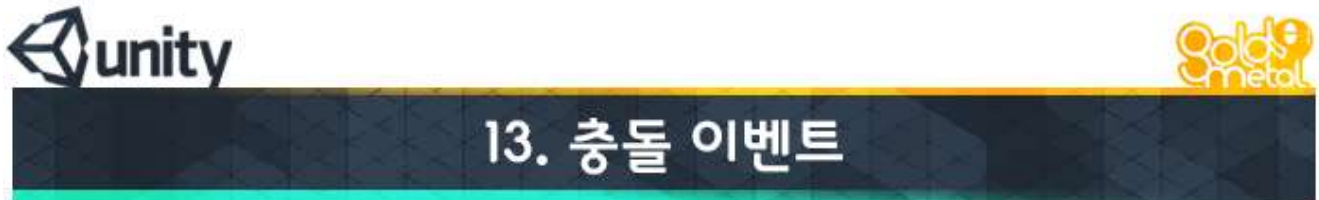


안녕하세요.  
골드메탈입니다.

오늘은 유니티에서 중요한 기술 중 하나인  
충돌 이벤트에 대해서 살펴보겠습니다.



충돌을 위해 필자는  
발판 버튼을 하나 만들어 놓았습니다.  
물론 이벤트 로직을 담은 스크립트도 장만해줍니다.

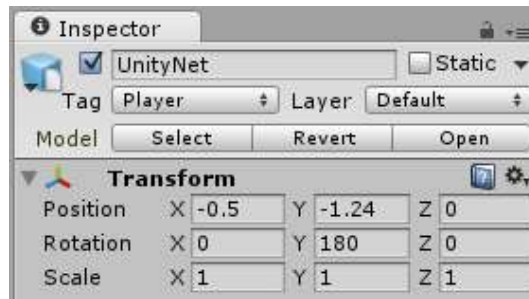
```
public class Button_Event : MonoBehaviour {
    GameObject player;

    void Awake ()
    {
    }
}
```

12편에서는 public 전역변수로 외부 오브젝트를 가져왔지요.  
이번에는 **태그(Tag)**를 통하여 불러올 것입니다.

```
void Awake ()
{
    player = GameObject.FindGameObjectWithTag ("Player");
}
```

여기에 GameObject 클래스의 **FindGameObjectWithTag()** 함수를 사용합니다.  
매개변수로 태그 이름을 넣어주면 되는데,  
그럼 태그는 어떻게 설정해야 할까요?



바로 속성 창 맨 위쪽에 있습니다.  
태그는 기본적으로 제공하는 것도 있지만  
직접 만들어 쓸 수도 있지요.

이제 이벤트 로직을 만들어 보겠습니다.

```
void OnTriggerEnter(Collider other)
{
}

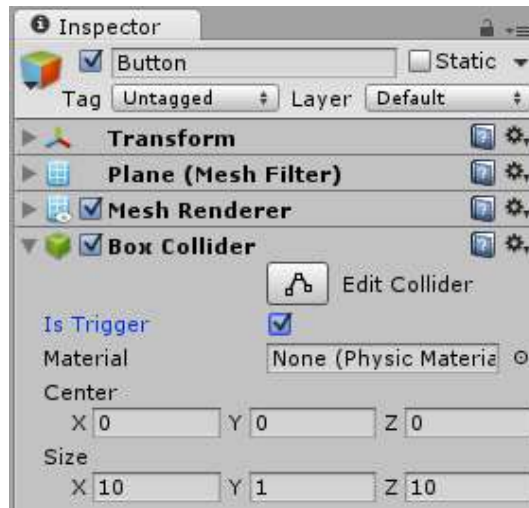
void OnTriggerExit(Collider other)
{
}
```

충돌 이벤트는 두 개의 충돌체(Collider)가  
만나거나 헤어질 때 발생합니다.

만날 땐, **OnTriggerEnter**  
헤어질 땐, **OnTriggerExit**



이 함수들을 사용해서  
타격하여 피해를 준다던가  
어느 구역에 들어가면 함정이 발동된다던가  
많은 것들을 만들 수 있습니다.



이 함수를 사용하기 위해선  
Is Trigger 가 활성화된 충돌체 컴포넌트가 하나라도 있어야 합니다.

그럼 물리효과까지 사용하는 오브젝트는  
충돌체가 두 개 이상이 되겠지요?

```
void OnTriggerEnter(Collider other)
{
    if (other.gameObject == player){
        Vector3 originPoint = new Vector3();
        originPoint.x = 0f;
        originPoint.y = 10f;
        originPoint.z = 0f;

        player.transform.position = originPoint;
    }
}
```

이 이벤트는 플레이어만 반응하도록 만들 것입니다.  
그러기 위해선 확인 먼저 해야겠지요.

충돌체의 게임 오브젝트를 태그로 가져왔던 플레이어 변수와 일치하는지 확인하고  
그 다음 알맞은 이벤트 로직을 넣어주면 되겠습니다.



발판을 밟는 순간 이벤트가 발생하는 것을 볼 수 있습니다.  
게임 안에서는 이런 방식의 수 많은 이벤트를 가지고 있습니다.

```
public class Button_Event : MonoBehaviour {  
  
    GameObject player;  
  
    void Awake ()  
    {  
        player = GameObject.FindGameObjectWithTag ("Player");  
    }  
  
    void OnTriggerEnter(Collider other)  
    {  
        if (other.gameObject == player){  
            Vector3 originPoint = new Vector3();  
            originPoint.x = 0f;  
            originPoint.y = 10f;  
            originPoint.z = 0f;  
  
            player.transform.position = originPoint;  
        }  
    }  
  
    void OnTriggerExit(Collider other)  
    {  
    }  
}
```

-----  
이번 편의 스크립트를 주욱 살펴보고  
열세번째 강좌 마무리 하겠습니다.

다음 시간에는 프리팸에 대해 다뤄보겠습니다.  
그럼.