Jiachen Liu

404337478

Report

1. Notable obstacles:

When trying to implement the process, trying to account for all the possible situations is really hard to implement. I tried to write a function examine whether a 1,2 or 3 characters part of a string is a valid command or not. But after an hour trying to figure it out, I don’t think it’s possible to implement that function, so I try to account for as many possible syntax error cases as possible.

A lot of the checking bugs cases contradict each other so I tried to use debugger to figure out when there’s a compile error in my code.

Also the p3tester doesn’t work so well for me so I spent a lot of time on that too.

2.

**A design of my program:**

Declare some important constants, like FG, BG, HORIZ, VERT.

Mention some function that will be used in this project.

Mention the Boolean isValid, which is really important in my code (instead of return 2, to prevent the program from immediately exiting)

int main()

{

The main function that the professor gave us.

}

bool plotline(some variables)

{

I first accounted for all the successful situation:

if (direction is valid)

{

if (foreground&background is valid )

{

if(the plotChar is valid)

{

if (the existing row and column is valid)

{

use a switch statement for each condition. There are four conditions in my case.

If (foreground)

Swtich (dir)

0: if distance>0, do column++, if distance <0, distance=abs(distance), column=column-1

1: if distance>0, do row++, if distance <0, distance=abs(distance), row=row-1

if (background)

switch (dir)

0: if distance>0, if empty: do column++, if distance <0, if empty: distance=abs(distance), column=column

1: if distance>0, if empty: do row++, if distance <0, if empty: distance=abs(distance), row=row-1

}

if any of them is invalid, return false for this function

}

}

}

}

int executeCommands(declare a bunch of variables. All of them being reference parameters except the string command)

{

declare some variables I use in this function:

length of the string

r1 (row position)

c1 (column position)

int (someothervariables);

the pointer also points to some parameters in the int main function.

for (i increment to size )

{

if the previous char is H/h or V/v (won’t mention the case after this one), if this char is +, declare this as invalid

}

for (i increment to size )

{

take account for H-, without a digit after ‘-’ problem

}

for (i increment to size )

{

takes care of V-, without a digit after ‘-’ problem

}

for (i increment to size )

{

takes care of a number after C problem

}

for (i increment to size )

{

takes care of a space after H V F or B

}

for (i increment to size )

{

take care of cases like ‘-number-’

}

for (i increment to size )

{

take care of the first character is not H V F B or C

}

I feel like I’m boring you, so I’m going to stop right here. There are a lot of other cases I took care of.

//now plotting stuff

for (increament i)

if H

{

declare dir

take distance from the string

take account if out of bound situation

}

for (increament i)

if V

{

declare dir

take distance from the string

take account if out of bound situation

}

for (increament i)

if B

{

set current mode to background mode

}

for (increment i)

if F

{

set current mode to foreground

}

for (increment i)

{

if C, cleargrid;

}

if valid, return 0;

if invalid, return 1

if out of bound, return 2

else return 3;

}

that’s the pseudo code

3.

**(by works I mean output the expected result)**

test result:

H1 works

V21 works

H21 works

CH21 works

H15V10B$H2 works

H15V10F$H2 works

F#H+25H?V3! works

B@H works

C12 works

H12V12C works

H12V12CH12 works

Q3V4# works

V03C H123# works

H12V3H-5 works

For H18H-123#,

I think the program can also interpret it as H-123 as one command, which makes it a outofbound error.

V-1 doesn’t works

H-1 doesn’t works ---//now it works after I fixed, here’s something that doesn’t work:

H—doesn’t work, it output badposition as 1 instead of 3.

It’s probably because it overlaps with some lines before it, and since I put as the lefiest one, the most left one might be 1, rather than 3.