



# [SCC] 빗물받는 르탄이 게임 만들기



매 주차 강의자료 시작에 PDF파일을 올려두었어요!

## [수업 목표]

1. 유니티 다뤄보기
2. C# 기본 문법 익히기
3. 유니티 기본 사용법 익히기

## [목차]

01. 오늘 배울 것 - [빗물받는 르탄이]
02. 기본 씬 구성
03. 애니메이션 맛보기
04. 캐릭터 움직이기
05. 빗방울 구현하기 - 빗방울 코딩하기
06. 빗방울 구현하기 - 랜덤한 빗방울
07. 빗방울 구현하기 - 반복 생성 로직
08. 점수 올라가게 하기
09. 게임 끝내기



모든 토글을 열고 닫는 단축키

Windows :

`Ctrl + alt + t`

Mac :

`⌘ + ⌥ + t`

## 01. 오늘 배울 것 - [빗물받는 르탄이]

### ▼ 게임개발종합반 수업의 목표와 범위



기본적으로 게임개발 종합반 수업은 "스스로 찾을 수 있는 단계"로 만드는 데에 있습니다.

### • 유니티 공부하는 방법

- 반복이 답이다 - 결국 우리가 쓰는 코드는 돌고 돈다-는 것을 알게 되실 거예요.
- 언어는 C# - 함께 찬찬히 C#부터 공부해 보아요. 🐼

### ▼ 한 번에 보는 강의 커리큘럼

- 4개 프로젝트 + 1개 추가 기능

- 1. 빗물 받는 르탄이 : 유니티 세팅, 기초 문법 연습
- 2. 풍선을 구해라! 백만 다운로드 게임 따라만들기 : 유니티 기초 복습
- 3. 고양이 밥 주기 게임 : hp바, 레벨 연습하기
- 4. 르탄이 카드 뒤집기 게임 : 보드 게임 기초 구현하기
- 5. 주변 기능 학습 : 스플래시 화면 구성, 광고 붙이기, 배포하기, 무료 에셋 구경하기

▼ 우리가 이번 프로젝트에 배울 것

- raindrop - 친환경 게임: 빗물 받는 르탄이


[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/12970db0-a2fa-4090-8721-dabc106195c7/빗물\\_받는\\_르탄이.mp4](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/12970db0-a2fa-4090-8721-dabc106195c7/빗물_받는_르탄이.mp4)

▼ [빗물받는 르탄이] 프로젝트 만드는 순서

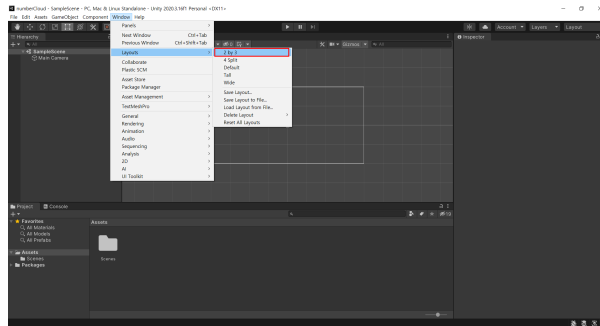
- 빗물 받는 르탄이 만드는 순서 보기
  1. 유니티 기본 세팅 및 씬 구성하기
  2. 캐릭터 움직이기, 캐릭터 클릭 시 방향 전환 구현하기
  3. 비 내리기 구현하기
  4. 비 충돌 구현하기
  5. UX (남은 시간 / 숫자합) 구현하기
  6. 게임 오버(팝업) 구현하기

## 02. 기본 씬 구성

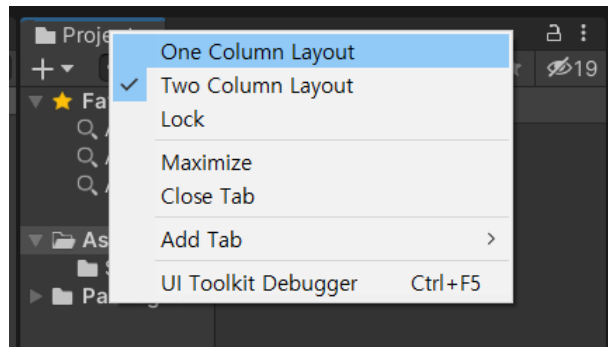
▼ 유니티 기본 개념

 유니티란? 충돌/중력/탄성/카메라 이런 것들을 쉽게 코딩할 수 있는 툴

- 유니티를 사용하는 이유
  - 게임 개발에 최적화된 개발 환경이기 때문입니다.
  - 특히 2D 게임은 거의 100% 유니티로 개발한 것으로 생각하면 됩니다.
- 유니티 화면 세팅 방법
  1. 프로젝트 생성 후 Window → Layouts → 2 by 3



## 2. project → 오른쪽 클릭 → one column layout 클릭



### • 유니티 개발 화면 개념 알기

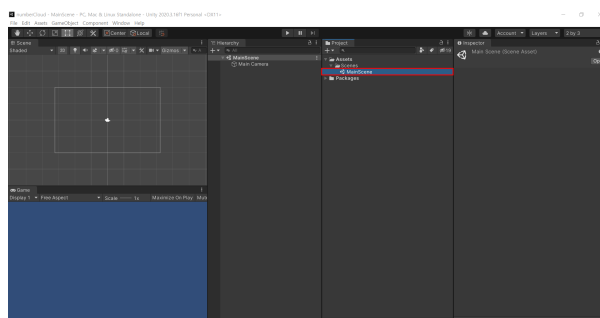
- Scene : 실제 게임의 구성요소를 보는 곳. 실질적인 게임 개발 씬
- Game : 게임이 실제로 보여지는 곳. play 버튼 클릭 후 볼 수 있음
- Hierachy : 게임 내 구성요소를 볼 수 있는 곳. 개발 시 자주 필요
- Project : 이 프로젝트에 포함된 파일들을 모아볼 수 있는 곳
- Inspector : 클릭한 요소의 속성과 정보를 보여주는 곳(차차 보면 알게 됨!)

### ▼ 기본 씬 구성하기

#### • 배경 세팅하기

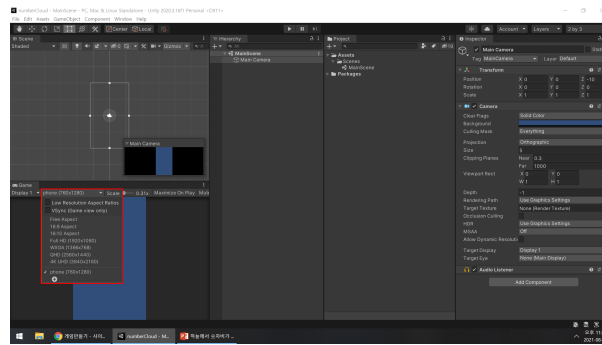
##### 1. 메인 씬 이름 바꾸기

→ Project에서 오른쪽 클릭 후 MainScene (대, 소문자 구분)으로 변경



##### 2. Game 씬 사이즈 바꾸기

→ +버튼을 클릭하고 760 x 1280 Phone을 입력 → Phone으로 변경

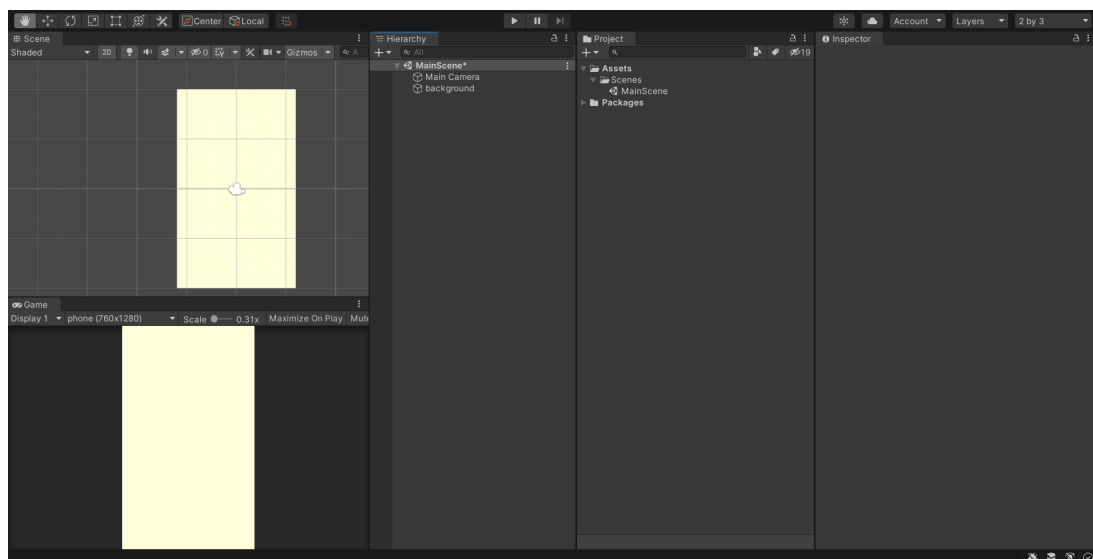


### 3. 배경 입히기

→ MainScene(우클릭) → Game Object → 2D Object → Sprite → Square 클릭 → background로 이름 바꾸기

→ SpriteRenderer의 Color 흰색 영역을 누르고 R,G,B,A 값을 255,255,220,255 로 맞추기

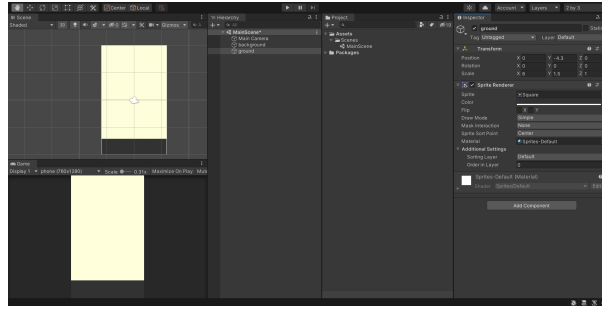
→ Transfrom의 Scale을 X:6, Y:10 으로 맞추기



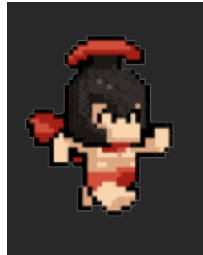
### ▼ UI 박스(점수) 세팅하기

#### • 검은색 박스 만들기

1. MainScene(우클릭) → 2D Object → Sprite → Square 클릭 → Ground로 이름 바꾸기
2. SpriteRenderer의 Color 흰색 영역을 누르고 R,G,B,A 값을 50, 50, 50, 255 로 맞추기
3. Transfrom의 Scale을 X:6, Y:1.5 으로 맞추기  
Transform의 Position은 Y:-4.3 으로 맞추기
4. Order in layer를 1로 맞추기



