Math Simulator Use Guide

The Math Simulator

Prerequisites

Installing Python

On mac:

1. Enter the command: 'brew install python3' (or www.python.org/downloads)
    a. The program requires the following packages (that may need to be installed):
        i. Anaconda package (install instructions here)
            1. Numpy
            2. Matplotlib
            3. Pandas
        ii. Math (default)
        iii. Random (default)
        iv. Itertools (defualt)
        v. Tkinter (default, for the gui)
2. Enter the command: 'which python'
3. At the top of the main.py file, change the first line, with the hashbang to the path found: like #!/opt/homebrew/bin/python3, or #!/usr/local/bin/python
    a. 'echo $path' allows you to see what directory will be used for executable commands - this may allow you to see where the python command is searched for.. If it's not in the list, where it's installed.. Then add it to the path variable. (instructions referenced here)
4. Set the mode of the file to executable by: 'chmod +x file.command'
5. ??? profit? I didn't get this to work by the instructions, but the audience currently isn't huge, so contact me if you need assistance installing.

On Windows:

1. Go to https://www.python.org/downloads/, follow the instructions
2. Add python path to environment variables… (needing more info, probably)
3. Anaconda package (install instructions here)
    a. Numpy
    b. Matplotlib
    c. Pandas

4. Other package dependencies come standard in Python, as in 1.a.ii+ in the last section

Reel sizes, game window, etc:

The simulator currently only allows 3x3 and 5x3 configurations, 3 or 5 reels respectively. When building the spreadsheet windows, it's important to remember that counting for programming begins with 0, not 1.

Excel File Formatting

The excel formatting is important - both in column name and in cell values.
The placement of the input file as a default is in the /Math_Simulator/assets/ folder, the default name being PARishSheets.xlsx. The name and location doesn't matter, except that if it's different from the default it will need to be selected with the file input button at the top of the program.

The following sections will describe each of the necessary sheets for the simulator to function.

Reels

In the "Reels" sheet, the reel columns should be labeled as "Reel #", # being 1 through 3 to 5 in the first row. The program dynamically reads in the data, but the reel 'strip'/column needs to be continuous, no empty cells.
While reading debug data, the reel position starts counting at zero, and because we also have a header row, this means that the # that gets displayed is actually cell Reel#[n+2].

Reel 1	Reel 2	Reel 3	Reel 4	Reel 5
M1	M1	M1	M1	M1
M2	M2	M2	M2	M2
M3	M3	M3	M3	M3
M4	M4	M4	M4	M4
F5	F5	F5	F5	F5
F6	F6	F6	F6	F6
F7	F7	F7	F7	F7
F8	F8	F8	F8	F8
F9	F9	F9	F9	F9

Paylines

In the “Paylines” sheet, the reel columns should be labeled as “Reel #”, # being 1 through 3 to 5 in the first row. The cells represent the reel (starting from zero to total-1), and the positions (the placement on the reel).
For instance, the whole reel 1 column should have 0 as the first numeral, comma, followed by the position on the 1st reel that the payline will use. So the value ‘0,0’ is the first/leftmost reel, followed by using the first/top position.
At higher debug levels, you can see this being loaded in during the slot machine setup phase.

Reel 1	**Reel 2**	**Reel 3**	**Reel 4**	**Reel 5**
0,0	1,0	2,0	3,0	4,0
0,1	1,1	2,1	3,1	4,1
0,2	1,2	2,2	3,2	4,2
0,0	1,1	2,2	3,1	4,0
0,2	1,1	2,0	3,1	4,2

Paytable

The reel columns should be labeled as “Reel #”, # being 1 through 3 to 5 in the first row; plus a ‘win’ row (column name doesn’t matter, but it has to be the final column after the reels.). .. this one has two specifics: it needs to have only the columns designated, and for 5 reels, the empty cells mean 3 or 4 cells-to-win and is used to find winners.
In the debugging and math, the winning line rows are processed sequentially from the top, so the lines need to be ordered by decreasing win/prize amount.

Reel 1	Reel 2	Reel 3	Reel 4	Reel 5	win
W	W	W	W	W	100
W	W	W	W		80
W	W	W			70
M1	M1	M1	M1	M1	100

M1	M1	M1	M1		80
M1	M1	M1			70

Math / RTP

In the “RTP” sheet, there should be two columns: “RTP” and “Volatility”. These are necessary for the math text boxes and the magenta ‘Expected RTP ##%’ line in the math plot.

Usage

Once the prerequisites are completed: the python environment and the excel file in the /Math_Simulator/assets/ folder, the simulator is ready to use.

1. Compile / run the program.
    a. From a command prompt, in the Math_Simulator folder: python3 main.py
    b. (or run it as a mac command, as mentioned above)
    c. (or compile it for running in windows…)
    d. Note: at writing, it takes approximately 8 seconds to load on a dev workstation. Much of this is due to the imported packages.
2. Input File - ensure the input file is correct if it isn’t using the default convention
3. Check / change the Bet Per Line, initial Starting Credits, and check if you want infinite credits.
    a. Debug level is also read in at this step, if you need more verbose messaging.
    b. Caution: debug 1 starts getting quite slow after 10000 runs, debug 2+ are slow even at the 100-1000 range. Those are more for tracing, proving math and compliance, and development.
    c. Machine Credits is a read-only field, it’s a visual indicator of whether the simulation has been run/not. It also updates at the end of the simulation run to show the final credits
4. Click ‘2. Build Virtual Slot’ - the prior settings are all included here. This is the first active step, and will need to be repeated prior to each round of simulations.
5. Set Simulator Total Spins:
    a. default is low. With 0 debug, 5 reel/25 paylines and infinite credits at 100k spins takes about 45 seconds. Higher debug, more complex paylines/paytables, and million+spins take time.

    b. Note: lower spins and limited credits are a good way to watch the reel/paytable behavior play out over the credits plot..
    c. Higher number of simulated spins should show erratic credit plots, but should show converging RTP plots of different simulations.

6. Click ‘3. Run Simulation’
    a. Once complete, the read-only math variables should populate at the bottom of the simulator.
    b. By default, the simulation runs quiet and notifies you when complete.
    c. Once the simulation is complete, there should be a notice in the console output. (may change in the future)
7. If you need the win/spin simulated output, click the Save File button.
8.
9. 4b. Plot Credits Over Time
    a. This graph shows the credit totals over the course of the simulation.
    b. To plot multiple Credit lines over multiple simulations, only click the button once after each simulation. Clicking RTP plotting will clear the graph and use the RTP parameters.
10. 4c. Plot RTP Over Spins
    a. This graph shows the running Return To Player over the course of the simulation.
    b. To plot multiple Return to Player lines over multiple simulations, only click the RTP button once after each simulation. Clicking Credit plotting will clear the graph and use the Credit parameters.