

Lab 1B Rapport

TEST OF PATHFINDING ALGORITHMS.

Benjamin Vesterlund | SOOO6D | 05/02-19

Problemspecifikation

Testa tre olika metoder för att hitta en path genom 3 olika 2d kartor där man kan röra sig mellan rutor vertikalt, horisontellt och diagonalt. Utan att gena genom hörn.

Tre kartor är utdelade av Lab handledare och är gjorda med fyra olika tecken för att skilja mellan unwalkable, walkable, spawn och goal.

```
|XXXXXXXXXXXXXXXXX
XS000000000000X
X0000000000XX00X
X00XXX00000XX00X
X00XXX00000000X
X00XXX00000000X
X00XXX000XXX000X
X00XXX000XXX000X
X00000000XXX000X
X00000000XXX000X
X000000000XX000X
X0000X00000000X
X000XXX0000000X
X00XXXXX000000X
X000000000000GX
XXXXXXXXXXXXXXXXX
```

Tre algoritmer som ska testas är A*, bredden och djupet först, och en egen algoritm (som inte behöver vara bra, bara snabb och hitta en giltig path)

Den slutgiltiga pathen ska visas grafiskt.

Användarhandledning

Projektet finns på <https://git.gscept.com/benves-7/door2d.git>

I Git projektet så finns det två kurser, både GameEngine och AI, under AI mappen finns två mappar, mappen "Lab 1B".

Startup är "PathTester.py", den kör och testar alla algoritmer och maps. Ritar ut pathen på mapsen.

Systembeskrivning

Har en fil **FileLoader.py** som laddar in en textfil och returnerar en lista på alla rader.

Har en fil **MapLoader.py** som innehåller en klass som skapar ett grid av Nodes i form av en array av listan från FileLoader.py, har även funktioner som hittar grannarna till den nuvarande noden. Innehåller även klassen Node som innehåller all data för olika punkter.

Har en fil **Window.py** som innehåller allt som behövs för att rita ut allt.

Har en fil **PathFinder.py** som innehåller alla sökfunktioner.

Lösningens begränsningar

Har fått en väldigt bra lösning där jag inte hittat några begränsningar, tid sökningen kan vara lite för dålig då vissa sökningar ger tid åtgång o.o s, vilket inte är jätte bra..

Diskussion

Intressant att DepthFirst är den snabbaste och att A* är den långsammaste sökningen (om jag inte gjort nått fel)

Min egna algoritm kör ett håll åt tills den stöter på en vägg, om den har testat alla riktningar så backar den tills den kommer till en nod som har kvar en riktning att utforska. Slutresultatet blir en väg till målet.. det går fort att hitta, men långt ifrån den bästa vägen.