEN OTSIKOT

Kuva 1 Luuanimaatiotyökalu Spine, jossa animaatiohahmo on valmisteltu animaatiota varten (Markus Norrgran 2015, TIGERIA).	4
Kuva 2 Pelimoottorivalmis atlas kuvan 1 hahmosta (Norrgran 2015).	5
Kuva 3 Osa kuvasekvenssianimaatiota samasta animaatiosta kuin kuvan 1 hahmo ja kuvan 2 atlas.	6
Kuva 4 Pieni peikkoanimaatio Photoshopin aikajanalla (Norrgran 2016)	7
Kuva 5 Spinen animaatioiden vientityökalu.	8
Kuva 6 Unityn animatorissa tehty ennakointianimaatiojärjestys (Anticipation).	11
Kuva 7 Unityn partikkelieditori, sekä Spinessä määritelty partikkelien syntyalue.	13
Kuva 8 Vartalon liikkeen suunnat ja lihas- sekä niveltyytit ryhmiteltyinä (Norrgran, 2015).	14
Kuva 9 Pelissä Ori and the Blind Forest niin lavastus kuin hahmotkin ovat graafisia ja kauniita. (Moon Studios, 2015).	17
Kuva 10 Sly Cooper tasohyppelypelisarja käyttää välianimaatioinaan laadukkaita, täysanimoituja videoita.	20
Kuva 11 The Witcher 3 - Wild Hunt -pelin välianimaatio (CD Projekt Red, 2015).	21
Kuva 12 Klein Don't Starven intro on lyhyt, tyylikäs ja toteutettu pelin sisäisin asetein.	21
Kuva 13 Final Fantasy IV ja taistelu Magus Sistersia vastaan (Square Enix, 1991).	22
Kuva 14 Unityn kuvasekvenssityökalu ja Spinen vientityökalu.	25
Kuva 15 Otos Galactic Echoes -pelin kohtauksesta Titanin pinta, missä kameralla on monimutkaisia ajoja.	27
Kuva 16 Kohtauksessa Hissi maan uumeniin, missä liikettä korostetaan ulkopuolisin objektein.	29
Kuva 17 Välianimaation luuston laajuus on valtava.	30
Kuva 18 Jääpyramidin sisällä.	30
Kuva 19 Sanalangan ensimmäisen kohtauksen vanha (vas.) ja uusi versio.	32
Kuva 20 Toinen kuva Sanalangan ympäristöanimaatioista.	33
Kuva 21 Papun juoksuanimaation Spinessä haamuavuin.	34

TAULUKOT

Taulukko 1 Nivelryhmät	15
Taulukko 2 Sanalangan hahmojen animaatiotaulukko.	34

SANASTO

Animaatio-ohjelma	Opinnäytetyössä sanaa animaatio-ohjelma käytetään kuvaamaan tarinallisia animaatioita erotuksena pelianimaatioista.
Assetit	Pelinkehityskontekstissa assetit tarkoittavat kaikkia objekteja pelissä. Tässä opinnäytetyössä termillä asset viitataan erityisesti graafisiin asetteihin ja animaatioasetteihin (Unity).
Mesh warp	Mesh warp on luuanimaatiotekniikka, jossa samaan grafiikkaan vaikutetaan useilla luilla. (Esoteric Software).
Renderointi	Renderointi tarkoittaa kuvan luomista raa'asta datasta matematiikan avulla. Opinnäytetyössä renderoinnilla viitataan erityisesti pelimoottorin toteuttamaan renderointiin, eli siihen, miten pelaaja näkee grafiikan pelissä. (Autodesk).
Runtime	Runtimet ovat pelimoottoriliitännäisiä, jotka pelianimaatiokontekstissa tarkoittavat liitännäisiä, joiden tehtävä on toistaa toteutettu animaatio pelimoottorissa. Runtimet ovat riippuvaisia siitä, millä ohjelmalla animaatio on toteutettu. (Esoteric Software).
Skinnaus	Skinning, luuanimaatiossa käytetty tekniikka, missä yhteen luurankoon liitetään useita grafiikoita. (Esoteric Software).
Slot ja region	Opinnäytetyössä slot ja region ovat Spine-animaatiotyökälussa esiintyviä työkaluja, joihin tallennetaan graafista data. (Esoteric Software).
Sprite Swap	Sprite swap on tekniikka, jossa luuhun kiinnitetty grafiikka vaihdetaan toiseen. Käytännössä se on eräänlainen kuva-sekvenssianimaatio luuanimaation sisällä. (Esoteric Software).

1 JOHDANTO

Työn aiheena oli pelianimaatiot. Sanalangan otin vastaan Turku Game Labilta. Työssä kehitettiin niin väli-, mikro- kuin pelianimaatioitakin tuotteeseen. Ikoniac Oy:n Galactic Echoes on mukana tuomassa laajempaa näkökulmaa pelianimaation tuottamiselle. Animaatiotyökaluksi valikoitui Esoteric Softwaren Spine sen saatavuuden ja opinnäytetyön tekijän osaamisen vuoksi. Animaatiotyökalun tuli sopia niin introanimaation- kuin pelin sisäistenkin animaatioiden tekoon. Suurin osa 2D-pelianimaatio-ohjelmistojen lisensseistä on kertaostoslisenssejä, joten ohjelmistoksi valittiin mieluiten sellainen ohjelmisto, joka pysyy ajan tasalla ja on käytettävissä myöhemmissäkin projekteissa.

Introanimaatioiden kohdalla ongelma oli se, että mobiililaitteissa kovalevyn tila on hyvin rajattua, eikä ollut mahdollista käyttää renderoitua animaatiota niiden koon vuoksi. Myös kuvasekvenssianimaatio tiedettiin haastavaksi ja raskaaksi käyttää välianimaatioissa. Mobiilituotteen ruutunopeus on yleensä 60 kuvaa sekunnissa, joten 16:9 -kuvakoon animaatio olisi hyvin raskas, sillä minuutin animaatioissa olisi tuhansia kuvia. Oli etsittävä muunkaltaisia ratkaisuja, kuten luuanimaatioita. Toiveena myös oli, että pelianimaatiot ovat helposti muokattavasti, jos tarvetta tulee. Kuvasekvenssianimaatiot eivät vastaa suoraa tähän toiveeseen. Näiltä pohjilta aloitettiin opinnäytetyön projektin toteutus.

Sanalangan haaste oli sen koko jo ilman animaatioita. Tiedettiin, että oli luotava animoituja objekteja ja elementtejä, joita voi kierrättää useissa pelin ruuduissa ja kohtauksissa, että kovalevytilaa säästyy.

Jo johdannossa voinee mainita, että opinnäytetyössä on myös paljon tekniikan käsitteilyä, eikä pelkästään animaatioprosesseja. Luuanimaatio, johon käytännönsuuden projektit perustuvat, on animaatiotapa, joka toimii peliulottuvuudessa hyvin syvässä yhteistyössä pelimoottorin ja koodin kanssa. Luuanimaatiotekniikoita hyödyntävälle pelianimaattorille on hyväksi tuntee myös projektissa käytettävää pelimoottoria omien graafisten työkalujensa lisäksi. Näin animaattori voi tuottaa paljon kiinnostavampaa ja monipuolisempaa animaatiota.

2 2D-PELIANIMAATIOIDEN TUOTANTOMETODIT

Pelianimaation vaatimukset ovat nousseet samanaikaisesti kun pelien sisältö on monipuolistunut. Varhaiset, grafiikkaa sisältäneet pelit olivat nopeita, yksinkertaisia mekaniikkoja toistavia pelejä, jotka perustuivat uudelleen yrittämiselle, testaamiselle ja nopeille pelisykleille. Samanlaisille asioille, mille monet nopeat mobiilipelit perustuvat. Vuosien mittaan peleille on tullut enemmän vaatimuksia ja valinnanvapauden ja liikkeen kokemuksesta on tullut tärkeä osa pelattavuutta. Pelaajalle ei enää riitä tasaisella nopeudella liikkuvat hahmot liikeradoilla, jotka eivät perustu todellisuuteen. Vaikka tämä vaatii enemmän animaattorilta kuin alkuaikojen yksinkertaiset pelit, toisesta näkökulmasta katsoen teknologiasta on tullut mahdollistaja sen sijaan, että se oli ennen rajoittaja. Mukaan on tullut myös teknisiä ominaisuuksia kuten hahmon fysiikka, pintojen liukkaus ja erilaisten objektien osuminen vaihtelevin tavoin pelaajahahmoon. Animaattorin onkin hyvä tuntea myös tekniikkaa. (Vince, 2012; Rogers 2010.)

Pääsääntöisesti hyvä pelianimaatio syntyy samoilla ohjenuorilla kuin hyvät animaatio-ohjelman animaatiot. Toteutustavoissa on kuitenkin eroa. Animaatio-ohjelman animaatiot suunnitellaan suoraan kuhunkin näytökseen käyttäen hyväksi animaation rytmiä, kuvakulmia, lavastusta ja laskennallisesti raskaitakin elementtejä. Peleissä pelaajan toiminta on vapaata, eikä pelaaja katso rajoituksia hyvällä. Peleissä pyritäänkin hälventämään rajoittuneisuuden tunnetta. Poikkeuksiakin toki on, kuten kauhupeleissä tapahtumien graafisuuden jättäminen kuuloaistin varaan rajoittamalla kuvakulmia tai vaikkapa interaktiivisen visuaalisen tarinan tarinankerrontaa korostavat keinot. Esimerkiksi Tecmon arvostettu kauhupelisarja Project Zeron pelihahmot ovat hitaita ja avuttomia yliluonnollisen ja todellisuuden välissä elävässä pelimaailmassaan, mutta silti pelisarjan pelit saavat jatkuvasti hyviä arvosanoja arvosteluissa (Adams, 2013).

Pelianimaation esitoteutukseen kuuluu kattava suunnittelu. Pelisuunnitteludokumentti (Game Design Document tai GDD) on tärkeä apuväline etenkin alkutuotannossa. Animaatio-osuuden kuuluu pitää sisällään kaikki tarvittavat animaatiot kategorisoituna kuhunkin animoitavan objektin tai hahmon oheen. Suunnitteludokumentissa määritellään objektin tai hahmon luonne ja seikat, jotka vaikuttavat olemukseen tai liikkeeseen. Esimerkiksi puujalka, liehuvat helmat tai raskas taakka voivat olla tällaisia asioita. (Adams, 2013).

Pelisuunnitteludokumentin animaatio-osuuden pohjalta lähdetään toteuttamaan varsinaista pelisisältöä. Tuotantoa tarkkailee Art Director muun graafisen laaduntarkkailun ohella. Suurissa tuotannoissa työstä vastaa pääanimaattori. (Rogers, 2010).

Pelianimaation toteuttamiseen on useita tapoja, joilla saadaan erilaista materiaalia ja eri käyttötarkoituksiin uskottavaa sisältöä. Kaikissa animaatiotavoissa on etunsa ja miinuspuolensa. Myös grafiikan tuottaminen erilaisiin animaatiotapoihin poikkeaa toisistaan. Animaatiotapoja käsitellään luvuissa 2.2 ja 2.3. Tuotantotapakin mainitaan erilaisissa pelisuunnitteludokumenteissa, mikäli se koetaan tarkoituksenmukaiseksi. Näin toimitaan erityisesti laajoissa tuotannoissa, joiden toteutus saattaa olla yli vuoden pituinen projekti. Tällaista pelisuunnitteludokumenttia kuvailee mm. Scott Rogers kirjassaan Level Up! The Guide to Great Video Game Design.

Pelissä animaatio voi olla monipuolista ja räiskyvää, mutta sitä sitoo samat säännöt kuin muutakin graafista toteutusta: se ei saa olla sekavaa, sillä silloin pelattavuus kärsii. Nyrkkisääntöjä ovat animaation rytmikkyys ja samansuuntaisuus. Myös animoitavan kohteen siluetin selkeyttä ja originaalisuutta korostetaan erityisesti pelianimaatioissa, jossa on paljon nopeita liikkeitä ja toimintaa. Tästä puhuu hyvin muun muassa Skullgirlsin animaatioiden toteutuksesta vastannut Mariel Cartwright (2014).

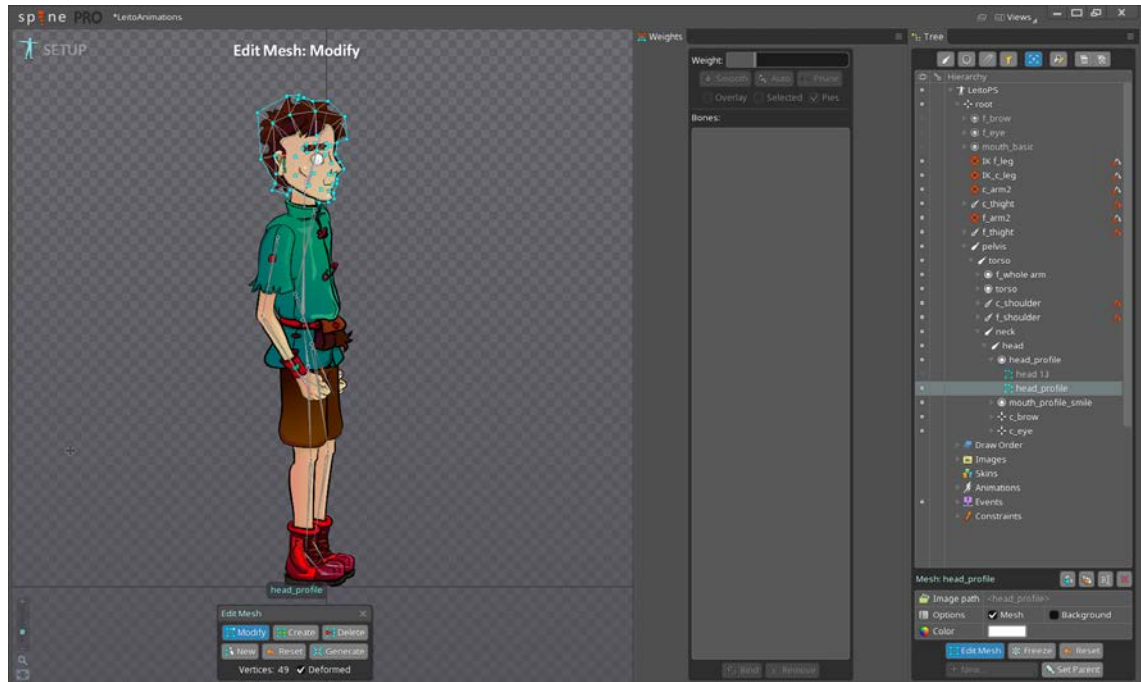
2.1 Esituotanto

Nykyään, teknologian mahdollistaessa pelianimaation toteutuksen rajoituksitta on lähettävä siitä, millainen on pelin haluttu tyyli, ja millaisia liikeratoja halutaan toteuttaa (John Vince 2014). Animaation 12 periaatteen 4. kohta määrittelee animaatiometodit suoraan toimintaan (Straight Ahead Action) ja asennosta asentoon (Pose to Pose) (Johnston & Thomas 1981). Näistä ensimmäinen sopii erinomaisesti abstrakteihin animaatioihin, jälkimmäisellä on hyvä toteuttaa olentojen ja objektien liikettä. Tekniikoita voidaan myös yhdistää. Esimerkiksi Suoraan toimintaan -metodia voi hyödyntää esimerkiksi asennosta asentoon -metodilla toteutetun hahmon hiuksiin. (Becker 2015.) Muita hyviä Suora toiminta -metodin animaatioita ovat esimerkiksi tuli, vesi ja kangas, eli ylipäätään sellaiset animaatiot, jotka liikkuvat jonkin verran ennustamattomaksi. Tarkempaa määritelmää on vaikea sanoa, sillä se, mikä grafiikka toimii minkäkin logiikan mukaan, päätetään pelisuunnittelussa ja saattaa vaihdella tuotannoittain (Rogers, 2010).

