

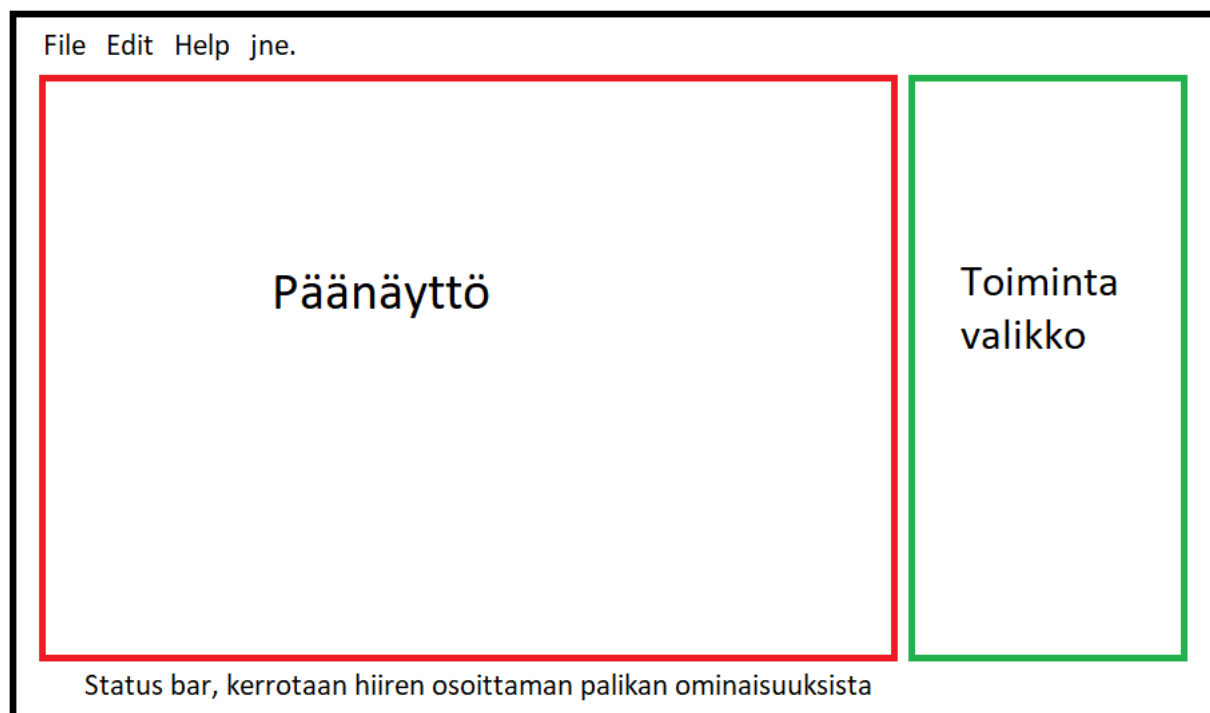
Yleiskuvaus ja vaikeustaso

Projektin aiheena on luoda vuoropohjainen strategiapeli, jossa vastustajana on älykäs tietokonevastustaja. Inspiraation lähteenä projektissa aion käyttää Civilization-pelisarjan pelejä ja sen ominaisuuksia. Tavoitteenani on luoda maailma, joka perustuu neliönmuotoisiin palasiin (tile) ja luoda maailma siten, että siihen on helppo lisätä eri mahdollisuuksia.

Aion luoda useampia erilaisia käytettäviä hahmoja (unit) ja lisätä näille eri ominaisuuksia, kuten esimerkiksi hyökätä, suojautua ja levätä. Aion aluksi luoda pelikartan itse, mutta mahdollisuuksien mukaan aion luoda kartat siten, että ne muodostuvat satunnaisesti. Tavoitteena on myös lisätä peliin karttaeditori, eli pelaaja voi itse luoda oman pelikartan.

Tavoitteenani on siis suorittaa projekti vaatimusasteella haastava ja pyrin pääsemään tähän tavoitteeseen ja uskon, että omaan hyvät mahdollisuudet siihen.

