# Toteutusdokumentti

**Pakkaukset**

- Pakkauksia ei ole, vain kansiorakenne.

**Kansiorakenne**

Kansion /src alla ovat kansiot

- bsp sisältämät luokat
  - BSPDungeon: Generoi luolaston hyödyntäen BSP-algoritmia.
  - Node: Solmu, joka sisältää tilan tiedot ja viitteen solmun vasempaan ja oikeaan lapseen.
- main
  - Main: Pyörittää vuorollaan sekä BSPDungeon:in että VoronoiDungeon:in. Tämän jälkeen suoritetaan Visualisation, josta voidaan tarkastella algoritmien aikaansaannoksia.
- visualisation
  - Visualisation: Piirtää molempien algoritmien tuotoksia näkyville valitun tason (level) mukaan.
- voronoi
  - Edge: Sisältää särmän tiedot
  - Event: Kuvastaa tapahtumaa, joka voi olla tontti- tai kehätapahtuma (site event & circle event).
  - Parabola: Sisältää tiedon paraabelista, sen lapsista ja vanhemmista.
  - Point: Yksittäisen pisteen tiedot.
  - Room: Luodun huoneen tiedot.
  - VoronoiDungeon: Generoi luolaston Voronoi-algoritmia hyödyntäen.

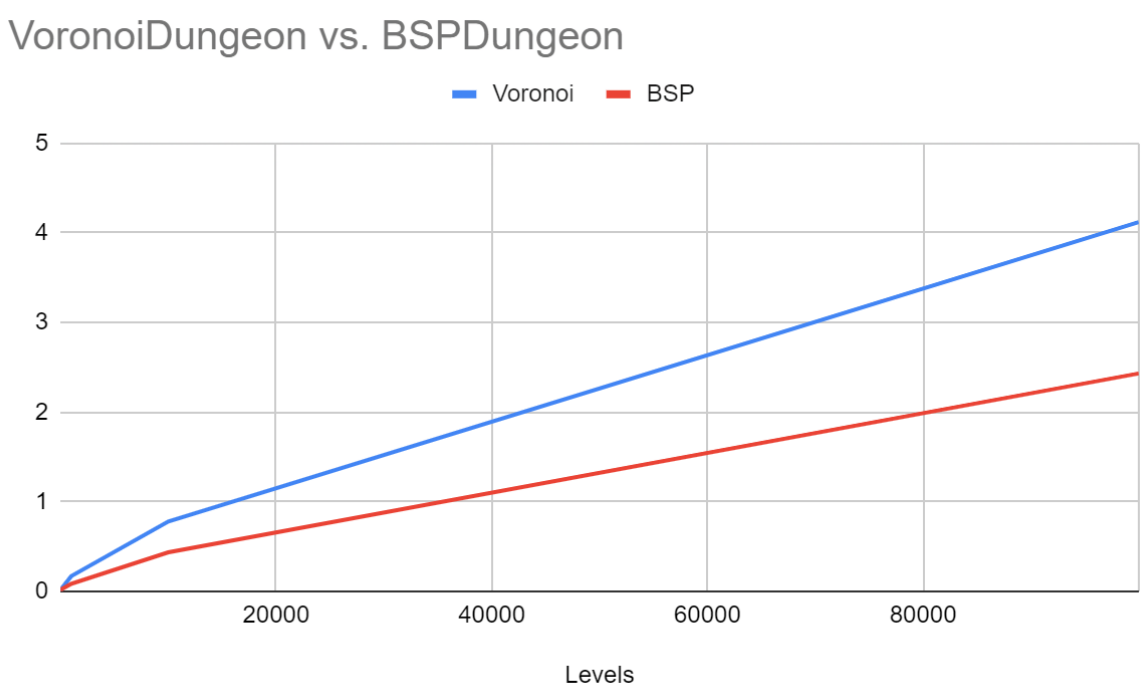
**Saavutetut aika- ja tilavuudet**

1. VoronoiDungeon
   - Tekemäni Voronoi-algoritmi perustuu Fortunen algoritmiin, jonka aikavaativuus on O(n log n) ja tilavaativuus on O(n). Todellisuudessa tämä ei toteutunut, sillä käyttämäni algoritmi vaati lisäkseen muita algoritmeja ja ratkaisuja, jotta saavutettaisiin haluttu luolasto. Todellisuudessa luolaston luomisen aikavaativuus on O(n^2) ja tilavaativuus O(n^2).
2. BSPDungeon
   - Tekemäni BSP-algoritmi luolaston luomisineen toteutuu aikavaativuudella O(n) ja tilavaativuudella O(n^2).

**Vertailu**

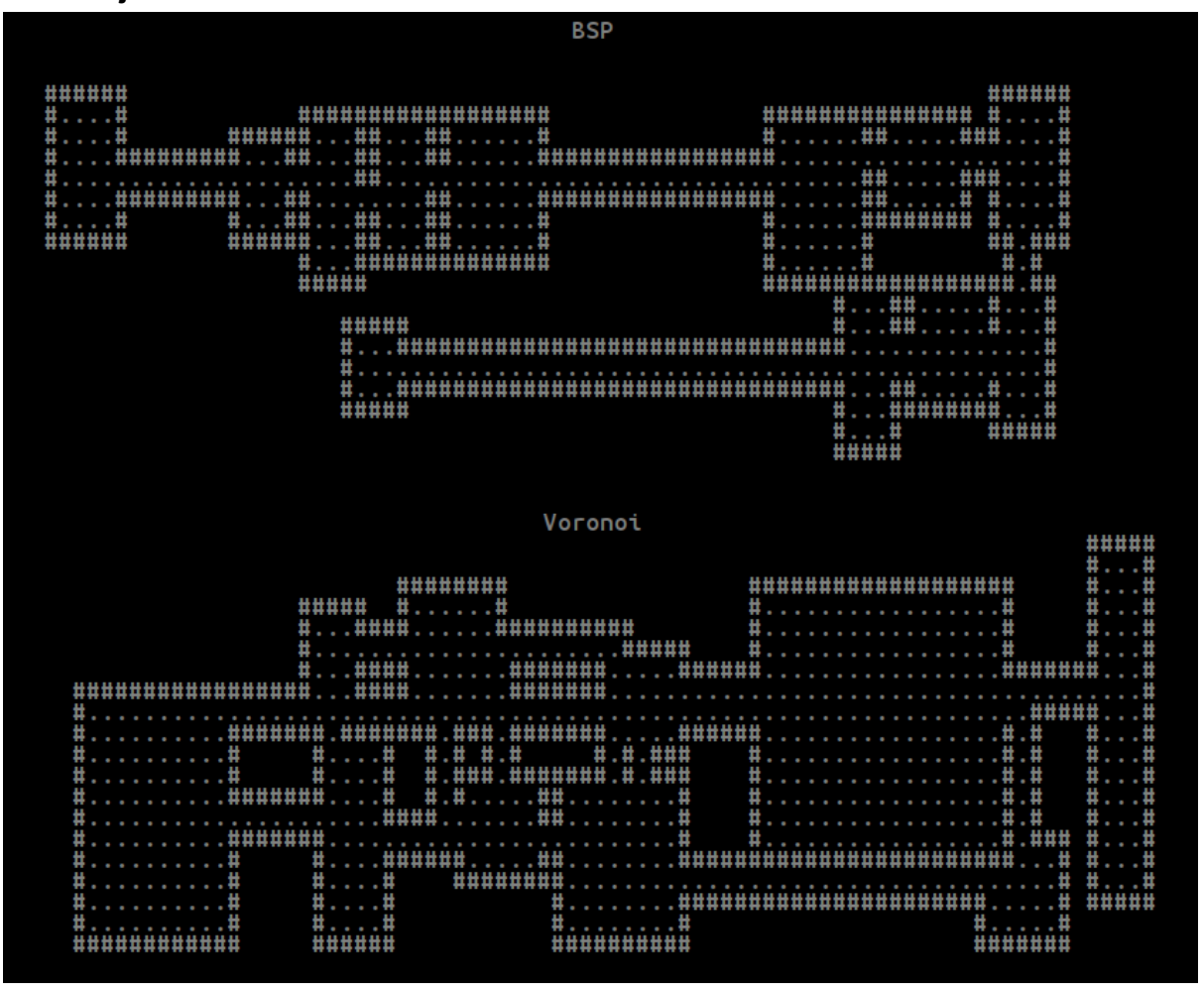
Ajoin luolasto-algoritmit eri määrillä luolastoja.

