3D콘텐츠 이론 및 활용

12주(1). Sound 오브젝트

- Audio
- 3D Audio

학습개요

학습목표

- 사운드 처리를 위한 설정을 이해하고 활용할 수 있다.
- 3D 사운드 입체감을 위한 설정과정을 이해하고 활용할 수 있다.

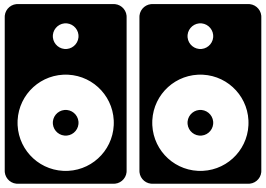
학습내용

- Audio Source 컴포넌트
- Audio Clip
- 3D Sound 처리

1. Sound

1) 사운드 오브젝트

- 오디오 리스너(Audio Listener) - 헤드셋
- 오디오 소스(Audio Source) - 플레이어
- 오디오 클립(Audio Clip) - 음악파일



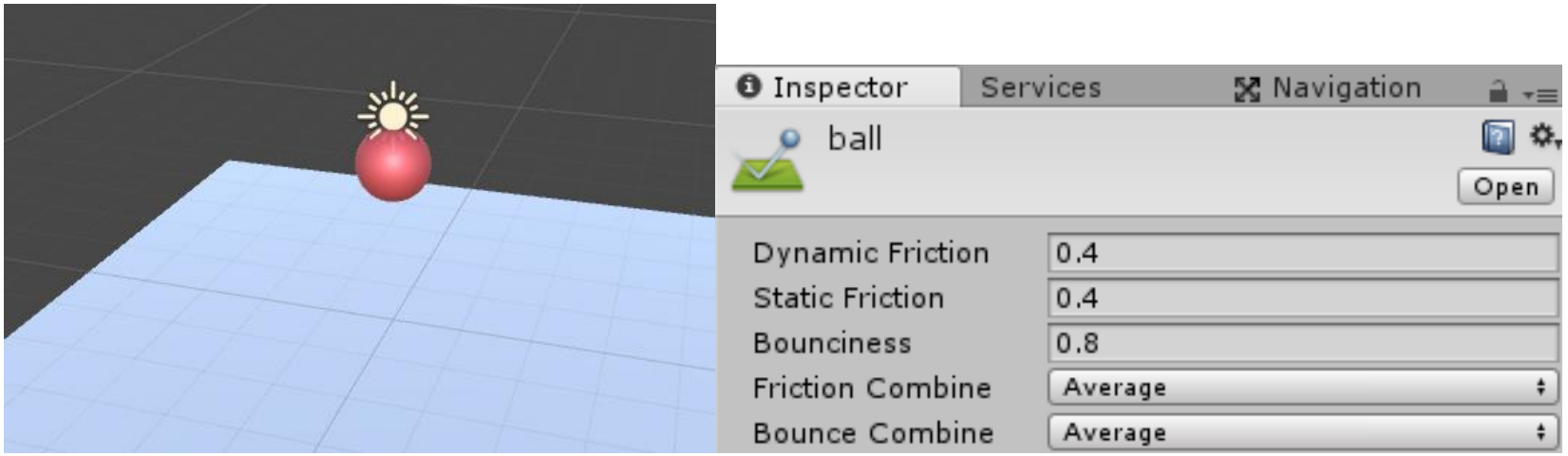
2. Play 메서드

1) Play ()

