

3D콘텐츠 이론 및 활용

11주(1). Trigger 처리

- Trigger
- Tag

학습목표

- 플레이어가 벽과 충돌 했을 때, 또는 아이템과 충돌 했을 때 각각 처리하는 방식의 차이점을 이해한다.
- 플레이어가 아이템과 같은 오브젝트를 획득할 때 일어나는 일과 처리해야 일을 스크립트로 작성할 수 있다.

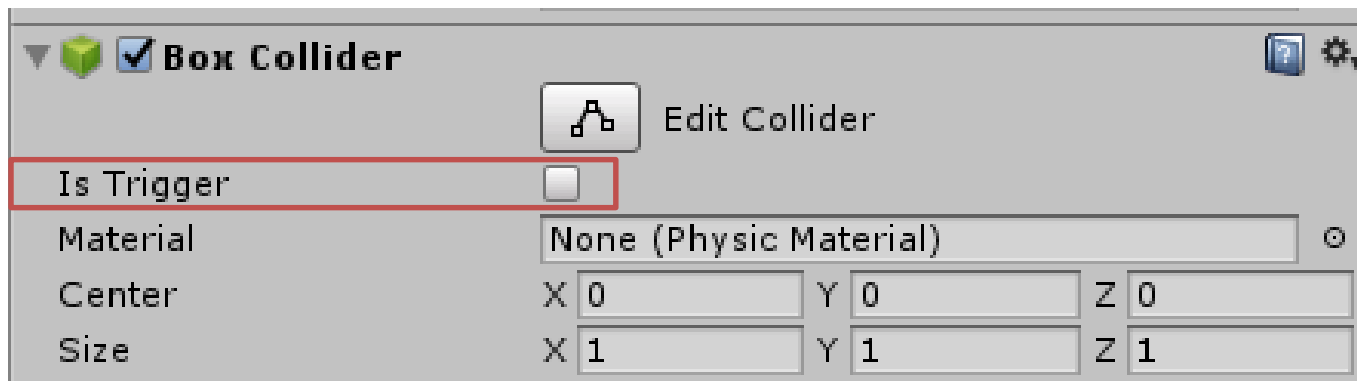
학습내용

- Tag 처리
- Trigger 처리 스크립트

Trigger 함수의 종류

- 두 객체가 접촉했을 때 서로 튕겨 나가지않고 그냥 통과하게 됨.
- Trigger를 쓰기 위해서는 해당 Collider의 [Is Trigger] 항목을 체크해야 함

