Software Design & Development Logbook

12/05/2020

As testing the server and clients will prove to be difficult in the school because of me not having access, I'm going to start developing the client interface. Using tkinter, creating a parent tkinter object and laying out all the buttons and options will be the first step. A child object will be created to pop up as a dialogue box when the program is first run, to specify the connection and user.

After considering creating a new pop up dialogue box, I've opted to the client just opening in the connection option initially and then after connecting, resize and place the buttons on the window into the normal client interface.

19/05/2020

The testing of the server and clients works fine in the school, apparently. Thought that the school didn't give me access to make socket objects or whatever, but after talking to my dad about it yesterday, he said it shouldn't be like that, so I tried today using a simple server and client, and it worked. That being said I should work on the server then, since that's the actual program.

21/05/2020

Referring back to the messaging and echo servers I made in the holidays, I made a server that currently can:

- Accept connections from multiple different devices on the same network
- Send them a confirmation of connection message in the form of a welcome message
- Send data/command based on the user's input from the server
- Receive data from the client and store it

Currently trying to work on efficiently storing answers. Eventually, it will need to spit it out into a CSV file so it can be put on a excel file, but right now I need a way of storing everyone's answers accurately. If I was using small basic, I would've used a 2D array that looks like this, with the x being question number and y being client number:

Answers	1	2	3	4	5
1	A	C	D	A	B
2	D	C	A	A	B
3	B	B	B	B	B

But since python cannot do 2D arrays with lists, I was thinking of doing a list with lists in each cell, like this:

Answer_lists	1	2	3	4	5
	Client1_list	Client2_list	Client3_llist	Client4_list	Client5_list

And each client list would be:

Client1_list	1	2	3	4	5
	A	B	D	A	C

So that if I wanted to search for client3's answer for question….

I JUST REALIZED THIS DOESN'T WORK

Instead I'm going to just export the answers after the all the answers come into a CSV and use 1 list to temporarily store the answers each time.

25/05/2020

Had some problems trying to install matplotlib, which is for graphing stuff, and pip install was trying to use the python 2 that was installed on my computer, so I uninstalled all of python 2 and 3, pycharm 1 and 2 and reinstalled them all. After some trouble with it, I successfully installed python 3, updated pip and installed matplotlib.

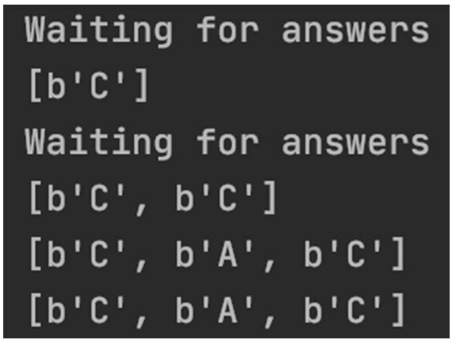
26/05/2020

Had some trouble actually receiving the answers, the first problem was that the sockets actually disconnected before receiving the question, but that was an easy fix, because I was just missing the loop. The next problem was the server wasn't actually receiving the answers. I was originally going to have separate functions to record the answers on the clientside and then send it to the server, but the function had some problem or something because when I removed it and just did it and it worked fine. The final problem was receiving the answers, and then figuring out who sent it, then ordering them so that they can be put into a list so that the client[counter]'s answer goes in answer[counter]. I did this by just going through the client list and receiving from them one by one, and if they don't send anything in that moment it just goes next by setting socket blocking to false. If an answer is received, it finds the index, then inserts it into the index of the answer list. This way, no matter the order they arrive in, the answers will be in order of the clients.

02/06/2020

Working on the answering part of the program, which is where the server receives the answers and has to pick which answer is correct. What I've done is collected the answers and put them into a dictionary, with the key being the answer in a bytes string, such as b"a", and made the value equal to the number of clients who answered that answer. I've made it so that the values are easy to punch into a graphing function, but other than that I think it works fine.

Ran into some trouble where the answer from one client counted as two, but it's an easy fix, where I just see if that client has already answered or not, and so if I receive another answer from that client, it just overwrites the one before it.



```
Waiting for answers
[b'C']
Waiting for answers
[b'C', b'C']
[b'C', b'A', b'C']
[b'C', b'A', b'C']
```

🡐 This happens when I answer C on client 1, then A on client 2, then C on client 1 again.

