



[스파르타코딩클럽] 게임개발 종합반 - 2주차



매 주차 강의자료 시작에 PDF파일을 올려두었어요!

▼ PDF 파일

[수업 목표]

1. 유니티 기본 사용법 복습해보기
2. 유명 게임을 완성해보기
3. 베스트 스코어 기록해보기

[목차]

- 01. 2주차 오늘 배울 것
- 02. 기본 씬 구성하기
- 03. 풍선 만들기
- 04. 마우스 만들기
- 05. 네모 만들기(1) - 내려오기
- 06. 네모 만들기(2) - 나타나기
- 07. 시간 올라가게 하기
- 08. 게임 끝내기(1) - 판넬 만들기
- 09. 게임 끝내기(2) - 판넬 나타내기
- 10. 최고 점수 나타내기
- 11. 풍선 애니메이션 전환하기
- 12. 숙제 - 떨어지는 네모를 없애기!
- HW. 2주차 숙제 해설



모든 토글을 열고 닫는 단축키

Windows : `Ctrl` + `alt` + `t`

Mac : `⌘` + `⌥` + `t`

01. 2주차 오늘 배울 것

▼ 1) 2주차 수업의 목표와 범위



오늘의 키워드! 기억이 새록새록

- 1주차에 배웠던 것을 복습 → 복습 → 복습 하는 게 전체적인 수업의 구성이에요!
→ 이번 주는 조금 더 익숙해진 자신을 볼 수 있을 것이랍니다.
→ 참고로 **오늘, 쉽습니다!** (복습이 많기 때문에! 😎)
 - Rise Up! 이란 게임을 유사하게 만들어보면서, 유니티의 기초를 다시 다집니다.
- ▼ 실제 게임 모습 보기: 전세계 1억 다운로드 이상의 히트작이랍니다.



추가 정보

업데이트 날짜

2021년 8월 31일

크기

기기에 따라 다릅니다.

설치 수

100,000,000+

현재 버전

3.1.2

필요한 Android 버전

5.0 이상

콘텐츠 등급

전체이용가
자세히 알아보기

인앱 상품

항목당 US\$1.99 -
US\$3.99

권한

세부정보 보기

신고

부적절한 앱으로 신고

제공

Serkan Özyilmaz

개발자

웹사이트 방문

magnificentgames2018@gmail.com

개인정보처리방침

Istanbul

▼ 2) 오늘 만들 것 : 풍선을 지켜라

- 네모가 풍선에 닿으면 끝! 오래 살아남는 게 목표랍니다.
→ 새롭게 배우는 것: 마우스로 제어하기, 베스트스코어 기록, 애니메이션 전환

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/f4ad2974-6c4c-4664-8330-39e12915d4fb/Untitled.mp4>

▼ 3) 만들 순서

1. 기본 씬 구성하기 - 배경, 풍선, 마우스, 네모, 시간
2. 풍선 애니메이션 더하기
3. 마우스 움직임 더하기
4. 시간 가게 하기
5. 네모 내려오게 하기 + 충돌 구현
6. 게임 끝내기(1): 판넬 만들기
7. 게임 끝내기(2): 베스트 스코어 기록하기
8. 풍선 애니메이션 전환하기

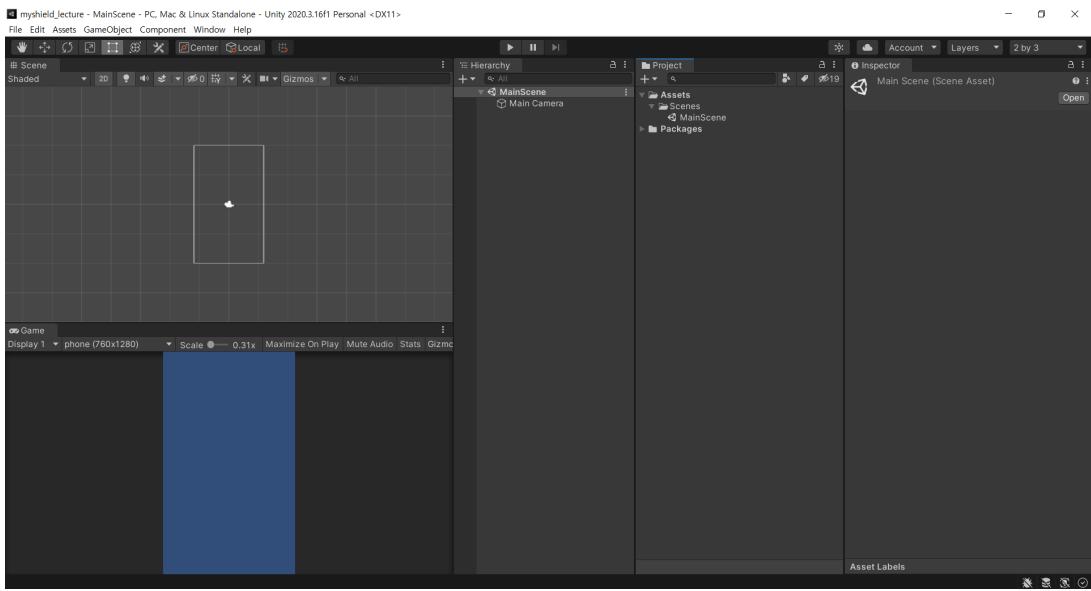
02. 기본 씬 구성하기



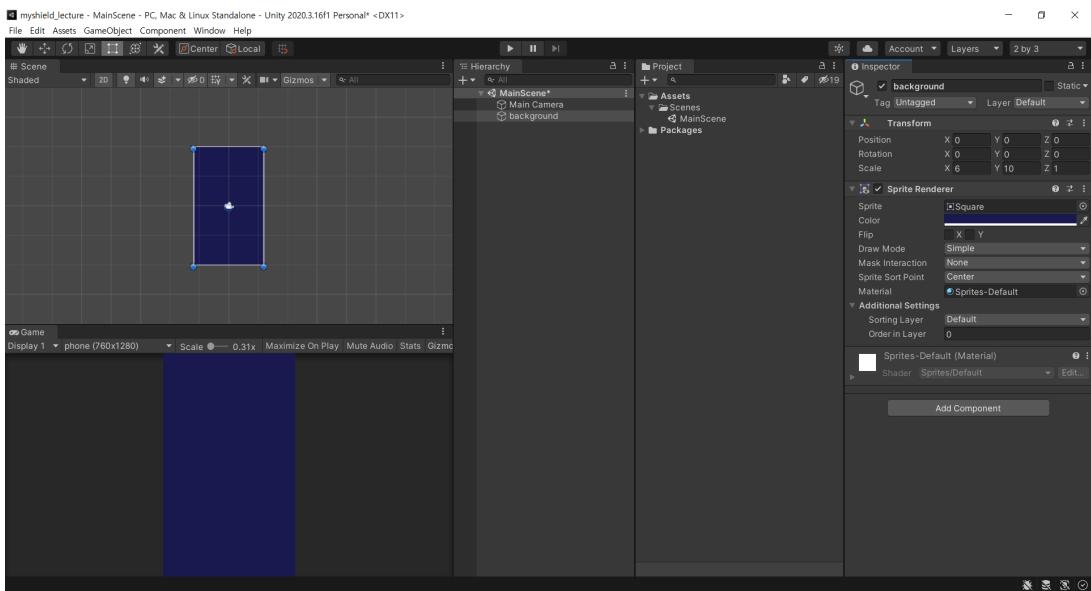
myshield라는 프로젝트를 만들고 시작할게요!

▼ 1) 기본 세팅하기 & 배경 만들기

1. windows → 2x3 layout 클릭! free aspect → phone 클릭!



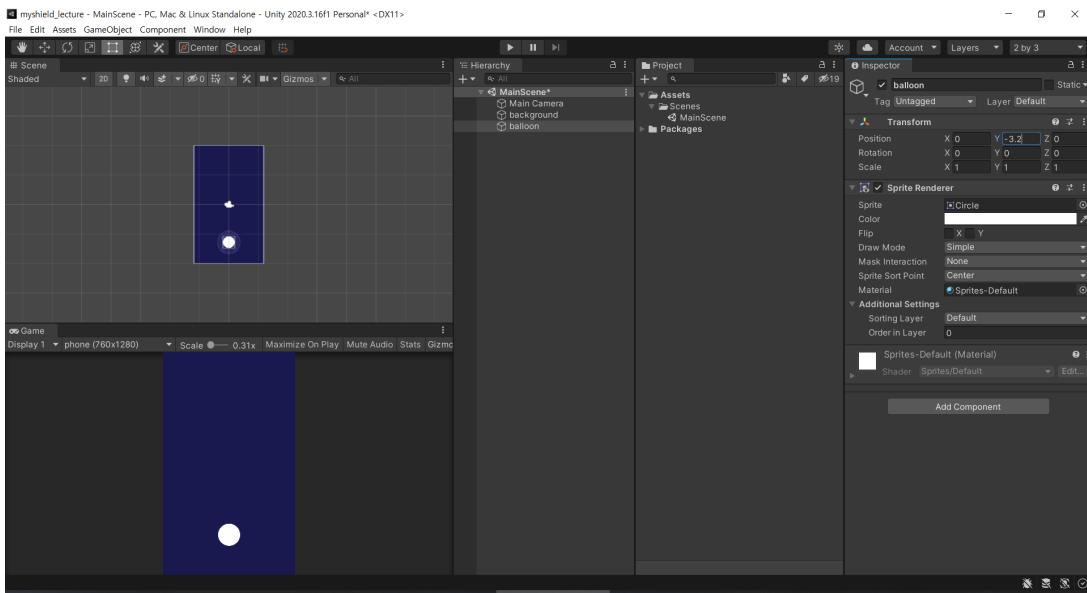
2. 배경 색은 $\text{rgb} \Rightarrow 20, 20, 80$ 으로 할게요! 사이즈는 $x: 6, y: 10$



▼ 2) 풍선, 마우스 만들기

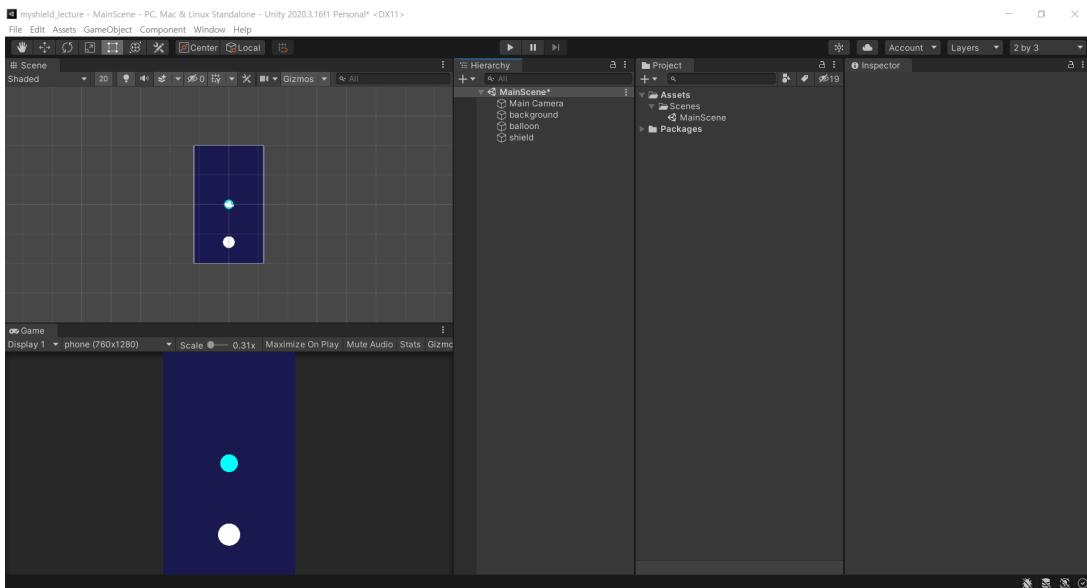
1. 간단한 풍선을 만들어둡니다. (balloon)

→ position $x: 0, y: -3.2$ 에 맞춰두기!