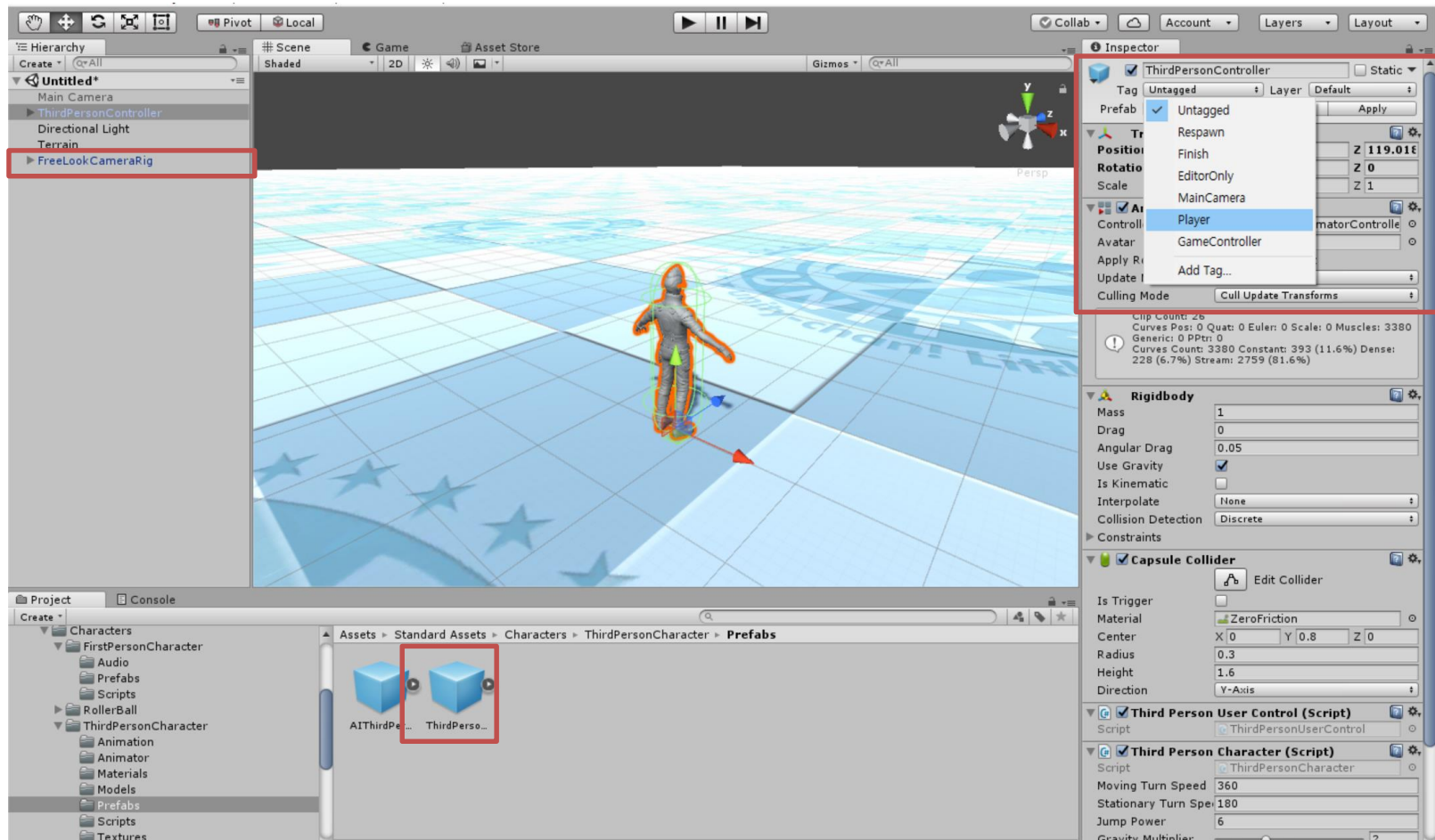
- void OnTriggerEnter(Collider any) { }
- void OnTriggerStay(Collider any) { }
- void OnTriggerExit(Collider any) { }



1. 기본 캐릭터 배치

1) Ethan Characters import

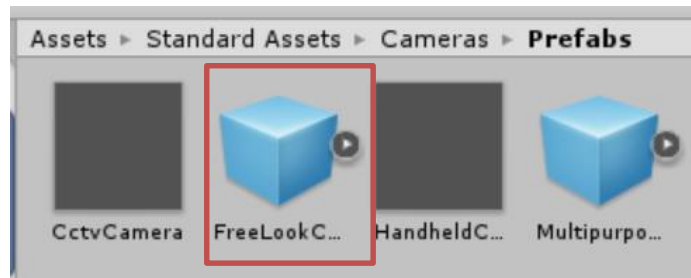
- \Assets\Standard Assets\Characters\ThirdPersonCharacter\Prefabs



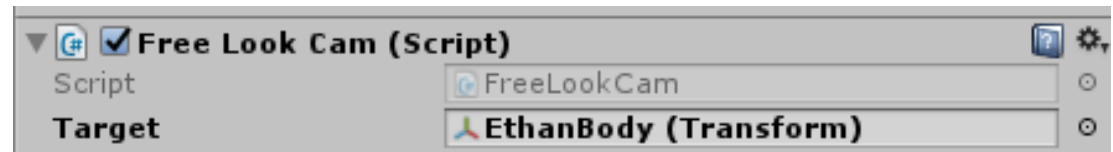
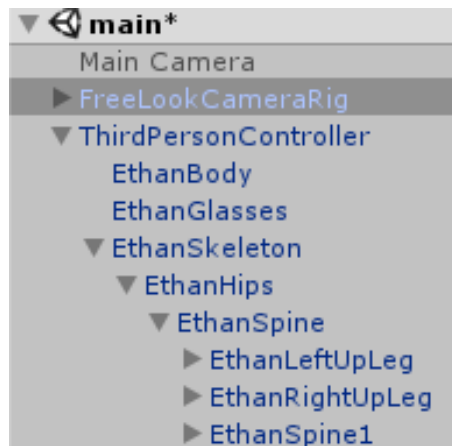
2. 자유시점 카메라 적용

