

# Hand Tracking and Gesture Recognition in VR

20172122 조민지

## | 선정 배경

- 허공 제스처: 허공의 손동작으로 컴퓨터와 상호작용하는 방법



- ▲ 컨트롤러를 사용하여 손 제스처를 인식하는 컨트롤러(leap motion)

## | 선정 배경

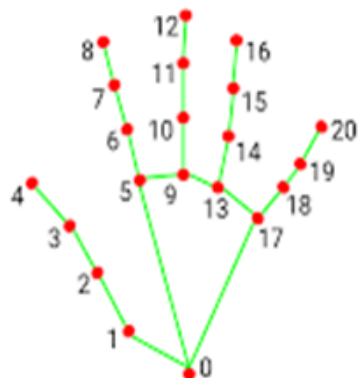
- 허공 제스처: 허공의 손동작으로 컴퓨터와 상호작용하는 방법



▲ 컨트롤러를 사용하여 손 제스처를 인식하는 컨트롤러(leap motion)

## | 구현방법

1. mediapipe Hand를 이용하여 웹캠에서 실시간 손 인식 구현 (main.py)

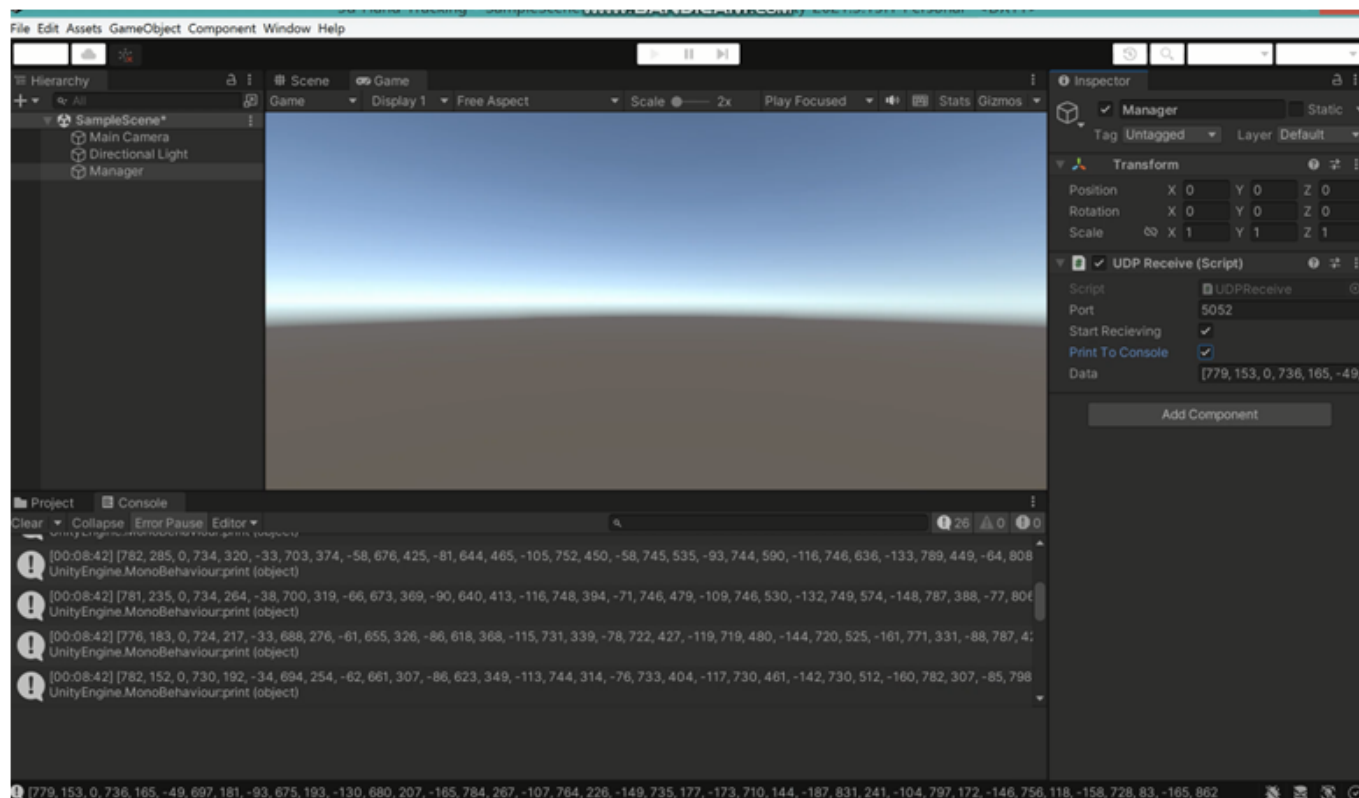


- |                       |                       |
|-----------------------|-----------------------|
| 0. WRIST              | 11. MIDDLE_FINGER_DIP |
| 1. THUMB_CMC          | 12. MIDDLE_FINGER_TIP |