Esituotantovaiheessa suunnitellaan liikkeiden koko ja näkö. Siinä, missä vaikkapa kukan heilumista tuulella ei tarvitse hirveästi perustaa muulle kuin sille, että liike sopii tyyliin ja näyttää hyvältä, niin hahmon liikkeet, liikkeiden rata ja näyttävyys perustetaan sille, mitä pelikäsikirjoituksessa ja suunnitteludokumentissa sanotaan hahmosta. Vaihe voi sisältää ääriasetojen luonnoksia. (Cartwright 2014; Rogers 2010.)

2.2 Luuanimaatio

Luuanimaatioissa animoitavalle objektille tai hahmolle luodaan luuranko. Luihin sidotaan animoitavan kohteen grafiikat ja luita pyörittämällä ja liikuttamalla saadaan aikaan animaatiota. Tällainen työkalu on esimerkiksi Spine, jota on käytetty tämän opinnäytetyön projekteissa. Ensimmäisessä kuvassa on Spinessa valmisteltu hahmo. Kuvan hahmolle on asetettu luut, hahmon käsiin ja jalkoihin on toteutettu IK:t ja hahmon pää on valmisteltu mesh warpia varten.



Kuva 1 Luuanimaatiotyökalu Spine, jossa animaatiohahmo on valmisteltu animaatiota varten (Markus Norrgran 2015, TIGERIA).

Luurangon liiketieto tallennetaan tietokantamuotoon, esimerkiksi JSON (JavaScript Object Notation, joka nimestään huolimatta on kieliriippumaton datastruktuurimuoto), jota käyttää Spine-animaatio-ohjelmisto. Pelimoottorissa sitten kutsutaan tätä tietokantaan tallennettua dataa. Työkalu tähän tarkoitukseen on liitännäinen, josta käytetään termiä runtime. Runtime tarkoitus on toistaa animaatiotyökalussa toteutettu animaatio pelimoottorissa. Runtime on elementti, joka näyttää tärkeää osaa kaikissa luuanimaatiotyökaluissa. Toiset, jotka koskevat kaikkia luuanimaatiotyökaluja, ovat atlasit, joihin grafiikka tallennetaan. Runtime avulla kutsutaan grafiikkaa ja luodataa ja animaatio toistuu. (Kestrelmoon 2016, Esoteric Software 2016.)

Grafiikka-atlaksien koko on yleensä kahden potensseissa, sillä se optimoi pelin suorituskykyä. Tämä johtuu tietokoneiden datan käsittelykyvystä. Mikäli grafiikkakoko ei ole kahden potensseissa pikselileveydeltään ja -korkeudeltaan (64 px * 64 px, 2048 px * 1024 px jne.), pelimoottorin renderointi skaalaa kuvan automaattisesti kahden potenssiin. Tämä saattaa tuottaa huonolaatuista kuvaa, joka ilmenee sumuisuutena tai sahalaistoina, tai vähintäänkin kuva vie enemmän suorituskykyä tietokoneelta. Suorituskyvylle tai laadulle ei ole vaarallista, mikäli tuotteessa tai skenessä (scene) on muutamia virheellisesti tallennettuja kuvia, mutta kun niitä on useita (skene saattaa koostua tuhansista objekteista), suorituskykyongelmia syntyy (Katsbits, 2014).



Kuva 2 Pelimoottorivalmis atlas kuvan 1 hahmosta (Norrgran 2015).

Kuvassa 2 on TIGERIA -peeliin toteutettujen sivuanimaatioiden atlas. Sen koko on 241 kt ja .JSON on 49 kt. Kokonaisuus sijaintidatan kanssa on yhteensä 293 kt. Pelihahmon korkeus koottuna on 700 pikseliä, joten se voidaan piirtää peliruudulle melko suurena niin, että laatu säilyy. Atlas on kooltaan 2048 px * 1024 px, eli kahden potenssissa, joten suorituskyky pysyy hyvänä.

Kun sama hahmo pakattiin kuvasekvensseiksi (kuva 3), kaikkien sivuanimaatioiden koko oli yhteensä 12 Mb, mikä ei tietokonetuotteelle ole vielä paljoa, mutta mobiilialustoilla se olisi suuri. Applikaatiokaupoista voi ladata kerralla 50 Mb:n kokoisia asennuspaketteja ilman WiFiä ja 100 Mb:n kokoisia paketteja WiFi:n kanssa. Muulle pelin sisällölle ei jäisi hirveästi tilaa, ja loppujen objektien graafinen laatu olisi tilanpuutteen vuoksi oltava melko alhainen verrattuna hahmoon. Se saisi pelin näyttämään hiomattomalta. Tietenkin pelin kokoa voi kasvattaa päivityksin. Esimerkiksi iOS-laitteille voi päivitysten avulla ladata jopa 4 GB:n kokoisia applikaatioita (Apple Developer Guide, 2016).

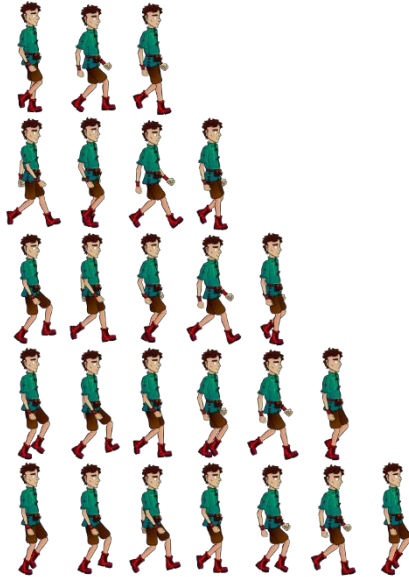
Luanimaatio-ohjelmat tarjoavat myös toimintoja, jotka mahdollistavat vuorovaikutuksen pelimoottorin kanssa, kuten IK-perusteiset polygonit ja skinnaukset, joiden avulla tavaroita tai grafiikoita voidaan helposti vaihtaa perustuen animoidun objektin eli sen slotien sijaintiin. (Esoteric Software 2016.)

2.3 Kuvasekvenssianimaatio

Kuvasekvenssianimaatio on vanha, mutta yhä validi tapa toteuttaa animaatioita. Skullgirlsin pääanimaattori Mariel Cartwright toteuttaa animaationsa kuvasekvenssein. Hän toteuttaa nopeatempoisia pelejä, joten kuvasekvenssit ovat nopeita ja tiiviitä. Ne sisältävät yleensä avainruudut, sekä muutaman väliruudun. (Cartwright 2014.)

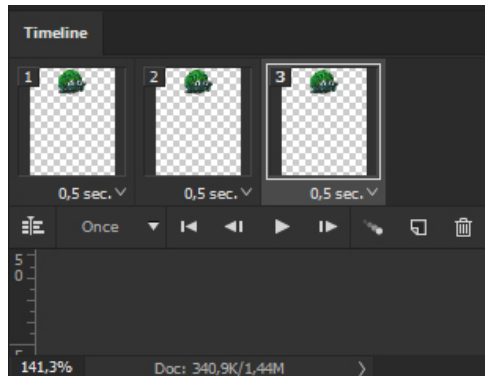
Pelianimaatioiden nopeuden vuoksi huomataan animaatoruutujen vähenneen vuosien mittaan. Samalla kuitenkin nähdään, että animaation siluetit ovat tehostuneet ja tulleet entisistä dynaamisemmiksi (vertaa Skullgirls ja MegaMan 8).

Kuvasekvenssianimaatio on siis yhä validi animaatiometodi, ja pelinkehityksen esituo-
tannossa onkin päätettävä, millainen metodi tuottaa parhaiten lopputuloksen, mikä vas-
taa design-dokumentissa osoitettua tyyliä ja mitä art director valvoo. Kuvassa 3 on ku-
vasekvenssianimaationa kävely kuvan 1 hahmosta. (Rogers 2010.)



Kuva 3 Osa kuvasekvenssianimaatiota samasta animaatiosta kuin kuvan 1 hahmo ja kuvan 2 atlas.

Kuvasarja-animaatiossa on teknisiä ongelmia kuten joustamattomuus, adaptoitumatto-
muus tietokoneen toistonopeuteen ja sen rytmittämisen vaikeus. Lisäksi erilaisia tvii-
nauksia on vaikea toteuttaa rytmittämään animaatiota. Tämä saattaa ilmetä esimerkiksi
grafiikan välkyntänä tai pätkimisenä. Tämä johtuu siitä, että pelimoottorit pyrkivät toista-
maan animaatiota vähintään 60 ruutua sekunnissa (nykyään monet laadukkaista peli-
näytöistä ovat 144 Hz, ja nopeampiakin on jonkin verran markkinoilla). Animaatio-ohjel-
missä taas ruutunopeus on 24 ruutua sekunnissa ja esimerkiksi 1950–1970 luvuilla to-
teutettiin paljon animaatioita jopa niin, että sekuntiin piirrettiin 12 kuvaa ja kutakin ku-
vaa näytettiin kahdesti. Niin kuin luvussa 1 todettiin, nykyisiin peleihin olisi kallista to-
teuttaa hyvännäköistä renderoitua- tai kuvasekvenssianimaatiota niin, että se toimisi
pelille normaalilla nopeudella ilman keinotekoisia rajoituksia. (Williams 2001; Unity Do-
cumentation 2016.)

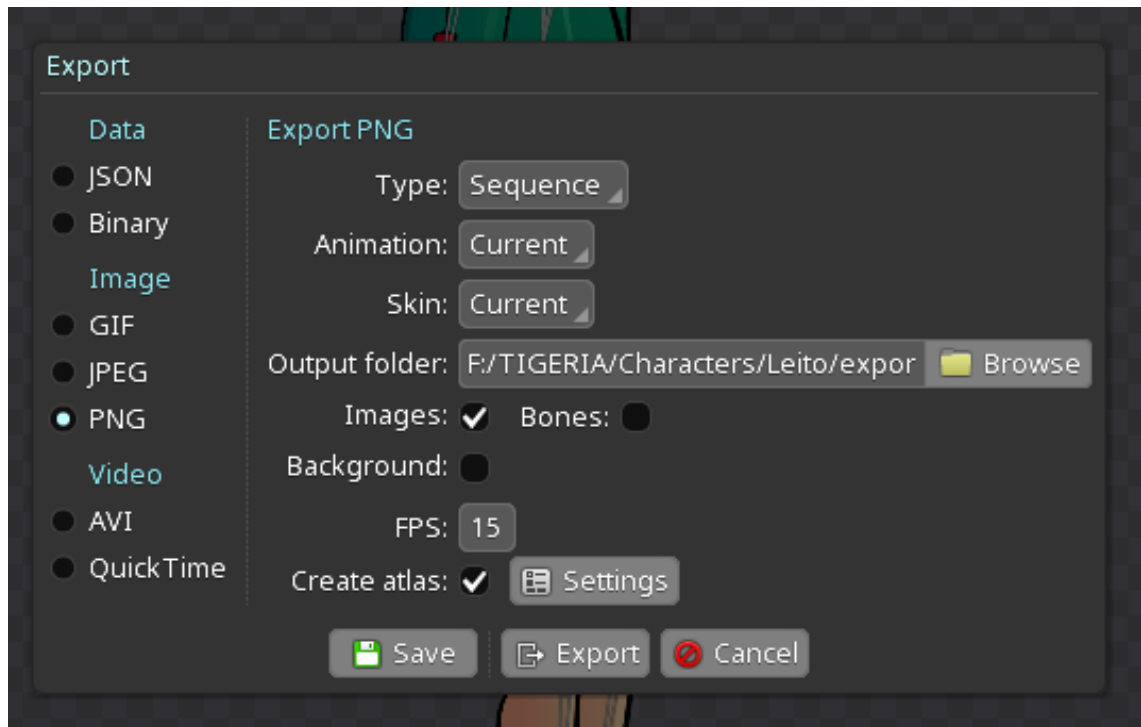


Kuva 4 Pieni peikkoanimaatio Photoshopin aikajanalla (Norrgran 2016)

Kuvasekvenssianimaatio toteutetaan yleensä piirtämällä ne tietokoneen piirto-ohjelmilla tai joskus jopa käsin. Myös luuanimaatiotyökaluista voi yleensä viedä myös kuvasekvenssianimaatioita, mikäli tuotannossa se jostain syystä koetaan hyödylliseksi. Esimerkiksi Adobe Photoshop tarjoaa tällaisen animaatiotavan avuksi aikajanan, johon voidaan toteuttaa ruutuanimaatio, jonka toistonopeuden voi säätää haluamukseen. Hyvä esimerkki tästä on kuvan 4 peikkoanimaatio. Aikajانات ovat hyviä aputyökaluja kuvasekvenssianimaatiota toteutettaessa.

Kuvassa 5 esitellään vuorostaan luuanimaatiotyökalu Spinen asetukset, joilla viedään kuvasekvenssianimaatio pelimoottorikäyttöön. Työkalulla määritellään, miten monta kuvaa kukin animaatio sisältää. Esimerkiksi kuvassa viedään pelimoottoriin kuvasekvenssianimaatiota, jonka ruutu nopeus on 15 kuvaa sekunnissa.

Kuvassa esitelty vientityökalu on kuitenkin epätarkempi kuin vaikkapa Photoshopissa ruuduittain piirretty animaatio. Tämä johtuu siitä, että työkalulla ei voi säätää tarkkoja hetkiä, joihin vientityökalu tallentaa yksittäiset ruudut. Piirtotyökalulla voi piirtää haluamansa määrän muutoksia ja kokeilla aikajanalla, miten tasalaatuisen jäljen piirtojalkei tuottaa. Animaatio-ohjelmien vientityökalut voivat näin ollen olla huomattavasti epätarkempia ja perustua enemmän kokeiluun kuin tietoon.



Kuva 5 Spinen animaatioiden vientityökalu.

Rotoskopia on myös tekniikka, mikä on hyödyllinen erityisesti kuvasekvenssianimaatioissa, mikäli haetaan käsitteitä perinteisestä animaatiosta. Rotoskopiassa artisti tai animaattori piirtää kuvattujen liikesarjojen päälle animaatiota ja näin saadaan uskottavasti toteutettua monimutkaisiakin animaatioita, kuten vaikka tanssimista. Tämä on tekniikka, mitä on perinteisesti toteutettu aina Disneyn alkuaajoista lähtien nykypäivän Studio Ghibliin. Tehostettuna se on hyödynnettävissä myös pelianimaatioissa (ks. Mariel Cartwrightilla lähteistetyt osiot). (Williams 2002.).

3 PELIANIMAATION TOTEUTUS

Pelien ensimmäinen prioriteetti on aina pelattavuus. Tuotteena pelin on oltava hauska ja hyvä jopa ilman grafiikkaa. Hyvä kokonaisuus taas muodostaa pelikokemuksen. Optimaalisessa tuotannossa kaikki sisältö, myös animaatiot, tukevat pelattavuutta. Animaatio ei koskaan saisi olla pelattavuuden esteenä. Mikäli pelaaja käyttää kaoottista valinnanmahdollisuuttaan hyväksi, esimerkiksi päättämällä liukumisen hyppyyn yllättävässä paikassa, animaatio ei saa hidastaa pelaajaa. Parhaimmillaan animaatio tukee mekaniikkaa. (Adams, 2013)

Peleissä, joissa on taistelua tai vuorovaikutusta tekoälyvastustajien kanssa, pelihahmolla on hyvä olla hieman vähemmän animaatoruutuja tai animaation kestoa (animaatiotavasta riippuen), jotta pelikokemus on mahdollisimman intuitiivinen. Esimerkiksi Skullgirlsissa pelaajan hahmolla on vahvat siluetit ääriasennoissa ja yhdestä kolmeen väliruutua. Näin pelaaja aistii väliruudut, vaikkei tietoisesti näe niitä, ja intuitiivisuus säilyy. Vihollista voi taas hieman hidastaa lisäämällä hieman animaation kestoa joko ajallisesti tai muutamalla välikuvalla. (Cartwright, 2014)

Pelianimaatiota toteutetaan samoin ehdoin kuin animaatio-ohjelmankin animaatiota. Laadukkaat animaatiot parantavat pelikokemusta siinä, missä hyvin animoitu animaatio-ohjelma on positiivisen katselukokemuksen kulmakivi. Merkittävänä erona on mainittu pelattavuuden tukeminen. John Vincen (2012) mukaan pelianimaation vaatimukset ovat lisääntyneet pelien ottaessa vaikutteista muiden medioiden kerronnasta. Animaation halutaan herättävän tunteita ja sitouttavan pelaajaa kokemukseen passiivisen viihteen tavoin. Pelien etu on interaktiivisuus, jonka takia pelaajalle voidaan kertoa yhä mielekkäämpiä tarinoita, joihin pelaaja pääsee itse vaikuttamaan ja vaikka kaksikulotteiselta grafiikalta ei odoteta samankaltaista realismia kuin kolmiulotteisilta tuotteilta (esimerkiksi Remedyn Quantum Break tai Rockstarin GTA V), niin kerronnan ilmeikkyyden odotetaan olevan vahvaa ja kiinnostavaa (Daedlic Entertainmentin Deponia -peli-sarja tai vaikkapa Rovion Angry Birds tarinaa syventävine pikkuanimaatioineen). Sen lisäksi tyyli saa olla liioiteltua, sarjakuvamaista, minkä onnistuvuutta voi seurata App Annien ja Steamien tilastoista.

