

Teemu Rasinkangas

2D-peligrafiikka

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mediatekniikan koulutusohjelma

Insinöörityö

11.4.2014

Tekijä Otsikko	Teemu Rasinkangas 2D-peligrafiikka
Sivumäärä Aika	56 sivua 11.4.2014
Tutkinto	insinööri (AMK)
Koulutusohjelma	mediatekniikka
Suuntautumisvaihtoehto	digitaalinen media
Ohjaaja	yliopettaja Harri Airaksinen
<p>Insinööriyön tarkoitus oli tutkia grafiikan tuottamista valmiiseen 2D-peliin. Peligrafiikan tekemisessä oli otettava huomioon monia teknisiä ja rajoittavia tekijöitä, joita ilmenee peli-projektissa. Insinööriyössä perehdyttiin muun muassa ajan optimointiin, referenssin hyödyntämiseen ja pelin osa-alueiden, kuten hahmojen tai pelikentän, suunnitteluun ja toteuttamiseen.</p> <p>Projektissa todettiin, että monet kasuaali- ja mobiilipelit ovat nykyään 2D:tä. Laitevaatimukset ja trendit olivat vaikuttaneet nykypäivän 2D-pelin ilmeeseen, mutta vanhoja retropelejä ja kasuaalipelejä ei ole unohdettu. 2D-pelejä on helppo tehdä, koska ne eivät vaadi monimutkaista tuotantoprosessia, niin kuin 3D-pelin tuottaminen. 2D-pelit voivat olla myös vahvasti sidoksissa niiden taiteelliseen tyyliin, ja ne vetoavat pelaajiin persoonallisella grafiikalla. Niiden ei kuitenkaan tarvitse olla riippuvaisia graafisesta ilmeestä, ja ne voivat painottaa käyttöliittymän toiminnallisuuteen, kuten esimerkiksi managerointipeleissä.</p> <p>Insinööriyöprojektissa tehtiin grafiikka Toy Party -peliin, joka sisälsi useamman minipelin. Peli vetosi leikki-ikäisiin lapsiin pirteillä hahmoilla ja sopivan haastavilla pelimekaniikoilla. Toteutus oli aloitettava projektin määrittelyllä ja suunnittelulla, koska projekti alkoi ilman graafista ohjeistusta. Toy Partyn minipelit olivat yksinkertaisia ja viihdyttäviä kasuaalipelejä. Pelien tyyliä ohjasivat päätökset graafisen projektion ja ohjelmiston valinnan suhteen, joten niiden määrittely alussa oli keskeinen osa pelin esituotantoa. Grafiikan toteutusta helpottivat monet työkalut ja tekniikat, kuten onion skinning -tekniikan imitoiminen animoinnissa. Vaikka pelisuunnittelu oli jakautunut usealle henkilölle, aikarajat ja tehtävät pysyivät hallussa ja lopputuloksena oli onnistunut peli, jota ei kuitenkaan julkaistu.</p>	
Avainsanat	2D-pelit, peligrafiikka, pelitaide

Author Title	Teemu Rasinkangas 2D game graphics
Number of Pages Date	56 pages 11 April 2014
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Digital Media
Instructor	Harri Airaksinen, Principal Lecturer
<p>The purpose of this thesis was to study the making of 2D game graphics from design to a finished product. Making graphics for a game was not the same thing as just painting pretty pictures. There was a lot to know about the technical side of implementing graphics in to a complete game. This thesis will go through matters like optimizing time, reference usage, character designing and making of menus.</p> <p>Graphics were made for a game called Toy Party in part with this thesis. Toy Party was funded by the Game Cluster -project. The game development ended with a successful game collecting six small games with different mechanics. The game targeted kids as audience which meant that the graphics were compromised from harmonic color schemes to a more playful appearance.</p> <p>Even though modern games have developed a lot, 2D retro games and casual games haven't been forgotten. The six small games of Toy Party were also simple and casual. The style of the game was driven by the choices we made concerning the software and graphical projection. There was a swarm of tools and techniques available out there to make the game development easier.</p>	
Keywords	2D games, game art, game graphics

Sisällys

Sanasto

1	Johdanto	1
2	2D-pelit	2
2.1	2D-pelit nykypäivänä	2
2.2	Pikselit, vektorit ja vokselit	4
2.3	Graafiset projektiot	6
3	Pelituotannon kulku	13
3.1	Esituotanto	13
3.2	Työnkulun optimointi	15
4	Graafikon työkalut	17
4.1	Laitteisto	17
4.2	Ohjelmistot	19
5	Grafiikan tuottaminen	22
5.1	Graafinen ohjeistus	22
5.2	Referenssi	22
5.3	Hahmon luonti	24
5.4	Kenttä	27
5.5	Graafinen käyttöliittymä	29
5.6	Animaatio	29
6	Toy Party	34
6.1	Toy Party -pelit	36
6.2	Toteutus	42
6.3	Ongelmat	47
6.4	Työnkulku ja aikataulu	51
7	Yhteenveto	52
	Lähteet	53

1 Johdanto

Onko 3D syrjäyttänyt 2D:n kokonaan? Ovatko kamera-ajot pelaajalle niin tärkeitä, että jokaisen pelin pitää olla kolmiulotteinen? Jotkut voivat pitää enemmän 2D-peleistä ja jotkut taas 3D-peleistä, mutta täytyykö niiden välillä olla ikuinen taistelu? Tai edes eroa? Nykyisten pelimoottorien tekniikan puolesta eroa ei ainakaan tarvitse olla, koska usein 2D-pelit tehdään 3D-pelimoottorilla. 3D:llä ja 2D:llä on omat hyvät ja huonot piirteensä, ja siksi niitä on hyvä jopa sekoittaa keskenään. Vaikka 3D-pelit ovat pelimaailman kärjessä edelleen kehittyvänä, 2D-pelit kiinnostavat vielä monia niiden yksinkertaisuuden ja esteettisyyden ansiosta. Ne ovat myös löytäneet oman paikkansa mobiili- ja indie-markkinoilla.

Varsinaisia ohjeita, miten tehdä täydellinen peli, ei ole. Monien taiteilijoiden opetusmateriaalit perustuvat asioihin, joita he ovat harrastelijana tai ammattuurallaan oppineet. Nämä taidot nopeuttavat työntekoa, mutta ne eivät ole ainoa tapa tehdä pelejä, varsinkin kun tyylejä on monia. Pienessä peliprojektissa artisti voi vaikuttaa hyvinkin paljon pelin visuaaliseen suunnitteluun, kun taas suuremmissa tuotannoissa työnkuva suppeenee erikoisosaamisalueisiin.

Insinöörityöprojektina tehdään Toy Party -niminen peli. Projekti toteutetaan osana Game Cluster -hanketta, joka auttaa tuomaan esille Metropolia Ammattikorkeakoulun pelituotantoa. Insinöörityössä tutkitaan myös vakiintuneita työnkulun menetelmiä ja niihin liittyviä teknisyyksiä, joita peligrafiikan tuottamiseen vaaditaan. Esillä on 2,5D-pelejä, koska ne ovat usein hyvin lähellä 2D-pelejä. Monia insinöörityön asioita voidaan nähdä myös 3D-pelin tuotannossa. Näkökantana on usein peligraafikon ja pelisuunnittelijan näkökulma.

2 2D-pelit

2.1 2D-pelit nykypäivänä

Kaksiulotteisia pelejä voidaan pitää esteettisempinä ja paikoitellen realistisempina kuin kolmiulotteisia pelejä, koska 2D-grafiikka on usein maalattua. 3D:ssä kamerakulmien vaihtaminen ei ole ongelma, toisin kuin 2D:ssä, jossa se ja muiden kameran ominaisuuksien muuttaminen vaatii koko piirroksen uudelleen tekemisen. Realistinen grafiikka kiinnostaa monia pelaajia, ja 3D täyttää heidän toiveitaan. 3D-pelit ovat nykypäivää, mutta vielä silloin tällöin markkinoille ilmestyy Rayman Legendsin kaltaisia 2D-pelejä. Rayman yhdistää vauhdikkaan pelattavuuden ihanteellisella grafiikalla ja samaistuttavilla hahmoilla [1]. Vaikka 3D ottaa kiinni koko ajan, 2D:n hyviä puolia ovat vielä kehityksen helppous, esteettisyys, vähäiset laitevaatimukset ja nostalgia.

2D:n ja 3D:n yhdistymistä erilaisilla tavoilla kuvaillaan 2,5D:nä. Termi on monipuolinen ja vaihtuva riippuen käyttökohteesta. Englanniksi 2,5D:tä voi kutsua myös pseudo 3D:ksi. Tällöin puhutaan kolmiulotteiselta vaikuttavasta pelistä, vaikka takana ovat 2D-työkalut. Peliharrastelijapiireissä 2,5D voi merkitä 3D-pelimootorilla tehtyä sivulta kuvattua tasohyppelyä. Jos pelin grafiikka koostuu kuvista eli puhtaasti 2D:stä mutta se on aseteltu kolmiulotteiseen avaruuteen eli xyz-koordinaatistoon, kuvailisin peliä 2,5D:ksi. 2,5D-peliksi käsittäisin myös pelin, jossa 3D-objektit on aseteltu xy-koordinaatistoon kameran liikkuesssa pelkästään sivu- ja pystysuunnassa. Toy Party -projekti, jota käsitelen luvussa 6, on 2,5D-peli, koska siinä käytetään paikoitellen 3D:tä, vaikka se on selvästi painottunut 2D-grafiikkaan.

Vaikka 3D-pelit ovat vallanneet pelimarkkinat, indie-pelejä tehdään vielä paljon 2D:nä. Termi indie tulee englannin kielen sanasta independent eli itsenäinen. Itsenäiset pelinkehittäjät ovat usein pienen ryhmän jäseniä tekemässä pieniä pelejä pienellä budjetilla. Koska riskejäkään ei ole paljoa indie-pelin kehityksessä, voidaan peleistä tehdä erikoisempia kuin AAA-pelit. AAA-pelit ovat suuren budjetin ja uuden sukupolven pelejä suurimmilla pelialustoilla. Kuvassa 1 on Incredipede, jossa on omalaatuinen visuaalinen ilme. Indie-pelin kehittäjät ovat monesti myös suuria pelifaneja, jotka arvostavat pelien nostalgia-arvoa tekemällä erikoisia moderneja retropelejä. [2.]



Kuva 1. Incredipede-peli [3].

2D-pelien yksinkertaisuus ja lähestyttävyyys tuo suosiota sosiaalisilla alustoilla ja mobiilipelikaupoissa. Niin sanotut kasuaalipelit pyrkivät tarjoamaan pelaajalle yksinkertaisen ja addiktoivan pelikokemuksen. Mobiilipelit vetoavat pelaajiin miellyttävällä pelikokemuksella ja persoonallisella grafiikalla. Ostamisen kynnyks voi olla matala, kun graafinen ilme miellyttää; mobiilipelithän eivät ole kuin muutaman euron hintaisia tai jopa ilmaisia. 2D-pelien kehittäminen on nykyään myös entistä helpompaa lukuisilla pelinkehitysohjelmistoilla. Koska työvoimaa ei tarvita 3D:n monimutkaiseen työprosessiin, voidaan 2D-mobiilipelin kehityksessä keskittyä markkinoimiseen, visuaaliseen ilmeeseen ja siihen, mitä kaikkien pelien pitäisi olla eli hauskuuteen. Tämän perusteella 2D-peleillä on vielä hyvät oltavat verrattuna 3D-peleihin varsinkin mobiili- ja kasuaalipelien markkinoilla. [4, luku 3.]

Facebook on lisännyt mobiilipelien suosiota. Nämä ”sosiaaliset pelit” ovat vetäneet puoleensa uusia pelaajia, joille pelaaminen on aivan uutta. Facebook-pelien pelaajien keskimääräinen ikä vuonna 2010 oli 43 [5]. Koska pelejä pelataan lyhyen aikaa, töiden tai keskustelun ohessa, käytettävyys nousee etusijalle. Käytettävyyden kannalta tärkeäksi nousee käyttöliittymä ja grafiikan selkeys mobiilipeleissä. Mobiililaitteet ovat kehittyneet paljon viime aikoina, ja toisin kuin aikaisemmin, niissä on sisäänrakennettu näyttönohjain, joka mahdollistaa raskaampien pelien pelaamisen. Mobiililaitteiden teho ja näytön koko ovat kuitenkin rajattuja. Pelejä on optimoitava sopiviksi koodin ja grafiikan puolesta. Optimointi auttaa pelin suorituskykyä ja tekee pelistä sopivan mobiililaitteille tai internetiin. Joidenkin tablettitietokoneiden tehot, kuten Microsoftin Surface Pron, voi

vastata koti PC:n tehoa. Älypuhelimet ovat hyvin lähellä tablettitietokoneiden tehoa. Tarkoittaako tämä sitä, että on vain ajan kysymys, milloin pelataan uuden sukupolven pelejä puhelimella? Voi olla, mutta se tuskin vaikuttaa 2D-pelien suosioon, koska 2D-pelit toimivat yksinkertaisuuden ansiosta paremmin pienillä näytöillä.

2.2 Pikselit, vektorit ja vokselit

2D-pelit ovat yleensä bittikarttagrafiikkaa tai vektorigrafiikkaa. Bittikartat ovat kuvia, jotka koostuvat pikseleistä. Näitä ovat tavalliset digitaaliset kuvat, kuten valokuvat tai maalaukset. Bittikarttakuvilla on aina pikseliulottuvuus, joka määrittää kuvan tarkkuuden. Vektorikuva on piirto-ohjelmalla tehty xy-koordinaatistossa sijaitsevien pisteiden, viivojen, käyrien tai muotojen kokoelma. Niiden näkyminen näytöllä ei riipu pikselien määrästä vaan suunnasta, joten ne voivat olla kuinka tarkkoja tahansa, toisin kuin bittikarttakuvat. Vektorit ovat yksinkertaisia ja tiedostokooltaan pieniä, joten ne soveltuvat hyvin mobiilipelien grafiikaksi. Ne eivät kuitenkaan ole yhtä realistisia kuin pikselikuvat. [4, luku 3.]

Vektorikuvatiedostot eivät ole niin hyvin tuettuja kuin pikselikuvat, ja niiden muokkaaminen on vaikeampaa. Toisaalta pelien graafinen käyttöliittymä on useasti tehty vektoreilla, koska niitä on helppo skaalata eri kokoon. Skaalautuvuus on tärkeää, kun peli tehdään usealle näytön kuvasuhteelle. Usein pelit on tehty kuitenkin kokonaan bittikarttoina käyttöliittymää myöten.

Vektoripelejä näkee usein flash-pelisivustoilla. Nimitys flash-peli tulee Adobe Flash -ohjelmasta, joka on saanut paljon suosiota. Flash-pelejä tehdään Flashilla, ja ne toimivat internetympäristössä yleistyneen Flash-selainlisäosan ansiosta. Tämän vuoksi monet harrastelijat ovat tarttuneet Flashiin. Flashin on pitkään epäilty häviävän HTML5:n kehittyessä, koska HTML5 ei tarvitse erillistä liitännäistä. Pelinkehityksen kannalta tällä hetkellä Flash on silti ylivoimaisesti yleisempi.

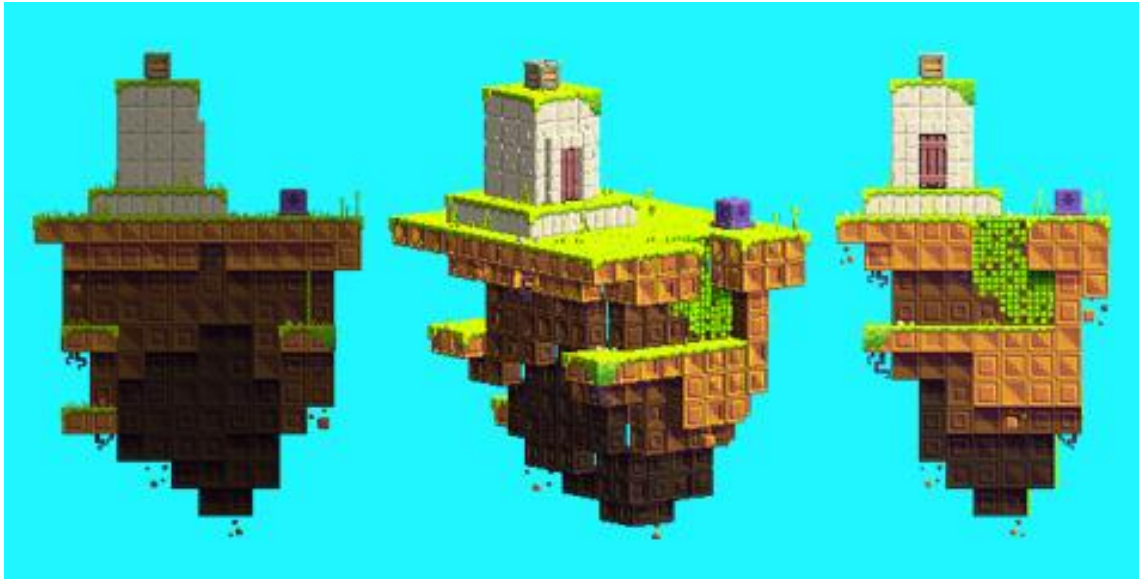
Pikseligrafiikka on tyylisuunta, joka ottaa vaikutteita vanhoista peleistä. Kuvan 2 peli on Superbrothers: Sword & Sworcery EP, joka on tehty retrolla pikseligrafiikalla. Siinä on yhdistetty visuaalisen ilmeen kauneutta ja nykypäiväistä tunnelmaa pikseligrafiikkaan. Pikseligrafiikassa pikselit ovat selvästi erotettavissa ja värit ovat rajattu. Modernit retro-pelit tehdään usein pikseligrafiikalla. Retropelejä tekevät nykyään melkein ainoastaan

itsenäiset pelinkehittäjät, jotka haluavat vaalia vanhoja 8- ja 16-bittisiä pelejä. Pikselipelien etuna on niiden toteutuksen helppous, soveltuvuus mobiilialustoille ja nostalgia-arvo.



Kuva 2. Superbrothers: Sword & Sworcery EP -peli [6].

Monet niin sanotut vokselipelit ovat tyyliltään retropelejä. Ne ovat teknisesti enemmän 3D:tä, mutta vokselien ulkonäkö saa ne näyttämään retrolta pikseligrafiikalta. Yksinkertaisesti selitettynä vokselit ovat pikseleitä 3D-avaruudessa. Ne ovat siis yksivärisiä laatikoita xyz-koordinaatistossa, aivan kuten pikselit ovat ruutuja xy-koordinaatistossa. Ne rakentuvat suhteessa toisiinsa ruudukossa toisin kuin polygonipinnat, jotka voivat olla mihin tahansa suuntaan missä tahansa.



Kuva 3. Vokseligrafiikan käyttö FEZ-pelissä [7].