| 2. THUMB_MCP          | 13. RING_FINGER_MCP   |
| 3. THUMB_IP           | 14. RING_FINGER_PIP   |
| 4. THUMB_TIP          | 15. RING_FINGER_DIP   |
| 5. INDEX_FINGER_MCP   | 16. RING_FINGER_TIP   |
| 6. INDEX_FINGER_PIP   | 17. PINKY_MCP         |
| 7. INDEX_FINGER_DIP   | 18. PINKY_PIP         |
| 8. INDEX_FINGER_TIP   | 19. PINKY_DIP         |
| 9. MIDDLE_FINGER_MCP  | 20. PINKY_TIP         |
| 10. MIDDLE_FINGER_PIP |                       |



## | 구현방법

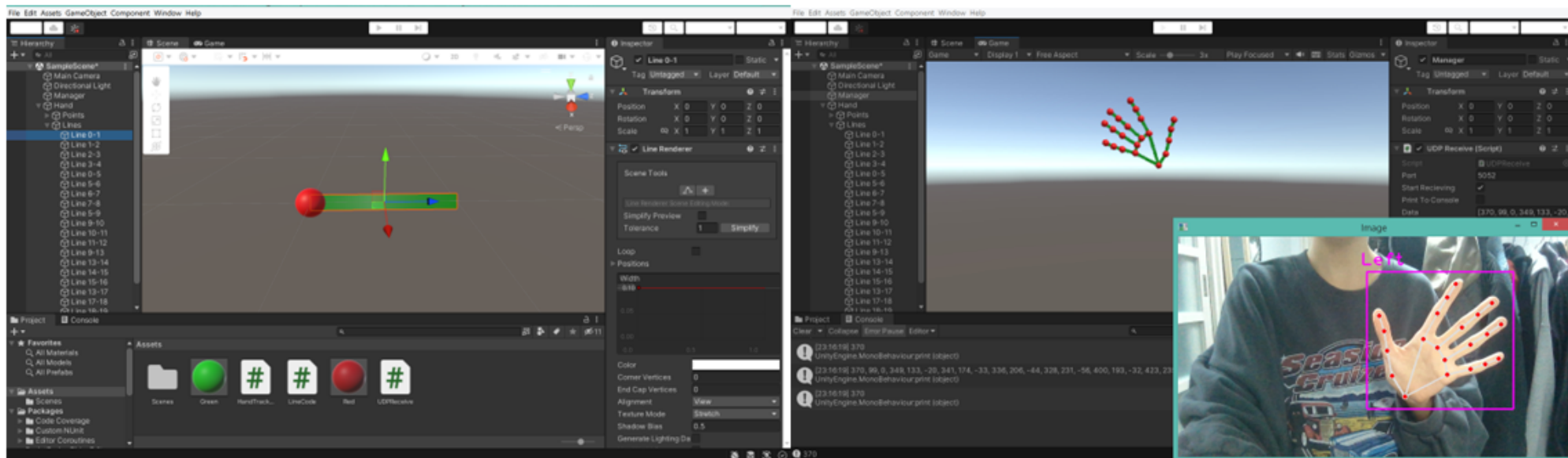
### 2. 랜드마크 좌표 추출, UDP(User Data Protocol) 이용하여 파이썬에서 유니티로 데이터 전송 (UDPReceive.cs)



