

# OHJELMOINTI 3: PROJEKTIDOKUMENTAATIO

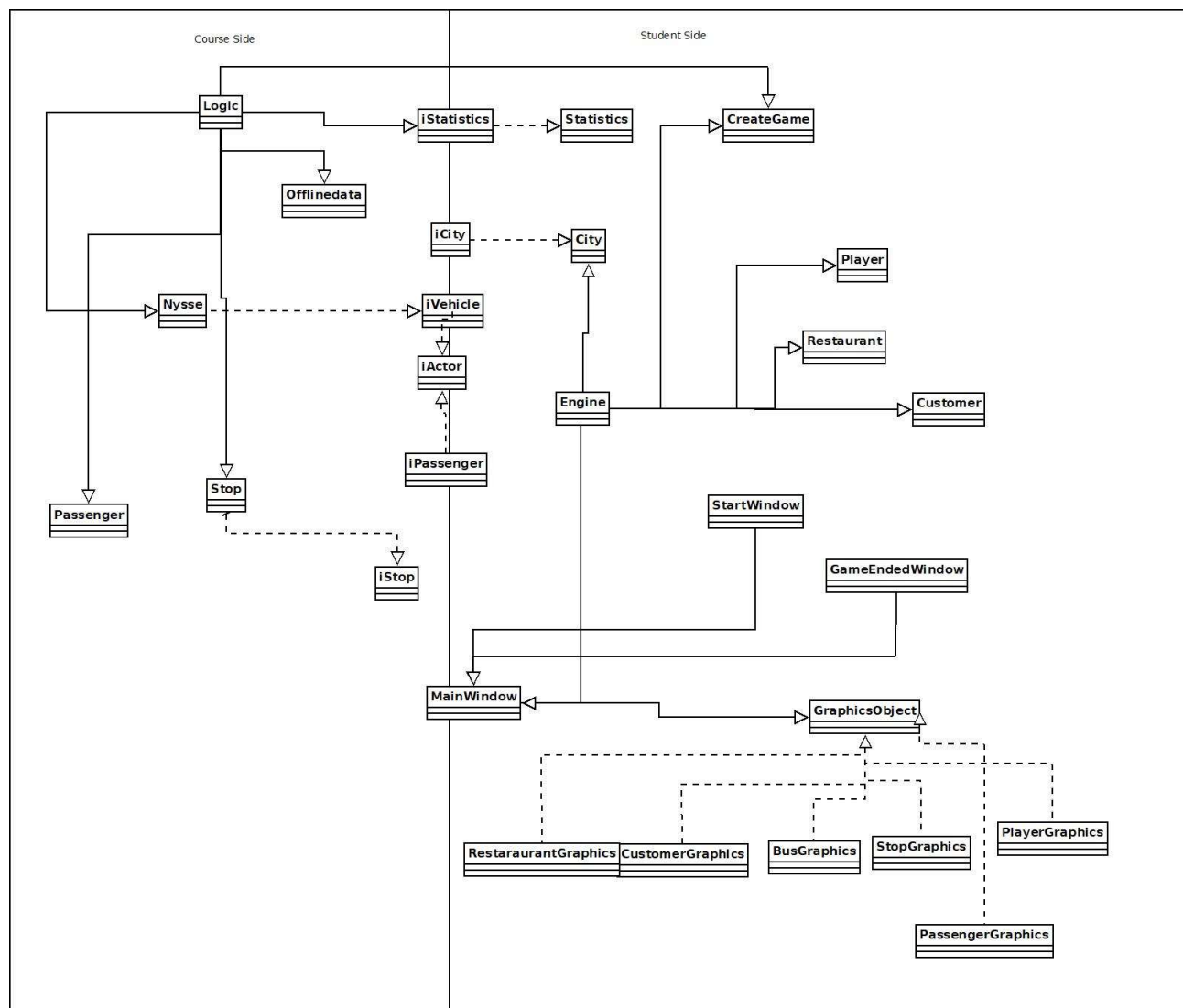
Jere Aho ja Titta Kemppi

## Sisällysluettelo

|                                  |   |
|----------------------------------|---|
| Ohjelman rakenne .....           | 3 |
| Luokkien vastuujako .....        | 4 |
| Ohjelman toiminta.....           | 4 |
| Dokumentoidut luokat .....       | 4 |
| Vapaaehtoiset ominaisuudet ..... | 5 |
| Työnjako .....                   | 6 |
| Peliohje .....                   | 6 |
| Ongelmat ja puutteet .....       | 6 |

