Liisa Särkiö

Y2 projektisuunnitelma – tartuntatauti

1. Henkilötiedot

Projektini aihe on tartuntatauti.
Henkilötiedot: Liisa Särkiö, 100582910, Sähkötekniikan kandidaatti (Bioinformaatioteknologia), suunnitelma tehty 24.2.2022

2. Yleiskuvaus ja vaikeustaso

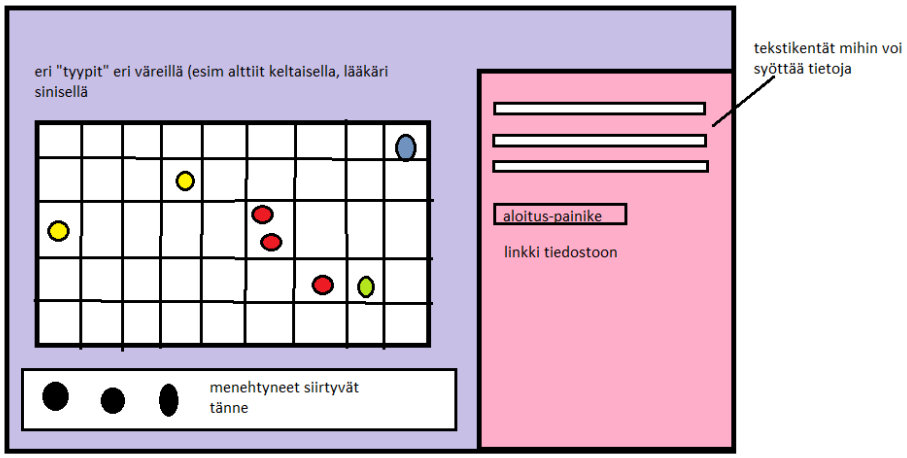
Projektini tarkoituksena on tuottaa simulaatio, joka mallintaa tartuntataudin leviämistä agenttipohjaisesti, eli yksittäisten ja itsenäisten toimijoiden/”hahmojen” toimintaa seuraten. Simulaation agentit ovat joko taudille alttiita, tartuttavia, parantuneita, tai simulaatiosta poistettuja eli menehtyneitä. Simulaation käyttäjä voi valita henkilöiden määrän, tartuttavien henkilöiden määrän alkutilanteessa, sekä muitakin alkutietoja. Lisäksi ohjelma kerää ja dokumentoi simulaation tuloksia tiedostoon. Tiedoston lisäksi käyttäjä pystyy tarkastelemaan simulaation etenemistä sekä kommunikoimaan sen kanssa graafisen käyttöliittymän avulla.