▲ 파이썬 실행 후 유니티 실행하여 웹캠에 손 움직이면 랜드마크의 좌표값들이 콘솔창에 나타남

## | 구현방법

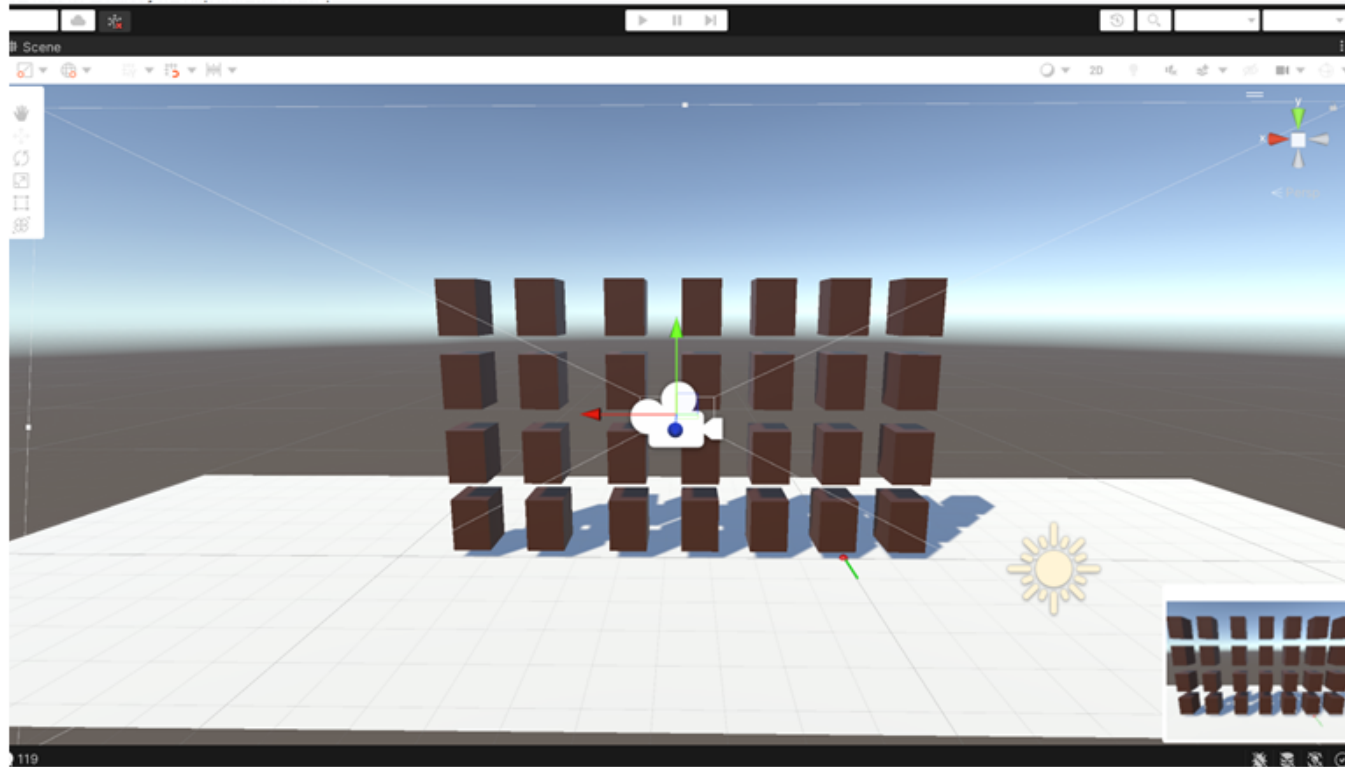
3. Hand Tracking. 손 모양을 나타내기 위해 오브젝트 생성 후 랜드마크 좌표값에 나타냄.  
(LineCode.cs, HandTracking.cs)



