

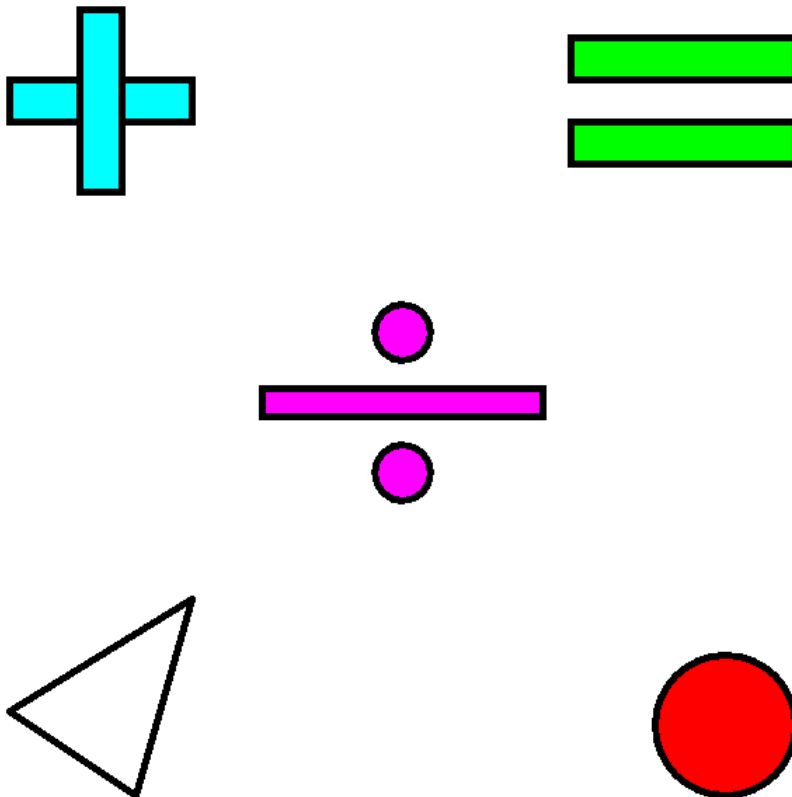
***CMPSCI 120
Extra Credit #1
Professor William T. Verts***

Setting Up

In this assignment you are to create a Web page that contains a client-side image map. This assignment does not build on any earlier assignment. For this assignment you will need the tools you used in earlier assignments (PuTTY or ssh, WinSCP or Fugu, Windows Notepad or Text Edit). You will need a graphics drawing package such as Mac PaintBrush (available from <https://sourceforge.net/projects/paintbrush/>) or Windows Paint. You may use other graphics packages if you feel more comfortable with them, as long as they can create .gif or .png images.

Part 1: Background for links

On the class site is a link to the following image, called **MyLinks.gif**:



The image is 600×600 pixels, and the background color (yellow) is set to transparent, so the objects in the image appear to be floating separately from one another. ***Download that image*** to your personal computer, and then use FTP (WinSCP or Fugu) to ***copy the image*** to your `public_html/cmpsci120` folder on the `elsrv3` server.

Part 2: Background Pattern

Create a new image in your favorite bitmap graphics package. This image must be **exactly** **8×8 pixels** in size. In this image create a random “speckle” pattern of your favorite colors. Here is an example. The full-sized image that you create is the tiny view on the left, but a “zoomed” version is shown on the right, where each colored square represents a single pixel. **Do not use** my pattern; create your own.



Save the image as **MyPattern.gif**. **Note:** you are allowed to save this particular image as **MyPattern.png** if you so choose, but throughout this document it will be referenced as a .gif file – you’ll have to make the appropriate substitutions in the code to make a .png file work correctly. Use FTP (WinSCP or Fugu) to **copy the image** over to your public_html/cmpsci120 folder on the elsrv3 server.

