COGSCI 131 – Assignment 7
DUE: March 21st at class start

In this assignment, you will use an existing physics engine in order to implement and test the idea that people—well, you—are capable of mental physics simulations. We have provided template code which gives a friendly interface to the pymunk physics library (be sure you are using Python 3). To use it, you will have to install pymunk and pygame (e.g. via pip). The template code defines and creates a BlockTower, which is a stack of blocks at varying x-locations (mean is mid screen, standard deviation = xSD is fixed in the code) with a single black block at the top. The goal of you and of the physics simulation is to predict whether the black block will ever touch the ground once physics starts going.

The BlockTower class takes one argument: a vector of x-positions for blocks (the y-positions are fixed). It contains two methods that will be useful:
BlockTower.run_person – opens a display window showing the blocks. If you press "y" or "n", you can watch the simulation start and run for 10s. This returns a tuple of whether people predicted yes (True) or no (False) that the black block would hit the floor, and whether it actually did (True/False)
BlockTower.simulate – runs the same simulation but does NOT show it in a window. This simply returns a True/False for whether the black block will touch the floor within 10s of simulation time. This runs much faster than 10s.

1. [20pts] Your first task is to write an "experiment" that will run you in 200 trials with random starting positions, saving the starting positions, your predictions, and the physics simulation outcome. A recommended way to do this is to save these variables in a .csv file or a pickle file as the program runs so that if for some reason it stops or you get interrupted, you won't lose all of your data. Once you are sure that you are saving response and positions of the blocks you saw, you should run yourself through the experiment. (Each trial is 10s, so 200 of them should take 2000 seconds, or 33 minutes, plus a little for responding)

2. [30pts] Now you should write code that will load your saved positions and use the BlockTower.simulate() method to generate predictions from a noisy physics model. For this, assume that the noise is normal with a SD of S=10 (pixels) on the recorded starting positions from Q1. Run 100 simulations on each item you answered, with new noise each time, in order to compute a model predicted probability of the black box hitting the floor. To visualize this data, plot on the y-axis your averaged prediction from Q1 (True=1, False=0, averaged over items to yield a number in [0,1]) on items for which the model predicted <10%, 10-20%, 20-30%, etc. That is, group your 200 saved items (starting positions) by model prediction deciles and plot your measured responses for each group of items. What would a "good" model of you look like in this plot, and how does your look?

3. [10pts] Plot Q2 for S=1, 5, 10, 15, 20, 50, 100. How do you determine which S value matches best to your responses and what is the value of S? How well does the best S value match with your responses? If no value works well, come up with a hypothesis as to how people may be thinking about it differently than the model, based on patterns you see in the plots.

3. [+15pts extra credit] Use the simulation to create 20 items which the model thinks are very close to 50% likely to have the black block fall to the floor. Show all of these items to 5 friends and have them predict whether the box will hit the floor, without telling them that the model predicts 50%. What is their mean response? Does their average prediction agree with the model or not, and if not why might that be?