

Viikko 1

Yleinen

Ensimmäinen viikko on parhaimmillaankin ollut puolikas; aihe valittiin keskiviikkona ja joulukiireet söivät pois suuren osan tehokkaasta työajasta. Lopustakin iso osa meni Gitin ja Netbeansin käyttöä uudelleen opetellessa, sekä ihan perus-java -koodaamista mieleen palautellessa. Olkoonkin miten pyörällä ajon kaltaista, jos ei vuoteen ole juurikaan töissä tai kouluhommissa päässyt toteuttamaan, niin hetken kestää ennen kuin pääsee vauhtiin.

Jos jotain jouluksi toivoisin, niin sitä että Gittiä käytettäessä voisi ohittaa muutaman välivaiheen ja että Visual Studiolla voisi koodata javaa. NetBeansista löytyvät kyllä kaikki työkalut, mutta jotenkin se on asteen pari VS:ää kankeampi käyttää.

Mitä tein tällä viikolla

Varsinaisen toteutuksen kannalta viikko meni A*-teoriaan tutustumisessa sekä perustoteutuksen luomisessa. Lopputuloksena toteutus, joka luo annetun kokoisen (tyhjän) kartan ja osaa laskea reitin kahden pisteen välillä kulkien x- ja y-akseleita pitkin. Kartassa on mahdollisuus asettaa noodeja läpikulkemattomiksi, mutta käytössä ominaisuus ei vielä ole.

Ohjelma on käytettävissä vain yksikkötestien kautta, ja varmaan on vielä hetken aikaa. Täytyy vähän miettiä kartan syöttämistä ja sen vaatimuksia, ja niiden kautta mahdollista käyttöliittymää. Yksikkötestejä on vielä vähän ja ne lähinnä testaavat reitinhakua korkealla ja karkealla tasolla.

Mikä jäi epäselväksi

Annetun noodin viereisten noodien läpikäynti ja reitin kustannusten kohdalla tuli kiire ja piti etsiä netistä esimerkkikoodia. Näyttäisi toimivan ja teoria on aika selkeä, mutta toteutuksen kokonaiskuva on vielä vähän hämärä. Oletan että se selkenee jähka ehtii tehdä muutaman monimutkaisemman yksikkötestin ja debugata ne lävitse. Mitään ylitsepääsemättömiä ongelmia tai kokonaan hämärän peitossa olevia juttuja ei ole vielä tullut vastaan, koska toteutus on vielä hyvin yksinkertainen.

Aika- ja tilavaativuuksia en ehtinyt vielä katsomaan sillä tasolla että niitä voisi dokumentoida määrittelydokumenttiin.

Mitä teen seuraavaksi

Reitinhaku pelkillä x- ja y-akseleilla ei tunnu kauhean mielekkäältä, mutta sen toteutus on nyt suhteellisen hyvin hallussa, vaikka apua piti googlata parikin kertaa. Seuraavaksi lienee syytä monimutkaistaa hakua testaamalla esteiden käyttöä, ja harkita neljän liikkumissuunnan asemesta kahdeksan käyttöä. Jos aikaa jää, voinee hifistellä eri liikkumiskustannuksilla, mutta todennäköisesti tätä ennen on syytä ottaa käyttöön LinkedListien asemesta ensin ehdotettu PriorityQueue ja sitten tehdä oma kekototeutus. Png:stä generoidun kartan käyttöönoton priorisoin jonnekin oman kekototeutuksen jälkeiselle ajalle.

Koodia pitäisi abstrahoida, alustavasti varmaan niin että erottaa reitinlaskennan logiikan omaksi luokakseen jotta se on korvattavissa.

Viikko 2 ja 3

Yleinen

Joulu- ja uusivuosi söivät jonkin verran aikaa, mutta yllättävän hyvinhän tämä on edennyt. Pakko tiristää työ mahdollisimman valmiiksi toiseksi viimeiseen palautukseen, koska viimeisellä viikolla pitää jättää aikaa toisen kurssin esseille.

Mitä tein tällä viikolla

Toteutettu ensin minimikeko Integer-arvoilla ja laajennettu se sitten käsittelemään kartan solmuja (Noodi). Lisäksi kekon piti tehdä mm. metodi jolla tutkitaan löytyykö annettu olio jo keosta sekä keon sisäisen tietorakenteen kasvatus keon koon kasvaessa yli sille alustetun maksimikoon.

Karttatoteutukseen lisätty mahdollisuus alustaa kartta konstruktorin parametrinä saadulla tiedoston nimellä. Mitään varsinaisia tarkastuksia tälle ei ole toteutettu, vaan aika monella oletuksella mennään tässä vaiheessa.

Yksikkötestejä järkevöitetty ja suorituskykytestausta aloiteltu (ks. testausdokumentti).

Mikä jäi epäselväksi

Epäselväksi jäi seuraavanlainen teoreettinen kysymys: voidaanko minimipuusta poistaa jokin muu arvo kuin pienin arvo ja suorittaa sen jälkeen heapify niin että minimikeon ehto toteutuu? Todennäköisesti voidaan, mutta sen toteuttaminen ei onnistune normaalilla minimikeon toteutuksella, vaan pitäisi esim. rakentaa koko keko uudelleen jäljelle jääneillä arvoilla.

Kysymys jäi lähinnä mieleen kun PriorityQueue.remove(T) noin nopeasti vilkaistuna poistaa PQ-rakenteesta annetun olion, kun taas minimikeossa voidaan poistaa vain pienin alkio. Tässä

tapauksessa PQ:n kutsu voidaan korvata minimikeon poisto-operaatiolla, koska käsiteltäväksi alkioiksi on aiemmin haettu minimikeon pienin alkio.

Noin muuten avoimet kysymykset liikkuvat aika pitkälle ihan algoritmiin liittymättömässä perustoteutuksessa kuten erilaisissa tarkistuksissa ja virhetilanteista selviämisessä, joihin löytäneet vastaukset lähinnä tekemällä.

Mitä teen seuraavaksi

Lienee korkea aika tehdä ohjelmasta sellainen että sitä voi ajaa komentoriviltä käsin. Runkototeutus on nähdäkseni lähes valmis (kaikenlaisia tarkistuksia on toki syytä lisätä ajan salliessa, mutta ne eivät liene harjoitustyön tärkein asia). Manuaalinen testaus on kokonaan suorittamatta, yksittäistestausta pitää vielä tarkistella kriittisesti, sekä tehdä lisää automatisoituja integraatiotestejä.