2) import Cameras

- ₩Assets₩Standard Assets₩Cameras₩Prefabs



- 동적움직임에 따라 캐릭터를 추적하도록 Free Look Cam 스크립트에서 추적 대상을 지정한다.
- 메인 카메라는 해제

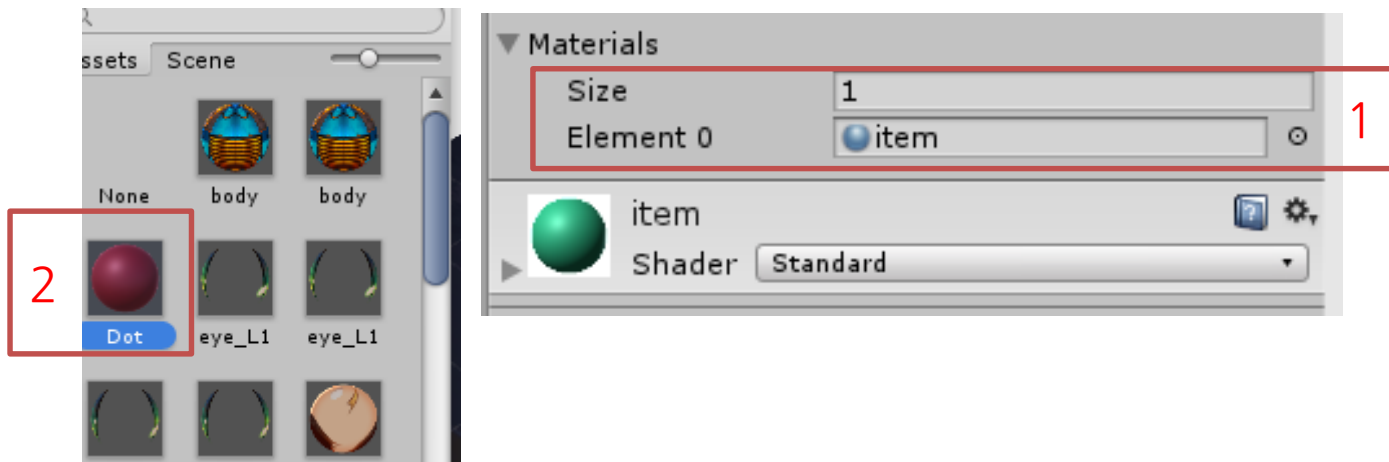


3. 아이템 만들기

플레이어가 게임을 하면서 획득할 아이템 제작

1) 아이템 생성 및 매트리얼 적용

- Dot 오브젝트를 선택하고 Inspector항목에서 Mesh Renderer - Materials - Element 0을 만들어 놓은 item로 지정 (매트리얼을 아이템에 드래그)

