

2D- JA 3D-PELIGRAFIIKAN OMINAISUUDET

LAHDEN
AMMATTIKORKEAKOULU
Tekniikan ala
Mediatekniikan koulutusohjelma
Tekninen visualisointi
Opinnäytetyö AMK
Kevät 2015
Hanna Mansikka

Lahden ammattikorkeakoulu

Mediatekniikan koulutusohjelma

MANSIKKA, HANNA: 2D- ja 3D-peligrafiikan ominaisuudet

Teknisen visualisoinnin opinnäytetyö, 45 sivua

Kevät 2015

TIIVISTELMÄ

Opinnäytetyön tavoitteena on tutkia peleissä käytettävien graafisten elementtien ominaisuuksia, niiden etuja ja haittoja sekä käyttötarkoituksia. Tietokonegrafiikka on tärkeä ilmaisukeino, joka välittyy jokapäiväiseen elämään useiden eri välineiden kautta, erityisesti viihdeteollisuudessa erilaisina tehosteina ja animaatioina, mutta myös painettuna materiaalina ja havainnollistamassa informaatiota mitä erilaisimmissa yhteyksissä.

Peleissä käytettävä grafiikka valitaan yleensä käyttötarkoituksen perusteella: millaista informaatiota pelaajalle halutaan välittää ja kuinka monipuolisesti se on pelaajan tarkasteltavissa tai muuttuu pelitilanteiden vaihdellessa. 2D-grafiikkaa voidaan luoda vektorien tai pikseleiden avulla, joista molempien käytöllä on etunsa. Vektoripiirrolla voidaan luoda sarjakuvamaisia ja skaalattavia 2D-kuvia, kun taas bittikarttagrafiikka on laajemmin muokattavissa ja jäljittelee todellisuutta paremmin. Realistisin peliympäristö on tietysti kolmiulotteinen. 3D-grafiikalla saavutetaan realistisin lopputulos, mutta sen saavuttaminen vaatii runsaasti suunnittelua, mallintamista ja lopputuloksen renderöinti voi viedä paljon aikaa. Siksi myös 3D-ympäristöjen renderöinnissä oikaistaan ja renderöidään valmiiksi valoja ja varjoja peliympäristöön ja kappaleisiin, jolloin tietokoneella on vähemmän informaatiota laskettavana pelitilanteessa, joka muutoin renderöityy reaaliaikaisesti.

Peleiltä odotetaan nykyisin myös yhä enemmän vuorovaikutteisuutta pelaajan ja pelin välille, peliympäristön tulisi renderöityä reaaliaikaisesti ja esitysgrafiikan olla mahdollisimman laadukasta. Peliteollisuus onkin hyvin merkittävässä osassa ratkaisemassa tietokonegrafiikan esittämiseen liittyviä ongelmia ja kehittämässä graafista esittämistä yhä laadukkaammaksi.

Asiasanat: peligrafiikka, tietokonegrafiikka, projektio, konseptitaide, renderöinti

Lahti University of Applied Sciences

Degree Programme in Media Technology

MANSIKKA, HANNA:

2D and 3D graphics in games

Bachelor's Thesis in Visualization Engineering, 45 pages

Spring 2015

ABSTRACT

The thesis aims to examine graphic features which are used in computer games. There are many advantages and disadvantages in different graphic styles. The end use determines what graphic style should be used. Computer graphics is transmitted to our daily lives through various instruments, particularly in the entertainment industry as special effects and animations, infographics and illustrations in various contexts, and also as printed material.

Game graphics is used to provide information to the player: how the game environment appears to the player, and how the environment and objects adapt to the changes in the game situation. 2D graphics can be created with vectors or pixels, which both have their advantages. Vector graphics is more cartoon-like and can be scaled infinitely while pixel graphics is widely customizable and realistic. The most realistic game environment is three dimensional. The most realistic result can be achieved with 3D graphics, but it requires a lot of design and modelling to create a believable game environment and its characters. Rendering also takes time, depending on how detailed the 3D scene is. To reduce long rendering times, it is common to render some parts of lighting and shadows in advance. With some pre-rendered lighting and shadows, the computer needs to calculate less information when rendering in real time.

Games are expected to be interactive, render in real-time and graphics should be displayed with the highest possible quality. The game industry is developing at a fast pace, so it has an important role in the development of computer graphics and in solving related problems.

Key words: game graphics, computer graphics, projection, concept art, rendering

SISÄLLYS

1	JOHDANTO	1
2	TIETOKONEGRAFIikka	2
2.1	Tietokonegrafiikan esittäminen	2
2.2	Tietokonegrafiikan kehittyminen	2
2.3	Peligrafiikan varhaiset vaiheet	4
3	2D-GRAFIikka	7
3.1	Tietokonegrafiikka 2D:nä	7
3.2	Bittikarttagrafiikka	8
3.3	Vektorigrafiikka	9
3.4	2D-peleissä käytettäviä graafisia keinoja	11
3.4.1	Sprite-grafiikka	11
3.4.2	2D- ja 3D-grafiikoiden yhdistäminen	12
4	3D-GRAFIikka	14
4.1	Kolmiulotteisuus	14
4.2	3D-grafiikan käyttösovellukset	16
4.3	3D-grafiikka peleissä	16
4.3.1	3D-mallit	16
4.3.2	Interaktiivinen ja reaaliaikainen 3D	17
5	CONCEPT ART – KONSEPTITAIDE	19
6	3D-YMPÄRISTÖN PROSESSOINTI	21
6.1	Renderöinti	21
6.1.1	Renderöintiprosessi	21
6.1.2	Reaaliaikainen renderöinti	21
6.2	Renderöity kuva	22
6.2.1	Renderöidyn kuvan ominaisuudet	22
6.2.2	Shading - varjostus	23
6.2.3	Tekstuurit ja pinnanmuodot	23
6.2.4	Varjot	24
6.2.5	Valon käyttäytyminen	27
7	PELEISSÄ KÄYTETTÄVÄT TASOPROJEKTIOT	28
7.1	Paralleeliprojektio	28
7.1.1	Ortografinen projektio	28

7.1.2	Aksonometrinen projektio	29
7.1.3	Oblique-projektio	31
7.2	Perspektiiviprojektio	32
7.2.1	Perspektiiviprojektion muodostuminen	32
7.2.2	Fixed 3D	34
7.2.3	First person -perspektiivi	35
7.2.4	Second person -perspektiivi	37
7.2.5	Third person -perspektiivi	37
8	YHTEENVETO	39
	LÄHTEET	40

1 JOHDANTO

2D- ja 3D-grafiikka näyttäytyy jokapäiväisessä elämässä hyvin monin tavoin. Digitaalisia valokuvia, animaatioita, erilaisia 3D-malleja ja monia muita graafisia elementtejä käytetään tehostamaan infograafeja, mainoksia, pelejä ja muuta ihmisen havaitsemaa kuvamateriaalia. Tietokonegrafiikan kehitys vektorein esitetyistä käyristä aina kolmiulotteisiin peliympäristöihin on hyvin kiinnostava ja monipuolinen. Kaksiulotteinen kuva on paljon muutakin kuin esittävä piirros tai digitaalinen valokuva. Esimerkiksi 3D-mallien tekstuurit ovat kaksiulotteisia bittikarttakuvia, joten kolmiulotteinen kappalekaan ei toistu realistisesti ilman kaksiulotteista kuvaa.

Pelien graafisen ilmeen kehitys sisältää monia elementtejä, joita pelaaja ei välttämättä tule ajatelleeksi. Bittikarttagrafiikan avulla tuotetut animaatiot ja kolmiulotteisuuden ilmaisu projektioiden avulla, joita varhaisissa peleissä käytettiin, ovat edelleenkin tärkeitä pelien graafisen ilmeen luomisessa. Esimerkiksi retro-henkiset uudet pelit luodaan 3D-kappaleilla, mutta projektion avulla ympäristö ikään kuin "litistetään" muistuttamaan 80-luvun isometristä tietokonepeliä. Vaikka tietokoneiden muistikapasiteetti sallii yhä monimutkaisempien 3D-ympäristöjen toistamisen reaaliaikaisesti, vaatii peliympäristön toteutus edelleenkin joitain perinteisiä menetelmiä. Realistisuuteen pyrkivä esittäminen on yksi 3D-grafiikan haasteista, mutta erityisesti pelimaailmassa 3D-ympäristöistä halutaan myös täysin epätodellisia, mutta joissa valojen ja varjojen tulisi käyttäytyä todellisuutta vastaavasti.

Vaikka tietokonegrafiikka on kehittynyt muutaman vuosikymmenen aikana suurin harppauksin, on edelleen monia haasteita, joihin pyritään löytämään ratkaisuja. Peliteollisuus onkin hyvin keskeisessä roolissa löytämään ratkaisuja graafisen esittämisen ongelmiin.

2 TIETOKONEGRAFIikka

2.1 Tietokonegrafiikan esittäminen

Grafiikka-nimitys on pohjimmiltaan kreikkalainen ilmaisu “grafein”, joka tarkoittaa kaivertamista, piirtämistä tai kirjoittamista. Suomen kielessä siitä käytetään kuvataiteessa yleisnimitystä menetelmille, joissa painolevystä vedostetaan kuvia paperille tai muulle materiaalille. Englanniksi käsite grafiikka eli graphics on kuitenkin paljon laajempi, ja oikeastaan kaikki kuvalliseen ilmaisuun liittyvä kuuluu käsitteen alaisuuteen. (Taidegrafiikka 2003.)

Tietokonegrafiikka on tietokoneella tuotettua kuvallista informaatiota. Yksinkertaisuudessaan tietokonegrafiikan voidaan sanoa olevan päällekkäin sijoitettuja kuvia, joita tarkastellaan tietokoneen näytöltä. Tietokonegrafiikka voi olla kaksiulotteista tai kolmiulotteista. Varhaiset tietokoneet pystyivät käsittelemään vain kaksiulotteisia harmaasävyisiä grafiikoita. Tietokoneiden kehittyessä myös värien toistaminen kehittyi aina 16 väristä 256 väriin toistamiseen saakka. Nykyisin grafiikkaa pystytään toistamaan miljoonissa eri väreissä. (TechTerms.com 2009.)

Tietokonegrafiikan havainnointi ei rajoitu ainoastaan tietokoneella toteutettiin toimintoihin. Tietokoneella toteutetut kuvat näkyvät jokapäiväisessä elämässä televisiossa, säätiedotuksissa, lääkärissä visuaalisina ilmaisukeinoina sekä painettuna aineistona päivän lehdessä. Tarkoin valitulla graafisella esityksellä voidaan muuntaa muuten vaikeasti havainnoitavia esitystapoja, kuten kaaviota ja luetteloita, helpommin ymmärrettävään muotoon. Tietokonegrafiikka käsitteenä voidaan jakaa seuraaviin aihealueisiin: kaaviot, esitykset, piirtäminen ja suunnittelu, kuvankäsittely, tieteellinen kuvitus, kartoitustyökalut ja grafiikkakirjastot. (IT 2015.)

2.2 Tietokonegrafiikan kehittyminen

Ensimmäiset tietokonegrafiikan sovellukset kehitettiin Yhdysvalloissa Whirlwind-projektissa jo 1940-luvun lopulla ja 1950-luvulla. Projektissa oli tarkoitus kehittää riittävän nopeasti toimiva digitaalinen tietokone, jolla voi-

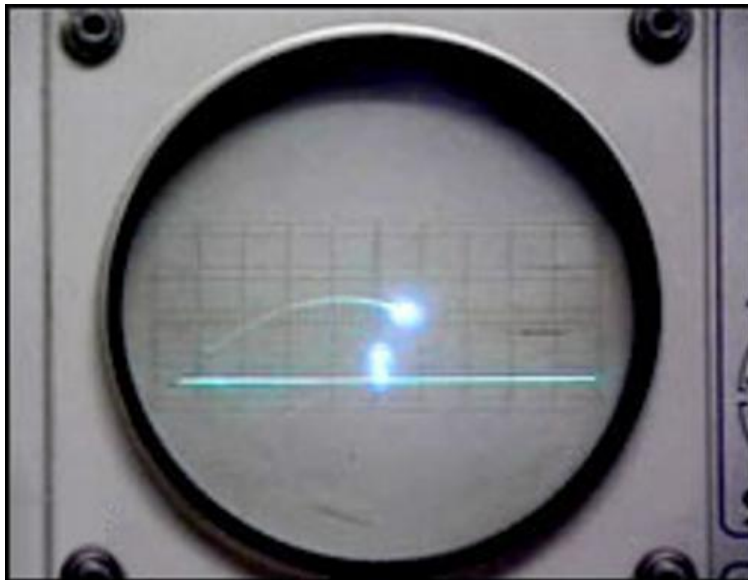
taisiin ohjata lentosimulaattoria. Lentosimulaattorin pohjalta kehitettiin valvontajärjestelmä, jonka tutkakuvat esitettiin käyttäjälle vektorigrafiikan avulla. Tietokonegrafiikka löysi tiensä myös viihdeteollisuuteen ensimmäisen kerran 1950-luvun lopulla. Alfred Hitchcockin Vertigo-elokuvassa avuksessa on vektorigrafiikalla luotuja geometrisia muotoja. Termiä "tietokonegrafiikka" käytti ensimmäisen kerran lentokoneiden suunnittelija William Fetter, kun hän tutki lentäjän sijoittamista koneen ohjaamoon rautalankamallin avulla. Samoihin aikoihin IBM kehitti ensimmäisen kaupallisen CAD-järjestelmän ja vuonna 1961 MIT:n (Massachusetts Institute of Technology) tiedemiehet ohjelmoivat PDP1-tietokoneelle Spacewar-pelin, ensimmäisen reaaliaikaisen graafisen tietokonepelin. (Puhakka 2008, 25 - 26.)