Kuitenkin on asioita, joihin animaattorin on kiinnitettävä huomioita. Pelien hauskuuden tullessa interaktiivisuudesta monet animaatio-ohjelman mahdollisuudet ja temput draaman ja ulkoasun luomiseen ovat mahdottomia, mutta toisaalta ohjelmointi luo tukun uusia työkaluja animaattorin käyttöön. Tärkeimpiä eroja animaation ja animaatio-ohjelman välillä ovat muun muassa:

- Pelaajan aktiivinen vaikutus pelimaailman tapahtumiin ja siten animaatioihin omalla toiminnallaan.
- Toisin kuin animaatio-ohjelmia, pelejä ei renderoida, eli pelin on pyörittävä suoraan tietokoneen muistista.
- Pelihahmon kaoottinen muuttuvuus pelaajan mielihalujen mukaisesti (esimerkiksi pelaaja vaihtaa pelihahmon päällä olevia vaatteita, käsissä olevia tavaroita jne.).

Perinteisessä animaatiossa dramatisoidaan hahmon, olennon tai elementin luonnollisia liikkeitä niin, että ne pysyvät yhä uskottavina. Myös pelinkehityksessä tämä on hyvän

tavan mukaista. Tässä luvussa sovelletaan animaatio-ohjelmien tuotantokeinoja pelinkehitykseen, että saadaan kaapattua traditionaalisen animaation keinoja, joilla luodaan mielekkään fantasian tunnetta tuotteeseen. Lähdemateriaalina käytetään erityisesti Animator's Survival Kitiä ja Disney Animation: the Illusion of Life'n lukua 12 Principles of Animation (ks. liite 12 ohjenuoraa (12 Principles of Animation) (Williams, 2001; Johnston & Thomas, 1981).

3.1 Hahmoanimaation toteutus ja tehokeinot

Hahmovetoisissa peleissä pelaaja kokee pelielämyksensä pelihahmon kautta. Pelihahmon animaatioiden tulisi kuvantaa tunteita ja tunnelmia, mitä pelisuunnittelijat ja artistit ovat pelihahmoille kuhunkin tilanteeseen suunnitelleet.

Esimerkiksi jos hahmo on kuvattu härkäpäiseksi henkilöksi, jolla on huono itsehillintä, voisi hahmo suutuessaan hengittää koko vartalollaan kuin härkä ja olla hyökkäävästi etukenossa. Introvertti ja hiljainen hahmo voisi puolestaan tärinä suuttumuksesta ja nostaa hartioitaan ollessaan vihainen.

Hyvässä suunnitteludokumentissa on lista kaikista pelihahmoon tarvittavista animaatioista sekä hahmon kuvaus, joiden perusteella animaattori voi suunnitella ja toteuttaa uskottavan ja hyvän sarjan animaatioita.

Mikäli pelityyppi on sellainen, missä hahmo haavoittuu tai saa vaikkapa nopeutta kenttään ripotelluista erikoiskyvyistä, sen on hyvä antaa näkyä animaatioissa. Hahmo voi esimerkiksi menettää panssarinsa osia, kun vihollinen osuu häneen tai välkkyä kirkkaissa väreissä saadessaan lisävoimia. (Rogers 2010; Vince 2012.)

Tällaisia lisäominaisuuksia voidaan lisätä luuanimaatioon pelimoottorissa, mikä tukee varjostimia ja valoja kaksiuulotteisissa grafiikoissa. Esimerkiksi Unity (Unity 2016).

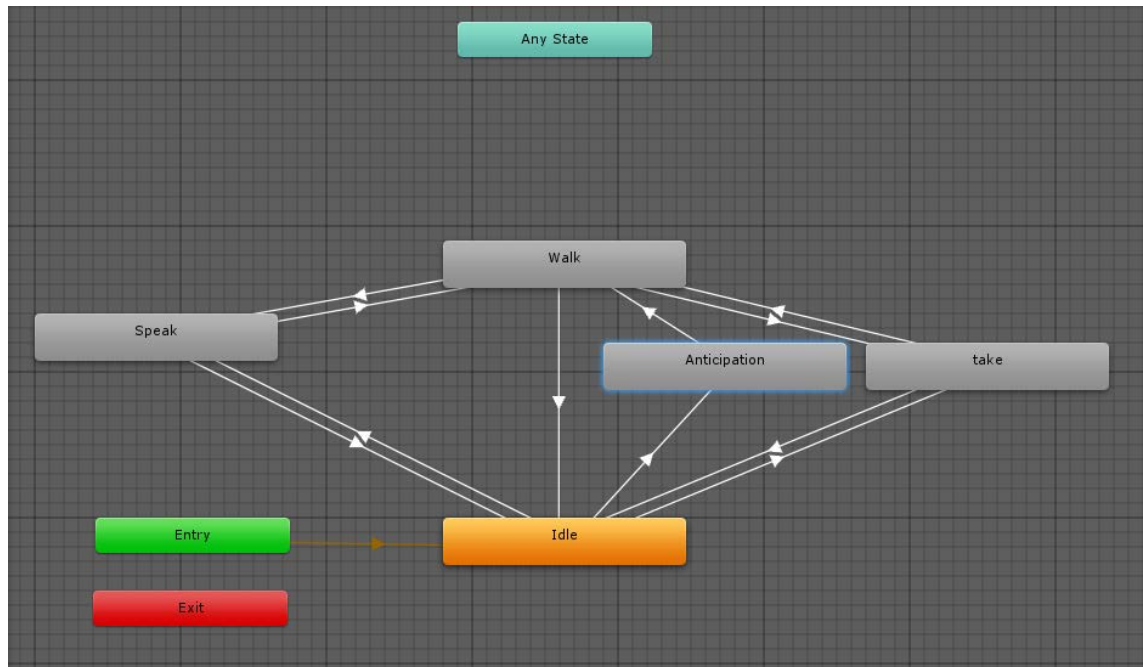
3.2 Liikkeen tehokeinot Mariel Cartwrightin ja 12 Principles of Animationin mukaan

Animaatio ei ole elokuvaa, vaan se on karakterisoitua ja tehokeinoin rikastettua toimintaa, mikä tekee animaatiosta erityisen. Yksi animaation periaatteista on litistys ja venytys, joka kattaa animaation tehosteet. Tässä luvussa haluan kuitenkin laajentaa ymmärrystä jakamalla periaatteen niin kuin Mariel Cartwright tekee vuoden 2014 GDC luentonsa Making Fluid and Powerful Animations For Skullgirls. Käytän pohjana myös Richard Williamsin the Animators Survival Kitiä (2002).

Ennakointi

Ennakointi on hyvin tavallinen efekti kaikenlaisessa animaatioissa. Se tarkoittaa sitä, että hahmo ottaa ikään kuin vauhtia. Esimerkiksi lähtiessään kävelemään hahmo koukistaa polviaan, ennen kuin ottaa ensimmäisen askeleen.

Ennakoinnin valmistelu on helpompaa tehokkaissa toiminnoissa ja sellaisissa kontrolleissa, jossa hahmon on tarkoitus aloittaa jokin nopea liike, kuten taistelupelissä lyönti. Animaattorin täytyy kuitenkin huomioida, että toimintapeleissä pelaajahahmon liikkeiden tulee olla nopeampia kuin vihollisten, jotta pelaaja kokee hallitsevansa hahmoaan. Vihollisen pitää antaa pelaajalle aikaa reagoida.



Kuva 6 Unityn animatorissa tehty ennakoitiantimaatiojärjestys (Anticipation).

Tavallisemmissa animaatioissa, kuten seikkailupelin kävelyanimaatioissa, ennakoitinta on lähestyttävä niin, että ennakoitinta on erillinen animaatio, joka toistetaan ennen liikkeellelähtöä. Kuvassa 6 on asetettu kävelyanimaation ennakoitinta paikalleen Unityn animatorissa yksinkertaiseen animaatiopuuhun. Tavallisen liikkeen ennakoinnilla peliin luodaan sujuvuutta ja uskottavuutta.

Liikkeen painotus avainasentoihin

Nopeissa, pelaajalle mahdollisimman intuitiivisissa liikkeissä tai liikesarjoissa ei ole aikaa monimutkaisille, liikkeen jatkuvuutta kuvaaville väliruuduille. Tämä tekee liikkeestä nopeasti reagoivaa, mutta äärimmäisen vahvaa. Mikäli liikehdintä jää liian heikoksi, pelaajan on vaikea käsittää liikettä.

Seuraava liike

Seuraava liike viittaa osittain animaation 4. periaatteeseen. (Johnston & Thomas 1981). Seuraava liike tarkoittaa nopean animaation avainkuvien yhdistämistä jollakin abstraktilla liikkeellä, kuten takinliepeen seuraamisella tai hiusten liehahduksella.

Tällaiset liikkeet ovat käteviä toteuttaa Suoraan toimintaan -animaatiometodilla, vaikka muutoin hahmo olisi avainkuvien perusteella animoitu.

Välikuvan epäselvyys

Tämä metodi koskee erityisesti välikuvaa. Liike ei saa loppua epäselvään viivaan, tai illuusio särkyä. Välikuvan suttuisuudella voidaan kuitenkin korostaa liikkeen nopeutta. Etenkin jos ruudunvaihto on nopeaa. Tällöin pelaaja ei varsinaisesti näe kuvan epäselvyyttä, vaan ainoastaan käsittää liikkeen nopeuden. Epäselvyyden on oltava liikkeen suuntaista

Yliampuus

Tämä on tavallaan ennakoimiselle käänteinen metodi, jonka avulla toteutetaan liikkeelle terävyyttä. Käytän taas esimerkkinä taistelupelien lyöntiä. Tämä voi tarkoittaa sitä, että toiseksi viimeisessä kuvassa hahmon sormet tai nyrkki kurottuu hieman pidemmälle kuin se, mihin isku viimeisessä kuvassa loppuu. Tämä saa liikkeen tuntumaan voimakkaalta.

Riko fysiikka

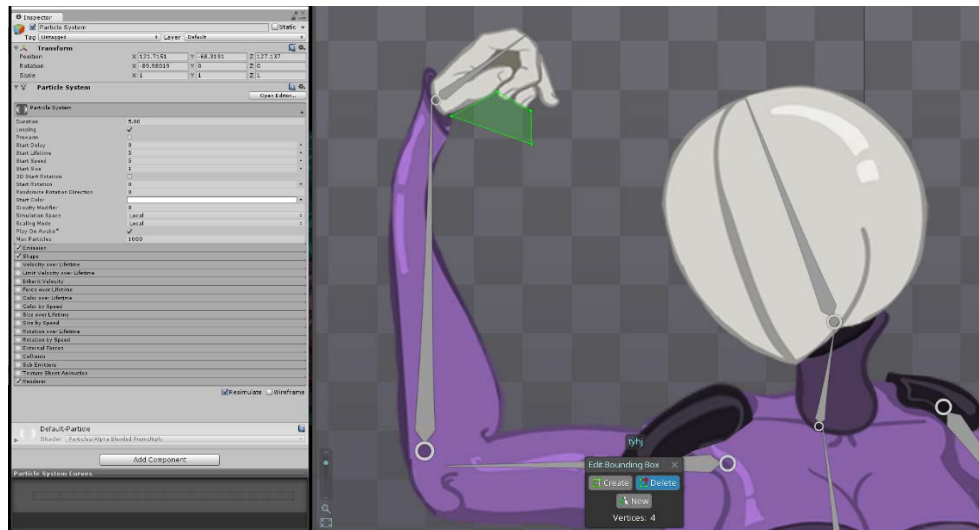
Mikäli liike on pitkä tai sisältää vahvoja muutoksia, animaattori saa rikkoa fysiikan. On kuitenkin pidettävä huoli siitä, ettei liike lopu rikkiin fysiikkaan, sillä se rikkoisi hahmon kokonaisuuden ja ehjän piirtämisen illusion.

3.3 Partikkelipilvet

Pelitekniikassa partikkelit (particles) tarkoittavat matemaattisesti toteutettuja pilviä, joiden grafiikkana toimii hyvin pienet grafiikkaelementit jotka liikkuvat niille syötetyn matemaattisen kaavan mukaan. Partikkelipilven muotoa, nopeutta, kiertoa tai vaikkapa väriskaalaa käsitellään

Nykyisissä pelimoottoreissa tähän käyttöön on monipuolisia graafisia liittymiä joissa halutun laisia partikkelipilviä voi luoda kätevästi, mikäli osaa käyttää editoria. Myös jotkut pelianimaatio- ohjelmat tarjoavat työkaluja joilla voidaan animoida ja luoda partikkelipilvien nollapisteitä, eli kohtia, mistä partikkelipilvi syntyy. Opinnäytetyön käytännönosiossa käytettävä Spine -animaatiotyökalu tarjoaa mahdollisuuden luoda

partikkelipilvien syntyypisteitä. Tällainen on luotu kuvassa 7 hahmon käteen. Samalla työkalulla luodaan muun muassa törmäysalueita.



Kuva 7 Unityn partikkelieditori, sekä Spinessä määritelty partikkelien syntyalue.

Esimerkkejä partikkelipilvien mahdollisista käyttötarkoituksista ovat abstraktit ilmiöt kuten tuli, savu, veden virtaus, sade ja lasersäde. Partikkelipilviä käytetään usein myös tärkeiden kohtien korostamiseen, heijastusnäytön erottamiseen pelimaailmasta (HUD tai heads-up display eli valikko, jossa osoitetaan pelaajan tietoja, kuten terveyttä tai eri-koiskyvyn palautumisaikaa). (Unity 2016; Rogers 2010)

3.4 Aktiivisten elementtien ja objektien animaatiot

Opinnäytetyössä hahmoanimaatio käsittää kaikkien aktiivisten objektien animaatiota kuten pelaajahahmoa, autoja, pelin vihollisia, ympäristön vuorovaikutteisia tekijöitä ja niin edelleen. Käytännössä kaikki objektit, jotka eivät kuulu taustaan ja lavastukseen kuuluvat hahmoanimaation piiriin.

Hahmoanimaatiot ovat peleissä tärkeässä osassa, ja niiden väliseen toimintaan perustuu pelin vuorovaikutus.

Ihminen samaistuu parhaiten ihmismäisesti toimiviin hahmoihin. Tästä syystä eksoottiset ja fantasiaolennot sovitetaan peleissä humanoidin luurankoon. Ihmissilmää kiehtoo vieraus tuttuudessa ja tätä tietoa on syytä käyttää hyväksi hahmojen luonnissa (Rogers 2010).

