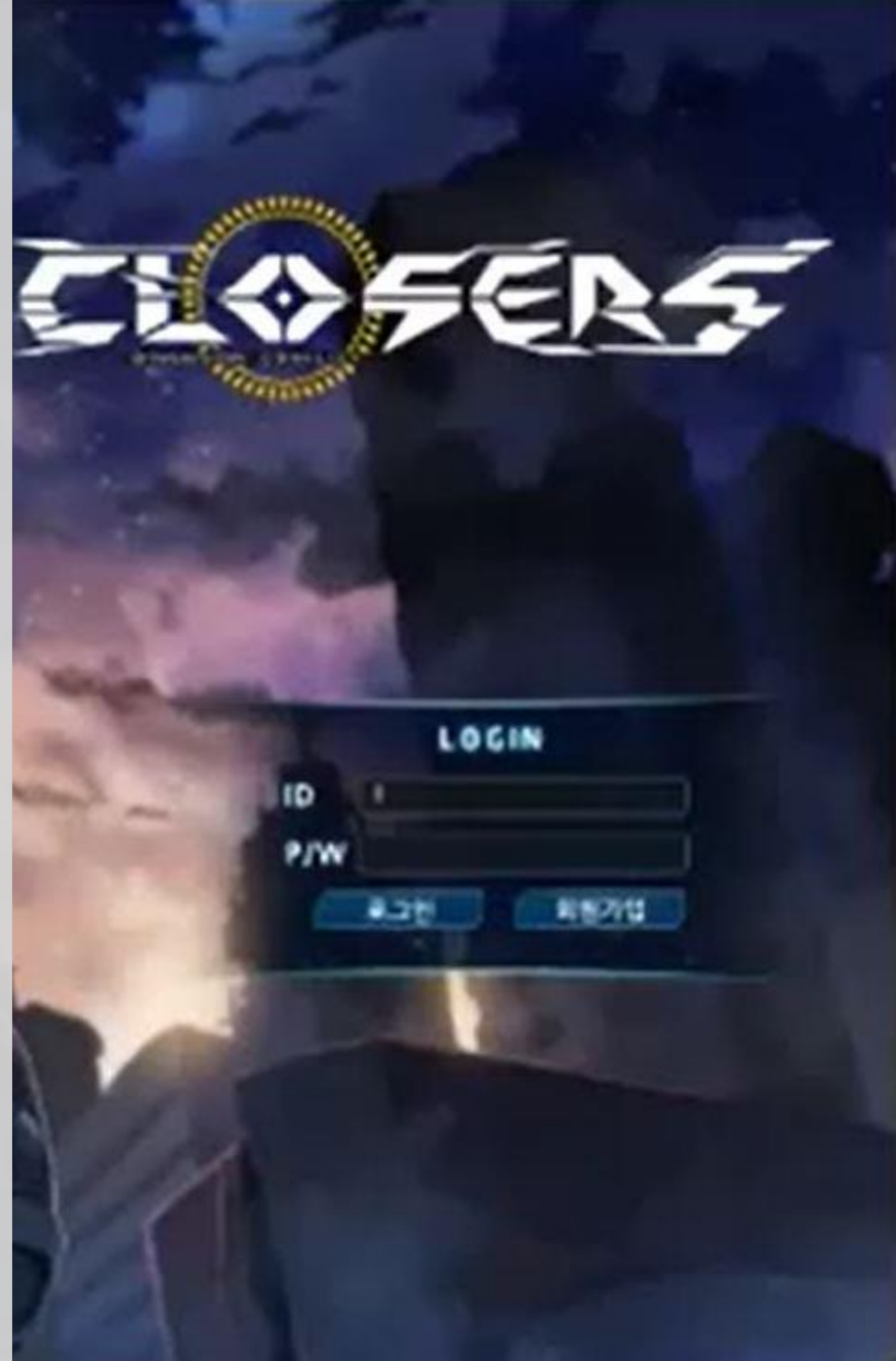




Unity 포트폴리오 소개서

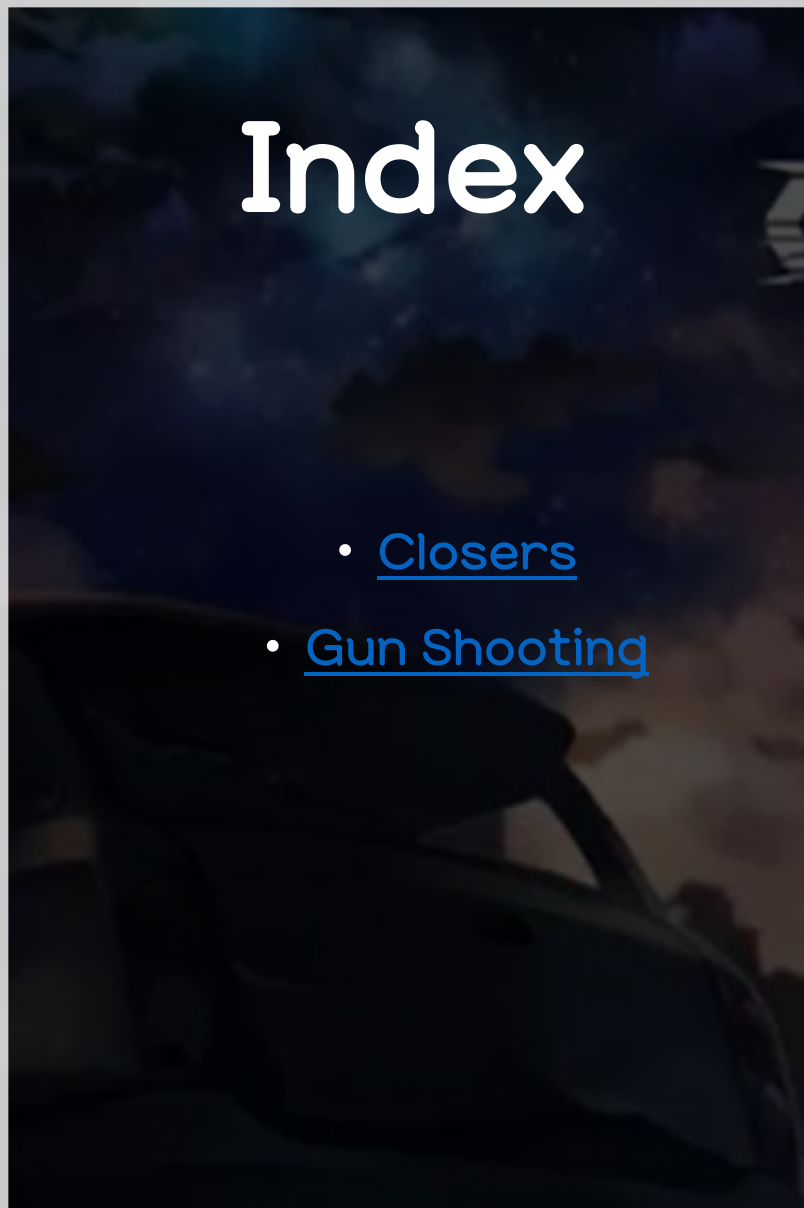
정다경





Index

- [Closers](#)
