Segon Puzzle

Itead PN532 NFC amb Ruby i GTK3

1. Objectiu

La segona part del projecte consisteix en implementar una interfície gràfica del Puzzle 1, amb una finestra principal formada per un *Label* que demana login amb la targeta RF—carnet UPC o Mifare— i un *Button* de Clear. Quan es llegeix la targeta, apareix l'UID en format hexadecimal.

2. Instal·lació de GTK3

Per al desenvolupament del Puzzle 2, va ser necessari instal·lar la biblioteca GTK3, ja que aquesta proporciona les eines necessàries per construir interfícies gràfiques d'usuari (GUI). GTK3 (GIMP Toolkit version 3) és un conjunt d'eines multiplataforma de codi obert per a la creació d'interfícies gràfiques.

La biblioteca GTK per a Ruby es pot obtenir a través de la gemma gtk3, que forma part del conjunt de gemmes publicades a RubyGems, el repositori de paquets més utilitzat per al llenguatge Ruby.

Per instal·lar la biblioteca, vaig utilitzar la següent comanda a la terminal de la Raspberry Pi:

```
➔  gem install gtk3
```

3. Programació

3.1. Gràfics

Utilitzem `Gtk.init` per inicialitzar GTK. A continuació, podem afegir els elements gràfics:

- **Window**: Finestra amb el títol "PUZZLE 2". Es defineix una mida 500x150 i un marge de 10 píxels. Afegim la caixa *vbox*, mostrem la finestra i els seus elements amb *show_all* i finalment, permetem el tancament de l'aplicació quan es prem la "x" de la finestra.

```
window = Gtk::Window.new("PUZZLE 2")
window.set_border_width(10)
window.set_default_size(500, 150)
      …
window.add(vbox)
window.show_all
window.signal_connect("destroy") { Gtk.main_quit }
```

- **Label:** Inicialment, el text de l'etiqueta és "Please, login with your university card", amb mida 20. El fons de l'etiqueta és blau per indicar que s'està esperant la targeta, i el text blanc. Un cop escanejada la tarjeta, canviem a color vermell i actualitzem el text amb l'UID. També, és important destacar que per poder utilitzar la variable *label* a dins del mètode *scan*, és necessari afegir "@" davant.

```
@label = Gtk::Label.new

#Estat inicial:

@label.set_markup("<span font='20'>Please, login with your
university card</span>")
@label.override_background_color(:normal, Gdk::RGBA.new(0, 0, 1,
1))
@label.override_color(:normal, Gdk::RGBA.new(1, 1, 1, 1))
@label.set_size_request(500, 100)

#Un cop fet l'escaneig:

@label.set_markup("<span font='20'>UID: #{uid} </span>")
@label.override_background_color(:normal, Gdk::RGBA.new(1, 0, 0,
1))
```

- **Botó de *Clear*:** Quan l'usuari fa clic en aquest botó, es torna a l'estat inicial, demanant que es torni a passar la targeta per autenticar-se.

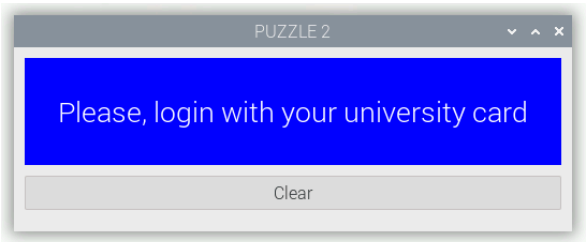
```
button = Gtk::Button.new(label: "Clear")

button.signal_connect("clicked") do
      … #codi a executar quan es prem el botó
end
```

- **Caixa vertical (vbox)**: Organitza els elements de la interfície en una disposició vertical. Els elements (*label* i *button*) es col·loquen dins d'aquesta caixa amb *pack_start*.

```
vbox = Gtk::Box.new(:vertical, 0)
vbox.pack_start(@label, expand: true, fill: true, padding: 0)
vbox.pack_start(button, expand: true, fill: true, padding: 10)
```

Finalment, `Gtk.main` inicia el bucle principal de l'aplicació. El resultat és el següent:



3.2. Importació del Puzzle 1

La classe Rfid, resultat del Puzzle 1, s'importa a través de `require_relative 'read_uid'`, que permet accedir a la funció `read_uid`, responsable de llegir el UID de la targeta. Aquesta funció es crida dins el mètode `scan`, que crea un nou objecte `Rfid`, crida `read_uid`, i obté el UID de la targeta escanejada `(uid = Rfid.new.read_uid)`.