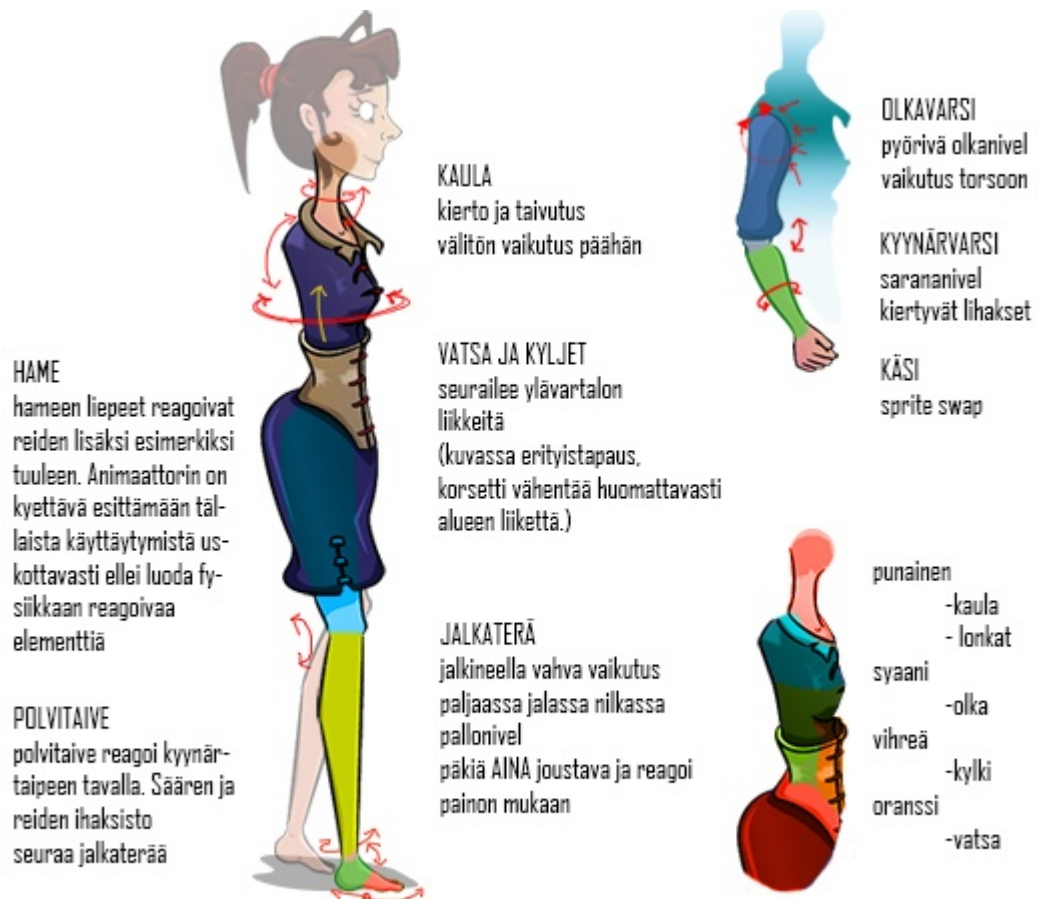
Sujuva animaatio vaatii anatomian keskeisten pointtien kuten lihasryhmien ja luuston ymmärtämistä, että animaatio olisi uskottavaa. Tietenkään tämä ei vaadi lääkäritason tai edes kuvataiteilijatasen yksityiskohtaista pienten lihasten, luiden ja jänteiden ymmärtämistä vaan nivelten ja lihasryhmien liikkeen liikesuuntien sisäistämistä. Tähän kuuluu myös käsitys siitä miten vartalo reagoi erilaisissa toimintatilanteissa, kuten juoksussa, kävelyssä ja hypyssä. Mikäli tällaisia asioita ei esitetä oikein, pelaaja käsittää, että jokin on pielessä.

On tärkeää myös tunnistaa, miten hahmon vaatteiden materiaalit reagoivat hahmon toimintaa. Sillä on paljon vaikutusta, piirretäänkö hahmoille istuvat, jämäkstä kankaasta olevat varusteet vai jotain kevyttä, mikä tarttuu helposti vaikkapa tuuleen.

Mikäli edellä mainittuja seikkoja osaa tarkastella jo hahmon luonnissa, pystyy jo suunnitteluvaiheessa ottamaan kantaa, miten paljon resursseja jonkin asian toteuttaminen vie. Tämä on hyödyllistä tuotantokustannusten kannalta. (Rogers 2010; Ukko 2015)

Orgaanisten hahmojen rakenne

aikka animaatioissa hahmon liikeradat ovat melko vapaita ja tehokeinoja saa käyttää paljon – fysiikkaa jopa rikkoa – pitää animoitavan hahmon päätyä uskottavaan kokonaisuuteen. Uskottavuus ei tarkoita luurangon ja fysiikan realistista rakentamista vaan sitä, että esimerkiksi nivelet ja niiden alla liikkuvat lihakset toimivat tutulla tavalla (ks. luku 3.3). Kuvassa 8 On ihmishahmon lihas- ja nivelryhmät selvitettyä Tigeria -pelin hahmossa.



Kuva 8 Vartalon liikkeen suunnat ja lihas- sekä niveltyypit ryhmiteltynä (Norrgran, 2015).

Monissa animaatio-ohjelmistoissa on liitännäisiä erilaisia hahmoluurankoja, jotka sisältävät tiedon perusliikkeistä. Tämä on erittäin hyödyllistä luotaessa hahmoja väliaikaisiin tarkoituksiin (placeholder), jotka uusitaan myöhemmin varsinaisilla hahmoilla.

Karikatuyrissa hahmogallerioissa ei voi nojata animaatiotyökalujen valmiiksi generoituihin luurankoihin, vaan animaattorin on luotava itse omansa. Tällöin on hyvä tutustua eri luuryhmien toimintaperiaatteisiin. Jenó Barkcsayn Anatomia, opas taiteilijoille -kirjassa (2004) nivelet jaetaan toimintaperiaatteeltaan Taulukko 1:n ryhmiin.

Taulukko 1 Nivelryhmät

Niveltyypit	Esimerkkisijainti
Laakanivel	Ranne- ja nilkkanivelet
Sarananivel	esim. polvet, kyynärpäät
Ratasnivel	värttinän ja kyynärän kiertävät liikkeet
Pallonivel	Lonkka- ja olkanivelet
Satulanivel	esim. peukalo

Laakanivelet

Laakanivelten liike voi olla hyvin kolmiulotteista (käden ja jalkaterän kierrot esimerkiksi käännyttäessä). Jos kääntymisanimaatiot ovat rauhallisia, nilkan laakanivelen kierto kannattaa huomioida. Laakanivel on käytännössä kaksi luulevyä, jotka kiertyvät toistensa ympäri.

Sarananivelet

Sarananivelet ovat yksinkertaisimpia käsittää ja ne harvoin vaativat suurta huomiota. Niiden toiminta muistuttaa saranan toimintaa ja niiden kierto on vähäistä. Avattaessa sarananivelet myös lukittuvat tiettyyn kohtaan liikekaarta ja tämän lukkopisteen tunnistaminen onkin tärkeintä sarananivelten toimintaa animoidessa. Tärkeitä tällaisia niveliä ovat polvet ja kyynärpäät.

Pallonivelet

Olkanivelet ja lonkat ovat animaattorille haastavimpia niveliä. Olkanivel liikuttaa laajoja alueita ylävartalosta ja liikkuu melko vapaita liikeratoja. Animaatiossa olkavarren liikkeiden vaikutusta ylävartaloon ja kylkiin demonstroidaan harvoin anatomisesti oikein. Joskus olkavarsi ja torso eivät lainkaan vaikuta toisiinsa. Tämä on hyvä tehostekeino ja sopii moniin tuotantoihin, mutta olkanivelen todellista toimintaa on syytä tutkia.

Lonkat ovat samankaltaisia animaation kannalta, mutta niiden liiketaso on horisontaalinen. Lonkkanivelten toiminnan tunnistaminen korostuu esimerkiksi uskottavissa pysähdyksissä juoksuvauhdista tai vaikkapa kääntymisanimaatioissa.

Satulanivel

Satulanivel on melko vapaasti liikkuva nivel ikään kuin satulana luiden päällä. Kaksiulotteisissa animaatioissa siihen pitää kiinnittää huomiota, jos animaattori käyttää tarkkoja lähikuvia käsistä.

Nivelryhmien tunnistaminen on hyödyllistä, sillä niiden avulla käsittää paremmin liikkeen mahdollisuudet, joita voi osaamisen kanssa rikkoa. Kun animaattori tunnistaa nivelet, työ myös tehostuu.

Mesh Warp

Kaikki tuotannollisesti merkittävät luuanimaatio-ohjelmat tarjoavat mesh warpia työkaluna (mm. Spine, Spriter ja Dragon Bones). Mesh warpin avulla hahmoihin saadaan syvyyttä luomalla verteksiverkko grafiikan ympärille. Luilla vaikutetaan verteksiverkon solmupisteisiin ja näin venytetään grafiikka. Tällä tavalla huijataan katsojan silmää ja näkymä vaikuttaa kolmiulotteiselta. Tekniikalla voi luoda monipuolisia animaatioita.

Luvun 3.2 animaatiotapoja seurattaessa mesh warp on usein hyödyllinen työkalu. Esimerkiksi kohta Seuraava liike. (Esoteric Software 2016).

Neli- ja monijalkaiset olennot

Neli- ja monijalkaisten olentojen ilmiselvän ongelman lisäksi (poikkeava fysiikka) on neli- ja monijalkaisilla hahmoilla muitakin selkeitä eroja humanoideihin. Nelijalkaisten olentojen animaatiot vievät huomattavasti enemmän tilaa pelialueelta kuin kaksijalkaiset humanoidiolennot. Alueet, jossa neliraajaisia olentoja liikkuu, suunnitellaan niiden vaatiman pinta-alan mukaan. Liian ahdas liikkumatila ei näytä kiinnostavalta tai ole edes hauskaa. Tämä korostuu, jos pelaajahahmo on nelijalkainen olento.

Nelijalkaiset olennot ovat käynniltään matalampia kuin humanoidit jos tarkkaillaan suunnilleen yhtä pitkiä olentoja. Tämä tarkoittaa sitä, että interaktiiviset objektit pelikentässä on sijoitettava niin, että neli- tai moniraajainen olento voi toimia niiden kanssa mielekkäästi. Animaatiosuunnittelussa pitää päättää, että onko interaktiivisten objektien sijoittelu suunniteltu humanoidi- vai nelijalkaisten olentojen mukaan. Tällä on merkitystä vaikutelmaan. Mikäli interaktiivisten objektien sijoittelu toteutetaan humanoidien mittasuhteisiin, luodaan vierauden tunnetta. Neli- tai moniraajaisen olennon mukaan suunniteltu interaktiivisuus taas luo tuttuuden tunteen, jos ympäristö on esimerkiksi metsä.

Kolmas seikka on jo mainittu fysiikka. Neli- ja monijalkaiset olennot ovat usein verrattain vahvempia kuin humanoidit. Nelijalkaisuudesta on etua nopeuden ja lihaksiston ra-

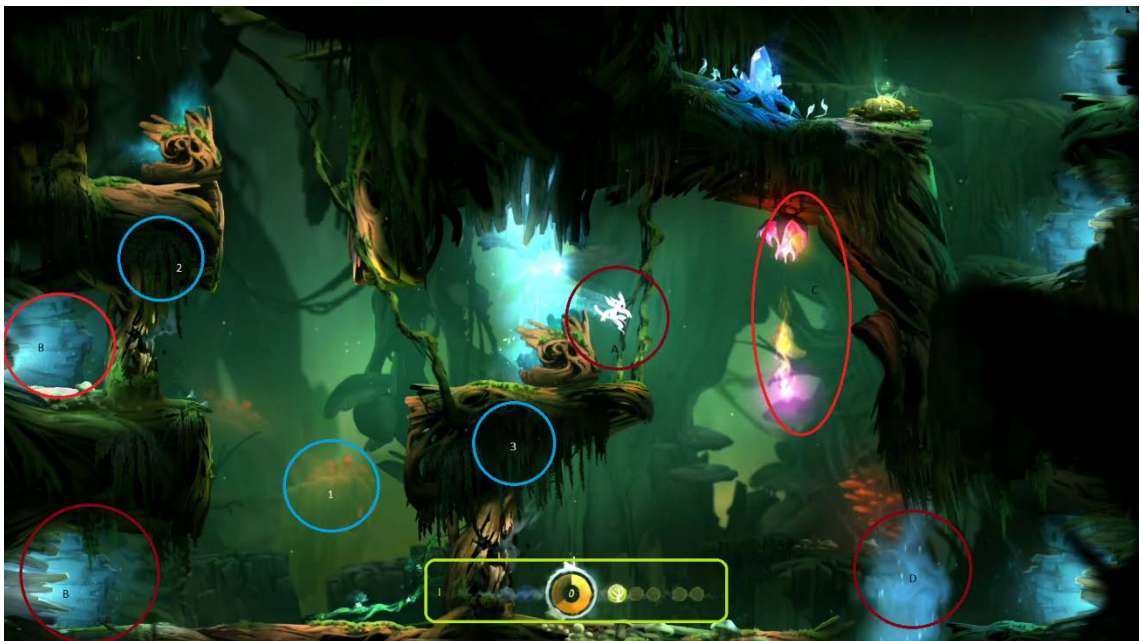
kenteen suhteen. Monet neli- ja monijalkaiset olennot käyttävät paljo hampaitaan työkaluina ja niiden kaula on muotoutunut siten, että se kestää paljon enemmän painetta. Tämä näkyy usein olentojen rakenteessa ja toissijaisissa liikkeissä, mikä on tärkeää animaation kannalta.

Nelijalkaisten olentojen uskottava tuntuma on erilainen kuin humanoidien. Nelijalkaiset olennot ovat usein nopeampia ja voimakkaampia (esimerkiksi ne voivat hypätä pidemmälle). Lisäksi ne käyttävät enemmän suutaan toimintaan kuin vaikkapa etukäpäliä. (Rogers 2010, 109.)

3.5 Passiivisten objektien ja taustan animaatiot

Kirjassa *12 Principles of Animation* määritellään yhdeksi animaation periaateksi lavastus (3 – Staging), mikä tarkoittaa kaikkea sitä, mikä ympäröi kohdetta, jonka kuuluu olla katsojan mielenkiinnon kohteena katsotussa kohtauksessa. Esimerkiksi Mikki Hiiri löytää jonkin jännittävän esineen. Tällöin lavastus tulee rakentaa niin, että värit ja muodot korostavat tilanteen jännittävyyttä ja intensiivisyyttä eivätkä vie huomiota varsinaiselta asialta. (Johnston & Thomas 1981.)

Peleissä tilanne tarkoittaa pelihahmoa tai jotain ympäristön mielenkiintoista kohdetta, kuten vaikka vipua, mitä kääntämällä ovi aukeaa.



Kuva 9 Pelissä *Ori and the Blind Forest* niin lavastus kuin hahmotkin ovat graafisia ja kauniita. (Moon Studios, 2015).

1. Tausta ja parallaksi

Parallaksilla ei pelitekniikassa ole paljoakaan tekemistä varsinaisen matemaattisen laskutavan kanssa. Pelitekniikassa parallaksilla (parallax) tarkoitetaan taustan sy-

vyysvaikutelmaa, joka syntyy pelihahmoon nähden eri nopeuksilla liikkuvista taustan osasista. Taustan osien syvyysvaikutelmaa voidaan vahvistaa väriteknisin keinoin kuten vähentämällä yleistä värikylläisyyttä ja usvaisemmalla piirtojäljellä sen mukaan, mitä etäämmällä kamerasta ollaan.

2. Lavasteen yksinkertainen animaatio

Lavasteet, eli taustan osaset joiden tehtävä on ainoastaan elävöittää maailmaa, voivat sisältää pientä liikettä kuten huojumista ja toimia vaikkapa leijailevien partikkeleiden keskipisteenä. Samoja lavasteita voidaan käyttää useissa paikoissa, elleivät ne ole liian originaaleja ja huomiota herättäviä. Nämä palaset eivät saisi herättää liikaa huomiota, jotta ne eivät tee ruudun toiminnasta sekavaa ja vie huomiota olennaisesta.

3. Tasanne (myös maa kuuluu tähän)

Tasanne tarkoittaa passiivista maastonosaa, jonka päällä hahmo voi seistä. Tasanteessa voi olla törmäyksen tunnistin, mikä estää pelaajaa hyppäämästä sen läpi. Tasanteessa voi olla myös yksisuuntainen törmäyksen tunnistus, mikä estää pelaajaa putoamasta tason läpi.

Tasanne on samalla etäisyydellä kamerasta kuin hahmokin. Tämä pitää muistaa, kun toteuttaa taustan yksityiskohtaisuutta ja väritystä. Pääsääntöisesti yksityiskohtaisuuden ja värikylläisyyden tulisi olla samaa luokkaa hahmon kanssa.

a. Pelihahmo

Hahmo ei tietenkään kuulu taustaan tai passiivisiin objekteihin, mutta pelihahmoa vasten peilataan tausta ja passiiviset objektit. Pelihahmo on se vertailukohta, minkä mukaan määritellään taustan ja passiivisten objektien koko, värikylläisyys, animaatioiden muoto ja niin edelleen. Pelihahmon on oltava mukana kunkin osasen suunnittelussa.

b. Kulkureitit

Kulkutiet on osoitettu sinisin partikkelein ja valoin.

c. Aktiivinen toimija tai vihollinen

Tässä kohtauksessa aktiivisena toimijana on kukka, joka sylkee punaisia ammuksia jotka läiskähtävät sinisiksi partikkelipilviksi osuessaan kohteeseen – maahan tai pelaajaan.

d. Partikkeleita

Partikkeleilla on monenlaisia käyttötarkoituksia. Tässä kohtauksessa partikkelipilvillä on korostettu kuvaruudun tärkeitä kohtia. Ympyröidyssä kohdassa ne osoittavat kohtaa, mihin viholliskukan ammuksiset hajoavat. Kuvaruudussa niitä on käytetty myös muun muassa kulkureittien korostuksessa valojen ohella ja näkymättömämmin taustan elävöityksessä ja tunnelman luonnissa.

I. Heijastusruutu eli HUD

Heijastusruutu on erotettu varsinaisesta pelimaailmasta erilaisen väripaletinsa ja sitä ympäröivän voimakkaanvärisen partikkelipilven avulla. Näin siitä on tehty erillinen, pelaajakokemukseen vaikuttava elementti josta näkee samalla selvästi, ettei se kuulu varsinaiseen pelimaailmaan. Nykyiset heijastusruudut on tehty hienovaraisiksi. Pelihistoriassa ne ovat usein olleet raskaita palkkeja ruudun alareunassa (kuten vanhoissa räiskintäpeleissä kuten DOOM tai Hex ja yhä se on käytössä sellaisissa roolipeleissä kuin Diablo ja Path of Exile). (Rogers 2010; Ukko 2014)

3.6 Pelien välianimaatiot

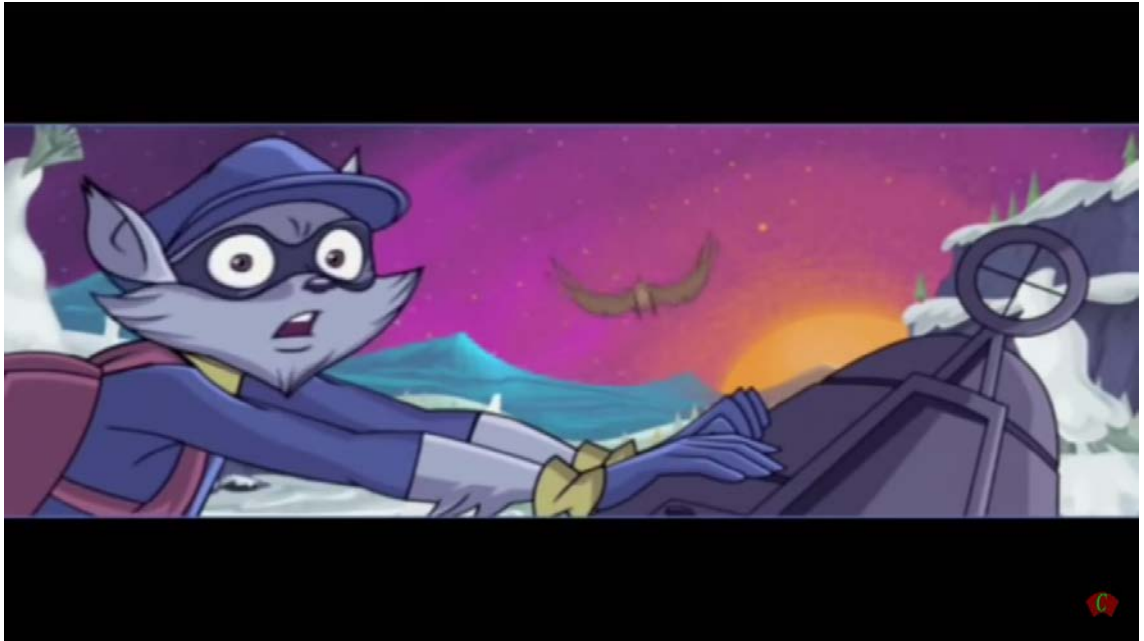
Kokemukseni mukaan välianimaatioiden budjetointi niin ajallisesti kuin työtuntienkin osalta saattaa olla hyvin haastavaa etenkin pienille studioille. Vaikka pelitilanteissa on ongelmansa, välianimaatioiden tuotanto muistuttaa kaikissa tapauksissa ainakin jossain määrin animaatio-ohjelman toteutusta. Tässä opinnäytetyössä ei ole tilaa tutkia välianimaatioita syvällisemmin, mutta koska ne ovat merkittävä osa pelianimaatiota, luodaan tässä luvussa niihin yleiskatsaus.

Kirjassaan *Level UP! the Guide to Great Video Game Design* Scott Rogers jakaa pelien välianimaatiot viiteen, opinnäytetyön kannalta merkittävään, luokkaan joista neljä on mielekkäitä animaatiotapoja 2D-pelianimaation kannalta:

Täysanimoidut Välivideot (Animated Cutscenes)

Animaatio-ohjelmien tapaan animoidut välivideot ovat perinteisiä animaatiopätkiä, jotka on viety pelimoottoriin formaatissa, mitä pelimoottori toistaa. Esimerkiksi .oggina, .mp4:nä ja niin edelleen. Tällä tavalla on mahdollista luoda korkealaatuista ja vaihtelevaa animaatiota. Huonoina puolina voidaan mainita animoitujen välivideoiden tilaa vievä muoto, jonka takia tämän kaltaiset välivideot eivät sovi esimerkiksi mobiilipeleihin. Rovio on kuitenkin luonut täysanimoitujen välivideoiden kaltaista lisäsisältöä *Angry Birds* -pelisarjaansa. Kompromissina näitä animaatioita voi katsella erikseen Rovion omilta pelisivuilta ja esimerkiksi Youtubesta.

Hyvälaatuisten täysanimoitujen välivideoiden toteutus on kallista ajallisesti ja rahallisesti. Siksi ne harvoin sopivat pienten peliyrittäjien välianimaatoratkaisuksi. Tosin pelin perustuessa animaatiotarjaan, tuotantostudio voi käyttää animaatio-ohjelman animaatiopätkiä välivideoina. Tällöin välianimaatio on jo valmiina, eikä toteutukseen tarvitse laittaa lisäresursseja. Kuva 10 on otos laadukkaasta välianimaatiosta *Sly Cooper* -lastenpelistä 2000-luvun puolesta välistä.

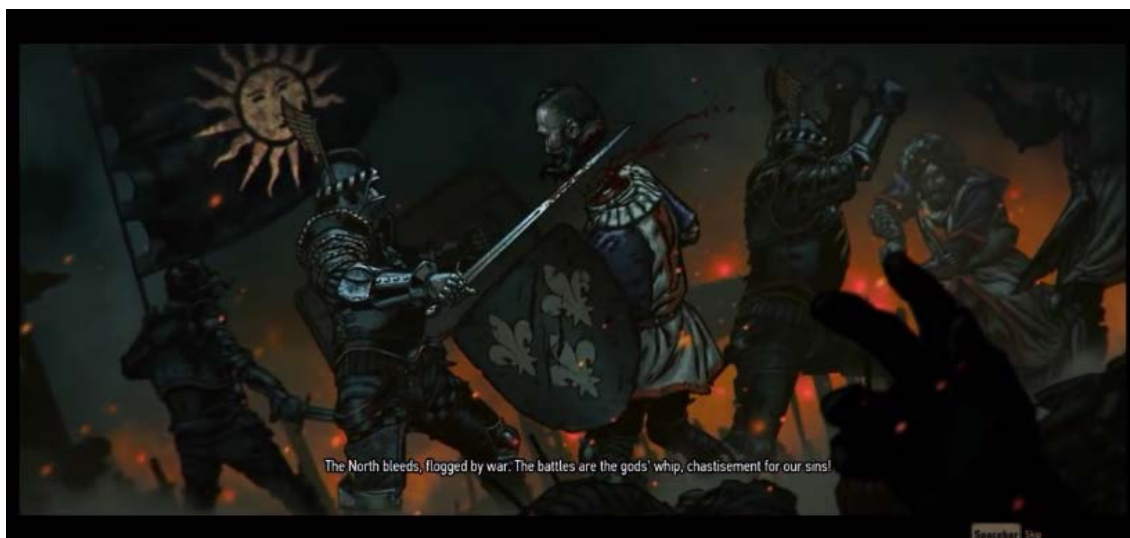


Kuva 10 Sly Cooper tasohyppelypelisarja käyttää välianimaatioinaan laadukkaita, täys-animoituja videoita.

Animatikit (Animatics)

Hyvä huomio on, että Scott Rogers kutsui tätä välianimaatioiden toteutustapaa animaatiosekvensseiksi (Animation Sequences). Koska opinnäytetyö koskee 2D-pelejä, vaihdan termin animatikeiksi, sillä tekniikan jälki muistuttaa animatikeja. Pelien kehityksessä animaatiosekvenssi taas tarkoittaa pelin sisäisiä kuvasekvenssianimaatioita.

Välianimaatioina animatikit ovat käytännössä yksinkertaistettuja animoituja välivideoita. Tekniikka on ollut viime aikoina muutaman ison pelistudion suosiossa ja tähän tyyliin on tuottanut sisältöä muun muassa Blizzard ja CD Projekt Red, jonka Witcher -pelisarjan välianimaatiosta on esimerkki kuvassa 11.



Kuva 11 The Witcher 3 - Wild Hunt -pelin välianimaatio (CD Projekt Red, 2015).

Nukketeatteri Puppet Show)

Tässä kontekstissa nukketeatteri on puhdasta pelimaailman aseteilla tuotettua animaatiota. Tämän kaltainen animaatio tuotetaan pelimoottorin sisällä pakottamalla pelilanteen hahmot animaatioon ja viemällä pelaajalta pois pelihahmon ohjattavuus. Tästä esimerkkinä kuva 12. Klei-yhtiön animaatio on myös hyvä esimerkki siitä, miten animaatiotyylillä voi korostaa pelin tyyliä.



Kuva 12 Klein Don't Starven intro on lyhyt, tyylikäs ja toteutettu pelin sisäisin asetein.

Huomaamattomat välianimaatiot



Kuva 13 Final Fantasy IV ja taistelu Magus Sistersia vastaan (Square Enix, 1991).

Huomaamattomat välianimaatiot ovat pelin kulkuun upotettuja pienanimaatioita, joilla voidaan korostaa kohtauksen draamaa, mikä muutoin saattaa olla hyvin hankalaa. Tällaisia animaatioita ovat esimerkiksi klassisista japanilaistyyillisistä roolipeleistä tutut päävastusanimaatiot, jossa merkittävä vihollinen muuttua ulkoasuun kesken taistelun kulun tai vaikka pelikentän kohtaus, jossa hahmo lentää halki ilman ja ympärillä tapahtuu räjähdyksiä. Tällaisia pelihetkiä pelaaja ei miellä välianimaatioksi, mutta ne ovat silti tärkeitä, pelinkehittäjän kontrollissa olevia kohtauksia, joilla lisätään dramaattisuutta. Tästä vanha, mutta pätevä esimerkki kuvassa 13. Kuvan Magus Sisters saattavat muuttua muuttua yhdeksi hirviöksi kesken taistelun ja se on upotettu osaksi pelin kulkua.

4 SPINE-ANIMAATIO PELIMOOTTORISSA

Nykyisillä pelianimaatiotyökaluilla on mahdollista lisätä hahmon luurankoon myös teknisiä ominaisuuksia kuten eventejä tai polygon-alueita. Mikäli ohjelmoijan ja animaattorin välinen yhteistyö on sujuvaa, ohjelmointi standardisoitua ja tekijät ovat asiantuntevia, tuotanto tehostuu merkittävästi. Onkin suotavaa, että myös ohjelmoijat tutustuvat animaatiotyökalujen tarjoamien runtimejen mahdollisuuksiin ja näin kehitetään tiiminä toimivampia kokonaisuuksia.

Luanimaatiotyökaluista ainakin Spine (Esoteric Software) ja Dragon bones (Dragon Bones) hyödyntävät runtimeissaan paljon koodia, mihin pelinkehittäjä voi vaikuttaa pelimoottorille kirjoitetun koodin kautta. Näin on mahdollista toteuttaa mielenkiintoisia animaatioita, joille animaattori antaa animaatiotyökaluissa toiminnollisuuksia, joita hyödynnetään pelimoottorissa. Tällaisia toiminnollisuuksia voidaan yhdistää esimerkiksi kurveihin, jotka venyttävät animaatiota tai grafiikkaa fysiikan mukaisesti esimerkiksi törmäystilanteissa. Törmäystilanteet voivat olla hyvin vapaita aina esteistä tuuleen (ns. wind zone on muun muassa Unityssa käytettävä vyöhyke, joka liikuttaa aseteja, joilla on törmäysalue, joka on asetettu reagoimaan wind zoneen. Näin saadaan aikaan esimerkiksi puiden huojuntaa). Luanimaatiossa onkin hyödyllistä, että myös animaattori tuntee pelimoottorin tarjoamaa tekniikkaa ja uskaltaa laajentaa osaamisaluettaan myös tekniikan puolelle.

Animaatiotyökalun tuottaja tarjoaa usein valmiit runtimet merkittäville pelimoottoreille. Mikäli runtimea tarjotaan, kannattaa sitä usein käyttää. Yleensä ne on optimoitu erinomaisesti ja toimivat lähellä rautaa. Esimerkiksi Esoteric Softwarin Spine tarjoaa myös pseudokoodatun version runtimeesta, minkä kanssa voi toteuttaa oman runtimen. Sen toteuttamista kannattaa kuitenkin lähteä tekemään vain erityistapauksissa. Esimerkiksi tilanteessa, jos käyttää omaa pelimoottoria. Tällöinkin kannattaa tutustua eri skriptikielillä toteutettuihin yleisiin runtimeihin. Spinen runtimeja löytyy ainakin C#:lle, LUAlle ja JavaScriptille. (Esoteric Software 2016.)

4.1 Polygon

Polygon 2D-grafikassa on käsitteenä sama kuin 3D-grafiikasta tuttu termi. Ero on siinä, että polygonit asettuvat ainoastaan xy-akselille ilman syvyyttä. Animaatio-ominaisuuksien lisäksi polygon voidaan asettaa esimerkiksi partikkelipilven alkupisteeksi. Polygoneilla voidaan määritellä suoraan myös osuma-alueet, mitä tosin kannattaa käyttää harkiten mobiilituotteissa, sillä animaatiotyökalulla toteutetut polygon-pohjaiset osuma-alueet ovat aina mesh-tyyppisiä. Tämä tekee niistä melko raskaita eivätkä ne välttämättä sovi useimmille vanhemmille mobiililaitteille. Mikäli kuitenkin mesh-törmäysalueita käytetään, sen voi integroida jo tässä vaiheessa. Ainakin spinen polygon toimii törmäyspintojen havaitsijana pelimoottoreissa, joille on valmiit runtimet (mm. Unity ja Cocos2D). (Esoteric Software 2016.)

