

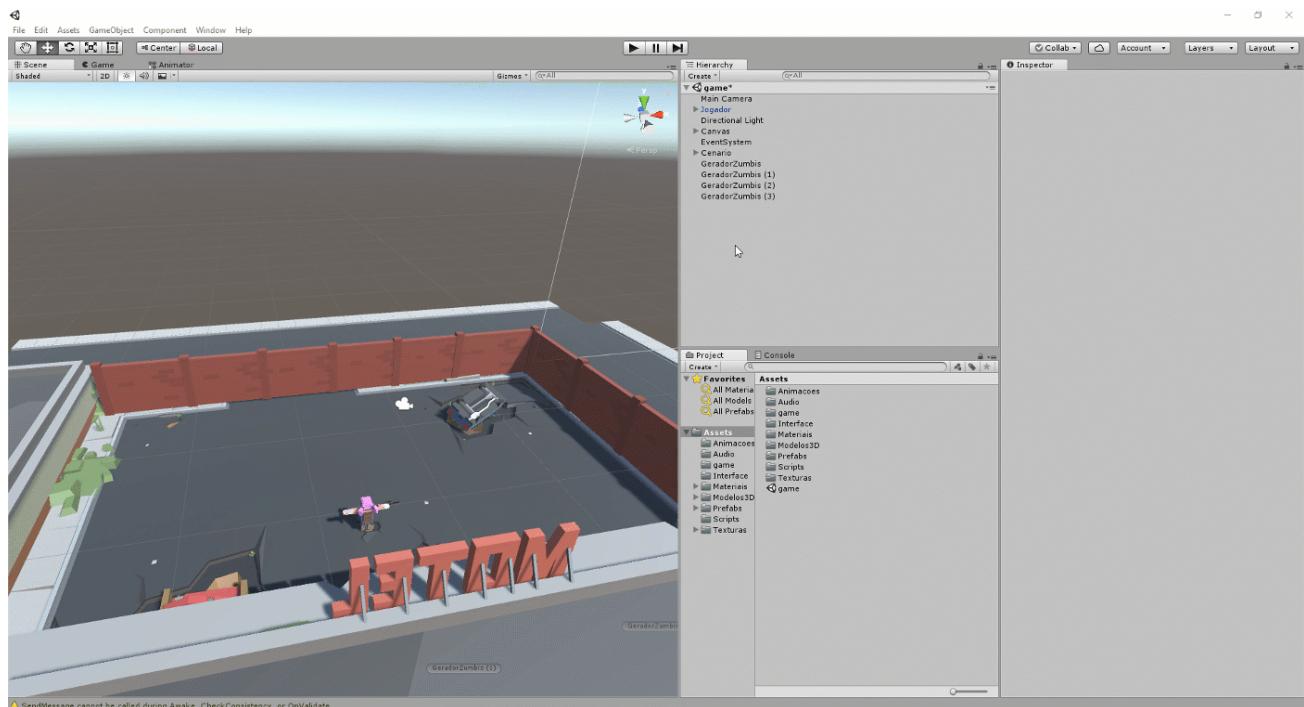
03

Música no jogo

Temos que colocar agora áudio no nosso jogo. Primeiramente [clique aqui](https://github.com/alura-cursos/unity-shooter-2/raw/master/Audio.zip) (<https://github.com/alura-cursos/unity-shooter-2/raw/master/Audio.zip>), faça download dos arquivos, extraia a pasta e jogue os arquivos para a Unity. Lembre-se de criar uma pasta para os arquivos ficarem organizados.

Com os áudios dentro da Unity, vamos colocá-los num objeto. Para isso vamos criar um objeto um objeto vazio, nomear como **ControlaAudio**, e adicionar um componente do tipo **AudioSource**.

Agora é só arrastar a música `You_Cant_Dodge_a_Laser` para a propriedade `AudioClip` no Inspetor do seu **AudioSource** e marcar a opção `Loop`.



Temos agora a música no jogo! Mas e os outros sons? Para colocá-los precisamos programar esse comportamento.

Para isso vamos criar um script chamado **ControlaAudio** e arrastá-lo para o nosso objeto **ControlaAudio**.

Neste código vamos definir uma variável que irá fazer uma guardar o **AudioSource** do nosso objeto.

```
private AudioSource Meu AudioSource;
```

Agora vamos fazer essa variável valer o nosso **AudioSource** mas ao invés de fazer isto no `Start` vamos fazer no método `Awake` . Por que no `Awake` ?

Podemos querer tocar um som quando o jogo começa, então a gente está garantindo que nosso **AudioSource** já tenha a possibilidade de tocar sons antes do jogo começar, já que o `Awake` roda antes do `Start` .

```
public class ControlaAudio : MonoBehaviour {  
  
    private AudioSource Meu AudioSource;  
  
    void Awake()  
}
```

```
{  
    Meu AudioSource = GetComponent< AudioSource >();  
}  
}
```

Agora que já temos a variável pronta, só temos que criar uma forma de fazer os sons tocarem em outros scripts. Isso é importante pois esse código não sabe o que está acontecendo no momento para tocar o som correto na hora certa.

Para isso criaremos uma forma de acesso a esta classe. É diferente do que fizemos até agora, pois até agora nós iríamos até cada script criar uma variável que referencia este código, e a partir dela tocarmos os sons.

Só que isso teria que ser feito quase em todos os códigos do jogo. Para facilitar, vamos utilizar um padrão (pattern) conhecido como *Singleton*. Não vamos implementar o padrão por completo, mas teremos o princípio que queremos.

É interessante lembrar que estamos utilizando o padrão forma simples, somente para tocar os sons. Porém, ao implementar *Singletons* para algo mais complexo podem surgir problemas. [Clique aqui](#) (<http://blog.caelum.com.br/singletons-e-static-perigo-a-vista/>) se quiser ler mais sobre o assunto. Cuidado!

Bom, para implementar o *Singleton* vamos criar uma variável com o modificador `static`. Este modificador garante que essa variável vai valer a mesma coisa independente de onde esta classe está sendo chamada.

```
public static AudioSource instancia;
```

E no `Awake` vamos colocar o valor desta variável como a nossa outra variável de `AudioSource`.

```
public class ControlaAudio : MonoBehaviour {  
  
    private AudioSource Meu AudioSource;  
    public static AudioSource instancia;  
  
    void Awake()  
    {  
        Meu AudioSource = GetComponent< AudioSource >();  
        instancia = Meu AudioSource;  
    }  
}
```