[유니티 기초] 16. 점수 시스템

유니티·개발

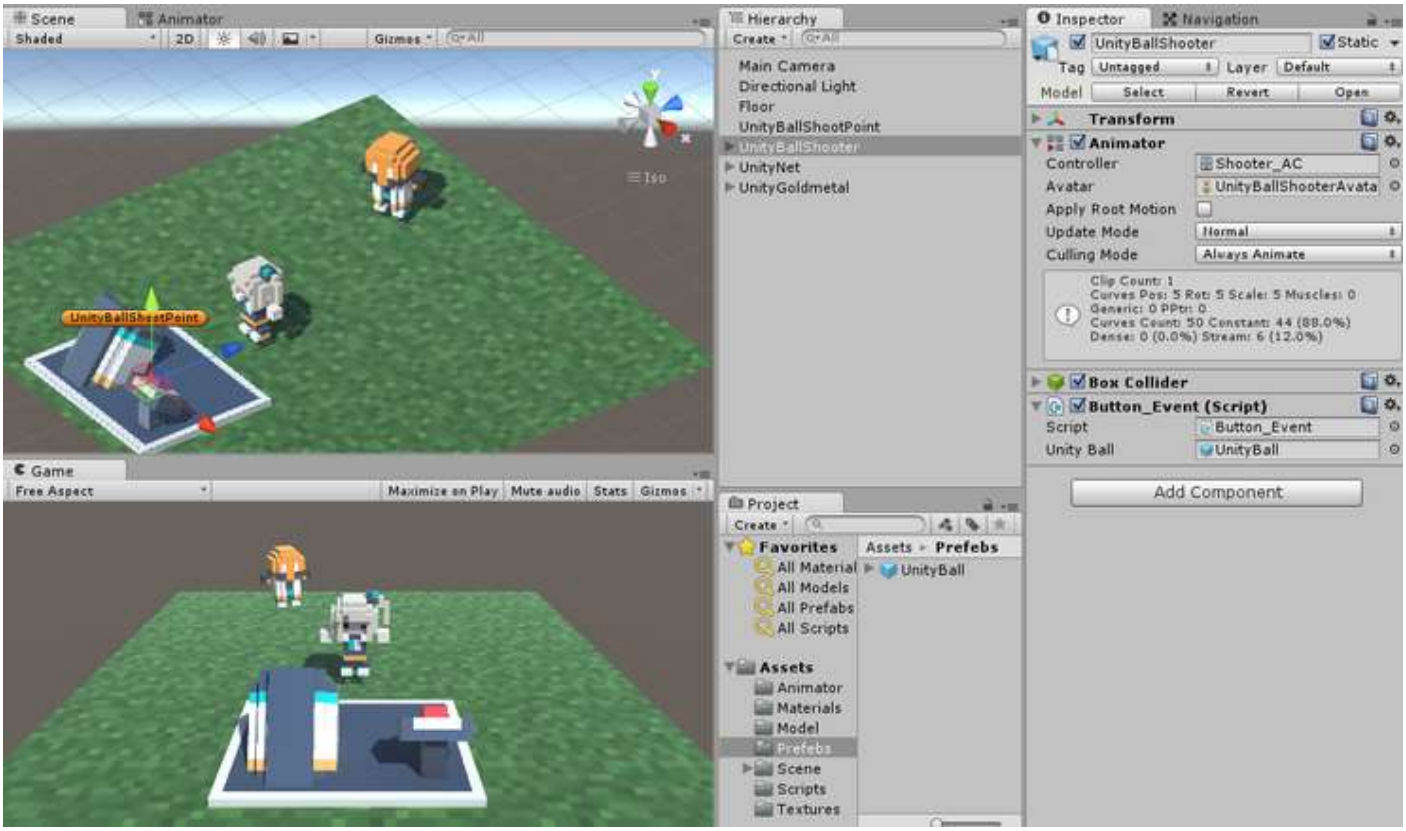
2015. 10. 29. 21:27

http://blog.naver.com/gold_metal/220523485312

안녕하세요.
골드메탈입니다.

이번 편은 정적 변수와 UI를 사용한
점수 시스템을 만들어보도록 하겠습니다.

원활한 강좌를 위해 15편에서 다뤘던 내비 컴포넌트를
비활성화 해놓고 시작하겠습니다.