Polygoneilla voidaan luoda myös hahmoanimaation reaktiivisuutta ja painopisteitä. Esimerkiksi jos pelihahmo painaa sormellaan nappia, reagoi hahmon koko käsi ja käsi-

varsi animaattorin määrittelemällä tavalla hyödyntäen IK-tekniikkaa. Näin saadaan hahmon toimintaan luonnollisuutta sen toimiessa ympäristönsä kanssa. (Esoteric Software 2016; Unity 2016.)

4.2 Event

Event animaatioissa on sama kuin liipaisin ohjelmointikielissä. Se muuttaa määrättyä muuttujaa ja laukaisee näin erilaisia toimintoja. Esimerkiksi juoksevasta hahmosta lähtee askelääni aina soittaessa sitä keyframea, jolla jalka osuu maahan. Tämä on kätevää sillä esimerkiksi frameraten hidastuminen tietokoneen tehosta johtuen ei riko animaation ja äänen välistä dynamiikkaa. Toisena esimerkkinä annetakoon lyöntianimaatio, missä pelaajahahmon halutaan tekevän vahinkoa kohteelle ainoastaan tietyssä kohdassa animaatiota (esimerkiksi miekkailuanimaation osuessa kohteeseen). Tämä toiminto voi myös laukaista esimerkiksi partikkelipilven.

Teknisesti tämä tarkoittaa sitä, että tietyssä kohtaa animaatiota animaatioskripti muuttaa määrättyä muuttujaa, mikä laukaisee halutun tapahtuman. Tyypillisesti tämä tarkoittaa sitä, että pelimoottori tekee kutsun pelimoottorin omaan AnimationState-funktioon. Tämä voidaan toki toteuttaa myös omalla kuuntelu-funktiolla, mutta yleensä se ei ole järkevää, sillä nykyiset animaatiotyökalujen runtimet ovat mahdollisimman lähellä rautaa. (Esoteric Software 2016.)

4.3 Skinnaus (skin)

Animaatioluuranko on nykyaikaisissa pelianimaatiotyökaluissa sellainen, että sitä voi animaatioineen kierrättää eri skineihin. Tällöin pelimoottori kierrättää samaa luurankoa, mutta kutsuu grafiikat siitä atlaksesta, josta ohjelmoija ne koodissa kutsuu. Ajallisesti tämä on hyvin voittoisaa. Animaattori voi luoda yhteisen luurangon pelin kullekin samanlaiselle, geneeriselle elementille, mutta pelaaja ei sitä huomaa, sillä grafiikat vaihtuvat. Tämä on hyvä keino monipuolistaa pelimaailmaa.

Tässäkin tapauksessa on tosin huomioitava kohdealusta, sillä kovin monipuoliset ja vaihtuvat grafiikat kuormittavat nopeasti mobiilialustojen grafiikkasuorittimia. Sopivan vaihtuvuuden saavuttamiseksi voi tutkia julkaistuja pelejä ja harjoittaa testaamista yleisöllä. (Esoteric Software 2016.)

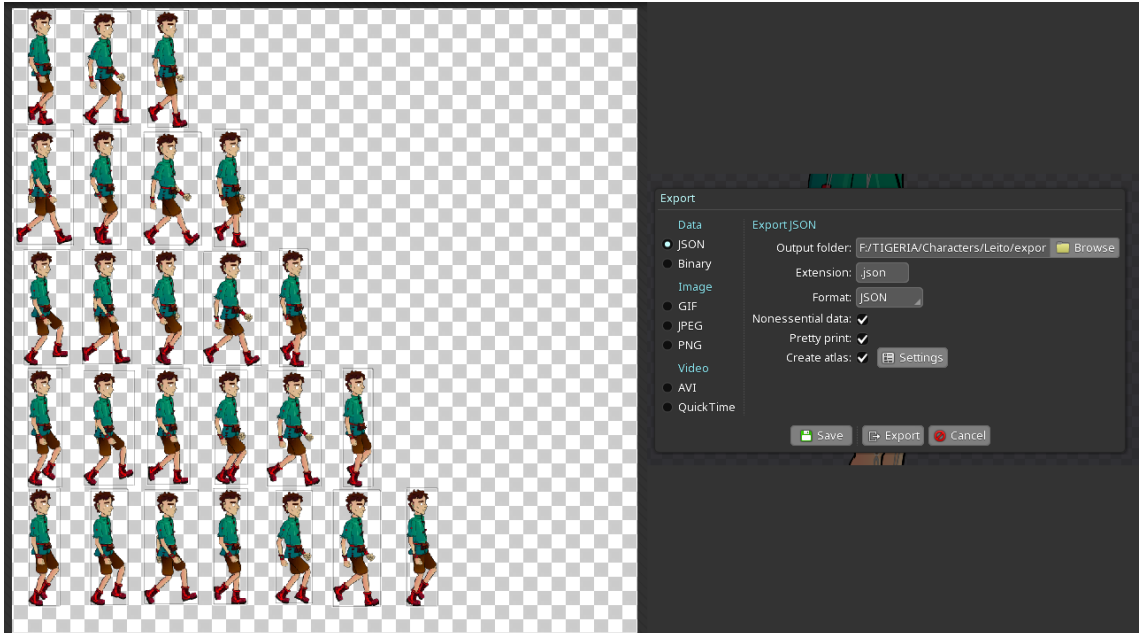
4.4 Animaatioiden toimitus pelimoottoriin

Animoinnin jälkeen animaatiot on vielä vietävä pelimoottoriin ja pelimoottorissa on tarkastettava animaation rytmi, saumattomuus ja mahdolliset interaktiiviset ja ohjelmointiriippuvaiset toiminnot kuten skinnaus ja polygon.

Animaation toistotekniikka riippuu toteutustavasta. Nyrkkisääntönä on, että luuanimaatiot käyttävät omaa runtimeaan ja kuvasekvenssianimaatiot jaetaan ohjelmallisesti tai

graafisin työkaluin ruutuihin, mitkä asetetaan toistuvaksi kuvasarjaksi jälleen joko ohjelmallisesti tai graafisen käyttöliittymän kautta (Esoteric Software 2016; Unity 2016).

Kuvassa 14 on rinnan Unityn kuvasekvenssityökalu ja Spinen vientityökalu .JSON -asetuksin. Oikealla puolella osituksessa on kuvan 3 animaatio.



Kuva 14 Unityn kuvasekvenssityökalu ja Spinen vientityökalu.

Kun työskentely siirtyy pelimoottoriin niin vastuu siitä, kenen vastuulla animaation saattaminen pelikuntoon riippuu työtiimin tai yrityksen toimintaperiaatteista. Joskus artistin tai animaattorin tehtävänä on saattaa animaatio mukaan peliin, joskus asiasta vastaa tekninen artisti tai moottorin parissa työskentelevä henkilö (Rogers, 2010).

5 PROJEKTIT SANALANKA JA GALACTIC ECHOES

Viime vuosina 2D-pelianimaatiotyökalut ovat kehittyneet merkittävästi. Kuvasekvensseistä ja moottorien sisäisistä, yksinkertaisista animaatiotyökaluista on siirrytty monipuolisiin, erilaisia tekniikoita tukeviin universaaleihin työkaluihin, jotka toimivat runtimejen avulla pelimoottoreissa.

Minä olen käyttänyt työssäni niin Spriteria, Spinea kuin Harmonyakin, sekä joitain pelimoottorien sisäisiä animaatiotyökaluja, kuten Dragon Bonesia ja Unityn sisäisiä työkaluja. Näistä pelimoottoriin riippuvaiset työkalut sekä Spriter ja Harmony ovat osoittautuneet käyttökelvottomiksi minulle. Näistä Spriter on liian yksinkertainen ja toiminnoiltaan joustamaton. Harmony taas on kallis sisältäen kaikki animaatio-ohjelmien tuottamiseen tarkoitettut työkalut. Lisäksi sen animaatiot toimivat pelimoottoreista vain Unityn kanssa, mikä vähentää mahdollisia asiakasyhteyksiä.

Spine on onnistunut erottautumaan laadukkailla runtimeillaan sekä jatkuvina päivityksinä, jotka ovat lisänneet ominaisuuksia ja käytettävyyttä (vuonna 2016 4 merkittävää päivitystä). Tämä on monipuolistanut työskentelyä ja parantanut laatua, jota tuotteella voi toteuttaa.

Omana työnäni esittelen kahta projektia, joihin olen luonut keskenään suhteellisen erilaista sisältöä, ja eri tarkoituksiin. Toinen tuote on Ikoniac Oy:lle toteutettu markkinointitarkoituksiin, ja Galactic Echoes -pelin aloitusvideoksi tarkoitettu intro-animaatio ja toinen Sanalanka-tuotteeseen toteutettu animaatio sisältö, joka käsittää niin pelianimaatiot kuin välianimaatiot.

5.1 Galactic Echoes -introanimaatio

Ikoniac halusi tuotteeseensa animaation, joka avaisi pelin tarinaa. Lisäksi animaation oli oltava sellainen, että se toimii suoraan kännykältä, toisin sanoen videopalvelusto Internetin välityksellä ei kelvannut. Tuote oli kansainvälinen, ja sitä markkinoitiin myös alueilla, missä verkkoyhteyttä ei ole kaikkialla. Lisäksi Ikoniac Oy oli tietoinen siitä, että useissa maissa liittymisiin kuuluu datakatto, mikä saattaisi estää videon toistamisen esimerkiksi Youtube-peilauksen kautta. Peli itsessään oli jo melko iso mobiilipeliksi (noin 70 MB), mikä jätti minulle noin 20 MB toteutusvaraa, että peli olisi yhä ladattavissa yhtenä pakettina mobiilikaupasta.

Jo tässä kohtaa oli selvää, että renderoity animaatio ei kelpaisi, sillä animaation tuli olla hyvälaatuinen myös suurilla näytöillä. Tällöin ainakin 720P-laatu on ehdoton, mutta se tarkoittaa sitä, että .avi tai .mp4 veisi noin minuutin animaatioina kymmeniä megatavuja.

Oli siis tehokkaampaa, että animaatio toteutettiin Spinella runtime-animaationa, mikä toistaa animaation atlatseen koottujen grafiikoiden ja ohjelman luoman JSONin pohjalta. Tällöin kokoa lisää pääasiassa atlatset.

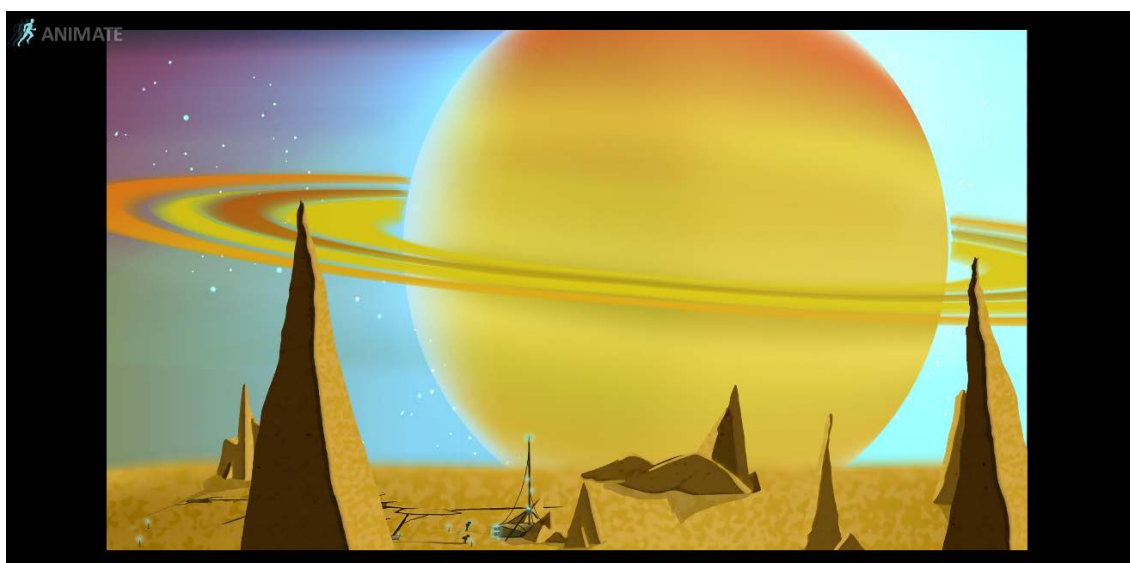
Aluksi ratkaisu oli kuitenkin ongelmallinen, sillä Internet tai kirjastot eivät tarjonneet minulle referenssiä vastaavan toteutustavan hyödyntämisestä välianimaatioissa. Olin siis

teorioiden kanssa melko omillani. Teknologia on kuitenkin melko uusi. Tekniikka on kuitenkin yleistynyt. Esoteric Software lanseerasi PR-tarkoituksissa Twitter-merkintänsä #madewithspine, jonka tarkoituksena oli innostaa artisteja jakamaan töitään suurelle yleisölle. Lanseeraus onnistui hyvin, ja nykyään julkaistaan suuria määriä animaatioita kyseisellä merkinnällä.

Animaatio käsikirjoitettiin niin, että siinä oli neljä kohtausta (Titanin pinta, Hissi maan uumeniin, Jääluola ja Jääpyramidi). Animaation pituudeksi sovittiin 45-60 sekuntia ja toteutusajaksi 2 kk, jonka jälkeen kaksi palautekierrosta, joiden pohjalta tehtiin korjaukset animaatioon.

Animaation toteutus

Spine on ensisijaisesti pelianimaatiotyökalu, joten välianimaation toteutuksessa on huomioitava joitain erityisheitteitä. Käyttäjälle näkyvin on ruutukoko. Älypuhelimien ruutujen kuvasuhteen varioivat paljon. Peli oli kuitenkin suunniteltu melko uudelle älypuhelinsukupolvelle, joten tuotannossa oli huomioitava pääasiassa 16:9 ja 16:10 sekä tähän väliin osuvat erikoiskoot. Spinen ollessa erityisesti animaatiotyökalu pelin sisäisille animaatioille ei se sisällä esimerkiksi kameroita, joten sellaiset täytyy luoda keinoitellisesti.



Kuva 15 Otos Galactic Echoes -pelin kohtauksesta Titanin pinta, missä kameralla on monimutkaisia ajoja.

Esimerkiksi Kuvan 15 kohtauksessa Titanin pinta, oli monimutkaisia kamera-ajoja. Tällöin vaarana oli, että laajemmassa kuvaformaattissa 16:9, kameran liike olisi jatkunut varsinaisen animaatioalueen ulkopuolelle. Ratkaisuna taustan avaruus piirrettiin niin, että se jatkui animaatioalueen ulkopuolelle 100 pikseliä. Lopulta ratkaisu ei ollut kovin raskas, mutta koska animaation liike ei riko animaation kuva-alueen rajoja missään tilanteessa, jälki pysyy ammattimaisena eikä riko vaikutelmaa.

Sen lisäksi erilaiset efektit, kuten tuulta kuvastavat partikkelit, jääkiteiden välke ja niin edelleen, on luotava pelimoottorissa. Tämä toteutettiin niin, että animaatiota varten on oma kohtaus, minkä mittapuissa oli mahdollista ajastaa erilaisia asioita, kuten partikkeleita tai varjostimia.

Varsinaisessa animaatiotyössä ongelmat taas ovat lähinnä perinteisiä, sellaisia, mitä animaatio-ohjelmienkin tuotannossa kohdataan. Työkaluja tosin rajoittaa se, että niiden on oltava sellaisia, että pelimoottori voi toistaa niitä tietueen pohjalta. Esimerkiksi polkuanimaatiot on toistettava bézier-käyrin.

