CSC 111 Assignment 5

NOTE: Your programs must compile and execute using the Eclipse environment in ECS 242. If you do your work on your own computer, be sure to test it in ECS 242 before you submit it.

Programming instructions

Assignment 5 consists of two parts. Each part requires a separate C program in a separate .c source file and the files must be named as per instructions below. In Part I, you generate randomized CSC 111 art similar to the image in Figure 1 below. For Part I submit you're the source code (i.e., .c file) and your best art (i.e., html file). In Part II, you write a set of array manipulation routines. These Part II routines are typical Midterm 2 and final exam questions. Part 1 and Part I each are worth 50% of the A5 mark.

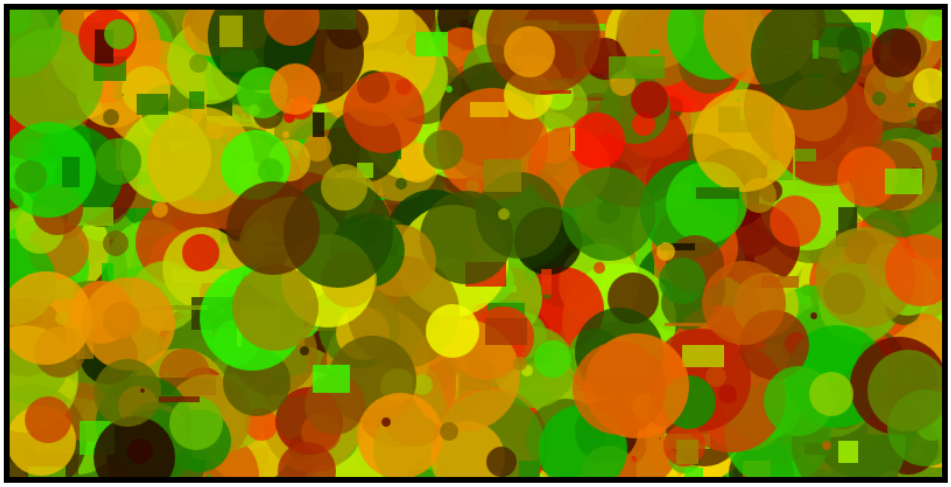

Figure 1: CSC 111 Art

Part I

A syntactically correct C program has been provided in the file `A5P1Template.c` (posted on CourseSpaces). Read through this program carefully before starting.

Your task is to implement the following four functions in the Part I template:

- `Main()` – Open an HTML/SVG file called the `myCSC111art.html`, call the function `genCSC111Art`, and close the file. To generate a different piece of art every time the program is run, seed the random number generator using the system clock.
- `Circle()` – Outputs an SVG `circle` tag to the file to draw a random circle
- `Ellipse()` – Outputs an SVG `ellipse` tag to the file to draw a random ellipse
- `Rectangle()` – Outputs an SVG `rect` tag to the file to draw a random rectangle
- `Square()` – Outputs a square using the SVG `rect` tag to the file to draw a random square
- `genCSC111Art` – Generates up to 1000 randomly chosen shapes each with random positions, sizes, colours, and opacities.

