

Määrittelydokumentti

Toteutan luolaston generointialgoritmin Javalla. Luolastojen generointiin on useita erilaisia algoritmeja. Mielenkiintoisin löytämäni on kuitenkin tämä. Toteuttamani algoritmi perustuu seuraavaan linkkiin:

https://www.reddit.com/r/gamedev/comments/1dlwc4/procedural_dungeon_generation_algorithm_explained/

Algoritmi koostuu useammasta vaiheesta.

- 1) Valitaan generoitavien solujen määrä. Jos solu on tarpeeksi suuri, kutsutaan sitä huoneeksi. Solujen leveyden ja korkeuden määrittämiseen voidaan käyttää Park-Miller normaalijakaumaa. En ole vielä varma tulenko käyttämään sitä. Solut generoidaan tietyn säteen omaavan ympyrän sisään. Tässä vaiheessa solut ovat erittäin todennäköisesti päällekkäin. Tämä vaihe onnistuu lineaarisessa ajassa $O(n)$ solujen määrään nähden.
- 2) Solut ovat generoimisen jälkeen todennäköisesti päällekkäin. Nyt täytyy kehittää algoritmi, jolla huoneet saadaan irti toisistaan. Tähän ilmeisesti voidaan käyttää apuna "separation steering behaviouria". Tämän aikavaativuudesta en vielä osaa sanoa.
- 3) Nyt mahdollisia tyhjiä välejä täytetään 1×1 kokoisilla soluilla. Tämän tarpeesta ja aikavaativuudesta en vielä osaa sanoa.
- 4) Solut ovat erillään, ja tarpeeksi suuret solut ovat huoneita. Nyt huoneista tehdään verkko käyttämällä "delanuay triangulationia". Wikipedian mukaan tämä pitäisi onnistua ajassa $O(n \log n)$ huoneiden määrään nähden.
- 5) Sillä kaikkien huoneiden ei haluta olla yhteydessä toisiinsa, tehdään verkosta virittävä puu käyttäen Primin algoritmia. Primin algoritmin aikavaativuus on $O(E \log V)$, jossa E on kaarien määrä ja V on solmujen määrä.
- 6) Lopuksi lisätään L-muotoisia käytäviä menemään huoneista toiseen, jos huoneet eivät ole vierekkäin. Olettaisin, että tämä onnistuu lineaarisessa ajassa $O(n)$ huoneiden määrään nähden.

Aputietorakenteina tarvitsen ainakin ArrayListin kaltaista listarakennetta, taulukoita, verkkoja. Syötteiksi ohjelma saa solujen määrän ja ympyrän säteen, jonka sisälle solut aluksi generoidaan.

Muut lähteet:

https://en.wikipedia.org/wiki/Delaunay_triangulation

<https://www.cs.helsinki.fi/u/floreen/tira2013/tira.pdf>