

# 중간 대체 AR

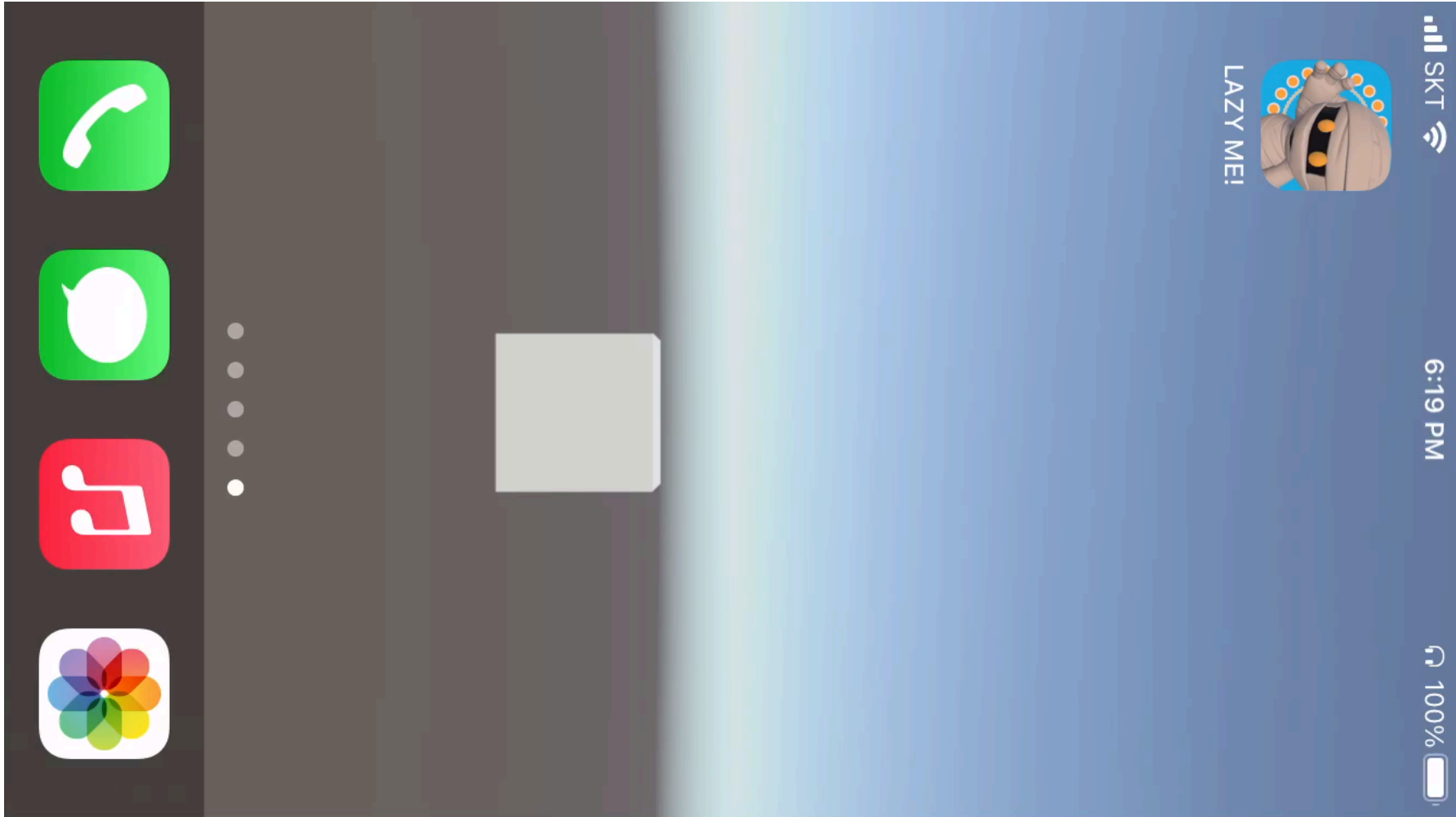
~제작자의 의도와 그 풀이~

# 만든 것

- 자기소개가 주제라 게으른 저 자신을 고백하는 AR 게임을 만들었습니다.
- 비대면 수업 동안 일반적인 제 하루 루틴을 담았습니다.

