```
% MATLAB Final
% Will McClain
% EGR 101-01
% Due: 4/18/23


clear; clc;

TRIALS = 1000;

matchups = zeros(TRIALS, 4, 3); %3d matrix of gamestates (x, y, z). x is current trial num, y is which behavior matchup is being played, z is possible outcomes (1 for player 1 win, 2 for player 2 win, 3 for tie)

for idx = 1:TRIALS
    % simulate_bot_match() returns {the name of who won, the id of who won}
    % we're interested only in the second element
    dummy_vs_dummy = simulate_bot_match(["Masone" "dummy"; "Twolia" "dummy"]);
    dummy_vs_novice = simulate_bot_match(["Masone" "dummy"; "Twolia" "novice"]);
    novice_vs_dummy = simulate_bot_match(["Masone" "novice"; "Twolia" "dummy"]);
    novice_vs_novice = simulate_bot_match(["Masone" "novice"; "Twolia" "novice"]);

    % for each behavior matchup, mark down the outcome of each trial
    % mark 1 if that outcome comes to pass, 0 if it doesn't
    % (x, y, 1) + (x, y, 2) + (x, y, 3) = 1 always because only one player can win
    matchups(idx, 1, dummy_vs_dummy{2}) = 1;
    matchups(idx, 2, dummy_vs_novice{2}) = 1;
    matchups(idx, 3, novice_vs_dummy{2}) = 1;
    matchups(idx, 4, novice_vs_novice{2}) = 1;
end

% mean (matchups, 1) averages probability of each outcome across all trials
% permute() changes matrix from ugly 1x4x3 to pretty 3x4
% then, mult by 100, convert to string, and append "%"
data = string(permute(mean(matchups, 1), [3 2 1])*100)+"%";

% make it a pretty table
Outcomes = ["Player 1 Win"; "Player 2 Win"; "Tie"];
DummyVsDummy = data(:, 1);
DummyVsNovice = data(:, 2);
NoviceVsDummy = data(:, 3);
NoviceVsNovice = data(:, 4);

% display that groovy data
disp(table(Outcomes, DummyVsDummy, DummyVsNovice, NoviceVsDummy, NoviceVsNovice))
```