

KYMENLAAKSON AMMATTIKORKEAKOULU

Viestinnän koulutusohjelma / Digitaalinen media

Toni Mattila

PELIPROJEKTI NEMESIS

Opinnäytetyö 2014

TIIVISTELMÄ

KYMENLAAKSON AMMATTIKORKEAKOULU

Viestintä

MATTILA, TONI

Peliprojekti Nemesis

Opinnäytetyö

31 sivua

Työn ohjaaja

Marko Siitonen

Lokakuu 2014

Avainsanat

pelisuunnittelu, pelihahmot, pelit, peligrafiikka

Opinnäytetyö käsittelee ryhmätyönä toteutettua peliprojektia ja sen vaiheita. Projektin työryhmään kuuluvat Kymenlaakson ammattikorkeakoulun digitaalisen median opiskelijat Henri Nuuttila ja Toni Mattila, jotka vastasivat pelin kolmiulotteisesta toteutuksesta ja Taija lumitähti, joka vastasi pelihahmojen konseptista ja kaksiulotteisesta toteutuksesta. Lisäksi ryhmään liittyivät myöhemmin ohjelmoijat Ilmari Karulinna ja Jaakko Kauranen, jotka vastasivat pelin lähdekoodin tuottamisesta. Projekti aloitettiin tammikuussa 2013 osana Finnish Game Jam – pelitapahtumaa. Game Jam –tapahtuman aikana tuotettiin yksinkertainen shakista inspiroitunut vuoropohjainen strategiapeli, jonka jatkokehitys alkoi kesällä 2013.

Kesän aikana peliin tuotettiin kokonaan uudet kolmiulotteiset pelihahmot pintamateriaaleineen ja animaatioineen, uusittu pelikenttä ja laaja pelisuunnitelma. Peli sai työnimekseen Nemesis. Syksyn aikana peliin tuotettiin uusittu lähdekoodi, joka kuitenkin jäi keskeneräiseksi risteävien aikataulujen vuoksi.

Oma osa-alueeni Nemesis peliprojektissa oli yhteisen pelisuunnittelun lisäksi mallintaa puolet kolmiulotteisista pelihahmoista ja tuottaa niihin animaatiot. Pelin hahmot, pelikenttä ja animaatiot tuotettiin käyttäen hyväksi Autodeskin 3ds Max -ohjelmaa. Pelimoottoriksi valitsimme ilmaisen Unity Game Enginen, josta työryhmällä oli kokemusta kurssien ja aiempien projektien kautta. Lisäksi suoritin laajan pelisuunnitelman eli Game Design Documentin käännöksen englanniksi.

Peliprojektina Nemesis oli mielestäni yleisesti onnistunut. Nemesis oli alunperinkin ammatillista osaamista edistäväksi tarkoitettu projekti, ja harjoitusluonteisena se täytti tarkoituksensa. Pelisuunnitelma, pelihahmot ja pelikenttä valmistuivat aikataulun mukaan. Pelin lähdekoodin tuottaminen eteni tiettyyn pisteeseen asti, mutta jäi lopulta kesken. Projekti antoi työryhmälle kokemusta ja kuvaa siitä mitä pelialalla työskentely voi olla.

ABSTRACT

KYMENLAAKSON AMMATTIKORKEAKOULU

University of Applied Sciences

Media Communication

MATTILA, TONI

Game Project Nemesis

Bachelor's Thesis

31 pages

Supervisor

Marko Siitonen, Lecturer

October 2014

Keywords

game design, game characters, games, game graphics

This thesis covers the documentation of a videogame project made by a group of students at Kymenlaakso University of Applied Sciences. The work group consists of students Henri Nuuttila and Toni Mattila, who developed the 3D assets for the game and Taija Lumitähhti, who developed the character design and made all the 2D assets of the game. The source code was developed by programming students at Kouvolan seudun ammattiopisto, Kouvola Region Vocational College Ilmari Karulinna and Jaakko Kauranen who joined the project later. The project initially started in 2013 as a part of the Finnish Game Jam game development event. As a result of the event, a simple turn-based chess inspired strategy game was created. The game was further developed later on during the summer of 2013.

Completely new versions of the 3D characters, game level and the game design were completed by the end of the summer. A game design document was also written containing all the essential information about the game. The project was given the working title Nemesis. The source code of the game was developed during the autumn of 2013, but was not entirely completed due to problems with the schedule.

Apart from the joint work with the design of the game, my part of the project was to model and animate half of the 3D characters in the game. All the 3D assets and animations were made using Autodesk 3ds Max 3D modeling program. Unity Game Engine was chosen as the game engine for the game, as the working group had experience of using it from previous projects and courses. The thesis author's part was also to carry out the translation of the game design document to English.

I found Nemesis to be a rather successful project. The purpose of the project as a practical work experience and a help in further developing our professional skills were fulfilled. The game design document, the characters and the level of the game were all completed within the schedule and the production of the source code was developed to a point where the game was playable, but was left unfinished. The project as whole, gave the working group plenty of experience as to what working in the game industry might be.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO	5
2	PROJEKTIN TAUSTAT	5
	2.1 Projektin aloitus	5
	2.2 Toteutuksen lähtökohdat	7
3	SUUNNITTELU JA DOKUMENTOINTI	8
	3.1 Suunnitteluprosessi	8
	3.2 Pelin idea	10
	3.3 Pelin kulku	12
	3.4 Pelattavat hahmot	14
	3.5 Viholliset	19
4	PELIHAHMOJEN TOTEUTUS	22
	4.1 Hahmojen kolmiulotteinen toteutus	22
	4.2 Pelihahmojen animaatiot	27
	4.3 Pelimoottori	29
	4.4 Ohjelmointi ja jatkokehitys	30
5	LOPPUTULOKSET	30
6	LÄHTEET	32

1 JOHDANTO

Tässä tekstissä käsittelen ryhmätyönä toteutettua peliprojektia, sen vaiheita ja toteutusta. Kyseessä on ammatillista osaamista edistäväksi tarkoitettu harjoitusmuotoinen peliprojekti. Projekti toteutettiin Kymenlaakson ammattikorkeakoulun viestinnän koulutusohjelman digitaalisen median ryhmän Vi10Di opiskelijoiden Henri Nuuttilan, Taija Lumitähden ja Toni Mattilan toimesta.

Projektissa ei ollut varsinaista ulkopuolista toimeksiantajaa. Projekti oli omalähtöinen ja lähtökohtaisesti oppimista edistäväksi tarkoitettu käytännön harjoitustyö. Projekti jaettiin ryhmän kesken työalueisiin, ja tämä teksti painottuu projektin dokumentointiin. Käsittelen tekstissä projektin lähtökohtia ja taustoja, itse peliä ja omaa työaluetani sen toteutuksessa. Projektin suunnittelutyö on tehty pääosin yhteisesti työryhmän kesken.

2 PROJEKTIN TAUSTAT

2.1 Projektin aloitus

Peliprojektimme lähti alunperin liikkeelle Global Game Jam -tapahtumasta tammi-kuussa 2013. Game Jam -tapahtuma on maailmanlaajuinen pelialan tapahtuma, joka järjestetään samanaikaisesti useassa eri maassa ja useissa eri tapahtumapaikoissa. Game Jam on innovatiivisuutta ja uusia ideoita painottava pelinkehitystapahtuma pelialalla työskenteleville, pelialan opiskelijoille ja pelialasta kiinnostuneille. Tapahtumaan ovat tervetulleita niin uudet tekijät kuin veteraanitkin osaamistasosta riippumatta. Game Jam -tapahtumassa on tarkoitus oman tiimin voimin luoda peli 48 tunnissa. Tiimi koostuu usein pelialalla tutuista rooleista, kuten grafiikan tuottajista, ohjelmoijista ja mahdollisesti ääniefektien tuottajasta. Vuosittain sadat tiimit tuottavat Game Jam -tapahtumissa tuhansia pelejä, jotka sivuavat tiettyä vuosittain vaihtuvaa teemaa. (Global Game Jam. 2014.)

