

# ARKit を用いた AR アプリ制作

## 1. 目的・動機

近未来の AR 機器を使用したアクションアニメ映画をみて憧れを持ったから。

## 2. 使用機器、ソフトウェア

- Macbook Air 13 インチ (Apple M1 チップ)
- iPhone 13mini (ios15 以上)
- Xcode Ver13.2.1
- Blender Ver2.93.3

## 3. 作業内容

AR アプリの制作

< 1 学期 >

Macbook を使用したアプリ制作環境に慣れるため、ネットや本で紹介されている簡単なアプリを制作した。

～制作したアプリの種類～

- 特徴点の表示
- 平面の生成
- タップした地面を崩す
- タップでオブジェクトの生成

< 2 学期 >

(1) 1 学期に引き続き、サンプルアプリを制作し、その中から 1 つのアプリに改良を加え、発表するための作品を完成させた。

～制作したアプリの種類～

- MotionCapture
- PeopleOcclusion
- FaceTracking ※発表用として完成させた。

```
1 import UIKit
2 import SceneKit
3 import ARKit
4
5 //フェイストラッキング(ブレンドシェイプのキャラクターの表示)
6 class ViewController: UIViewController, ARSCNViewDelegate {
7     @IBOutlet var sceneView: ARSCNView!
8
9     //ノード
10    private var originalJawY: Float = 0
11    private var jawNode: SCNNode!
12    private var eyeLeftNode: SCNNode!
13    private var eyeRightNode: SCNNode!
14    private var jawHeight: Float = 0.0
15
16    //ローに常に呼び出される
17    override func viewDidLoad() {
18        super.viewDidLoad()
19
20        //シーンの作成
21        sceneView.scene = SCNScene()
22
23        //ARSCNViewプロパティの設定
24        sceneView.delegate = self
25    }
26
27    //ビュー表示時に呼び出される
28    override func viewWillAppear( animated: Bool) {
29        super.viewWillAppear(animated)
30    }
31
32    //フェイストラッキングの開始
33    let configuration = ARFaceTrackingConfiguration()
34    configuration.isLightEstimationEnabled = true
35    sceneView.session.run(configuration)
36 }
```

```
37 //ビュー表示時に呼び出される
38 override func viewWillDisappear( animated: Bool) {
39     super.viewWillDisappear(animated)
40     //セッション一時停止
41     sceneView.session.pause()
42 }
43
44 //追加されたARアンカーに対応するノードを提供
45 func renderer( renderer: SCNSceneRenderer, nodeFor anchor: ARAnchor) -> SCNNode? {
46     //ARFaceAnchorの取得
47     guard let faceAnchor = anchor as? ARFaceAnchor else { return nil }
48
49     //キャラクターのノードの生成
50     let contentNode = SCNReferenceNode(named: "face")
51     let jawNode = contentNode.childNode(withName: "jaw", recursively: true)!
52     let eyeLeftNode = contentNode.childNode(withName: "eyeLeft", recursively: true)!
53     let eyeRightNode = contentNode.childNode(withName: "eyeRight", recursively: true)!
54     let (min, max) = jawNode.boundingBox
55     let originalJawY = jawNode.position.y
56     return contentNode
57 }
58
59 //ARアンカーの位置をARアンカーに対応するノードに反映し正確に呼び出される
60 func renderer( renderer: SCNSceneRenderer, didUpdate node: SCNNode, for anchor: ARAnchor) {
61     //ARFaceAnchorの取得
62     guard let faceAnchor = anchor as? ARFaceAnchor else { return }
63
64     //ブレンドシェイプの値の取得
65     let blendShapes = faceAnchor.blendShapes
66     //キャラクターの目の位置に値の更新
67     eyeLeftNode.scale.z = 1 - eyeBlinkLeft
68     eyeRightNode.scale.z = 1 - eyeBlinkRight
69     jawNode.position.y = originalJawY - jawHeight + jawOpen
70 }
71
72 //SCNReferenceNodeの拡張
73 extension SCNReferenceNode {
74     //リソース名でノードを生成
75     convenience init(named resourceName: String, loadImmediately: Bool = true) {
76         let url = Bundle.main.url(forResource: resourceName,
77                                 withExtension: ".scn", subdirectory: "art.scnassets")!
78         self.init(url: url!)
79         if loadImmediately {
80             self.load()
81         }
82     }
83 }
```

