

Viikkoraportti 3

Aloitin hommat lauantaina 13.8. klo 17 ja jatkoin niitä klo 24 asti. Sunnuntaina jatkoin klo 21-23. Viime viikolla Dijkstran toteutus jäi kesken. Tällä viikolla sain Dijkstran valmiiksi. Muuttelin myös ohjelman rakennetta pariin otteeseen: muutin Keko-luokkaa niin, että solmujen painot määrittävät solmun paikan keossa. Toisin sanoen uuden solmun lisäys sekä heapDecKey vertailevat solmun painoa muiden solmujen painoihin. Vanhemmat ja lapset löytyvät solmun järjestysnumeron avulla, joka muuttuu aina kun solmun sijainti keossakin muuttuu. Loin myös algoritmiluokilleni yläluokan Algoritmi. Algoritmi sisältää sekä Dijkstrassa että Bellman-Fordissa käytettävät apumetodit. Luokkahierarkia noin karkealla tasolla kaipaa vielä fiksausta, koska ensinnäkään muutaman apumetodin sisältävä luokka (huolimatta siitä, että siellä on myös Bellmanin ja Dijkstran yhteiset luokkamuuttujat) ei ansaitse nimeä Algoritmi eikä toisaalta ansaitse sellaisenaan yläluokan asemaa. Tuota pitää vielä fiksata myös.

Algoritmeissa käytetty polku- sekä etäisyystaulukko poistettu ja ne on korvattu luokkamuuttujilla. Jokainen solmu sisältää luokkamuuttujana yhden solmun, nimikkeellä path, joka ajaa polkutaulukon asian. Naapurisolmuihin etäisyys on aina 1, sillä ohjelma merkitsee naapurin kohdalle null, mikäli naapurina on seinä. Solmujen parametri paino ajaa distance-aulukon asiaa, myönnettäköön että nimi on huono, sillä se ei puhu kaaren painosta vaan solmun etäisyydestä alkusolmuun. Etäisyystaulukon poisto helpottaa myös keon ja dijkstran yhdistämistä, sillä se vaikuttaa olevan helpoin tapa saada keon ja dijkstran metodit puhumaan samaa kieltä. Lisäksi toteutin Bellman-Fordin ja päivitin lopuksi vielä javadocin.

Toisen testin sain myös toimimaan. Se oli niinkin helppoa että taulukko piti vain testimetodin alussa ensin luoda, ennenkuin sitä voi testata. Nyt on siis kaksi testiä, jotka testaavat labyrinthitaulukon luontia ja naapurisolmujen hakua.

Ohjelma aiheuttaa kummallakin algoritmilla jostain syystä nullpointterin ja syyksi aiheutuu metodin lyhinPolku() rivi 80 Algoritmiluokasta, siis tuo while-ehto. Debuggauksen perusteella solmu u eli maalisolmun path on ilmeisesti null. Koska en ole toteuttanut equalssia solmujen vertailuun, suoritan sen vertailemalla solmujen x- ja y-koordinaatteja. Ohjelma suoritettiin monta kertaa jolloin labyrinthi on aina erilainen, mutta virhe silti sama. Ei voi siis olla kiinni pääsystä lähtösolmusta maalisolmuun. Tarkoitus olisi tulla ensi viikolla näyttämään kyseistä ohjelmaa koululle. Samalla tarvitsen apua Bellmanin toteutukseen: Olenko ymmärtänyt oikein että käyn jokaisen taulukon solmun läpi järjestään ja jokaisen kohdalla relaxaan kaikki kaaret kyseisestä solmusta sen vierussolmuihin? Tarvitseeko lopuksi käytävää tarkistusta tehdä, sillä painot ovat positiivisia?

Ensi viikolla yritän saada ohjelmaa toimimaan, toteutus alkaa toivottavasti olla kasassa. Myös testejä voi alkaa tehdä sekä jonkin sortin käyttöliittymää. Turhaa on silti ahnehtia liikaa.