Part 3: The Basic Web Page

Telnet to the server (PuTTY or ssh), change into the public_html/cmpsci120 folder, and use emacs to create **MyLinks.html** (the file name must be exactly as specified or else you’ll lose points – we will use that name to link to your file for grading). Create the following Web page framework as replacing the blanks with **your first name**. Save it.

```
<!DOCTYPE html>

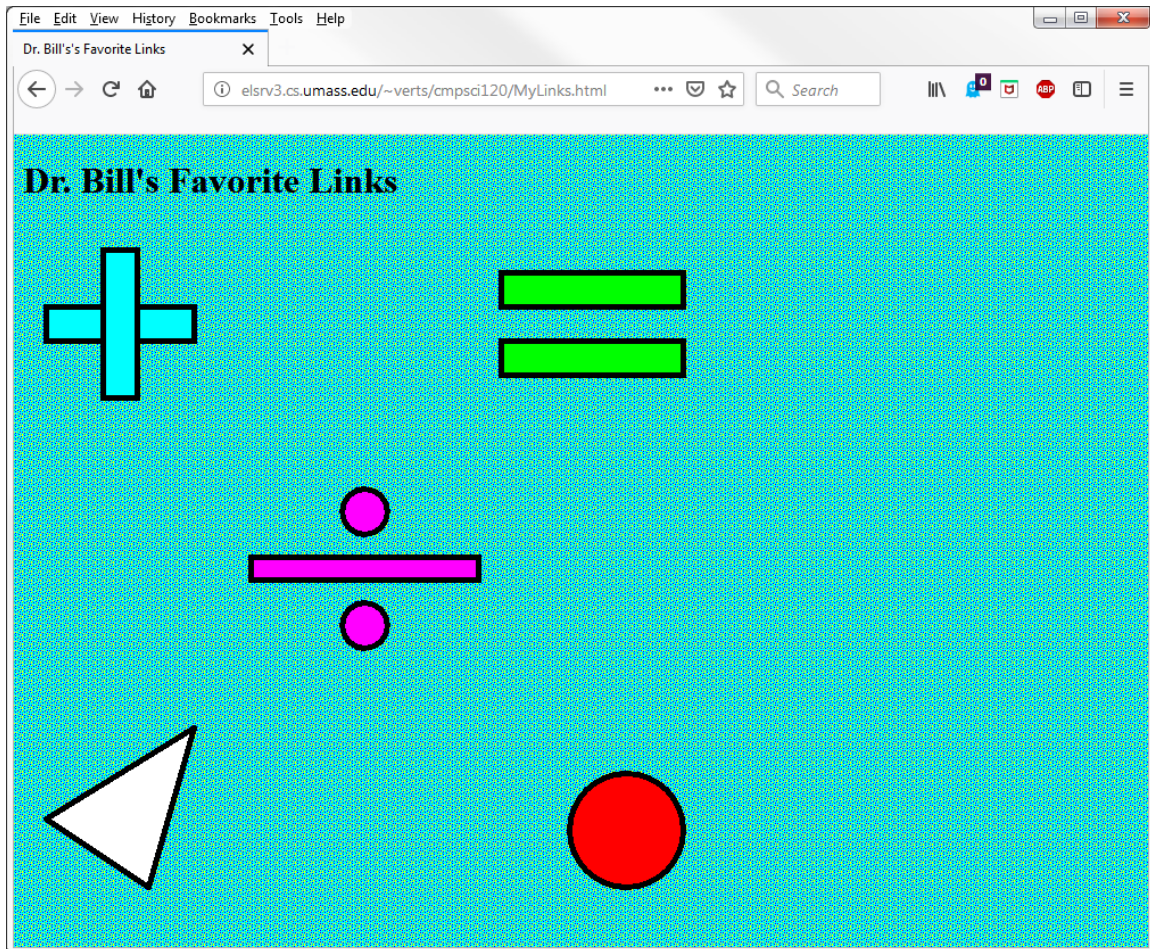
<HTML>
  <HEAD>
    <TITLE>_____ 's Favorite Links</TITLE>

    <STYLE TYPE="text/css">
      BODY {background-color:#0FF ;
            background-image:url('MyPattern.gif');}
    </STYLE>
  </HEAD>

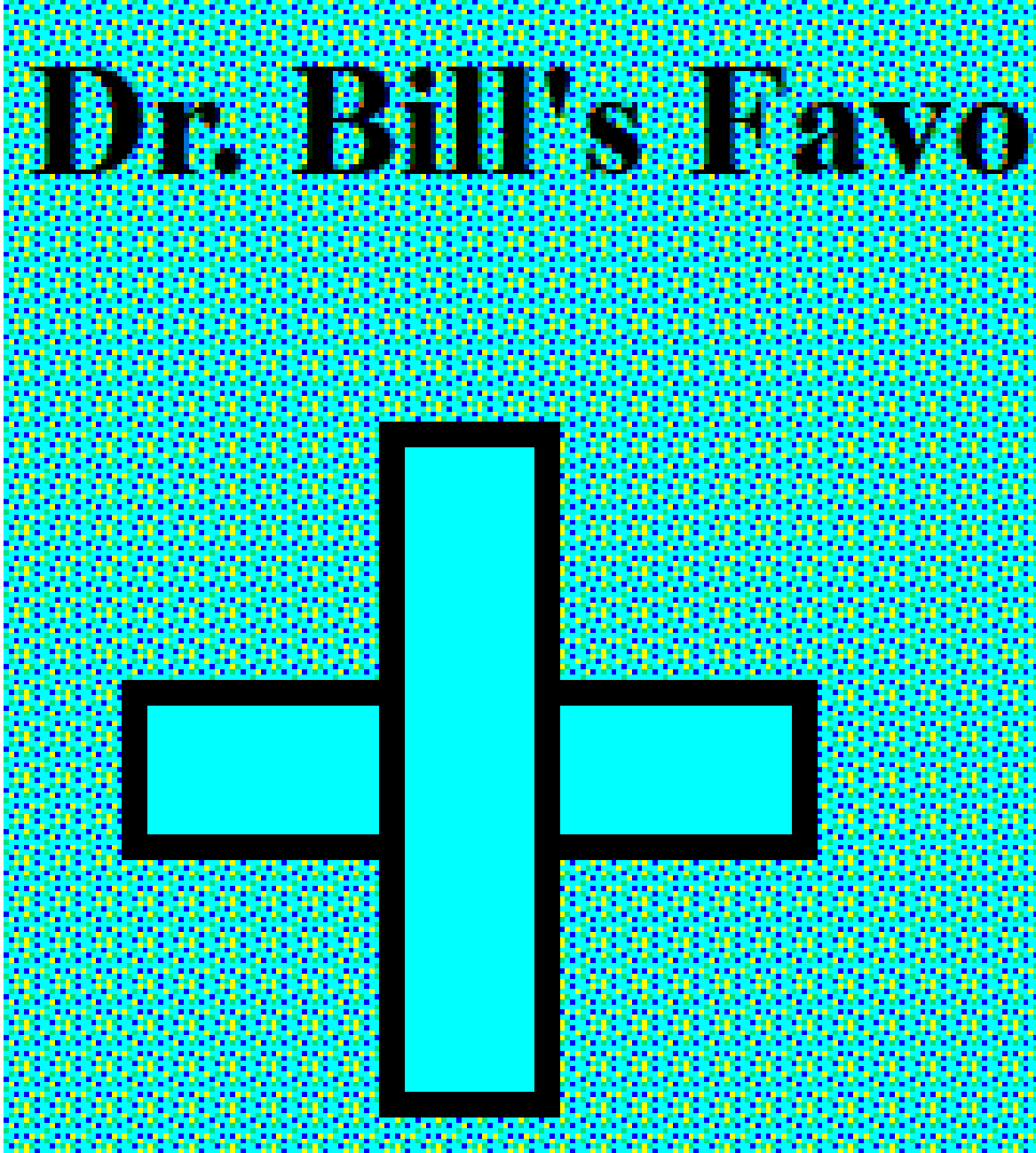
  <BODY>
    <H1>_____ 's Favorite Links</H1>
    <IMG SRC="MyLinks.gif" BORDER="0">
  </BODY>
</HTML>
```

