

移动游戏开发的音效优化

Ben Houge

伯克利音乐学院
音乐技术创新
西班牙瓦伦西亚



GAME DEVELOPERS CONFERENCE™ CHINA

SHANGHAI INTERNATIONAL CONVENTION CENTER

SHANGHAI, CHINA · OCTOBER 19-21, 2014

音效优化

- 不同于电影配乐music score或音乐CD
- 数据大小data size vs. CPU使用率
- 我们必须对每一代进行优化
- 在限制范围limited scope内最大限度地提高音频质量

《Leisure Suit Larry: Love for Sail!》

- MS-DOS 5.2或Windows 3.1或更高版本
- 486 DX2处理器(66GHz)
- 8MB 内存(DOS), 12MB(Windows)
- 22MB Hard硬盘空间
- 超级视频图形适配器(SVGA), 256色
- 1996年的规格！

与iPhone 6比较

- 1GB内存
- 64位双核1.38GHz处理器
- 16到128GB的硬盘空间
- 2014年的规格！

音效优化optimazation

- 有效地注记(Spotting)和剪辑editing游戏音效
- 数字音频理论与格式
- 创造性的音效再利用
- 文件管理与中间件 middleware

有效地注记和剪辑游戏音效

节省游戏音效占用空间的最好办法：

不要把音效放在你的游戏当中！

注记(Spotting)游戏

- 了解游戏信息
- 考虑哪些东西对用户来说是重要的
- 不要认为在所有的时间点上都需要音效
- 原型prototype和迭代
- 记录你的音效并进行分析
- 让专业音效人员较早地参与到开发工作当中

剪辑游戏音效

- 紧凑compactly地剪辑音效
- 剪辑edit to到过零点(zero crossings)
- 记录高音量音效, 在软件中调低音量

音频质量和文件大小

三个参数影响比特率

- 采样率
- 比特深度
- 通道数

采样率

- 从麦克风采样模拟信号
- 振幅测量
- 线性PCM(未压缩)
- 采样率:采样的频率
- 奈奎斯特频率
- CD音质为44100Hz

比特深度

- 每个样品测量的分辨率
- 低比特深度增加噪音
- CD音质为16比特

采样率/比特深度演示

感知音频编码

- 分析信号，丢弃我们不会错过的频率
- 有损压缩：部分数据永远消失
- 基于输入进行变化：噪音压缩很少，空白压缩很多

例子

- MP3
- Ogg Vorbis
- AAC
- WMA/XMA
- AC3

感知音频编码

- 有时可以为了美学目的使用编码
- 音效解压缩的性能损失
- 在某些平台上 (iPhone、Xbox One...) 支持硬件加速功能

无损音频压缩

- 例子
 - FLAC
 - 苹果无损
- 保留了所有的数据
- 不是非常适合游戏

谨慎压缩

- 不用对所有的音效使用相同的设定
- 花费更多的时间，但能提高音质
- 保留高分辨率版本，工作使用副本

创造性的音效再利用

再利用的好处

- 从现有数据中创造出新的音效
- 最小化占用空间，最大化变动
- 地图音频参数集成到游戏数据
- 密切同步和浸入

例子：脚步声

- 从一组声音当中随机选择
- 实时操控音调pitch和音量
- 模仿自然界的物理效果
- 在CPU占用方面非常有效率

例子：组合 (Combinatoriality)

- 多层随机音效
- 组合数指数上升
- 一致性也提高
- 仍然十分有效

例子：随机循环

- 短循环的无穷变化
- 扩展性和参数化
- 一种粒子合成法

例子：生成音乐

- 多个独立层
- 间歇性的乐句
- 同步到节拍器
- 游戏事件选曲
- 可扩展到游戏玩法

例子:《终结战争》(EndWar)加载音乐

- 极其有限的资源
- 内存很少, 没有光盘
- 音乐响应加载时间
- xxx kb

实时效果

- 先进的数字信号处理
- 无需多个版本
- 可通过游戏参数进行扩展
- 无穷的变化

《终结战争》(EndWar)中的音效

- 无线电音效的过滤和失真
- 用于距离模拟的过滤器
- 用于爆炸的音频粒子系统

iOS7的音频效果

kAudioUnitSubType_BandPassFilter
kAudioUnitSubType_DynamicsProcessor
kAudioUnitSubType_Delay
kAudioUnitSubType_AUFilter
kAudioUnitSubType_GraphicEQ
kAudioUnitSubType_HighPassFilter
kAudioUnitSubType_HighShelfFilter
kAudioUnitSubType_PeakLimiter

kAudioUnitSubType_LowPassFilter
kAudioUnitSubType_LowShelfFilter
kAudioUnitSubType_MultiBandCompressor
kAudioUnitSubType_MatrixReverb
kAudioUnitSubType_NetSend
kAudioUnitSubType_ParametricEQ
kAudioUnitSubType_SampleDelay
kAudioUnitSubType_Pitch

利用iOS8的新AVAudioEngine能够更容易地使用！

合成

- 拥有巨大潜力的话题
- 比如音乐制作当中的“软件合成”(soft synths)
- 更多的CPU占用率, 但数据要少得多
- 实施操控的机会
- MIDI的黄金时代?
- 也可以用于音效

文件管理与中间件

游戏音效数据管理

- 需要组织
- 追踪许多小文件
- 强制执行一种命名规则

音效引擎 vs. 游戏引擎

- 游戏引擎调用声音事件
- 音效引擎管理声音事件
- 音响设计师定义音效行为
- 明确有效的分工

可能的音效行为

- 播放声音、播放多个声音、播放声音变化、停止声音播放、播放一个声音的同时停止其他声音、改变音量、应用效果等等
- 音效执行者并不需要知道这一点，只需要调用事件

使用中间件

- 常见的解决方案
 - Wwise (音频动力学)
 - FMOD
- 能为最简单的游戏之外的所有游戏带来巨大收益
- 价格合理(在某些情况下是免费的)
- 可用于Android、iOS和其他平台

中间件的好处

- 游戏内音效混合用于紧凑的重复循环
- 记录和分析音效数据使用情况
- 避免冗余
- 导出到多个平台
- 管理音效优先级
- 整合音效插件

结论

总结

- 小心地标注游戏音效
- 紧凑地剪辑音效
- 分别地压缩音效
- 创造性地再利用音效
- 仔细地追踪音效资产
- 反复地混合和分析音效

有问题吗？

Ben Houge

伯克利音乐学院

音乐技术创新

[bhough@berklee.edu](mailto:bhouge@berklee.edu)

推特/微博：@AleaBoy

- 单击此处剪辑母版文本样式
 - 第二级
 - 第三级
 - 第四级
 - 第五级