E-Portfolio Instructions

PART 1: Create an Image from Scratch

In the first part of the project, you will start by making a default image of size 256x256 and calculating a value for each RGB component (a value for red, a value for green and a value for blue).

Your task is to write your own code in the style of the examples below

1. Create an image of size 256x256 and use each pixel's location as part of an arithmetic expression to generate red, green and blue values; creativity is encouraged! Save your image so you can add it to your web page for this part of the assignment (see more details about web pages in Part 4).
2. Copy your final version of your JavaScript program that generated your image into your web page.
3. Write a description of the algorithm you develop to share with others and add that to your web page, too.

Each pixel in the image must be set to some color value based on the pixel's location (its x and y coordinates in the image); beyond that, you have enormous flexibility to be creative! See examples below for ideas.

Create at least one image to complete this part of the assignment, but feel free to create more than one. Really interesting projects with clearly described anticipated results (as indicated in the description of the algorithm) may be chosen to demonstrate instructor feedback or shown as an example for future students!

We provide several examples to help you understand how to create images. You can use these examples as models and as inspiration in completing this part of the project:

Arithmetic Image Examples for Part 1

PART 2: Modify an Image

In the second part of the assignment, you will modify an existing image with a different algorithm than you developed in Part 1.

Pick any digital image you want to start with. It could be an image of your own or you can find an image on the web. Your task is to write your own code in the style of the examples below:

1. Modify an existing image of any size by creating a different algorithm than you used in Part 1 and applying it to the image; keep a copy of both the original image file and the modified image to add to your web page for this part of the assignment (see more details about web pages in Part 4).
2. Copy your final version of your JavaScript program that generated your image into your web page.
3. Write a description of the algorithm you designed and add it to your web page.

In this case, not every pixel from the original image may be changed as you might want to have parts of the original image still show. We suggest you include comments in your algorithm description about the parts of your code that result in the interesting effects. See examples below for ideas.

Create at least one image, but feel free to create more than one. Again, really interesting projects may be chosen to demonstrate instructor feedback or shown as an examples for future students.

We provide several examples to inspire you here:

Modified Image Examples for Part 2

PART 3: Hide an Image in Another Image—Steganography

You learned how to hide and extract an image in another image by hiding the image in half a pixel. For the third part of the project, you will modify your code to hide one image in another image with a different number of bits.

Pick two images and make sure they are the same size. You may need to crop one of them to make sure they are the same size.

Your task:

1. In the first Steganography code you wrote, the four left-most, or most significant bits of the original image were maintained and the four right-most bits were used to hide an image. In this version you should keep the six most-significant bits and hide the image in the two right-most bits. Keep at least one copy of both original images after cropping, as well as a copy of the combined image to add into your web page for this part of the assignment (see more details about web pages in Part 4).
2. Then modify your steganography code to extract the hidden image, which is taking up the right-most 2 of the 8 bits for every RGB component. Keep a copy of the extracted image for your web page as well.
3. Write a written description of the algorithm you created to solve this problem and add it to your web page.
4. Discover the hidden message inside of the lower two bits of this **picture** and write it at the end of your web page.

See the a demonstration of hiding an image in 2 bits here:

[Steganography Examples for Part 3]

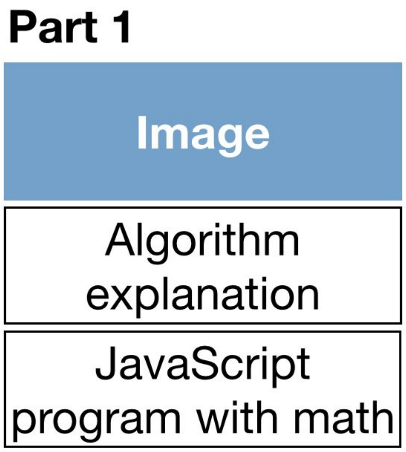
PART 4: Completing the E-Portfolio

Now it is time to show off your work by putting the images you have created and the code you have written on web pages.

Your tasks:

1. Create a web page for the first algorithm you created in Part 1 and include on the page:
    a. The image you created from scratch
    b. A description of the algorithm you designed
    c. and the JavaScript program code
       You can cut and paste the code and put it between the two tags <pre> and </pre> to give it a similar look to the coding environments.

You might lay out the content in this way on a web page; clear title and labels are highly recommended! It is in your best interest to make it easy for your peers to understand your work!



