



# Unityはじめるよ ～すごいよサウンド機能～ 実践編

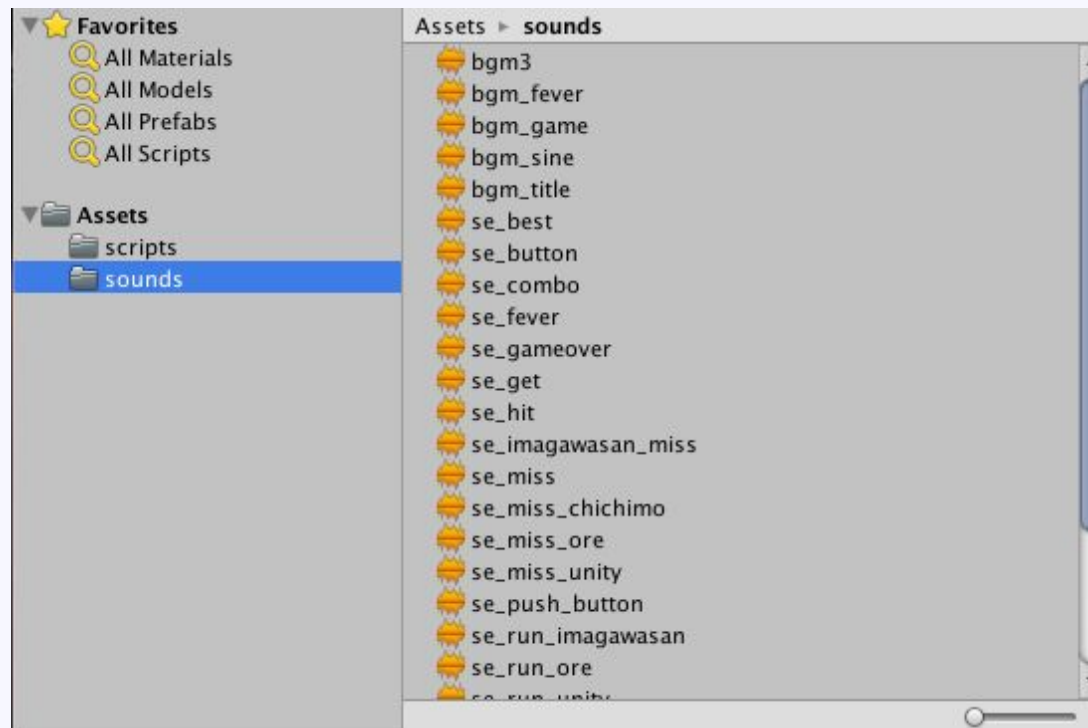
統合開発環境を内蔵したゲームエンジン  
<http://japan.unity3d.com/>

※いろんな職業の方が見る資料なので説明を簡単にしている部分があります。正確には本来の意味と違いますが上記理由のためです。ご了承ください。  
この資料内の一部の画像、一部の文章はUnity公式サイトから引用しています。