- 오디오 소스 컴포넌트에 오디오 클립을 이용해 사운드를 출력하기 위한 메서드

2) 실습 환경 설정

- 떨어지는 공에 물리효과와 사운드를 넣기 위한 준비



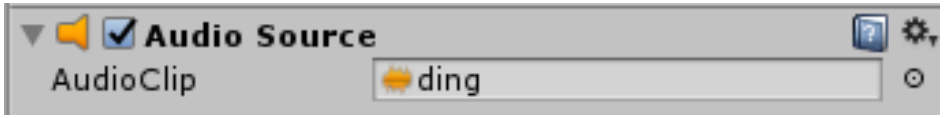
2. Play 메서드

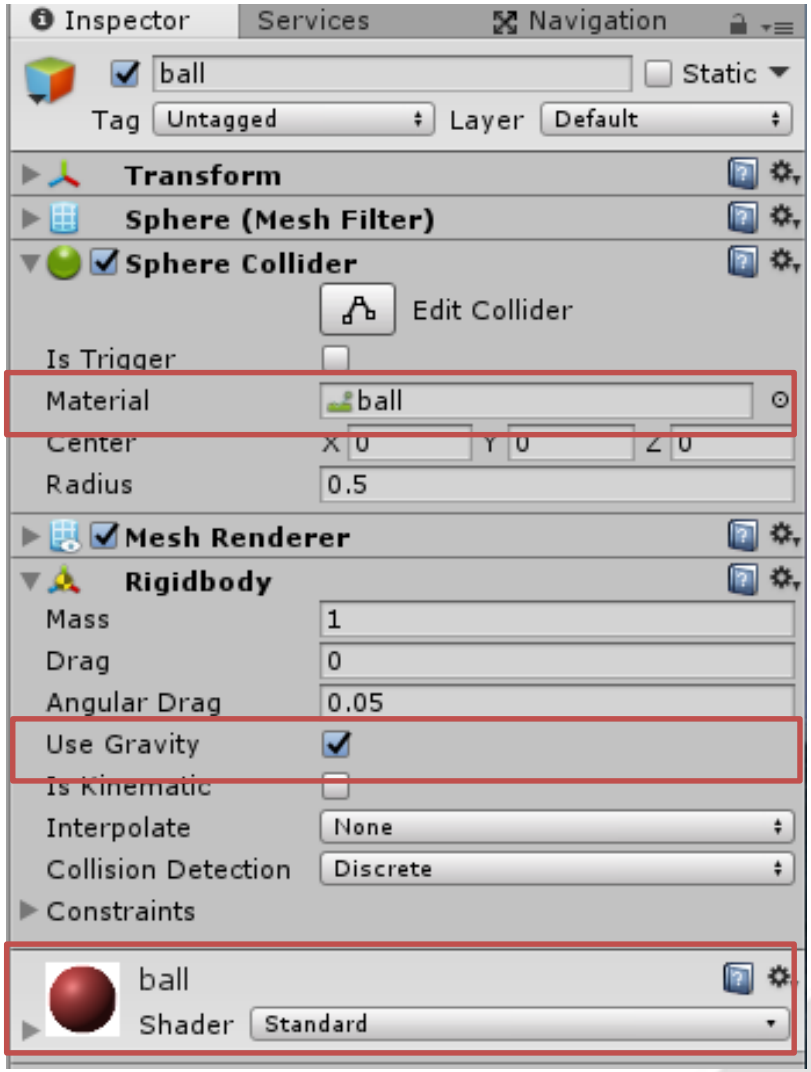
3) 사운드 적용하기

- 볼 인스펙터 뷰 설정 확인

4) Audio Source 컴포넌트 추가

- 오디오 소스 추가
- 오디오 클립 연결





2. Play 메서드

5) 공이 바닥에 닿을 때 사운드 출력 스크립트 작성

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

public class SoundPlay : MonoBehaviour {
    // Use this for initialization
    void Start () {
    }

    // Update is called once
    void Update () {
    }
    void OnCollisionEnter(Collision hit)
    {
        GetComponent<AudioSource> ().Play ();
    }
}
```

3. PlayOneShot 메서드

1) PlayOneShot ()

- 오디오 소스 컴포넌트 외부에 있는 오디오 클립을 출력하는 용도

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

public class PlayOneShot : MonoBehaviour {
    public AudioClip clip;
    void OnCollisionEnter (Collision hit)
    {
        //GetComponent<AudioSource> ().Play ();
        GetComponent<AudioSource> ().PlayOneShot(clip, 1f);
    }
}
```

4. Destroy 처리

1) Destroy 처리의 문제점

- 공이 낙하 후 바닥과 충돌하면 소리를 한 번 재생하고 자신의 게임오브젝트를 제거하도록 함.

```
void OnCollisionEnter(Collision hit)
{
    GetComponent<AudioSource> ().Play ();
    Destroy (this.gameObject);
}
```

위와 같이 코딩 후 결과를 보면, 공이 낙하 후 바닥과 충돌하면서 사운드를 출력하고 자신의 게임오브젝트가 제거되어야 하지만 사운드 출력 시간 부족으로 소리가 들리지 않게 됨.