플레이어가 발사한 공으로
NPC을 맞추면 점수를 얻는 시스템을
만들어보려 합니다.

```
void OnTriggerEnter (Collider other)
{
    if (other.gameObject.tag == "Ball" && other.isTrigger == false){
        status = "Damaged";
        onDamage();
    }
}

//-----------------------------------------------------[ Animation Function ]

void AnimationUpdate ()
{
    if(status == "Damaged"){
        animator.SetTrigger("onDamage");
        return;
    }

    animator.SetBool("isWalking",false);
}

//-----------------------------------------------------[ Damage Function ]

void onDamage ()
{
    Vector3 knockbackDegree = new Vector3(0f,1f,1f);
    rigidbody.AddForce(knockbackDegree * 350f, ForceMode.Impulse);
}
```

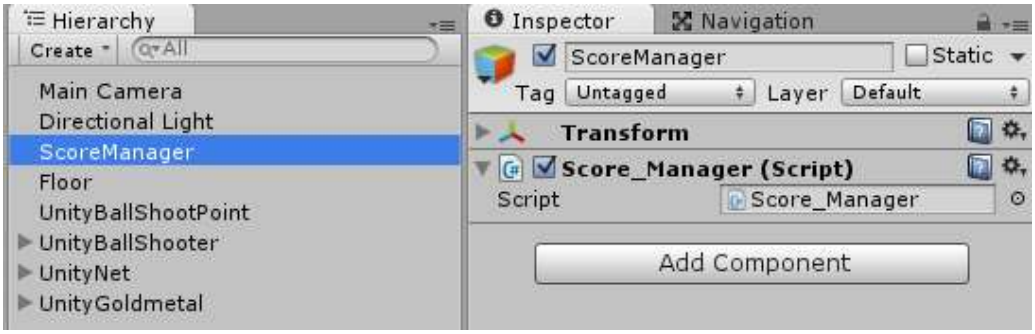
NPC 오브젝트에 트리거 충돌체 하나 만들고
위처럼 스크립트를 만들어줍니다.

OnTriggerEnter
트리거 충돌체에 접촉한 다른 충돌체를 태그와 타입으로 확인합니다.
지난 편을 보면 공에 두개의 트리거가 있는데
isTrigger를 통해 안쪽의 물리효과를 주는 충돌체만 인식하겠다는 것이지요.

AnimationUpdate
트리거를 통해 얻은 status를 확인하여
애니메이션을 실행합니다.

onDamage
공과 충돌할 때, 뒤로 넉백되는 것을
AddForce로 간단하게 구현하였습니다.

이렇게 셋팅된 상태로,
점수를 관리하는 오브젝트를 만듭니다.



빈 오브젝트 하나 만들고
매니저 스크립트를 새로 작성합니다.

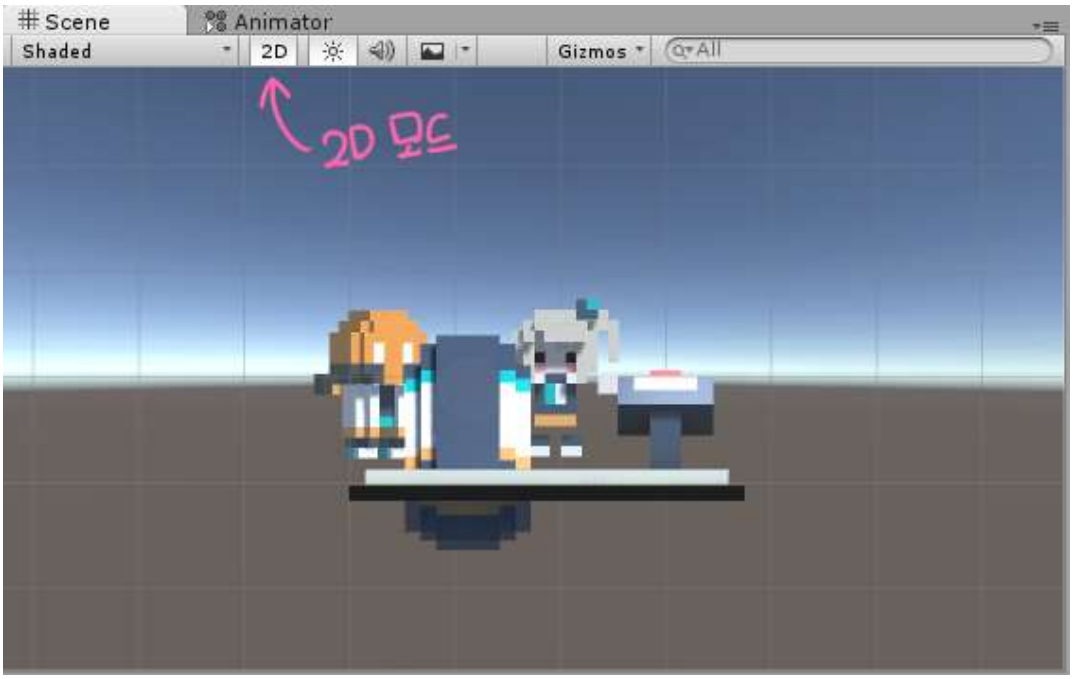
```
public class Score_Manager : MonoBehaviour {

    public static int score;

    void Awake ()
    {
        score = 0;
    }
}
```

