

12.2.2020

Daniel Juola
daaajuol@student.jyu.fi
Pääaine: Tietotekniikka

TIEA306 Ohjelmointityö: Ultima-kloon

Ensiksi olisi parasta selittää, mitä tarkoitetaan ”Ultima-kloonilla”. Yleisesti termillä viitataan peleihin, jotka ovat samankaltaisia kuin Ultima-pelisarjaan kuuluvat pelit (<https://fi.wikipedia.org/wiki/Ultima>). Valtaosa näistä peleistä julkaistiin 1980- ja 1990-luvuilla, mutta vielä tänä päivänäkin voi löytää satunnaisen Ultima-kloonin vaikka App Store:sta. Mitä Ultima-pelit sitten olivat? Ne olivat urauurtavia tietokoneroolipelejä, joita oli yhteensä yhdeksän. Pelisarjaan kuului myös spinoff-pelejä jotka erosivat sarjan pääpeleistä, joista kuuluisin on Ultima Underworld, yksi ensimmäisistä 3D-roolipeleistä. Sarjan ensimmäinen peli, Ultima I, julkaistiin vuonna 1981, ja viimeinen peli, Ultima IX, julkaistiin vuonna 1999. Pelisarja kattaa siis valtaosan tietokonepelien varhaisesta historiasta, ja se näkyy itse peleissä, joiden ulkonäkö kehittyi teknologian mukana. Oma projektini on ”kloon” sarjan keskivaiheen peleistä, Ultima IV ja Ultima V, jotka vielä käyttivät yksinkertaista 2D-grafiikkaa ja olivat tiilipohjaisia.

”Ultima-kloon” ei tietenkään ole ohjelman lopullinen nimi, mutta edellä annetun selityksen jälkeen se antaa kuvan siitä, minkälainen projekti on kyseessä. Tavoite on moderneilla työkaluilla luoda uudelleen 80-luvun peli, jossa on ainakin valtaosa samoista ominaisuuksista. Sisällön määrä ei tietenkään tule olemaan samalla tasolla, koska aikaa on vain kuusi kuukautta ja tekijöitä vain yksi. Kyseessä ei myöskään ole täysi ”kopio” vaan kloon-sana voisi niin vihjata. Kaikki sisältö luodaan tyhjästä, eikä tekijänoikeuksia loukata käyttämällä samoja nimiä tms. Projektin tavoitteena on luoda peli, joka tuntuu samalta kuin Ultima-peli, olematta kuitenkaan täysin samanlainen.

Mitä ominaisuuksia Ultima-kloonilla sitten pitäisi olla? Ultima-pelien keskeisimpiä ominaisuuksia olivat avoin maailma, karma-moraalimittari joka ei ole vain hyvä/paha dikotomia, syvä dialogi pelihahmojen kanssa, taikasysteemi joka käyttää riimuja ja ainesosia perinteisen ”mana”-mittarin sijaan, elävän ja interaktiivisen pelimaailman simulaatio sekä kumppanit joiden kanssa pelaaja seikkailee. Olen jaotellut nämä sekä muut ominaisuudet eri vaiheisiin. Perusominaisuudet ovat alkuvaiheissa ja pyrin toteuttamaan ne ensin, vähemmän tärkeitä ja monimutkaisemmat ominaisuudet ovat myöhemmissä vaiheissa eivätkä välttämättä esiinny projektin lopullisessa versiossa. Tavoitteenani olisi ainakin toteuttaa kaikki vaiheissa 1-3. Lopullisessa dokumentissa ominaisuudet selitetään laajemmin.

- Vaihe 1: Perusasiat: Tiilipohjainen 2D-kartta, jossa on liikuteltava pelihahmo. ASCII-grafiikka. Karttojen luonti ja tallennus. Siirtyminen kartasta toiseen. Viestiloki.
- Vaihe 2: Interaktio: Yksinkertainen keskustelu ja taistelu muiden pelihahmojen kanssa. Kerättävät esineet. Inventaario.
- Vaihe 3: Ultima-pelien ominaisuudet: Taikasysteemi. Karma-mekaniikka. Kumppanit. Monimutkaisempi keskustelu. Tehtävät joita voi suorittaa.
- Vaihe 4: Työläät luksusominaisuudet: Kustomoidut pikseligrafiikat tiileille. Yö/päivä sykli, jota pelihahmot noudattavat esim. menemällä nukkumaan. Enemmän esineitä ja interaktiota pelimaailman kanssa. Päätarina, jonka voi suorittaa alusta loppuun. Ääniefektit?

Projekti toteutetaan C#-koodikielellä käyttäen Visual Studio 2019-ohjelmaa kehitysympäristönä. Lisäksi suunnittelin käyttäväni RogueSharp-

12.2.2020

(<https://github.com/FaronBracy/RogueSharp>) ja RLNET-kirjastoja (<https://bitbucket.org/clarktravism/rlnet/src/master/>). RogueSharp on pääasiassa tarkoitettu roguelike-genren pelien kehitykseen, mutta siinä on monia ominaisuuksia joita voin hyödyntää, kuten polunetsintää, satunnaislukuja ja näkökentän luomista. RLNET puolestaan luo pohjan tiilipohjaiselle pelille. Versionhallintaan käytän github:ia. Projektin rakenne tulee olemaan todennäköisesti yksinkertainen: Yksi kansio luokkatiedostoille ja toinen mahdollisille resurssitiedostoille. Tarkempi rakenne selviää kun työ aloitetaan.

Viimeisintään projektin pitäisi tietenkin olla valmiina puolessa vuodessa. Pyrin kuitenkin saamaan vaiheet 1-3 tehtyä noin kolmessa kuukaudessa. Ainakin 1-2 ovat toteutettavissa siihen mennessä, mutta vaiheeseen 3 saattaa mennä pitempään. Kun nämä perusominaisuudet ovat kasassa, niin laajennan niiden ympärille, toteuttaen vaiheen 4 ominaisuuksia ainakin osittain. Tämä ei ole ensimmäinen ohjelmointiprojektini, joten jonkinlaista kokemusta minulla kyllä on. Isompi ongelma aikataululle on se, että tulen käyttämään kirjastoja joiden kanssa en ole vielä kovinkaan tuttu. Internetistä kyllä löytyy tutoriaaleja perusasioille, mutta kun alan toteuttamaan monimutkaisempia ominaisuuksia niin kirjastoista on vähemmän hyötyä ja pitää itse selvittää miten ne voidaan tehdä. Laajemman kokemuksen puute tarkoittaa että en ole vielä varma, miten haastavia tiettyjen ominaisuuksien toteuttaminen tulee olemaan. On mahdollista, että vaiheet 3 ja 4 vielä muuttuvat.