

Viikko 3.

30.1. Perjantai – 2.2. Maanantai

Koska olin nyt saanut ohjelmani ytimen toteutettua, aloitin implementoimaan seuraavaa vaihetta. Meinasin aluksi tehdä JPS eli Jump Point Searchin, mutta päädyin loppujen lopuksi tekemään liikkuvan maalin osaksi siksi, koska minulla ei ollut mitään kokemusta JPS:n implementoimisesta ja sen tutkimiseen ja toteuttamiseen olisi voinut kulua paljon aikaa. Luulin, että liikkuva maali olisi siis järkevämpi ja hieman selkeämpi minulle ja siksi otin sen. Myös pelien niin sanottu dynaaminen reitinhaku aiheena on kiinnostavaa, koska pelien monimutkaistuessa ja tehon vaatimusten kasvaessa optimoinnit ovat tärkeässä asemassa ja tästä syystä aluksi mielessäni olikin JPS, joka on huomattavasti A*-algoritmia tehokkaampi, koska sen ei tarvitse käydä kuin murto-osa A*:iin vaadittavista solmuista.

Viikonloppu koodauksen parissa ei sujunut kuitenkaan hyvin. Sain toteutettua esi-asteen liikkuvalla maalille niin, että maalin paikkaa pystyi vaihtamaan aina ajojen välissä. Tämä ei kuitenkaan ollut alkuperäinen suunnitelmani, vaan halusin dynaamisen reitinhaun. Ongelmat kuitenkin alkoivat heti, kun aloin implementoimaan suunnitelmaani. Minulla ei ollut nimittäin kauheasti kokemusta Java Swingin toimintatavoista, joten pitkän koodausurakan jälkeen olin tilanteessa, jossa algoritmini kyllä toimi, mutta kun ohjelma piirsi, niin se lukitsi muun ohjelman käytön. Toisin sanoen siis piirto-operaatio jostain syystä lukitsi kaiken muun GUI:n käytön enkä voinut kuunnella esimerkiksi klikkauksia laisinkaan.

Päätin toteuttaa Piste-luokan selkeyttämään koordinaattien toiminnallisuutta lopuksi.

3.2. Tiistai – 5.2. Torstai

Olin jo hieman epätoivoinen, koska tuntui että kulutan kauheasti aikaa ongelmaan, joka ei liittynyt suoraan algoritmiini laisinkaan. Onneksi sain kuitenkin apua ongelmaan ja ratkesi, että Java Swingin pääsäie oli sellainen, jossa ei saanut ajaa raskaita ja pitkiä ohjelmia. Ratkaisu ongelmaan oli siis tehdä erillinen SwingWorker-luokan ilmentymä, joka pystyi hoitamaan piirtämisen erillään pääsäikeestä. Nyt sain toteutettua kuuntelijan niin, että piirron pystyi keskeyttämään klikkaamalla, joka oli erittäin hyvä asia.

Tein myös samalla ArrayList-tietorakenteelleni testit ja laitoin sen toimimaan sillä tasolla, jolla luulen, että itse tulen käyttämään sitä. Eli tarpeeksi tehokas, mutta kaikki metodeja ei minun todennäköisesti tarvitse luoda, koska nille ei tule olemaan tarvetta.

Lopuksi tein niin, että hiiren klikkauksilla piirto lopetettiin ja algoritmi alustettiin uudelleen jatkaen kohdasta, jossa viimeinen solmu analysoitiin.

3. Viikko kokonaisuudessaan

Olin hieman pettynyt viikkoon ja siihen kuinka paljon aikaa kulutin ongelman kanssa. Kuitenkin kaikki toimii nyt miten pitää ja ehdin tutustua materiaaleihin mm. D*:sta ja dynaamisesta reitinhausta. Ensi viikolla pääsen toteuttamaan liikkuvaa maalia urakalla. Minun täytyy vain ensiksi yhdistää A*-algoritmin ja piirron välistä suhdetta, koska nyt toiminnallisuus menee niin, että ensin A* ajetaan taustalla ja sitten sen palauttama polku piirretään. Jotta dynaamisuutta saataisiin olisi piirron hyvä tapahtua melkein A* toimesta, jolloin kaikki tapahtuisi reaaliajassa.