3.3. Solució a l'ús d'un mètode bloquejant en aplicacions gràfiques (Thread auxiliar)

En aplicacions gràfiques, l'ús d'un *thread* auxiliar és necessari a l'hora d'executar un mètode bloquejant, com és el cas de `read_uid`. Això és degut a que l'aplicació gràfica funciona en un *thread* principal dedicat exclusivament a la gestió de la interfície d'usuari (detecció d'inputs de l'usuari, redimensionat de l'aplicació a la pantalla i actualització de canvis). Aquest *thread* principal té una execució seqüencial que provoca que tota la resta d'operacions es posin en espera fins que certa operació acaba. Per tant, si `read_uid` s'executés en el *thread* principal, l'aplicació gràfica es congelaria fins que es completés la lectura i l'usuari no podria interactuar amb l'aplicació.

En aquest cas, el mètode `scan` crea un *thread* auxiliar amb `Thread.new` dins del qual s'executa el mètode bloquejant `read_uid`. D'aquesta manera, l'aplicació pot continuar responent a altres accions mentre s'executa la lectura. Un cop llegida la targeta, per actualitzar l'etiqueta gràfica amb l'UID de manera segura donat que estem treballant amb *threads*, és recomanable utilitzar `GLib::Idle.add` per invocar la funció al *thread* principal de GTK.

3.4. Codi

```
require 'gtk3'
require_relative 'read_uid'

class GUI

    def scan

        Thread.new do #comença l'escanejat en thread
            puts "Waiting for a card scan..."
            uid = Rfid.new.read_uid
            puts "Card scanned!"
            GLib::Idle.add do #GLib::Idle.add permet actualitzar la GUI de manera
segura
                @label.set_markup("<span font='20'>UID: #{uid} </span>")
                @label.override_background_color(:normal, Gdk::RGBA.new(1, 0, 0,
1))
                @scanning = false
                false #atura GLib::Idle.add
```

```
            end
        end

    end

    def finestra

        Gtk.init

        window = Gtk::Window.new("PUZZLE 2")
        window.set_border_width(10)
        window.set_default_size(500, 150)

        vbox = Gtk::Box.new(:vertical, 0)

        @label = Gtk::Label.new
        @label.set_markup("<span font='20'>Please, login with your university
card</span>")
        @label.override_background_color(:normal, Gdk::RGBA.new(0, 0, 1, 1))
        @label.override_color(:normal, Gdk::RGBA.new(1, 1, 1, 1))
        @label.set_size_request(500, 100)

        button = Gtk::Button.new(label: "Clear")

        vbox.pack_start(@label, expand: true, fill: true, padding: 0)
        vbox.pack_start(button, expand: true, fill: true, padding: 10)

        scan #primera execució de scan
        @scanning = true

        button.signal_connect("clicked") do #torna a l'estat inicial: demanar login
i escanejar
            if !@scanning #evita que s'executi un nou thread quan l'usuari
pressiona el botó clear mentre s'escaneja
                @label.set_markup("<span font='20'>Please, login with your
university card</span>")
                @label.override_background_color(:normal, Gdk::RGBA.new(0, 0, 1,
1))
                @scanning = true
                scan
            end
        end

        window.add(vbox)
        window.show_all
        window.signal_connect("destroy") { Gtk.main_quit }

        Gtk.main

    end
```

```
end

if __FILE__ == $0

    GUI.new.finestra

end
```