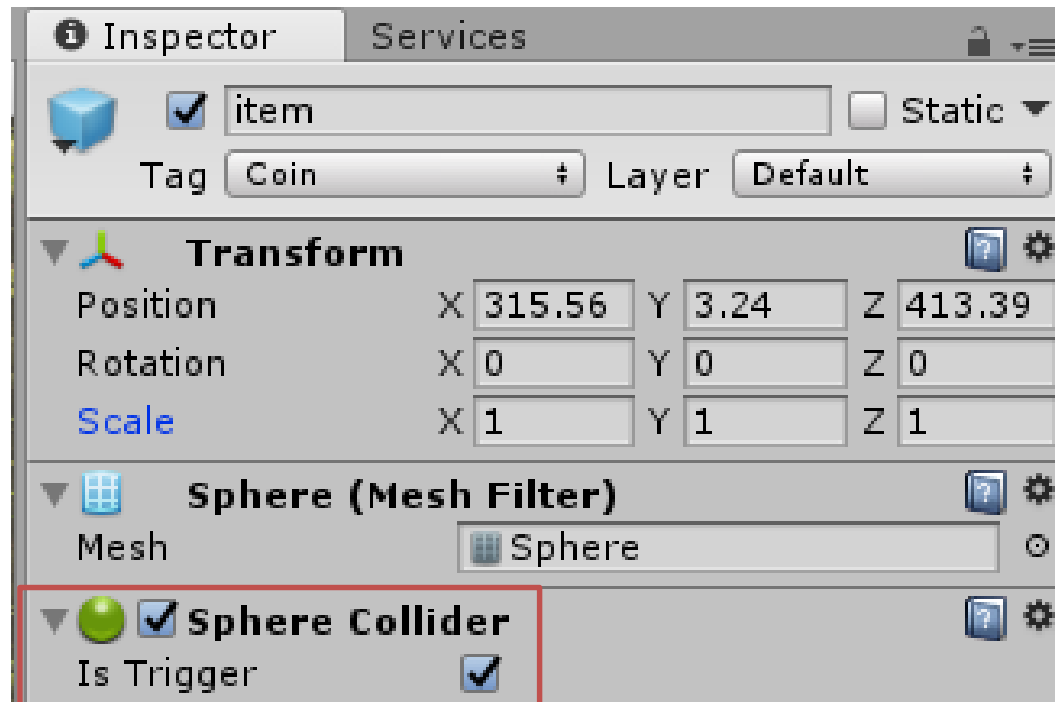


4. 플레이어가 아이템을 획득하도록 구현

아이템 객체도 하나의 장애물이기 때문에 플레이어의 진로를 막지 않도록 함

1) Collider를 Trigger 처리하도록 지정하기

- Dot 오브젝트의 Sphere Collider에서 Is Trigger 항목을 체크한다.
 - 앞으로 많이 사용되는 item객체와 같은 아이템은 프리팹으로 저장해 두면 생성하기 편리하다.



4. 플레이어가 동전을 획득하도록 구현

2) 동전을 획득하도록 보이기(코딩)

- 플레이어가 트리거 동전과 충돌하면 동전을 씬에서 보이지 않도록 소멸되도록 구현

```
void OnTriggerEnter (Collider hit) {
    Destroy(hit.gameObject);
}
```

// 게임오브젝트가 트리거와 충돌하면 OnTriggerEnter()가 호출되고
 // Destroy()의 함수에 의해 충돌된 객체가 소멸된다.

5. 태그로 씌 안에 있는 Coin 아이템 개수 카운트

월드에서 남아 있는 아이템의 개수를 세어 게임 점수를 매김

1) 태그 설정

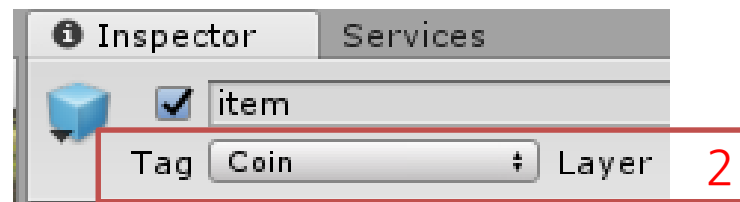
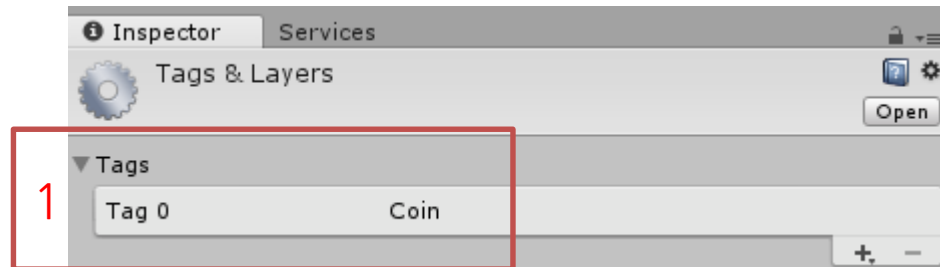
- Edit - Project Setting - Tags and Layers 클릭하여 Inspector에 표시

