

HNRS 2200
COGITATIONS ON COMPUTATION
Honors Nature of Science, Technology, & Mathematics
Spring 2019

Assignments 0423 and 0509

This assignment seeks to give you firsthand exposure to a central premise of current approaches in artificial intelligence and machine learning: with large data comes great predictive power. With enough data, patterns emerge that allow effective decision-making or forecasting. The trick is to *discern* those patterns from the mass of information. This assignment does the reverse, in the hope of enlightening the same premise: by generating randomized data following certain rules, we hope to see how *more* data can lead to a particular (known) pattern, behavior, or computation.

Background Reading/Viewing

Due to the unanimous use of Python in the previous assignment, let's make the switch to HTML/CSS/JavaScript for this one. As mentioned before, I gravitate to the Mozilla Developer Network for details, sample code, and the occasional tutorial:

<https://developer.mozilla.org>

To build on our earlier Khan Academy experience but within the more flexible CodePen environment instead, we shall turn to the *p5* JavaScript library for graphics and animation:

<https://p5js.org>

What to Do

“Game of Coins:A Song of Dice & π ”

The assignment's title is a mash-up of the three options provided. They have these common:

- The user specifies how much data to generate
- An automated routine generates this data
- The results are visually displayed on the fly

Specifics then vary per option. Have fun with the display portion: meet the minimum requirements then get creative with how you show the coins, dice, or darts!

Option 1: Game of Coins

This program simulates a bag of 100 coins. “How much data” represents the number of times the bag is to be “thrown” into the air. The program should track how many times a particular coin lands as heads. The program's visual display should then a histogram that illustrates how many coins landed as heads at various frequencies.

Option 2:A Song of Dice

This program simulates the classic “2d6” dice combination: a pair of six-sided dice. “How much data” represents the number of times the dice are to be “rolled.” The program should track how many times each die lands as 1–6. The program's visual display should then be a table with rows representing how one die landed and columns representing the other, thus illustrating how many times each die rolled 1–6 as well as how many times a particular dice total (2–12) was rolled.

Option 3: & π

This program tries to estimate π ...by throwing darts! Using the $x^2 + y^2 = r^2$ definition of a circle, a randomly thrown “dart” within the square ranging from (0, 0) to (1, 1) can be determined to have landed either inside or outside the quarter-circle whose origin is at (0, 0) with a radius of 1. The ratio of darts inside the quarter-circle to the total number of darts approximates one-fourth the area of that circle, and because the radius of that circle is 1, then its area is π . (drawing a picture may help) “How much data” represents the number of darts to “throw” at the square. The program should track where each dart lands, and its visual display should show each location as well as whether that location is within the quarter-circle. Finally, the program should display its estimate for π .

What to Submit

Work on your programs in CodePen and submit the link to Brightspace. Students who work in pairs are individually responsible for submitting the link under their respective Brightspace assignment slots.