- [Gun Shooting](#)



Closers

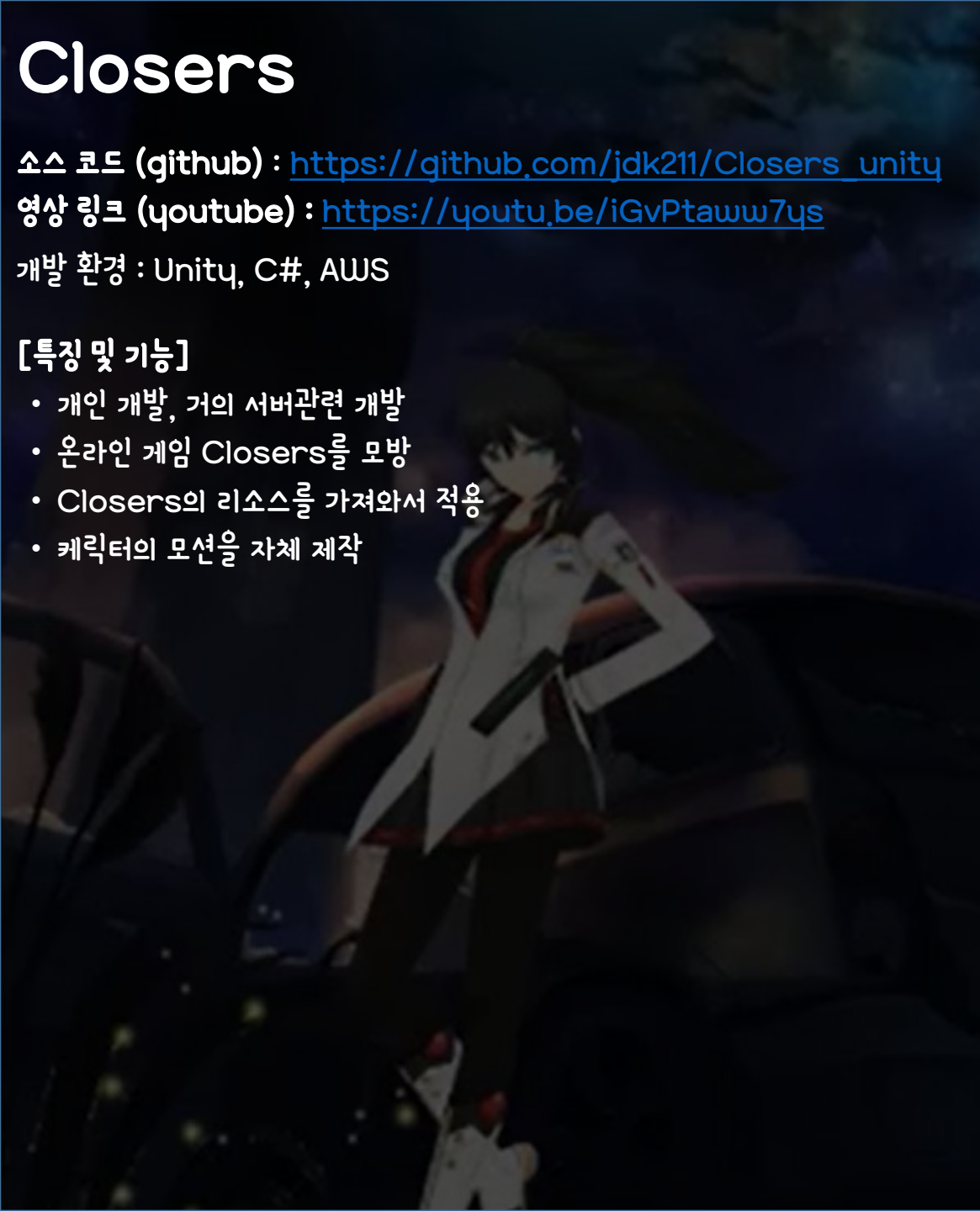
소스 코드 (github) : https://github.com/jdk211/Closers_unity

