

## **5. Simple epidemic simulator**

TiLa2 loppuprojekti

**Antti Lumppio**  
015 111 609

## 1 Introduction

Projektissa tehtävänä oli luoda satunnaisia kävelijöitä  $N \times N$  tasolle. Kävelijöillä on jokaisella ajan hetkellä määritelty paikka  $x$ - ja  $y$ -koordinaatein. Kävelijät voivat ajan hetkellä liikkua satunnaiseen suuntaan 1 askeleen, kuten yhden askeleen positiiviseen  $x$ -suuntaan. Kävelijät eivät voi liikkua pois tasosta, vaan ne "ilmestyvät" uudelleen tason toiselta puolelta. Oletetaan, että taso on nyt määritelty  $8 \times 8$  tasoksi ja kävelijä sijaitsee pisteessä  $(8, 1)$  ja kävelijä liikkuu yhden askeleen negatiiviseen  $y$ -suuntaan. Tällöin kävelijä liikkuu pisteeseen  $(8, 8)$ .

Kävelijöillä on myös sijainnin lisäksi aina määritelty jokin terveydentila. Tila voi olla **terve**, **sairas** tai **immuuni**. Tasolla liikkuu sairaus, joka voi tarttua kävelijöiltä toisille, mikäli ne ovat samassa ruudussa sairaan kävelijän kanssa. Alussa on määritelty tietty määrä sairaita, jotka voivat tartuttaa muita samassa ruudussa olevia terveitä kävelijöitä määrätyllä todennäköisyydellä. Vain terveet voivat sairastua tautiin. Sairaot voivat muuttua immuuneiksi jokaisella aika-askeleella määrätyllä 1% todennäköisyydellä.

Kun taso ja kävelijät on luotu, tarkastellaan miten tauti lähtee etenemään tasossa. Projektissa piti ylläpitää dataa siitä, miten monta kävelijää jokaisella ajan hetkellä on sairaana tai immuunina tautiin. Näistä tiedoista piti luoda kuvaaja. Yksi tehtävistä on myös kuvata tasoa ajan hetkillä jollain ohjelmalla esimerkiksi tehtävänannon suosittelemalla Ovitolla. Tason kuvaajasta nähdään kävelijän sijainti, sekä sen tila. Kerättyä dataa piti tarkastella katsomalla, miten sairaus kehittyy eri alkumääreiden perusteella. Dataa kerättiin muuntelemalla tason kokoa suhteessa kävelijöihin

## 2 Methods

Ohjelmassa luon tietyn määrän satunnaisia kävelijöitä satunnaisiin paikkoihin. Kävelijät voivat liikkua yhden askeleen satunnaiseen suuntaan jokaisella aika-askeleella. Tasolle ei erikseen tarvitse tehdä omaa taulukkoa, vaan voidaan suoraan käyttää kävelijöiden taulukkoa ja tällä mallintaa tasoa. Taso on määritelty kävelijöiden taulukossa siten, että kävelijöiden  $x$ - ja  $y$ -rajat on tason sivun pituus. Kävelijöiden liikkumissuunta määritellään satunnaisen numeron avulla väliltä  $[0, 1[$ . Tämä väli on tasaisesti jaettu neljään osaan eli  $+x$ ,  $-x$ ,  $+y$  ja  $-y$  -suuntaan. Samaten terveen kävelijän sairastuminen samassa ruudussa olevan sairaan kävelijän takia määritellään satunnaisen numeron avulla verraten tätä sairastumisen todennäköisyyteen.

Kävelijöiden sijainti on määritetty  $x$ - ja  $y$ -koordinaateilla kokonaisluvuil-

la  $1, 2, \dots, N$  asti, missä  $N$  on tason sivun pituus. Kävelijöiden terveydentila on määritelty myös kokonaisluvulla. Tila voi olla joko: **0=Terve**, **1=Sairas** tai **2=Immuuni**. Alussa on määritelty tietty määrä sairata ja immuuneja. Fortran-ohjelmani luo kaksi tekstitiedostoa: xyz-tiedoston Ovitolle ja datataulukon kuvaajaa varten. XYZ-tiedoston formaatti on seuraava: ensimmäiselle riville kirjoitetaan kävelijöiden määrä, toiselle riville haluama kommentti (ohjelmassani se kertoo aika-askeleen) ja loppuilla riveillä on kävelijöiden tieto eli sijainti  $(x,y)$  sekä terveydentila  $(0/1/2)$ . Sairastietojen kuvaajan teen Python-kielellä.

### 3 Implementation of the methods

Ohjelma koostuu yhdestä moduulista ja pääohjelmasta. Pääohjelma kirjoittaa käyttäen moduulia ja kirjoittaa dataa kahteen tiedostoon. Moduulissa on määritelty pääohjelmassa kutsutut aliohjelmat.

Moduulissa ensiksi on määritelty ohjelmassa käytetty oma muuttuja tyyppi **dalsijat**, joka koostuu kolmesta kokonaisluvusta. Kokonaisluvut ovat  $x, y$  ja  $s$ . Muuttujat  $x$  ja  $y$  ovat kävelijän sijainti  $N \times N$  tasossa. Muuttuja  $s$  taas on kävelijän terveydentila eli **0=Terve**, **1=Sairas** tai **2=Immuuni**. Moduulissa on tämän jälkeen määritelty kaksi aliohjelmaa.

Ensimmäinen aliohjelma on aliohjelma **luodallaajat**, mikä luo tason satunnaiset kävelijät. Aliohjelmassa luodaan **dalsijat**-tyyppiselle muuttujalle  $N^2$  kokoinen taulukko. Koko on siis tason kokoiseen taulukon mahtuvien alkoiden maksimimäärä. Taulukko sisältää kaikkien kävelijöiden tarvittavat tiedot eli sijainnin ja terveydentilan. Aliohjelmalle annetaan kutsussa 5 arvoa: käytetyn muuttujan nimi, kävelijöiden määrä, tason sivun pituus, alussa sairaiden kävelijöiden määrä ja alussa tautiin immuunien kävelijöiden määrä. Aliohjelma luo do-luupin avulla taulukon. Kävelijöiden sijainti määritellään satunnaisen numeron avulla. Satunnainen numero saadaan väliltä  $[0, 1]$ , mikä skaalataan sijainniksi tasolla Fortranin sisäänrakennetun floor-funktion avulla. Satunnainen numero sisäänrakennetusta aliohjelmasta **random\_number**, jota kutsutaan komennolla **call random\_number**. Alussa sairaiden ja immuunien määrä määritellään if-luupin avulla. Sairaita satunnaisia kävelijöitä luodaan ensimmäisestä alkioista kutsussa annettuun lukuun asti. Tämän jälkeen määritellään alussa immuunien kävelijöiden määrä. Loput kävelijät ovat terveitä.

Moduulissa on toinen aliohjelma nimeltä **liikkuu**. Tämä aliohjelma liikuttaa **dalsijat**-tyypin muuttuja taulukon kaikkia kävelijöitä yhden aika-askeleen verran. Kävelijät liikkuvat siis yhden askeleen satunnaiseen suuntaan. Alioh-

jelmalle annetaan parametreinä kävelijöiden taulukon nimi, kävelijöiden määrä ja tason sivun pituus. Kävelijöiden liikkumissuunta päätetään satunnaisen numeron avulla. Satunnainen muuttuja kutsutaan aliohjelmalla **random\_number**, joka antaa luvun väliltä  $[0, 1[$ . Tämän väli on jaettu neljään yhtä suureen väliin, joista päätetään etenemissuunta. Väli  $[0, 0.25[$  on suuntaan -x, väli  $[0.25, 0.5[$  on suuntaan +x, väli  $[0.5, 0.75[$  on suuntaan -y ja väli  $[0.75, 1.0[$  on suuntaan +y. Sille välille mille satunnainen numero osuu, niin kävelijä liikkuu.

