**Project 3 Grading Guide**

|  |  |  |  |
| --- | --- | --- | --- |
| **Item no.** | **Description** | **Correctness** | **Style** |
| PROBLEM 1 | mwIllusion.m | Total = 9 | Total = 1 |
| REMINDER | MATLAB may scale diagrams proportionally to fit the figure window. This is fine! |  |  |
| 1 | There are n rectangles of color cg (1C) which span the entire width of the diagram (1C). | 2 |  |
| 2 | The rectangle height and spacing is equal to 1. | 1 |  |
| 3 | The lower left corner of the diagram is at coordinates (a,b). | 1 |  |
| 4 | There are two "stacks" of n-1 rectangles in color cs (1C) with width f\*w (1C). The stacks are drawn with 1 or 2 loops, and no loops are nested (1S). | 2 | 1 |
| 5 | The stacks are horizontally centered in the diagram, with the same amount of space left of the left stack, between the stacks, and right of the right stack. | 1 |  |
| 6 | If f >= .5, f is set to .3 before drawing. | 1 |  |
| 7 | The student uses DrawRectNoLine rather than DrawRect. | 1 |  |
|  | showIllusion.m | Total = 3 | Total = 0 |
| 8 | Calls mwIllusion three times with different parameter values. | 1 |  |
| 9 | The first two images demonstrate the illusion. (Don’t worry about whether the third image is a “weaker” illlusion.) | 1 |  |
| 10 | The three illusions are placed side-by-side with no overlap. | 1 |  |
| PROBLEM 2 | varySegment2.m | Total = 5 | Total = 4 |
| 11 | Uses an outer loop to iterate over the given values of r2. | 1 |  |
| 12 | Uses an inner loop to correctly compute compute the values D(beta) for 0<beta<2pi. | 1 |  |
| 13 | Values for beta and D(beta) are correctly stored in vectors. (-1C for a mistake with vector indices, -2 max.) | 2 |  |
| 14 | The plot(...) statements are placed correctly inside the loop so that each line is plotted exactly once. | 1 |  |
| 15 | Values are displayed in a plot with an informative title (1S), axis labels (1S), and correct legend (1S). |  | 3 |
| 16 | A horizontal line is added but does not appear on the legend. |  | 1 |
| PROBLEM 3 | gameGapN.m | Total = 3 | Total = 1 |
| NOTE | There is a typo in the book. It mistakenly says that you should keep running the game while |# of tosses - # of heads| < N but this should actually be the |# of tails - # of heads| < N. Accept either stopping condition. |  |  |
| 17 | Correctly simulates flipping a coin with probability ph (e.g. by using rand()). | 1 |  |
| 18 | Uses a loop which terminates when the stopping condition is reached. | 1 |  |
| 19 | Loop condition is fully simplified. No nested loops are used. |  | 1 |
| 20 | Correctly computes and returns the score s. | 1 |  |
|  | statsGapN.m | Total = 9 | Total = 3 |
| NOTE | It’s OK if they split the code for part a) and b) into two separate blocks. |  |  |
| 21 | An outer loop varies N in a reasonable range (this should at least include N=5 to N=10). | 1 |  |
| 22 | An inner loop performs multiple trials for each value of N (1C). There are at least 100 trials for each value of N (1S). | 1 | 1 |
| 23 | The average score is computed correctly. | 1 |  |
| 24 | The proportion of “short games” (score <=4N) is computed correctly. | 1 |  |
| 25 | The above statistics are correctly computed for the ph=0.75 case. | 1 |  |
| 26 | One figure plots expected score vs N (1C). It is titled and labeled (1S) and contains two curves (1C) corresponding to the ph=0.5 and ph=0.75 cases. | 2 | 1 |
| 27 | Another figure plots the “short game probability” vs N (1C). It is titled and labeled (1S) and contains two curves (1C) corresponding to the ph=0.5 and ph=0.75 cases. | 2 | 1 |
| GENERAL |  |  | Total = 10 |
| 28 | Script starts with a concise comment describing the program.  Function comment follows function header. |  | 1 |
| 29 | Code is sufficiently (but not excessively) commented. |  | 1 |
| 30 | Line lengths are not excessively long (80 columns).  NOTE: It's ok if a couple lines are a little too long, especially if they are print statements |  | 1 |
| 31 | No extra output (debugging output) produced |  | 1 |
| 32 | Proper indentation is always used. |  | 1 |
| 33 | Use meaningful variable names. Do not overwrite MATLAB keywords. **Note:** For this assignment, it is OK to overwrite the keyword “beta”. |  | 1 |
| 34 | Name important parameters as variables (constants). |  | 1 |
| 35 | No superfluous code (e.g., an empty if or else branch or a useless loop). Of course some students will have code that is awkward or unclear or inefficient. This point is specifically for not having code that does literally nothing. |  | 1 |
| 36 | Reasonably efficient code. |  | 1 |
| 37 | Does NOT put semicolon at wrong places, e.g., at the end of these lines: "if", "elseif", "else"," for","while", "function". |  | 1 |
| TOTAL |  | 29 | 19 |

**Penalties**

|  |  |  |
| --- | --- | --- |
| P1 | Student's code does not execute (or student provides a script when a function is required and vice-versa) | -1 from final score |
| P2 | Student's code crashes or does not terminate (infinite loop) for normal cases. | -1 from final score |
| P3 | All function headers and file names match those specified in the project description exactly. All input and output variables should be of the correct type. | -1 from final score |

**Grade Calculation**

|  |  |
| --- | --- |
| Total Possible Correctness Points | TC = 29 |
| Total Possible Style Points | TS = 19 |
| Student Correctness Points | C = min( \_\_\_ + 1 freebie point, TC) |
| Student Style Points | S = min( \_\_\_ + 1 freebie point, TS) |

Exceptions: If any file is missing/unacceptable, no freebie points can be applied to that file and subtract 3 style points for each missing/unacceptable file.

|  |  |
| --- | --- |
| Student’s final score | ([(C/(TC))+(S/TS)] X 5) - Penalties  (Out of 10; 1 decimal; no negative score; round to NEAREST) |