# AudioMixerのサンプルアプリを作る

## ■ 音素材のインポート

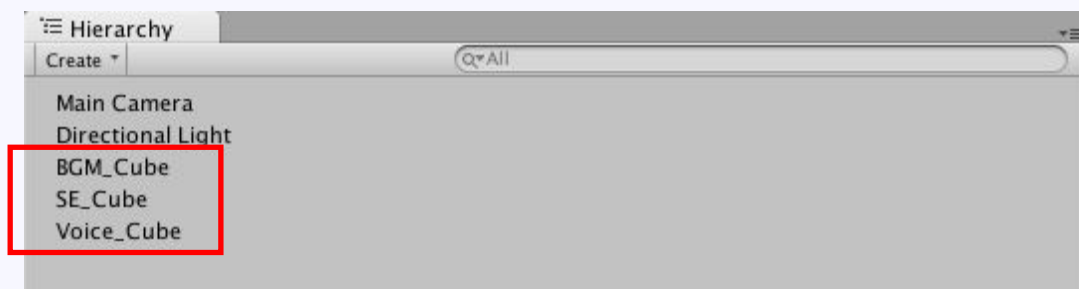
soundsフォルダを作成して、音ファイルをインポート



## ■ 音を鳴らすGameObjectの作成

BGM用  
SE用  
Voice用

の3つを用意。



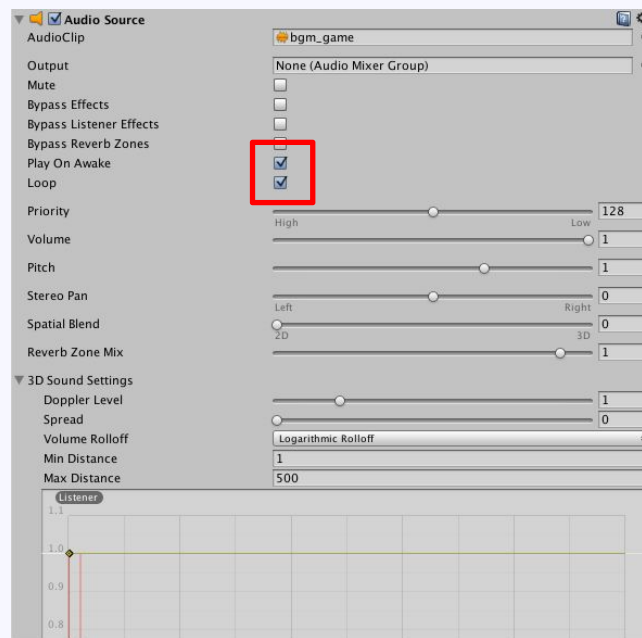
## ■ AudioSourceの追加

さっき作ったGameObjectに  
オーディオソースとオーディオクリップを追加



※BGMは  
自動再生開始用に  
「Play On Awake」  
フラグにチェック

ループ再生用に  
「Loop」  
フラグにチェック



### ■ 音を鳴らすためのスクリプトを作成

BGMは自動再生。

SEとボイスはキーボードのキーを押した時に音が鳴るようにする。  
なので、下記の二つのスクリプトを作成

SE再生用「SEPlayer」

ボイス再生用「VoicePlayer」

## SEPlayer.cs

```
using UnityEngine;
using System.Collections;

public class SEPlayer : MonoBehaviour {

    AudioSource mAudioSource; // オーディオソース
    AudioClip mAudioClip; // オーディオクリップ
    bool mKeyUp = true; // キー押しっぱなしで音が連続で鳴ってしまうのを防ぐフラグ

    void Start () {
        // オーディオソース取得
        mAudioSource = gameObject.GetComponent<AudioSource> ();

        // オーディオクリップ取得
        mAudioClip = mAudioSource.clip;
    }

