

KARELIA-AMMATTIKORKEAKOULU
Media-alan koulutus

Kuisma Myller

SPINE-OHJELMISTON KÄYTTÖ VIDEOPELIHAHMOJEN ANI-
MOINNISSA

Opinnäytetyö
Toukokuu 2018



OPINNÄYTETYÖ
Toukokuu 2018
Media-alan koulutus

Tikkarinne 9
80220 JOENSUU
+358 13 260 600

Tekijä(t)
Kuisma Myller

Nimeke
Spine-ohjelmiston käyttö videopelihahmojen animoinnissa

Toimeksiantaja
Karelia-amk

Tiivistelmä

Tässä opinnäytetyössä esitellään Esoteric Softwaren julkaisema Spine-ohjelmisto, ja kuinka ohjelmistoa käytetään 2D-hahmojen luomiseen pelimoottoriin. Opinnäytetyössä käydään läpi koko prosessi hahmojen ideoinnista lopulliseen testaamiseen pelimoottorissa.

Opinnäytetyössä käytän esimerkkeinä useita eri pelihahmoja Firetail Gamesin Drunkenpants-pelistä. Esittelen suunnitteluperiaatteita hahmojen taustalla, sekä yleisiä ja toiminnan kautta opittuja käytäntöjä. Keskeisenä teoriana käytetään animaation kahtatoista periaatetta. Opinnäytetyön sisältöön lukeutuu hahmojen konseptointi, animaation suunnittelu, hahmon valmistelu Spineen tuotavaksi, Spine-ohjelmiston käyttö ja animoinnin eri vaiheet, sekä hahmon tuominen Unity-pelimoottoriin.

Kieli
suomi

Sivuja 33

Asiasanat

2D -animaatio, videopelit, hahmosuunnittelu



THESIS
May 2018
Media degree

Tikkarinne 9
80220 JOENSUU
+358 13 260 600

Author (s)
Kuisma Myller

Title
Use of Spine software in video game character creation

Commissioned by
Karelia UAS

Abstract

This thesis showcases Spine-software made by Esoteric Software and how to utilize the program in animating 2D characters. This thesis will go through the process from character ideas and concept to final animated characters imported to the chosen game engine.

In the thesis I will use multiple characters from Firetail Games' Drunkenpants –video game. The design principles will be presented behind the characters, as well as practices and conventions discovered through the process. Central theory for animation conventions are based on the twelve principles of animation. The thesis content includes character design, animation design, preparation for export and importing final character files into a Unity-game engine project.

Language

Finnish

Pages 33

Keywords

2D animation, video games, character design

Sisältö

1	Johdanto	7
2	Animaatio	8
3	Hahmokonsepti	9
4	Pelihahmon piirtäminen	11
5	Animaation suunnittelu	14
6	Animaation periaatteet	15
6.1	Venytytys ja litistys	15
6.2	Ennakointi	16
6.3	Suoraan eteenpäin ja asennosta asentoon	17
6.4	Läpivienti ja päällekkäinen toiminta	18
6.5	Hidastus sisään ja ulos	19
6.6	Kaari	20
6.7	Toissijainen toiminta	21
6.8	Ajoitus	21
6.9	Liioittelu	22
6.10	Vetovoima	22
6.11	Kiinteä piirros	23
6.12	Lavastus	24
7	Spine-ohjelmisto	25
8	Setup-tila	25
9	Animate-tila	27
10	Drunkenpants-pelihahmot	28
11	Pelihahmojen animaatio	30
12	Unity	34
13	Tulokset	35
14	Pohdinta	37
	Lähteet	39

Sanasto

Atlas	Atlas on kuvatiedosto, johon Spine sijoittaa eri kuvat ja ottaa tietyt alueet pelimoottorissa kuvien renderöintiä varten (Esoteric Software 2018).
Attachment	Attachment on luuhun kiinnitettävä liite. Kun animoitavaa luuta käsitellään, attachment muuttuu sen kanssa. Yleensä attachmentit ovat kuvia, mutta voivat olla myös esimerkiksi bounding box. (Esoteric Software 2018.)
Clipping	Clipping -toiminnolla määritetään polygonialue, joka peittää alemmat piirtojärjestyksessä olevat kuvat (Esoteric Software 2018).
Curves	Graafinäkyvässä voi muokata käyriä, joiden avulla voidaan muokata keyframejen väliin piirrettävien ruutujen nopeutta, esimerkiksi lisäämällä keyframen alkuun ja loppuun hitaamman liikkeen kuin keskellä. (Esoteric Software 2018).
Dopesheet	Dopesheetissä on animaation nopeuteen ja animointitapaan liittyvät asetukset. Siitä voidaan määrittää ulostuotavan animaation ruudunnopeus, sekä luodaanko inbetweenejä ollenkaan. (Esoteric Software 2018.)
Freeform deformation	Tarkoittaa sitä, kun kuvalle määritellään polygon mesh, eli polygoniverkko, jota manipuloidaan. Kuvaa voidaan litistää, venyttää, käännellä yksittäisiä pisteitä (vertices) muokkaamalla. (Esoteric Software 2018.)

Ik Constraint	Inverse kinematics constraint luodaan kolmesta luusta. Pääluu, alaluu ja kohdeluu. IK constraint määrittää pääluun ja alaluun sijainnit kohdeluun perusteella, joka nopeuttaa animaatiota huomattavasti. (Esoteric Software 2018.)
JSON	JavaScript Object Notation. JSON on tekstitiedosto, jolla säilytetään tai vaihdetaan dataa (Introducing JSON 2018).
Keyframes	Avainkuvat, joiden mukaan animaatio toteutuu. Keyframet ovat Spinessä mitä tahansa animaatioon vaikuttavan työkalun tai arvon muuttamista. Keyframet vaaditaan, jotta Spine pystyy luomaan inbetween-kuvat. (Esoteric Software 2018).
Konseptiartisti	Vastaa videopelin visuaalisesta ilmeestä ja työskentelee usein pelisuunnittelijan kanssa yhteistyössä (Pardew 2009, 1).
Mesh	Mesh on kuvan sisään luotu polygoni, jota voidaan manipuloida. Tämä sallii kuvan liiskaamisen ja venyttelyn pisteiden manipuloinnin avulla. (Esoteric Software 2018.)
Path Constraints	Rajoittaa kuvan tai luun tietylle radalle (Esoteric Software 2018).
Polygon Mesh	Polygoniverkko. Polygonit ovat monikulmioita jotka määritellään kolmiulotteisilla pisteillä ja suorilla viivoilla, jotka yhdistävät ne (edges.) Viivojen sisään jää pinta (face). Nämä kolme ovat polygonin peruskomponentit. Yleensä polygonit ovat kolmipintaisia kolmioita. Monta pintaa yhdistämällä ne muodostavat polygoniverkon. (Autodesk Maya 2015.)
Region	Region on yksinkertainen, neliskulmainen kuva-attachment. Nimi tulee siitä, että ajon aikana kuva piirretään alueesta, eli regionista, kuva-atlaksesta. (Esoteric Software 2018.)

Runtime library	Komentosarjakirjasto eli ryhmä eri komentosarjoja, joita voi hyödyntää viittaamalla kirjastoon (Wikipedia 2018).
Script	Komentosarja eli automatisoitu ryhmä komentoja, jotka ohjelma suorittaa, jotka ovat mahdollista käyttäjänkin toteuttaa (Wikipedia 2018).
Skeletal animation	Tietokoneanimaatiossa käytetty tekniikka, joka koostuu kahdesta osasta: pinnoista ja luista. Pinta voi olla esimerkiksi hahmon visuaalinen osa, kuten käden grafiikka ja siihen kiinnittyy luu. Luita manipuloimalla animoidaan pinta. (Marionette Software 2016.)
Skin	Skin on kokonaisuus attachmentteja, joita voidaan käyttää samassa luurangossa. Skinejä voi olla useampia samalla hahmolla, joten voi tehdä eri variaatioita samasta hahmosta ilman, että tarvitsee luoda täysin uutta luurankoa ja animaatioita. (Esoteric Software 2018.)
Transform Constraint	Rajoittaa animaation kuvan tai luun eri arvot joko niin, että se on sidottu toiseen kuvaan tai luuhun, tai ettei sitä voi muuttaa millään tavalla. Tämä on hyödyllinen esimerkiksi pyörien animointiin. Sitomalla takapyörän etupyörään tarvitsee vain animoida toinen pyöristä. (Esoteric Software 2018.)

1 Johdanto

Videopelit ovat käyttäneet 2D-animaatiota ensimmäisistä konsoleista lähtien. Nintendo Entertainment System ja Sega Master System mahdollistivat esittävämmän grafiikan. Laitteiden kehittyessä myös grafiikka kehittyi. Super Nintendon ja alkuperäisen NESin välillä on suuri ero. Grafiikkaa alettiin puskemaan jo PlayStation järjestelmällä ja Segan uusi Saturn -konsoli muutti arkkitehtuuria 3D-yhteensopivaksi. 2D-grafiikka jäi elämään käsikonsolleille vielä tässä vaiheessa. Nykypäivänä 3D-pelien tekeminen on helpompaa kuin koskaan ennen lähes ilmaisten kehitysalustojen, kuten Unreal Engine ja Unity, avulla. Myös Blender ja monet muut 3D-mallinnusohjelmat ovat ilmaisia. Silti moni pelinkehittäjä käyttää 2D-grafiikkaa ja animaatiota. Moni kehittäjä kokee 2D-grafiikan mahdollistavan asioita, joita 3D-grafiikoilla ei pysty. 2D-pelien sarjakuvamainen grafiikka antaa kehittäjille vapauksia ja ei tarvitse pohtia, ovatko pelin grafiikat realistiset tai oikeat. 2D-grafiikka on myös edullisempaa pienten kehittäjätiimien toteuttaa kuin 3D. 2D-peleille on yhä kysyntää ja pelinkehittäjien vastuu on selvittää, minkälainen graafinen ilmaisu sopii heidän pelille parhaiten. (Ohlew 2014).

Tässä opinnäytetyössä kerron 2D-hahmojen animoinnista Spine-ohjelmalla. Spine on monipuolinen ja tehokas animointiohjelma, jota käytetään videopelialalla. Spinen kilpailijoita ovat mm. Spriter (www.brashmonkey.com), Nima (<https://www.2dimensions.com>), Dragonbones Pro (www.dragonbones.com). Esittelen työvaiheeni neljän eri pelihahmon avulla, jotka tein Firetail Gamesin Drunkenpants-videopeliin.

Opinnäytetyössä esittelen oman työprosessini pelihahmon animaatioon ja käytän eri hahmoja esimerkkeinä Spinen työkalujen käyttämiseen. Käyn läpi myös Spinen sisältämiä työkaluja ja miten niitä voi hyödyntää. Esittelen myös animaation periaatteita, ja mitä olen oppinut videopelien animoitujen hahmojen suunnittelusta. Käyn läpi prosessin alusta loppuun hahmojen konseptoinnista pelimoottoriin tuomiseen.