Kun kaikki kävelijät ovat liikkuneet yhden askeleen johonkin suuntaan, niin tarkastetaan onko kävelijöitä samoilla laatoilla. Mikäli on, niin terveillä kävelijöillä on moduulissa määritetyn **p\_tar** todennäköisyys sairastua tautiin, jos terve kävelijä on sairaan kanssa samalla laatoilla. Tämä tarkistus tehdään kahden do-luupin ja viiden if-luupin avulla. Mikäli kävelijät ovat samassa ruudussa jokin toisen kävelijän kanssa, niin niiden koordinaatit ovat samat. Tämän takia voidaan tarkastella aluksi kävelijöiden x-koordinaatteja. Mikäli x-suunnan sijainti on sama, niin siirrytään tarkastelemaan y-koordinaatteja. Jos y-koordinaatitkin ovat samat, niin if-luuppien avulla sairas kävelijä sairastuttaa terveen kävelijän. Immuuneja ei voi enää sairastuttaa. Sairastuminen tapahtuu aliohjelma **random\_number** avulla. Mikäli kutsuttu satunnainen numero on pienempi kuin **p\_tar**, niin kävelijä sairastuu.

Lopuksi aliohjelmassa käydään do-luupilla läpi kaikki kävelijät, jotka ovat sairaita. Mikäli nämä ovat sairaita näillä on  $p_{imm} = 0.01 = 1\%$  todennäköisyys parantua sairaudesta ja tulla immuuniksi sille. Tämä myös tehdään aliohjelman **random\_number** avulla. Mikäli satunnainen luku on pienempi kuin 0.01, niin kävelijä tulee immuuniksi sairaudelle.

Pääohjelma käyttää moduulia. Pääohjelmassa on määritelty aliohjelmalla **random\_seed** satunnaiselle numerolle siemen, jotta saataisiin yhteisiä tuloksia satunnaisille numeroille. Ohjelmassa avataan kaksi tekstitiedostoa **kavelijat.txt** ja **sairastiedot.txt**. **Kavelijat.txt**-tiedostoon kirjoitetaan kävelijöiden tieto jokaisella aika-askeleella XYZ-formaatissa. Tekstitiedostoon **sairastiedot.txt** kirjoitetaan kävelijöiden sairastiedot jokaisella aika-askeleella taulukkona. Taulukossa järjestys on: aika-askel, terveiden määrä, sairaiden määrä ja lopuksi immuunien määrä. Kävelijät luodaan kutsumalla moduulin aliohjelmaa **call luodallaajat** halutuilla parametreillä. Alku tiedot nyt kirjoitetaan molempiin tiedostoihin. Tämän jälkeen tehdään halutun kokoinen do-luuppi, jossa kävelijät liikkuvat. Jokaisella aika-askeleella kirjoitetaan molempiin tiedostoihin kävelijöiden tiedot. Kävelijät liikkuvat moduulin aliohjelman kutsulla **call liikkuu**. Kun do-luuppi loppuu eli kävelijät ovat liikkuneet halutun määrän verran, niin tekstitiedostot suljetaan ja tallennetaan.

Haluttuja tuloksia muokataan muokkaamalla pääohjelman ja moduulin eri parametrejä. Animoitu kuva kävelijöiden liikkumisesta tasolla luodaan Ovi-to ohjelman avulla käyttäen XYZ-formaatissa olevaa tekstitiedostoa **Kavelijat.txt**. Kuvaaja luodaan triviaalilla Python-ohjelmalla, joka käyttää kirjastoa **matplotlib.pyplot** luomaan kuvaajan. Ohjelmaa kääntäessä täytyy ensiksi linkittää moduuli pääohjelmaa, jonka jälkeen saadaan käännettyä ohjelma, joka voidaan suorittaa. Kääntämiseen löytyy tarkemman ohjeet src-kansiosta README-tekstitiedostosta.

## 4 Results

Tarkastellaan muuttelemalla parametrien arvoja, että miten tulokset vaihtelevat. Oletussairastumis todennäköisyys on  $p_{tar} = 0.4 = 40\%$ . Sairaita alussa on aina prosentti kävelijöistä. Ohjelmaa testataan vaihtelemalla kävelijöiden määrää suhteessa pinta-alaan, vaihtelemalla sairastumisen todennäköisyyttä sekä asettamalla osan väestöstä immuuneiksi tautiin jo alussa.

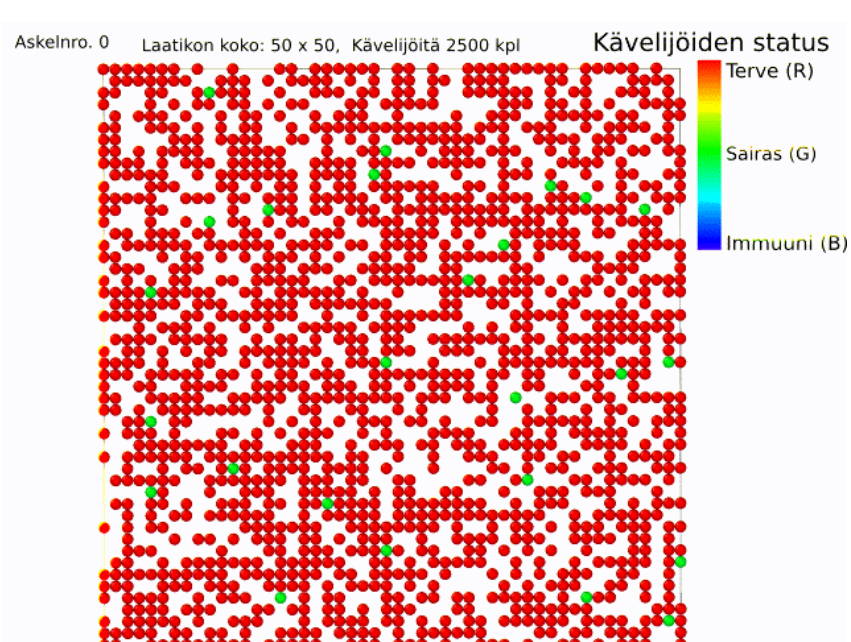
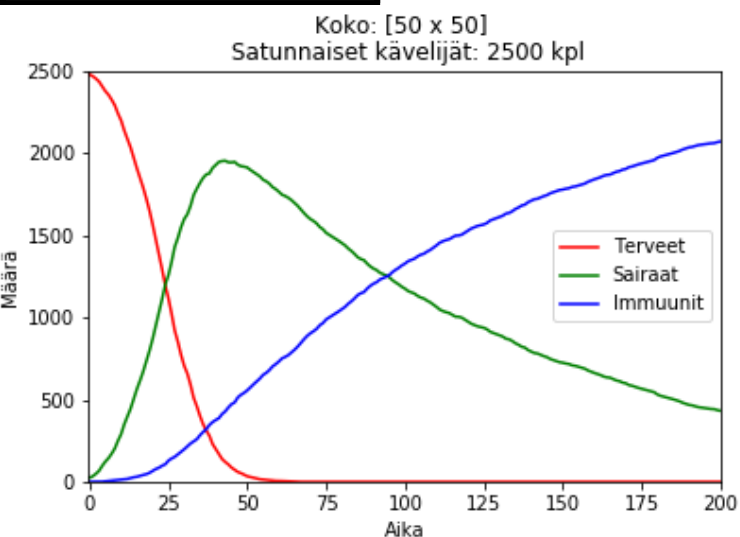
Kävelijöiden määrän merkitystä testataan neljällä eri arvolla: 1, 0.68, 0.36 ja 0.08 (kävelijät/maksimi pinta-ala). Näissä testauksissa käytetään oletus sairastumistodennäköisyyttä.

