

# Research maze algorithms

## Onderzoek.

Omdat ik moeite heb met algoritmes maken/implementeren wil ik graag vantevoren goed op orde hebben hoe ik dit ga aanpakken dus ik pak de volgende referentie punten:

- [https://en.wikipedia.org/wiki/Maze\\_generation\\_algorithm](https://en.wikipedia.org/wiki/Maze_generation_algorithm)
- <https://stackoverflow.com/questions/38502/whats-a-good-algorithm-to-generate-a-maze>

### Aldous broder algorithm.

Referentiepunt:

<https://weblog.jamisbuck.org/2011/1/17/maze-generation-aldous-broder-algorithm>

Het lijkt erop dat het relatief makkelijk te implementeren maar het gaat dan ook gepaard met extra performance kosten, dat is wel iets wat ik wil vermijden.

### Kruskal's algorithm.

Referentiepunt:

<http://weblog.jamisbuck.org/2011/1/3/maze-generation-kruskal-s-algorithm>

Als ik het test op de website zie ik hoe het een stuk sneller is dan het Aldous-Broder algoritme, omdat het algoritme verschillende punten tegelijkertijd pakt en vanuit daar werkt en Aldous-Broder pakt 1 punt en werkt vanaf daar. Maar volgens de wikipedia maakt het een hoop dead-ends aan.

### Prim's algorithm.

Referentiepunt:

<http://weblog.jamisbuck.org/2011/1/10/maze-generation-prim-s-algorithm>

Ziet er goed uit maar ik ben nog niet helemaal overtuigd omdat het net zoals Kruskal's algoritme "biased toward many short dead ends" en dat wil ik voorkomen.

### Wilson's algorithm.

Referentiepunt:

<https://weblog.jamisbuck.org/2011/1/20/maze-generation-wilson-s-algorithm>

Ik ben van mening dat deze maze het mooist eruit ziet van alle, enkel volgens het referentiepunt en wat ik zie is het heel traag; zo traag dat ik durf te zeggen dat er heel weinig mensen zijn die zo lang willen wachten voordat ze eindelijk kunnen spelen.

## Conclusie.

Voor dit project kies ik ervoor om **Kruskal's algoritme** te gebruiken om verschillende redenen, het lijkt erg op Prim's algoritme (volgens de referentiepunten) maar ik denk dat Kruskal sneller is dan Prim daarom kies ik Kruskal boven Prim.

Als ik performance niet zal meerekenen zou ik Wilson's algoritme gebruiken omdat die het meest "random" is en zo min mogelijk dead ends maakt, enkel ben ik totaal niet tevreden met de performance en is het dus in mijn ogen niet viable.