**Project 4: Development of Robots**

Due date: 11/24/2014

1. **Project description:**

In this project, you will be asked to develop several robots with varying behavior and use them to accomplish one task which is to finish a race. A race is described as 100 squares and the starting point starts at the square 1. Each square represents a possible position along the race. The finish line is at square 100. The first robot to reach or pass square 100 is rewarded with a box of Chocolate. The race weaves up the side of a slippery mountain. There is a clock that ticks once per second. With each tick of the clock, your robots should move according to the *move* function built in each robot. A robot would have a position (all starts at square 1) and a function move would modify that position. In your program, you should provide a common superclass called Robot whose *move* function does nothing. Then you will develop subclasses called RandomRobot, FunctionRobot, and UnpredictedRobot. Each of these robots has a different strategy of racing, meaning different *move* functions. The RandomRobot simple makes random moves forward as shown in Table 1. FunctionRobot will use a function to determine the next move. This function is shown in Figure 1. UnpredictedRobot behavior moves as shown in Table 2.

|  |  |
| --- | --- |
| 20% of the time | RandomRobot moves forward by 1 square |
| 30% of the time | RandomRobot moves forward by 2 squares |
| 35% of the time | RandomRobot stay at the same position |
| 15% of the time | RandomRobot will jump forward 6 square |

Table 1: RandomRobot’s behaviors

At starting point: FunctionRobot moves 4 squares ahead

Otherwise: FunctionRobot will move f(x) squares with f(x) = 2\* log2(x) with x is the current distance from the starting point. (Round this up to be integer value).

Figure 1: FunctionRobot’s behaviors

|  |  |
| --- | --- |
| 30% of the time | UnpredictedRobot moves forward by 3 squares |
| 20% of the time | UnpredictedRobot moves backward by 2 squares |
| 10% of the time | UnpredictedRobot stay at the same position |
| 25% of the time | UnpredictedRoobot jump forward by 6 squares |
| 15% of the time | UnpredictedRobot jump backward by 5 squares |

Table 2: UnpredictedRobot’s behaviors

Begin the race by printing “Ready! Set go”

If the three robot landed on the same square, you should print out “Ouch!” at that position. You will print R for RandomRobot, M for MemoryRobot and U for UnpredictedRobot.

After printing each line, test whether each robot has reached or passed square 100. If so, print the winter and terminate the simulation. If all three robot pass, print “This is a tie”.

1. Submission details:

All \*.java files need to be in Dropbox Project 4 by the midnight of the due date.