Tämän jälkeen koodia testataan muuttamalla sairastumisen todennäköisyyttä. Käytetään kahta eri arvoa pientä  $0.1 = 10\%$  ja suurta  $0.9 = 90\%$ . Testataan sairastumisen todennäköisyyden vaihtelua kävelijät/pinta-ala -arvoille: 1, 0.68, 0.36. Arvolle 0.08 ei tarvitse testata, sillä siinä sairaus leviää hyvin hitaasti.

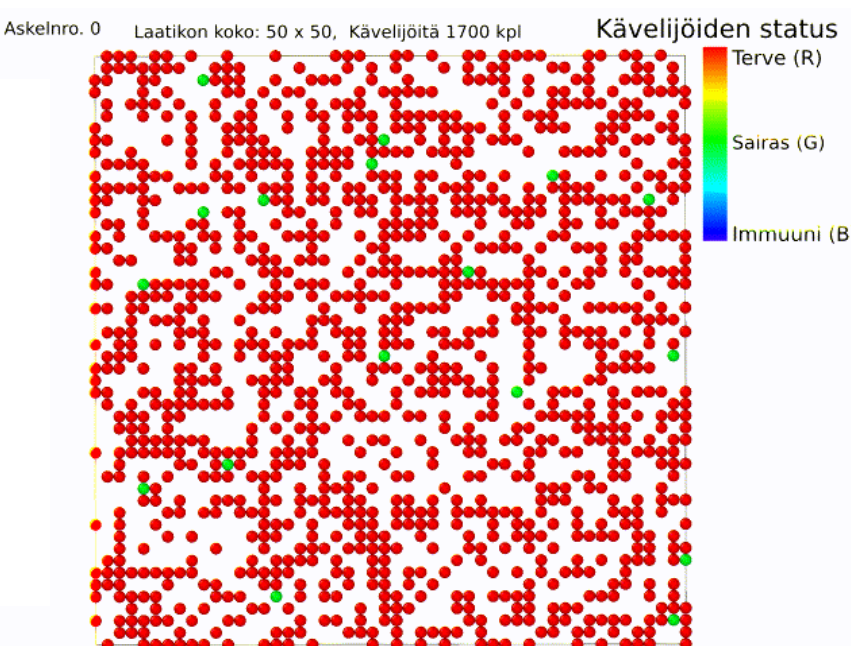
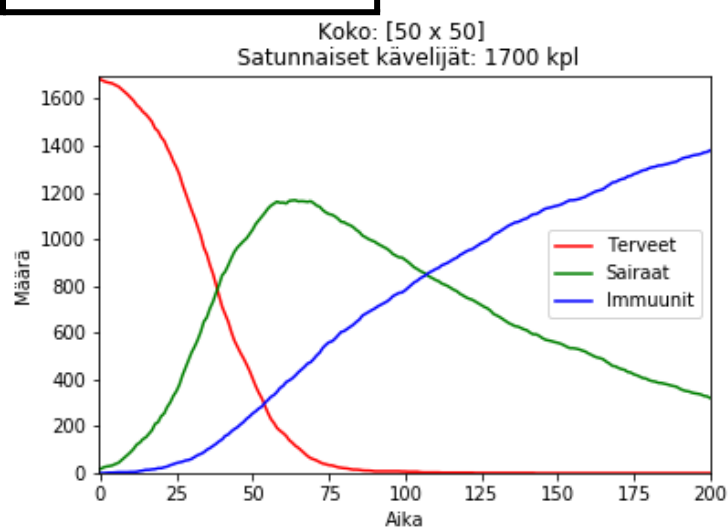
Lopuksi muutetaan osa väestöstä alussa jo immuuniksi tautiin. Valitaan, että  $0.1 = 10\%$  väestöstä on alussa jo immuuneja taudille. Testataan alussa jo immuunien vaikutusta kävelijät/pinta-ala -arvoille: 1, 0.68, 0.36. Arvolle 0.08 ei taaskaan tarvitse testata, sillä siinä sairaus leviää hyvin hitaasti.

Seuraavilla sivuilla on kuvat kuvaajista ja gif-animaatioiden tasojen alkua-setelmat eli ensimmäiset kuvat tasoista eri testauksilla. Jokaisesta testatus-tasosta on gif-animaatiot **run**-kansiossa. Gif-animaatioissa ei korostetusti erota päällekkäisiä kävelijöitä. Animaatiot näyttävät muutokset aina kymmenen aika-askeleen välein, koska muuten tiedostokoot olisivat liian suuria. Korkeimman arvon väri on päällimmäinen eli joko **sairas=1** tai **immuuni=2**. Terveydentilaa kuvataan väreillä seuraavasti: **terve=punainen, sairras=virheä, immuuni=sininen**.

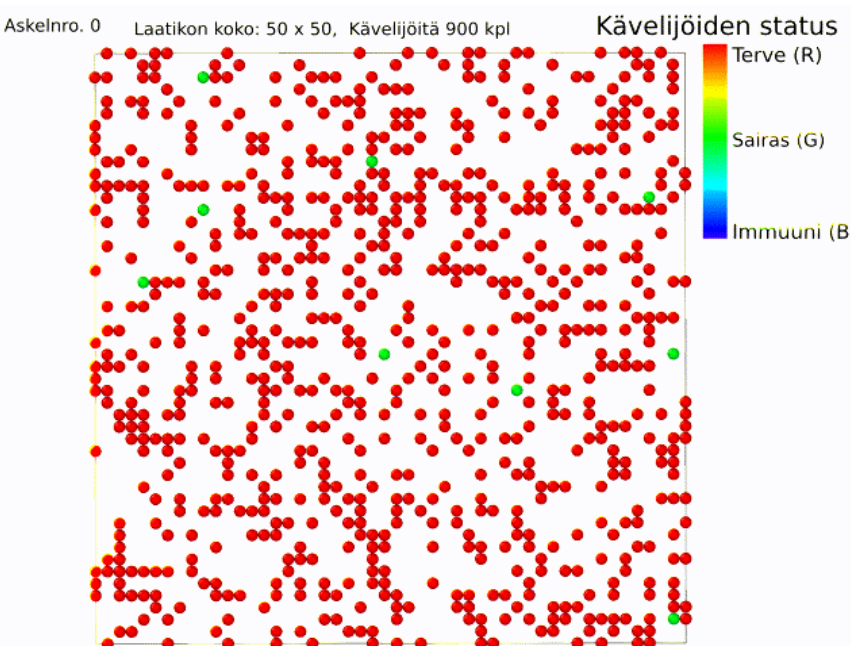
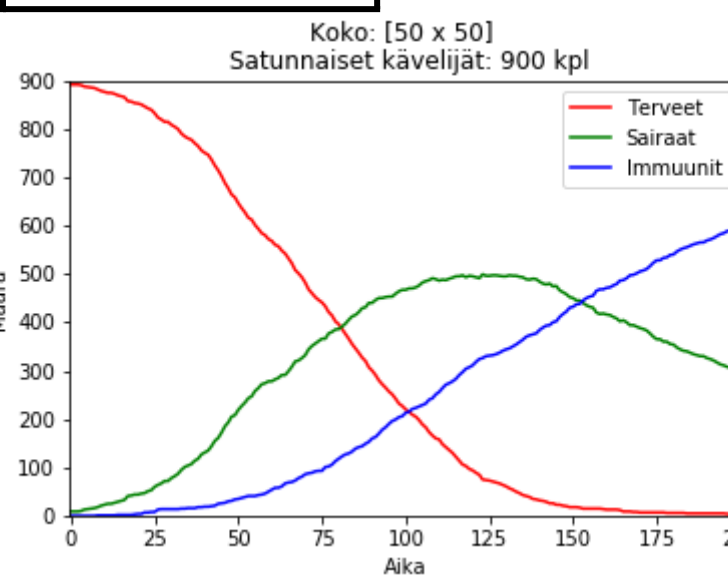
p\_tar=0.4  
Immuuneja alussa=0