    void Update () {

        // Sキーが押されたら音を鳴らす
        if (Input.GetKey (KeyCode.S)) {
            if (mKeyUp == true) {
                mAudioSource.PlayOneShot (mAudioClip);
                mKeyUp = false;
            }
        }
        else {
            mKeyUp = true;
        }
    }
}
```

## VoicePlayer.cs

```
using UnityEngine;
using System.Collections;

public class VoicePlayer : MonoBehaviour {

    AudioSource mAudioSource;          // オーディオソース

    void Start () {
        // オーディオソース取得
        mAudioSource = gameObject.GetComponent<AudioSource> ();
    }

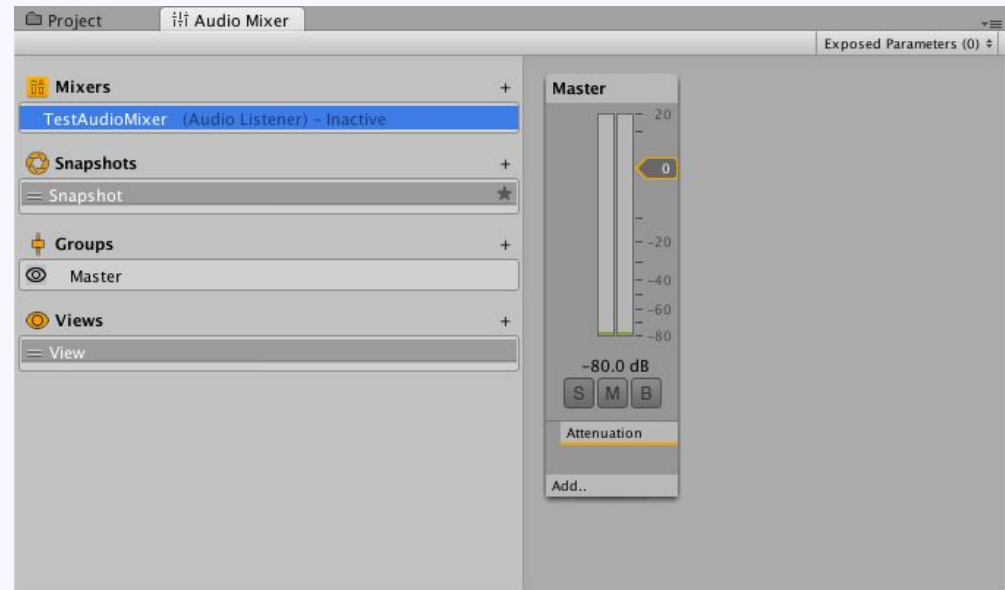
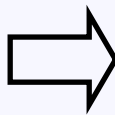
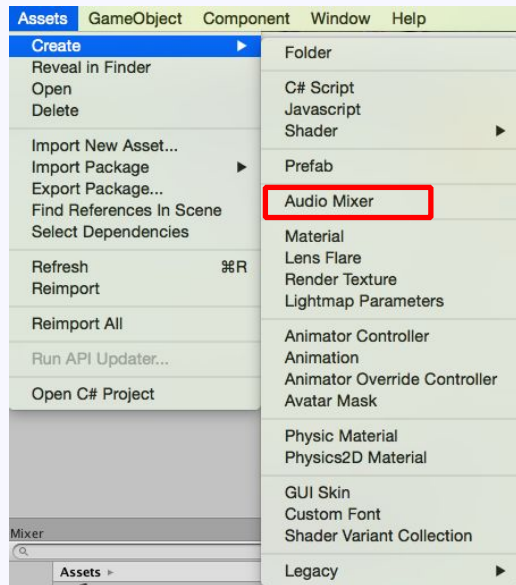
    void Update () {

        // Vキーが押されたら音を鳴らす
        if (Input.GetKey(KeyCode.V)) {
            if (mAudioSource.isPlaying == false) {
                mAudioSource.Play ();
            }
        }
    }
}
```