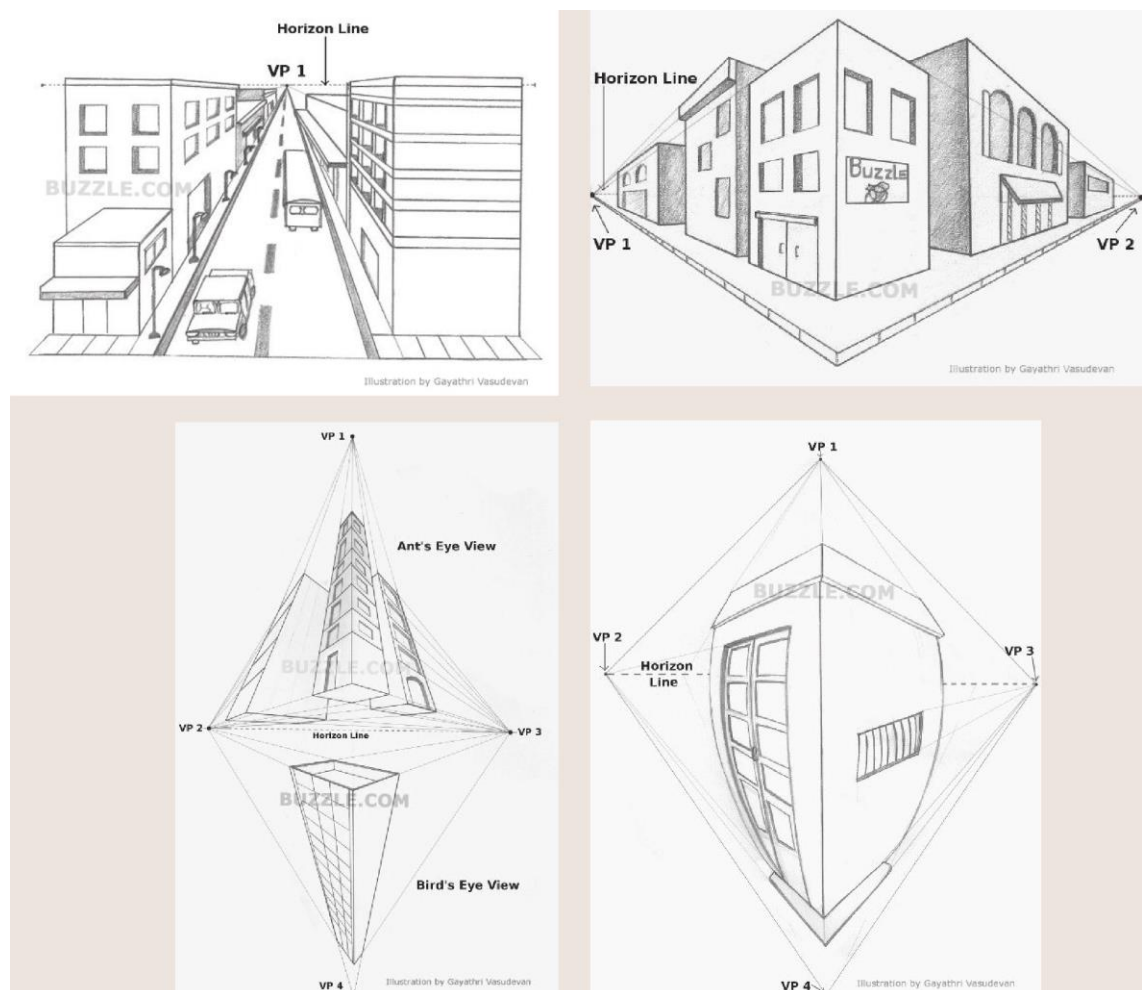
Kuvasta 3 näkee, kuinka kaksiulotteinen ilme on toteutettu vokselipeli Fezissä. Vokselit saavat sen näyttämään perinteiseltä pikseligrafiikalta, mikä on esteettiseltä kannalta arvo retropelaajalle. Fezin tapauksessa pikselimäisen grafiikan tekoon sovellettiin studion omaa muokattua vokselia eli trikseliä, joka mahdollistaa pelin omalaatuisen mekaniikan. [8.]

2.3 Graafiset projektiot

2D-pelit toimivat usein vain yhdellä kamerakulmalla. Kamerakulmia voi olla esimerkiksi ylhäältä, sivusta tai yläviistosta. Kamerakulman lisäksi pelin kameralle määritellään graafisen projektion tyyppi, joka määrittää sen, kuinka kolmiulotteiseksi kuviteltu maailma piirtyy kaksiulotteiselle ruudulle. Pelilajityyppi ja grafiikka ovat vahvasti sidoksissa kamerakulmaan ja graafiseen projektiioon. Niiden muuttaminen kesken pelituotannon voi aiheuttaa grafiikan uudelleen tekemisen tai muokkaamisen suurelta osin. Kun peliin aletaan sekoittaa useampaa kuvakulmaa, tai jopa useampaa graafista projektiota, joudutaan yleensä tukeutumaan 3D:hen. Tällöin alamme puhua jo 2,5D:stä. [9.]

Perspektiivinen projektio

Perspektiivisyys vaatii pakopisteet toimiakseen. Pakopisteitä on kuvassa tavallisesti yhdestä kolmeen, mutta pisteitä voi olla loputtomiin. Mitä enemmän pakopisteitä, sitä enemmän kuva näyttää kalansilmäperspektiivistä kuvatulta. Kuvassa 4 on neljä kuvaa, joissa on pakopisteitä yhdestä neljään. Yhden pisteen perspektiivillä on helppo piirtää esimerkiksi huone, koska vastakkaiset seinät ovat helposti nähtävillä. Kahden pisteen perspektiivillä voi tehdä mitä vain esineistä ympäristöihin. Kolmen pisteen perspektiiviä näkee harvoin 2D-peleissä, koska se ilmenee parhaiten silloin, kun katsotaan kauaksi ulottuvaa kohdetta sen edestä. Neljän tai useamman pisteen perspektiiviä ei 2D-peleissä ole luultavasti ollenkaan, koska se on epäluonnollisen näköinen ja hankala piirtää.



Kuva 4. 1-, 2-, 3- ja 4-pisteperspektiivit [10].

Vanhoissa flip3D-peleissä käytettiin yhden pisteen perspektiiviä tuomaan kolmiulotteisuuden tunnetta kaksikulotteiseen peliin [11]. Sama ajatus on pseudo-3D-ajopeleissä, kuten vanhemmissa arcade-ajopeleissä.

Kaksi- tai kolmepisteperspektiiviä näkee usein osoita ja klikkaa -peleissä (engl. point and click), joissa hahmon pitää pystyä liikkumaan syvyys suunnassa. Syvyys voi olla toteutettu xyz-koordinaatistossa tai huijatuksi xy-koordinaatistossa. Kuvitellaan, että peli on toteutettu vain 2D-grafiikalla. Hahmo on oma sprite-grafiikkansa erillään taustasta. Spritet ovat 2D-kuvia tai animaatiota, jotka ovat yleensä osa useampaa kuvaa kokoaavaa spritesheet-kuvatiedostoa. 3D-pelimootorissa hahmoa voi liikuttaa syvyys-, korkeus- ja leveys suunnassa taustagrafiikan ollessa paikallaan. Saman voi toteuttaa 2D-pelimootorissakin. Jos tausta on perspektiivissä, maataso häviää horisonttia kohden. Taustan alakulma näyttää olevan lähellä, ja mitä ylemmäs liikutaan, sitä syvemmäksi kuva vaikuttaa siirtyvän. Jos hahmoa siirtää ylöspäin taustan päällä, näyttää kuin hahmo siirtyisi horisonttia kohden. Jos kuvan perspektiivi on syvä, hahmoa voi joutua kutistamaan perspektiivin mukaan, kuten joutuisi kuvassa 5, jos kamera ei liikkuisi hahmon mukana.



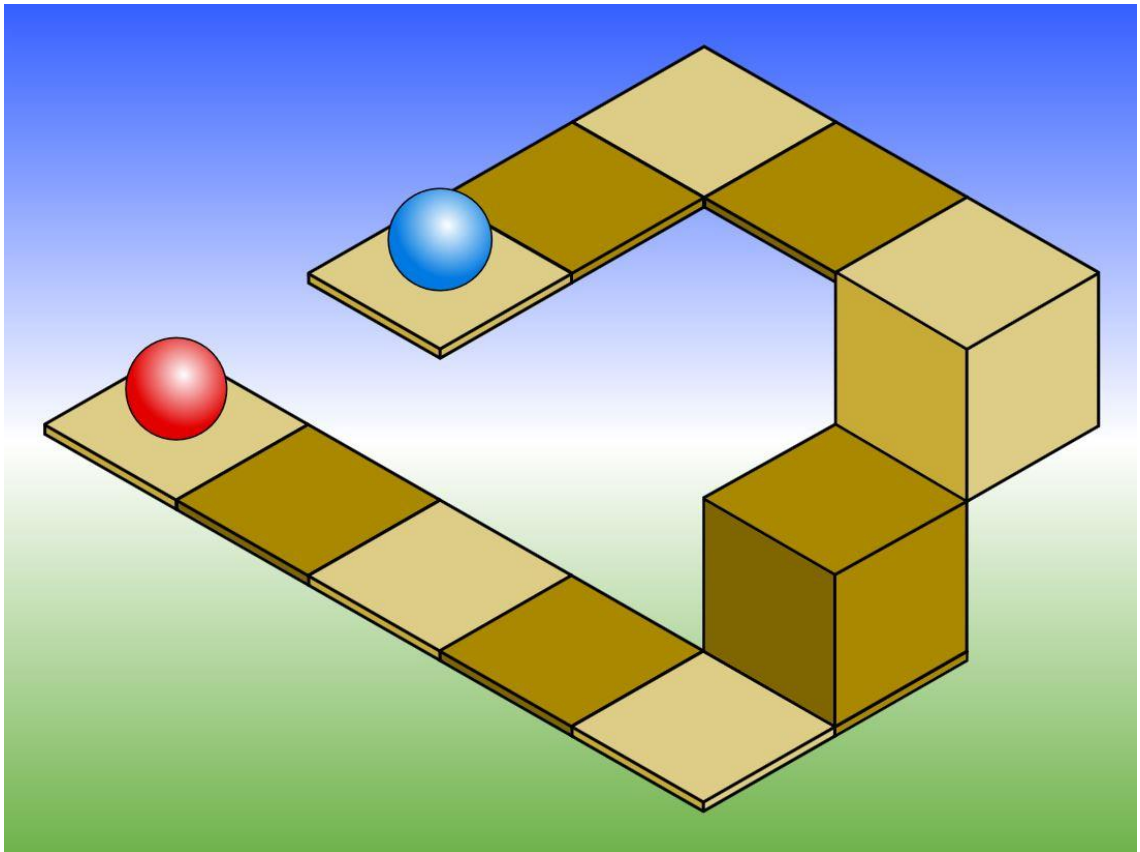
Kuva 5. Perspektiiviset virheet Don't Starve -pelissä [12].

Perspektiivisen 2D-pelin ongelma on pakopisteissä. Kaikkien peliobjektien kulmien on osoitettava oikeaan pakopisteeseen. Objektin koon muuttaminen tai sen liikuttaminen muuttaa pakopisteen paikkaa ja saa objektin näyttämään siltä, kuin se olisi vinossa. Super Marion kaltaisten sivulta kuvattujen pelien projektio on ortogonaalinen (tästä myöhemmin). Jos Super Marion grafiikka olisi tehty perspektiivissä, se antaisi luonnot-

toman vaikutelman kun tausta liikkuu, koska pakopisteet siirtyvät liikkeen mukana. Hyvän esimerkin teknisesti epäonnistuneesta, vaikka persoonallisesta perpektiivistä, tarjoaa peli nimeltä Don't starve (kuva 5). Pelin kenttä ja esineet ovat xyz-koordinaatistossa, vaikka ne on piirretty 2D:nä. Maataso, joka on samalla tausta, häviää pakopistettä kohti. Esineiden sijainti liikkuu siis myös pakopistettä kohti, ja ne pienenevät sen mukaan mitä kauempana ovat. Pakopisteet eivät muutu suhteessa esineeseen. Tämä aiheuttaa vääristyneen perspektiivin. Tämän lisäksi pelin esineillä on vaihtelevat kuvakulmat ja graafiset projektiot; jotkut ovat perspektiivissä, jotkut ovat erilaisissa aksonometrisissä projektioissa ja jotkut ovat jopa vinoprojektiossa (aksonometrinen projektio ja vinoprojektio käydään läpi luvussa 2.2.2). Kuvassa 5 ilmenee neljän eri esineen graafisen projektion ero. [13.]

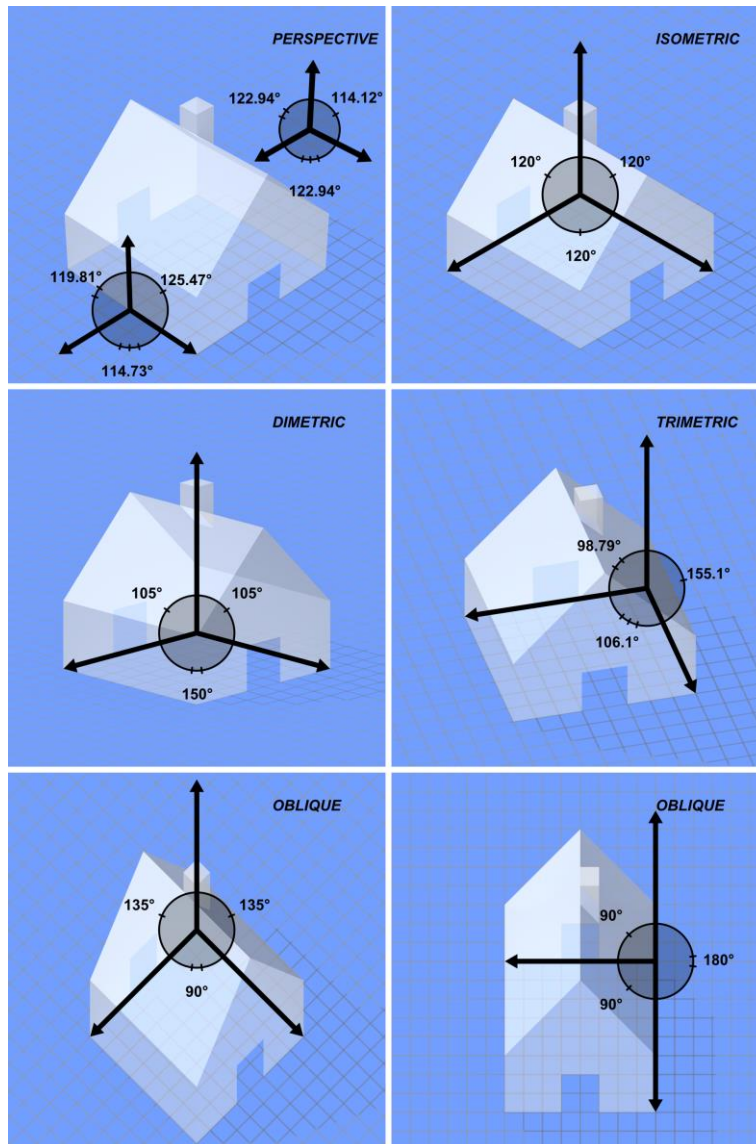
Yhdensuuntaisprojektiio

Yhdensuuntaisprojektiio (engl. parallel projection) jakaantuu ortografiseen projektioon ja vinoprojektioon. Yhdensuuntaisprojektiiossa ei ole pakopisteitä. Tämän vuoksi kaukaiset objektit ovat samankokoisia kuin edessä olevat objektit. Tämä voi aiheuttaa ongelmia, kuten kuvassa 6. Korkeutta ja syvyyttä on hankala havaita. Ortografisuus mahdollistaa kuitenkin elementtien asettelun ruudulle ilman, että tarvitsee pelätä, ovatko pakopisteet oikein. [14.]



Kuva 6. Aksonometrisen projektion ongelma [14].

Aksonometrinen projektio on ortografian alalaji. Siihen lukeutuu isometrinen, dimetrinen ja trimetrinen projektio (kuva 7). Isometrisessä projektiossa objektia katsotaan yläviistosta niin, että sen kolmen akselin, $x:n$, $y:n$ ja $z:n$, väliset kulmat ovat samansuuruisia. Kuvan 6 projektio on isometrinen. Isometriset 2D-pelit koostuvat usein tileistä eli laatoista, joita on levitetty pseudosatunnaisen lopputuloksen aikaansaamiseksi sinne tänne. Kuvan 6 laatat ovat hyvin yksinkertaisia laatikoita. Laatat voivat koostua niin kokonaisista rakennuksista kuin yksittäisistä ruohonkorsistakin. Satunnaisuus syntyy, kun esimerkiksi multa-grafiikkalaatan päälle lisätään laatta, jossa on ruohoa. Multa-laattoja ja ruoho-laattoja on saatettu tehdä muutamia erilaisia, joten yhdistelmiä syntyy lukuisia. [15.]



Kuva 7. Graafisen projektion tyypit [16].

Isometrisyys sekoitetaan monesti dimetriseen ja trimetriseen projektioon, koska näiden kolmen välillä on vain pieni ero. Toisin kuin isometrisessä projektiossa, dimetrisessä on vain kaksi identtistä kulmaa ja trimetrisessä ei yhtään. Kulmien tiedostaminen on tärkeää, koska grafiikka on tehtävä niiden mukaan. Ne antavat myös hieman erilaisen tunnelman peliin. Kuvassa 8 on kaksi erilaista peliä. Trimetristä projektiota käytetään Faloutissa (oikealla) ja dimetristä Final Fantasy Tactics Advancessa (vasemmalla). Faloutissa kuvakulma on suurempana ja talon yksi seinä nähdään paremmin, toisin kuin Final Fantasyssä, jossa kenossa oleva kuvakulma painottaa pelin taktisia piirteitä. [15; 9.]



Kuva 8. Projektion vertailua Final Fantasy Tactics Advancen ja Falloutin välillä [17; 18].

Vino projektio (engl. oblique projection) on kuin isometrinen projektio. Erona on kuitenkin se, että kameran lähinnä oleva pinta on suoraan kameraa kohden. Tätä projektiota käytetään lähinnä arkkitehtuurisissa piirustuksissa, kun halutaan näyttää etupinta mahdollisimman tarkasti ilman perspektiivin aiheuttamaa vääntyvyyttä. Kuvan 7 alimmaisen rivin oikean kuvan mukaista vinoprojektiota on käytetty muun muassa The Legend of Zelda: A Link to the Pastissa.

3 Pelituotannon kulku

Peliprojekti lähtee käyntiin konseptista, jota aletaan suunnitella, kunnes todennäköisesti todetaan, ettei se toimi. Näin ainakin useimmissa tapauksissa. Jos konseptin prototyyppi eli ensimmäinen pelattava versio vaikuttaa hauskalta ja lupaavalta, voidaan projektissa edetä. Pelituotannon sykli pitää sisällään pelituotannon vaiheet. Niitä ovat esituotanto, tuotanto, testaus ja jälkituotanto [19, s. 4]. Konseptitaiteen tekeminen aloitetaan esituotantovaiheessa, mutta varsinaisen peligrafiikan teko alkaa tuotantovaiheessa.

