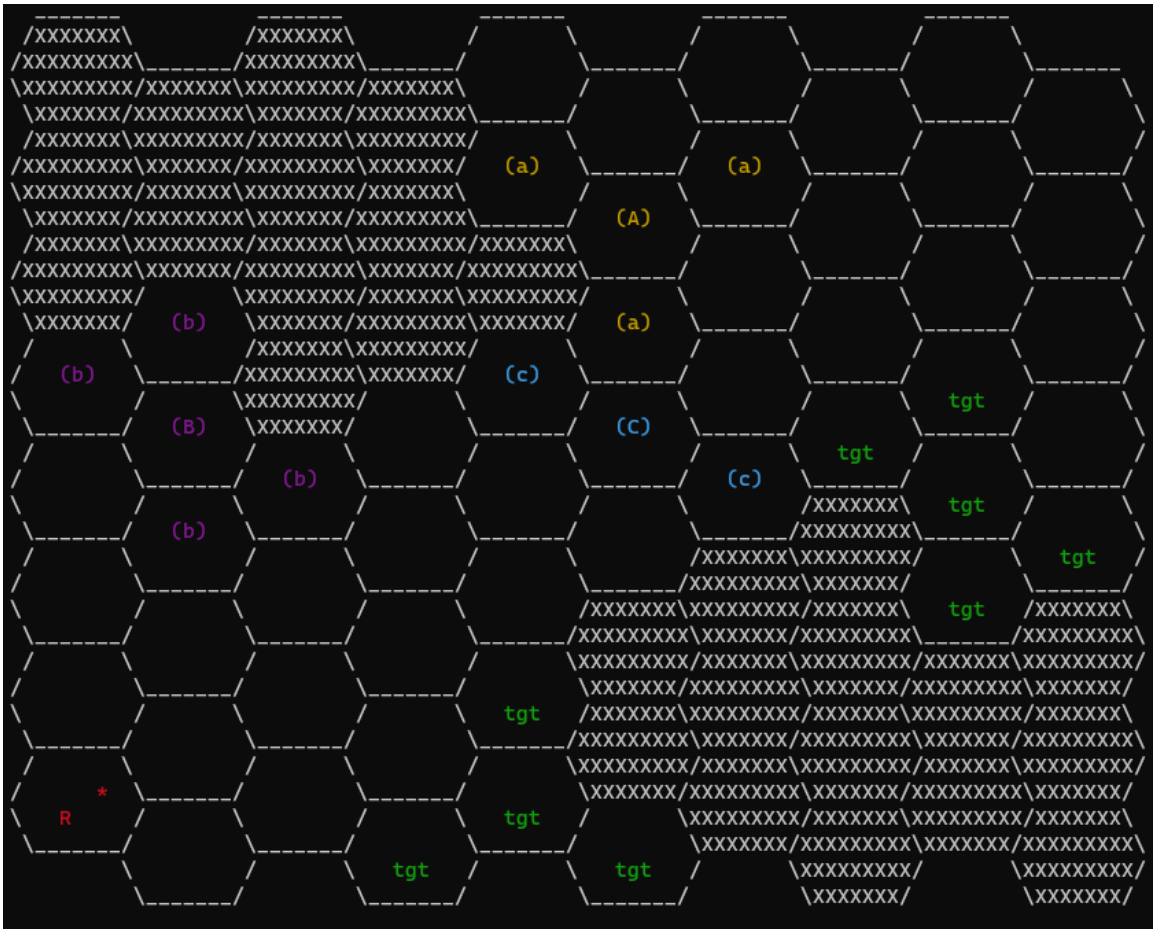
HexBot Robot Environment

COMP3702 Artificial Intelligence 2022



You have been tasked with developing a search algorithm for automatically controlling HexBot, a multi-purpose robot which operates in a hexagonal environment, and has the capability to push and pull ‘Widgets’ in order to reposition and rotate them to target locations and orientations. To aid you in this task, we have provided a simulator and visualisation for the HexBot robot environment which you will interface with to develop your solution.

