



Veera Varrio

2D-pelihahmon animaatiotekniikat

Metropolia Ammattikorkeakoulu

Muotoilija (AMK)

Muotoilun tutkinto-ohjelma

Opinnäytetyö

17.4.2024

Tiivistelmä

Tekijä(t): Veera Varrio
Otsikko: 2D-pelihahmon animaatiotekniikat
Sivumäärä: 45 sivua
Aika: 17.4.2024

Tutkinto: Muotoilija (AMK)
Tutkinto-ohjelma: Muotoilun tutkinto-ohjelma
Pääaine: Visuaalisen viestinnän muotoilu
Ohjaaja(t): Lehtori Samuli Homanen,
Lehtori Lauri Huikuri

Opinnäytetyön tarkoituksena on perehtyä niihin eri tekniikoihin, joilla 2D-pelihahmoa voidaan animoida. Tavoitteena on havainnoimalla, vertailemalla ja kokeilemalla selvittää, mitä eroja, hyötyjä ja heikkouksia tekniikoilla on ja millaisiin tarkoituksiin eri tekniikat sopivat.

Animaatiotekniikoita eritellään opinnäytetyössä käytännön peliesimerkkien avulla. Lisäksi rigataan hahmo Unityssa. Opinnäytetyössä esitellään myös hahmoanimaation tekemisessä käytettäviä yleisimpiä ohjelmia sekä pelimoottoreihin sisäänrakennettuja ominaisuuksia tähän liittyen. Opinnäytetyössä perehdytään animoitavan pelihahmon suunnitteluun muun muassa valitun alustan, peligenren, peliperspektiivin ja visuaalisen tyylin näkökulmista. Työssä läpikäydään myös pelihahmojen historiaa, sillä teknologian kehittyminen on vaikuttanut paljon pelihahmojen suunnitteluun ja animaatioon.

Opinnäytetyössä tehtyjen havaintojen perusteella saatiin eväitä hahmosuunnittelun tueksi sekä löydettiin animaatiotekniikoita vertailemalla niiden etuja ja rajallisuuksia. Kriteereinä vertailulle käytettiin muun muassa ajankäyttöä, budjettia ja visuaalista tyyliä. Animaatiotekniikoiden yhdistäminen johti usein haluttuihin lopputuloksiin.

Asiasanat: 2D, hahmoanimaatio, pelihahmo, pelimoottori

Opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

Abstract

Author(s): Veera Varrio
Title: Animation Techniques for a 2D Game Character
Number of Pages: 45 pages
Date: 17 April 2024

Degree: Bachelor of Culture and Arts
Degree Programme: Design
Major: Visual Communication Design
Instructor(s): Senior lecturer Samuli Homanen,
Senior lecturer Lauri Huikuri

The purpose of this thesis was to examine various techniques for animating 2D game characters. The aim of this thesis was to identify the advantages, limitations and practical applications of each technique through comprehensive observation, comparison and testing.

This thesis exemplifies character animation techniques through their implementation in various games. Rigging a character in a game engine is also provided as a practical example. This thesis addresses animation systems in game engines and example softwares used for character animation. It is also discussed how the chosen platform, visual style, genre and perspective of the game affect the character design process. The historical context of game characters is also covered in order to fully understand the influence of technological advancements on game character design and animation.

The findings of this thesis offer insights for character design and aid in the selection of the most appropriate animation technique. The advantages and disadvantages of each animation technique were identified through comparisons across various criteria, such as time efficiency, budget and visual style. Combining various techniques often resulted in achieving the desired outcome.

Keywords: 2D, character animation, game character, game engine

This thesis has been checked using Turnitin Originality Check service.

Sisällys

1	Johdanto	1
2	Katsaus 2D-peleihin ja pelihahmoihin	2
2.1	Hahmoanimaation merkitys pelissä	3
2.2	Huomioitavaa animoitavan pelihahmon suunnittelussa	4
2.3	Pelihahmojen historiaa	6
3	2D-hahmon animaatiotekniikat ja työkalut	13
3.1	Frame by frame -animointi	13
3.1.1	Rotoskooppaus	16
3.1.2	Pikselitaide	19
3.1.3	Stop motion	21
3.2	Pala-animaatio	22
3.3	Luurankoanimaatio	24
3.3.1	Käänteinen kinematiikka eli IK	26
3.3.2	Meshien muokkaaminen	28
4	Hahmo pelimoottorissa	29
4.1	2D-hahmon rigaaminen Unityssa	29
4.2	Animaation toisto pelimoottorissa	32
5	Tekniikoiden vertailua	35
6	Yhteenveto	37
	Lähteet	39
	Kuvalähteet	44

1 Johdanto

Tässä opinnäytetyössä tavoitteenani on esitellä lukijalle erilaisia 2D-pelihahmon animoimiseen käytettäviä tekniikoita perehtymällä niiden käyttötarkoituksiin, vahvuuksiin ja rajoitteisiin sekä lopuksi vertailla niiden eroja. Vaikka 3D onkin etenkin suuremmissa pelituotannoissa se yleisin valinta, on 2D-peleillä tänäkin päivänä paljon mahdollisuuksia erottua edukseen. Tarkastelenkin tässä opinnäytetyössä niitä seikkoja, jotka pelihahmon animaatioon vaikuttavat, ja sitä, millä tekniikoilla hahmoanimaatiota voidaan 2D-peleissä toteuttaa. Olen ajatellut opinnäytetyöni olevan suunnattu kaikille pelialasta ja animaatiosta kiinnostuneille. Erityisesti siitä voi olla hyötyä 2D-hahmoja ja niiden animaatioita suunnitteleville peligraafikoille ja muille hahmoanimaation parissa työskenteleville.

Pohjustan työtäni ensin kertomalla hahmoanimaation merkityksestä videopeleissä. Pohdin animoitavan pelihahmon suunnittelussa huomioitavia seikkoja muun muassa alustan, kontrollien, peligenren, peliperspektiivin ja visuaalisen tyylin näkökulmista. Avaan myös pelihahmojen animaation historiaa, sillä teknologian kehitys on luonnollisesti vaikuttanut paljon hahmojen ja niiden animaation kehitykseen.

Toisessa osiossa erittelen tarkemmin niitä tekniikoita ja työkaluja, joita nimenomaan 2D-pelihahmon animoimisessa voidaan hyödyntää. Olen jakanut tekniikat karkeasti kolmeen kategoriaan: frame by frame -animaatio, pala-animaatio ja luurankoanimaatio. Tämän jaottelun olen tehnyt sillä perusteella, että tällä tavoin selkeimmät erot tekniikoiden välillä tulevat ilmi. Lisäksi käsittelen tekniikoiden yhteydessä niihin liittyviä erilaisia menetelmiä ja työkaluja. Kerron myös eri tekniikoiden käyttötarkoituksista. Rajaus tekniikoiden välillä ei ole mitenkään yksiselitteistä, sillä niitä usein käytetään yhdessä. Olen kuitenkin työssäni pyrkinyt johdonmukaisesti erittelemään eri tekniikoiden ominaisuuksia, jotta lukijalle muodostuisi selkeä kuva niiden sisällöstä. Tekniikoiden käytännöllisempää puolta avatakseni hyödynnän valikoituja peliesimerkkejä, joiden

hahmoanimaatioita havainnoin. Laajemman käsittelyn ulkopuolelle olen rajannut ne ohjelmat, joilla hahmoanimaatioita usein toteutetaan. Tämän olen tehnyt sen vuoksi, että useimmilla ohjelmilla on samanlaisia ominaisuuksia, ja haluan työnsäni keskittyä nimenomaan eri tekniikoiden eroihin ja ominaisuuksiin. Mainitsen ohjelmista vain lyhyesti.

Tekniikoiden erittelyn jälkeen kerron vielä erikseen animaatioiden toistosta pelimoottoreissa. Käyn läpi yleisimpiä sisäänrakennettuja ominaisuuksia pelimoottoreissa liittyen hahmoanimaatioon. Hyödynnän esimerkkeinä Unity-pelimoottoria sekä omia hahmojani. Lisäksi esittelen hahmon rigaamista pelimoottorissa. Lopuksi vertailen 2D-pelihahmon animoimiseen käytettäviä tekniikoita niiden vahvuuksien ja rajoitteiden kautta.

2 Katsaus 2D-peleihin ja pelihahmoihin

Nykypäivän teknologia ja yhä kehittyneemmän 3D-grafiikan käyttö mahdollistavat miltei todellisuutta vastaavan realismin saavuttamisen videopeleissä. Lisäksi pelaajien vaatimukset pelejä kohtaan nousevat kohisten. 2D-pelien suosio ei kuitenkaan ole lopahtanut. 2D-pelien uniikki, käsintehty tunnelma välittyy pelaajalle saman tien. Monet asiat vaikuttavat siihen, päädytäänkö peli tekemään 2D- vai 3D-toteutuksena vai kenties niiden välimuotona. Pelin genre vaikuttaa toteutustavan valintaan. Esimerkiksi tasohyppely-, pulma- ja taistelupelit suosivat 2D-alustoja, kun taas rooli-, toiminta- ja seikkailupelit ja etenkin ammunta- ja taistelupelit suosivat 3D-alustoja. Usein 2D-pelien kehitys on nopeampaa kuin 3D-pelien ja budjettikin on tavallisesti pienempi. Varsinkin *indie*-pelit ovat usein 2D-toteutuksia. (Kevuru Games 2023.) Indie-videopelillä tarkoitetaan yksityishenkilön tai pienen tiimin luomaa peliä, joka kehitetään ilman julkaisijan taloudellista tukea (Helpshift 2023). 3D-toteutuksella voidaan kuitenkin saavuttaa realistisia avoimia pelimaailmoja (*open world*), mikä mahdollistaa erityisen immerstiivisen pelaajakokemuksen. Etenkin tällaisten AAA-pelien budjetti on valtava, ja peliä työstää sadoista ihmisistä koostuva tiimi. 3D-pelit mahdollistavat hahmon liikuttamisen mihin tahansa suuntaan ja maailman katsomisen mistä tahansa

näkökulmasta – nämä seikat ovat luonnollisesti 2D-peleissä hyvin rajoitettuja. (Kevuru Games 2023.)

2.1 Hahmoanimaation merkitys pelissä

Laadukkaan hahmoanimaation tarkoituksena on parantaa pelaajakokemusta. Vaikka periaatteessa hyvän pelin on oltava toimiva ja kiinnostava kokonaisuus jo itsessään ilman hienoja grafiikoitakin, tuovat visuaaliset elementit sille paljon lisäarvoa. Sen lisäksi, että tarkoituksenmukainen hahmoanimaatio tekee pelistä visuaalisesti miellyttävän ja kiinnostavan ja tukee pelin tarinankerrontaa, välittää se ennen kaikkea pelaajalle informaatiota: hahmon reaktiot, liikkeet ja kehonkieli viestivät paljon muun muassa siitä, mikä tunnetila hahmolla on ja millainen sen luonne on. Pelihahmojen käyttäytyminen vaikuttaa pelaajan käyttäytymiseen. Esimerkiksi NPC:n (*non-playable character*) eli ei-pelattavan hahmon ilmaisemat tunnetilat ja teot voivat ohjata pelaajaa tekemään erilaisia valintoja pelissä – haluatko miellyttää NPC:tä vai suututtaa sen? Millaisia reaktioita ja asenteita teoistasi seuraa?

Usein ilman animaatiota tiedon välittäminen tulee tehdä näkyväksi tekstillä, mikä ei ole pelaajakokemuksen kannalta aina optimaalisin vaihtoehto, sillä pelin luoma immersio voi hajota. (Porokh 2023; Moleman 2009.) Videopelien immersivisyydellä tarkoitetaan sitä, että pelaaja kykenee uppoutumaan pelissä olevaan maailmaan ja eläytymään siellä tapahtuviin asioihin sulkien ulkopuolisen maailman hetkellisesti pois mielestään (Wirtz 2023). Vaikka joku voisi olettaa, että immersiota tapahtuu helpommin realistisemmissä 3D-peleissä tai vielä tehokkaammin esimerkiksi virtuaalitodellisuudessa, voi 2D-pelinkin visuaalinen maailma ja mukaansatempaava tarinankerronta imaista pelaajan mukaansa pelimaailman syövereihin. Luonnollisesti myös dynaaminen hahmoanimaatio toimii yhtenä elementtinä immerstiivisen pelaajakokemuksen rakentumisessa.

2.2 Huomioitavaa animoitavan pelihahmon suunnittelussa

Animoitavan pelihahmon suunnittelussa on huomioitava monia asioita. Käsitte- len tässä aluvussa animoitavan pelihahmon suunnittelussa huomioitavia seik- koja muun muassa pelialustan, kontrollien, peligenren, visuaalisen tyylin ja peli- perspektiivin näkökulmista. Tavoitteena on, että hahmoanimaatio tukee pelin toiminnallisuutta ja vuorovaikutuksellisuutta ja on yhtenäinen pelin tarinallisuu- den ja muun grafiikan kanssa.

Pelialustalla tarkoitan tässä yhteydessä sitä laitetta, jolla peliä pelataan. Pääasi- assa jaan ne tietokoneeseen, mobiiliin ja konsoleihin. Vaikka nykyään sama peli julkaistaankin yleensä useammalle alustalle, on se usein alun perin suunniteltu jollakin tietyllä laitteella pelattavaksi. Mobiililaitteella toteutettava peli on tavalli- sesti yksinkertaisempi kuin konsoli- tai tietokonepeli sen pienen näyttökoon puo- lesta. Käsikonsoleissa, kuten Nintendo Switchissä, on toki myös pienehkön näy- tön rajoitus, mutta laite on kuitenkin optimoitu pelaamiseen, eli pelit voivat te- hoiltaan ja grafiikoiltaan olla vaativampia kuin mobiilissa. Nimenomaan suoritus- kyvyn puolesta mobiili on usein pelialustoista heikoin. Koska mobiililaitteen ruutu on pieni, tulee hahmon olla hyvin tunnistettavissa jo pienessä koossa ja erottua selkeästi muusta peligrafiikasta. (CGhero 2023.) Hahmon tunnistettavan ja persoonallisen siluetin luominen on tärkeä osa hahmon suunnitteluprosessia (Melling 2019). Koska mobiilipelihahmo on usein yksinkertaisempi, ei sen ani- mointi välttämättä vaadi monimutkaisempaa rigin rakentelua. Tämän vuoksi mo- biilipelien hahmoanimaatiossa hyödynnetään usein frame by frame -tekniikkaa. Esimerkiksi Innerslothin vuonna 2018 julkaisemassa, suureen suosioon nous- seessa Among Us -moninpelissä hahmot ovat erittäin yksinkertaisia eikä niillä ole monimutkaisia animaatioita. Tarinankerronnallisesti ja minimalistisen visuaa- lisen tyylin puolesta ne kuitenkin palvelevat onnistuneesti pelin tarkoitusta.

