

## Ohjelman rakennesuunnitelma

Ohjelman voisi jaotella ainakin kahteen olennaiseen toiminnalliseen osa-alueeseen, valikoihin ja itse peliin. Tämän lisäksi ohjelmassa on pelitilanteen asetusten muokkaaminen, tallentaminen ja lataaminen, jotka ovat varsinaisen pelin tukitoimintoja. Kuvassa on pelin toiminnallinen logiikka. Kuvassa ei ole otettu kantaa asetusten muokkaamiseen, pelin tallentamiseen tai lataamiseen. Nämä ovat jokseenkin helposti mukaan toteutettavia asioita, jos peli vain toimii moitteetta.

Peli-luokka sisältää pelin kulun kannalta olennaiset asiat, se luo muut luokat ja tämän jälkeen kutsuu niiden metodeita, kuljettaessa peliä eteenpäin. Tämä luokka sisältää asetukset ja vastaa niiden implementoinnista peliin.

Kartta-luokka sisältää pelin kartan. Karttaa luodessa, luokka luo jokaiseen koordinaattipisteeseen oman ruudun, jolla on ominaisuutena mm. maastotyyppi. Kartta luodaan satunnaisesti pelin alussa ja peitetään pelaajilta.

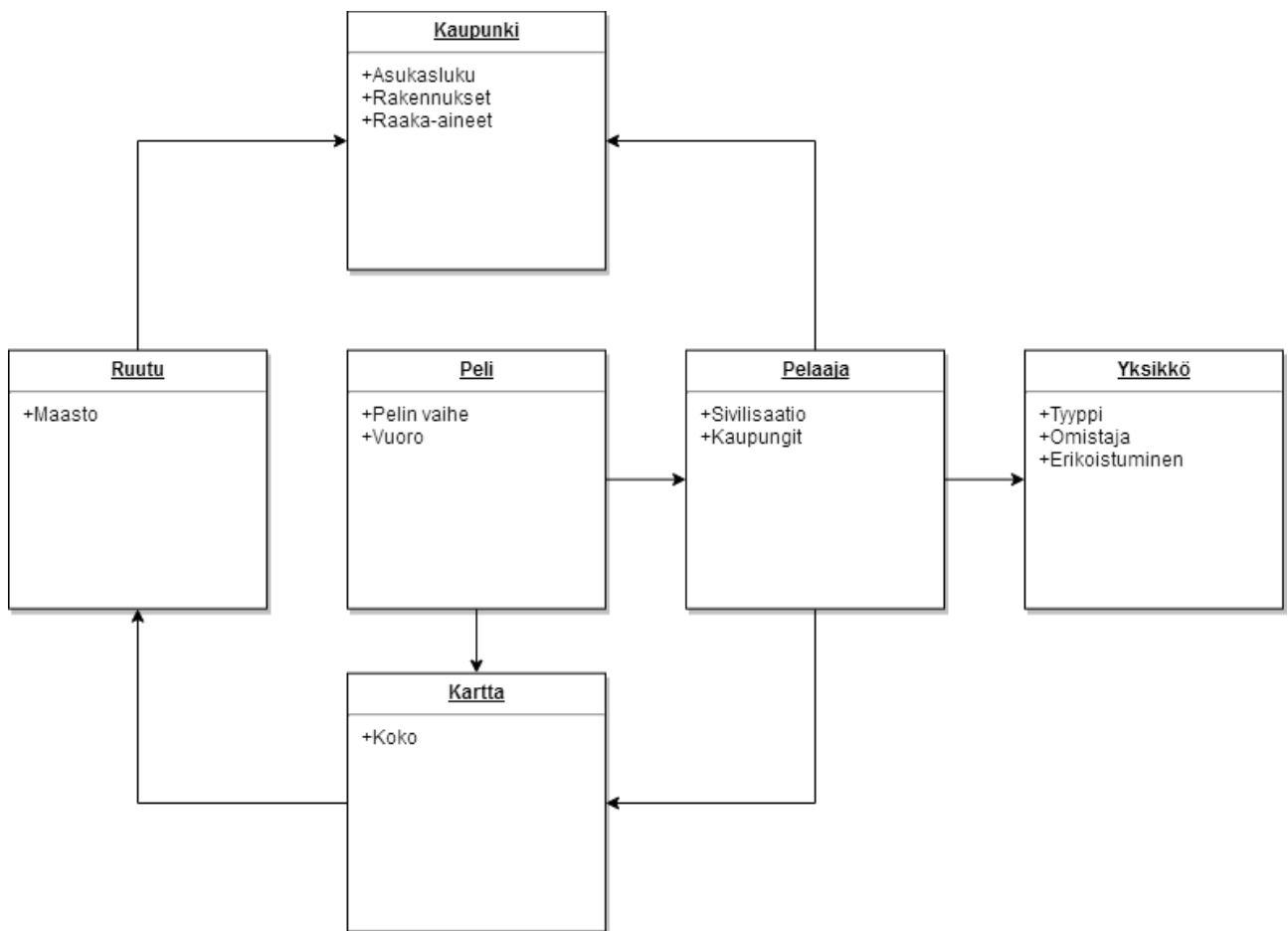
Ruutu-luokka kuvaa yhtä ruutua kartalla. Yhdessä ruudussa on aina oma maastotyyppi, se voi sisältää kaupungin ja/tai yksiköitä.

