

国内第一本介绍FMOD游戏音频制作的图书

FMOD资深研究者薛乔与知名游戏音乐制作团队“小旭音乐”共同打造，十年经验浓缩而成
国内最大的电脑音乐网站MIDI fan倾情推荐

FMOD

游戏音频制作快速上手

薛乔 小旭音乐 编著



清华大学出版社

FMOD

游戏音频制作快速上手

薛乔 小旭音乐 编著

清华大学出版社

北 京

内 容 简 介

本书是国内第一本介绍 FMOD 游戏音频引擎工具的书, 书中由浅入深地介绍了如何使用 FMOD Designer 2010 进行声音设计。本书侧重介绍音频引擎的概念、使用音频引擎进行声音设计的工作流程及以音频引擎为基础的声音制作思路等内容。

本书共 8 章。涵盖的内容有: 音频引擎概念及 FMOD 的获取方式; FMOD Designer 2010 基本界面; FMOD 系统的重要概念和基本操作; FMOD 中的简单事件概念及应用; 声音定义的概念及应用; 多轨事件的概念及应用; 多轨事件的应用; 4 个综合实例。本书的附录列举了书中没有提及的属性和功能。

本书适合所有音效制作者、游戏策划人员及游戏音频爱好者阅读; 也适合正在使用或者即将使用 FMOD 引擎作为声音设计平台的音频设计人员及希望在游戏作品中使用音频引擎的程序员阅读; 对于开设了音频设计课程的各院校, 本书也不失为一本好教材。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

FMOD 游戏音频制作快速上手 / 薛乔等编著. —北京: 清华大学出版社, 2013.2
ISBN 978-7-302-30776-1

I. ①F… II. ①薛… III. ①电子游戏—音乐制作—应用软件 IV. ①J619-39

中国版本图书馆 CIP 数据核字(2012)第 285861 号

责任编辑: 夏兆彦

封面设计: 欧振旭

责任校对: 徐俊伟

责任印制: 王静怡

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社总机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京世知印务有限公司

装 订 者: 三河市李旗庄少明印装厂

经 销: 全国新华书店

开 本: 185mm×260mm	印 张: 18.25	字 数: 440 千字
版 次: 2013 年 2 月第 1 版	印 次: 2013 年 2 月第 1 次印刷	
印 数: 1~4000		
定 价: 69.00 元		

产品编号: 047218-01

游戏音频，放飞梦想

（序一）

10 年前，我与颜东成先生合作编写了一本电脑音乐教程《MIDI 技巧与数字音频》，引起了读者的很大反响。这是我参与编写的第一本电脑音乐类图书，那时我正好 20 出头，刚加入茫茫的北漂大军中。十年后的今天，我一手经营起来的“小旭音乐”团队又参与编写了《FMOD 游戏音频制作快速上手》一书，这是我及团队参与编写的第 13 本电脑音乐技术类著作。与之前的 12 本图书不同的是，本书讨论的内容集中在电脑音乐中的一个细分领域——游戏音频制作。这十年，我积极参与了电脑音乐技术在中国的推广，而且也组建了“小旭音乐”团队。团队从最初的几个人发展到现在的 30 多个人，我们团队的中心工作就是致力于游戏音乐的创作、音效制作和游戏音频理念的研究与推广。

游戏音乐在中国出现已经超过 10 个年头了，尽管网络游戏行业在此期间得到了飞速发展，但依然还没有将游戏音乐这个与游戏研发相关的产业带入一个应有的发展高度。这可能与声音没有游戏画面那么直观有一定关系，但我想更多的是与整个游戏产业不重视音频，而且网游市场相对比较浮躁有关。作为从业 10 多年的行业老兵，“小旭音乐”团队一直在为改变这种局面而努力。FMOD 这个游戏音频专业引擎的出现和它越来越广泛的应用为我们打开了一扇窗，无论是游戏音频从业者和音频程序员，还是对声音感兴趣的策划和游戏美工人员，都应该认识并了解 FMOD 的特性，为游戏研发提供更友好、更直观的音频接口，在与像“小旭音乐”这样的外包创作团队进行沟通时也会更加方便。

研发 FMOD 的澳大利亚 Firelight 公司相关负责人 2012 年到访了“小旭音乐”团队，给我们带来了 FMOD 2013 年新版本上市的相关消息，并与我们讨论了未来一系列深度战略合作计划。其中最重要的就是从 2013 年开始，Firelight 公司将与小旭音乐合作，在国内进行 FMOD 技术普及的系列讲座。我们将邀请更多的游戏策划、音频程序员和音频设计师等参与本次活动。这将是游戏音频行业的一次开放性盛宴，对 FMOD 及游戏声音引擎的相关概念和技术在中国的推广也将起到重要的作用。