2) 태그 추가

- Tags에 새로운 Tag 0항목 추가
- New Tag를 Coin으로 수정

3) Prefabs에 태그 지정

- 생성된 태그 적용



5. 태그로 씬 안에 있는 Coin 아이템 개수 카운트

4) 승부 판정하기

- 씬에 남은 동전 아이템의 개수를 세어 게임의 승부를 판정하는 스크립터 작성 추가

```

using UnityEngine.SceneManagement; // 추가
void OnTriggerEnter (Collider hit) {
    Destroy(hit.gameObject);
    if (GameObject.FindGameObjectsWithTag("Coin").Length == 0)
    {
        SceneManager.LoadScene("main");
    }
}
    
```

5) 게임 실행

- 5개의 아이템을 추가해서 동작 시켜 보자

응용 실습

- 1 특정 지역을 지나가면 비바람(파티클 활용)이 몰아 치도록 월드를 구성해 봅시다.

```
public GameObject Prefab;  
// Use this for initialization
```

```
void OnTriggerEnter (Collider hit)  
{  
Instantiate (Prefab, transform.position, transform.rotation);  
}
```

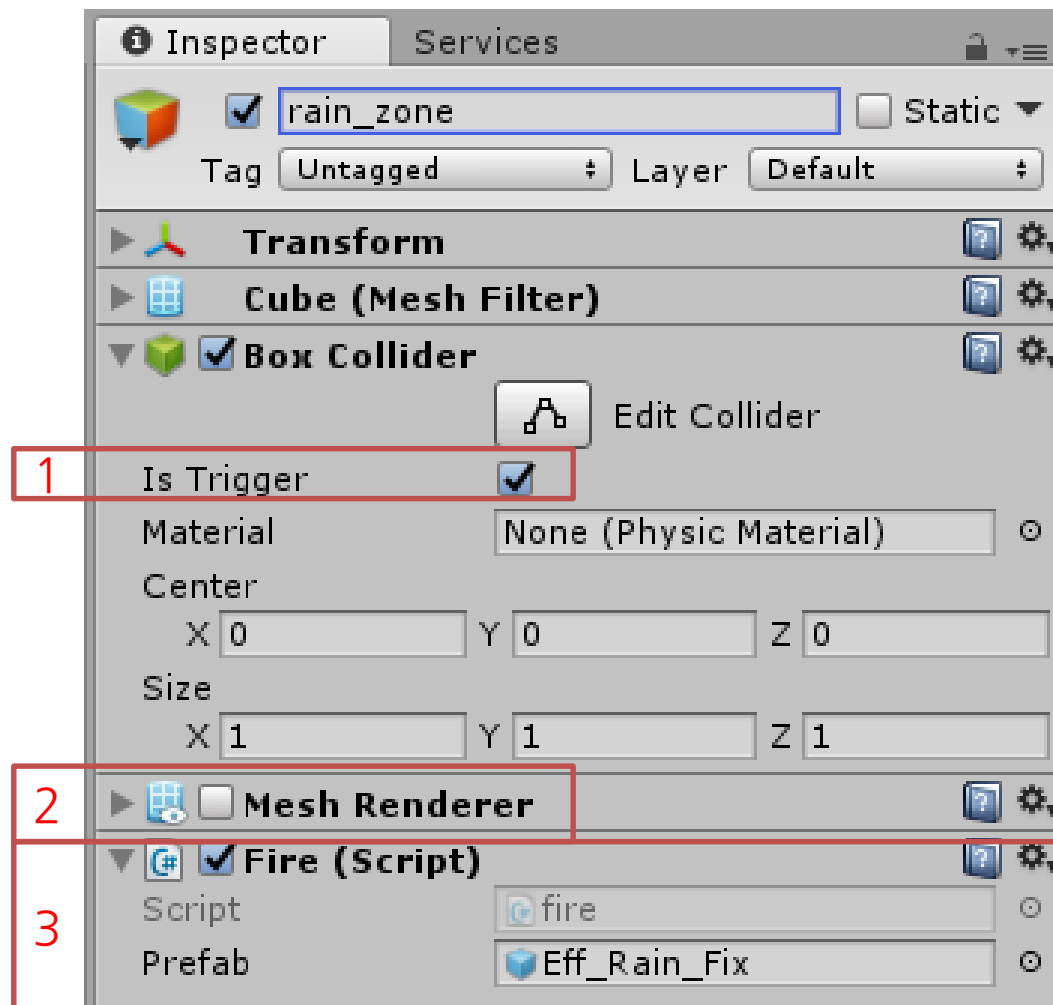
- 2 무인도에 숨겨진 아이템(드래곤볼) 7개를 모두 획득한 경우, 분하 구에서 불꽃(파티클 처리)이 피어나도록 구성해 봅시다.