Vuonna 1963 MIT:n Ivan Sutherland esitteli Sketchpad-järjestelmän, joka mahdollisti interaktiivisen piirron vektorigrafiikkaan perustuvan näytön avulla, ja loi näin perustan nykyaikaiselle graafiselle käyttöliittymälle. 1970-luvulle asti vektorinäyttö oli tärkein tietokonegrafiikan näyttölaite, sillä se soveltui rautalankamallien esittämiseen. Vuonna 1970 saataville tuli jo yhden kilotavun kokonainen RAM-muistipiiri, joka mahdollisti kohtuuhintaisen videopuskuroinnin. Näytölle piirrettävä pikselikuva kirjoitetaan videopuskuriin ennen sen siirtämistä. Pikseleihin perustuvat rasterinäytöt korvasivat vektorinäytöt täysin, jolloin tietokonegrafiikka halpeni ja tuli yhä useampien käyttäjien ulottuville. (Puhakka 2008, 25 - 26.)

1970-luvulla ja 1980-luvun alussa kehitettiin useita yhä tänä päivänäkin käytössä olevia tekniikoita, joilla pyrittiin pintojen realistisempaan piirtoon valaistuksen ja sävytyksen avulla. Myös pintojen materiaalit, heijastukset ja teksturointi kiinnostivat tutkijoita. 1980-luvulla Autodesk-yhtiö toi markkinoille ensimmäisen PC:llä toimivan CAD-ohjelmiston ja tietokonegrafiikkaa alettiin käyttää teollisuudessa yhä laajemmin. 1982 perustettiin yritys, joka tunnetaan nykyisin nimellä SGI. SGI alkoi valmistaa erityisesti tietokonegrafiikkaa varten suunnattuja työasemia. Vuonna 1992 yhtiön kehittämän piirtorajapinnan pohjalta luotiin avoin OpenGL-piirtorajapintastandardi, joka on nykyisesti yleisesti käytössä oleva grafiikkakirjasto. (Puhakka 2008, 25 - 26.)

2.3 Peligrafiikan varhaiset vaiheet

Varhaiset tietokonepelit olivat hyvin yksinkertaisia ja ensimmäiset versiot olivat tekstipohjaisia. Vuonna 1958 William Higinbotham kehitti alkeellisen tennispelin Tennis for Two (kuva 1), jossa oli ensimmäisenä graafinen käyttöliittymä. Tennis for Two -peliä pelattiin kuitenkin oskilloskoopilla, joten on kiistanalaista onko se oikeastaan tietokonepeli. Tietokonepelien kehittyessä syntyi useita pelejä, jotka onnistuivat luomaan kiehtovan peliympäristön graafisesta yksinkertaisuudestaan huolimatta. Alkuaikoina pelien graafinen suunnittelu oli kuitenkin hyvin rajoittunutta, sillä käytettävissä oleva muisti oli vielä hyvin vähäistä. Suunnittelijoiden tulikin tasapainotella muistin käytössä graafisen laadun ja pelin nopean latautumisen välillä. (Morris & Hartas 2003, 10.)



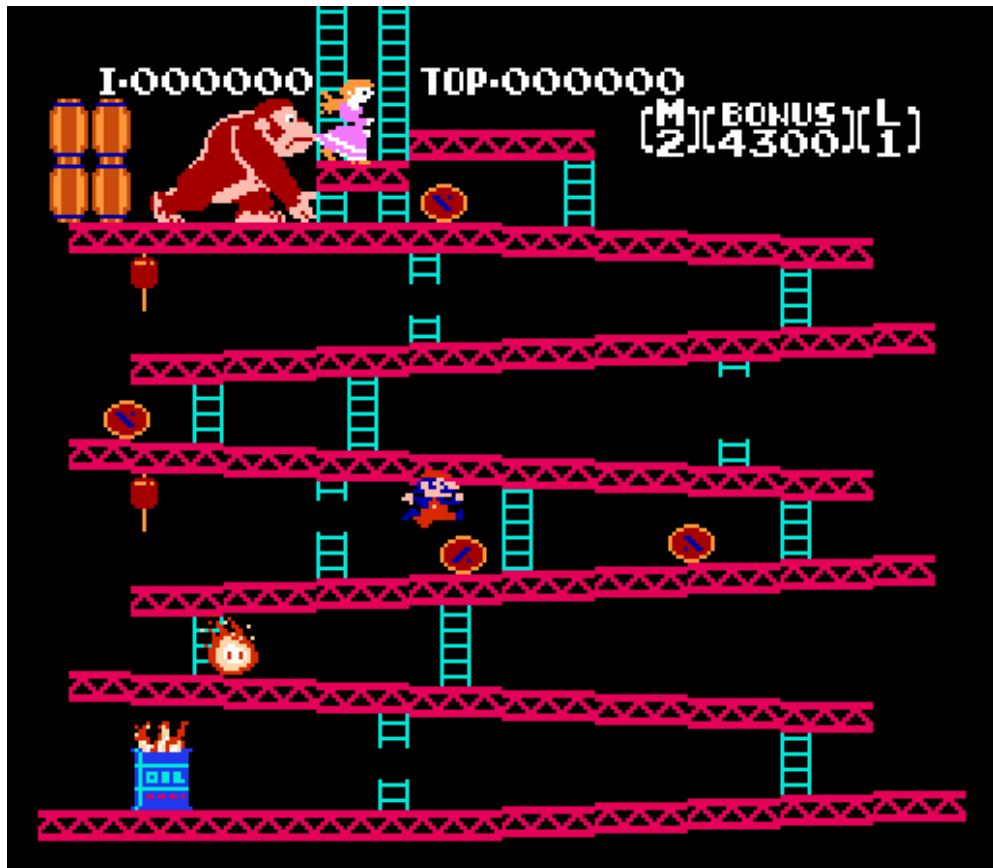
KUVA 1. Tennis for Two -peli (Pongmuseum.com 2015)

Peliteollisuudessa tapahtui tärkeä läpimurto vuonna 1977, kun Cinematronics julkaisi Spacewars-pelin. Spacewarsin graafinen käyttöliittymä käytti vektoreita, jotka toimivat pohjana nykyaikaiselle real-time 3D-grafiikalle. Tekniikan kehittyessä julkaistiin myös kolmiulotteisia pelejä. Esimerkiksi Vectorbeams-yhtiön julkaisema Speed Freak oli autopeli, jossa ulottuvuutena oli myös syvyys. Nykystandardien mukaan 1970- ja 1980-luvun pelit ovat karkeatekoisia, ne kuitenkin ovat tavoitelleet samoja standardeja kuin nykyisin tavoitellaan sekä luoneet pohjan koko pelikehitykselle. Vuoden

1977 alkusysäyksen jälkeen pelikehitys on ollut hyvin voimakasta ja peliteollisuudesta on tullut merkittävä ala myös taloudellisesti. (Morris & Hartas 2003, 12.)

Tietokonepelit yleistyivät 1980-luvulla, kun kotitietokoneet ja videopelikon-solit yleistyivät kotitalouksissa. Vielä tuolloin pelit perustuivat taustakuvan päällä liikkuviin 2D-hahmoihin eli "spriteihin". 1980-luvulla syntyi useita peliklassikoita, kuten Pac-Man ja Donkey Kong, josta esimerkki kuvassa 2. (Puhakka 2008, 27.)

Kehityksen myötä markkinoille saapui grafiikkaan keskittyneempiä pelikoneita, kuten Amiga ja Atari. Pelikoneissa laitteen muisti oli varattu vain yksittäisen pelin pelaamisen, joten niiden oli mahdollista näyttää monimutkaisempia graafisia muotoja ja useampia värejä. Pelikoneissa oli myös suuremmat näytöt, joten tavalliset kotitietokoneet jäivät pelikoneiden varjoon monissa grafiikan esittämismomenteissa. Tallennuskapasiteetin ja prosessoritehojen lisääntyessä pelien taustakuvista voitiin tehdä yksityiskohtaisempia, animaatioista parempia ja näin lisätä pelikokemuksen vaikuttavuutta. (Morris & Hartas 2003, 12.) Vuonna 1993 julkaistiin uudella tavalla 3D-grafiikkaa käyttänyt Doom-peli, jossa käytettiin realistisia tekstuureja. Doom avasi tien 3D-grafiikan laajemmalle käytölle tietokonepeleissä. (Puhakka 2008, 27.)



KUVA 2. Donkey Kong -peli vuodelta 1981 (Donkey-kong.info)

3 2D-GRAFIikka

3.1 Tietokonegrafiikka 2D:nä

Kaksiulotteinen grafiikka eli 2D-grafiikka on pääasiassa kaksiulotteisten kuvien, kuten geometrinen muotojen, tekstin ja digitaalisten kuvien esitystapa. 2D voi tarkoittaa tekniikkaa, jolla kuvia tarkastellaan, tai kuvan ominaisuutta. Kaksiulotteisuus-käsitteen hahmottaminen helpottuu, kun kuvitellaan kaksiulotteisessa tasossa oleva kuva käännettäväksi sivuttain, jolloin kappaleesta näkyisi pelkkä viiva eli suora. 2D-grafiikka kehittyi perinteisten piirustus- ja painomenetelmien pohjalta erilaisiin tarpeisiin, kuten teknisen piirtämisen ja markkinoinnin apuvälineeksi. 2D-grafiikka kehittyi 1950-luvulla vektorigrafiikkaa käyttävien näyttölaitteiden pohjalta, mutta paria vuosikymmentä myöhemmin rasterigrafiikka yleistyi ja syrjäytti vektorigrafiikan. (Wikipedia 2015a.)

2D-peleissä perspektiivillä voidaan luoda vaikutelmaa syvyydestä, vaikkei peliympäristö itsessään olisikaan kolmiulotteinen. Kaksiulotteisessa ympäristössä kuvataan kappaleita luomalla illuusio kolmiulotteisuudesta pelkän perspektiivin keinoin. 2D-peleissä yksinkertaisimmin kolmiulotteinen vaikutelma luodaan käyttämällä aksonometristä projektiota, joka tarkastelee kuvaa yläviistosta eikä pyri esittämään mittasuhteita todellisuutta vastavasti, vaan liioitellen syvyyden vaikutelmaa. (OpenGameArt.org 2015.)

2D-peleissä käytettävää perspektiiviä kutsutaan usein isometriseksi projektiksi. Projektilla tarkoitetaan kappaleen esittämistä kaksiulotteisessa tasossa. Isometrinen projektio on yksi kolmesta aksonometrisestä eli yhdensuuntaisesta projektioista. Sillä pyritään luomaan illuusio kolmiulotteisuudesta. Todellisuudessa isometrinen projektio on tarkkaan määritelty piirustustekniikka ja sitä usein virheellisesti käytetään kuvaamaan mitä tahansa yläviistosta tarkasteltavaa peliä. (Giantbomb 2014b.)

Ennen reaaliaikaisen 3D-grafiikan kehittymistä, useissa peleissä käytettiin aksonometristä projektiota, jolla pyrittiin luomaan kolmiulotteinen vaikutelma. Usein tällaisissa peleissä sanottiin olevan "isometrinen grafiikka",

vaikka tarkalleen ottaen kyseisissä peleissä käytettiin dimetristä projektioita, joka on myös yksi aksonometrisistä projektioista. (Giantbomb 2014b.) Peleissä käytettäviä projektioita käsitellään laajemmin luvussa 6.

3.2 Bittikarttagrafiikka

Bittikartta- eli rasterigrafiikka koostuu pikseleistä. Pikselit ovat ruudukossa sijaitsevia pieniä erivärisiä pisteitä, jotka yhdessä muodostavat näkyvän kuvan. Bittikarttakuvat ovat riippuvaisia näytön erottelutarkkuudesta eli resoluutiosta. Resoluutio yleisesti ilmaisee kuvassa olevien pikseleiden lukumäärän. Resoluution yksikköinä käytetään lyhenteitä dpi, joka ilmaisee pisteiden määrän tuumaa kohden, tai ppi, joka ilmaisee pikseleiden määrän tuumaa kohden. Esimerkiksi bittikarttakuvaa tulostettaessa tulee resoluution olla 150 - 300 ppi. (About Tech 2015a.)

Bittikarttojen ollessa riippuvaisia resoluutiosta on niiden kokoa vaikea muuttaa ilman, että kuvan tarkkuus eli laatu heikkenee. Kuvaa pienennettäessä kuvainformaatiota katoaa, mikä tarkoittaa käytännössä pikseleiden häviämistä. Kuvaa suurennettaessa ohjelmiston tulee luoda uusia pikseleitä kuvakoon kasvaessa. Ohjelmiston luodessa lisää pikseleitä. Se arvioi luotavien pikseleiden värin ympäröivien pikseleiden värin perusteella. (About Tech 2015a.)

Pikseleiden määrä on suorassa suhteessa kuvan terävyyteen. Kuvan koostuessa useammista pikseleistä tarkoittaa se myös kuvan sisältävän informaatiomäärän olevan suurempi. Suurempi informaatio tarkoittaa käytännössä parempaa tarkkuutta. Esimerkiksi HD- eli teräväpiirtotelevisio toistaa vaakasuuntaisesti 1920 pikseliä ja pystysuuntaisesti 1080 pikseliä, jolloin pikselien kokonaislukumäärä on 2 073 600 pikseliä. (Beane 2012, 57.)

