Viikko 2.

23.1. Perjantai – 25.1. Sunnuntai

Aloitin perjantaina työstämään käyttöliittymää projektiini, koska projektiini kuului olennaisena osana, että siihen pystyi lataamaan kuvan analysoitavaksi. Tämän kuvan avulla luodaan kartta A*-algoritmille, joka sitten suorittaa reitinhaun perustuen piirrettyyn kuvaan. Sain kaikki toimimaan ja piirrettyä kuvan sekä siihen liittyvän lyhimmän polun. Lisäsin myös visuaalisena analysoidut solmut, joten algoritmin ajon aikana näkee kätevästi, miten algoritmi tuli tulokseen, että kyseinen polku on lyhin mahdollinen.

Omien testieni nojalla tulin siihen tulokseen, että parhaimmat kartat olivat .bmp-muotoisia ja noin 50 x 50 – 500 x 500 kokoisia kuvia ja joissa sininen väri esittää vettä, musta esteitä, punainen maalia ja vihreä lähtöä. Koin testatessani kuvia huomattavia ongelmia tehokkuuden kanssa, koska HD-tasoisessa kuvassa on esimerkiksi 1920 x 1050 pikseliä, joka on 2 miljoonaa pikseliä per kuva. Tämän laajuinen kuva saa A*:ni kyykkyyn, koska se analysoi jokaisen pikselin joka kuvasta löytyy verkon luomisessa. Kuitenkin algoritmini pystyy tehtävään n. 10-20 kertaa nopeammin kuin Dijkstra, joka on huomattava parannus.

26.1. Maanantai

Jatkoin graafista käyttöliittymääni lisäämällä siihen ominaisuuden lisätä klikkaamalla uusia esteitä. Näin saan kartasta hieman dynaamisemman ja jos koen paljon ongelmia liikkuvan maalin kanssa, meinaan toteuttaa todennäköisesti dynaamisen reitinhaun projektiini. Se olisi nykyisellä käyttöliittymällä helppoa tehdä ja näyttäisi mielestäni todella hyvältä ja visualisoisi A*-algoritmin toimintaa enemmän.

Aloitin myös samalla omien tietorakenteiden toteuttamisen siltä varalta, että projektistani tulisi liian suppea. Keskeiset tietorakenteet, joita projektini käyttää ovat jono, minimikeko sekä lista. Jonoa käytetään pääasiassa polun tallentamiseen, minimikeon avulla taas A* toimii ahneesti ja lista on tällä hetkellä vain auttamassa muutamassa metodissa, jossa laskentatehoa ei tarvita kauheasti.

27.1. Tiistai

Aloitin liikkuvan maalin toteuttamista, mutta koin suuria ongelmia tehokkuuden kanssa. En oikein tiennyt miten saan maalin liikkumaan tehokkaasti samalla päivittäen A*-algoritmiin tarvittavaa karttaa solmuista ja niiden naapureista. Ongelman pääsyynä on sekä piirron visualisointi jollain tavalla sekä liikkuvan maalin liikkumisen simulointi tehokkaasti.

28.1. Keskiviikko- 29.1. Torstai

Huomasin, että algoritmini heurestiikan laskemisessa oli pieni ongelma ja päätin lisätä myös algoritmiini mahdollisuuden kulkea myös viistosti. Nyt ohjelmani löytää lyhyimmän polun jopa entistä nopeammin, mutta karttaan se toi pienen rajoituksen; kulman on oltava aina vähintään L:n muotoinen, tai muuten viistossa oleva voidaan vahingossa tulkita kuljettavaksi solmuksi vaikka välissä olisikin seinä.

2. Viikko kokonaisuudessaan

Sain viikolla mielestäni ihan kohtuullisesti aikaan ainakin käyttöliittymän nojalla. Valitettavasti itse algoritmin toteuttaminen parannuksista huolimatta jäi vähälle ja olen nyt hieman kahden vaiheilla,

että toteutanko rinnalle paljon puhutun JPS:n (Jump Point Search) sillä se on vieläkin tehokkaampi kuin A* vai toteutanko dynaamisen reitinhaun vai sitten liikkuvan maalin. Näissä kaikissa on kuitenkin omat ongelmansa ja aika on rajallinen, vaikka tosiasiassa haluaisin toteuttaa nämä kaikki ominaisuudet.

Alla olevassa kuvassa ohjelmani, kun kartta on ladattu ja siihen on ajettu A*-algoritmi. Punainen tarkoittaa analysoitua solmua ja sininen optimaalista polkua maaliin.