p\_tar=0.4  
Immuuneja alussa=0

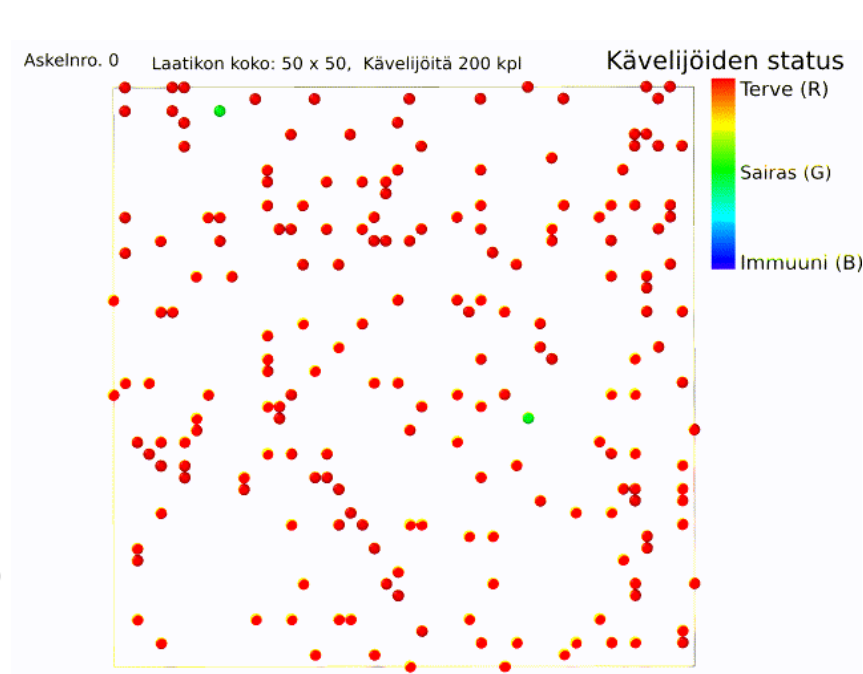
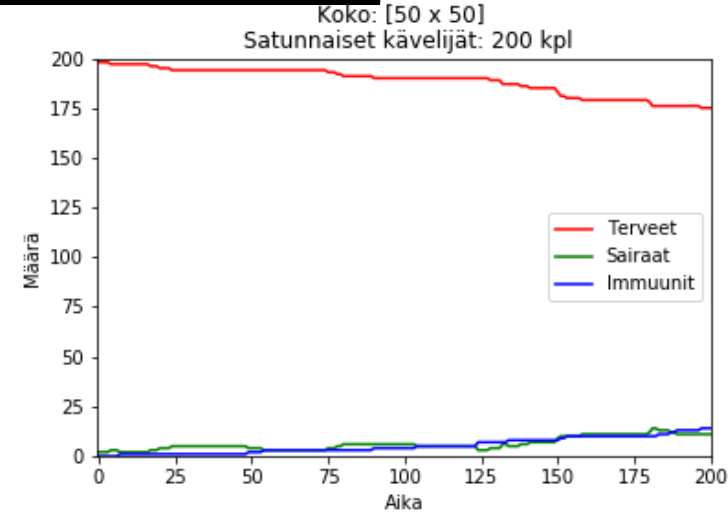


p\_tar=0.4  
Immuuneja alussa=0

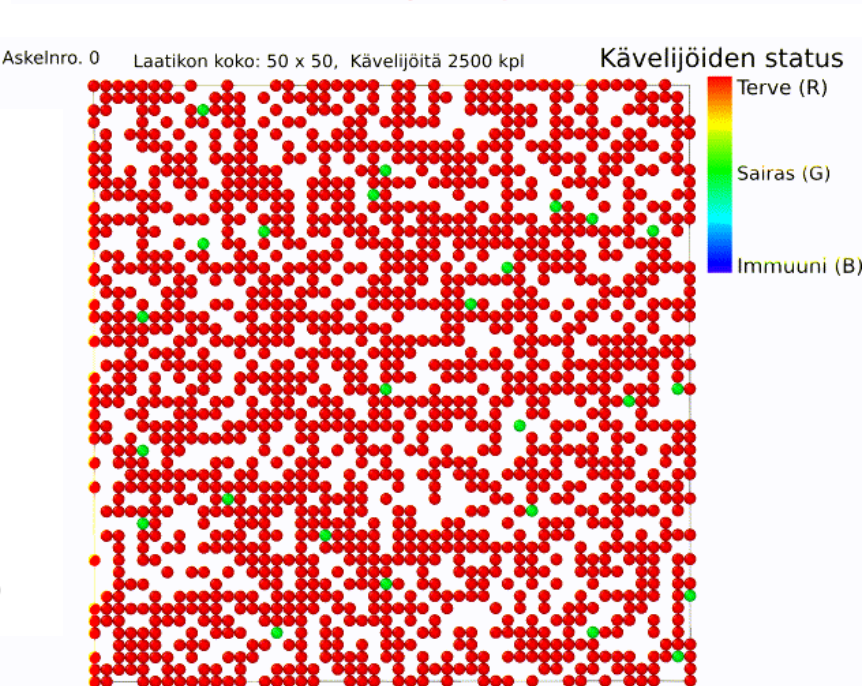
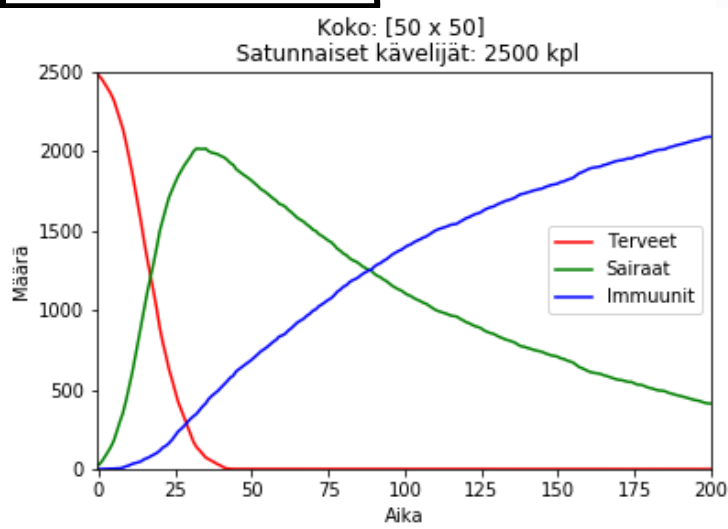