中国网络游戏经 10 年历练而愈见成熟。我们相信，游戏音频的概念在整个游戏研发产业中将会受到越来越多的重视，我们也希望能与更多国内外同行和技术专家进行交流，为中国游戏音频行业的发展做出自己的努力和贡献，为此我和小旭音乐团队一直努力着。希望未来的一天，中国的游戏音乐也能发展起来，走向世界，这是我们的梦想。让我们为实现这个目标而一起放飞梦想吧！

各位读者在阅读中如有任何疑问，欢迎与我们微博互动：<http://e.weibo.com/gamemusic>，或邮件交流 xx@GameMusic.com.cn 谢谢！



创始人：卢小旭

www.GameMusic.com.cn

踏上游戏音频的新征程

(序二)

游戏音频长期以来是作为游戏中的重要组成部分而存在的。早在任天堂黄卡的时代,《马里奥》、《魂斗罗》等经典游戏中的音乐音效就让人印象深刻。尽管在当时的游戏中受到诸如容量、运算能力等限制,为了带来更好的游戏体验,开发者们向游戏中添加足够的音乐音效的欲望一点也没有衰减的迹象。游戏《马里奥》就使用了4个关卡音乐、数十个音效。《魂斗罗》更是每个关卡都是不同的背景音乐,至今人们还对这些音乐津津乐道。

在今天,一些早期经典游戏中的音频应用已经成为了现在游戏音频设计的参照标准。中国的游戏开发者当然也是在这些经典游戏的影响下成长起来的,自然在作品中也继承了这些特征,比如一个不同场景关卡使用不同的背景音乐,像是大家所熟知的《仙剑》;游戏中角色动作触发的事件都有对应音效,特殊场合还有激发类型的音乐或者音效(Sting),像是《三国群英传》等。这些比较传统的游戏音频应用方案一直被沿用至今。

而在我们现在的游戏中有一个不太好的迹象:画面随着3D引擎技能的提升在提升,而游戏中很多设计思路却只是追随国外的脚步,还有一些内容甚至一直没有变过,游戏音频设计思路就是一个很典型的例子。在20年前,BGM对于游戏来说就是一个场景重复播放一个音乐,在20年以后仍然是这样。音效也不需要考虑变通和规避单调性,单一动作仍然是触发单一音效。当然对于氛围音效来说,可能就更加没有考虑过这个概念了。然而事实上在这个领域,别人早就走在了我们的前面。游戏《战神》使用了随节拍同步触发的音乐,《植物大战僵尸》则是无缝循环且可以进行变奏的循环音乐。《街霸3.3》,一个1999年发售的老牌街机格斗游戏,就已经将音乐上的切片概念应用到游戏BGM当中,用来创造回合过渡中的强烈冲击与自然衔接。知名网游里程碑巨作《魔兽世界》,通过解析游戏数据文件可以发现,同样是一段表现童话般森林的BGM(艾尔文森林主题),它被拆解成了3段可以自然过渡的音乐使用。实际上,这3段来自同一主题音乐的变奏则会被随机触发,创造连绵不绝的听觉氛围。而我们的游戏中的BGM,现在仍然在无缝循环和间断循环两者中进行二选一。

再来说音效,对于很多游戏来说,氛围音效就是没有考虑过也不存在的事物。而实际上在很多游戏的数据文件中,制作人还要为压缩氛围音效的文件长度而感到头痛。又或者多版本的音效随机触发,这个小设计其实在欧美游戏中非常普遍。大到大型RPG,小到Flash游戏都使用了这个思路,其实这是一个很简单也不需要增加太多预算的事情,类似于准备4个版本的脚步声文件用于触发,再在每次触发的时候将音频文件做些微的随机化调整用来创造真实的音效反馈。这样简单的增加一些随机化,就可以极大地改善游戏的听觉表现,规避重复性等问题,使得游戏体验更加平滑。可惜笔者的游戏设计项目中使用这个思路的项目屈指可数,其他包括随游戏内容关联的动态音效就更见不到了。

这样的现状当然并不局限于音频设计方面,国产游戏中缺乏细节也不仅仅是体现在音频上,还体现在画面人设剧本等方面。这个跟现在游戏界乃至整个社会对于盈利上“短平

快”的追求是分不开的，这里也不多做讨论。只是令人惋惜的是，在有些设计精良的国产大作中，音频上虽然没有被忽视，甚至可以说投入了大量的精力和金钱，但是距离先进的设计思路和游戏体验还是有距离的，因此才有了本书需要介绍的内容。