Tulee ottaa huomioon, että tässä yhteydessä kuvan skaalaaminen on eri asia, kuin kuvan suurentaminen tai pienentäminen. Skaalaus tarkoittaa näytöllä tapahtuvaa kuvan koon muuttamista vetämällä kuvaa kulmista suuremmaksi. Skaalaus on väliaikainen toimenpide, kun taas kuvan suu-

rentaminen tai pienentäminen tallennetaan ja kuvassa olevan informaation määrä muuttuu. Skaalaus ei muuta kuvan pikseleiden määrää suuremmaksi tai pienemmäksi. Pienemmäksi skaalatessa bittikarttakuva näyttää usein selkeämmältä, mutta pikseleiden määrä kuvassa ei ole muuttunut. Kuvassa 3 näkyy selvästi, kuinka suuremmaksi skaalatessa kuvan pikselit suurenevat ja kuvasta tulee epätarkka. Suuremmaksi skaalattu kuva ja kuva, jonka kokoa on suurennettu ja lisätty pikseleitä, ovat erinäköisiä. (About Tech 2015a.) Tietokone tallentaa bittikarttakuvan tiedot matriisiin, jossa jokaisella pikselillä on väri- ja paikkatieto. Pikselin väri koostuu punaisen, vihreän ja sinisen sävyistä, esimerkiksi ensimmäinen piste on 120 punainen, 14 vihreä, 50 sininen ja 20 läpinäkyvä. (OpiGIMP 2013.)



KUVA 3. Osasuurennos bittikarttakuvasta

3.3 Vektorigrafiikka

Vektorikuvat koostuvat useista skaalattavissa olevista objekteista. Objektit määrittyvät matemaattisten yhtälöiden avulla, joten toisin kuin pikselit, vektoreilla piirretyt kuvat ovat aina samanlaatuisia. Vektorien muodostamiseen käytettävät objektit sisältävät viivoja, käyriä ja muita muotoja. Ne voivat sisältää erilaisia muokattavissa olevia attribuutteja, kuten täyttöväri ja ääriviivat. Attribuuttien muokkaus ei vaikuta objektiin. Objektin muokkaaminen voi tapahtua attribuutteja muokkaamalla, mutta itse objektia voi muotoilla muokkaamalla kappaleen muodon määrääviä pisteitä, esimer-

kiksi lisäämällä niitä ja muokkaamalla niiden olemusta. (About Tech 2015b.)

Vektorikuvat eivät ole riippuvaisia resoluutiosta, koska vektoreiden kokoa voi muuttaa vaikuttamatta lopulliseen kuvaan. Vektorikuvia voi suurentaa tai pienentää ilman että kuva muuttuisi epätarkaksi, kuten kuvan 4 osasuurenoksessa on esitetty. Kirjasintyypit eli fontit ovat myös eräänlaisia vektorikuvia. Vektorikuvien käytössä on useita hyviä puolia, mutta yksi suurin haitta on, ettei vektoreilla voi tuottaa fotorealista kuvitusta. Vektorikuvat sisältävät yleensä yksivärisiä alueita tai muutaman värin vivahteita, eikä niillä voi piirtää pieniä sävyeroja toisin kuin pikseleillä. Vektorikuvat vaikuttavat usein sarjakuvamaisilta värierojen ja vivahteiden puuttumisen vuoksi. Vektorigrafiikka kehittyy kuitenkin kaiken aikaa ja vektoripiirto-ohjelmilla voidaan luoda erilaisia varjostuksia ja läpinäkyvyyksiä, joita ei aiemmin vektoreilla piirtäessä pystytty saavuttamaan. Nykyisin on myös mahdollista lisätä vektoriobjekteihin tekstuureja, jotka tuovat fotorealisuuden tuntua. (About Tech 2015b.)

Vektorikuvat ovat aina tietokoneella tuotettuja; käsin piirrettyä kuvaa ei ole mahdollista skannata ja jäljittää kuvankäsittelyohjelmalla sitä täysin alkuperäistä vastaavaksi vektorikuvaksi. Vektorikuvat on myös hyvin yksinkertaista muuntaa eli rasteroida bittikarttakuviksi. Rasteroidessa tulee huomioida lopullisen bittikarttakuvan käyttökohde, sillä lopullisen kuvan resoluutio tulee valita ennen kuvan rasterointia. Alkuperäisen vektorikuvan säilyttäminen alkuperäisessä formaatissa on hyvin tärkeää, sillä useimmat kuvankäsittelyohjelmat muuntavat vektoritiedot rasteritiedoiksi, eikä rasteroitu kuvatiedosto muunnu takaisin vektoreiksi edes vektoripiirto-ohjelmaan avattaessa. (About Tech 2015b.) Vektorikuvaa tallentaessa tietokone käsittelee objektin tietoja pisteiden, kaarien ja muiden ominaisuuksien perusteella, esimerkiksi monikulmio, jonka kulmat sijaitsevat pisteissä $1x$, $1y$ ja $1x$, $2y$ ja $4x$, $1y$ ja $4x$, $2y$, musta reuna 3 pistettä, punainen täyttö. (Opi-GIMP 2013.)



KUVA 4. Osasuurennos vektorikuvasta

3.4 2D-peleissä käytettäviä graafisia keinoja

3.4.1 Sprite-grafiikka

Spritet ovat yksittäisiä bittikarttakuvia, joista bittikarttakuvalle tyypillinen tausta on maskattu pois. Spritesheetit ovat kokonaisuuksia, jotka koostuvat useista spriteistä. Spritesheetit voivat sisältää staattisia tai animaation sisällytettäviä kuvia. Esimerkiksi peleissä käytetään spritesheetejä, joihin on koottu bittikarttakuvia pelissä käytettävistä objekteista ja ikoneista. Erittäin erityisesti 1980- ja 1990-luvuilla oli tavallista käyttää spritesheetejä grafiikoiden sisällyttämiseksi peleihin. Esimerkki pelissä käytettävästä spritesheetistä kuvassa 5. (TechTerms.com 2012.)

Peliohjelmoijat viittasivat spriteihin lähdekoodissa ja määrittivät niiden käytölle erilaisia ominaisuuksia, kuten milloin halutun kuvan tuli näkyä ja min-käläinen vuorovaikutus kuvalla oli toisten spritejen kanssa. Jotkin pelit käyttävät edelleen 2D-spritejä, mutta pääsääntöisesti peleissä käytetään kaksiulotteiseksi muunnettuja 3D-polygoneja. Nykyisissä tietokoneissa ja pelikonsoleissa käytettävät 3D-videokortit renderöivät 3D-objekteja tehokkaammin kuin 2D-spritejä. (TechTerms.com 2012.)



KUVA 5. Donkey Kong -pelin spritesheet (Mario Universe 2015)

Spritesheeteillä pyritään nopeuttamaan kuvien näkyvyyttä: on tehokkaampaa noutaa kaikki kuvat eli spritet yhdestä kuvakokonaisuudesta. Spritesheetit koostuvat kahdesta osasta: frames eli ruudut sekä cycles eli kierrot. Ruutu on yksi yksittäinen kuva eli sprite. Spritesheetien avulla luodussa animaatiossa yksittäisistä ruuduista luodaan kierto, jolloin saadaan vaikutelma jatkuvasta liikkeestä, kuten kävely tai juoksu. Animaatiossa vaihdetaan spritesheetistä haettuja kuvia hyvin nopeasti peräjälkeen, jolloin näytettävässä animaatiossa syntyy illuusio liikkeestä. Tekniikka on hyvin samanlainen kuin filmiprojektorilla esitettävissä elokuvissa. (Tuts+ 2013.)

3.4.2 2D- ja 3D-grafiikoiden yhdistäminen

2.5D tarkoittaa “kaksi ja puoli-ulotteista”, josta voidaan käyttää asiansynteystä riippuen termejä $\frac{3}{4}$ perspektiivi ja pseudo-3D. 2.5D-termiä käytetään kuvaamaan hyvin monenlaisia 2D-projektioita, joiden esittämisessä yritetään jäljitellä kolmiulotteista vaikutelmaa. 2.5D ei kuitenkaan ole kolmiulotteista, muun muassa siksi, ettei ympäristössä olevia objekteja voi tarkastella niiden takapuolelta. 2.5D-termiä käytetään pääasiallisesti vain peleissä, joissa pelaajan liikkeet tai ympäristö on sidottu kaksiulotteiseen tasoon. (Wikipedia 2015b.)

2.5D-peleiksi voidaan kutsua pelejä, joissa graafinen kokonaisuus on luotu polygoneista koostuvista kolmiulotteisista malleista eli objekteista, mutta pelin tapahtuma-ala on kaksiulotteinen taso. Tällaista peliä tarkastellaan

kiinteästä perspektiivistä, ja pelaajalla harvoin on mahdollisuutta vaikuttaa kameran perspektiiviin tai hahmon liikkeeseen kolmiulotteisesti. Rajoittamalla pelaajan liikkeitä saadaan erilaisia vapauksia pelin suunnittelun tekniseen ja graafiseen toteutukseen. Aiemmin 2.5D:llä viitattiin grafiikoihin, joilla yritettiin simuloida kolmiulotteisuutta, mutta silloin ei käytetty 3D-polygoneista koostuvia objekteja. Myös pelejä, joissa oli 3D-ympäristöt, mutta 2D-spriteillä tuodut hahmot, kutsuttiin termillä 2.5D. (Giantbomb 2014a.)

4 3D-GRAFIikka

4.1 Kolmiulotteisuus

3D tarkoittaa kolmea ulottuvuutta: leveys, korkeus ja syvyys. Ihmisen fyysinen elinympäristö on kolmiulotteinen, ja ihmisellä on kyky havainnoida ympäristöä kolmiulotteisesti. Tärkein eroavaisuus kaksiulotteisuuteen on kyky erottaa syvyys. Todellisuudessa, kun ihminen tarkastelee ympäristöä, muodostuu silmän verkkokalvolle kaksiulotteinen kuva, jonka aivot kuitenkin prosessoivat kolmiulotteiseksi havainnoksi. (Mediacollege.com 2012.)

Kolmiulotteisen kuvan voidaan sanoa syntyneen vuonna 1838, kun Charles Wheatstone rakensi stereoskoopin varhaisen version. Tarkasteltaessa kohdetta stereoskoopilla syntyi illuusio myös syvyydestä. Laitte näytti ihmisilmälle saman aiheen kuvattuna kahdesta eri kuvakulmasta, jolloin syntyi syvyysvaikutelma. Stereoskooppi levisi maailmalle vuoden 1851 maailmannäyttelyn jälkeen, kun yleisölle näytettiin kolmiulotteisia kuvia. (Historia 2012.) Stereoskoopin toimintaperiaate on tullut tutuksi viime vuosikymmeninä yleisimmin View Master -laitteen kautta (kuva 6).



KUVA 6. View Master -laitteen komponentit ja stereoskoopin toimintaperiaate (ScottDutton/Catspaw dynamics 2004)

Stereoskoopilla luotavan kolmiulotteisen vaikutelman luomiseen on olemassa useita eri tekniikoita, joista kaikki pohjautuvat samaan periaatteeseen. Vasemmalle ja oikealle silmälle heijastetaan katsottavaksi muuten identtinen, mutta kuvakulmastaan lievästi eroava kuva. Ero on yhtä suuri kuin ihmisen kahden silmän näkökulmat. Aivot sulauttavat kaksi kuvaa yhdeksi, mutta kulma-eron takia syntyy vaikutelma syvyydestä. (Elitisti 2011.) Tulee ottaa huomioon että kolmiulotteisuuden illuusio ja 3D-grafiikka eroavat toisistaan siinä, että 3D-grafiikassa kappaleet ovat todellisuudessa kolmiulotteisia (3dRender.com 2002). Esimerkiksi stereoskoopilla tuotettu vaikutelma syvyyssuuntaisesta ulottuvuudesta ei ole todellisuudessa kolmiulotteinen vaan sen illuusio.

4.2 3D-grafiikan käyttösovellukset

3D-grafiikkaa hyödynnetään mitä erilaisimmissa yhteyksissä, muun muassa erilaisissa suunnittelutehtävissä, lääketieteessä, simulaatioissa sekä elokuvissa ja mainoksissa. Esimerkiksi kaupunkien, rakennusten ja tilojen mallintaminen 3D-ohjelmistojen avulla on hyvin yleistä. 3D-grafiikalla voidaan luoda hyvin todentuntuja näkymiä ja tuotekuvia tuotteista, joiden kuvaaminen kameralla olisi liian vaikeaa tai hyvin kallista. (Puhakka 2008, 24.)

Yksi kiinnostavimpia 3D-grafiikan sovellusalueita on lisätty todellisuus, jossa ihmiselle pyritään keinotekoisesti luomaan todelliselta tuntuva maailma. Keskeisimpiä 3D-grafiikan käyttökohteita ovat kuitenkin pelit. Peliteollisuus on hyvin keskeisessä roolissa 3D-grafiikan laitteistojen ja standardien kehityksessä. Tietokonepelejä on ollut olemassa jo vuosikymmeniä, mutta vasta viimeisen kahden vuosikymmenen aikana on hyödynnetty 3D-grafiikkaa realistisen oloisten peliympäristöjen luomiseen. (Puhakka 2008, 24 - 25.)