영상 링크 (youtube) : <https://youtu.be/iGvPtaww7ys>

개발 환경 : Unity, C#, AWS

[특징 및 기능]

- 개인 개발, 거의 서버관련 개발
- 온라인 게임 Closers를 모방
- Closers의 리소스를 가져와서 적용
- 캐릭터의 모션을 자체 제작



Server

01

Test ID로 로그인하고 가져온 DB의 정보를 보여줍니다.

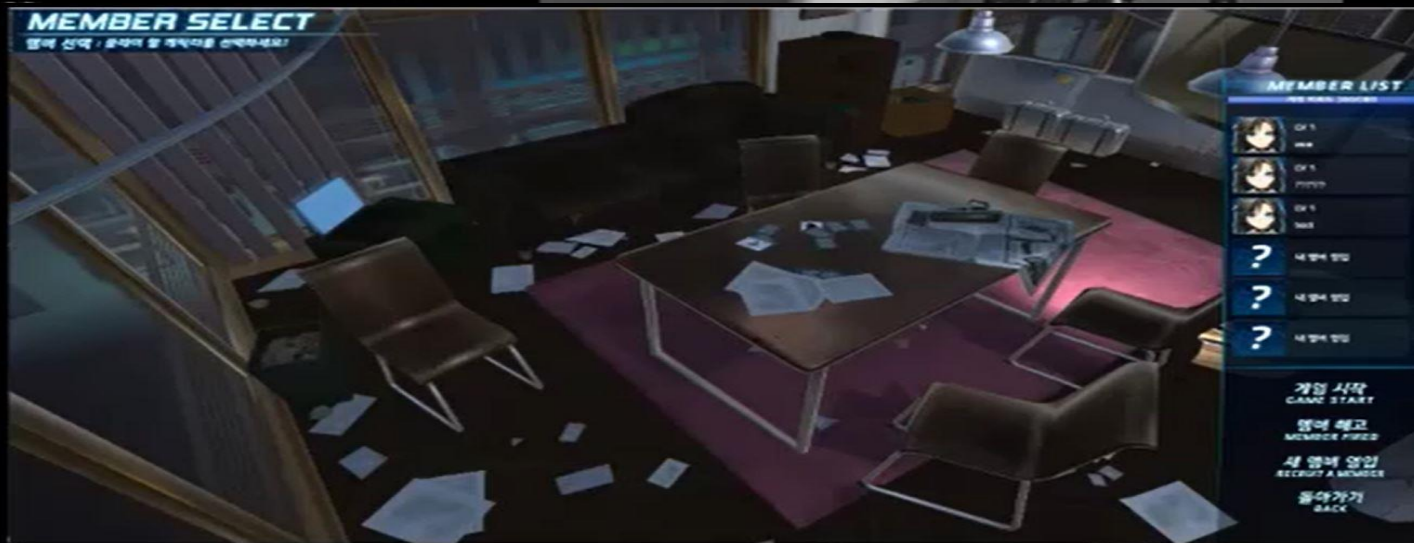


```
ubuntu@172.18.1.30:~$ mysql
mysql> use closer5
mysql> select * from users;
+----+-----+
| id | password |
+----+-----+
| 1234 | 1234 |
| 12345 | 1234 |
| test | 1234 |
| test1 | 1234 |
| test123 | 1234 |
+----+-----+
5 rows in set (0.00 sec)

mysql> select * from users;
+----+-----+
| id | password |
+----+-----+
| 1111 | 1234 |
| 12345 | 1234 |
| test | 1234 |
| test1 | 1234 |
| test123 | 1234 |
+----+-----+
5 rows in set (0.00 sec)

mysql>
```

우선 테스트하던 아이디로 Login 해보겠습니다.



```
ubuntu@172.18.1.30:~$ mysql
mysql> use closer5
mysql> select * from Unit;
+----+----+-----+-----+-----+
| id | num | nickname | level | job |
+----+----+-----+-----+-----+
| 1234 | 1 | 1234 | 1 | 1234 |
| 12345 | 1 | test | 1 | 1234 |
| test | 1 | test1 | 1 | 1234 |
| test1 | 1 | test123 | 1 | 1234 |
+----+----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Unit 테이블에 있는 정보입니다.

그 중 Login한 id를 조건으로 갖고있는 unit정보를 가져와 Member List에 출력합니다.

Server

02

새로운 ID로 캐릭터를 만드는 장면



Server

code

AWS 서버를 사용해서
Uwww로 쿼리를 날려서
DB에 정보를 가져왔습니다.
(쿼리문을 가진 스크립트는 소실...)

자세한 설명은 영상을 참고해주세요.