Osallistuimme Finnish Game Jam 2013 kymenlaakso -tapahtumaan, joka järjestettiin Kouvolan seudun ammattiopiston liiketalouden yksikön tiloissa. Ryhmäämme kuuluivat itseni lisäksi Henri Nuuttila ja Taija Lumitähhti, joiden kanssa olin jo aiemmin kurssien yhteydessä toteuttanut osittaisen peliprojektin. Pelit ja pelaaminen ovat kaikille ryhmän jäsenille yhteinen harrastus ja kiinnostuksen kohde. Game Jam -

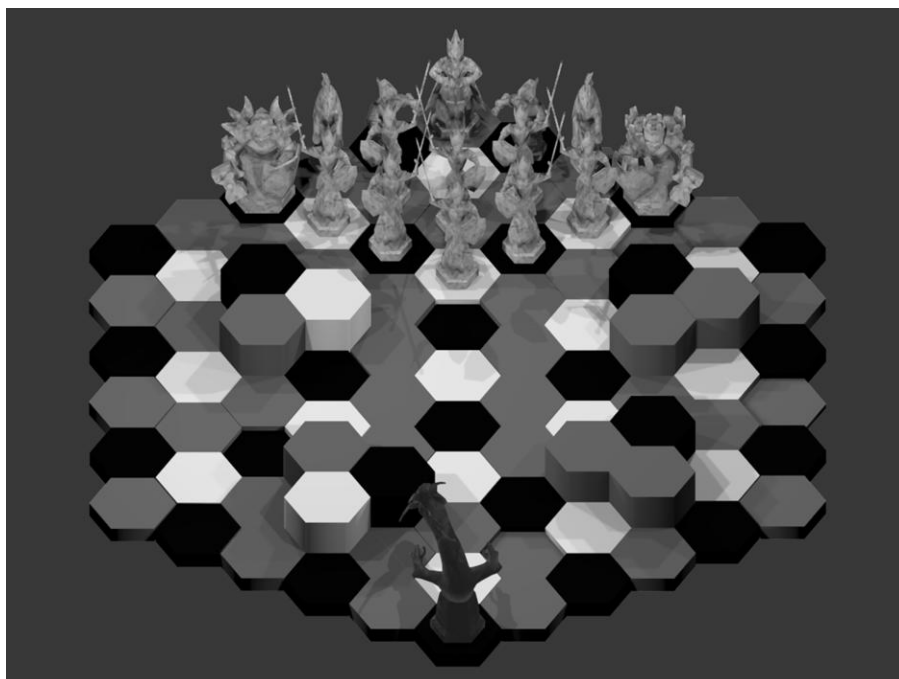
tapahtuman aikana oma työalueeni Nuutilan kanssa oli pelihahmojen ja pelikentän kolmiulotteinen mallinnus, Lumitähti suunnitteli hahmojen ulkoasun ja toteutti kaiken 2D-kuvituksen ja visuaalisen konseptin. Lisäksi ryhmäämme kuului Game Jam -tapahtuman ajan opiskelija Lauri Kuparinen, joka vastasi pelin ohjelmakoodin tuottamisesta.

Game Jam -tapahtuman aikana suunnittelimme ja toteutimme 3D-tietokonegrafiikkaa hyödyntäen pelin, jolle annoimme nimen HeXart. Pelimme oli shakista vaikutteita ottanut vuoropohjainen strategiapeli, jossa pelaaja ohjaa hahmoja kuusikulmaisista ruuduista muodostuvalla pelikentällä (kuvat 1 ja 2). Tavoitteena pelissä on yksittäisen voimakkaan vihollisen tuhoaminen erilaisista hahmoista koostuvalla joukolla pelattavia hahmoja.

Game Jam -tapahtuman 48 tunnin työrupeaman aikana saimme aikaan yksinkertaisen version visioimastamme pelistä. Pelin ydinmekaniikat olivat pelattavissa, mutta suurin osa käyttäjäpalautteesta, visuaalisuudesta ja käyttöliittymäelementeistä puuttuivat. Lisäksi peli sisälsi paljon niin sanottuja bugeja, eli virheitä pelin ohjelmakoodissa, jotka saattoivat tehdä pelin satunnaisesti pelikelvottomaksi tai aiheuttaa ei-toivottuja asioita pelin toiminnassa tai pelihahmojen käyttäytymisessä.



Kuva 1. Game Jam peli HeXart



Kuva 2. Ruutukaappaus alkuperäisestä pelimme Game Jam -versiosta

2.2 Toteutuksen lähtökohdat

Koska yksi viikonloppu on tavattoman lyhyt aika pelin kehittämiseksi, päätimme myöhemmin yhdessä Taija Lumitähden ja Henri Nuutilan kesken jatkaa pelin kehittämistä. Kokonaan uuden peliprojektin aloittaminen ei tuntunut hyvältä vaihtoehdolta, koska Game Jamin aikana toteutetussa pelissä oli jo valmiina peli-idea ja tietynlainen pohjatyö. Annoimme pelille projektin alkaessa uuden työnimen Nemesis. Pelin suunnittelu ja tuottaminen antaisi mahdollisuuden hyödyntää kaikkia opiskelun aikana oppimiamme taitoja hahmosuunnittelusta 3D-mallintamiseen ja konseptikuvittamisesta pelin varsinaiseen koodaamiseen.

Kesän 2013 aikana laadimme Nemesis-peliin kokonaan uuden pelisuunnitelman, hahmosuunnitelmat ja visuaalisen konseptin. Sen lisäksi tuotimme kokonaan uudet 3D-mallit pelin hahmoista ja pelikentästä animaatioineen. Tavoitelaitteistoksi ja alustaksi pelille määritettiin nykyaikainen keskiverto PC. Työskentelimme päivittäin ryhmänä, keskittyen kukin omaan työalueeseemme. Projektia organisoidaksemme ja pitääksemme pelin designin yhtenäisenä järjestimme viikoittain kokouksia. Näissä kokouksissa kävimme läpi projektin edistymistä ja laadimme aikatauluja arvioiduista projektin vaiheiden valmistumisesta. Oma ryhmämme keskittyisi lähinnä pelin visuaalisuuteen, konseptiin ja sen rakennuspalikoiden, kuten kolmiulotteisten

pelihahmojen tuottamiseen. Koska pelihahmot ja kenttä eivät yksinään tee mitään ilman pelin toiminnallisuudet luovaa ohjelmakoodia, olisi myös tämä tärkeä työalue projektista saatava toteutettua, jotta peliämme voisi kutsua peliksi. Oman koulutuksemme painottuessa juuri pelituotannon visuaalisiin osa-alueisiin tarvitsimme ulkopuolista apua pelin ohjelmoimiseksi.

Game Jam -tapahtuman aikana meihin otti yhteyttä kouvolaalaisen pelistudio Blight Gamesin edustaja, Mika Syrjänen, joka oli kiinnostunut pelistämme. Blight Games Oy työsti tuolloin mobiilipeliä, joka oli tyyliltään samankaltainen omamme kanssa. Meille ehdotettiin, että tekisimme yhteistyötä jakaen osaamistamme toistemme peleihin. Meillä oli ryhmänä tarjota visuaalista näkemystä ja ideoita, kun taas meille luvattiin apua oman pelimme koodin kanssa. Kun projektin jatkokehitys myöhemmin aloitettiin, otimme heihin yhteyttä yhteistyöhön liittyen. Aluksi kävimme viikoittain kokouksissa Blight Games Oy:n tiloissa samalla omaa peliämme työstäen. Nopeasti kuitenkin selvisi, ettei heidän aikataulunsa päätoimisen peliprojektinsa kanssa sallinut heille aikaa tuottaa koodia peliimme. Yhteistyö jäi Nemesis-projektin osalta valitettavan lyhyeksi. Päädyimme kuitenkin myöhemmin tuottamaan 3D-malleja Blight Gamesin päätoimiseen Black Stairs -kauhupeliin.

3 SUUNNITTELU JA DOKUMENTOINTI

3.1 Suunnitteluprosessi

Nemesis-peliprojekti eteni kuitenkin omissa käsissämme hyvin. Peli-ideana shakista inspiroitunut strategiapeli ei ollut mitään uutta, mutta alunperin ajattelimme sen olevan hyvä ja tarpeeksi yksinkertainen toteutettavaksi Game Jam -tapahtuman aikarajan puitteissa ja myöhemmin harjoitusprojektina. Alusta asti oli selvää, että halusimme tehdä pelistä oppimista edistävän harjoitusprojektin. Pelin kaupallisuus olisi toissijainen tavoite. Myöhemmin, pelin valmistuttua, olisi mahdollista suunnitella pelistä kaupallisesti kannattavampi, esimerkiksi mobiililaitteille optimoitu versio ja sisällyttää siihen pelinsisäisillä mikromaksuilla hankittavia lisäkenttiä ja hahmoja.

Suunnittelun lähtökohtana käytimme pelin olemassa olevaa versiota, jonka pohjalta aloimme suunnitella parannuksia uudistettuun Nemesis-peliin. Pelin vanhan version hahmomallit ja pelikenttä olivat hyvin tiukalla aikataululla luotuja, eikä niiden laatu täyttänyt vaatimuksiamme. Taija Lumitähti suunnitteli vanhojen hahmojen pohjalta

täysin uudistetut hahmokonseptit, joiden pohjalta uudet hahmot mallinnettaisiin. Suunnittelimme ryhmänä hahmojen erikoiskyvyt ja voimatasot. Tasapainotimme ja hioimme jokaista hahmoa ja kykyjä projektin edetessä. Tavoitteena oli saada jokaisesta hahmosta erilainen, niin kyvyiltään kuin ulkonäöltäänkin.