## Ohjelman rakenne

Alla on pelimme luokkakaavio, joka pääpiirteissään kuvaa toimivia luokkia ja niiden vuorovaikutusta toistensa kanssa.



Kuva 1: Pelin luokkakaavio

Vasemmalla puolella ovat kurssin puolelta valmiiksi saamamme pohjan luokat ja oikealla luokat, jotka olemme itse lisänneet peliin. Keskellä ovat Interface-luokat, jotka virtuaalisia luokkia, joiden metodeille teimme toiminnallisuudet niistä periytetyille luokille, eli Citylle ja Statisticsille. Muita lisäämiämme luokkia ovat esimerkiksi oma MainWindow:mme, jolla korvasimme SimpleMainWindowin sekä GraphicsObject, jolla korvasimme kurssipuolen SimpleActorItemin. Graphicsobjectista periytyy oma grafiikkaluokka kaikille pelin aktoreille. Loimme myös CreateGame luokan ja Enginen, joka pyörittää peliä. Enginen suorassa yhteydessä

ovat omat actor-luokkamme Player, Restaurant sekä Customer. Lisäsimme peliin myös kaksi lisäikkunaa, alun dialogi-ikkunan StarWindow ja ikkunan GameEndedWindow, joka ilmoittaa pelin päättymisestä.

## Luokkien vastuujako

Pelin toiminta perustuu luokkien Engine ja MainWindow yhteistoiminnalle, eli nämä kaksi yhdessä koostavat ohjelman rungon. MainWindowin puolella hoituu kaikki näkyvä, eli siellä actorit (pelaaja, bussit, matkustajat, ravintolat ja asiakkaat) lisätään kartalle grafiikkoineen. Mainwindowissa reagoidaan myös käyttäjän antamiin inputteihin peliä pelatessaan. Engine taas pyörittää itse peliä: se asettaa pelin init-tilaan, käynnistää pelin, muuttaa actorien koordinaatteja, pysäyttää pelin, tarkastelee actorien vaikutusta toisiinsa ja tarkistaa, onko peli hävitty tai voitettu. Pelin grafiikat koostuvat kantaluokasta graphicsobject, josta on periytetty jokaiselle actorille oma grafiikaluokkansa.

## Ohjelman toiminta

Ohjelma lähtee käyntiin siten, että mainissa toteutetaan pointer engineen, josta pelin pyöritys käynnistyy. Enginessä luodaan kaikki mitä peli tarvitsee toimiakseen: logiikka, mainwindow, ajastin, pelaaja sekä statistiikkaolio. Rakennuksen jälkeen peli saatetaan init-tilaan, eli sille luodaan city-pointteri, jonka avulla saadaan käyttöön taustakuva, bussit sekä stopit ja saadaan ne asetettua kartalle. Myös ajastin pistetään toimintaan, että bussit pääsevät liikkumaan. Peliä päivitetään advance-slotin avulla timerilla määritellyin aikavälein, eli actoreita liikutellaan, poistellaan, lisäillään ja niiden tilaa päivitetään. Täällä tarkistetaan myös olioiden vuorovaikutus toistensa kanssa. Advancen avulla päivitetään myös käyttäjälle näkyviä pelitietoja, kuten pelaajan healthbaria tai pistetilannetta. Advancessa kutsutaan myös isGameOver-metodia, joka tarkistaa, onko peli hävitty tai voitettu.

Pelin ympäristö, city-olio, toimii ikään kuin ohjelman varastona. Sieltä löytyvät vectorit, joihin säilötään actorit ja joita päivitetään aina muutoksien tapahtuessa. Cityssä luetaan myös ravintoloiden tiedot sisältävä JSON-tiedosto ja lisätään ne omaksi actor-vectorikseen.

