

# TIE-02402 Ohjelmointi 3: Tekniikat

## Nysse-peli dokumentaatio

Harri Lehtonen, Toivo Wuoti

### Sisällys

1. Pelin kulku ja tavoite .....	1
2. Ohjelman toiminta ja luokkien vastuujako.....	2
3. Työnjako .....	3
4. Vapaaehtoiset ominaisuudet .....	3
5. Tiedossa olevat puutteet.....	3

### 1. Pelin kulku ja tavoite

Peli aloitetaan syöttämällä ensimmäisen dialogi ikkunan kahteen spinboxiin arvot. Toinen arvoista on pelin aloitusaika, joka tarkoittaa aikaa päivästä, jolloin peliä halutaan pelata. Esimerkiksi kellonajan 14.00 syöttämien aloittaa pelin kellonajasta 14:00, jonka mukaan myös busseja on enemmän liikenteessä kuin esimerkiksi kello 22:00. Toinen spinbox määrittelee, kuinka pitkään pelaajaa haluaa pelata. Tässä spinboxissa luvut tarkoittavat pelin sisäisiä minutteja. Jos pelaaja ei syötä lainkaan aloitusaikaa (ts. aloitusaikana 0.00) pelin ajaksi otetaan tietokoneen tämänhetkinen aika. Toisessa spinboxissa syötetään pelin aikaraja minuiteissa. Tämänkin voi jättää täyttämättä (0.00), jolloin pelin aikaraja poistetaan kokonaan, ja pelaaja saa pelata niin kauan kuin haluaa.

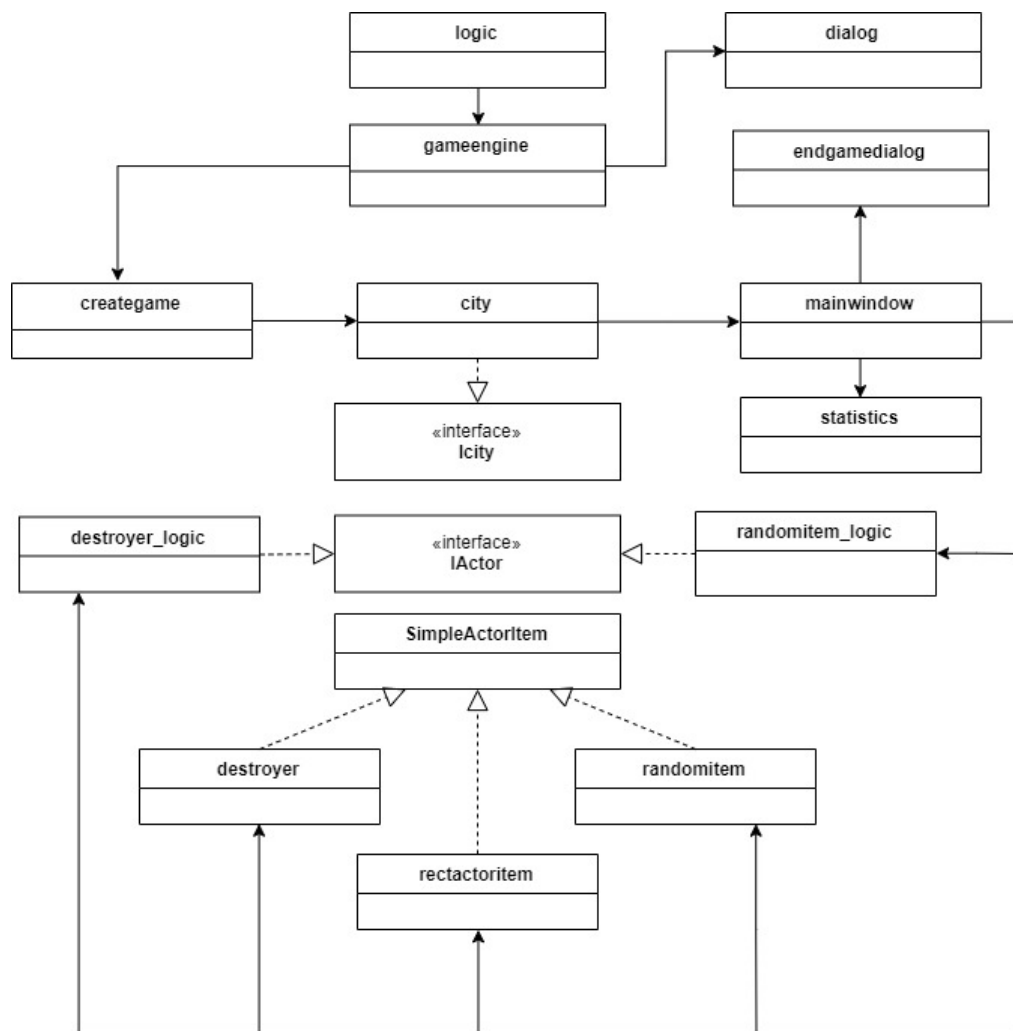
Näiden valintojen jälkeen start-nappulaa tai entteriä painamalla peli-ikkuna avautuu. Peli alkaa heti ikkunan avauduttua ja exit-nappulaa painettaessa peli luonnollisesti loppuu (lopetusdialogin kautta). Ruudulta löytyvät oikeasta laidasta pelaajan pistesaldo, bussien lukumäärä liikenteessä, matkustajien määrä peliruudulla, sekä alhaalla pelin sisäinen kello ja lopetusajankohta. Lopetusdialogi tulee esille myös pelaajan loppuessa. Tässä dialogissa kerrotaan pelaajalle pelin loppuneen, sekä hänen keräämänsä pisteet.

