

Lab 02 – Monte Carlo Simulation

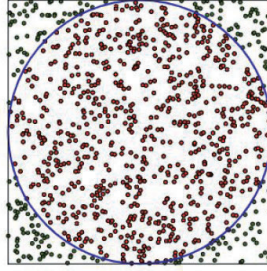
Objectives:

- Learn to use Java to generate random numbers
- Learn to use Java to perform simulation

1. **Monte Carlo Method** is a method of solving problems using statistics. Given the probability, P , that an event will occur in certain conditions, a computer can be used to generate those conditions repeatedly. The number of times the event occurs divided by the number of times the conditions are generated should be approximately equal to P .

We can estimate the value of π using the Monte Carlo method. If a circle of radius r is inscribed inside a square with side length $2r$, then the area of the circle will be πr^2 and the area of the square will be $4r^2$, and the ratio of the area of the circle to the area of the square will be $\pi/4$. Therefore, if you pick n points at random inside the square and find m points inside the circle, then π can be approximated as follows: $\pi \approx 4m/n$

Monte-Carlo Method

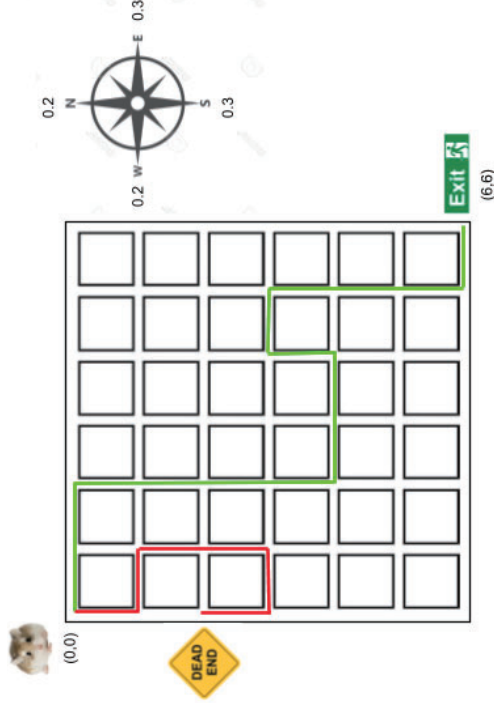


$$\pi \approx \frac{783}{1000} \times 4 = 3.132$$

You are asked to write a simulation program to estimate the value of π . By the law of large numbers, increase the number of sample points in the above simulation will make the approximation closer to the expected value 3.1415.

```
Output - Lab (run) x BacktrackSolver.java x Solver.java x Sudoku.java x
RUN:
This program approximates PI using the Monte Carlo method.
Please enter number of throws (n): 1000
Computed PI = 3.124, Difference = 0.017592653589793006
BUILD SUCCESSFUL (total time: 6 seconds)
```

2. **[Homework]** A mouse is trying to escape from a maze where it starts moving from the top-left corner (0,0) to the exit in the bottom-right corner (6,6). It moves according to the following rules:
 - The mouse randomly goes to one of the four directions - north, east, south and west based on the probability (0.2 or 0.3) specified below.
 - The new position must be within the boundaries of the maze.
 - The new position must not have been visited previously.
 - If the new position does not fulfil the criteria above, the mouse will randomly pick a new direction again. Every decision is independent of the others.
 - If the mouse reaches a dead-end - all directions are not enterable, it fails to leave the maze.
 - The mouse continues to move until it reaches the exit.



Write a simulation program using the Monte Carlo method to estimate the probability (P) of the mouse successfully escaping from the maze. The simulation should run one million times. Your program should print out the simulation results clearly and round the success rate P to 3 digits after the decimal point as follow:

The Monte Carlo simulation result of one million runs:
 No. of successful escape: 406254
 Success Rate P : 0.406

The acceptable range of the success rate is 0.405 - 0.410 and the execution time should be less than 5 seconds.

In addition to the above functional requirements, your program should also be object-oriented and follow the Java naming convention. That is, you need to design one or more classes to model the simulation program. Programming style will be considered in the marking.

All your classes should be organised in the package **hw1.sDDDDDDDD** where the D-string is your 8-digit student ID. Your `main` method should be located in the `Main` class as shown below.

```
package hw1.sDDDDDDDD; // DDDDDDDD should be replaced by your real SID

public class Main {

    public static void main(String[] args) {

        // entry point of your program

    }

}
```

- END -