## ■ AudioMixerの作成

メニュー > Asset > Create > Audio Mixerで` AudioMixerを作成する。

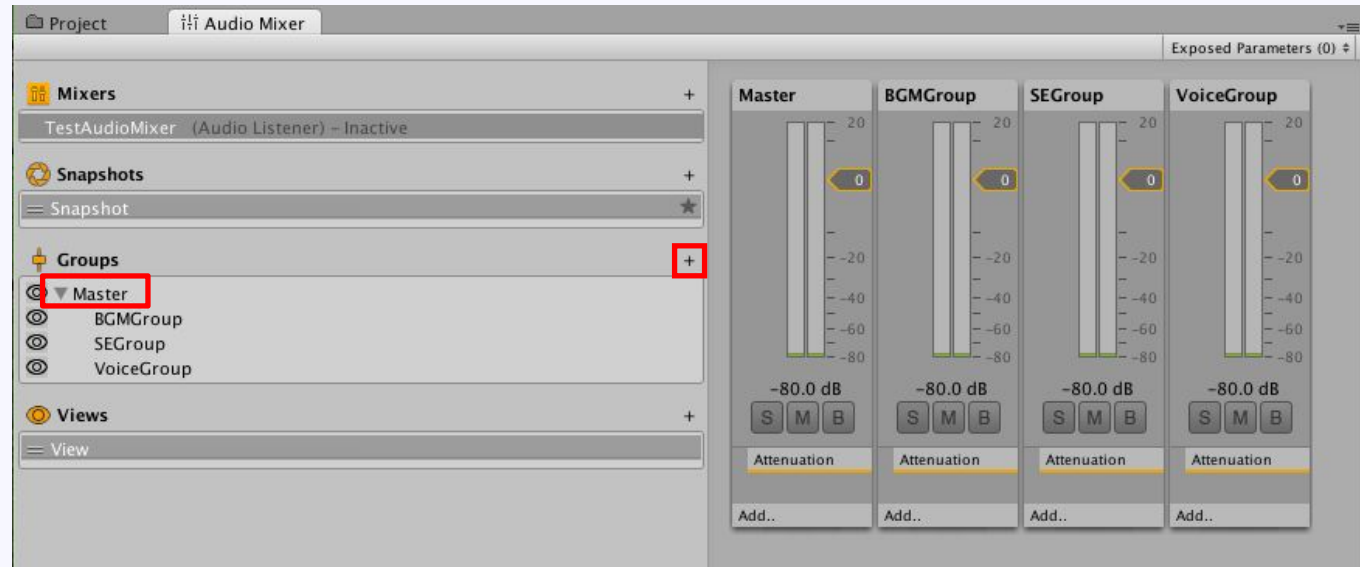


## ■ AudioGroupの作成

Masterを選択した状態で、Groupsの右上の「+」ボタンからGroupを作成。

BGM  
SE  
Voice

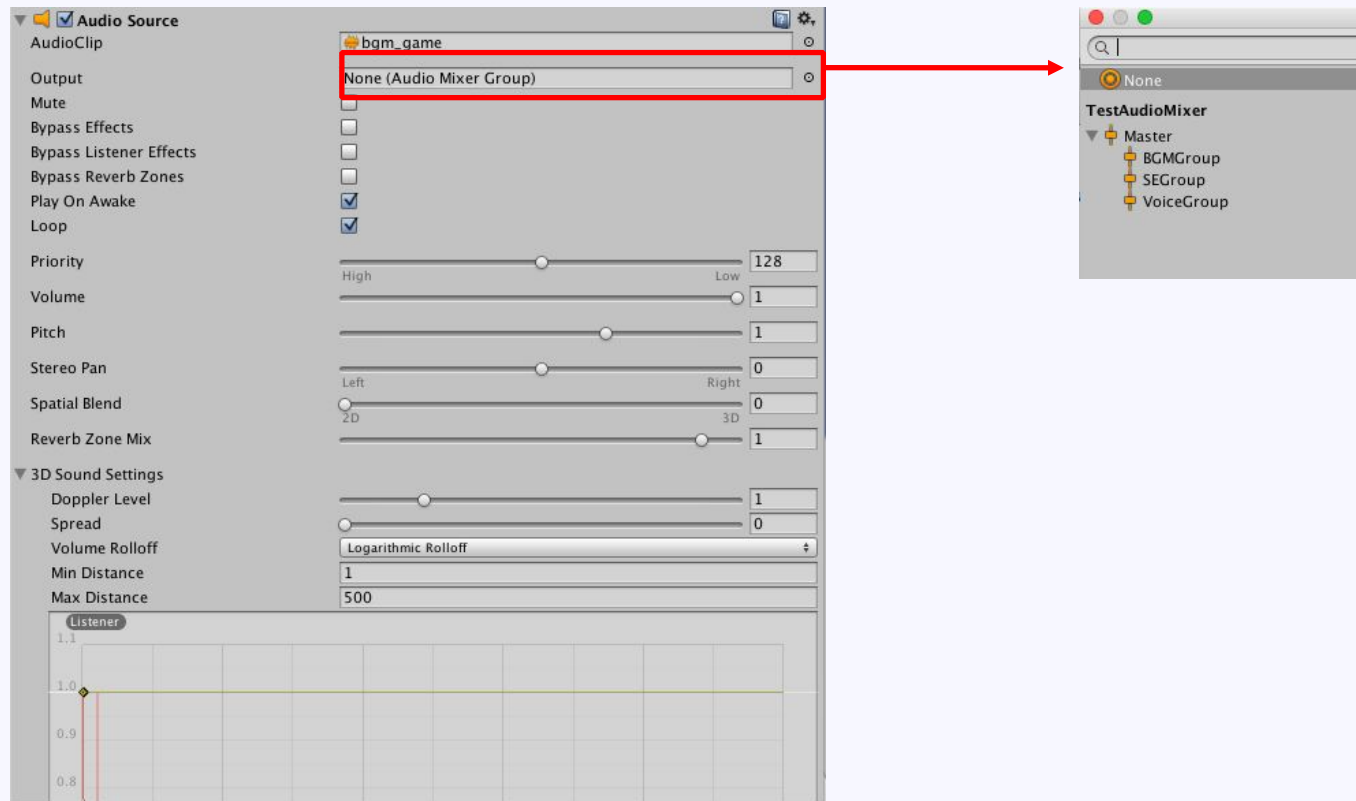
の3グループを作成する。



## ■ AudioSourceにグループを指定

各GameObjectのAudioSourceのOutputからグループを選択。

BGM、SE、Voiceそれぞれグループを指定しておく。

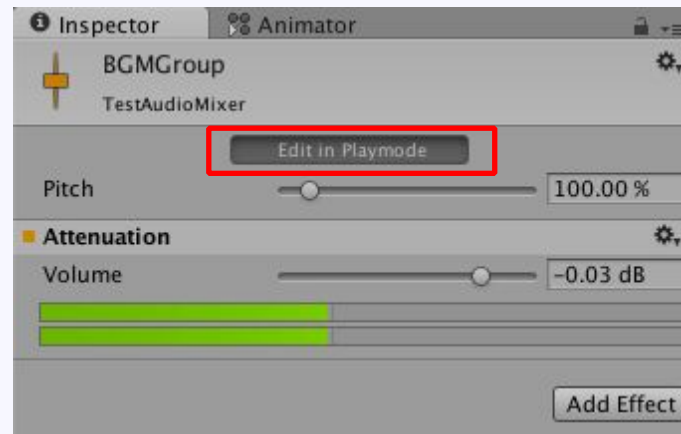


### ■ とりあえず一旦再生

EditInPlayModeボタンでリアルタイムに編集可能

※但し、ここでの編集はPlayをやめた状態でも生き残るので注意が必要

※Masterは全体のすべてに影響を与える



グループ名

ボリュームスライダー

ソロ再生ボタン  
このボタンが押されているグループのみ再生される

ミュートボタン  
このボタンが押されているグループはミュートとなる

バイパスボタン  
このボタンが押されているグループはエフェクトが無視される

エフェクト  
現在適用されているエフェクトの一覧  
Addボタンでエフェクトの追加をする

Master	Music	Reverb	Effects
Volume: -2	Volume: 0	Volume: 0	Volume: 0
S M B	S M B	S M B	S M B
Attenuation	Attenuation	Attenuation	Attenuation
Add..	Lowpass ●	Return	S: Reverb/Return
	Compressor ●	SFX Reverb ●	Add..
	Flange ●	Add..	
	Distortion ●		
	Echo ●		
	Highpass ●		
	S: Reverb/Return		
	Add..		

