```
using UnityEngine;
using System.Collections;
/*
 * Cade Sperlich
 * Eddie Gurnee
 */
public class BallAudio : MonoBehaviour {

	//parameters for tweaking
	public float pitchMin;
	public float pitchMax;
	public float volMin;
	public float volMax;
	public float transitionRate;
	public float falloffRate;
	public float normFactor;
	public float cutOffSpeed;
	public float minCollisionSpeed;

	//audio sources
	private AudioSource rollingAudio;
	private AudioSource hitAudio;

	//varibles for logical calculations
	private int frameCount;
	private float curSpeed;
	private float prevSpeed;
	private bool fadeOut;

	void Start() {
		//connect the audio sources and initialize variables
		AudioSource[] audioSources = GetComponents<AudioSource>();

		rollingAudio = audioSources[0];
		hitAudio = audioSources[1];

		prevSpeed = rigidbody.velocity.magnitude;
		rollingAudio.volume = 0.0f;
		rollingAudio.Play ();
		fadeOut = false;
		frameCount = 0;
	}

	void OnCollisionStay(Collision collision) {

		if(++frameCount > 10)	//if we've been on a surface
for 10 frames
			fadeOut = false;


		curSpeed = this.rigidbody.velocity.magnitude;// -
collision.rigidbody.velocity).magnitude;

		//Debug.Log(curSpeed);
		//if we've had a strong enough impact
		if (Mathf.Abs(curSpeed - prevSpeed) > minCollisionSpeed){
			hitAudio.volume = Mathf.Lerp(rollingAudio.volume,
prevSpeed/normFactor, Time.deltaTime * transitionRate);
			hitAudio.Play();
		}
```

```
                    //interpolate the volume and pitch
                    if (curSpeed > cutOffSpeed)
                              rollingAudio.volume =
Mathf.Lerp(rollingAudio.volume, curSpeed/normFactor, Time.deltaTime *
transitionRate);
                    else
                              rollingAudio.volume =
Mathf.Lerp(rollingAudio.volume, 0.0f, Time.deltaTime * transitionRate);
                    rollingAudio.pitch = Mathf.Lerp(rollingAudio.pitch,
Mathf.Clamp (curSpeed/normFactor, pitchMin, pitchMax), Time.deltaTime );

                    //mute
                    if (rollingAudio.volume <= volMin)
                              rollingAudio.mute = true;
                    else
                              rollingAudio.mute = false;
                    prevSpeed = curSpeed;
          }


          void OnCollisionExit(Collision collision) {
                    //if we're not still touching another object
                    if(Physics.OverlapSphere(transform.position, 0.25f).Length
== 1){
                              frameCount=0;               //reset the frame
counter
                              fadeOut = true;             //start fading out in
Update()
                    }
          }

          void Update() {
                    if (fadeOut){
                              rollingAudio.volume -= falloffRate;  //unity
won't let volume fall below 0.0
                    }
          }
}
```