2. 마우스를 만들어둡니다. (shield)

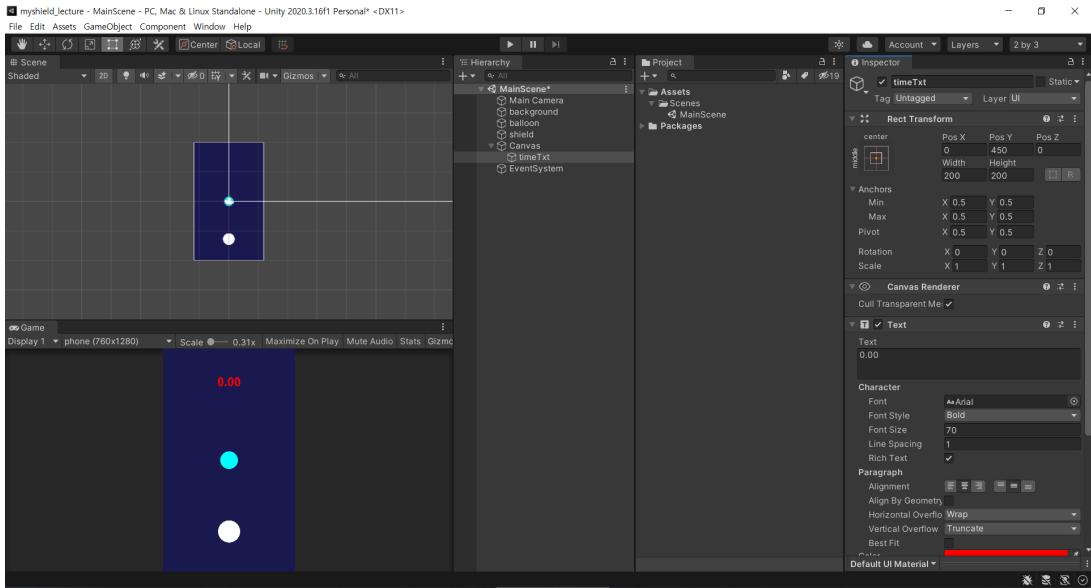
- scale `x: 0.5, y: 0.5` 로 세팅해두기! position은 그대로 둘게요!
- `rgb = 0, 0, 255` 로 가겠습니다!



▼ 3) 타이머 만들기

- 시간 라벨 만들기 (timeTxt)
 - UI → Text를 활용!
 - `font size 70`, `position x: 0, y: 450` 으로 맞춰주세요!
 - width: 200, height :200

→ posY: 450으로 맞춰주세요!



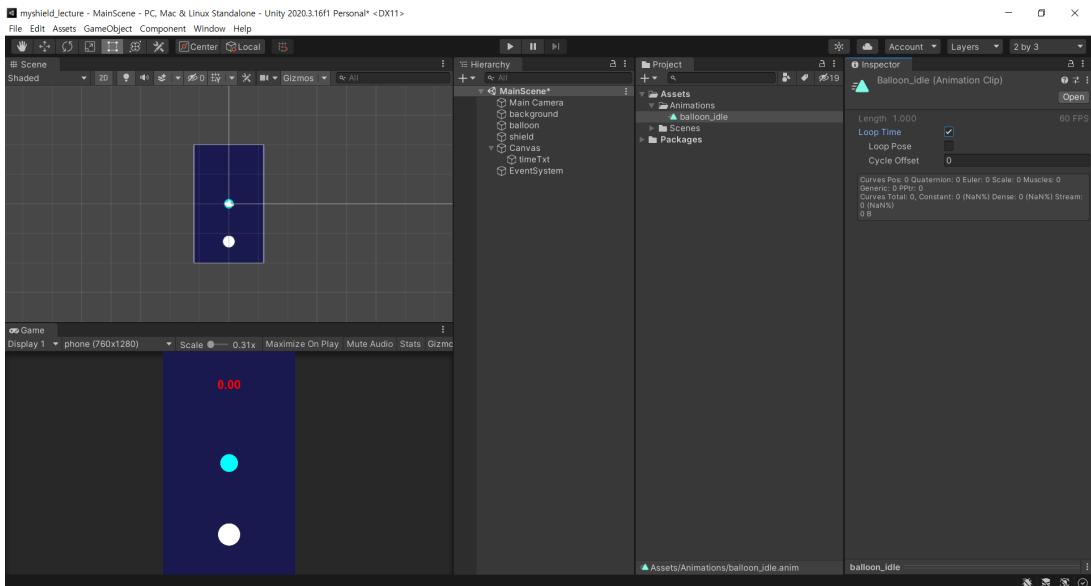
03. 풍선 만들기

▼ 1) 애니메이션 더하기

1. Animations 폴더 → 애니메이션 만들기 (balloon_idle)

→ 이따가 풍선이 "터지는" 애니메이션도 만들어야 하니, 이것은 idle(기본 상태)로 둘게요!

→ Loop Time에 체크하는 것 잊지 말기!

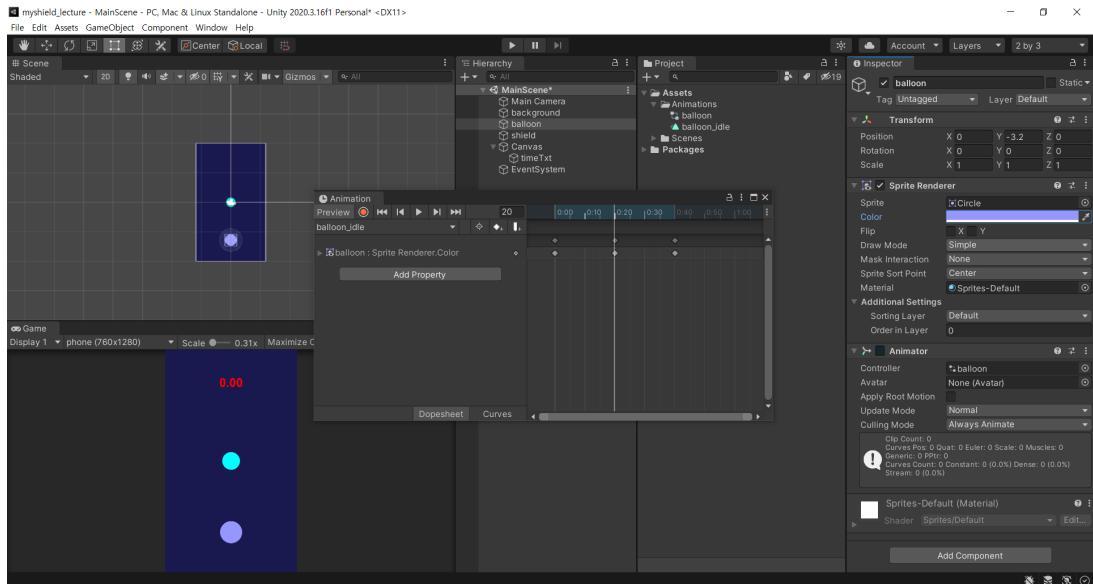


2. 풍선에 끌어다 놓고 애니메이션 만들기

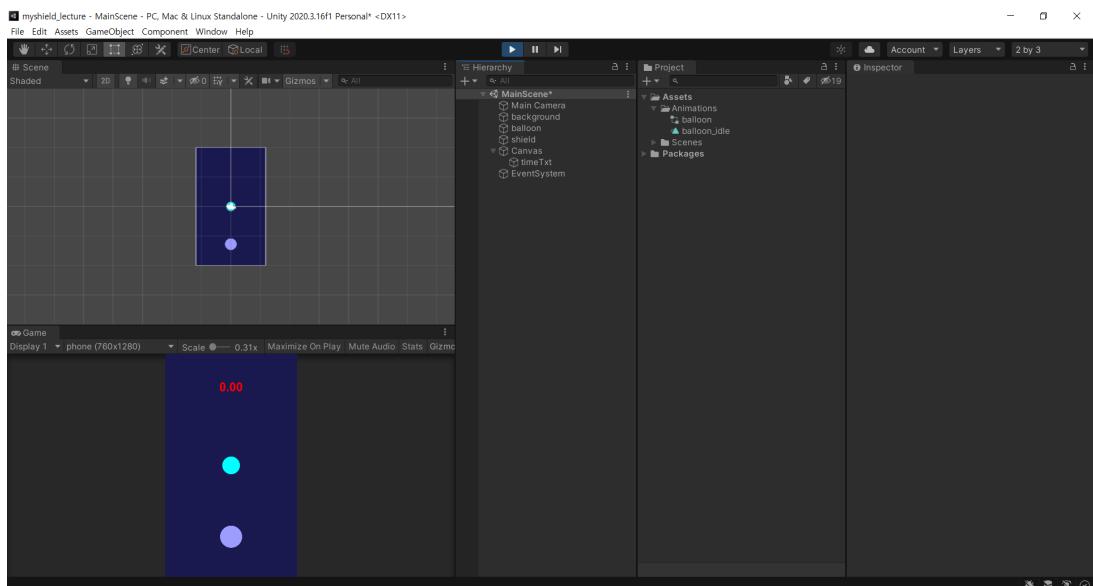
→ 레코드 (빨간색 동그라미!)를 눌러서 같이 세팅합시다!