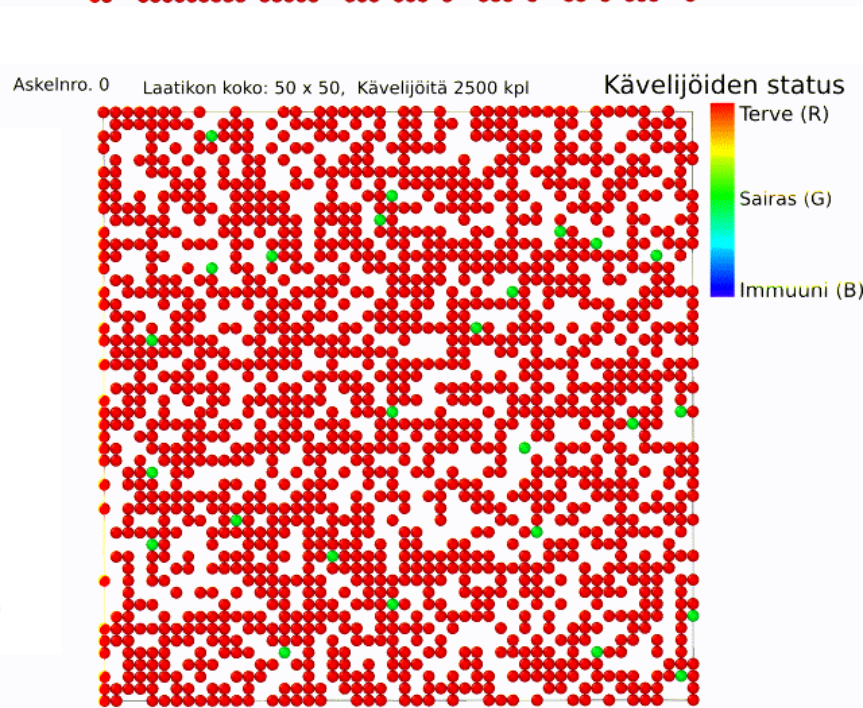
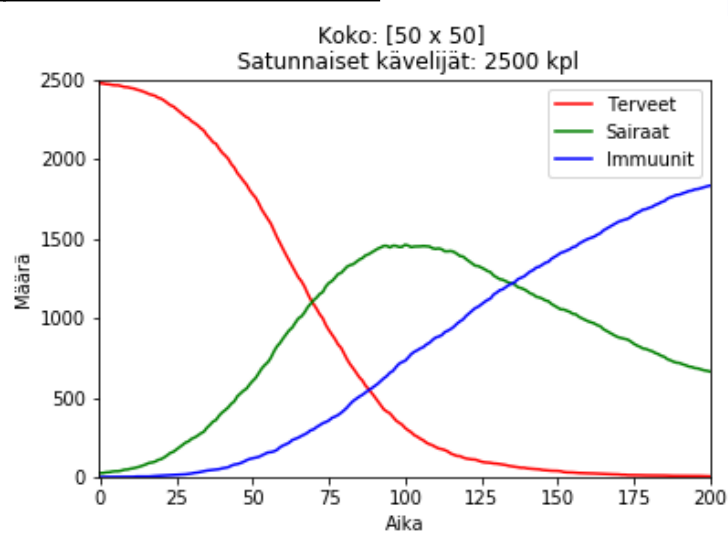
$p_{tar}=0.4$   
Immuuneja alussa=0



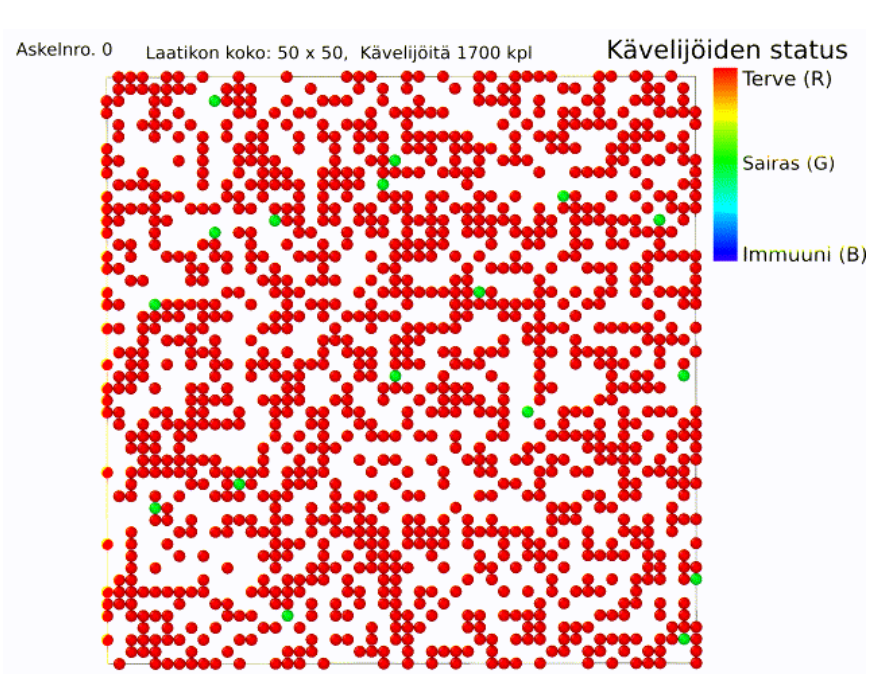
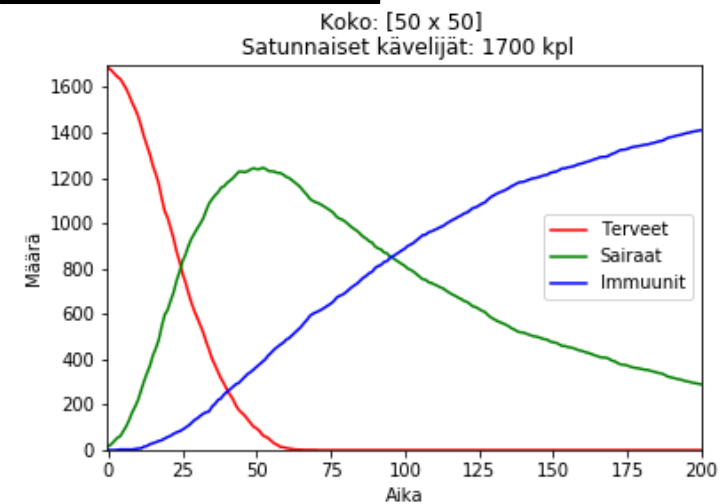
$p_{tar}=0.9$   
Immuuneja alussa=0