Pelaaja-luokka mallintaa peliin osallistuvaa pelaajaa. Sitä voi kontrolloida joko ihminen tai tekoäly. Pelaajalla on hallussaan sekä kaupunkeja, että yksiköitä. Pelaajat voivat kommunikoida keskenään löydettyään toisensa.

Kaupunki-luokka mallintaa pelissä kaupunkien toimintaa. Jokainen kaupunki kuuluu kerrallaan vain yhdelle pelaajalle. Yhdellä pelaajalla voi olla useita kaupunkeja. Kaupungilla on erinäisiä arvoja, kuten asukasluku, materiaalit sekä tuotantokapasiteetti. Lisää ominaisuuksia voidaan harkita pelin monipuolistamiseksi. Kaupunki voi myös majoittaa ja tuottaa yksiköitä omistamalleen pelaajalle.

Yksikkö-luokka kuvaa pelaajan komennossa olevia yksiköitä, jotka voi jakaa sotilaisiin, laivoihin ja siviileihin. Sotilaat voivat valloittaa ja tuhota muiden pelaajien omaisuutta, siviilien rakentaessa lisää kaupunkeja ja käydessä kauppaa. Kaikki näistä yksiköistä piirretään kartalle, ja niitä voi liikuttaa vapaasti. Laivat eivät voi kulkea maalla eivätkä sotilaat merellä, ilman että samassa ruudussa on laiva.

Pelin logiikan lisäksi, kuten aiemmin mainittiin toteutetaan peliin valikot. Valikoista voi valita pelaajien määrän, sivilisaatiot, kartan koon, kartan tyyppin ja muita asetuksia, sekä yhtenä tärkeimmistä, tallentaa ja ladata vanhan pelin. Asetuksista tehdään configure-tiedosto, jonka peli lukee luodessaan maailmaa. Tästä tiedostosta käy ilmi peliin alussa valitut asetukset.



