

# TIE-02402 Ohjelmointi 3: Tekniikat

## Nysse-peli dokumentaatio

Valtteri Huhdankoski ja Niilo Santaharju

### 1. Johdanto

Peli aukeaa dialogi- ikkunaan. Ikkunassa on vaihtoehtoina Lore- nappi, jonka painaminen on hyvin suositeltavaa, ja toimivat quit ja play napit.

Play nappia painettaessa peli avaa suoraan mainWindowin, jossa on tavoitteena WASD- näppäimillä navigoiden ohjata Pohjois- Korealaista UFOa nappaamaan mahdollisimman nopeasti 500 tutkimuskohdetta tampereen kaduilta. TKL yrittää parhaansa mukaan suojella kansaa ajamalla heitä pois, mutta turhaan.

Pelikentällä pelaajahahmo on violetti-punainen ufo, bussipysäkit ovat punaisia STOP- merkkejä, ja matkustajia on merkitty pienillä pisteillä pysäkkien päällä. Liikkuvat Nysset ovat (Autekin)keltaisia koulubusseja, ja ne matkaavat kellonajasta riippuen aikataulunsa mukaan. Peliä ei suositella pelaamaan yöllä, sillä se on paljon pelottavampaa ilman busseja.

Jokaisesta napatusta ihmisestä pelaajan captured- laskuri kasvaa, ja sen yläpuolella näkyy kulunut aika. Tavoitteena on saada ihmiset kyytiin mahd. nopeasti, ja muuttua sitten näkymättömäksi ja lentää pois paikalta.

### 2. Ohjelman luokkarakenne, ja vastuuluokat

Asiaankuuluvat kuvat löytyvät tiedoston lopusta.

Ohjelmisto noudattaa Model View Controller- arkkitehtuuria (MVC), pelin alustaa main, initializer, ja createGame, mutta sitten Logiikka toimii pelimootorina, city toimii mainWindowin ohjaajana, ja mainWindow tuottaa visuaalisen ulosannin.

Poikkeuksena on, että kollisio tutkitaan mainWindowissa, ja pelin aloitus ja päätös on siirretty Cityyn, sillä niiden täytyi olla opiskelijoiden kontrolloitavissa.

Peli pyörittää käyntiin mainissa, jossa aloitetaan QApp instanssi. Tämän jälkeen main kutsuu CreateGame- funktiota, jonka tehtävänä on alustaa kaikki tarpeellinen pelin aloitusta varten. Se luo logiikkaolion, ja antaa sille shared\_pointerin ICity:stä periyettyyn instanssiin, jotta logiikalla on keino toteuttaa muuhun sovellukseen muutoksensa. City toimii primäärinä tietovarastona ja ohjaajana, se käynnistää pelin, luo peli- instanssille kellon, taustakartan, ja lisää kaikki actorit mainWindowiin.

mainWindow pitää kirjaa kaikista visuaalisesta, se tietää missä mikäkin QGraphics instanssi on kartalla, joille on puolestaan annettu Cityssä sijaitsevat vastin- oliot, joita Logic ohjaa.

mainWindow pääasiassa seurailee olioidensa muutostiloja, ja päivittää niitä mainWindowiin, mutta siinä on myös toteutettu Pelaajan UFO, sen liikkuminen, ja sen kollisiot pysäkkien kanssa, jolloin pysäkillä olevat matkustajat "kerätään kyytiin".

Luokkien vastuujako meni vähän ristiin rastiin, karkea jako oli kuitenkin front-, ja back-end luokat.

Rajapinnat käytiin etukäteen jokseenkin läpi, ja headereihin tehtiin declaraatiot, jotta toisen vastuulla olevaa luokkaa pystyi "käyttämään" jo ilman toteutusta.

Kaikki merge konfliktit vältettiin siten, että kommunikoimme aina, että mitä luokkaa muokkasimme, ja toinen ei IKINÄ koskenut samaan tiedostoon. Paitsi kun näin kävi, mutta lukuunottamatta 2 konfliktia strategia toimi loistavasti.

### 3. Työnjako

Työnjako suunniteltiin alustavasti kun peliä ideoitiin, heti kun saimme gitin toimimaan, submodulelet paikoilleen, ja meillä oli jotain hajua miten peli toimisi, niin jaoimme traditionaalisesti työt Fronttiin ja Backiin.

Valtteri otti koko backendin, johon karkeasti kuuluu city, mainin rakenne, createGame ja initialization.

Niilo otti frontin, johon kuuluu mainWindowin toteutus, dialogit, ja kaikkien actoritemien grafiikat.

Työn edetessä lisä työn jakoa tehtiin, ja työnjako tehtiin enemmän ongelmanratkaisupainotteisesti. Yksikkötestit ja monet muut loppu pään toteutukset tehtiin enemmän tai vähemmän yhteistyönä.