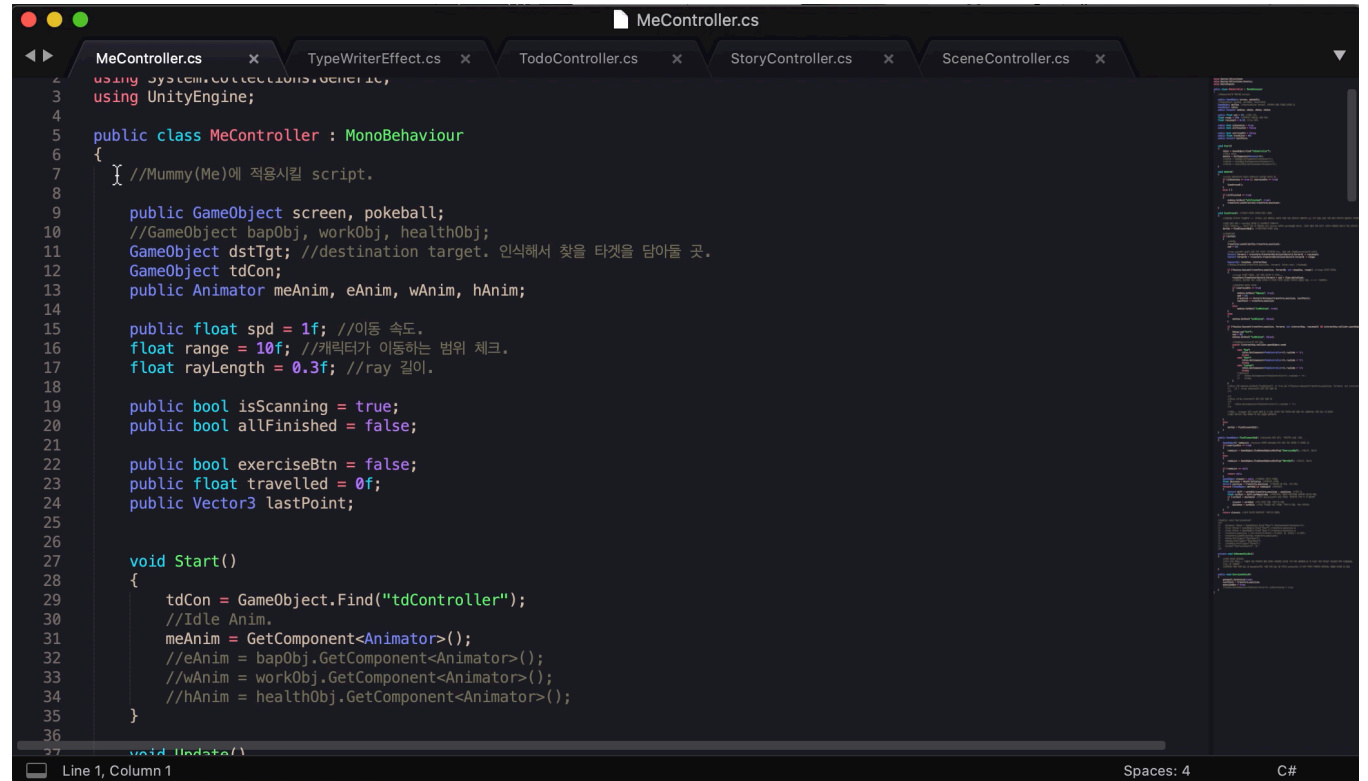


# 플레이 영상



# 주요 스크립트

- MeController
- TypeWriterEffect(Edited)
- TodoController
- StoryController
- SceneController