- The ranges of the random numbers should fit the variable—positions and sizes should "fit" the canvas range, the three RGB values (i.e., red, blue and green) of the colours must be constrained to ranges between 0 and 255, opacities must be in the range 0.0 to 1.0. The min and max values for all ranges and all other constants or literals must be defined at the top of the program using `#define` preprocessor directives.
- Output the characteristic values of the first 10 shapes to the console as depicted in Figure 2 below; one line per shape.
- Develop this program incrementally one step at a time.
- Play with the constants to make your image aesthetically pleasing.
- Note proper decomposition of your program is critical. Recommended structure of the program: (1) include files and constant definitions (many); (2) prototypes of all the functions used in the program; (3) main program; (4) functions for generating random numbers (i.e., integers and floats); (5) functions for generating appropriate random numbers for drawing a circle, rectangle, square and ellipse; (6) functions for generating SVG code for drawing a circle, rectangle, square and ellipse; and (7) a function for controlling the frequency of shapes (not required).

```
 1 Squ: (x,y,s)           (r,g,b)       op
 1 Squ: (972,115,  9)     (255,150,  0) 0.77
 2 Rec: (x,y|w,h)         (r,g,b)       op
 2 Rec: (198,290| 11, 19) (255, 21,  0) 0.73
 3 Rec: (x,y|w,h)         (r,g,b)       op
 3 Rec: (953,249| 18, 25) (255,167,  0) 0.99
 4 Ell: (x,y|rx,ry)       (r,g,b)       op
 4 Ell: (574,129| 25, 17) (255,176,  0) 0.99
 5 Squ: (x,y,s)           (r,g,b)       op
 5 Squ: (456,198, 15)     (255, 36,  0) 0.81
 6 Cir: (x,y,r)           (r,g,b)       op
 6 Cir: (120,292, 19)     (255, 97,  0) 0.98
 7 Cir: (x,y,r)           (r,g,b)       op
 7 Cir: (523,414, 17)     (255,139,  0) 0.95
 8 Rec: (x,y|w,h)         (r,g,b)       op
 8 Rec: (240,438| 10,  5) (255,121,  0) 0.70
 9 Squ: (x,y,s)           (r,g,b)       op
 9 Squ: (331,145, 15)     (255, 60,  0) 0.78
10 Cir: (x,y,r)           (r,g,b)       op
10 Cir: (367, 32, 11)     (255,168,  0) 0.92
```

Figure 2: Console Output for Shape Variables

Part II

A syntactically correct C program has been provided in the file `A5P2Template.c` (posted on CourseSpaces). Read through this program carefully before starting. You are to complete the set of array manipulation routines outlined in the template. Develop this program incrementally one function at a time. Note the structure of the program: (1) include files and constant definitions; (2) prototypes of all the functions used in the program; (3) main program; (4) functions to test and exercise the functions you are writing; and (5) stubs for 18 functions you are writing. For most functions you have two write two lines of

code. The main program calls four different functions which exercise array functions and test code as follows:

```
// setVectors();
rotateVectors();
// findVectorValues();
// addMulVectors();
```

When you develop the code for 18 functions, comment out three of the calls in the main program. In other words, develop your functions incrementally. Maintain a syntactically correct program while developing your functions.

Assignment submission instructions

CSC 111 assignments will only be accepted electronically through the assignment page on the CSC 111 CourseSpaces site. Your submission will consist of ***two C source files*** and ***one HTML file*** named by the following convention: If your student ID is ***V00123456***, your C source files for parts 1 and 2 (respectively) must be named **V00123456A5P1.c, V00123456A5P1.html and V00123456A5P2.c**. In addition, your ***full name, student ID,*** and ***Assignment name*** (e.g., Assignment 5) must appear in a ***comment section at the beginning of each C program***.

For example (in Part 1) – not required for the HTML file:

```
Name: Polar Bear (Replace this with your name)
UVicID: V00123456 (Replace this with your student number)
Date: 2017/10/25 (Replace this with the date you wrote the program)
Assignment: A5
File name: V00123456A5P1.c (Replace V00123456 with your student number)
Description: This program generates CSC 111 Art.
```

Please submit only the source file and not the executable file. To verify that you have submitted the correct file, you are strongly encouraged to download your submissions from the site and test that they work correctly in your Eclipse environment. Submissions that do not follow the guidelines above will receive a mark of zero.

Since this assignment only requires three source files, CourseSpaces will only allow you to submit three files. However, until the assignment due date, you may change your submission by deleting and resubmitting your source files multiple times. After the due date, no submissions will be accepted. When grading is complete, your assignment mark and comments will appear in the ***Gradebook*** section of CourseSpaces.