| Luotujen tasojen määrä | VoronoiDungeon aika sekunteina | BSPDungeon aika sekunteina |
|---|---|---|
| 1 | 0.003950762 | 0.001619672 |
| 10 | 0.009693405 | 0.006220246 |
| 100 | 0.032035706 | 0.023299234 |
| 1000 | 0.169022381 | 0.083794476 |
| 10000 | 0.778783845 | 0.435967088 |
| 100000 | 4.119794431 | 2.431233696 |




Voimme nähdä, että luolastojen luomisen ajallinen ero on huomattava BSP-algoritmin eduksi, mutta luotujen luolastojen määrä ei vaikuta kummankaan algoritmin suoritukseen muuten kuin lineaarisesti toteutuksellani.

**Luolastojen visuaaliset erot**




Kuten kuvasta voidaan huomata, lopputulos BSP- ja Voronoi-luolaston välillä on suhteellisen suuri. BSP-luolasto luo tyypillisesti hillittyjä, putkimaisia ja pienen oloisia luolastoja, toisin kuin Voronoi-luolasto, jonka luomat luolastot ovat kaaottisempia, avoimempia ja olemukseltaan massiivisempia luolastoja. Toki molempien luolastojen huoneiden luomis ehdoissa on viilaamisen varaa, mutta Voronoi vaikuttaa ainakin omaan silmääni luonnollisemmalta ja miellyttävämmältä. Se minkä näköistä sitten kukin haluaa käyttää on täysin riippuvaista siitä minkälaisen tunnelman ja paikan tuntua haluaa kuvastaa luolastollaan.

**Työn puutteet ja parannusehdotukset**

Puutteet ja parannukset kohdistuvat täysin toteuttamaani Voronoi-algoritmiin. Puutteellinen ymmärrykseni algoritmin toiminnasta ajoi minut tekemään laastariratkaisuja, jotta algoritmi pysyisi toimivana. Parannuksena olisi algoritmin muokkaaminen kokonaisvaltaisesti vastaamaan haluamaani luolaston luomista niin, että turhat askeleet voitaisiin poistaa välistä. Kenties luolasto luotaisiin Voronoin luomisen yhteydessä, eikä jälkikäteen kömpelösti luomisen tuotoksista.

**Lähteet**

https://en.wikipedia.org/wiki/Binary_space_partitioning
https://en.wikipedia.org/wiki/Fortune%27s_algorithm
https://gamedevelopment.tutsplus.com/tutorials/how-to-use-bsp-trees-to-generate-game-maps--gamedev-12268
https://eskerda.com/bsp-dungeon-generation/
https://github.com/serenaz/voronoi
https://stackoverflow.com/questions/982928/confused-with-voronoi-diagram-algorithm-fortunes-sweepline
https://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm
A Sweepline Algorithm for Voronoi Diagrams, Steven Fortune, Algorithmica, 2: 153-174, 1987
An Efficient Implementation of Fortune's Plane-Sweep Algorithm for Voronoi Diagrams, Kenny Wong, Department of Computer Science, University of Victoria, BC, V8W 3P6, Canada