- 3 무인도에 숨겨진 7개 아이템(드래곤볼)을 하나씩 획득할 때마다 남은 아이템의 개수가 화면에 보여지도록 GUI를 구성해 봅시다

응용 실습

1

특정 지역을 지나가면 비바람이 몰아 치도록 월드를 구성해 봅시다.

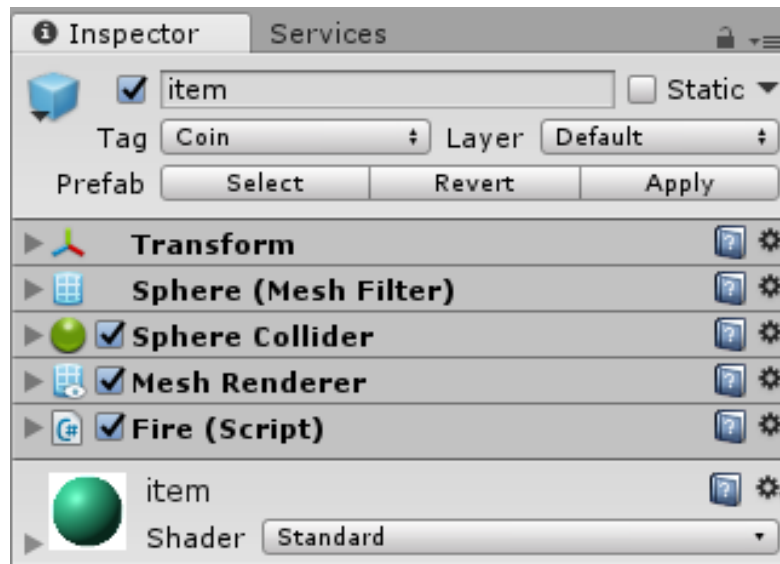


응용 실습

2

무인도에 숨겨진 아이템(드래곤볼) 7개를 모두 획득한 경우, 분하 구에서 불꽃(파티클 처리)이 피어나도록 구성해 보시다.

```
public GameObject Prefab;  
// Use this for initialization  
void OnTriggerEnter (Collider hit)  
{  
    Instantiate (Prefab, transform.position, transform.rotation);  
}
```



응용 실습

3

무인도에 숨겨진 7개 아이템(드래곤볼)을 하나씩 획득할 때마다 남은 아이템의 개수가 화면에 보여지도록 Text UI를 완성해 보시다



응용 실습

3

무인도에 숨겨진 7개 아이템(드래곤볼)을 하나씩 획득할 때마다 남은 아이템의 개수가 화면에 보여지도록 GUI를 구성해 봅시다

```
using UnityEngine.UI; //추가
```

```
public class itemText : MonoBehaviour {  
    public Text mText;  
    public int coin;  
  
    void Start () {  
        mText = GetComponent<Text> ();  
    }  
  
    // Update is called once per frame  
    void Update () {  
        coin = GameObject.FindGameObjectsWithTag ("Coin").Length;  
        mText.text = "남은 아이템: " + coin.ToString ("0");  
    }  
}
```

응용 실습

4

Query-chan을 모델로 활용하기

