using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class SoundManager : MonoBehaviour {

//Unity 자습서(튜토리얼)을 참고한 Script

//주소 >> https://unity3d.com/kr/learn/tutorials/projects/2d-roguelike-tutorial/audio-and-sound-manager?playlist=17150

//-------------------------------------------------------------------------

public AudioSource efxSource; //Drag a reference to the audio source which will play the sound effects.

public AudioSource musicSource; //Drag a reference to the audio source which will play the music.

public static SoundManager instance = null; //Allows other scripts to call functions from SoundManager.

public float lowPitchRange = .95f; //The lowest a sound effect will be randomly pitched.

public float highPitchRange = 1.05f; //The highest a sound effect will be randomly pitched.

void Awake()

{

//Check if there is already an instance of SoundManager

if (instance == null)

//if not, set it to this.

instance = this;

//If instance already exists:

else if (instance != this)

{

//Destroy this, this enforces our singleton pattern so there can only be one instance of SoundManager.

//Destroy(gameObject);

}

//Set SoundManager to DontDestroyOnLoad so that it won't be destroyed when reloading our scene.

//DontDestroyOnLoad(gameObject);

}

//Used to play single sound clips.

public void PlaySingle(AudioClip clip)

{

//Set the clip of our efxSource audio source to the clip passed in as a parameter.

efxSource.clip = clip;

//Play the clip.

efxSource.Play();

}

//RandomizeSfx chooses randomly between various audio clips and slightly changes their pitch.

public void RandomizeSfx(params AudioClip[] clips)

{

//Generate a random number between 0 and the length of our array of clips passed in.

int randomIndex = Random.Range(0, clips.Length);

//Choose a random pitch to play back our clip at between our high and low pitch ranges.

float randomPitch = Random.Range(lowPitchRange, highPitchRange);

//Set the pitch of the audio source to the randomly chosen pitch.

efxSource.pitch = randomPitch;

//Set the clip to the clip at our randomly chosen index.

efxSource.clip = clips[randomIndex];

//Play the clip.

efxSource.Play();

}

}