The screenshot shows a code editor with a dark theme. The active window is titled 'MeController.cs'. The code is in C# and defines a 'MeController' class that inherits from 'MonoBehaviour'. The class contains several public fields for game objects and animators, and a 'Start' method that initializes these fields. The code is as follows:

```
using System.Collections.Generic;
using UnityEngine;

public class MeController : MonoBehaviour
{
    //Mummy(Me)에 적용시킬 script.

    public GameObject screen, pokeball;
    //GameObject bapObj, workObj, healthObj;
    GameObject dstTgt; //destination target. 인식해서 찾을 타겟을 담아둘 곳.
    GameObject tdCon;
    public Animator meAnim, eAnim, wAnim, hAnim;

    public float spd = 1f; //이동 속도.
    float range = 10f; //캐릭터가 이동하는 범위 체크.
    float rayLength = 0.3f; //ray 길이.

    public bool isScanning = true;
    public bool allFinished = false;

    public bool exerciseBtn = false;
    public float travelled = 0f;
    public Vector3 lastPoint;

    void Start()
    {
        tdCon = GameObject.Find("tdController");
        //Idle Anim.
        meAnim = GetComponent<Animator>();
        //eAnim = bapObj.GetComponent<Animator>();
        //wAnim = workObj.GetComponent<Animator>();
        //hAnim = healthObj.GetComponent<Animator>();
    }

    void Update()
```

The editor interface includes a tab bar at the top with other files like 'TypeWriterEffect.cs', 'TodoController.cs', 'StoryController.cs', and 'SceneController.cs'. The status bar at the bottom indicates 'Line 1, Column 1', 'Spaces: 4', and 'C#'.

# 참고 스크립트

- Coroutine 이용,  
텍스트를  
타자기 효과로  
출력하는  
스크립트.
- <https://welcomeheesuk.tistory.com/m/38>

```
1 using UnityEngine;
2 using System.Collections;
3 using UnityEngine.UI;
4
5 public class TypewriterEffect : MonoBehaviour
6 {
7     //변경할 변수
8     public float delay;
9     public float Skip_delay;
10    public int cnt;
11
12    //타이핑효과 변수
13    public string[] fulltext;
14    public int dialog_cnt;
15    string currentText;
16
17    //타이핑확인 변수
18    public bool text_exit;
19    public bool text_full;
20    public bool text_cut;
21
22    //시작과 동시에 타이핑시작
23    void Start()
24    {
25        Get_Typing(dialog_cnt,fulltext);
26    }
27
28
29
30    //모든 텍스트 호출완료시 탈출
31    void Update()
32    {
33        if(text_exit==true)
34        {
35            gameObject.SetActive(false);
36        }
37    }
38
39
40    //다음버튼함수
41    public void End_Typing()
42    {
43        //다음 텍스트 호출
44        if (text_full == true)
45        {
46            cnt++;
47            text_full = false;
48            text_cut = false;
49            StartCoroutine(ShowText(fulltext));
50        }
51        //텍스트 타이핑 생략
52        else
53        {
54            text_cut = true;
55        }
56    }
57 }
```

```
58 //텍스트 시작호출
59 public void Get_Typing(int _dialog_cnt, string[] _fullText)
60 {
61     //재사용을 위한 변수초기화
62     text_exit = false;
63     text_full = false;
64     text_cut = false;
65     cnt = 0;
66
67     //변수 불러오기
68     dialog_cnt = _dialog_cnt;
69     fulltext = new string[dialog_cnt];
70     fulltext = _fullText;
71
72     //타이핑 코루틴시작
73     StartCoroutine(ShowText(fulltext));
74 }
75
76 IEnumerator ShowText(string[] _fullText)
77 {
78     //모든텍스트 종료
79     if (cnt >= dialog_cnt)
80     {
81         text_exit = true;
82         StopCoroutine("showText");
83     }
84     else
85     {
86         //기존문구clear
87         currentText = "";
88         //타이핑 시작
89         for (int i = 0; i < _fullText[cnt].Length; i++)
90         {
91             //타이핑중도탈출
92             if (text_cut == true)
93             {
94                 break;
95             }
96             //단어하나씩출력
97             currentText = _fullText[cnt].Substring(0, i + 1);
98             this.GetComponent<Text>().text = currentText;
99             yield return new WaitForSeconds(delay);
100        }
101        //탈출시 모든 문자출력
102        Debug.Log("Typing 종료");
103        this.GetComponent<Text>().text = _fullText[cnt];
104        yield return new WaitForSeconds(Skip_delay);
105
106        //스킵 지연후 종료
107        Debug.Log("Enter 대기");
108        text_full = true;
109    }
110 }
111 }
112 }
```

# 참고 스크립트

- FindGameObjectsWithTag(),  
GameObject Array 생성해  
이분 비교 반복으로 가장  
가까이 있는 특정 태그  
gameObject 반환.
- 가장 가까이 있는 오브젝트  
판별에 사용.