```
42  /** HTTP GET 방식 통신 처리 */
43  public void get(int id, string url)
44  {
45      WWW www = new WWW(LocalPath + url);
46      StartCoroutine(WaitForRequest(id, www));
47  }
48
49  /** HTTP POST 방식 통신 처리 */
50  public void post(int id, string url, IDictionary<string, string> data)
51  {
52      WWWForm form = new WWWForm();
53
54      foreach (KeyValuePair<string, string> post_arg in data)
55      {
56          form.AddField(post_arg.Key, post_arg.Value);
57      }
58
59      WWW www = new WWW(LocalPath + url, form);
60      StartCoroutine(WaitForRequest(id, www));
61  }
62
63  /** 통신 처리를 위한 코루틴 */
64  private IEnumerator WaitForRequest(int id, WWW www)
65  {
66      // 응답이 올때까지 기다림
67      yield return www;
68
69      // 응답이 왔다면, 이벤트 리스너에 응답 결과 전달
70      bool hasCompleteListener = (OnHttpRequest != null);
71
72      if (hasCompleteListener)
73      {
74          OnHttpRequest(id, www);
75      }
76
77      // 통신 해제
78      www.Dispose();
79  }
80
81
82  }
```

*http 통신

```
56  private void CompareID()
57  {
58      m_dicLoginData.Add("ID", IDInput.text);
59      m_dicLoginData.Add("PW", PWInput.text);
60
61      http_helper = http.Instance;
62      helper.OnHttpRequest += OnHttpRequest;
63      helper.post(100, "/Login", m_dicLoginData);
64
65      m_dicLoginData.Clear();
66  }
67
68  void OnHttpRequest(int id, WWW www)
69  {
70      Debug.Log(www.text);
71      if (www.error == null)
72      {
73          Debug.Log(www.text);
74          SceneManager.LoadScene("CharacterSelectScene");
75      }
76      else
77      {
78          Debug.Log("[Error] " + www.error);
79          LoginFailed.SetActive(true);
80      }
81
