

Assignment 1 Phase 1

Qasim Abbas (qsa1), Cathy Pei (cwp33)

February 19, 2017

a) Interface

Interface: Unity Engine

map1.txt, map2.txt, map3.txt, map4.txt, map5.txt

5 maps

10 start/goal pairs each

visualize h, g, f for each cell (be able to select with mouse)

b) Algorithms

- uniform cost search
uninformed search; do not know where goal is
 $f(n) = g(n)$
fringe is a priority queue

Main ()

$g(s_{start}) := 0$

$fringe := \emptyset$

$visited := \emptyset$

$fringe.Insert(s_{start})$

while $fringe \neq \emptyset$

$s := fringe.Pop();$

if $s = s_{goal}$ **then**

return "path found";

$visited := visited \cup s;$

foreach $s' \in neighbors(s)$

if $s' \notin fringe \vee visited$

$fringe.Insert(s');$

else if $c(s, fringe.Contains(s')) > c(s, s')$

$updatecostof s' in fringe;$

$fringe.Remove(s');$

$fringe.Insert(s');$

- A*
- Weighted A*

c) Optimization

sorted list for visited;

won't have to search through whole list to see if a node has been checked

see if a goal exists

if not, then UCS

- Uniform Cost Search (UCS)
- A*
add a dynamic variable w for weighted A*
- Weighted A*

d) Heuristics

1. best conditional Euclidean/Manhattan?
find Euclidean and Manhattan,
compare,
if Manhattan not overestimation then use Manhattan
else use Euclidean
2. Euclidean
3. Manhattan inadmissible bc overestimation
4. Highways (try to use highways as much as possible)
5. No hard to traverse areas

e) Evaluation

- UCS
avg run time:
avg path length (vs. optimal length):
avg # nodes expanded:
memory requirements:
- A*
avg run time:
avg path length (vs. optimal length):
avg # nodes expanded:
memory requirements:
- Weighted A*
avg run time:
avg path length (vs. optimal length):
avg # nodes expanded:
memory requirements:

f) Results and Observations

Progress:
MapGen
unblocked cells
hard to traverse cells
working on highways
working on parser
GUI
works

Algorithm
need a priority queue
otherwise need weight for A*

Explain your results and discuss in detail your observations regarding the relative performance of the different methods. What impact do you perceive that different heuristic functions have on the behavior of the algorithms and why? What is the relative performance of the different algorithms and why?

Bibliography

GUI: quill18creates YouTube
PriorityQueue BlueRaja GitHub
UCS alg: <http://aima.cs.berkeley.edu/algorithms.pdf>