- 'Assets' 폴더에 캐릭터 넣어두기



#### ▼ [코드스니펫] 르탄이 이미지

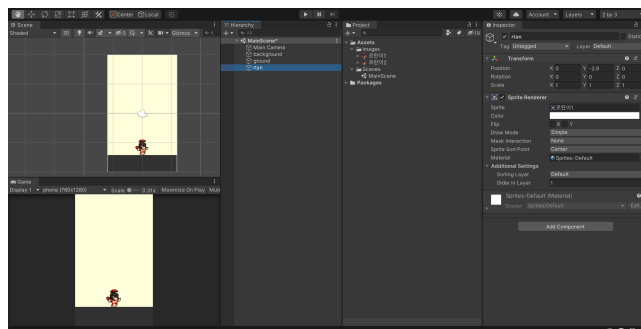
[https://s3.ap-northeast-2.amazonaws.com/materials.spartacodingclub.kr/game\\_new/week01/rtan.zip](https://s3.ap-northeast-2.amazonaws.com/materials.spartacodingclub.kr/game_new/week01/rtan.zip)

rtan.zip

1. Assets에서 Images 폴더 생성 → 르탄이 이미지 압축 풀고 끌어다놓기

#### • 르탄이 캐릭터 만들기

1. 2D Object → Sprite → Square 클릭 → rtan으로 이름 바꾸기
2. Sprite 부분에 르탄이1 이미지 끌어다 놓기
3. Order in Layer를 1로 바꾸기
4. Transfrom의 Position 값을 Y: -2.9 로 바꾸기

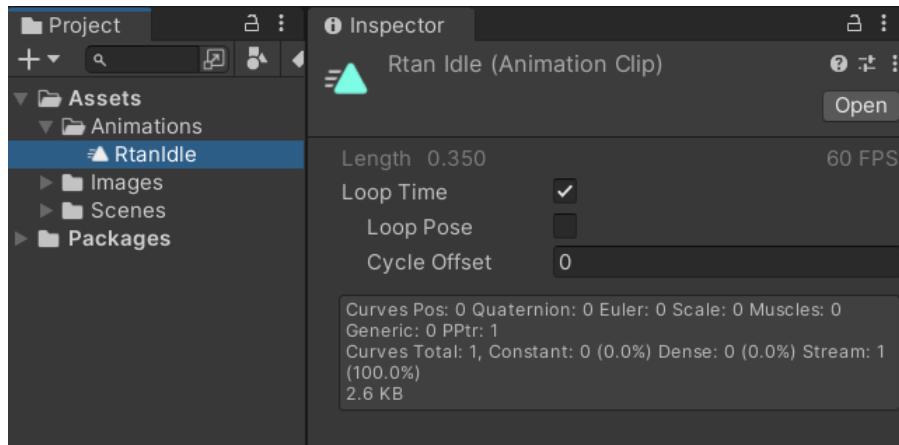


## 03. 애니메이션 맛보기

▼ 간단한 애니메이션을 입혀보기

- 애니메이션을 만드는 과정

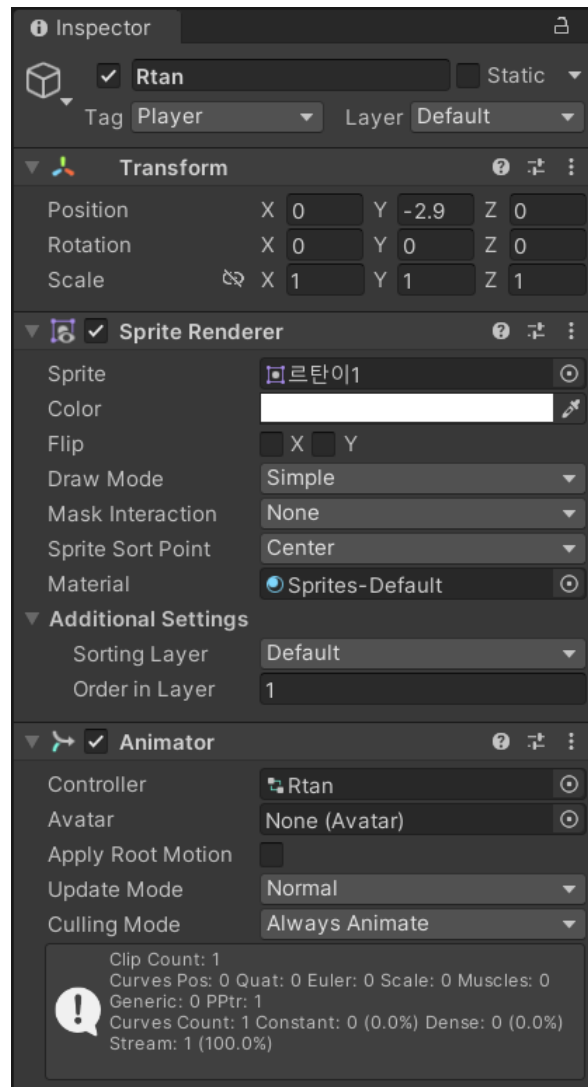
1. 애니메이션 폴더 만들기 (Assets → Animation)
2. 애니메이션 파일을 만들고, Loop Time에 체크



3. 이것을 만들어둔 르탄 캐릭터에 sprite에 끌어다 놓기

4. Controller가 생긴 것을 확인!

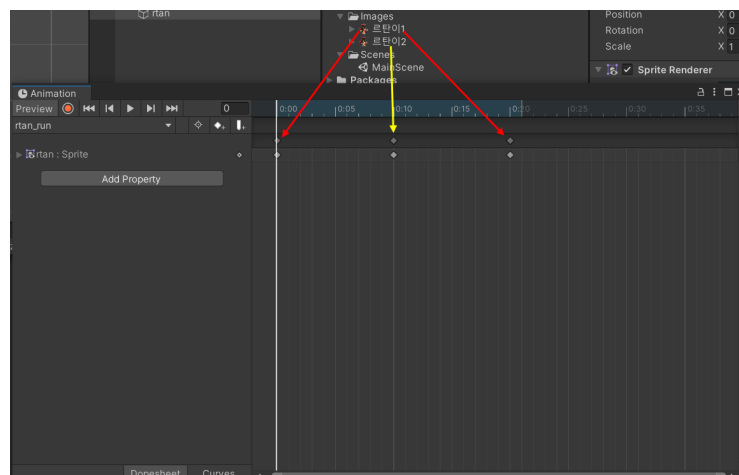
- Controller는 : Animation을 컨트롤하는 것  
(예 - 보통 상태 / 맞을 때 / 뛸 때 어떤 애니메이션을 써라)
- Animation은 : 동작 파일



#### ▼ 기본 Animation 만들어보기

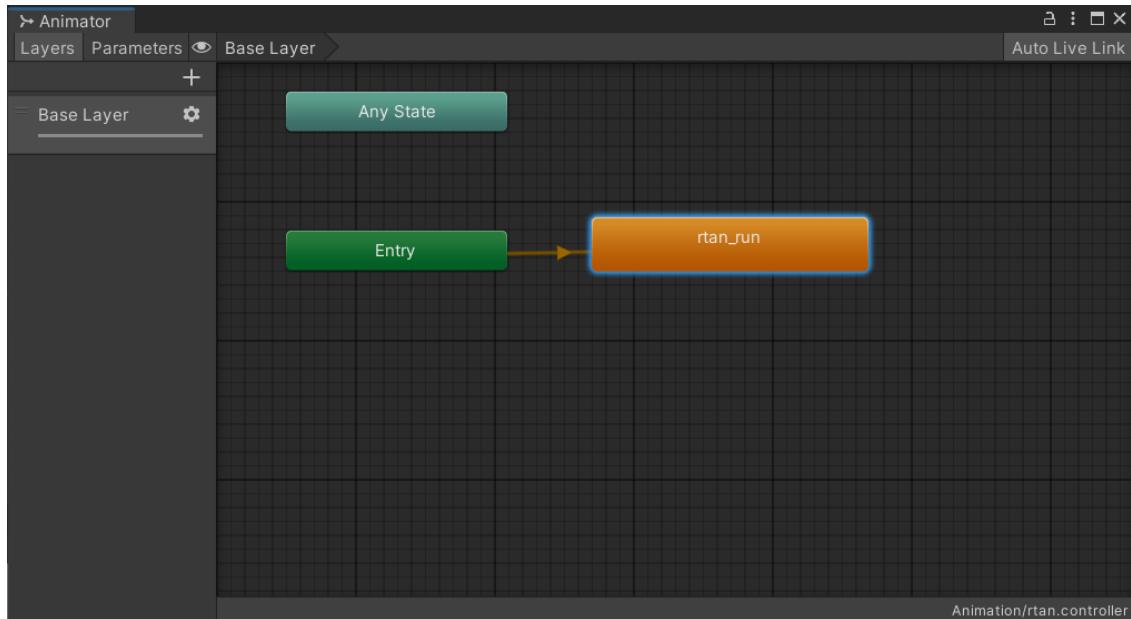
##### • Animation 만드는 방법

1. **RtanIdle.anim** 더블 클릭 후 → 르탄이 캐릭터 클릭
2. 르탄이1, 2파일을 적당한 시간 간격으로 끌어다 두기



- **Animator의 동작원리**

- rtan\_run은 **끝이 없는 애니메이션임**
- 시작하면 무조건 rtan\_run을 실행하게 됨



## 04. 캐릭터 움직이기

### ▼ 캐릭터에 코딩을 더하는 법

- **캐릭터가 코드를 가진다면?**

- 캐릭터에 코드를 입히는다는 의미는? → "캐릭터가 동작할 수 있게 만드는 것"
- 동작에서 꼭 알아야 할 두 가지 함수 → Start (너는 태어날 때) / Update (매 순간 이렇게 해라)

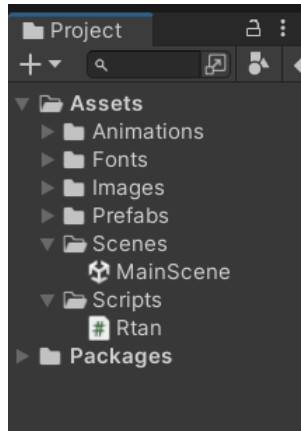
### ▼ Script 만들기

1. **Assets** 우클릭
2. **Create** 클릭
3. **Folder** 만들기 이름은 **Scripts**
4. **Create** 클릭
5. **C# script** 클릭하기 이름 **Rtan**

### ▼ [참고] - C#이란?

Microsoft가 개발한 코딩 개발 언어. 희한하게 유니티에서만 주류로 쓰이고 있다.