Pelillisiä uudistuksia pohtiessamme totesimme yhden vihollisen ja pelikentän käyvän pelaajalle nopeasti tylsäksi. Myös pelihahmojen määrä ja asettelu oli Game Jam -versiossa vakio, eikä antanut pelaajalle mitään vapautta valintoihin. Halusimme pelaajalla olevan valittavanaan joukon erilaisia hahmoja joista luoda mielensä mukainen joukko. Hahmojen monipuolisuuden ja niiden yhdistelemien ajattelimme olevan tärkeä tekijä pelin uudelleenpelattavuuden kannalta. Hahmojen valintaa varten suunnittelimme kokonaan uuden valintaruudun, jossa hahmot ostettaisiin osaksi joukkoa värväyspisteitä käyttäen. Pisteiden lisääminen mahdollisti myös vaikeusasteiden lisäämisen osaksi peliä. Pelattavien kenttien lukumäärä nousi yhdestä kolmeen kahden uuden vihollishahmon myötä.

Suunnittelimme huomattavia pelimekaanisia parannuksia myös pelin käyttöliittymän suhteen. Pelin aikaisemmasta versiosta puuttuivat kokonaan käyttöliittymäelementit, jotka antavat pelaajalle palautetta pelin kulusta. Pelin uudistettua versiota varten luonnostelin ulkoasun erääseen tärkeimmistä käyttöliittymäelementeistä, hahmojen valintaruutuun. Käyttöliittymäelementtien toteutusta jarrutti projektin edetessä pelimoottorina käyttämämme Unityn rajoitukset kaksikulotteisten elementtien luomiseen. Työryhmällä oli tiedossa Unity-pelimoottoriin suunniteltu päivitys, joka tekisi käyttöliittymän luomisesta helpompaa. Tuolloin erityisesti pelissä esiintyvien hahmokorttien luominen oli haastavaa, ja päivityksen viivästyisestä johtuen käyttöliittymäelementit jäivät pääosin luonnosasteelle.

Syksyllä työryhmäämme liittyivät Kouvolan seudun ammattiopiston pelialan opiskelijat Ilmari Karulinna ja Jaakko Kauranen, jotka alkoivat tuottaa ja muokata pelimme lähdekoodia. Tähän mennessä suurin osa pelin visuaalisista elementeistä, pelikenttä ja pelin hahmot, olivat jo valmiina. Ilman ohjelmakoodia nämä pelin rakennuspalikat eivät itsessään tehneet mitään, ja koodi ja sen tuottaminen oli peliprojektimme vakavin puutos. Koska alunperin suunniteltu yhteistyö Blight Gamesin kanssa kariutui, viivästyi pelimme ohjelmoinnin aloitus valitettavan paljon suhteessa muihin projektin työalueisiin. Alkuperäisen työryhmän koulutuksen

painottuessa visuaalisten elementtien suunnitteluun ja toteutukseen oli erittäin tärkeää, että saimme ulkopuolista apua ohjelmakoodin osalta.

Laadimme peliä varten myös laajan pelisuunnitelman eli Game Design Documentin (GDD). Laaja pelisuunnitelma kuuluu jokaiseen ammattimaiseen peliprojektiin ja sisältää kaiken olennaisen tiedon pelistä (Ryan 1999). Tähän kuuluvat niin pelin idea, taustatarina ja hahmot kuin myös pelin ydinmekaniikat ja toiminnallisuudet.

Pelisuunnitelman on pelin dokumentoinnin lisäksi tarkoitus antaa mahdollisimman paljon tietoa pelistä mahdollisille ulkopuolisille tahoille, jotka peliä mahdollisesti jatkokehittävät. Pelin sisäinen kehitystiimi saattaa myös muuttua pelin kehityksen aikana, jolloin pelisuunnitelma auttaa uutta ryhmän jäsentä pääsemään sisään projektiin. (Bethke 2003, 102-105.) Laaja GDD laadittiin ensin yhdessä suomeksi projektin kuluessa, tämän jälkeen suoritin käännöksen englanniksi, joka on yleisesti käytetty työkieli pelialalla.

3.2 Pelin idea

Nemesis on peligenreltään vuoropohjainen strategiapeli, jossa pelaaja ohjaa pelattavia hahmoja taistelussa vihollista vastaan. Pelaajalla on valittavana useita erilaisia hahmoja, joista kaikilla on erilaiset vahvuudet, heikkoudet ja erikoiskyvyt. Pelin tavoitteena on muiden pelattavien hahmojen avulla suojella Keisari-hahmoa ja tuhota tietokoneen ohjaama vihollishahmo.

Keisari-hahmo eli Emperor on pelattavista hahmoista tärkein ja on aina kiinteänä osana pelihahmojoukkoa. Emperor-hahmon lisäksi pelaajalla on valittavana viisi erilaista hahmoa: Guardian, Scout, Soldier, Stallion ja Tower. Jokaisella hahmolla on erilaiset kykytasot elämäpisteissä, hyökkäysvoimassa ja nopeudessa. Elämäpisteet määrittävät kuinka paljon vahinkoa hahmo kestää tuhoutumatta. Hyökkäysvoima taas määrittää kuinka paljon vahinkoa hahmo tekee hyökätessään. Vahingon tekeminen vähentää elämäpisteitä. Hahmojen taso nopeudessa määrittää kuinka monta ruutua hahmo voi liikkua pelikentällä vuoron aikana. Jokaisella hahmolla on lisäksi yksi erikoiskyky, joka määrittää eri hahmojen rooleja. Jokainen hahmo maksaa lisäksi eri määrän väräyspisteitä.

Pelaaja valitsee näistä hahmoista mieleisensä joukon käytössä olevien väräyspisteiden puitteissa. Tietyt hahmot maksavat enemmän pisteitä kuin toiset,

joten pelaajalla on mahdollisuus luoda hyvin erilaisia joukkoja ja erilaisia taktiikoita. Pelaajalla on pelin alkaessa käytössään tietty määrä pisteitä, joiden määrä määräytyy pelaajan valitseman vaikeusasteen mukaan. Mitä korkeampi pelin vaikeusaste on, sen vähemmän pelaajalla on värväuspisteitä käytössään. Tämä tekee jokaisesta valinnasta tärkeän antaen suuremman haasteen ja vähemmän varaa virheille.

Pelaajan hahmojoukkoa vastassa ovat tietokoneen ohjaamat viholliset Lich Lord, Behemoth ja Dragon, jotka ovat toinen toistaan voimakkaampia ja joista pelaaja kohtaa yhden kerrallaan pelin kolmessa eri kentässä. Kuten pelaajan ohjaamalla pelihahmoilla, myös jokaisella tietokoneen ohjaamalla vihollisella on omat vahvuutensa, voimatonsa ja kykynsä. Pelaajalla on taistelukentällä määrällinen ylivoima vihollista vastaan, mutta jokainen vihollinen on pelaajahahmoihin nähden ylivoimainen voimassa ja kestävyudessa. Vihollinen saattaa tuhota helposti pelaajahahmoja yhdellä iskulla, ja pelaajan on liikutettava hahmojaan taktisesti ja harkiten, jotta vihollisen voittaminen olisi mahdollinen.

Tyylillisesti peli muistuttaa yksinkertaista lautapeliä. Pelikenttä muodostuu kuusikulmion muotoisista ruuduista, joita pitkin hahmojen ja vihollisten liikkuminen tapahtuu. Pelihahmot muistuttavat ulkoisesti patsaita ja liikkuvat pelinappuloiden tavoin. Pelin visuaalinen ilme on hyvin minimalistinen ja hieman abstrakti. Väripaletti koostuu lähes kokonaan harmaasävyistä ja muutamasta tehosteväristä.

Yksi pelin tärkeimmistä elementeistä ovat hahmokortit (kuva 3). Pelissä hahmot esiintyvät kolmiulotteisten mallien lisäksi hahmokortteina. Jokaisella hahmolla on oma hahmokorttinsa, joka tulee näkyviin, kun kyseinen hahmo on aktiivinen. Hahmokortissa on näkyvillä kaikki oleellinen tieto hahmosta, sen kykytasoista ja kestopisteistä. Lisäksi jokaisessa hahmokortissa on visuaalisena elementtinä digitaalisesti maalattu kuva kustakin hahmosta. Hahmokortit ovat pelin aikana näkyvistä käyttöliittymäelementeistä tärkein ja ne sisältävät suurimman osan pelaajalle olennaisesta informaatiosta.



