

# 音频音效设计

## 1. 设计目标与总体思路

### 1.1 设计目标

实现全游戏统一的音效与音乐播放管理，避免场景树中散落大量  
AudioStreamPlayer 节点

保证音效零卡顿、背景音乐无缝切换、死亡/重开/结局时声音状态完全干净  
支持后续轻松扩展音量设置、淡入淡出、独立总线分类调节  
在拾取 buff 时间减慢时保持音效音调不失真  
所有声音调用集中、可追踪、易于替换资源

### 1.2 模块分工

所有音效资源统一放在 Sounds/ 目录，命名即为调用 key  
SoundManager 作为唯一音频调度中心，负责预加载、播放、循环、停止、  
清理  
Master.gd + MainUI.gd + 其他脚本-只负责什么时候该播什么，不直接持  
有 AudioStreamPlayer  
default\_bus\_layout.tres -集中管理总线与全局效果

## 2. 音频资源目录结构 (Sounds/)

- .wav: 短音效 (射击、受伤、UI、破坏、拾取等)
- .mp3 / .ogg: 长背景音乐 (intro、boss\_music、ending 等)
- .fms: 部分遗留或临时音效包 (已全部转为 wav/mp3 使用)

## 3. 核心管理器: SoundManager.gd

### 3.1 注册方式和播放实现细节

在 project.godot 的 autoload 段注册:  
SoundManager="\*res://Scripts/SoundManager.gd" 后游戏启动即实例化，常驻  
整个生命周期。

每次播放动态实例化 AudioStreamPlayer 并挂到 SoundManager 节点下  
单次音效：播放完自动 queue\_free()

循环音效：登记进 looping\_sounds 字典，使用 while + await finished 实现

所有播放器默认 process\_mode = Node.PROCESS\_MODE\_PAUSABLE，受 HitStop 时间缩放影响，音调会变低，制造打击感

#### 4. 生命周期与状态管理

初始化：游戏启动-Autoload 加载 SoundManager-Master.\_ready() 播放开场音乐

正常流程切换：所有背景乐切换均先 stop\_sound() 旧音乐-play\_sound() 新音乐，避免重叠

死亡/结局：统一调用 stop\_all\_sounds() 确保无残留音效，再播放对应死亡或结局音乐/视频

hitstop 时：当前 AudioStreamPlayer 会受 time\_scale 影响，HitStop 时 音调会自然降低，增强打击感。

#### 5. 主要调用点与触发逻辑

游戏启动/开场动画：Master.gd (\_ready) 调用 play\_sound("start\_intro")，播放单次开场音乐 start\_intro

开始游戏按钮按下：Master.gd 调用 stop\_sound("start\_intro") - play\_sound("intro", loop=true)，停止开场音乐并切换为循环主玩法背景乐 intro

射击（开场动画 10 连发）：Master.gd 调用 play\_sound("shoot") × 10，连续播放 10 次射击音效 shoot

玩家受伤：MainUI.gd 调用 play\_sound("hurt")，受伤时同时触发 HitStop 并播放受伤音效 hurt

UI 按钮获得/失去焦点：startgameUI.gd 调用 play\_sound("UI\_move")，菜单导航时播放移动音效 UI\_move

UI 确认/开始游戏：Master.gd 调用 play\_sound("UI\_SELECT")，确认选择时播放确认音效 UI\_SELECT

玩家死亡: Master.gd (restart\_game) 调用 stop\_all\_sounds() – play\_sound("death"), 清空所有音效后播放死亡音效 death

游戏通关: Master.gd (finish\_game) 调用 stop\_sound("boss\_music") – stop\_all\_sounds() – 播放 ending.ogv (自带音轨), 清空所有音效后播放带自带音轨的结局视频

场景切换/重开: Master.gd 调用 stop\_all\_sounds(), 任何重开或场景切换前强制清空所有正在播放的所有音效, 保证声音状态完全干净