4.3 3D-grafiikka peleissä

4.3.1 3D-mallit

Suurinta osaa nykyaikaisista peleistä pelataan reaaliaikaisesti 3D-ympäristössä, jossa voi liikkua leveys-, korkeus- ja syvyyssuunnassa. Myös 2D-peleissä käytettävä tekniikka on usein renderöity 3D-ohjelmista kaksiulotteiseksi. Pelin kehittämisprosessin ensimmäinen vaihe on pelihahmojen ja ympäristön mallintaminen, teksturointi ja animointi. Useimmissa 3D-malleissa käytetään polygoneja mallien kehikkona ja niiden tekstuurit ovat kaksiulotteisia maalattuja bittikarttoja. (Morris & Hartas 2003, 110.)

Pelihahmon luominen aloitetaan sijoittamalla se mallinnusohjelmassa olevan äärettömän 3D-tilan origoon eli nollapisteeseen. Tämä siksi, että origosta voidaan laskea etäisyyksiä, jolloin kappaleelle voidaan määrittää

koko haluttuun mittakaavaan. Hahmot muodostuvat yleensä polygoneista, jotka ovat tasoja pisteiden koordinaattien välillä. Todellisuudessa hahmon työstäminen aloitetaan yksinkertaisesta perusmuodosta, kuten kuutiosta, pallosta tai kartiosta. Hahmon työstämistä ei käytännössä aloiteta juuri koskaan pelkästään pisteitä ja polygoneja muodostamalla, vaikka ne ovatkin perusta hahmon rakenteelle. (Morris & Hartas 2003, 110 - 111.)

3D-mallinnuksessa polygonit ovat nykypäivänä yleisimmin käytetty geometria. Polygonit ovat helposti muokattavissa, ja niiden kanssa työskentely on suhteellisen yksinkertaista. Polygonit koostuvat kolmesta tai useammasta kulmasta eli verteksistä. Verteksejä yhdistäviä viivoja kutsutaan nimellä "edge" eli särmä. Useimmat 3D-ohjelmistot luovat pinnanmuodot täyttämällä verteksien väliset alueet, jolloin syntyy "face" eli taho. Verteksien, särmien ja tahojen muodostamaa kokonaisuutta voidaan pyörittää, siirtää ja skaalata 3D-ympäristössä. Renderöidessä havaitaan vain polygonien tahot, eikä verteksit ja särmät näy renderöidyssä kuvassa. (Beane 2012, 136.)

Suurempi määrä polygoneja tarkoittaa parempaa tarkkuutta hahmon pinnan muodoissa, jolloin tietokoneen tulee suorittaa myös enemmän laskentaa. Laitteistot kehittyvät nopeammiksi kaiken aikaa, joten pelisuunnittelijat voivat luoda malleista entistä yksityiskohtaisempia lisäämällä polygoneja ja saada hahmojen pinnanmuodoista mahdollisimman tasaisia. Nopeasta kehityksestä huolimatta on grafiikan esittämisessä edelleen rajoitteita, koska koneet eivät pysty laskemaan sijainteja ja liikettä käyttäjän liikkeiden tahdissa. (Morris & Hartas 2003, 110 - 111.)

4.3.2 Interaktiivinen ja reaaliaikainen 3D

Monet 3D-grafiikan sovellukset ovat interaktiivisia, joka tarkoittaa käyttäjän ja piirrettävän mallin kanssa tapahtuvaa vuorovaikutusta (Puhakka 2008, 435 - 436). Interaktiivinen sovellus vastaa esitettyihin pyyntöihin ja antaa käyttäjälle toimintavaihtoehtoja eli polkuja, jolloin käyttäjä voi vaikuttaa tapahtumien kulkuun tekemällä erilaisia valintoja (BusinessDictionary 2015).

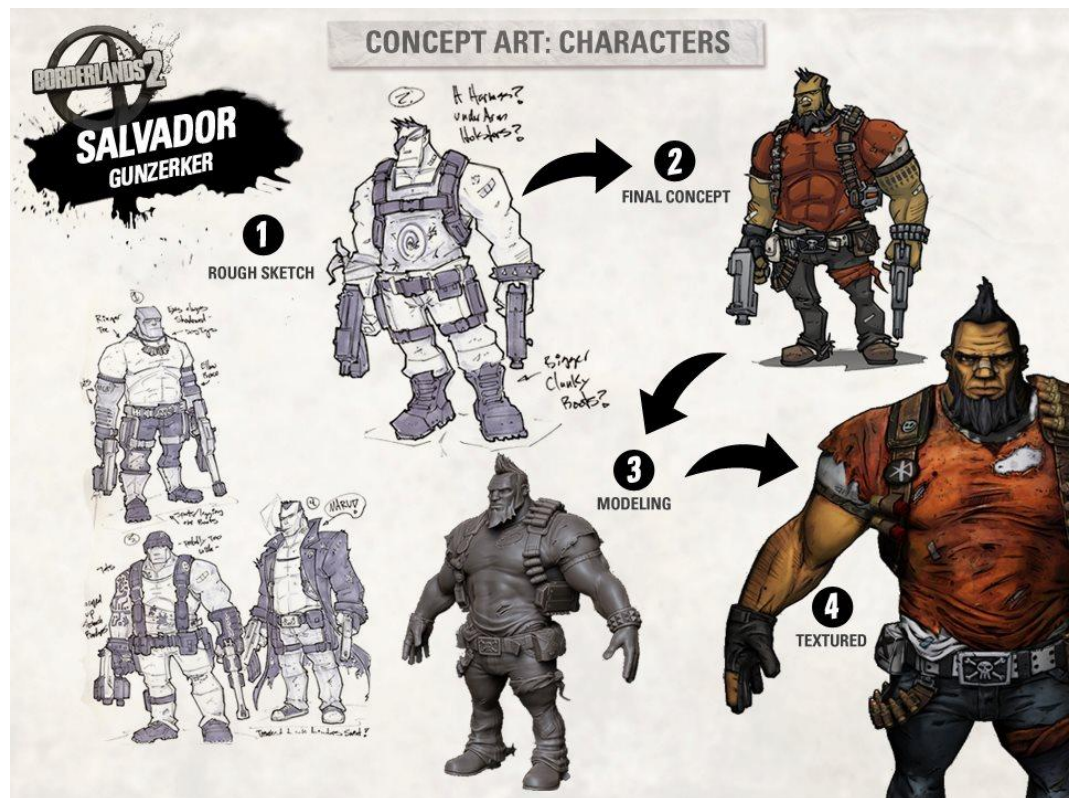
Real time eli reaaliaikainen mainitaan nykyisin useissa tietotekniikkaan liittyvissä yhteyksissä, muun muassa reaaliaikainen käyttöjärjestelmä vastaa syötteeseen välittömästi tai sosiaalisessa mediassa voi ottaa osaa tv-ohjelman keskusteluun reaaliaikaisesti. Pelimaailmassa reaaliaikaisella tarkoitetaan tapahtumien ja toimintojen tapahtumista välittömästi, kuten todellisuudessa juuri sillä hetkellä tapahtuu. Esimerkiksi animaatiossa reaaliaikaisesti tapahtuva kappaleiden liike näytöllä tapahtuu samalla nopeudella kuin oikeassa elämässä. (Webopedia 2015.)

5 CONCEPT ART – KONSEPTITAIDE

Concept art eli konseptitaide on pelisuunnittelussa käytettävä käsite. Konseptitaiteen avulla esitellään pelin graafista yleisilmettä luonnosten avulla. Konseptitaiteen avulla määritellään pelin yleinen tunnelma ja se, miten hahmot, ympäristöt ja itse toiminta pelaajalle näyttäytyy. Konseptitaide on hyödyllisimmillään pelisuunnittelun alkuvaiheessa, jolloin määritellään pelin yleistä tunnelmaa. (Fullerton 2014, 441 - 442.)

Konseptitaiteen suunnittelu vaatii suunnittelijalta hyvää tietämystä aiheesta, jonka pohjalta konseptiluonnosta tehdään. Ollakseen uskottava yleisölle konseptin tulee olla yksityiskohtia myöden huolellisesti työstetty. Esimerkiksi hyvä tietämys tieteestä ja historiasta ovat hyvän konseptitaiteilijan tärkeimpiä ominaisuuksia. Konseptitaiteen suunnittelu vaatii myös hyvää ideointikykyä, suunnittelutaitoa ja kykyä esittää ideat ja suunnitelmat visuaalisesti esimerkiksi luonnosten avulla. (Radbin 2012.)

Konseptitaiteen suunnitelmassa on yleensä neljä päävaihetta, joissa esitellään suunnitelma pelin hahmoista, ympäristöstä, key art -kuvituksesta sekä pelin rekvisiitasta. Pelihahmon luonnoksissa esitellään hahmon anatomia, historia, kulttuuritausta, persoona, elämäntavat ja muut piirteet jotka määrittävät pelihahmosta muodostuvan kokonaiskuvan. Kuvassa 7 näkyy hahmon kehityksen vaiheet, kun hahmoa työtetään concept art -luonnoksesta valmiiksi 3D-malliksi. Peliympäristön luonnokset esittelevät muun muassa peliympäristön kulttuurihistoriaa, sääolosuhteita ja valaistusta. Key art on yksittäinen kuva jostakin toiminnasta, jonka kautta pelin yleistä tunnelmaa halutaan esittää. Key art -kuvaa piirrettäessä tulee huomioida kuvakulmat ja valaistus. Key art -kuva on ikään kuin still-kuva toimintakohtauksesta, jolla kuvataan peliin haluttua tunnelmaa. Pelin rekvisiittia voi olla rakenteeltaan hyvin monimutkaista ja muokkautua pelitilanteissa. Rekvisiitan luonnoksissa esitellään tavaroiden ja erilaisten pelissä käytettävien välineiden, kuten ajoneuvojen, rakennetta, muotoilua, toimivuutta ja liikettä. (Radbin 2012.)



KUVA 7. Borderlands 2 -pelihahmon concept art (Wikia 2012)

6 3D-YMPÄRISTÖN PROSESSOINTI

6.1 Renderöinti

6.1.1 Renderöintiprosessi

Renderöinti on toimenpide, jossa 3D-mallista projisoidaan eli luodaan näytölle 2D-kuva. Renderöinti suoritetaan sekä still-kuville että videolle. Renderöinti voidaan suorittaa ennalta tai reaaliaikaisesti niin kutsuttuna real time -renderöintinä. Real time -renderöintiä käytetään erityisesti 3D-videopeleissä, joissa vuorovaikutus pelaajan ja ympäristön välillä on suuressa roolissa. (WhatIs.com 2009.)

Renderöintiprosessi määrittää mallinnetun kolmiulotteisen ympäristön kuvaksi, joka on kuvattu määritellystä kuvakulmasta, esimerkiksi pelihahmon perspektiivistä. Renderöinti käsittelee ympäristössä olevat valot, varjot, värit, tekstuurit ja muut optiset elementit ja sen kuinka ne vaikuttavat kappaleeseen tilassa. Renderöinnillä voidaan tavoitella todellisuutta vastaavaa kuvaa tai vaihtoehtoisesti sen avulla voidaan liioitella ja saada kuva näyttämään epätodelliselta. (3dRender.com 2002.)

Renderöidessä ohjelmisto ikään kuin valokuvaa kuvan jokaisen pikselin. Yhden pikselin värin laskettamiseen voi mennä paljon aikaa, sillä ympäristöön asetetut valot aiheuttavat valonsäteiden "liikettä" ympäri kuvaa ja aiheuttaa paljon sävyeroja, vaikka kappaleet olisivat samanvärisiä. Esimerkiksi pitkien 3D-animaatioelokuvien renderöinti vie aikaa kuukausia tai vuosia, vaikka niitä renderöisi kymmeniä tai satoja tietokoneita. (3dRender.com 2002.)

6.1.2 Reaaliaikainen renderöinti

Kun suoritetaan renderöinti etukäteen, voidaan suorittaa paljon raskaampia toimenpiteitä kuin reaaliaikaisesti renderöitäessä. Esimerkiksi animaatioelokuva on aina renderöity etukäteen valmiiksi kokonaisuudeksi, kun taas peleissä osa ympäristöstä voidaan renderöidä etukäteen ja osa reaa-

liaikaisesti. Peliympäristön toiminnot ja animaatiot eroavat elokuva- ja tv-animaatioista siinä, että peliympäristössä reaaliaikaisesti renderöidessä koneen tulee voida laskea käyttäjän liikkeen mukaan tapahtuvat muutokset, ja vielä kyetä näyttämään ne reaaliaikaisesti. (Morris D. & Hartas L. 2003, 111.)

Videopelin pelimoottori renderöi reaaliaikaista kuvaa pelitilanteesta 15:sta 100:an kuvaan sekunnissa. Reaaliaikaisesti renderöitäessä ikään kuin huijataan, jotta pelinäkömä saadaan näyttämään mahdollisimman hyvältä. Esimerkiksi tekstuurit, valaistus ja varjot voidaan määrittää kappaleille etukäteen, jolloin tekstuurit, valot ja varjot ovat kappaleessa kiinteinä. Renderöinti tapahtuu kevyemmin, kun pelimoottorilla on vähemmän informaatiota laskettavana. Kiinteiden ominaisuuksien ennalta määrittäminen kuitenkin toimii vain kappaleille, jotka eivät ole liikkeessä, jolloin niihin kohdistuvat valot ja varjot eivät myöskään vaihda paikkaa. Reaaliaikainen renderöinti aiheuttaa monenlaisia rajoitteita, muun muassa käytettävien tekstuurien tiedostokooissa sekä hahmojen ja kappaleiden yksityiskohtien tarkkuudessa. Pelimoottorien ja tietokoneiden kehittyessä voidaan renderöidä yhä parempaa kuvaa myös reaaliaikaisesti. (Beane 2012, 284.)

