

# Loppudokumentaatio

Tämä on Markus Nuujan ja Jaakko Rainan loppudokumentaatio Ohjelmointi 3 Tekniikat kurssin harjoitustyöstä. Doxygendokumentaatio löytyy gitin hakemistosta Dox.

## Sisällys

Peliohjeet ja säännöt: .....	1
Luokkien vastuujako.....	1
Ryhmän työnjako .....	2
Lisäominaisuudet .....	3

## Peliohjeet ja säännöt:

Pelissä liikutaan ratikalla ympäri Tampereen keskusta-alueen karttaa. Tarkoituksena on kerätä pisteitä pelastamalla bussipysäkeillä tylsää ja vanhanaikaista Nysseä odottavat matkustajat ratikan kyytiin. Pelissä on minuutti aikaa saada ratikan kyytiin mahdollisimman monta ihmistä. Pelaaja ohjaa ratikkaa wasd-näppäimillä ja matkustajan saa kerättyä, kun raitiovaunu osuu matkustajaan.

Pelissä täytyy myös väistellä busseja ja hyvin uhanalaisia liito-oravia. Kumpaankin osuminen päättää pelin välittömästi. Bussit liikkuvat kartalla reaaliajassa niiden oikeita reittejään pitkin. Liito-oravat tipahtelevat kartan yläreunasta satunnaiseen tahtiin.

Peli näyttää reaaliajassa ratikan keräämien matkustajien määrän pistelaatikossa, sekä pitää kirjaa käyttäjän kaikkien aikojen korkeimmasta pistemäärästä, ja näyttää sen high score ruudussa käyttöliittymässä.

Pelin alussa pelaajalta kysytään alustusikkunassa mistä kolmelta pysäkiltä pelaaja haluaa pelinsä alkavan. Lisäksi saatavilla on kolme eri nopeusvaihtoehtoa ratikan liikkumiseen. Rules painikkeella ilmestyvät tässä esitellyt säännöt.

Pelin päätyttyä liike kartalla loppuu ja ruudulle ilmestyy ikkuna, jossa ilmoitetaan pelin päättymisen syy, sekä pistemäärä. Painamalla Retry painiketta ilmestyy jälleen asetusikkuna ja peli on mahdollista käynnistää uudelleen.

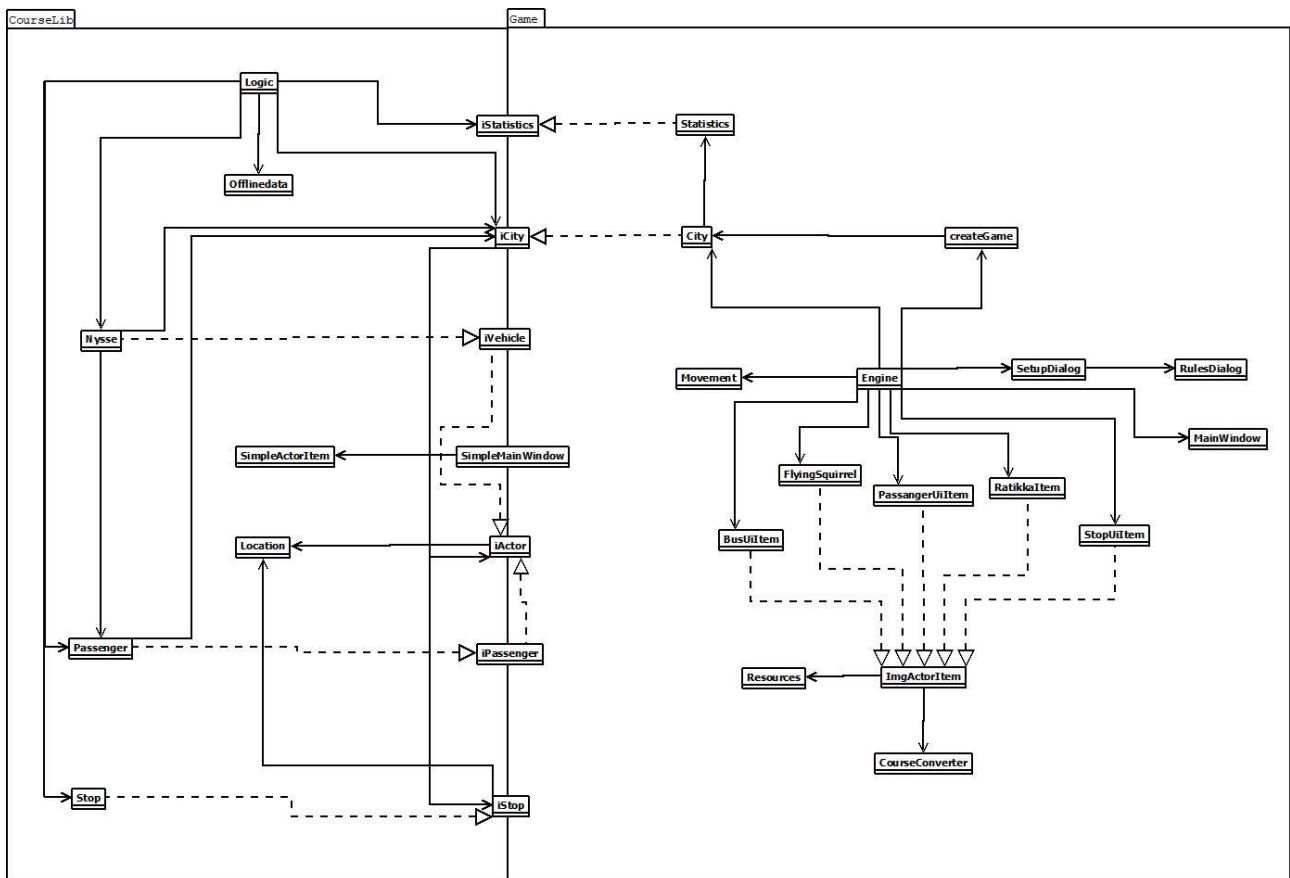
## Luokkien vastuujako ja toiminta

Projektin puolen vastuujako on seuraavanlainen. Mainissa luodaan Engineolio joka toimii pelimoottorina. Pelimoottori aukaisee asetusikkunan, jonka sulkeminen lopettaa koko pelin. Asetusikkunan välitettyä käyttäjän syöttämät tiedot pelimoottorin sisällä kutsutaan createGamea,