Hexagonal Grid

The environment is represented by a hexagonal grid. Each cell of the hex grid is indexed by (row, column) coordinates. The hex grid is indexed top to bottom, left to right (i.e. the top left corner has coordinates (0, 0) and the bottom right corner has coordinates (n_rows-1, n_cols-1)). Even numbered columns (starting from zero) are in the top half of the row, odd numbered columns are in the bottom half of the row. e.g.

```
row 0, col 0            row 0, col 2                ...
            row 0, col 1            row 0, col 3
row 1, col 0            row 1, col 2                ...
            row 1, col 1            row 1, col 3
    ...         ...         ...         ...
```

Two cells in the hex grid are considered adjacent if they share an edge. For each non-border cell, there are 6 adjacent cells.

Robot

The HexBot robot occupies a single cell in the hex grid. In the visualisation, the robot is represented by the cell marked with the character ‘R’. The side of the cell marked with ‘*’ represents the front of the robot. The state of the robot is defined by its (row, column) coordinates and its orientation (i.e. the direction its front side is pointing towards).

The robot has 4 available actions:

- Forward → move to the adjacent cell in the direction of the front of the robot (keeping the same orientation)
- Reverse → move to the adjacent cell in the opposite direction to the front of the robot (keeping the same orientation)
- Spin Left → rotate left (relative to the robot’s front, i.e. counterclockwise) by 60 degrees (staying in the same cell)
- Spin Right → rotate right (i.e. clockwise) by 60 degrees (staying in the same cell)

The robot is equipped with a gripper on its front side which allows it to manipulate Widgets. When the robot is positioned with its front side adjacent to a widget, performing the ‘Forward’ action will result in the Widget being pushed, while performing the ‘Reverse’ action will result in the Widget being pulled.

Obstacles

Some cells in the hex grid are obstacles. In the visualisation, these cells are filled with the character ‘X’. Any action which causes the robot or any part of a Widget to enter an obstacle cell is invalid (i.e. results in collision). The outside boundary of the hex grid behaves in the same way as an obstacle.

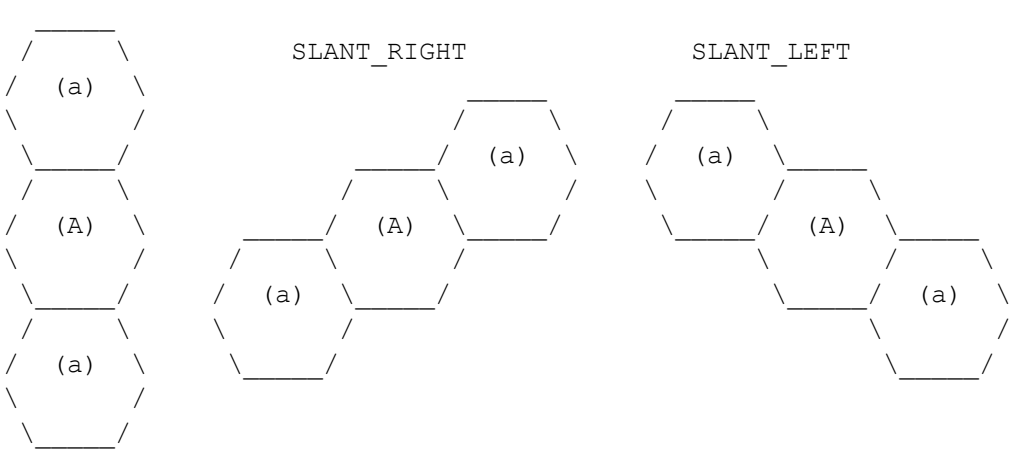
Widgets

Widgets are objects which occupy multiple cells of the hexagonal grid, and can be rotated and translated by the HexBot robot. The state of each widget is defined by its centre position (row, column) coordinates and its orientation. Widgets have rotational symmetries - orientations which are rotationally symmetric are considered to be the same.

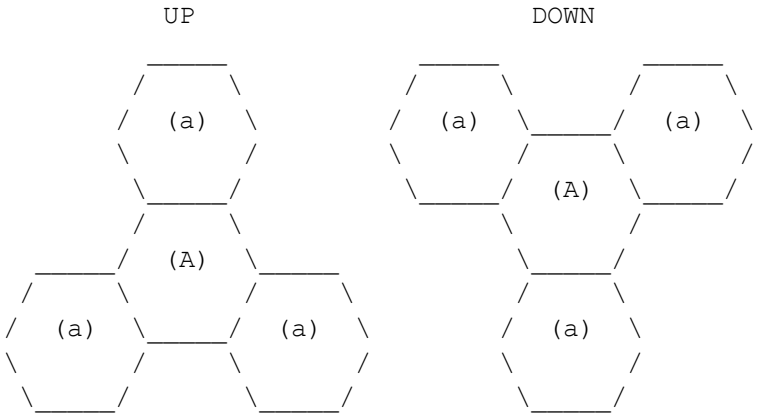
In the visualisation, each Widget in the environment is assigned a unique letter ‘a’, ‘b’, ‘c’, etc. Cells which are occupied by a widget are marked with the letter assigned to that widget (surrounded by round brackets). The centre position of the widget is marked by the uppercase version of the letter, while all other cells occupied by the widget are marked with the lowercase.

