Calvin Liu

804182525

Report

1. Most of the obstacles:

-Off by one errors

-How to check if a string was syntactically valid

-The accuracy of the circle function

-The swap algorithm threw me off in the flip function

-Making sure if plotted characters in bound and not those that are out of bound, and not producing the error message

2.

eightWordIsWrong:

check if the 6 digits are all digits using a passed in beginning parameter to indicate where you’re at. If it’s wrong return the position

tenWordIsWrong:

same concept as eightWordIsWrong

textCommIsWrong:

same concept as the above functions, except need to check if the second delimiter is there.

flipIsWrong:same concept

\*\*All functions must only setChar if the coordinate is on the grid

plotHorizontal:

just make a loop adjusting the columns not the rows

plotVertical:

just make a loop adjusting the rows not the columns

plotRectangle:

Combination of the vertical and horizontal line

plotCircle:

If the diameter was positive odd, I increased it by 1 by using the % to check

Loop, changing theta as it goes around the circle like polar coordinates

Cast the radius to an int and calculate using cmath functions

int rowPosition = static\_cast<int>(((radius \* cos(theta)) + .5));

int colPosition = static\_cast<int>(((radius \* sin(theta)) + .5));

int yPosOffset = r+rowPosition;

int yNegOffset = r-rowPosition;

int xPosOffset = c+colPosition;

int xNegOffset = c-colPosition;

This will calculate the x and y for each loop

setChar(yPosOffset, xPosOffset, ch);

setChar(yPosOffset, xNegOffset, ch);

setChar(yNegOffset, xPosOffset, ch);

setChar(yNegOffset, xNegOffset, ch);

Will plot the x and y calculated for each loop

Use this algorithm, which is based off polar coordinates to sweep out a circle of radius = diameter/2

clearPicture:

just loop through the grid in the row and columns, making each coordinate a space

invertRectangle:

loop through the whole grid using 2 for loops

if the coordinate is on the grid,

make the character at the spot, if it is the one asked into a space

if the coordinate is on the grid

make the character at that spot into the character asked if it was a space

flipHorizontal:

for each row

loop through 0-half the width, exchanging the last and first letter and moving inward

Moving inward would require a beginning+something and end-something and that something will be increased for each loop

inGrid:

Loop through the whole grid, asking if the character is there

Executecommands:

Make all the letters lowercase

Loop through the whole entire string

Look at the first letter and sort it into certain cases. Once it enters a case, you need to convert the numbers into digits while parsing it

Then do the function that is under the case and it will do it on the grid

Increase the position that you are at with the length of the string that matches with the function

\*\*The cases are in a loop and will keep going until it reaches the end of the string. The cases will increase the point. If it is syntactically valid, then it will end where it is supposed to.

3. Test Cases:

-If the command was short just 1 short such as r04050607

-If the command was truncated r04050607tt1011

-If the command was had a wrong character r04050607gl

-if the text delimiter did not have something that closed it

-If the text delimiter was inside the string

-If the delimiter showed up 3 times in the code

-If 2 plotText function had 2 of the same delimiters in different places

-If the command was missing a number r0450607t

-If the picture goes out of bounds

-If there is a character where an integer should be

-If the text has the delimiters next to one another

-Multiple inputs with one run

Some of the test cases used:

01020304r

r40050607\*

t0405\_PPPPjfnkjfn\_fmv

t0405\_PPPPjfnkjfn\_

h040140\*

i04050706\*

C070611$X

Xz

zX

T0705/Hat/C070611$R051003049

T0705/Hat/C070611$fR051003049

r050t

t0405 Magic! f04050106

t0405 Magic f04050106

t0405 Magic f04050006

t0405 Magic f04050105

t0405 Magic f04050104

t0506\*rjgnjrn\*r04050607Pc101010d