

COMP 1600 Fall 2025

Lab 8: Visual Approximation

Due by the end of class

Your mission is to make a visualization for the Monte Carlo approximation for pi that we discussed in class in the context of `for` loops.

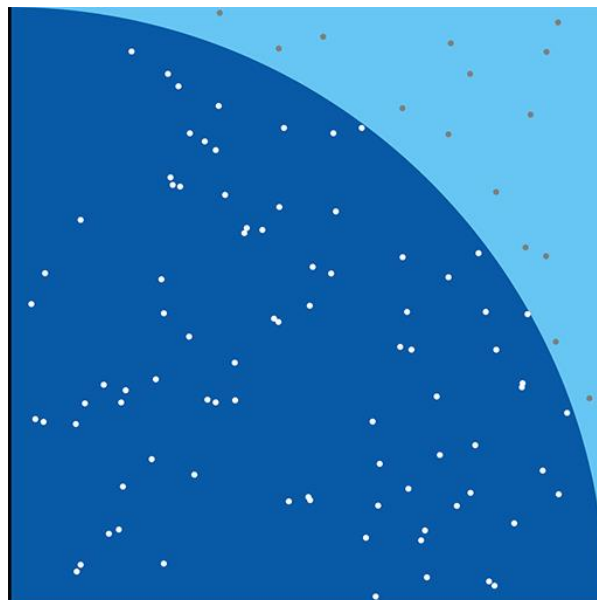
Approximating pi is a useful task, although we are not doing so in the most efficient way. Nevertheless, a Monte Carlo approximation for pi is very straightforward. Now that we have the `StdDraw` library, we can make an attractive visualization to accompany the approximation.

Specification

Create a project called `Lab8`. Add a class called `Approximation`. Your program will prompt the user for the number of "darts" to throw. Your program will then run the simulation that number of times, throwing random darts into the region defined by $0 \leq x < 1$ and $0 \leq y < 1$. Below is sample output for when a user enters `100`.

```
How many darts would you like to throw? 100  
Your approximation of pi is: 3.4
```

The accompanying visualization for this run of the program is below. Note, of course, that this program uses random values and will consequently have different output each time it is run, even with the same input.



Recall that you should generate random x and y points in the range $[0, 1)$ for each dart. If the dart falls inside the unit circle, increment the counter for the total number of hits. When you have thrown all of your darts, the final approximation to π is 4.0 times the hits divided by the total.

The points inside the circle are given by the equation $x^2 + y^2 \leq 1$. If you plug the x and y values for a specific point into this formula and the result is less than or equal to 1, it is a hit. Since we are only in Quadrant I, all values are positive.

Drawing

To match the sample output, you should set the pen color to `StdDraw.BOOK_LIGHT_BLUE` and draw a square filling the region in question. Then, set the pen color to `StdDraw.BOOK_BLUE` and draw a circle of radius 1 centered at the origin.

After setting up the circle and square, set the pen radius to `0.01`. Then, set the pen color to `StdDraw.BLACK` and put a line from the origin to $(0,1)$ and to $(1,0)$ to show the x and y axes.

When actually finding the random location of a dart, set the pen color to `StdDraw.WHITE` if the dart is a hit, and set the pen color to `StdDraw.GRAY` if it is a miss. Then, mark the point.

To use `StdDraw`, you must download `StdDraw.java` from [here](#). You can either save it in your project `src` folder once you have created your project or you can add a `StdDraw` class to your project and paste the text in.

Methods

This lab uses many different methods. Here is a list of the ones you will need:

Math methods

- `Math.random()`

StdDraw methods

- `StdDraw.setPenColor()`
- `StdDraw.setPenRadius()`
- `StdDraw.filledCircle()`
- `StdDraw.filledSquare()`
- `StdDraw.point()`
- `StdDraw.line()`

Turn In

Turn in your code by uploading `Approximation.java` from the `Lab8\src` folder wherever you created your project to [Brightspace](#). **Do not** upload the entire project. I only want the `Approximation.java` file.

All work must be done individually. Never look at someone else's code. Please refer to the course policies if you have any questions about academic integrity. If you have trouble with the assignment, I am always available for assistance.