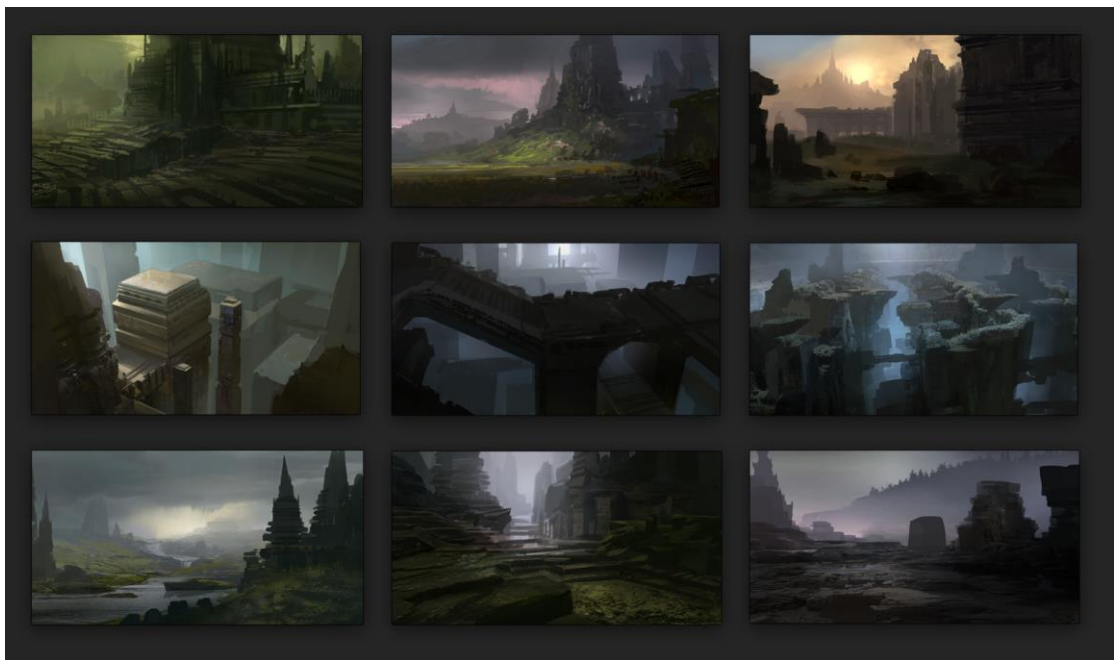
Peliprojekti etenee virstanpylväiden (engl. milestone) tahdittamana. Kun peli on siinä vaiheessa, että sitä voidaan jo pelata, julkaistaan ensimmäinen pelattava versio ja siitä edelleen alpha- ja beta-versiot. Jossakin vaiheessa pelin uuden koodin kehitys lopetetaan. Tästä edespäin koodaajat keskittyvät vikojen korjaamiseen. Kun viat on korjattu, pelin koodi hyväksytetään tuottajalla (engl. code release) [19, s. 144]. Julkaisuvalmista versiota kutsutaan gold masteriksi [19, s. 2].

3.1 Esituotanto

Esituotantovaihe alkaa, kun päätetään tehdä peli. Ensimmäiseksi tehdään konsepti eli idea. Esituotantoon kuuluu konseptoinnin lisäksi monia hallinnollisia asioita, kuten projektin vaatimusten määrittely ja riskien kartoittaminen. Grafiikan kannalta oleellisia osia esituotannossa ovat kuvakäsikirjoitus, konseptitaide ja visuaalinen ohjeistuksen laatiminen. [19, s. 6.]

Ennen kuin pelikonsepti siirtyy tuotantotiimille, projektiin liittyy konseptitaiteilija. Konseptitaiteilijan tehtävä on luoda visuaalinen kuva pelikonseptista. Konseptitaide voi sisältää hahmosuunnittelua, ympäristösuunnittelua, käyttöliittymäsuunnittelua ynnä muuta, mikä vaatii suunnittelua. Muutoksiin on varaa esituotantovaiheessa, joten konseptitaiteilijalla on vapaat kädet, kunhan vain seuraa pelisuunnittelijan näkemystä. Taiteilija käy läpi paljon lähdeaineistoa ja luo sen ja mielikuvansa pohjalta oman version. Konseptitaiteen tekeminen on nopeaa verrattuna muuhun taiteeseen, koska siinä keskitytään ongelman ratkomiseen ja tuotteen suunnitteluun. [20.]

Tyylillä ja visuaalisella ilmeellä ei ole niin paljon merkitystä kuin ajatusten kommunikomisella ja ideoiden tuottamisella. Kuvassa 9 on Eytan Zanan ympäristöluonnoksia. Näitä konseptin pieniä raakaluonnoksia kuvaillaan englanniksi thumbnailiksi. Niillä saa aikaan nopeasti monia ideoita ja vaihtoehtoja. Raakaluonnoksista valitaan paras, josta tehdään sitten viimeistelty konseptikuva. Raakaluonnokset voivat olla esimerkiksi hahmoista tai ympäristöistä. Ympäristösuunnittelu voikin olla isossa osassa konseptitaidetta. Sen tarkoitus on antaa kuva ympäristön tunnelmasta ja asioista, jotka ovat oleellisia paikalle. Hahmosuunnittelu on myös tärkeää; varsinkin päähahmon suunnittelu. Myös pelaamattomille hahmoille on hyvä tehdä konsepti, ovat ne sitten päähahmoja tai sivuhahmoja. [21, s. 97, 98, 114, 115, 116.]



Kuva 9. Ympäristöluonnoksia [22].

Taiteilijan apua voi tarvita myös mockupin tekemiseen. Mockupin eli valmiin pelin mallipiirroksen tekeminen auttaa osoittamaan, millaiseen lopputulokseen tähdätään. Mockupin tekeminen ei vaadi taiteellisia lahjoja, joten jopa pelisuunnittelija voi tehdä tämän. Sen voi koota valmiista kuvista tai maalata käsin; miten vain ajatuksen saa parhaiten julki.

Elokuvallisesta pelistä tai pelin välianimaatioista voi tehdä kuvakäsikirjoituksen. Kuvakäsikirjoitus on käsikirjoitusta vastaava visuaalinen representaatio pelin tapahtumista sarjakuvamaisesti. Kuvakäsikirjoituksen tarkoitus on ilmaista tapahtumien rytmiä ja

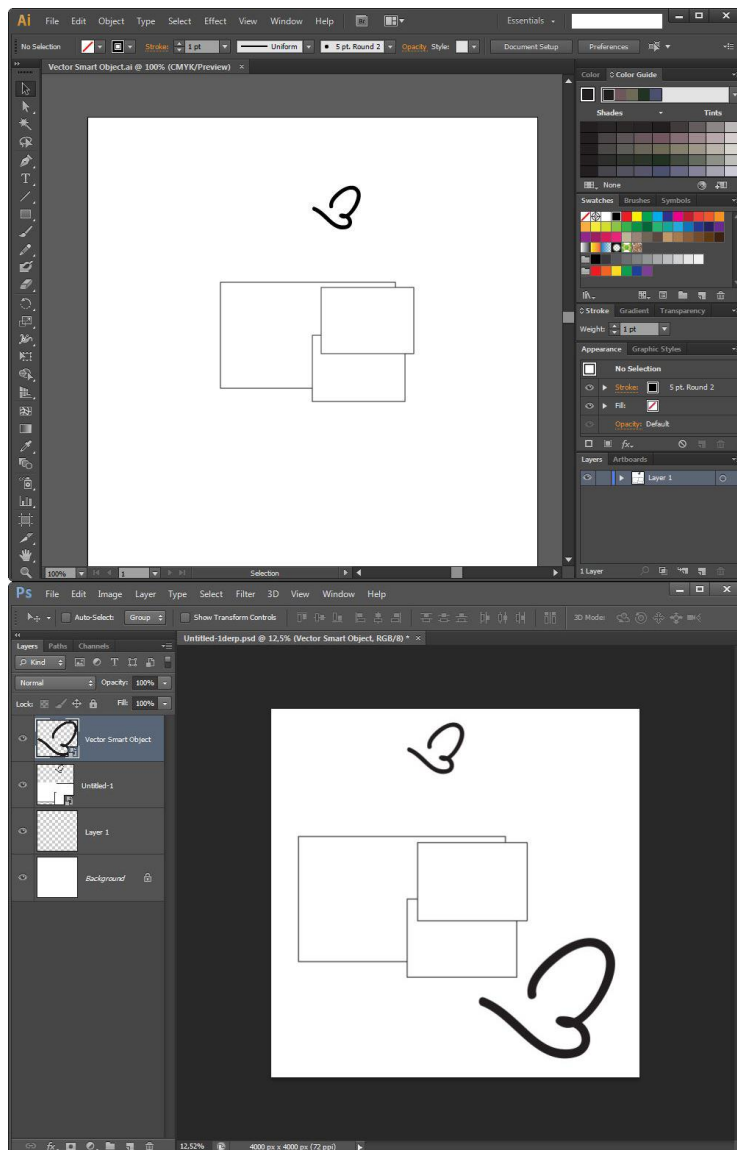
ajoitusta. Siihen merkitään yleensä kuvakulmat, kuvakoot, toiminta, aika, äänitiedot, keskustelut ja narraatiot. Jos kohtaukset vaativat draaman luontia, voidaan kuvakäsikirjoituksesta tehdä tiettyihin hetkiin sijoittuva. Tällöin yhdellä kuvalla yritetään sanoa mahdollisimman paljon. Jos animaation sulavuus on tärkeämpää, voi kuvakäsikirjoitukseen piirtää kohtauksen päätoiminnot nopealla tempolla. Tällöin voi olla syytä tehdä animatikki. Se on video kuvakäsikirjoituksen kuvista. Sen tarkoituksena on antaa pohjaa valmiiseen animaatioon. [23, s. 14–15; 24, luku 3; 25, luku 8.]

3.2 Työnkulun optimointi

Työnkulun optimointi nopeuttaa työskentelyä. Nopeus on usein tärkeää peliprojektissa. Peliteollisuudelle on hyvin yleistä, että aikarajat ovat kireät. Jatkoajalle on jopa oma termi: crunch time. Jos grafiikkaa tehdään sekalaisesti eri tehtävien välillä, voi jossain vaiheessa huomata, että mikään ei tule valmiiksi. Jos mikään ei tule valmiiksi, koodaajat joutuvat jatkuvasti hyödyntämään sijaisgrafiikkaa (engl. placeholder), joka ei anna täydellistä kuvaa lopputuloksesta. Hyvä tapa pitää itsensä oikeilla raiteilla on pitää todo-listaa, johon on merkitty tehtävät prioriteetin mukaan. Prioriteettiin vaikuttaa eniten se, mihin suuntaan tuottaja tai pelisuunnittelija on viemässä peliä. Tämä ei kuitenkaan estä todo-listan pitämistä, koska jokaisen tehtävän voi aina pilkkoa pienempiin osiin.

Ohjelmien välillä joutuu hyppimään, jos pelin toteutus niin vaatii. 3D:tä voi käyttää hyväksi esimerkiksi animaation tekoon. Hahmon voi animoida 3D-ohjelmalla ja renderöidä kuviksi. Kun oikeat kuvat eli animaation kehykset on valittu, ne jälkikäsitellään näyttämään piirrettyiltä. 3D mahdollistaa sulavan animoinnin ja mittasuhteet, jotka on hankalampi toteuttaa piirtämällä.

Työtiedostoja voi joutua avaamaan parin ohjelman välillä toistuvasti. Tämä on otettu huomioon Adoben tuotteissa, niin kuin näkee kuvasta 10. Kuvassa ylempänä on Illustrator ja alempana Photoshop. Objekteja voi kopioida Illustratorista suoraan Photoshopiin pikanäppäimellä. Jos elementin tai tiedoston kopioi Photoshopiin älykkäänä objektina, sitä voi muokata kummassakin ohjelmassa samaan aikaan.



Kuva 10. Samanaikainen työskentely yhdellä tiedostolla Adobe Photoshopilla ja Illustratorilla.

Yksinkertaistaminen säästää aikaa ja helpottaa kuvan tulkintaa. Liiottelu voi olla avain yksinkertaistamiseen, koska se helpottaa kuvan lukua. Aivan kuten karikatyyripiirroksissa: jos hahmolla on iso nenä, se kiinnittää huomion helpommin. Yleensä katsojan silmä ei käy läpi kaikkea maalauksessa tai piirroksessa. Maalauksen kohdat, joihin ei ole syytä kiinnittää huomiota, voidaan karsia yksityiskohdista. Yksinkertaistaminen on myös hyvin luonnollista, jos verrataan, kuinka silmä toimii: emme näe yksityiskohtia kaukaa. Tällöin tärkeäksi muodostuvat puolestaan mittasuhteet, valööri ja väri, jotka ovat helpommin havaittavissa. Digitaalinen maalaaminen mahdollistaa monia helpottavia tekniikoita. Maalaus voidaan esimerkiksi tehdä ensiksi harmaasävyinä ja värittää jälkepäin. Grafiikan uudelleenkäyttämistä kannattaa myös tehdä, kun se on sallittavaa.

4 Graafikon työkalut

Peligraafikolle on paljon eri työkaluja, joista valita mieleisensä. Ohjelmistoja on joka tarkoitukseen. Grafiikkaohjelmistojen lisäksi peligraafikon on käytettävä ohjelmia kuten GitHub, johon grafiikkansa laittaa ja josta pelin senhetkisen version voi ladata. Peligrafiikassa määrä on tärkeämpi kuin laatu verrattuna esimerkiksi kuvitustaiteeseen, jossa yhden maalauksen parissa kuluu useita päiviä. Työn eteneminen aikataulussa on siis tärkeää. Laitteiden ja ohjelmistojen hallitseminen nopeuttaa työskentelyä.

4.1 Laitteisto

Näppäimistö on osa digitaalista työskentelyä. Jokaisessa ohjelmassa on pikanäppäimiä, jotka nopeuttavat työskentelyä. Pikanäppäimet suorittavat ohjelman eri toimintoja, ja niillä voi valita työkaluja nopeasti. Pikanäppäin voi olla yhden tai useamman näppäimen yhdistelmä. Vaikka ne helpottavat ohjelman käyttöä, ne voivat olla epämiellyttävissä paikoissa ja vaatia käden liikuttamisen jatkuvasti. Käden liikuttaminen voi vaatia katseen siirtymisen näppäimistöön, jos pikanäppäinyhdistelmä on hankalassa paikassa. Miten käy, kun maalaa ja joutuu katsomaan toisaalle jatkuvasti? Välttämättä ei tunne mitään, mutta todellisuudessa keskittyminen kärsii joka kerta. Tämän huomaa, kun pikanäppäimet mukauttaa omaan työtapaansa. Kun tietää käyttävänsä tiettyjä pikanäppäimiä, ne kannattaa sijoittaa mahdollisimman helppoon paikkaan, jotta voi keskittyä työntekoon. [26; 27.]

Työskentelyä voi nopeuttaa edelleen eri laitteilla. Näppäimistön muoto ei välttämättä miellytä kaikkia. Jotkut USB-laitteet on suunniteltu ergonomiseksi. Niihin voi asettaa esimerkiksi pikanäppäimiä ja niissä on helpottavia ominaisuuksia, kuten rullasäädin. Näissä laitteissa on mahdollista asettaa macro-näppäimiä, joilla voi joko kiertää ohjelman pikanäppäinrajoituksia tai asettaa näppäinyhdistelmiä. Jos kuitenkin haluaa käyttää näppäimistöä, ja ongelmana ovat ohjelman rajoitukset pikanäppäinten sijoituksen suhteen, ei kannata luovuttaa. Internetissä on ladattavana muutamia ilmaisia ohjelmia, kuten KeyTweak, joilla voi asettaa näppäimistön näppäimet siten kuin haluaa. [27.]

Peligrafiikkaa tehdään nykyään digitaalisesti, joten jokaisella graafikolla pitäisi olla digitaaliseen piirtämiseen tarkoitettu piirtopöytä (kuva 11). Niitä on pienistä, noin A6-kokoisista piirustuslaudoista isompiin 24-tuumaisiin monitoreihin. Perinteisiä piirustus-

ja maalausvälineitä imitoivia pelejä on, mutta niiden grafiikka on joka tapauksessa tuotu digitaaliseen muotoon viimeisteltäväksi. Digitaalinen työskentely mahdollistaa myös paljon helpotuksia eikä vaadi paljon välineistöä, toisin kuin perinteiset menetelmät. Digitaaliseen maalaamiseen ei tarvita piirtopöydän lisäksi muuta kuin tietokone ja ohjelma. Tablettitietokone voi olla tietokone ja piirtopöytä samaan aikaan. Monessa tabletissa on jopa kynän paineentunnistustasoja, joten tuntuma ei paljon eroa erillisistä piirtopöydistä. Tablettitietokoneen näytön koko ei silti ole tarpeeksi suuri kaikille.



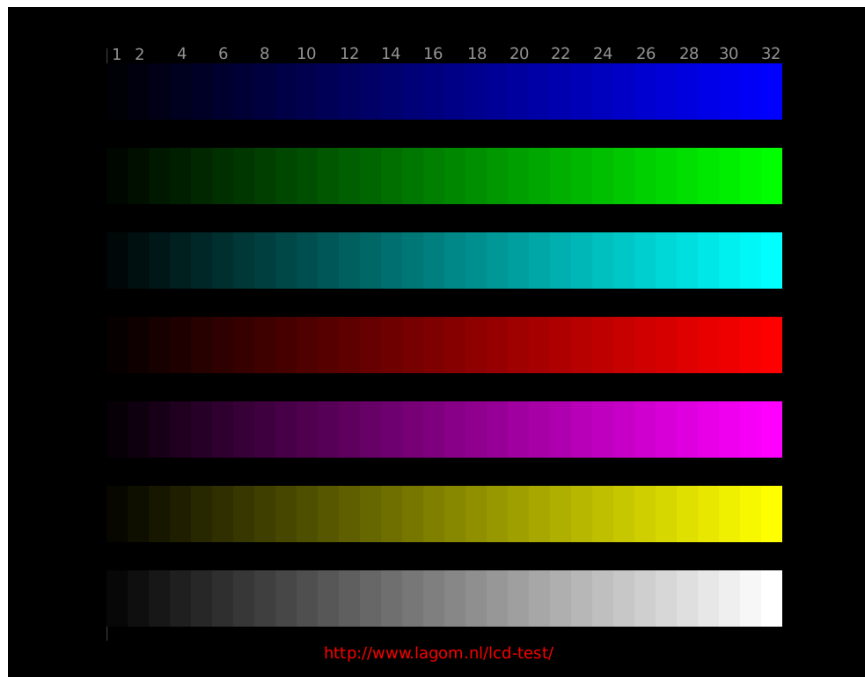
Kuva 11. Vaihtoehtoisia piirustusvälineitä digitaaliseen työhön.