6.2 Renderöity kuva

6.2.1 Renderöidyn kuvan ominaisuudet

Renderöity kuva sisältää useita ominaisuuksia, joilla haluttu vaikutelma saadaan aikaan. Osa ominaisuuksista perustuu algoritmeihin ja tekniikoihin ja osa vaikutuksista syntyy eri ominaisuuksien keskinäisestä vuorovaikutuksesta. Vuorovaikutuksessa keskenään ovat esimerkiksi ympäristöön määritettyjen valot ja materiaalit, joista aiheutuu erilaisia heijastumia ja varjoja. (Wikipedia 2014.)

Kolmiulotteisen vaikutuksen luomiseksi kuvaan tulee saada varjot, valon heijastumat, materiaalit, pinnanmuodot ja värit toistumaan mahdollisimmat luonnollisella tavalla. Haluttu vaikutus luodaan lisäämällä 3D-ympäristöön

valoja sekä määrittämällä kappaleille materiaaleja. (Cornell University - Program of computer graphics 1998.)

6.2.2 Shading - varjostus

Shader eli varjostin sisältää kappaleen materiaalitiedot. Tietokoneen grafiikkaprosessorin ajama ohjelma käsittelee kappaleen materiaalitiedot, kun kuva renderöidään. Shaderit tuottavat kuvaan, ympäristöön määritettyjen tietojen ja ominaisuuksien perusteella, värit, tekstuurit, valot ja muut kappaleen ominaisuudet. Käsite shader - varjostin on oikeastaan hieman harhaanjohtava, koska todellisuudessa on kyse laskennallisesta prosessista, eikä se liity perinteisessä piirtämisessä käytettävään varjostuskäsitteeseen. (Processing 2013.)

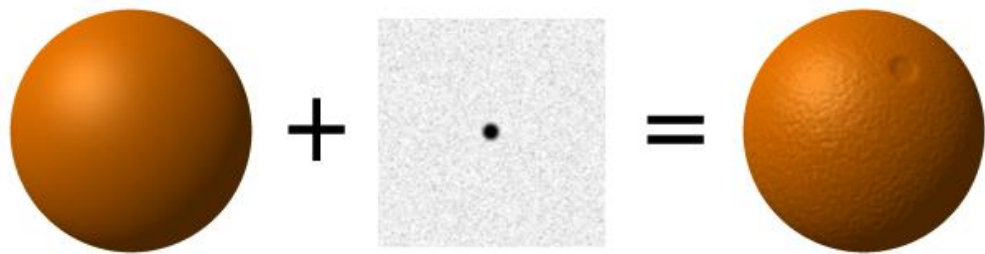
Shaderin valinnalla voidaan vaikuttaa renderöidyssä kuvassa näkyviin väriin ja valojen käyttäytymiseen, esimerkiksi kuinka valot reagoivat törmätessään kappaleen pintaan. Shaderin perusominaisuudet ovat kappaleelle määritetty väri tai tekstuuri, ambient – ympäristön värien vaikutus kappaleeseen sekä läpinäkyvyys. Niin kutsutut ”Complex shaderit” ovat monimutkaisempia ja tarjoavat useampia vaihtoehtoja valon käyttäytymisen laskemiselle. (Beane 2012, 163.)

6.2.3 Tekstuurit ja pinnanmuodot

Teksturoidessa pyritään jäljittelemään joko konseptitaiteessa määriteltyjä ominaisuuksia tai mallin esimerkkinä toimivan todellisen kappaleen ominaisuuksia. Kappaleelle lisätään väri- tai tekstuuri- ja pintaominaisuudet halutun ulkonäön saavuttamiseksi. Esimerkiksi renderöidyssä kuvassa näkyvän puupöydän materiaalin määrittäminen tapahtuu teksturoinnin avulla, jolloin kappaleelle lisätään puista materiaalia muistuttava tekstuuri-map (bittikarttakuva). (Beane 2012, 39.)

Bump-mapin avulla korostetaan tekstuurin mukaisia pinnanmuotoja eli käytännössä bump-mapissa määritetään, millaiset valo- ja varjokohdat kappaleen pinnassa on. Bump-mapit ovat harmaasävyisiä bittikarttakuvia,

joissa eri harmaan sävyillä on merkityksensä pinnanmuodon muodostumisessa. Bump-mapin keskiharmaat kohdat eivät vaikuta kappaleen pinnanmuotoon, mustat kohdat ovat varjokohtia, esimerkiksi lommoja tai uria pinnassa, valkoiset kohdat ovat pinnasta ulkonevia kohtia, esimerkiksi kuhmuja tai muuten pinnasta ulospäin kohoavia alueita. Bump-map vaikuttaa shaderin eli varjostimen käsittelemään tietoon muuttamalla kappaleen pinnan normaaleja, jolloin syntyy realistinen vaikutelma lommoista tai kohoamista kappaleen pinnassa. Kuvassa 8 esitetään, kuinka pelkästään bump-mapin lisääminen saa pallon näyttämään appelsiinilta. (Beane 2012, 165.)



KUVA 8. Palloon lisätty bump-map (Wikimedia commons 2010)

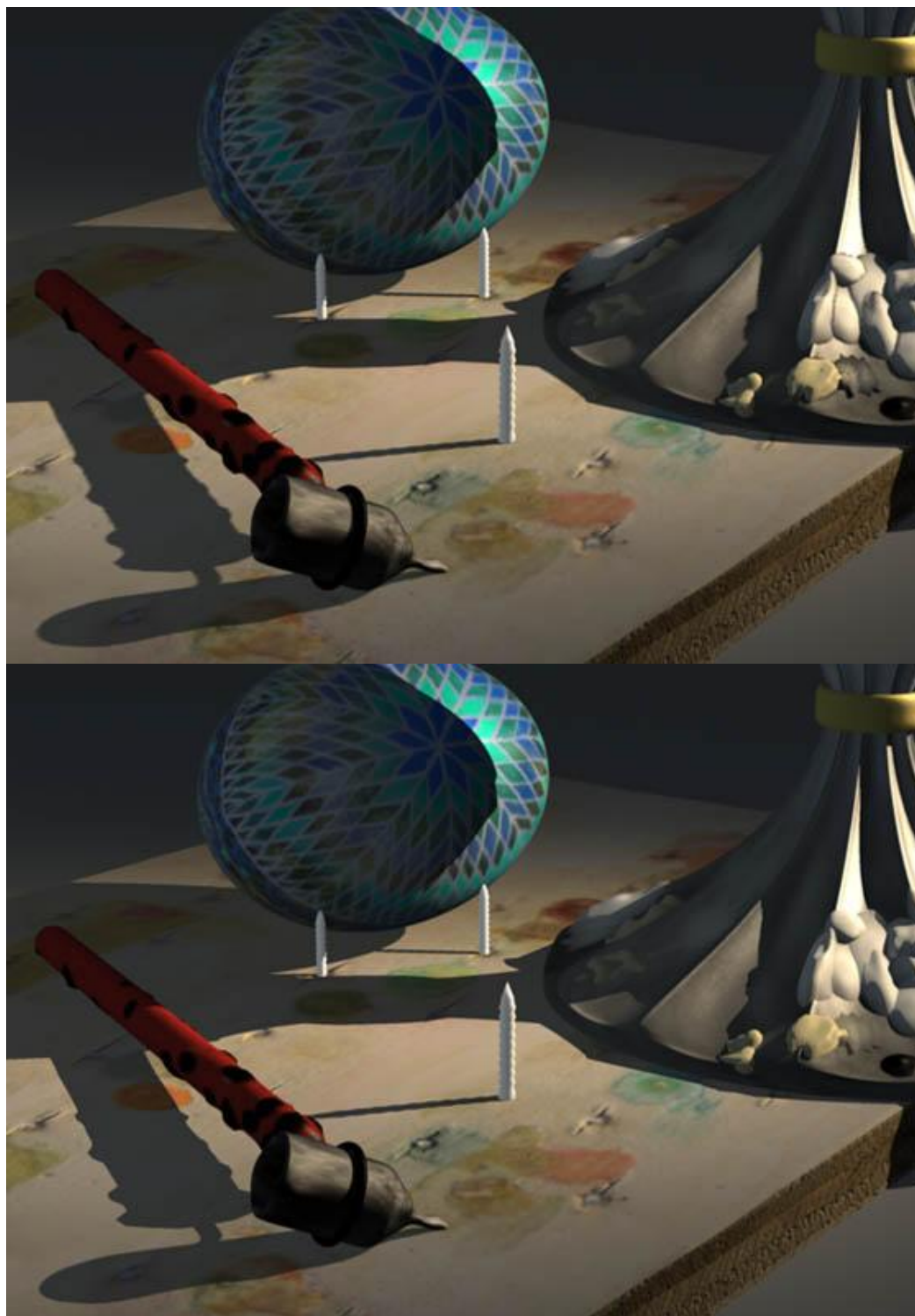
6.2.4 Varjot

3D-ympäristön varjojen luomiseen on useita tapoja. Varjojen kaksi päätyyppiä ovat raytracing-varjot ja depth map -varjot. Varjojen laskettaminen vie aikaa ja tuottaa ympäristön varjot eri tarkkuudella varjojen luomistavasta riippuen. Raytraced shadows eli säteenjäljityksellä tuotetut varjot ovat lähimpänä realistisia varjoja. Raytracingin avulla voidaan muodostaa hyvin tarkkapiirtoiset, reunoista selkeät varjot, jotka kuitenkin ovat realistiset ja niin sanotusti ”oikealla tavalla pehmeät”. Raytracingillä tuotetut varjot myös läpäisevät läpikuultavat objektit, kuten lasin, muovin ja veden, ja heikkenevät läpäistessään materiaa, eli varjon tummuus vaalenee. Varjot syntyvät valojen kimpoillessa ympäristössä olevien kappaleiden pinnoilta langettaen varjoja valonsäteiden ja kappaleiden yhteisvaikutuksen perusteel-

la. Raytracing on todella realistinen tapa tuottaa varjoja, mutta tarkan laskennan vuoksi ympäristön renderöinti vie huomattavan määrän aikaa.

(Beane 2012, 231.)

Depth map -menetelmällä luodut varjot voidaan renderöidä todella nopeasti ja oikein käytettynä sillä voidaan saavuttaa kohtalaisen hyvä lopputulos. Depth map -varjoilla voidaan luoda sekä teräväreunaisia että pehmeitä varjoja. Suurin puute depth map -menetelmässä verrattuna raytracing-menetelmään on, ettei sillä voida luoda läpikuultavaa varjoa, joka syntyisi esimerkiksi valon läpäistessä lasisen kappaleen. Depth map -varjoja renderöitäessä huomioidaan vain valojen läheisyydessä sijaitseva geometria. Jos valon läheisyydessä ei ole kappaleita, ei myöskään valon laskettamista tapahdu. Vain valojen läheisyydessä sijaitsevat kappaleet valaistaan ja vain ne kappaleet langettavat varjoja. Depth map -varjojen laatuun voidaan vaikuttaa muutamilla säädöillä, mutta vaikka resoluutio olisi korkea ja varjojen pehmeyttä ja laatua säätäisi, ei depth map -varjoilla voida päästä aivan yhtä realistiseen lopputulokseen kuin raytracingillä. (Beane 2012, 232.) Kuvassa 9 verrataan raytracing- ja depth map -menetelmillä renderöityjä kuvia. Raytraced-varjojen reunat erottuvat selkeämmin ja varjojen sävy vaihtelee luonnollisesti.

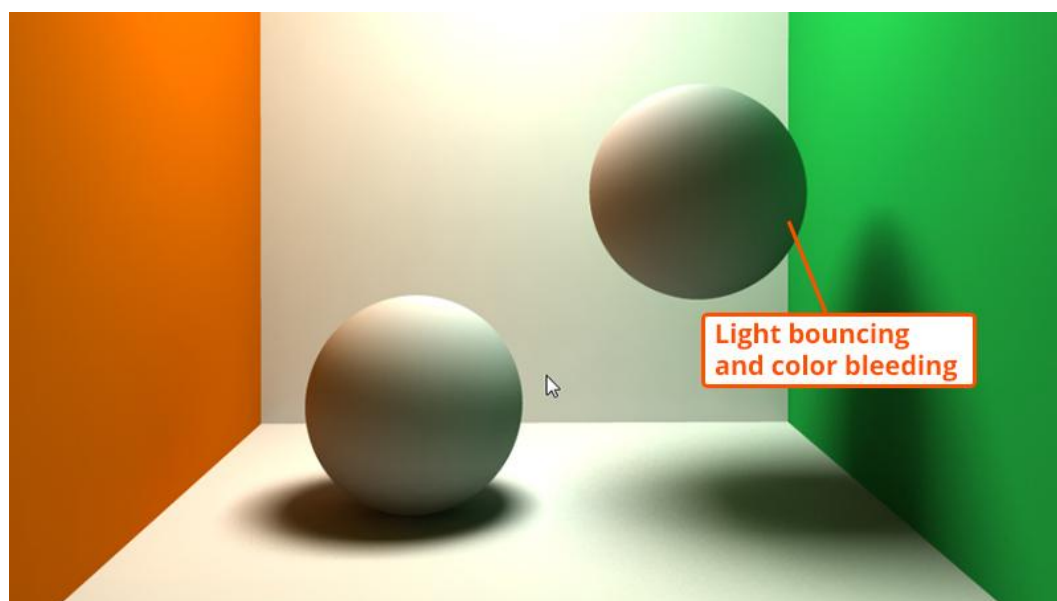


KUVA 9. Ylemmässä kuvassa depth map -varjot ja alemassa kuvassa raytraced-varjot (Dan Bailey)

6.2.5 Valon käyttäytyminen

Reflectivity eli heijastuvuus määrittelee, kuinka heijastava kappale on. Heijastus voi aiheutua joko raytracingillä suoritetusta valon laskettamisesta tai ennalta määritetystä heijastus-mapista. Refraction ja diffraction ovat ominaisuuksia, jotka määrittävät valon taittumissuunnan läpikuultavien kappaleiden sisällä. Valon taittuminen ja hajaantuminen määräytyy valon nopeudesta valon läpäistessä läpikuultavan kappaleen. Refraction- ja diffraction-ominaisuuksilla saadaan aikaan luonnollisia vääristymiä, esimerkiksi vesilasin läpi tarkasteltuna vesilasin takana olevat kappaleet näyttävät vääristyneiltä. (Beane 2012, 164.)

