

Project Resources/Research

GUI:

1. Tkinter: A module that comes with python that allows people to create GUI's.
 - a. Documentation - <https://docs.python.org/3/library/tk.html>
 - b. Here are some videos on Using Tkinter with python
 - I. <https://www.youtube.com/watch?v=jE-SpRI3K5g>
 - II. <https://youtu.be/YXPyB4XeYLA>
 - III. <https://www.youtube.com/watch?v=ELkaEpN29PU>
 - c. A simple Hello World Program using Tkinter
<https://docs.python.org/3/library/tkinter.html#tkinter-modules>
 - d. Tkinter Dialog Boxes where it asks the user for input like what file to open
<https://docs.python.org/3/library/dialog.html>
 - e. Here is Tkinter message prompts:
<https://docs.python.org/3/library/tkinter.messagebox.html#module-tkinter.messagebox>
2. Another way to do it is using PyQt5, PyQt5 is implemented as a set of python modules
 - a. Documentation - <https://doc.qt.io/qtforpython/> ,
<https://www.riverbankcomputing.com/static/Docs/PyQt5/>
 - b. Some videos
 - I. <https://www.youtube.com/watch?v=Vde5SH8e1OQ>
 - II. <https://www.youtube.com/watch?v=ksW59gYEI6Q&t=12s>

Simulating Terminal:

1. This is a way I found to potentially do this:
<https://stackoverflow.com/questions/11634626/basic-terminal-emulation-in-python>
 - a. Shared by Nathan <https://vincent.bernat.ch/en/blog/2017-write-own-terminal>
2. Here is how to create your own interactive shell
<https://code-maven.com/interactive-shell-with-cmd-in-python>

Recording Keyboard Input: Here are some ways to record the keyboard input

1. <https://code.activestate.com/recipes/134892-getch-like-unbuffered-character-reading-from-stdin/>
 - a. <https://stackoverflow.com/questions/13207678/whats-the-simplest-way-of-detecting-keyboard-input-in-a-script-from-the-terminal>
2. There is also using pynput which is a library that can be used to control the input of devices such as the keyboard. On windows to install pynput go on cmd (assuming you have python installed to the path on your computer) and type pip install pynput
 - a. Here are some example on how to use pynput: <https://pypi.org/project/pynput/>

Here is a keylogger using pynput that reads the keyboard and writes it into a textfile:

```
1  import pynput
2
3  from pynput.keyboard import Key, Listener
4
5  count = 0
6  keys = []
7
8  def on_press(key):
9      global keys, count
10     keys.append(key)
11     count += 1
12     print("{0} pressed".format(key))
13
14     if count >= 1:
15         count = 0
16         write_file(keys)
17         keys = []
18
19  def write_file(keys):
20     with open("log.txt", 'a') as f:
21         for key in keys:
22             k = str(key).replace("'", "")
23             if k.find("enter") > 0:
24                 f.write('\n')
25             elif k.find("space") > 0:
26                 f.write(" ")
27             elif k.find("Key") == -1:
28                 f.write(k)
29
30
31  def on_release(key):
32     if key == Key.esc:
33         return False
34
35  with Listener(on_press =on_press, on_release=on_release) as listener:
36     listener.join()
37
```

It writes the output to log.txt and stops recording when the ESC key is pressed. Its a sample program should be able to modify it so that it works with a gui

General:

1. Just some general things I found while researching some stuff.
 - a. Here is an application that records and plays back keyboard input on github using a GUI application : <https://github.com/Zuzu-Typ/Input-Recorder>
 - b. here is an article on how a user can record their terminal activity using script which is a UNIX command-line application
<https://www.2daygeek.com/linux-script-command-record-replay-terminal-session-activity/>
 - c. Here is another example but this one doesn't use Tkinter:
<https://github.com/RMPR/atbswp/tree/master/atbswp>