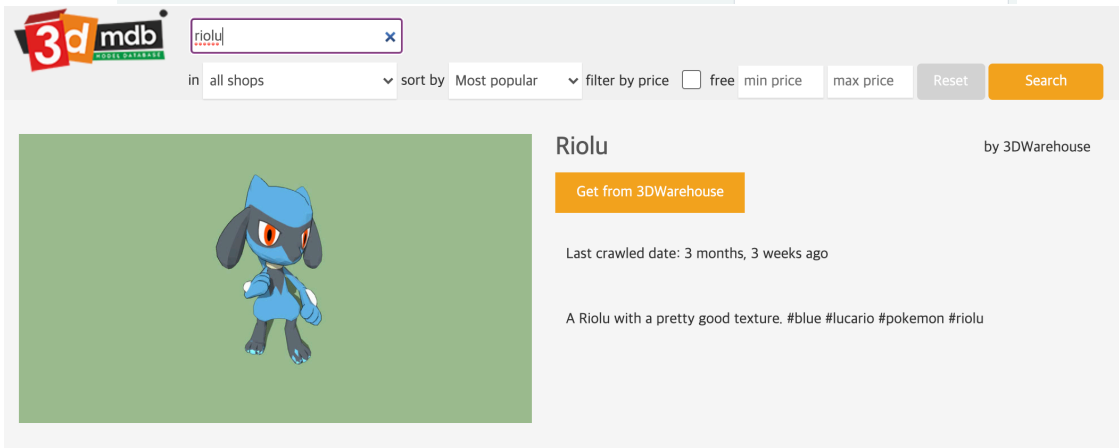
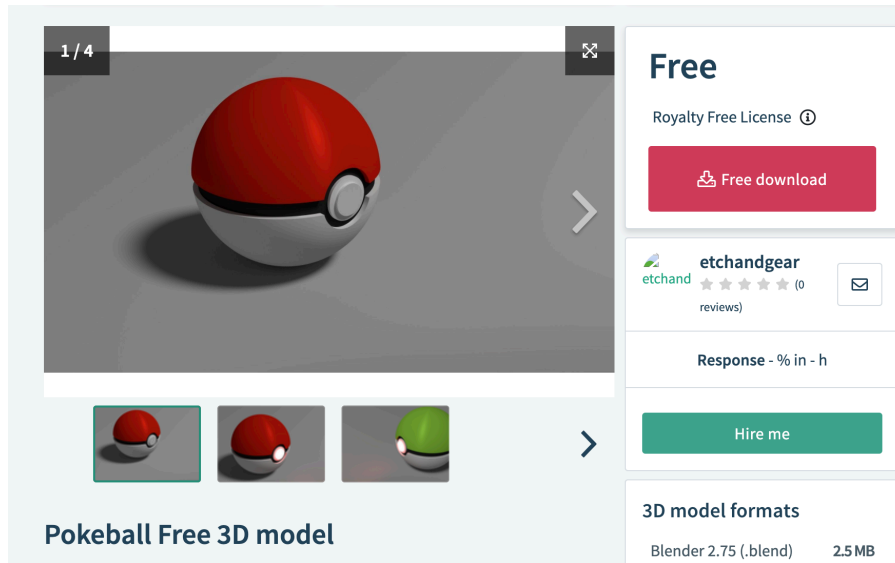
Another example:

```
// Find the name of the closest enemy

using UnityEngine;
using System.Collections;

public class ExampleClass : MonoBehaviour
{
    public GameObject FindClosestEnemy()
    {
        GameObject[] gos;
        gos = GameObject.FindGameObjectsWithTag("Enemy");
        GameObject closest = null;
        float distance = Mathf.Infinity;
        Vector3 position = transform.position;
        foreach (GameObject go in gos)
        {
            Vector3 diff = go.transform.position - position;
            float curDistance = diff.sqrMagnitude;
            if (curDistance < distance)
            {
                closest = go;
                distance = curDistance;
            }
        }
        return closest;
    }
}
```

# 사용 예셋



LUMO-ART 3D  
FREE Casual...  
3.7 MB  
Purchased: a day ago  
Organization: so...