Global illumination (tai indirect illumination) on prosessi, joka simuloi epäsuoran valon vaikutusta 3D-ympäristössä, kuten valon kimpoamista kappaleista ja kappaleiden värien heijastumista toisiin kappaleisiin. Global illuminatiolla saavutettavaa lopputulosta havainnollistetaan kuvassa 10. Värien heijastumia kutsutaan yleisesti käsitteellä ”color bleeding”. Sana indirect – epäsuora tarkoittaa, että lankeava valo kimpoaa jonkin kappaleen pinnasta eikä osu kappaleen pinnalle suoraan valonlähteestä. Näin kimpoavaan valoon vaikuttaa kimmottavan kappaleen väri, joka heijastuu siten muihin tilan kappaleisiin. (Digital-tutors 2014.)



KUVA 10. Esimerkki global illuminationista (Digital-tutors 2014)

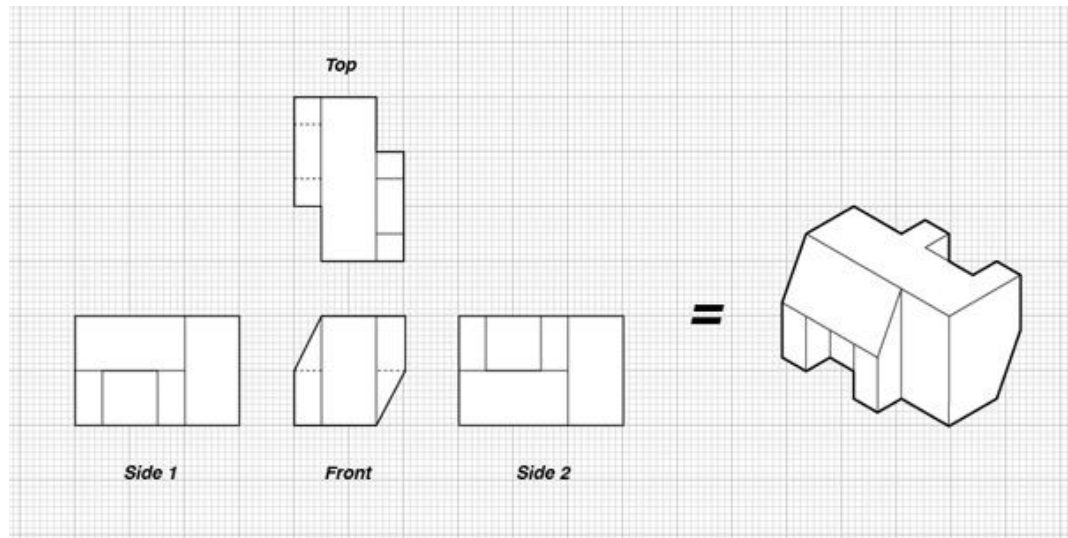
7 PELEISSÄ KÄYTETTÄVÄT TASOPROJEKTIOT

7.1 Paralleeliprojektiot

7.1.1 Ortografinen projektio

Paralleeliprojektiossa, jossa projektorisuorat ovat kohtisuorassa kuvaustasoon nähden eli ovat tason normaalin suuntaisia, kutsutaan ortografiseksi projektioksi. Ortografisessa projektiossa pääakselin suuntaisesti tarkasteltava kuutio näyttää katsojan silmään neliöltä. Ortografista projektiota voidaan havainnollistaa parhaiten moniulotteisemmalla muodolla ja tarkastelemalla sitä ortografisen projektien eri suunnista, kuten kuvassa 11. Ortografisessa projektiossa pääakseleiden suunnilla ja kuvattavan mallin rakenteella on läheinen yhteys. Esimerkiksi suorakulmaista taloa kuvatessa malli on yleensä rakennettu niin, että x- ja y-akselit ovat maan tasossa katu- ja suuntaisina ja y-akseli osoittaa taivaalle. Tällöin ortografinen projektio jaotellaan sen mukaan, miten projektiosuunta suhtautuu pääakseleihin. Esimerkiksi projektiota, joka ei ole minkään pääakselin suuntainen, kutsutaan aksonometriseksi projektioksi. (Puhakka 2008, 179.)

Ortografisen projektion vaikutus perustuu perspektiivin puuttumiseen ja akseleiden välisiin suhteisiin. Käytännössä se tarkoittaa sitä, etteivät kappaleet pienene kohti pakopistettä, koska pakopistettä ei todellisuudessa ole. Myös akselien suhteelliset pituudet säilyvät, vaikka kuvaa vieritettäisiin. Tasohyppelypeleissä käytetään usein suoraan sivusta tarkasteltua, pääakselin suuntaista ortografista projektiota. Kameran sijaitessa hieman yläviistossa (edelleen kuitenkin sivussa), jolloin katsontasuunta alas maailmaan pelihahmon ylitse, luodaan lintuperspektiivi, joka jäljittelee katsojalle paremmin hahmon näkemää ympäristöä. Sivukuvakulma ja lintuperspektiivi ovat eniten käytettyjä perspektiivityylejä 2D-peleissä. Lintuperspektiivin voidaan joskus ajatella luovan illuusion kolmiulotteisuudesta, mutta siinä käytetyt y- ja z-akseli toimivat samassa avaruudessa eikä kappaleita tarkastella kuin yhdeltä sivulta. (SignificantBits 2009.)



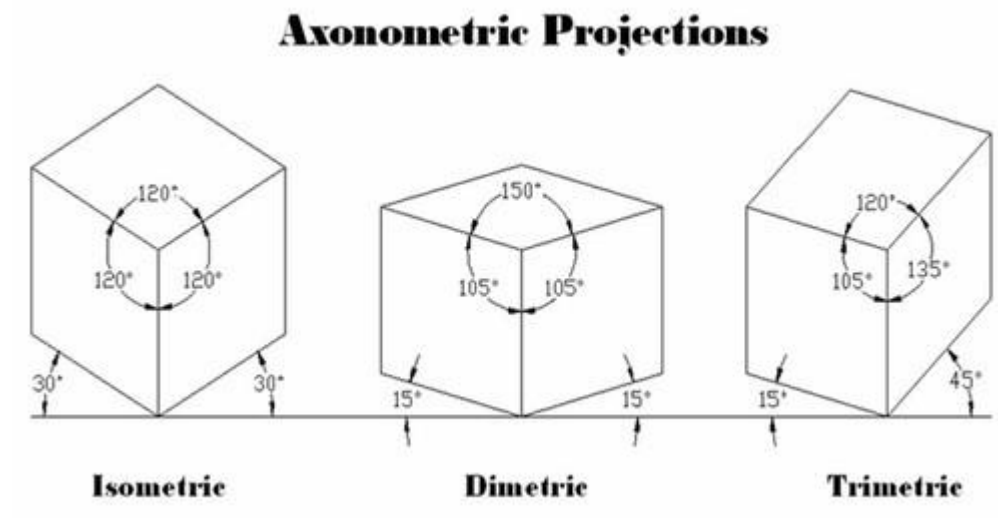
KUVA 11. Ortografisen projektion eri versioita (Tuts+ 2011)

7.1.2 Aksonometrinen projektio

Aksonometrinen projektio on pakopisteetön yhdensuuntainen projektio, joka ei ole minkään pääkselin suuntainen. Toisin kuin realistiseen esittämiseen pyrkivän perspektiiviin piirtämisen perusominaisuuksia on yksi tai useampi pakopiste, aksonomisessa projektiossa ei ole pakopistettä lainkaan. Pakopisteen puuttuminen mahdollistaa saman perspektiivin säilymisen kuvaa vieritettäessä. Aksonometrisessä projektiossa tarkasteltavat kappaleet säilyvät samankokoisina, vaikka niiden tarkasteluetaisyys muuttuisi. Aksonometriselle projektiolle tyypillinen yläviistosta suunnattu tarkastelu mahdollistaa kappaleiden esittämisen kolmiulotteisesti, esimerkiksi kuutiosta voidaan näyttää kolme sivua samanaikaisesti. (Compu Phase 2011.)

Aksonometrisen projektion kolme päätyyppiä (kuva 12) ovat isometrinen, trimetrinen ja dimetrinen. Päätyypit määritellään x-, y-, z-koordinaattien kulmalukujen perusteella. Isometrisessä projektiossa kaikki kolme kulmaa ovat yhtä suuria, jolloin esimerkiksi tasakylkisen kuution kaikki sivut ovat myös isometrisessä projektiossa yhtä pitkiä. Dimetrisessä projektiossa koordinaatiston kaksi kulmaa ovat yhtä suuria. Trimetrisessä projektiossa

jokainen kulma on erisuuri. (Bing-Yu Chen 2008.) Isometristä projektiota käytettiin yleisesti simuloimaan kolmiulotteisuutta 2D-peleissä, josta esimerkki kuvassa 13.



KUVA 12. Aksonometrisen projektion kolme päätyyppiä (Mr. Rush's site 2015)



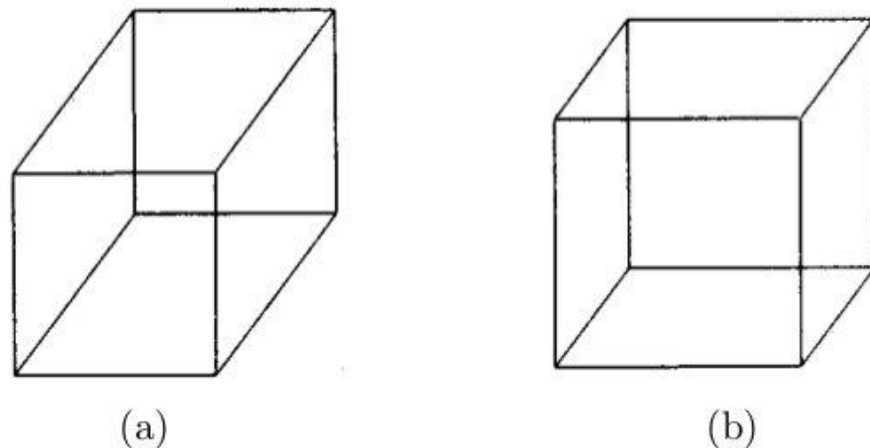
KUVA 13. Isometristä projektiota hyödyntävä Qbert -peli vuodelta 1982 (Wikipedia 2007).

7.1.3 Oblique-projektio

Paralleeliprojektiota, jossa projektorisuorat ovat vinossa kuvatasoon nähden, kutsutaan oblique-projektiksi. Oblique-projektio on ortografisen projektion vastakohta, koska projektion suunta ei ole kohtisuorassa kuvatasoon nähden. Kabinetti-projektio ja kavaljeeri-projektio ovat molemmat oblique-projektityyppejä. Esimerkiksi kavaljeeriprojektiossa kuvataason ja projektiosuunnan välinen kulma on 45 astetta. Kavaljeeriprojektio säilyttää sekä kuvataason suuntaisen että kuvaustasoon nähden kohtisuorassa olevien viivojen pituudet. Kavaljeeriprojektiossa tarkastellaan yhtä sivua kohtisuorasti, jolloin kuutio näyttää neliöltä ja kuvaustasoon kohtisuorien viivojen pituus säilyy ennallaan. Viivojen pituuden säilyminen luo vaikutelman venymisestä. (Puhakka 2008, 180 - 181.) Kavaljeeriprojektion tarkasteltua kuutiota voidaan pitää venyneenä ja muistuttavan enemmän perspektiiviprojektiossa tarkasteltua suorakulmaista särmiötä, kuten voidaan havaita kuvassa 14 esitettävästä oblique-projektion vertailusta.

Kabinettiprojektiossa kuvaustasoon ja projektiosuunnan välinen kulma on 63,4 astetta, jolloin kohtisuorien viivojen pituudet lyhentyvät puoleen. Viivojen pituuden puolittumisen vuoksi kabinettiprojektion avulla voidaan tuottaa realistisemmän oloinen kuva kuin kavaljeeriprojektion avulla. (Puhakka 2008, 181.)

Kavaljeeriprojektion aiheuttaman kappaleiden venymisen vuoksi sitä ei juurikaan käytetä peleissä. Kabinettiprojektiota puolestaan käytetään jonkin verran. Myös kabinettiprojektiossa tarkastellaan yhtä sivua kohtisuorasti, jolloin kuutio näyttää neliöltä. Muut sivut projisoidaan yleensä 45 asteen kulmassa projektion muodostaman neliön sivuun nähden. Kuvaustasoon ja projektiosuunnan kulman aiheuttama kohtisuorien viivojen pituuden puolittuminen luo vaikutelman syvyydestä. (SignificantBits 2009.)



