Flow fields

# Types of fields (3 types)

## Cost field

* Hold cost-to
* Used as input for building the flow field
* 8-bit field
* Possible value 0-255
  + 255 is wall
  + 1-254 traversable
* Minimum cost is 1

## Integration field

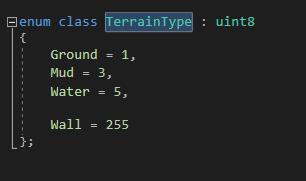
* 24-bit field
* First 16 bits -> Total integrated cost
* Second 8 bits -> used for integration flags
* Optionally spend more memory bot better flow results (eg: 32-bit float integrated cost)

## Flow field

* 8-bit fields
* First 4 bits -> index into direction lookup table
* Second 4 bits -> flags

# Implementation

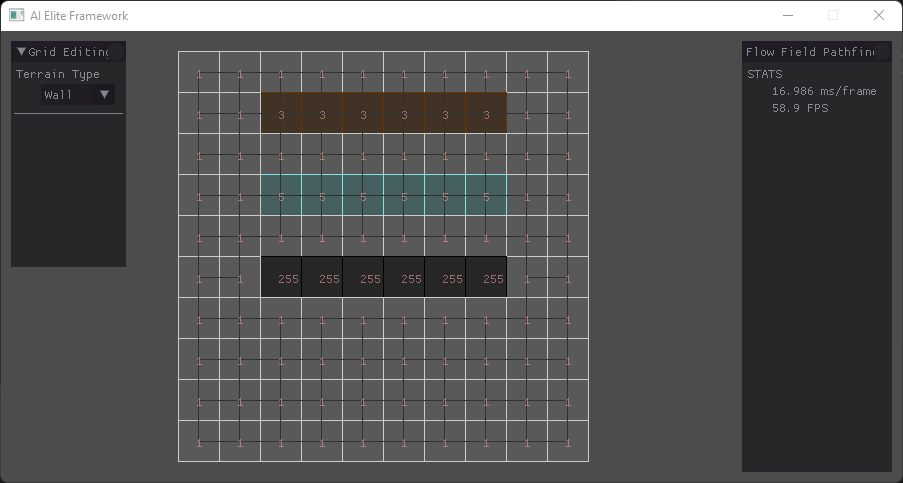
## Step 1: Generate Cost-field

It’s not necessary to generate an extra cost field, since all the nodes contain the cost to travel to them.

Currently there are 4 types of terrain:

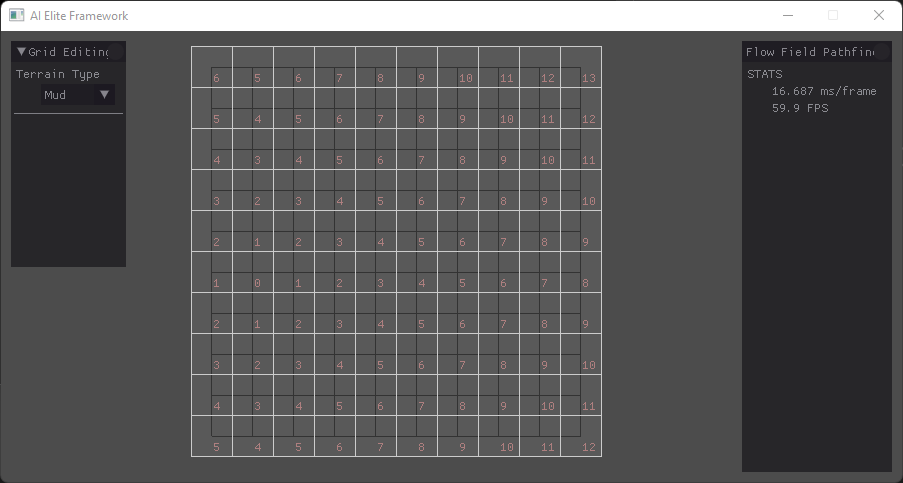
* Ground (cost =1)
* Mud (cost = 3)
* Water (cost = 5)
* Wall (cost = 255)

The cost/terrain is stored in a uint8, since we don’t need more numbers for the scale of this demo. If more would be needed, it would be as easy as using a larger size integer or even a float.

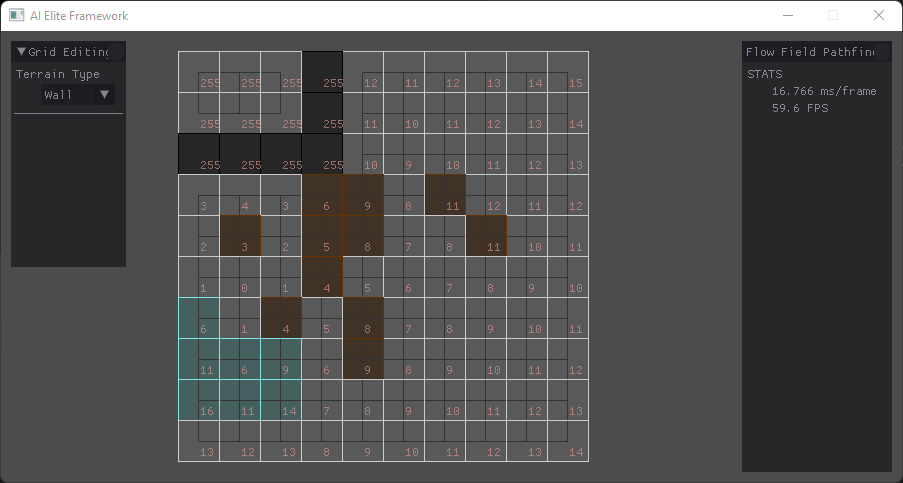
This is how the ‘Cost-field” looks with debug rendering.

## Step 2: Generate integration field

The integration field assigns new costs to each cell. The target cell always has a cost of 0. Starting from this cell, we check every other cell. For every cell, we check the neighbors and remember the lowest neighboring score. We add this score with the cost field score of the current cell and this is our final cost.



This is how the integration field looks, without different type of terrain.

This is how the integration field looks, with different type of terrain.

This demo does not have any flags implemented/used.