Kuva 3. Esimerkki hahmokortista

3.3 Pelin kulku

Ennen pelin alkamista pelaaja valitsee pelin vaikeustason. Vaikeustaso määrittää kuinka monta pistettä pelaajalla on käytettävänä hahmojen valitsemiseen. Tämän jälkeen pelaaja valitsee hahmot, joita hän ohjaa pelissä. Pelaajan aloittaessa uuden pelin, aukeaa niin sanottu värväysruutu (kuva 4). Tässä näkymässä pelaaja valitsee hahmojen joukosta haluamansa yhdistelmän pelattavia hahmoja. Pelaajan valintaa rajoittaa käytössä olevien värväyspisteiden määrä. Alkuperäisessä versiossa tätä hahmojen valintaa ei ollut lainkaan. Suunnittelemamme värväysruutu värväyspisteineen oli selkeä kehitysaskel.

Hahmojen valinnassa pelaajalle tuodaan esiin hahmokorttien merkitys pelissä. Pelaaja valitsee hahmot ruudun alareunassa olevista hahmokorteista. Valitut hahmot tulevat näkyviin niin ikään hahmokorteina ruudun yläreunaan eräänlaisena korttipakkamaisena yleissilmäyksenä pelaajan valitsemasta hahmojoukosta, jota pelaaja voi edelleen muokata mieleisekseen käytettävissä olevien pisteiden puitteissa. Hahmokortteja valitessa ruudun reunaan tulisi myös näkyviin kyseessä olevan hahmon 3D-malli. Pelin varsinainen ensimmäinen kenttä alkaa hahmojen valinnan jälkeen.

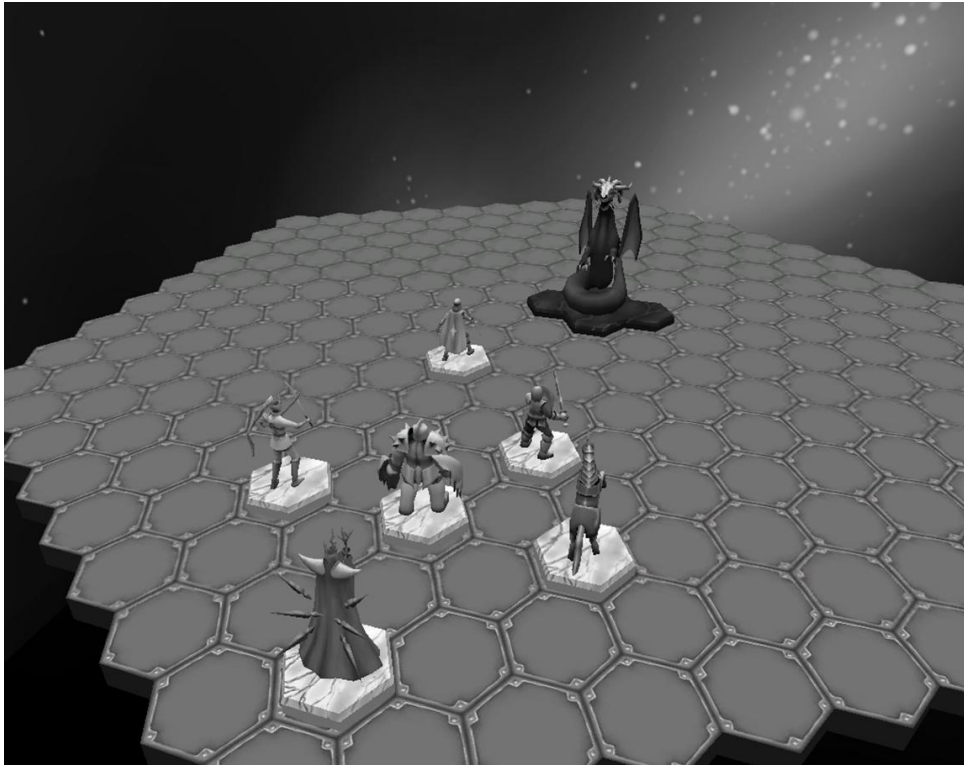


Kuva 4. Luonnos hahmojen valintaruudusta

Pelin alkaessa pelaajan valitsemat hahmot ja vihollinen ovat sijoitettuna vastakkain kuusikulmaisista ruuduista muodostuvalle pelikentälle (kuva 5). Pelin varsinainen kulku on jaettu pelaajan ja tietokoneen ohjaaman vihollisen vuoroihin. Pelaajan tehtävä on tietokoneen ohjaaman vihollisen tuhoaminen omaa joukkoaan käyttäen. Tätä tavoitetta varten pelaaja voi vuoronsa aikana joko liikkua tai hyökätä. Pelaaja voi liikuttaa vuorossa vain yhtä hahmoa tai hyökätä yhdellä hahmolla. Tietokoneen ohjaama vihollinen niin ikään liikkuu kohti pelaajan hahmoja ja hyökkää kun siihen on mahdollisuus. Aivan kuten pelaajan tehtävä on vihollisen tuhoaminen, vihollishahmon päätavoite on Emperor-hahmon tuhoaminen, ja pelaajan on myös muistettava suojella sitä muiden hahmojen avulla. Jos Emperor-hahmo tuhoutuu häviää pelaaja pelin. Pelaajan voittaessa vihollisen alkaa pelissä seuraava kenttä, kunnes viimeinenkin vihollinen on voitettu pelin viimeisessä kentässä. Pelikenttien välillä pelaajalla on mahdollisuus muokata hahmoujoukkoaan uuteen haasteeseen sopivaksi värväysruudulla.

Pelaaja ohjaa hahmoja pelissä tietokoneen hiirtä käyttäen. Pelaaja valitsee hahmon klikkaamalla sitä, jolloin hahmosta tulee aktiivinen ja sen hahmokortti tulee näkyviin näytölle. Kun hahmo on aktiivinen, pelaaja voi klikata ruutua pelikentällä hahmon liikkumisalueen sisäpuolella liikuttaakseen hahmon kyseiseen ruutuun. Kun hahmo aktivoidaan, ruudut, joihin hahmon on mahdollista liikkua korostuvat tehostevärillä.

Myös hyökkäys viholliseen tapahtuu klikkaamalla vihollista aktiivisella hahmolla. Hahmon on oltava viereisessä ruudussa viholliseen nähden hyökätäkseen. Pelin kameraa kontrolloidaan hiiren oikealla napilla, jolloin kameraa voidaan pyörittää aktiivisen hahmon ympärillä. Kameran etäisyyttä voidaan lähentää tai loitontaa hiiren rullasta.



Kuva 5. Ruutukaappaus pelikentältä

3.4 Pelattavat hahmot

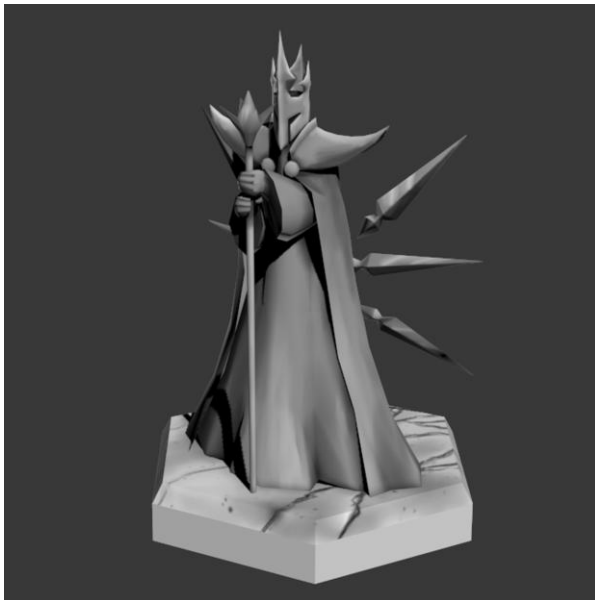
Pelin lautapelimäisyydestä johtuen suunnittelimme myös pelin hahmot pelinappulamaisiksi. Hahmot eivät kuitenkaan ole täysin staattisia pelinappuloita. Koska kyseessä on peli, saatoimme rikkoa oikean maailman rajoja. Suunnittelimmekin hahmoista eräänlaisia eläviä patsaita. Täysin uudistetut pelihahmot ja pelikenttä tuotettiin Taija Lumitähden konseptikuituksen mukaan (kuva 6).



Kuva 6. Pelin hahmotyypit ja viholliset

3D-mallit ja animaatiot uudistettuihin pelihahmoihin työstimme yhdessä Henri Nuuttilan kanssa. Oma työalueeni oli tehdä 3D-mallit ja animaatiot Lich Lord, Dragon, Guardian, Scout ja Stallion-hahmoista. Emperor, Soldier, Tower ja Behemoth ovat Henri Nuuttilan 3D-mallintamia ja animoimia hahmoja. Taija Lumitähti suunnitteli ja maalasi hahmojen pintamateriaalit ja hahmokortit.