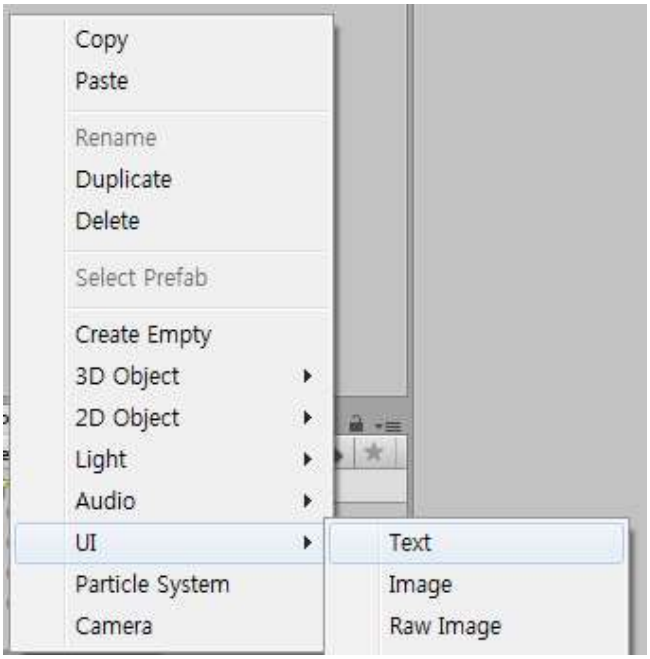
점수를 저장한 변수를 선언해주는데
static 키워드 (정적) 까지 넣어줍시다.
static이 뭐길래 쓰는걸까요?

이유는 나중에 알려드리겠습니다.

