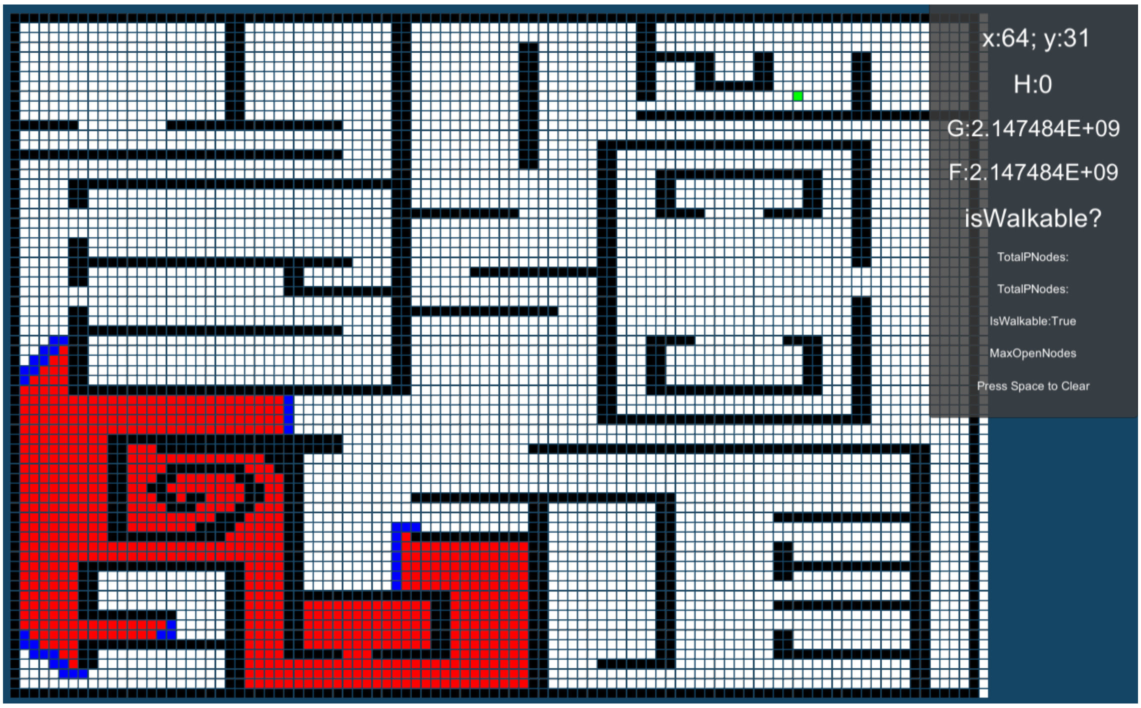
TÉCNICO LISBOA

Inteligência Artificial para Jogos

2024-2025

1st IAJ Project – Efficient Pathfinding in a Grid



x:64; y:31

H:0

G:2.147484E+09

F:2.147484E+09

isWalkable?

TotalPNodes:

TotalPNodes:

IsWalkable:True

MaxOpenNodes

Press Space to Clear

In this first practical project of the IAJ (or AIG) Course, we want you to create different levels of efficiency pathfinding algorithm for room based scenarios (where the traditional distance based heuristics fail to properly guide the search algorithm). Levels 1-3 will be implemented during Lectures and Labs (but of course you should start as soon as you want)

For this first project, you will need to write a report **with 4-6 pages**, explaining and justifying some of the decisions made by your group and a comparison between different algorithms (Level 4).

You should submit a zip file (named P1 Group XX, where XX is your group number) with both Unity source code and with the report (a pdf/doc file) via Fenix, **until 14:59 of September 22th**. I will accept submissions after the deadline, but with a 0.5 value penalty for each hour after the deadline. If the size of the Unity Project is too large for Fenix, please send an email with a download link to me, pedro.santos@tecnico.ulisboa.pt, with the subject “Project 1 Group XX”.

Bonus Levels: Bonus Levels will contribute to substituting your lowest test with a better grade, they will not affect the grade of the project

Level 1– Traditional A* (4 points)

- Implement the A* search algorithm with different options for data structures as described in the Lecture Guide.

Level 2– Node Array A* (4 points)

- Implement the Node Array A* search algorithm.

Level 3 – Gateway (5 points)

- Implement the Gateway pathfinding algorithm.

Level 4 – Comparing the pathfinding algorithms (5 points)

- Analyse the differences in performance between the original A* algorithm (w Euclidean distance), the original A* algorithm (w Euclidean distance and tie-breaking), the NodeArray A* algorithm (w Euclidean distance and the Gateway algorithm. This analysis should be in the report.
- Consider aspects such as fill, nodes visited, size of the open and closed list, total processing time and processing time per node (and other aspects if relevant). You can use the UI Manager and the canvas to capture some of these metrics.
    - Use the tables from the Lab 2 (but combine them).
    - Which version has the best performance and why? Make sure to include this analysis in the report.
    - In order to compare the different algorithms, make sure to use the giant grid (one with lots of nodes) and use equal paths (such as those available in Keys 1,2,3,4,5 (you can change these if you see fit). Use at least two different paths and indicate which ones you used.

Bonus Level – Additional Optimization.

- Implement a relevant efficiency optimization to your A* algorithm. Make sure to prove that it is an optimization and include it in the report.
    - Optimizations include creating a better heuristic, more efficient data structures, a new algorithm itself and code improvements (be careful, these must have a significant impact on performance) etc...

Report Quality (2 points)