### 4. Extra ominaisuudet

Tavoittelimme arvosanaa 3-4. Pelissä on toteutettu oma mainWindow, joka päivitetään ainoastaan kun joku scenessä liikkuu, yksi lisäätori (UFO), jota voi liikuttaa WASD näppäimillä, ja lisäksi teimme kaikille actoreille omat grafiikat. Eräs ominaisuus on pelin tilan seuranta ikkunan oikeassa reunassa, jossa esitetään matkustajien määrä ja muuta статистиikkaa reaaliajassa.

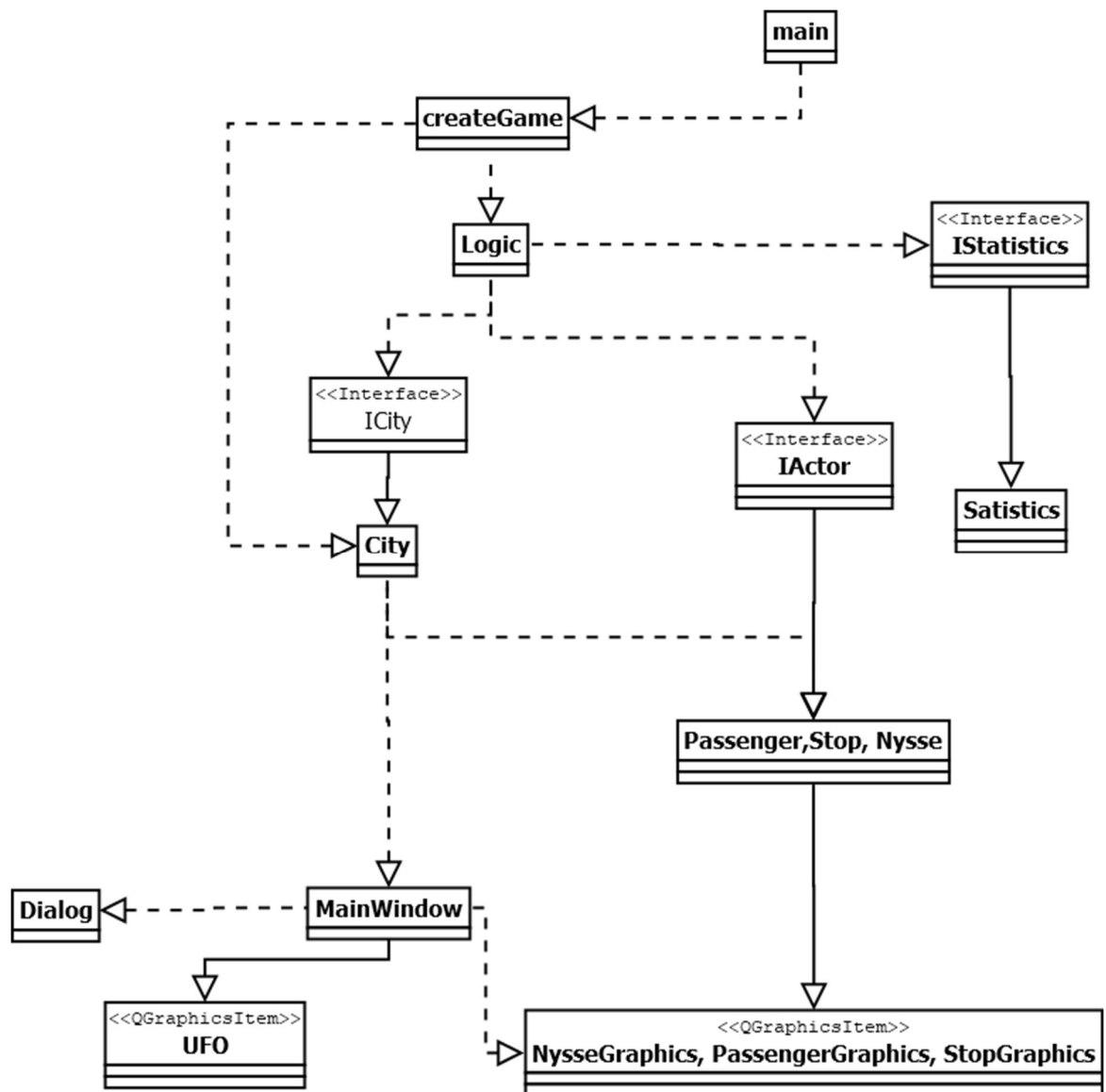
Pelissä on myös toteutettu yksikkötestaus Statistics luokalle.

