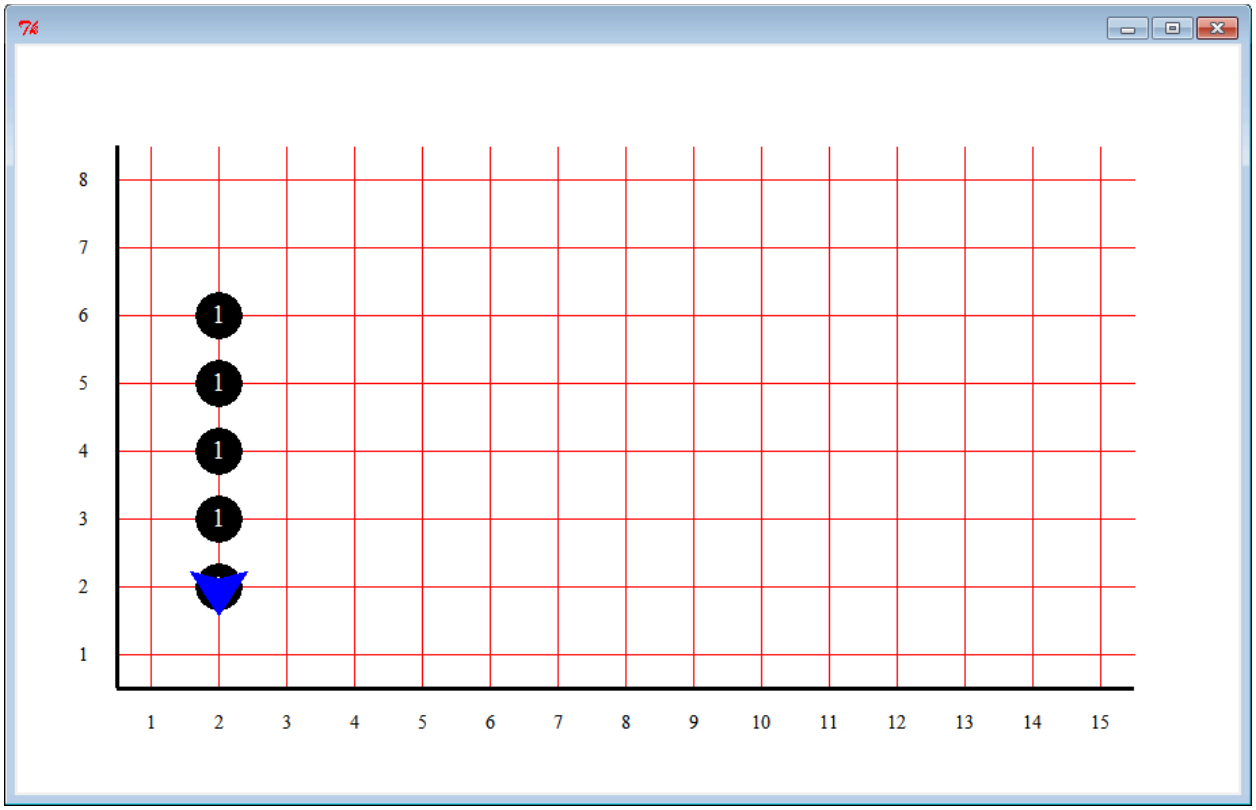
Lab 7: Functions and Conditionals

Objective(s):

- You will define functions with conditional if-statements in the definition

Directions:

1. Create a new file named **Lab7.py**
2. Use the **empty** world
3. If you feel like you want to define other functions to help you, then define them. Examples: **turnRight(), turnAround()**, etc.
4. Define a function named **drawOne(myRobot)** that draws a numerical one (1) from bottom to top using 5 beepers and returns back to where it started. The robot will move and place beepers repeatedly to accomplish this. Remember, after it finishes placing the beepers, it should move back to its starting position (at the bottom of the 1)



5. Test out your function by making a function call to see if it works. Remember, you can set the starting direction of the robot like so:

```
karel = Robot(wld, x=3, y=3, direction=north, beepers=5)
```

6. If your function call works, what happens when you change the robot's starting direction?

   What happens when the robot starts facing south? east? west?

   What you should see is that your **drawOne()** function does not always draw a vertical, numerical 1 because it depends on the starting direction of the robot

7. You need to add several if-statements to your **drawOne()** function definition so that before the robot moves and places beepers, you correctly reorient the robot. In order for the robot to always draw a vertical one, it must be facing North before it moves. Add if-statements that check what direction the robot is facing, and then command the robot the appropriate number of turns so that it faces North before it begins to move.

Example:

```
if myRobot.facingEast():
  myRobot.turnLeft()
```

Robot Condition Functions

```
anyBeepersInBeeperBag() - True when robot has beepers, False otherwise
frontIsClear() - True when front of robot is clear, False otherwise
leftIsClear() - True when left of robot is clear, False otherwise
rightIsClear() - True when right of robot is clear, False otherwise
backIsClear() - True when back of robot is clear, False otherwise
facingNorth() - True when robot is facing North (up), False otherwise
facingSouth() - True when robot is facing South (down), False otherwise
facingEast() - True when robot is facing East (right), False otherwise
facingWest() - True when robot is facing West (left), False otherwise
nextToABeeper() - True when robot is on top of a beeper, False otherwise
nextToARobot() - True when robot is on top of another robot, False otherwise
```

7. Test out your code again with each of the four directions?

   Does North work? South? East? West?

   Does the robot draw a vertical one in every scenario? If it does not, then debug.

8. Define a new function named **drawZero(myRobot)** that draws a numerical zero (0) of height 5 beepers and width 4 beepers. You decide what direction to draw the zero, but be sure to use if-statements at the beginning of the function definition to correctly orient your robot.

9. Test **drawZero()** by using a function call, and be sure to test it with different starting directions.

10. Create several new robots (name them whatever you want) and command them to write out your age in binary using ones (1s) and zeros (0s). To figure out your age in binary, just Google it (or find a decimal->binary converter). If you need more space in the world or you want the robots to move faster, you can always change the initial parameters of the world like so:

```
wld = World("empty", height=8, width=20, delay=0.2)
```

11. Demonstrate you program to Mr. Bui

12. Submit the **Lab7.py** file to Google Classroom