Building a GUI in Python for MaPSA testing

Mentee: Aliyah Montgomery (Howard University)

Mentor: Doug Berry (Fermilab)

Introduction to the project and our goals

A GUI, or a graphical user interface is a form of user interface that allows users to interact with other electronic devices through graphical ions and audio indicators. Building this GUI for MaPSA testing in python is critical for the USCMS HL-LHC Outer Tracker Upgrade and allows the performance goal of this research to be attained. Once completed, novice student researchers should be able to conduct their own studies using the code shortcuts created in python.

I first created a “Hello World” python gui with a button and a check box in Tkinter to familiarize myself with the software before moving onto adding a test box. From there I was given the code:

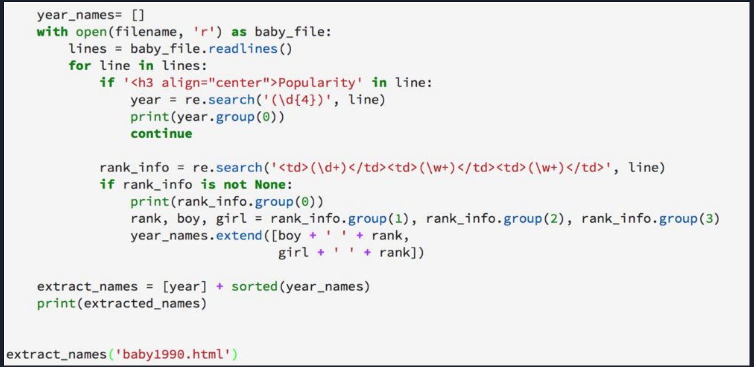
```
def pa ():
    pixel_alive=
    cal_pixelalive(ref_valCAL=250, ref_valTHR=250)
```

which allowed us to observe threshold within our own test area and eventually implement the codes created in there into Hannsjörg’s GUI code.

Tool used for Research

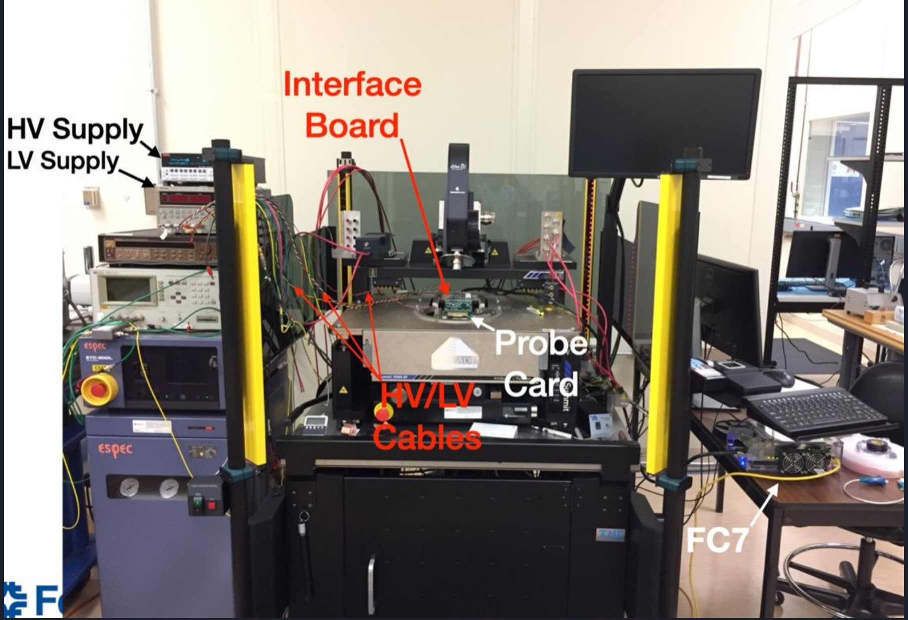
01

- Tkinter
  - De facto GUI Python interface
  - Combined with Python to create a faster and easier way to actually create GUI applications
  - Created a rather simple space for me to understand the codes that I was running so I could ultimately complete the buttons within my code.



