Mars Rover: Pathfinder and Exploration Visualiser

Concept

A standard A-star and Djikstra pathfinder tool, with a variation: instead of functioning on a level surface with walls (a maze), this functions on rough terrain. It operates on an nxm grid, wherein each cell has a designated `height`, and the cost of going to another cell is a function of the difference in the `height`s of the two cells. This function can differ, based on whether we're optimising for time spent or energy consumption.

For a better idea of the concept, take a look at concept.html.

Program Structure

```
- main.js
- functional/
    - cell.js
    - grid.js
    - canvashelpers.js
    - terrainhelpers.js
- pathfinding/
    - astar.js
    - djikstra.js
    - pathhelpers.js
- exploration/
    - basicmove.js
    - explorer.js
    - scanners.js
```

Major Concepts Used

The A-Star and Djikstra Search

- Both are relatively straightforward algorithms, the only major addition done is the visualisation through timeout delays.

AStar implementation

Djikstra implementation

Visualising the algorithms

- The grid is updated by calling painter functions after they've already been updated internally
- The calls are decoupled to improve performance

- The function does the search and initiates cellPainters based on a delay for and animated effect.

Code that deals with cell updates and general grids can be found in functional

Generating the terrain

- Using a hill data structure
- Pick a random hill
- Apply the hill to a point in the grid
- repeat

The terrain generation algorithm and its helpers can be found in terrainhelpers

Scanning the terrain

- Setup scanners in 4 directions
- In each direction, a loop scans a triangle from inside-out
- The minimum_height is the height below which a cell cannot be accurately identified. It is updated row-wise

The scanners can be found here

Movement for exploration

- Only a basic movement algorithm was implemented
- The algorithm scans clockwise through every single cell in the grid, without consideration for hard-to-cross paths.
- A more advanced one would have to backtrack for completion, and even then completion is not guaranteed

The movement algorithm is defined here