Jotta hahmoanimaatio palvelee pelin toiminnallisuutta, on se tavallisesti interak- tiivista. Pelihahmo on vuorovaikutuksessa peliympäristön kanssa, ja pelaajalla on hahmosta kontrollin tunne. Mobiilipeleissä hahmoanimaation toteutus on haasteellisempaa rajallisten kontrollien vuoksi. Mobiilipelin kontrollit perustuvat yleensä kosketusnäytön toimintaan eli sormen pyyhkäisyyn tai näpäytykseen.

Sen sijaan konsoli- ja tietokonepeleissä hahmon kontrolloinnin mahdollisuudet ovat laajemmat. (CGhero 2023.) Mobiilipeleihin voidaan lisätä animaatiota kuitenkin myös muilla keinoin. Esimerkiksi turkkilaisen peliyhtiön MobGen Oddmar-viikinkipeli hyödyntää motion comic -tekniikkaa, eli pelissä on kohtauksia, jotka toimivat kuin liikkuvana sarjakuvana (IMGA i.a.).

Kontrollien lisäksi hahmon suunnitteluvaiheessa tulee ottaa huomioon, mitä liikkeitä hahmolta odotetaan. Yleensä pelihahmoilla on jokin etenemisanimaatio, esimerkiksi kävely- tai juoksuanimaatio. Lisäksi pelistä riippuen hahmolla voi olla hyppäys-, hyökkäys- ja keräilyanimaatiot sekä muita tarvittavia toimintoja kuvaavia animaatioita. Yksinkertaisemmat hahmot eivät luonnollisesti vaadi niin paljon animoimista. Esimerkiksi Ludeon Studiosin vuonna 2018 julkaiseman Rimworld-scifi-simulaatiopelin maailma ja hahmot ovat hyvin pelkistettyjä. Hahmoilla ei oikeastaan ole raajoja, jotka voisivat kävellessä liikkua – hahmot vain lipuvat eteenpäin.

Sekä pelin visuaalinen tyyli että peligenre osiltaan vaikuttavat siihen, millaisia hahmoja suunnitellaan ja miten ne liikkuvat. Hahmoanimaatioiden tulee palvella pelin muuta grafiikkaa. Usein realistisempien pelien hahmoilta odotetaan myös realistisempaa liikettä. Sen sijaan yksinkertaisemmat ja tyylitellymmät pelit eivät aseta niin korkeita odotuksia hahmoanimaation suhteen. Pikseligrafiikkaa hyödyntävissä peleissä hahmon ei odoteta liikkuvan kovinkaan realistisesti. Piirroselokuvamainen ympäristö taas voi antaa viitteitä siitä, että hahmoanimaatioissa nähdään paljon liioittelua ja ilmeikkyyttä. Peligenreistä pulmapelit ja kasuaalipelit asettavat hahmoanimaatiolle ehkä vähemmän odotuksia kuin rooli- ja toimintapelit. Taistelupeleissä hahmojen luonnollisesti odotetaan taistelevan. Visual novel -peleissä taas ei välttämättä nähdä hahmoanimaatiota lainkaan.

Animoitavan pelihahmon suunnitteluun vaikuttaa lisäksi myös peliperspektiivi, eli mistä näkökulmasta pelaaja katsoo peliä. 2D-peleissä hahmoa ja pelimaailmaa saatetaan katsoa ylhäältä päin (*top-down*) tai hahmo on pelaajaan nähden sivuttain (sivuttaisivierittävät pelit). Hahmon animointiin vaikuttavat myös sen kulkusuunnat – kulkeeko hahmo sivuttain ainoastaan eteen- ja taaksepäin vai

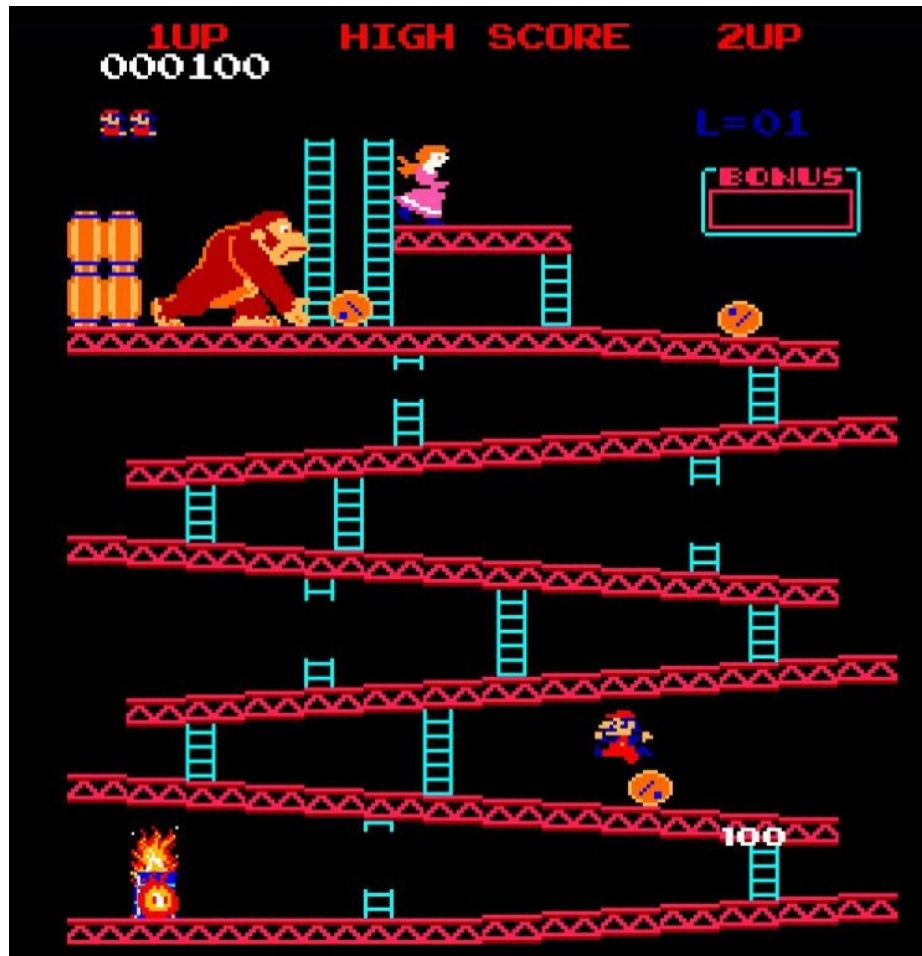
ylhäältä katsottuna kenties jopa kahdeksaan eri ilmansuuntaan? Koska hahmo on kaksiulotteinen, tulee sen jokainen uusi kuvakulma piirtää ja animoida erikseen.

2.3 Pelihahmojen historiaa

Varsinaisten videopelien historia lienee saanut alkunsa vuonna 1952, kun tietojenkäsittelytieteen professori A.S. Douglas kehitti OXO:n eli ristinollan osana ihmisen ja tietokoneen välistä vuorovaikutusta käsittelevää väitöskirjaansa. Tästä voidaan kuitenkin olla montaa mieltä, sillä peli ei koskaan ollut avoin yleisölle eikä se sisältänyt minkäänlaista reaaliaikaisesti päivittyvää grafiikkaa. (Cohen 2019.) Pysin tässä osiossa kuitenkin keskittymään lähinnä pelihahmojen historiaan, joten en jatkossa erityisemmin käsittele laitteiden ja konsolien evoluutiota, vaikka se taustalla kulkeekin mukana.

Ensimmäisten konsoli- ja tietokonepelien hahmoanimaatioissa hyödynnettiin lähinnä frame by frame -tekniikkaa ja sprite-grafiikoita. Niitä luotiin käsin piirtämällä, digitoimalla, rotoskooppaamalla ja jopa videokuvaamalla. Teknologian kehittyessä hahmot ovat saaneet monipuolisempien liikkeiden lisäksi myös lisää luonnetta ja persoonaa. Hahmot ovat tietenkin kehittyneet myös visuaalisesti, kun tietokoneet ja konsolit ovat mahdollistaneet yhä korkearesoluutioisempien grafiikoiden käyttöä.

Ensimmäisenä varsinaisena pelattavana hahmona voidaan kaikeksi pitää Pac-Mania vuodelta 1980. Heti sen jälkeen, vuonna 1981, kaikille tuttu Super Mario teki ensimmäisen ilmestymisensä Jumpmanina Miyamoton Nintendolle luomassa Donkey Kongissa. (Furniss 2016.) Näiden kahden pelin päähahmojen olemuksissa on selviä eroja: Pac-Man on hyvin yksinkertainen sokkelossa seikkaileva keltainen pallo, jolla ei suun avaamisen ja sulkemisen lisäksi ole muita animaatioita tai asentoja. Sen sijaan Jumpman on jo selkeästi ihmishahmo, joka juoksee, kiipeilee, lyö vasaralla ja hyppii esteiden yli. Donkey Kong -gorillalla on myös lajityypillisiä eleitä ja asentoja, sekä erilaisia animaatioita, kuten kiipeily, hyppy ja tynnyreiden tiputtaminen (ks. kuva 1).



Kuva 1. Vuoden 1980 Donkey Kongissa hahmoilla on useampia sprite-animaatioita. Donkey Kong tiputtaa tynnyreitä, ja Jumpman hyppää tynnyreiden yli. (Classicgaming i.a.)

Näissä aiemmin mainitsemissani videopeleissä oli siis jo selkeää animaatiota. Tällainen yksittäisen kuvan liikuttaminen ruudulla oli mahdollista sprite-grafiikoiden eli *spritejen* ansiosta. Spritet ovat yksinkertaisuudessaan kaksiulotteisia kuvia, joita voidaan liikuttaa ruudulla itsenäisesti ilman, että se vaikuttaa taustaan tai muuhun peligrafiikkaan. Aiemmin liike perustui koko näytön päivittymiseen aina uudelleen ja uudelleen, kun liikettä tapahtui. (Ahoy 2014.) Sprite-grafiikan teko oli tuohon aikaan kuitenkin melko haastava prosessi – esimerkiksi vuoden 1984 Super Mario Bros. -pelin spritet on piirretty ja väritetty käsin ja vasta sitten digitoitu, jonka jälkeen ne piti vielä saada toimimaan pelissä. Spritejen käyttöön liittyi myös paljon rajoituksia laitteistosta riippuen esimerkiksi sen suhteen, kuinka monta väriä tai spriteä ruudulla voidaan näyttää yhtä aikaa. Spritet

perustuvat bittikarttagrafiikkaan ja koska tietokoneet eivät olleet kovin tehokkaita, ovat vanhat pelihahmot hyvin matalaresoluutioisia, eli niissä on vain vähän pikseleitä. (Blacksteinn 2022.) Tätä nostalgista pikseliyyliä havitellaan kuitenkin vielä tänäkin päivänä useissa videopeleissä, vaikka teknologia on jo huomattavasti kehittynyt. Puhun pikseliyyliä enemmän luvun Frame by frame -animointi yhteydessä.

Lisäksi peleissä nähtiin ja nähdään yhä myös ei-interaktiivista animaatiota.

Cutscenet eli välianimaatiot ilmestyvät pelissä yleensä silloin, kun pelaaja saavuttaa jonkin tietyn pisteen tai suorittaa tehtävän. Myös Donkey Kongissa ja Pac-Manissa oli lyhyitä cutscenejä, jotka toimivat lähinnä hauskana lisänä. Kuitenkin vuoden 1983 videopelin Bega's Battlen (jap. *Genma Taisen*) välianimaatiot toimivat jo selvästi pelin tarinankerrontaa tukevinä elementteinä. Välianimaatiot olivat *full-motion videota* eli etukäteen tallennettua videomateriaalia samannimisestä elokuvasta. Animen taidetta hyödynnettiin myös pelin taustoissa (ks. kuva 2). (Sini 2023.)



Kuva 2. Bega's Battle -pelin taustoissa *full-motion videona* hyödynnetty anime- taide erosi huomattavasti pelihahmojen matalaresoluutioisista grafiikoista. (Ruutukaappaus: Insert Coin 2020.)

Spritejä luotiin myös muilla tavoin. Jordan Mechner kehitti taistelupelin Karatekan vuonna 1984 Apple II -tietokoneelle. Hän halusi tietokoneen ja omien piirustustaitojensa rajoituksista huolimatta luoda mahdollisimman sujuvalta näyttävää hahmoanimaatiota, joten hän hyödynsi rotoskooppausta. Hän siis jäljensi videokuvattua materiaalia läpi piirtämällä, jotta sai muun muassa haastaviin taistelu-kohtauksiin realistista liikettä. Mechner hyödynsi samaa tekniikkaa myös pelissä Prince of Persia vuonna 1989. (Leong 2023.) Kerron rotoskooppauksesta animaatiotekniikkana vielä tarkemmin osiossa Frame by frame -animointi.

Pelisarja The Legend of Zelda on myös Miyamoton kädenjälkeä ja sai alkunsa vuonna 1986 mullistaen samalla seikkailupelien genren. Pelisarjan uusin peli julkaistiin vuonna 2023, joten tunnistettava päähenkilö Link on ehtinyt teknologian kehittyessä muuttua paljon noiden 37 vuoden aikana (ks. kuva 3). (Webster 2016.)



Kuva 3. Link-hahmo vasemmalla ensimmäisessä Legend of Zelda -pelissä vuonna 1986 ja oikealla uusimmassa pelissä Legend of Zelda: Tears of the Kingdom, 2023. (Eisen ym. 2018; Parkin 2023.)

Ensimmäisessä Legend of Zeldassa pelin perspektiivi on tosiaan pelaajaan nähden ylhäältä alas (*top-down*) ja Link on animoitu liikkumaan neljään eri

suuntaan: oikealle, vasemmalle, eteen ja taakse. Lisäksi Linkillä on hyökkäysanimaatio. *Top-down*-peleissä hahmon liikkuminen tapahtuu perinteisesti juuri neljään suuntaan, usein myös jopa kahdeksaan. Oikealle tai vasemmalle kulkevan hahmon animaation voi kätevästi peilata, mutta eteen- ja taaksepäin menevä hahmo täytyy piirtää erikseen, koska kuvakulmat ovat niin erilaiset.

Siirtyminen kolmannen sukupolven konsoleista neljännen sukupolven konsoleihin alkoi vuonna 1987 ja oli iso harppaus videopelien kehityksessä. 8-bittisten laitteistojen väistyessä 16-bittisten tieltä videopelien värimaailma ja muu grafiikka monipuolistuivat. Vuonna 1992 Disneyn piirroselokuva *Aladdin* sai niin suuren suosion, että heti seuraavana vuonna Sega julkaisi siitä myös tasohypelypelin. Rima hahmoanimaatioille oli korkealla, ja yli 30 animaattoria työskenteli projektin parissa. Animaattorit hyödynsivät spritejen peilaamista animaatioloopeissa tietokoneen muistin optimoimiseksi, ja esimerkiksi Aladdinin kiipeilyanimaatiossa puolet kuvista on peilattu. (Strafofox 2021a.) Kuva 4 havainnollistaa Aladdinin peilattujen spritejen käyttöä.