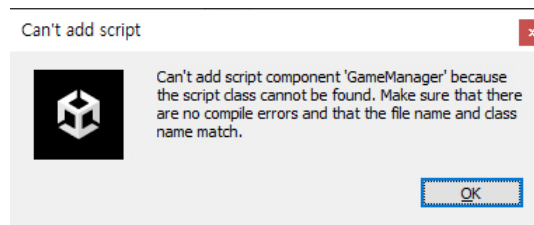


#### ▼ 캐릭터 좌우 움직임 코딩하기

##### • 캐릭터 오른쪽으로 이동하기

###### 1. 아래와 같이 입력하고 캐릭터에 스크립트를 끌어다 놓기

###### ▼ [여기서 잠깐!] 스크립트 끌어다 놓기 하였을 때 오류가 발생한다면?



스크립트를 끌어다 놓았을 때 현재 화면과 같은 오류가 발생하는 경우가 있습니다.

이 오류는 스크립트의 이름과 내부 코드의 클래스명이 일치하지 않는 경우에 주로 발생하는 오류로, 스크립트의 이름과 클래스명은 대, 소문자까지 일치해야 합니다!

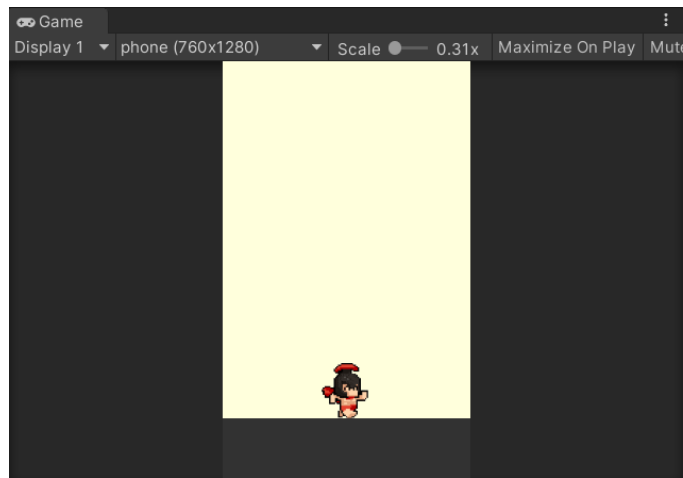
- rtan.cs



###### ▼ [코드스니펫] 캐릭터 오른쪽으로 이동하기

```
void Update()
{
    transform.position += Vector3.right * 0.05f;
}
```

###### 2. Play 버튼을 눌러 캐릭터가 오른쪽으로 이동하는 것 확인하기



### • 코드 동작 원리 알아보기

- transform의 의미: 캐릭터의 위치와 중, position을 계속 바꿔달라는 것
- `transform.position += Vector3.right * 0.05f;`
  - Vector3.right → Vector3(1, 0, 0);  
사용하기 편하게 유니티에서 제공하는 변수!
  - Vector3 (x,y,z) → x,y,z 의 3 차원 값  
Vector3.right \* 0.05f → Vector3(1, 0, 0) \* 0.05f;  
→ Vector3(1 \* 0.05f, 0 \* 0.05f, 0 \* 0.05f) → Vector3(0.05f, 0, 0);
  - += :  
transform.position = transform.position + Vector3.right \* 0.05f 의 축약!  
트랜스폼 안의 포지션을, Vector3 방향으로 계속 더해주세요
  - f란? ⇒ float입니다! 소수점을 나타내는 자료형이죠! 즉, 소수를 쓰고 싶으면 뒤에 f를 붙여줘야 합니다.
- 방향을 변수로 선언하기
  - 변수란?  
데이터를 담아 놓을 수 있는 박스 같은 것!  
반복해서 사용할 데이터/값을 박스에 담아 놓고 꺼내 쓴다!

```
float direction = 0.05f;

// Start is called before the first frame update
void Start()
{

}

// Update is called once per frame
void Update()
{
    transform.position += Vector3.right * direction;
}
```

- 르탄이 속도 일정하게 만들기

```
// Start is called before the first frame update
void Start()
{
```

```
Application.targetFrameRate = 60;
}
```

- frame : 1초에 호출/계산되는 횟수  
기기 성능이 좋을수록 frame이 높아지므로, 기기 성능과 관계없이 1초에 60번만 호출/계산하도록 고정!

- 캐릭터가 벽에 닿으면 다른 방향을 보게 하기

- 실시간 르탄이 위치 확인하기

- ▼ [코드스니펫] Debug.log

```
Debug.Log(transform.position.x);
```

- 앞으로 향하다 벽을 만나면 다른 방향으로 걷게 하기

```
float direction = 0.05f;

// Start is called before the first frame update
void Start()
{
    Application.targetFrameRate = 60;
}

// Update is called once per frame
void Update()
{
    if (transform.position.x > 2.8f)
    {
        direction = -0.05f;
    }
    transform.position += Vector3.right * direction;
}
```

- 뒤로 향하다 벽을 만나면 다른 방향으로 걷게 하기

```
float direction = 0.05f;
// Start is called before the first frame update
void Start()
{
    Application.targetFrameRate = 60;
}

// Update is called once per frame
void Update()
{
    if (transform.position.x > 2.8f)
    {
        direction = -0.05f;
    }
    if (transform.position.x < -2.8f)
    {
        direction = 0.05f;
    }
}
```

```
transform.position += Vector3.right * direction;
}
```

#### ▼ 캐릭터 벽에 닿으면 좌우 전환하기

##### • 캐릭터가 벽에 닿으면 다른 방향을 보게 하기

- 함께 해보기 - 벽에 닿으면 방향 전환하기

##### ▼ 힌트 보기

1. SpriteRenderer 컴포넌트에서 이미지의 두 버튼을 번갈아가며 눌러보기.
2. 코드로 어떻게 가져와서 조작할까? 고민하고 검색해 보기.  
(keyword : SpriteRenderer 스크립트 제어)
3. 이런 코드를 만나면 굿! `GetComponent<SpriteRenderer>();`

##### ▼ [코드스니펫] 방향 좌우 반전하기

```
SpriteRenderer renderer;           // 컴포넌트를 담아줄 변수(박스)를 만들고

void Start()
{
    Application.targetFrameRate = 60;
    renderer = GetComponent<SpriteRenderer>();    // 시작할 때 넣어준다.
}

void Update()
{
    ....(생략)
    renderer.flipX = true;
}
```

##### ▼ [완성코드] 방향 좌우 반전

```
float direction = 0.05f;
SpriteRenderer renderer;

// Start is called before the first frame update
void Start()
{
    Application.targetFrameRate = 60;
    renderer = GetComponent<SpriteRenderer>();
}

// Update is called once per frame
void Update()
{
    if (transform.position.x > 2.6f)
    {
        direction = -0.05f;
        renderer.flipX = true;
    }
    if (transform.position.x < -2.6f)
    {
        direction = 0.05f;
        renderer.flipX = false;
    }
}
```

```

    }
    transform.position += Vector3.right * direction;
}

```

▼ 마우스 클릭 시 캐릭터 방향 바꾸기

1. 마우스 클릭하면 → 움직이는 방향/이미지 방향 바꾸기

▼ [코드스니펫] 마우스 클릭 시 방향 바꾸기

```

if (Input.GetMouseButtonDown(0))
{
    sprite.flipX *= !sprite.flipX;
    direction *= -1;
}

```

```

float direction = 0.05f;
SpriteRenderer renderer;

// Start is called before the first frame update
void Start()
{
    Application.targetFrameRate = 60;
    renderer = GetComponent<SpriteRenderer>();
}

// Update is called once per frame
void Update()
{
    if (Input.GetMouseButtonDown(0))
    {
        direction *= -1;
        renderer.flipX = !renderer.flipX;
    }

    if (transform.position.x >= 2.6f)
    {
        direction *= -1;
        renderer.flipX = true;
    }
    if (transform.position.x <= -2.6f)
    {
        direction *= -1;
        renderer.flipX = false;
    }

    transform.position += Vector3.right * direction;
}

