

## Lunar Lockout

The lunar lockout game has several pieces. First, we have a 5x5 game board with a red square marked in the middle. Next, we have 5 helper spacecraft in various colors and then 1 red spacecraft. We are also given card that specify initial setup position for some subset of spacecraft. On the back of each card is a solution.

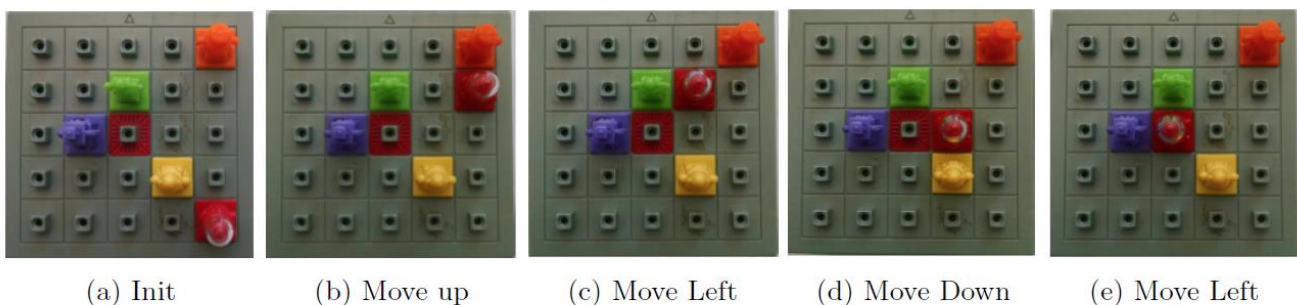


### Game Description

The goal of this game is to move the red spacecraft to the center red square. One can move any spacecraft but they are limited to moving up-down or left-right. Whenever a spacecraft moves, it continues moving until it hits another spacecraft. Note that it is not legal for the spacecraft to move off the board. Therefore, all spacecraft need to cooperate together to stay on the board and help the red spacecraft reach the goal.

### Game Play

Two beginner examples below represent the gameplay. Puzzle 1 is the beginner puzzle shown above. Starting from the initial setup, the red spacecraft moves up, left, down, then left to finally end up at the goal position. In this simple puzzle, only the red robot had to move. In the more complex puzzles, we also require other helper spacecraft to move. Puzzle 2 is an example requiring another helper robot to move as well.



Puzzle 1. The solution only requires moving the red spacecraft.



(a) Init (b) P - Move Down (c) P - Move Right (d) R - Move Right (e) R - Move Down

Puzzle 2. The solution requires first moving the purple spacecraft and then the red

The provided code implements a breadth first search algorithm to explore successor states until it finally arrives upon the solution. It can be executed by running the following command:

```
python run.py -p Puzzle1.txt
```

The text file contains the objects followed by their xy locations in tab separated form. For eg.

```
r 1 2
g 3 4
etc...
```

The output presents the board at each state in the plan.  
For eg.

```
- - r - o
- - - -
- - - -
- p - g -
- - - y -
-----
- - r - o
- - - -
- - - -
- - p g -
- - - y -
-----
- - - - o
- - - -
- - r - -
- - p g -
- - - y -
-----
```

Number of states traversed: 3