### **Experiment 4: UIKit Experiment with Custom RGB Color Picker**

### **Objective**

To create a custom RGB color picker using **UISliders**, **UISwitches** and dynamically update the background color based on user input.

### **UI Components**

1. Add three **UISliders** (for Red, Green, Blue) to adjust RGB values.
2. Add a **UILabel** to display the current RGB color values (e.g., RGB(255, 0, 128)).
3. Add a **UIView** as a color preview box to show the selected color in real time.
4. Add **UISwitch** to enable and disable UISliders.

#### **1.Add Components in Storyboard**

1. **Add three sliders** to the View Controller:
   * Label them **Red**, **Green**, and **Blue**.
   * Set their minimum value to 0 and maximum value to 255 in the **Attributes Inspector**.
2. **Add a UILabel** below the sliders:
   * Set the default text to RGB(255, 255, 255).
3. **Add a UIView** as a color preview box:
   * Place it above or below the sliders.
   * Set its initial background color to **white**.
4. **Add three UISwitch** to enable and disable Sliders on Toggle.

A diagram of a cell phone

AI-generated content may be incorrect.

Code:

//

// ViewController.swift

// ColourPicker

//

// Created by KJSCE on 18/02/25.

//

**import** UIKit

**class** ViewController: UIViewController {

**@IBOutlet** **weak** **var** redSlider: UISlider!

**@IBOutlet** **weak** **var** greenSlider: UISlider!

**@IBOutlet** **weak** **var** blueSlider: UISlider!

**@IBOutlet** **weak** **var** redSwitch: UISwitch!

**@IBOutlet** **weak** **var** greenSwitch: UISwitch!

**@IBOutlet** **weak** **var** blueSwitch: UISwitch!

**@IBOutlet** **weak** **var** LockButton: UIButton!

**@IBOutlet** **weak** **var** colourView: UIView!

**@IBOutlet** **weak** **var** rgbValueLabel: UILabel!

**var** isLocked = **false**

**override** **func** viewDidLoad() {

**super**.viewDidLoad()

redSlider.value = 0.5

greenSlider.value = 0.5

blueSlider.value = 0.5

redSwitch.isOn = **true**

greenSwitch.isOn = **true**

blueSwitch.isOn = **true**

}

**@IBAction** **func** redSwitch(\_ sender: UISwitch) {

updateColor()

}

**@IBAction** **func** greenSwitch(\_ sender: UISwitch) {

updateColor()

}

**@IBAction** **func** blueSwitch(\_ sender: UISwitch) {

updateColor()

}

**@IBAction** **func** redSlider(\_ sender: UISlider) {

updateColor()

}

**@IBAction** **func** greenSlider(\_ sender: UISlider) {

updateColor()

}

**@IBAction** **func** blueSlider(\_ sender: UISlider) {

updateColor()

}

**@IBAction** **func** LockButton(\_ sender: UIButton) {

isLocked.toggle()

sender.setTitle(isLocked ? "Unlock" : "Lock", for: .normal)

// updateColor()

}

**func** updateColor() {

**if** isLocked{

**return**

}

**var** red: CGFloat = CGFloat(redSlider.value)

**var** green:CGFloat = CGFloat(greenSlider.value)

**var** blue:CGFloat = CGFloat(blueSlider.value)

**if** !redSwitch.isOn {

red = 0.0

}

**if** !greenSwitch.isOn {

green = 0.0

}

**if** !blueSwitch.isOn {

blue = 0.0

}

colourView.backgroundColor = UIColor(red: red, green: green, blue: blue, alpha: 1.0)

rgbValueLabel.text = String(format: "RGB: (%.0f, %.0f, %.0f)", red \* 255, green \* 255, blue \* 255)

}

}

Output:









