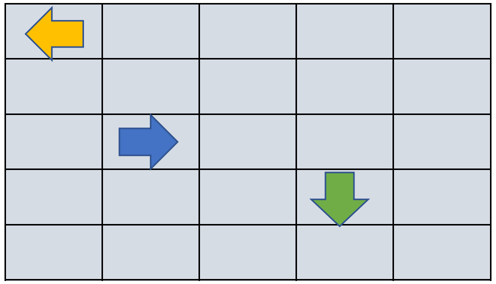
Fig. 1 Example with three Rovers; Mars surface
We will need to support *n* Rovers of different, unknown types in future, with various instrumentation



Assumptions

**1. Each Rover will need its own identifier.**

**2. This cannot be a stateless system, because:**

2.2 Rovers must know where other Rovers are;
2.3 Rovers must re-use their last locations as input parameters

3. **Mars is a simple, risk-free place for now:**

3.1 No sophisticated errorr-throwing (No water! No aliens! No gradients! A thermostable environment! Parts that never fail!)

Fig. 2 movement permutations and logic for x and y.
x and y are initial and end point horizontal and vertical coordinate points.
This is also the basis for movement logic unit tests.

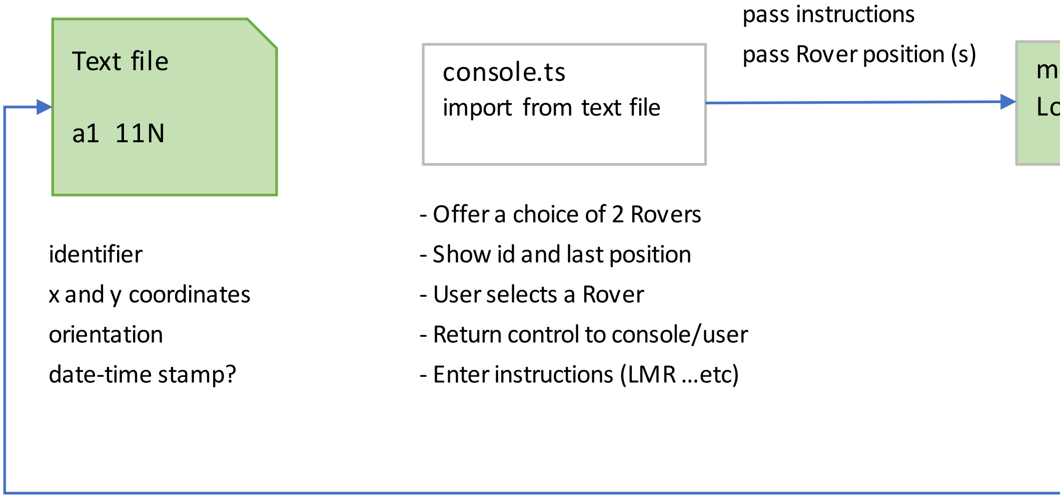
| | | | |
|---|---|---|---|
| x, y, N | x, y, N | x, y, N | **Initial coordinate point** |
| L | R | M | Movement instruction |
| x, y, E | x, y, W | x, y + 1, N | End point |
| x, y, S | x, y, S | x, y, S | **Initial coordinate point** |
| L | R | M | Movement instruction |
| x, y, W | x, y, E | x, y - 1, S | End point |
| x, y, E | x, y, E | x, y, E | **Initial coordinate point** |
| L | R | M | Movement instruction |
| x, y , S | x, y, N | x - 1, y, E | End point |
| x, y, W | x, y, W | x, y, W | **Initial coordinate point** |
| L | R | M | Movement instruction |
| x, y, N | x, y, S | x + 1, y, W | End point |

**If instruction === M:**
Re-set y or x according to initial orientation N, S, E or W
if N: y = y +1, if S: y = y - 1, if E: x = x - 1, if W: x = x + 1

**If instruction === L:**
Re-set orientation from initial orientation
if N: orientation = E, if S: orientation = W, if E: orientation = S, if W: orientation = N

**If instruction === R:**
Re-set orientation from initial orientation

Goal for minimum scope is reading an original position and instructions from a



Text file

a1 11N

identifier

x and y coordinates

orientation

date-time stamp?

console.ts
import from text file

pass instructions

pass Rover position (s)

m
Lc

- Offer a choice of 2 Rovers
- Show id and last position
- User selects a Rover
- Return control to console/user
- Enter instructions (LMR ...etc)

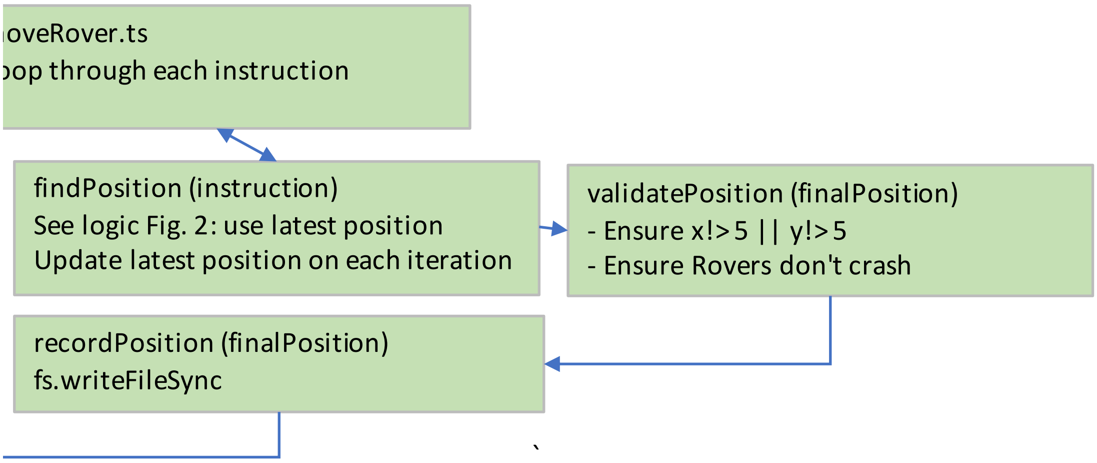
Data structures:

```
const instructions = [
'L',
'R',
'M'
]
as const;
```

```
class Rover {
id: Roverid = validID[0];
//can have other properties in future:
instrumentation, Rover type,
};
```

```
interface posi
  orientatior
 x: number
 y: number
 lastUpdatec
```

Instructions will be fed in as string but parsed to array text file and writing out the result again to the file.



oveRover.ts
oop through each instruction

findPosition (instruction)
See logic Fig. 2: use latest position
Update latest position on each iteration

validatePosition (finalPosition)
- Ensure x!>5 || y!>5
- Ensure Rovers don't crash

recordPosition (finalPosition)
fs.writeFileSync

```
tion {
n: string //or could be a custom Type (N, S ...)


d: Date //can I use a Date object?
```