Version: 1.0 • Apr 19, 2017

First release

[+ Add label](#) [Hide asset](#)



NEXT LEVEL 3D  
HQ Laptop C...  
7.1 MB  
Purchased: a day ago  
Organization: so...

Version: 1.01 • May 10, 2016

Texture improvement. Minor fixes.

[+ Add label](#) [Hide asset](#)



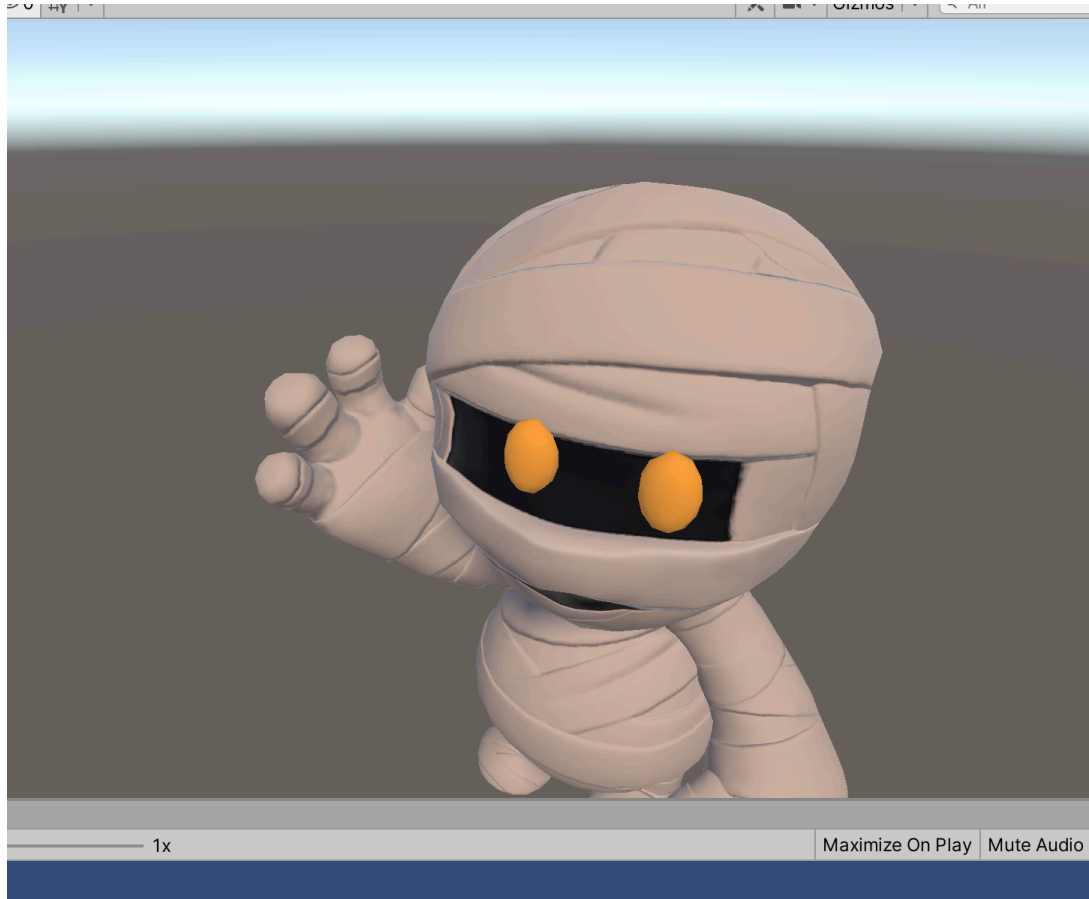
AMUSEDART  
Free Mummy...  
1.6 MB  
Purchased: 4 months ago  
Organization: so...

Version: 1.0 • Dec 4, 2018

First release

[+ Add label](#) [Hide asset](#)

# 직접 제작 이미지 – 앱 아이콘





# 직접 제작 이미지 – 스토리 씬과 아이콘

