# SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA INDORE

| CLASS WORK |
| --- |
| SESSIONAL WORK |

| ASSIGNMENT No 1 |
| --- |
| EXPERIMENT |

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

NAME Anushka Tomar ROLL NO 191400BTCSEMA05474

CLASS CS-K IIIrd year DEPARTMENT CS (B. Tech.)

SUBJECT V.R and Arg 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 augment reality application. To help us we get familier using ARKit.

ARKit combines device motion tracking, camera scene capture, advanced scene processing, and display conveniences to simplify the task of building an AR experience. You can create many kinds of AR experiences with these technologies using the front or rear camera of an iOS device.

It is an integrated ios device camera and motion feature to produce augmented reality experiences on/in ~~on~~ our application or game.

Q23 Develop an simple view app to add an 3D object using Scene kit

Ans - import sceneKit
import ARKit
class ViewController : UIViewController
ARSCNViewDelegate {

Override func viewWillAppear (animated: Bool)
{
super.viewWillAppear (animated)
let configuration()
sceneView.session.run (configuration)
}
override func viewWillDisappear
(_ animated : Bool)
Super.viewWillDisappear (animated)
let configuration = ARWorldTrackingConfiguration()
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/stellite.scn")!
sceneView.scene = scene
}

Q3→ Define the only uses of Spritekit, Realitykit?

Ans→ SpritKit:-

Spritekit is a general purpose framework for drawing shapes, particles, text, images and video in two dimention. 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 apps. Using a rich set of animations and physics behaviors, you can quickly add life to your visual elements and gracefully transition between screens.

Realitykit.→

Use the RealityKit framework to implement high-performance 3D simulation and rendering. RealityKit leverage information provided by the ARKit framework to seamlessly integrate virtual objects into the real world.

# Assignment - 2

Q1 → Create a Single view application to facilitate with AR example to how an object onto 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 = [ARSCNDebugOptions.
                          showWorldOrigin]
}

Override func viewwillAppear(_animated:Bool){
super.viewwillAppear(animated)
let Configuration = AR Tracking Configuration
SceneView.Session.run(configuration)
}

@IBAction func resetButton(_Sender:UIButton)
{
sceneview.session.pause()
sceneview.session.run(Configuration.option:
                      [.resetTracking]).
{
```

```
Func showshape() {
  let node = SCNNode()
  node.geometry = SCN sphere(radius: 0.05)
  node.geometry?.first.Material?.diffuse.
  Contents = UIcolor.blue
  node.position = SCNVector3(0,0,0)
  sceneView.scene.rootNode.addChildNode
                              (CNode)
    }
  }
```

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

```
import UIKit
import ARkit
import Scenekit
class ViewController : UIViewController,
    ARSCNViewDelegate {
@IBoutlet var SceneView : ARSCNView!
@IBoutlet var Xslider : UISlider!
@IBoutlet var Yslider : UISlider!
@IBoutlet var Zslider : UISlider!
let Configuration = ARWorldTrackingConfiguration()
overrid fun ViewDidLoad() {
  Super.View.Delegate = Self
  SceneView.showStatistics = [ARSCN RedueObjects
  option.showWorldOptions]
}
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 = SCNNode()
    node.geometry?.firstMaterial.diffuse
    contents = UIColor.blue
    node.position = SCNVector3(xSlider.value,
    ySlider.value, zSlider.value)
    node.name = "Sphere"
    sceneView.scene.rootNode.addChild
                        Node(node)
    }
}
```

# SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA INDORE

| CLASS WORK |
| --- |
| SESSIONAL WORK |

| ASSIGNMENT / EXPERIMENT No 3 |
| --- |

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

NAME ..... ROLL NO .....

CLASS ..... DEPARTMENT .....

SUBJECT ..... CODE NO .....

Signature of Student Signature of Professor

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

```
import UIKit
import SceneKit
import ARKit.
Class ViewController : UIViewController
ARSCNViewDelegate {
@IBoutlet var sceneview : ARSCNView!
let configuration = ARWorldTracking
                    configuration()
@IBAction func resetButton(Sender: UI
                           Button)
{
  Sceneview.session.pause()
  Sceneview.session.run(configuration,
  option : [resetTracking,]),}
Override func viewDidload() {}
  Super.viewDidload()
  sceneview.delegate = self
  Sceneview.debugoption = [ARSCNDebug
                          -option2.showworldorigin]
```

{ Sceneview . showstatistics = true
override fun viewwillAppear (animated : Bool)
{ Super . viewwillAppear (animated)
Sceneview . session . run (configuration { }

P2 → Dap adding sphere in the position related to the current world origin

import UIKit
import scenekit
import ARKit
class viewcontroller : UIView Controller, ARSCNView Delegate {

@ IBoutlet var scene view : ARSCN view!
@ IBoutlet var X Slider : UI slider!
@ IBoutlet var Y slider : UI slider!
@ IBoutlet var Z Slider : UI Slider!
let configuration = ARWorld Tracking Configuration ()
Override fun viewDidload () {
Super . viewDidload ()
Sceneview . show . delegate = self
Sceneview . showstatistics = true
Sceneview . debugoptions = [ARSCN Debug Option
. { Show World Origin]

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: 0.04)
node.geometry?.firstMatrial.diffuse.
    contents = UIColor.blue
    node.position = SCNVector3 (xslider.value,
Yslider.value, Zslider.value)
    node.name = "sphere"
    sceneView.scene.rootNode.add
    Child Node (Node)
    }
    }
```