

Kuva 1 havainnollistaa ohjelmamme luokkien väliset kytkökset. Ohjelman tärkeimmät komponentit keskustelevat alussa siten, että ensin main.cpp:ssä luodaan osoitin uuteen engine-luokan olio on pelilogiikan pyörittämistä varten. Seuraavaksi mainissa kutsutaan mainWindowin rakentajaa. Kyseisen rakentajan sisällä luodaan ja näytetään sekä suljetaan dialogi. Dialogista saadaan talteen pelihahmon nimi.

Dialogin sulkeuduttua ohjelma palaa mainWindowin alustamisen jälkeen takaisin mainiin, jossa engine->takeUi() ottaa sen parametrinaan. Tämän jälkeen kutsutaan engine->startGame(), jonka sisällä ensimmäiseksi alustetaan uusi tilastoluokan olio. Tämä olio yhdessä aiemmin luodun mainWindowin kanssa syötetään parametrinä seuraavaksi uudelle citylle, jolloin city saa tarvitsemansa oliot omien metodiensa toteuttamiseen. Engine->takeCity() saa parametrinaan luomamme cityn. Edelleen startGame() sisällä luetaan kurssin puolelta tarjotuin menetelmin busseja ja pysäkkejä koskevat datat, jonka jälkeen peli voidaan käynnistää. Nyt on siis luotu engine, sille pääikkuna ja dialogi, jonka jälkeen on tilastoluokan ja pääikkunan avulla voitu luoda uusi city, joka nyt annetaan enginelle, ja pelin keskeinen logiikka on valmis pyörimään. Pääosissa siis city ja mainWindow, joille kuitenkin tarvii kutsua muita luokkia alustusvaiheessa.

YLEISTÄ

Pelin tarkoituksena on estää lepakkopandemian hallitsematon leviäminen. Pelin alussa 20 opiskelijaa saa tartunnan. Opiskelijat voivat edelleen jatkotartuttaa muita lähistöllään olevia. Pelaaja ohjaa dronea, jolla voi kulkea ympäri kaupunkia keräämässä resursseja. Kerättyään jokaista resurssia yhden kappaleen kykenee pelaaja parantamaan yhden opiskelijan. Jos keräät 10 jokaista resurssia voit päivittää. Päivitykset tuplaavat joko resurssien saamis- nopeuden tai kantaman ja lisäksi voit nostaa liikkumisnopeuttasi 5:llä päivittämällä. Peli päättyy voitokkaasti, mikäli pelaaja onnistuu parantamaan kaikki tartunnan saaneet. Peli päättyy tappioon, mikäli tartuntojen määrä kasvaa liian korkeaksi. Oletusarvoisesti häviökynnys on säädetty siten, että peli päättyy, kun 10% matkustajista on saanut tartunnan. Tartunnan levittäminen eteenpäin on säädetty siten, että lepakkotartunnan saanut jatkotartuttaa lähistöllään olevan ihmisen 5%:n todennäköisyydellä.

Loppupelissä, kun olet saanut paljon resursseja, niin voit päivittää pelaajan kantamaa tarpeeksi suureksi, että se kattaa koko pelialueen ja massaparantaa kaikki. Pelin loputtua, jos pääsit top 10 listalle, saat nimesi ja tuloksesi talteen, kunnes joku muu syrjäyttää sinut top 10 listalta. Pisteitä saa yhden per pelisekunti ja jokaisesta resurssista saat yhden pisteen ja vessapaperista tuplat. Parannetusta opiskelijasta saat 1000 pistettä.

Mikäli tahdot testauksen ajaksi säätää pelin kestoa, niin se onnistuu menemällä Game-city.h ja vaihtamalla siellä losingThresholdin arvoa, joka oletusarvoisesti on 10,0. Voit myös säätää tartuttamisen todennäköisyyttä menemällä Game->mainWindow::updatePandemic(), ja säätämällä siellä sisimmän if-lauseen parametria.

YKSIKKÖTESTIT

Toteutettu statistics-luokalle omana aliprojektinaan nimeltään "StatsTest".

LISÄOMINAISUUDET

- Vieritettävä kartta
 - Näytettävä pelialue siirtyy pelihahmon mukana. Karttaa voi myös rullata sekä pysty- että sivusuunnassa palkkien avulla.
- Pelihahmon tasainen liike
 - WASD-syötteellä pelaaja liikkuu tasaisin askelin. Askel on oletusarvoisesti 5 ja sitä voidaan päivittää, jokaisen päivityksen pidentäessä askelta 5 yksikköä.

- Pelin tilan seuranta
 - Lepakoiksi muuntuneiden lukumäärä näytetään peliruudulla
 - Resurssien määrä ja ominaisuuksien arvot ovat myös näkyvillä
 - Ruudulla myös pelissä kulunut aika sekunnin tarkkuudella
- Top10-lista
 - Kääntämisen jälkeen ensimmäinen peli luo build-kansioon "top_scores.txt", jossa pidetään kirjaa parhaista pisteistä. Mikäli kyseessä on ensimmäinen peli, niin pelin alussa peliin luetaan tekijöiden manuaalisesti laatima Game-kansiossa sijaitseva ":/top_scores.txt", joka pelin päättyä muokattuna build-kansioon, josta se luetaan seuraavilla pelikerroilla.
- Tasainen ruudunpäivitys
 - 7 millisekunnin välein kutsutaan ruudunpäivitys.
 - Tämä on (ainakin teoriassa) noin 143 ruudunpäivitystä sekunnissa.
- Minimaalinen ruudunpäivitys
 - Päivittää vain pelaajasta alle 500 pikselin päässä sijaitsevat kartalle piirrettävät elementit
- Pelihahmon päivitykset
 - Pelihahmo voi kerätä resursseja, joilla on mahdollista parantaa tartunnan saaneita.
- OMA LISÄOMINAISUUS: Musiikki
 - Pelin aikana voi halutessaan laittaa musiikin soimaan. Pelin päättyessä soitetaan myös musiikkia.
- OMA LISÄOMINAISUUS: fps-mittari
 - Olemme toteuttaneet fps-mittarin, joka mittaa kuinka monta ruudunpäivitystä peli tekee sekunnissa. Tämä lukema näytetään peliruudulla.

MUUTA

Reset-nappia ei vaadittu ohjelman perustoteutukseen. Sellainen koitettiin toteuttaa, mutta ei täysin bugittomaksi ikinä saatu. Toisinaan napin painaminen kaataa pelin. Kurssin puolen koodi ei myöskään tarjoa kovin näppärää tapaa pelin täydelliseen alkutilaan saattamiseksi ilman pelin uudelleenkäynnistämistä. Peli on myös hyvin raskas cpu:lle ja ramille. Käännösaika saattaa olla kohtalaisen pitkä, odotathan kärsivällisesti.

Työnjako

- Vesa
 - Actor, actoritem, gameengine, resources, stop, grafiikat
 - Suurin osa gamewindow, myöskin dokumentaatiossa mukana
- Mikael
 - Dialog, Passenger, Statistics, Yksikkötestit, dokumentaatio, vehicle
 - Suurin osa City