Monitori on graafikolle kuin ateljee taiteilijalle. Ruudun koko ja pikselitiheys määrittää työtilan tilavuuden. Laatu vaikuttaa esimerkiksi väreihin ja kontrastiin. Nämä ominaisuudet ovat tärkeitä painetussa grafiikassa, jotta monitorin värit vastaisivat tulostettua työtä. Pelejä pelataan kuitenkin monilta eri näytöiltä. Jos grafiikka tehdään huonolaatuisella monitorilla ja pelaaja käyttää laadukkaampaa monitoria, jää värien potentiaali uupumaan kummaltakin. Laadukas monitori ei takaa parempia värejä, jos pelaajalla on käytössään huonolaatuinen monitori, mutta se voi helpottaa työskentelyä, koska värit ovat helpommin erotettavissa.

Monitorin paneelityyppi kertoo näytön laadusta. Arvelisin, että pelaajien tavallisin paneelityyppi on TN (twisted-nematic), koska sitä on myyty paljon edullisuutensa vuoksi. Siinä on lisäksi nopea vasteaika, mikä on tärkeää pelaajalle. IPS-paneelin vasteaika on huonompi, mutta sen värien toisto ja katselukulma on parempi, mikä tekee sen soveltuvaksi graafiseen työhön. Työskenneltäessä IPS-paneelin monitorilla on hyvä tarkistaa työn jälki TN-paneelin monitorissa. [28.]

Näytön värejä voi säätää kalibroimalla. Kalibroinnin voi suorittaa tietokoneen käyttöjärjestelmän oman kalibroitsovelluksen, muun sovelluksen tai erillisen kalibroitilaitteen

avulla. Windowsin kalibrointi käy läpi kirkkauden, kontrastin, värit ja terävyyden. Kalibroinnin säädöt suoritetaan osaksi ohjelman sisäisesti ja osaksi monitorin asetuksista. Lagom.nl-internetsivun LCD-testin voi tehdä pelkän monitorin asetusten avulla. Sen kontrastitesti on kuvassa 12. Huonoilla näytöillä on hankala erottaa värien kylläisimpiä ja tummimpia arvoja kontrastitestistä [29]. Tällaiset manuaaliset testit vievät aikaa ja voivat olla hankalia. Kalibrointilaite helpottaa kalibrointiprosessia ja takaa värien toimivuuden. Sitä ei tarvitse kuin laittaa monitorin eteen ja suorittaa ohjelma kalibroimiseksi.



Kuva 12. Kontrastitesti monitorille [29].

4.2 Ohjelmistot

Peligrafiikka tuodaan pelimoottoriin, jossa varsinainen peli rakennetaan. Grafiikan voi luoda halutussa ohjelmassa, koska niissä on usein tallennusvaihtoehtona useampia tiedostomuotoja. Ammattimaisessa projektissa tarvitaan ohjelma vielä projektinhallintaan, kommunikaation, tiedostonjakoon ja versiohallintaan. Työtiedostoja voi siirtää esimerkiksi pilvipalvelun tai ftp-palvelimen kautta, mutta miksi ei vaikka sähköpostinkin välityksellä, jos se on miellyttävän ratkaisu. Valmiin grafiikan voi ladata myös suoraan peliin versiohallintaohjelman, kuten GitHubin, avulla.

Pelimoottorit

2D-pelit eivät ole rajoittuneita pelkästään 2D-pelimoottoreihin. Niitä voi tehdä myös 3D-pelimoottoreilla, kunhan pelimoottorissa on mahdollista luoda ortografinen kamera. Pelimoottoreita on tarjolla maksullisina ja maksuttomina. Joillakin pelistudioilla, kuten Ubisoftilla, on käytössään oma uniikki pelimoottorinsa 2D-peleille. Rayman Origins ja Rayman Legendsin kenttäsuunnittelu on tehty artistiystävällisellä Ubisoftin Ubiart-ohjelmalla. Pelimoottori itsessään riittää valmiin pelin tekemiseen. Se sisältää yleensä ohjelmistokehyksen, renderöijän, fysiikkamoottorin ja useita muita toiminnallisuuksia.

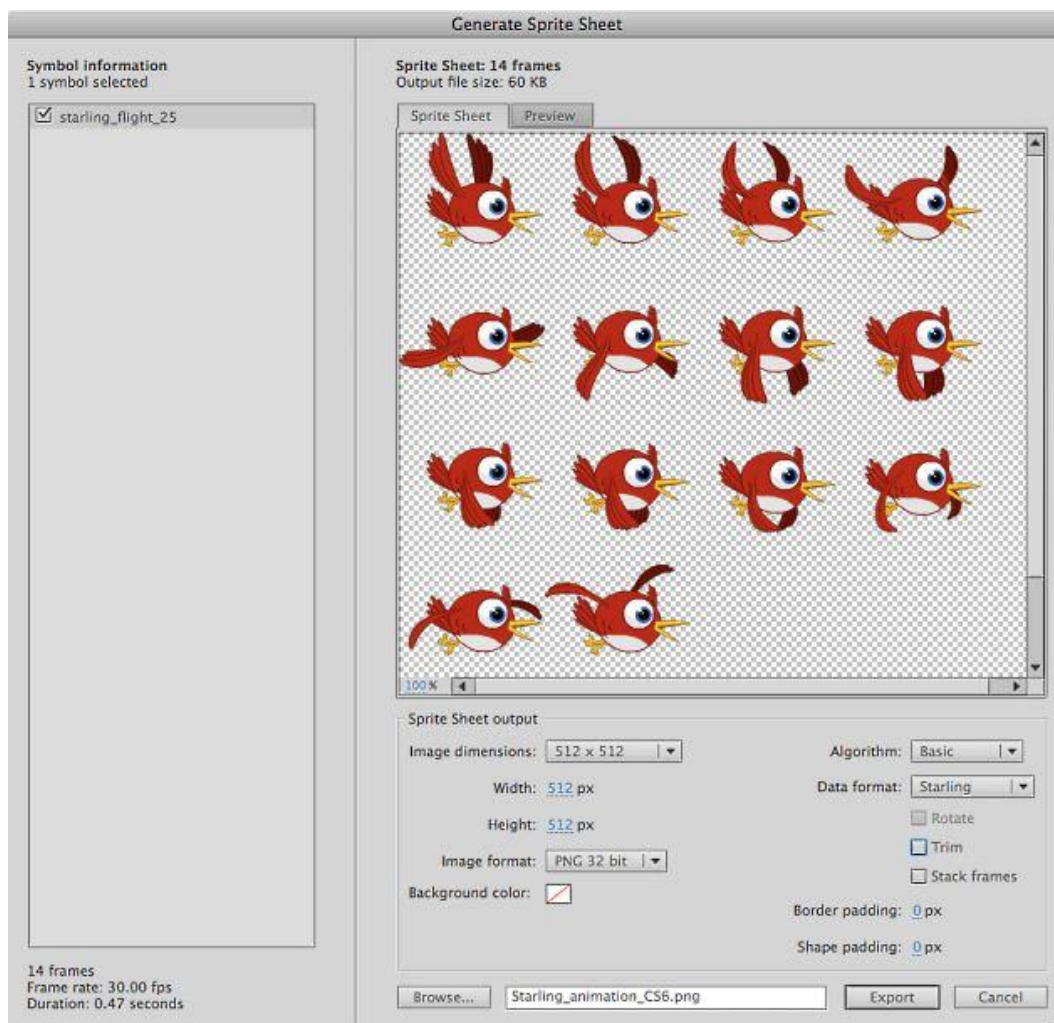
Ohjelmistokehykset voivat olla osa pelimoottoria tai oma ohjelmansa. Monesti ohjelmistokehyks katsotaan pelimoottoriksi. Ohjelmistokehykset sisältävät koodikirjaston, jota käytetään rakentamaan peliä. Renderöijä muuttaa työtilan objektit kuvaksi ja ottaa samalla huomioon esimerkiksi valaistuksen ja fysiikan. Fysiikkamoottori huolehtii pelin luonnollisesta liikkeestä, kuten painovoimasta ja objektien törmäyksestä. Pelimoottoreissa on yleensä tuki yhdelle koodikielelle. Yleisimmät ovat C, C++, Python, Java, Javascript ja Actionscript [30]. Graafikon ei yleensä tarvitse ymmärtää paljoa koodista, mutta pelimoottorin käyttöjärjestelmän hallitseminen voi olla aiheellista.

Grafiikkaohjelmat

Grafiikkaa voi tehdä eri ohjelmilla riippuen pelistä. Maalaukset voi tehdä millä tahansa maalausohjelmalla, kuten Paint tool SAI:lla. Vektoreita voi tehdä vaikka Adobe Illustratorilla. Flash on myös hyvin monipuolinen ohjelma. Sillä voi tehdä vektorigrafiikkaa, animaatioita tai vaikka kokonaisia pelejä. Adoben tuotteet ovat suosittuja, mutta kalliita. Photoshop on kuvankäsittelyn ja maalamisen standardi, mutta tarjolla on monia muitakin vaihtoehtoja maksullisista maksuttomiin. Maksuttomat voivat olla ihan yhtä hyviä kuin maksullisetkin peruskäytössä. Photoshopin voi korvata esimerkiksi Kritalla ja Illustratorin Inkscapeella. Jos pelistä halutaan retromainen, kannattaa harkita pikselitaide-ohjelmia, kuten Pyxeledit tai GraphicsGale.

Animaation voi periaatteessa tehdä maalausohjelmalla, mutta animaatio-ohjelmia on myös erikseen. Animaation voi toteuttaa myös koodilla ja pelimoottorin sisällä. Selainpelien tai mobiilipelien animaatio voi olla niin yksinkertaista, ettei varsinaista animaatio-ohjelmaa tarvita. Sprite-ohjelmista voi olla hyötyä, koska niillä voi helposti koota ani-

maation spritesheetiksi. Tähän voi käyttää esimerkiksi Adobe Flashin Generate Sprite Sheet -työkalua (kuvassa 13).



Kuva 13. Adobe Flashin Generate Sprite Sheet -työkalu [31].

Generate Sprite Sheet -työkalu huolehtii muun muassa rivityksestä, kuvan formaatista ja resoluutiosta. Kuvia ei tarvitse erikseen sijoittaa riveihin ja sarakkeisiin pikselilleen. Jos kuvia ei ole sijoitettu spritesheettiin tarkasti, animaatioissa voi nähdä tärinää tai kuvien leikkausta.

5 Grafiikan tuottaminen

5.1 Graafinen ohjeistus

Graafiseen ohjeistukseen on merkitty tyylimäärittelyt, joita noudattamalla pyritään yhtenäistämään pelin ilmettä. Yhtenäisyys muodostuu erinäisistä asioista, kuten värivalinnoista tai taidetyylistä. Taiteilijan on kyettävä kaappaamaan pelin tunne ja noudattamaan tyyllisiä ehtoja. Lasten pelit poikkeavat aikuisten peleistä. Kulttuureissa on eroja. Graafiseen ohjeistukseen on yleensä laitettu irtonaisia kuvia tai jopa moodboard. Moodboard on kokoelma useista kuvista, joita käytetään referenssinä esimerkiksi jollekin tärkeälle esineelle.

Graafikon tehtävä ei ole tehdä kohderymäanalyysiä, mutta joitain asioita on silti pidettävä mielessä yleisöä ajatellen. Graafisissa ohjeissa taidettyli voidaan määrittellä vaikka sarjakuvamaiseksi leikki-ikäisille sopivaksi tyyliksi. Lapset pitävät värikkäistä helposti lähestyttävistä peleistä. Punaisen käyttöä joissain asioissa on vältettävä, koska sen voi liittää helposti vereen.

Väripaletti voi olla hyvin tarkasti määritelty. Jos värejä ei ole rajoitettu ja valikoitu teeman ympärille, tulee peliruudusta tukkoisen oloinen. Värit kuvastavat tunnelmaa, vaikuttavat sommitteluun ja auttavat silmää löytämään elementit, kuten pelihahmon tai vivun seinässä. Paikoilla on aina jonkinlainen teema. Kun pelissä edetään, väripaletti voi muuttua lokaatioiden mukana. Kun sankari etenee viimeistä lokaatiota kohden, on hyvin yleistä, että värit muuttuvat ystävällisistä lämpimistä väreistä pelottaviin kylmiin väreihin ja ystävällisistä kylistä pelottaviin eksoottisiin kohteisiin. Pelien väripaletti on yleensä hyvin suppea. Puut on rajoitettu pariin väriin. Huomiopisteet, kuten noukittavat esineet, ovat pääosin yhtä väriä. Värien yksinkertaisuus auttaa luettavuutta. Esineen voi tunnistaa, ennen kuin ymmärtää, mikä tietty esine on kyseessä, jos värit ovat tarpeeksi selkeät.

5.2 Referenssi

Ammattimaista piirrosta ei tehdä kylmiltään. Ennen kuin kynä edes koskettaa paperia, taiteilijan on hyvä katsoa referenssiä tai vähintään olla hyvin tietoinen siitä, mitä on tekemässä. Referenssi voi olla kuvia, videoita tai tekstiä, joita voi käyttää lähteenä mo-

nenlaisille visuaalisille ideoille. Referenssiä käytettäessä on pidettävä mielessä, että tarkoitus ei ole kopioida, vaan oppia uutta ja soveltaa sitä omalla tapaa.

Nopea tapa oppia piirtämään jotain tiettyä materiaalia on etsiä internetin hakupalvelusta referenssiä, mutta koska taiteilijan on hyvä osata piirtää mitä vain, onkin järkevää pitää silmät auki kaikkialla. Arvostettu konseptitaiteilija Feng Zhu painottaa monissa Youtube-opetusvideoissaan, kuinka tärkeää on kerätä ”visuaalista kirjastoa” eli mielikuvaa kaikista mahdollisista asioista. Myöhemmin tämä kirjasto auttaa piirtämään mielikuvituksesta. [32.]

Joitakin asioita ei kuitenkaan voi tietää tai muistaa. Asioiden piirtäminen muistista on eri asia kuin piirtäminen mallista. Kun piirretään muistista, muistetaan lähinnä oleellimmat asiat, kuten hevosen jalan z-muoto tai hännän harjamaisuus. Jos animoidaan nyt hevonen, sen jalat vaikuttavat elottomilta tukeilta, jotka hyppivät eri tahtiin, ja häntä tulee perässä kuin takapuoleen ammuttu nuoli. Syynä tähän on tietämättömyys hevosen anatomiasta. Luiden rakenne määrittelee jalkojen liikeradan ja asennon, kuten takajalkojen käänteisen polven. Lihakset supistuvat ja antavat uskottavan muodon. Häntä on kuin lippu, joka piiskaa ilmaa hevosen juostessa. On monia asioita, joihin ei tule kiinnitäneeksi huomiota, vaikka olisi kuinka kiinnostunut ympäristöstä. [33.]

Referenssi voi toimia inspiraationa. Taiteilija tarvitsee inspiraatiota antamaan suuntaa omaan henkilökohtaiseen tyyliinsä. Suunnitteluvaiheessa inspiraatio antaa suuntaa koko pelille. Referenssiä kannattaa olla monesta eri subjektista, ja ideoita kannattaa hakea eri lähteistä. Hahmosuunnitteluun kannattaa hakea ideoita esimerkiksi mytologiasta, eläimistä tai elektroniikasta, jotta hahmon voisi helposti liittää oikeaan maailmaan ja pitää hahmon tyylin yhtenäisenä. Jos pelin maailma perustuu historialliseen aikakautteen tai hakee vaikutteita sieltä, kannattaa referenssiä hakea historiankirjoista, jotta kykenee näkemään maailman toisin silmin kulttuurin ja esimerkiksi arkkitehtuurin kautta. [34.]

Laadukkaiden kuvien etsiminen internetistä johtaa usein maksullisille sivuille. Referenssiä voi kuitenkin löytää paljon muualtakin. Kuvakirjoja on aiheesta kuin aiheesta, ja niissä on paljon korkealaatuisia kuvia, joissa on otettu valaistus huomioon. Anatomiaa on esimerkiksi nyrkkeilyvideoissa. Animaation kannalta on tärkeä nähdä liikettä, joten mikä olisi parempi paikka kuin Vimeo tai Youtube, jossa voi pysäyttää videon ja analysoida liikettä. [34.]

5.3 Hahmon luonti

Pelin hahmoiksi luokitellaan kaikki ei-pelattavat hahmot ja pelaajahahmot. Ei-pelattavat hahmot eli tekoälyllä toimivat hahmot ovat vihollisia, ystäviä tai sivustakatsojia. Joissakin peleissä on vain pelaajahahmoja, ja joissain peleissä ei ole tarvetta hahmoille ol- lenkaan. Monessa pelissä hahmot ovat kuitenkin tärkeässä asemassa. [35, s. 2, 4.]

3D-hahmon luonnin työkulku on moniportainen, mutta 2D-hahmo voi olla yksi piirros hahmosta. Hahmosta voidaan tehdä konsepti, kuvitus ja animoitava peliobjekti. Lisäksi hahmolle voi olla erilaisia vaatteita ja välineitä. Hyvä hahmo on muun muassa selkeä, yhtenäinen, persoonallinen ja sopiva pelimaailmaan. Hahmon miellyttävyyttä lisää eri- koisuus ja taidokas toteutus, jota vahvistavat peligraafikon taiteelliset taidot eli lähinnä anatomian, valöörien, värien ja sommittelun hallinta.

