# What is the problem?

Ik ben op zoek naar een doolhof alogrithme die snel kan generen en makkelijk is om te programmeren. En heft het doolhof een natuurlijk gevoel van een doolhof.

# How did I solve this?

Ik heb gebruik gemaakt van de **Liturte study** om uit te zoeken **welke soorten doolhof generators** er zijn. De volgende critera’s komen er in voor het **makkelijk toepassen**, en het moet een natuurlijk **gevoel geven** van een doolhof.

Daarnaast heb ik ook gebruik gemaakt van de **Gap analysis** met een **Bench Mark test**. Om te uit te zoek welke het **snelste**.

What is the Result

I heb 2 website gezocht waarvan 1 zeker simulaties uitvoert plus het laat ook de code zien die ik nodig heb.  
[jamisbuck website](https://weblog.jamisbuck.org/2011/1/3/maze-generation-kruskal-s-algorithm) heeft de simulatie van elke doolhof algorithme die er is.

[Rozzeta Code](https://rosettacode.org/wiki/Maze_generation) heeft ook verschillende doolhof generators.  
allebij hebben ze code openbaar maar het is helaas wel in Python. Dus zou ik het zelf moeten vertalen.

# Validation

Hierin doe ik een korte bench mark en de Gap analysis verdeel het resultaat in colomen hoe snel ze zijn met generen van een doolhof. Alle doolhoven worden genereerd in een 16\*16 grid. Dus 256 tegels.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Doolhof algoritme | Snelheid | Overal toegang | Makkelijk te op te programeren | Maakt het doolhof natuurlijk gevoel |
| kruskal | 6 seconden | ja | gemiddeld | Nee maakt korte eindpunten maar heeft wel langer opeenvolgende stukken |
| prim | 5 sec | Ja | gemiddeld |  |
| Growing tree | 11 seconden | Ja | moeilijker | Meer bochten ook rechte stukken. Begin van generatie is random |
| sidewinder | 7 seconden | Ja | moeilijk | Nee een gedeelte blijft hetzelfde met ieder generatie |
| Binary tree | 6 seconden | Ja | Zeer moeilijk | Nee n gedeelte blijft hetzelfde met ieder generatie |
| Aldous-Broder | 1 min en 28 second | ja | Moeilijker op te lossen | Ieder doolhof is uniek omdat het startpunt random is. |

Door het zoeken van verschillende algoritmes heb ik gekozen voor de Growing Tree. Deze kwam als 4 de plaats uit de test maar, het is algoritme die meest gebruikt wordt en het makkelijkste is om te implementeren. Als hier in dit [Youtube](https://www.youtube.com/watch?v=OutlTTOm17M) filmpje wordt gebruikt.

# Next Steps

De volgende stappen is om Growing tree een prototype te maken in unity.