Make certain that all your permissions are set correctly. All three files, `MyLinks.html`, `MyLinks.gif`, and either `MyPattern.gif` or `MyPattern.png`, should be present in your `public_html/cmpsci120` folder and should have 644 (`rw-r--r--`) permissions.

Load up your favorite browser and point it at the `MyLinks.html` file. Your URL will be `http://elsrv3.cs.umass.edu/~____/cmpsci120/MyLinks.html`, with your username in the blank. You should see the following appear in your browser, with your name instead of mine.



Here is a zoomed view of the upper left corner of the screen. Notice that the `MyPattern.gif` pattern file has been repeated across the width and height of the browser window, and that the background pattern shows through the transparent color of the `MyLinks.gif` image.



Do not proceed until your basic page works correctly as shown here. At this point, none of the pictures in the `MyLinks.gif` image actually do anything (the link-regions are not yet active). We will be making modifications to this file later, but for now you should be able to tell if everything done so far is correct.

Part 4: Building the Client-Side Image Map

The images are now complete, and the basic page should look OK.

Your next task is use emacs to modify the `MyLinks.html` file to **turn the image into a client-side image map** called `MyMap`, where each of the five pictures in the `MyLinks.gif` image links to a Web site. See page 186 in the Companion for information about client-side image maps.

There are five related regions in the image, indicated by their fill-colors (cyan, green, magenta, white, and red). Each links to a different Web page from Wikipedia, as follows:

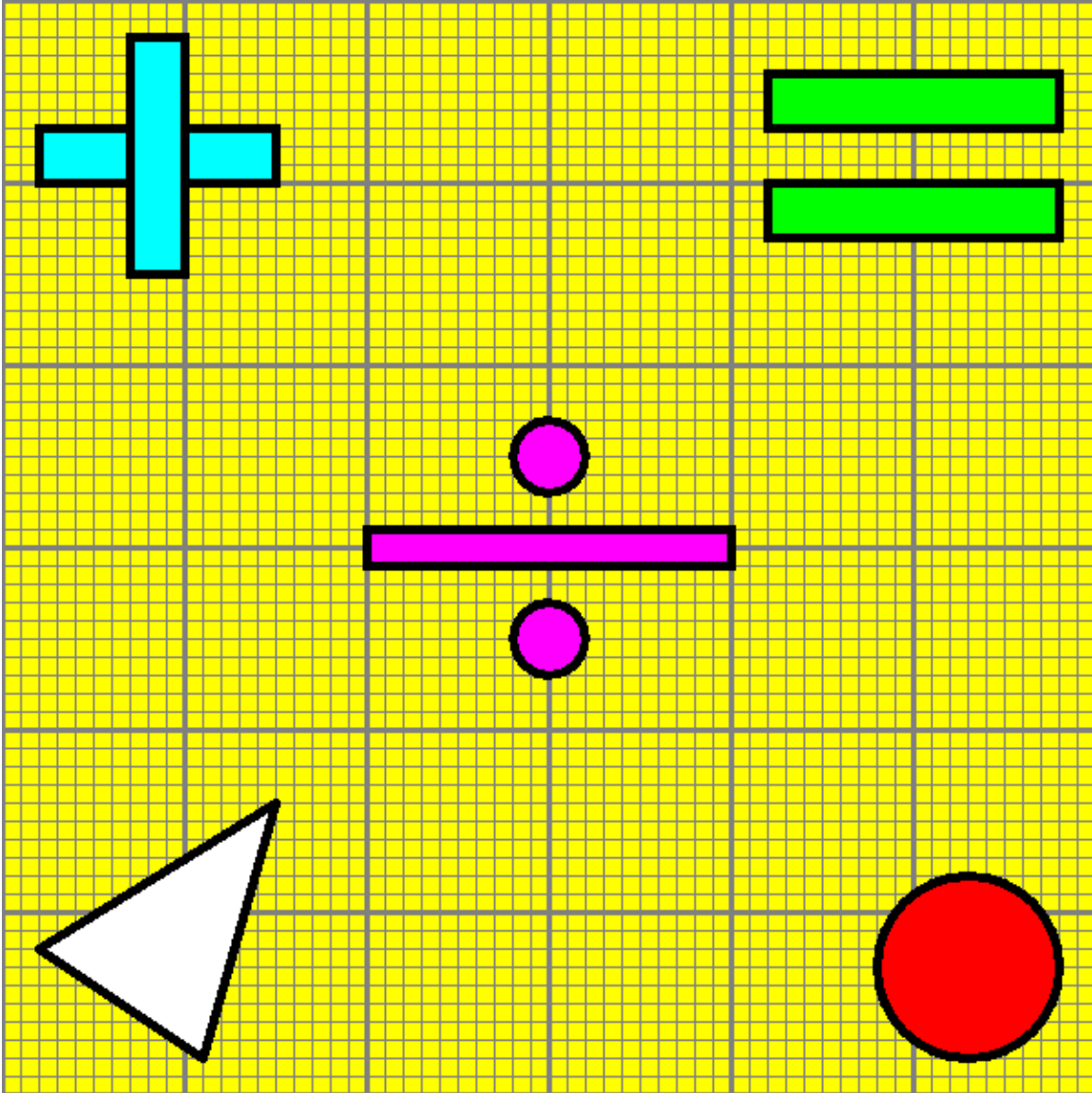
Cyan:	https://en.wikipedia.org/wiki/Addition
Green:	https://en.wikipedia.org/wiki/Equals_sign
Magenta:	https://en.wikipedia.org/wiki/Division_algorithm
White:	https://en.wikipedia.org/wiki/Triangle
Red:	https://en.wikipedia.org/wiki/Circle

