SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA

CLASS WORK
SESSIONAL WORK

ASSIGNMENT NO- 1
EXPERIMENT

SUBMITTED ON ..... MARKS OR GRADE OBTAINED .....

NAME Yash Goswami ROLL NO. 19100BTCSEMA05507

CLASS CS-K III rd year DEPARTMENT CS (B.E)

SUBJECT VR & Aug. reality CODE NO. .....

Signature of Student Signature of Professor

Q1	What is ARKit ? Why we use it ? ARKit → It acts as a platform for you to develop your augmented reality ~~world~~ applications. To help us we get familier using ARKit. By using ar-kit with the help of it we can create augmented reality appli-cations by focusing on the unique features of our application rather than on the details of detecting, displaying and drawing Virtual objects on the real world entity. It is an Integrated ios device camera and motion feature to produce augmented reality experiences on/in our application or game.

Lab Code | Roll No. | Exp. No. | Page

Q. Develop an simple iOS application to add an 3-D object using scenekit.

```
import SceneKit
Import ARKit
class ViewController : UIViewController
ARSCNViewDelegate {

override func viewWillAppear (animate : Bool)
{
  super.viewWillAppear (animated)

let configuration = ARWorldTracking
  configuration()

sceneView.session.run (configuration)
}
override func viewWillDisappear
(-animated : Bool) {
  super.viewWillDisappear (animated)
  sceneView.session.pause()
}
@IBOutlet var sceneView : ARSCNView!
override func viewDidLoad()
  sceneView.delegate = self
sceneView.showStatistics = true.
```

let scene = SCNScene (named: "art.scnassets/Satellite.scn")!
sceneView = scene
}

Q3 Define the only uses of SpriteKit, RealityKit ?

→ SpriteKit

It is used to add high performance of 2D content with smooth animations to our application, or to create a game with a high level set of 2-Dimentional game based tools.

It is a general purpose framework for drawing shapes, particles, text, images and video on two dimension. It leverages metal to achieve high performance rendering, while offering a simple programming interface to make it easy to create games and other graphics-intensive applications.

→ RealityKit

We use the realityKit framework to implement high performance 3-Dimentional

Lab Code | Roll No. | Exp. No. | Page

Simulation and rendering.
It leverages information provided by the ARKit framework to seamlessly integrate virtual objects into the real world.

SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA

CLASS WORK
SESSIONAL WORK

ASSIGNMENT NO-2
EXPERIMENT

SUBMITTED ON ..... MARKS OR GRADE OBTAINED .....

NAME Yash Goswami ROLL NO. 19IRVBTCSEMA055O7

CLASS CS-K (Mob. app^n) DEPARTMENT CSE (B.Tech)

SUBJECT VR and Aug. Reality CODE NO. .....

Signature of Student Signature of Professor

Q 1 Create a single view app^n to facilitate a view AR eg. to show an object on the current view of camera.

```
Import UIKit
Import SceneKit
Import ARKit
class ViewController : UIViewController
ARSCNViewDelegate {

@IBOutlet var sceneView :
    ARSCNView !
override func viewDidLoad () {
    super.viewDidLoad ()
    sceneView.delegate = self
    sceneView.showStatistics = true
    sceneView.debugOptions = [ ARSCN
        - DebugOptions.showFeaturePoints]
}
Override func viewWillAppear
    (_ animated : Bool) {
```

Lab Code | Roll No. | Exp. No. | Page

```
super.viewWillAppear(animated)
let configuration = ARWorldTrackingConfiguration()
sceneView.session.run(configuration)
}

@IBAction func resetButton(sender: UIButton)
{
  sceneView.session.pause()
  sceneView.session.run(configuration,
    options: [.resetTracking])
}
func showShape() {
  let node = SCNNode()
  node.geometry = SCNSphere(radius: 0.03)
~~node.geometry = SCNSphere~~
node.geometry?.firstMaterial?.diffuse.
  contents = UIColor.blue
  node.position = SCNVector3(0,0,0)
sceneView.scene.rootNode.addChildNode(node)
}
}
```