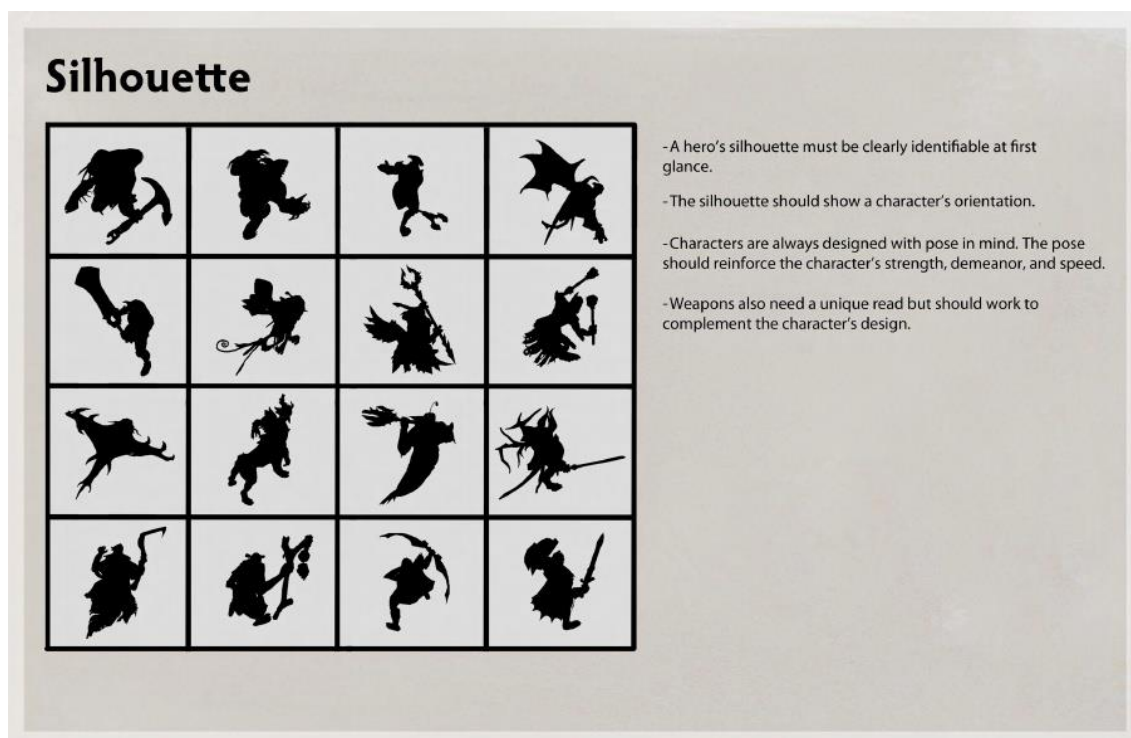
Graafiset ohjeet antavat pohjaa hahmon visuaaliselle ilmeelle. Referenssin etsiminen on suositeltavaa. Se parantaa ideointia ja taiteilijan omaa visuaalista kirjastoa. Referenssistä kannattaa tehdä moodboard, joka kokoaa kuvia anatomialle, tyyllille ja materi- aaleille. Kannattaa myös tutkia muiden pelien samankaltaisia hahmoja ja merkitä muis- tiin asioita, jotka koskevat hahmoa. Näitä ovat esimerkiksi taustatarina ja persoonalli- suus. Hahmon on sovittava pelimaailmaan ja toimittava pelilajityypissään. Hahmo- suunnittelija tekee päätöksensä värien, tunnelman, sommittelun ja omistusesineiden suhteen. Hän pääsee myös perehtymään hahmon sanattomaan viestintään eli eleisiin, ilmeisiin ja asentoihin.

Sivuhahmolle riittää stereotyyppinen yksiulotteinen persoonallisuus, joten tämä ei vaadi suurempia tarinankerronnallisia taitoja graafikolta. Jos kyseessä olisi päähahmo, peli- suunnittelija olisi luultavasti jo luonut hahmon persoonallisuuden, koska se on olennai- nen asia hahmoon samaistumisen kannalta. Hahmojen persoonallisuuksia ja muita hahmonkehityksen olennaisia apukeinoja on tvtropes.orgin kaltaisilla verkkosivuilla. Sivun on vapaan sisällön tietosanakirja tropeille. Tropet ovat tarinankerronnan yleistymiä ja kirjoittajan apukeinoja. Feng Shui opettaa monessa opetusvideossaan, kuinka kirjat avaavat taiteilijan luovuutta [34]. Mikä olisikaan parempi tiedon lähde luovuuden avaa- miselle kuin valmiit yleistymät, joita kirjailijatkin käyttävät hyväkseen [36]?

Vihollishahmojen häviäminen taustaan on epämiellyttävää pelaajalle, mutta se voi olla tarkoituksellista esimerkiksi kauhupelissä, jossa näkymätön vihollinen lisää jännitystä.

Pelaajahahmon erottuvuutta taustasta voi parantaa oikeilla väreillä, kontrastia nostamalla ja äärirajojen paksuudella. Myös eri hahmojen välillä pitää olla ero, jotta pelaaja erottaisi hahmon mahdollisimman nopeasti. Tämän eron löytää helposti, kun tietää, mikä on hahmon tarkoitus. Onko hän päähahmo vai sivuhahmo, kauppias vai vihollishahmo? Onko hahmo ihminen, antropomorfinen esine, fantasiarotu vai jotakin muuta? Onko hahmolla jokin esine tai ase mukanaan, joka täydentää hahmon persoonallisuutta?

Hahmo on hyvä aloittaa tekemällä raakaluonnoksia. Kuvassa 14 on online-strategiapeli DOTA 2:n graafisten ohjeiden raakaluonnoksia. Raakaluonnoksissa voi olla jonkin verran yksityiskohtia, kuten värejä, mutta yleensä ne ovat siluetteja eli mustavalkoisia luonnoksia, joiden tehtävänä on kuvailla hahmoa yksinkertaisesti. Siluetteja tehdään useita, koska niiden tekeminen on nopeaa. Hahmosta saa useita eri vaihtoehtoja, kun niitä kopioi, pyöryttää ympäri tai muokkaa vähän asentoa. Siluetin on oltava tunnistettava ja yksinkertainen, joten yksityiskohtiin ei kannata keskittyä. Siluetin kuvakulma lisää tunnistettavuutta. Hyvä esimerkki tästä on kuvassa 14. Jos hahmon siluetin piirtää edestäpäin, ei siitä välttämättä erota, miten päin hahmo on. Jos hahmon piirtää yläviistosta, kuten DOTAn hahmot, voi tunnistaa, mihin suuntaan hahmo katsoo. Siluetti voi kuitenkin aina olla kummin päin tahansa, mutta yläviistosta vastakkainen kulma on epäluonnollisen oloisesti alhaalta viistosta, joten sitä on hankalampi nähdä. Jos siluetti toimii, voidaan ymmärtää, miten hahmo liikkuu. Tämä on tärkeää, kun kyseessä on vihollishahmo. [37.]



Kuva 14. Raakaluonnoksia [37].

Hahmon osien, kuten varusteiden, sijoittaminen visuaalisesti järkeviin paikkoihin on sommittelua. Hahmon sommittelu alkaa jo raakaluonnoksista, kun piirretään kehon suhteita ja päällekkäisyyksiä. Jos hahmoa tarkastellaan tarkemmin, täytyy hahmo olla hyvin sommiteltu siluetin sisältäkin. Pelihahmot ovat tavallisesti hyvin yksinkertaisesti sommiteltuja, mutta pelaajan hahmo voi olla yksityiskohtaisempi. Sommitteluun lukeutuu usein valöörisommittelu ja värisommittelu.

Valöörisommittelussa keskitytään musta-valkotasapainoon. Se ohjaa katseen oikeaan paikkaan hahmossa; eihän haluta kiinnittää huomiota vihollisen jalkoihin, kun tarvittava informaatio on aseessa, jota hahmo pitelee käsissään uhkaavan oloisesti. DOTAn hahmoissa käytetään valöörigradiattia ohjaamaan katse ylös hahmojen kasvoja kohti. Hahmot on lisäksi jaettu valöörisiin, jotka helpottavat hahmon lukua. Kuvassa 15 on neljä DOTA 2:n valmiita hahmoa väreineen. Värit kannattaa pitää yksinkertaisina, koska hahmot ovat vain osa koko peliä. Yksinkertainen ja värikylläinen hahmo erottuu nopeammin kuin turhan monivärinen tai väritön hahmo. Kun hahmojen tyyliohjeisiin on määritetty värien käyttö kunnolla, saadaan hahmoista samantyyliisiä. DOTA 2:n hahmoille valitaan ensiksi yksi pääväri, jonka perusteella valitaan kaksi muuta väriä. Hyvä väritasapaino saadaan, kun tärkein väri peittää noin 60 prosenttia hahmosta, toinen

väri 30 prosenttia ja kolmas väri 10 prosenttia. Valitsemalla eri värit, kuin pelimaailmas-
sa on muualla, saadaan hahmot erottumaan taustasta. [37.]



Kuva 15. Valmiita hahmoja [37].

Yksityiskohtia tehdessä kannattaa pitää mielessä pelilaitteen ikkunakoko ja hahmon suhde siihen. Jos kysessä on mobiilipeli, turhien yksityiskohtien käyttöä pitäisi varoa, koska ne eivät piirry näytölle, ellei hahmo ilmesty koko ruudun koossa. Yksityiskohdat voivat myös hankaloittaa hahmon tulkitsemista. Hahmosta voi kuitenkin olla erikseen yksinkertainen pelihahmo ja yksityiskohtainen kuvitus. DOTA 2:n ohjeet painottavatkin yksityiskohtien käyttöä vain vähän. [37.]

5.4 Kenttä

Riippumatta ohjelmasta 2D-pelien sommittelu eli kompositio tehdään usein tasoilla. Skullgirls-taistelupelissä (kuva 16) on selkeät tasot. Päällimmäisenä on graafinen käyttöliittymä. Taaimmaisena on taustakuva eli kenttä. Keskitasossa sijaitsevat peliobjektit, kuten hahmot ja erikoisefektit.



Kuva 16. Skullgirls-taistelupeli [38].

Peliruudun etutason käyttöliittymä eli HUD (heads-up display) on tärkeä, koska siinä ilmenee paljon tietoa pelaajalle. Siinä voivat olla esimerkiksi pelaajan tavarat tai pisteet. Joskus pelaajalla on vain lyhyt silmäys aikaa vilkaista omat pisteensä, koska peli vaatii keskittymistä. Pisteiden pitäisi olla siis selvästi esillä. Taustataso ja keskitaso vaikuttavat etutason sijaintiin. Jos pelin toiminta tapahtuu oikeassa reunassa, olisi varmasti paras laittaa pistetaulu vasempaan laitaan. Kannattaa myös muistaa, että etutaso tulee muiden tasojen päälle, joten viholliset voivat piileksiä sen takana, jos se ei ole läpinäkyvä tai tarpeeksi pieni.

Pelin toiminta tapahtuu keskitasossa. Hahmojen, vihollisten, esineiden ja erikoisefektien tulisi erottua taustatasosta. Koska keskitaso on tärkeämpi osa pelattavuuden kannalta kuin tausta, kontrasti ja värit ovat vahvempia siinä. Hahmoja voi erottaa taustasta valokehällä. Valokehää käytetään hahmon ympärillä yleensä, kun hahmo on valittuna tai on jonkin erikoisvoiman alaisena. Valokehää voi käyttää myös esineissä. Kerättävien esineiden on oltava selkeitä ja uskottavia. Usein niillä on myös selkeä tärkeysero. Arvokkaat esineet loistavat (valokehä) ja ovat houkuttelevamman näköisiä. Vaarallisten esineiden ja objektien on erotuttava ystävällisistä objekteista, jotta pelaaja ei koe ikäviä takauksia. [39.]

Taustataso tuo pelin sommittelun yhteen. Se on usein ympäristökuva. Alhainen kontrasti, yhtenäinen väripaletti ja tarkoin suunniteltu sommittelu on tärkeää, jotta se ei ole liian sekava. Jos taustaa ei ole suunniteltu kunnolla, voi pelaajalle tulla ongelmia ym-

märtää, mihin pitää mennä. Valoa ja kontrastia voi käyttää ohjaamaan silmää tärkeisiin paikkoihin. [39.]

5.5 Graafinen käyttöliittymä

Käyttöliittymän (engl. user interface eli UI) tärkein ominaisuus on käytettävyys. Kun käyttäjä näkee ensimmäistä kertaa päävalikon, hänen on ymmärrettävä, mitä valikossa voi tehdä. UI voi olla graafinen käyttöliittymä (engl. Graphical user interface eli GUI) tai tekstipohjainen (engl. Command-line interface eli CLI). Yksinkertaisissa mobiilipeleissä käyttöliittymä on oleellinen osa peliä. Esimerkiksi managerointipeleissä käyttöliittymä voi olla koko peli, eikä sen tarvitse olla visuaalisesti erikoinen. Useassa pelissä graafinen käyttöliittymä on kuitenkin pakollinen. Valikoita käytetään moneen tarkoitukseen. Valikoita ovat esimerkiksi päävalikko, latausruutu, asetukset-valikko, lopputekstit ja lennähtävät vahvistusruudut. [40, s. 147.]

Valikot johtavat toisiin valikoihin. Valikkorakenne on suunniteltava loogiseksi, ja turhien alavalikoiden käyttöä pitäisi varoa, ettei pelaaja eksy tai koe valikoissa suunnistamista turhauttavaksi. Hienot siirtymisanimaatiot voivat olla sadannen pelikerran jälkeen puuduttavia, joten ne kannattaa tehdä nopeiksi. Animaation ja visuaalisen ilmeen pitää kuvastaa pelin ilmettä, koska esimerkiksi aloitusruudun tarkoitus on viestittää pelaajalle pelin ilmettä ja mainostaa peliä. [40, s. 148, 149.]

Käyttöliittymän nappulat ja typografia viestittävät pelin tyyliä ja hyvin tehtynä helpottavat navigoimista. Sellaisissa peleissä kuin Angry Birds ei tarvita tekstiä ollenkaan, koska nappulat, kuvakkeet ja symbolit on niin selkeästi toteutettu. Selkeyttä auttavat vaikiintuneet symbolit kuten pause-merkki. Nappuloiden värillä on voi olla symbolinen merkitys: palkintoruutu on usein kullankeltainen ja poistumis-nappula negatiivisen punainen.

5.6 Animaatio

Pelitaiteilijan on tärkeä osata animoida, koska kaikkea ei voi tehdä koodilla. Kun meiniin tai hahmoihin laitetaan liikettä, ne heräävät eloon ja auttavat pelaajaa ymmärtämään pelin tunnelmaa tai jopa samaistumaan hahmoon. Hahmojen kehonkieli, kuten myös kamera-ajot ja valikkoanimaatiot, ovat osa persoonallista tarinankerrontaa. Ani-

maation voi toteuttaa bittikarttagrafiikkana, vektoreina tai niiden yhdistelmänä. 3D voi toimia myös apukeinona. Animointiin voi käyttää monia ohjelmia. Maalausohjelmalla-kin voi tehdä yksinkertaista animaatiota. Vektoreita on helppo animoida Adobe Flashilla. Modernin retropelin tekoon voi käyttää Photoshopia, mutta niiden tekoon on lukuisia erikoistuneita ohjelmia, kuten GraphicsGale.

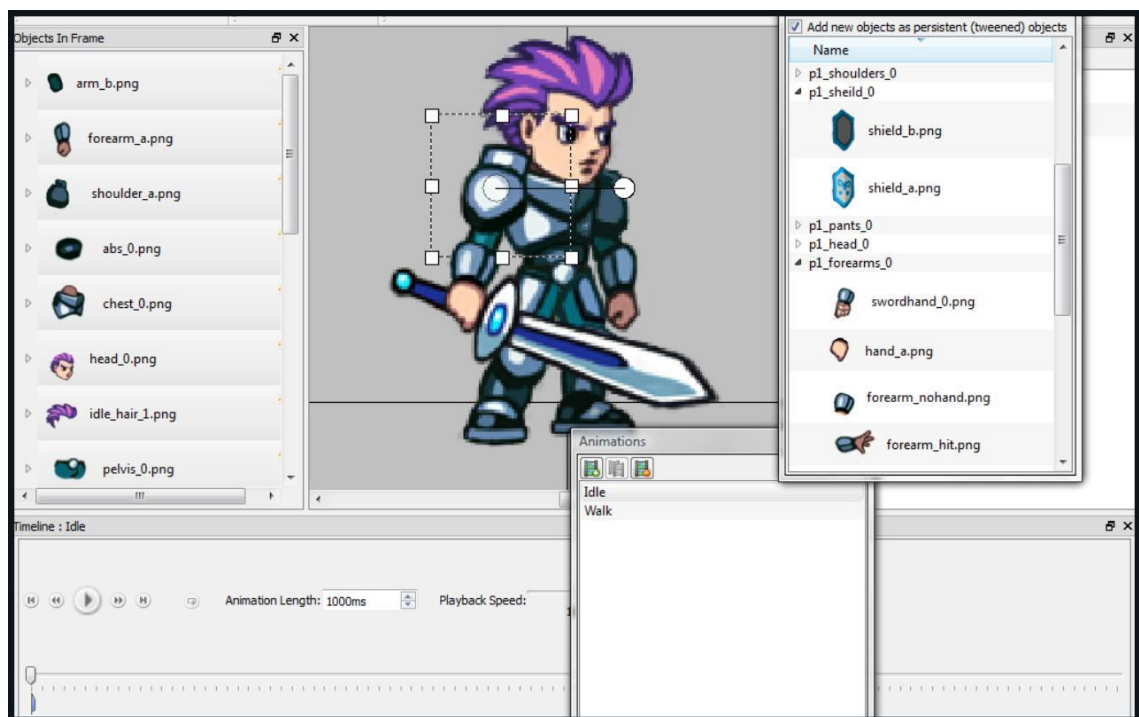
Luemme tavallisesti mikroilmeitä ja kykenemme näkemään tarkasti jokaisen puun lehden liikkeen, mutta pelissä halutaan olla varmoja, että kaikki on selvää pelaajalle. Vaikka kyseessä olisi realistinen peli, halutaan animaatiota yksinkertaistaa ja liioitella. Jo pelin tyylikin voi rajoittaa animaation mahdollisuuksia: Mitä jos peli on kuvattu kaukaa lintuperspektiivistä? Tällöin hahmon on suurin piirtein lennettävä, kun häntä ammutaan, vaikka näin ei tosielämässä kävisi.

Hyvin vähällä animaatiolla pärjää, jos pelin tyyli antaa näin olettaa. Otetaan esimerkiksi maailman suurin animaatioiden tuottaja, Japani. Japanilainen animaatio eli anime on jokapäiväistä paikallisessa televisiossa. Jotta animaattorit pysyvät aikarajoissaan, he käyttävät mahdollisimman paljon valmiiksi piirrettyjä kuvakehyksiä ja kohtauksia. Kamera-ajoilla ja erikoisefekteillä säästetään myös aikaa ja vaivaa. Peleissä on usein samaa ideaa. Pelianimaatiot eroavat siinä määrin animaatiosarjoista, että pelianimaatioita uudelleenkäytetään ja yksinkertaistetaan vielä enemmän kuin sarjoissa. Jos pelissä on sata hahmoa, olisi hankalaa animoida kaikki erikseen ja laadukkaasti. On paljon helpompaa käyttää animaatioita uudelleen. Sulavan animaation sijasta haluamme, että animaation ymmärtää ja se on ajoitettu oikein. Anime-sarjassa nyrkin iskua voidaan maustaa kaikenmoisin erikoisefektein, symboloivien salamaniskujen ja kamerakulman pyörimisen säestämänä. Taistelupelissä tämä isku voi olla toteutettu yhdellä kuvakehyksellä, joka toistuu parin sekunnin ajan.

Liikkuva hahmo antaa pelaajalle palautetta. Jos hahmoa ammutaan, hahmoa sattuu. Luodista muodostuva efekti ilmaisee osuman tehokkuuden. Jos pelissä on paljon erikoistaitoja, kuten taistelupelissä, voi olla parempi antaa prioriteettia efektien näkyvyydelle. Kun pelaaja oppii paremman iskun, joka on voimakkaampi kuin aikaisempi, erikoisefekti on myös suuremman ja tehokkaamman oloinen.

