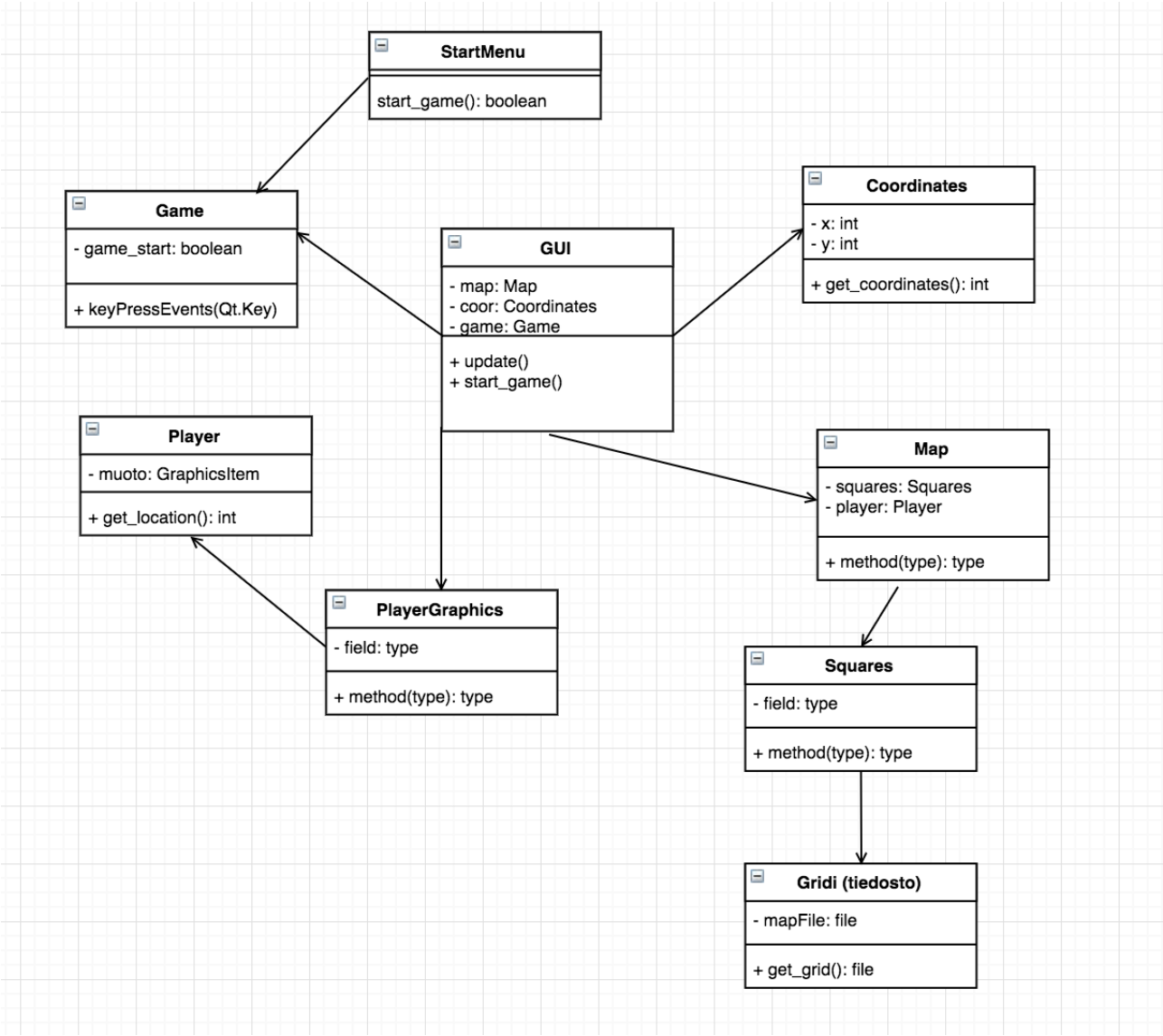
Y2 Tekninen suunnitelma

Aapo Linjama 587895, Elektroniikka ja sähkötekniikka, 3. vuosikurssi, 5.3.2019

Ohjelman rakennesuunnitelma



Käyttötapauskuvaus

Käyttäjä käynnistää pelin ajamalla peliä pyörittävän tiedoston Eclipsessä. Ensin näytölle aukeaa aloitusnäyttö, jossa on esimerkiksi teksti ”Y2-tasohyppely”, lisäksi ikkunassa on painike ”Aloita peli”, kun käyttäjä painaa tätä nappia saa ohjelmassa itse pelin aloittamiseen liittyvä boolean tyyppinen muuttuja arvon True ja pelilooppi lähtee käyntiin. Avautuu uusi ikkuna, jossa näkyville tulee pelikenttä, joka on jaettu neliöihin. Osa pelikentän osista on esteitä ja osa vapaasti kuljettavaa aluetta. Käyttäjä voi nyt alkaa ohjailemaan pelihahmoaan kentällä pyrkien pääsemään esteiden yli. Ohjaaminen tapahtuu näppäimistöllä. Ainakun käyttäjä painaa jotain ohjelmaan

määriteltyä näppäimistön nappulaa tutkitaan se ns. ’’keypresseventtinä” ja ohjelma etenee näppäimeen määritellyn operaation mukaisesti. Peli loppuu, kun käytäjää pääsee joko maaliin tai mahdollisesti toteutettavat viholliset saavat pelaajan ’’tuhotuksi’. Yksi mahdollisuus on myös aikaraja, jonka rajoissa pelikentästä on selviydyttävä.

