10531 Video Game Technologies Proj. 3 - 07/05/2017

Path Finding

Supervisor: Abel Gomes *Scribe: A. Gomes*

In this project, the goal is to code a pathfinder in C++. Before going any further, please have a look at:

`https://qiao.github.io/PathFinding.js/visual/`

for a brief glance at a number of pathfinders at work.

1 Specific Learning Goals

After completing this project, students should know and be able:

1. To master pathfinding problems.
2. To deal with a pathfinder for 3D games .

2 Web Links

General stuff: graph traversal algorithms and codes:
`https://en.wikipedia.org/wiki/Graph_traversal`
`https://github.com/qiao/PathFinding.js`
`https://vlebb.leeds.ac.uk/bbcswebdav/orgs/SCH_Computing/FYProj/reports/0405/HallM.pdf`
`http://cirg.cs.up.ac.za/thesis/anguelov.pdf`
`https://project.dke.maastrichtuniversity.nl/games/files/msc/Schnieders_thesis.pdf`

A* algorithm
`https://en.wikipedia.org/wiki/A*_search_algorithm`
`http://theory.stanford.edu/~amitp/GameProgramming/`
`http://wiki.roblox.com/index.php?title=A*`
`http://code.activestate.com/recipes/577457-a-star-shortest-path-algorithm/`
`http://www.redblobgames.com/pathfinding/a-star/implementation.html#cpp-astar`
`https://github.com/justinhj/astar-algorithm-cpp/tree/master/cpp`
`http://www.geeksforgeeks.org/a-search-algorithm/`

Best-first search algorithm
`https://en.wikipedia.org/wiki/Best-first_search`
`http://www.stoimen.com/blog/2012/09/24/computer-algorithms-graph-best-first-search/`
`http://www.geeksforgeeks.org/best-first-search-informed-search/`
`https://www.daniweb.com/programming/software-development/threads/329811/best-first-algorithm`
`http://wiki.roblox.com/index.php?title=Best-first_search`
`https://www.reddit.com/r/readablecode/comments/19wmef/best_first_tree_search_c/`

Breadth-first search algorithm
`https://en.wikipedia.org/wiki/Breadth-first_search`
`https://www.tutorialspoint.com/data_structures_algorithms/breadth_first_traversal.htm`
`https://www.khanacademy.org/computing/computer-science/algorithms/breadth-first-search/`

a/the-breadth-first-search-algorithm
https://www.hackerearth.com/practice/algorithms/graphs/breadth-first-search/tutorial/
http://www.programming-techniques.com/2012/07/breadth-first-search-in-c-algorithm-and.html
http://www.redblobgames.com/pathfinding/a-star/implementation.html#cpp-breadth-first

Jump point search algorithm
https://en.wikipedia.org/wiki/Jump_point_search
www.nicta.com.au/pub-download/full/7771/
https://gamedevelopment.tutsplus.com/tutorials/how-to-speed-up-a-pathfinding-with-the-jump-point-search-algorithm--gamedev-5818
https://harablog.wordpress.com/2011/09/07/jump-point-search/
https://github.com/fgenesis/jps
https://www.gamedev.net/resources/_/technical/artificial-intelligence/jump-point-search-fast-a-pathfinding-for-uniform-cost-grids-r4220
http://kimwagner.de/jps-pathfinding-in-c/

Iterative deepening A* (IDA*) search algorithm
https://en.wikipedia.org/wiki/Iterative_deepening_A*
https://github.com/nihilus/idapathfinder
https://codereview.stackexchange.com/questions/5942/iterative-deepening-and-a
http://www.aaai.org/ocs/index.php/SOCS/SOCS12/paper/download/5404/5682
http://www.gamasutra.com/view/feature/131564/pawn_captures_wyvern_how_computer_.php?print=1
http://www.diva-portal.se/smash/get/diva2:949638/FULLTEXT02.pdf

3 Project Steps

This project consists of the following main steps:

1. To read a grid-based HOG map in memory; it is mandatory to use maps from "Dragon Age: Origins" and "Warcraft 3" available at `http://movingai.com/benchmarks/`.
2. To generate a graph from the HOG map.
3. To implement the pathfinder.
4. To display the HOG map with the path between two nodes (start and end nodes), as well as the visited nodes.

After completing your project, send it by email to `agomes@di.ubi.pt`. The .zip archive name should include the student number (e.g., `12345-project3TJV.zip`).

The deadline to deliver the project is 5 June (Monday, 23:59). **Projects not working or received after the stipulated deadline will not be evaluated.**

4 Algorithms to Students

Christian Lopes: Jump point search
Tiago Mendes: A*
José Manteigueiro: Breadth-first search

António da Silva: A*
Fábio Machado: IDA*
André Louro: Breadth-first search

Miguel Rodrigues: Best-first search
João Silva: IDA*
José Nunes: Jump point search