Ruutukoko

Käytännössä tein kehykset, joilla rajasin valitussa kuvasuhdehaarukassa näkyvät kuvasuhteet. Näin sain määriteltyä turvallisen rajan, jonka sisällä näytettäisiin varsinainen toiminta sekä rajan, jonka sisällä grafiikan pitää jatkua täydellisenä, että laajakuvaisilla-kin ruuduilla näkymä on mielekäs.

Vaihtoehtoisesti elokuvakokoja voi määrittellä mustilla palkeilla, joiden sisään mahtuu esimerkiksi 16:10 ja näin 16:9-koolla 16:10:tä laajempi alue näkyy mustana.

Kohtauksittaisuus ja kameran liike

Kameran puute ei varsinaisesti vaikeuttanut kamera-ajattelua, sillä maailman liikuttaminen kameran sijasta on tekniikka, joka käytetään monenlaisissa toteutuksissa aina animaatio-ohjelmista peleihin. Kohtausten ankkurointi taas toteutti enemmän päänvaivaa. Lopuksi oivaksi ratkaisuksi osoittautui se, että jokaiselle kohtaukselle luotiin oma perusluu, jonka päälle kaikki kohtauksen objektit tuotiin. Kunkin perusluun nollapiste oli maailman nollapisteessä, missä sitä siirreltiin.

Spinessa kaikki lapsiluut perustavat oman sijaintinsa vanhempiluun sijaintiin, mikä kameran liikkeissä helpottaa erillisten kohtausobjektien kohdistamista. Tämä helpotti erityisesti Titanin pinta ja Hissi maan uumeniin -kohtausten animointia, joissa oli dynaaminen kamera.

Kesällä tullut päivitys toi mukanaan polkuanimaatiot, eli bézier-käyrät, joita luut ja luiden sisältämät animaatiot voivat seurata. Näin pystyin liikuttamaan kohtauksen perusluuta ja luoda samalla sujuvia maailman liikkeitä. Ennen päivitystä tämä oli ongelmalista, sillä sujuvuutta piti huoltaa erilaisin tviinauksin, mikä on hidasta työtä. bézier-käyrät tehostivat työtä selkeästi.



Kuva 16 Kohtauksessa Hissi maan uumeniin, missä liikettä korostetaan ulkopuolisin objektein.

Lisäksi liikettä korostettiin kohtausten liikkeiden avulla, kuten hissikohtauksen hissiliikkeellä ja sitä ympäröivässä jäässä olevien partikkelien liikkeillä. Partikkeleja oli lopulta viisi erilaista, joiden grafiikan väriä ja kokoa muutettiin niiden ehtiessä kuvan ylälaitaan. Sen jälkeen partikkelit siirrettiin ylhäältä alas ja käytettiin eri näköisinä samaan tarkoitukseen. Tästä esimerkkinä kuva 16.

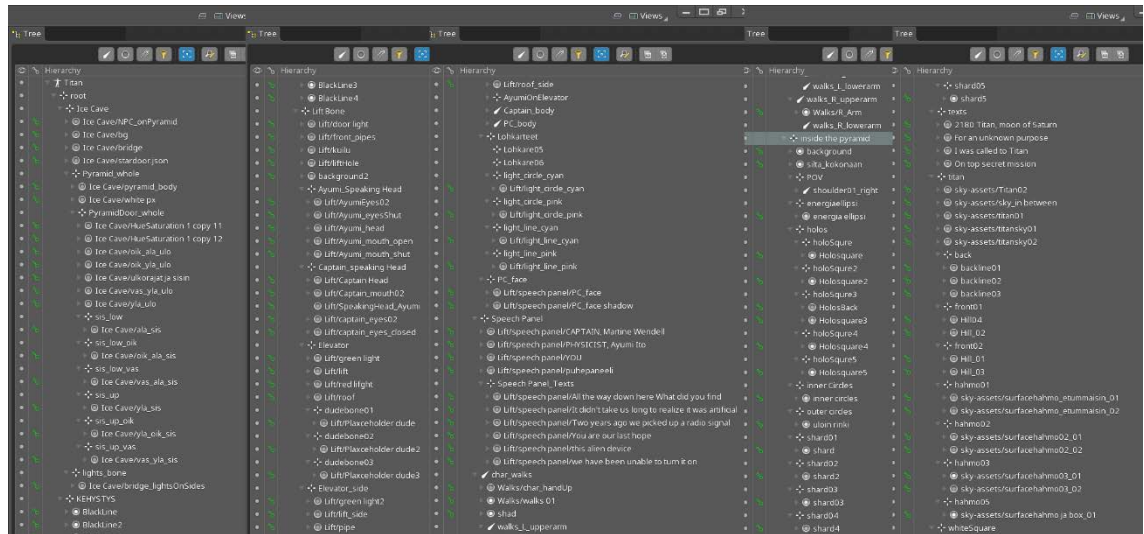
Objektina hissi ei liiku, joten sen liikettä mallinnetaan muiden animaation objektien avulla. Muun muassa hissien ulkopuolella sijaitsevien partikkelien liikkeellä hissien ohitse.

Animointiprosessi

Projektissa työskentelin varsinaisen työaikani hyvin itsenäisesti. Asiakkaan kanssa olimme kontaktissa palautekierroksissa ja välitösten kohdalla. Pyysin myös palautetta.

Projektin edetessä ongelmia tuotti animaation laajuus. Kokonaisuutta ohjasi satoja luita, joiden kanssa meni helposti sekaisin. Sen hallinta vaati valtavasti organisointia. Kuvassa 17 on esitelty hierarkian koko.

Itse työ onnistui jouhevasti ja itsenäisesti. Olen tottunut työskentelemään etäpisteeltä asiakkaan kanssa, joskus toiseen maahan, joten oli suorastaan yksinkertaista työskennellä turkulaiselle yritykselle. Työskentelyssä ei tarvinnut huomioida aikaeroja, tai vaikkapa kulttuurillisia poikkeavuuksia työtavoissa.

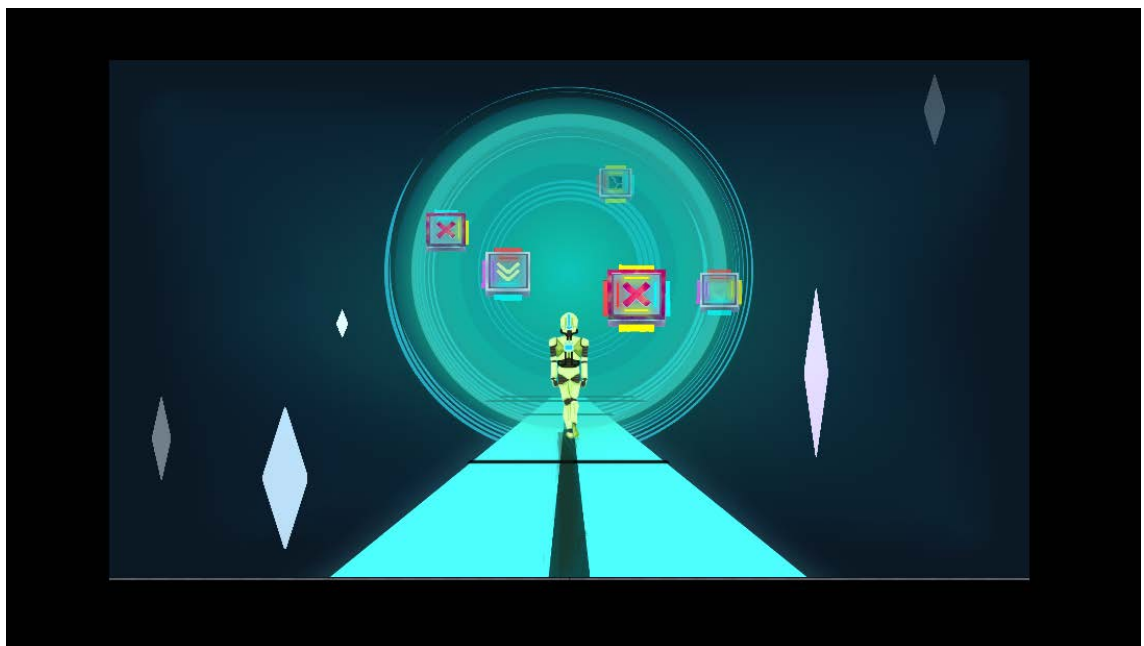


Kuva 17 Välianimaation luuston laajuus on valtava.

Selkeästi nimetty luusto kuitenkin pysyi hyvin hallussa, ja suurta luumäärää oli lopulta helpompaa hallita listasta kuin valita ruudulta yksittäisiä luita.

Varjostimet, partikkelipilvet ja äänet

Varjostimet ja partikkelipilvet toimivat ajastetusti. Tähän Unityn partikkelityökalun ja Shader Forgen varjostintyökalun ajastimet olivat riittäviä. Partikkeleilla luotiin lähinnä syvyyttä ja usvaisuutta Jääpyramidin sisällä -kohtaukseen. Varjostimella luotiin rakeisuutta samaiseen tilanteeseen.



Kuva 18 Jääpyramidin sisällä.

Mitä tekisin toisin

Spinen runtime sisältää suuren määrän kooditiedostoja, jotka toteuttavat runtimen eri toimintoja. Galactic Echoesin introanimaatio käytti noin puolta tarjotuista toiminnoista, joten loput olisi voinut poistaa ennen levitystä. Tämä olisi pienentänyt animaation kokoa ja tehostanut sen toiminnallisuutta vielä lisää.

Myös partikkelien ja varjostimien toiston olisi voinut toteuttaa kauniimmin. Niin kuin ääniä, myös näitä komponentteja olisi voinut kutsua eventein, mikä olisi nopeuttanut muokkausta, sillä odotusaikaa ennen efektin toistumista ei olisi tarvinnut säätää, vaan efekti olisi tapahtunut halutussa kohdassa event-kutsun kautta.

Työprosessin aikana Esoteric Software julkaisi päivityksen, joka toi mukanaan animaatiopolut, joita käytettiin muun muassa dynaamiseen kameraan. Nyt käyttäisin sitä myös Jääpyramidin sisällä -kohtauksen useisiin liikkuviin osiin, sillä se pehmentäisi liikettä, joka nyt töksähtelee hieman.

5.2 Sanalangan peli- ja mikroanimaatiot sekä introanimaatio

Tässä luvussa esittelen poikkeavat metodit, joilta jouduin käyttämään erilaisen kerrontatavan vuoksi.

Sanalangassa toteutin myös pelin sisäiset animaatiot. Tähän kuuluu hahmoanimaatiot, mikroanimaatiot ja toiminta-animaatiot, jotka pelaaja käynnistää klikkailemalla pelikohtausten kiinnekohtia. Tuotannossa toimi myös muita peliartisteja, joten grafiikat eivät olleet yksin minun käsialaani. Myös osalla ympäristöanimaatioita oli eri tekijä (osa metsäkohtauksen sienianimaatioista oli toisen artistin käsialaa). Tämä takia tuotantoon tuli vielä yksi aspekti: tyylien yhdistäminen. Muina artisteina toimi Teemu Seppänen (avaruusaluksen grafiikka ja avautuminen, sienet ja joidenkin sienien animaatio, paljon minipelikuvia), Natasha Bulatovic trygg (taustamaalaukset). Minä vastasin minipelianiimaatioista (sekä niiden grafiikoista ja osasta minipelikortteja), hahmoanimaatioista ja hahmojen grafiikoista, introanimaatiosta ja ympäristöä elävöittävästä mikroanimaatioista.

Lopputuloksena oli hyvin yhdistetyt tuotantotyylit, joista asiakas ei onnistunut nostamaan eri piirtäjien ja animaattoreiden grafiikoita irti toisistaan. Silti ulkoasusta ei tullut latteaa tai yksinkertaista. Kuvan 18 oikean puoleisessa osassa on kaikkien pelituotannon artistien toteuttama ulkoasu nähtävillä.

Pelin protagonistina on Miima, avaruusperheen onnettomuudessa maahan perheensä kanssa maahan laskeutunut avaruusolentolapsi. Pelissä pelaaja tutustuu kieleen Miiman kautta oppaanaan orava Papu.

Grafiikat ovat värikkäitä ja lapsellisia, ja animaatiot ovat satukirjamaisia, vahvoja ja lapsille sopivia. Sanalangan tärkein kohderyhmä on 3-8-vuotiaat lapset, jotka tarvitsevat eritasoisista logopedistä terapiaa.

Animaatioiden uusiokäyttö

Puheterapiayritys Contextia, joka testaa Sanalankaa, toivoi palautekierroksellaan, että ympäristöihin tuotaisiin enemmän animaatioita, joita lapsi voisi käynnistää klikkailemallaan ympäri ruutua. Lapset, jotka pelaavat peliä, ovat kielitaidoiltaan monen tasoisia, joten toiveena oli, että ne, jotka osaavat kieltä liian heikosti metodisille minipeleille, saisivat silti mahdollisimman paljon pelistä irti, vaikka eivät ymmärtäisi pelin sisältöön kiedottua tarinaa.



Kuva 19 Sanalangan ensimmäisen kohtauksen vanha (vas.) ja uusi versio.

Pelin grafiikka oli jo valmiiksi melko tilaa vievää, joten päätimme tehdä pieniä animaatioita jo olemassa oleville elementeille kuten sienille, avaruusalukselle ja puille. Näiden objektien tuli olla myös uusiokäytettäviä. Kuvassa 19 ja 20 näkyy ympyröitynä ympäristön objekteja, joihin lisättiin liikettä ja animaatiota. Sienet ja uuden avaruusaluksen mallin toteutti Teemu Seppälä, muissa näkyy minun käsialani. Samoja objekteja, kuten sieniä ja lintuja käytetään useissa kohtauksissa. Tämä säästää kovalevytilaa ja samalla saadaan monipuolista sisältöä kunkin kohtauksen ympäristöön.

Uusiokäytettävyys ja keveys toteutettiin niin, että ohjelmoija loi peliobjektiin liitettävän koodin, joka luki objektin spine-animaation ja törmäyksentunnistimen, joka oli merkitty sovitulla tavalla (tässä tapauksessa TapCollider). Näin pystyin teknisenä artistina luomaan peliobjekteja, jotka sisälsivät animaation ja grafiikan kokonaisuuden, kuten vaikkapa linnun lennon ja oksalla istumisen. Luodusta objektista tehtiin prefabeja, joita pystyi laittamaan useita pitkin pelikenttää.



Kuva 20 Toinen kuva Sanalangan ympäristöanimaatioista.

Hahmoanimaatiot

Sanalangan hahmojen yleinen animaatiokaala ei sisällä kovin erikoisia animaatioita tai animaatioita, jotka perustuisivat voimakkaaseen dynamiikkaan. Syynä tähän on pelin luonne. On mahdotonta, että peli sisältäisi tällaista materiaalia, sillä se veisi huomiota pois pelin ja sen hahmojen tarkoitukselta (ks. luku 3). Sen sijaan hahmot liikkuvat ja vuorovaikuttavat pelaajan tahdon mukaan.

Pelaajahahmo-Miimalla on luonnollisesti kaikkein laajin animaatiokaala. Muut hahmot on animoitu ja toteutettu niiden tarpeiden mukaan, millaista roolia ne pelissä esittävät. Esimerkiksi Miiman vanhemmat ovat lähinnä Avaruusaluksen laskeutumiskohtauksessa alkuanimaation jälkeen. Miiman vanhemmat eivät siis tarvitse esimerkiksi monimutkaisia kävelyanimaatioita.

Miima ja hänen vanhempansa olivat yksinkertaisimmat animoida. Heidän vartalonsa ovat kahdella jalalla kävelevän humanoidin vartaloita. Ihmismäiset, melko tavalliset liikkeet sopivat heidän kuvaamiseensa erinomaisesti. Toteutuksessa hyödynsin tekniikoista sprite swapia (ilmeet ja kädet), mesh warpia (erityisesti raajojen liikkeet, mutta myös yleinen luonnollisuus ja soljuvuus) sekä puhdasta luuanimaatiota (esimerkiksi kulmakarvoissa ei ole lainkaan muita tekniikoita).