```

## 05. 빗방울 구현하기 - 빗방울 코딩하기

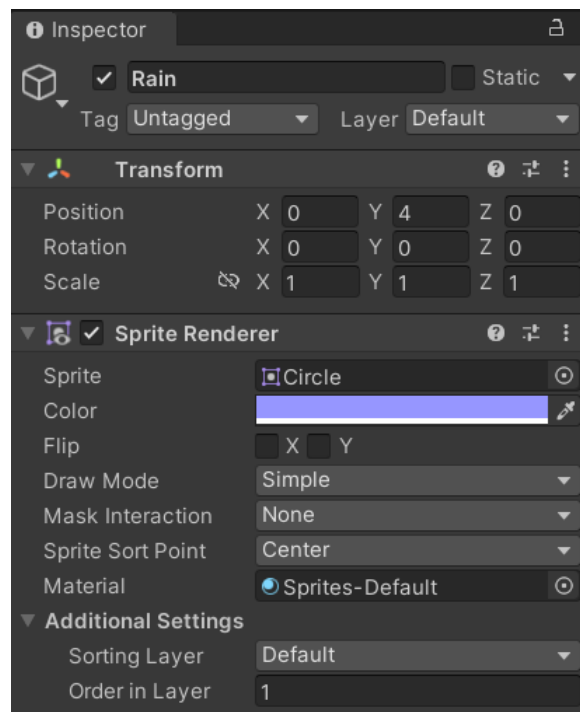
### ▼ 빗방울 내리게 하기

#### • 빗방울 특징

- 빗방울은 "하늘 랜덤 한 위치에서 내림"
- 큰 / 중간 / 작은 빗방울 존재 (3-2-1점)
- 캐릭터와 부딪히면 점수 더하기

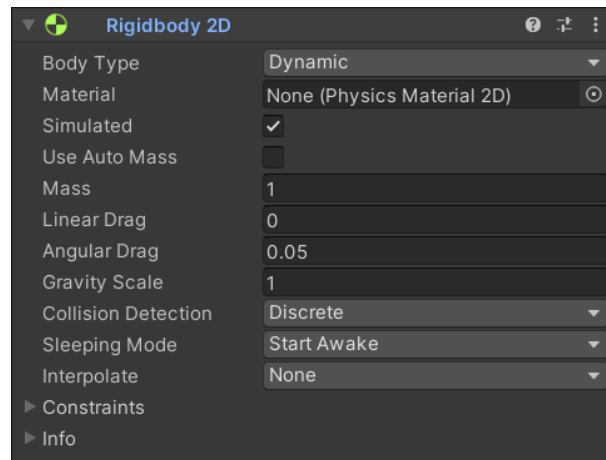
#### • 빗방울 그리기

- Hierarchy 탭에서 Sprite → Circle 클릭 → rain으로 이름 바꾸기
  - SpriteRenderer의 Color 흰색 영역을 누르고 R,G,B,A 값을 150,150,255,255 으로 맞추기
  - Transform의 Position 값을 Y:4 세팅하기



#### • 빗방울 떨어지게 하기

- rigidbody 2D를 달아 중력의 영향을 받게 하기



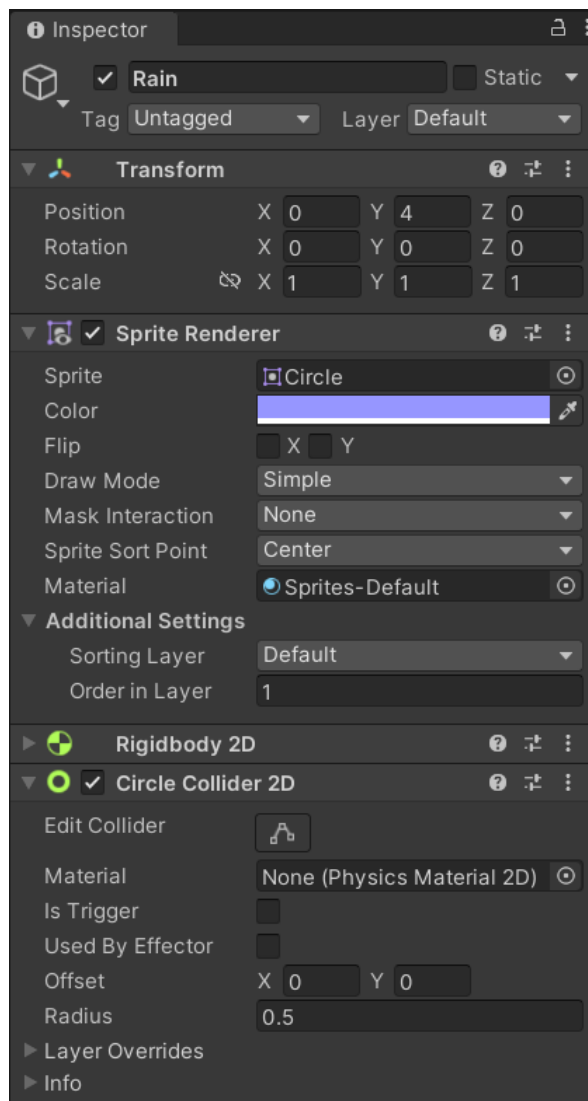
- 땅에 닿으면 없어지게 하기(충돌 세팅)

#### [충돌 조건]

- 충돌의 기본 조건은 둘 다 Collider 가 있어야 한다
- 둘 중 하나는 Rigidbody가 있어야 한다.

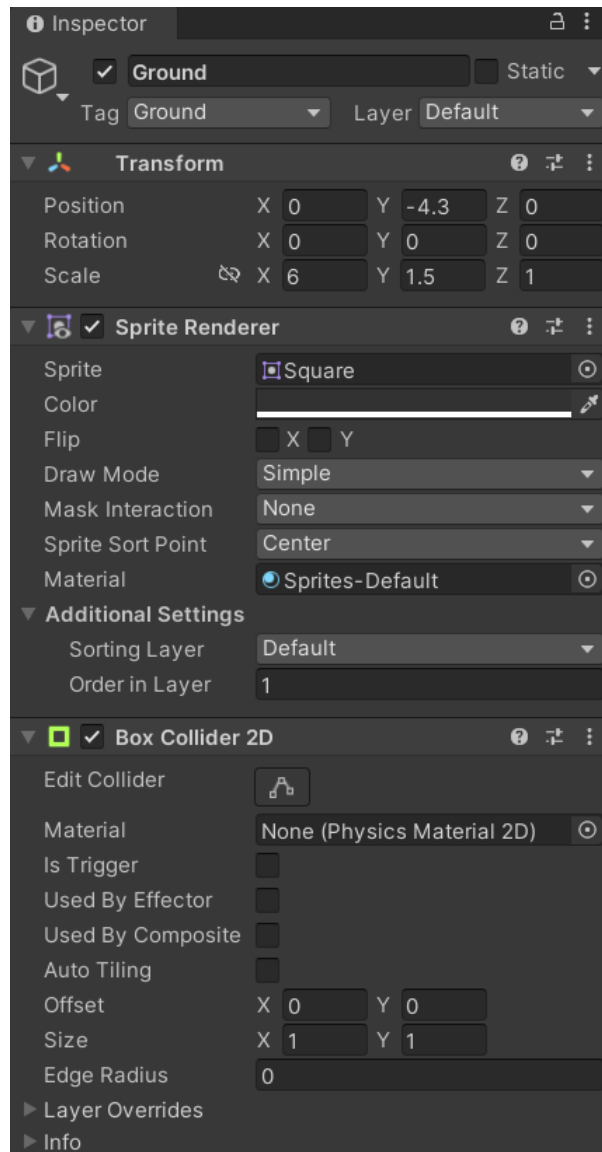
#### [충돌 세팅하기]

1. **Add component** → **circle collider 2d**를 달고, 반경 조정. 자세히 보면 초록색 선이 보임!

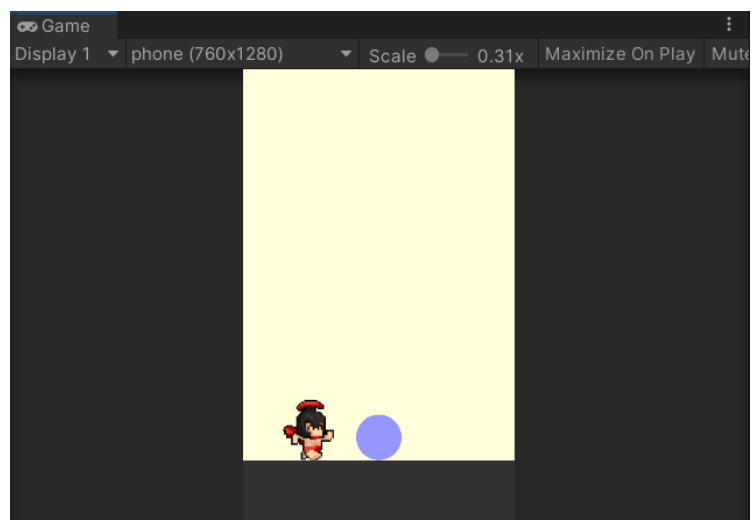


## 2. 바닥에도 box collider 2d를 달아주기



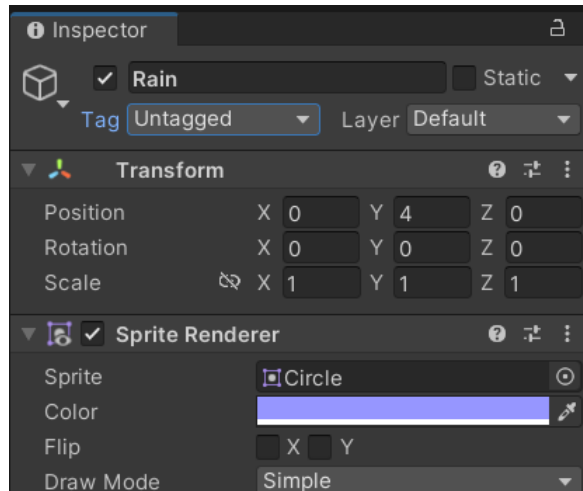


3. 게임을 실행하면 땅과 충돌을 합니다.