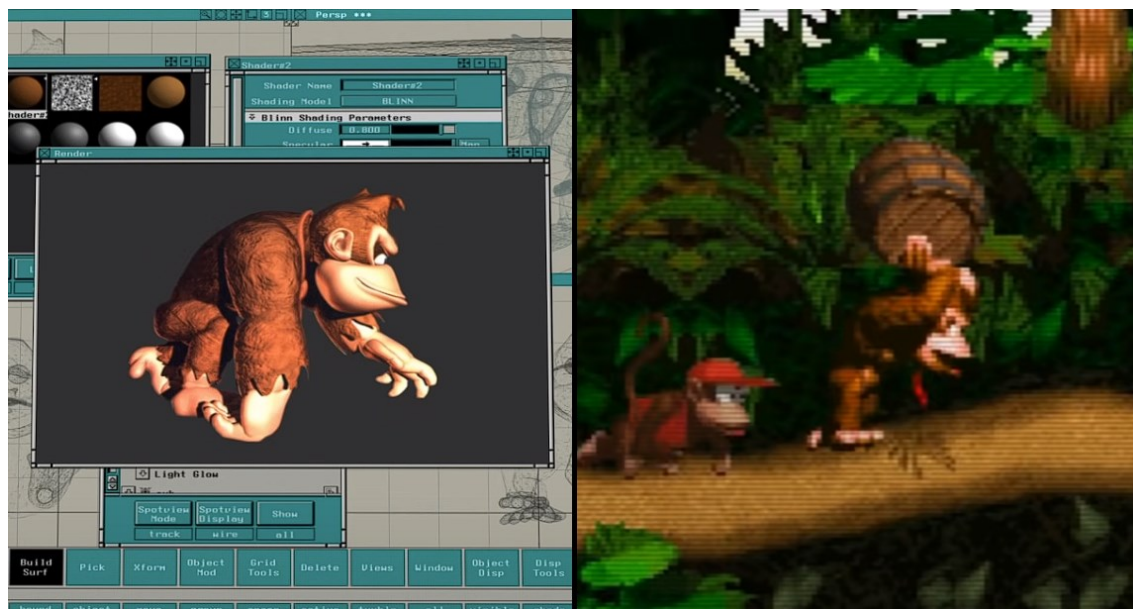
Kuva 4. Kuvassa yllä on Aladdinin kiipeily-frames ja alla samat frames peilattuna. Näiden kuvien peräkkäinen toisto luo saumattoman animaatioloopin. Ruutu-kaappaukset pelistä *Disney's Aladdin* (1992).

Animaatiolooppi on syklisen liikkeen kuvasarja, jota voidaan toistaa loputtomasti ilman liikkeen katkeamista. Kerron animaatioloopeista perusteellisemmin luvussa Hahmo pelimoottorissa. Kaikki Aladdin-pelin hahmot piirrettiin käsin paperille, skannattiin ja väritettiin digitaalisesti. Animaatioiden siirto peliä toistavalle tietokoneelle oli kuitenkin monimutkaisempi prosessi, sillä korkearesoluutiset kuvat oli pakattava sopivaan muotoon, ja pikselöitymisen myötä jokaista framea täytyi vielä manuaalisesti korjailla. Näillä piirroselokuvaan ja käsin piirrettyyn animaatioon perustuvilla pelihahmoilla oli hyvin dynaamisia ja yllättäviäkin liikkeitä ja asentoja verrattuna aiempiin peleihin, joissa perinteisesti hahmoilla oli hyvin tarkkarajainen sprite-alue, jolla liikkeet saattoivat tapahtua. (Strafofox 2021a.)

Vaikka käsinpiirrettyjä spritejä käytettiin peleissä paljon, vuonna 1992 julkaisussa *Mortal Kombat*issa käytetyt spritet olivat itse asiassa videokuvattua materiaalia. Kamppailulajien ammattilaisia kuvattiin yksiväristä taustaa vasten, kun he suorittivat peliin tarvittavia hahmoanimaatioita. Kuvatun materiaalin saturaa-tiota ja kontrastia tehostettiin, jotta kuvat toimisivat paremmin matalaresoluutioisina sprite-grafiikkoina. Tausta jouduttiin kuitenkin jokaisesta kuvasta poistamaan erikseen manuaalisesti. (Strafofox 2022.)

90-luvulla käsin piirrettyjen animaatioiden suosio alkoi kuitenkin pikkuhiljaa hii-pua, kun videopeleissä alettiin hyödyntää 3D-grafiikka ja markkinoille tuli 32- ja 64-bittisiä laitteistoja. 3D:tä oli toki nähty jo aiemmin, kuten jo vuonna 1984 *Atarin I, Robot* -nimisessä ammuntopelissä, mutta vasta konsolien viides sukupolvi toi 3D:n laajemmin käyttöön. Ennen kuin reaaliaikainen 3D oli teknisesti mahdollista, hyödynnettiin paljon pseudo-3D:tä tai 2.5D:tä, eli illuusion luomista 3D-maailmasta, vaikka oikeasti oltiin vielä kaksiulotteisessa ympäristössä. Esimerkiksi vuonna 1994 julkaistussa *Donkey Kong Country*ssä hahmot tehtiin kolmiulotteisiksi tietokonemallinnuksen avulla, mutta niiden pohjalta luodut hahmo-spritet olivat yhä kaksiulotteisia (ks. kuva 5). Tekijät jopa odottivat tämän tekniikan yleistyvän alalla, mutta teknologia kehittyi niin nopeasti, että pian reaaliaikainen 3D oli jo mahdollista. (Eden i.a.; Strafofox 2021b.) Pelimoottorit alkoivat yleistyä 90-luvun puolivälissä, ja ensimmäisenä 3D-renderöinnin mahdollisti

Quake engine vuonna 1996 (Layzelle 2021). Monet vanhat tutut hahmot saivat uuden ulkonäön kolmiulotteisuuden myötä, kuten aiemmin mainitut Donkey Kong, Mario sekä Link.



Kuva 5. Donkey Kong -pelin tietokonemallinnetuista hahmoista käytettiin pelissä 2D-spritejä. (Ruutukaappaukset: Straffox 2021b.)

2000-luvulla selainpelit valtasivat alaa, sillä ammattikäyttöön kehitetty Flash laajeni yllättäen amatöörien keskuuteen. Etenkin Tom Fulpin luoma Newgrounds-verkkosivusto toimi julkaisualustana monille innokkaille pelinkehittäjille ja taiteilijoille. Selainpelien keskiössä ei kuitenkaan ollut erityisesti laadukas hahmoanimointi, vaan ennemminkin luovuus ja tekijöiden kokeilunhalu sekä ajankoh- taisten asioiden ja ilmiöiden tuominen pelattavaan muotoon. Flashin aikakausi vaikutti indie-skenen syntyyn, sillä yhtäkkiä itsenäisillä ja riippumattomilla pelinkehittäjillä oli mahdollisuus nousta ihmisten tietoisuuteen. Flashia ei ole tuettu enää vuoden 2020 jälkeen. (Reeves 2018.) Internetin myötä pelit saivat monin- peliominaisuuksia, ja niiden luonne laajeni vielä lisää sosiaalisen median vaikutuksesta.

2010-luvulla videopelien budjetit alkoivat kasvaa hurjasti verrattuna aiempiin vuosiin ja muun muassa pilvipalvelut, virtuaalitodellisuus ja livestriimaus yleistyivät. Myös mobiilipelien suosio alkoi kasvaa räjähdysmäisesti. (Beresford 2019.)

Hyödynnän jatkossa nimenomaan tämän aikakauden pelejä esimerkkeinä, kun kerron hahmoanimaatiotekniikoista seuraavassa luvussa.

3 2D-hahmon animaatiotekniikat ja työkalut

Animaatiotekniikoilla tarkoitan niitä menetelmiä, joita käytetään 2D-hahmon animoimiseen. Olen jakanut ne kolmeen kategoriaan, joita ovat frame by frame -animaatio ja sen erilaiset toteutustavat, pala-animaatio ja luurankoanimaatio. Joskus puhutaan animaatiotyyleistä, mutta tässä opinnäytetyössä käytän nimenomaan tekniikka-käsitettä, sillä se sopii mielestäni työni tavoitteisiin paremmin. Näitä tekniikoita on osin haasteellista erottaa toisistaan, sillä niillä on paljon samankaltaisia ominaisuuksia, mutta erottavimpien tekijöiden perusteella olen jakanut ne näihin kolmeen kategoriaan.

3.1 Frame by frame -animointi

Frame by frame tarkoittaa yksinkertaisesti suomennettuna ”kuva kovalta”, ja siinä jokainen kuva eli kehys (*frame*) piirretään erikseen joko perinteisesti käsin tai digitaalisesti tietokoneohjelmalla. Kuvat voi luoda myös valokuvaamalla. Peräkkäiset toistetut kuvat luovat illuusion liikkeestä. Perinteiset frame by frame -animaatiot piirrettiin käsin paperille tai läpinäkyvälle selluloidikalvolle, jotka sitten valokuvattiin. Nykyteknologia mahdollistaa digitaalisen piirtämisen ja ylipäänsä kehittyneiden tietokoneohjelmien käytön, jotka helpottavat ja nopeuttavat prosessia huomattavasti. (Ip 2024.) Yhä kuitenkin nähdään aivan perinteiseen tapaan luotuja animaatioita, sillä muilla keinoilla nostalgista estetiikkaa on haasteellista saavuttaa.

StudioMDHR eli veljekset Chad ja Jared Moldenhauer sekä heidän tiiminsä julkaisivat Cuphead-ammuntapelin vuonna 2017. He ihailivat 1930-luvun Fleischerin ja Disneyn piirroselokuvia ja halusivat tuoda tämän tyylin myös omaan peiliinsä. Kaikki pelin animaatiot on siis käsin piirretty paperille ja taustakuvat maalattu vesiväreillä. Veljekset kokivat perinteisen animaatiotekniikan olevan ainoa keino saavuttaa 1930-luvun estetiikka, ja mielestäni he onnistuivat siinä

erinomaisesti (ks. kuva 6). Lopulta käsinpiirretyt animaatiot väritettiin Photoshopissa. Pelin animaatioissa myös ruutunopeus (*fps, frames per second*) on 24, kuten myös vanhoissa piirroselokuvissa. (Unity i.a.) Ruutunopeus tai kuvataajuus tarkoittaa sitä, kuinka monta uutta kuvaa piirtyy ruudulle sekunnin aikana, ja korkeampi kuvataajuus merkitsee usein sujuvampaa liikettä (Wikipedia 2024a). Lisäksi Cuphead on animoitu ”ykkösillä”, eli jokaisessa 24 kuvassa tapahtuu muutos yhden sekunnin aikana. Esimerkiksi kakkosilla animoitaessa vain joka toisessa kuvassa tapahtuu muutos, eli tällöin sekunnin aikana näkyy vain 12 eri kuvaa. Kakkosilla animointi on yleistä, sillä se säästää aikaa ja rahaa, mutta erityisesti nopeat liikkeet on syytä animoida ykkösillä. (Unity i.a.; Williams 2009.)

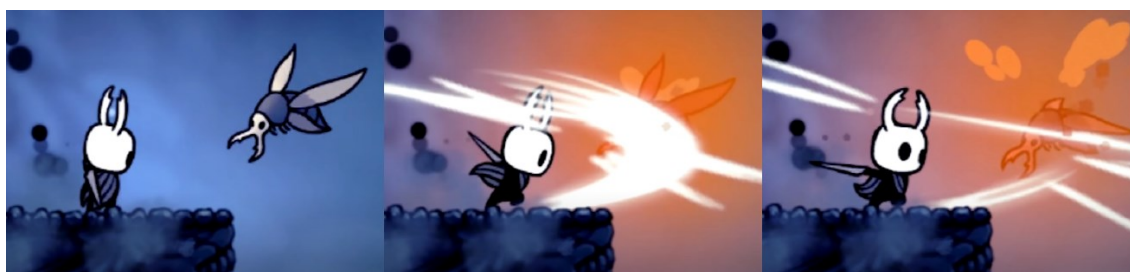


Kuva 6. Cuphead-pelin taustat on maalattu vesiväreillä ja piirroselokuvamaiset hahmoanimaatiot toteutettu frame by frame -tekniikalla. (Ruutukaappaus: Pro-safiaGaming 2020.)

Useat pelintekijät, jotka havittelevat animaatioihinsa tällaista perinteistä visuaalista ilmettä, luovat animaatiot frame by frame -tekniikalla. Australialainen indie-peliyhtiö Team Cherry julkaisi toimintaseikkailupelinsä Hollow Knightin vuonna 2017. Ari Gibson piirsi käsin kaikki hahmot ja ympäristöt, jotka sitten Photoshopissa muutti digitaaliseen muotoon ja animoi. Pelin hahmot ovat erilaisia

hyönteisiä, eli ne ovat aika yksinkertaisia ulkoasultaan, eikä esimerkiksi päähahmolla ole käsiä – tämä on varmasti nopeuttanut animaatioiden luomista. Muista käsittelemistäni 2D-peleistä poiketen Hollow Knightissa on pelin taustojen asettelussa hyödynnetty Unityn 3D-ympäristöä parallaksin tehostamiseksi. Tämän vuoksi pelin visuaalinen syvyysvaikutelma on voimakas, vaikka liikutaan 2D-ympäristössä. Pelin animaatioiden elävöittämisessä on lisäksi hyödynnetty fysiikkasimulaattorin partikkelianimaatioita (*physics-based particle animations*), eli esimerkiksi kun hahmo juoksee eteenpäin tai hyppää, maassa olevat lehdet leijailevat tai pölypilvi pölyyää. Nämä ovat siis pelimoottorin fysiikkasimulaattorin painovoimaan perustuvia partikkeleja, joiden animoimista ei ole tehty käsin. (ThatGuyGlen 2020; Video Game Animation Study 2019.) Nämä pienet lisäanimaatiot kuitenkin tehostavat hurjasti käsinpiirrettyä hahmoanimaatiota, mutta en käsittele opinnäytetyössäni enempää tällaisia hahmoanimaation ulkopuolisia ominaisuuksia.