Part 1

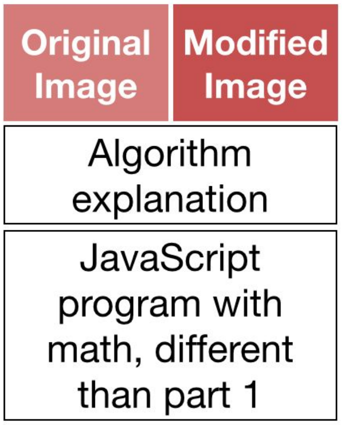
Image

Algorithm explanation

JavaScript program with math

2. Create a web page for the second algorithm you created in Part 2 and include on the page:
    a. Your original image you started with and the modified image
    b. A description of the algorithm you designed, remembering that it should be different than the algorithm used in Part 1
    c. and the JavaScript program code

Part 2



Original Image

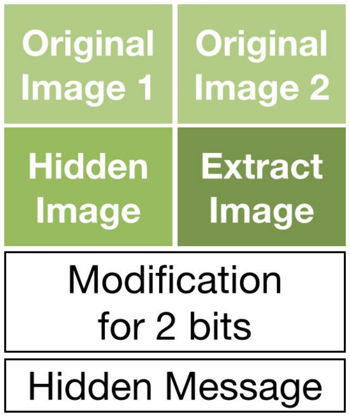
Modified Image

Algorithm explanation

JavaScript program with math, different than part 1

3. Create a web page for the steganography program you wrote for Part 3 and include on the page:
    a. Four images:
        i. Two original images of which you plan to hide one image in the other image
        ii. Your combined image that has the other image hidden in it with the hidden image hiding in the lower 2 bits
        iii. The extracted hidden image
    b. An explanation of how you modified your code to hide an image in 2 bits
    c. The hidden message found in the provided picture above that is hiding in the lower two bits

Part 3



Original Image 1

Original Image 2

Hidden Image

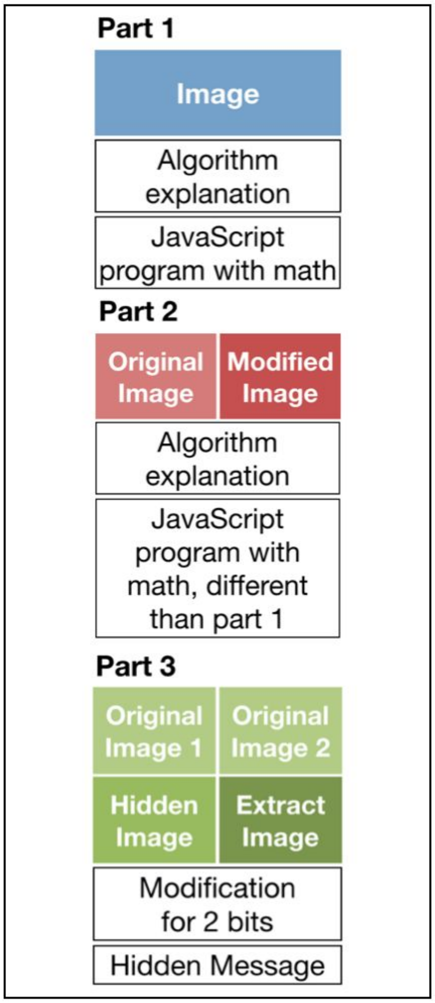
Extract Image

Modification for 2 bits

Hidden Message

4. Create a blog on Codepen.io and connect to the three pages for 1-3.

Part 4



Here is a sample e-portfolio that Susan constructed:

[Example E-Portfolio]

Review Criteria

You will submit your e-portfolio by copying and pasting the link to your codepen.io blog into a textbox. Reviewers will look at your e-portfolio to confirm that all the required parts are complete.

Basic Requirements Checklist

- Did you provide a link to your e-portfolio web page on CodePen?
- Did you include a web page to showcase the image you created from scratch using math and include on that page:
    a. the image
    b. a description of the algorithm that includes arithmetic expressions or math functions
    c. and the JavaScript code
- Did you create a web page to showcase the modified image you created and include on that web page:
    a. the original image
    b. the modified image
    c. a description of the algorithm that includes arithmetic expressions or math functions, which is different from that used in Part 1
    d. and the JavaScript code
- Did you create a web page to showcase your steganography work and include on that web page:
    a. your two original images
    b. the image you created that has another image hidden inside the lower two bits of each pixel
    c. the extracted image
    d. a description of your modification of the steganography algorithm
    e. and the hidden message in the provided image

Please remember, all descriptions, program code, and feedback should be provided in English.