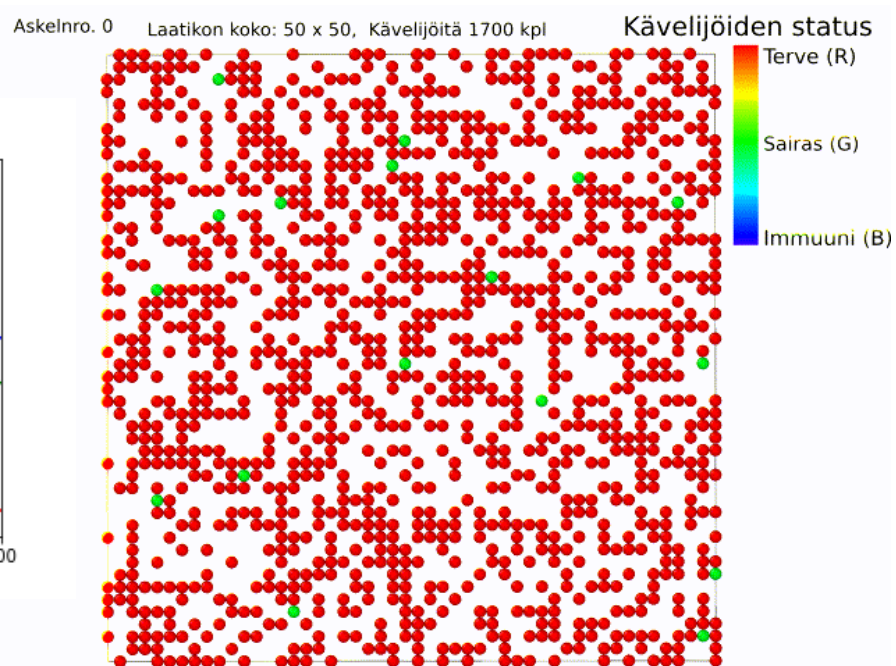
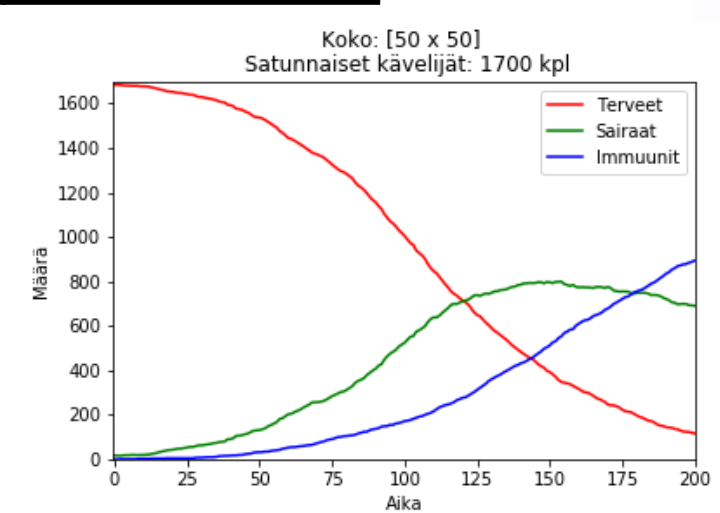
$p_{tar}=0.1$   
Immuuneja alussa=0



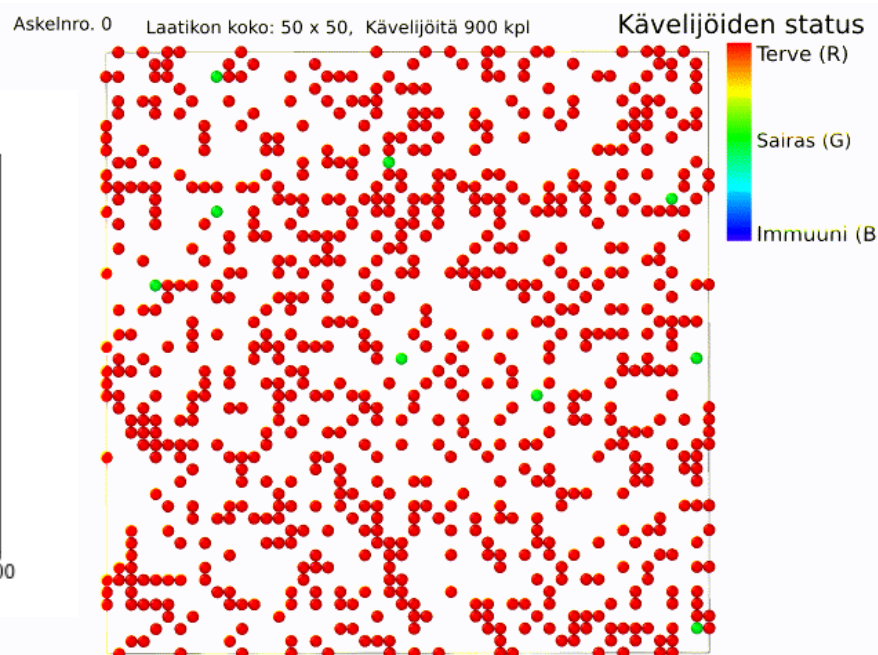
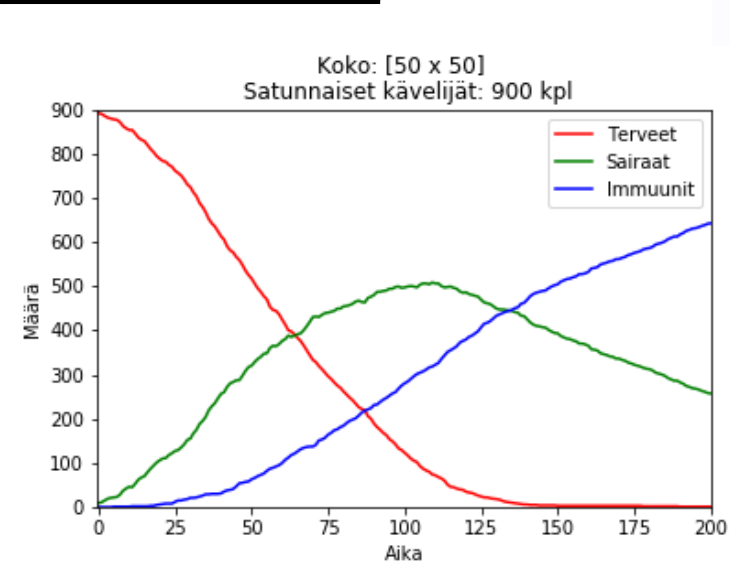
$p_{tar}=0.9$   
Immuuneja alussa=0



