Task 5.3D RPi - Blink Morse code using GUI - Don Athalage

Q1: Take a video of your system demonstrating the outcome, and upload it to youtube. Include the link here.
https://youtu.be/1v2tax9RAzM

Q2: Create a repository on Github, name it Morse_GUI. Include the link to your repository here.
https://github.com/Kushan-Nilanga/Morse_GUI

Q3: How did you test your system?

User Input

Morse code is defined only for alphanumeric characters [A-Z, 0-9], initially I didn't have a filter out the special characters. This resulted in several crashes while testing different character sequences. Special characters such as (_ * %) throws a key error in **morse_lookup**. This was identified by using different characters to break the behaviour of the code. Since, a new **fliter** function has used to strip out any special characters

Helllo_World! -> HelloWorld
H3L$0 -> H3L0

System output

The system output had to be tested extensively and changes to time delays had to be applied. This was done using trying out different characters and write down the LED output on a paper. This helped diagnose different issues in the time delays

```
17  def send_morse(code):
18      for signal in code:
19          if signal == '.':
20              GPIO.output(40, GPIO.HIGH)
21              time.sleep(0.2)
22              GPIO.output(40, GPIO.LOW)
23
24          if signal == '-':
25              GPIO.output(40, GPIO.HIGH)
26              time.sleep(1)
27              GPIO.output(40, GPIO.LOW)
28          time.sleep(0.5)
```

In the earlier versions of code the delay at the line 28 did not exist and this cause the last dot or dash of current binded with the first dot or dash of the next character corrupting the signal. Writing down the output of the signal I sent can comparing them helped me identify this issue.