Hanchong Jin, Aredo Plloci

Roscoe Giles

EK128 Engineering Computation++

03 May 2017

Final Project Report: GraphIt!

Report Narrative:

The goal of our project is to build a GUI application that is also a game. The game displays a number of dots/buttons, asks the user to draw a specific shape by connecting the dots, and then gives the user a result of whether he/she draws the correct shape. The graphic user interface consists of eight different parts, including a shape name display, a result display for each round, a display of the rules, a score board, a drawing board, a refresh button, a submit button and a reset button. The functions of each part is stated below:

- The shape name display will display the shape needed to be drawn at the start of every round.
- The result display will display whether the user has drawn the correct shape at the end of each round.
- An area on the right side of the interface shows the rules the user need to know in order to play the game.
- The score board displays the total score the user has gained; the user starts with a score of zero and each correct answer will add one to the total score.
- The drawing board is where the user can connect the dots shown on the screen to draw the specific shape.
- The refresh button at the bottom right corner of the interface will let the user to switch to another randomly generated shape.
- The submit button will check if the user has entered the correct shape.

- The reset button at the bottom left of the interface will erase whatever that has drawn by the user and let the user start the same round again with the same shape.

In this project, we used PyQt5, Qt Designer, python 3 as well as Photoshop. First, we used Qt Designer and Photoshop to design the front end. We used Photoshop to create icons we wanted for the buttons and used Qt Designer to put the entire graphic user interface together. Next, we connected Qt Designer to python 3 by using the built-in "uic.loadUiType" function from PyQt5. Finally, we used python 3 skills we learned from previous classes to write the back-end algorithms (a lot of functions) for the game.

This Project has been extremely challenging for both of us, since none of us had experience of working with GUI before. However, it has also been extremely rewarding since we eventually completed our application and learned a lot through the process of writing it. We first had to look for online tutorials to get a general understanding of how PyQt and Qt Designer work. Then we had to look for more tutorials on how to connect PyQt5 and Qt Designer to our python 3 file (We figured out a way to directly load the ".ui" file without converting it to a ".py" file, which saved us a lot of time during the process of debugging). We also figured out a way to represent different shapes in python code: we originally decided to use matrices filled with 0s and 1s, with 0 representing no connection between two dots and 1 representing there is a connection; however, since python 3 does not have a matrix variable type, we decided to use dictionaries in which there are numbers and lists. When it comes to actually drawing things using PyQt5, we faced a great difficulty. The "drawLine" function must be called inside a "paintEvent" function which can only be executed once when we run the program. This prevented us from continuously drawing as we kept clicking the dots. Nevertheless, we found a function called "update()" which would call the paintEvent again when needed, and we wrote a loop to prevent the previous drawn lines to be erased. Through

this project, we have learned both the front end and the back end of building an application. In addition, we learned to meet and spend time outside of class with each other to finish our project.

Besides constantly communicating with each other, we have divided the task between the UI/pattern design and the shape recognition algorithm. Again, both of us understand holistically how our application works. We just divided the tasks to save us time.

Our final application does exactly what we originally planned with 10 different shapes we designed. Yet we know that this project can be further improved in many ways. For example, we can add sub windows/popups after the user clicks the submission button; we can add hints for the user if he/she keeps inputting wrong answers multiple times; we can display the correct answer in a separate window after the user has given up; we can even make a mobile version of the game, or build a standalone ".exe" application and put it online. Overall, we believe our project is a big success.

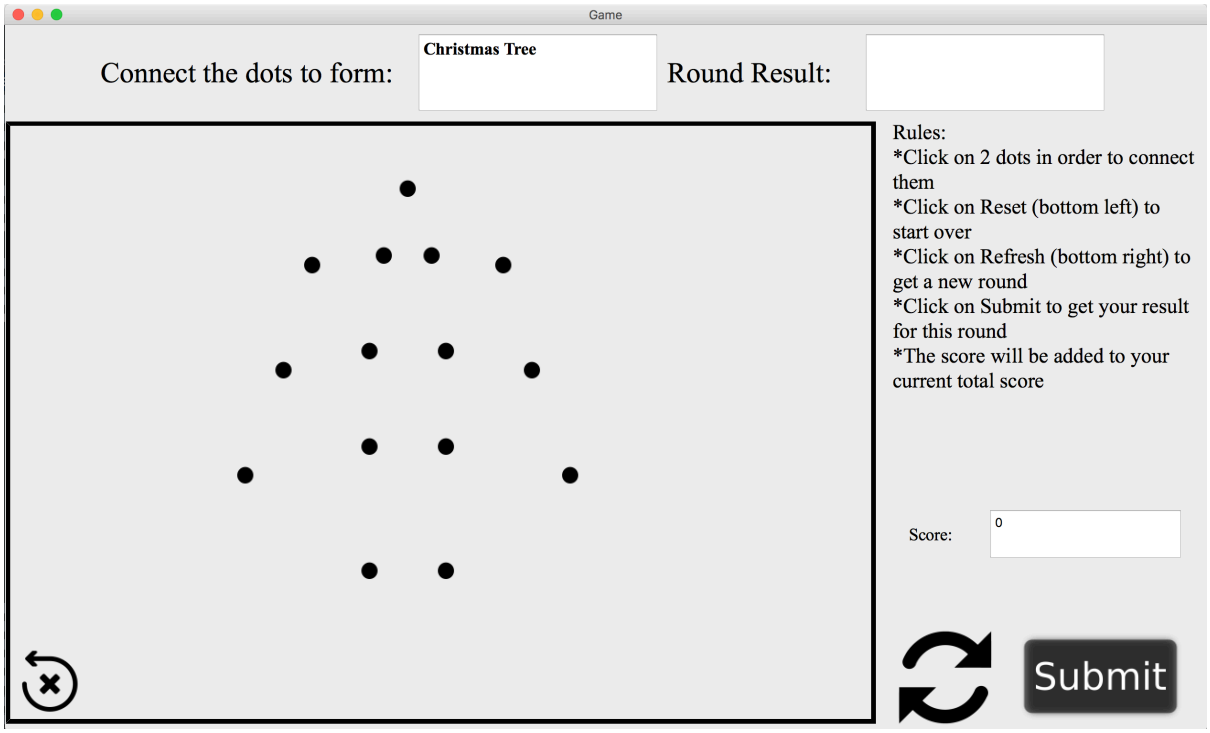
Code Appendix:

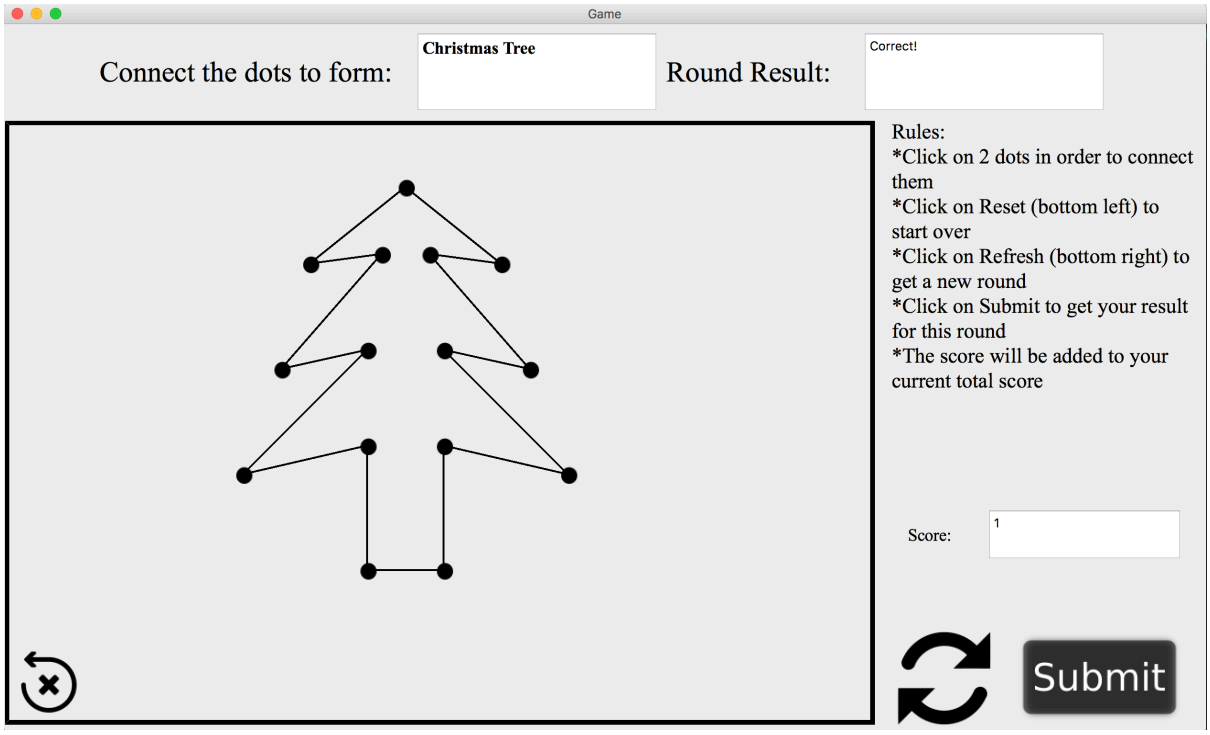
https://github.com/JackJin96/GraphIt-

(The .py file needs to be run with everything else in the same file, including the images and the .ui file)

Media (Screenshots):

(See next page of this document)


Game
Connect the dots to form:
Christmas Tree
Round Result:
Rules:
*Click on 2 dots in order to connect them
*Click on Reset (bottom left) to start over
*Click on Refresh (bottom right) to get a new round
*Click on Submit to get your result for this round
*The score will be added to your current total score
Score:
0
Submit


Game
Connect the dots to form:
Christmas Tree
Round Result:
Correct!
Rules:
*Click on 2 dots in order to connect them
*Click on Reset (bottom left) to start over
*Click on Refresh (bottom right) to get a new round
*Click on Submit to get your result for this round
*The score will be added to your current total score
Score:
1
Submit