Animaatiotekniikat peleissä

Perinteinen animaatiotekniikka, jossa kuvat eli kehykset piirretään yksi kerrallaan, on nimeltään frame-by-frame. Sitä käytetään lähinnä animaatioelokuvissa, mutta hyödynnetään vielä nykyään pelien sprite-animaatioissa. Sprite-animaatiot voivat olla yksi- tai monitasoisia. Visuaalisesti niillä ei ole juurikaan eroa, mutta teknisesti paljonkin. Yksitasoinen sprite koostuu hahmon tapauksessa itse hahmosta sekä kaikista erikoisefekteistä ja esineistä, joita hahmo pitää mukanaan. Monitasoisessa spritessä hahmo on omassa tasossaan erillään efekteistä ja esineistä. Jopa hahmo itsekin voi koostua useammasta tasosta. Esineillä on omat animaationsa ja hahmolla omat. Tämä mahdollistaa animaatioiden ajoittamisen aikajanalla taso kerrallaan. Kuvan 17 hahmo on koottu monesta eri osasta, jotta animaatiota voi hallita taso kerrallaan.



Kuva 17. Hahmon osat Spriter-ohjelmassa [41].

Tween-animaatiota voi tehdä esimerkiksi Adobe Flashilla tai Toon Boomilla. Sitä voi tehdä vektoreilla tai bittikartoilla, mutta vektoreita käytetään usein, koska Flashissa on paljon vektorityökaluja. Flashissa on helppo animoida esimerkiksi pallo, joka hyppii pystysuunnassa ja litistyy iskiessään maahan. Tämä ei vaadi frame-by-frame-animaatiota. Animaation voi tehdä yksinkertaisesti liikuttamalla palloa ja luomalla aikajalle avainkehykset, jotka määrittävät, mistä mihin pallo liikkuu. Litistyminen onnistuu

skaalaamalla palloa korkeussuunnassa. Kaksiulotteisen pallon pyörimistä on kuitenkin hankala simuloida ilman frame-by-frame-animaatiota. Kehykset voidaan piirtää suoraan Flashissa vektoreina, tai ne voidaan piirtää maalausohjelmassa bittikarttoina ja tuoda Flashiin yhdistettäväksi aikaisempaan pomppimisanimaatioon.

Flashissa ja muissa samankaltaisissa ohjelmissa on yleensä luu-työkalu, jonka avulla voi animoida hahmon raajoja erikseen. Luu-työkalua, tween-animaatiota ja perinteistä animaatiota yhdistelemällä voi saada ammattimaisen lopputuloksen. Uusissa Rayman-peleissä hyödynnetään näitä tekniikoita. Työprosessi on kuitenkin monimutkainen, koska kaikki erikseen animoitavat osat, kuten raajat tai kasvot, on piirrettävä omalle tasolle. Kuvassa 18 on Rayman-hahmo pilkottuna osiin. Jokaista osaa animoidaan erikseen. [42.]

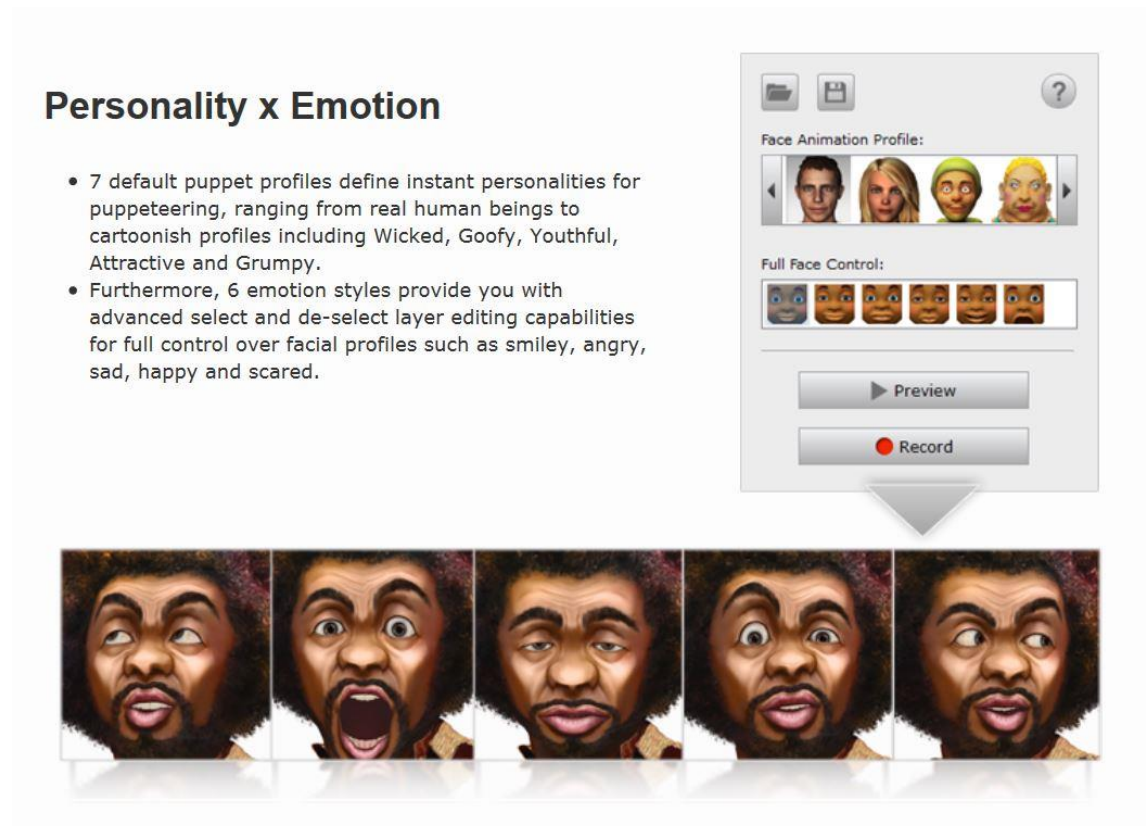


Kuva 18. Rayman-hahmon erikseen animoitavat osat [42].

Visual novel -lajin pelit ovat ääriesimerkki animen ja pelin risteymästä. Niissä on teksti pääosassa eikä animaatiota juurikaan näe, ellei laske mukaan hahmojen asentojen vaihtelua. Monet pelit ovat soveltaneet visual novel -pelien tyyppistä keskustelua ja vieneet sitä eteenpäin. Trinity universe -roolipelissä keskusteluja koristavat hahmojen pienet liikehdinnät ja puheanimaatiot. Animaatioiden tarkoitus on tuoda hahmoja eloon.

Trinity universen tyyppistä animaatiota näkee usein. Vanillaware käyttää peleissään samantyyppistä tekniikka. Peliohjaaja ja taiteilija George Kamitani kuvailee mielenkiintoista animaatiotekniikkaansa tebineriksi, joka toteutetaan Flash-tyyppisellä ohjelmalla

[43]. Siinä 2D-hahmo saadaan näyttämään melkein 3D:ltä. Tällaisen animaation toteuttamiseen on monia työkaluja. Animaatio on toteutettu ensiksi piirtämällä valmis hahmo ja sitten muokkaamalla hahmoa liikkeen imitoimiseksi. Photoshopissa kuvaa voi muokata muuntotyökalulla, nukketeatteri-käyritysmuuntotyökalulla tai sulata-suotimen avulla. Crazytalk-ohjelmassa (kuva 19) on face puppet -niminen työkalu, jolla kaksiulotteisia kasvoja voi muokata ja vääristellä realistisesti. Sillä on helppo animoida valmiiden tunnetyyliä avulla. Lopputulos vaikuttaa 3D:ltä, vaikka pohjana on 2D-kuva. [43; 44.]

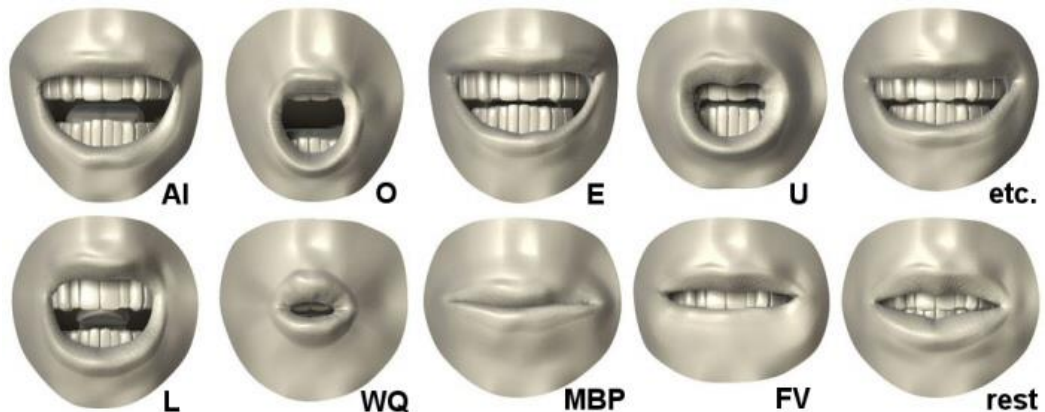


Kuva 19. CrazyTalk-ohjelman mahdollistamia ilmeitä [45].

Huulisynkronointi

Jos äänien tuottaminen on pelin esituontantovaiheessa, ne voivat inspiroida graafikon tyyliä. Kaikista äänistä graafikon ei kuitenkaan tarvitse huolehtia. Pelihahmon askeleen äänet voidaan tuoda peliin koodilla, joten animaattorin ei tarvitse tahdittaa askelia. Huulisynkronointi on luultavimmin yksi harvoista tilanteista, joissa graafikon pitää olla tekemisissä äänten kanssa. Huulisynkronointi tarkoittaa ääniraidan ja puheanimaation yhdistämistä. Synkronoinnin voi tehdä automaattisesti ja manuaalisesti.

Ennen synkronointia kannattaa omaksua ääniraita kunnolla, jotta eläytyisi hahmoon. Itse synkronointi aloitetaan jakamalla ääniraita vokaaleihin ja konsonantteihin, jotka vastaavat foneemeja. Kuvassa 20 ovat Breston Blairin esimerkkifoneemit. Huulisynkronoinnissa foneemiksi kutsutaan kirjainta vastaavaa suun asentoa. Asennot voi yksinkertaistaa aluksi neljään: avoimeen ja suljettuun sekä laajaan ja kapeaan suun asentoon. Tästä jatketaan piirtämällä yksityiskohtia animaatioon, jotta animaatio ei tuntuisi tasapaksulta. [46.]



Kuva 20. Breston Blairin foneemit [47].

Automaattinen synkronointi voi olla aiheellista, jos laadulla ei ole niinkään väliä ja äänimateriaalia on paljon. Automaattinen synkronointi ei ole täydellistä, mutta tarpeen mukaan synkronointia voi parannella manuaalisesti. Synkronoinnin voi tehdä automaattisesti esimerkiksi Papagayo-ohjelmalla. Se osaa liittää foneemit arvioiduille kohdille aikajanalla, mutta vaatii jonkin verran manuaalista kohdistamista. Papagayosta viedään äänitiedosto foneemeineen dat-tiedostona. Animointia voi jatkaa sitten toisessa ohjelmassa, kuten After Effectsissä. [46; 48.]

6 Toy Party

Toy Party -projekti alkoi Game Cluster -hankkeen aloituksesta. Haluttiin tehdä pelejä edustamaan Metropolia Ammattikorkeakoulun pelituotantoa. Projektin tavoite oli antaa opiskelijoille kokemusta pelien teosta. Työskentely tapahtui oppilaitoksen tiloissa. Projekti tehtiin yhteistyössä muiden Metropolian projektien kanssa, ja sen mekaniikkaa sovellettiin asiakasprojektina Mimmit-pelissä. Peli tehtiin Unity 3D:llä. Visuaalisesti peli näyttää 2D:ltä, mutta Unity 3D:n 3D-ominaisuuksia hyödynnettiin myös. Unity 3D:n 2D-

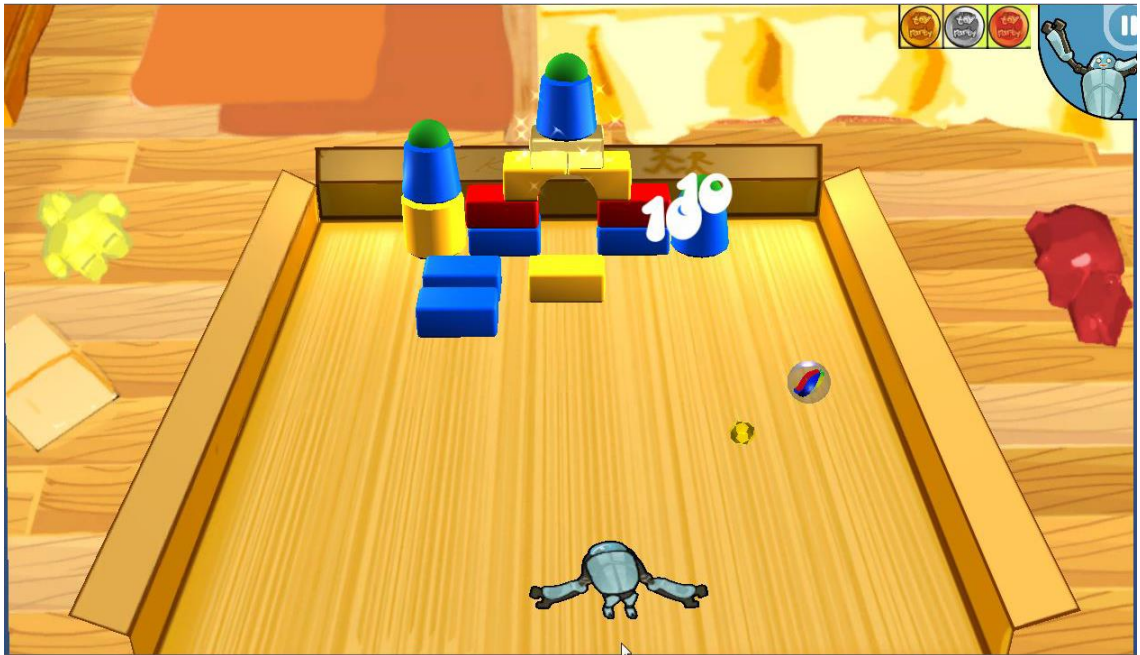
työkalut julkaistiin pari kuukautta projektin päättymisen jälkeen, joten niitä ei valitettavasti käytetty projektissa. Peligraafikoita oli kaksi, koodaajia viisi ja pelisuunnittelijoita yksi. Jotkut projektilaisista tekivät opetusmateriaalia muille opiskelijoille. Jotkut tekivät lopputyötään projektista. Omaan ensisijaisena tehtävänäni oli tuottaa grafiikkaa toisen graafikon kanssa, mutta pääsin tekemään myös paljon pelisuunnittelua.

Pelin teema kehittyi alkupuoliskolla. Ennen kuin pelin nimeä oli keksitty, tiedettiin jo, että haluttiin lapsille suunnattu pelikokoelma. Pelejä oli loppujen lopuksi kuusi yhden pelin sisällä. Lelut osoittautuivat toimivaksi teemaksi, joten päädyin suunnittelemaan leluhahmoja, jotka tekevät erinäisiä asioita lastenhuoneessa. Jokaisella pelillä oli oma päähahmonsa ja pieni tarina. Ennen jokaisen pelin alkua näytetään pelaajalle kahden ruudun mittainen sarjakuva, jossa selitetään pelin säännöt tarinanomaisesti. Mekaniikka ja pelilajityyppi vaihteli peleittäin. Peligraafikkona, ja ennen kaikkea suunnittelijana, sain luoda tarinan ja hahmot peliin, sekä olla mukana suunnittelemassa pelin mekaniikkaa.

6.1 Toy Party -pelit

Brick

Brick-pelissä lelurobotti potkaisee marmoripalloa. Pallo iskeytyy palikkaan, jonka alta paljastuu aarre, kuten kuvassa 21. Robotti alkaa rikkoa palikoita innoissaan.



Kuva 21. Brick-peli.

Brick on yläviistosta kuvattu arkanoid-tyyppinen peli. Se on ainoa peli, jossa perspektiivi on erotettavissa. Tavoitteena on kerätä paljon pisteitä rikkomalla lelupalikoita marmoripallolla. Pelissä liikutaan sivusuunnassa iskien palloa, joka kimpoaa palikoihin rikkoen ne. Palikoiden alta on maagisia tähtiä, jotka antavat erikoisvoimia. Tähdet joko helpottavat tai vaikeuttavat peliä, kun niitä kerää. Robotti voi esimerkiksi laajentaa käntensä tähdillä, jolloin marmoripalloon on helpompi osua, tai kasvattaa aseensa selästään ja ampua palikoita.

Flight

Pehmolelut ovat valtaamassa pahvilohikäärmeen huonetta. Lohikäärmeen on puolustauduttava (kuva 22.)



Kuva 22. Flight-peli.

Flight on sivusta kuvattu ampumispeli. Pahvilohikäärme ampuu pöllö- ja orava-pehmolelujä paperitulipalloilla. Pehmolelut koettavat iskeytyä lohikäärmettä päin tai ampuä sitä kohti. Pehmolelut putoavat, kun niihin osuu. Lohikäärmeen on voitettava pehmolelujen pomo, joka pikku hiljaa menettää osan aluksestaan, kun sitä ampuu.

Grab

Räsynukke huomaa lohikäärmeen alasammuttuja pehmoleluja ja juoksee pelastamaan niitä (kuva 23).



Kuva 23. Grab-peli.

Grab on sivusta kuvattu keräilyveli, jossa pehmoleluja putoaa ylhäältä alas tietty määrä. Pelaajan eli nuken täytyy pelastaa pehmoleluja hatullaan liikkumalla niiden alle. Pehmolelujen joukossa on myös vaarallisia esineitä, kuten silitysrautoja, joita pitää vältellä.

Maze

Mekaaninen hiiri seikkailee kolosta koloon etsien juustoa (kuva 24).



Kuva 24. Maze-peli.

Maze on ylhäältä kuvattu keräilypele. Hiirtä ohjataan kallistamalla pelilaitetta. Jos laite ei tue kallistusta, hiirtä voi liikuttaa myös suuntanäppäimillä. Hiiri kerää juustoa saadakseen paremman pistemäärän. Juustoja on erilaatuisia. Laatu kertoo juustosta saatavista pisteistä. Koloista pääsee kentän toisiin osiin. Kentällä on ansoja ja lammikoita, jotka haittaavat hiiren kulkua. Lopuksi hiiren on löydettävä kotiin.

Flip

Pehmolelukissa pelaa muistikorttipeliä (kuva 25).



Kuva 25. Flip-peli.

Flip on ylhäältäpäin kuvattu muistikorttipeli. Peli toimii kuin muistipeli, jossa pitää löytää parit. Pelin alkaessa on kolme sekuntia aikaa katsoa kortteja. Korteissa on pelin hahmojen kuvitukset, ja jokaisella kortilla on oma erottuva värinsä.