Käyttötapauskuvaus ja käyttöliittymän luonnos

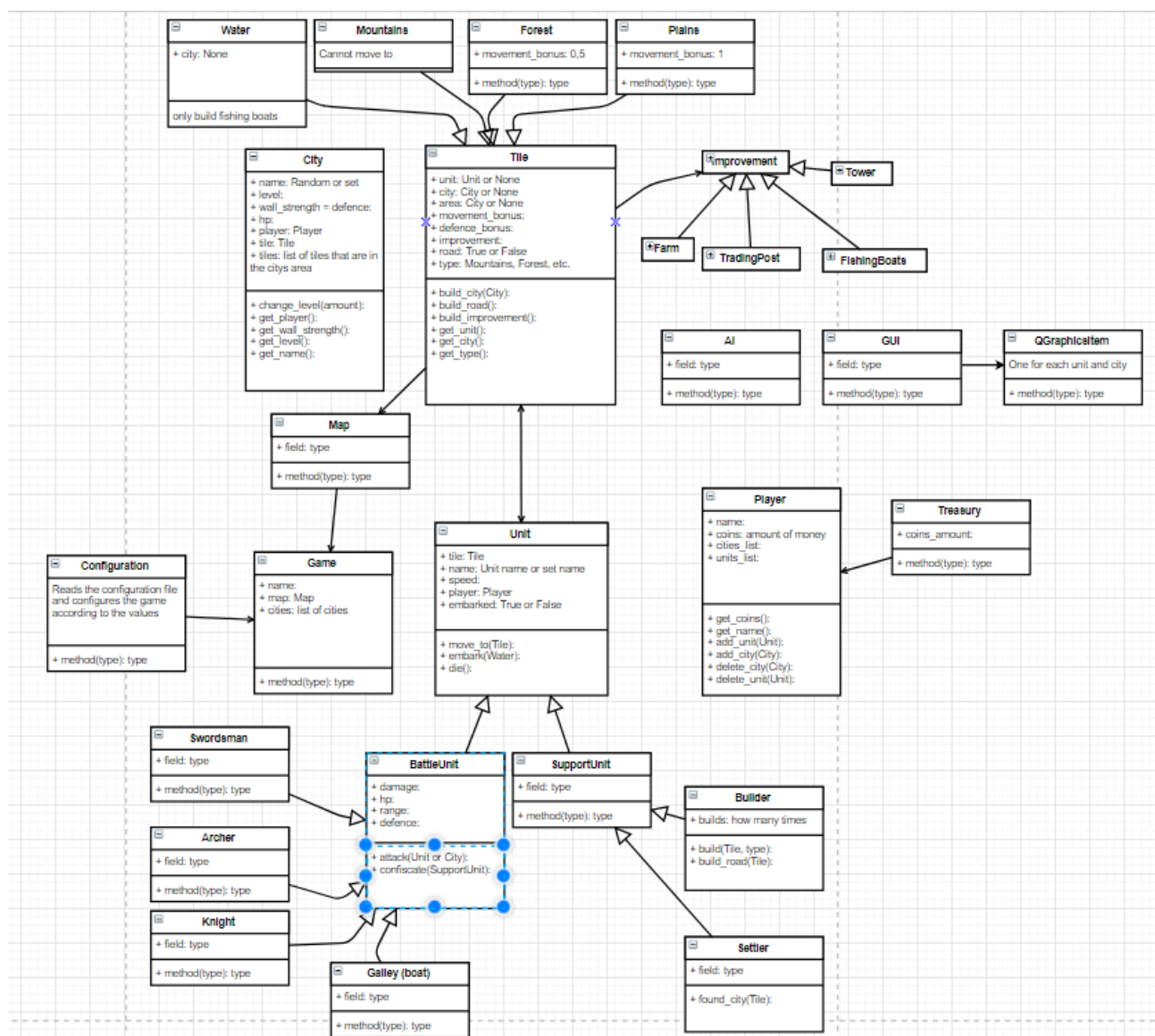


Ideana on luoda pääikkunalle pelimaailma, jossa hiirellä voi valita, mitä hahmoa haluaa liikuttaa tai tehdä jotakin. Tähän interaktioon aion luultavasti luoda myös pikanäppäimiä (esim. S on "sleep" eli nuku). Toimintavalikkoon aion listata klikatun hahmon ominaisuuksia sekä tämän käyttömahdollisuuksia esim. liikkua tai hyökätä.

