Määrittelydokumentti

Toteutin ohjelmoinnin harjoitustyönä luolastoseikkailupelin, jonka vihollisten polunetsintä perustui lähinnä pelaajan takaa-ajamiseen ja satunnaisuuteen. Siispä päätin toteuttaa tira-harjoitustyönä erilaisia pathfinding-algoritmeja, joista parhaat voin käyttää pelin jatkokehitykseen.

Toteutustapa

- Lähtökohtaisesti algoritmin tulee löytää reitti ruuduista koostuvan sokkelon läpi, pisteestä A pisteeseen B.

Testaan kutakin algoritmia myös muissa tilanteessa:

- Kohderuutu liikkuu
    - Tällä simuloidaan pelaajan hahmoa takaa-ajavaa vihollista: pelaaja liikkuu tätä pakoon
    - Algoritmi joutuu koko ajan sopeutumaan uuteen tilanteeseen
- Muuttuva sokkelo
    - Tällä simuloidaan esimerkiksi porttien ja siltojen avautumista pelimaailmassa
    - Algoritmi joutuu tarkistamaan, onko vanha reitti yhä käyttökuntoinen
- Jää ja suo
    - Eli ruudut joissa hahmo liikkuu eri nopeudella/solmujen kaarilla on eri painot
    - Algoritmi siis suosii jäätä, jonka pinnalla liukuessa solmujen välisten kaarten paino on pienempi, suolla se taas on suurempi koska liike on tahmaisempaa.
- Yllämainitut yhtä aikaa

Sokkelot ladataan tekstitiedostosta, jossa määritellään seinien ja lattioiden lisäksi ovet ja eri lattiatyypit. Toisessa tiedostossa on tallennettuna maalipisteen alkusijainti liike, kolmannessa lähtöpiste. Sokkelot ovat suuria ja monimutkaisia.

Algoritmien testauksessa mitataan maaliin pääsyyn vaadittavien askelten määrä sekä reitin laskemiseen käytetty aika.

Tutkittavat algoritmit

Tutkin yleisimpiä polunetsintään käytettyjä algoritmeja:

- A*
- Dijkstran algoritmi
- Syvyyshaku
- Leveyshaku

Lähteet

- Yleisesti:
    - http://en.wikipedia.org/wiki/Pathfinding
- Sokkelogeneraattori:
    - http://www.delorie.com/game-room/mazes/genmaze.cgi
- A*:
    - http://www.policyalmanac.org/games/aStarTutorial.htm
    - http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html
    - http://web.mit.edu/eranki/www/tutorials/search/
    - http://rocketmandevelopment.com/blog/a-heuristics/
- Dijkstra:
    - http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm
- Syvyyshaku:
    - http://en.wikipedia.org/wiki/Depth-first_search
- Leveyshaku:
    - http://en.wikipedia.org/wiki/Breadth-first_search
- Keko:
    - http://opendatastructures.org/ods-java/10_1_BinaryHeap_Implicit_Bi.html
    - http://www.cse.hut.fi/en/research/SVG/TRAKLA2/tutorials/heap_tutorial/taulukkona.html
    - http://en.wikipedia.org/wiki/Binary_heap
- ArrayList:
    - http://grepcode.com/file/repository.grepcode.com/java/root/jdk/openjdk/6-b14/java/util/ArrayList.java
- HashSet:
    - http://stackoverflow.com/questions/4558754/define-what-is-a-hashset
- HashMap:
    - http://eclipsesource.com/blogs/2012/09/04/the-3-things-you-should-know-about-hashcode/
    - http://en.wikipedia.org/wiki/Hash_table

- http://codereview.stackexchange.com/questions/62049/simple-implementation-of-hashmap
- http://www.javamadesoeasy.com/2015/02/hashmap-custom-implementation.html