实际上笔者并没有持有这样的观点：要做好一个东西就需要花很多时间、花很多钱，就是慢工出细活。对于优秀的作品来说，时间成本和物质成本只是必要条件中的一部分。很多时候并不是投入成本就能收到效果，极端的情况是投入的成本甚至会造成反效果。比如，把游戏做成赌局，投入更多的时间去研究怎样让这个赌局更刺激，让更多的买家投更多的钱进来，钻研这种东西的游戏有什么意义呢？这根本就不是游戏；或者采用过时的技术，类似于 KOF03 以前系列的 KOF 作品，仍然是用 MIDI 来实现游戏的 BGM，虽然在一代一代地更新音源，最终还是扛不住时代的车轮，直接使用音频文件作为 BGM 了。想必看到这里，读者也有了一定的推断，确实，本书想要介绍的除了是对于一个新工具的使用以外，更多的是想给未来想从事这个行业但不是很了解的人介绍一个更新、更有效的思路，即使用音频引擎来解决游戏中的音频实现的问题。

首先使用音频引擎这样一个新工具，是不是就需要花很多时间来熟悉，花很多资金来购买许可？答案是否定的。笔者本身虽然不是一线程序员，但是对于使用 FMOD 方案来实现游戏音频在程序员中的评价已经有了很多了解。基本评价是一致的：使用 FMOD 来实现是前所未有的方便。以往是用 OpenAL、DirectMusic 需要很多代码才能实现的功能，使用 FMOD 就可以很简单地实现，并且控制做得更好、更稳定，系统开销更小。不仅如此，Unity3D、虚幻 3 等引擎早就集成了 FMOD，而很多正在使用上述引擎的项目组并没有尝试使用 FMOD 的功能，这难道不是一种浪费吗？

而对于项目制作人来说，使用 FMOD 引擎系统就更是音频实现上的利器了。有谁不想让自己的大型 3DMMORPG 能让玩家玩得身临其境呢？可是如果没有成型的音频实现系统，实现良好的氛围音效就比较困难，那么这个方案只能砍了。山洞里面的混响，怎么做呢？在游戏里面集成一个混响的算法？找一个第三方的库？好吧，这个方案也只能砍了。做了一个动作游戏，想在 KO 的时候来个华丽的结束，需要冲击性的音效效果，需要动态滤波，这又得需要查书去实现。实际上，哪怕想从 MP3 文件中的某个精确时间位置开始播放，都需要程序员计算偏移量才能得以实现，这导致很多引擎里面自带的程序接口需要所有的 BGM 不得不从头开始播放。但是 FMOD 引擎却完全可以省去这些工作，直接一个调用就开播了，何必去解决这些鸡毛蒜皮又影响实现的功能？更不用说它对各种音频格式的全面支持了。使用音频引擎，很多以前想实现但不好实现的音频效果现在就可以轻松解决。

音频设计师肯定会觉得 FMOD 是一个福音，因为他们的设计成果终于从砍了又砍的状态变成可以实现了，同时还不需要他们自己去了解程序如何实现。《胧村正》这个游戏的 BGM 过渡是完全无缝的，这对于很多项目来说肯定是一个想尝试但是无从下手的東西。但是用 FMOD 之后这个就完全变成音频设计师和音乐家的工作了——他们只需要创作可以拆解为两个部分的 BGM 即可，剩下的无缝过渡交给 FMOD 解决，而对于他们来说并不需要掌握程序知识，仅需要使用 FMOD Designer 工具设置出一个事件即可。再比如创作子弹时间效果，不再需要音频设计师和程序员进行怎样深度的沟通，音频设计师可以脱离编程，完全依靠原始音频素材和 FMOD Designer 制作一个子弹时间事件，调节参数声音就好像变慢了，调回参数声音就又变快了。实际的游戏开发中，程序员只需要简地触发这个事件并且调节参数就可以完成这个效果，这对于双方来说都是多么的方便！

其实 FMOD 的意义不仅在于方便了音频程序员和音频设计师，对于像“小旭游戏音乐”

这样国内的游戏音乐外包团队来说也是一种福音，在音乐和音效创作之前了解最终这些素材的实现机制，能够更有利于创作并提升最终玩家听觉品质。

说了这么多，难免让人觉得有吹嘘之嫌。其实不然，想要达到很好的效果仍然需要投入精力研究，才能取得实际的成绩。笔者在这本书中所述的只是通过介绍一个新的工具来向读者展示一个新的思路，用来对比以往的旧的思路，从而取得更好的效果。FMOD 也不是万能的，工具总是在进化，随时能跟上最新的技术才是应该有的姿态。