Emperor on pelaajan hahmoista tärkein (kuva 7). Jos se tuhoaan on peli ohi. Emperor-hahmon suojeleminen muiden hahmojen avulla on yksi pelin tärkeimpiä asioita vihollisen tuhoamisen lisäksi. Emperor on aina automaattisesti osana pelaajan hahmojoukkoa. Pelaajan valitessa hahmoja on se aina lukittuna yhdeksi joukon jäseneksi. Se on voimaltaan, nopeudeltaan ja elämäpisteiltään keskivertoa heikompi, eikä ole erityisen hyvä taistelukyvyiltään. Emperor-hahmon rooli onkin lähinnä toimia muiden hahmojen tukena. Se voi erikoiskykynään nostaa ylös pelikentän muodostavia kuusikulmionmuotoisia ruutuja muodostamaan esteitä viholliselle. Pelaaja itse aktivoi Emperorin erikoiskyvyn. Ruutujen nostaminen jäi pelimekaniikkana toistaiseksi suunnitteluasteelle.



Kuva 7. Emperor

Guardian on pelattavista hahmoista voimakkain ja sen rooli on toimia pelattavan hahmojoukon eliittinä (kuva 8). Sen kaikki kykytasot ovat korkeat tehden siitä erittäin tehokkaan ja kestävän hahmon. Guardianin ainoa heikkous on sen korkea värväysmaksu. Guardianin passiivisena erikoiskykynä on sen mahdollisuus jokaisella lyönnillä tehdä kaksinkertainen määrä vahinkoa viholliseen tai tehdä vihollinen tilapäisesti toimintakyvyttömäksi.



Kuva 8. Guardian

Soldier on pelattavan hahmojoukon selkäranka (kuva 9). Se maksaa hahmoista vähiten väräyspisteitä, muttei myöskään ole erityisen voimakas, kestävä tai nopea. Soldier-hahmon erikoiskyky tekee siitä voimakkaamman jokaista vierekkäisessä peliruudussa olevaa toista Soldier-hahmoa kohden. Sen ulkonäkö on yksinkertainen ja koristelematon.



Kuva 9. Soldier

Scout on hahmoista nopein ja voi liikkua suurimman määrän ruutuja yhdessä vuorossa (kuva 10). Vastapainona sen suurelle nopeudelle Scout tekee vain vähäisen määrän vahinkoa, eikä itse kestä kovin paljoa vahinkoa tuhoutumatta. Scout-hahmo voi erikoiskykynään liikkua vihollisen luo, lyödä vihollista ja liikkua vielä tietyn matkaa vihollisesta pois päin saman vuoron aikana. Scout on kooltaan melko pieni ja sen ulkonäössä korostuu hahmon nopeus ja ketteryys. Hahmoon on haettu vaikutteita peleissä usein esiintyvistä varas- ja salamurhaajahahmoista.



Kuva 10. Scout

Stallion on rooliltaan taktisin ja joustavin hahmo (kuva 11). Sen kaikki kyvyt ovat keskivertoja, ja se voi erikoiskykynään ylittää Emperor-hahmon nostattamia esteitä pelikentällä. Stallion-hahmo on tavallaan oma näkemyksemme shakin ratsunappulasta. Etenkin sen erikoiskyky on melko vahvasti inspiroitunut shakista. Stallionin sotaratsumaisessa ulkonäössä korostuu sen kolme sarvea ja haarniskointi.



Kuva 11. Stallion

Tower on pelattavista hahmoista kestävin ja toimii pelissä puolustavana hahmona (kuva 12). Tower-hahmon kestävyystaso on kaikista pelattavista hahmoista korkein. Tower pystyy erikoiskykyään jähmettymään paikalleen muuttuen tuhoutumattomaksi. Pelaaja aktivoi tämän kyvyn, ja kyvyn ollessa aktiivinen ei Tower-hahmo voi liikkua tai hyökätä. Towerin ulkonäössä halusimme korostaa sen kestävyyttä ja roolia puolustavana hahmona. Tower on kooltaan suuri ja mittasuhteiltaan raskas ja lihaksikas. Towerilla on päällään raskas haarniska ja suuri kahteen osaan jaettu kilpi, jota se käyttää sekä hyökkäämiseen että kilvenpuolikkaat yhdistämällä puolustukseen.



Kuva 12. Tower

3.5 Viholliset

Pelaajan hahm joukkoa vastassa on kolme erilaista tietokoneen ohjaamaa voimakasta vihollishahmoa. Pelaaja kohtaa viholliset yhden kerrallaan pelin kolmella eri kentällä. Suunnittelimme vihollishahmojen ulkonäön kuvastamaan niiden voimatasoa ja luonnetta.

Pelin vihollishahmot ovat tietokoneen ohjaamia. Tämä tarkoittaa, että pelin sisäinen ohjelmoitu tekoäly ohjaa vihollisia tiettyjen ohjelmoitujen käyttäytymismallien mukaan. Tekoäly pyrkii tarkkailemalla ympäristöään toteuttamaan sille ohjelmoituja tavoitteita. (Russell & Norvig 2003, 27, 32–58.) Vihollisten tekoälyn ensisijainen

tavoite on Emperor-hahmon tuhoaminen. Tekoäly ohjaa vihollishahmoa joka vuorolla kohti Emperoria yrittäen tarpeen mukaan kiertää ja tuhota pelaajan ohjaamia hahmoja tieltään. Se myös pyrkii välttämään ylivoimaista vastarintaa selviytyäkseen.

Pelin ensimmäisessä kentässä pelaaja kohtaa Lich Lord -hahmon (kuva 13), joka on vihollisista tasoltaan heikoin. Sillä on kolmesta vihollisesta heikoin lyöntivoima ja kestävyys, mutta sen suhteellista heikkoutta tasapainottaa sen erikoiskyky. Lich Lordin erikoisuus on sen kyky liikkua hetkessä mihin tahansa ruutuun pelikentällä. Tämä tekee siitä vaarallisen vihollisen sen pystyessä siirtymään hetkessä Emperor-hahmon viereen tai pois pelattavien hahmojen ulottuvista. Lich Lord -hahmoon haimme vaikutteita shamaaneista ja taikoja käyttävistä aave maagihahmoista, joita esiintyy useissa videopeleissä.



Kuva 13. Lich Lord

Pelaaja kohtaa pelin toisessa kentässä Behemothin, joka on tasoltaan Lich Lordia vahvempi. Se on valtava alkukantainen ja pelottava haarniskoitu jätti, joka henkii olemuksellaan raakaa voimaa (kuva 14). Behemoth on hidas, mutta erittäin voimakas ja kestävä. Se tekee erikoiskykynään vahinkoa kaikkiin ympärillä oleviin pelaajan hahmoihin lyömällä maahan aseellaan. Behemoth ja Dragon eroavat muista pelin hahmoista siinä, että ne suuren kokonsa takia vievät pelilaudalta neljä ruutua yhden

sijaan. Tämä ero aiheutti suuren määrän lisäsuunnittelua ja työtä ja osoittautui lopulta melko haastavaksi toteuttaa koodin kannalta.



Kuva 14. Behemoth

Dragon on pelin todellinen haaste ja pelaajan kohtaamista vihollishahmoista voimakkain ja kestävin (kuva 15). Dragon voi erikoiskykynään laskea Emperor-hahmon nostamia esteitä ja luoda kuoppia, jotka toimivat samaan tapaan kuin Emperorin nostamat ruudut estäen pelaajan ohjaamien hahmojen kulun.

