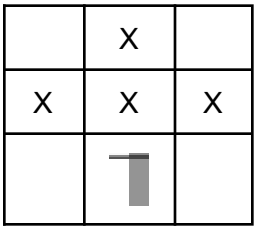
Lab 6.
Task for all.
Implement value iteration algorithm for situation when you have three possible actions: {rotate left 90, rotate right 90, make a step ahead}. What are the new states in this setup? Define probabilities for successful rotation and moving forward using rules from variant.

Create a medium-sized room with obstacles, one in-patch and one out-patch (so you have several similar ways out). An agent will appear on in-patch. When the agent comes to out-patch (or other terminal patches), this round is finished. In each variant, you need to create a simulation procedure that measures the average reward for the policy.
The policy is calculated as a result of the value iteration algorithm, but you should compare different policies calculated using different epsilon and delta.
Initially, you have four types of patches: black (reward -1, empty space), white (wall, can't go there), red (fire pit, the game finishes when going there), blue (out-patch, the game finishes when agent steps there). Rewards for patches you can set as you think suitable.

Variants.
0. Add teleports - pairs of patches, connected with each other, when the agent comes in one patch, he appears on the other with extra cost, defined from the interface.
Probabilities: when moving forward, with probability p - move forward, with probability 1 - p - stay on patch. When rotating - with probability q rotate as intended, 1-q - rotate other direction.
1. Add moving stripes (escalators), which increase moving speed twice. Show how they affect utility function.
Probabilities: when choosing one choice, with probability p - do this action, with (1 - p) / 2, make one of the other choices.
2. Add poisonous barrels that spread poison around. Make the radius adjustable, and the percent of poison (25% for neighbors4, 10% for next neighbor cells) is the probability of getting a big negative reward (-100, for example). How to include this case in the model?
Probabilities: Rotating is always done as commanded, but moving forwards sometimes moves backward.
3. Ice skating. Replace black patches with light blue ice patches - now each step forward is a lottery when you perform the intended action with probability p, and with (1 - p)/4, you move in any of crossed patches if possible (see image below).



Extra task for all (for extra credits).

1. Implement one box pushing from any point A to point B (need to make possible to define in the interface) for a given maze. If this is not possible - the model should show a message about it.