▲ 유니티 실행 시 손의 실시간 움직임을 받아 화면에 표시

## | 구현방법

4. 오브젝트로 바닥(Plane)과 블록(Cube) 생성. 각 블록에 Rigidbody 추가



## 소스코드

```
import cv2
from cvzone.HandTrackingModule import HandDetector
import socket

# Parameters
width, height = 1280, 720

# Webcam
cap = cv2.VideoCapture(0)
cap.set(3, 1280)
cap.set(4, 720)

# Hand Detector
detector = HandDetector(maxHands=1, detectionCon=0.8)

# Communication
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
serverAddressPort = ("127.0.0.1", 5052)

while True:
    # Get the frame from the webcam
    success, img = cap.read()
    # Hands
    hands, img = detector.findHands(img)

    data = []
    # Landmark values - (x,y,z) * 21
    if hands:
        # Get the first hand detected
        hand = hands[0]
        # Get the landmark list
        lmList = hand['lmList']
        # print(lmList)
        for lm in lmList:
            data.extend([lm[0], height - lm[1], lm[2]])
        # print(data)
        sock.sendto(str.encode(str(data)), serverAddressPort)

    cv2.imshow("Image", img)
    cv2.waitKey(1)
```

▲ main.py

```
1 using UnityEngine;
2 using System;
3 using System.Text;
4 using System.Net;
5 using System.Net.Sockets;
6 using System.Threading;
7
8 public class UDPReceive : MonoBehaviour
9 {
10
11     Thread receiveThread;
12     UdpClient client;
13     public int port = 5052;
14     public bool startReceiving = true;
15     public bool printToConsole = false;
16     public string data;
17     public void Start()
18     {
19         receiveThread = new Thread(
20             new ThreadStart(ReceiveData));
21         receiveThread.IsBackground = true;
22         receiveThread.Start();
23     }
24     // receive thread
25     private void ReceiveData()
26     {
27         client = new UdpClient(port);
28         while (startReceiving)
29         {
30             try
31             {
32                 IPEndPoint anyIP = new IPEndPoint(IPAddress.Any, 0);
33                 byte[] dataByte = client.Receive(ref anyIP);
34                 data = Encoding.UTF8.GetString(dataByte);
35
36                 if (printToConsole) { print(data); }
37             }
38             catch (Exception err)
39             {
40                 print(err.ToString());
41             }
42         }
43     }
44 }
```

▲ UDPReceive.cs



```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class LineCode : MonoBehaviour
6 {
7     LineRenderer lineRenderer;
8
9     public Transform origin;
10    public Transform destination;
11
12    // Start is called before the first frame update
13    void Start()
14    {
15        lineRenderer = GetComponent<LineRenderer>();
16        lineRenderer.startWidth = 0.1f;
17        lineRenderer.endWidth = 0.1f;
18    }
19
20    // Update is called once per frame
21    void Update()
22    {
23        lineRenderer.SetPosition(0, origin.position);
24        lineRenderer.SetPosition(1, destination.position);
25    }
26 }
27

```

▲ LineCode.cs

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class HandTracking : MonoBehaviour
6 {
7     // Start is called before the first frame update
8     public UDPReceive udpReceive;
9     public GameObject[] handPoints;
10    void Start()
11    {
12
13    }
14
15    // Update is called once per frame
16    void Update()
17    {
18        string data = udpReceive.data;
19
20        data = data.Remove(0, 1);
21        data = data.Remove(data.Length-1, 1);
22        print(data);
23        string[] points = data.Split(',');
24        print(points[0]);
25
26        //0      1*3      2*3
27        //x1,y1,z1,x2,y2,z2,x3,y3,z3
28
29        for (int i = 0; i<21; i++)
30        {
31
32            float x = 7-float.Parse(points[i * 3])/100;
33            float y = float.Parse(points[i * 3 + 1]) / 100;
34            float z = float.Parse(points[i * 3 + 2]) / 100;
35
36            handPoints[i].transform.localPosition = new Vector3(x, y, z);
37
38        }
39    }
40 }

```

▲ HandTracking.cs

## | 실행 화면