Lab Code | Roll No. | Exp. No. | Page

Q2) Write a program for adding node in the AR program created.

```
import UIKit
import ARKit
import SceneKit
class ViewController : UI ViewController,
    ARSCNViewDelegate {
@IBOutlet var sceneView : ARSCNView!
@IBOutlet var xSlider : UI Slider!
@IBOutlet var YSlider : UI Slider!
@IBOutlet var z slider: UI Slider!

let configuration = AR worldTracking configuration()
override func ViewDidLoad () {
    super.View.Delegate = self
    SceneView.showStatistics = true
    sceneView.debugOptions = [ ARSCN Debug
        options.showWorldOptions ]
    }
    override func ViewWillAppear (animated: Bool)
{
    super.ViewWillAppear(animated)
    sceneView.session.run(configuration) }
@IBAction func addButton (sender: UI Button)
{ showShape () }
```

Lab Code | Roll No. | Exp. No. | Page

```
Func showShape() {
    let node = SCNNode()
node.geometry = SCNSphere(radius: 0.8)
node.geometry?.firstMaterial.diffuse.
    contents = UIColor.blue
node.position = SCNVector3(xSlider.value,
    ySlider.value, zSlider.value)
node.name = "Sphere"
sceneView.scene.rootNode.addChildNode(node)
}
}
```

SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA

CLASS WORK
SESSIONAL WORK

ASSIGNMENT NO 3
EXPERIMENT

SUBMITTED ON ..... MARKS OR GRADE OBTAINED .....

NAME Yash Goswami ROLL NO. 19100

CLASS CSK-MA (Apple) DEPARTMENT CSK-Mobile

SUBJECT AR & Aug reality CODE NO. .....

Signature of Student Signature of Professor

Q1 Write a program for resetting the world origin in AR app.

=>

```
import UI Kit
import Scene Kit
import AR Kit.

class ViewController : UI View Controller
ARSCNView Delegate {

@IBOutlet var sceneView : ARSCNView!

let configuration = AR World Tracking
                    configuration ()
@IBAction func resetButton (sender:
                    UI Button)
{
sceneView.session.pause()
sessionView.session.run (configuration,
  option : [.resetTracking]) }
```

Lab Code | Roll No. | Exp. No. | Page

```
override func viewDidLoad() {
    super.viewDidLoad()
sceneView.delegate = self
sceneView.debugOptions = [ARSCNDebugOptions
                .showWorldOrigin]
sceneView.debug showStatistics = true
}
override func viewWillAppear(animated: Bool)
{ super.viewWillAppear(animated)
sceneView.session.run(configuration)
}}
```

(Q2) CAP for adding sphere on the position related to the current world origin.

```
import UIKit
import SceneKit
import ARKit

class ViewController: UIViewController,
    ARSCNViewDelegate {
@IBOutlet var sceneView: ARSCNView!
@IBOutlet var x slider: UISlider!
@IBOutlet var y slider: UISlider!
```

Lab Code | Roll No. | Exp. No. | Page

```
@IBOutlet var xSlider : UISlider!

let configuration = ARWorldTrackingConfiguration()
override func viewDidLoad() {
    super.viewDidLoad()
    sceneView.delegate = self
    sceneView.showStatistics = true
    sceneView.debugOptions = [ARSCNDebug
        Options.showWorldOrigin]
    }
override func viewWillAppear(animated: Bool)
{
 super.viewWillAppear(animated)
 sceneView.session.run(configuration)
}
 @IBAction func addButton(sender:UIButton)
{ showShape()
    }
     func showShape() {
     let node = SCNNode()
     node.geometry = SCNSphere(radius:0.01)
node.geometry?.firstMaterial.diffuse.
     contents = UIColor.blue
     node.position = SCNVector3(xSlider.value,
```

Lab Code | Roll No. | Exp. No. | Page

```
Y slider.value, Z slider.value)
node.name = "Sphere"
scene.scene.rootNode.addChildNode
(node)
}
}.
```