Background

Blue Ridge Computer Games, Inc. is very pleased with the first version of screen graphics for the tic-tac-toe game board. It is time to move to the next phase of development and add the “X’s” and “O’s” that will mark the moves made by each player during the game.

Additionally, the project director has indicated that performance is an issue with the previous design. To address this, the software will be re-engineered to incorporate the following design changes:

1. All code shall be moved out of function main() and into user-defined functions. The only code remaining in function main() should be control structures and function calls.
2. All computations and “drawing” actions shall be done behind the scenes in a buffer. Once the buffer is completely updated with changes the new view shall be transferred to the terminal window.

An example of what the game board might look like with two moves displayed is shown below.

![Figure 1 Game Board](image-url)
Additional Guidance

The “O” shown in Figure 1 is essentially a circle having a radius of 1 and its center at the center of the top left board cell. Your program should implement a generic circle drawing function. The function should accept as parameters, the column and row value that represent the center of the circle, the circle radius, and the character to use to draw the circle. Given the coordinates for the center of a circle and its radius, we can determine if any "point" on the terminal screen falls on (or close enough) to the circle. For example, consider a circle with center \((a, b)\) and radius \(r\). For any "point" \((col_n, row_n)\) on the screen, we can conclude that the point falls on the circle if:

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

This represents a relatively simple, but perfectly acceptable approach. Ideas for more sophisticated solutions can be found at:


The “X” shown in Figure 1 is simply the intersection of two lines. We can define a “radius” of the “X” such that it becomes trivial to compute the end points of the two lines. Your program should implement a generic line drawing function that uses these end points to draw the lines that will form the “X”. 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:

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

and y-intercept is given by:

\[
b = row_1 - (m(col_1))
\]

Of course vertical and horizontal lines are special cases. However, in general, we can consider each screen 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](http://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm)
Programming Skills

The programming skills, in addition to those from previous projects, required to complete this assignment include:

- User defined functions
- Arrays as function arguments
- Geometric shape drawing techniques

Submission Details

What to submit: One C++ source code file. The file name should be <netID>P4.cpp.

Due date/time: Wednesday, 21 MAR 2012, no later than 11:59pm

Point Value: 100 points

Include the following comments at the start of your program:

/**************************************************
*                                                *
* <FileName>.cpp                                 *
*                                                *
*  COSC 051 Spring 2012                          *
*  Project #4                                    *
*                                                *
*  Due on: MAR 21, 2012                          *
*  Created on: MMM DD, 2012                      *
*  Last edited on: MMM DD, 2012                  *
*  Author: <your name>                           *
*  netID: <your netID>                           *
*                                                *
**************************************************/

Academic Integrity

This is an individual project and all work must be your own. No documentation of references is required for this project. Otherwise, you must follow all guidelines specified in the Academic Honesty section of this course syllabus.
Grading

This graded assignment is worth 100 points and will be counted as part of the Programming Projects category for the course. Points are allocated as follows:

<table>
<thead>
<tr>
<th>Category</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error checking and data validation</td>
<td>5</td>
</tr>
<tr>
<td>Comments</td>
<td>5</td>
</tr>
<tr>
<td>Functions correctly implemented to remove most code out of main()</td>
<td>20</td>
</tr>
<tr>
<td>Circle drawing function correctly implemented</td>
<td>15</td>
</tr>
<tr>
<td>Line drawing function correctly implemented</td>
<td>15</td>
</tr>
<tr>
<td>Output (game board accurately drawn given variable center column and row data)</td>
<td>10</td>
</tr>
<tr>
<td>Use circle drawing function to draw a “O” in at least one cell</td>
<td>10</td>
</tr>
<tr>
<td>Use line drawing function to draw an “X” in at least one cell</td>
<td>10</td>
</tr>
<tr>
<td>File naming conventions are followed exactly</td>
<td>5</td>
</tr>
<tr>
<td>Correct file, in correct format, is posted to Blackboard</td>
<td>5</td>
</tr>
</tbody>
</table>