Haluan toteuttaa projektin vaativana

3. Käyttötapakuvaus ja käyttöliittymän luonnos

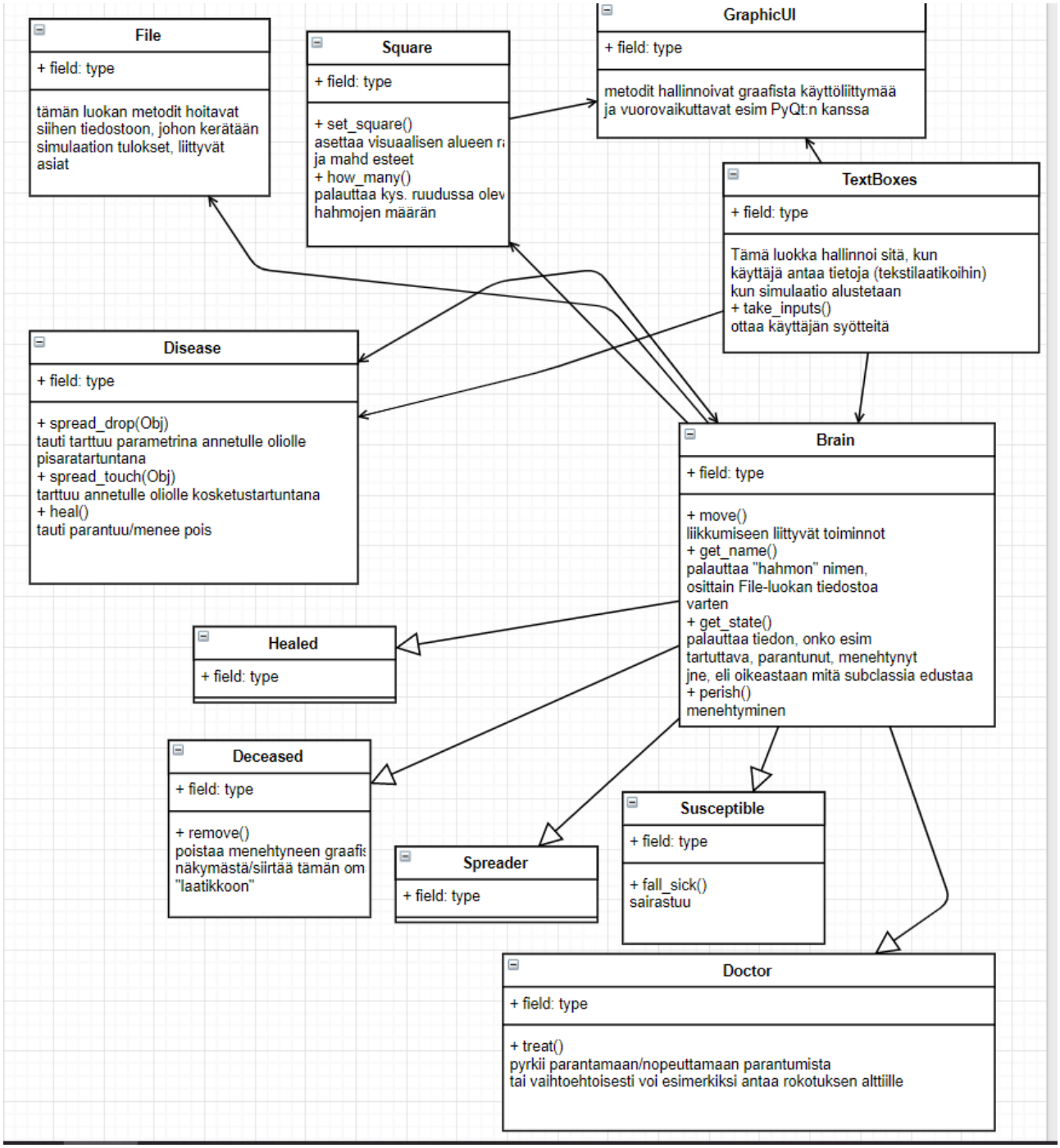
Ohjelma kommunikoi käyttäjän kanssa graafisen käyttöliittymän avulla, jonka voisi toteuttaa esimerkiksi PyQt:llä. En vielä kovin tarkasti osaa kuvailla käyttöliittymää, tai sitä mitä on mahdollista tehdä taitojeni ja käytössä olevien välineiden puitteissa, sillä en ole vielä tehnyt kurssin tehtävien graafinen käyttöliittymä -kierrosta.

Käyttäjä pystyy antamaan ohjelmalle komentoja kirjoittamalla tekstikenttiin tekstiä (esimerkiksi henkilöiden määrän alussa, säätämään tartuttavuutta (eli todennäköisyyttä, jolla tauti tarttuu)) sekä aloitus- ja lopetuspainikkeiden avulla. Ohjelman käyttötilanne voisi mennä yksinkertaisena esimerkiksi näin:
käynnistys → ennakkotietojen asettaminen (jonkinlainen main-moduuli saa nämä, graaf. UI:hin liittyvät luokat aktivoituvat) → aloitus → tekoälyt aloittavat toimintansa ja liikkumisen (eli eri ”rooleja” kuvaavat luokat aktivoituvat, kts. UML-malli alla) → tiedostoon aletaan kirjata tapahtumia ja tilastojen etenemistä, eli esim. tartuntojen ja menehtymisten määrää → simulaatio loppuu, kun tietyt ehdot täyttyvät tai kun käyttäjä lopettaa simulaation.



Yksinkertainen visualisaatio käyttäjälle näkyvästä ikkunasta yllä.

4. Ohjelman rakennesuunnittelu



Yllä hyvin alustava UML-suunnitelma projektini luokkarakenteesta. Ohessa luokkien keskeisiä metodeja, sekä lyhyet selitykset niiden tehtävistä. **Brain** on parent class eri ”rooleille” joita henkilöt simulaatiossa toteuttavat. Nämä roolit, siis esimerkiksi luokat **Susceptible**, **Healed** ja

Doctor perivät luokan **Brain.** Luokat **Square** ja **TextBoxes** liittyvät graafiseen käyttöliittymään, jonka toiminta ja luokkien väliset suhteet ovat minulle vielä melko mysteeri. Vaihtoehtoisuutta ja valintoja liittyen luokkarakenteeseen on esimerkiksi se, kuinka paljon taudin (**Disease**) toiminnasta – esimerkiksi tarttuminen, parantuminen jne. – toteutetaan kyseisessä luokassa, ja kuinka paljon tekoälyjen luokissa (**Brain** ja sen subclassit). Eräs tavoista, joilla suunnittelen tekeväni ohjelmasta edukseen erottuvan, on taudin tarttumisen jakaminen kosketus- ja pisaratartuntaan. Nämä tarttuisivat luonnollisesti eri periaattein. Lisäksi suunnittelen lisääväni projektiaiheen ohjeessa listattujen roolien lisäksi jonkinlaisen lääkärihahmon (**Doctor**) joka joko pyrkii parantamaan tai esimerkiksi rokottamaan muita hahmoja.