Hollow Knightin hahmoanimaatioissa näkyy hyvin se, kuinka frame by frame -tekniikalla animoitaessa täytyy tehdä valintoja framejen poisjättämisen suhteen. Dan Root on tehnyt Hollow Knight -pelin animaatioista huomioita Youtube-videollaan (2019). Esimerkiksi pelin päähahmon hyökkäysanimaatiossa (ks. kuva 7) hahmolla on vain yksi ennakointi (*anticipation*) -frame, jossa tikari on hahmon takana valmiina hyökkäykseen. Heti seuraavassa framessa näkyy iso valokaari ilmentämässä tikarin reittiä, vaikka se on vielä melkein alkuasennossa. Seuraavassa framessa tikari on palautunut hahmon selän taakse ja valoviiva on hieman heikentynyt.



Kuva 7. Hollow Knight -pelin päähahmon hyökkäysanimaatiossa vain olennaisimpia frameja on käytetty. Valoviiva ilmentää tehokkaasti nopeaa tikarin iskua. (Ruutukaappaukset: Video Game Animation Study 2019.)

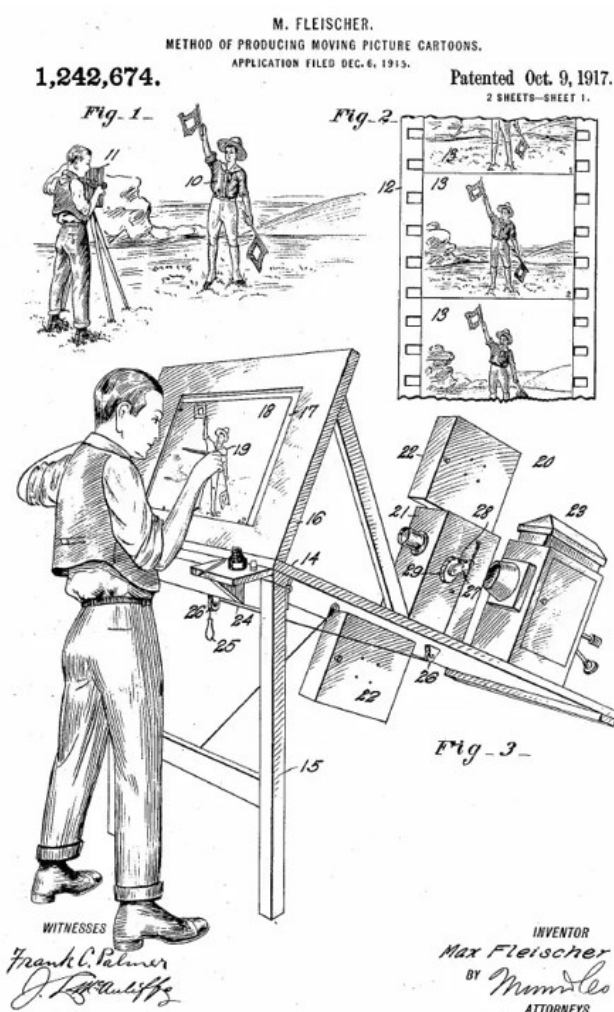
Miekkaa ei missään vaiheessa piirretä näkyväksi tälle reitille, vaan valoviiva kertoo katsojalle, että miekka on siitä kulkenut. Tämä muutaman kuvan kestävä hyökkäysanimaatio kertoo siitä, että liike tapahtuu hyvin nopeasti, ja kaikki tarpeellinen tulee näistä muutamasta kuvasta ilmi. (Video Game Animation Study 2019.) Videopeleissä pelaajalla täytyy olla kontrollin tunne, eikä aikaa ole jokaisen framen näyttämiseen. Pelaaja turhautuisi nopeasti, jos animaatiot olisivat yksityiskohtaisia, sillä liikkeen tuntu hidastuisi ja on epäolennaista näyttää frameja, jotka eivät palvele pelin toiminnallisuutta.

Myös Skullgirls-videopelin pääanimaattori Mariel Cartwright on puhunut videopelikehittäjien konferenssissa eli GDC:ssä (*Game Developers Conference*) siitä, kuinka tärkeää frame by frame -animaatioissa on keskittyä hahmon avainkehysiin (*keyframe*) eli pääasentoihin. Avainkehukset kertovat liikkeen pääkohdat, eli niistä tapahtuma on selkeästi luettavissa. Välikehukset (*inbetweens*) taas täydentävät liikettä kahden avainkehysten välillä. (Williams 2009.) Cartwright halusi luoda hahmoilleen avainkehysiin perustuvaa liikettä, joka olisi luettavissa mahdollisimman vähällä välikehysten käytöllä. Sen lisäksi että tämä säästää frame by frame -tekniikassa aikaa, palvelee se pelin tarkoitusta, kuten aiemmin Hollow Knightsin kohdalla mainitsin – kaikki turha liike on syytä karsia. Cartwright puhui konferenssissa myös siitä, kuinka hän alussa ”ylianimoii” hahmojen liikkeitä ja huomasi pian niiden olevan turhan yksityiskohtaisia ja palvelevan huonosti pelin funktiota. Hän korjasi animaatioita poistamalla turhia välikehyskiä ja sai täten ne paljon ytimekkäimmiksi ja vaikuttavammiksi. (GDC 2016.)

3.1.1 Rotoskooppaus

Rotoskooppaus (*rotoscoping*) on animaatiotekniikka, jossa videokuvattua materiaalia jäljennetään läpi piirtämällä kuva kovalta. Kun oikea ihminen, esimerkiksi näyttelijä, tekee liikkeen, voidaan saavuttaa mahdollisimman realistinen lopputulos. Tekniikka sai alkunsa vuonna 1915 animaattori Max Fleischerin kehitettyä rotoskoopin (ks. kuva 8) helpottamaan realistisemman animaation luomista. Laitte heijasti filmin kuvat lasipaneelille, jolta animaattori pystyi jäljentämään ne paperille yksi kerrallaan. (Fleischer Studios i.a.) Nykypäivänä tietokoneet ovat

toki helpottaneet rotoskooppausta käytännössä. Rotoskooppauksen mahdollisuudet voivat kuitenkin myös kääntyä animaatiota vastaan: liian realistinen liike imitoi liikaa todellisuutta, eivätkä animaation peruseriaatteet, kuten liioittelu, pääse toteutumaan. Rotoskooppaus voi toimia hyvänä työkaluna, jos sitä käytetään maltillisesti. Darlie Brewster, yksi Thinking animation -teoksen kanssa yhteistyötä tehneistä animaattoreista, sanoo, ettei animaation tarkoituksena ole imitoida oikeaa maailmaa, vaan vangita jokin vaikutelma. Videokuvattu ihminen ei voi pystyä siihen, mihin animaatio pystyy liioittelun ja karrikoinnin ansiosta. (Jones & Oliff 2008.)



Kuva 8. Rotoskoopin avulla jäljentäminen filmiltä tuli mahdolliseksi, ja aidomman liikkeen vangitseminen animoituihin hahmoihin helpottui. (Fleischer Studios i.a.)

Viime vuonna (2023) julkaistun tasohyppelypelin Lunarkin suunnittelija Johan Vinet on kertonut käyttäneensä pelin hahmoanimaatioissa rotoskooppaustekniikkaa. Hän kuvasi itseään tekemässä tarvittavia liikkeitä, toi videomateriaalin After Effectsiin ja asetti resoluution ja kehysnopeuden (*frame rate*) vastaamaan pelin asetuksia. Sitten hän jäljensi jokaisen framen käsin luodakseen yhtenäisen animaation. Pikseliä käyttämällä hän havitteli peliin entisaikojen tasohyppelypelien estetiikkaa. (WayForward 2023.) Verrattuna noihin 90-luvun peleihin, päähahmo Leon liikkeet ovat hyvin realistisia ja jouhevia korkean kehysnopeuden ansiosta. Hahmo ei voi myöskään hypätä liioitellun korkealle tai vaihtaa suuntaa ilmassa, koska Vinet halusi hahmon liikkumisen jäljittelevän tosielämää mahdollisimman paljon. Mielestäni hahmoanimaatiosta kuitenkin puuttuu kiinnostavuus, vaikka ilmiötä on vaikea selittää. Richard Williams pohtii kirjassaan *The animator's survival kit* rotoskoopauksen sudenkuoppia. Videokuvattua liikettä ei voi täydellisesti jäljentää paperille, koska informaatiota on yksinkertaisesti liikaa, ja täten jäljitely animaatio menettää uskottavuutta. Liikkeestä voi tulla leijuvaa ja painotonta, sillä painovoima menettää tehoaan. Uskottavuuden säilyttämiseksi rotoskoopatun liikkeen ääriasentoja tulisi liioitella hieman. *Uskottavuus* on tärkeämpää kuin realismi. Williamsin mukaan pelkätään videokuvatusta materiaalista täydellisesti jäljitely liike voi tuntua katsojasta oudolta – materiaali ei ole elokuvaa mutta ei myöskään animaatiota. (Williams 2009.) Lunarkissa tapahtuu mielestäni jotakin tämänkaltaista, sillä päähahmon liike on hieman leijuvaa ja painotonta. Lisäksi matalaresoluutioisen hahmon liikkuminen hyvin realistisesti luo mielessäni jonkinlaisen ristiriidan. Henkilökohtaisesti koen, että pikseleitä olisi voitu käyttää hieman runsaammalla kädellä, jotta tämä liikkeen sujuvuus kohdattaisiin. Lunarkissa on kuitenkin mielenkiintoinen tyylien yhdistelmä.

Rotoskoopkaus on siis oivallinen työkalu informaation ja havaintojen keräämiseen liikkeistä, mutta animaattorin on tiedettävä, mitä liikettä on syytä alleviivata tai olla huomioimatta. Kuten Williams (2009) teoksessaan mainitsee, saavutetaan tosielämän imitaatio videokuvaamalla, mutta animaattorilla on valtaa keksiä jotain uutta.

3.1.2 Pikselitaide

Kuten aiemmin kirjoitin, oli niin sanottu ”pikselityyli” pelihahmojen varhaishistoriassa juuri teknologian rajoituksista johtuen ainoa keino luoda peligrafiikkaa. Kuitenkin nykypäivänä pikselityyli tai pikselitaide on muotoutunut omaksi, nostalgiaa tavoittelevaksi tyyliuunnakseen. Miksi pikselitaide nousi uudelleen pinnalle ja on pystynyt säilyttämään asemansa ja suosionsa videopeleissä? Nykypäivän teknologia mahdollistaa uskomattomia realistisia ja moniulotteisia pelimaailmoja. Pikselitaide taas on ilmaisumuotona minimalistinen ja rajallinen ja vaatii kärsivällisyyttä. Ehkä juuri tämä rajoittuneisuus toimii hyvänä vastapainona nykyajan hektisille digitaalisille ympäristöille. Pysyvyytensä ansiosta pikselitaide luo katsojassa tuttuuden tunteita, vaikka nykypäivänä se onkin huomattavasti korkearesoluutioisempaa ja värien käyttö ei ole teknologian puolesta rajoitettua. Lisäksi markkinoille on tullut kehittyneitä ohjelmistoja pikselitaiteen luomiseen. (Verma 2023.)

Peleissä, joissa on hyödynnetty pikselityyliä, on hahmoanimaatiot useimmiten tehty frame by frame -tekniikalla hyödyntäen *sprite sheetejä*. Sprite sheet on käytännössä pienistä kuvista koostuva iso kuva, jossa on läpinäkyvä tausta. Eli esimerkiksi hahmon kävelysyklin kaikki framet eli yksittäiset kuvat ovat tallennettuna isoon kuvaan. Tämä parantaa pelimoottorin suorituskykyä, sillä jokaista animaatioon kuuluvaa kuvaa ei tarvitse ladata yksittäin, vaan ne löytyvät samasta tiedostosta. (Loew 2023.) Hahmoanimaation sprite sheetin luonti onnistuu joissakin ohjelmissa automaattisesti, kuten esimerkiksi Adobe Animateissa. Kerron sprite sheetien ominaisuuksista vielä tarkemmin luvussa Hahmo pelimoottorissa.

Aiemmin esittelemäni Lunark on esimerkki pikselityyliä edustavasta pelistä. Vuonna 2016 suuren suosion saavuttanut Stardew Valley on myös tehty pikselityyllillä, mutta hahmoanimaatiot on Lunarkiin verrattuna toteutettu hieman eri tavalla. ConcernedApe eli Eric Barone loi maatilasimulaattorin Stardew Valleyn yksin neljässä vuodessa. Hahmoanimaatiot on luotu frame by frame -tekniikalla hyödyntäen aiemmin mainittuja sprite sheetejä. Legend of Zelda -peliä käsitellessäni mainitsin *top-down*-perspektiivin, ja myös Stardew Valleyn hahmot on

tällä tavoin animoitu liikkumaan neljään eri ilmansuuntaan. Animaatioiden peilaaminen oikealle tai vasemmalle onnistuu nopeasti, mutta eteen- ja taaksepäin menevä hahmo tulee piirtää uudestaan eri kuvakulmista. ConcernedApe on selvästi hyödyntänyt animaation perusperiaatteita hahmoanimaatioissaan, kuten muun muassa ennakointia (*anticipation*), liioittelua (*exaggeration*) ja ajoitusta (*timing*). Esimerkiksi hahmon louhiessa kiviä hahmo nostaa ensin hakun päänsä päälle, jonka jälkeen iskee sillä kiveä (ks. kuva 9). Iskun aikana hahmo on pysähtyneenä ilmassa samassa asennossa, kunnes kivi on hajonnut. (Stardewvalley.net i.a.; Foxy Fern Animation 2021; Williams 2009.)



Kuva 9. Stardew Valley -pelihahmon louhinta-animaation kuusi ensimmäistä framea, joissa kivi hajoaa jo ennen hakun siihen osumista. Ruutukaappaukset pelistä *Stardew Valley* (2016).

Se, että hahmo pysyy ilmassa useamman framen ajan, vahvistaa tunnetta siitä, että hän käyttää voimaa. Kuva kuvalta tarkasteltuna huomataan, että kivi itse asiassa hajoaa jo ennen hakun siihen osumista. Tämä kuitenkin näyttää pelissä luonnolliselta. Kiven hajoaminen vasta hakun siihen osuttua näyttäisi todennäköisesti liian hitaalta tapahtumalta. Lisäksi hahmo palautuu alkuasentoon nopeasti kiven hajoamisen jälkeen ja hakku katoaa näkyvistä. Olennaisin liike on jo

tapahtunut, ja hahmon tulee siirtyä sujuvasti seuraavaan animaatioon. Louhimisanimaatio ei siis ole realistinen, mutta juuri tällä tavalla se on uskottava.