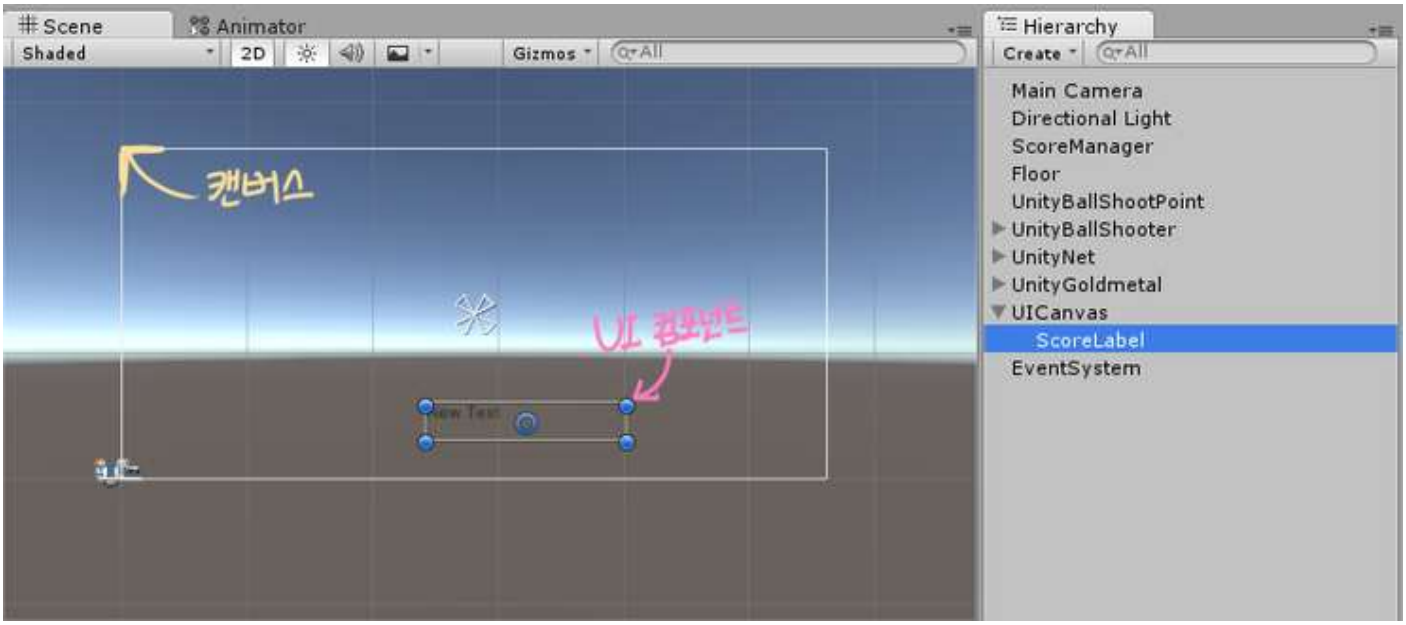


매니저가 완성되었으니
화면에 표시하기 위해 UI를 만들 차례입니다.

Scene 창에서 2D 버튼을 눌러줍시다.

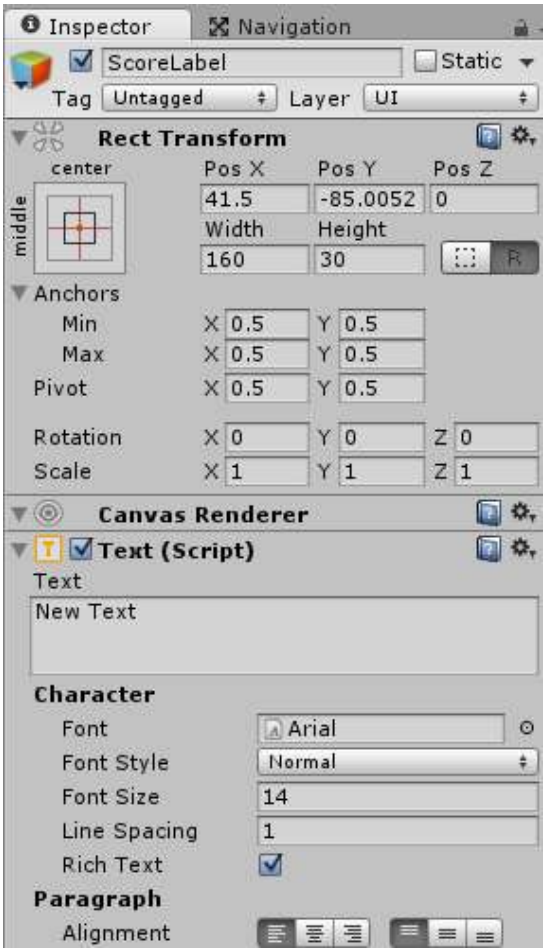


그러고나서 **UI - Text** 메뉴로
텍스트 UI를 추가해보죠.



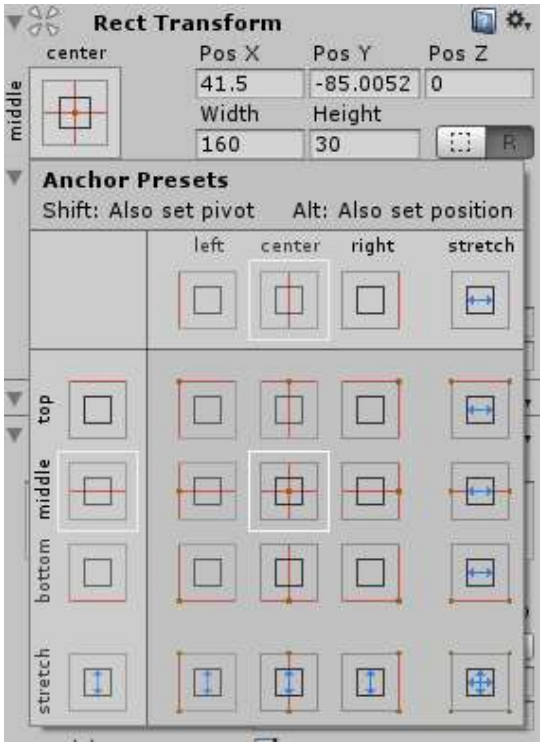
짜라~
UI는 이런 모습으로 작업을 합니다.

