Index

Contenido

[Understanding Data Input 1](#_Toc178503116)

[Establishing Data Structure 1](#_Toc178503117)

[Functions 1](#_Toc178503118)

[Classes 1](#_Toc178503119)

[Scrum 2](#_Toc178503120)

Assume we have the following units:

Take this into account :

“Notes:

The order of the actions is determined by the destination state whose identifier is the lowest, that is, if different (partial) destinations can be reached at a given point (intersection), they will be visited in increasing numerical order”.

## Understanding Data Input

**address**: zone of Albacete we are in .

**distance**: idk

**initial**: initial state

**final**: goal state

**intersections**: a list of dictionaries with attributes **identifier**, **longitude** and **latitude**

**segments**: a list of dictionaries with attributes **origin**, **destination**, **distance** and **speed**

## Establishing Data Structure

### Functions

|  |  |  |
| --- | --- | --- |
| Function | Input | Output |
| search | (Problem, State) | return listOfActions |
| testGoal | (State) | return boolean |
| succesor | (Node) | return Node |

search: can be only one function that prompts the user for a number between 0 and 2 being

0: depth-first

1: breadth-first

2: random

I don’t really see the idea of search being an abstract class.

Procedures

|  |  |
| --- | --- |
| Procedures | Input |
| initializeOpen() | (initial) |

### Classes

|  |
| --- |
| State |
| -accumulatedCost: float |
| initializeOpen() |

|  |
| --- |
| Node |
| -state: State  -parent: pointer |
|  |

|  |
| --- |
| Search (abstract) |
|  |
|  |

|  |
| --- |
| Problem |
|  |
|  |

|  |
| --- |
| Action |
|  |
|  |

|  |
| --- |
| DepthFirst **inherits** Seach |
|  |
|  |

|  |
| --- |
| BreadthFirst **inherits** Seach |
|  |
|  |

## Scrum

|  |  |  |
| --- | --- | --- |
| Tasks | Agus | Jesús |
| -Implement DepthFirst  -Implement BreadthFirst  -Implement RandomSearch  -Define Problem  -Define Node  -Define State  -Define Action |  |  |