3.1.3 Stop motion

Stop motion -tekniikassa hyödynnetään valokuvausta, eli fyysinen hahmo luodaan esimerkiksi muovailuvahasta. Sitten hahmon liikkeet valokuvataan kuvakuvulta, jotta luodaan illuusio liikkeestä. Olennaista on siis fyysisen hahmon käyttäminen käsinpiirretyn sijaan. Muovailuvahan lisäksi hahmon voi luoda paperinpaloista tai käyttää vaikka käsinukkeja. Tapoja on useita, mutta stop motion -tekniikka on hyvin tunnistettava, uniikki ja vaatii tekijöiltä kärsivällisyyttä. (Adobe i.a.)

Vaikka aiemmin olen tekniikoita käsitellessäni hyödyntänyt esimerkkeinä nimenomaan 2010-luvun pelejä, mennään tässä kohtaa hieman ajassa taaksepäin. Vuoden 1996 klikkailemalla etenevä ongelmanratkaisupeli The Neverhood on luotu stop motion -tekniikalla. Syy, miksi käytän tässä esimerkkinä 90-luvun peliä, on se, että monissa nykypäivänä toteutetuissa stop motion -peleissä on hyödynnetty paljon myös 3D:tä. Pyrkimyksenä on työssäni keskittyä nimenomaan 2D-toteutuksiin. Stop motion on peliteollisuudessa tekniikkana harvinaisempi. Neverhood-pelissä sekä pelin taustat että Klaymen-niminen päähahmo on rakennettu muovailuvahasta. Pelistä kertovassa dokumentissa tekijät mainitsevat, että matkan varrella he pohtivat myös 3D:n hyödyntämistä pelitaiteessa, mutta halusivat loppujen lopuksi toteuttaa sen mahdollisimman perinteisillä menetelmillä. (Robobin 2016.) Stop motion -tekniikassa hahmon kontrollointi voi olla haastavaa, sillä frame by frame -valokuvattujen liikkeiden tulee olla hienovaraisia. Vähäisetkin virheet tallentuvat kameralle, ja niitä on jälkeenpäin vaikeaa korjata. Tekniikka on hyvin hidas ja kärsivällisyyttä vaativa, mutta voi olla joissakin tapauksissa matalampi budjetiltaan kuin kalliit digitaaliset toteutukset. Klaymenillä on useita erilaisia animaatioita, joita pelaaja triggeröi klikkailemalla maailmassa näkyviä esineitä. Hahmolla on tunnistettava ja persoonallinen ulkomuoto, ja se on luonnollisesti myös sopiva muovailuvahatoteutukseen.

3.2 Pala-animaatio

Pala-animaatio (*cutout animation*) vaatii hahmon pilkkomisen osiin eli paloihin. Yleensä perinteisestä ihmishahmosta siis erotellaan omiksi paloikseen keskivartalo, pää ja raajat. Puhun tämän opinnäytetyön yhteydessä nimenomaan digitaalisesta pala-animaatiosta. Perinteinen pala-animaatio sen sijaan on stop motion -tekniikkaa vaativa toteutus, jossa hahmon paloja liikutellaan ja valokuvataan kuva kuvalta liikkeen illuusion luomiseksi. Digitaalisessa pala-animaatiossa prosessia voidaan nopeuttaa automaattisen *tweeningin* avulla sekä *rigaamalla* hahmo eli luomalla sille luuranko. Kerron luurankoanimaatiosta yksityiskohtaisemmin seuraavassa alaluvussa. Lisäksi hahmon osien *meshejä* muokkaamalla animaatioon voi saada lisää elävyyttä. (Toon Boom i.a.) Pala-animaatio voi olla hyvin yksinkertainen tai monimutkainenkin riippuen hahmon ulkonäöstä ja sille suunnitelluista liikkeistä. Digitaalisissa piirto-ohjelmissä hahmon osat ovat yleensä omilla tasoillaan eli *layereilla*.

Thunder Lotus Game -peilyhtiön vuonna 2020 julkaisema Spiritfarer yhdistää frame by frame -tekniikkaa pala-animaatioon. Spiritfarer on sivuttaisvierittävä hallinnointipeli (*management game*), jossa pelaajan tehtävänä on opastaa henkimaailman ystävät tuonpuoleiseen. Pelin tekemiseen on käytetty Toon Boom Harmony -animointiohjelmaa ja Unity-pelimoottoria. Animaattorit halusivat luoda hahmoilleen mahdollisimman ilmeikästä, uskottavaa ja yksilöllistä animaatiota. Pääasiassa hahmot on animoitu Toon Boom Harmonyssa rigin avulla, sitten pakattu sprite sheeteihin ja tuotu lopulta pelimoottoriin. Rigin luominen edellyttää hahmon pilkkomista osiin ja mahdollistaa myös nopeamman animoinnin tweeningin ansiosta. Tweening eli inbetweening tarkoittaa välikehysten luomista avainkehysten välille, kuten luvussa Frame by frame -animointi käytiin läpi. Tässä yhteydessä kuitenkin tietokoneohjelma laskee automaattisesti avainkehysten perusteella niiden väliin tulevat frame't, eikä niitä tarvitse manuaalisesti piirtää tai asetella paikoilleen. Tämä säästää huomattavasti aikaa verrattuna kokonaan frame by frame -tekniikalla tehtävään animaatioon. Spiritfarerin animaattorit joutuivat kuitenkin joitakin kohtauksia tai liikkeitä piirtämään kokonaan uudelleen kuva kuvalta, sillä rigillä animoidessa vastaan tulee tiettyjä rajoituksia.

2D-rigi toimii hyvin kaksiulotteisessa liikkeessä, mutta sillä ei pystytä muuttamaan syvyyttä tai kuvakulmia kovinkaan tehokkaasti. (Hartley 2021; Academy of Interactive Arts & Sciences 2021.) Joitakin työkaluja tähän kuitenkin on olemassa, vaikka suurempi kuvakulman muutos vaatii todennäköisesti frame by frame -tekniikkaa.

Kanadalainen pelistudio Klei Entertainment on käyttänyt pelissään Mark of the ninja mielenkiintoista pala-animaation verrattavissa olevaa tekniikkaa. Kyseessä on hiiviskelypeli (*stealth game*), jossa pelaaja pääsee kokeilemaan taitojaan varjoissa hiippailevana ninjana (Klei i.a.). Pelin sulavat hahmoanimaatiot tukevat mielestäni erinomaisesti ninjateemaista tarinankerrontaa ja auttavat pelaajaa eläytymään hiiviskelevän ninjan olemukseen. Ninjahahmo kiipeilee, hiiviskelee ja hyppii, joten etenemisanimaatiot ovat monipuolisia. Lisäksi peli sisältää paljon erilaisia taistelukohtauksia, joissa hahmon animoimiseen on tarvittu runsaasti eri asentoja. Klei Entertainment kertoi prosessistaan hahmoanimaatioiden taustalla vuonna 2014 videopelikehittäjien konferenssissa. Tekijät kertoivat luoneensa hahmoanimaatiot Flashilla hyödyntäen *symboleita*, joista kerron vähän myöhemmin lisää. (GDC 2017.) Adobe Animate CC, joka tosiaan ennen vuotta 2016 tunnettiin nimellä Flash Professional, on animaatioiden tekoon tarkoitettu ohjelma, joka mahdollistaa sekä rasteri- että vektoripohjaisten animaatioiden luomisen useille eri alustoille (Wikipedia 2024b). Mark of the ninjan tekijät eivät riganneet hahmoja animaatioita varten, mutta pala-animaation periaatteen mukaisesti kuitenkin pilkkoivat hahmot osiin. Animointia varten jokainen hahmon osa tehtiin Flashilla omaksi symbolikseen (ks. kuva 10), ja automaattinen tweening nopeutti näiden vektorigrafiikalla toteutettujen hahmojen animointia. (GDC 2017.) Ei ole myös lainkaan tavatonta pilkkoa pelkästään frame by frame -tekniikalla animoitavaa hahmoa osiin, sillä omilla layereillaan olevia osia on helpompi hallinnoida.

Animatessa on kolme symbolityyppiä: *graphic*, *movieclip* ja *button*. Graphic -symboli säilöo sisällään kokoelman frameja, eli kuten Mark of the ninjan esimerkissä, animaattorit säilöivät yhteen symboliin useita kuvakulmia ja asentoja samasta kehon osasta. Tällöin esimerkiksi käsivarren asento on nopeasti

vaihdettavissa animaation edetessä, eikä sitä siis tarvinnut piirtää aina uudelleen. Movieclip-symboli sen sijaan säilöö animaatiopätkiä, jotka ovat kätevästi uudelleen käytettävissä projektin aikana. Button-symbolit taas toimivat interaktiivisina painikkeina. (Adobe 2023.)



Kuva 10. Mark of the ninja -pelin päähahmo pilkottuna osiin, joista jokainen on Adobe Animate:ssa oma symbolinsa. (Ruutukaappaus: GDC 2017.)

Mark of the ninjan hahmojen liikkeet ovat hyvin dynaamisia, ja tässä tapauksessa rigi olisi saattanut rajoittaa hahmon liikettä. Lisäksi tekijät kertovat konferenssissa, että perinteinen frame by frame -tekniikka ja sen perusteella koostetut korkearesoluutioiset sprite sheetit olivat pelimoottorille aivan liian työläitä, joten siksi he päätyivät tällaiseen toteutustapaan. (GDC 2017.)

3.3 Luurankoanimaatio

Luuranko- tai luuanimaatio vaatii hahmon pilkkomisen osiin, kuten pala-animaatiokin. Nämä tekniikat osin limittyvät toisiinsa ominaisuuksiltaan. Pala-animoitu hahmo ei kuitenkaan vaadi rigin luomista, kun taas rigin luominen melkein poikkeuksetta vaatii hahmon paloittelun. Hahmon rigaaminen tarkoittaa sitä, että

hahmolle luodaan ranka ja luuranko, jonka nivelet ja luut mahdollistavat hahmon liikkeiden kontrolloimisen. Useimmat animaation tekoon tarkoitettut ohjelmat, kuten esimerkiksi Adobe Animate, Toon Boom Harmony ja Spine sisältävät erilaisia työkaluja hahmon rigin luomiseen ja kontrolloimiseen. (Brown 2024.) Humanoidin hahmon rigin eli rangan ”juuri” tai alku sijaitsee yleensä lantiossa, sillä liike lähtee kehon keskiosasta. Luut liitetään rankaan ja toisiinsa nivelten (*joint*) avulla. Itse luut ovat taipumattomia, mutta nivelten avulla ”luuketjujen” eli raajojen liikuttelu ja taivuttaminen on mahdollista. Nivelet siis tulevat hahmossa niihin kohtiin, joista sen on tarkoitus taipua. (Maestri 2006.)

Ubisoftin vuonna 2013 julkaisemassa Rayman Legends -tasohyppelypelissä hahmojen liikkeet on toteutettu rigaamalla. Hahmot on pilkottu osiin ja osat on pakattu sprite sheeteihin. Osista kootaan hahmo, jolle rakennetaan luuranko. Ubisoft hyödyntää myös *sprite swappausta* eli spritejen vaihtelemista keskenään. Esimerkiksi minotaur-hahmon sprite sheetillä on hahmon suu-spriteille useampi vaihtoehto (ks. kuva 11), jolloin hahmon ilmettä voidaan helposti muuttaa vaihtelemalla spritejä keskenään. (HardwareHeaven 2013.)

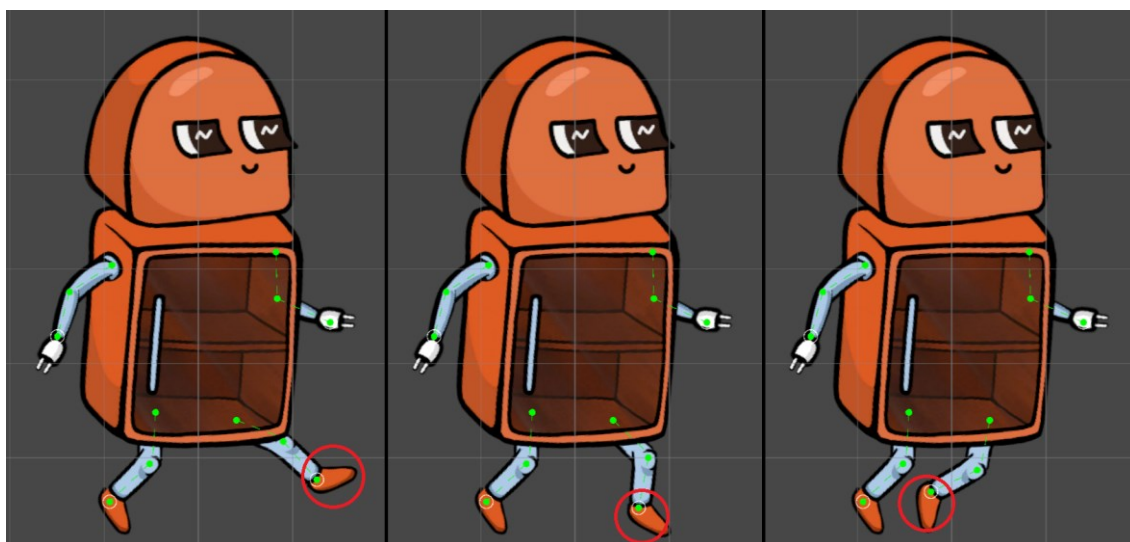


Kuva 11. Minotaur-hahmon suu-spritein eri vaihtoehdot. (Ruutukaappaus: HardwareHeaven 2013.)

Rayman Legendsin päähahmollekin on erilaisia silmiä ja suita, jotta ilmeiden vaihtaminen onnistuu helposti. Tällainen palasten vaihtelu tuo liioittelun mahdollisuuden myös luurankoanimoidulle hahmolle, sillä tavallisesti se voi luuanimaatioissa olla jokseenkin rajoitettua – ainakin verrattuna frame by frame -tekniikkaan. Lisäksi samaa luurankoa voidaan käyttää useammalle hahmolle, joten hahmoanimaation tuotantoprosessi nopeutuu huomattavasti. (HardwareHeaven 2013.)