Sovelsin pelihahmojen animaatioon pitkälti perinteisen elokuva-animaation kah-tatoista periaatetta, jotka Frank Thomas ja Ollie Johnston esittelevät kirjassaan Disney Animation: The Illusion of Life (1984). Lopuksi käyn läpi, miten Spine-projektin voi tuoda Unity-pelimoottoriin ja siihen liittyviä toiminnallisuuksia. Opin-näytetyö toimii ohjeistuksena Spine-ohjelmiston käyttöön.

Spine on Esoteric Softwaren tekemä ja julkaisema 2D-animaatio-ohjelma, ja se on tarkoitettu ensisijaisesti videopelituotantoon (Esoteric Software 2018). Spinen lisäksi käytin Adobe Photoshopia hahmojen piirtämiseen, mutta sen käytöstä ker-ron vain animoinnin kannalta keskeiset asiat. Käytän pelihahmojen piirtämiseen Adobe Photoshopia sen monipuolisten työkalujen ja hyvän yhteensopivuuden Spine-ohjelmiston kanssa.

Esittelen lyhyemmin grafiikan tuomista Unity-pelimoottoriin ja ominaisuuksia mitä siellä voi käyttää, mutta en perehdy syvällisesti niihin, saati kerro ohjelmointiin liittyvistä asioista. Opinnäytetyö keskittyy 2D-hahmojen animointiin.

Pelien nykypäiväisestä 2D-animaatiosta on yllättävän vähän lähde-tekstejä, joten olen käyttänyt lähteinä pääsääntöisesti opinnäytetöitä, pelisuunnitteluoppaita, ja perinteisen animaatioelokuvatekstejä.

2 Animaatio

Frame-by-frame on animaatiotapa, jossa jokainen kuva piirretään erikseen. Pe-rinteinen animaatio on toteutettu tällä menetelmällä. Perinteisessä animaatiossa pyritään 24 kuvaan sekunnissa. Keyframe -animaatiossa toteutetaan hahmon liikkeen tärkeimmät kohdat, kuten toiminnan alku ja loppu. Frame-by-frame -ta-valla toteutettaessa artisti joutuu kumminkin tekemään jokaisen liikkeen välissä olevan kuvan itse. (Bloop Animation 2015.)

2D-animaatio mielletään usein pelkästään käsin piirrettyyn animaatioon, mutta se voi myös viitata tietokoneella toteutettuun vektorianimaatioon. Vektorianimaation suosioon 90-luvun lopulla vaikutti Flash-ohjelmisto, jota moni artisti käytti kaistarakoitusten vuoksi. Flash -tiedostot olivat usein lyhyitä, mutta pienikokoisia animaatioita. Flash-ohjelmisto mahdollisti perinteisen frame by frame -tavan lisäksi luoda rigejä, joiden avulla hahmon eri osia pystyi liikuttamaan itsenäisesti ilman, että koko kuvaa tarvitsi piirtää uusiksi. (Bloop Animation 2015.)

Keyframe-animaatiolla tarkoitetaan animaatiota, jossa määritetään keskeiset kuvat ensin ja niiden väliin luodaan sitten väliin piirrettävät kuvat (inbetween frames, kuva 1), jotka muodostavat kokonaisen animaation. (Thomas & Johnston, 1984.)



Kuva 1. Esimerkkikuva keyframeista. Animaattori määrittelee keyframet, jonka jälkeen ohjelma laskee väliin piirrettävät kuvat (läpinäkyvät hahmot).

3 Hahmokonsepti

Pelihahmon toteutus lähtee ideoista ja hahmokonseptista. Konseptiartistin vastuulla on visuaalisen ilmeen suunnittelu ja hän toimii pelisuunnittelijan kanssa yhteistyössä. Tässä on hyvä pitää mielessä, mitä projekti tarvitsee ja minkälaisia hahmoja halutaan. Peleissä grafiikka on ensimmäinen asia mihin kiinnitetään huomiota. Hyvä konseptitaide vaikuttaa pelin vastaanottoon, ja sitä voi hyödyntää markkinoinnissa tai julkaisijan hankkimisessa. Konseptiartistin haasteena on ideoida hyviä hahmoja ja hahmonluominen on aikaa vaativa prosessi. Artisti voi

tehdä useita luonnoksia, ennen kuin mieluinen hahmo löytyy. (kuva 2). Inspiraatiota voi hakea tarkkailemalla oikeaa elämää, tai seuraamalla muita artisteja ja medioita. (Pardew 2009,133-137.)



Kuva 2. Drunkenpants-pelin hahmojen ideointia.

Hahmosuunnitelman on tuotava esille hahmon luonne sekä henkiset ominaisuudet. Epäinhimilliset ja luotaantyöntävät hahmot, kuten pahat vastustajat tai hirviöt ovat usein helpompi keksiä, mutta pidettävien ja ihmismäisten sankarien luominen on vaikeata. Hahmon muoto viestii hahmon persoonallisuutta. Huomattavan ylipainoinen hahmo on harvoin sankari, sillä sankarilta oletetaan hyvää ulkonäköä. (White 2006, 30–36.)

Muodon lisäksi hahmon suunnittelussa on tärkeä miettiä mittasuhteita. Perinteisesti ihminen arvioidaan kahdeksan pään mittaa korkeaksi. Pää toimii mittayksikönä ja loppuruumis on seitsemän pään mittainen. Tosin sarjakuvamaiset hahmot noudattavat eri ”päänmittaa.” Tätä ajattelua soveltamalla tai etäntymällä kahdeksan pään mitasta saadaan liioitellumpia hahmoja. (White 2006, 37.)

Animoitavat hahmot kannattaa suunnitella useasta eri näkökulmasta, sillä keskittymällä vain yhteen kuvakulmaan hahmon piirteet eivät välttämättä erotu tai toimi muissa. Malliliuska (Model sheet) on niin sanottu pohjapiirustus hahmolle, joka määrittelee hahmon mittasuhteet, koon ja rakenteen. Tämän rakenteellisen malliliuskan lisäksi voidaan tehdä muita, jossa esitellään hahmon luonnetta tai eri asentoja. Malliliuskan lisäksi määritellään myös värimallit ja tarvittavat tekstuurit,

jossa voidaan myös määritellä piirustustapoja, kuten viivojen paksuus tai väritystavat. (White 2006, 38–39.)

Hahmon suunnittelussa on myös mietittävä muitakin projektin visuaalisia elementtejä, ja suunnittelussa otetaan huomioon etuala/taka-alayhteensopivuus. Tämä tarkoittaa, että hahmo sopii ympäristöihin. Tämän takia on tärkeää testata hahmoa ja eri ympäristöjä, ennen kuin viimeistelee tuotoksen. Tämä yhteensopivuus on erittäin tärkeää perinteisessä animaatiossa, mutta suotavaa myös muisakin medioissa. Jos hahmo ei sovi ympäristöön, hahmoa voi olla vaikea seurata tai lopputulos on muuten epäselkeä ja amatöörimäinen. (White 2006, 39-40). Videopeleissä on tärkeää, että pelaaja tunnistaa hahmonsä eri tilanteissa, joten etuala ja taka-alan yhteensopivuuteen kannattaa keskittyä huolella. Loppujen lopuksi hahmo- ja ympäristösuunnittelu on subjektiivista, eikä ole mahdollista tyydyttää kaikkia. Artistin pitää luottaa omaan näkemykseensä.

Pelihakmojen suunnittelussa on ehdottoman tärkeää pitää mielessä, että, muoto seuraa toimintaa. Pelihakmon ulkomuoto voi vaikuttaa myös pelisuunnitteluun. Hahmon ulkonäön pitää sopia peliin. Pelihakmojen persoonallisuutta pitää miettiä prosessin aikana ja löytää luonteenpiirteitä, jotka kuvaavat hahmoa hyvin. Nämä piirteet voi hyödyntää visuaalisessa ilmeessä, sillä muoto kertoo katsojalle hahmon persoonallisuudesta. Hahmoilla kannattaa olla selkeät siluetit, sillä niiden avulla pelaaja näkee nopeallakin vilkaisulla, kuka kukin on ja onko kyseessä vihollinen vai ystävä. Moninpelin hahmojen siluetit kannattaa suunnitella yhtä aikaa, että hahmot pysyvät samantyyppisinä, mutta yksilöinä. Siluetti auttaa myös irrottamaan hahmot taustasta. (Rogers 2014, 94–105.)

4 Pelihakmon piirtäminen

Ideoidusta hahmosta piirretään viimeistely ja yksityiskohtainen kuva. Hahmon kuva keskittyy yhteen hahmoon ja määrittelee ulkonäön lisäksi luonnetta. Hahmokuvalla kerrotaan tiimille tietoa hahmosta ja käytetään inspiraationa ja se toimii

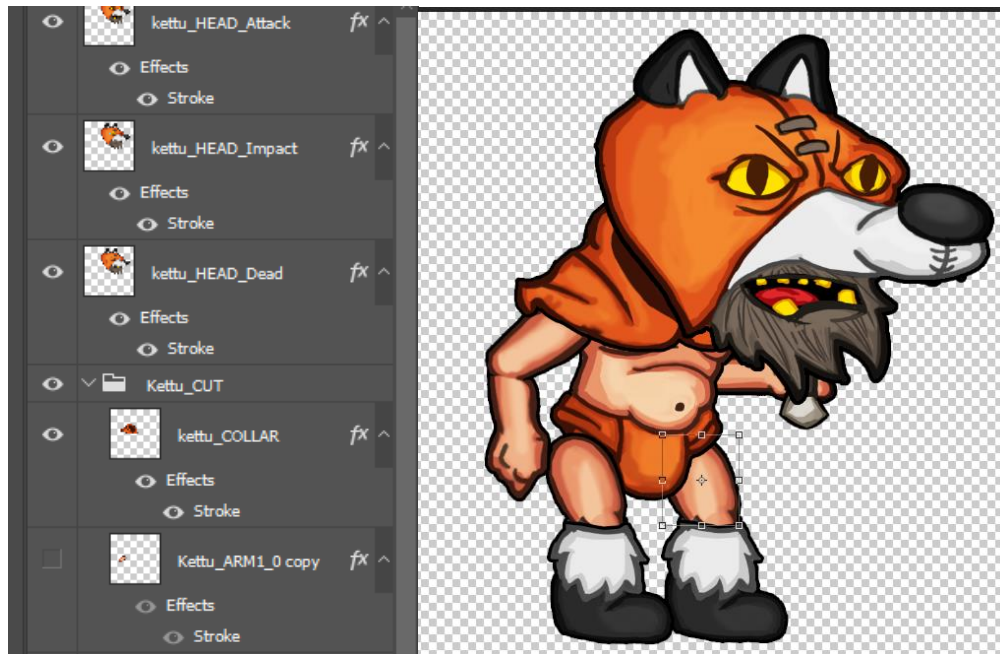
pohjana pelihahmon grafiikalle ja animaatioille. Niitä käytetään myös markkinointiin, joten selkeät ja viimeistellyt kuvat ovat suotavia. (Pardew 2009,153–154).

Pelihahmo piirretään valmiin konseptin pohjalta. Pelihahmo suunnitellaan staattiseen asentoon. Lähtökohtana pelihahmon asennolle paras on se pelihahmon "idle" -animaatio, eli se mistä siirrytään joka animaatioon.