Kartalla valkoiset pallot ovat matkustajia, harmaat pallot bussipysäkkejä ja bussit sinisiä busseja. Pelaaja voi tässä kohtaa alkaa liikuttamaan huutis-hymiötä painelemalla wasd-näppäimiä. Tämä pelihahmo voi liikkua snake-pelin tapaan kartan toiselta laidalta toiselle puolelle.

Pelin tavoite on tappaa mahdollisimman paljon matkustajia ja siten kerätä pisteitä ennen kuin aika loppuu. Matkustajien tappaminen tapahtuu ajamalla huutis-emojilla matkustajan yli. Matkustajia voi kuitenkin tappaa vain, jos he ovat pysäkillä odottamassa bussia, busseissa he ovat turvassa. Jokaisesta taposta pelaaja saa yhden pisteen, joka päivittyy ikkunan oikeaan reunaan pisteosioon. Jokaisella matkustajan tapolla on mahdollisuus tuottaa kartalle banaaneita, joita hakemalla voi kerryttää satunnaisen määrän pisteitä 5:n ja 15:n väliltä. Matkustajien tappaminen lisää pisteitä yhdellä.

## 2. Ohjelman toiminta ja luokkien vastuujako

Kuvassa 1 on esitetty luokkien toteutus diagrammina. Ohjelman alussa main:stä kutsutaan gameengine-luokkaa, jonka vastuulla on pelin alustus. Gameengine-luokka kutsuu creategame-luokkaa, joka vain luo jaetun osoittimen city-luokkaan, sekä kutsuu dialogin, ja alustaa dialogin arvoilla kurssipuolen logiikka-luokan. City-luokka tekee kaikki muutokset tietorakenteissa ja kutsuu aina tarvittaessa mainwindow-luokkaa ja päivittää siihen tarvittavat muutokset. Mainwindow-luokka on QWainWindow-luokasta periytetty graafinen käyttöliittymä, jonka vastuulla on ylläpitää käyttöliittymää ja sen graafisia tapahtumia. Statistiikka-luokka puolestaan pitää kirjaa esimerkiksi kartalla näkyvien matkustajien määrästä. Tälle statistiikka-luokalle on myös kirjoitettu yksikkötestit.



Kuva 1: Luokka-diagrammi

Peliin on toteutettu kaksi omaa toimijaa: banaani (randomitem) ja destroyer. Näillä molemmilla on grafiikkakansiossa luokka, joka luo niiden grafiikan ja pitää kirjaa esim. niiden positioista. Näiden lisäksi niille on toteutettu myös omat logiikka-luokat, joihin on toteutettu esimerkiksi destroyerin liikuttamis-metodi. Grafiikka kansioista löytyy myös rectactoritem- luokka, joka on toteutettu vain bussien kuvanpiirtoa varten.

### 3. Työnjako

Emme sopineet alun perin mitään erikoisempaa työnjakoa, sillä annetun ohjelmakoodin tulkitsemisessä oli molemmilla haasteita. Alussa teimmekin pitkälti niin, että olimme molemmat paikan päällä kirjoittamassa ja debuggaamassa koodia samalla koneella, ja tästä syystä committeja onkin huomattavasti vähemmän harri\_haarassa kuin toivo\_haarassa. Lopulta kun saimme pohjakoodista tolkkua ja itse pelin pyörimään, jaoimme vastuualueet niin, että Toivo teki GUI-puolen ja Harri statistiikkaa, yksikkötestit sekä dokumentaation. Tästä jaosta huolimatta teimme myös toistemme hommia ja debuggasimme spagettia yhdessä.

### 4. Vapaaehtoiset ominaisuudet

Tällä hetkellä pelimme saavuttaa perustoteutuksen vaatimukset. Käyttöliittymämme on tehty itse, sekä pelihahmoa voi liikutella kartalla wasd-näppäimillä. Peliin on myös toteutettu yksi uniikki toimija lisää, joka on sattumalla ilmestyvä banaani, jonka hakemalla saa lisäpisteitä.

Lisäominaisuuksia pelissä on muutama. Minimaalinen ruudunpäivitys on toteutettu; ikkunaan päivitetään vain ne ominaisuudet, jotka ovat viime hetkeltä muuttuneet. Toinen ominaisuus on pelin tilan seuranta ikkunan oikeassa reunassa, jossa esitetään matkustajien määrä ja muuta statistiikkaa reaaliajassa. Osassa grafiikoista on myös käytetty png kuvia pallojen piirron sijaan. Lisäksi peliin on lisätty loppu-dialogi, joka näyttää pelin lopuksi pisteet ja maailmankuvaus löytyy erillisestä pdf-dokumentista.

### 5. Tiedossa olevat puutteet

Peliin ei ole toteutettu pelin uudelleen aloitusta, koska logiikka ei siihen tarjonnut hyvää toteutustapaa. Peli toimii vallan mainiosti molemmilla, Toivolla 5.12.4 Linuxilla ja Harrilla 4.12.4 Windowsilla. Etätyöpöydällä kuitenkin jostain syystä peli kaatuilee välillä, kun koittaa syödä banaaneita. Epäilemme tämän johtuvan etätyöpöydän latenssista.