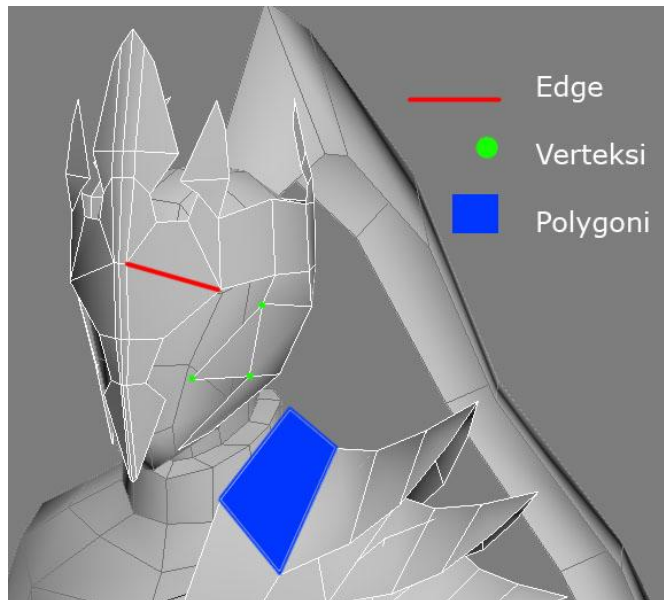


Kuva 15. Dragon

4 PELIHAHMOJEN TOTEUTUS

4.1 Hahmojen kolmiulotteinen toteutus

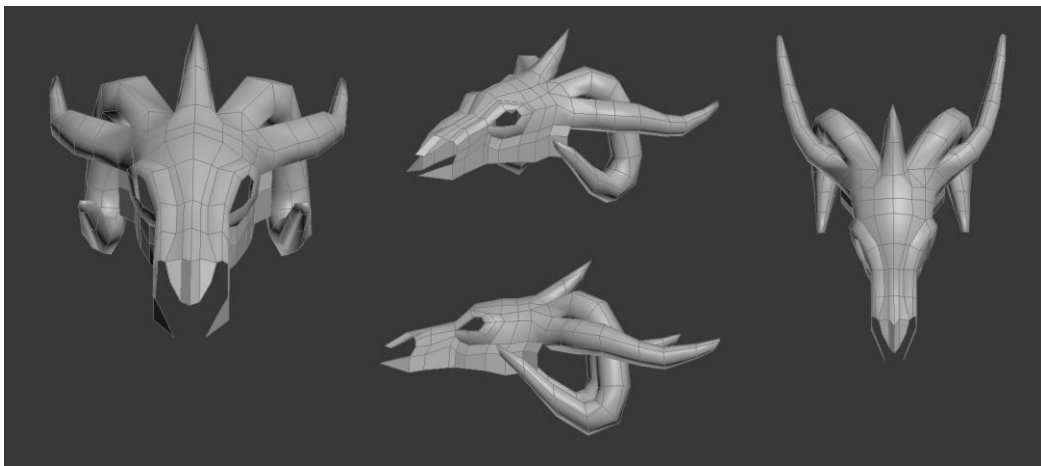
Pelimme on kokonaisuudessaan toteutettu 3D-tietokonegrafiikalla. Käsitteellä 3D tarkoitetaan kolmiulotteista. Tämän projektin yhteydessä käsite viittaa kolmiulotteisiin 3D-mallinnusohjelmilla tuotettuihin esineisiin ja asioihin. 3D-mallintamisella tarkoitetaan kolmiulotteisten mallien tuottamista asiaan tarkoitetuilla tietokoneohjelmilla. 3D-mallit muodostuvat useista litteistä pinta-tasoista eli polygoneista, jotka ovat useimmiten neliön muotoisia. (Mitchell 2012, 126.) Mallintaminen tapahtuu eri tavoin näitä polygonipintoja, polygonin reunoja, eli edgejä, ja polygonin kulmapisteitä, eli verteksejä, lisäämällä, liikuttelemalla ja kääntelemällä (kuva 16).



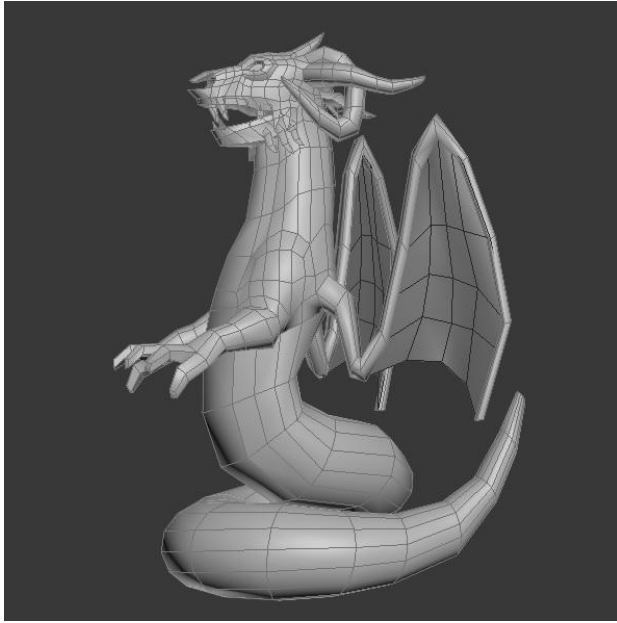
Kuva 16. 3D-mallintamisen perusmuuttujat: verteksi, edge ja polygoni

Käytin pelimme hahmojen mallintamisessa Autodesk 3ds Max -ohjelmaa, jota olemme käyttäneet kurssien puitteissa, ja joka oli sekä tuttu, että tehokas työkalu mallien luomiseen, teksturointiin ja animointiin. Kun kolmiulotteisia objekteja luodaan pelikäyttöön, on tärkeää optimoida hahmomallin polygonimäärä ja pintamateriaalin, eli tekstuurikuvien koko suhteessa kohdelaitteiston muistimäärään (Autodesk 2006, 190). Usein pelejä varten optimoituja 3D-malleja luodessa puhutaan niin sanotusta low poly -mallintamisesta. Low poly -mallintamisella tarkoitetaan 3D-mallintamista mahdollisimman vähällä polygoneilla muistiresurssien säästämiseksi (Derakhshani & Munn 2008, 240). Polygonimäärän lisäksi eniten laitteistoresursseja kuluttavat tekijät ovat pintamateriaalit, eli tekstuurit, ja animaatiot. Low poly mallintaminen on yleistä erityisesti kolmiulotteisiin videopelisiin tuotetuissa 3D-malleissa, pelihahmoista kaikkiin pelin sisäisiin esineisiin ja asioihin. Tavallisesti pelin päähahmon polygonimäärä on noin 4000, vaikkakin määrä voi vaihdella rajusti laitteistosta riippuen (Mitchell 2012, 127). Mobiilipeleissä keskimääräinen polygonimäärä hahmoa kohden voi olla muutamia satoja, kun taas nykyaikaisessa PC-pelissä määrä voi olla yli 50000. Low poly -mallinnuksen vastakohtaa, High poly -mallinnusta, jossa polygoneja saattaa olla miljoonia mallia kohden, tuotetaan useimmiten 3D-animaatioelokuvia ja yksittäisiä renderöityjä kuvia varten. Pelimme hahmojen polygonimäärä on keskimäärin alle 1500 polygonia, ja se sopisi pienellä optimoinnilla myös mobiililaitteille.

Oma pääasiallinen työalueeni oli mallintaa ja animoida pelimme hahmoista Guardian, Scout, Lich lord, Dragon ja Stallion. Aloitin mallinnustyön tuomalla Taija Lumitähden luomat hahmoluonnokset Autodesk 3ds Max -ohjelmaan. Asettelin kutakin hahmoa eri suunnista kuvaavat luonnokset litteille Plane-pinnoille ja asettelin ne eri kulmiin. Sain näin aikaan eräänlaisen virtuaalitudion ohjelman 3D-työtilaan. Nämä hahmoluonnokset toimivat referenssinä työlleni antaen ohjenuorat hahmojen mittasuhteille ja yleiselle muotokielelle. Käytin mallintamisessa pääosin apunani 3ds Max -ohjelman Edit Poly -työkalua. Tämä työkalu on ohjelman perustyökalu kolmiulotteisen kappaleen muokkaamiseen polygonitasolla. Mallintaessa käytin tekniikkaa, jossa loin ensin yhden litteän Plane-pinnan, jonka reunoille lisäsin pala kerrallaan lisää pintoja ikään kuin "vetämällä" reunasta esiin uuden pinnan (kuva 17). Kymmenistä ja sadoista yksittäisistä pinnoista muodostui näin yhdistämällä kokonainen hahmomalli (kuva 18). Usein mallintaminen aloitetaan käyttämällä hyväksi kuution muotoista valmiiksi suljettua peruskappaletta, jonka muotoa hiotaan pikkuhiljaa haluttuun suuntaan lisäämällä polygonipintoja ja liikuttamalla niitä. Itse koin henkilökohtaisesti tarkemmaksi ja mielekkäämmäksi lähtökohdaksi pelin hahmojen työstämiseen luoda yksittäinen yhdestä polygonipinnasta koostuva litteä Plane-objekti. Käytin lisäksi apunani symmetristen asioiden mallintamiseen Symmetry-työkalua, joka loi automaattisesti reaaliajassa mallin toisen puoliskon säästäten näin aikaa ja työtä.