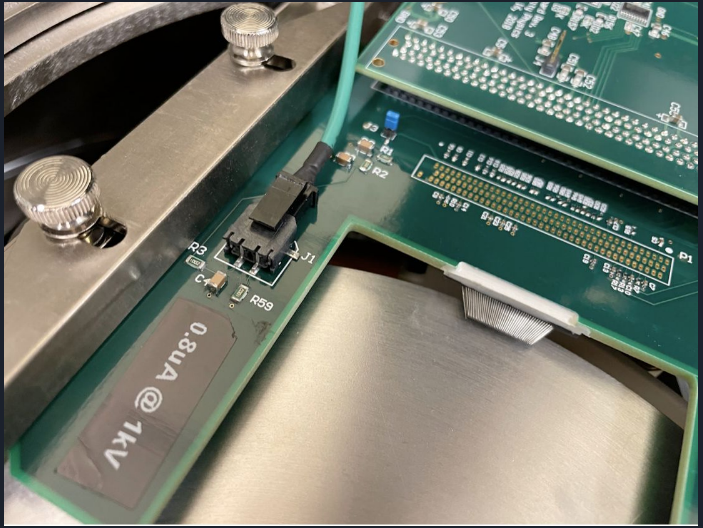
```
    year_names= []
    with open(filename, 'r') as baby_file:
        lines = baby_file.readlines()
        for line in lines:
            if '<h3 align="center">Popularity' in line:
                year = re.search('(\d{4})', line)
                print(year.group(0))
                continue

            rank_info = re.search('<td>(\d+)</td><td>(\w+)</td><td>(\w+)</td>', line)
            if rank_info is not None:
                print(rank_info.group(0))
                rank, boy, girl = rank_info.group(1), rank_info.group(2), rank_info.group(3)
                year_names.extend([boy + ' ' + rank,
                                   girl + ' ' + rank])

    extract_names = [year] + sorted(year_names)
    print(extracted_names)


extract_names('baby1990.html')
```

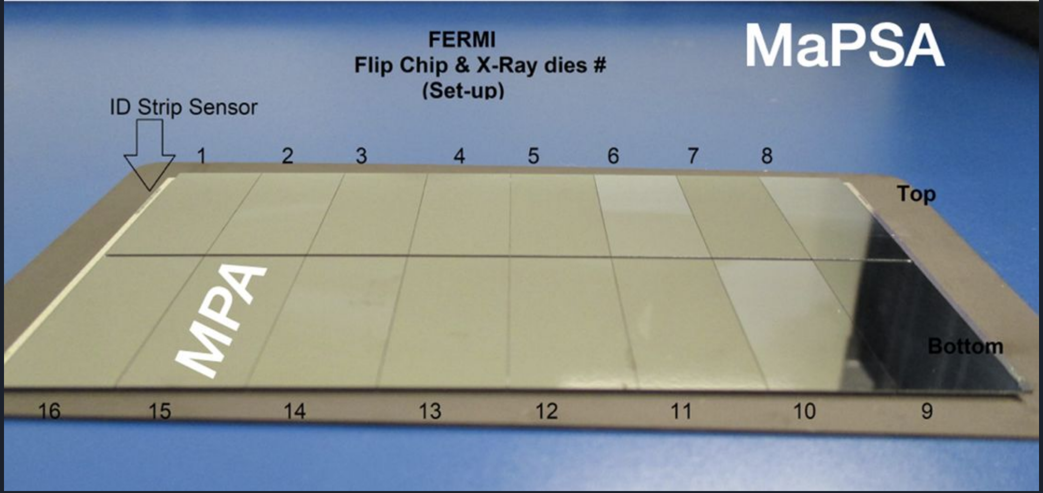
Probe Test Station

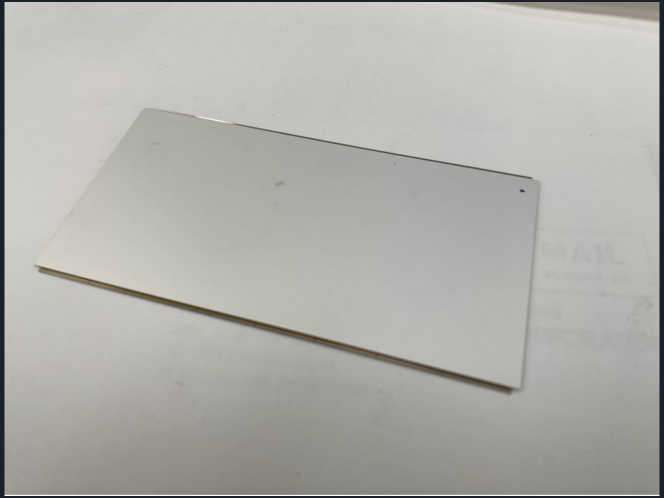


This is a physical station where the codes I have programmed are ran using the MaPSA's. Probe stations allow users to position electrical, optical or RF probes onto a device that will test that device's response to external stimuli