Hahmon piirtämisen jälkeen hahmo paloittellaan. Paloitteluun on kaksi vaihtoehtoa: Joko luonnoksen päälle piirretään joka raaja yksitellen, tai sitten piirretään valmis hahmo, josta irrotetaan paloja ja korjataan leikkauksista jääneet virheet (Kukkonen 2011, 8). Itse koin jälkimmäisen paremmaksi tavaksi pitää hahmon mittasuhteet ja ilme yhtenäisenä, kun taas pala kerrallaan piirtämisellä hahmoista tuntui tulevan oudonnäköisiä, oudoilla mittasuhteilla varustettuja paperinukkeja. Piirtäjän kannattaa kokeilla kumpaakin tapaa ja valita niistä itselleen paremmin soveltuva.

Pelihahmon valmiista kuvasta aletaan animaatiosuunnitelman mukaisesti erottelemaan liikuteltavia osia. Esimerkiksi kädet, jalat, jalkapöydät, kämmenet, pää, kaula, ylä- ja alaruumis. Mitä useampi osa, sitä parempaa animaatiota saa aikaan, mutta siinä on myös enemmän työtä. Myös uusia paloja kannattaa luoda, esimerkiksi peittävät palat eri nivelten kohdille. Näin voi liikuttaa niveliä monipuolisemmin ilman, että palojen kääntyessä ja liikkuesssa jää rakoja hahmon eri kohtiin. (Kukkonen 2011,11).

Hahmo muodostuu lopulta eri tasoilla olevista raajoista, jotka kannattaa nimetä selkeästi (kuva 3). Esimerkiksi omassa työssäni nimeämiskäytäntönä olen pitänyt "hahmonnimi_raaja_numero." Tämä nopeuttaa seuraavaa työvaihetta, kun pelihahmon kuvat tuodaan Spine-ohjelmaan ja helpottaa työskentelyä, jos hahmoa työstää toinen graafikko.



Kuva 3. Kokonainen hahmo eri tasoille paloitetuna Photoshopissa.

Photoshopista kuvien tuominen Spineen on helppoa erillisen skriptin kautta, joka löytyy Spinen asennuskansiosta. Tätä käyttämällä kuvatiedostosta tulee valmis kansiorakenne ja Spinen tarvitseman atlas-kuvan (kuva 4) sekä JSON-tiedoston. Hahmo on myös Spinessä samassa asennossa, kuin se Photoshopissa on. Eri tasoryhmiä voi myös käyttää Spinen skin-ominaisuudessa.

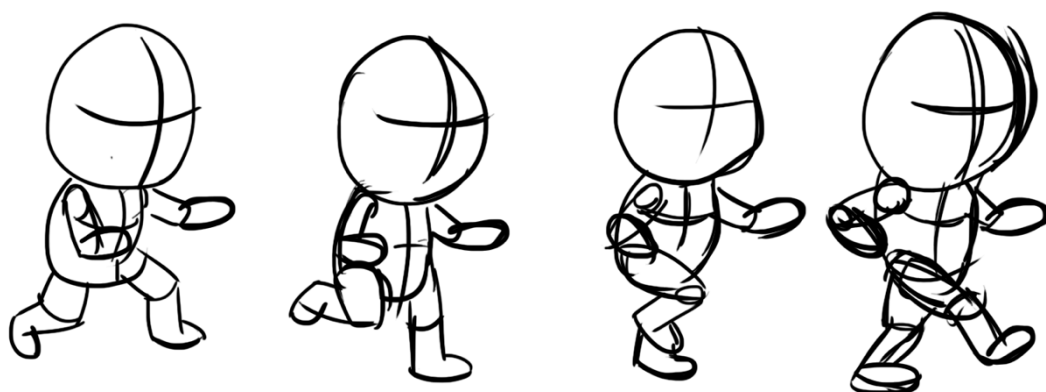
Spinessä hahmo on valmiiksi asennossa ja kuvat ovat piirtyvät sille korkeudelle kuin ne on Photoshopissa määritelty. Seuraava työvaihe on luoda hahmolle luuranko, jonka avulla animointi tapahtuu.



Kuva 4. Hahmon atlas-kuva, josta näkyy, kuinka hahmo on paloitetu.

5 Animaation suunnittelu

Animoinnin suunnittelun voi aloittaa, kun hahmon konsepti on lyöty lukkoon. Animaatiota on hyvä suunnitella ja miettiä etukäteen minkälaiset liikkeet voisivat tulla ongelmallisiksi (kuva 5). Hahmojen perspektiiviä voi myös rajoittaa. Tässä vaiheessa animaation suunnittelu kannattaa pitää hyvin yleisenä, koska kokeilemalla ja korjaamalla päästään parhaaseen lopputulokseen ja tämän takia lopullisen animaation tekeminen vaatii hahmojen tuomisen pelimoottoriin välillä. Hahmojen animaatioita kannattaa luonnostella ja kehitellä perinteisellä frame-by-frame -tavalla, luonnostelemalla tärkeimmät framet.



Kuva 5. Esimerkki juoksuanimaation suunnittelusta.

Yksinkertaiset animaatiot myös hyödyttävät siinä, kuinka pelimoottorin sisällä animaatioiden sekoittamista käytetään. Spinen komentosarjakirjastot tarjoavat mahdollisuuden ohjelmoinnin kautta sekoittaa animaatioita keskenään, esimerkiksi juoksuanimaation ja lyönnin. Tämä nopeuttaa animointia, sillä ei tarvitse tehdä erikseen animaatioita, kun hahmo lyö paikoillaan ja liikkeessä. Liian monimutkaiset animaatiot eivät toimi halutulla tavalla animaatioita miksaamalla, sillä miksaus yhdistää vain animaation keyframejen mukaan, missä luurangon osissa ei ole määritelty mitään toimintaa: Jos lyöntianimaatiossa on jalkojen luut määritelty

tiettyyn asentoon, lyöntianimaation ja juoksuanimaation sekoittaminen aiheuttaa sen, että jalat liikkuvat vain määrittämättömistä kohdista tai ominaisuuksista.

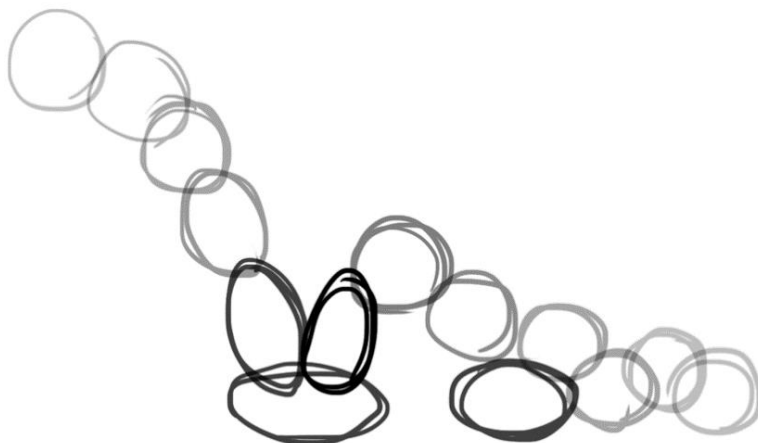
6 Animaation teoriaa

6.1 Animaation 12 periaatetta

Animaation kahtatoista periaatetta, jotka esitellään Frank Thomasin ja Ollie Johnstonin (1984, 30) kirjoittamassa *Disney Animation: the Illusion of Life*, voidaan soveltaa myös videopelihahmoihin. Kyseinen kirja kertoo kuinka Disneyn piirretyt elokuvat ja animaatio ovat kehittyneet, ja perehtyy syvällisesti työprosesseihin. Esittelen kaikki periaatteet ja kuinka mahdollisesti hyödynsin niitä omassa työssäni tai miten niitä voi soveltaa videopeleihin. Tärkeintä on muistaa, että liikkeiden ei tarvitse olla realistisia vaan "eläviä." Näitä periaatteita hyödynnetään etenkin perinteisessä animaatioissa, ja osa niistä sitoutuu pelimekaniikkaan. Pelin kehittäjän ja graafikoiden pitää yhdessä selvittää, kuinka niitä parhaiten soveltaa. Nämä kaksitoista periaatetta ovat: venytys ja litistys, ennakointi, suoraan eteenpäin ja asennosta asentoon, läpivienti ja päällekkäinen toiminta, hidastus sisään ja ulos, kaari, toissijainen toiminta, ajoitus, liioittelu, kiinteä piirros sekä lavastus.

6.2 Venytys ja litistys

Tosielämässä vain jäykät ja kovat esineet, esimerkiksi tuolit, pysyvät liikkeessä yhdenmuotoisina. Mikä tahansa elollinen liikkeessaan muuttaa muotoaan, esimerkiksi hauiksen pullistuminen kättä liikuttaessa tai rintakehän laajeneminen hengittäessä. On tärkeä huomata, että kuvaa venyttäessä tai litistäessä kuvan tilavuus pysyy samana, vaikka pituus tai leveys muuttuisi. Kuvan venyttäminen ja litistäminen tuovat luonnetta ja kertovat kuvan ominaisuuksista, esimerkiksi pomppivan pallon elastisuudesta (kuva 6). Venytyksen ja litistykseen liioittelu vaikuttaa myös toiminnan koomisuuteen. (Thomas & Johnston 1984, 15–19.)

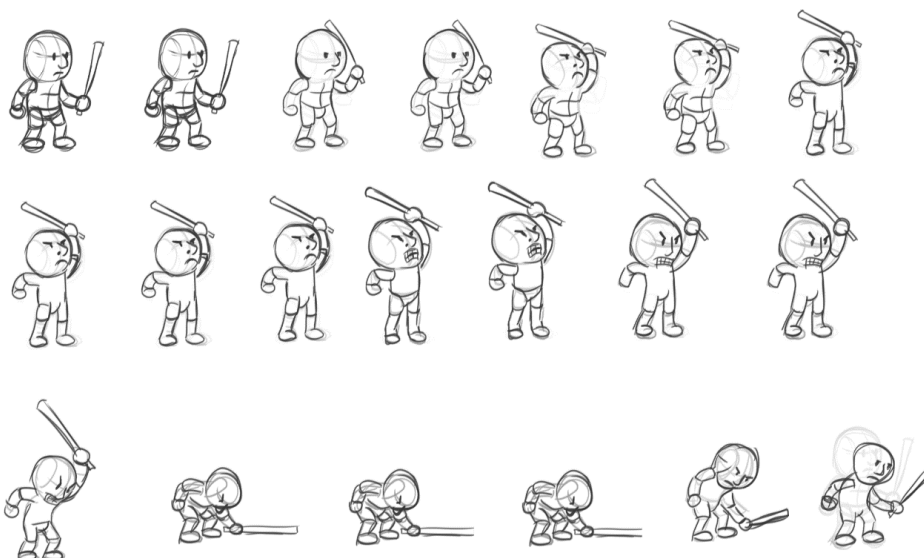


Kuva 6. Esimerkki venytyksessä ja litistyksessä pallolla, joka osuu maahan ja pomppahtaa. Pallon pomppimisen piirtäminen on hyvä harjoitus venytykselle/litistykselle.