## Käyttötapauskuvaus

Pelaaja käynnistää pelin. Peliin aukeaa päävalikko, jossa pelaajalle esitetään graafisesti eri vaihtoehdot. Jos pelaaja haluaa aloittaa pelaamisen, hän klikkaa "Aloita uusi peli" -vaihtoehtoa. Tämän jälkeen ohjelma avaa uuden configure-tiedoston, johon se on valmis tallentamaan pelaajan määrittämät asetukset. Pelaaja näkee eri vaihtoehtoja, joilla hän voi määrittää pelaajien lukumäärän, joista saa määrätä ihmispelaajat sekä tekoälyn kontrolloimat. Näille saa myös määrät sivilisaatiot tai jättää kentät tyhjäksi, jolloin ne arvotaan. Kaikki mikä voidaan, esitetään yksinkertaisina graafisina valikoina, osa tiedoista on tarkoituksenmukaisinta täyttää tyhjiin kenttiin. Kun asetukset ovat valmiit, configure-tiedosto on valmis, ja se tullaan tallentamaan myöhemmin, jos peli tallennetaan. Nyt "Peli"-funktio on valmis toimimaan. Se alustaa pelin käyttäjän laatimien määritelmien mukaisesti. Kartta luodaan satunnaisgeneraattorilla annettujen ohjeiden mukaan (saaristo, vähävetinen, metsäinen, aavikkoinen, kuinka iso, jne.), kartta luo tyhjä ruudut ja sitten täyttää ne maastotyypeillä. Tämän jälkeen kartta arpoo aloituspaikat pelaajille, joka ei voi olla merellä, eikä liian pienellä saarella. Tämän jälkeen peli luo pelaajat. Jokaiselle pelaajalle annetaan alussa yksi uudisraivaaja-yksikkö, joten pelaaja saa aloituspaikastaan huolimatta vaikuttaa kaupunkinsa sijaintiin. Jos pelaajalle ei valittu sivilisaatiota, peli arpoo sen, kuitenkin siten, että jokaisella pelaajalla on uniikki sivilisaatio. Tämän jälkeen peli on kaikilta osin alkanut. Pelaaja voi suorittaa siirtonsa.

## Algoritmit

|               |   |
|---------------|---|
| "Pathfinding" | Ajatuksena toteuttaa hyvin yksinkertaisesti siten, että yksikkö liikkuu molemmilla akseleilla yhtä aikaa (kartta on ruudukko, kuten vaikkapa shakissa) lähemmäs kohdekoordinaatteja. Yleensä saavutetaan toinen oikeista koordinaateista ennen varsinaista kohdetta, jolloin yksikkö siis liikkuu vain toista akselia pitkin, kunnes saavuttaa kohteensa. |
| RNG           | Tarkoituksena arpoa kartta arpomalla ensin ns. siementen koordinaatit. Tämän jälkeen maata aletaan luomaan näiden koordinaattien ympärille, aina sitä epätodennäköisemmin, mitä kauempana siemenkoordinaateista ollaan. Tarkoituksen kokeilla   |

erilaisia tekniikoita ja tallentaa esim. saaristopitoisin saaristo-vaihtoehdoksi.

|                 |  |
|-----------------|--|
| Tekoäly         | Tekee asioita riippuen saatavilla olevasta informaatiosta, alussa rakentaa kaupunkoja ja laajenee, jos tilaa riittää laajenee lisää, jos muut pelaajat aiheuttavat ahtautta, varustautuu sotaan, jos tilanne suotuista, hyökkää. Tekoälyn on tarkoitus olla varsin yksinkertainen, mutta kuitenkin järkevä.                    |
| Kartan tutkimus | Kartta on tarkoitus näyttää pelaajalle sitä mukaan kun pelaaja sitä tutkii, eli liikuttaa yksiköitään tuntemattomille alueille. Kartan paljastaminen tapahtuu laskemalle ympyrä yksikön ollessa sen keskipisteessä ja yksikön "tiedustelukyvyn" ollessa ympyrän säde. Helposti toteutettavissa käyttämällä pythagoraan kaavaa. |

## Tietorakenteet

Merkittävimpänä tietorakenteena pelissä on varmaankin pelinsisäiset listat, jotka pitävät sisällään tiedon pelaajan omistamista yksiköistä ja kaupungeista. Listoille on tarkoitus myös tallettaa tieto kartan sisällöstä. Listat ovat helppo valinta, sillä tieto on hyvin geneeristä, mutta sitä on kohtuullisia määriä. Listat olisivat dynaamisia, sillä yhden vuoron aikana pelaajan omistamat kaupungit tai yksiköt saattavat muuttua useilla kymmenillä instansseilla.

