### **Lab Exercise: Creating Transition Animations Programmatically in iOS Swift**

**Objective:**

Learn how to create and implement transition animations programmatically in iOS Swift. This exercise will guide you through setting up simple transition animations between views.

**Step 1: Set Up the Project**

* Create a new Xcode project:
* Open Xcode and create a new project.
* Choose "App" under iOS.
* Name the project TransitionAnimationLab.
* Set Up the View Controller:
* Open ViewController.swift.

**Step 2: Create Views for Transition Animation**

* Add Two Views Programmatically:
* Open ViewController.swift and add two views that will transition between each other.

import UIKit

class ViewController: UIViewController {

var firstView: UIView!

var secondView: UIView!

var thirdView: UIView!

var currentViewIndex = 0

override func viewDidLoad() {

super.viewDidLoad()

view.backgroundColor = .white

setupViews()

setupTransitionButtons()

}

func setupViews() {

firstView = UIView()

firstView.backgroundColor = .red

firstView.translatesAutoresizingMaskIntoConstraints = false

view.addSubview(firstView)

NSLayoutConstraint.activate([

firstView.leadingAnchor.constraint(equalTo: view.leadingAnchor),

firstView.trailingAnchor.constraint(equalTo: view.trailingAnchor),

firstView.topAnchor.constraint(equalTo: view.topAnchor),

firstView.bottomAnchor.constraint(equalTo: view.bottomAnchor)

])

secondView = UIView()

secondView.backgroundColor = .blue

secondView.translatesAutoresizingMaskIntoConstraints = false

secondView.isHidden = true

view.addSubview(secondView)

NSLayoutConstraint.activate([

secondView.leadingAnchor.constraint(equalTo: view.leadingAnchor),

secondView.trailingAnchor.constraint(equalTo: view.trailingAnchor),

secondView.topAnchor.constraint(equalTo: view.topAnchor),

secondView.bottomAnchor.constraint(equalTo: view.bottomAnchor)

])

thirdView = UIView()

thirdView.backgroundColor = .green

thirdView.translatesAutoresizingMaskIntoConstraints = false

thirdView.isHidden = true

view.addSubview(thirdView)

NSLayoutConstraint.activate([

thirdView.leadingAnchor.constraint(equalTo: view.leadingAnchor),

thirdView.trailingAnchor.constraint(equalTo: view.trailingAnchor),

thirdView.topAnchor.constraint(equalTo: view.topAnchor),

thirdView.bottomAnchor.constraint(equalTo: view.bottomAnchor)

])

}

func setupTransitionButtons() {

let transitionButton = UIButton(type: .system)

transitionButton.setTitle("Next Transition", for: .normal)

transitionButton.translatesAutoresizingMaskIntoConstraints = false

transitionButton.addTarget(self, action: #selector(performTransition), for: .touchUpInside)

view.addSubview(transitionButton)

NSLayoutConstraint.activate([

transitionButton.centerXAnchor.constraint(equalTo: view.centerXAnchor),

transitionButton.bottomAnchor.constraint(equalTo: view.bottomAnchor, constant: -50)

])

}

@objc func performTransition() {

let views = [firstView!, secondView!, thirdView!]

let fromView = views[currentViewIndex]

currentViewIndex = (currentViewIndex + 1) % views.count

let toView = views[currentViewIndex]

let options: UIView.AnimationOptions = [.transitionFlipFromLeft, .showHideTransitionViews]

UIView.transition(from: fromView, to: toView, duration: 1.0, options: options, completion: nil)

}

}

**Configure Views for Transition:**

Set up two full-screen views (firstView and secondView), and initially hide the second view.