Venytystä ja litistystä voi soveltaa monella tavalla. Omassa projektissa hyödynsin sitä etenkin hahmon tehdessä pyörähdyksen maassa. Tämä ei ole realistista, mutta tuo liikkeeseen eloa. Hahmo venyy hieman ensin syöksyessään kohti maahan ja maassa pyörähtäessään litistyy. Hienovaraisesti käytettynä se toi nopeaan liikkeeseen elävyyttä.

6.3 Ennakointi

Ennakoinnilla annetaan katsojalle tieto siitä, mitä tulee tapahtumaan hetkeä ennen päätapahtumaa. Ilman ennakointia katsojan on vaikea ymmärtää, mitä tapahtuu, jos toiminnoilla ei ole selkeää rakennetta tai jatkumoa. Ennakointi toteutetaan niin, että ennen suurta toimintaa on jokin pienempi liike sen edellä. (kuva 7). Liike voi olla pieni tai suuri, esimerkiksi ilmeen muuttuminen tai ponnistus juoksuun kyykystä. (Thomas & Johnston 1984, 19–20.)



Kuva 7. Lyöntiä ennakoidaan pitkään – hahmo nostaa hitaasti kepin pänsä yläpuolelle ja ilme muuttuu vihaiseksi, ennen kuin tuo sen alas.

Ennakointia käytin vaihtelevasti ja pienimuotoisesti, sillä tietyissä liikkeissä ennakoinnin priorisointi animaatiossa olisi vaikuttanut jo pelisuunnitteluun: lyöntianimaatiossa hahmo nostaa hyvin pieneksi hetkeksi aseensa pänsä yläpuolelle, ennen kuin tuo sen alas iskuna. Jos hahmo pitäisi ennakossa kättään todella pitkään, lyönnillä on vaikea osua kehenkään, sillä vastapelaajilla on liikaa aikaa reagoida lyöntiin. Joissain toiminnoissa jätin ennakoinnin kokonaan pois, sillä niiden pitää olla välittömiä tapahtumia napinpainalluksesta. Tästä on esimerkkinä pyörähdysanimaatio, jota käytetään pelissä iskujen ja hyökkäysten väistämiseen: liian pitkä viive tekee siitä vaikean ja turhauttavan käyttää pelaajalle.

6.4 Suoraan eteenpäin ja asennosta asentoon

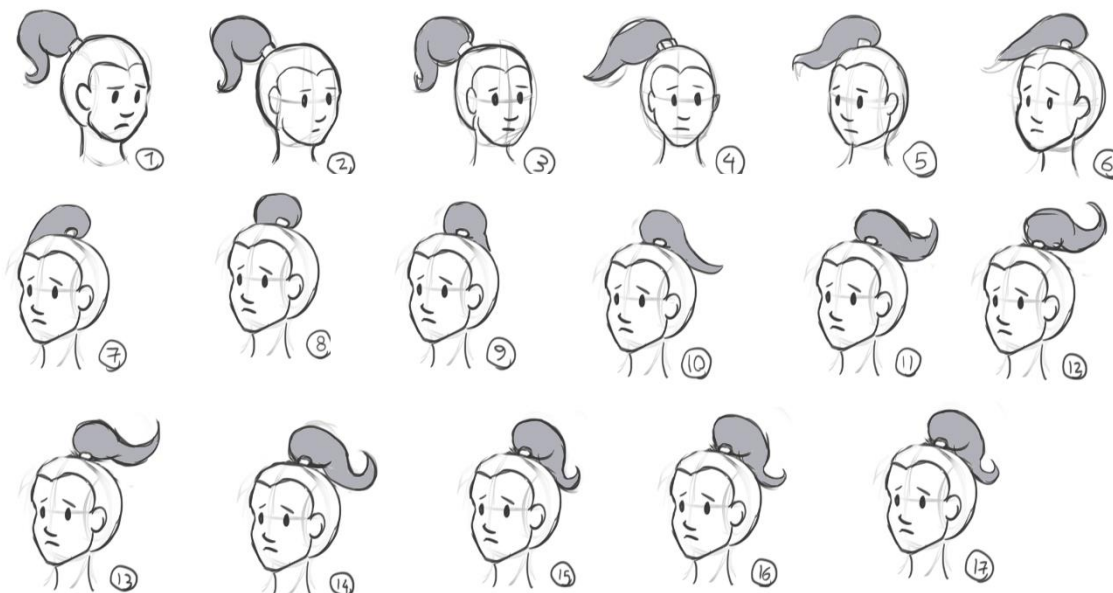
Animaatiota voi lähestyä kahdella tavalla. Suoraan eteenpäin (straight ahead) ja asennosta asentoon (pose to pose). Suoraan eteenpäin animoimalla tarkoittaa sitä, että animaatio työstetään ensimmäisestä piirroksesta suoraan eteenpäin. Asennosta asentoon animaattori suunnittelee toiminnan, kehittää tärkeimmät kuvat toiminnan esittämiseen ja luo väliin piirrettävät kuvat jälkikäteen. Asennosta asentoon toiminnalle tulee selkeyttä ja voimakkuutta, ja suoraan eteenpäin -animaatiolla spontaanisuutta. (Thomas & Johnston 1984, 24–26.)

Spine soveltuu työkaluna erittäin hyvin asennosta asentoon -animaatioon. Animaattorin tarvitsee asettaa animoitava hahmo vain tärkeimpiin kohtiin animaatioissa ja Spine luo väliruudut eli inbetweenit automaattisesti keyframejen välille. Suoraan eteenpäin -tavalla animaatioista voi tulla nykiviä ja huonosti rytmitettyjä, esimerkiksi juoksuanimaatiota animoidessa ensimmäisiä kokeiluja tehdessäni liikkeen rytmi oli outo: osa askelista oli nopeampia kuin toiset ja jalat nousivat liikaa tai liian vähän. Yleisesti animaatioissa suoraan eteenpäin on loistava tapa animoida spontaaneja ja eläviä asioita, kuten liekit nuotiossa tai savupilvi. Hahmoille koin järkevimmäksi toteuttaa animaatiot asennosta asentoon.

6.5 Lämpivienti ja päällekkäinen toiminta

Lämpivienti ja päällekkäinen toiminta on vaikea erottaa toisistaan, mutta on viisi eri luokkaa:

1. Jos hahmolla on mitä tahansa lisäkkeitä, kuten pitkät korvat jäniksellä tai ihmisellä pitkät hiukset, näiden liike jatkuu vaikka hahmo olisi muuten pysähdyksissä (kuva 8).
2. Keho ei liiku yhtä aikaa vaan kehon eri osat liikkuvat, kääntyvät ja kiristyvät vasten. Kun kehon osa on saavuttanut pysähdyspisteen, toiset voivat olla yhä liikkeessä. Ihmisen juoksusta pysähtyessä saattavat kädet vielä heilahtaa kerran, ennen kuin ihminen vie ne sivulle.
3. Hahmon löysät tai irralliset osat liikkuvat hitaammin kuin luisevat osat. Esimerkiksi lihavan hahmon vatsa hölskyy eri tahdissa kuin jalat liikkuvat juostessa.
4. Miten toiminta päättyy, kertoo hahmosta enemmän kuin itse toiminnan liike.
5. Liikkeen pysähtyminen. Liikkeen päättymisen korostaminen hienovaraisesti, esimerkiksi hahmon noustessa varpaille iloisesti hymyillen korostetaan nostamalla hahmo aivan ylös ja muuttamalla virnettä mahdollisimman äärirajalle, vaikka hahmo olisi muuten samassa asennossa. Tämä kertoo liikkeen sekä hahmon luonteesta. (Thomas & Johnston 1984, 27-30.)



Kuva 8. Hiusten heilahtaminen on hyvä esimerkki läpiviennistä ja päällekkäisestä toiminnasta. Pään kääntymisen jälkeen hiukset jatkavat vielä liikettä hieman.

6.6 Hidastus sisään ja ulos

Animaatiossa liikettä voi painottaa ja korostaa luomalla enemmän ruutuja liikkeen ääripäihin, esimerkiksi hahmon juoksussa hahmon ponnistava jalalla on enemmän ruutuja kuin jalan liikkussa ponnistuksesta ylös heilautukseen ja takaisin maahan. Ilman liikkeen hidastusta sisään ja ulos liikkeestä tulee luonnotonta ja mekaanista. (Thomas & Johnston 1984, 30.)

Luontevalle liikkeelle kuuluu, että liike alkaa ja loppuu hitaammin kuin mitä se liikkeen keskellä on. Tämä tarkoittaa myös sitä, että toiminnan ääripäissä on enemmän yksityiskohtia kuin liikkeen keskellä. Esimerkiksi jos hahmo nousee seisomaan, eniten ruutuja olisi alussa, ja kun hahmo nostaa itsensä lattialta, ja lopussa, kun hahmo on lähes pystyssä ja suoristaa selkensä. Liikkeen keskellä, lattialla istumisen ja seisallaan ollessa, on vähiten liikettä.

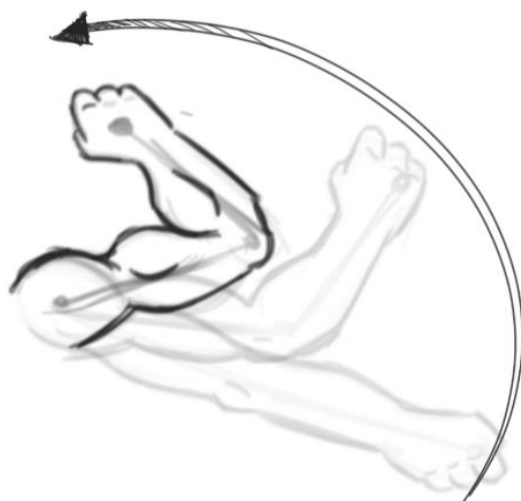
Liikkeen hidastus äärikohdissa oli erittäin olennaista pelihahmojen elävyyden luomiseksi. Jos jalkojen ja käsien liike on tasaista heilumista juoksussa, hahmo näyt-

tää enemmän robotilta kuin ihmiseltä. Kuinka paljon liikkeen äärikohtille luo hidastusta, vaikuttaa myös pelihahmojen liikkeeseen. Omassa työssäni koin sopivana pienen hidastuksen määrän, koska pelihahmojen jalat liikkuvat nopeasti, ja liikettä testatessa hahmot ”vipelsivät” huvittavalla tavalla, joka sopi pelin koomiseen tyyliin.

6.7 Kaari

Hyvin harva elävä olento tekee liikettä, jolla on selkeä liikerata pisteestä pisteeseen. Kaikki liike noudattaa jonkinlaista kaarta. Liike, joka menee pisteestä pisteeseen, tuntuu konemaiselta. Käden liikuttaminen ylös ja alas ei ole suora viiva, vaan siinäkin on kaarimainen liike, koska nivelen liikerata sallii kaarevan liikkeen (kuva 9). Joitain poikkeuksia on, mutta lähes kaikessa hahmon animaatioissa sopii miettiä ja suunnitella liikkeet kaarevasti. (Thomas & Johnston 1984, 30.)

Kaarevuus liikkeissä tuo elävyyttä hahmojen toimintaan. Ensimmäisissä animaatioissani liike näytti usein pahalta sen takia, että liike eteni suoraan aloituspisteestä loppuun ilman nousua tai laskua. Liikkeen kaartuvuutta oli myös vaikea hahmottaa alkuun. Omassa työssäni auttoi tutkia erilaisia liikkeitä, kuten heiluttaa kättään peilin edessä.



Kuva 9. Hauiksen ojennus ja koukistus on kaareva liike.

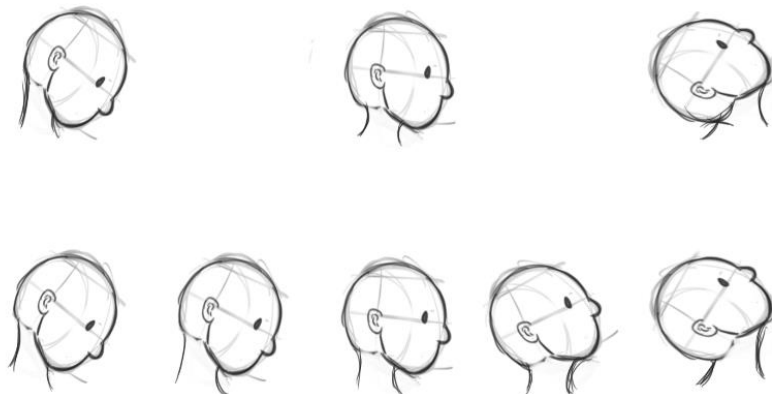
6.8 Toissijainen toiminta

Toissijaisella toiminnalla tarkoitetaan liikkeitä, eleitä ja toimintoja, jotka tukevat pääasiallista toimintaa. Jos toissijaisesta toiminnasta tulee liian kiinnostava tai se vie huomion tärkeämmältä toiminnalta, animaation viesti katoaa. Esimerkkejä toissijaisesta toiminnasta on, kun surullinen hahmo poistuu ja pyyhkäisee kääntyessään kyyneleen poskeltaa. Toissijainen toiminta kannattaa myös ajoittaa niin, ettei se sekoita tärkeämpää toimintaa. Useimmiten toissijainen toiminta kannattaa aloittaa vain hieman ennen tai jälkeen ensisijaista toimintaa. (Thomas & Johnston 1984, 31–32.)

Toissijainen toiminta kannattaa toteuttaa tärkeemmän toiminnan jälkeen animoinnissa. Omassa työssäni huomasin, että yksityiskohtiin keskittyminen liian aikaisin aiheuttaa ongelmia animaatiossa ja lopputulos voi olla sekava. Pääasiallinen toiminta auttaa myös miettimään, kuinka toissijaiset liikkeet toteutetaan.

6.9 Ajoitus

Ajoituksella tarkoitetaan, kuinka monta kuvaa liikkeeseen käytetään (Kuva 10). Se määrittelee animaation pituutta. Sama toiminta eri kuvamäärillä vaikuttaa toiminnan luonteeseen. Pään kääntyminen vasemmalta oikealle lyhyessä ajassa, esimerkiksi kahdella kuvalla, vaikuttaa siltä, että päähän osuu suurella voimalla jotain. Useammalla kuvalla taas sama pään kääntyminen voi näyttää venyttelyltä. (Thomas & Johnston 1984, 33.)



Kuva 10. Kuvien määrä vaikuttaa liikkeen luonteeseen.

Pelianimaatiossa on tärkeä miettiä ajoituksen vaikutusta pelituntumaan sekä pelimekaniikkoihin. Liian pitkä lyöntianimaatio voi tehdä lyöntitoiminnallisuudesta hyödyttömän tai jopa haitallisen sitä käyttävälle pelaajalle, jos lyöntianimaatio esittää muun toiminnan lyönnin ajaksi.

6.10 Lioittelu

Toiminnan liioittelu tuo toimintaan uskottavuutta ja persoonallisuutta. Esimerkiksi lihava hahmo vetää syvään henkeä ja hahmo paisuu kuin ilmapallo. Tämä ei ole realistista, mutta se korostaa liikettä ja tuo toimintaan persoonallisuutta ja kertoo paljon hahmosta ja tilanteesta. (Thomas & Johnston 1984, 31–32.) Jos samaa henkeään pidättelevää hahmoa lyötäisiin vatsaan, hän saattaisi päästää ilmat ulos nopeasti huulet väristen ja litistyä kuin tyhjenevä ilmapallo. Lioittelun määrä vaikuttaa myös tunteeseen, jota toiminta välittää. Lioittelulla voidaan saada koomisuutta toimintaan.

Pelianimaatiossa liioittelua sopii käyttää, jotta liikkeen ja toiminnan tunnistaa varmasti kaiken muun toiminnan seasta. Liian pelkistetyt toiminnot eivät erotu jatkuvasti muuttuvasta ympäristöstä ja tilanteesta, ja pelaajalle voi jäädä epäselväksi, mitä mikäkin hahmo tekee.

6.11 Vetovoima

Vetovoima liittyy hahmosuunnitteluun ja animaatioon. Vetovoimalla tarkoitetaan sitä, että hahmo ja toiminta miellyttää silmää, oli kyse pahasta ja rumasta hahmosta tai suloisesta ja hyvästä hahmosta. Kaikilla hahmoilla ja niiden käyttäytymisellä pitäisi olla vetovoimaa. Huonot piirustukset ja animaatiot eivät herätä mitään tunteita. Vetovoimainen animaatio kiinnostaa katsojaa ja pelkästään toiminnan katselu herättää mielenkiintoa. (Thomas & Johnston 1984, 36.)

Vetovoimaisuus on subjektiivista ja jokaisella on omat mielipiteensä siitä, mikä näyttää hyvältä. Hyvin suunnitellut hahmot kuitenkin herättävät mielenkiintoa ja ajatuksia. Drunkenpants-projektissa esimerkiksi kehitysryhmä näki kettupäähiineeseen pukeutuneen hahmon hauskana ja mielenkiintoisena hahmona, joka halettiin peliin mukaan (kuva 11).



Kuva 11. Piirros kettuhahmosta.

6.12 Kiinteä piirros

Perinteisessä animaatioissa pitää osata piirtää hahmoja useista eri kulmista ja asennoista. Animaattori pystyy esittämään vain piirtämistaitonsa verran eleitä, ilmeitä ja liikkeitä.

Animaattorin piirrosten pitää tuntua siltä, että ne ovat kolmiulotteisia: niillä on pitää olla paino, syvyys ja tasapaino. Hahmojen pitää sopia tilaan, missä he ovat.

Vaikka puhutaan kaksiulotteisista kuvista, tärkeätä on luoda kolmiulotteinen vaikutelma. Hahmon painoon, syvyyteen ja tasapainoon vaikuttaa myös niiden toiminta. Jos hahmo tekee raajoillaan, kuten käsillä, samanlaista liikettä samaan aikaan, puhutaan ”twinningistä.” Tämä on epäluonnollisen näköistä liikettä. (Thomas & Johnston, 21-24).

Luurankoanimaatiossa tätä voi olla vaikea soveltaa. On helppo sortua paperinukkejen oloisiin hahmoihin. hahmot ovat samasta kuvakulmasta ja ne on koottu luurangon päälle. Kolmiulotteinen olemus riippuu piirtäjän taidosta ja taidetyylistä.

6.13 Lavastus

Lavastus (staging) on hyvin yleisesti soveltuva periaate, koska se viittaa ja esiintyy monessa asiassa. Periaate on, että toimintaa tai ideaa esitetään niin, että sen viesti on täysin selkeä. Esimerkkinä voisi olla, jos animaatioon halutaan pelottava tunnelma, lisätään pelottavia symboleja tai perinteisiä asioita, kuten taustaksi ränsistynyt talo ja vääristyneitä puita. Toiminnan esimerkkinä olisi se, että hahmo potkaisee. Silloin ei leikata kuvaa hahmon vyötäröstä alaspäin, ja piiloteta potkun liikettä ulos ruudusta. Pääasiana on ohjata mitä katsoja näkee. (Thomas & Johnston 1984, 34-36.)

Lavastus on vaikea sovellettava videopelihin, sillä se nojaa eniten perinteiseen tarinankerrontaan elokuva-animaatiossa. Videopelien interaktiivisen luonteen takia ei voi määrätä tarkalleen, kuinka asiat tapahtuvat ja missä järjestyksessä. Tosin lavastuksen periaate, että ohjataan sitä mihin halutaan kiinnittää huomiota, voidaan soveltaa pelin. Iso ja vaarallinen hyökkäys voidaan haluta korostaa animaatioilla, äänillä ja efekteillä. Jos pelaajaa halutaan ohjata tekemään tietty asia, kuten pysymään tietyllä alueella, voidaan korostaa kenttägrafiikalle haluttu alue tai keksiä muu visuaalinen apukeino. Drunkenpants-pelissä esimerkkinä olisi, kun peli päättyy jonkin pelaajan voittoon: Voittajan hahmoon keskitetään kamera ja korostetaan valokeilaefektillä (kuva 12). Tämän tukena on myös teksti sekä äänet korostamassa, että peli on päättynyt ja voittaja ratkennut.



Kuva 12. Drunkenpants-pelin voittotilanne.

7 Spine-ohjelmisto

Spineä ohjelmana voi kokeilla ilmaiseksi, mutta omia töitään ei voi tallentaa, saati tuoda ulos Spinestä. Esoteric Software tarjoaa myös Essential, Professional sekä Enterprise-lisenssejä. (Esoteric Software 2018). Oman työni olen tehnyt Professional-lisenssillä. Spinestä kiinnostuneen kannattaa huomata, että Essential-lisenssistä puuttuu esimerkiksi meshin luomiseen ja manipulointiin liittyvät työkalut.

Spinen käyttö jakautuu kahteen eri työtilaan, Setup- ja Animate-tilaan. Setup-tilassa tuodaan hahmon vaatimat kuvat, ja muu data. Setup-tilassa luodaan myös hahmolle muut tarvittavat ominaisuudet, joilla itse animointi tapahtuu, kuten hahmon luuranko ja niiden riippuvuudet, erilaiset rajoittimet, ja kuvan verkon luominen. Animate-tilassa hyödynnetään näitä eri ominaisuuksia.

8 Setup-tila

Kun kuvat on tuotu Spineen skriptin avulla, luodaan hahmolle luuranko. Luurankon luihin kiinnitetään kuvatiedostot, jonka jälkeen luuta liikuttamalla kuva liikkuu

samalla tavalla. Luurangon luut noudattavat hierarkiaa, jossa ylempi luu vaikuttaa alempiin luihin, esimerkiksi ylemmän luun pyörittäminen liikuttaa siinä kiinni olevaa alempaa luuta. Spine osaa nimetä luut automaattisesti kuvan nimen perusteella, joten siksi edellisessä työvaiheessa tehty nimeäminen helpottaa työtä huomattavasti. Luut voi myös värjätä selkeyden kannalta. Kuvatiedostoja, jotka ovat luussa kiinni, voi vaihtaa eikä ole tarvetta luoda uutta luurankoa.