82      http_helper = http.Instance;
83      helper.OnHttpRequest -= OnHttpRequest;
84  }
85  }
```

*적용

GunShooting

소스 코드 (github) : https://github.com/jdk211/Top_Down_Gun_Shooting_Base

개발 환경 : Unity, C#,

[특징 및 기능]

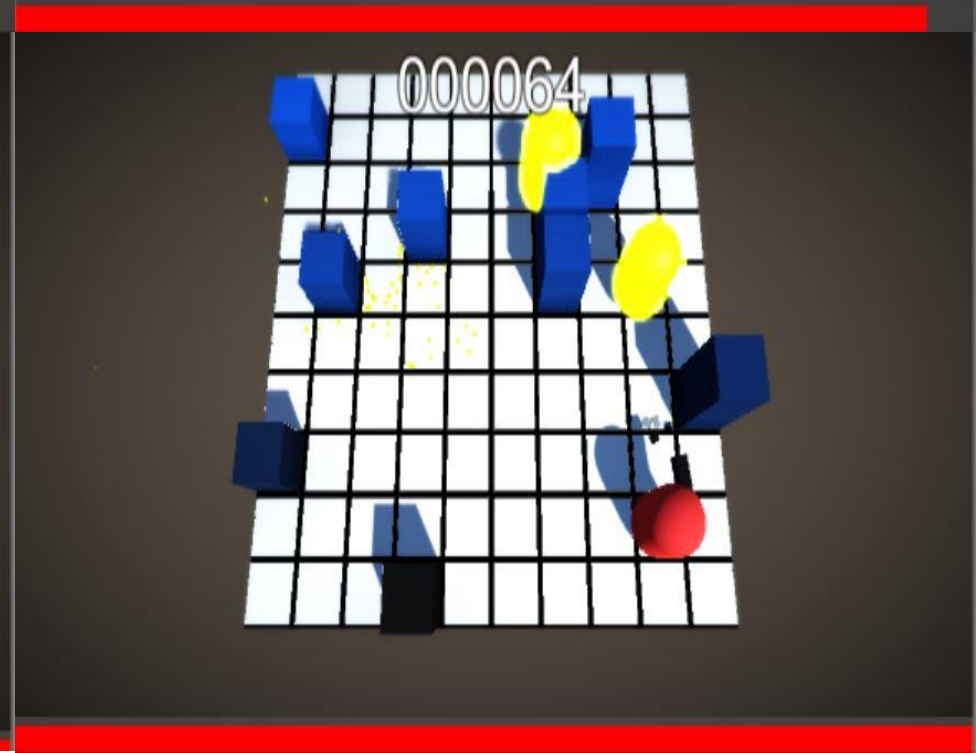
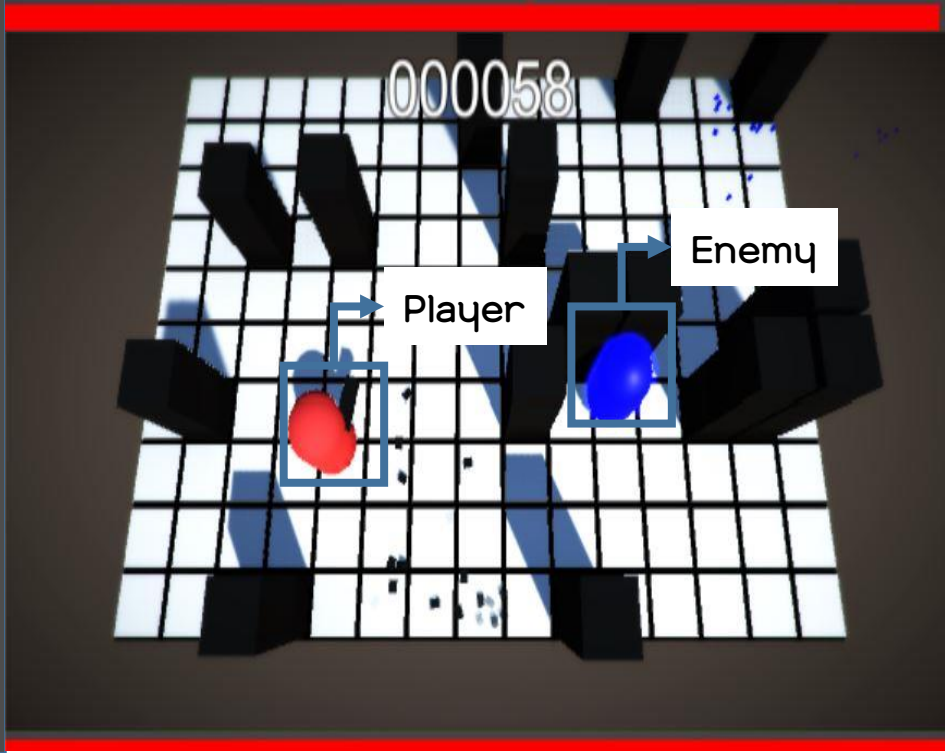
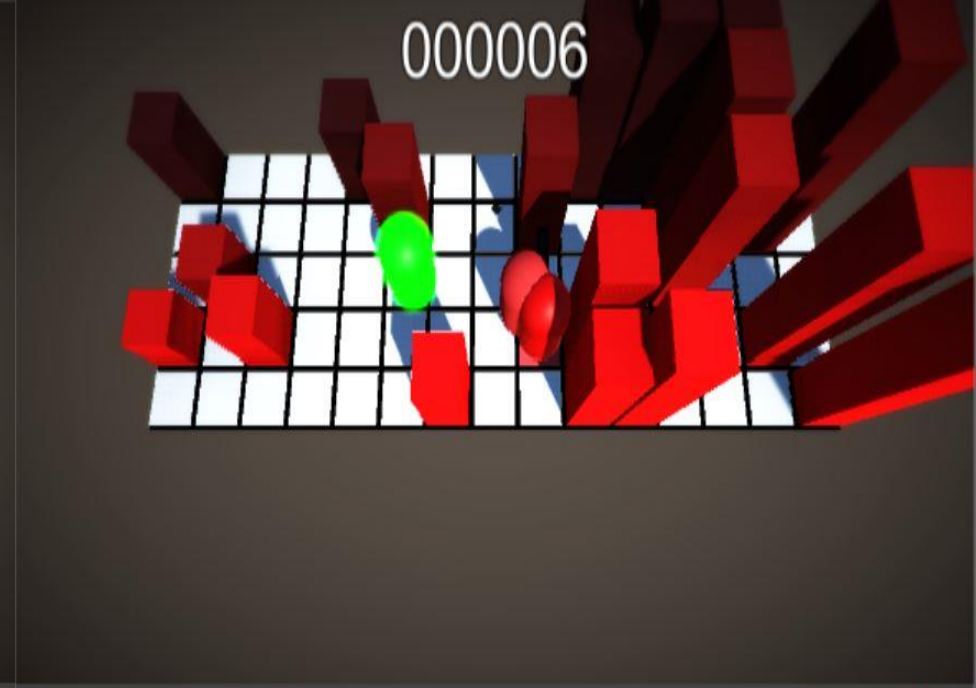
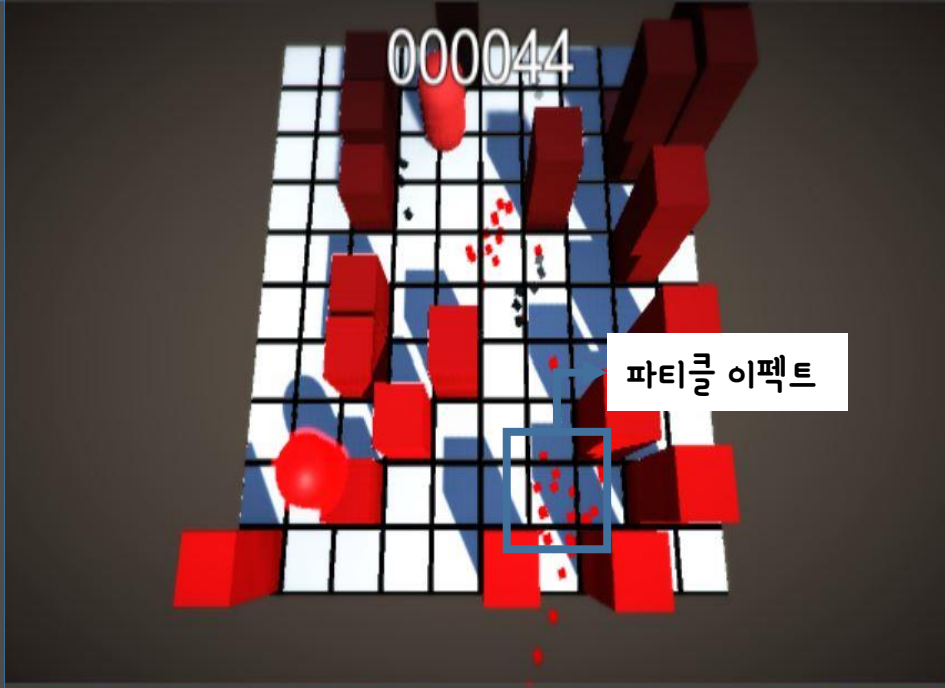
- YouTube 강좌 보면서 만든 게임
- Base를 기반으로 몇몇 기능을 추가
- Object Pool 사용해서 적군 생성
- Stage의 정보를 csv로 저장해서 읽어와서 Stage를 생성

- Wave 1 -
Enemies : 5

Play

간단한 Shooting 게임
Stage마다 적들의 능력치가 다르다
(색, 생성 속도, 이동속도)

적은 죽을때 파티클로
이펙트를 뿌리면서 죽는다.

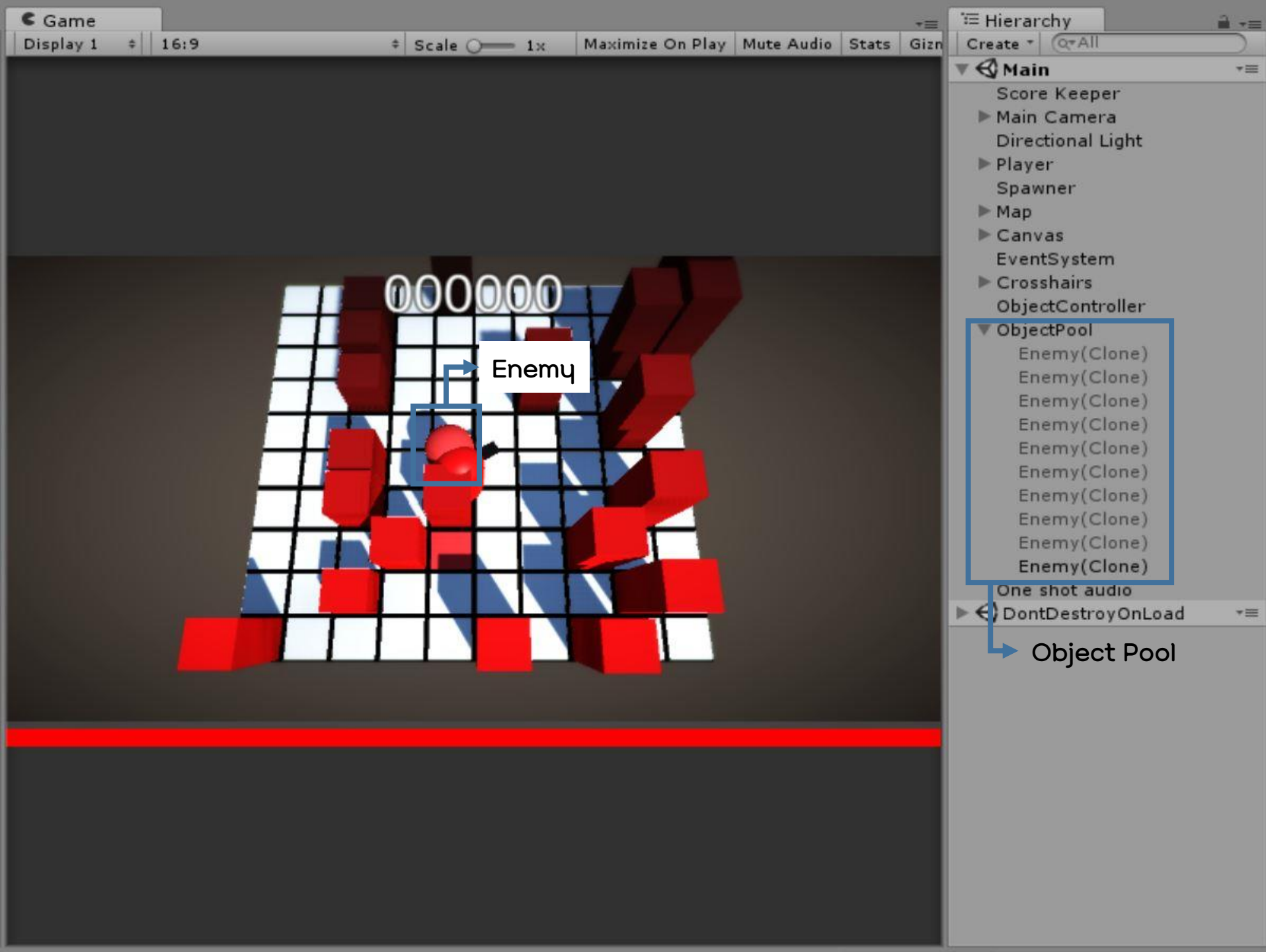


Object Pool

01

Instantiate는 Prefab

생성 비용이 커서 미리 만들어둔 object들
을 썼다 썼다하면서 재사용한다.



Object Pool

code

실시간으로 Instantiate는 Prefab
생성 비용이 커서 미리 만들어둔 object들
을 썼다 썼다하면서 재사용한다.