Fixed it by just optimizing the receiving of answers. I had it going through two while loops when I could just use 1 so I did that.

03/06/2020

Instead of inserting an answer in the correct position when an answer is received, I just gave everyone an initial answer of nothing, so that I can just replace it with the actual answer when it comes. This way, I can actually let the client change the answer.

04/06/2020

Had some problems installing matplotlib again, well I had it installed but it wasn't in the right folder so python wasn't able to see it. I just moved it in the right folder and it sees it now, so it works. Going to work on the graphing of the answers after the answers are collected.

There was a problem saying that one of the functions needed to make it working isn't in the __init__.py folder and I was having trouble, but after I went home and showed my dad who was interested, told me that I have to import matplotlib as matplotlib.pyplot for it to work properly, and it is working properly.

09/06/2020

Got the graphing working, ran into a problem where if there was an answer that no one chose, it wouldn't show up on the graph, meaning if everyone chose True in a true/false question, you would just see one really fat bar on a graph. Just fixed it by giving all answers a value of 0 so that it at least shows up on the graph.

Figuring out how to export as a CSV now. I know I have to use commas to move to the next cell, I just have to figure out how I want to do it.

, , client1, client2, client3, clientn
Question, 1, client1ans, client2ans, client3ans, clientnans

		Client1	Client2	Client3	ClientN
Q. Type	1	Client1ans	Client2ans	Client3ans	ClientNans
Q. Type	2	Client1ans	Client2ans	Client3ans	ClientNans
Q. Type	3	Client1ans	Client2ans	Client3ans	ClientNans
Q. Type	N	Client1ans	Client2ans	Client3ans	ClientNans

09/06/2020

As I've already done some File I/O before, making the .csv wasn't that hard. So, most of the functionality is done, I just need to code the buzzer, then make it graphical, I think. I had an idea where I changed the client's interface, so the small main interface just shows the Score, address and maybe the name, and whenever a question is asked, the appropriate answering prompt opens up in a new window.

For the buzzer, I'll make it so that the first person who sends the server an answer gets written down, then display their name on the bar graph. I made it so that when the client answers, it sends the name of the client, so that when the answer is shown, it just says the client's name on the graph.

17/06/2020

It seems that I have all the functionality finished, so I'll start working on the graphics using Tkinter. For the clients, I'll have a main window that only shows the client's name, server's ip address, maybe score, maybe ranking. When a question is received, an answer window will open up, depending on the question. After answering, the answer window will stay open until everyone has answered or the teacher stops the question or the timer runs out.

Just going to make another class object called Interface() for the client's interface. I was gonna mess around with class heritage and whatever but I don't know how to do that so I'm not going to do that. Encountered a problem where the window will not close after it's done its job. Other than it not closing, the rest works properly.

18/06/2020

Found a problem. I don't know if it's a problem within the code or the computer, but basically, if I run the program on my laptop, the windows that open up freeze instead of closing. I ran the program on the school desktops and they work fine.

22/06/2020

There's a massive problem now. I'm having trouble connecting clients to a host server. If I host the server using my laptop, the two desktops can connect, but if I host using one of the desktop computers, the clients don't seem to connect. The laptop still has the problem of the main window not closing but instead freezing, but another weird thing is the client opened on the other desktop will not connect. The window comes up, takes the info and then closes, but it just does nothing for like 10 seconds and then closes. According to the error, the server is apparently not responding.

It can connect on if the host and client are on the same computer, but currently the connection between other computers is not working for some reason. I didn't change any of the connection things from before, so it probably has something to do with the school network. Just in case, I'll go and test it and home.

Deciding how to write the graphical for the answering. Once I receive a question, a window according to that will open, so what needs to happen is I need to not use mainloop(), because that means it will get stuck in that loop without doing anything else. It should escape that loop once the server sends it a message for it to stop, but the mainloop() isn't something I can edit, so instead I'll use update().

22/06/2020

I think I figured out why the window doesn't want to close. The OS can't see if the application is open if it doesn't have a window, so even when the window has been destroyed, it tries to show something to show it is still running. I was changing my coding a bit so instead of creating a new class object called Interface that will handle the interface, I'll just do everything in the client class.

To fix the previous problem, instead of destroying the whole window, I'll re-define the window to show the name and connected address.