Three widget types are possible, called Widget3, Widget4 and Widget5, where the trailing number denotes the number of cells occupied by the widget. The shapes of these three Widget types and each of their possible orientations are shown below.

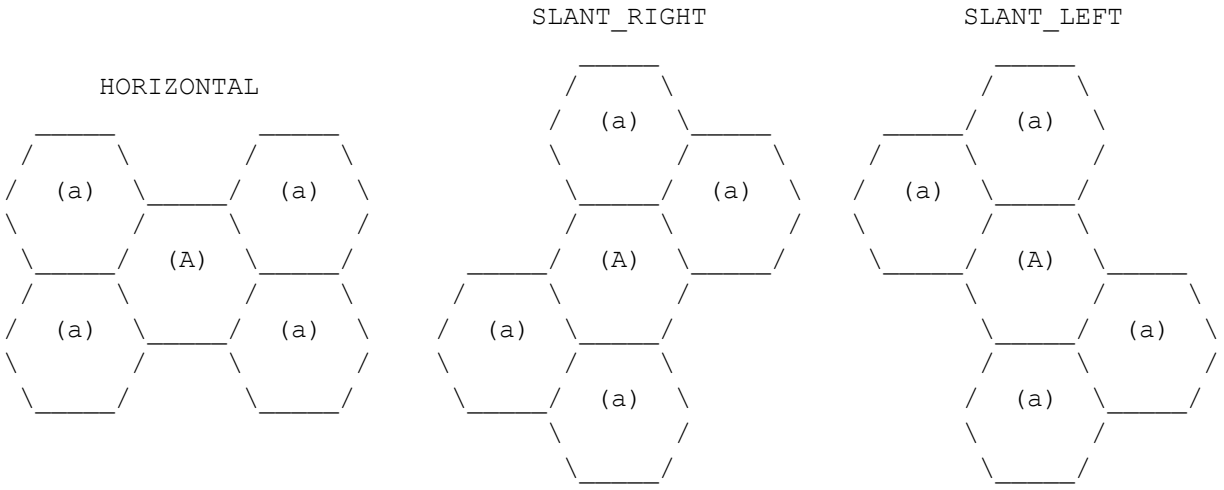
Widget3



Widget4



Widget5



Two types of widget movement are possible - translation (change in centre position) and rotation (change in orientation).

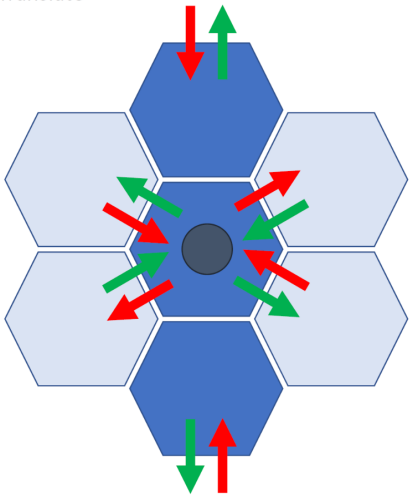
Translation occurs when the robot is positioned with its front side adjacent to one of the widgets cells such that the robot's orientation is in line with the widget's centre position. Translation results in the centre position of the widget moving in the same direction as the robot. The orientation of the widget does not change when translation occurs. Translation can occur when both 'Forward' or 'Reverse' actions are performed. For an action which results in translation to be valid, the new position of all cells of the moved widget must not intersect with the environment boundary, obstacles, the cells of any other widgets or the robot's new position.