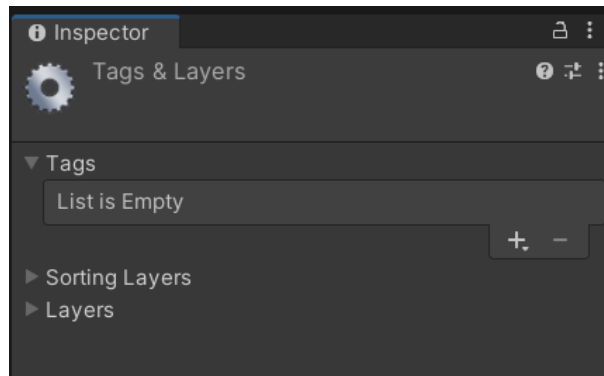


- 땅에 닿으면 없어지게 하기(충돌 조작)

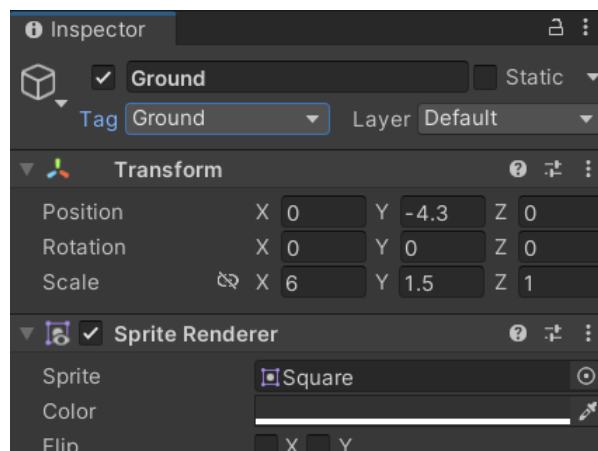
1. "땅"인지 알 수 있게, **Ground**라고 tag를 주기  
Untagged 클릭 > Add Tag 클릭



2. **Tags** 박스의 + 버튼을 눌러 **Ground**를 추가



3. 다시 **Ground** 게임 오브젝트에 **Ground** 태그 적용



4. 땅에 닿았는지 확인하기

- a. **Rain** 스크립트를 만들고, 빗방울에 붙이기
- b. `OnCollisionEnter2D` 함수는 다른 콜라이더에 부딪혔을 때 실행되는 내장함수
- c. coll (부딪힌 것의) tag 가 ground 이면, 땅에 닿았는지 확인하기

▼ [코드스니펫] 땅에 닿았는지 확인하기

```
private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.CompareTag("Ground"))
    {
        Debug.Log("닿았다!");
    }
}
```

5. 비가 없어지게 하기

- Debug.Log 대신, `Destroy(gameObject)`
- `gameObject` 는 나 자신 = 빗방울

▼ [코드스니펫] 비가 없어지게 하기

```
Destroy(gameObject);
```

▼ [완성코드] 비가 없어지게 하기

```
private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.CompareTag("Ground"))
    {
        Destroy(gameObject);
    }
}
```

- 빗방울 스프라이트가 ground 태그라는 물체와 부딪히면 빗방울 스프라이트를 없애기

## 06. 빗방울 구현하기 - 랜덤한 빗방울

▼ 빗방울 랜덤하게 나타나게 하기

1. 랜덤하게 위치 잡아주기

- **Start()** 함수에 랜덤 position 세팅하기

```
void Start()
{
    float x = Random.Range(-2.4f, 2.4f);
    float y = Random.Range(3.0f, 5.0f);
    transform.position = new Vector3(x, y, 0);
}
```

2. 랜덤하게 사이즈(큰/중간/작은) 잡아주기

- 어떤 사이즈로 나올지 생각하고
- 사이즈 변경: `transform.localScale = new Vector3(size, size, 0);`
- 색 변경: `GetComponent().color = new Color(100 / 255f, 100 / 255f, 255 / 255f, 255 / 255f);`

▼ 참고 : 색 변경할 때 255?

255.0f로 나눠주는 게 핵심!

▼ [코드스니펫] type이 1일 때

```
size = 0.8f;
score = 1;
GetComponent<SpriteRenderer>().color = new Color(100 / 255f, 100 / 255f, 255
```

▼ [코드스니펫] type이 2일 때

```
size = 1.0f;
score = 2;
GetComponent<SpriteRenderer>().color = new Color(130 / 255f, 130 / 255f, 255
```

▼ [코드스니펫] type이 3일 때

```
size = 1.2f;
score = 3;
GetComponent<SpriteRenderer>().color = new Color(150 / 255f, 150 / 255f, 255
```

▼ [완성코드] 랜덤한 사이즈의 빗방울 만들기

```
float size;
int score;

SpriteRenderer renderer;

// Start is called before the first frame update
void Start()
{
    renderer = GetComponent<SpriteRenderer>();

    float x = Random.Range(-2.7f, 2.7f);
    float y = Random.Range(3.0f, 5.0f);
    transform.position = new Vector3(x, y, 0);

    int type = Random.Range(1, 4);

    if(type == 1)
    {
        size = 0.8f;
        score = 1;
        renderer.color = new Color(100 / 255f, 100 / 255f, 1f, 1f);
    }
    else if(type == 2)
    {
        size = 1.0f;
        score = 2;
        renderer.color = new Color(130 / 255f, 130 / 255f, 1f, 1f);
    }
    else if(type == 3)
    {
        size = 1.2f;
```

```

        score = 3;
        renderer.color = new Color(150 / 255f, 150 / 255f, 1f, 1f);
    }

    transform.localScale = new Vector3(size, size, 0);
}

```

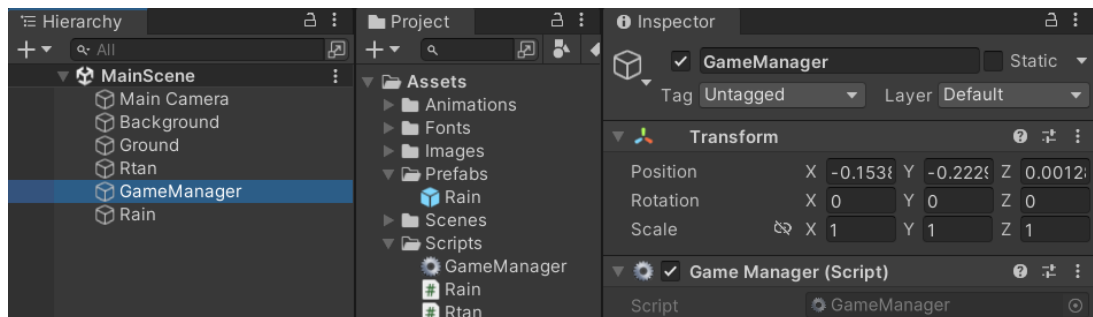
## 07. 빗방울 구현하기 - 반복 생성 로직

▼ 빗방울 계속 나오게 하기

👤 빗방울이 계속 나오게 하려면,  
GameManager 오브젝트를 만들고 → 빗방울을 Prefabs로 틀을 만들고 → Instantiate 복제합니다!

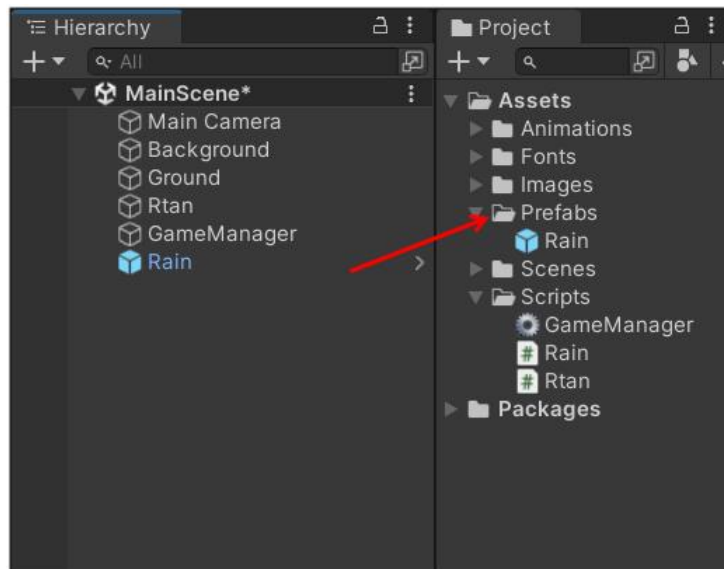
### 1. GameManager 만들기

- GameManager란? **게임 전체를 조율하는 오브젝트!**
  - 예) 점수 / 다시 시작 / 3번째 다시 시작에 부스터 / 광고 보기 등
- a. Hierarchy 창 빈 곳에 오른쪽 마우스 버튼 클릭  
Create Empty 버튼 클릭  
이름을 "GameManager"로 만들어줍니다.
- b. 마찬가지로 스크립트도 만들어 붙입니다. (어떻게 알았는지 아이콘 모양이 다르네요!)



### 2. 빗방울 복제하기 - Prefabs

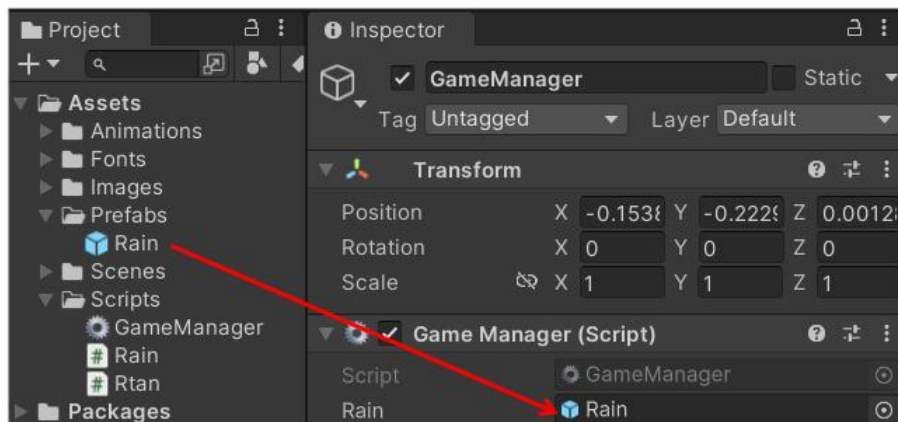
- 스프라이트는 봉어빵, 프리팹은 봉어빵 틀!
  - 빗방울 스프라이트 하나로는 빗방울을 복제할 수 없어요!
  - 그래서 빗방울 프리팹으로 복제할 수 있는 상태로 만듭니다!
- 프리팹 구현하기
  - Prefabs 폴더를 Project 탭에 하나 만듭니다
  - Hierarchy 탭의 rain 스프라이트를 Prefabs 폴더로 끌어다 놓기 & 오브젝트는 삭제하기



### 3. 빗방울 복제하기 - Instantiate

#### a. GameManager 스크립트에서 Prefabs의 rain을 끌어다 놓기

```
public GameObject rain;
```



#### b. 0.5초마다 한 번씩 실행되는 코드

##### ▼ [코드스니펫] InvokeRepeating 함수

```
InvokeRepeating("MakeRain", 0, 0.5f);
```

##### ▼ [코드스니펫] MakeRain 함수

```
void MakeRain()
{
    Debug.Log("비를 내려라!");
}
```

##### ▼ [완성 코드] - 1초마다 한 번씩 실행되는 코드

```
void Start()
{
    InvokeRepeating("MakeRain", 0, 1f);
}

void MakeRain()
{
    Debug.Log("비를 내려라!");
}
```

#### c. 빗방울 프리팹을 복제하기

##### ▼ [코드스니펫] Instantiate 함수


```
Instantiate(rain);
```

##### ▼ [완성 코드] - 빗방울 프리팹을 복제하기

```
void MakeRain()
{
    Instantiate(rain);
}
```

## 08. 점수 올라가게 하기

### ▼ 점수 보드 만들기

 점수는 UI라는 것을 사용해서 보여줄 거예요!