(a) Cavalier projection and (b) cabinet projection.

KUVA 14. Kavaljeeriprojektion ja kabinettiprojektion vertailu (StackExchange 2011)

7.2 Perspektiiviprojektio

7.2.1 Perspektiiviprojektion muodostuminen

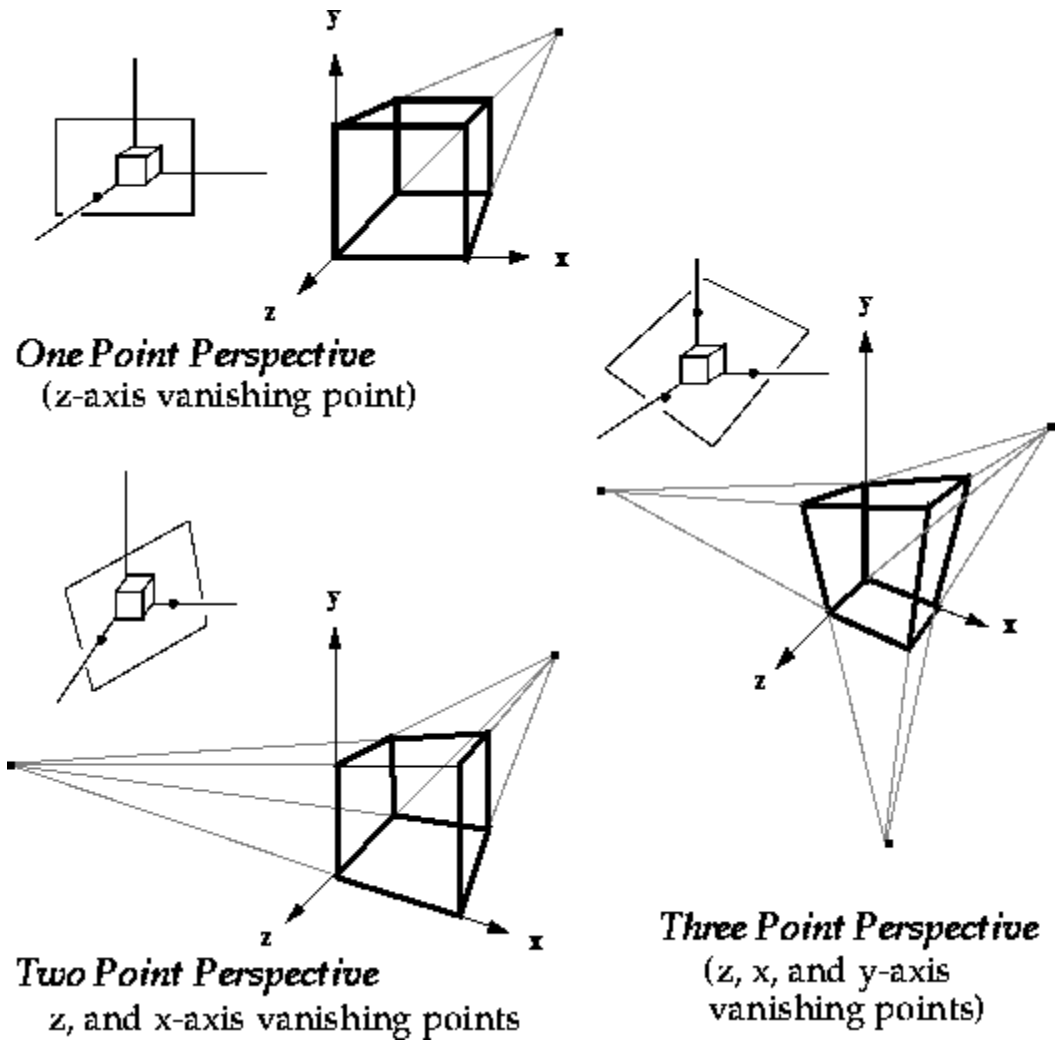
Perspektiiviprojektio on yleisin tapa muodostaa kaksiulotteinen kuva kolmiulotteisesta kappaleesta. Idea perspektiiviprojektioista syntyi ensimmäisen kerran renessanssiajalla taiteilijoiden etsiessä realistista tapaa kuvata maailmaa. Voidaan sanoa, että perspektiiviprojektio tuottaa luonnollisimman näköisen kuvan. Perspektiiviprojektioilla tuotetussa kuvassa kauempana olevat kappaleet näyttävät pienemmiltä kuin etualalla olevat. Perspektiiviprojektioita voidaan käyttää sekä 2D- että 3D-ympäristöjen esittämiseen. (Puhakka 2008, 183 - 184.)

Perspektiiviprojektiossa kaikki projektorisuorat kulkevat projektion keskipisteen kautta, jota voidaan kutsua projektiokeskukseksi tai kuten kuvataiteessa yleisimmin pakopisteeksi tai katoamispisteeksi. Kappaleessa olevat samansuuntaiset suorat eivät näy kuvassa samansuuntaisina, vaan pisteiden kuvitteelliset jatkeet kohtaavat toisensa projektion keskipisteessä. Pro-

jehtiokeskuksen voidaan ajatella olevan katsoja ja katsomissuunta z-akselin suuntainen. Kappaleen kulmapisteistä lähtevät projektorit kohtavat projektiokeskuksessa eli katsojassa, jossa perspektiivi havaitaan. (Puhakka 2008, 184.)

Perspektiiviprojektio ikään kuin siirtää kappaleiden kärkipisteitä katsojaa kohti. Laskenta perustuu kappaleiden kärkien sijaintiin. Kauimpana z-akselin suuntaisesti sijaitsevat kärkipisteet liikkuvat suhteessa vähemmän kuin lähempänä katsojaa sijaitsevat kärkipisteet. Kärkipisteiden liikkeeseen vaikuttaa myös niiden x- ja y-suuntainen etäisyys projektiokeskuksesta. (Learning Modern 3D Graphics Programming 2012.) Perspektiiviprojektio voidaan piirtää yhden (z-akseli), kahden (z- ja x-akseli) tai kolmen (z-, x- ja y-akseli) pakopisteen avulla, riippuen halutusta lopputuloksesta. Yhden tai useamman pakopisteen perspektiiviprojektion piirtäminen esitetään kuvassa 15.

Pakopistettä hyödyntämällä luotua perspektiiviprojektiota käytettiin yleisesti varhaisissa 2D-peleissä, joissa taustan kuvitus oli perspektiiviin piirretty. Tällöin syvyysuuntainen vaikutelma luotiin sijoittamalla oletettavasti kauempana sijaitsevat kappaleet lähemmäs peliruudun yläreunaa pienempikokoisina ja lähempänä sijaitsevat kappaleet lähemmäs peliruudun alareunaa suurempikokoisina. Perspektiiviin piirretty taustakuva myös auttoi kappaleiden välisten etäisyyksien arvioimisessa ja niiden skaalaamisessa suhteessa oikean kokoisiksi. Nykypäivänä tämän tyylisesti toteutettu perspektiiviprojektion jäljittely ei toistu enää uskottavasti, mikä johtuu näyttöjen hyvästä resoluutiosta. (SignificantBits 2009.)



KUVA 15. Kolme erilaista perspektiiviprojektiota (Viewing history 1995)

7.2.2 Fixed 3D

Fixed 3D on perspektiiviprojektio, jossa ympäristö ja siellä sijaitsevat kappaleet ovat ennalta renderöityjä. Fixed 3D -perspektiivissä kolmiulotteiset kappaleet voidaan myös renderöidä reaaliaikaisesti 2D-kuvan päälle. Fixed 3D -perspektiivi on kiinteä, eikä pelaaja voi muuttaa tarkastelun kuvakulmaa. 2D-peleissä käytettävien projektioiden ja Fixed 3D -perspektiivin eroavaisuuksia on vaikea kuvata sanatarkasti. Merkittävin ero voidaan sanoa olevan 3D-kappaleiden käyttö kaksikulotteisen taustan päällä. Kun ympäristön näkymä renderöidään, voi lopputulos muistuttaa hyvin paljon, kuvakulman suunnasta riippuen, 2D-ympäristöjen esittämiseen käytettäviä

projektiota. Kuvassa 16 esimerkki Fixed 3D -perspektiivissä pelattavasta SimCity4-pelistä, jonka tarkastelukuvakulma muistuttaa hyvin paljon isometristä projektiota. (SignificantBits 2009.)

Perspektiiviprojektion käyttö peleissä tuli suosituksi vasta 3D-pelien myötä. 3D-peleissä käytettävä perspektiiviprojektio tarvitsee tuekseen erilaisia kappaleille määritettyjä tekstuureja ja varjostuksia, jotka vaikuttavat suuresti lopulliseen kokonaiskuvaan. 3D-peleissä käyttäjällä on mahdollisuus vaikuttaa tarkastelun kuvakulmaan ja peliympäristö ja siinä sijaitsevat kappaleet ovat tarkasteltavissa laajemmin. (SignificantBits 2009.)



KUVA 16. Fixed 3D -perspektiivissä tarkasteltava SimCity 4-peli (EA 2009)

7.2.3 First person -perspektiivi

First person -perspektiivissä käytettävällä projektiokeskuksella simuloidaan käyttäjälle vaikutelma peliympäristön tarkastelusta pelihahmon silmin. Yleensä first person -perspektiiviä käytetään First person shooter -

peleissä ja rallipeleissä. First person shooter -pelit nousivat suosioon vuonna 1992 Wolfenstein 3D -pelin myötä, jota seurasi Doom vuonna 1993 ja Quake vuonna 1996. Ensimmäisenä first person shooter -pelin konsolille toi Rareware ja Nintendo julkaistessaan GoldenEye-pelin Nintendo64-konsolille. First person -perspektiiviä on sittemmin hyödynnetty muissakin peleissä, kuten seikkailu- ja roolipeleissä. (Giantbomb 2015a.)

First person -perspektiiviin ei ole nykyaikana tapahtunut suurta kehitystä, vaan sen parantelu on keskittynyt perspektiivin ominaisuuksien hiomiseen, kuten valojen ja heijastusten oikeellisuuteen ja niiden korostamiseen. Esimerkiksi Halo-pelisarjassa pelinäköymä pyritään esittämään niin, että pelaaja tarkastelee ympäristöä visiirin takaa ja heads-up -näyttö eli tarpeelliset yksityiskohdat, kuten pelihahmon energiataso, ovat pelaajan tarkasteltavissa kaiken aikaa. Myös useissa nykyisissä rallipeleissä ja muissa ajopeleissä käytetään first person -perspektiiviä, jolloin pelaaja näkee radan lähempää ja ympäristön tarkastelun laajuus muuttuu. Tyypillinen esimerkki first person -perspektiivistä kuvassa 17, jossa matalalla sijaitsevalla perspektiivillä luodaan pelaajalle realistisempi vaikutelma autolla ajamisesta. (Giantbomb 2015a.)



KUVA 17. First person-perspektiivi Mario-Kart 7-pelissä (Giantbomb 2015 a)

7.2.4 Second person -perspektiivi

Yleensä second person -termi viittaa pelin kertojaan, erityisesti esimerkiksi tekstipohjaisissa seikkailu- ja roolipeleissä. Kertojan tarkoituksena on viestiä pelihahmon käyttäytymisestä pelaajalle. Näissä tapauksissa ei voida oikeastaan puhua näkemällä koetusta kuvallisesta perspektiivistä. Peleissä käytettävästä visuaalisesta second person -perspektiivissä pelaaja tarkastelee peliympäristöä jonkin muun, kuin pelin päähenkilön silmin. (Giantbomb 2014c.)

Vaikutelma luodaan yleensä kameran kuvakulmalla, jolloin pelaaja ikään kuin tarkastelee pelattavaa hahmoaan toisen pelihahmon näkökulmasta. Second person -perspektiiviä käytetään peleissä hyvin harvoin, mutta esimerkiksi Battletoads-pelissä pelaaja tarkastelee omaa pelihahmoaan vastustajansa silmin, mutta ohjaten edelleen oman pelihahmon toimintaa. (Giantbomb 2014c.)

7.2.5 Third person -perspektiivi

Third person -perspektiivissä pelihahmoa voidaan tarkastella eri etäisyyksiltä, ja kameran kuvakulma voi olla joko muutettavissa tai kiinteässä perspektiivissä. Yleensä third person -käsitteellä viitataan pelihahmon takaa aukeavaan kuvakulmaan, jolloin pelihahmon on osaksi tai kokonaan pelaajan nähtävillä. Yksi third person -perspektiivin etuja on pelihahmon liikkeiden näkyminen pelaajalle ja third person -perspektiiviä käytetäänkin useimmiten ammunta- ja seikkailupeleissä. (Giantbomb 2015b.)

Third person -perspektiivissä kontrolloidaan niin kutsuttua kolmatta persoonaa eli pelihahmoa, toisin kuin first person -perspektiivissä, jossa pelaajalle luodaan tunne, että pelaaja on itse sisällä pelissä ja on oma pelihahmonsansa. Yleensä pelihahmo on sijoitettuna kuvaruudun vasempaan laitaan tai keskelle kuvaruutua, jolloin pelaaja ei milloinkaan voi nähdä pelihahmon kasvoja. Kun pelihahmo sijoitetaan kuvaruudun keskelle, on mahdollisuus kääntää näkymää kääntämällä pelihahmoa kameran ollessa kiinnitettynä pelihahmoon. Tyypillinen esimerkki third person

-perspektiivissä pelattavasta pelistä on Tomb Raider III, josta kuvakaappaus kuvassa 18. (Giantbomb 2015b.)