→ 0:00, 0:40은 처음 모습 그대로

→ 0:20은 `rgb → 200, 200, 255` 로 세팅하기



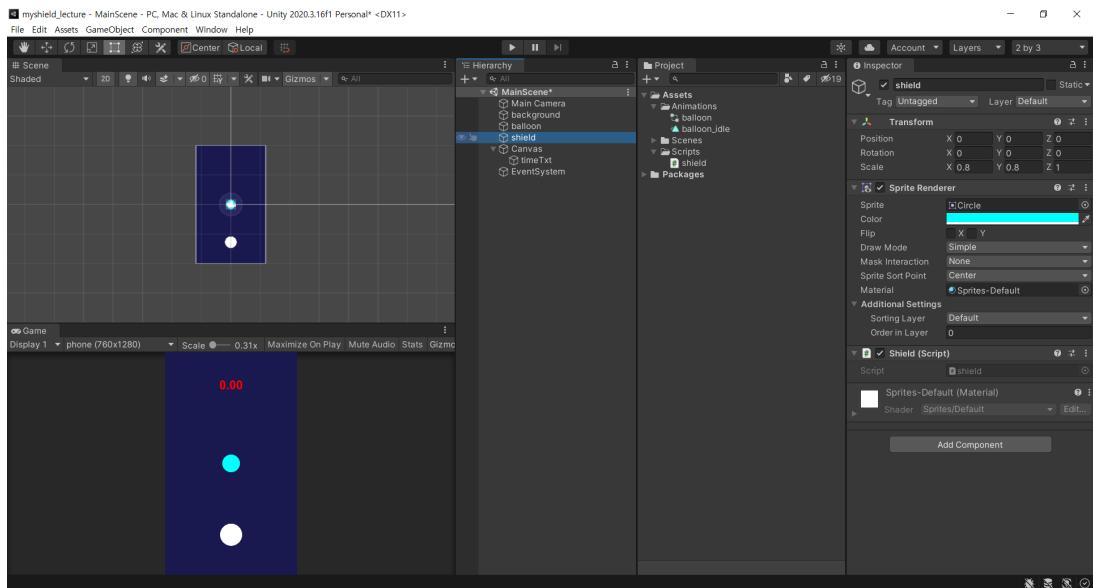
3. `Play` 버튼을 눌러서 확인하기



04. 마우스 만들기

▼ 1) 마우스에 움직임 더하기

1. Scripts → shield.cs 만들기 + shield에 붙이기



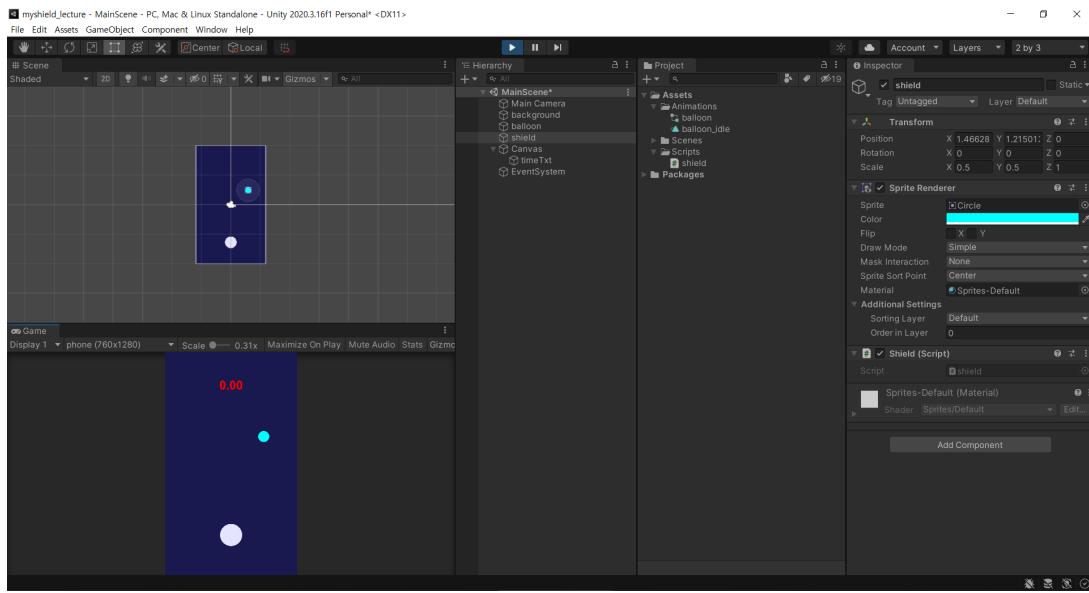
2. 마우스 포인터를 따라 움직이게 하기

→ 외우지 말고, 나중에도 보고 쓰는 코드랍니다. 튜터도 외우고 있지 않아요!

→ mouse 의 좌표계를 카메라 좌표계로 바꾸고, shield의 위치에 넣어주기

```
void Update()
{
    Vector3 mousePos = Camera.main.ScreenToWorldPoint(Input.mousePosition);
    transform.position = new Vector3(mousePos.x, mousePos.y, 0);
}
```

3. **Play** 해서 움직임 확인하기

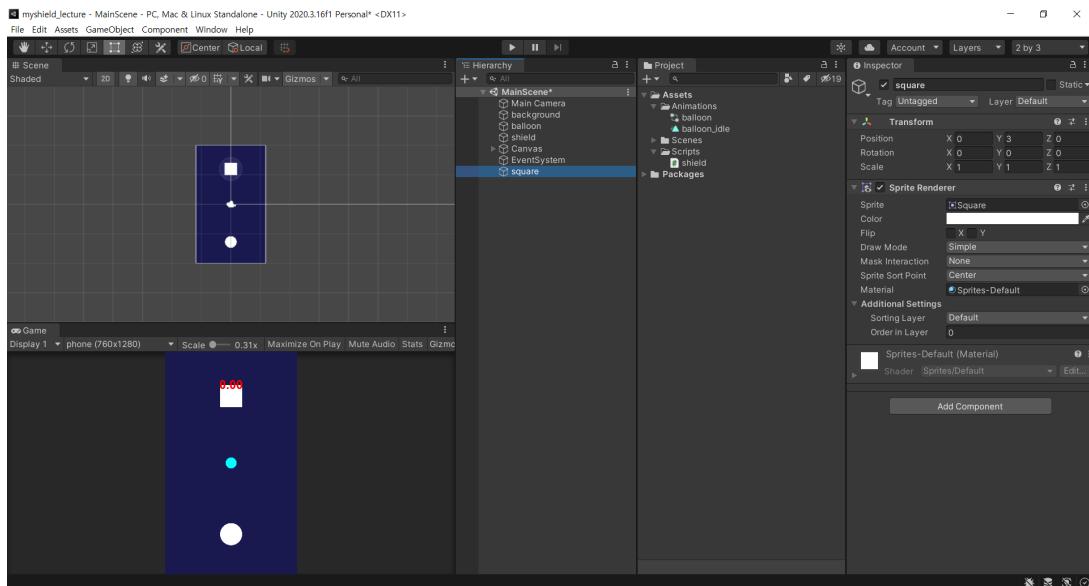


05. 네모 만들기(1) - 내려오기

▼ 1) 네모 만들기

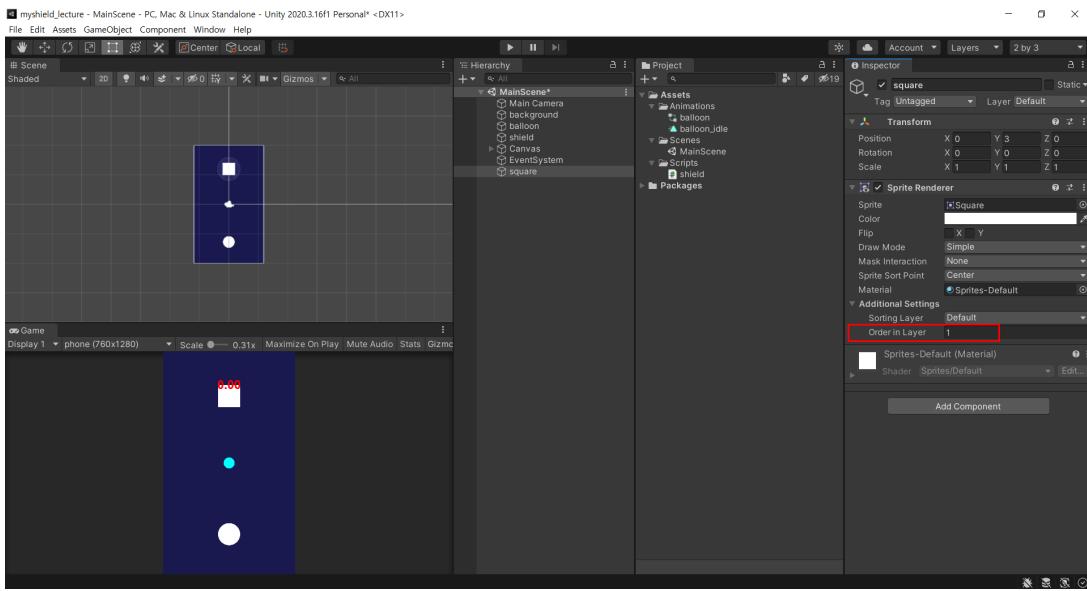
1. sprite → square로 만들고, 이름 바꿔두기 (square)

→ position `x:0, y:3`에 맞추기



2. order in layer 맞추기

→ 네모, 마우스, 풍선 모두 order in layer를 1로 바꿔줍니다! (참고로 배경은 0 !)

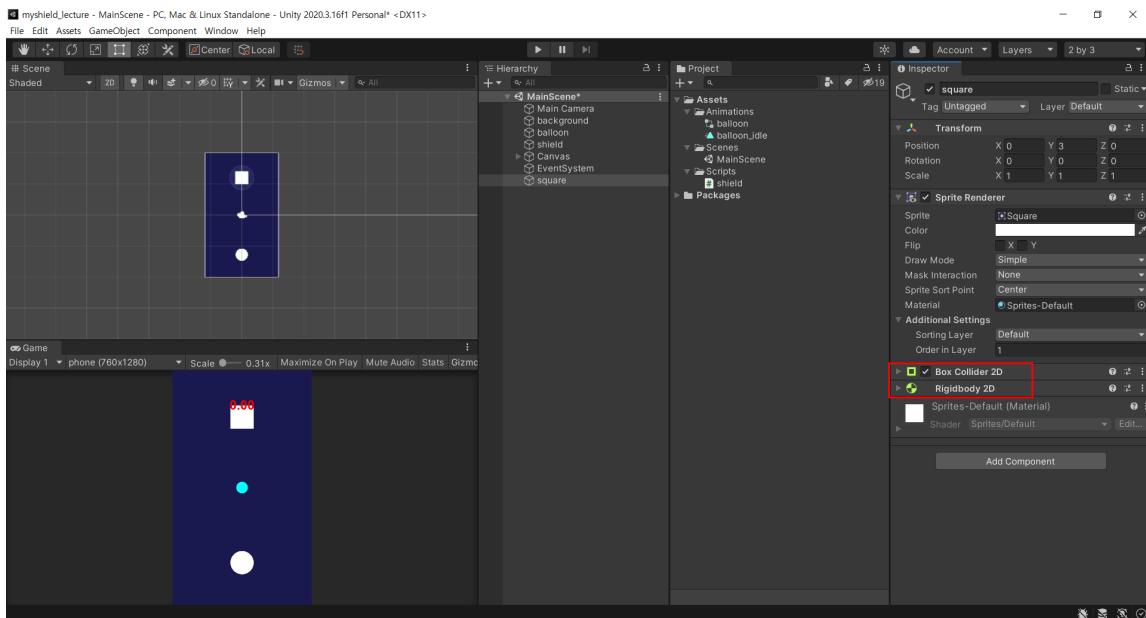


▼ 2) 네모 떨어지게 하기/충돌효과 주기



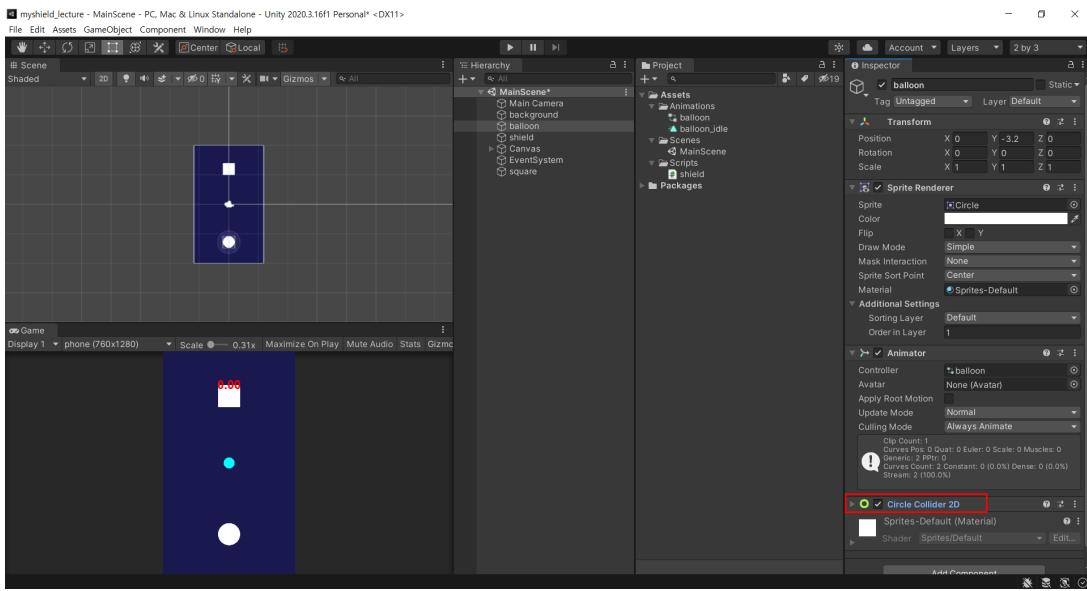
리마인드 - 충돌은 **한쪽에 rigidbody** 를 주고 + **양쪽에 collider** 가 있으면 완성!