4. Destroy 처리

2) Destroy 해결책

- 사운드 재생이 완료되기를 기다린 후 게임오브젝트가 제거 되도록 함.

```
public class SoundPlay : MonoBehaviour {
    private AudioSource myAudio;
    // Use this for initialization
    void Start () {
        myAudio = GetComponent<AudioSource> ();
    }

    void OnCollisionEnter(Collision hit)
    {
        myAudio.Play ();
        Destroy (this.gameObject, myAudio.clip.length);
    }
}
```

Destroy (this.gameObject, 2); // 2초 뒤 소멸처리
myAudio.clip.length 속성 // 오디오 클립의 재생시간

참고

1) Static int account = 0;

프로그램내에서 공유할 수 없는 변수는 Static으로 선언하고 사용함.

2) Static MyClass _myclass;

- 프로그램내에서 공유할 수 없는 인스턴스 객체를 하나만 만들어서 사용해야 하는 경우 싱글톤으로 클래스를 제작함.
- Static [클래스명]으로 선언하고 사용하며, 허가된 함수를 통해 접근이 가능함.

5. AudioManager

1) 사운드의 싱글톤(singleton) 처리

- 싱글톤 처리를 위한 AudioManager 작성

```
public class AudioManager : MonoBehaviour {
    static AudioManager _instance = null;
    public static AudioManager Instance ()
    {
        return _instance;
    }
    // Use this for initialization
    void Start () {
        if (_instance == null)
            _instance = this;
    }
    public void PlaySfx(AudioClip clip)
    {
        GetComponent<AudioSource> ().PlayOneShot (clip);
    }
}
```

5. AudioManager

2) 싱글톤을 이용한 사운드 재생

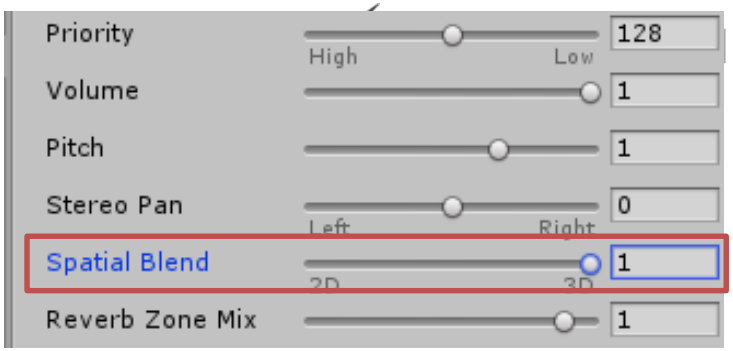
- csAudioManager.cs

```
public class csAudioManager : MonoBehaviour {
    public AudioClip clip;

    void OnCollisionEnter(Collision hit)
    {
        AudioManager.Instance ().PlaySfx (clip);
        Destroy (this.gameObject);
    }
}
```

6. 3D 사운드

1) 3D사운드 재생 실습



- csAudioManager.cs

```
public class cs3Dsound : MonoBehaviour {
    float speed = 10.0f;
    // Use this for initialization
    void Start () {
    }
    // Update is called once per frame
    void Update () {
        float v = Input.GetAxis("Vertical");

        v = v * speed * Time.deltaTime;
        transform.Translate (Vector3.forward * v);
        if (Input.GetButtonDown ("Fire1")) {
            GetComponent<AudioSource> ().Play ();
        }
    }
}
```

6. 3D 사운드

2) 거리에 따른3D사운드 음량설정

- Min Distance
  - 보통소리의 음량
- Max Distance
  - 작아지면서 들리는 거리