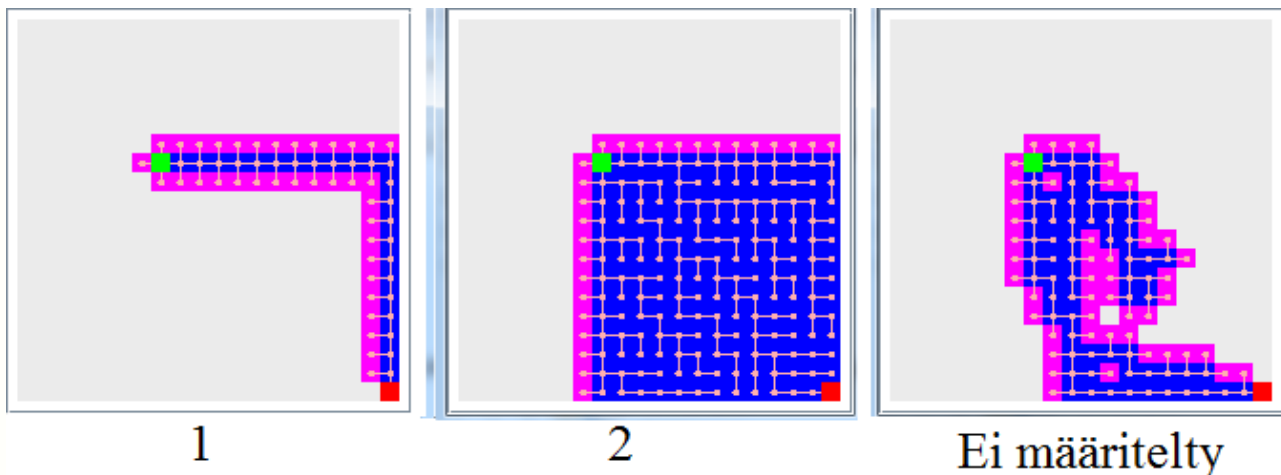


- mitä opin tällä viikolla / tänään?

- Java priority queue ei tue decrease key
 - Alkio pitää poistaa ja lisätä takaisin avaimen muuttamisen jälkeen
- Opiskelin netistä miten Fibonacci-keko toimii.
- A*: saman potentiaalin omaavat solmut voidaan järjestää kahdella tavalla
 1. Maalia lähempänä olevat solmut saavat isomman prioriteetin. (Käytän tätä nykyään)
 2. Lähtöä lähempänä olevat solmut saavat isomman prioriteetin.
- Jos näin ei tehdä niin prioriteettijonon toteutus määrittelee järjestyksen. (Käytin tätä ennen)
 - Kaikki kolme toimii mutta vaihtoehto yksi vaikuttaa keskimäärin tehokkaammalta



- mikä jäi epäselväksi? Vastaa tähän kohtaan rehellisesti, koska tarjoan sinulle tarvittaessa apua tämän kohdan perusteella.

- miten ohjelma on edistynyt?

Algoritmit alkavat olla valmiit ja tietorakenteiden tekeminen aloitettu. Bugeja korjattu.

- mitä teen seuraavaksi?

- Fibonacci-keon toteutus
- Binäärikeon toteutus
- Vertailu
 - Aika (millisekunnit/nanosekunnit)

- A* binäärikeolla
 - A* fibonacci-keolla
 - Dijkstra binäärikeolla
 - Dijkstra fibonacci-keolla
 - Tutkittujen solmujen määrä (kpl)
 - A*
 - Dijkstra
-
- (Lisää ominaisuuksia graafiseen käyttöliittymään)
 - Etene askel napista
 - Hyppää loppuun napista
 - Suorita loppuun napista
 - Aloitus- ja lopetusruudun asettaminen