Peli oli suunniteltu siten, että pelin tallennus, ja lähtöarvojen muokkaus olisi mahdollisimman yksinkertaista, mutta emme aikataulusyistä pystyneet toteuttamaan loppuja toimintoja.

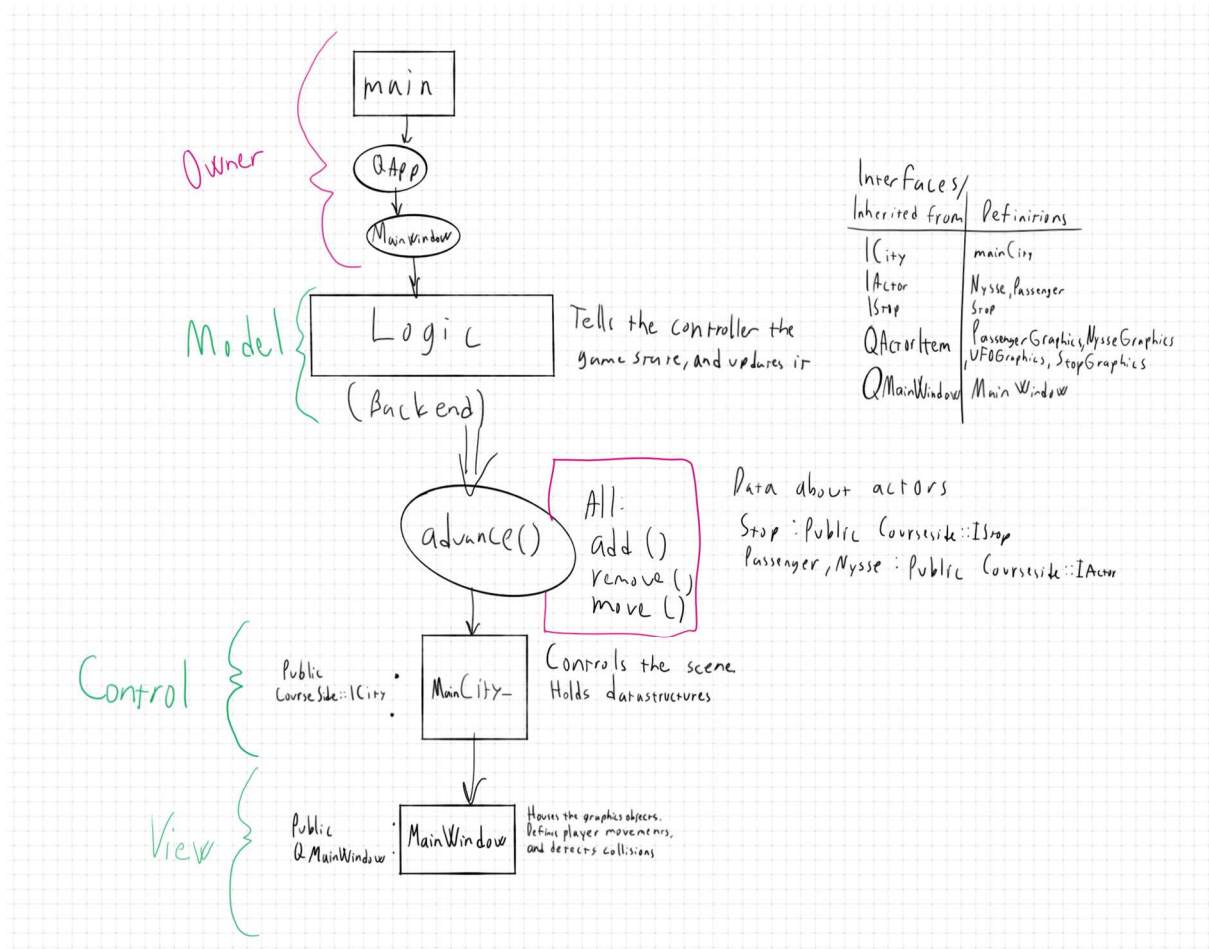
### 5. Tiedossa olevat puutteet

Lähes ei mitään. Peli pyörii nopeasti, eikä jää jumittamaan minnekkään. Ainoa löydetty ongelma on kartalla välkkyvät passengerobjectit jotka eivät oikeasti ole sielä. Bugia selvästi tapahtuu aina välillä, mutta määrä on vaikea tietää tarkasti. Kokeilimme muuttaa tietorakenteita, ja scenepäivitysjärjestelmiä useasti, mutta passengerit välkkyvät regardless. Uskomme bugin johtuvan useasta päällekkäin piirretystä passengerista, ja QT:n tasapainosta objektien määrän kasvaessa.

Koodin versiolokit löytyvät tiedostosta CHANGELOG



Kuva 1: Luokkakaavio.



Kuva 2: Toimintalogiikka