

# CS1112 Fall 2014 Project 3 Part A due Friday 10/3 at 11pm

(Part B will appear in a separate document. Both parts have the same submission deadline.)

You must work either on your own or with one partner. If you work with a partner you must first register as a group in CMS and then submit your work as a group. *Adhere to the Code of Academic Integrity.* For a group, “you” below refers to “your group.” You may discuss background issues and general strategies with others, but the work that you submit must be your own. In particular, you may discuss general ideas with others but you may not work out the detailed solutions with others. It is not OK for you to see or hear another student’s code and it is certainly not OK to copy code from another person or from published/Internet sources. If you feel that you cannot complete the assignment on your own, seek help from the course staff.

## Objectives

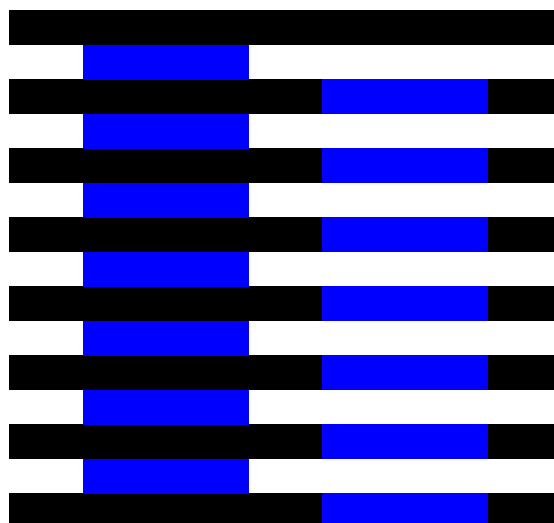
Completing this project will solidify your understanding of user-defined functions and vectors. Part 1 focuses on user-defined functions and graphics; Part 2 additionally involves vectors.

## 1 The Munker-White Illusion

Go to this website to check out the “Munker-White Illusion”:

<http://www.michaelbach.de/ot/lum-white/>

The website has an interactive graphic and gives a short description of the illusion. Use the slider next to the graphic to slide away the “grid rectangles” (black rectangles if you use the default graphic); you will see that the two stacks of colored rectangles (green in the default graphic) actually have the same brightness (luminance)! To the right is a MATLAB implementation of the Munker-White Illusion. The two stacks of blue rectangles appear to have different brightness—the illusion is more pronounced if you look at it from a distance—but in fact they were drawn using the same blue color.



You will explore the properties of the M-W Illusion by writing a function. If all we want to do is to draw one illusion, one could just write a script with the dimensions and colors “hard-coded”. To facilitate exploration, however, you will write a function whose parameters are the properties that we want to explore. For example, one may want to find out what combinations of color and rectangle dimensions would give a strong illusion and what values would result in a weak or ineffective illusion. Implement the following function as specified:

```
function mwIllusion(n, w, cg, cs, f, a, b)
% Display the Munker-White Illusion in the current figure window.
% The horizontal width of the diagram is w, and the rectangles that span
% the width of the diagram (the grid rectangles) is in color cg, where cg
% is a predefined color name such as 'k', 'b', ..., etc, or an rgb vector.
% Both the height of each rectangle and the vertical space between the
% rectangles are 1. There are n grid rectangles. The lower left corner of
% the diagram is at coordinates (a,b).
% There are two "stacks" of n-1 rectangles in color cs, where cs is a
% predefined color name or an rgb vector. The horizontal width of these
% rectangles is f*w, where f is a fraction less than 0.5. These two 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.
% Check parameter f: if f>=.5, set f to .3
```

You will then complete the script `showIllusion` to call function `mwIllusion` three times to produce three different illusions, placed side by side in one figure window. The first two should “show off” the illusion—give a strong illusion of differing brightness in the two stacks of rectangles—while the last illusion should be a weak(er) one. Try different combinations of parameter values!

## Specifics and hints

1. Start by reading *Insight* §5.3 (The Betsy Ross flag) first—it’ll help you with this problem!
2. Download the m-file `DrawRect.m` from the *Insights* page of the course website. Read the function comments and code. This function draws a rectangle with a black outline. In order to draw the M-W Illusion, you need to modify `DrawRect` to draw a colored rectangle without a black outline. This can be done simply by modifying the last statement in the file, which calls the built-in function `fill`, to be

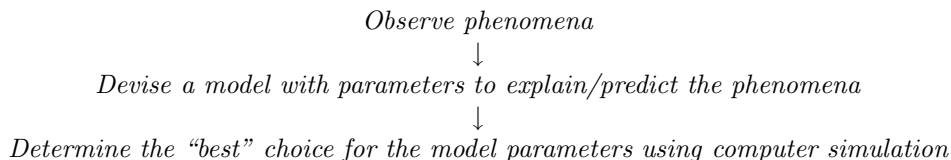
```
fill(x,y,c,'line','none')
```

Change the function name (and therefore filename) to `DrawRectNoLine`. Make effective use of `DrawRectNoLine` in drawing the illusion!

3. In function `mwIllusion`, use the command `hold on` at the beginning and `hold off` at the end.
4. In the provided (incomplete) script file `showIllusion`, the given code sets up a wide figure window for you to place three M-W Illusions in it side by side.
5. We have been using MATLAB’s predefined color names, e.g., ‘r’ for red. Instead of using a color name, we can specify the exact color vector, e.g., `DrawRect(0,0,1,1,[1 0 0])` will draw a red square because the “rgb vector” `[1 0 0]` specifies red (full contribution of red, zero green and blue). Take a look at this webpage that shows color swatches and their corresponding rgb values:  
<http://prideout.net/archive/colors.php> The last column of the table shows the values that should go into the rgb vector.

## Think about this . . .

Appreciate this problem as a metaphor. In computational science the sequence of actions



is typical. The problem-solving behind the design of `mwIllusion`—choosing the function parameters—essentially involves the same sequence.

Submit your files `mwIllusion.m` and `showIllusion.m` only. We will use our version of `DrawRectNoLine` when we test your code.

Part B of Project 3 will appear in a separate document.