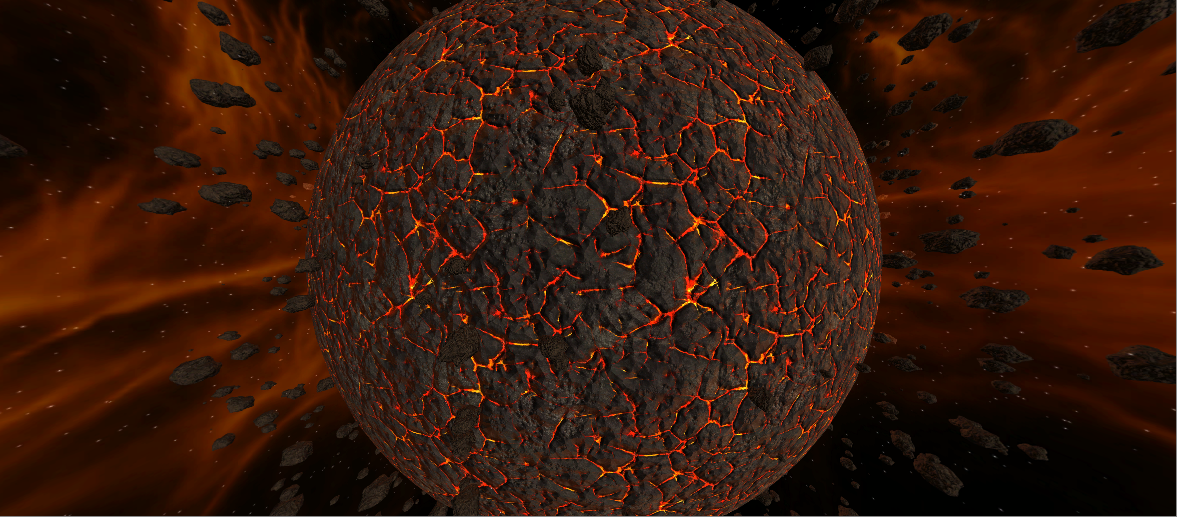
**Splash Screen**

**Objective:** This lesson has 2 parts. First we’re going to add a splash screen with music that will lead us into our game. You’re going to figure out how to code it using the previous project for examples. Second we’re going to have the music play without interruption from to splash screen to the level.

1. Create a new scene. Go to **File> New Scene**
2. Remove the **directional light**
3. Add your chosen skybox to the skybox in this scene
4. Point the **camera** towards something in your skybox and adjust your **Field** **of** **View** to get a good loading screen view.



1. Save Scene as **Splash**
2. Create an **Empty** **Game** **Object** and name it **Music Player** and don’t forget to Reset its position.
3. Add an **Audio** **Source** component to it.

**Challenge:**

* Find a song that you like for an intro and add it to the Audio Source.
* Create a script called **MusicPlayer**
  + use the invoke method to switch levels and
  + create a **levelLoadDelay** variable
* Get the next level to load. Don’t worry if the song cuts off.

**ANSWER**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class MusicPlayer : *MonoBehaviour*{

    [*SerializeField*] float levelLoadDelay = 2f;

    void Start(){

        Invoke("LoadNextScene", levelLoadDelay);

    }

    void LoadNextScene(){

        SceneManager.LoadScene(1);

    }

    void Update(){

    }

}

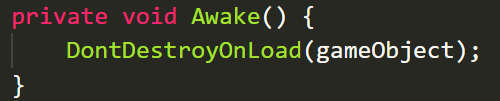
**Continuous Music**

**Objective:** We’re going to have the music play without interruption from to splash screen to the level.

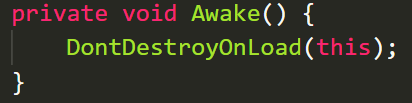
**Background**: Its important to know how Unity executes the methods. We’re going to learn about a method called [Awake](https://docs.unity3d.com/Manual/ExecutionOrder.html). The important thing to note that Awake happens before Start and before Update keeps repeating.

Also! [DontDestroyOnLoad](https://docs.unity3d.com/ScriptReference/Object.DontDestroyOnLoad.html) method. Pretty much you give it an object to not destroy the target Object when loading a new Scene.

1. Create the **Awake** method in the **MusicPlayer** file and inside call the **DontDestroyOnLoad** method. Give it **gameObject** as the argument since its an instance of the object GameObject.



So what's the context here? We are in a MusicPlayer script, that music player script is attached to the scene. A music player is a component of the music player Game Object. So when we say game object with a small g from inside this script, we are talking about the thing I've got highlighted in blue, the very music player itself, that game object. So its by saying, don't destroy on load game object, we're saying don't destroy the thing I'm attached to.

Alternate way you could replace **GameObject** with **this**

1. Go ahead and test your changes. Make sure the music plays without interruption.

**ANSWER**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class MusicPlayer : *MonoBehaviour*{

    [*SerializeField*] float levelLoadDelay = 2f;

    private void Awake(){

        DontDestroyOnLoad(this);

    }

    // Start is called before the first frame update

    void Start(){

        Invoke("LoadNextScene", levelLoadDelay);

    }

    void LoadNextScene(){

        SceneManager.LoadScene(1);

    }

    // Update is called once per frame

    void Update(){

    }

}