Toteutusdokumentti

Ohjelmassa on toteutettuna A*-algoritmi ja Dijkstran algoritmi sekä tietorakenteena minimikeko, joka toimii prioriteettijonona algoritmien toteutuksessa.

Ohjelman interaktiota käyttäjän kanssa ohjaa luokka UI, joka toimii yksinkertaisen tekstipohjaisen käyttöliittymän tavoin. Luokan avulla luetaan uusi kartta annetusta tiedostosta ja luodaan kartta polunhakua varten TileMap-luokan avulla.

Polunhakualgoritmiä varten ohjelmassa on toteutettu luokka TileMap, joka edustaa yksittäistä karttaa, johon on asetettu lähtö- ja maalipisteet ja jossa on eripainoisia pisteitä. TileMap saa kartan pisteiden arvot UI-luokan lukemasta tiedostosta ja tallentaa ne kaksiulotteiseen taulukkoon Pathfinder-luokan reitinhakualgoritmeja varten.

Polunhaku suoritetaan luokassa Pathfinder. Kun luokan olio otetaan käyttöön, päätetään halutaanko käyttää A*-algoritmia vai Dijkstran algoritmia mode-muuttujan avulla. Algoritmit on toteutettu minimikeon avulla, joka on toteutettu luokassa MinHeap. Luokka ei käytä erikseen luotua verkkoa, vaan luo solmuja Node-luokan avulla ja asettaa niitä taulukkoon sitä mukaa, kun kartassa edetään.

Kun reitti lähdön ja maalin välillä on löydetty, saadaan kuljettu reitti seuraamalla Node-luokasta luotujen solmujen vanhempia, jotka on asetettu aina kun uuteen solmuun on menty tai solmuun on löydetty lyhyempi reitti.

MinHeap-luokassa on toteutettuna A*-algoritmin ja Dijkstran algoritmin toiminnassa tarvittava minimikeko. Minimikeon aikavaativuus on O(log n) uusia solmuja lisättäessä ja poistettaessa pienin alkio keosta, mutta käytössä oleva metodi satunnaisen solmun poistamiselle poistaa aikavaativuudella O(n), koska sen täytyy ensin etsiä poistettava solmu.

Lähteet:

http://theory.stanford.edu/~amitp/GameProgramming/

https://en.wikipedia.org/wiki/A*_search_algorithm

https://en.wikipedia.org/wiki/Heap_(data_structure)