$p_{tar}=0.1$   
Immuuneja alussa=0

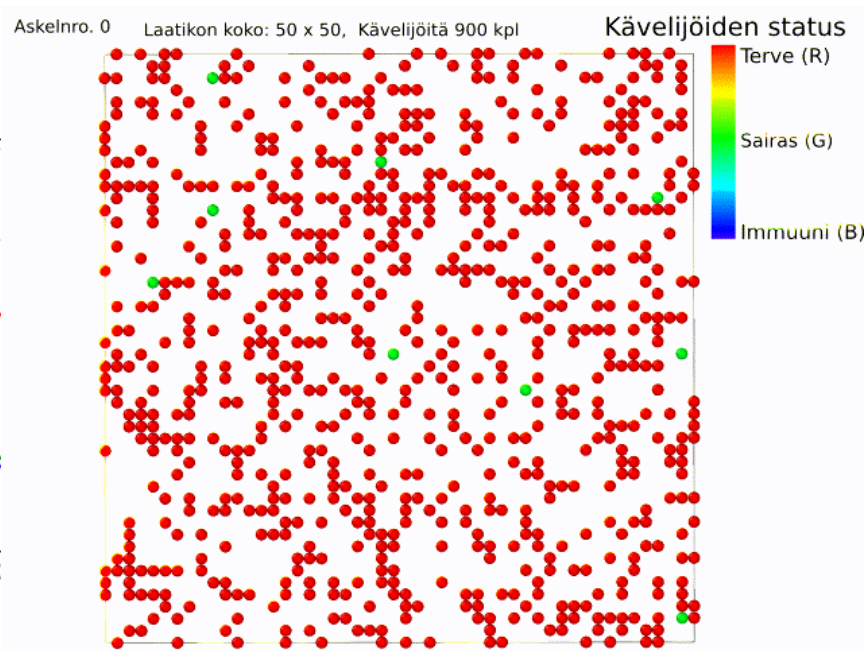
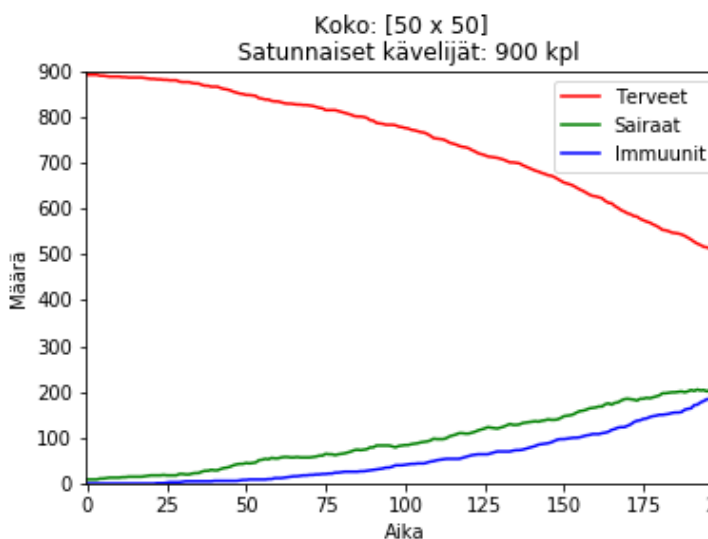


$p_{tar}=0.9$   
Immuuneja alussa=0

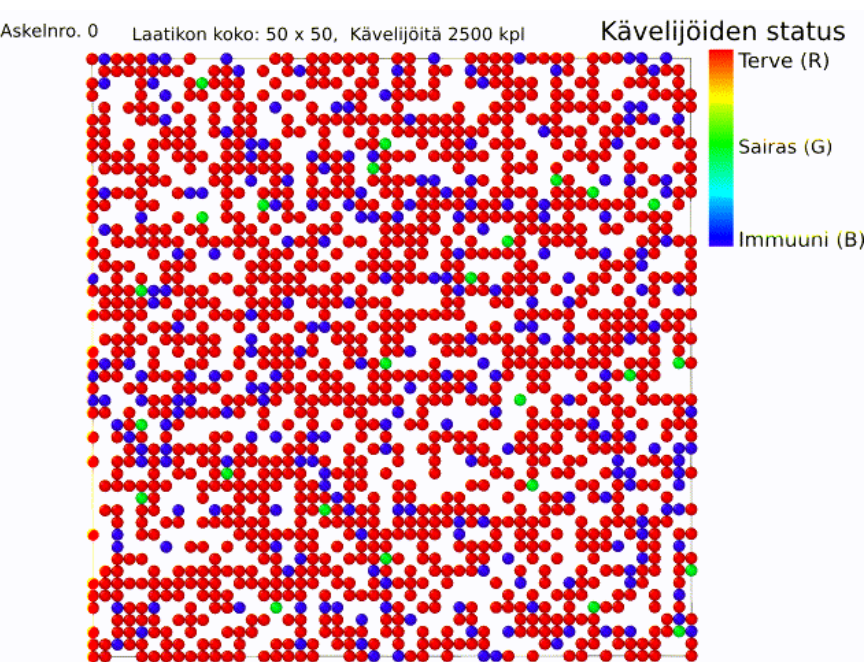
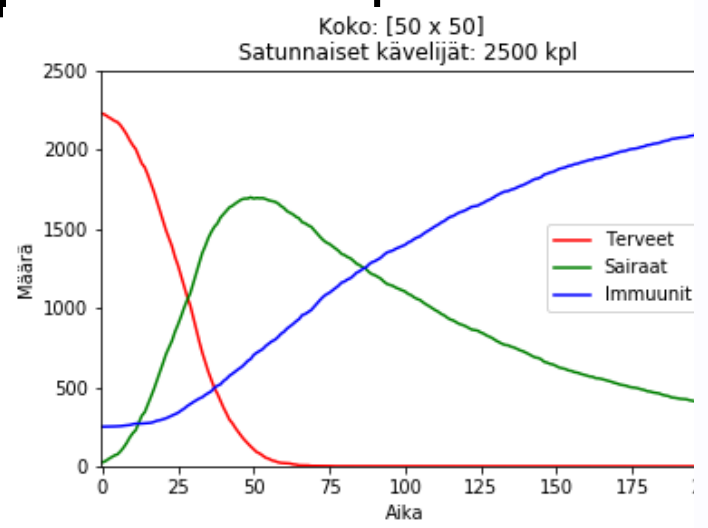




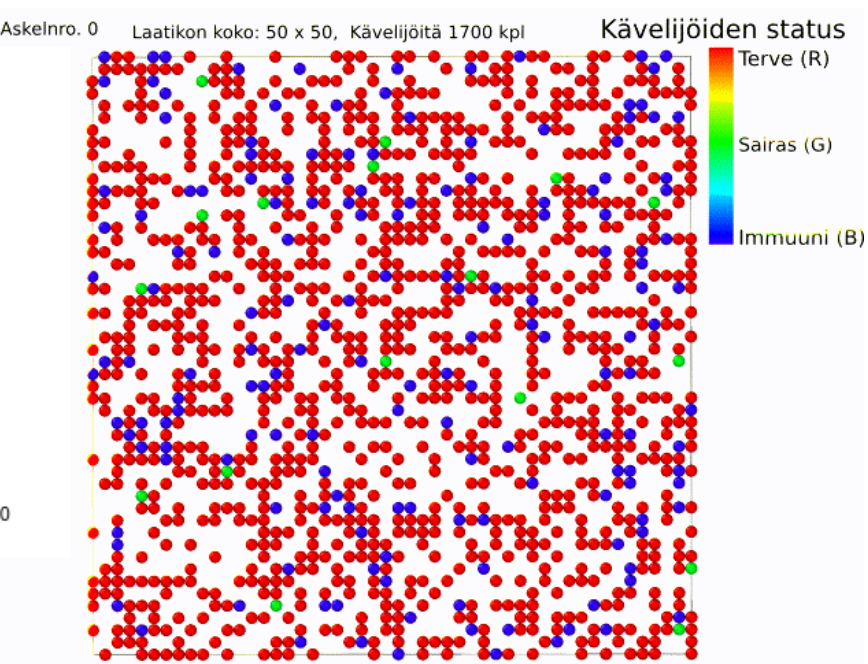
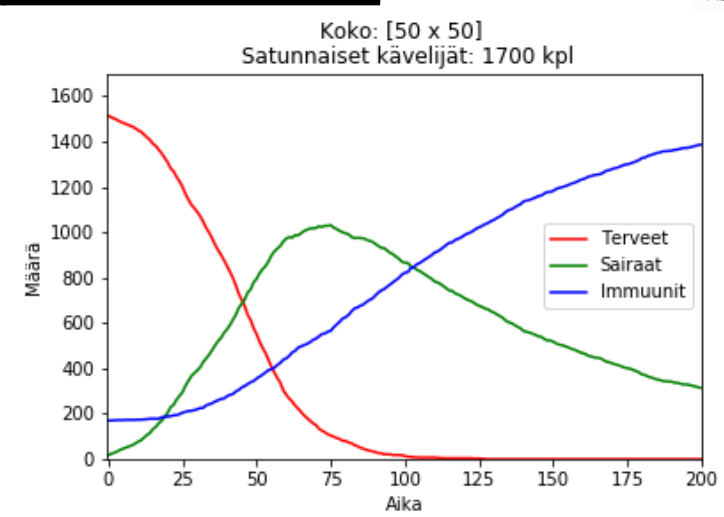
p\_tar=0.1  
Immuuneja alussa=0