#### ▼ [참고] UI(User Interface)란?

UI는 Canvas라는 도화지 위에 그려지고, 카메라 위치와는 관계가 없이 보여집니다.

→

버튼 / 텍스트 / 순위를 보여줄 때에만 써줍니다. 😊

### 1. 폰트 적용하기

- Assets에 fonts 폴더 만들고 옮겨두기

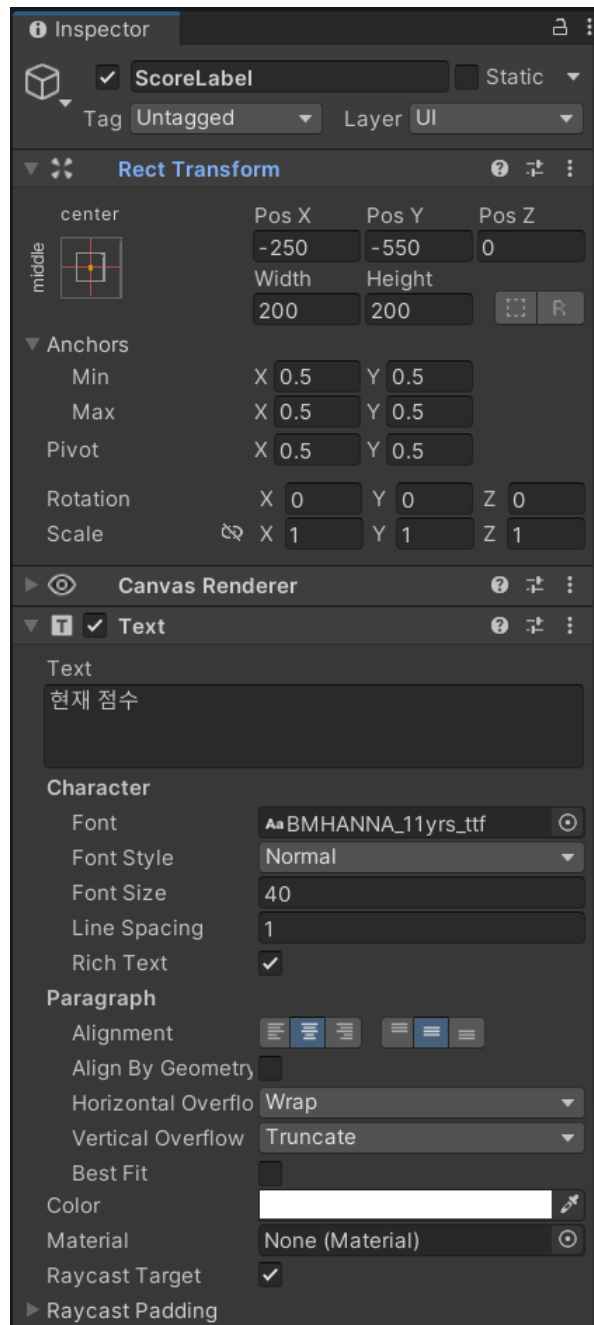
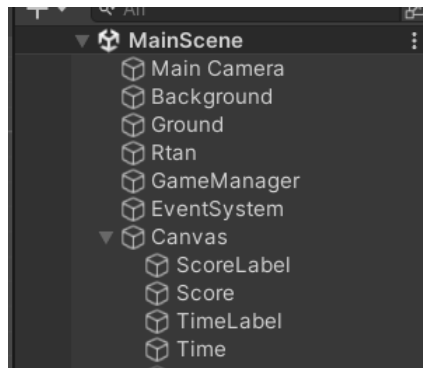
#### ▼ [코드스니펫] 배민-한나체

```
http://pop.baemin.com/fonts/hanna11yrs/BMHANNA_11yrs_ttf.ttf
```

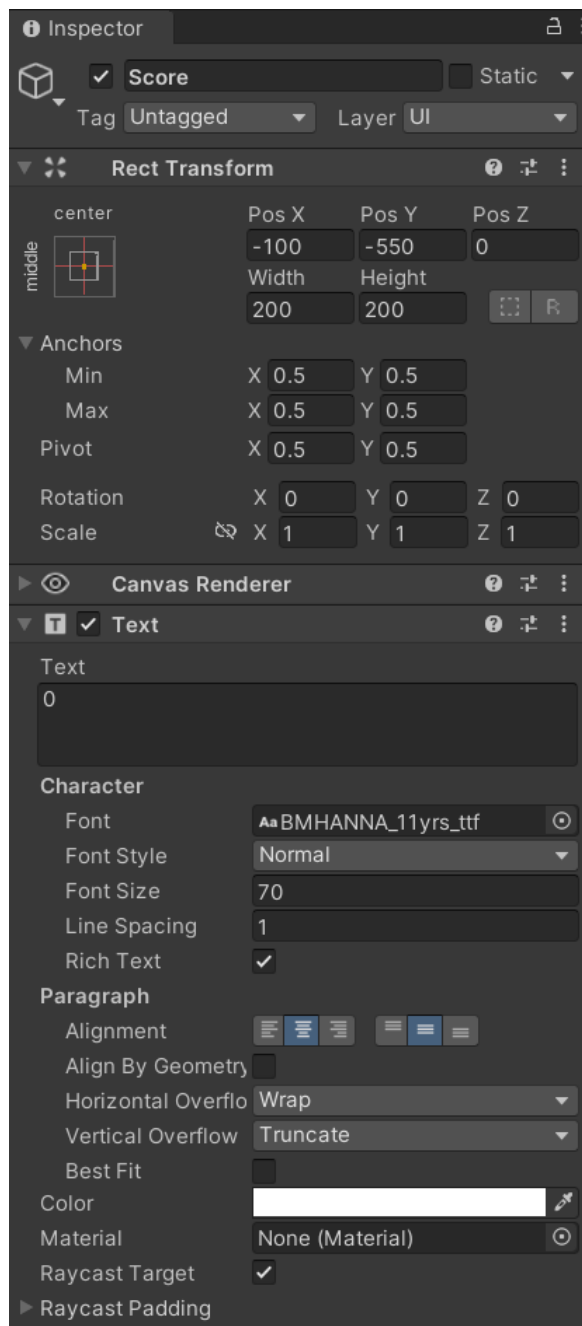
### 2. Sprite vs UI 그리고 Canvas

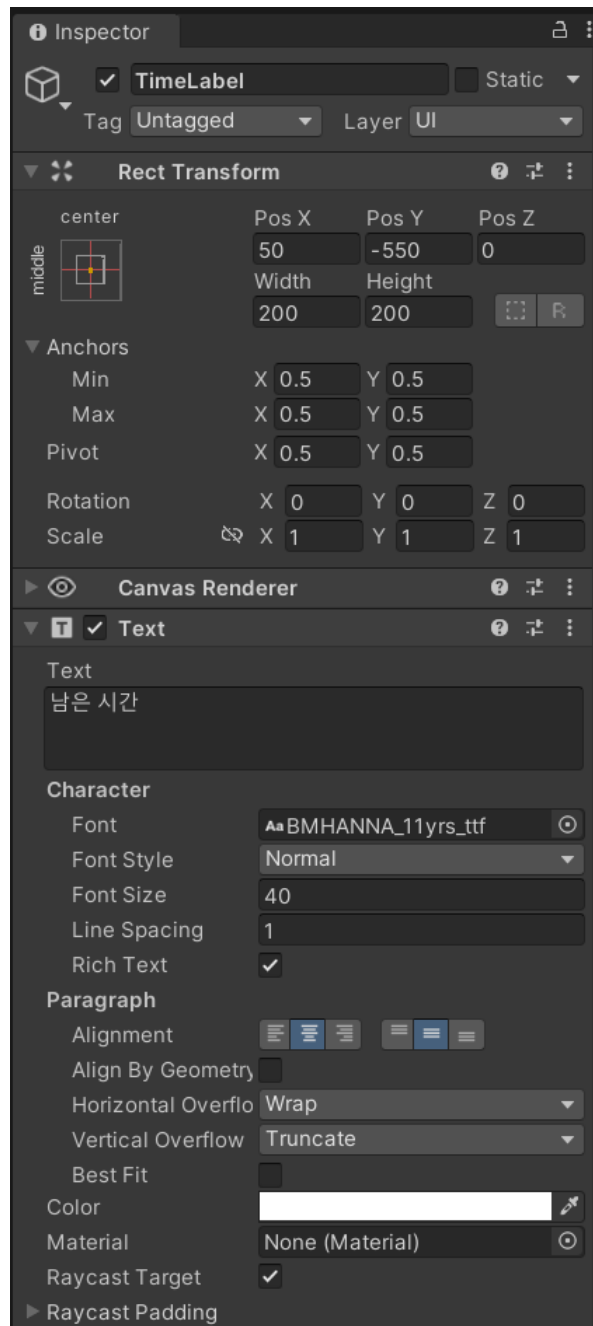
- UI** → **Legacy** → **Text** 클릭 → 아래 설정을 따라 하기 (폰트 사이즈, 위치 등)