→ **rigidbody 2D** 와, **box collider 2D** 를 줍니다.

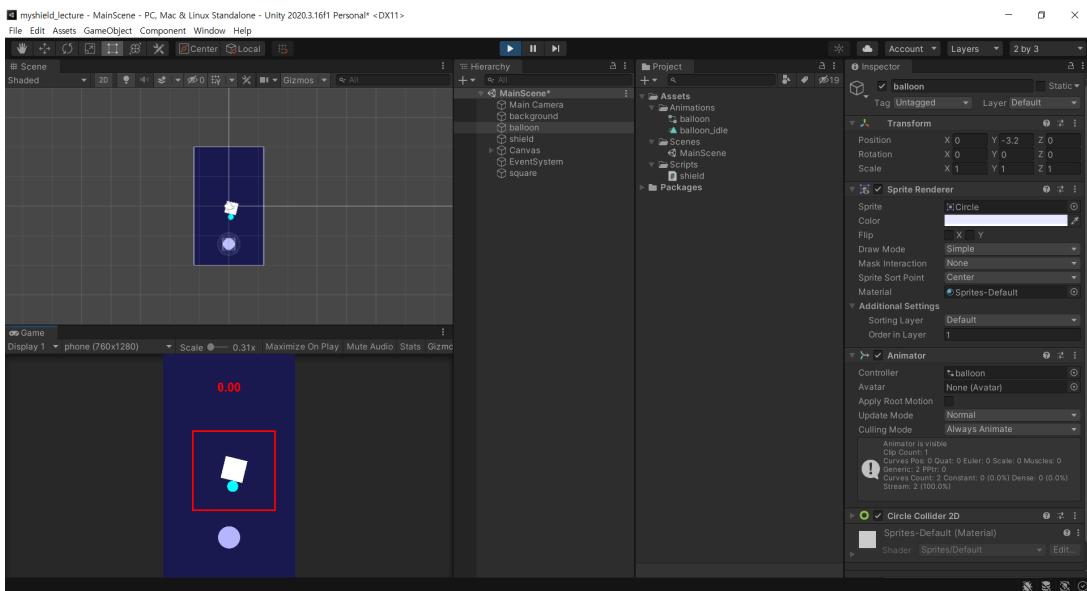


▼ 3) 풍선, 마우스에도 충돌효과 주기

1. 풍선, 마우스에 **circle collider 2D** 달기



2. Play 버튼을 눌러 마우스와 네모가 부딪히는지 보기



06. 네모 만들기(2) - 나타나기

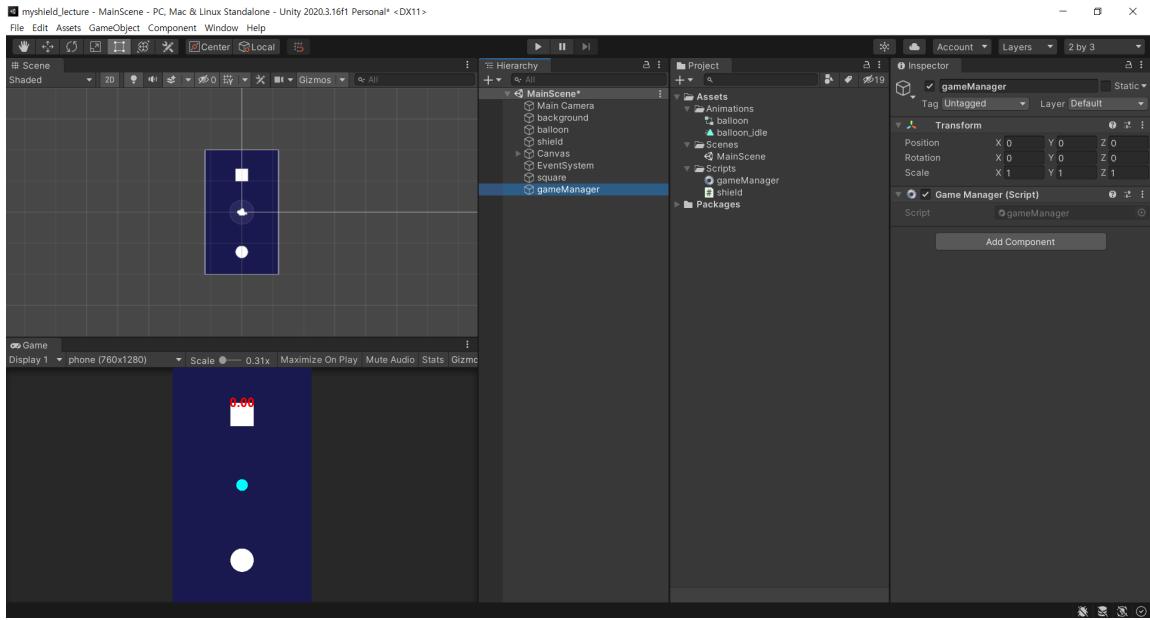
▼ 1) gameManager 만들기



리마인드 - gameManager는? 가운데서 게임을 관장하는 역할!

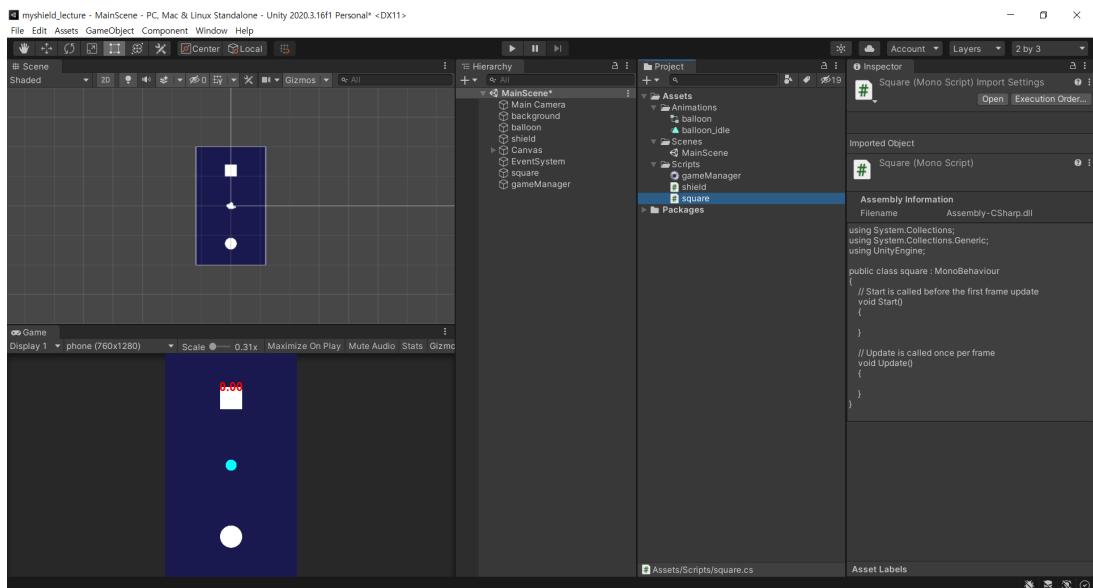
→ gameManager gameObject 와 script 를 만들고 서로 붙여줍니다!

→ 이후 이 gameManager에서 네모를 만들어 줄 예정!



▼ 2) 네모 랜덤으로 나타내기

1. `square.cs` 만들고, 네모에 붙이기



2. 랜덤 위치에서 생성하기

→ `x: -3.0f ~ 3.0f, y: 3.0f ~ 5.0f`

→ 저장 후 `Play` 해서 확인하기

```
void Start()
{
```

```

        float x = Random.Range(-3.0f, 3.0f);
        float y = Random.Range(3.0f, 5.0f);

        transform.position = new Vector3(x, y, 0);
    }

```

3. 랜덤 사이즈로 생성하기

- `size: 0.5f ~ 1.5f`
- 저장 후 `Play` 해서 확인하기

```

void Start()
{
    float x = Random.Range(-3.0f, 3.0f);
    float y = Random.Range(3.0f, 5.0f);

    transform.position = new Vector3(x, y, 0);

    float size = Random.Range(0.5f, 1.5f);
    transform.localScale = new Vector3(size, size, 1);
}

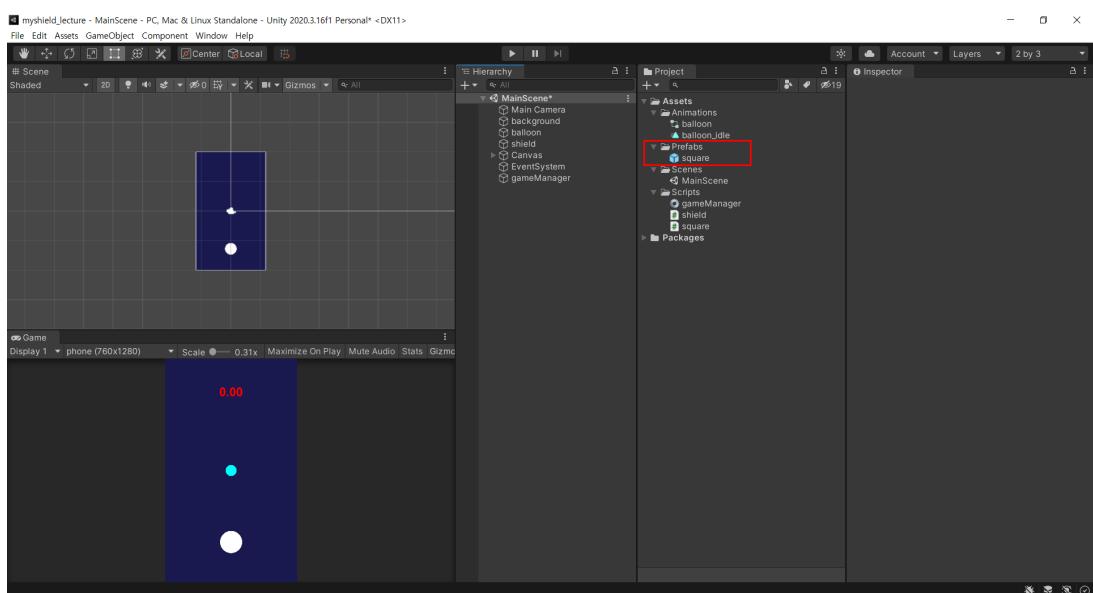
```

▼ 3) 네모를 prefab으로 만들기



리마인드 - 프리팹은? 복제를 위해 미리 만들어둔 아이템

1. Prefabs 폴더를 만들고 → 끌어다넣기
2. 기존의 square 오브젝트는 과감하게 삭제!



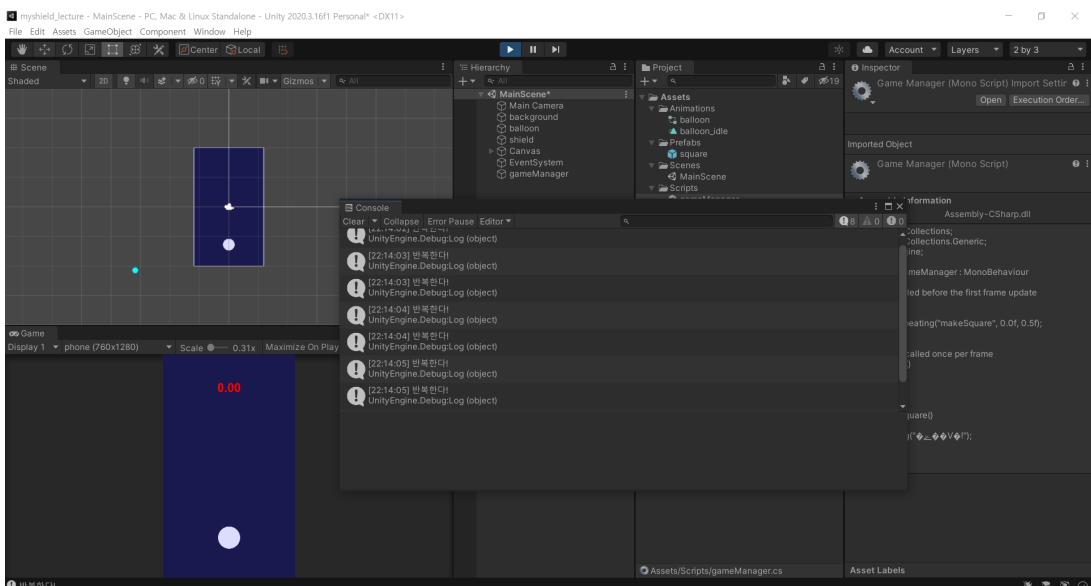
▼ 4) `gameManager.cs` 에서 네모를 만들기

1. 반복 실행하게 하기

→ 0.5f마다 `makeSquare` 함수를 실행!

```
void Start()
{
    InvokeRepeating("makeSquare", 0.0f, 0.5f);
}

void makeSquare()
{
    Debug.Log("반복한다!");
}
```