```
4 public void MakeObject(string strKey, GameObject pPrefab, int nAddItemCount = ObjectPoolItem.DEFAULT_CREATE_ITEM_COUNT)
5 {
6     if (m_dictObjectPool.ContainsKey(strKey)) // 동일한 키가 있으면 예외처리
7     {
8         Debug.Log("Duplicate ObjectPool Key [ " + strKey + " ] ");
9         return;
10    }
11
12    ObjectPoolItem pPoolItem = new ObjectPoolItem(strKey, pPrefab);
13    pPoolItem.AddItem(nAddItemCount);
14
15    m_dictObjectPool.Add(strKey, pPoolItem);
16 }
17
18 public ViewObject GetObject(string strKey)
19 {
20     if(!m_dictObjectPool.ContainsKey(strKey))
21     {
22         Debug.Log("Not Contains ObjectPool Key [ " + strKey + " ] ");
23         return null;
24     }
25
26     ObjectPoolItem pPoolItem = m_dictObjectPool[strKey];
27
28     ViewObject pViewObject = pPoolItem.GetItem();
29
30     pViewObject.DoAwake();
31
32     return pViewObject;
33 }
34
35 public ViewObject CreateItem() // Object Create
36 {
37     GameObject pObj = GameObject.Instantiate(m_pPrefab);
38     ++m_nItemCount;
39     ViewObject pViewObject = pObj.GetComponent<ViewObject>();
40     pViewObject.SetObjectPoolKey(m_strKey);
41
42     pObj.transform.parent = ObjectPool.Instance().transform;
43     pObj.SetActive(false);
44
45     return pViewObject;
46 }
47
48 public void AddItem(int nAddItemCount = ObjectPoolItem.DEFAULT_CREATE_ITEM_COUNT) // nAddItemCount 만큼 미리 만들어 놓기
49 {
50     for (int i = 0; i < nAddItemCount; i++)
51     {
52         m_stackUnusedItem.Push(CreateItem());
53     }
54 }
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
```

*Make Object

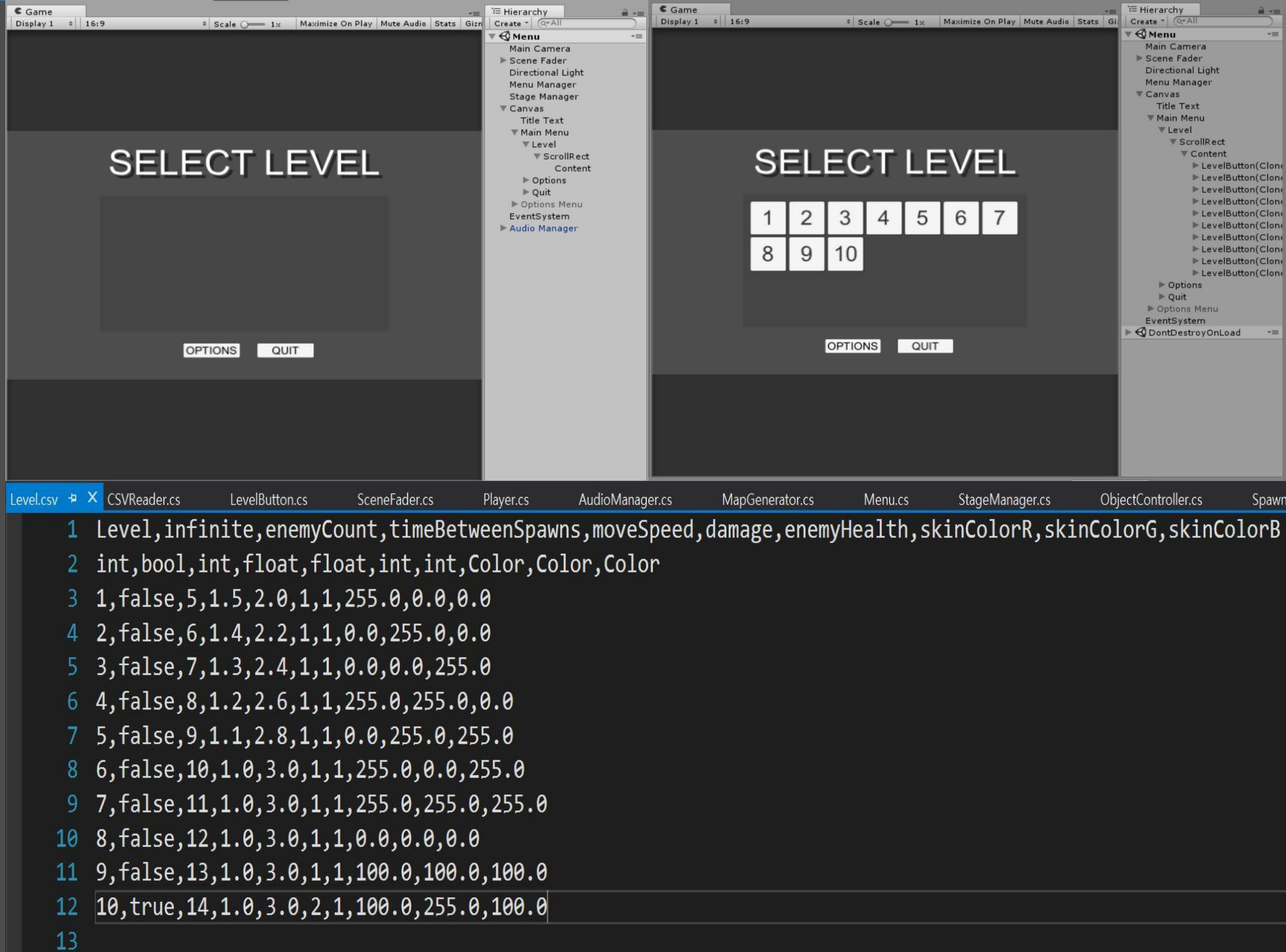
*Return Object