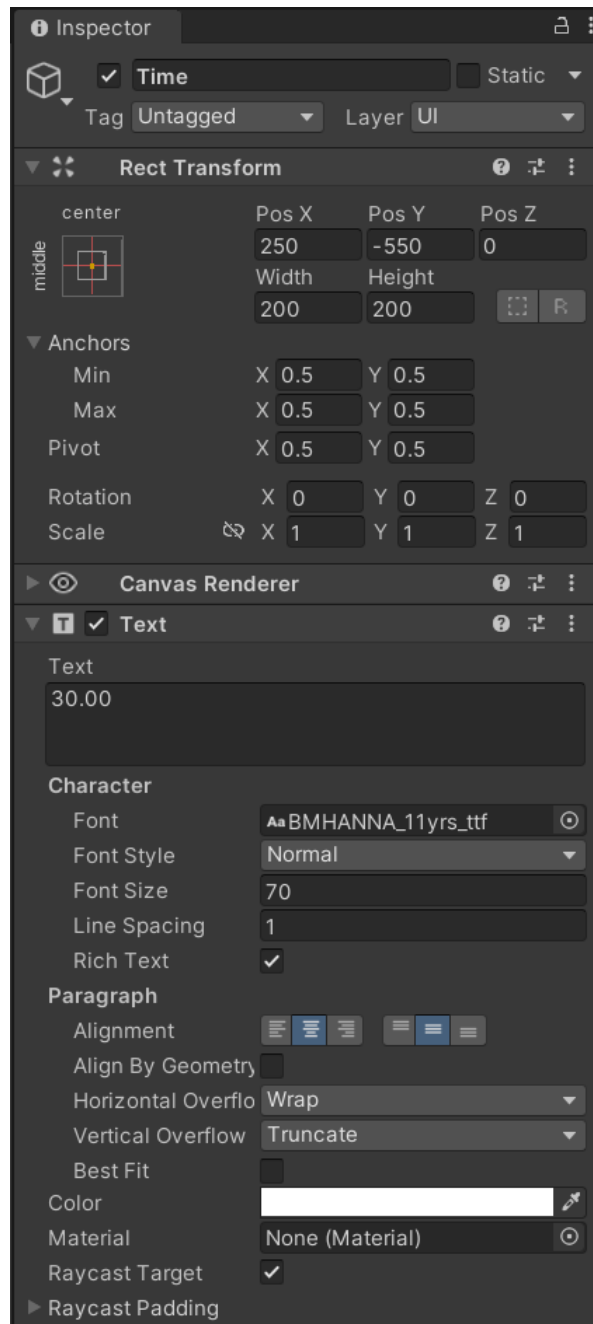
**Text**를 네 번 복사→붙여넣기 해서 아래와 같이 맞추기 (단축키 : Ctrl + D)











#### ▼ 싱글톤 화

👩 싱글톤이란? 어디서든 부를 수 있는 '하나'로 만들어주는 것! 곧 보게 되실 거예요!

#### ▼ [코드스니펫] 싱글톤 화

```
public static GameManager Instance;

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

- 실제 싱글톤은 훨씬 복잡하지만, 간단한 것부터 차근차근 하나씩 추가해 나가 보겠습니다!

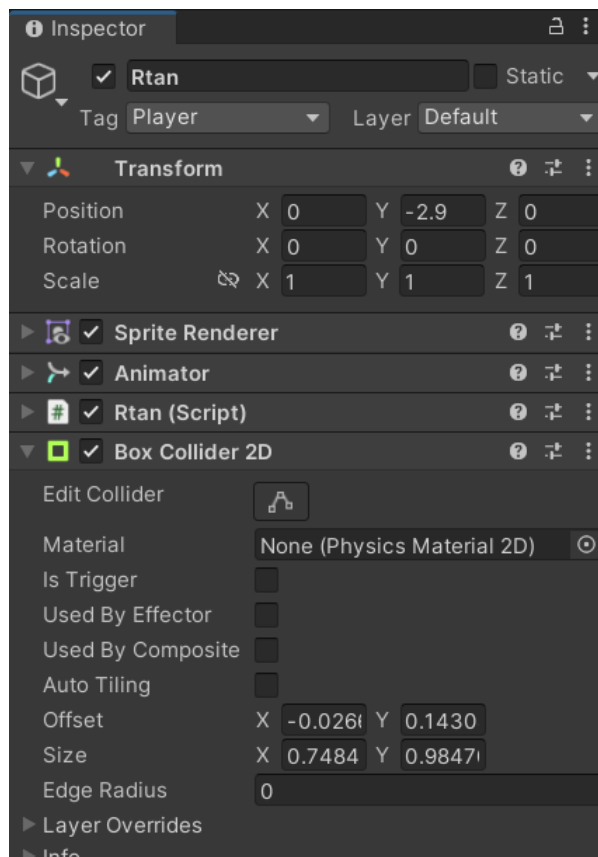
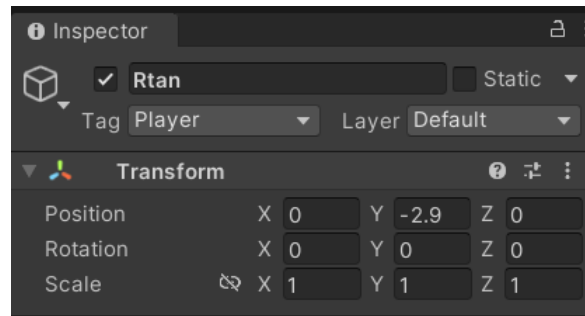
#### ▼ 점수 올라가는 함수 만들기

```
int totalScore = 0;

public void AddScore(int score)
{
    totalScore += score;
}
```

▼ 캐릭터에 맞으면 점수 올라가게 하기

#### 1. 캐릭터에 "Player" tag 적용 + collider 추가



#### 2. 빗방울 - 캐릭터에 맞으면 점수 올라가고 + 사라지기

```
private void OnCollisionEnter2D(Collision2D collision)
{
    if(collision.gameObject.CompareTag("Ground"))
    {

```

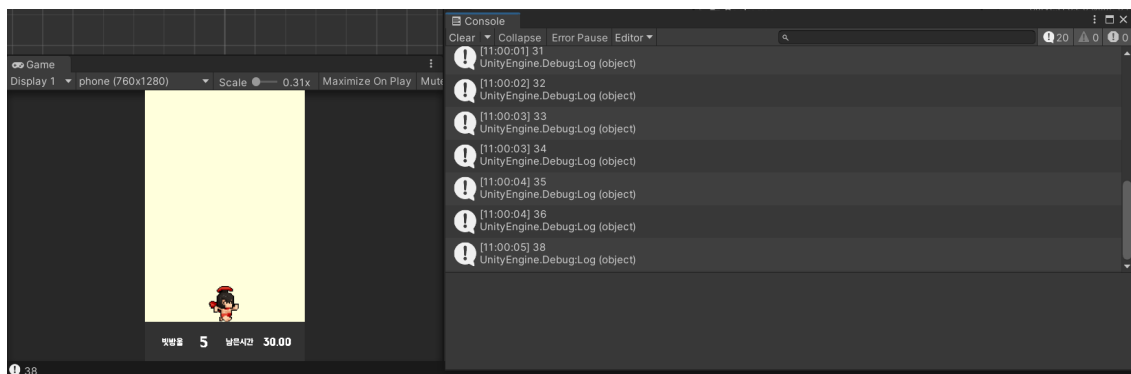
```

        Destroy(this.gameObject);
    }

    if (collision.gameObject.CompareTag("Player"))
    {
        GameManager.Instance.AddScore(score);
        Destroy(this.gameObject);
    }
}

```

### 3. GameManager - AddScore 함수에 Debug.Log를 걸어서 확인 → 잘 된다!



#### ▼ 올라가는 점수 표기하기

##### 1. UI Text 받기

```
using UnityEngine.UI;
```

```
public Text totalScoreTxt;
```

##### 2. Text 바꿔주기

- 점수가 오를 때마다 Text가 변경되도록 합니다

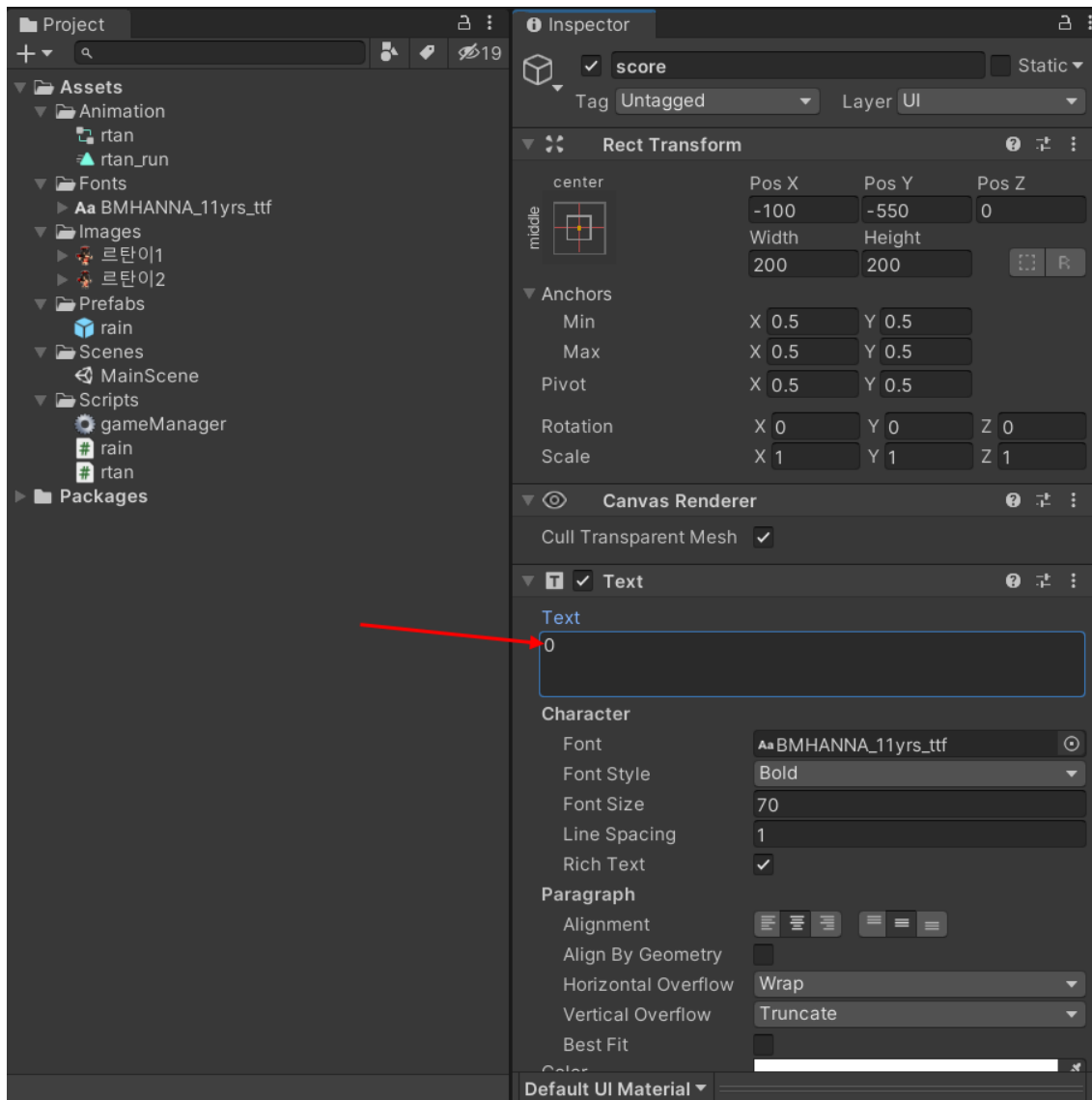
```