Rotation occurs when the robot's current position is adjacent to the centre of the widget but the robot's orientation does not point towards the centre of the widget. Rotation results in the widget spinning around its centre point, causing the widget to change orientation. The position of the centre point does not change when rotation occurs. Rotation can only occur for the 'Forward' action - performing 'Reverse' in a situation where 'Forward' would result in a widget rotation is considered invalid.

The following diagrams show which moves result in translation or rotation for each widget type:

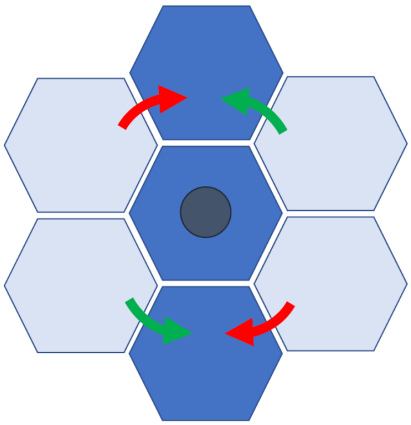
Forward Translate

Reverse Translate



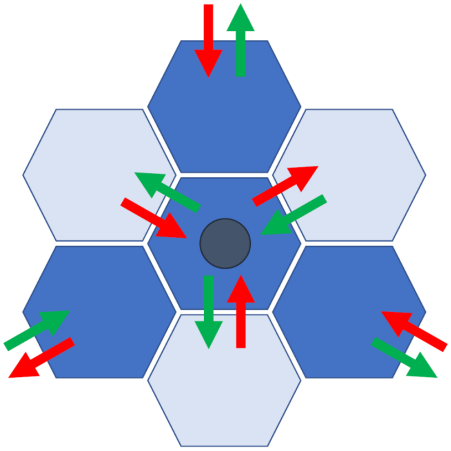
Forward Rotate CW

Forward Rotate CCW



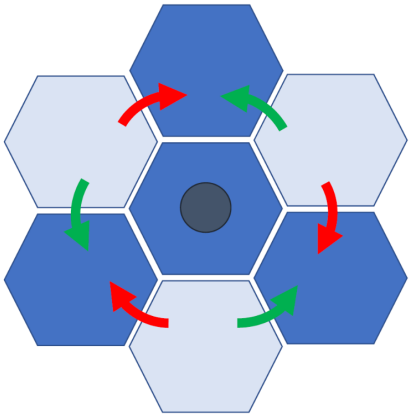
Forward Translate

Reverse Translate



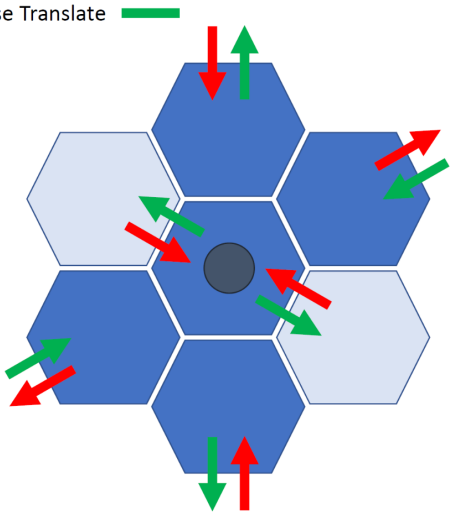
Forward Rotate CW

Forward Rotate CCW



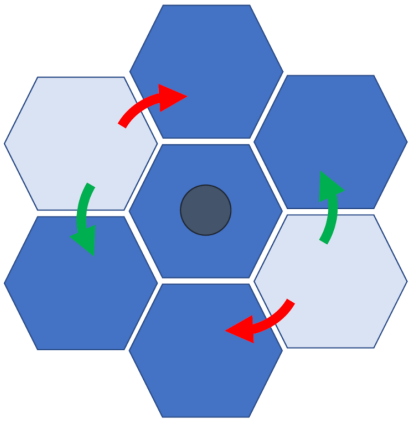
Forward Translate

Reverse Translate



Forward Rotate CW

Forward Rotate CCW



The arrows indicate directions from which the robot can push or pull a widget in order to cause a translation or rotation of the widget. Pushing in a direction which is not marked with an arrow is considered invalid.

Targets

The hex grid contains a number of ‘target’ cells. In the visualisation, these cells are marked with ‘tgt’. For a HexBot environment to be considered solved, each target cell must be occupied by part of a Widget. The number of targets in an environment is always less than or equal to the total number of cells occupied by all Widgets.