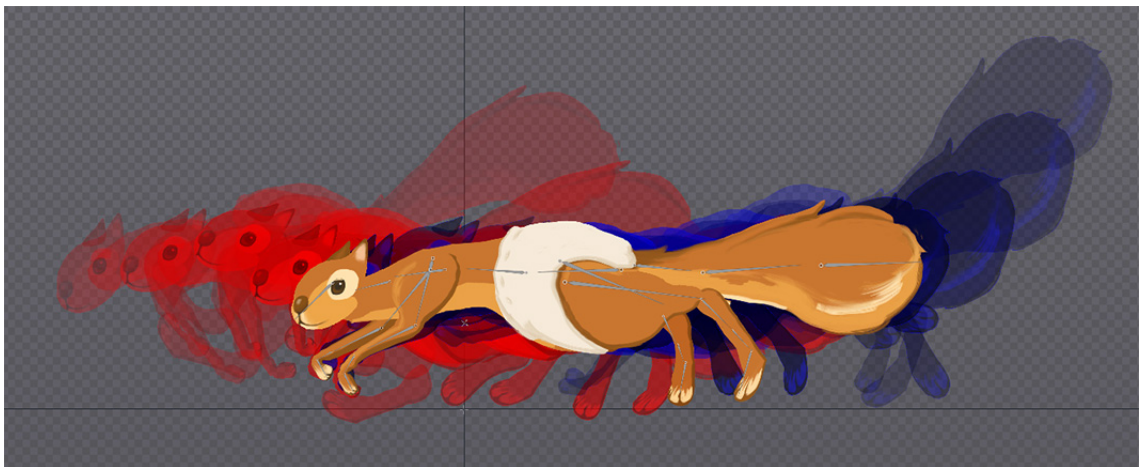
Animaatiotyön apuna käytin animaatiotaulukkoa (Taulukko 2 on Sanalangan hahmojen animaatiotaulukko), johon luetteloin eri animaatiot ja yhdistin ne hahmoihin. Näin jokaisen hahmon animaatiosisältöä ei tarvinnut miettiä erikseen, vaan animaatioiden määrittely kuului suunnitteluvaiheeseen ja ne sai siirrettyä

Taulukko 2 Sanalangan hahmojen animaatiotaulukko.

Animaatiot	Miima	Avaruusvanhemmat	Avaruusalus	Papu
Paikallaan	x	x	x	x
Sivuttaiskävely	x			x
Kohti kameraa/ poispäin kamerasta	x			
Ottaminen	x			
Vuorovaikuttaminen	x	x		x
Ilmeet	x	x	x	
Erikoisanimaatiot	istuminen		avautuminen	hyppy

Papu oli animaation kannalta hankala kohde. Sen lisäksi että orava on nelijalkainen olento, jolla on varsin rytmikäs ja eläväinen liikehdintä, sen suunnittelussa oli annettu eläimelle paljon inhimillisiä piirteitä, kuten vaate ja suorassa istuminen. Muutos istumisesta juoksevaan eläimeen oli haasteellinen. Ratkaisuna käytin Mariel Cartwrightin (2014) mukaista Riko fysiikka -metodia. Lisäksi animaation vaihdolle tehtiin ennakointi pienellä hypyllä ennen liikkeen muutosta, esimerkiksi pysähtymisen ja kävelyn välillä.

Lopulta pelikohtaukset osoittautuivat pinta-alaltaan melko ahtaiksi suurikokoiselle oravahahmolle juosta. Tämä johtui siitä, että tila, missä hahmot liikkuvat, oli suunniteltu kaksijalkaisten hahmojen mukaan ja lähinnä päähenkilö Miimaa ajatellen. Tästä syystä liikkumatila nelijalkaiselle olennolle oli huomattavan rajattua, ja liikkumatilaa olisi ollut lähinnä pienille pyrähdyksille. Oravan liike olisi näyttänyt paremmalta, jos se olisi toteutettu kapeampana tai liike olisi toteutettu pelkin hypyin sijaintien välillä. Kuvassa 21 näkyy Papun juoksuanimaation leveys.



Kuva 21 Papun juoksuanimaation Spinessa haamuavuin.

Minipelianimaatiot

Minipelianimaatiot olivat sinänsä yksinkertaisia toteuttaa. Käytännössä ne ovat animaatioita mitä erilaisimmista verbeistä. Esimerkiksi imeä, juosta, syödä, nakertaa, kaivaa ja jolkottaa. Minipelianimaatioita käytetään pääasiassa Muodosta lause -korttipeleissä, joissa lapsi valitsee subjektin, verbin ja objektin korteista, joita on kutakin viisi ruudulla. Korteista verbit on toteutettu animaatioin (myöhemmin myös osa adjektiiveista).

Ongelmana kuitenkin oli animaatioiden suunnittelu ja niiden pieni koko (vain muutama sentti fyysiseltä mitaltaan ruudulla). Animaatioiden täytyi olla mahdollisimman univertsaaleja ja abstrakteja, mutta sellaisia, että lapsi ymmärtää ne helposti. Aikaisempien korttipeliversioiden ongelmana oli ollut nimenomainen huono ymmärrettävyys. Käytännössä tämä ratkaistiin niin, ettei toiminnalla ollut koskaan selkeää subjektia. Esimerkiksi kaivaa-verbissä lapio kaivoi maata ilman kaivajaa. Verbissä nakertaa hampaat nakersivat abstraktia, porkkanan kaltaista muotoa ja niin edelleen. Jokainen korttianimaatio hyväksytettiin testaajalla.

Unitysissä animaatioista tehtiin prefabit, jotka sisälsivät spine-animaation ja kutsun animaatioille, mikäli se arvotaan ruudulle minipelin korttipakasta. Ensimmäisten yritysten jälkeen päädyttiin ratkaisuun, jossa kortin animaatio lisättiin taustaan jo Spinessa, kun staattisissa, animaatiottomissa korteissa tausta lisättiin vasta, kun objekti arvottiin ruutuun. Tämä toteutettiin siksi, että osa subjekteista saattoi näytellä myös objektin osaa ja toisin päin.

Introanimaation eroavaisuudet Galactic Echoesin introanimaatioon

Galactic Echoesin ja Sanalangan erot pohjautuvat kohderyhmien eroon. Siinä missä Galactic Echoesin kohderyhmä oli ongelmanratkaisupeleihin kasvaneet kolmikymppiset aikuiset, niin Sanalanka on tuote 3–7-vuotiaille lapsille, jotka tarvitsevat tukea kielenoppimiseen. Samanlaiset tuotantometodit ja samanlainen lopputuote eivät käy päinsä. Sanalanka vaatii yksinkertaisempaa kerrontaa.

Sanalangan introanimaation käsikirjoituksen raakaversioon toteutti Jenny Wiik, mutta käsikirjoitusta muokattiin sopimaan paremmin animaatiokerrontaan. Silti, monia kohtia selitettiin tekstillä enemmän kuin olisi selitetty lapsille, jotka eivät tarvitse puheenymmärtämiseen apua.

Apuna käytettiin myös vahvoja ja kontrastisia väriskaaloja. Contextian edustaja kertoi, että tämä auttaa lapsia keskittymään kohteisiin. Myös avaruuskohtausten yksitoikkisuutta vähennettiin sarjakuvaruuduin, jotka ilmestyivät animaatioon kertomaan, mitä avaruusaluksen ohjaamossa tapahtuu.

Luurakenteesta tuli yhtä suuri kuin Galactic Echoesin animaatiossakin, mutta animaatiokurvityökälun päivitykset olivat ehtineet ilmestyä animaatioprosessin alettua. Näin muun muassa planeettojen kiertoradoista sai nopeammin kauniita.

5.3 Projektien tulokset

Projekteista Galactic Echoesin introanimaatio saatiin kokonaan valmiiksi ja integroitua tuotteeseen. Laatu oli hyvää, ja asiakas piti lopputuloksesta. Myös välianimaatioista ja episodien välianimaatioista on puhuttu markkinointitarkoituksissa, mutta niiden toteutusta ei ole aloitettu.

Spine osoittautui hyväksi työkaluksi ja se ratkaisi animaatio-ongelman koon ja animointinopeuden suhteen.

Sanalankaan toteutetut animaatiot integroitiin myös tuotteeseen, mutta tuotteen kehittäminen on muutoin yhä vaiheessa. Tämänhetkisistä animaatioista on kuitenkin hyvä jatkaa. Lopputulos näyttää hyvältä ja tuntuu uskottavalta. Jatkokehitys jatkunee talvella 2017.

Luananimaatio osoittautui hyväksi työkaluksi tilanteisiin, joissa toteutetaan hyvälaatuista animaatiota pieneen tilaan, mikä on tärkeää esimerkiksi mobiilialustoille kehitettäessä. Suurissa tuotannoissa, joissa animaattoreita on monta sekä aikaa ja tilaa riittävästi, kuvasekvenssianimaatio tuottaa parhaan jäljen, mikäli suurten animaatiostudioiden tyyppillinen jälki on sellainen, mitä haetaan. Luuanimaation tuottama jälki saattaa olla tyylikkyys myös suuremman budjetin tuotteissa, mutta tekniset ja tyyllilliset kysymykset eivät aina vastaa toisiaan.

Molemmissa tuotteissa introanimaatiot pakkautuivat niin pieneen tilaan, että ne voidaan ladata latauspaketin mukana mobiilipelikaupasta. Tässä suhteessa projektien tavoite saavutettiin erinomaisesti. Galactic Echoesissa, jossa grafiikan koko oli suurta, yllätyin positiivisesti, sillä olin odottanut pelimoottoriin viedyn animaation koon olevan noin 20 Mb, mutta lopullinen animaation koko olikin 12 Mb.

6 LOPPUPÄÄTELMÄ

Opinnäytetyön projekteissa innovatiivisin osuus oli välianimaatioiden tuotantotapa. Vaikka välianimaatioita tuotetaan erilaisin työkaluin, jopa pelin sisäisin objektein, niin puhtaasti luuanimaatiolla toteutettuja täysanimaoituja välianimaatioita, joissa on paljon tuotantoarvoa, ei ole tuotettu paljoa. Turkulaisista peliyrityksistä muut kuin ne, jotka käyttävät minun animaatioitani, eivät ole tuottaneet vastaavalla tekniikalla välianimaatioita. Tulevaisuudessa tilanne saattaa olla toinen, sillä etenkin mobiilituotteisiin voidaan tällä tuotantotavalla tuottaa huomattavasti parempilaatuisia väli- ja tarinankerronta-animaatioita.

Tuotantotavan vaatimuksena on varteenotettavien työkalujen ja osaavien animaattoreiden lisäksi organisointikyky, suurten tietomäärien hallintataito ja käsitys pelimoottoreiden ja grafiikan välisestä yhteistyöstä. Lisää syvyyttä saa, mikäli osaa toteuttaa monipuolisesti varjostimia ja hallitsee hyvin partikkelipilvityökalut.

Jos toteuttaisin pelianimaatiot uudelleen, rajoittaisin palautekierrosten välistä aikaa jo sopimusvaiheessa. Tekisin myös enemmän luonnoksellisia pohjia animaatiokokonaisuudesta, joiden avulla animaatio rytmittäisiin. Näin rytmitykseen ei tarvitsisi puuttua palautekierroksissa sillä suurten luumäärien uudelleenorganisointi on aikaa vievää työtä. Kohtausten nopeuttaminen ei hoidu pelkästään avainruutujen välien skaalauksella, vaan kohtausten tapahtumat on organisoitava uudelleen. Hyödyntäisin myös tehokkaammin avainasennosta avainasentoon -metodia hahmo- ja ympäristöanimaatioissa, sillä animaatiotapa nopeuttaa huomattavasti työsykliä. Näin ei tarvitsisi työskennellä aamuyön pitkiä tunteja, vaan voisin käyttää ne tehokkaammin esimerkiksi nukkumiseen. Opin hallitsemaan aikaa huomattavasti aiempaa paremmin suunnilleen projektien puolesta välissä.

Prosessin aikana Galactic Echoes viimeisteltiin tuotteena ja laitettiin levitykseen. Sen kehä on sulkeutunut, eikä siinä ole enää jatkokehittävää ennen seuraavien episodien ilmestymistä, josta on mainittu alustavasti. Toteutettu animaatio on lopullinen tuote, ja se sai positiivista palautetta.

Sanalanka on projektina vielä vaiheessa, ja sen laajennusmahdollisuudet ovat hyvät. Peliin voi tuoda lisää hahmoja, tarinallisia animaatioita ja minipelisisältöä niin paljon, kuin teknologian rajoissa on mahdollista sisällyttää tuotteeseen. Animaatioiden suuri määrä on logopedien toive. Logopedien mukaan monipuolisella graafisella materiaalilla tuotteesta saadaan kiinnostavampi lapsille, joilla on vaikeuksia sisäistää kieltä. Monipuolinen ja elävä graafinen materiaali pitää heidät kiinnostuneina, vaikka verbaalinen sisältö ja tarkoitus eivät heti avautuisikaan. Tämä on ollut ongelma vanhoilla pahvikorttimateriaaleilla.

Tulevaisuudessa jatkan pelianimaatioiden parissa. Olen laajentanut animaatio-osaamistani 3D-animaatioon ja uskon, että löydän mielenkiintoisia tapoja yhdistää kaksi- ja kolmiulotteista animaatiota toisiinsa. Suomalaisessa pelinkehityksessä animaatio on kuitenkin vähemmän kilpailtu ala kuin vaikka ympäristö- ja konseptigrafiikka, joten uskon, että aihe työllistää jatkossakin.

LÄHTEET

Apple Developer Guide 2016. On-Demand Resources Essentials. Viitattu 30.11.2016. Setting Up On-Demand Resources > On-Demand Resources Essentials

Cartwright, M. 2014. Making Fluid and Powerful Animations For Skullgirls (GDC 2014). Viitattu 13.11.2016. <https://www.youtube.com/watch?v=Mw0h9WmBlsw>

Esoteric Software 2016. Spine User Guide 2016. Viitattu 03.05.2015. <http://esotericsoftware.com/spine-user-guide>

Johnston O. & Thomas F. 1995. The Illusion of Life: Disney Animation. 2. painos. Glendale: Disney Editions

KatsBits 2016. Better Textures For Games, "Power of Two" & Proper Image Dimensions. Viitattu 31.11.2016. <http://www.katsbits.com/tutorials/textures/make-better-textures-correct-size-and-power-of-two.php>

Rogers, S. 2010. Level Up! the Guide to Great Video Game Design. West Sussex: John Wiley & Sons, Ltd.

Sanders, A. 2015. Animating for Video Games vs. Animating for Movies. viitattu 05.06.2015. <http://animation.about.com/od/videogameanimation/a/gamesvsmovies.htm>

Suuronen, H. 2012. How to Make Killer Game Loops. viitattu 25.05.2015. http://www.gamasutra.com/view/news/173732/Video_How_to_make_killer_game_loops.php

UBM 2016. Ubi-Art cultivates both creativity and culture at Ubisoft. Viitattu 10.6.2015. http://www.gamasutra.com/view/news/200015/UbiArt_cultivates_both_creativity_and_culture_at_Ubisoft.php

Ukko, R. 2014. Making Games: Designer's Journey. Viitattu 26.05.2015. Luentosarja Aalto-yliopistossa: Monetization Course f2p Games.

Unity Technologies 2016. Particle Systems. Viitattu 12.11.2016. <https://docs.unity3d.com/Manual/ParticleSystems.html>

Vince, J. 2002. Handbook of Computer Animation. 2. painos. Lontoo: Springer-Verlag

Williams, R. 2002. Animator's Survival Kit. 2., laajennettu painos. Lontoo: Faber & Faber.