Luultavasti aion luoda jonkin hyppyikkunan, joka kertoo pelaajalle esim. hyökkäyksen vaikutuksista sekä esim. mitä vastustaja onnistui tekemään omalla vuorollansa. Myös pääikkunan yläreunaan olisi tavoite luoda jokin infonäkymä, joka ilmaisisi mm. pelaajan rahamäärän.

Kun käyttäjä käynnistää ohjelman, ohjelma kysyy, mistä tiedostosta haluaa pelin jatkuvan. Jos mahdollista ja tarpeeksi simppeä lisätä ohjelmaan, pyrin, että pelin käynnistämässä on mahdollisuus ”jatka”-nappiin, eli ohjelma avaa automaattisesti aikaisemman tallennustiedoston. Käynnistämisen jälkeen pelaaja voi taas suorittaa oman vuoronsa mahdollisia toimintavaihtoehtoja, jotka tulevat graafisen käyttöliittymän toimintavalikkoon.

Ohjelman rakennesuunnitelma



(Aion tallentaa UML-mallin myös Gittiin.)

Aion luoda Game-olion pelin luomiseksi ja Map-olion kartan luomiseksi ja kartta koostuu Tile-olion alaluokista. Hahmot taas ovat Unit-olion alaluokissa. Pyrin ohjelmoimaan nämä luokat siten, että voin itse vapaasti luoda uusia olioita ja nämä olisivat helppo lisätä peliin. UML-mallissa on vähintään ne hahmot ja maastotyytit, jotka aion luoda, mutta mahdollisuuksien mukaan pyrin tekemään niitä lisää. Erikseen ovat vielä luokka City, joka muodostaa pelin kaupungit. Siitä voi tehdä ehkä Tile-olion alaluokka. Sitten on vielä GPU ja AI. GPU:hun tulee vielä useita alaluokkia ja luultavasti teen omat QGraphicsItemit jokaiselle Tile- ja Unit-oliolle.

Game-olion tehtävänä on siis luoda ja ohjata peliä. Pelin ohjaamiseen käytettävät parametrit tulevat Configuration-oliolta, joka lukee konfiguraatiotiedostot ja palauttaa nämä arvot pelin käyttöön. Game vastaa pelin vuorosysteemin toiminnasta ja ns. kaikki yleinen pelin hallinta kirjoitetaan tähän luokkaan. Luokkaan tulee metodeiksi mm. pelaajan tekemien toimintojen ohjaaminen ja näiden kutsuminen muiden luokkien metodeista.

Unit-olion tehtävä on olla yläluokka kaikille pelin hahmoille. Se antaa yleiset raamit pelin hahmoille. Tärkeimmät metodin muodostetaan alaluokissa BattleUnit tai SupportUnit. SupportUnit on vain selvennykseksi tehty olio, jonka metodit määritellään alaluokissa Settler ja Builder. Builder rakentaa ja Settler perustaa kaupunkeja. BattleUnitin metodit ovat taas hyökätä toista BattleUnittia kohtaan tai varastaa SupportUnit itselleen.

Tile-olion tehtävä on toimia yläluokkana kaikille mahdollisille maastoille, jota peliin aion lisätä. Sen tärkeimmät metodit muuttavat sen attribuutteja esim. rakentaa tie kyseiselle kentälle. Näiden metodien käyttöä voidaan rajoittaa alaluokissa, mutta metodit ovat yleisiä raameja, mitä Tileillä voidaan tehdä.

Map-olio on pelin kartta, joka koostuu lukuisista Tile-olioista. Kartta pitää ajan tasalla kaikki pelin hahmojen sijainnit ja pelitilanteen.

Tavoitteena on siis luoda hyvät pääluokat, joiden raameihin on mahdollista tehdä laajennuksia ilman, että peli kärsii.

