

Unity x Wwise 游戏音频系统

音频设计文档

项目 FPS Demo · 个人作品集

引擎 Unity 6000.4.4f1 · Wwise 2024.1.13.9056

负责人 胡锦涛

版本 v1.0 · 2026.05

一、项目说明

基于 Unity FPS Microgame 模板开发的个人音频作品集 Demo，场景为科幻风格基地——敌人为悬浮机器人（HoverBot）与固定炮台（Turret），背景为开阔的科幻室外环境。

项目目标是验证 Wwise 在 Unity 中的核心集成能力，重点不在音效品质的精细打磨，而在系统逻辑的完整性与稳定性。

共实现五个音频系统，使用 2 个 SoundBank、1 个 State Group、1 个 Switch Group、1 个 RTPC。

二、美学概念

本项目音频风格定义为科幻写实，参考《星球大战》系列的音效设计思路：不依赖纯合成音色，而是用真实采样素材重构出科幻质感——如用弦乐震弓（Vibe_String）作为炮台启动的音色基础，用打击乐（Vibe_Percussion）推进 Boss 战氛围。

场景为科幻室外基地，音频整体基调：空旷 · 机械 · 紧张。

三、系统设计

3.1 动态音乐系统（State Machine）

State Group: GameState

States:

- None 默认，无音乐
- Exploration (预留，当前未激活)
- Boss 炮台检测到玩家时触发

触发逻辑

游戏事件	State 变化	同步触发 Event
炮台检测到玩家	→ Boss	Play_Turret_Power_up + Play_Boss_Music
炮台销毁	→ None	Stop (Boss Music 停止)

实现脚本: EnemyTurret.cs

```
// 检测到目标时
AkSoundEngine.PostEvent(WwisePowerUpEventName, gameObject);
AkSoundEngine.SetState(WwiseBossStateGroup, WwiseBossStateName);
AkSoundEngine.PostEvent(WwiseBossMusicEventName, gameObject);

// 销毁时
AkSoundEngine.SetState(WwiseBossStateGroup, "None");
```

Boss 音乐 (Play_Boss_Music) 为 Infinite 类型，随 State 切换持续播放至炮台销毁。

3.2 实时参数心跳（RTPC）

Game Parameter: Player_Health

范围: 0 → 100
 默认值: 100
 更新方式: 每帧实时更新

实现逻辑

```
// 游戏开始时启动心跳音效
AkSoundEngine.PostEvent("Play_Heartbeat", gameObject);

// 每帧更新血量参数
AkSoundEngine.SetRTPCValue("Player_Health", m_Health.GetRatio() * 100f, gameObject);

// 胜利时停止心跳
AkSoundEngine.ExecuteActionOnEvent(
    "Play_Heartbeat",
    AkActionOnEventType.AkActionOnEventType_Stop,
    gameObject, 0,
    AkCurveInterpolation.AkCurveInterpolation_Linear
);
```

Play_Heartbeat 为 Infinite 类型，心跳音效的强度通过 Player_Health RTPC 在 Wwise 内部曲线控制，血量越低压迫感越强。

3.3 地面材质脚步声 (Switch)

Switch Group: Meterial (工程内命名保持原样)

```
Switches:
├─ Metal    金属地面 (13 个音频变体)
├─ Sand    沙地 (25 个音频变体)
└─ Default 默认地面
```

Unity 触发方案

通过检测碰撞体 Tag 切换 Switch 值，动画事件触发 Foot_Player:

```
// 地面标签检测
if (hit.collider.CompareTag("Footstep_Metal"))
    switchValue = "Metal";
else if (hit.collider.CompareTag("Surface_Sand"))
    switchValue = "Sand";
else
    switchValue = "Default";

AkSoundEngine.SetSwitch("Meterial", switchValue, gameObject);

// 动画事件触发
AkSoundEngine.PostEvent("Foot_Player", gameObject);
```

金属 13 个变体 / 沙地 25 个变体，Wwise 内设置 Avoid Repeating Last N，避免重复感。

3.4 环境音 (Ambience)

事件: Play_Amb_Base_Wind / Stop_Amb_Base_Wind

场景为科幻室外基地，环境音以持续风声为主干，叠加科幻质感的电气音和飞船掠过音效，营造空旷的外星基地氛围。

音频素材

素材	用途
Amb_Base_Wind.wav	持续底层风声

素材	用途
SciFiElectrical_S08SF.275.wav	科幻电气背景音
SpaceshipPassBy_S011SF.714.wav	飞船掠过一次性音效

Play_Amb_Base_Wind 为 Infinite 类型，场景加载时启动，关卡结束时调用 Stop_Amb_Base_Wind 停止。

3.5 武器与角色音效

事件列表

Event	触发时机	时长
Player_Gunshot	玩家开枪	0.175—0.220s
Gun_ClodDown	弹药耗尽冷却	0.889s
Play_Player_Jump	跳跃	0.206s
Play_Player_Land	落地	0.361s
Turret_Gunshot	炮台射击	0.128—0.161s
Play_Turret_StopFire	炮台停火（失去目标）	0.339s
Play_Turret_Power_up	炮台激活	3.643s
Play_Emenry_destory_Turret	炮台销毁	2.771s
Play_HoverBot_GunShot	悬浮机器人射击	1.280s
Play_Emenry_Hovery	悬浮机器人悬停	1.125s
Emenry_destory	悬浮机器人销毁	2.971s
Pickup_Health	拾取血包	3.217s

炮台激活（Play_Turret_Power_up）使用 Vibe_String.wav 作为底色，与 Boss 音乐进入协同，强化"战斗开始"的听觉信号。

四、SoundBank 结构

SoundBank	内容	加载时机
Init	全局初始化配置	游戏启动
Main	全部 16 个 Event + 43 个音频文件	关卡加载

所有资源集中在 Main Bank，Demo 规模下无需拆分。

五、已知问题记录

问题	说明
命名拼写	Meterial (Material)、Emenry (Enemy)、cLod down (cool down) 为工程内原始命名，保留以维持引用一致，后续版本可统一规范
postWwiseevent.cs	空模板脚本，未实现功能
Exploration State	已在 Wwise 中定义，暂未接入触发逻辑，预留作后续扩展

六、验证记录

系统	验证项	结果
State Machine	炮台检测/销毁触发 Boss 音乐切换	✓
RTPC	血量 0→100 心跳变化连续	✓
Switch	Metal / Sand / Default 脚步切换	✓
Ambience	风声场景启动/停止	✓
武器	玩家 / 炮台 / HoverBot 枪声	✓
全系统	Profiler 压力测试音频事件触发率	✓ 100% 稳定触发

胡锦涛 · 2026.05