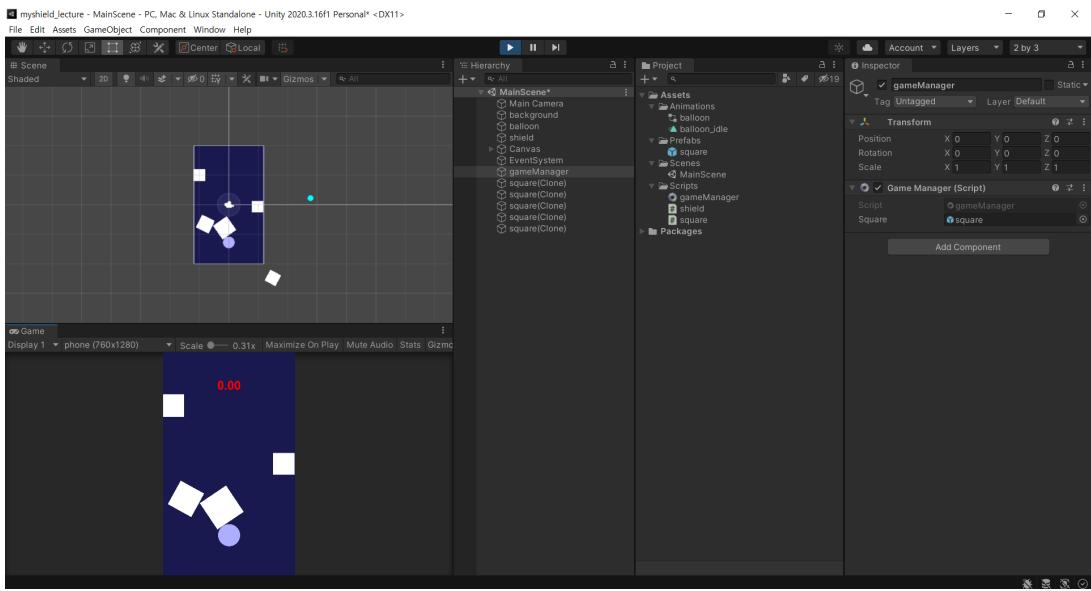


2. 네모 만들기

→ `square` 프리팹을 받아서, 복제하기

```
public GameObject square;

void makeSquare()
{
    Instantiate(square);
}
```



07. 시간 올라가게 하기

▼ 1) 시간 올리기

1. UI text 받기

```
using UnityEngine.UI;

public Text timeTxt;
```

2. 시간 올리기

```
float alive = 0f;

void Update()
{
    alive += Time.deltaTime;
    timeTxt.text = alive.ToString("N2");
}
```

08. 게임 끝내기(1) - 판넬 만들기

▼ 1) 게임 종료 판넬 만들기



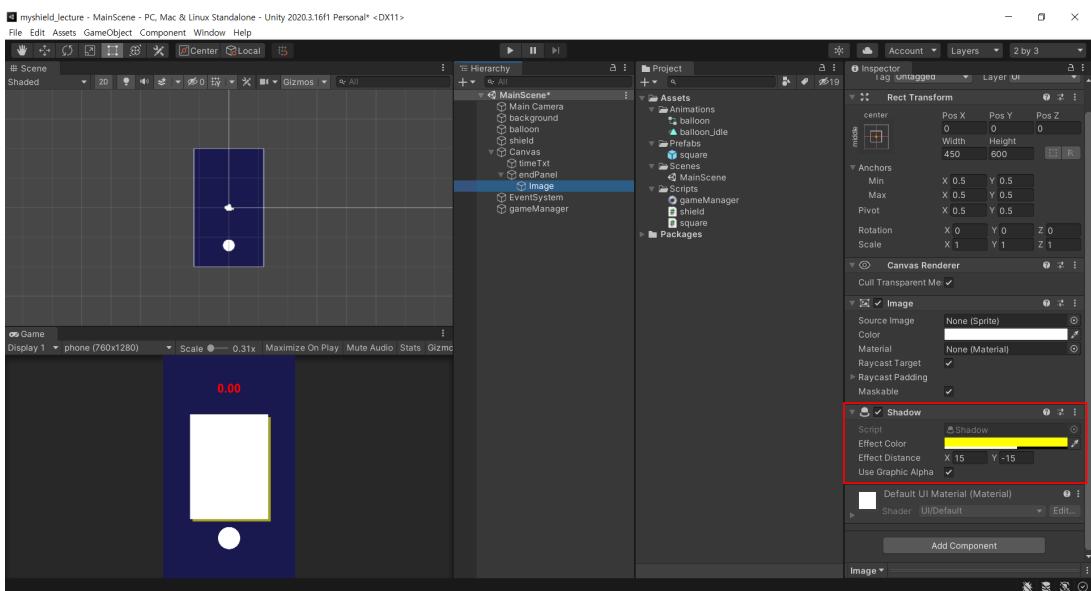
Canvas → endPanel 게임오브젝트로 모두 끌습니다.

1. Image 만들기

→ 사이즈 `x: 450, y: 600`

→ shadow 효과주기 : `rgba ⇒ 255, 255, 0, 150` (Add Component로 추가!)

→ 그림자 위치는 `x: 15, y: -15` 로 맞추기

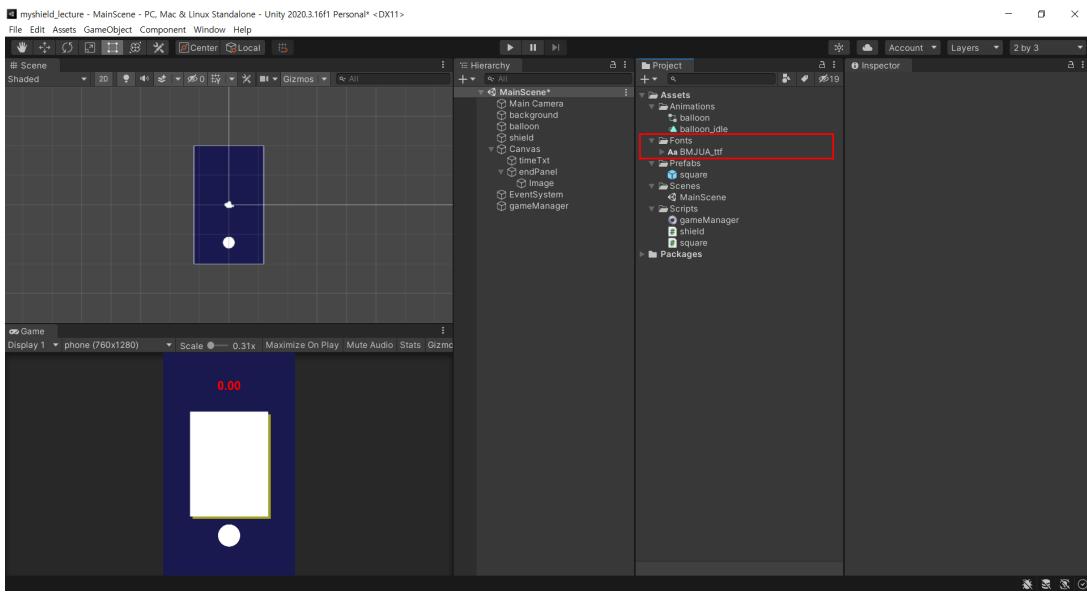


2. 폰트 가져오기

▼ [코드스니펫] 배달의민족 주아체

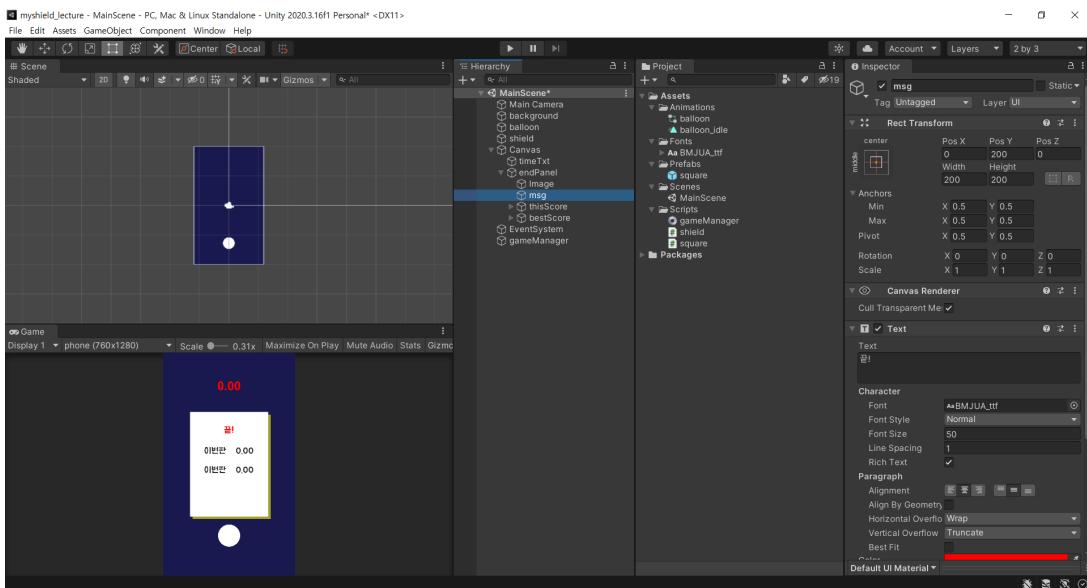
http://pop.baemin.com/fonts/jua/BMJUA_ttf.ttf

→ Fonts 폴더 만들고 끌어다넣기



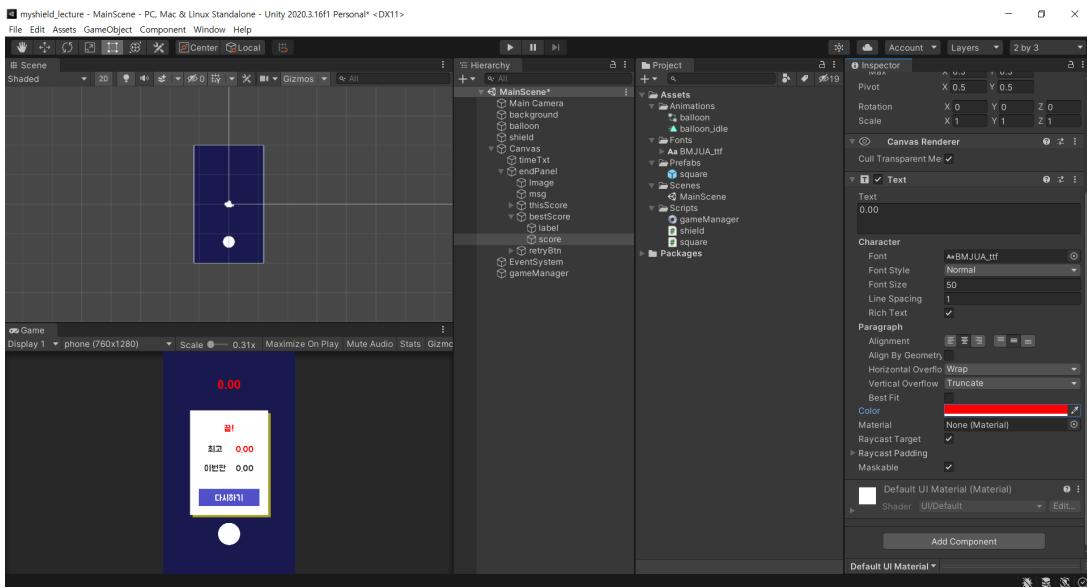
3. 끝 메시지, 현재 스코어, 최고 스코어 만들기

- 폰트 사이즈는 메시지는 50, 라벨은 40으로 해주세요!
- position (-100, 100), (150, 100), (-100, 0), (150, 0)으로 맞춰주세요!
- **ctrl+d** (복제) 를 이용하면 무척 편하답니다.