일단 도화지 격인 **Canvas**가 있고,
그 안에 UI 컴포넌트들을 담아 넣는 형식입니다.

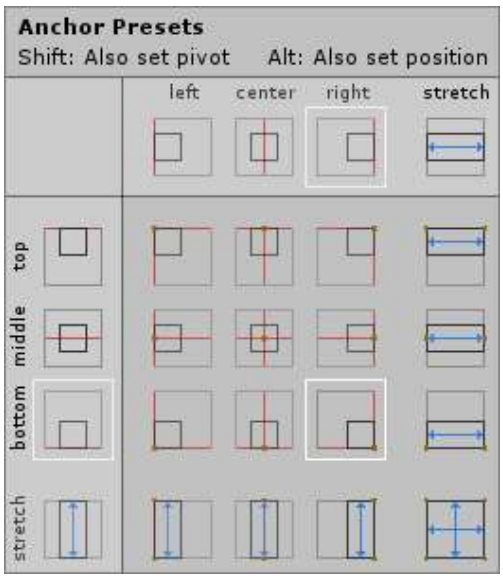


이번 편은 Text 컴포넌트만 보겠습니다.

위쪽엔 위치와 넓이, 기준점이 있고,
아래쪽에는 텍스트 값과 폰트를 설정할 수 있습니다.



UI는 위치가 중요하죠.
왼쪽 사각형을 클릭하면 위치 설정이 나옵니다.
하지만 평상시는 기준점만 옮겨집니다.



여기서 Alt 키를 누르면
UI까지 함께 이동하는 방식으로 바뀝니다.
필자는 오른쪽 아래에 놓도록 하겠습니다.

어느정도 설정했다면, 이 텍스트 컴포넌트에
스크립트를 새로 만들어줍니다.

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class Score_Update : MonoBehaviour {

    Text scoreLabel;

    void Awake ()
    {
        scoreLabel = GetComponent<Text>();
    }

    void Update ()
    {

    }
}
```

꼭 넣어야 UI 선언 가능

이전에 한 것처럼
컴포넌트를 변수에 넣은 다음,

```
void Update ()
{
    scoreLabel.text = Score_Manager.
}
```

score
UnityEngine.MonoBehaviour
print
UnityEngine.Object
Destroy
DestroyImmediate

public static int
score

텍스트 값에 점수를 넣어줍니다.

여기서 중요한 것은
점수 매니저를 따로 변수로 선언하지 않고 불러서
static으로 선언했던 변수에 바로 접근한다는 점입니다.

점수나 라이프 등
게임 상에서 글로벌하게 사용되는 요소는
이렇게 매니저를 두어 static으로 관리할 수 있습니다.

```
void Update ()
{
    scoreLabel.text = Score_Manager.score.ToString();
}
```

텍스트 값은 오직 문자열만 취급하기 때문에
ToString으로 형변환해주었습니다.

```
//-----------------------------------------------------[ Damage Function ]

void onDamage ()
{
    Vector3 knockbackDegree = new Vector3(0f,1f,1f);
    rigidbody.AddForce(knockbackDegree * 350f, ForceMode.Impulse);

    Invoke("onDeath",0.5f);
}

void onDeath ()
{
    Score_Manager.score += 10;
    Destroy(gameObject, 0f);
}
```

다시 NPC 스크립트로 돌아와서
공으로 맞출 때 점수를 얻는 로직을 만들어줍니다.

그리고 해당 캐릭터가 죽었다고 가정하고,
Destory(GameObject, Delay) 함수로 장면에서 제거해버립시다.

이처럼 공을 맞은 후
뽕 사라지면서 점수를 얻는 장면이 되었습니다.

이렇게하여
열여섯번째 강좌를 마무리하였습니다.

다음은 플레이어가 아이템을 줍고 놓는
모션에 대해 다뤄보도록 하겠습니다.

그럼.