joka palauttaa Cityolion sharedpointterin. Tämän jälkeen moottori kutsuu logiikkaa, jossa luodaan bussit ja matkustajat. Lopuksi alustetaan vielä pääikkuna ja luodaan ratikka.

Moottorissa pyörii funktio updatePositions(), jota kutsutaan 60 kertaa sekunnissa ja joka huolehtii kartan alueella olevien aktorien liikuttamisesta kartalla. Moottori myös tarkkailee, onko peli päättynyt.

Cityluokan päävastuita ovat kartan alueella olevien aktorien päivittäminen vektoriin, sekä tilastoluokan kanssa kommunikointi tilastojen päivityksessä.



Kuva 1: Projektin luokkakaavio.

## Ryhmän työnjako

Projekti aloitettiin usean tunnin mittaisilla yhteisillä brainstormsessioilla. Pelin idean jalostuttua timanttiseksi suunnittelutyö aloitettiin. Suunnitelmien selkiydyttyä koodaustyö aloitettiin ensin yhteisillä sessioilla saman ruudun ääressä. Alkuun päästyä Jaakolla tuli kiireisempi koulujakso, jolloin projektista oli viikoittaisia palavereja, mutta Markuksella oli päävastuu koodaustyöstä. Kun bussit oltiin saatu liikkeelle ja oma mainwindow luotua oli työn jatkaminen helppoa eri haaroissa. Jaakon vastuulla oli asetusikkunan, raitiovaunun, ja liito-oravan toteuttaminen, kun Markus samaan aikaan kehitti pelimekaniikkoja, kuten törmäyksen tunnistamista, liikkeiden optimointia ja

sulavuutta. Lopussa Markuksella oli päävastuu yksikkötestien tekemisestä ja Jaakolla loppudokumentaatiosta.

Työnjako oli selkeää ja odottamattomia konflikteja ei tapahtunut. Molempien koodatessa samaan aikaan saatiin selkeästi sovittua mitä toinen tekee ja mitä toinen. Ominaisuuksien tekeminen omissa featurebrancheissa oli erittäin toimiva ratkaisu.

Markuksen koodiosaaminen oli jo ennen kurssia parempi kuin Jaakolla, joten Markus hoiti hieman haastavampien osuuksien varsinaisen koodaustyön, kuten suurilta osin pelimoottorin toteutuksen. Näin mahdollistettiin paras mahdollinen toteutus käytettävissä olevalla ajalla.

Työ aloitettiin nopeasti sen julkaisun jälkeen, mikä edesauttoi projektin valmistumista etuajassa suuresti. Muutamia hiljaisempia hetkiä tuli, kun muut kurssit veivät enemmän aikaa tai kurssin puolen koodiin odotettiin päivityksiä.

## Lisäominaisuudet

1. Tasainen ruudunpäivitys. Pelialueelle päivitetään tasaisin väliajoin kertyneet muutokset, sen sijaan että jokainen muutos aiheuttaisi erillisen päivityksen. `Engine::updatePositions()` päivittää pelin aktoreiden paikkoja 60 kertaa sekunnissa.
2. Minimaalinen ruudunpäivitys. Toimijan tilan muutoksen seurauksena vain sen välitön ympäristö päivitetään pelialueelle, sen sijaan että koko kartta piirrettäisiin uudelleen. Keräämme nearby vektoriin tiedot kaikista lähellä olevista aktoreista, joten kartalle piirretään ja päivitetään vain karttaikkunassa näkyvät aktorit niiden liikkeessä.
3. Vieritettävä kartta. Pelialue ei ole sidottu vain Tampereen keskusta, vaan esitettävä alue vaihtuu pelihahmon liikkeen perusteella.
4. Pelihahmon liike on lukittu keskelle karttaa, jos kartta ei ole ulkoreunastaan kiinni ikkunassa. Kartan scrollaus on estetty.
5. Pelihahmon tasainen liike. Välittömän paikanvaihdon sijaan pelihahmo käskytetään liikkumaan johonkin suuntaan, minkä jälkeen se etenee sinne sopivalla nopeudella. Pelihahmo etenee sulavasti nuolinäppäimen pohjaan painettuaan tasaisella nopeudella haluttuun suuntaan. Pelihahmo voi edetä kaikkiin pää- ja väli-ilmansuuntiin.
6. Kartalla pysäkeillä olevat ihmiset on hajautettu yhden pisteen sijasta pienelle alueelle niin, että on mahdollista nähdä suuntaa antavasti paljonko ihmisiä pysäkillä on.
7. Pelin tilan seuranta. Pelin aikana kerättäviä tilastotietoja esitetään reaaliajassa pääikkunassa. Ikkunan oikeassa alareunassa ratikan kyytiin poimittujen ihmisten määrä näkyy reaaliajassa.

8. Pelialue on lukittu karttaan, eli ratikka ei pääse pelialueen ulkopuolelle.
9. Peliin on toteutettu pelaajan kaikkien aikojen high score ominaisuus, joka tallentaa korkeimman pistemäärän.