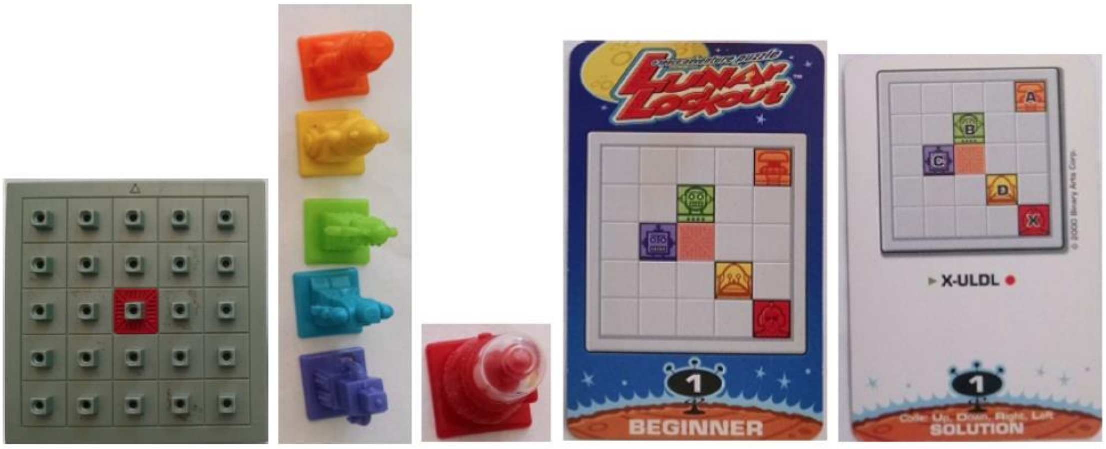
Lab 1 and Homework 1

1 Lunar Lockout Game

Game Pieces

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 cards that specify initial setup position for some subset of spacecraft. On the back of each card is a solution.



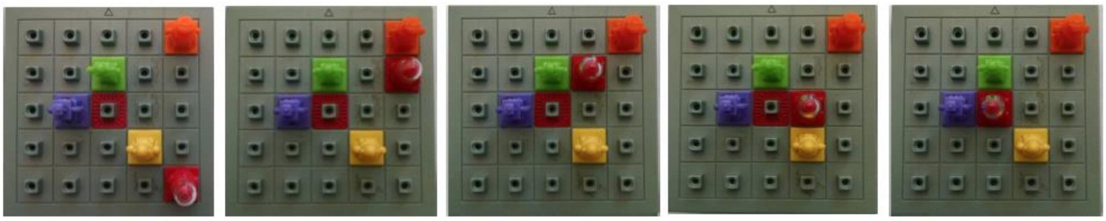
(a) 5x5 Board (b) Helpers (c) Red (d) Puzzle 1 (e) Solution 1

Game Description

The goal of this game it 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 on the board. Therefore, all spacecraft need to cooperate together to stay on the board and help the red spacecraft reach the goal.

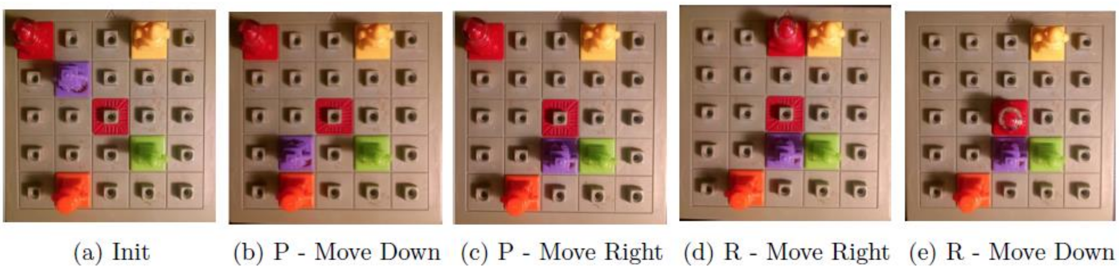
Game Play

We present two simple beginner puzzles to show typical game play. 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. In this assignment, you will be defining a representation for this puzzle and also implementing a planner to solve them.



(a) Init (b) Move up (c) Move Left (d) Move Down (e) Move Left

Figure 2: 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

Figure 3: Puzzle 2. The solution requires first moving the purple spacecraft and then the red Spacecraft.

a) [20 pts]

Define two very different ways to represent the lunar lockout world (objects, states, actions, domain axioms). Ensure there is a big difference in the set of possible states between both representations.

b) [20 pts]

Explain the number of possible states for both representations that you came up with. Do they also differ in the number of feasible states? Explain why each representation would be preferable to the other.

c) [60 pts]

Write the PDDL files required for a planner of your choice to solve this problem (Hint: a state-space search is likely the most simple). We should be able to run your code with something like the following command: /lunar-lockout <domain.pddl> <problem.pddl>. The <domain .pddl> should define the lunar lockout world actions and predicates. If you wish, you can hardcode the domain representation into your suggested planner but be sure to state this in your README. The <problem .pddl> contains the objects, init state, and goal state. The output of your planner should be a sequence of actions that solves the given puzzles. Explain how we should interpret the output. It is OK to use a more compact representation.

We recommend using **the PDDL language** to represent the problem. That way you can evaluate your problem definition with existing planners like: **FF planner.**
You can use visual studio on windows.