

[12] Try it out! (20 minutes)

1. Find the “mazeGraph.txt” file, with a similar format containing a maze game graph (but: for each node, we now specify its connection in each of the 4 directions up, right, down, left, being empty if there is no connection there; adjust your code to be able to read this format instead, and store the information appropriately in the node, e.g. mapping from direction to node ID instead). The first line in this file specifies the width and height of the grid represented.
2. Add the nodes read from the file into a `HashMap` in the `GameState` class as our maze representation, mapping from node ID to the node object.
3. In the `ForwardModel.next()` method, we can now apply the player actions (and check if a player is trying to do a valid action according to our maze graph), e.g.
`mazeGraph.get(playerPosition).contains(action)` – returns true if player can move in that direction, and `nextPlayerPosition = mazeGraph.get(playerPosition).get(action)` ;
 - a) If that can't be executed, the player might try to not move, or to tag another player. In the second case, we need to check if exactly 1 opponent is neighbouring the player, and if so, apply the tagging action).
 - b) The actions array contains one action per player, where the index in the array corresponds to the player ID, e.g. player ID 0 takes action at index 0. Apply all actions in the given `gameState`. Remember that we defined the actions as follows previously: 0 – do nothing, 1 – move up, 2 – move right, 3 – move down, 4 – move left, 5 – tag.
 - c) For now, you can allow players to move on top of each other (i.e. have the same position).
4. With this in place, we can run the game! How does it work with random players?

[12] Additional Materials

- The “mazeGraphDraw” folder is a package containing several Java classes. Running the DrawGraph class from this package draws an initially empty grid of a particular size (set in the main method).
 - You can left-click and drag your mouse to draw lines on top of the dashed ones, which are seen as walls and cut off the connections between the grid cells (the click point identification is a bit off, so the lines actually drawn might not be **quite** what you intended – drawing needs to be adjusted a bit for this to be more precise).
 - A right-click would remove the last drawn line segment (a 1-cell long part of the line you just drew).
 - Closing the drawing window (NOT stopping the program) will print to console the graph corresponding to the grid you’ve drawn.
- The text file suggested in the exercise uses this tool to generate it. Feel free to draw your own mazes of different sizes and experiment with those too.
- In case the display is not right in your PC, adjust the parameters at the top of DrawGraph file (offsetX, offsetY, cellSize), or the dimension of the window drawn (in the getPreferredSize method in this class).