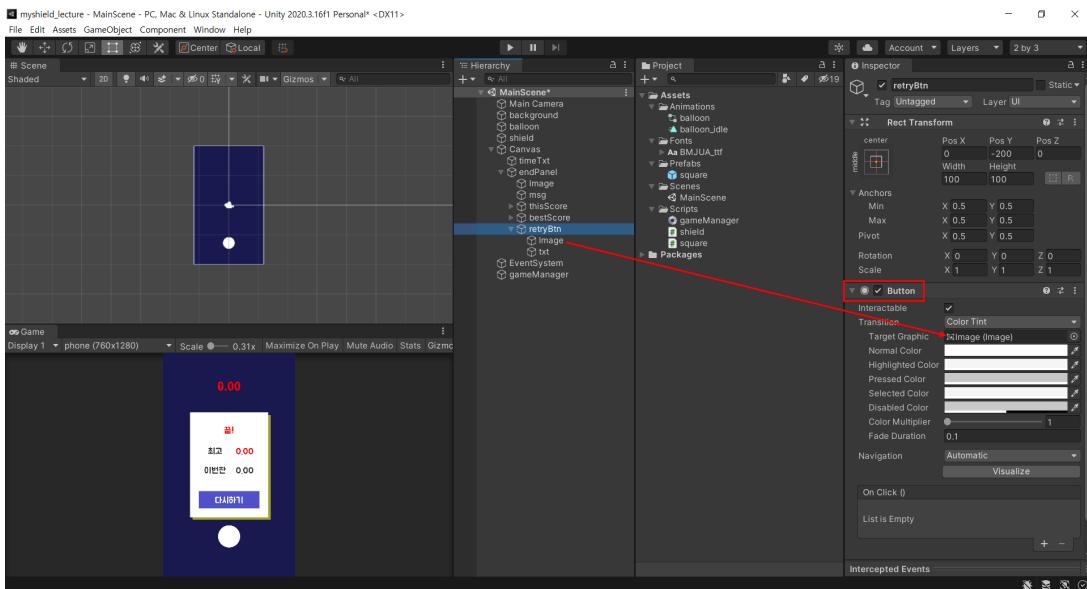
4. retry 버튼 만들기

- retryBtn 오브젝트 안에 만들게요!
- 이미지 색은 `rgb = 80, 80, 200` 으로 할게요!
- width: 300, height: 100
- posY : -200



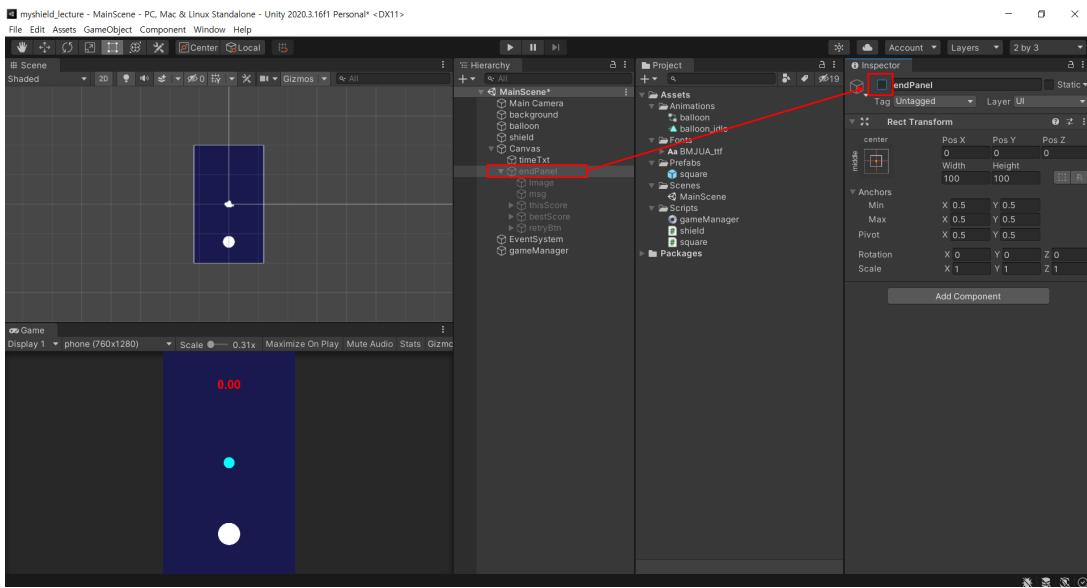
5. 버튼에 `button` 속성 달고 Image 끌어다놓기

→ 그래야 클릭 할 때 color 틴트가 일어납니다!



6. 우선, 판넬 전체를 숨겨둡니다.

→ `SetActive(true)` 로 나중에 켜 것이랍니다!



09. 게임 끝내기(2) - 패널 나타내기

▼ 1) gameManager 싱글톤 처리하기



리마인드 - 싱글톤이란? **너는 딱 하나야!** 라고 할 수 있게 세팅해두는 것
(그래야 다른데서 나를 부를 수 있습니다!)

```
public static gameManager I;

void Awake()
{
    I = this;
}
```

▼ 2) 게임 종료하기

1. `gameManager.cs` 에 종료 함수 만들어두기

```
public GameObject endPanel;

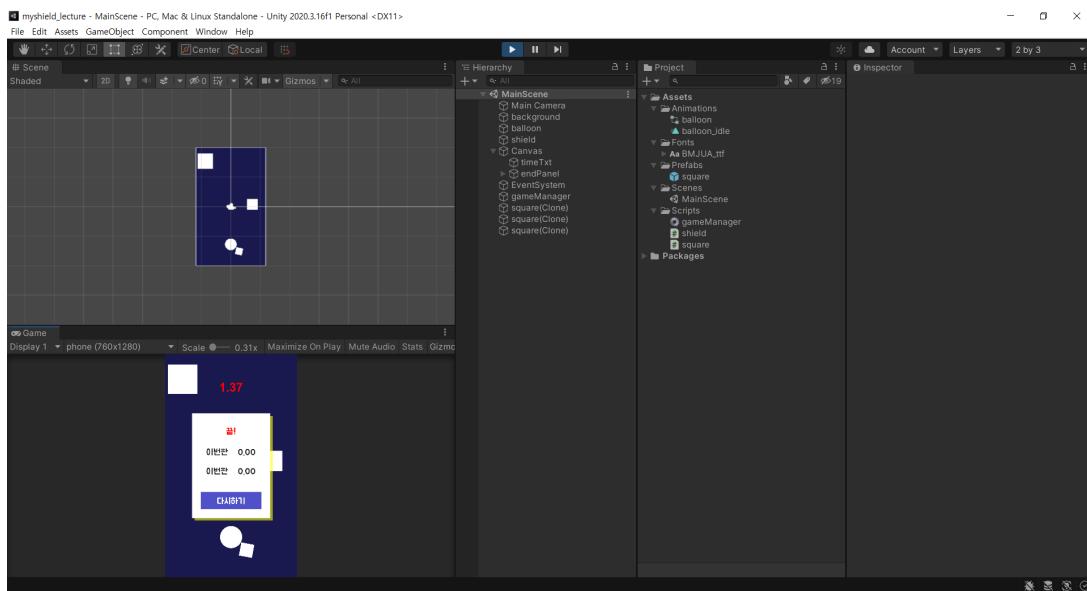
public void gameOver()
{
    Time.timeScale = 0.0f;
    endPanel.SetActive(true);
}
```

2. `square.cs` 네모가 풍선과 부딪히면 게임 종료하기

→ 우선, 풍선에 "balloon"이라는 tag를 주기

```
void OnCollisionEnter2D(Collision2D coll)
{
    if (coll.gameObject.tag == "balloon")
    {
        gameManager.I.gameOver();
    }
}
```

3. `Play` 해서 확인하기



▼ 3) 현재 점수 보여주기

1. `thisScoreTxt` 가져오기

```
public Text thisScoreTxt;
```

2. `gameOver()` 수정하기

```
public void gameOver()
{
    Time.timeScale = 0.0f;
    thisScoreTxt.text = alive.ToString("N2");
    endPanel.SetActive(true);
}
```

3. 한걸음 더 : Update() 함수를 멈추게 하기

→ Update()와 gameOver() 간의 약간의 시간차가 있기 때문에, 이것을 제어해보겠습니다.

→ 그래야 `timeTxt` 와 `thisScoreTxt` 가 같은 값으로 나온답니다!



```
bool isRunning = true;

void Update()
{
    if (isRunning)
    {
        alive += Time.deltaTime;
        timeTxt.text = alive.ToString("N2");
    }
}

public void gameOver()
{
    isRunning = false;
    Time.timeScale = 0.0f;
    thisScoreTxt.text = alive.ToString("N2");
    endPanel.SetActive(true);
}
```

▼ 4) 다시하기 만들기

1. `gameManager.cs` - 다시하기 함수 만들기

```

using UnityEngine.SceneManagement;

public void retry()
{
    SceneManager.LoadScene("MainScene");
}

```

2. 시간을 다시 켜주기

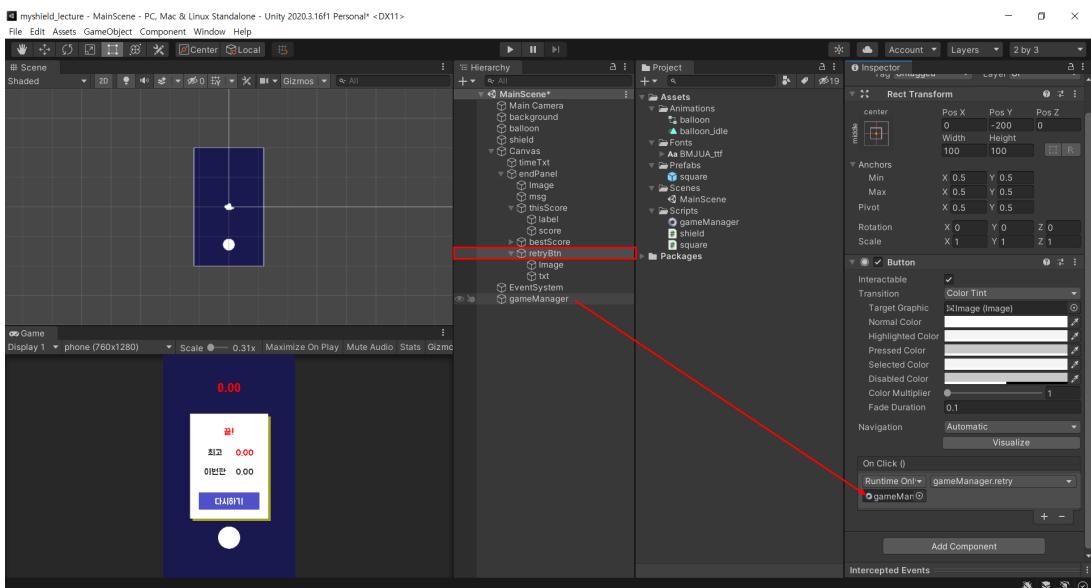
→ 이렇게 다시 할 때에는 반드시 "시간"도 되돌려 놓아야 합니다!

```

void Start()
{
    Time.timeScale = 1.0f;
    InvokeRepeating("makeSquare", 0.0f, 0.5f);
}

```

3. 다시하기 버튼에 retry() 함수 붙이기



10. 최고 점수 나타내기

▼ 1) 데이터를 보관하는 방법: [PlayerPrefs](#)



앱을 껏다 켜도 데이터가 유지되게 - 유니티에서 데이터를 보관하는 방법!

