```
using UnityEngine;
using FMODUnity;
using FMOD.Studio;


public class LineCastOcclusion : MonoBehaviour
{

    //Fmod variables
    [Header("FMOD Audio File")]
    [SerializeField]
    [EventRef]
    private string SelectAudio;
    private EventInstance Audio;
    private EventDescription AudioHeader;
    private StudioListener listen;
    private bool AudioIsVirtual;
    public GameObject enemy;
    public GameObject player;

    [SerializeField]
    private LayerMask OcclusionLayer;
    private Color colour;
    //Declaring positions of linecasts
    private Vector3 LeftOffset,RightOffset,CentreOffset,FaceOffset,FeetOffset,EnemyPosition,PlayerPosition;
    [SerializeField]
    private bool togglelines;
    private float LinesCollided;
    // Start is called before the first frame update
    void Start()
    {
        //Fmod variable declaration
        Audio = RuntimeManager.CreateInstance(SelectAudio);
        //Attaches Audio to gameobject
        RuntimeManager.AttachInstanceToGameObject(Audio, GetComponent<Transform>(), GetComponent<Rigidbody>());
        Audio.start();            //Starts audio
        Audio.release();          //Removes FMOD event when audio stops playing
        AudioHeader = RuntimeManager.GetEventDescription(SelectAudio);
        //Finds location of studio listener
        listen = FindObjectOfType<StudioListener>();


        togglelines = true;
        LinesCollided = 0;

        //Initializing positions of linecasts
        LeftOffset = new Vector3(0, 2.0f, 0.4f);
        RightOffset = new Vector3(0, 2.0f, -0.4f);
        CentreOffset = new Vector3(0, 2, 0);
        FaceOffset = new Vector3(0, 3, 0);
        FeetOffset = new Vector3(0, 1, 0);
    }
    // Update is called once per frame
    void FixedUpdate()
    {
        Audio.isVirtual(out AudioIsVirtual);

        LinesCollided = 0;

        PlayerPosition = player.transform.position;
        EnemyPosition = enemy.transform.position;

        //Draws line to the left
        LineCastDraw(EnemyPosition+LeftOffset, PlayerPosition + LeftOffset);
        //Draws line to the right
        LineCastDraw(EnemyPosition + RightOffset, PlayerPosition + RightOffset);
        //Draws line to the centre
        LineCastDraw(EnemyPosition + CentreOffset, PlayerPosition + CentreOffset);
        //Draws line to the centre
        LineCastDraw(EnemyPosition + FaceOffset, PlayerPosition + FaceOffset);
        //Draws line to the centre
        LineCastDraw(EnemyPosition + FeetOffset, PlayerPosition + FeetOffset);


        SetFMODAudioParameter();
    }
    //Raycasts are draw with functions to detect if individual lines are passing through terrain
    private void LineCastDraw(Vector3 StartPoint,Vector3 EndPoint)
    {

        RaycastHit hitObject;
        Physics.Linecast(StartPoint, EndPoint, out hitObject, OcclusionLayer);

        if (Physics.Linecast(StartPoint, EndPoint, out hitObject, OcclusionLayer))
        {
            LinesCollided++;
            if (togglelines == true)
            {
                //Makes line red if raycast collides with object
                Debug.DrawLine(StartPoint, EndPoint, Color.red);
            }


        }
        else
        {//Makes line green if raycast doesnt collider with object
            if (togglelines == true)
            {
                Debug.DrawLine(StartPoint, EndPoint, Color.green);
            }
        }
    }
    private void SetFMODAudioParameter()
    {
        //function changes audio volume my amount of lines hit and mimicks occlusion
        Audio.setParameterByName("Occlusion", LinesCollided*0.25f);
    }



}
```