Kuva 17. Dragon-hahmon mallinnuksen alku. Jokainen ruutu vastaa yhtä polygonipintaa



Kuva 18. Dragon-hahmon valmis 3D-malli ilman pintamateriaalia

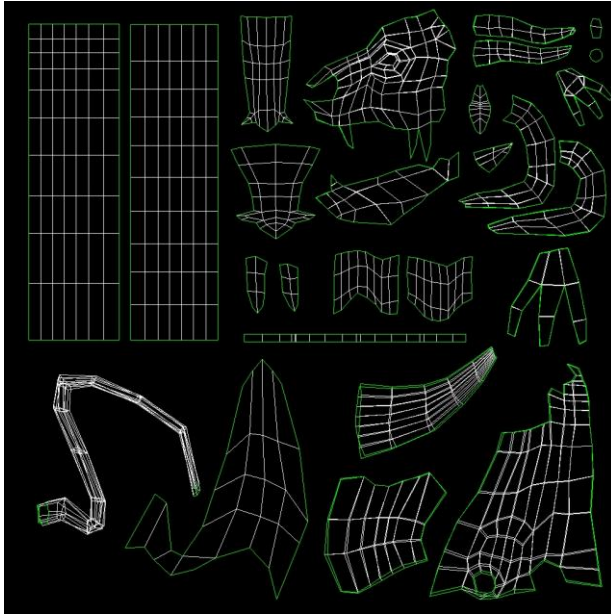
Kun hahmot oli mallinnettu, olivat ne valmiit teksturoitaviksi. Teksturoimisella tarkoitetaan 3D-mallien yhteydessä pintamateriaalien asettamista mallin pinnalle. Pintamateriaalit, eli tekstuurit, ovat tavallisesti kuvatiedostoja, joiden avulla malleihin saadaan luotua yksityiskohtia ja materiaalin vaikutelmaa, ja esimerkiksi häivytettyä teräviä kulmia tai rajoja (Manninen 2007, 231). Varsinkin low poly -mallintamisessa on tärkeää, että yksityiskohtia voidaan luoda tekstuurikarttojen avulla varsinaisen 3D-geometrian sijaan ja näin säästää jopa tuhansia polygoneja.

Tekstuurikartoista yleisin on perusmateriaali, eli diffuse-kartta, joka sisältää tavallisesti materiaalikuvan ilman valoja tai varjoja (kuva 19). 3D-malleihin voidaan lisätä myös muita tekstuurikarttoja, jotka luovat malliin tarvittaessa ominaisuuksia, kuten kiiltoa, heijastuksia tai läpinäkyvyyttä. Lisäksi peleissä käytetään usein niin sanottua normal mappia tai bump mappia, joilla voidaan luoda keinotekoisesti lisää pintaa ja monimutkaisia yksityiskohtia, kuten suomuja tai naarmuja hyödyntämällä valoja ja varjoja. (Novak 2008, 168.) Nämä yksityiskohdat eivät ole varsinaisesti osa mallin 3D-geometriaa, ja mappeja käyttämällä on mahdollista luoda erittäin yksityiskohtia malleja käyttäen vain murto-osan laitteistoresursseista.



Kuva 19. Taija Lumitähden maalaama Dragon-hahmon diffuse-tekstuurikartta

Jotta tekstuurikartat saataisiin aseteltua oikein kolmiulotteisen mallin pinnalle, on mallin pintojen koordinaatit kartoitettava eli mapattava (Gahan 2009, 30-35). Käytin hahmojen mappaukseen 3ds Max -ohjelman sisäänrakennettua Unwrap UVW -työkalua. Unwrap UVW -työkalun avulla ikään kuin avasin hahmojen mallien eri osat tasaisiksi pinnoiksi ja tallensin ne rautalankamallin kaltaisiksi kuvatiedostoksi, jonka päälle pintamateriaali luotaisiin (kuva 20). Tätä voisi yksinkertaisimmillaan verrata vaikkapa paperikuution auki taittelemiseen. Tämän jälkeen Taija Lumitähti maalasti hahmojen tekstuurit luomienne UVW-karttojen päälle, jotka saatoin sitten liittää mallintamiini hahmoihin.



Kuva 20. Dragon-hahmon UVW-kartta

4.2 Pelihahmojen animaatiot

Jotta valmiit 3D-hahmoihin saataisiin heräämään eloon pelissä, oli niihin luotava animaatiot. Suuriin peliprojekteihin verrattuna, joissa yhdellä hahmolla saattaa olla useita kymmeniä eri animaatioita juoksusta ja kävelystä aina jousen virittämiseen tai lukon tiirikointiin, oli projektimme suhteellisen yksinkertainen. Hahmot liikkuvat pelilaudalla liukumalla pelinappuloiden liikuttelun tapaan, eivätkä minkäänlaiset kävelyanimaatiot olleet tarpeellisia. Hahmot heräävät kuitenkin eloon hyökätessään vihollisen kimppuun, joten loin omalle osaani hahmoista lyöntianimaatiot.

Käytin myös animaatioiden ja luiden luomiseen Autodesk 3ds Max -ohjelmaa. Animaatioiden tekoa varten hahmoihin oli luotava luuranko jonka avulla hahmon liikkeitä kontrolloidaan. Luut itsessään ovat tavallista 3D-geometriaa ja usein vain yksinkertaisia laatikoita. Luut eivät itsessään sisällä mitään toiminnallisuuksia ennen kuin ne yhdistetään hahmomallin geometriaan ohjelman Skin-työkalulla (Bousquet 2005, 141). Luiden yhdistämistä hahmoon kutsutaan skinnaukseksi, ja tuloksena hahmon pinta käyttäytyy ihon tapaan liikkuen luiden mukana niitä liikuteltaessa. Skin-työkalu liittää luut hahmomalliin automaattisesti, ja tulos on harvoin sellaisenaan halutunlainen. Luiden painoarvoja niiden vaikutusalueita Skin-työkalun asetuksista muuttamalla saadaan säädettyä verteksin tarkkuudella, kuinka paljon jokainen hahmon luista vaikuttaa mihinkin kohtaa hahmosta. Tarkkuutta saadaan parannettua

asettamalla luut mahdollisimman tarkasti hahmon raajojen sisäpuolelle ja tekemällä luista mahdollisimman sopivan kokoiset hahmoon nähden. Luiden käyttö helpottaa huomattavasti orgaanisten hahmojen animointia rajoittaen hahmon raajojen liikkeit niiden luonnollisiin liikeratoihin. Luut asetetaan usein hahmomallin sisäpuolelle, eivätkä ne ole näkyvissä lopullisessa hahmomallissa. Poikkeuksena voivat olla mekaaniset hahmot, kuten robotit, joiden mekaaniset raajat itsessään toimivat hahmomallin varsinaisina luina. Luiden tekoa, asettelua ja muokkaamista hahmomalliin sopiviksi kutsutaan riggaamiseksi (Slick 2014).

3ds Max -ohjelmassa on sisäänrakennettuna erilaisia valmiita luujärjestelmiä, joista osaan hahmoja käytin valmista Character Animation Toolkit -järjestelmän (CAT) ihmisen perusluurankoa. Käytin CAT-perusluurankoa pohjana Guardian ja Scout-hahmoihin niiden ollessa muodoltaan tavallisia ihmisiä. Perusluurankoa oli helppo muokata luiden kokoa skaalaamalla hahmoihin sopivaksi. Dragon, Lich Lord ja Stallion-hahmot olivat muodoltaan huomattavasti monimutkaisempia ja uniikimpia, ja näiden hahmojen kohdalla riggaaminen vaati hieman enemmän työtä. Stallion-hahmon kohdalla sain luurangon muokattua valmiista hevosen luustosta. Käytin myös Dragon-hahmon pohjana valmista lohikäärmeen luurankoa, josta käyttökelpoisia osia olivat lähinnä eturaajat, siivet ja selkäranka. Dragon-hahmon kohdalla jouduin muokkaamaan luurankoa todella paljon. Valmiin luurangon pohjana käyttö säästi tästä huolimatta aikaa sen sisältämien valmiiden liikerajoittimien takia. Lich Lord -hahmon kohdalla tein luut alusta asti itse hahmoon sopiviksi.

Animaatioiden teko tapahtui key frame -animaatiotekniikalla, jossa useat peräkkäiset pysäytetyt kuvat, eli framet, muodostavat yhdessä kokonaisen animaation (Novak 2008, 168). Käytin animaatioita työstäessä eurooppalaista PAL-formaattia, jossa yhdessä sekunnissa on 25 kuvaa. Animointi tapahtuu liikuttamalla hahmoa haluttuun asentoon luiden avulla, jonka jälkeen asento tallennetaan animaation aikalinjalle yksittäisen kuvakehykseen eli key frameen. Tätä toistetaan, kunnes peräkkäisistä kuvakehyksistä muodostuu kokonainen animaatio, joka tallennetaan animaatiotiedostoksi. Jokaista framea ei kuitenkaan ollut tarpeellista erikseen animoida ja tallentaa vaan ohjelma täytti automaattisesti tyhjät framet yhtenäiseksi liikkeeksi. Animaatiosta tulee kuitenkin sitä sulavampi, mitä enemmän key frameja tallennetaan manuaalisesti.

