

# Problem Wk.14.1.2: Robot on a grid map

Please read the Software Lab 14 handout.

If a move would cause a collision, the next state should be the same as the current state. **The returned cost should be the same as if the move had been legal.**

You can test in Idle by adding this to your file. This creates a small grid map for testing.

```
import lib601.gridMap as gridMap
class TestGridMap(gridMap.GridMap):
    def __init__(self, gridSize):
        (self.xN, self.yN) = (5, 5)
        self.xStep = gridSize
        self.yStep = gridSize
        self.xMin = self.yMin = 0.0
        self.xMax = self.yMax = gridSize * 5
        self.grid = util.make2DArray(5, 5, False)
        for i in range(5):
            self.grid[i][0] = True
            self.grid[i][4] = True
        for j in range(5):
            self.grid[0][j] = True
            self.grid[4][j] = True
        self.grid[3][3] = True

    def robotCanOccupy(self, (xIndex, yIndex)):
        return not self.grid[xIndex][yIndex]
```

You should then be able to execute the test cases run below.

```
class GridDynamics(sm.SM):
    legalInputs = None
    def __init__(self, gridMap):
        pass
    def getNextValues(self, state, inp):
        pass
```

MIT OpenCourseWare  
<http://ocw.mit.edu>

6.01SC Introduction to Electrical Engineering and Computer Science  
Spring 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.