Tietorakenteet sekä tiedostot ja tiedostoformaatit

Luultavasti tietorakenteina tulen käyttämään Pythonin omia tietorakenteita eli listoja ja mahd. sanakirjoja, koska en ainakaan vielä näe syytä, miksi pitäisi vaihtaa muuhun. Pelin tallentamiseen tiedostoon tulen luultavasti käyttämään Comma Separated Value-tiedostorakennetta sen ns. simppeliyden vuoksi. Aion luultavasti kuitenkin tallentaa simppeleitä rakenteita myös lohkoihin tämän tiedoston sisälle, jos se on vain kannattavaa. Tavoite on kuitenkin käyttää helpointa ja simppeleintä tallennusmetodia.

Algoritmit

Tietokonevastustajan luomiseen aion tutkia Utility AI:ta, jonka projektiassistenttini vinkkasi minulle ja en ole siihen ehtinyt perehtyä suuremmin. Yksi matemaattinen luomus, jonka joudun luomaan, on pelin vahinkosysteemi (eng. damage system) eli mitkä asiat vaikuttavat hahmon luomaan vahinkoon ja mitä kaikkea otan siinä huomioon. Tämän systeemin tulen luultavasti luomaan aika vahvasti valmiin mallin pohjalta, koska uskon, että jo nyt muissa peleissä oleva systeemi, jota on käytetty jo pitkään, on parempi kuin oma yrittämäni tässä.

Testaaminen

Heti aluksi testauksesta tulee mieleen se, että ohjelma tarkistaa, että konfigurointi- sekä tallennustiedosto ovat oikein. Testauksessa tulee ottaa huomioon ainakin myös pelin vuorosysteemi, että se estää toista tekemästä liikkeitä, kun on toisen vuoro.

Uskon, että osaan suunnitella metodini siten, että tiedän, mikä on oikeanlainen syöte ja mitä metodin pitäisi palauttaa, jolloin pystyn testata nämä metodit heti, kun olen metodit saanut valmiiksi.

Yksikkötestauksia pyrin toteuttamaan mahdollisimman säännöllisesti ja pyrin tekemään siitä rutiininomaisesti. Metodien luomisessa pyrin myös tekemään siitä testausystävällisen, jos vain mahdollista, joka säästää vaivaa myöhemmin.

Kirjastot ja muut työkalut

PyQt

Aikataulu

Aion tehdä aluksi ohjelman perusraamit kuntoon, eli hahmot ja palikat ja tämän jälkeen kartan ja pelin. Mietin, että voisin luoda tässä kohtaa GPU:n esiversion tässä kohtaa, jotta voin kunnolla nähdä omin silmin koodin toimivuutta. Tämän jälkeen luultavasti siirryn palikoiden liikkumiseen, jossa uskon, että voi tulla hankaluuksia. Myös tietokonevastustajan luomista alan tässä kohtaa aloittamaan, koska jos se on niin haastava kuin luulen, en halua jättää sitä viime tinkaen.

1. Tile ja Unit luokkien raamit – 5h
2. Game, Map ja Player luokat – 10h
3. GPU esiversio – 10h
4. Hahmojen liikuttaminen – 10h
 - a. Välipalautukseen 1 mennessä uskon, että pelin hahmot voivat liikkua
5. Päivityksien tekeminen Tileihin – 10h
6. Tietokonevastustaja – 10h
7. Pelitiedoston tallentaminen ja lataaminen – 10h
 - a. Tässä kohtaa pitäisi olla Välipalautus 2 aikana
8. GPU loppuun saakka – 5h
9. Dokumentaatio kuntoon – 3h

Aikataulutuksesta puuttuu testit, mutta testit kuuluvat nykyiseen aikataulutukseen, eli luon testit samalla, kun luon ohjelmien osat.

Kirjallisuusviitteet ja linkit

Tässä kohtaa en osaa vielä sanoa mitä kaikkea aion käyttää. Utility AI:n toimintaa tutkin tietokonevastustajaa tehdessä. Aion myös varmasti tutkia Civilization pelin vahinkosysteemiä, jotta voin implementoida sen omaani.