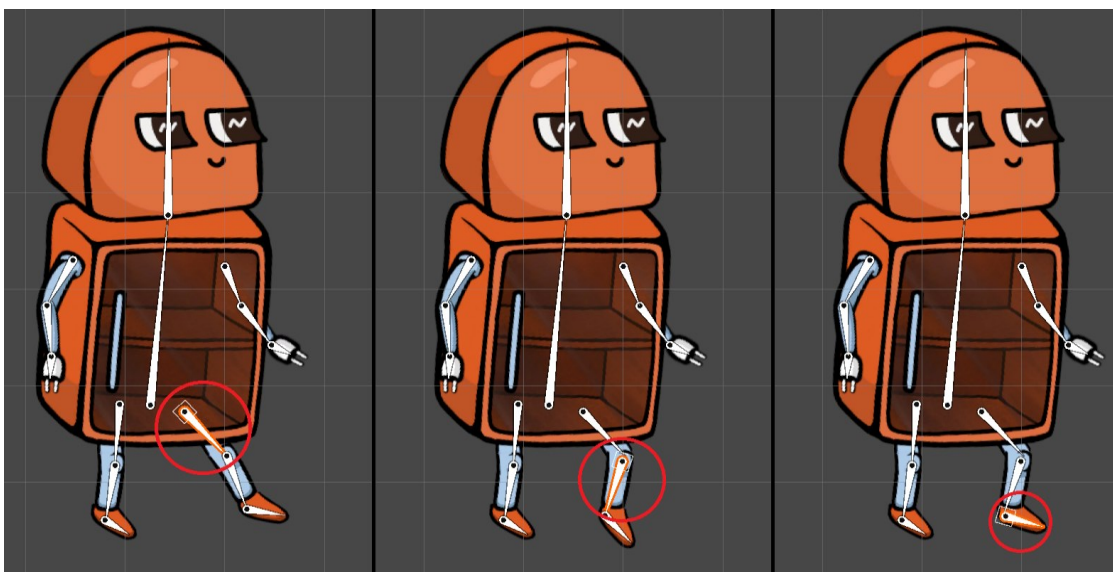
3.3.1 Käänteinen kinematiikka eli IK

2D-hahmon luurangolla ei välttämättä tarvitse olla kaikkia luita, sillä etenkin yksinkertaisten pelihahmojen kädet ja jalat eivät välttämättä taivu kyynärnivelten ja polvien kohdalta. Oli luuranko sitten monimutkainen tai yksinkertainen, luut ovat joka tapauksessa hierarkkisessa järjestyksessä toisiinsa nähden. *Inverse kinematics* eli käänteinen kinematiikka (IK) nopeuttaa rigatun hahmon animointia ja auttaa jäljittelemään luonnollisia liikeratoja. Käytännössä käänteinen kinematiikka toimii siten, että jos hahmon jalkaterästä otetaan kiinni ja liikutetaan sitä ylös tai alas, hahmon sääriluu ja reisiluu seuraavat mukana ja polvi voi taipua. (Maestri 2006.) Esimerkkinä tästä toimii luomani jääkaappi-hahmo, jonka jalkaterää liikutettaessa aiempi luuketju seuraa perässä (ks. kuva 12).



Kuva 12. Hahmon jalan liikuttaminen IK:n periaatteilla. Pelkästään jalkaterän liikuttaminen muuttaa nopeasti koko jalan asentoa.

Tällöin esimerkiksi kävelysykliä animoitaessa ei tarvitse siirtää jokaisen luun paikkaa erikseen manuaalisesti, vaan käänteinen kinematiikka mahdollistaa nopeamman työskentelyn. *Forward kinematics* eli suorakinematiikka sen sijaan tarkoittaa sitä, että tässä esimerkissä ensin taivutetaan hahmon reisiluu oikeaan kulmaan, jonka jälkeen taivutetaan sääriluuta ja vasta sitten jalkaterää (ks. kuva 13). Prosessi on huomattavasti hitaampi, sillä jokaista luuta täytyy säätää erikseen. (Maestri 2006.)



Kuva 13. Hahmon jalan liikuttaminen FK:n periaatteilla. Punaisella merkittynä näkyy, missä järjestyksessä luita *rotatoidaan* halutun lopputuloksen saavuttamiseksi.

IK ei kuitenkaan luonnostaan tunnista realistisia liikeratoja, sillä se ei tiedä, mihin suuntaan nivelet eivät saa taipua. Aiemmin mainitsemassani esimerkissä (kuva 12) sääriluu siis voisi reisiluuhun nähden taipua väärään suuntaan. Tämä voidaan korjata asettamalla IK:lle rajoituksia (*constraints*), jolloin liikeradasta saadaan luonnollisempi ja taas nopeutetaan animaation työnkulkua. Koska käänteinen kinematiikka ei aina toimi toivotulla tavalla, animaattorilla on oltava tietämystä anatomiasta. (Maestri 2006.) Kerron IK:n luomisesta hahmolle vielä tarkemmin luvussa Hahmo pelimoottorissa.

3.3.2 Meshien muokkaaminen

Sekä luuranko- että pala-animaatioon liittyy olennaisesti meshien muokkaaminen (*mesh deformation* tai *free-form deformation*). Tämä on hieman erilaista eri ohjelmista riippuen, mutta pyrin kertomaan meshien muokkaamisesta lyhyesti olennaisimmat pääpiirteet. Joskus mesheistä puhuttaessa tarkoitetaan spritejen *geometriaa*. Käytän tässä kuitenkin selkeyden vuoksi mesh-termiä. Tämä työkalu tai ominaisuus mahdollistaa kaksiulotteisten pintojen hienovaraisen muokkauksen. Ohjelmasta riippuen, hahmon osille eli spriteille luodaan *polygon mesh* eli polygoniverkko. Perinteisesti 3D-grafiikassa polygoniverkko on tietenkin kolmiulotteinen, mutta tässä tapauksessa nimenomaan hahmon osien kaksiulotteiset pinnat jaetaan polygoneihin. Useissa ohjelmissa tämän verkon muodostuminen on automaattista, mutta sitä voidaan hienosäätää manuaalisesti. Polygonien verteksejä eli kärkiä siirtämällä kappaleen muoto muuttuu. Tällainen hienovarainen osien säätäminen voi lisätä hahmon animaatioon elävyyttä. Esimerkiksi vaatteiden kankaiden heiluminen voi olla nopeampaa luoda meshiä muokkaamalla sen sijaan, että piirtäisi tällaista liikettä kuva kuvalta. Liikkeen lopputulos voi myös olla sulavampi, sillä animaatiota voidaan esimerkiksi hidastaa ilman, että liike muuttuu jumittavaksi – kuten frame by frame -tekniikalla luodun liikkeen kohdalla kävisi. Meshiä ei kuitenkaan voida muokata kovin paljon, tai osan alkuperäinen muoto muuttuu liikaa, eikä lopputulos näytä enää uskottavalta. Monimutkaiset meshit voivat myös hidastaa pelin toimintaa, joten on tärkeää huomioida, mikä pelin toiminnallisuuden kannalta on tarpeellista. (Wulverblade 2014.) Esimerkiksi Spiritfarer-pelissä päähahmon seisoessa ulkona paikoillaan, hänen hattunsa lieri liehuu hienovaraisesti tuulessa. Lisäksi erilaiset pään asennot muuttavat hatun lierin perspektiiviä, ja juuri tällaiset hienovaraiset muutokset voi kätevästi saavuttaa hatun lieri -kappaleen meshiä muokkaamalla.

Meshin muokkauksen mahdollistavissa ohjelmissa voi olla paljon eroja sen suhteen, kuinka hyvältä lopputulos näyttää. Esimerkiksi Spine on suosittu ohjelma mesh-deformaatiota hyödyntävien tekijöiden keskuudessa, mutta se voi olla hienokas joidenkin tekijöiden mielestä. Seuraavassa luvussa käsittelen hahmon riggaamista pelimootorissa, jolloin esittelen konkreettisia esimerkkejä rigin luomiseen liittyvistä välivaiheista ja huomioista.

4 Hahmo pelimoottorissa

Pelimoottorit ovat videopelien ohjelmistokehyksiä, jotka helpottavat ja yksinkertaistavat videopelien luomista. Ne sisältävät erilaisia työkaluja pelinkehittämistä varten, kuten esimerkiksi renderöintimoottorin, fysiikkasimulaattorin, animoinnin, ohjelmointikielen ja äänet. Esimerkiksi Unity ja Unreal Engine ovat nykypäivän suosituimpia pelimoottoreita. Pelimoottoreiden sisäiset animaatiojärjestelmät (*animation systems*) helpottavat animaatioiden hallintaa. (Wikipedia 2024c.) Tässä luvussa käsitelen hahmon rigaamista pelimoottorissa, animaation eri toistotapoja sekä animaatiolooppeja. Kaikissa esimerkeissä hyödynnän Unitya pelimoottorina.

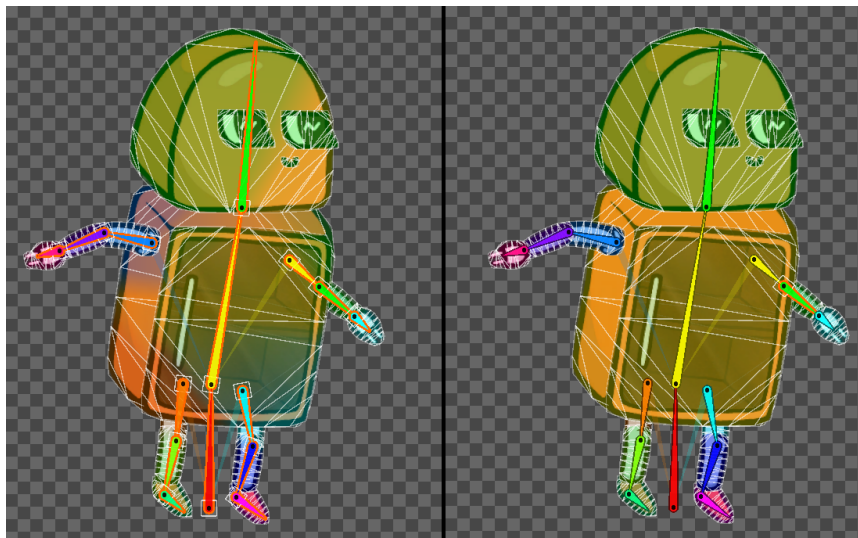
4.1 2D-hahmon rigaaminen Unityssa

Aiemmin olen maininnut erilaisista ohjelmista, joilla hahmon rigin rakentaminen on mahdollista. Hahmo voidaan kuitenkin rigata ja animoida suoraan pelimoottorissakin. Kuten on käynyt ilmi, hahmon rigin optimaalinen kontrollointi on mahdollonta ilman hahmon pilkkomista osiin. Hahmon raajat ja oikeastaan kaikki liikuttavat (eli luihin kiinnittyvät) osat tulee siis pilkkoa omiksi paloikseen. Tässä esimerkissä hyödynnän jo tuttua jääkaappi-hahmoa, jonka olen luonut Photoshopilla. Hahmolle on tarkoitus luoda juoksusykli, ja koska hahmo liikkuu vain eteenpäin eli yhteen suuntaan, on se nopeinta ja kätevintä animoida luurankotekniikalla. Koska haluan animoida kaikkia hahmon osia erikseen, täytyy osien olla omilla layereillaan. Hahmon voi tuoda Unityyn .psb-tiedostona. Kuvasta 14 nähdään, kuinka Unity tunnistaa omilla layereillaan olevat hahmon osat. Ne näkyvät *sprite editorissa* eroteltuina spriteinä. Photosopissa luodut layerit tulevat Unityyn myös oikeassa järjestyksessä niin, että esimerkiksi silmä-spritet pysyvät pää-spritin edessä. Järjestystä voi toki vielä muuttaa Unityssa.



Kuva 14. Unityyn tuotu .psb-tiedosto hahmosta *sprite editorissa*, jossa Unity erottelee omilla tasoillaan olleet hahmon osat.

Luurangon luominen hahmolle onnistuu *skinning editorissa create bone* -toiminnolla. Skinning tarkoittaa spritejen kiinnittämistä luihin. Luut on tärkeää nimetä johdonmukaisesti, sillä se helpottaa työskentelyä jatkossa. Luurangon luomisen jälkeen hahmolle luodaan geometria, mikä onnistuu nopeasti *auto geometry* -toiminnolla. Geometrian luomisen tuloksena jokaisella spritellä on nyt muokattava mesh, josta kerroin aiemmin. Hahmossa näkyy myös erilaisia värialueita, jotka kertovat luiden vaikutusalueesta kuhunkin spriteen ja meshiin. Vaikutusalueen laajuuden voi nähdä spriteissä olevista väreistä: vaikuttavien luiden värit ”valuvat” spriteihin (ks. kuva 15). Koska vaikutusalueet ovat liian laajat, täytyy niitä muokata hieman *bone influence* -toiminnolla. Tavoitteena on, että kukin luu vaikuttaa vain oman spritensä meshiin eli esimerkiksi olkavarren luu vaikuttaa vain olkavarteen ja niin edelleen. Muutoin luiden liikuttelu johtaa epämuodostumiin.



Kuva 15. Vasemmalla nähdään luiden vaikutusalueet ennen muokkausta ja oikealla luiden korjatut vaikutusalueet.

Vaikka hahmo on rigattu, ei sen animoiminen onnistu vielä kätevästi. On rakennettava käänteinen kinematiikka, ja se onnistuu *IK Manager 2D* -toiminnon avulla luomalla hahmolle *limb solvereita*. Tämän komponentin avulla voidaan luoda raajojen luuketjujen päihin *effectorit*, joiden avulla raajoja liikutellaan (ks. kuva 16).



Kuva 16. Unityssä luodut *limb solverit* ja niitä määrittävät *effectorit* kullekin raajalle.

Jos nivel ei kuitenkaan taivu oikeaan suuntaan, *solverin* asetuksista on syytä laittaa *flip*-toiminto päälle. Kun hahmon rigi on luotu ja *effectorit* toimivat moitteettomasti, voidaan hahmoa alkaa animoida. Tällä tavoin toteutettu hahmo animoidaan Unityssä keyframien avulla. Kerron animaation toistotavoista seuraavassa alaluvussa.

4.2 Animaation toisto pelimoottorissa

Animaation toistotapoja on erilaisia riippuen käytetystä animaatiotekniikasta. Käsittelen tässä alaluvussa sprite sheet -animaatiota, keyframe-animaatiota ja kuvajonoja. Lisäksi kerron animaatioloopeista, jotka ovat olennainen osa pelihahmojen toimintaa.

Sprite sheet -animaatio

Sprite-animaatioita voidaan luoda kahdella tavalla, joko sprite sheetiin perustuen tai sitten keyframe-animaationa. Käsittelen näistä ensin sprite sheet -animaatiota ja seuraavassa kappaleessa keyframe-animaatiota. Sprite sheet -animaatio perustuu luonnollisesti jo tuttuun sprite sheetiin, joka sisältää kussakin animaatioissa toistettavat peräkkäiset frameet. Sprite sheet voi sisältää esimerkiksi hahmon kävelysyklin, eli jokainen kävelysyklänimaatioissa tarvittava frame on pakattuna yhteen kuvaan, yleensä png-tiedostoon. Kuvassa 17 voidaan nähdä esimerkki sprite sheetille asetellusta hahmon yksinkertaisesta kävelysyklisestä. Olen luonut hahmon kävelysyklin frame by frame -tekniikalla Resprite-ohjelmalla, joka on pikseligrafiikan luomiseen käytetty ohjelma. Unityyn tuotu sprite sheet pilkotaan osiin käyttämällä *slice*-työkalua, jotta pelimoottori tunnistaa yksittäiset frameet. Tällä tavoin aseteltu sprite sheet on kätevintä pilkkoa gridin avulla, jolloin kukin sprite varaa yhtä suuren alueen sprite sheetillä. Asettelussa on tärkeää, että spritet on aseteltu tasaisin välein toisiinsa nähden sujuvan liikkeen saavuttamiseksi. Sprite sheetin luominen animaatiosta onnistuu useilla digitaalisilla piirto-ohjelmilla automaattisesti, jolloin spritejä ei tarvitse manuaalisesti asetella. Sprite sheetissä olevat kuvat toistetaan peräkkäin järjestyksessä animaatioklippinä Unityssä. (Unity Learn 2022.)