## ■スナップショットの作成

Snapshots右上の+ボタンからスナップショットを追加（図1）  
設定したいスナップショットを選択した状態で  
ミキサーをいじるだけで記録される（図2）

デフォルトとボイス用を作成し実行してみる

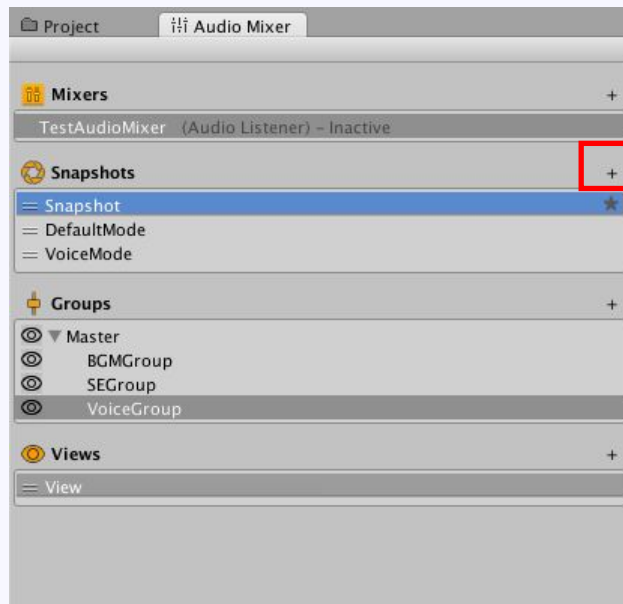


図1

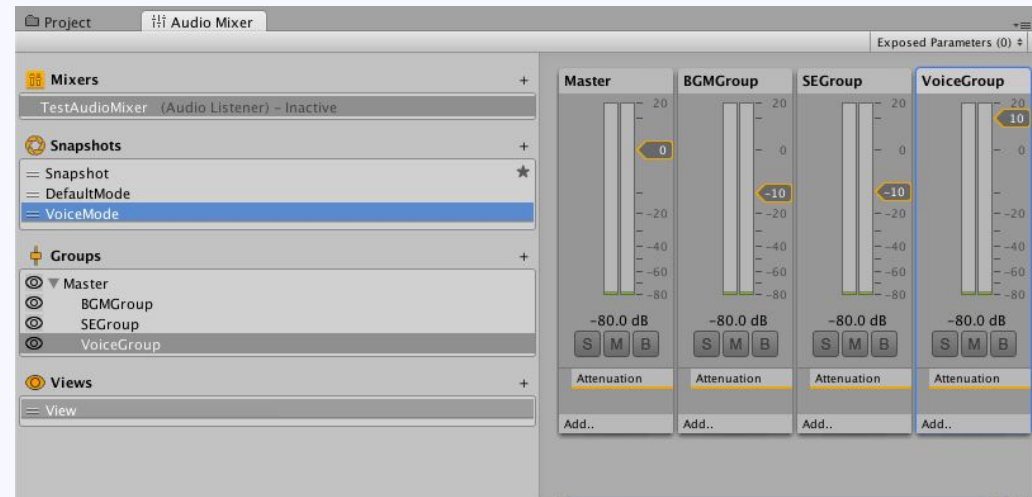


図2

### ■ スクリプトからスナップショットを制御

- ・ スナップショットの切り替え  
「TransitionToSnapshots」を使う。

#### TransitionToSnapshotsの説明

引数には「スナップショット」と「ターゲット値」と「変化にかける時間」を指定。  
スナップショットは配列で指定可能。

#### 例

```
AudioMixerSnapshot[] snapshots = new AudioMixerSnapshot[2] { DefaultModeSnapShot, VoiceModeSnapShot };  
float[] weights = new float[2] { 0.0f, 1.0f };  
Mixer.TransitionToSnapshots ( snapshots, weights, 2.0f );
```

上記の例は、  
デフォルトスナップショットを0に、  
ボイススナップショットを1に向けて、  
2秒かけて変更するコード。  
weightは0.0f~1.0fの値で指定する。  
0.5fとかにすれば、それぞれのスナップショットがミックスされた結果となる。

## SnapshotTester.cs

```
using UnityEngine;
using UnityEngine.Audio;
using System.Collections;

public class SnapshotTester : MonoBehaviour {

    public AudioMixer      Mixer;           // AudioMixer
    public AudioMixerSnapshot DefaultModeSnapshot; // 通常の状態のスナップショット
    public AudioMixerSnapshot VoiceModeSnapshot; // Voice再生用スナップショット

    bool mDefaultMode = true;           // デフォルトとボイスを交互に切り替えるためのフラグ
    bool mKeyUp = true;                 // キー押しっぱなしで反応してしまうのを防ぐフラグ

    void Start () {
    }

    void Update () {

        // Cキーが押されたらスナップショット切り替え
        if (Input.GetKey (KeyCode.C)) {
            if (mKeyUp == true) {
                // スナップショットを切り替え
                // デフォルトからヴォイスへ
                if (mDefaultMode == true) {
                    mDefaultMode = false;

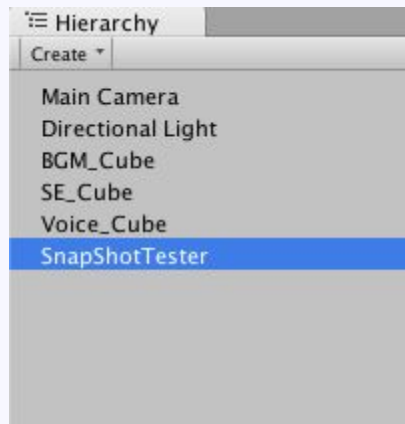
                    AudioMixerSnapshot[] snapshots = new AudioMixerSnapshot[2] { DefaultModeSnapshot, VoiceModeSnapshot };
                    float[] weights = new float[2] { 0.0f, 1.0f };
                    Mixer.TransitionToSnapshots ( snapshots, weights, 2.0f );
                }
                // ヴォイスからデフォルトへ
            } else {
                mDefaultMode = true;

                AudioMixerSnapshot[] snapshots = new AudioMixerSnapshot[2] { DefaultModeSnapshot, VoiceModeSnapshot };
                float[] weights = new float[2] { 1.0f, 0.0f };
                Mixer.TransitionToSnapshots ( snapshots, weights, 2.0f );
            }
            mKeyUp = false;
        }
    }
    else {
        mKeyUp = true;
    }
}
}
```

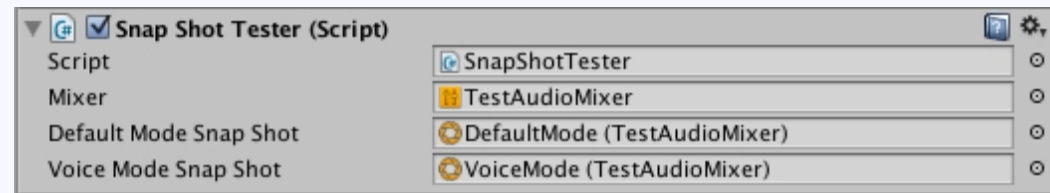


### ■ スクリプトをテスト

スクリプトテスト用のゲームオブジェクトを作成し、スクリプトを追加する。  
合わせて、スクリプトにAudioMixerとSnapShotを追加しておく。  
実行して動作を確認。



ゲームオブジェクトを作成し  
スクリプトを追加



スクリプトに、AudioMixerとスナップショットを追加

ご清聴ありがとうございました