Algoritmit

Kontrolleihin tarvittavat algoritmit vaativat sitä, että kun ohjelma saa käyttäjältä erilaisia keypresseventtejä, pitää ohjelmassa käynnistyä erilaisia prosesseja, jotka muokkaavaat pelihahmon koordinaatteja. Koordinaattimuutokset voivat tapahtua esimerkiksi kahden tai useamman kierron aikana, jotta liikkuminen vaikuttaa sulalavalta näytöllä. Eli käytännössä x ja y koordinaatteihin lisätään tai vähennetään jokin vakio summa, jonka jälkeen hahmot piirretään uudestaan seuraavalla peliloopin kierroksella. Muutokset tehdään siis hahmojen ’’älyyn’’, jonka jälkeen graafiset objektit piirretään ruudulle samaan paikkaan kuin äly.

Pelikentän piirtämiseen vaadittava algoritmi toimii siten, että tiedostosta luetaan merkki kerrallaan dataa, joka on string-muodossa. Tämän jälkeen riippuen mitä syöte sisältää piirretään pelikentän neliö joko vapaasti kuljettavaksi tai esteeksi.

Tietorakenteet

Pelikenttien varastoimiseen parhaiten sopivat tiedostotyyppiset tietotyypit, sieltä ne voidaan kätevästi lukea esimerkiksi listatyyppiseen muuttujaan. Listamuuttuja on helppo iteroida läpi ja piirtää pelikenttä. Pelihahmolle ja mahdollisille vihollisille ei oikeastaan tarvita erillisiä säiliöitä, sillä tarkoituksena ei ole luoda pelin aikana uusia vihollisia tai pelihahmoja. Tosin jos aikaa jää implementoida pelaajalle jokin ominaisuus, jolla hän voi tuhota vihollisia täytyy luoda säiliö, joka pitää kirjaa vihollisten määrästä. Mikäli kenttäeditoriin jää myös aikaa voidaan käyttäjän luoma kenttä lukea neliö kerrallaan tiedostoon, jossa taas jokainen erilainen neliö merkitään erilaisella tunnnuksella.

Aikataulu

Ensimmäiset pari viikkoa tulevat varmasti menemään siihen, että tutustuu PyQt5-kirjastoon ja muihin tarvittaviin kirjastoihin ja alkaa tutustumisen pohjalta kokeilemaan erilaisia ominaisuuksia. Viikkotasolla olisi hyvä saada ainakin alkuun käytettyä projektiin n. 10h. Toteutuksen voisi aloittaa sillä, että saa ensin ikkunan auki, jonka jälkeen alkaa luomaan objekteja ikkunaan, eli pelikenttää. Alkuun kenttää ei lueta vielä tiedostosta, vaan tarkoituksena on saada vain piirrettyä haluttuja objekteja näytölle. Tämän jälkeen voisi alkaa implementoida sitä, miten objektit saadaan liikkumaan näytöllä. Tärkeää on saada ensin pelikenttä ja pelihahmo näkyviin ikkunaan, jonka jälkeen lisäominaisuuksia voi lähteä implementoimaan. Graafista käyttöliitymää varten on toteutettava luokat itse pelihahmolle ja pelikentän osille eli neliöille, jonka jälkeen näille ’’äly’’-luokille voi alkaa luomaan graafisia objekteja omissa luokissaan.

Yksikkötestaussuunnitelma

Keskeisimpiä asioita ohjelman toiminnan kannalta ovat esimerkiksi se, että tiedostosta lukeminen onnistuu halutulla tavalla, sillä muuten pelikenttää ei saada oikeaan muotoon. Täytyy siis toteuttaa testejä, joissa varmistetaan, että kun tiedostosta luetaan dataa, niin piirretty pelikenttä on myös halutun lainen. Jokaiselle pelikentän eri tyyliselle osalle on määritelty jokin merkki esimerkiksi, jos luettu merkki on ’’0’’, niin ruutu pelikentällä on este.
Toinen oleellinen osa ohjelman kannalta on se, että käyttäjän antamat käskyt aiheuttavat halutun vasteen. Eli esim. kun painetaan ’’aloita peli’’ -nappia, niin luonnollisesti pelin täytyy käynnistyä, tai kun käyttäjä painaa kontrollinappeja niin pelihahmon täytyy liikkua halutulla tavalla. Näitä voidaan kokeilla yksikkötesteillä antamalla luokille ja metodeille, jotka näitä tapatumia käsittelevät ennalta määrättyjä käskyjä, joiden oletetut vasteet tiedetään.

Kirjallisuusviitteet ja linkit

Qt-kirjaston omat dokumentaatiot:

https://doc.qt.io/qt-5/

PyQt:n omat kirjastot

https://pypi.org/project/PyQt5/

Myös stackoverflow tulee olemaan varmasti käytössä:

https://stackoverflow.com/