- 데이터 저장하기

```
PlayerPrefs.SetFloat("bestScore", 어떤숫자값);
PlayerPrefs.SetString("bestScore", 어떤문자열);
```

- 데이터 불러오기

```
어떤숫자값 = PlayerPrefs.GetFloat("bestScore");
어떤문자열 = PlayerPrefs.GetString("bestScore");
```

- 데이터를 저장했었는지 확인

→ 있으면 `true` 없으면 `false`

```
PlayerPrefs.HasKey("bestScore")
```

- 데이터를 모두 지우기

```
PlayerPrefs.DeleteAll();
```

▼ 2) 최고 점수 보여주기

1. 로직 생각하기

```
if (최고 점수가 없으면)
{
    최고점수 = 지금점수
}
else
{
    if (최고점수 < 지금점수)
    {
        최고점수 = 지금점수
    }
}
```

2. 구현하기

```
public void gameOver()
{
    isRunning = false;
    Time.timeScale = 0.0f;
```

```
thisScoreTxt.text = alive.ToString("N2");
endPanel.SetActive(true);

if (PlayerPrefs.HasKey("bestScore") == false)
{
    PlayerPrefs.SetFloat("bestScore", alive);
}
else
{
    if (PlayerPrefs.GetFloat("bestScore") < alive)
    {
        PlayerPrefs.SetFloat("bestScore", alive);
    }
}
}
```

3. 최고 점수 띄워주기

```
public Text bestScoreTxt;

bestScoreTxt.text = PlayerPrefs.GetFloat("bestScore").ToString("N2");
```



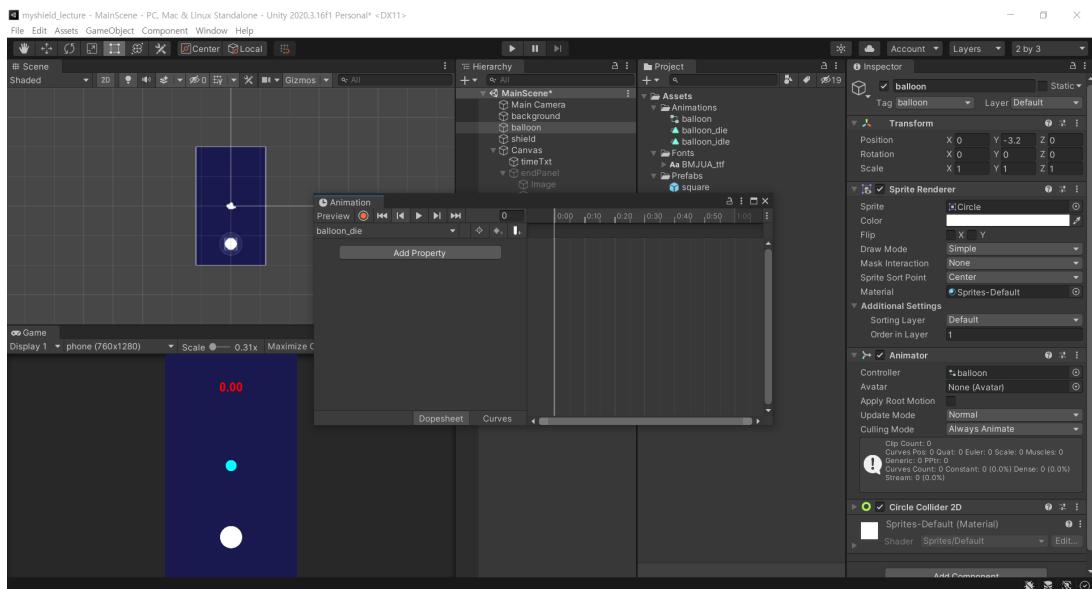
11. 풍선 애니메이션 전환하기

▼ 1) 풍선이 터지면서 끝나게 하기



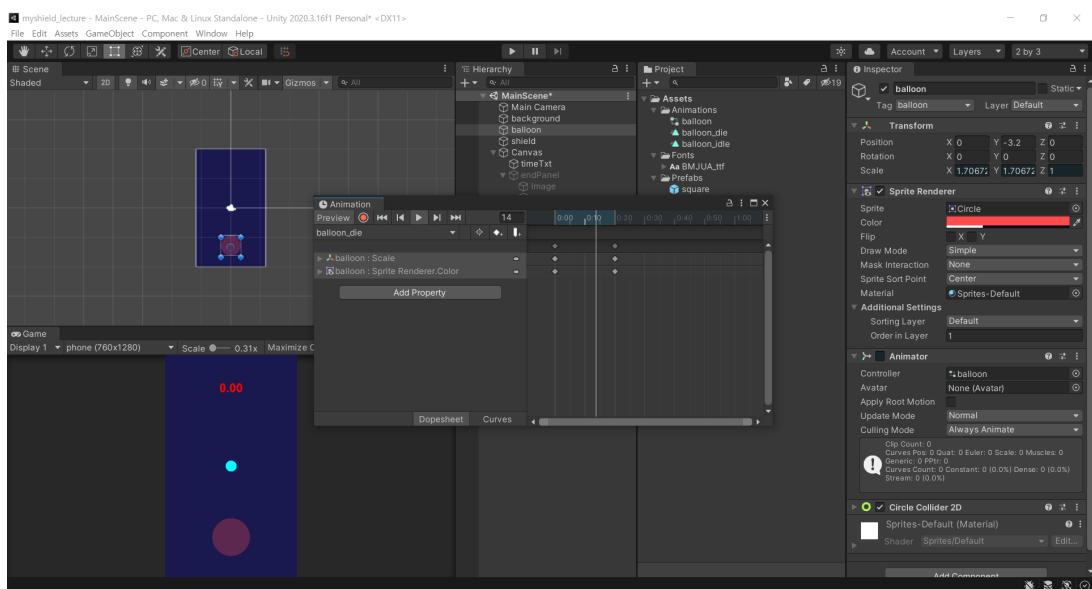
터지는 애니메이션을 만들어두고 → 풍선이 네모와 닿으면 "전환"하게 하는 것이죠!

- 터지는 애니메이션(balloon_die) 만들고 balloon에 끌어다 놓고 add New Clip 해 주기



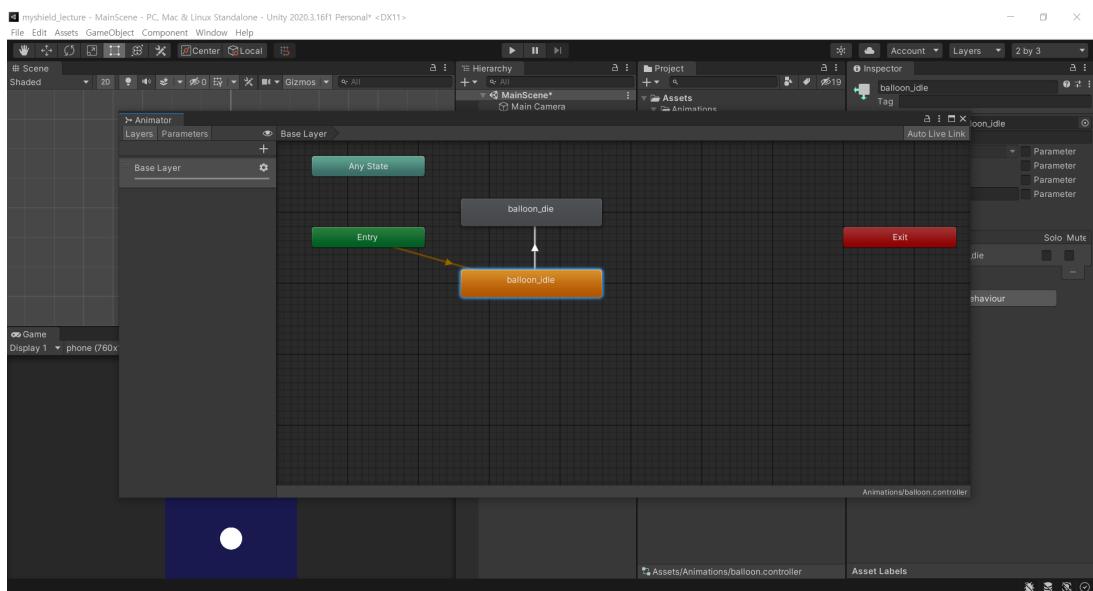
- 0:20 에 기록하기

→ `x:2, y:2` 그리고 `rgba ⇒ 255, 0, 0, 125` 으로, 터지는 것처럼!

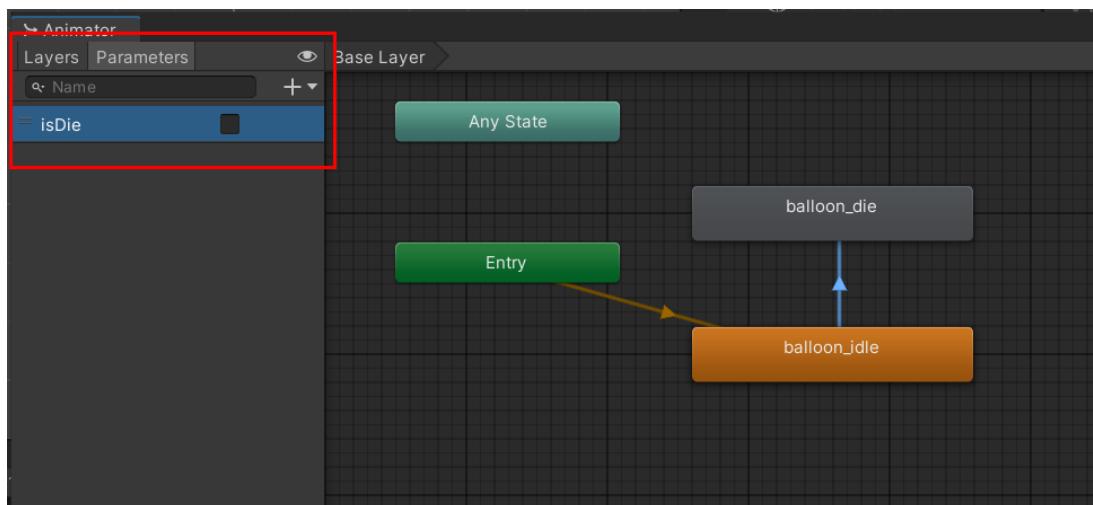


3. balloon animator를 열어서, idle → die로 transition 만들기

→ 마우스 오른쪽 클릭후 make transition 하면 됩니다!