The background of the `MyLinks.gif` image does nothing (i.e., there is no default link for the image map).

Your goal in creating the image map is to **exactly capture** the shape of each link-region in the `MyLinks.gif` image. That means that for each of the five pictures you are to capture just the outline of the link-regions, without allowing any of the background pattern to be used as part of the link.

Several of the link-regions consist of multiple objects, which must be captured as separate entries in the image map. For example, the green region consists of two separate rectangles, so the image map must contain two distinct rectangle entries, one for each rectangle, and each must link to the same Web site. The empty area between the two rectangles must not link to anything.

Here is a detail view of `MyLinks.gif` with a grid showing 10-pixel (thin lines) and 100-pixel (thick lines) coordinates. Use this image to determine the coordinates for each of the link-regions in the map. All coordinate values are multiples of 10 (that is, a point might be at location $\langle 40, 30 \rangle$ but will never be at $\langle 43, 32 \rangle$). Remember that location $\langle 0, 0 \rangle$ is the upper left corner of the image.



When your `MyLinks.html` page is complete, **test it in the browser**. As you float the mouse over the `MyLinks.gif` image, there should be no active link in any portion of the page except in the link-regions of the five pictures, and clicking in any valid link-region should jump to the appropriate Web site. No portion of the background pattern should link to anything.

What To Turn In

When your page is correct and visible on the Web, send an email to the **literacy@cs.umass.edu** account. The subject line must be set to the exact phrase **COMPSCI 120 EXTRA CREDIT ASSIGNMENT #1**, and the body of the message must contain your name, your username, and your fully specified URL.

Grading

We will be grading the assignment based on whether or not it works, as well as on how closely you follow the directions. You will lose points if any of the files have the wrong name, if the background pattern does not show up, if clicking outside of the five link-regions (i.e., on any portion of the background pattern) jumps to any site, if any part of an object does not link to some site, if the background pattern image is the wrong size, or if there are any errors in the HTML code.

Assuming that when we click on **MyLinks.html** we actually see something, you'll start with 10 extra credit points, and the following infractions reduce that score:

1 point: **MyLinks.html**

A: -1 for syntax errors or bad indentation in HTML code

3 points: **MyPattern.gif** or **MyPattern.png** (8×8 Background Pattern)

B: -3 for no pattern at all, otherwise:

C: -1 for wrong file name

D: -1 for wrong size (must be 8×8)

2 points: **MyLinks.gif**

E: -2 for no image at all:

4 points: Client-Side Image Map

F: -4 for no image map at all, or does not work, otherwise:

G: -1 for naming the image map something other than MyMap

H: -1 for an invented **SHAPE** (such as `SHAPE="triangle"`, which doesn't exist)

I: -1 for improper coverage (active link in region outside an object or dead region inside)

J: -1 at least one region links to the wrong URL

1 point: General

K: -1 for **MyLinks.html** exists but never claimed (no email to literacy)