Valmiin luurangon lisäksi kuvatiedostot voi muuttaa meshiksi eli verkoksi, jolloin kuvaan muodostuu polygoneja (kuva 13). Tämä on hyödyllistä optimointimielessä, mutta myös mahdollistaa myös niiden venyttelyn ja muokkaamisen monipuolisesti. Tällä tavalla voi tehdä esimerkiksi luontevan liikkeen poninhännälle, venyttää ja pyöristää kumipalloa tai antaa kolmiulotteisen vaikutelman liikuttamalla meshin keskeisiä verteksejä.



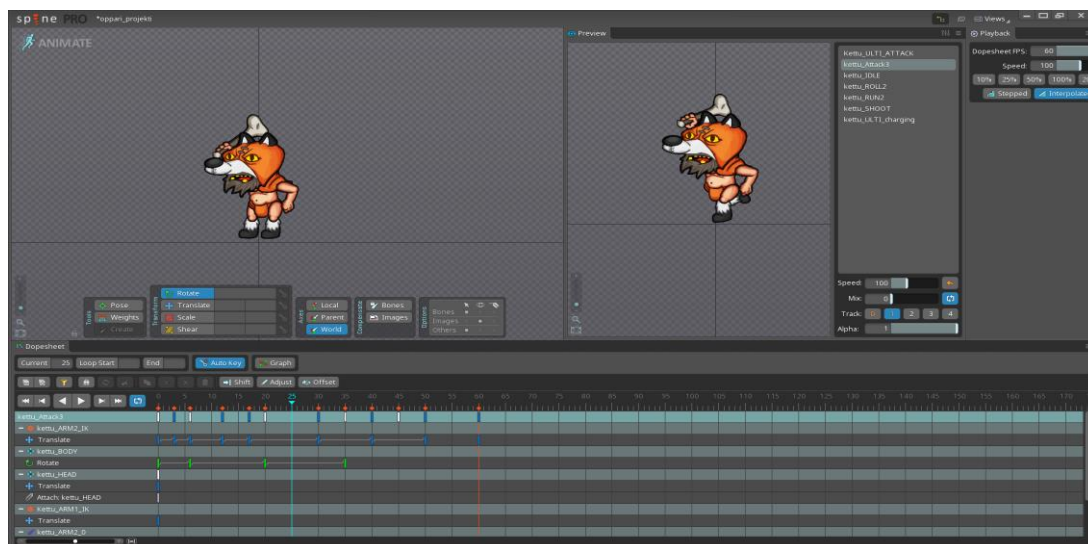
Kuva 13. Hahmon pään päällä on keskeneräinen mesh-verkko.

Inverse Kinematics Constraint on erittäin hyödyllinen esimerkiksi raajojen animointiin, sillä se estää ylitaipumisen tiettyyn suuntaan eikä anna luiden mennä täysin päällekkäin. IK simuloi suhteellisen hyvin nivelten realistista taipumiskulmaa, ja sillä voi vain yhtä luuta liikuttamalla tuottaa realistisempaa liikettä.

9 Animate-tila

Animaatiot itsessään luodaan animate-tilassa. Tässä tilassa hahmoa voi manipuloida miten haluaa, oli se sitten vain luiden liikuttamista tai kuvien vaihtamista, tai meshien manipulaatiota (kuva14).

Spinessä on helppo luoda useita eri animaatioita samalle hahmolle. Hahmolle tarvitaan vain keskeisimmät kuvat animaatioissa eli keyframet ja ohjelma luo niiden väliin tarvittavat tween-kuvat. Animaatioita voi myös sekoittaa Unity -pelimoottorissa eli yhdistellä kahta tai useampaa eri animaatiota. Tämä nopeuttaa työprosessia huomattavasti, sillä hahmolle ei tarvitse luoda samasta animaatiosta eri variaatioita, esimerkiksi hahmon lyönti paikallaan ja hahmon lyönti juostessa.



Kuva 14. Esimerkki animate-tilasta. Vasemmalla on hahmon näkymä, oikealla on preview-ikkuna animaatioiden sekoittamisesta, sen vieressä dopesheet ja alhaalla aikajana.

Keyframien luominen tapahtuu aikajanalla. Keyframe on yksinkertaisimmillaan luun liikuttamista koordinaatistossa. Keyframeja voi luoda mille tahansa hahmon luurangon ja luiden ominaisuuksille. Esimerkiksi hahmon kättä voi pyöräyttää luomalla ensimmäiselle framelle keyframen, ja muutaman framen päähän uuden keyframen, jossa hahmon käden luuta on pyöräytetty 180 astetta. Puuttuvat tweenit ohjelma luo automaattisesti.

10 Drunkenpants-pelihahmot

Hahmon luonti alkaa ideasta ja pelisuunnitelman tarpeista. Tässä peliprojektissa tarvitaan vähintään neljä erilaista pelaajahahmoa, joten keskeinen suunnittelukysymys on luoda yhtenäinen ilme hahmoille ja että hahmot erottuvat toisistaan. Pelin teemana on "suomalaiset juntit" taistelemassa keskenään arkisista asioista liiallisella väkivallalla. Pelin teema on humoristinen ja hahmoissa pitäisi tämä myös näkyä.

Ensimmäiset luonnokset tein aivan mielikuvituksen ja mahdollisten referenssien pohjalta (kuva 15). Tässä vaiheessa on vapaat kädet ja tärkeintä on vain tuottaa nopeasti ideoita ja löytää ne parhaat. Tärkeää on olla estoton omien piirustusten suhteen. Ideoita kannattaa luoda useammallekin hahmolle ja tutkia piirtämällä eteenpäin, mitkä hahmot ovat parhaita. Valinta perustui paljolti siihen, mitkä miellyttävät silmään eniten ja missä hahmossa on eniten persoonallisuutta.



Kuva 15. Esimerkkejä Drunkenpants -hahmojen ideoinnista

Hahmoja piirtäessä tärkeimpänä pidin mielessä yleiset mittasuhteet. Vaikka osa hahmoista olisikin lihavia, lihaksikkaita tai erittäin laihoja, tietyt periaatteet päätin

heti alkuun. Hahmojen tunnistettavin osa on pää, joten se on iso hahmosta riippumatta. Pään ollessa jopa kolmasosa hahmon "pinta-alasta" ei jää paljoa tilaa yksityiskohdille muualla kehossa.

Pelihakmon piirsin noin neljä kertaa isompana, kuin mitä se pelissä tulee olemaan. Tässä työssä käytin piirtoalueena 1000 kertaa 1000 pikseliä, jolloin hahmosta tulee noin 250 kertaa 250 pikselin kokoinen. Tämä auttaa myös pienten virheiden ja poikkeavuuksien hävittämiseen, kun piirtää aluksi isomman kuvan. Jos tarvitsee myös vaihtaa hahmon kokoa pelissä, koon skaalaaminen isommaksi ei muuta hahmon grafiikkaa epätarkaksi.

Pelin kuvakulma ja mittasuhteet vaikuttavat myös hahmon ulkonäköön. Kyseisessä pelissä peliruutu ei vieri ja kaikki informaatio näkyy koko ajan. Peli on suunniteltu nopeatempoiseksi ja refleksejä vaativaksi, joten grafiikka ei saa tulla missään vaiheessa esteeksi pelitilanteen ymmärtämisen suhteen. Tärkeä myös suunnitella hahmot niin, että ne tunnistetaan välittömästi eivätkä pelaajat sekoita niitä keskenään. Tässä apuna auttaa pelihakmojen siluettien suunnittelu ja eri päävärien käyttö.

Pelihakmon piirtämisessä pyrin myös piirtämään ääriiviivat mahdollisimman tarkkaan ja saman paksuisena (kuva 16). Valmista kuvaa voi myös pienentää ja käyttää testikuvissa, esimerkiksi mockup-kuvankaappauksessa, ja miettiä mitkä asiat kuvassa toimivat ja mitkä eivät. Tässä projektissa huomasin, että vahvalla sävytyksellä saa paremmin hahmon sisäpinnan ääriiviivat esille, kuin mustalla ääriviivalla. Hahmojen ulkoiset ääriiviivat pidin kuitenkin mustina, koska koin sen auttavan hahmojen erottamisen taustasta. Tässä projektissa on myös hyvä muistaa, että hahmot ovat niin pieniä, että tietyt asiat eivät erotu pelin ajan aikana.



kuva 16. Piirustuksia hahmoista.

Vaikka hahmot ovat erinäköisiä, on hyvä olla erilaisia yhdistäviä tekijöitä. Visuaalinen tyyli on tärkeä pitää yhtenäisenä. Pelin teemaan ja huumoriin sopii, että kaikki hahmot ovat alusvaatteisillaan. Tämä on lievä rajoitus hahmojen ulkonäköä suunniteltaessa, mutta auttaa pitämään ilmeen yhtenäisenä. Suunnittelun ja ideoinnin jälkeen luon lopullisen konseptin hahmolle. Konseptin piirrettyä aloitan myös miettimään hahmojen animointia, kuinka hahmot liikkuvat tai lyövät.

11 Pelihahmojen animaatio

Projektin pelihahmoille tein viisi animaatiota, jotka olivat: idle, run, attack, roll ja shoot. Tämä on tarvittava vähimmäismäärä, mutta erilaisia siirtymiä ja variaatioita on mahdollista tehdä, esimerkiksi useampi attack -animaatio, joita vaihdella, kun pelihahmo hyökkää. Pelin kohderuudunnopeus on 60 ruutua sekunnissa. Aloitan mieluiten idlestä, sillä se on oletettava perustila, mihin hahmo aina palautuu ja mistä kaikki liike myös alkaa.

Pienten hahmojen kohdalla mietin myös animaation tärkeimpiä asioita. Pelisuunnitelman mukaisesti kaikki liikkeet tulevat olemaan nopeita. Animaation yksityiskohtaisuus voi myös haitata toiminnallisuutta: siinä missä oikeasti maassa pyörähtäen palautuminen taas pystyasentoon on hieman hidasta, pelissä hahmot pyörähtävät nopeasti ja jatkavat samalla nopeudella. Tämän takia pyörimisanimointiin en edes suunnitellut palautumista, tai ennakoi liikettä millään tavalla.

Pienissä hahmoissa kannattaa muistaa, että yksityiskohtia ei oikein huomaa. Kaikkia liikkeitä ei kannata korostaa ja koin parhaaksi keskittyä siihen tärkeimpään elementtiin. Liian monta eri liikettä voi aiheuttaa myös sen, että hahmon toimintaa ei tunnista tai se näyttää ikävältä. Esimerkiksi lyöntianimaatiossa tärkeimpänä oli lyövän käden liike.

Hahmoille luon perustarpeiden mukaan yhteisiä asioita ja pyrin pitämään hierarkian ja nimeämisen samana: jos animaatioita tarvitsee hakea Unity -pelimoottorissa, ohjelmoijan ei tarvitse etsiä kauaa. Hahmoille tulee luuhierarkia, IK constraintit jalkoihin sekä käsiin ja mesh tarvittaessa päähän sekä keskivartaloon. Joka hahmolle tuon myös pelin eri aseista kuvia, jotta niitä voidaan vaihtaa pelimoottorin sisällä.