Grafiikat actoreille lisätään mainwindowin addActor-metodilla, joka erottelee actorit toisistaan tyyppinsä mukaan ja lisää ne mappiin, jossa avaimena on viite itse actoriin ja arvona actorin grafiikkaolio. Grafiikoiden päivitykseen on olemassa useita eri metodeja, joita kutsutaan enginen advancessa pelin tilan päivityksen yhteydessä.

Pelin loppumisen jälkeen peli voidaan myös käynnistää uudelleen. Kun pelin käynnistyy, kutsutaan luokkien omia reset-metodeja, jotka asettavat pelin siihen tilaan, jossa sen tulee olla pelin alussa.

## Dokumentoidut luokat

Kaikki Game-kansiosta löytyvät luokat on dokumentoitu.

## Vapaaehtoiset ominaisuudet

1. Tasainen ruudunpäivitys. Pelialueelle päivitetään tasaisin väliajoin kertyneet muutokset, sen sijaan että jokainen muutos aiheuttaisi erillisen päivityksen. Engine::Advance päivittää muutokset 30 kertaa sekunnissa. Mainwindowissa puolestaan kutsutaan QGraphicsScenen update-metodia samalla taajuudella.
2. Vieritettävä kartta. Kun käyttäjä liikuttaa pelaajaa, liikkuu kartta sen mukana. Koko kartta ei siis ole nähtävissä sillä hetkellä, kun peli käynnistetään.
3. Pelin uudelleenkäynnistys. Peli voidaan käynnistää uudelleen, kun se on joko hävitty tai voitettu. Pelin uudelleenkäynnistys tapahtuu painamalla uudestaan Start-nappia.
4. Ylimääräiset toimijat (asiakkaat ja ravintolat). Peliin on lisätty kaksi uutta actor-luokkaa. Asiakkaat ilmestyvät kartalle satunnaisesti. Ravintolat laitetaan kartalle valmiiksi määriteltyihin paikkoihin ja niihin kerääntyy ruokaa kahden sekunnin välein. Ravintoloilla on myös oma maksimiruokamääränsä, jonka se on voinut valmistaa kerralla. Kun pelaaja saapuu ravintolan luokse ja sillä on tilaa lämpölaatikossa, ravintolan ruokaa siirtyy ravintolasta pelaajalle. Ravintolan yllä oleva label kertoo saatavissa olevan ruuan määrän. Asiakas poistuu kartalta, kun pelaaja saapuu sen luokse ja antaa sille sen ruokamäärän, jonka se sattumanvaraisesti haluaa.
5. Piirretään vain kartan lähellä olevat actorit. MainWindow tarkistaa actoreita lisätessä, onko actor kartan läheisyydessä. Jos on, se piirretään, jos ei ole, sitä ei piirretä. Moveactor huolehtii pelialueelle liikkuvien actorien grafiikoiden lisäämisestä, sekä pelialueelta ulos liikkuvien actorien poistamisesta.
6. Minimaalinen ruudunpäivitys: QGraphicsViewin ViewportUpdateMode on asetettu arvoon MinimalViewportUpdate, jolloin ruudusta päivitetään nimenomaan vain muuttuneet asiat, eikä välttämättä koko ruutua.
7. Pelaajan sulava liikkuminen kahdeksaan suuntaan. Pelaajaa voi liikuttaa wasd-näppäinyhdistelmillä jokaiseen ilmansuuntaan, ilman, että pelaaja liikkuu töksähtelevästi (eli pelaaja ei pysähdy hetkellisesti ennen kuin sen liikesuunta muuttuu)
8. Pelin tilan seuranta. Peli-ikkunassa näkyy useita tilastotietoja, joiden avulla pelaaja saa tietoa pelin tilasta. MainWindowissa on pistelaskuri, pelaajan healthbar sekä laskuri, joka kertoo, kuinka paljon pelaajalla on ruokaa kyydissä. Näiden kolmen lisäksi kartalla ravintoloiden yllä on päivittyvä teksti, joka kertoo paljonko ruokaa niistä on saatavilla.
9. Oma data. Loimme oman JSON-tiedoston, josta löytyy ravintoloiden tietoja (id, maksimiruokamäärä, sijaintitiedot). Tämä data luetaan ja otetaan käyttöön city-luokan addRestaurants-metodissa. Omia resursseja ovat myös kuvat, jotka mapilla näkyvillä actoreilla on.