Esitettyjen luokkien lisäksi esimerkiksi jonkinlainen kokoava **Population**-luokka, joka esittäisi kaikkia simulaation henkilöitä voisi olla hyödyllinen esimerkiksi tiedostoon (**File**) liittyvissä asioissa.

5. Tietorakenteet

   Lyhyehkön alkututustumisen ja -suunnittelun jälkeen uskon esimerkiksi listojen olevan hyödyllinen tietorakenne projektini toteutuksessa. Listoja voi käyttää esimerkiksi pitämään kirjaa siitä (samalla kun tietoja kirjataan tiedostoon sekä auttamaan tässä), mitkä hahmoista ovat tartuttavia, mitkä alttiita jne. Muokattava tietorakenne on tähän hyvä, sillä hahmojen tila muuttuu, esim. tartuttava voi parantua.

6. Tiedostot ja tiedostoformaatit

   Ohjelman tulee tuottaa tiedosto, johon se siirtää ja ”outputtaa” tietoja simulaatiosta. Kyseinen tiedosto on tekstitiedosto. Lisäksi tiedostossa voisi olla jonkinlaisia graafeja tms., jotka havainnollistavat taudin leviämistä, mutta näiden toteuttaminen voi olla haastavaa.

   Koska käyttäjä antaa ohjelmalle tietoja graafisen käyttöliittymän avulla, en usko, että ohjelman tarvitsee ladata tietoja tiedostosta. Tosin voi olla, että käyttäjän antamien syötteiden ja ohjelman välille tarvitaan jokin tiedosto ikään kuin välittämään tiedot, mutta tämä selvinnee, kun tutustun paremmin graafisten käyttöliittymien toimintaan

7. Algoritmit

Ohjelmassa esiintyvien ongelmien ratkaisemiseen tarvitaan esimerkiksi jonkinlainen liikkumisalgoritmi, joka tulee todennäköisesti olemaan jonkinlainen yhdistelmä tekoälyjen ja roolien toimintaa sekä sattumanvaraisuutta. Ylipäätään ohjelmassa tullaan tarvitsemaan paljon todennäköisyyslaskentaa ja siihen liittyvien matemaattisten kaavojen käyttöä, kenties jonkinlaisilla painokertoimilla, jotka kuvaavat älykkyyttä.

Lisäksi tarvitaan tietoja ja kaavoja, jotka kuvaavat erilaisten tartuntatautien tarttumista.

8. Testaussuunnitelma

   Aion toteuttaa yksikkötestausta projektilleni, eli ohjelmassa täytyy olla jonkinlaisia testaukseen liittyviä luokkia, jotka siten ”throw Errors” jos niitä löytyy. Projektin ominaisuuksia, jotka on tärkeä ja mahdollista testata ovat esimerkiksi se, onko tartuttaminen ”oikeellista” (eli esimerkiksi EI tartuttava → parantunut), sekä hahmojen liikkuminen (eivät esimerkiksi liiku seinän sisään).

9. Kirjastot

En suunnittele käyttäväni muita kirjastoja, kuin mitä aiheen tehtävänannossa mainitittiin. Suunnittelen toteuttavani graaf. käyttöliittymän PyQt:llä, esimerkiksi PyQt 5:llä.

10. Aikataulu

Seuraava periodini on melko kiireinen, joten minulle sopivat hyvin checkpointtien 1 ja 2 ajankohdiksi projektiohjeessa mainitut takarajat, siis checkpoint 1 24.3 ja checkpoint 2 14.4.

Muuta aikataulutusta:

Lisää suunnittelua ja tutkimustyötä – 5 h

Lopullinen UML-malli – 10 h

KOODAAMISEN ALOITTAMINEN

CP 1

GRAAFISEN KÄYTTÖLIITTYMÄN SUUNNITTELUA JA TOTEUTUSTA 20 h

Testauksen tekemistä 20 h

CP 2

VIIMEISTELYÄ

Lopullinen palautus 12.5

11. Kirjallisuusviitteet ja linkit

Kurssimateriaali, python-dokumentaatio, nettisivut ja palstat kuten Stack Overflow (tieto- ei kopiointilähteenä!).