Idle-animaatiossa hahmo on vain paikallaan, joten animaatiossa ei tarvitse olla kuin pientä liikettä (kuva 17). Jokainen hahmo vain seisoo paikalla ja hengittää. Tämän toteutin vain siirtämällä hahmojen käsiä ja päätä hieman ylös ja palauttamalla takaisin. Keskivartaloon tehtyä meshiä litistämällä ja suurentamalla sain rintakehän liikkumaan niin, että hahmot näyttävät hengittävän.



Kuva 17. Idle-animaatiossa ei tarvitse olla paljon liikettä, koska hahmo ei tee mitään. Pienillä liikkeillä, kuten pään liikuttamisella ja vatsan venyttämällä hahmo näyttää hengittävän.

Run-animaatio vaatii enemmän tekemistä: Juokseminen, tai käveleminen, on suhteellisen vaikea perusanimaatio, joka jokaisen animaattorin kannattaa opetella (kuva 18). Se on myös hyvä esimerkki monista eri animaation periaatteista. Run-animaatiossa piti myös huomioida pelihahmojen liikenopeus pelissä ja ympäristössä. Lopullinen animaatio hahmoilla näytti koomiselta, jalkojen vipeltäessä nopeasti. Aluksi jalkojen animaatioiden lisäksi käytin aikaa käsien animoimiseen, että ne seuraavat juoksuanimaatiota oikein. Tässä tuli se ongelma vastaan testatessa, että animaatioiden sekoittaminen Unityssa ei toiminut oikein. Juoksuanimaatiossa määritetyt käsien keyframet menivät esimerkiksi lyöntianimaation keyframien päälle, ja lopullinen sekoitettu animaatio näytti hirveältä ja lyönti näytti

nykivältä heilumiselta. Käsien heilunta pitäisi tehdä omaksi animaatioksi. Jouduin siis poistamaan käsien liikkeen, että juoksua voi sekoittaa lyönnin ja ampuksen kanssa halutulla tavalla.

Juoksuun saa helposti myös hidastuksen sisään ja ulos käyttämällä Spinen curve -työkalua. Tämä muokkaa inbetween-kuvien ajoitusta kuvaajakäyrän mukaisesti, eli painottaa liikettä käyttäjän määrittämällä tavalla.

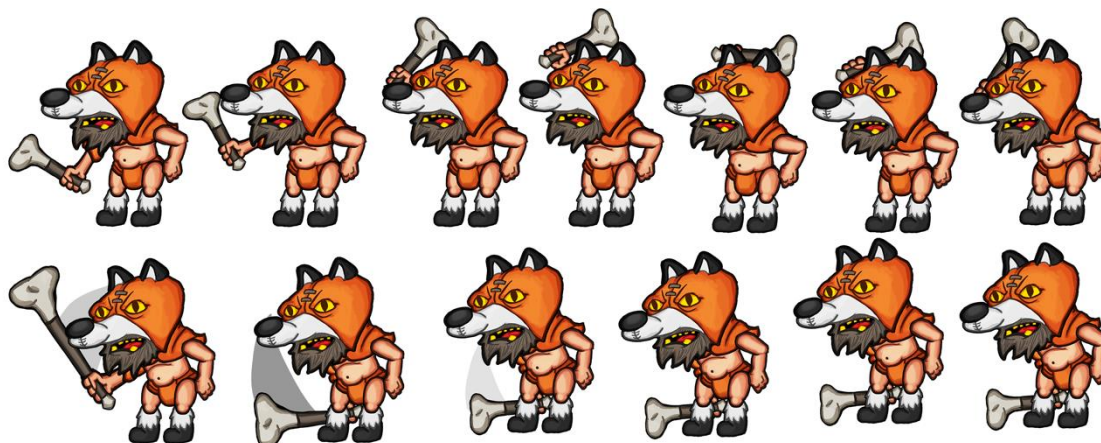


Kuva 18. Esimerkki juoksuanimaatiosta.

Aloitin juoksuanimaation tekemisen tärkeimmistä keyframeista. Tähän on hyvä etsiä referenssi, tai piirtää itse suunnitelma perinteisemmällä animaatiolla. Runanimaatiossa oli aluksi 12 keyframea. Tässä kuitenkin huomasin, että Spineohjelmiston luomat tween-kuvat käyttäytyivät hieman oudosti ja hahmon liikkeistä tuli nykivää. Muokkaamalla kahtatoista alkuperäistä keyframea ja luomalla uusia tarvittuihin kohtiin sain haluamani liikkeen tasaisuuden.

Attack-animaatiossa tärkein elementti oli saada lyöntiin voimantunnetta ja napakkuutta. Tässä käytin eniten ennakointia ja palautumista hyödyksi. Ensimmäisissä testeissä animaatiossa keskityin liikaa itse lyöntiin, kun ase laskeutuu hahmon yläpuolelta maahan: tämä teki liikkeestä laiskan ja voimattoman. Mitä vähemmän lyönnissä oli ruutuja, sitä napakamman iskun sain aikaiseksi. Hahmon nostaessa asetta ja palautuessa lyönnissä oli eniten ruutuja ja pientä liikettä, mutta silti liikkeen paino oli lyönnissä. Tässä sovelsin venyttämistä, ja ase venyy nopeasti lyönnin keskivaiheessa, ja luo parempaa liikkeentunnetta. Lisää tunnetta lyöntiin saa sillä, että vaihtaa hahmon pään kuvaa tavallisesta rauhallisesta ilmeestä vihaiseen kuvaan. Suosittelisin kuitenkin, että joko pelimoottorissa tai Spinessä luodaan visuaalinen efekti aseeseen, jotta liike selkeytyy. Tässä animaatiossa

teen lisäksi kaaren seuraamaan asetta, jotta liikkeeseen tulee lisää voimaa ja liikerata on selkeämpi katsojalle pelissä. (Kuva 18).



Kuva 18. Lyöntianimaatioesimerkki. Ase venyy yhden ruudun ajan, jotta liikerata on selkeämpi ja lyönti voimakkaampi. Lisäksi hahmon animaatioon on lisätty kaari vahvistamaan liikettä.

Roll-animaatio oli kaikkein haastavin. Tässä hahmon asento muuttuu eniten ja hahmon pyörähtäminen maassa tarvitsisi tosi paljon uusia kuvia ja yksityiskohdista animaatiota. Toisaalta itse pelissä roll-animaatio on määritelty niin nopeaksi, että liikkeen yksityiskohdat eivät edes näkyisi. Projektin aikataulun vuoksi animaatioille ei jää paljon aikaa hiomiseen. Kompromissina vain pyörähtän hahmoja ja litistän niitä joko muuttamalla hahmon skaalaa tai asettamalla luut niin, että hahmo on sikiöasennossa. Spinessä katsastaen pyörähdystä animaatio ei ole kovin miellyttävä, mutta pelissä siihen ei ehdi kiinnittää huomiota, ellei katsoa pysäytettyä videokuvaa tai ruutukaappausta.

Shoot-animaatio on hyvin yksinkertainen. Hahmo nostaa kättä rekyylin mukaisesti ja laskee sen takaisin alas hitaammin. Tässäkin hyödynnän hieman venyttämistä vahvemman liikkeen luomiseksi sekä kasvokuvan vaihtamista. Projektin tuon myös eri aseiden kuvia, jotta niitä voidaan vaihtaa pelimoottorissa riippuen siitä, minkä aseensa pelaaja poimii. Kaikki aseet käyttävät samaa animaatiota, mutta parempi olisi, jos jokaiselle tekisi omanlaisensa aseensa ominaisuuksista riippuen. Esimerkiksi naulapysyssä (Kuva 19, vasen hahmo) voisi olla vähemmän rekyyliä, kuin papuhaulikossa (Kuva 19, oikea hahmo).

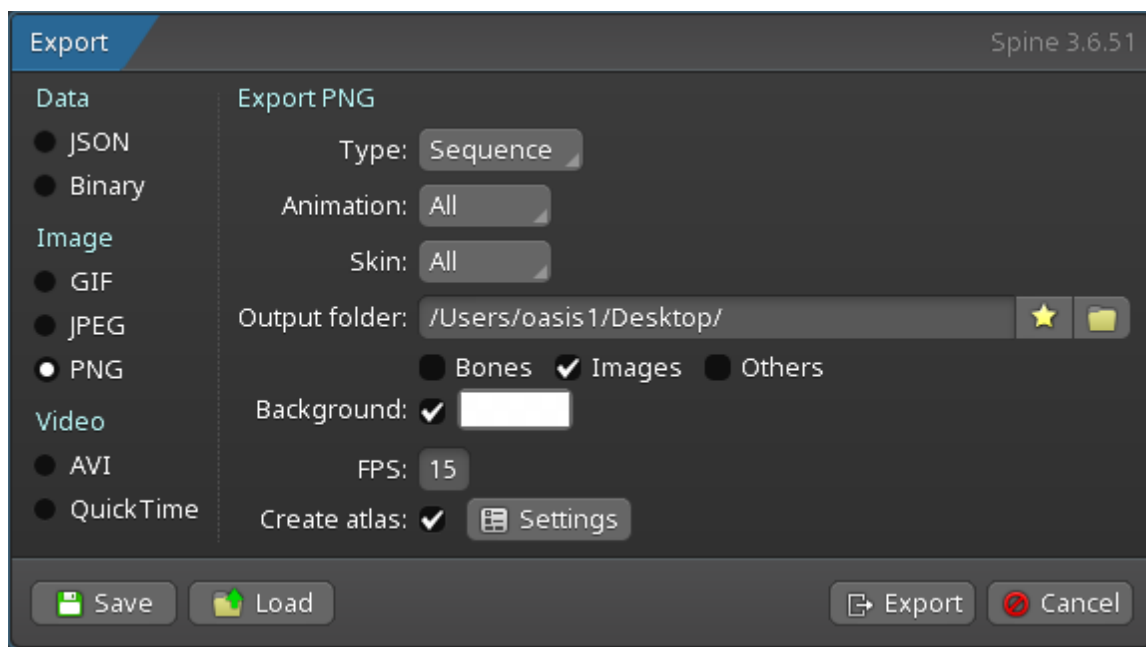


Kuva 19. Naulapyssy vasemmalla ja papuhaulikko oikealla. Kuvia voi vaihtaa joko animaation aikana tai pelimoottorissa.

12 Unity

Spine-tiedostojen käyttäminen Unityssa vaatii Spine komentosarjaokirjaston lataamisen. On myös mahdollista kääntää Spine-tiedostot Unityn omaan tiedostomuotoon, jolloin ei tarvitse ladata mitään lisätiedostoja. Tämä vaikuttaa kuinka paljon animaatioita voi manipuloida pelimoottorissa ja mitä ominaisuuksia on käytettävissä. Valmiin hahmon voi tuoda Unityyn käyttämällä export-ominaisuutta (kuva 20). Unityn puolella on laaja koodikirjasto Spine-tiedostojen muokkaamiseen ja manipuloimiseen.