Roll

Taikuriolio löytää karkin, mutta se on korkealla lipaston päällä. Olio käyttää taikavoimiinsa pudottaakseen makeisen alas (kuva 26).



Kuva 26. Roll-peli.

Roll on fysiikkapeli, jossa piirretään karkista tehtyä taikaviivaa kentälle. Viivan tarkoitus on johdattaa pudonnut makeinen taikurioliota kohden. Kun makeinen on tarpeeksi lähellä, olio syö sen. Leijailevista tähdistä saa pisteitä, jos makeinen osuu niihin matkalla.

6.2 Toteutus

Nimikkeeni peliprojektissa oli peligraafikko, mutta tein myös paljon suunnittelua. Halusin yhdistää peleissä 2D:tä ja 3D:tä, joten päädyin tekemään kumpaakin. Päätehtäväni oli hahmojen, UI:n ja samalla koko ilmeen luominen, mutta tehtäviä jaettiin aikarajojen mukaan. Projektin alussa tein pienimuotoisen tyyliohjeen artisteille. Siihen kuuluivat referenssi, paletin värit ja joitain tarkempia määritteitä. Ilme kehittyi pikku hiljaa, kun pelin tavoite selveni. Tyyli yritettiin pitää mahdollisimman yhtenäisenä, joten työtehtävät jaettiin mahdollisimman selkeästi.

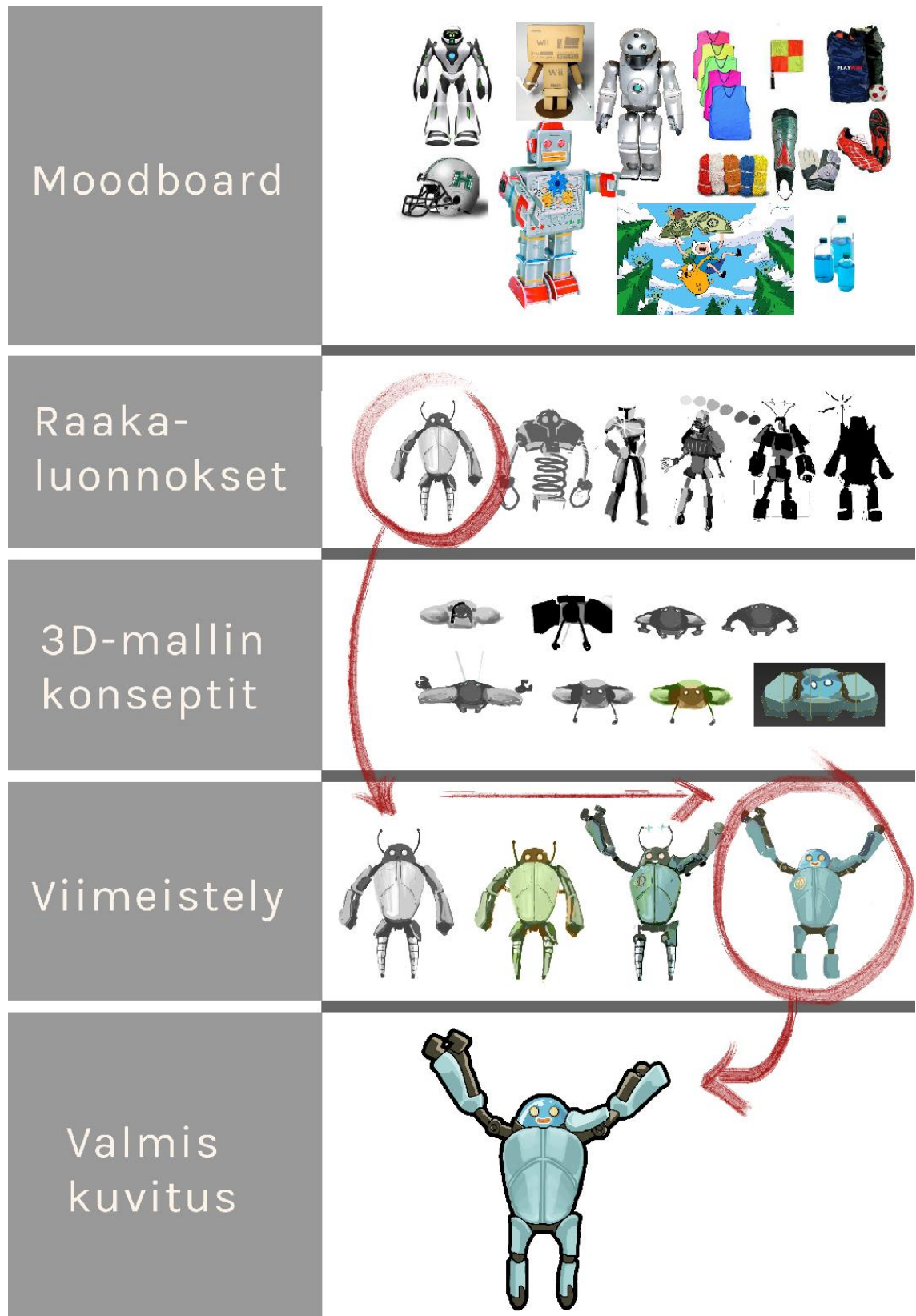
Hahmot

Luonnostelin ja maalasin hahmot (kuva 27) Photoshopilla. Hahmojen piti olla tunnistettavia ja helposti lähestyttäviä. Tein niistä myös hyvin erilaisia, jotta jokainen voi löytää pelistä hahmon, josta pitää. Aloitin aina raakaluonnoksella, jonka hyväksyin pelisuunnittelijalla. Raakaluonnoksesta tein hahmon kuvituksen. Kuvituksia käytettiin päävalikossa, aloitusruudussa, voittoruudussa, häviöruudussa, kunniainnatt-sivulla ja peliruudun HUD:ssa. Kuvituksesta sain hyvän pohjan seuraavaksi toteutettaville peliobjekteille. Hahmot piti myös animoida. Animaation tein frame-by-frame-animaatiolla Photoshopilla.



Kuva 27. Toy Partyn hahmot.

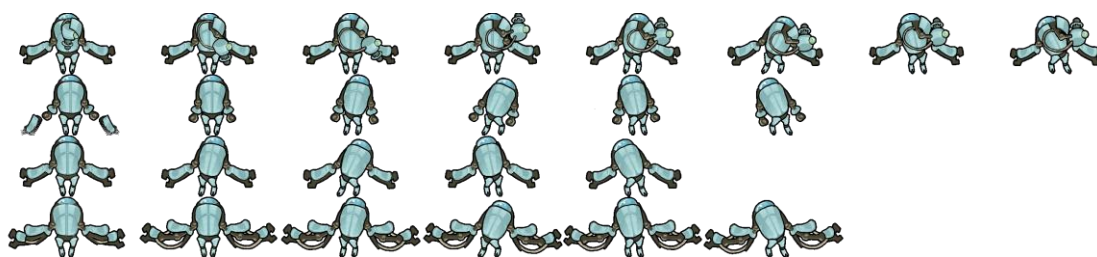
Kuva 28 näyttää robotin kehitysvaiheet. Aloitin hahmojen suunnittelun kokoamalla moodboardin useasta referenssikuvasta. Näin varmistin, että tiesin, mihin suuntaan veisin hahmoja. Jokaisen hahmon oli edustettava erilaista lelua ja sovittava pelimekaniikkaan. Hahmoilla oli myös pieni tarina takanaan, vaikka sitä ei päästy loppujen lopuksi ilmaisemaan kovinkaan näkyvästi. Tarina oli kuitenkin tärkeä, jotta hahmoille onnistuisi rakentamaan oikeanlaisen ilmapiirin. Tein muutaman raakaluonnoksen robotista, koska se oli hieman monimutkaisempi hahmo. Kun löysin hyvän siluetin, lähdin viimeistelemään sitä. 3D-mallin konseptit näyttävät, millainen robotista olisi tullut, jos se olisi ollut 3D:nä.



Kuva 28. Robotti-hahmon kehityksen vaiheet.

Brick-pelin kamerakulma oli yläviistosta, joten robotin sijoittaminen vaati säätämistä. Aluksi aion tehdä siitä kuutiomaisen 3D-mallin. Tyyli ei olisi kuitenkaan vastannut vaadittua tyyliä kovinkaan hyvin. Hahmot päätettiin tehdä 2D:nä. Koska hahmot sijaitsivat 3D-ympäristössä, sijoitettiin hahmon piirros png-kuvatiedostona yhden polygonin pinnalle. Brick-pelin tapauksessa perspektiivi vaikutti polygoniin, joten jouduin venyttämään piirrosta Photoshopissa, jotta se näyttäisi luonnolliselta pelimoottorissa. Muissa peleissä ei ollut perpektiivistä kameraa.

Hahmojen animointi oli yksinkertaista, joten tein senkin Photoshopilla. Imitoin onion-skinning-tekniikkaa Photoshopissa muuttamalla animoitavan kehyksen edeltävän ja jälkeisen tason läpinäkyvyyttä, koska Photoshopin CS5-versiossa ei ole frame-by-frame-animaatiolle omaa onion skinning -toimintoa. Sana tulee onionskinistä, joka on läpikuultavaa paperia. Sen avulla pystyy piirtämään animaation seuraavan kehyksen niin, että näkee samalla alla olevan kehyksen paperin läpi. Tekniikkaa käytetään myös digitaalisessa animoinnissa. Onion skinning -tekniikkaa käyttämällä sain helposti aikaan sulavan animaation. Sen jälkeen suoritin Photoshopissa koodin, joka muutti kehykset automaattisesti siistiksi Spritesheet-kuvaksi (kuva 29). Koodissa pystyi helposti määrittelemään, kuinka monta saraketta ja riviä kuvassa oli. Flight-pelin animaatio oli toteutettu monitasoisesti, mutta muiden pelien animaatio oli pelkästään yhdessä spritesheetissä. Useampitasoinen animaatio mahdollisti esimerkiksi pelin päävihollisen aluksen tuhoutumisen pala kerrallaan. [35.]



Kuva 29. Robotti-hahmon spritesheet.

Pelikenttä

Pelien taustatasolla oli ympäristökuva. Maze-peli oli kuitenkin poikkeus. Sen graafinen projektio oli ortografisesti ylhäältäpäin kuvattu. Siksi se näytti visuaalisesti 2D:ltä, mutta se oli todellisuudessa 3D:tä. Toinen graafikko teki kaikki paitsi yhden taustan. Olin mukana antamassa palautetta ja ideoita niihin. Annoin myös kenttäsuunnitteluun ideoita,

jotka koodajat toteuttivat. Itse tein korttipelin taustan. Kolmen pelin taustat esirenderöitiin 3D:nä, ja sitten maalattiin päälle. Flight-pelin tausta oli lisäksi toistuva panoraamakuva.

Kenttien keskitaso oli hieman erilainen riippuen pelistä. Brick-pelissä oli käytettävä 3D:tä palikoihin ja marmoripalloon, jotta pystyi hyödyntämään Unity 3D:n fysiikkamootoria. Muiden pelien keskitasossa oli pääosin kaksiulotteisia hahmoja, esineitä tai muita peliobjekteja.

Pelien etutasossa oli HUD-valikko, jossa oli pistepalkki ja hahmon muotokuva, joka oli samalla linkki pause-valikkoon. HUD vaihteli hieman peleittäin, mutta muotokuva koettiin pitää tyyliään samana. Roll-pelissä ei ollut pistepalkkia. Siinä oli lisäksi käytettäviä esineitä, jotka saattoi valita HUD:sta. Joissakin peleissä jäljellä oleva aika osoitettiin tiimalasilla.

Graafinen käyttöliittymä

Valikoiden suunnittelu ja toteutus oli minun tehtävänäni. Toinen graafikko auttoi silti joissakin nappuloissa. Pelin käyttöliittymä koostui aloitusruudusta, päävalikosta, kunniamaininnat-sivusta, opetusruudusta, HUD:sta, voittoruudusta ja häviöruudusta. Valikot koostuivat taustakuvasta, nappuloista ja kuvakkeista.

Etusivu on pelaajan ensimmäinen kokemus pelistä, joten sen oli osuttava kohderyhmään. Värisomittelusta tehtiin useita versioita. Opetusruudussa oli kuvan 30 kaltainen lyhyt sarjakuva, joka opetti tarinan avulla, kuinka peli toimii. Jokaisen pelin opetusruutu oli sommiteltu samalla tavalla. Vasemmalla oli pelin hahmon kuvitus. Oikealla oli kaksiruutuinen sarjakuva. Ruutujen sivussa oli puhekuplassa kuvake, joka selitti, miten pelata. Peli haluttiin pitää yksinkertaisena, jotta nuoret lapsetkin pystyisivät pelaamaan sitä. Melkein kaikki tekstit karsittiin pois. Kuvakkeet ja nappuloiden symbolit suunniteltiin helposti ymmärrettäviksi. Tekstiä ilmeni vain parissa paikkaa: pelin nimessä etusivulla, kunniamaininnat-sivulla, voittoruudussa ja häviöruudussa.



Kuva 30. Flip-pelin opetusruutu ennen pelin alkua.

Pelisuunnittelu

Oli hetkiä, jolloin huomasin olevani vastuussa kaikista senhetkisistä suunnittelun haasteista, koska pääsuunnittelija oli toisaalla ja toinen graafikko oli uppoutunut tärkeämpään työkuvaansa eli grafiikan tuottamiseen. Siksi sainkin vaikuttaa paljon peliin. Pääsin pohtimaan pelin mekaniikkaa, ilmettä, teemaa ja teknistä toteutusta. Keksinkin myös pelille nimen Toy's Party, joka muuttui myöhemmin Toy Partyksi kiitos amerikkalaisen työkumppanin.

6.3 Ongelmat

Suurin ongelma oli sellaisen henkilön puute, joka tekisi pelkästään pelisuunnittelua. Projektipäällikkö eli pääsuunnittelija oli vastuussa monesta muustakin asiasta, joten suunnittelutehtävät jakautuivat meille graafikoille useasti. Ongelma tässä oli se, että meillä ei ollut tarpeeksi aikaa suunnitella peliä loppuun asti, kun piti tehdä grafiikkaa samalla. Graafikoille tärkeintä oli grafiikan luominen ennen aikarajaa. Pelisuunnittelun ja dokumentoinnin vajavuuden vuoksi peli kehittyi ja suunnitelmat muuttuivat kesken kehityksen. Kun huomattiin, ettei jokin asia toimi, tai ilmeni jokin uusi mullistava idea, joka haluttiin hyödyntää, peli kehittyi hitaammin. Dokumentointi alkoi kunnolla vasta

projektin puolivälissä, joten jopa pelin mekaniikkaa joutui suunnittelemaan uudestaan ohjeistusten muuttuessa.

Yhtenäisen tyyliin pitäminen onnistui hyvin, vaikka projektissa oli kaksi graafikkoa ja tyylit olivat hyvin erilaiset. Kulttuurierot ja lievä kielimuuri aiheutti erimielisyyksiä. Ongelman esiintyessä otettiin joko paperia tai asiaan liittyvän ohjelmiston esiin ja selvennettiin mielikuvaa visuaalisesti. Jos yhteisymmärrykseen ei päästy, haettiin päätös pääsuunnittelijalta. Tyylin yhtenäistämistä auttoi tehtävien jako. Hahmojen ei odoteta aina olevan tyyllisesti samanlaisia kuin taustojen, joten siihen saatiin vedettyä taktinen raja.

Taiteilijana olen tottunut pitämään silmällä väriharmoniaa, joten alussa pirteiden värien käyttäminen oli hankalaa. Silmääni miellyttävät värit olivat monesti tylsiä muille. Korjasin ongelmat nopeasti jokapäiväisen palautteen ansiosta. Loppujen lopuksi värit eivät olleet välttämättä visuaalisesti hienostuneimpia, mutta ainakin niissä ilmeni leikkisyys, jota haluttiin peliin.

Sunnittelutehtävät vaikeuttivat grafiikan tekemistä, ja huomasinkin jatkuvasti, että seuraavaan aikarajaan oli kiire. Tästä seurasi, että UI:n nappulat olivat poikkeus työnkuluissa. Päädyin antamaan toiselle artistille joitain nappuloita tasapainottaakseni työtaakkaa. Tämän jälkeen ne tulivat takaisin minulle, kun työvastuut kääntyivät toisin päin seuraavassa aikarajassa. Ongelma tässä oli se, että meillä oli eri ohjelmat ja eri ruudukko (pohja, johon nappulat oli sijoitettu tasavälein). Aikaa hävisi, kun nappuloita joutui sijoittamaan uudelleen ruudukkoihin. Ongelmaa ei olisi ollut, jos olisi käytetty samaa ohjelmaa.

Tekniset ongelmat

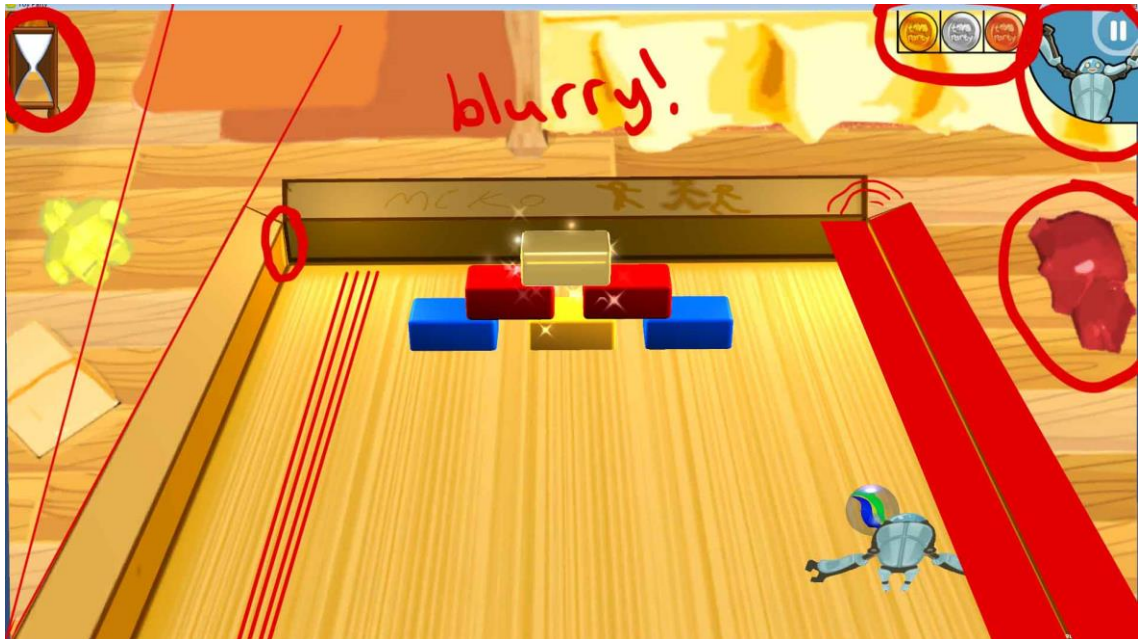
Jos en olisi käyttänyt tuttuja ohjelmia, olisin varmasti todistanut paljon enemmän teknisiä ongelmia. Huomasin tämän, kun toinen graafikko tuli jatkuvasti kysymään vastausta omiin ongelmiinsa. Varsinkin 3D-ohjelmat voivat aiheuttaa hankaluuksia opetteluvaiheessa.