Ammuntapeleissä tähtäin sijoitetaan yleensä pelihahmon olkapään yläpuolelle, jolloin tähdätessä kuvakulma siirtyy lähemmäs ja jäljittelee tarkentamista ampuessa. Joskus third person -perspektiiviä voidaan käyttää luomaan tunne elokuvan katselusta, jolloin tehosteena käytetään esimerkiksi veriroiskeita näytöllä ja voimakkaita valojen heijastuksia. Yleisimmin third person -perspektiivillä luodaan pelaajalle vaikutelma tilanteesta, jossa pelaaja on kuvauksen kohteena ja kameramies seuraa pelaajan toimintaa jatkuvasti. (Giantbomb 2015b.)



KUVA 18. Third person -perspektiivissä pelattava Tomb Raider III -peli (GameFAQs 2015)

8 YHTEENVETO

Pelin graafinen ilme voidaan luoda monin tavoin. Tietokoneiden kehityksen myötä lisääntyvä käytettävissä oleva muistikapasiteetti mahdollistaa yhä monipuolisemman ja vapaamman peligrafiikan suunnittelun. Tietokoneiden kyetessä renderöimään enemmän ja nopeammin suuriakin määriä kuvainformaatiota voidaan toteuttaa mitä erilaisimpia pelimaailmoja ja yksityiskohtaisempia hahmoja.

Peligrafiikan suunnittelussa yksi tärkeimpiä asioita on miettiä, millaisena pelimaailman halutaan pelaajalle näyttäytyvän. Huolellisella konseptitaiteen suunnittelulla luodaan yksityiskohtainen kuvaus pelin yleisestä tunnelmasta ja pelihahmojen persoonallisuuksista. On myös hyvin oleellista pohtia, millaista projektiota peliympäristön tarkasteluun käytetään. Projektion valinnalla vaikutetaan muun muassa siihen, kuinka pelaaja integroituu pelimaailmaan.

Graafisen tyylin valinnoilla peli voidaan suunnata tietylle käyttäjäryhmälle, tavoitella haluttua tunnelmaa ja luoda brändi pelin ympärille. Esimerkiksi Angry Birds -peli on itsessään brändi. Grafiikassa toistuvat teemat ja huolelliset yksityiskohdat näkyvät monissa Angry Birds -peleissä, ja ne ovat omaleimaisia ja tunnistettavia ympäri maailman.

Hyvä peligrafiikka ei pelasta huonoa peliä - oleellisinta on kuitenkin pelikokemus, mutta hyvin toteutettu peligrafiikka on merkittävässä osassa ja tuo suurta lisäarvoa. Nykyisin oikeastaan on lähes olettamus että peligrafiikan tulee olla kunnossa, suttuiset grafiikat voivat vaikuttaa pelikokemukseen negatiivisesti. Toisaalta taas, jos peli lähestulkoon nauraa itselleen, voi pelikokemus sivuuttaa suttuiset grafiikat, kuten Hill Climb Racing -peli osoittaa. Vaikuttaisi siltä, että peligrafiikan suunnittelussa tärkeintä on huolellinen ja johdonmukainen suunnittelu, jossa graafinen ilme tukee pelin tarinaa.

LÄHTEET

Beane, A. 2012. 3D Animation Essentials. Indianapolis, Indiana: John Wiley & Sons, Inc.

Fullerton, T. 2014. Game design workshop. A playcentric approach to creating innovative games. Boca Raton, Florida: Taylor & Francis Group, LLC.

Morris, D. & Hartas, L. 2003. Game Art. The graphical art of computer games. Iso-Britannia: Collins.

Puhakka, A. 2008. 3D-grafiikka. Helsinki: Talentum.

SÄHKÖISET LÄHTEET

3dRender.com. 2002. 3D Rendering [viitattu 8.1.2015]. Saatavissa: <http://www.3drender.com/glossary/3drendering.htm>

About Tech. 2015a. Vector and bitmap images [viitattu 9.1.2015]. About.com. Saatavissa: <http://graphicssoft.about.com/od/aboutgraphics/a/bitmapvector.htm>

About Tech. 2015b. Vector and bitmap images [viitattu 9.1.2015]. About.com. Saatavissa: http://graphicssoft.about.com/od/aboutgraphics/a/bitmapvector_2.htm

Bing-Yu Chen. 2008. Computer graphics [viitattu 14.1.2015]. National Taiwan University. Saatavissa: http://graphics.csie.ntu.edu.tw/~robin/courses/3dcg08/ppt/3dcg08_04view.pdf

BusinessDictionary. 2015. What is interactive? [viitattu 8.1.2015]. Web Finance, Inc. Saatavissa: <http://www.businessdictionary.com/definition/interactive.html>

Compu Phase. 2011. Axonometric projections [viitattu 14.1.2015]. Saatavissa: <http://www.compuphase.com/axometr.htm>

Cornell University - Program of computer graphics. 1998. What is graphics? [viitattu 22.12.2014]. Cornell University. Saatavissa: <http://www.graphics.cornell.edu/online/tutorial/>

Digital-tutors. 2014. Understanding global illumination [viitattu 23.2.2015]. Pluralsight. Saatavissa: <http://blog.digitaltutors.com/understanding-global-illumination/>

Elitisti. 2011. Varhainen 3D [viitattu 6.1.2015]. Elitisti-toimitus. Saatavissa: http://www.elitisti.net/artikkeli/2011/12/019602/elitistinen_katsaus_3d_elokuviin.html

Giantbomb. 2014a. 2.5D [viitattu 15.1.2015]. CBS Interactive Inc. Saatavissa: <http://www.giantbomb.com/25d/3015-660/>

Giantbomb. 2014b. Isometric viewpoint [viitattu 14.1.2015]. CBS Interactive Inc. Saatavissa: <http://www.giantbomb.com/isometric-viewpoint/3015-246/>

Giantbomb. 2014c. Secon-person perspective [viitattu 21.1.2015]. CBS Interactive Inc. Saatavissa: <http://www.giantbomb.com/second-person-perspective/3015-2722/>

Giantbomb. 2015a. First-person perspective [viitattu 21.1.2015]. CBS Interactive Inc. Saatavissa: <http://www.giantbomb.com/first-person-perspective/3015-330/>

Giantbomb. 2015b. Third-person perspective [viitattu 6.2.2015]. CBS Interactive Inc. Saatavissa: <http://www.giantbomb.com/third-person-perspective/3015-464/>

Historia. 2012. Milloin 3D-kuvat keksittiin? [viitattu 6.1.2015]. Bonnier publications. Saatavissa: <http://historianet.fi/kysy-meiltae/milloin-3-d-kuvat-keksittiin>

IT. 2015. Overview of computer graphics [viitattu 29.12.2014]. University of Leeds. Saatavissa:

http://iss.leeds.ac.uk/info/306/graphics/215/overview_of_computer_graphics/2

Learning Modern 3D Graphics Programming. 2012. Perspective projection [viitattu 17.1.2015]. Jason L. McKesson. Saatavissa:

<http://www.arcsynthesis.org/gltut/Positioning/Tut04%20Perspective%20Projection.html>

Mediacollege.com. 2012. What is 3D? [viitattu 29.12.2014]. Wavelength Media. Saatavissa: <http://www.mediacollege.com/3d/intro.html>

OpenGameArt.org. 2015. Chapter 3: Perspectives [viitattu 7.1.2015]. Saatavissa: <http://opengameart.org/content/chapter-3-perspectives>

OpiGIMP. 2013. Polut [viitattu 9.1.2015]. Saatavissa: <http://opigimp.gimp-suomi.org/?o=5>

Processing. 2013. Shaders [viitattu 11.2.2015.] Saatavissa:

<https://processing.org/tutorials/pshader/>

Radbin. 2012. What is concept art [viitattu 23.2.2015.] Randy Robin

Galiban. Saatavissa: <http://www.randbin.com/what-is-concept-art/>

SignificantBits. 2009. A Layman's guide to projection in videogames [viitattu 21.1.2015]. Saatavissa: <http://www.significant-bits.com/a-laymans-guide-to-projection-in-videogames>

Taidegrafiikka. 2003. Mitä taidegrafiikka on? [viitattu 22.12.2014]. Suomen Taidegraafikot ry. Saatavissa:

<http://www.taidegraafikot.fi/taidegrafiikka.html>

TechTerms.com. 2009. Graphics [viitattu 3.1.2015]. Saatavissa:

<http://techterms.com/definition/graphics>

TechTerms.com. 2012. Sprite [viitattu 11.1.2015]. Saatavissa:

<http://techterms.com/definition/sprite>

Tuts+. 2013. An introduction to spritesheet animation [viitattu 11.1.2015].

Envato Pty Ltd. Saatavissa:

<http://gamedevelopment.tutsplus.com/tutorials/an-introduction-to-spritesheet-animation--gamedev-13099>

Webopedia. 2015. Real time [viitattu 8.1.2015]. QuinStreet Inc. Saatavissa:

http://www.webopedia.com/TERM/R/real_time.html

WhatIs.com. 2009. Rendering [viitattu 28.12.2014]. TechTarget. Saatavissa:

<http://whatis.techtarget.com/definition/rendering>

Wikipedia. 2014. Rengering (computer graphics) [viitattu 28.12.2014].

Wikimedia Foundation, Inc. Saatavissa:

http://en.wikipedia.org/wiki/Rendering_%28computer_graphics%29

Wikipedia. 2015a. 2D computer graphics [viitattu 7.1.2015]. Wikimedia

Foundation, Inc. Saatavissa:

http://en.wikipedia.org/wiki/2D_computer_graphics

Wikipedia. 2015b. 2.5D [viitattu 15.1.2015]. Wikimedia Foundation, Inc.

Saatavissa: <http://en.wikipedia.org/wiki/2.5D>

KUVALÄHTEET

Dan Bailey. 2015. Comparison of Depth Map versus Raytraced Shadows

in both Maya and Mental Ray Renderers. [viitattu 10.2.2015]. Saatavissa:

<http://userpages.umbc.edu/~bailey/Courses/Tutorials/RenderingDemo/DepthRayShadowDemo.html>

Digital-tutors. 2014. Understanding global illumination [viitattu 23.2.2015].

Pluralsight. Saatavissa: <http://blog.digitaltutors.com/understanding-global-illumination/>

Donkey-kong.info. Original arcade games [viitattu 5.1.2015]. Saatavissa:

<http://www.donkey-kong.info/games.php>

EA. 2009. SimCity Societies Deluxe Screenshot [viitattu 11.2.2015].
Electronic Arts Inc. Saatavissa: http://www.ea.com/de/simcity-societies-deluxe-edition/images/060840ed034a3210enGB_to_deDEad65140aRCRD

GameFAQs. 2015. Tomb Raider III: Adventures of Lara Croft [viitattu 23.2.2015]. CBS Interactive Inc. Saatavissa:
<http://www.gamefaqs.com/pc/71538-tomb-raider-iii-adventures-of-lara-croft/images/screen-12>

Giantbomb. 2015a. First-person perspective [viitattu 21.1.2015]. CBS Interactive Inc. Saatavissa: <http://www.giantbomb.com/first-person-perspective/3015-330/>

Mario Universe. 2015. Enemies [viitattu 11.1.2015]. Saatavissa:
<http://www.mariouniverse.com/images/sprites/nes/dk/enemies.png>

Mr. Rush's site. 2015. Axonometric [viitattu 9.2.2015]. Edline. Saatavissa:
<http://www.ltps.org/webpages/lrush/index.cfm?subpage=888059>

Pongmuseum.com. 2015. The First "Electronic" Game Ever Made [viitattu 5.1.2015]. Saatavissa:
<http://pongmuseum.com/history/FirstElectronicGameEverMade.php>

ScottDutton/Catspaw dynamics. 2004. View-Master tribute poster [viitattu 6.1.2015]. Saatavissa: <http://www.catspawdynamics.com/view-master-tribute-poster/>

StackExchange. 2011. Draw a prism in TikZ or PSTricks [viitattu 10.2.2015]. Stack Exchange Inc. Saatavissa:
<http://tex.stackexchange.com/questions/37442/draw-a-prism-in-tikz-or-pstricks>

Tuts+. 2011. Working with Orthographic Projections and Basic Isometrics [viitattu 9.2.2015]. Envato Pty Ltd. Saatavissa:
<http://design.tutsplus.com/tutorials/working-with-orthographic-projections-and-basic-isometrics--vector-893>

Viewing history. 1995. Perspective projections: Vanishing points [viitattu 11.2.2015]. Brown University. Saatavissa:

http://cs.brown.edu/stc/summer/viewing_history/viewing_history_27.html

Wikia. 2012. Borderlands wiki [viitattu 23.2.2015]. Saatavissa:

<http://de.borderlands.wikia.com/wiki/Datei:Borderlands-2-salvador-concept-art-2.jpg>

Wikimedia commons. 2010. File:Bump-map-demo-full.png [viitattu

10.2.2015]. Saatavissa: <http://commons.wikimedia.org/wiki/File:Bump-map-demo-full.png>

Wikipedia. 2007. File:Qbert.png [viitattu 13.2.2015]. Wikimedia

Foundation, Inc. Saatavissa: http://en.wikipedia.org/wiki/Q*bert