Tätä opinnäytetöytä kirjoittaessa Unity ei tue atlas-tiedostoja suoraan, vaan export-asetuksissa pitää muokata tiedostonpääte muotoon atlas.txt, jolloin siitä tulee tekstitiedosto. Jos tämä jää tekemättä, pelihahmoja ei voi käyttää, sillä Unity ei pysty avaamaan ja käyttämään atlas -tiedostossa olevaa dataa.



Kuva 20. Export-ikkuna. Spinestä voi tuoda ulos animaatioita ja kuvia eri formaateissa.

13 Tulokset

Valmiit neljä hahmoa vaativat monta iteraatiota. Pelihahmojen animaatioihin vaikutti paljon pelimekaniikka sekä pelin nopeatempoisuus. Lyöntianimaatiota piti muuttaa useita kertoja, että siitä tuli kehitysryhmän mieleinen. Kuinka hahmot ovat ohjelmoitu toimimaan pelissä myös vaikutti animaatioihin, ja näitä piti yhdessä ohjelmoijan kanssa käydä läpi, että ne näyttävät siltä kuten halutaan.

Jokaiselle hahmolla piti myös miettiä yksilöllisesti, minkälainen liike toimii. Erilaiset lyöntiaseet vaativat erinäköisiä animaatioita. Ongelmallinen animaatio on esimerkiksi joulupukkihahmon lahjasäkki. Lahjasäkkiä piti muokata useasti, että sillä sai toteutettua iskevän animaation, jonka pelaajat huomaavat.

Animaatioita vielä hiottiin ja vahvistettiin luomalla partikkeliefektejä Unityn sisällä (kuva 21). Esimerkiksi pyörähdysanimaatioon tuli pölypilvi, joka korostaa liikettä ja mitä tapahtuu. Näitä piti kokeilla useita erilaisia ja muokata, jotta ne sopivat pelin visuaaliseen ilmeeseen eivätkä häiritse liikaa.



kuva 21. kuvankaappaus Drunkenpants-pelistä.

Pelianimaatiota toteuttaessa haaste oli huomioida animaatioiden vaikutus pelisuunnitteluun ja vastavuoroisesti. Animaatioita piti muokata ja suunnitella pelimekaniikkojen kautta, esimerkiksi pyörähdysanimaation ollessa erittäin nopea, yksityiskohtien hiominen vaikutti turhalta työltä. Myös se piti huomioida animaatiossa, että pelimekaanisesti toiminta tapahtuu välittömästi napinpainalluksesta: pyörähdystoiminto on pelissä usein reaktio hyökkäykseen, jotka tapahtuvat todella nopeasti. Liikettä ei pysty ennakoimaan yhtään, sillä pyörähdystoiminnan tapahtuessa viiveellä ja hahmon jäädessä ponnistamaan pelaajahahmo todennäköisesti vain kuolisi joka kerta yrittäessään väistää.

Pienessä kehitysryhmässä jouduin myös toimimaan graafikon ja animaattorin lisäksi myös pelisuunnittelijana. Tämä vaatii joustavuutta ja kykyä antaa rakentavaa palautetta pelin muista osa-alueista. Pelihahmojen liikkumisnopeus on esimerkiksi asia, joka vaikuttaa pelimekaanisesti todella moneen asiaan: kuinka helppoa pelaajien on pysyä hengissä tai osua muihin. Liikkumisnopeus vaikuttaa myös animaatioihin ja sitä liikaa muokatessa pitää myös animaatiot tehdä uusiksi, että nopeus ja hahmon toiminta sopivat yhteen.

Pelihahmojen tekemisen yhteydessä pohdimme myös hahmoille nimiä ja lyhyitä kuvauksia keitä he ovat. Tämä olisi ollut hyödyllistä miettiä jo ideointi ja luonnos-

vaiheessa, koska siitä olisi voinut löytää visuaalisia elementtejä hahmoille ja luonteenpiirteitä animaatioihin. Nyt hahmojen animaatiot ovat pelkistettyjä, mutta kuvaavat toimintaa, ja vähemmän hahmojen luonnetta.

14 Pohdinta

Ennen tätä opinnäytetyötä minulla oli hyvin vähän kokemusta animoinnista. Työharjoittelun kautta sain kokemusta Spinen käytöstä ja taitopohjaa tähän opinnäytetyöhön. Opinnäytetyön raportointi ja lähteisiin syventyminen paransi ymmärrystäni animaatiosta, hahmon suunnittelusta ja ylipäätään videopelien tekemisestä. Mielenkiintoista oli huomata, kuinka vanhat perinteisen animaation periaatteet toteutuvat tänäkin päivänä videopelianimaatiossa.

Spinen ja vastaavien animaatio-ohjelmiin liittyvää kirjallisuutta on julkaistu todella vähän. Suurin osa oppimastani perustuu internetistä löytyvään dokumentaatioon, esimerkkitiedostoihin ja videoihin. Spine-ohjelmistona on erittäin hyödyllinen 2D-animoinnissa. Ohjelmiston käyttöliittymässä ja käytettävyydessä on parantamisen varaa, sillä hyvin harvalle toiminnalle esimerkiksi on omia pikanäppäimiä. Myös käyttöliittymän liiallinen suoraviivaisuus aiheuttaa turhaa päänvaivaa, sillä animaatioidenkin luominen on kahden painalluksen takana, jotka ovat mahdollisimman kaukana toisistaan.

Spinen parhain ominaisuus oman työskentelyn kannalta oli ”meshien” luominen kuvista, sillä se mahdollistaa hyvin monipuolisen ja yksityiskohtaisen animaation. Valitettavasti ominaisuus on lukittu kalliimman lisenssin taakse. Halvempi lisenssi sisältää perinteisille animaatio-ohjelmille tuttuja työkaluja, mutta kilpailijoihin verrattuna se on rajoittuneempi. Esimerkiksi Spriter Pro on halvempi kuin Spine Pro -versio, mutta siinä on samojen vastaavien ominaisuuksien lisäksi myös muitakin. Spinen huonoihin puoliin luettelen sen käyttöliittymän: pikanäppäimiä on vaikea löytää, ja jotkin asiat voi tehdä vain painamalla näppäintä hierarkian alla, esimerkiksi uuden animaation luominen.

Työprosessi oli erittäin nopeatempoista. Hahmoja ollessa useita ja aikaa tehdä vain pari kuukautta piti opetella samalla piirtämään uudella tavalla sekä käyttämään Spine-ohjelmistoa. Toisaalta pelin ollessa vielä kehityksessä, hahmojen grafiikka ja animaatiot ovat riittävällä tasolla. Tämä on vain yksi esimerkki digitaalisen piirtämisen ja animaation vahvuudesta: iterointi ja tuottaminen on nopeaa, ja eri ideoita voi testata nopeasti ja muutoksien tekeminen on vaivatonta.

Suosittelen animaattoria keskustelemaan muun työryhmän kanssa siitä, kuinka eri toiminnot on suunniteltu, kuinka pitkäkestoisia ne ovat ja millä ajoituksella ne tapahtuvat. Pelihahmon toiminta vaikuttaa suuresti animaatioon. Spine-ohjelmisto ja ylipäätään tietokoneanimointi helpottaa animaatioiden muokkaamista jälkikäteen suuresti, etenkin jos pelin mekaniikat ja toiminnallisuudet muuttuvat. Animaatioita voi käsitellä myös Unity-pelimoottorissa tiettyyn pisteeseen asti, joten ohjelmoija voi sijoittaa pelin toimintoja animaatioihin.

Olen oppinut paljon perinteisestä animaatiosta sekä nykyisistä työkaluista ja kuinka yhdistellä niitä. Perinteisen animaation kaksitoista periaatetta on suunniteltu elokuvakerrontaa varten, mutta olen tyytyväinen kykyyni soveltaa periaatteita erilaiseen projektiin kuin mihin ne on tarkoitettu. Tulevaisuudessa toivon syventäväni taitojani videopelianaimaation suhteen, hyödyntää pelihahmojen toteutuksessa Spinen kaikkia ominaisuuksia, joita tässä projektissa ei tarvinnut. Spine on monipuolinen työkalu, ja luurankoanimaatiot ovat helppo tapa lähestyä 2D animaatiota.

Tietokoneohjelmien avustama animointi on nopeampaa kuin perinteinen, mutta hyvä animaatio vaatii paljon aikaa ja opiskelua. Jotta päästäisiin parhaaseen lopputulokseen, jokaisen hahmon eri animaatioille on jätettävä aikaa eri versioiden kokeilun lisäksi myös yksityiskohtien hiomiselle aikaa. Tässäkin projektissa olisin toivonut ehtiväni hyödyntämään monipuolisemmin Spinen lukuisia työkaluja, mutta kiireellisen aikataulun vuoksi jouduin pysyttelemään suht yksinkertaisissa ratkaisuuksissa. Jos osallistun vielä animointiin tulevaisuudessa, pyrin aikatauluttamaan työn paremmin ja hiomaan animaatioita.

Lähteet

- Autodesk Maya 2015. Introduction to polygons. <https://knowledge.autodesk.com/support/maya-lt/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/MayaLT/files/Polygons-overview-Introduction-to-polygons-htm.html>. 14.5.2018.
- Bloop Animation 2018. Types of Animation. <https://www.bloopanimation.com/types-of-animation/>. 20.2.2018.
- Doody, E. 19.8.2014. Wulverblade. <http://wulverblade.com/frame-vs-mesh-animations>. 3.2.2018.
- Esoteric Software. 2018. Spine User Guide. <http://esotericsoftware.com/spine-user-guide>. 20.2.2018.
- Falck, T. 2017. opinnäytetyö. 2D-deformaatio-animaatio peligrafiikassa. Metropolia ammattikorkeakoulu. <http://urn.fi/URN:NBN:fi:amk-2017060212183>. 2.2.2018.
- Introducing JSON 2018. <http://www.json.org/>. 27.5.2018.
- Kukkonen J. 2011. opinnäytetyö. Digitaalinen pala-animaatio. Turun ammattikorkeakoulu. <http://urn.fi/URN:NBN:fi:amk-2011060611069> 2.2.2018.
- Marionette Studio. 2016. Skeletal based animation. <https://marionettestudio.com/skeletal-animation/>. 3.2.2018.
- Ohlew, J. 10.10.2014 For the love of 2D: Why developers still make them. <https://www.usgamer.net/articles/for-the-love-of-2d>. 27.5.2018.
- Pardew, L. 2009. Beginning Illustration and Storyboarding for Games. Course Technology, Incorporated.
- Rogers, S. 2014. Level Up! the Guide to Great Video Game Design. US. John Wiley and Sons, Ltd.
- Thomas, F. Johnston, O. 1984. Disney Animation: The Illusion of Life. New York: Abbeville Press.
- Types of Animation. 2015. <https://www.bloopanimation.com/types-of-animation/>. 27.5.2018.
- White, T. 2006 Animation: from Pencil to Pixels. Oxford: Taylor & Francis Ltd.
- Wikipedia 2018a. Runtime library. https://en.wikipedia.org/wiki/Runtime_library. 2.5.2018.

Wikipedia 2018b. Scripting languages. https://en.wikipedia.org/wiki/Scripting_language. 2.5.2018.

Wikipedia 2018c. Polygon mesh. https://en.wikipedia.org/wiki/Polygon_mesh. 14.5.2018.