Still doesn't work for some reason. Considering threading at this moment. I kind of need two codes running at the same time. One to manage the interface, and one to actually run the program. I need the window to stay open while the program runs in the background, but every time I try to exit the loop of the window to continue the program, the window freezes and the code gets stuck on trying to connect to the server.

Apparently the blocking nature of the sockets causes the entire interpreter to freeze or something so I tried to connect using setblocking(False), but it raised an error about something like the operation is currently in progress, which is a really weird bug. I'll try to fix that bug first, and if it's impossible, I'll go rush a lesson on threading.

23/06/2020

Yeah I don't think it's possible to do this without threading. Whether the setblocking() is set to false or true, the connection to the server doesn't happen and just times out if a tkinter window is open. I'll have to learn threading or at least a basic level of threading in a week. I guess the command line interface version works, but all the problems with it will be fixed if it becomes a graphical interface.

Dad pointed out the problem that I was having, and that there was an easier fix, which is not basing the interface off of the socket object but rather basing the socket off the interface object. Basically, the main program will be the interface, and all the programming I wrote before will just be part of the button's commands. I'll have to majorly rewrite both programs, but it should fix the problem.

24/06/2020

I completed the server program up to binding a socket, accepting clients and sending the question. All works really well, and I also allowed for keyboard inputs. For the client program, I wrote all the way up to receiving the question, but it does not yet receive the question yet.
I also found a way to find the computer's IPV4 address without making the user go find it themselves, which is very nice.

26/06/2020

Completed converting the clients answering side. With that, the client side of the program is finished, except for the single answer part. Shouldn't be too hard actually, all I need to do is put an entry widget and a button, and the button will get the value from the entry widget. Now I need to write the server side to receive the answers.

29/06/2020

Receiving the answers don't really need graphics, other than showing who has answered or not, just to keep the host updated. So I pretty much just copy pasted my old code.
Used the video below to integrate the matplotlib graph into a tkinter window, and added the buttons to allow the host to pick the correct answer. Just need to add the functions that will actually do that though. For the graphing, I just used the code I wrote before, with some changes, like using figure and FigureCanvasTkAgg.
https://www.youtube.com/watch?list=PLQVvvaa0QuDclKx-QpC9wntnURXVJqLyk&time_continue=11&v=Zw6M-BnAPP0&feature=emb_title
Having trouble with allocating the correct points to the right people.
Whenever I press a button to select the right answer, all the functions say that the correct answer if False if you choose true or false. I don't know why this is happening at all.

01/07/2020

I'm going to try replacing the buttons with radio buttons and then add a confirm and close button. This way, the confirm and close button just gets which radio button was toggled, get either some variable it has, or just get its text, then uses that to continue giving the right people the scores, then closes.

https://stackoverflow.com/questions/22686587/tkinter-radiobutton-not-working

By using tk.Toplevel() instead of tk.Tk(), it opens a new window without creating a new tkinter object, which fixed the radiobuttons problem. I don't know why this fixes it but it does so I'm not going to mess with it further.

Changed the answer file's name to whatever the date is at the time, so that it can have multiple answer files without replacing or having the user rename it.

Program is fully functional now, even though it looks very bad. I'll have to add one last thing, which is the client list window that shows client's names and their scores.

02/07/2020

Created a simple setup python file that import matplotlib using pip. When first using A3, the user will need to first install python3, run the get-pip.py file that I will include, then run the A3Setup.py file to be able to use the sever.

If I compile the get-pip file and the A3Setup file into an exe, python doesn't need to be installed to run the program. There's a little problem where the A3Setup file doesn't seem to be doing anything.

Tried to run the server on the school desktop computers, but I wasn't able to connect to it with any other device, including other school desktop computers. This isn't meant to happen, so I kept testing it, and it kept failing. In the end, I tried testing on the minicomputer in the classroom and Mr Huang's laptop, but nothing was able to connect unless it was the server was on my laptop. Mr Huang asked the technician, and said it could be because of the ports. I just used port 1234 because I didn't think it mattered, but looks like a port can be open or closed, and port 1234 may be open for my laptop, but on the others it might not be open. I might have to allow the user to enter their own port, which won't be hard to make.

Tried testing port 3389 on the school desktops, but it came out with an OSError that said I have no permission to that port.

The technician came and opened port 1234 on one of the school desktop computers, and after he did that other computers could connect to that computer, so that was the problem.