Lab Work - Help Sheet 3

Recording sequences of movements.

Assuming you have managed to record and identify individual hand movements made in the range finder's beam, then you need to add their names to a list in order to recognise sequences of movements as defined in the task.

In order to know when a sequence of movements has been made by the user, the task tells you to wait for a period of inactivity of about 2 seconds and at that point to analyse the recorded sequence of movements, print out the mapped command name, and discard the recording in order to start another one.

Unless you have changed it, your sampling loop is using a delay value of 0.1 seconds in the sleep command. In that case there are 10 loops in each second. Thus the easiest way to identifiy a period of inactivity of 2 seconds is to count 20 loop iterations in which the range readings are all above the threshold you have chosen for hand movements - that is 20 readings that are all the distance to the ceiling. Whenever a reading below that threshold is received you would reset the counter to zero.

If you are recording the names of the hand movements you have identified in a list variable called *movementList* then the following example code illustrates how to decide when to analyse the *movementList* sequence. (This assumes a sleep value of 0.1 is being used, as described above).

```
nohandcounter = 0
while(True):
      … other code in this loop already …
      if (distance > threshold):
            nohandcounter = nohandcounter + 1
            if (nohandcounter >= 20):
                  identifySequence(movementList)
                  nohandcounter = 0
                  movementList = []
      else:
            nohandcounter = 0

def identifySequence(movementList):
      … look up the movement list in the command name dictionary …
```

This code calls a function called *identifySequence*, giving it the list of recorded movements. This function would need to be defined, and it would need to do the following things:

1. Check whether the given list has nothing in it and do nothing if so (this could be avoided in other ways too, but the example code above needs this check to be done).
2. Join the list into a single key string using the ***join*** command described in the previous help sheet.
3. Look up the name of the command mapped to that key string in a dictionary you have defined based on the table of commands given in the task.
4. If there is a command name, print it out, if there is not a command name matching that sequence, print out "sequence not recognised".

IMPORTANT!!

To look up the command name in the dictionary you might use something like this:

```
command = mydictionary[keystring]
print(command)
```

Where *mydictionary* is the name you gave to the dictionary based on the command name table you declared, and *keystring* is what you got by joining together the individual movement names in *movementList*.

BUT if there is no mapping for *keystring* in the dictionary, a **KeyError** exception will be raised. There are two solutions to this:

The better and easier way is to write

```
command = mydictionary.get(keystring)
```

instead of using the square-brackets syntax to read from the dictionary. In this case if there is no mapping to *keystring* the special value ***None*** is returned. You can then just print ***None*** or you can use an if statement to print whatever you like if the value was ***None***.

The other way is to enclose the dictionary access in a **try** block to catch the exception. This would look like this:

```
try:
      command = mydictionary[keystring]
      print(command)
except KeyError:
      print(keystring, "not recognised")
```

You can read lots of information about all this (probably too much!) in this article:

https://realpython.com/python-keyerror