```
80 public void ReturnObject(ViewObject pViewObject)
81 {
82     if(!m_dictObjectPool.ContainsKey(pViewObject.ObjectPoolKey))
83     {
84         Debug.Log("NOT Pool Object key [ " + pViewObject.ObjectPoolKey + " ] ");
85         return;
86     }
87
88     ObjectPoolItem pPoolItem = m_dictObjectPool[pViewObject.ObjectPoolKey];
89
90     pViewObject.DoDestroy();
91
92     pPoolItem.ReturnItem(pViewObject);
93 }
```


Stage

01

Excel로 csv file을 만들고
CSVReader로 게임에 적용



Stage

code

CSVReader 코드와 적용 코드

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

public class CSVReader
{
    static string SPLIT_RE = @"(?=([^\"]*" + "\"\"*" + ")*(?!\"\"))";
    static string LINE_SPLIT_RE = @"\r\n|\n\r|\n|\r";
    static char[] TRIM_CHARS = { '\n' };

    public static List<Dictionary<string, object>> Read(string fileName)
    {
        var list = new List<Dictionary<string, object>>();
        TextAsset data = Resources.Load(fileName) as TextAsset;

        var lines = Regex.Split(data.text, LINE_SPLIT_RE);

        if (lines.Length <= 1) return list;

        var header = Regex.Split(lines[0], SPLIT_RE);

        for(var i = 0; i < lines.Length; i++)
        {
            var values = Regex.Split(lines[i], SPLIT_RE);
            if (values.Length == 0 || values[0] == "" ) continue;

            var entry = new Dictionary<string, object>();
            for(var j = 0; j < header.Length && j < values.Length; j++)
            {
                string value = values[j];
                value = value.TrimStart(TRIM_CHARS).TrimEnd(TRIM_CHARS).Replace("\\", "");
                object finalValue = value;

                int n;
                float f;
                bool b;

                if (int.TryParse(value, out n))
                {
                    finalValue = n;
                }
                else if(float.TryParse(value, out f))
                {
                    finalValue = f;
                }
                else if(bool.TryParse(value, out b))
                {
                    finalValue = b;
                }

                entry[header[j]] = finalValue;
            }
            list.Add(entry);
        }

        return list;
    }
}
```

*CSVReader

*CSVReader 적용

```
private void CreateLevel()
{
    waves = new List<Wave>();

    List<Dictionary<string, object>> data = CSVReader.Read("Level");

    for (int i = 2; i < data.Count; i++)
    {
        Wave emptyWave = new Wave();
        emptyWave.level = (int)data[i]["Level"];
        emptyWave.infinite = (bool)data[i]["infinite"];
        emptyWave.enemyCount = (int)data[i]["enemyCount"];
        emptyWave.timeBetweenSpawns = (float)data[i]["timeBetweenSpawns"];
        emptyWave.moveSpeed = (float)data[i]["moveSpeed"];
        emptyWave.damage = (int)data[i]["damage"];
        emptyWave.enemyHealth = (int)data[i]["enemyHealth"];
        emptyWave.skinColor.r = (float)data[i]["skinColorR"];
        emptyWave.skinColor.g = (float)data[i]["skinColorG"];
        emptyWave.skinColor.b = (float)data[i]["skinColorB"];
        waves.Add(emptyWave);
    }
}
```


Thank you!

감사합니다!