

**1. Mitä opin tällä viikolla?**

Parin viime viikon aikana olen päässyt palauttamaan hyvin mieleeni ohjelmoinnin ja tietorakenteiden perusasioita, eikä A\*-algoritmin tai minimikeon toteutus lopulta ollut ihan niin ylitsepääsemätön urakka kuin miltä se aluksi tuntui, ainakaan nykyisellä tasolla ja tehokkuudella toteutettuna (varmasti molemmista löytyisi kyllä vielä rutkasti optimoitavaa). Sekä A\*:n että minimikeon toteutukseen minulla on toisaalta ollut hyvät pseudokoodit taustalla, mitkä ovat helpottaneet etenemistä.

Yksi suuri opetus tällä viikolla on ollut versionhallinnan hyödyllisyys - on kovin helppoa rikkoo jokin kohta jo kerran kuntoon saadusta koodista, ja jos vanhoja ohjelmaversioita ei olisi missään tallessa, virheen selvittämiseen olisi varmasti monessa kohtaa mennyt eksponentiaalisesti pidempi aika. Opin lisäksi niinkin yksinkertaisen asian kuin kellon käynnistämisen, joskin tehokkaampaa ja oikeellisempaa suorituskyyttestaustakin on varmasti mahdollista NetBeansissa tehdä. On ollut myös hyvä palautella mieleen esimerkiksi keko-tietorakennetta, joka oli jo hieman päässyt TiRan jälkeen unohtumaan.

**2. Mikä jäi epäselväksi**

Suorituskyyttestaus on minulle uusi asia, ja tämän(kin) tiimoilta tulen varmasti vierailemaan pajassa ensi viikolla. Nyt tein suorituskyyttestausta yksinkertaisesti vain käynnistämällä ja pysäyttämällä kellon samalla kun suoritan reittihaun labyrintissa pääohjelman kautta, mutta uskoisin, että testaukseen on validimpiakin menetelmiä. Tällä hetkellä näyttää siltä, että oma kekototeutukseni jää vielä jokseenkin reippaasti jälkeen Javan PriorityQueueesta, enkä ole ihan varma, mistä hitaus johtuu.

Aloin myös tutustua tänään Jump Point Search -algoritmiin sekä binomikekoon, sillä tavoitteeni on, että ehtisin laajentaa työtäni vielä (esimerkiksi) näistä toisen toteutuksella. Varsinkin Jump Point Searchin käytännön toteutus jäi vielä toistaiseksi itselleni vähän hämäräksi, mutta algoritmi vaikutti kyllä kiinnostavalta ja tehokkaalta.

**3. Miten ohjelma on edistynyt**

Sellaisina päivinä kun olen ehtinyt istua useamman tunnin työn ääressä, olen huomannut työn edistyvän yleensä melko tehokkaasti, kun taas päivinä, jolloin aikaa on ollut käytettävissä harjoitustyöhön vain pari tuntia, eteneminen on ollut hitaampaa ja pienenkin virheen ratkaiseminen on saattanut kestää pitkään. Jouluviikollakin ohjelmointiin jäi lopulta vähemmän aikaa, mitä olin ajatellut. Siihen nähden, että välillä eteneminen on ollut hitaampaa ja välissä on ollut päiviä että en ole ehtinyt koodaamaan yhtään, olen mielestäni kirjinyt työn tekemisessä ja etenemisessä tehokkaina päivinä ihan hyvin, ainakin varsinaisen ohjelmakoodin osalta.

**4. Mitä teen seuraavaksi?**

Ensi viikon aikana työ pitäisi saada valmiiksi, joten tehtävää kyllä riittää. Tavoitteeni on, että työ olisi sitä myöten valmiina perjantaina, että minun tarvitsisi enää ottaa huomioon mahdolliset demossa esiin tulevat kommentit ennen loppupalautusta.

Oman kekototeutuksen suorituskyky jää vielä jälkeen Javan PriorityQueueesta, joten katson seuraavaksi, pystyisikö suorituskykyä vielä jotenkin hieman parantamaan. Lisäksi tavoitteenani on vielä ehtiä toteuttaa vaihtoehtoinen etsintäalgoritmi tai kekototeutus ja päästä vertaamaan sitä tämänhetkiseen toteutukseen. Luonnollisesti tarkempi dokumentaatio ja aikavaativuusanalyysit (+visualisointi) pitää vielä myös saada valmiiksi.

Tarkoitus on myös, että Labyrintti-luokasta pystyisi generoimaan satunnaisesti erikokoisia labyrintteja (eli ei niin, että valittavana on vain muutama valmiiksi syötetty labyrintti). Tämän toteutus on edelleen työn alla.