MaPSA probe card

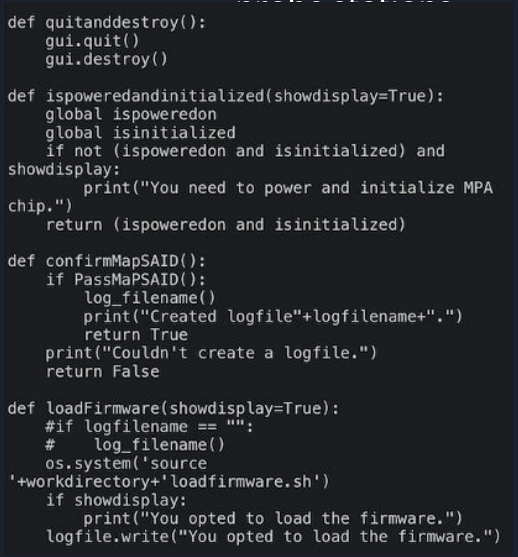




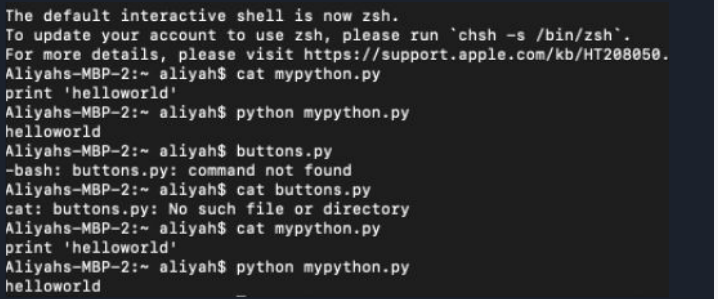


Major steps completed on my personal GUIs

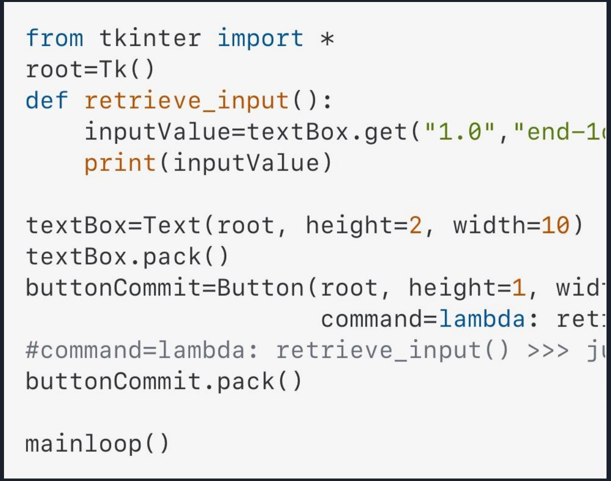
The goal of my personal GUIs was to create shortcut buttons using Tkinter that would make calling functions to run tests easier for movie coders. I ran local tests through trial and error with the help of my mentor to create these buttons and eventually enter them within Hannsjörg's code. I was given my own space to work an run the codes without affecting the actual probe stations, then once I completed and understood the buttons I could create, I was able to actually work towards running the codes through physical MaPSA's to be read using the probe stations.



```
def quitanddestroy():
    gui.quit()
    gui.destroy()

def ispoweredandinitialized(showdisplay=True):
    global ispoweredon
    global isinitialized
    if not (ispoweredon and isinitialized) and
showdisplay:
        print("You need to power and initialize MPA
chip.")
    return (ispoweredon and isinitialized)

def confirmMapSAID():
    if PassMaPSAID():
        log_filename()
        print("Created logfile"+logfilename+".")
        return True
    print("Couldn't create a logfile.")
    return False

def loadFirmware(showdisplay=True):
    #if logfilename == "":
    #    log_filename()
    os.system('source
'+workdirectory+'loadfirmware.sh')
    if showdisplay:
        print("You opted to load the firmware.")
    logfile.write("You opted to load the firmware.")
```



```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Aliyahs-MBP-2:~ aliyah$ cat mypython.py
print 'helloworld'
Aliyahs-MBP-2:~ aliyah$ python mypython.py
helloworld
Aliyahs-MBP-2:~ aliyah$ buttons.py
-bash: buttons.py: command not found
Aliyahs-MBP-2:~ aliyah$ cat buttons.py
cat: buttons.py: No such file or directory
Aliyahs-MBP-2:~ aliyah$ cat mypython.py
print 'helloworld'
Aliyahs-MBP-2:~ aliyah$ python mypython.py
helloworld
```



```
from tkinter import *
root=Tk()
def retrieve_input():
    inputValue=textBox.get("1.0","end-1
    print(inputValue)

textBox=Text(root, height=2, width=10)
textBox.pack()
buttonCommit=Button(root, height=1, wid
                    command=lambda: ret
#command=lambda: retrieve_input() >>> ju
buttonCommit.pack()

mainloop()
```

My experience/ What I learned

- Create a code shortcut in python for novice coders to utilize
- Add functions to call methods
  - MaPSA Testing
- How to utilize Tkinter and Python to create buttons, shortcuts and codes
- How to call/ recall a function to aid in the creation or addition of another code
- I thoroughly enjoyed working with my mentor, his team, my fellow interns and Fermilab holistically