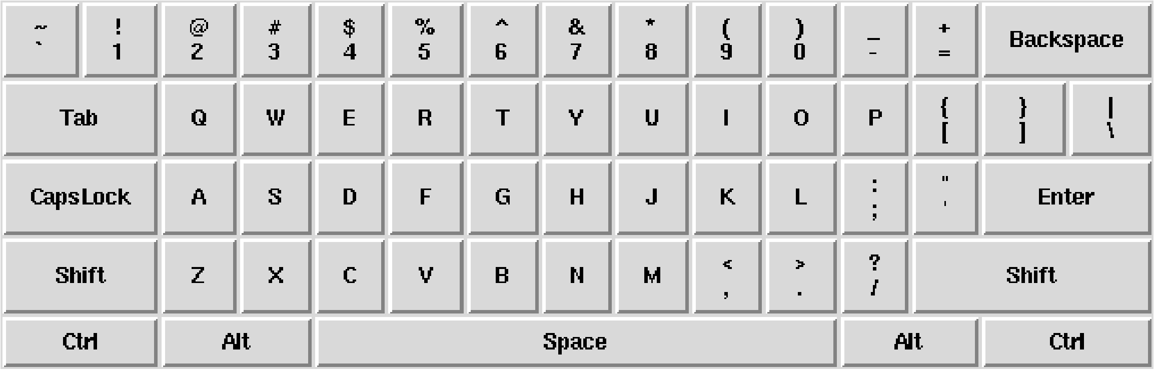
Homework #5 Due: 11/13

1. [100 points] Write a Python module `keyboard.py` that defines a class `Keyboard` (and perhaps another class or classes as needed) to implement a keyboard-like Tkinter object that looks like this:



(This might be used to enter text on a touchscreen-based tablet PC that did not have a keyboard attached to it.)

A `Keyboard` is a `Frame` whose constructor is `Keyboard(parent, callback)` where `parent` is the usual parenting `Frame` or `Toplevel` and `callback()` is a function invoked whenever the user “presses” (i.e. clicks on) a key on the keyboard. The argument passed to `callback()` is determined according to the key pressed and the state of the `Keyboard` instance itself (see below).

All keyboard layout information is contained in the data structure `layout` in the file `layout.py` accessible from the course web page via the link for this assignment. Note the comments in that file regarding its format. Your code must `import` this file and use it as-is.

At any time, a `Keyboard` is in one of three states: normal (the initial state), shifted, and cap. The value it passes to `callback()` depends on its state as well as the kind of key pressed. There are three kinds of keys:

- letters:
  In the normal state, these keys pass their lower case version. In the shifted and cap states, they pass their upper case versions.
- dual-symbol:
  In the normal and cap states, these pass the lower symbol in their labels. In the shifted state, they pass the upper symbol.
- special:
  These always pass their label (including the **Space** key, which returns ’Space’, except for the **Shift** and **CapsLock** key. These do not call `callback()`, but instead change the state of the keyboard as described below.

For the **Shift** key:

```
if the keyboard state is shifted or cap:
   set the state to normal
otherwise:
   set the state to shifted
```

For the **CapsLock** key,

```
if the keyboard state is cap:
   set the state to normal
otherwise:
   set the state to cap
```

In addition, the shifted state only applies to the next non-**Shift**, non-**CapsLock** key, after which the state returns to normal.

To demonstrate the usage of `Keyboard`, here is a test program you should include as your self-test. (You are expected to have a self-test in any case.)

```
from Tkinter import *
from keyboard import Keyboard

def press(key):
    print(key)

root = Tk()
root.title('Keyboard Module Self-Test')
kbd = Keyboard(root, press)
kbd.grid()
root.mainloop()
```

[10 pts. Extra Credit] Modify the appearance of the letter keys to reflect the keyboard state. That is, upper case for the cap and shifted states and lower case otherwise.