Kuva 17. Hahmon kävelysyklin sprite sheet Unityn *sprite editorissa* pilkottuna slice-työkalun toiminnolla "grid by cell count (5)".

Keyframe-animaatio

Toinen sprite-animaation toistotapa on keyframe-animaatio. Tässä voidaan hyödyntää sprite sheetin sijasta *sprite atlasia*. Sprite atlas on myös kuvatiedosto, joka sisältää spritejä, mutta ne eivät ole peräkkäin toistettavia frameja. Sen sijaan sprite atlas voi sisältää esimerkiksi useamman hahmon osat, kuten päät, vartalot ja raajat sekä erilaiset lisävarusteet. Sprite atlaseseen voidaan säilöä myös ei-animoitavia objekteja. Sprite atlas voi parantaa pelimoottorin suorituskykyä vähentämällä muistin kuormitusta ja nopeuttamalla latausaikoja. Keyframe-animatiossa voidaan tekniikkoina hyödyntää pala-animaatiota ja luurankoanimaatiota. Esimerkiksi aiemmin esittelemäni rigattu jääkaappi-hahmo voidaan animoida kävelemään asettelemalla hahmon raajoja sopiviin kohtiin ja tallentamalla kuhunkin keyframeen tehty muutos. Näiden keyframien eli avainkehysten perusteella pelimoottori laskee hahmolle tarvittavat välikehukset, ja hahmo saadaan liikkumaan. (Unity Learn 2022; Bleeding Edge Studio 2023.) Automaattinen tweening ei ole mahdollista sprite sheet -animaatiossa, jossa sekä avain- että välikehukset tulee piirtää valmiiksi.

Kuvajono

Joskus hahmoanimaatio toteutetaan kuvajonona eli kuvasekvenssinä, mutta tämä on harvinaisempaa. Yleensä kuvissa suositetaan png-tiedostomuotoa. Tämä eroaa sprite sheet -animaatiosta siten, että kuvat säilötään pelimoottorissa erillisinä tiedostoina sen sijaan, että ne olisi pakattu samaan tiedostoon. Samalla periaatteella kuitenkin kuvien peräkkäinen toisto luo illuusion liikkeestä. Unityssa kuvajonolle voidaan asettaa sopiva kehysnopeus animaation toistoa varten. Useampien kuvajonojen ja etenkin korkearesoluutioisten kuvien käyttö pelimoottorissa syö muistia ja voi hidastaa pelin toimintaa verrattuna aiemmin mainittujen sprite sheetien tai sprite atlaksien käyttöön, sillä jokainen kuva täytyy ladata yksittäin. Etenkin hahmoanimaatioissa huomattavasti kätevää on hyödyntää edellä mainittuja keinoja, jolloin myös tarvittavat muokkaukset hahmolle voidaan tehdä yhdestä paikasta.

Animaatioloopit

Peleissä animaatioloopit ovat olennainen osa pelihahmon toimintaa. Animaatioloopin avulla hahmon jotakin tiettyä liikettä voidaan toistaa loputtomasti ja liike pysyy saumattomana – ensimmäistä framea ei siis voida selvästi osoittaa. Perinteinen animaatiolooppi on kävelysykli. Kävelysyklin looppaaminen tapahtuu frame by frame -animaatiossa niin, että viimeisen framen liike on animaatioloopissa ensimmäistä framea edeltävä liike. Jos halutaan loopata rigattu hahmo, tulee samoin viimeisessä keyframessa olevan asennon edeltää ensimmäisessä keyframessa olevaa asentoa. Jos viimeinen asento on täysin sama kuin ensimmäinen asento, voi liike tuntua töksähtävältä, sillä nyt sama asento toistetaan kahden peräkkäisen framen tai keyframen ajan. Looppi on yleensä onnistunut, jos siinä ei ole yhtäkään samanlaista framea. (Rokoko 2022.)

Pelihahmon liikkeet koostuvat käytännössä tällaisista yhteen blendatuista animaatioloopeista. Blendaamisella tarkoitetaan kahden animaatioloopin, esimerkiksi kävelyn ja hypyn yhdistämistä niin, että siirtymä näyttää sulavalta. Siirtymä voi olla automaattinen, jolloin ensimmäisen animaation loppu menee hieman

limittäin seuraavan animaation alun kanssa, tai sille voidaan luoda ihan erikseen oma animaationsa. (Rokoko 2022.) Erillisen siirtymäanimaation tekeminen ei liene tarpeellista, jos blendattavien animaatioiden liikkeet ovat lähellä toisiinsa, kuten käveleminen ja juokseminen. Sen sijaan monimutkaisemmasta tai hyvin erilaisesta liikkeestä siirtyminen toiseen voi vaatia erillisen siirtymäanimaation tekemistä.

5 Tekniikoiden vertailua

Vaikka tekniikoita on osin mahdotonta eriyttää toisistaan, voidaan niiden ominaisuuksia vertailla keskenään. Tekniikoiden eroavaisuuksien, vahvuuksien ja rajoitteiden vertailu voi auttaa animaattoria sopivimman tekniikan valinnassa. Luonnollisesti myös muiden, aiemmin mainittujen seikkojen huomioiminen vaikuttaa tekniikan valintaan. Tekemäni vertailun ja havaintojen perusteella olen koonnut animaatiotekniikoiden vahvuuksia ja rajoitteita taulukkoon 1, josta tekniikoiden eroavaisuudet ovat nopeasti nähtävillä.

Frame by frame -tekniikan etuna on ehdottomasti sen uniikkisuus, sillä periaatteessa kaikki on hahmoanimaation puolesta mahdollista. Niin monimutkainen animaatio on mahdollista kuin tekijä vain kykenee piirtämään. Rajoittavana tekijänä toimii lähinnä aika. Lisäksi hyvin korkearesoluutioiset kuvat voivat hidastaa pelimoottorin toimintaa etenkin, jos kuvia on paljon. Sprite sheetit nopeuttavat frame by frame -animaation kuvien lataamista pelimoottorissa, mutta silti suhteettoman suurten sprite sheetien käyttö ei ole optimaalista. Pelin toiminnallisuuden kannalta on myös olennaista, että hahmoja ei ”ylianimoida”, mikä saattaa helposti käydä frame by frame -tekniikalla. Jälkeenpäin on kuitenkin aina mahdollista poistaa turhia frameja. Hitaammat tekniikat saattavat vaatia enemmän budjettia, kun animaation tekemiseen kuluu paljon aikaa. Etenkin stop motion -tekniikalla toteutettavat pelit vaativat hurjasti aikaa. Frame by frame -animaation luomiseen on kuitenkin paljon ilmaisia ohjelmia, sillä ohjelmalta ei vaadita monimutkaisempia rigin rakenteluun tarvittavia ominaisuuksia.

Taulukko 1. Animaatiotekniikoiden havainnoidut vahvuudet ja rajoitteet koottuna taulukkoon. Pala- ja luurankoanimaatiota käsitellään taulukossa saman sarakkeen alla, sillä niissä on niin paljon samanlaisia ominaisuuksia ja käytännössä ne usein sekoittuvat.

	Frame by frame	Pala- ja luurankoanimaatio
Vahvuuksia	<ul style="list-style-type: none"> - uniikkisuus - ilmaisia ja edullisia ohjelmia - monipuolisuus 	<ul style="list-style-type: none"> - ajankäytöllisesti tehokkaampia - rigin ja hahmon osien uudelleenkäyttö - keyframe-animaatio ja tweening
Rajoitteita	<ul style="list-style-type: none"> - tekniikkana hyvin hidas - iso määrä korkearesoluutisia kuvia pelimoottorissa ongelmallista 	<ul style="list-style-type: none"> - hahmon asentojen rajallisuus - mekaanisuus - ohjelmat voivat olla kalliita - rigin rakentaminen vaatii opettelua

Pala-animaatio ja luurankoanimaatio kulkevat usein käsi kädessä, joten käsittelem niitä tässä kappaleessa yhdessä. Tekniikoiden etuna ehdottomasti se, että hahmon osia eli paloja voidaan vaihdella ja uudelleen käyttää luoden monipuolisempia hahmoja tai täysin uusia hahmoja. Animointi ilman rigiä voi monimutkaisempien hahmojen kohdalla kuitenkin olla hidasta. Kuitenkin keyframe-animointi ja tweening mahdollistavat nopeamman työkulun kuin frame by frame -tekniikassa. Luurankoanimaation etuna on rigin uudelleenkäyttö, mikä säästää hurjasti aikaa animaatioprosessissa, kun useimmat hahmot voivat hyödyntää jo olemassa olevia animaatioita. Tämä ei tietenkään toimi täysin erilaisten hahmojen kesken – esimerkiksi humanoidin hahmon luurankoa tuskin on optimaalista hyödyntää anatomisesti erilaiselle hahmolle. Hahmon hiukset tai vaatteiden kankaat saadaan myös animoitumaan luiden ja meshin ansiosta nopeammin verrattuna siihen, että ne pitäisi piirtää kuva kovalta aina uudelleen. Sekä luu- että pala-animaatiossa on kuitenkin kiinnitettävä erityistä huomiota hahmon luomiseen, sillä palasten täytyy toimia yhteen hahmoa animoitaessa. On siis jo alussa tärkeää huomioida se, miltä esimerkiksi hahmon olkavarsi näyttää eri asennoissa. Vaikka rigin avulla animointi saattaa olla nopeaa, on tekniikalla myös heikkoutensa. Ensinnäkin monipuolista rigaamista tarjoavat animointiohjelmat eivät useinkaan ole ilmaisia. Kaksiulotteisen hahmon rigi on myös

loppujen lopuksi hyvin rajallinen, eivätkä esimerkiksi kuvakulman muutokset onnistu helposti. Lisäksi animaation lopputulos voi olla mekaanisen näköinen. Pelistä riippuen tämä voi olla jopa haluttu ominaisuus, mutta yleensä se haittaa hahmoanimaation uskottavuutta. Animaation peruseriaatteita on tärkeää noudattaa myös luurankoanimaation tekemisessä. Hahmon meshien muokkaamisella ja *sprite swappingilla* saadaan animaatioihin kuitenkin lisää elävyyttä.

6 Yhteenveto

Opinnäytetyöni tarkoituksena oli selvittää eri animaatiotekniikoiden käyttötarkoituksia, vahvuuksia ja rajoitteita sekä löytää perusteluja eri tekniikoiden käytölle. Tavoitteena oli myös luonnollisesti oppia itse aiheesta lisää, ja ammatillisesti kaipasin selkeyttä tekniikoiden jäsentelyyn ja erittelyyn. Koen tarjonneeni sekä tietoa että kelvollista tekniikoiden vertailumateriaalia hahmoanimaation tekijöille, jotka voisivat saada tukea sopivimman animaatiotekniikan valintaan. Esimerkkeinä hyödynnettyjen pelihahmojen animaatioiden tarkastelu auttoi havainnoimaan eri tekniikoiden ominaisuuksia käytännössä. Käytännönläheisyyttä toivottavasti lisäsivät myös omat kokeiluni pelimoottorissa, joiden kautta pelimoottoriin liittyviä aiheita käsittelin.

Hahmon animaatiotekniikan valintaan vaikuttavat monet asiat, kuten pelialusta, haluttu visuaalinen tyyli ja pelin toiminnallisuus sekä luonnollisesti myös resurssit, kuten budjetti ja aikataulu. Havaintojeni perusteella voin todeta, että useat pelintekijät lähtivät suunnittelemaan pelihahmojen animaatiota pelin toiminnallisuuden pohjalta. Prioriteettina pidetään siis itse pelin funktionaalisuutta, jota animaatioiden tulee palvella. Hahmoanimaation tulee olla ytimekästä ja informoivaa pelin kannalta. Useat pelintekijät olivat myös loogisesti jo etukäteen päättäneet pelinsä visuaalisen tyylin, jonka perusteella hahmo tulitisiin animoimaan. Etenkin perinteisen piirroselokuvan tyyliä havittelevat tekijät päätyivät perinteisempään frame by frame -tekniikkaan. Realistisempaa liikettä hahmolle voidaan saavuttaa lisäksi rotoskooppaamalla, mutta tyylitellymmät ja yksinkertaisemmat hahmot – etenkin esimerkiksi pikselityyllillä luodut – eivät tästä niinkään hyödy.

Haasteelliseksi koin animaatiotekniikoiden jäsentelyn tarkoituksenmukaisesti. Etenkin pala- ja luuanimaatiossa on paljon yhtäläisyyksiä eikä niitä voida täysin käsitellä erikseen. Pysin kuitenkin tarjoamaan analyysimäisen esittelyn kunkin tekniikan tärkeimmistä ominaisuuksista, toteutustavoista ja niihin liittyvistä työkaluista. Opin itse aiheesta lisää opinnäytetyön kirjoittamisen lomassa ja erityisen tärkeänä huomiona pidän animaatiotekniikoiden yhdistämisen mahdollisuutta. Havaintojeni perusteella huomasin, että useamman tekniikan yhdistäminen toi monelle pelintekijälle haluttuja lopputuloksia. Hahmon yksinkertaisemmat animaatiot, kuten kävelysykli, voidaan nopeasti animoida rigin avulla, ja käyttää sitten enemmän aikaa monipuolisempien, esimerkiksi eri perspektiiviä vaativien liikkeiden luomiseen frame by frame -tekniikalla. Hahmoanimaation tekijän on hyvä olla tietoinen eri tekniikoiden käyttötarkoituksista, jotta sopivimman tekniikan valinta juuri omalle hahmolle onnistuu. Jatkossa olisi mielenkiintoista lisätä 3D:n mahdollisuudet animaatiotekniikoiden yhdistämisen näkökulmaan. Etenkin peliteollisuudessa 3D:n ja 2D:n yhdistäminen on tavallista, mutta henkilökohtaisesti haluaisin perehtyä tähän vielä lisää ja tutkia asiaa nimenomaan hahmoanimaation perspektiivistä.

Lähteet

Academy of Interactive Arts & Sciences 2021. Spiritfarer - Nicolas Guérin & William Dubé of Thunder Lotus Games - Game Maker's Notebook Podcast. Verkko-video 11.1.2021. Youtube. 1:03:15. <https://www.youtube.com/watch?v=SvszPTLzTPs> (viitattu 10.3.2024).

