Pau Magro Alabarce
PBE-LAB

MEMÒRIA PUZZLE 2 (LCD)

INSTAL·LACIÓ DE BIBLIOTECA GTK

Primer de tot el primer pas que vaig fer va ser instal·lar-me la biblioteca GTK3 ja que com deia l'enunciat es imprescindible utilitzar-la.

Per tant, vaig obrir la terminal dins de la Raspberry utilitzant el ssh i vaig introduir aquest comando:

```
sudo apt-get install libgtk-3-dev
```

Una vegada instal·lada podem passar ja a la implementació del codi del puzle 2.

CODI

Com es una llarg el codi, he de comentar que he intentat que sigui lo més simplificat que he pogut, ho explicaré per blocs:

```
import gi
gi.require_version("Gtk", "3.0")
from gi.repository import Gtk
import lcddriver
```

Aquestes llibreries gi i Gtk son necessàries per crear interfícies gràfiques utilitzant com he dit abans el Gtk3, a més el gi.require_version("Gtk","3.0") això assegura que utilitzo la versió que necessito.

A més que utilitzo un altre vegada la llibreria lcddriver que vaig utilitzar per la implementació del puzle 1.

Bé, doncs ara podem crear la primera classe principal que la he denominat LCDApp.

```
class LCDApp(Gtk.Window):
```

Aquesta classe hereta de Gtk.Window, això significa que defineix una finestra de la aplicació gràfica.

Ara, dins de la classe LCDApp vaig fer el constructor de la finestra.

```
def __init__(self):
    super().__init__(title="Pantalla LCD")
    self.set_default_size(300, 200)
```

Com es pot veure truco al constructor de la classe per crear la finestra anomenada "Pantalla LCD" i seguidament amb self.set_default_size(300,200) això estableix la grandària de la finestra.

Després, vaig crear un objecte de la classe lcd() que representa el display i després neteja la pantalla perquè comenci buida.

Per la creació de la interfície amb Gtk.Box vaig crear un contenidor vertical VBox per organitzar els elements

```
# Creo un objeto lcd.
self.lcd = lcddriver.lcd()
self.lcd.lcd_clear()

# Creo un vbox para establecer widgets verticalmente.
vbox = Gtk.Box(orientation=Gtk.Orientation.VERTICAL, spacing=6)
self.add(vbox)
```

Ara passem per la creació de la interfície.

```
# Creo un textView para ingresar texto multilínea.
self.textview = Gtk.TextView()
self.textview.set_wrap_mode(Gtk.WrapMode.CHAR)
vbox.pack_start(self.textview, True, True, 0)
```

Vaig implementar el Gtk.TextView que serveix per introduir text en un àrea de text.

A més set_wrap_mode(Gtk.WrapMode.CHAR) fa que el text s'ajusti a l'amplada, tallant per caràcters i finalment s'afegeix el TextView al contenidor vertical VBox.

Per la implementació del botó que ens demana l'enunciat simplement vaig introduir aquestes expressions:

```
# Botón para mostrar en el display
btn_display = Gtk.Button(label="Display")
btn_display.connect("clicked", self.display_text)
vbox.pack_start(btn_display, False, False, 0)
```

Lo més importat d'aquí es que quan es fa clic, es truca la funció self.display_text i també el botó s'afegeix al VBox.

Ara, vaig crear un altre funció per mostrar el text en el display:

```
def display_text(self, _):
    buffer = self.textview.get_buffer()
    text = buffer.get_text(buffer.get_start_iter(), buffer.get_end_iter(), False).strip()
    lines = text.split("\n")[:4]  # Limita a 4 líneas
    print("Texto ingresado:", lines)

    self.lcd.lcd_clear()
    for i, line in enumerate(lines):
        self.lcd.lcd_display_string(line[:20], i + 1)  # Máximo 20 caracteres por línea
```

Com he dit abans aquesta funció s'executa quan el usuari fa clic al botó.

Llavors, vaig crear un get_buffer() en la qual obté el contingut escrit per l'usuari i el get_text() extreu tot el text com un string. Divideix el text en línies i neteja el display.

Finalment, l'execució del programa es fa d'aquesta manera :

```
if __name__ == "__main__":
    app = LCDApp()
    app.connect("destroy", Gtk.main_quit)
    app.show_all()
    Gtk.main()
```

Vaig crear una instància de la classe LCDApp, connecto l'esdeveniment de tancar la finestra (destroy) amb Gtk.main_quit, per finalitzar el programa correctament.

Llavors aram amb show_all() mostra tots els widgets de la finestra i el Gtk.main() inicia el bucle principal de GTK.

Per tant, aquí pots trobar el codi complet i ben format:

```
import gi

gi.require_version("Gtk", "3.0")

from gi.repository import Gtk

import lcddriver

class LCDApp(Gtk.Window):

  def __init__(self):

    super().__init__(title="Pantalla LCD")

    self.set_default_size(300, 200)
```

```
        # Creo un objeto lcd.
        self.lcd = lcddriver.lcd()
        self.lcd.lcd_clear()
        # Creo un vbox para establecer widgets verticalmente.
        vbox = Gtk.Box(orientation=Gtk.Orientation.VERTICAL, spacing=6)
        self.add(vbox)
        # Creo un textView para ingresar texto multilínea.
        self.textview = Gtk.TextView()
        self.textview.set_wrap_mode(Gtk.WrapMode.CHAR)
        vbox.pack_start(self.textview, True, True, 0)
        # Botón para mostrar en el display
        btn_display = Gtk.Button(label="Display")
        btn_display.connect("clicked", self.display_text)
        vbox.pack_start(btn_display, False, False, 0)
    def display_text(self, _):
        buffer = self.textview.get_buffer()
        text = buffer.get_text(buffer.get_start_iter(), buffer.get_end_iter(), False).strip()
        lines = text.split("\n")[:4]  # Limita a 4 líneas
        print("Texto ingresado:", lines)
        self.lcd.lcd_clear()
        for i, line in enumerate(lines):
            self.lcd.lcd_display_string(line[:20], i + 1)  # Máximo 20 caracteres por línea
if __name__ == "__main__":
    app = LCDApp()
    app.connect("destroy", Gtk.main_quit)
    app.show_all()
    Gtk.main()
```

Pau Magro Alabarce
PBE-LAB

Problemes trobats

-He tingut un problema amb la biblioteca que utilitzo: lcddriver. El que volia es utilitzar-la al meu programa però sense que el fitxer del lcddriver.py estigués en la mateixa carpeta que el puzzle2.py, però ho he intentat fer-lo creant un entorn virtual i després creant un directori on posar totes les llibreries juntes però a l'hora de posar les llibreries com externes amb python3, executo el programa però diu que no detecta les llibreries i per tant em surt error d'execució.

A part d'aquest inconvenient, no he trobat cap altre problema.