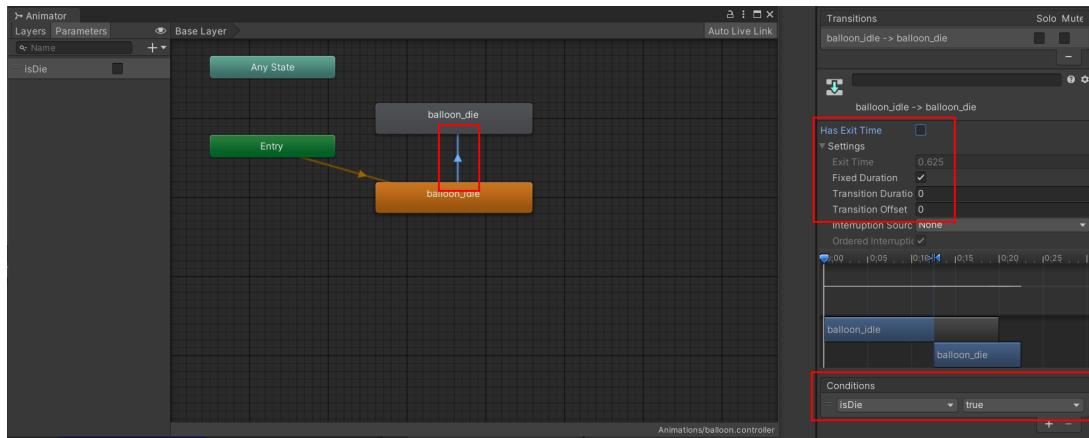


4. Parameters에, bool 형식의 `isDie` 를 만들기



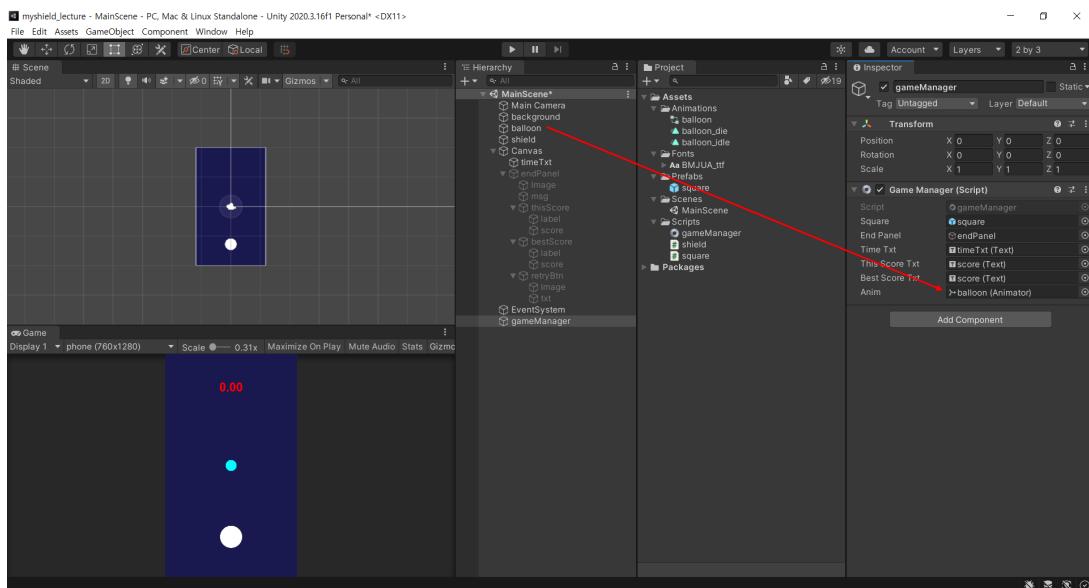
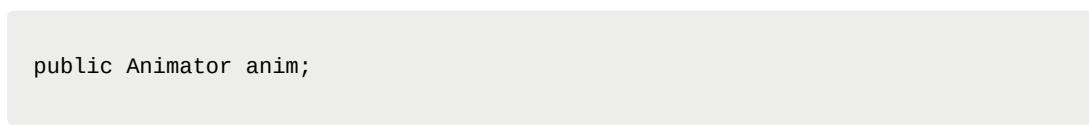
5. transition을 클릭하고 아래와 같이 세팅하기

→ has exit time 을 체크 해제해야 : 즉시 전환됩니다!



▼ 2) 풍선 애니메이션 전환하기

- gameManager.cs 에서 - animator를 받기



- gameOver() 할 때 isDie 값을 바꿔주기

```
public void gameOver()
{
    anim.SetBool("isDie", true);

    isRunning = false;
    Time.timeScale = 0.0f;
    thisScoreTxt.text = alive.ToString("N2");
    endPanel.SetActive(true);

    if (PlayerPrefs.HasKey("bestScore") == false)
```

```

    {
        PlayerPrefs.SetFloat("bestScore", alive);
    }
    else
    {
        if (PlayerPrefs.GetFloat("bestScore") < alive)
        {
            PlayerPrefs.SetFloat("bestScore", alive);
        }
    }
    bestScoreTxt.text = PlayerPrefs.GetFloat("bestScore").ToString("N2");
}

```

3. 확인하기 : 앗, 안된다!

- 그 이유는, 애니메이션이 나올 틈이 없이 시간을 멈추기 때문
- 0.5초 후에 시간을 멈추도록 Invoke로 처리하기!

```

public void gameOver()
{
    anim.SetBool("isDie", true);

    isRunning = false;
    Invoke("timeStop", 0.5f);
    thisScoreTxt.text = alive.ToString("N2");
    endPanel.SetActive(true);

    if (PlayerPrefs.HasKey("bestScore") == false)
    {
        PlayerPrefs.SetFloat("bestScore", alive);
    }
    else
    {
        if (PlayerPrefs.GetFloat("bestScore") < alive)
        {
            PlayerPrefs.SetFloat("bestScore", alive);
        }
    }
    bestScoreTxt.text = PlayerPrefs.GetFloat("bestScore").ToString("N2");
}

void timeStop()
{
    Time.timeScale = 0.0f;
}

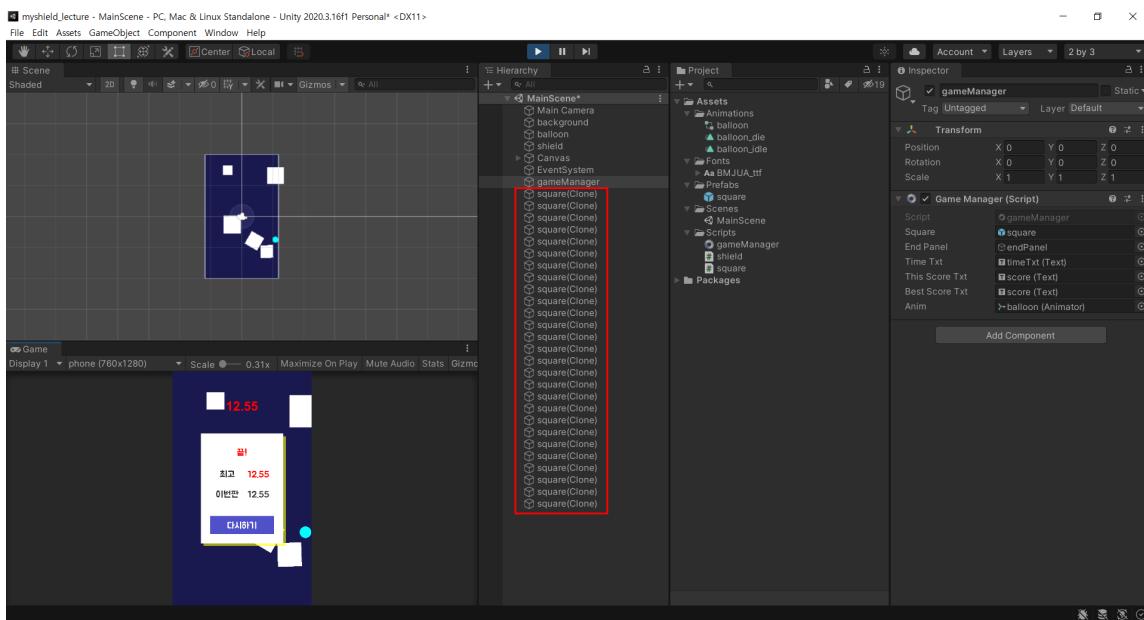
```

12. 숙제 - 떨어지는 네모를 없애기!



화면 밖으로 떨어진 네모들을 `Destroy(gameObject)` 해주기!

▼ 현재 상황

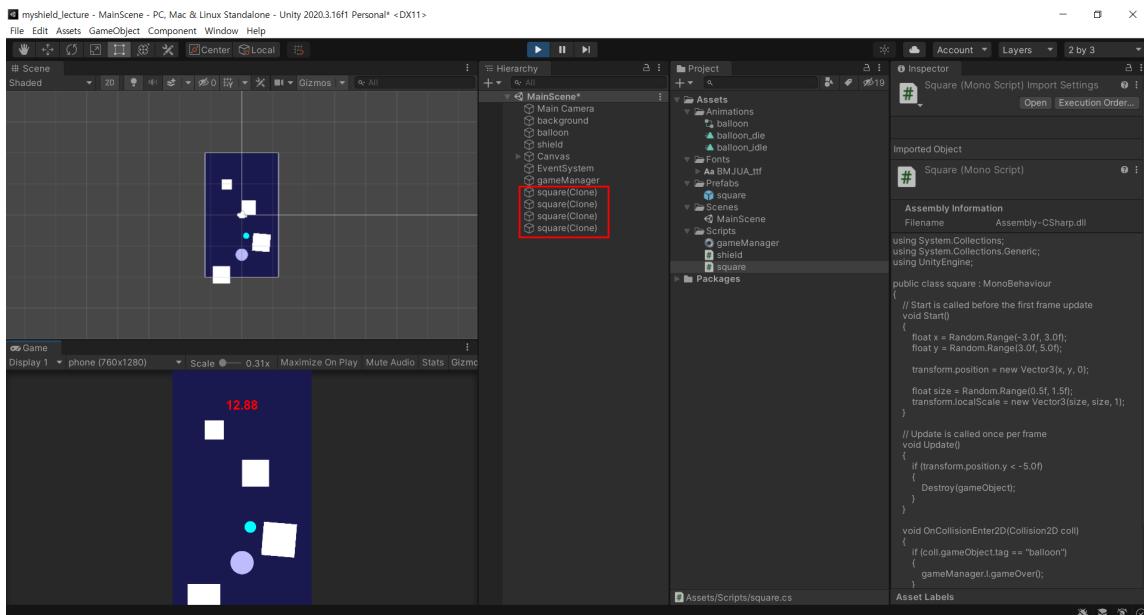


→ "뜨악" 시간이 지나면 네모가 계속 쌓이고 있었네요..!

→ 화면을 넘어가면 square를 Destroy 해줄 수 있을까요?

▼ 이렇게 되면 완성!

→ 화면에 보여지는 네모와 `square(clone)` 수가 일치하면 완성!



▼ 힌트요정 - 🧸

- `square.cs` 만 수정하면 된답니다!
- `Update()` 안에 딱 세 줄만 넣으면 됩니다! 딱 5분만 더 해보면 될 거예요!
- y좌표 구하기 ⇒ `transform.position.y` 기억나시죠!
- 없애라 ⇒ `Destroy(gameObject)` 기억나시죠!

```
Update()
{
    if (만약에 y좌표가 -5.0f 보다 작다면)
    {
        없애라;
    }
}
```

HW. 2주차 숙제 해설

▼ `square.cs` 코드

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

public class square : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        float x = Random.Range(-3.0f, 3.0f);
        float y = Random.Range(3.0f, 5.0f);

        transform.position = new Vector3(x, y, 0);

        float size = Random.Range(0.5f, 1.5f);
        transform.localScale = new Vector3(size, size, 1);
    }

    // Update is called once per frame
    void Update()
    {
        if (transform.position.y < -5.0f)
        {
            Destroy(gameObject);
        }
    }
}

void OnCollisionEnter2D(Collision2D coll)
```

```
{  
    if (coll.gameObject.tag == "balloon")  
    {  
        gameManager.I.gameOver();  
    }  
}
```

◀ 이전 주차

다음 주차 ▶

Copyright © TeamSparta All rights reserved.