Adobe i.a. Stop motion animation explained: definition, types and techniques. Verkkosivu. <https://www.adobe.com/creativecloud/animation/discover/stop-motion-animation.html> (viitattu 18.3.2024).

Adobe 2023. Elements. Verkkosivu. <https://helpx.adobe.com/animate/using/elements.html> (viitattu 6.2.2024).

Ahoy 2014. Sprite supreme: a brief history of graphics, part two. Verkko-video 18.11.2014. Youtube. 9:09. <https://www.youtube.com/watch?v=a1yBP5t-fSA> (viitattu 3.3.2024).

Beresford, Trilby 2019. How Gaming Changed in the 2010s. Verkkosivu. Hollywoodreporter.com. <https://www.hollywoodreporter.com/lists/how-gaming-changed-2010s-1256603/introduction-of-twitch/> (viitattu 9.3.2024).

Blacksteinn 2022. History of game graphics. Sprites and a bunch of colors. Verkkosivu. LinkedIn. <https://www.linkedin.com/pulse/history-game-graphics-sprites-bunch-colors-blacksteinn> (viitattu 29.2.2024).

Bleeding Edge Studio 2023. Exploring the significance of the atlas tool in Unity game development. Verkkosivu. <https://www.linkedin.com/pulse/importance-atlas-unity-bleedingedgestudio/> (viitattu 10.4.2024).

Brown, Liza 2024. The Only Guide You'll Ever Need to Learn That Make 2D Rigging Clear to You. Verkkosivu. Filmora. <https://filmora.wondershare.com/animation-tips/2d-rigging.html> (viitattu 16.2.2024).

CGhero 2023. The difference between mobile and PC game development. Verkkosivu. <https://cghero.com/articles/difference-between-mobile-and-pc-development> (viitattu 21.3.2024).

Cohen, D.S. 2019. OXO aka Noughts and Crosses - The First Video Game. Verkkosivu. Lifewire. <https://www.lifewire.com/oxo-aka-noughts-and-crosses-729624> (viitattu 18.2.2024).

Dodsworth, Duncan 2023. Fourth generation video game consoles: the 16-bit generation. Verkkosivu. History-computer.com. <https://history-computer.com/fourth-generation-video-game-consoles-the-16-bit-generation/> (viitattu 3.3.2024).

Eden, Martin i.a. History of 3D games: a long time ago in a 2D space. Verkkosivu. Meliorgames.com. <https://meliorgames.com/game-development/history-of-3d-games-a-long-time-ago-in-a-2d-space/> (viitattu 3.3.2024).

Fleischer Studios i.a. 100 Years of Rotoscoping! Verkkosivu. <https://www.fleischerstudios.com/mrotoscope.html> (viitattu 27.2.2024).

Foxy Fern Animation 2021. Game animator reviews: the art & animation of Stardew Valley - Stardew Valley Animation. Verkkovideo 10.2.2021. Youtube. 7:16. <https://www.youtube.com/watch?v=hoMUf8wvnQM> (viitattu 26.2.2024).

Furniss, Maureen 2016. A new history of animation. New York: Thames & Hudson.

Gass, Zach 2020. The 10 Oldest Video Game Characters. Verkkosivu. Screenrant.com. <https://screenrant.com/oldest-video-game-characters-mega-man-link/> (viitattu 18.2.2024).

GDC 2016. Making fluid and powerful animations for 'Skullgirls'. Verkkovideo 26.1.2016. Youtube. 21:05. <https://www.youtube.com/watch?v=Mw0h9WmBlsw&t=6s> (viitattu 10.2.2024).

GDC 2017. 2D Animation at Klei Entertainment. Verkkovideo 2.6.2017. Youtube. 23:13. https://www.youtube.com/watch?v=8_KBjd0iaCU (viitattu 6.2.2024).

HardwareHeaven 2013. How Rayman Legends Is Made! Verkkovideo 1.8.2013. Youtube. 19:14. <https://www.youtube.com/watch?v=y-chi097uV4> (viitattu 21.3.2024).

Hartley, Edward 2021. Thunder Lotus Games on animating the afterlife in Spiritfarer. Verkkosivu. Toon Boom. <https://www.toonboom.com/thunder-lotus-games-on-animating-the-afterlife-in-spiritfarer> (viitattu 26.2.2024).

Helpshift 2023. What is an indie game and why is it so popular? Verkkosivu. <https://www.helpshift.com/what-is-an-indie-game-and-why-is-it-so-popular/> (viitattu 9.3.2024).

IMGA i.a.. Oddmar. Verkkosivu. Imgawards.com. <https://www.imgawards.com/games/oddmar/> (viitattu 21.3.2024).

Ip, Monica 2024. What is frame-by-frame animation? (And when to use it effectively). Verkkosivu. Creativa.com. <https://creativa.com.au/what-is-frame-by-frame-animation/> (viitattu 2.3.2024).

Jones, Angie & Oliff, Jamie 2008. Thinking animation: bridging the gap between 2D and CG. Boston: Course Technology.

Kevuru Games 2023. 2D games vs 3D games: key features, pricing, and top genres. Verkkosivu. <https://kevurugames.com/blog/differences-between-2d-games-vs-3d-games/> (viitattu 26.2.2024).

Klei i.a. About Mark of the Ninja: Remastered. Verkkosivu. <https://www.klei.com/games/mark-ninja> (viitattu 22.2.2024).

- Latson, Jennifer 2015. How Donkey Kong and Mario Changed the World. Verkkosivu. Time. <https://time.com/3901489/donkey-kong-anniversary/> (viitattu 18.2.2024).
- Layzelle, David 2021. History of the game engine: part 2 - the 1990s. Verkkosivu. Ultimategamingparadise.com. <https://ultimategamingparadise.com/features/series/history-of-the-game-engine/history-of-the-game-engine-part-2/> (viitattu 3.3.2025).
- Leong, Tim 2023. Inside the making of 'Karateka'. Verkkosivu. Activisionblizzard.com. <https://newsroom.activisionblizzard.com/p/inside-the-making-of-karateka> (viitattu 3.3.2024).
- Loew, Andreas 2023. How to create a sprite sheet. Codeandweb.com. Verkkosivu. <https://www.codeandweb.com/texturepacker/tutorials/how-to-create-a-sprite-sheet> (viitattu 10.3.2024).
- Maestri, George 2006. Digital character animation 3. E-kirja. Kalifornia: New Riders. (viitattu 17.2.2014).
- Melling, Emily 2019. 10 principles of character illustration. Verkkosivu. <https://yesimadesigner.com/10-principles-of-character-illustration/> (viitattu 10.4.2024).
- Moleman, Christian 2009. The necessity of interactive animation for games. Verkkosivu. Game Developer. <https://www.gamedeveloper.com/design/the-necessity-of-interactive-animation-for-games> (viitattu 10.2.2024).
- Porokh, Alena 2023. What is animation in video games: basic principles and applications. Verkkosivu. Kevuru Games. <https://kevurugames.com/blog/what-is-animation-in-video-games-basic-principles/> (viitattu 14.2.2024).
- Reeves, Ben 2018. How flash games changed video game history. Verkkosivu. Medium. <https://www.gameinformer.com/2018/12/22/how-flash-games-changed-video-game-history> (viitattu 26.2.2024).
- Rokoko 2022. Tutorial: how to loop animation & make the perfect loop. Verkkosivu. <https://www.rokoko.com/insights/loop-animation> (viitattu 18.3.2024).
- Sini, Victoria 2023. Memoirs of a gamer vol 18: june 1983 in games. Verkkosivu. LinkedIn. https://www.linkedin.com/pulse/memoirs-gamer-vol-18-june-1983-games-studiotypes?trk=public_post (viitattu 9.3.2024).
- Stardewvalley.net i.a. Press. Verkkosivu. <https://www.stardewvalley.net/press/> (viitattu 26.2.2024).
- Strafefox 2021a. The making of Aladdin - cel animation in video games. Verkko-video 19.2.2021. 23:03. Youtube. <https://www.youtube.com/watch?v=kOnUIT-JqRQQ&t=0s> (viitattu 3.3.2024).

Strafofox 2021b. The making of Donkey Kong Country & Killer Instinct - pre-rendered graphics. Verkkovideo 30.5.2021. Youtube. 24:57. <https://www.youtube.com/watch?v=LPU7gNzWUE&t=0s> (viitattu 3.3.2024).

Strafofox 2022. The Making of Mortal Kombat - digitized graphics in video games. Verkkovideo 4.1.2022. Youtube. 27:59. <https://www.youtube.com/watch?v=226Nwq2CN5c> (viitattu 3.3.2024).

ThatGuyGlen 2020. How hollow knight was made and first appeared on Newgrounds. Verkkovideo 19.7.2020. Youtube. 22:56. <https://www.youtube.com/watch?v=kSAIVBFdC6k> (viitattu 2.3.2024).

Toon Boom Animation 2023. Harmony 22 Gaming - export to game engine and integration with Unity. Verkkovideo 4.4.2023. Youtube. 5:12. <https://www.youtube.com/watch?v=uVflE36yhal> (viitattu 18.3.2023).

Toon Boom i.a. Cut-out animation. Verkkosivu. <https://learn.toonboom.com/modules/animation-workflow/topic/cut-out-animation> (viitattu 18.2.2024).

Unity Documentation 2024. Unity manual: animation. Verkkosivu. <https://docs.unity3d.com/Manual/AnimationSection.html> (viitattu 18.3.2024).

Unity i.a. Cuphead by StudioMDHR. Verkkosivu. <https://unity.com/made-with/cuphead> (viitattu 2.3.2024).

Unity Learn 2022. Introduction to Sprite Animations. Verkkosivu. Learn.unity.com. <https://learn.unity.com/tutorial/introduction-to-sprite-animations#> (viitattu 25.3.2024).

Verma, Mayank 2023. Pushing pixels: exploring the world of pixel art in modern design. LinkedIn. Verkkosivu. <https://www.linkedin.com/pulse/pushing-pixels-exploring-world-pixel-art-modern-design-mayank-verma> (viitattu 10.2.2024).

Video Game Animation Study 2019. The animation of Hollow Knight. Verkkovideo 18.9.2019. Youtube. 11:35. <https://www.youtube.com/watch?v=VeN7dHLipkY> (viitattu 2.3.2024).

WayForward 2023. Lunark - Rotoscoping & Game Design (Featurette #2) Verkkovideo 16.3.2024. Youtube. 3:33. <https://www.youtube.com/watch?v=nX7yWiwGKCo> (viitattu 1.3.2024).

Webster, Andrew 2016. 30 years of Zelda: a timeline of the legend so far. Verkkosivu. Theverge.com. <https://www.theverge.com/2016/2/21/11063982/zelda-30th-anniversary-nintendo-history-link-smash-bros>

Wikipedia 2024a. Frame rate. Verkkosivu 10.2.2024. https://en.wikipedia.org/w/index.php?title=Frame_rate&oldid=1205895959 (viitattu 2.3.2024).

Wikipedia 2024b. Adobe Animate. Verkkosivu 19.2.2024. https://en.wikipedia.org/w/index.php?title=Adobe_Animate&oldid=1208917829 (viitattu 23.2.2024).

Wikipedia 2024c. Game engine. Verkkosivu 9.3.2024. https://en.wikipedia.org/w/index.php?title=Game_engine&oldid=1212737568 (viitattu 9.3.2024).

Williams, Richard 2009. The animator's survival kit: a manual of methods, principles and formulas for classical, computer, games, stop motion and internet animators. Lontoo: Faber & Faber.

Wirtz, Bryan 2023. The power of experience: The wonders of video game immersion. Verkkosivu. Gamedesigning.org. <https://www.gamedesigning.org/learn/game-immersion/> (viitattu 18.2.2024).

Wulverblade 2014. Frame vs. mesh animation. <https://wulverblade.com/frame-vs-mesh-animations/> (viitattu 10.3.2024).

Kuvalähteet

Kuva 1. Classicgaming i.a. The history of Donkey Kong. Verkkosivu. <https://www.classicgaming.cc/classics/donkey-kong/history> (haettu 10.4.2024).

Kuva 2. Insert Coin 2020. Bega's Battle Arcade. Verkkovideo 2.11.2020. Youtube. 19:49. <https://www.youtube.com/watch?v=6hv5WU8U7A> (ruutukaappaus 10.4.2024).

Kuva 3. Kuvakooste. Kuvat vasemmalta oikealle.

- Eisen, Andrew ym. 2018. The Legend of Zelda guide. Ign.com. Verkkosivu. <https://www.ign.com/wikis/the-legend-of-zelda/Walkthrough> (haettu 10.4.2024).
- Parkin, Jeffrey 2023. Where to find the Archaic Tunic in Zelda: Tears of the Kingdom. Polygon.com. Verkkosivu. <https://www.polygon.com/zelda-tears-of-the-kingdom-guide/23721453/archaic-tunic-location> (haettu 10.4.2024).

Kuva 4. Virgin Games USA 1992. Disney's Aladdin. (Ruutukaappaukset pelistä 20.3.2024).

Kuva 5. Strafefox 2021b. The making of Donkey Kong Country & Killer Instinct – pre-rendered graphics. Verkkovideo 30.5.2021. Youtube. 24:57. <https://www.youtube.com/watch?v=LPU7gNzWUE&t=0s> (ruutukaappaukset 25.3.2024)

Kuva 6. ProsafiaGaming 2020. Cuphead – full game walkthrough. Verkkovideo 20.1.2020. Youtube. 1:23:16. <https://www.youtube.com/watch?v=QF9tzn7UUlo> (ruutukaappaus 10.4.2024).

Kuva 7. Video Game Animation Study 2019. The animation of Hollow Knight. Verkkovideo 18.9.2019. Youtube. 11:35. <https://www.youtube.com/watch?v=VeN7dHLipkY> (ruutukaappaukset 10.3.2014).

Kuva 8. Fleischer Studios i.a. 100 Years of Rotoscoping! Verkkosivu. <https://www.fleischerstudios.com/mrotoscope.html> (haettu 27.2.2024).

Kuva 9. ConcernedApe 2016. Stardew Valley. (Ruutukaappaus pelistä 10.3.2024).

Kuva 10. Animation at Klei Entertainment. Verkkovideo 2.6.2017. Youtube. 23:13. https://www.youtube.com/watch?v=8_KBjd0iaCU (ruutukaappaus 6.2.2024).

Kuva 11. HardwareHeaven 2013. How Rayman Legends Is Made! Verkkovideo 1.8.2013. Youtube. 19:14. <https://www.youtube.com/watch?v=y-chi097uV4> (kuvankaappaus 21.3.2024).

Kuvat 12-17. Varrio, Veera 2024.