Chapter 1 - Algorithms and Abstractions\Robot Mazes\Abstractions\Abstraction HW (STUDENT)\Abstraction HW.html

```
<!DOCTYPE html>

<!--
 *
 * Maze Simulator (c) by Christopher Grattoni
 * Maze Simulator is licensed under a
 * Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.
 * You should have received a copy of the license along with this work.
 * If not, see <http://creativecommons.org/licenses/by-nc-sa/4.0/>.
 *
 * Last Edited: Aug 9, 2017
 *
 -->

<html>
<head>
    <title>
        Robot Maze Simulator
    </title>

    <style>
        canvas{
            background: #000000;
        }
    </style>
    <script type="text/javascript" src="maze.js"></script>
    <script type="text/javascript" src="speed.js"></script>
    <script type="text/javascript" src="security.js"></script>
    <script type="text/javascript" src="movementfunctions.js"></script>
    <script type="text/javascript" src="engine.js"></script>
    <script>
        /**
         * Functions you can use:
         *      moveForward(): The robot will move forward
         *          by one square relative to the direction
         *          it is currently facing. If you move into
         *          a white square, the game continues.
         *          If you move into a gray square, you win.
         *          If you try to move into a black square,
         *          you lose the game.
         *
         *      rotateRight(): The robot will rotate to the
         *          right relative to its current orientation.
         *
         *      rotateLeft(): The robot will rotate to the
         *          left relative to its current orientation.
         *
         *      goalReached(): The function returns true if
         *          you have reached the end of the maze. It
         *          returns false if you are still in a white
         *          square. This function can only be called
```

```
         *          100 times per maze to try to prevent the
         *          game from crashing.
         *
         *      canMove(direction): This function returns true
         *          if the robot can move in the specified direction
         *          relative to its current orientation. Otherwise,
         *          it returns false. You must replace the parameter
         *          'direction' with one of the following arguments:
         *              'forward'
         *              'backward'
         *              'left'
         *              'right'
         *          Note: you need to include the quotes since this function
         *          only accepts a string as its argument.
         */

        /**
         * INSTRUCTIONS
         * Solve the maze by creating and utilizing the following two abstractions (functions):
         *      1. multiMove(num)
         *          - This function takes a number as a parameter and moves forward the number of spaces given
         *      2. roundAbout()
         *          - This function will complete the 360 degree turns found at each corner. Each turn can be completed by calling this one function.
         * Your solution must use a combination of these two functions to complete the maze and finish in the gray box.  You must travel counter-clockwise!
         */
        function robotInstructions()
        {
            moveForward();
            rotateLeft();
            multiMove(4);
            roundAbout();
            multiMove(7);
            roundAbout();
            multiMove(7);
            roundAbout();
            multiMove(7);
            roundAbout();
            multiMove(2);
        }

        function multiMove(num)
        {
            // Code goes here
            for (let i = 0; i < num; i++)
            {
                moveForward();
            }
        }

        function roundAbout()
        {
            // Code goes here
            multiMove(2);
```

```
            rotateRight();
            multiMove(2);
            rotateRight();
            multiMove(2);
            rotateRight();
            multiMove(2);
        }
    </script>

</head>


<body onload="gameFrameworkInit()">

    <canvas id="myCanvas" width="400" height="500"></canvas>

</body>


</html>

```