## Työnjako

Jeren vastuualueisiin kuului Enginen ja MainWindowin runkojen luominen. Actorien lisääminen kartalle ja niiden liikuttelun toimivuus ovat pitkälti Jeren ansiota. Titta loi peliin kaikki grafiikkaoliot, statistiikat, ravintolat (sekä niiden datan), pistelaskennan sekä startWindowin. Jere teki luokat pelaajalle ja customerille, sekä GameEndedWindow:n. City-luokan tekeminen kuului Titan vastuualueisiin, mutta molemmat täydensivät sitä sitä mukaa kun sen toiminnallisuuksia omassa tekemisessä vaadittiin. Pelin vaikeustason määrittelyn loi Titta ja UnitTestit kasasi kokoon Jere.

## Peliohje

Pelissä toimit auktemopokuskena, joka kuljettaa ruokaa nälkäisille asukkaille ympäri Tampereen ydinkeskustaa (sillä jostain syystä Hervanta on muka liian kaukana). Pelin dialogi-ikkunassa, joka aukeaa Start-napin painalluksen jälkeen, valitset vaikeustason, joka määrittää kuinka monta pistettä pelin voittamiseen tarvitaan (Easy = 100, Medium = 250 ja Hard = 500). Pelin alettua sinulla on 300 sekuntia aikaa kerätä tämä pistesaalis tai häviät pelin. Pisteitä kerätään käymällä ravintoloissa hakemassa ruokaa, ja ravintolan yläpuolella oleva label indikoi, kuinka paljon ruokaa kyseisessä yrityksessä on valmiina. Jos ruokaa on valmiina, täyttyy pelaajan lämmityslaatikko (Trunk) ruualla 30 yksikköä/sekunti. Lämmityslaatikko on täynnä, jos siinä on 50 ruokaa. Pisteitä kerääntyy, kun nämä ruuat on toimitettu asiakkaalle ja asiakas häviää kartalta. Kannattaa siis huomioida, että pisteitä ei keräänny ennen kuin asiakas on saanut kaikki tilaamansa ruuat. Asiakkaat tilaavat myös eri määriä ruokaa, eli ovat satunnaisesti eri pistemäärien arvoisia. Auktemopokusia liikutetaan wasd-näppäimillä. Mopokuskin tulee myös kunnioittaa muita liikennevälineitä, eli Nyssen kanssa kolaroiminen aiheuttaa mopoon kolhuja, ja healthbarin tyhjentymisen jälkeen ei mopo enää ole käyttökelpoinen ja häviät pelin.

## Ongelmat ja puutteet

Pelimme on pelattavissa ja siitä löytyy useita lisäominaisuuksia, mutta onhan se selvästi vielä kehitysvaiheessa. Tällä hetkellä pelaaja voi poistua pelialueelta, mikä pitäisi pystyä pelaamaan. Peliä voisi myös kehittää siten, että sen pelaaminen uudestaan ja uudestaan kiehtoisi. Tällä hetkellä, kun olet läpäissyt pelin vaikean tason, ei pelissä ole juuri muuta tekemistä. Esimerkiksi top 10-listan lisääminen voisi lisätä pelin kiehtovuutta. StartWindowssa kysyttävä pelaajan nimi oli kysytty tähän tarkoitukseen, mutta tällä hetkellä sillä ei ole mitään funktiota. Myöskään customerit eivät osu ihan koko kartan alueelle, vaan vasen yläkulma ja oikea alakulma jäävät tyhjiksi, tämä paljastui rand-funktion erikoisuudeksi (tämä windowsilla, etätyöpöydällä toimii paremmin).