Huomasin muutaman kerran, että työtiedostoni oli korruptoitunut, vaikka olin tallentanut tiedostot Photoshopilla. Ratkaisin ongelman osin palauttamalla toimivat versiot varmuuskopiosta. Testasin ongelmaa, mietin mahdollisuuksia ja hain apua helpdeskistä, jotta en joutuisi kamppailemaan ongelman kanssa enää. Ongelma ei ollut siinä, että olisin sammuttanut ohjelman ennen tallennusta. Mahdollisuuksia oli joko viallinen levytila tai verkon ongelmat. Kiintolevyn olisi voinut eheyttää. Eheyttäminen olisi vaatinut kirjallisen luvan helpdeskiltä, ja tämä olisi vienyt pitkän ajan, koska oli loma-aika. Ratkaisin ongelman siirtämällä projektitiedostoni henkilökohtaiselle kiintolevylle. Tällöin ei tarvinnut huolehtia levyn tai verkon ongelmista. Päädyin myös käyttämään varmuuskopiointia kiintolevyni kanssa.

Oppilaitoksen koneilla työskennellessä tuli vastaan ongelmia ohjelmien asennuksessa. 3DS Max -ohjelman fbx-vienti-lisäosa oli päivittämätön, eikä sitä voinut päivittää ilman helpdeskin apua. Helposti ratkaistava ongelma, mutta aikaa vievä.

Testaus

Edellisten päivien versiopäivityksien tarkastaminen osoittautui hyödylliseksi rutiiniksi. Ongelmakohtien karkottaminen mekaniikan ja grafiikan suhteen Trelloon oli hyvä ajatus, koska vastuulliset pystyivät helposti lukemaan ongelmasta sieltä. Otin usein pelistä kuvakaappauksen, johon merkitsin virheet selkeästi. Kuvassa 31 on kuvakaappaus Brick-pelin kentästä, johon on merkitty ongelmakohtia punaisella.



Kuva 31. Peliruudun ongelmien kartoittamista.

Peliä testattaessa huomasin monesti taustagrafiikan olleen epätarkkaa. Tähän oli pariakin syytä. Unity 3D rajoittaa kuvien kokoa pitääkseen pelin tiedostokoon pienenä. Tätä asetusta oli kuitenkin muutettava moneen kertaan. Koodaajat eivät kyenneet näkemään epätarkkuutta, joten minun oli koko ajan testattava peliä tämän ongelman suhteen. Toinen syy oli, että taustagrafiikka oli liian huonolaatuista eli liian pientä. Itselläni ei ollut tätä ongelmaa, koska olen tottunut tekemään kaiken vähintään kaksi kertaa suurempana. Ongelma ilmeni kuitenkin pari kertaa työkaverillani, ja siitä seurasi kaksinkertainen työmäärä, kun tausta piti tehdä melkein kokonaan uudestaan.

Koodaajat kohtasivat jatkuvasti ongelman, jossa kuvat olivat venyneitä. Peli rakennetaan usealle kuvasuhteelle, joten kuvat venyvät helposti. Ongelma ratkaistiin testaamalla versioita useaan kertaan, asettelemalla ne uudestaan Unityssä tai muuttamalla alkuperäistä sommittelua. Vaikka ongelmaa ei voitu kokonaan poistaa, sitä voitiin sentään tehdä siedettävämmiksi testaamalla ja korjaamalla.

Palaute

Palautetta sai projektilaisten lapsilta eli kohderyhmän jäseniltä. Palaute oli tärkeää ja se otettiin vakavasti. Hahmojen oli oltava pirteän värikkäitä ja huvittavia. Vaikeusastetta ei saanut aliarvioida. Lapset ovat yllättävän hyviä pelaajia, kunhan vain tietävät, mitä pitää tehdä. Pelistä tehtiin tarpeeksi haastava, mutta helposti ymmärrettävä. Toimiva

mekaniikka oli tärkein elementti pelissä, mutta hauskuus tuli heti seuraavaksi. Väriharmonia oli toisarvoinen verrattuna silmiin hyppäävälle värimyrskylle.

6.4 Työnkulku ja aikataulu

Kun aloitin projektin, kahdesta pelistä oli jo pelattava versio. Pelien tavoite ei kuitenkaan ollut vielä selvillä, eikä graafista ilmettä ollut mietitty. Tiedettiin kuitenkin, että pelin alustana olisi ensiksi PC ja myöhemmin iOS. Ennen kuin paperille alkoi ilmestyä muuta kuin tikku-ukkoja, kaikki ryhmäläiset istuivat alas suunnittelemaan peliä. Palavereita pidettiin pitkin viikkoa. Pelin suunnan ja ilmeen määrittäminen oli tärkeää alkuvaiheessa. Aluksi peliin kuului neljä minipeliä, mutta projektin edetessä pelejä tuli kaksi lisää ja kenttien lukumäärä moninkertaistui peliä kohden.

Projektipäälliköltä saatiin viikoittainen tavoite joka viikko. Työntekijät saivat oman tavoitteensa, kuten myös koko tiimi. Tein tehtävistäni todo-listan, jota päivitin sitä mukaa, kuin tehtävät valmistuivat.

Pelin ydintoiminnallisuudet ehdittiin toteuttaa, ja jokaisessa pelissä oli vähintään kolme pelattavaa kenttää. Graafikkona olisin halunnut tehdä viimeistellympää jälkeä. Alkuvaiheessa oli hankala nähdä, kuinka vähän aikaa pelin toteutukseen oli. Huomasin tämän projektin loppuvaiheella, joten pyrin viimeistelemään jotain piirroksia. Vaikka pelit olivat jokseenkin viimeistelemättömiä, projektin päätyttyä meillä oli valmis kokonaisuus.

Työtunnit ja työtehtävät merkittiin muistiin. Työtunteja oli noin 300. Ne jakautuivat seuraavasti:

- pelisuunnittelu: 70 tuntia
- grafiikan suunnittelu ja konseptointi: 110 tuntia
- valmiin grafiikan teko: 90 tuntia
- ongelmien ratkominen: 10 tuntia
- palaverit: 20 tuntia.

7 Yhteenveto

2D-pelit ovat keräytyneet mobiili- ja indie-pelimarkkinoille. Monia pelejä voi kutsua 2,5D-peleiksi, koska pelimoottorien ja muiden pelintekijän työkalujen ansiosta 3D:tä ja 2D:tä on helppo yhdistellä pelin kehityksessä. Pelien tekeminen ei ole kuitenkaan yhtä helppoa kuin niiden pelaaminen. Ennen kuin grafiikkaa voi tehdä, on ymmärrettävä, kuinka esimerkiksi graafiset projektiot toimivat tai mitä sovellusta käyttää mihinkin tehtävään. Peligrafiikan tuottamiseen on myös monia tekniikoita, jotka nopeuttavat työskentelyä. Aikarajat aiheuttavat vaihteluja työnteon kiirrellisyyden kannalta.

Vaikka graafikon työ on muista osaanottajista mahdollisesti itsenäisin, on hänen kuitenkin oltava perillä pelin tavoitteista ja noudatettava ohjeistuksia. Hahmojen luonti vaatii muun muassa ihmisen ulkonäöllisten piirteiden ja persoonallisuuden tuntemista. Animaatio on persoonallista, mutta yksinkertaista. Ilmeen on oltava helposti ymmärrettävä ja yhtenäinen. Piirustustaito ei itsessään riitä peligrafiikan tekoon. Graafikon on ymmärrettävä, missä formaatissa grafiikka tehdään ja kuinka se toimii peliympäristössä. Grafiikkaa tarvitaan 2D-peleissä muun muassa peliobjekteihin, efekteihin, graafiseen käyttöliittymään, konseptitaiteeseen ja kuvituksiin.

Toy Party -projekti, jonka osa insinöörityö oli, päättyi odotetusti valmiiseen peliin. Lopputuloksena oli useampaa peliä kokoava 2,5D-peli, jossa on värikäs ilme ja useita pelimekaniikkoja. Monia ongelmia saatiin ratkaistua testaamalla ja palautteen avulla. Projektilla oli selkeät aikarajat, ja jokaisella työntekijällä omat tehtävälistansa. Grafiikan luonnin kannalta oli tärkeää ottaa huomioon koodaajien senhetkinen tilanne, koska se oli liitoksissa pelin etenemiseen. Tiivis ryhmätyönteko nopeutti kehitystä ja mahdollisti ongelmien ratkomisen heti niiden ilmetessä.

Pelin kehittäminen opetti paljon pelin ja sen grafiikan suunnittelusta, kuten myös grafiikan viimeistelystä. Projekti ei pelkästään opettanut grafiikan tekoa, vaan myös yleisesti pelin kehittämistä Unity 3D -pelimoottorin avulla. Pelisuunnittelu, palaverit ja raportointi veivät aikaa sisällöntuottamiselta, mutta ne olivat pakollinen osa projektia. Peliä ei julkaistu, mutta sen päätavoite eli pelien teon opettaminen onnistui moitteettomasti.

Lähteet

- 1 Kokoelma Rayman-pelin arvosteluista. Verkkodokumentti. Metacritic. <<http://www.metacritic.com/game/xbox-360/rayman-legends/critic-reviews>>. Luettu 5.3.2014.
- 2 IndieGames. 2012. What AAA can learn from indies -- according to indies. <http://indiegames.com/2012/12/what_aaa_can_learn_from_indies.html>. 30.12.2012. Luettu 20.12.2013.
- 3 Incredipede-peli. 2012. Verkkodokumentti. Games.cz. <<http://games.tiscali.cz/incredipede-15200/galerie/225899?articleId=61249#title>>. Haettu 29.1.2014.
- 4 Scolastici, Claudio & Nolte, David. 2013. Mobile Game Design Essentials. Packt Publishing.
- 5 Ingram, Mathew. 2010. Average Social Gamer Is a 43-Year-Old Woman. Verkkodokumentti. <<http://gigaom.com/2010/02/17/average-social-gamer-is-a-43-year-old-woman/>>. 17.2.2010. Luettu 10.12.2013.
- 6 Superbrothers: Sword & Sworcery EP -peli. 2012. Verkkodokumentti. Steam. <<http://store.steampowered.com/app/204060/>>. Haettu 29.1.2014.
- 7 FEZ-peli. Verkkodokumentti. Polytron. <<http://polytroncorporation.com/what-is-fez>>. Haettu 3.2.2014.
- 8 Bédard, Renaud. 2007. Behind Fez: Trixels (and why we don't just say voxels). Verkkodokumentti. <<http://theinstructionlimit.com/behind-fez-trixels-and-why-we-dont-just-say-voxels>>. 11.10.2007. Luettu 1.12.2013.
- 9 A laymans guide to projection in videogames. Verkkodokumentti. Significant bits. <<http://www.significant-bits.com/a-laymans-guide-to-projection-in-videogames>>. Luettu 10.12.2013.
- 10 Perspektiivi. Verkkodokumentti. Buzzle. <<http://www.buzzle.com/articles/types-of-perspective-drawings.html>>. Haettu 10.12.2013.
- 11 Sascha. 2009. RPG Design: Choosing the right Graphical Projection. Verkkodokumentti. <<http://blog.hexagonstar.com/rpg-design-graphical-projection/>>. Luettu 10.12.2013.
- 12 Don't starve -peli. 2012. Verkkodokumentti. Don't starve. <<http://www.dontstarvegame.com/>>. Haettu 20.1.2014.

- 13 Using Perspective to Convey Depth in 2D Games. 2012. Verkkodokumentti. Purple pwny studios.
<http://purplepwny.com/blog/using_perspective_to_convey_depth_in_2D_games.html>. 16.7.2012. Luettu 10.12.2013.
- 14 Yhdensuuntais projektio. Verkkodokumentti. Wikipedia.
<http://en.wikipedia.org/wiki/Parallel_projection>. Luettu 10.12.2013.
- 15 Aksonometrinen projektio. Verkkodokumentti. Wikipedia.
<http://en.wikipedia.org/wiki/Axonometric_projection>. Luettu 10.12.2013.
- 16 Isometrisen grafiikan käyttö peleissä. Verkkodokumentti. Wikipedia.
<http://en.wikipedia.org/wiki/Isometric_graphics_in_video_games_and_pixel_art>. Luettu 10.12.2013.
- 17 Fallout-peli. 2013. Verkkodokumentti. Wordpress.
<<http://daveden.wordpress.com/2013/08/13/how-to-enjoy-fallout/>>. Haettu 10.12.2013.
- 18 Final Fantasy Tactics Advance -peli. 2009. Verkkodokumentti. Giantbomb.
<<http://www.giantbomb.com/profile/bog/blog/bogs-top-30-games-of-all-time-prepare-to-disagree-/23446/>>. Haettu 10.12.2013.
- 19 Heather, Maxwell Chandler. 2013. The Game Production Handbook. 3rd Edition. Jones & Bartlett Learning.
- 20 Zhui, Feng. 2013. EPISODE 69 - Production Pitch. Verkkodokumentti. Youtube.
<<http://www.youtube.com/watch?v=905ct2EEbdI>>. Katsottu 5.12.2013.
- 21 Pardew, Les. 2005. Beginning Illustration and Storyboarding for Games. Course Technology PTR.
- 22 Zana, Eytan. 2010. Raakaluonnokset. Verkkodokumentti.
<<http://cghub.com/images/view/90008/>>. Haettu 3.2.2014.
- 23 Pihlaja, Toni. 2010. Tietokonepelin introanimaation tuotanto. Opinnäytetyö. Metropolia Ammattikorkeakoulu.
- 24 Cartwright, Steve. 2012. Pre-Production Planning for Video, Film, and Multimedia. Focal Press.
- 25 Jew, Anson. 2013. Professional Storyboarding. Focal Press.
- 26 Kohr, Matt. 2011. Custom Keyboard Shortcuts. Verkkodokumentti. Video.
<<http://ctrlpaint.com/videos/custom-keyboard-shortcuts>>. Katsottu 12.1.2014.

- 27 Kohr, Matt. 2011. Ergonomics and Hardware. Verkkodokumentti. <<http://ctrlpaint.com/videos/ergonomics-and-hardware>>. Katsottu 12.1.2014.
- 28 Brown, Michael. 2010. White Paper: LCD Technologies Explained and Compared. Verkkodokumentti. <http://www.maximumpc.com/article/features/white_paper_lcd_technologies_compared>. 4.2.2010. Luettu 29.1.2014.
- 29 Nienhuys, Han-Kwang. 2008. Kontrasti-testi. Verkkodokumentti. <<http://www.lagom.nl/lcd-test/contrast.php>>. Luettu 29.1.2014.
- 30 Lista pelimoottoreista. Verkkodokumentti. Wikipedia. <http://en.wikipedia.org/wiki/List_of_game_engines>. Luettu 21.12.2013.
- 31 Adobe Flashin Generate Sprite Sheet -työkalu. Verkkodokumentti. Adobe. <<http://www.adobe.com/devnet/flash/articles/using-sprite-sheet-generator.html>>. Haettu 25.1.2014.
- 32 Zhui, Feng. 2012. EPISODE 52 - Visual Library. Verkkodokumentti. Youtube. <<http://www.youtube.com/watch?v=dnfIBERf2zM>>. Katsottu 20.12.2013.
- 33 Kohr, Matt. 2011. Ergonomics and Hardware. Verkkodokumentti. <<http://ctrlpaint.com/videos/how-to-draw-anything-in-3-steps>>. Katsottu 12.1.2014.
- 34 Zhui, Feng. 2012. EPISODE 38 Reference Books part1. Verkkodokumentti. Youtube. <<http://www.youtube.com/watch?v=aIUbgJuAxTY>>. Katsottu 20.12.2013.
- 35 Pardew, Les. 2006. Game Character Animation All in One. Course Technology PTR.
- 36 Tropien määrittely. Verkkodokumentti. TV Tropes. <<http://tvtropes.org/pmwiki/pmwiki.php/Main/HomePage>>. Luettu 20.12.2013.
- 37 DOTA 2 -pelin graafinen ohjeistus. 2012. Verkkodokumentti. Valve Corporation. <http://www.scribd.com/fullscreen/158071778?access_key=key-192ffm0cxqe6fgaeq3ch&allow_share=true&show_recommendations=false&view_mode=scroll>. Luettu 20.11.2013.
- 38 Skullgirls-peli. 2013. Verkkodokumentti. The Fighters Generation. <<http://www.fightersgeneration.com/games/skullgirls.html>>. Haettu 10.1.2014.
- 39 Lichtner, Christian. 2012. The art of Diablo 3. Verkkodokumentti. <<http://gdcvault.com/play/1015306/The-Art-of-Diablo>>. Katsottu 10.12.2013.
- 40 Mitchell, Briar Lee. 2012. Game Design Essentials. Sybex.

- 41 Dearborn, Jonathan. Spriter-ohjelma. Verkkodokumentti.
<<http://www.dinomage.com/2012/10/spriter-api-for-c/>>. Haettu 20.12.2013.
- 42 Rayman Legend -pelin esittelyvideo. 2013. Verkkodokumentti. Youtube.
<<http://www.youtube.com/watch?v=y-chi097uV4>>. 31.7.2013. Katsottu 10.12.2013.
- 43 Sheffield, Brandon. 2009. King of 2D: Vanillaware's George Kamitani. Verkkodokumentti.
<http://www.gamasutra.com/view/feature/132486/king_of_2d_vanillawares_george.php?page=2>. 3.8.2009. Luettu 20.12.2013.
- 44 CrazyTalk 7.3 Tutorial - Custom Idle Motions for Web and iDevices. 2013. Verkkodokumentti. Youtube.
<http://www.youtube.com/watch?v=vqMDq7YSN_Y>. Luettu 3.1.2014.
- 45 CrazyTalk-ohjelma. Verkkodokumentti. Reallusion.
<http://www.reallusion.com/crazytalk/features_animation.aspx>. Haettu 3.1.2014.
- 46 How to make lip sync animation easy. 2009. Verkkodokumentti. Youtube.
<<http://www.youtube.com/watch?v=0fKjMvIP4CE>>. Katsottu 3.1.2014.
- 47 Chien, Kitty. Research. Verkkodokumentti.
<<http://www.sfu.ca/~kwc11/IAT445/Research.html>>. Haettu 3.1.2014.
- 48 Lango, Keith. 2001. Principles for lipsync animation. Verkkodokumentti.
<<http://www.keithlango.com/tutorials/old/lipSync.htm>>. Luettu 3.1.2014.