

1. 播放音频

我们可以通过生成 AudioBundle 的实体来播放声音。通过 PlaybackSettings 组件控制播放次数。

```
fn setup(mut commands: Commands, asset_server:
Res<AssetServer>) {
    commands.spawn(AudioBundle {
        source: asset_server.load("19_playing_sounds/
Windless Slopes.ogg"),
        settings: PlaybackSettings::LOOP,
    });
}
```

可以使用 GlobalVolume 资源来控制全局音量。注意，更改此值不会影响已经播放的音频。

```
insert_resource(GlobalVolume::new(2.))
```

Bevy 会将 AudioSink 组件添加到我们刚刚添加的 AudioBundle 实体中，以控制播放。

2. 控制播放

为了控制音频播放，我们可以使用 AudioSink 组件的方法。set_volume() 设置音量，set_speed() 设置速度，pause() 暂停，play() 播放，stop() 停止，toggle() 切换播放暂停。

```
fn control_audio(audio: Query<&AudioSink>, keyboard:
Res<ButtonInput<KeyCode>>) {
    let Ok(sink) = audio.get_single() else {
        return;
    };
    if keyboard.just_pressed(KeyCode::ArrowUp) {
        sink.set_volume(sink.volume() + 1.);
    }
    if keyboard.just_pressed(KeyCode::ArrowDown) {
        sink.set_volume(sink.volume() - 1.);
    }
    if keyboard.just_pressed(KeyCode::KeyW) {
        sink.set_speed(sink.speed() + 1.);
    }
    if keyboard.just_pressed(KeyCode::KeyS) {
        sink.set_speed(sink.speed() - 1.);
    }
    if keyboard.just_pressed(KeyCode::KeyP) {
        sink.pause();
    }
    if keyboard.just_pressed(KeyCode::KeyL) {
        sink.play();
    }
    if keyboard.just_pressed(KeyCode::KeyO) {
        sink.stop();
    }
    if keyboard.just_pressed(KeyCode::KeyT) {
        sink.toggle();
    }
}
```