public void AddScore(int score)
{
    totalScore += score;
    totalScoreTxt.text = totalScore.ToString();
}

```

##### 3. 처음 스코어는 0으로 만들어주기

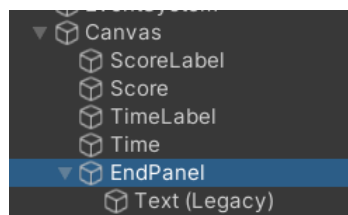
- 처음에 게임을 시작하면 0부터 시작해야겠죠?
- 첫 스코어를 0으로 시작하도록 만들어 줍시다
  - UI의 score 텍스트를 inspector 탭에서 Text 값을 0으로 하기




## 09. 게임 끝내기

### ▼ Retry 패널 만들기

- **Retry** 판넬을 만들기 위해서 할 것
  - image 사이즈: 400 / 250
  - txt 사이즈: 80
  - 글자 색상 (255, 255, 255, 255)
  - 배경 색상 (232, 52, 78, 255)
  - Inactive로 만들어두기



## ▼ 시간이 가게 하기

 gameManager 스크립트에서 시간이 가게 해 봅시다!

- 시간이 흐르게 하기

```
public Text scoreText;
public Text timeText;

int totalScore = 0;
float totalTime = 30.0f;

void Update()
{
    totalTime -= Time.deltaTime;
    timeText.text = totalTime.ToString("N2");
}
```

- 멈추게 하기

```
void Update()
{
    if(totalTime > 0f)
    {
        totalTime -= Time.deltaTime;
    }
    else
    {
        Time.timeScale = 0.0f;
        totalTime = 0.0f;
    }
    timeText.text = totalTime.ToString("N2");
}
```

## ▼ 0초에 Retry 패널 나오게 하기

1. Panel 받기

```
public GameObject endPanel;
```

2. Panel 나오게 하기

```
void Update()
{
    if(totalTime > 0f)
    {
        totalTime -= Time.deltaTime;
    }
    else
    {

```

```

        totalTime = 0f;
        endPanel.SetActive(true);
        Time.timeScale = 0f;
    }

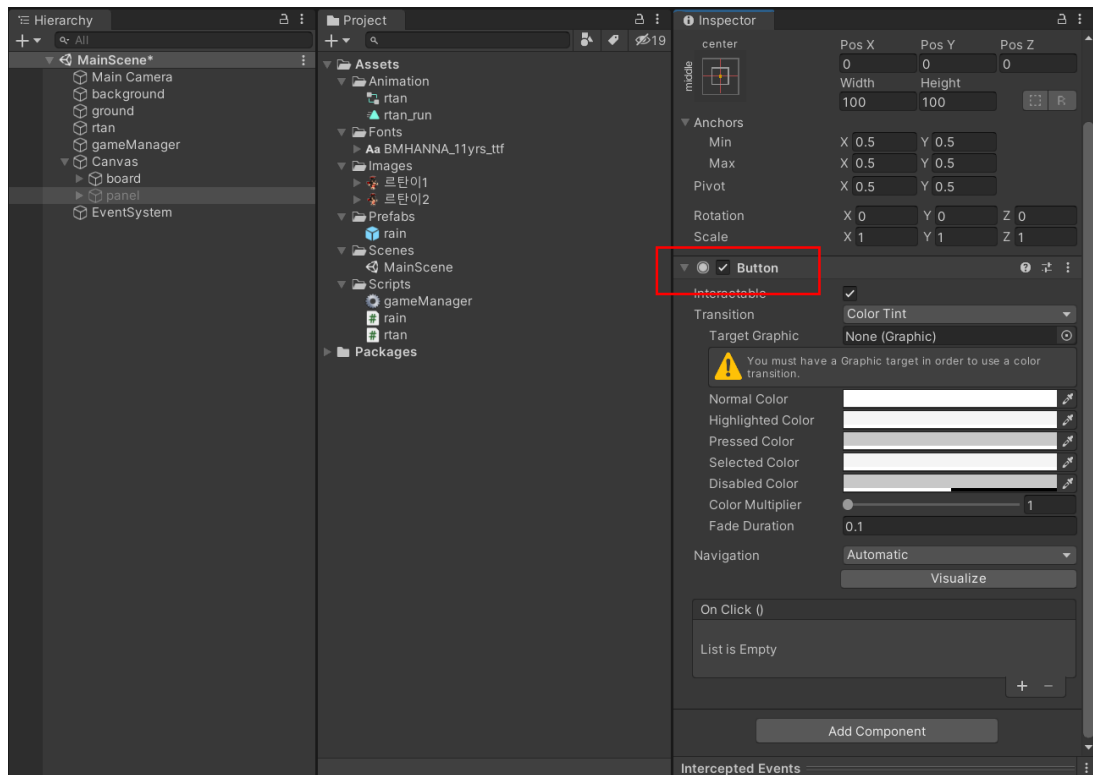
    timeTxt.text = totalTime.ToString("N2");
}

```

▼ 판넬 클릭하면 다시 시작하게 하기

### 1. 판넬에 button 달기

- panel을 클릭한 뒤, inspector 탭에서 **Button**을 클릭해주세요!



### 2. RetryButton 스크립트 만들고 판넬에 달기

- RetryButton 스크립트를 만들어줍니다.
  - 아래와 같이 코드를 작성해줍니다.

```

using UnityEngine;
using UnityEngine.SceneManagement;

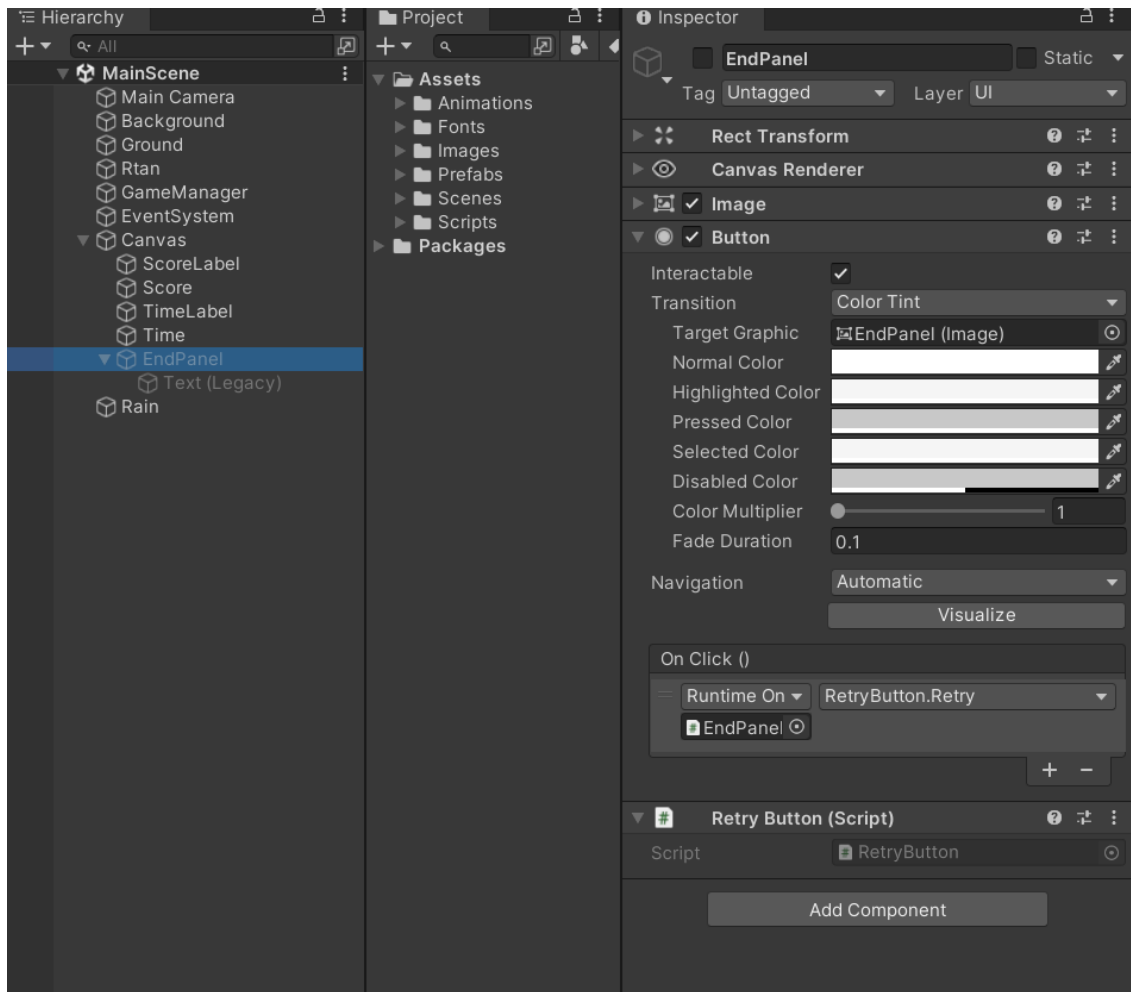
public class RetryButton : MonoBehaviour
{
    public void Retry()
    {
        SceneManager.LoadScene("MainScene");
    }
}

```



- EndPanel에 스크립트를 붙여줍니다.

### 3. onclick 연결하기



#### ▼ 초기화해주기

- 게임을 초기화하는 것은 어디서 하죠? 중앙에서!
  - 게임에서 중앙은 **GameManager!**
- 초기화해야 할 요소들은?
  - `timeScale`

```
private void Awake()
{
    Instance = this;
    Time.timeScale = 1.0f;
}
```


#### ▼ 수업 전체 코드 받기

👉 수업 전체 코드가 필요하다면?

👉 이전 주차

다음 주차 👉

 [SCC] 유니티 설치하기

 [SCC]풍선을 지켜라 게임 만들기

---

Copyright © TeamSparta All rights reserved.