Toinen merkittävä tietorakennetyyppi on tiedostot. Tekstitiedostoihin on tarkoitus tallentaa kaikki tieto millä on pelissä minkäänlaista merkitystä, jotta peliä voisi jatkaa aivan kuin se ei olisi koskaan keskeytynytkään. Tähän tarkoitukseen sopii chunk-tyyppinen tiedosto, joka voisi olla ihmisen luettava muokkaamisen helpottamiseksi. Tässä tietorakenteessa ensinäkin tiedoston alkuun merkitään yleiset asiat pelistä kuten tallennuspäivämäärä yms. Tämän jälkeen tekstissä on kohtia, jotka kertovat milloin mikäkin osuus alkaa. Esim. "Pelaaja1 : Slaavit" tarkoittaa, että seuraava teksti käsittelee ensimmäisen pelaajan tietoja, ja että pelaajan pelaama sivilisaatio on slaavit. Seuraava osuus listaisi pelaajan varat, yksiköt ja kaupungit sijainteineen ja omine tietoineen.

## Aikataulu

### Viikko 11

Toimivat valikot ja kartta. Kartta luo ruudut, ja määrää jokaiselle ruudulle maaston.

### Viikko 12

Viimeistellyt valikot, kartta ja ruutu. Toimivat yksikkö- ja pelaaja-luokat.

### Viikko 13

Viimeistelyt pelaaja-luokka. Toimiva kaupunki-luokka.

### Viikko 14

Viimeistely kaupunki. Peli-luokka alkaa toimimaan kaikilta osin. Myös yksiköiden toiminnot, kuten taistelemisen, kaupankäynti ja kaupunkien perustamien mahdollisia.

### Viikko 15

Asetukset ja tallennus toimii, pelin voi tallentaa milloin vain, ja se lataa oikein. Tiedot säilyvät kaikilta osin.

### Viikko 16

Graafinen osuus valmis, peli näyttää miellyttävältä, yksiköiden toiminta on mallinnettu siten, että pelkästään katsomalla peliä pitäisi olla selvää mitä on tapahtunut.

### Viikko 17

Pelimekaniikan viimeistelyä. Viimeisten toimintojen kuten kaupunkien valloituksen ja kehittämisen viimeistelyä. Pelin pitäisi olla pelattava.

### Viiko 18

Viimeiset testaukset ennen palautusta. Peliä pelattu ainakin yksi kierros alusta loppuun.

## **Yksikkötestaussuunnitelma**

Testaus tulee olemaan suurelta osin manuaalista. Idean on, että tehdessäni uutta luokkaa, vierellä on main-metodi, jossa kyseinen luokka luodaan ja sen metodeja testataan. Kun luokka on valmis, sitä hyödynnetään myöhempien luokkien testauksessa. Esim. kartta-luokan ollessa valmis, sitä hyödynnetään asetusten ja ruutu-luokan testauksessa. Monimutkaisten luokkien, kuten yksikön ja kaupungin kanssa voisi tehdä automaattisia testejä, mikä varmaan helpottaa monitahoisten luokkien testausta, joiksi tässä projektissa voisi lukea luokat yksikkö ja kaupunki. Uskoisin muiden luokkien testaamisen olevan suhteellisen helpohkoa. Kuten vaikkapa peli-luokka, jonka testaaminen on paljolti muiden luokkien käyttämistä.

## **Kirjastot**

Ajattelin käyttää PyGame-kirjastoa. Saatan ottaa myös muita, esim PyQt:n käyttöön jos näyttää tulevan tarve. Tarkoitus on lähteä ohjelmoimaan alusta alkaen, ja jos tarvetta ilmenee, implementoida hyödyllisiä metodeita kirjastoista.