[プログラム]

(2) Xcode 以外での AR の開発環境に触れる。

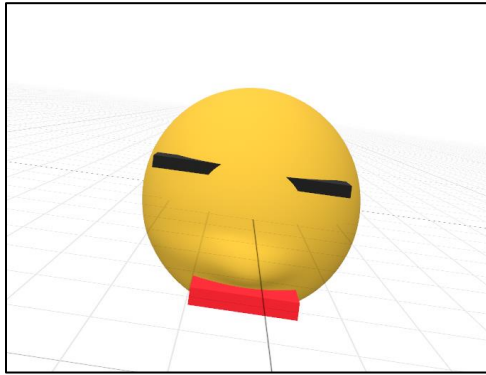
～挑戦した開発環境～

- ・ Unity での iPhone 向け AR アプリの開発
- ・ Adobe Aero と Xcode を用いた AR アプリの開発

< 3 学期 >

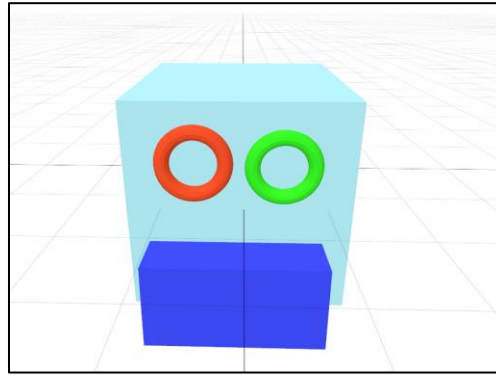
- ・ マスクの制作

Blender を用いて「マスク」をモデリングした。

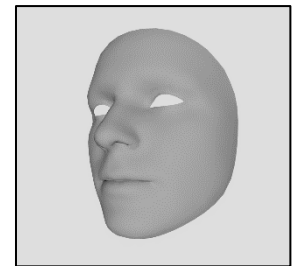


[デコイ]

- ・ 発表用アプリ FaceTraking の完成
- ・ レポート及びパワーポイントの作成



[ロボット]



[マスク]

#### 4. 作品内容 (FaceTraking)

FaceTraking とは顔の動きに合わせてマスクが動く AR 技術の一つである。

瞬きするとマスクの目が閉じ、口を開くとオブジェクトが上下する。

また、顔を横に動かすとマスクがついてくるように動く。

例) Snow (スマホアプリ)

3tene (VTube 等)

#### 5. 感想

私はこの課題研究を通じて改めて自分が作ったプログラムが動く楽しさを感じる事ができた。今回の課題研究は普段授業や実習とは違い、時間の都合上細かい構文から学ぶことができなかったため、プログラムの具体的な意味を理解することができなかった。基本的には、ネット等のサンプルプログラムを利用し、少し改良を加えてプログラムの仕組みを理解することに徹した。ネットに載っているサンプルプログラムの大半は解説がプログラムの真横に書いてあるので効率よく学習することができた。

この課題研究で使用した Xcode という慣れない開発環境でのプログラミングや実機での実行は簡単ではなかった。サンプルプログラムを導入してもバージョンの違いによるエラーが発生することも少なくはなかった。改良を加えた際も同じようにエラーが出ることもあった。しかし、どんなエラーなのかもわからない状況で解決方法を探し、試行錯誤をしているうちに修正できた時はプログラミングの本当の楽しさが分かった。初めての環境でうまくできるのか不安であったが、完成までもっていくことができてよかった。

#### 6. 参考文献

- ・ ARKit / フェイストラッキング npaka 著  
(<https://note.com/npaka/n/nb7723bla5567>)
- ・ Qiita  
(<https://qiita.com/>) ・ Swift で作る
- ・ ARKit 超入門第 3 版 北村愛美著