4.3 Pelimoottori

Pelimoottorilla tarkoitetaan pelien kehitystyökaluna käytettäviä ohjelmistoja, jotka sisältävät usein graafisessa työympäristössä paljon valmiita aikaa säästäviä työkaluja pelin kokoamista varten. Pelimoottoreita käyttämällä ei pelinkehittäjien tarvitse tehdä kaikkea työtä alusta, kun eräänlaisen universaali runko ja valmiit komponentit ovat sisäänrakennettuna. (Ward 2008.)

Pelimoottoriin kootaan kaikki pelin objektit, joita kutsutaan asseteiksi. Assetteihin kuuluvat niin pelin hahmot, kentät, kuin myös lähdekoodin palaset eli scriptit. (Watkins 2011, 4). Pelimoottorissa kaikki pelin assetit kootaan yhteen ja niistä rakennetaan valmis peli liittämällä toiminnallisuudet pelin visuaalisiin elementteihin (GameObjects 2014). Pelissä esiintyviin esineisiin ja asioihin voidaan lisätä ominaisuuksia, kuten fysiikkamallinnus ja törmäyksen tunnistus, jotka yhdessä scriptien kanssa saavat pelin assetit toimimaan ja vuorovaikuttamaan keskenään halutulla tavalla (Creating and Using Scripts 2014). Pelimoottorin kautta peliin voidaan myös lisätä ja ohjelmoida pelin käyttöliittymäelementit ja määrittää pelin kamerat ja kontrollit. Esimerkiksi omassa projektissamme hahmoihin lisätyillä scripteillä ja törmäyksen tunnistuksella saatiin hahmojen liikkuminen rajattua siten, etteivät hahmot koskaan liikkuneet samaan ruutuun toisen hahmon kanssa. Näin kokoamalla ja yhdistämällä pelimoottorissa lukuisia asetteja saatiin aikaan toimiva peliprototyyppi. Käytimme pelimme kehittämisessä apuna ilmaista Unity-pelimoottoria. Valitsimme Unityn pääosin juuri sen maksuttomuuden takia. Lisäksi Unity oli työryhmälle entuudestaan tuttu, ja jokainen oli jollain asteella työskennellyt sen parissa.

Kun 3D-objekteja viedään pelimoottoriin, muuntuu jokainen neliönmuotoinen polygonipinta kahdeksi kolmioksi. Tämä käytännössä tuplaa alkuperäisen kokonaispolygonimäärän. Tämä johtuu siitä, että kolmiot ovat polygonipintojen yksinkertaisin muoto ja ne ovat helpoin tietokoneelle laskea pelikentällä ollessa tuhansia polygoneja laskettavaksi yhtä aikaisesti (Chopine 2011). Tämä oli otettava mallinnettaessa huomioon, ja oma työni pelimoottoria ajatellen oli optimoida mallintamieni hahmojen geometria siten, että se olisi mahdollisimman suosiollinen polygonien kolmioiksi muunnosta ajatellen.

4.4 Ohjelmointi ja jatkokehitys

Nemesis-pelin ohjelmakoodin tuottamisesta vastasivat Kymenlaakson ammattioppilaitoksen pelituotannon opiskelijat Jaakko Kauranen ja Ilmari Karulinna. He liittyivät työryhmäämme suhteellisen myöhäisessä vaiheessa projektia, kun suunniteltu yhteistyö Blight Games Oy -pelistudion kanssa keskeytyi. Kauranen ja Karulinna aloittivat ohjelmoinnin osana opintojaan työskennellen itsenäisesti projektin parissa. Pelin Game Jam -version koodi oli mielestämme sellaisenaan korkeintaan pohja. Se oli hyvin puutteellinen ja täynnä virheitä. Suurin haaste pelimme ohjelmoimisessa oli vihollisten tekoälyn luominen. Tähän kuului jokaisen vihollisen käyttäytymismallien ohjelmoiminen. Tekoälyn ohjelmoiminen osoittautui erittäin vaikeaksi ja lopulta vain Lich Lord -hahmon tekoäly saatiin ohjelmoitua osittain. Tässä vaiheessa koko projekti keskeytyi muiden opintojen vaatiessa koodintuottajiemme huomion.

5 LOPPUTULOKSET

Oppimista edistävänä peliprojektina koen Nemesiksen olleen onnistunut, ja onnistuin mielestäni täyttämään työtehtäväni hyvin. Ryhmähenki oli kaiken kaikkiaan hyvä ja visio pelistä ryhmän kesken yhtenäinen. Yhteistyö Blight Gamesin kanssa, vaikkakin loppujen lopuksi lyhytaikaista, antoi hieman kuvaa siitä millaista työnteko pelialalla voi olla. Suurimmaksi ongelmaksi projektissa osoittautui pelin koodin tuottaminen. Ohjelmakoodin tuottamisen aloitus viivästyi, eikä sen luomiseen tahtonut löytyä tekijöitä. Myös tietokoneen ohjaamien vihollisten tekoäly osoittautui erittäin haasteelliseksi toteuttaa.

Koulutusohjelmani keskittyessä juuri pelituotannon visuaalisiin osa-alueisiin oli mielestäni tärkeää luoda yhteistyösuhteita muihin oppilaitoksiin, joissa keskitytään enemmän ohjelmointiin. Näin eri osa-alueisiin keskittyneet opiskelijat voivat yhteistyötä tekemällä täydentää osaamistaan ja luoda yhdessä kokonaisia projekteja. Jälkikäteen ajateltuna myös projektin organisointi olisi voinut olla tehokkaampaa. Saimme kuitenkin paljon kokemusta projektin aikataulutuksesta ja projektinhallinnasta. Suoritin oman työalueeni projektista valmiiksi aikataulun mukaisesti, vaikka projekti itsessään jäi lopulta kesken ohjelmakoodin osalta.

Jatkoa ajatellen en pidä mahdottomana projektin eteenpäin kehittämistä, mikäli pelin ohjelmakoodi saadaan valmiiksi. Seuraavana askeleena voisi olla pelin käänös

mobiililaitteille ja mahdollinen kaupallinen julkaisuversio. Tällöin olisi myös mahdollista kehittää peliin erilaista lisäsisältöä uusien vihollisten, hahmojen ja pelikenttien muodossa. Aika näyttäneen Nemesis-projektin lopullisen kohtalon ja sen, saako sen kehitys jatkoa. Keskenräisenäkin koen projektin kuitenkin antaneen mahdollisuuden soveltaa kaikkea sitä kokemusta ja osaamista mitä koulutusallani oli tarjota.

6 LÄHTEET

Autodesk. 2006. Autodesk 3ds Max 9 essentials. Oxford: Focal Press.

Bates, B. 2004. Game Design. Thomson Course Technology.

Bethke, Erik. 2003. Game development and production. Texas: Wordware Publishing

Bousquet, M. 2005. Model, RIG, Animate with 3ds max 7. Thousand Oaks: New Riders Publishing.

Chopine, A. 2011. 3D Art Essentials, The fundamentals of 3D modeling and Animation. Oxford: Focal Press.

Creating and Using Scripts. Unity3D.com. Saatavissa:

<http://docs.unity3d.com/Manual/CreatingAndUsingScripts.html> [viitattu 29.9.2014]

Derakhshani, D. & Munn, R. L. 2008. Introducing 3ds Max. Indiana: Wiley.

Gahan, A. 2009. 3Ds Max Modelling for Games. Oxford: Focal Press.

GameObjects. 2010. Unity3D.com. Saatavissa:

<http://docs.unity3d.com/Manual/GameObjects.html> [viitattu 29.9.2014]

Global Game Jam. Saatavissa: <http://globalgamejam.org/about> [viitattu 19.6.2014]

Manninen, T. 2007. Pelisuunnittelijan Käsikirja, Ideasta Eteenpäin. Tallinna: Printing Partners Oü.

Mitchell, B. L. 2012. Game Design Essentials. Indiana: Wiley.

Novak, J. 2008. Game Development Essentials. Delmar Cengage Learning.

Slick, J. 2014. Rigging. 3d.about.com. Saatavissa: <http://3d.about.com/od/Glossary-R/g/Rigging.htm> [viitattu 1.10.2014]

Russell, S.J. & Norvig, P. 2003. Artificial Intelligence: A Modern Approach. New Jersey: Prentice Hall

Ryan, T. 1999. The Anatomy of a Design Document. Saatavissa:

http://www.gamasutra.com/view/feature/3384/the_anatomy_of_a_design_document_.php [viitattu 23.9.2014]

Ward, J. 2008. What is a Game Engine? Saatavissa:

http://www.gamecareerguide.com/features/529/what_is_a_game_.php [viitattu 23.9.2014]

Watkins, A. 2011. Creating Games with Unity and Maya. Burlington: Elsevier Inc.