Programming Assignment 4 – Option B
COSC 051, Fall 2011
27 OCT 2011

Goal:
Implement graphics and animation on a text screen.

Assignment:
Write a C++ program that draws various geometric shapes on a text screen. Include animation capabilities that "move" the shapes across the screen.

For this assignment, consider the terminal to represent a drawing window that has 26 rows and 81 columns. Your program will allow the user to specify a shape to draw. Depending on the selected shape, the program will then request that the user enter parameters necessary to describe the shape. For example the specification of a circle would require the user to provide the coordinates of the center of the circle and the radius. Your program should recognize if the values provided place the selected shape "off screen" and should react accordingly. Further, your program should use a two dimensional array as a "buffer" for the graphics screen. Your calculations will first populate this buffer with data representing the shape. Once all calculations are complete, the program will output the image to the screen.

Your program must include the following shape options:
- Circle
- Line
- Extra Credit: Create a function that calls your line drawing function multiple times to draw a triangle

An example of the program output for a circle with center at Column = 60, Row = 14 and radius of 7 follows:
The previous screen capture shows the starting position of the circle on screen. The animation portion of your program should simulate moving the shape across the screen in a sequence of steps as shown below.

Successive images of circle as it scrolls across the screen from right to left

The skill set required for this assignment includes:
- Looping constructs
- Control structures
- Algorithm design
- Multi-dimensional array variables
- Error checking

One approach to "drawing" shapes in an array buffer is to use a formula for the shape under consideration. For example to plot a line that begins in \( col_1, row_1 \) and ends in \( col_2, row_2 \) we could use the slope/intercept formula of a line as follows:

\[
y = mx + b
\]

or,

\[
row = m(col) + b
\]

where,

slope is given by:

\[
m = \frac{row_2 - row_1}{col_2 - col_1} \quad \text{when} \quad (col_2 \neq col_1)
\]

and

\[
y\text{-intercept is given by:} \quad b = row_1 - (m(col_1))
\]
Of course vertical and horizontal lines are special cases. However, in general, we can consider each array element to be a point in space. Using the value of the column index, and the equation for a line from above, we can determine if the "point" actually falls on (or close enough) to the line.

The above description is a relatively simple, but perfectly acceptable approach. A more sophisticated algorithm can be found at: 
http://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm

Similarly, given the coordinates for the center of a circle and its radius, we can determine if any "point" in the array falls on (or close enough) to the circle. For example, consider a circle with center \((a, b)\) and radius \(r\). For any "point" \((col_n, row_n)\) in the array, we can conclude that the point falls on the circle if:

\[
r \approx \sqrt{(a - col_n)^2 + (b - row_n)^2}
\]

As before, this represents a relatively simple, but perfectly acceptable approach. Ideas for more sophisticated solutions can be found at: 
http://en.wikipedia.org/wiki/Bresenham%27s_circle_algorithm

◆

This graded assignment is worth 100 points and will be counted as part of your programming grade for the course.

The product that you submit must be your own work. Collaboration is allowed as specified within the syllabus for this course. For this assignment, you are not required to submit an acknowledgement statement.

Your programs must be posted to Blackboard no later than 11:59 pm on Thursday, November 3, 2011.