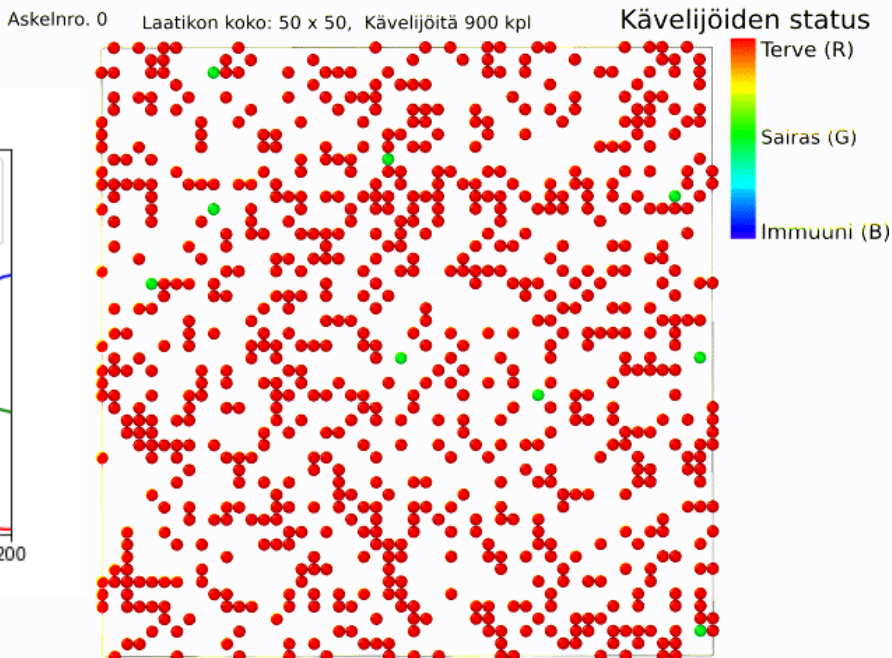
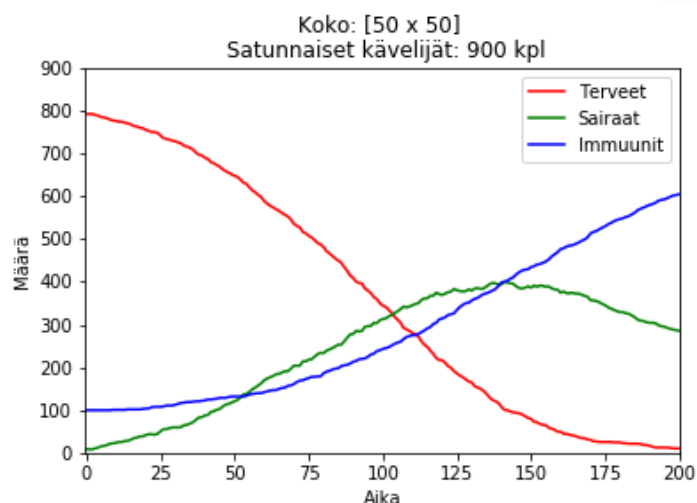
p\_tar=0.4  
Immuuneja alussa=250  
eli 10%



p\_tar=0.4  
Immuuneja alussa=170  
eli 10%



$p_{tar}=0.4$   
Immuuneja alussa=90  
eli 10%



## 5 Conclusions

Huomataan, että mikäli sairaus tarttuu, niin on todennäköistä, että ennen pitkään koko väestö tulee sairastamaan taudin. Täten myös pidemmän ajan myötä koko väestö tulee olemaan immuuni taudille. Tästä ei kuitenkaan voida olla aivan täysin varma, sillä kävelijät/pinta-ala arvolla 0.08 nähdään, että taudin leviäminen on todella hidasta. Tauti ei siis mahdollisesti leviä koskaan, jos taudin tarttuminen on alhainen ja väestön suhden pinta-alaan on pieni. Jo kävelijät/pinta-ala -arvolla 0.36 jo nähdään, että mikäli tauti on huonosti tarttuva eli  $p_{tar}=10\%$ , niin taudin leviäminen miltein lakkaa. Nähdään kuitenkin, että miltein kaikilla testatuilla parametreilla pidemmän ajanjakson myötä miltein kaikki kävelijät tulevat sairastamaan taudin. Kun tauti on helposti tarttuva, niin nähdään, että suurin osa väestöstä tulee nopeasti sairastamaan taudin. Täten myös iso osa väestöstä on nopeasti immuuni taudille. Immuunien kävelijöiden läsnäolo alusta lähtien hidastaa taudin leviämistä.

Näillä havainnoilla voidaan päätellä, että ohjelmassa kävelijällä parhain mahdollisuus olla sairastumatta tautiin on liikkua harvasti asutulla alueella, jossa on paljon immuuneja jo paikalla. Jos kävelijä kuitenkin haluaa sairastaa taudin mahdollisimman nopeasti, niin kannattaa elää alueella, joka on hyvin tiiviisti asuttu, ja jossa tauti on todella tarttuva.

Realistisemman tuloksen ohjelmalla saisi ainakin parantamalla kävelijöiden älykkyyttä. Tällä tarkoitan sitä, että terveet kävelijät aktiivisesti väistäisivät sairaita kävelijöitä, mutta pakon myötä voisivat tulla alttiiksi sairastumiselle. Satunnaiset kävelijät myös syntyvät satunnaisille kohdille. Tämä tarkoittaa sitä, että vaikka kävelijöiden määrä suhteessa pinta-alaan olisi 1, niin kävelijät eivät hyvin suurella todennäköisyydellä asetu niin, että jokaisessa pisteessä olisi yksi kävelijä. Kävelijät myös todennäköisesti asuisivat tietyssä rykelmässä omissa joukoissaan. Ohjelmassa kävelijä voi kävellä loputtomiin ilman määränpäättä. Tällainen järjestelmä ei oikein kykene hyvällä tarkkuudella ennustamaan, miten tauti leviäisi esim. ihmisten keskuudessa.