CSC3150
AS#2 Report

Hajun Lee
117010437

“Frog crosses the river”

• How did you design your program?

First, I used 2-demention list to store 10 X 50 unit map. There are 9 logs in the middle in the list of total. After that, I used struct called ob_log{} to create object of logs. Each object should have direction, length and y. (y : index of the left edge point of log)

I used rand() function to generate of logs. Each movement of log should control by the thread which need to create minimum 9 child threads.

As a frog, I defined G_Status and BRIDGE_TH(Threads) variables. I extended logs_move to move frog. In this function, I used while true loop for refresh map and control log of y and then, I used a mutex-lock to start thread change a map. One of the big reason or purpose is that not to see the old version of map again.

I used a for loop to process each block of the row and if y is bigger than 0 or less than column -2 – length, check block index range and if these all satisfied, I set this change to content of the block = and if not, set ‘ ’. Used list to save and then put parameter when create threads. If the numbers are same, find location of the frog first and if block is ‘ ’, let G_Status be -1 or change it to 0. Find frog location, direction and y. If y smaller than 0 or column -1, set 0 or column -1

Now let’s move on to the keyboard section. I used while loop again with kbhit() function whether to check user enter the keyboard and used getchar() to get the key what user enter. Used case of ‘a’, ‘s’, ‘d’, ‘w’ and ‘q’ to move it like x,y and for ‘q’, make G_Status to -2.

Finally, wait for the other child thread, used pthread_join and if status is 1, user win. If -1, user lost, if -2, should be quit.

• The environment of running your program.

Ubuntu 16.04.5 LTS / gcc Ubuntu 5.4.0

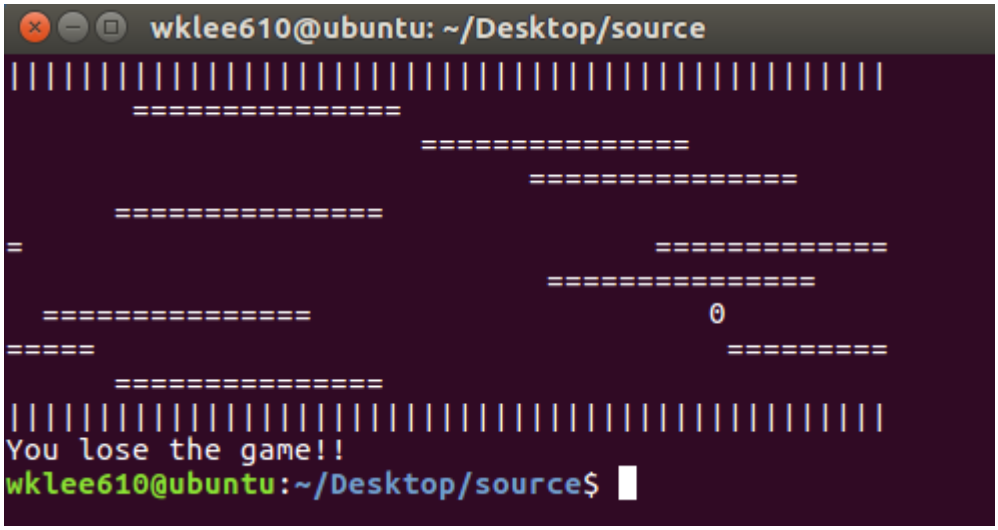
• The steps to execute your program.

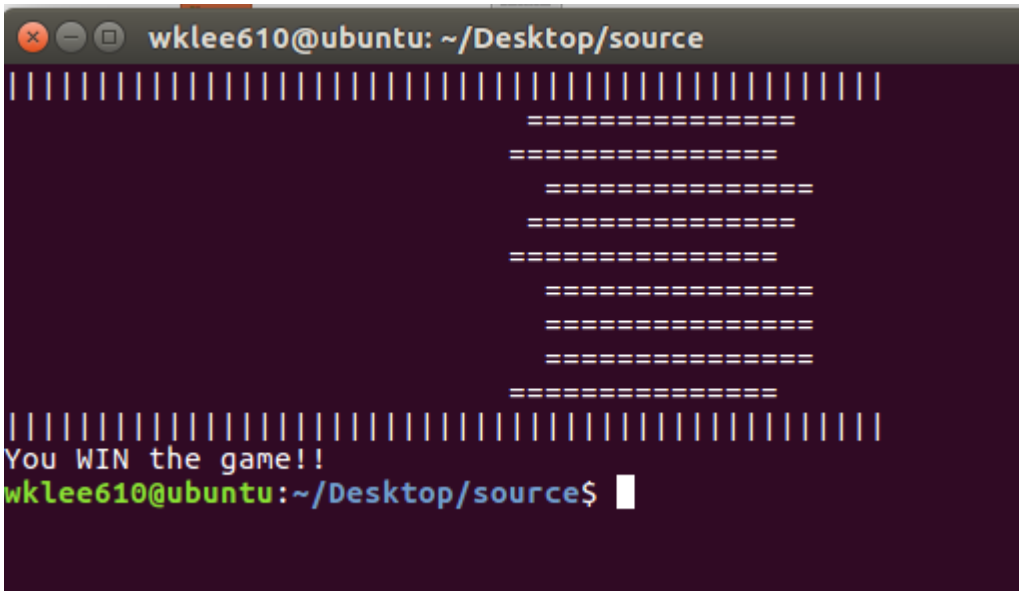
```
wklee610@ubuntu:~/Desktop$ cd source
wklee610@ubuntu:~/Desktop/source$ g++ hw2.cpp -lpthread
wklee610@ubuntu:~/Desktop/source$ ./a.out
```

To execute my program, first go to source directory and type ‘g++ hw2.cpp -lpthread’

And type ‘./a.out’ to execute.

• Screenshot of your program output.





•What did you learn from the tasks?

After this assignment, I got a knowledge about thread. I understood about how to create thread, how to execute, manage child thread and mutex-lock/unlock as well. Moreover, even though it's a little game, but still I can make a simple game at least. It was fun to think about how to design for this game.