本书作为一本 FMOD 游戏音频工具的使用手册，着重介绍了 FMOD Designer 部分，而对于 FMOD Ex (FMOD 的程序接口部分) 没有介绍。想要了解如何在何程序中使用 FMOD 的读者，还需要查阅相关的资料。以后如果有可能，我们打算对此在后续的出版物中进行介绍。本书对于从来没有了解过音频设计概念的读者来说非常值得一读，无论一个项目的制作人、音频设计师，还是程序员，都有必要了解一下本书的内容。对于设计师来说，阅读本书即可直接了解 FMOD 平台的音频设计全过程；而对于程序员和项目制作人来说，了解使用新工具可以实现什么样的新效果，能解决哪些问题，这更是必须的。因为最终在程序中启用 FMOD 是一个相对以前来说简单很多的工作，所以提前一步了解 FMOD 可以实现什么不是坏事。另外，有兴趣的读者可以阅读 FMOD 安装文件中自带的 FMOD Designer 使用手册，笔者所撰写的很多内容也是参考了该手册。细心的读者会发现，笔者略过了部分关于库、优化、多声道、配置器及互动音乐的部分。这是因为除了篇幅以外，还受制于笔者的工作环境不能有效地进行多声道事件的测试和监听。另外，关于库、优化和配置器的介绍不可避免地需要涉及程序知识，而它们也不是 FMOD 在效果实现上的直接关系部分（但是对于底层来说相当重要），因此在本书中对这个部分没有多做介绍。至于互动音乐系统，它是 FMOD Designer 2010 新增的系统，相对于 FMOD 整体来说比较独立，其中的各种应用也处于开发状态，所以它的相关内容本书也不做介绍。等这个系统更加成熟、稳定了，可以考虑专门写一本书来介绍它。

感谢卢小旭先生！他是我的良师益友。正是因为他的远见才有了这本书的诞生。作为第一次创作技术类出版物的笔者，能够被信任实属不易。希望以后能和小旭先生有更加长远的合作和交流。

感谢“小旭游戏音乐”公司的全体人员！大家对于游戏音频的热情和创造力是我能去钻研这个事物的原动力，正是因为热爱才有了进步，希望大家能再接再厉。

同时需要感谢张曼小姐！在我处于一堆杂乱的草稿状态中时，她提供了有力的 FMOD 手册翻译材料，并参与了相关编写。笔者对于 FMOD 手册中的附录内容头痛不已，但是她帮助我渡过了难关。读者可以在本书附录中阅读到这部分宝贵的内容。

感谢超哥~Tureleon！电话中所述的 FMOD 实现自然枪弹声的样例被我原样用到了第 8 章中，希望那天所描述的问题已经得到顺利解决。

感谢奶妈、大龙和何彦喆！跟你们讨论音效上的实现问题，才有了我对于这套系统的思路和理解。

感谢清华大学出版社的相关编辑！在我拖稿如此严重的状态下给予的包容和理解。

感谢汐岚及我的父母！在我漫长的写书过程中所提供的无私帮助和支持。

感谢我的朋友 Flycat，希望我们可以顺利地展开 FMOD Ex 相关书籍的编写工作。

感谢我的朋友亚菁，为本书写作所提供的帮助。

最后感谢各位读者，希望你们能在本书中找到自己想要的东西。

功能的介绍，到实际应用中的音频设计实例介绍，脉络清晰，一目了然。

7. 问题开放，拓展性强

本书除了讲解基本的知识点外，还给出了部分开放性问题，引导读者自己思考和动手实践，找出解决问题的方案。

8. 提供配套的工程实例

本书提供了书中所涉及的所有实例工程文件供读者下载，需要的读者请到清华大学出版社网站 (<http://www.tup.com.cn>) 上搜索到本书页面后即可下载。

本书内容

第 1、2 章为准备工作，主要向读者展示在新工具的辅助下的声音设计思路、FMOD 的获取方法及 FMOD Designer 程序外观。

第 3 章介绍通用组件的使用。FMOD Designer 系统中很多对象和工具与我们常用的其他音频软件有共同之处，所谓磨刀不误砍柴工，本章主要就对这些通用组件做介绍，为之后的阅读奠定基础。

第 4 章介绍简单事件。简单事件是 FMOD 2010 新提供的一个对象，它实际上是 FMOD 事件对象的一个核心简化版本。麻雀虽小，五脏俱全，它可以实现大量音频工作，但系统开销却非常节省。另外，它也是 FMOD 事件系统和声音定义系统两大核心概念的结合体。

第 5 章对声音定义这一游戏中的全新声音处理单位进行全面介绍。

第 6、7 章介绍多轨事件及其应用，同时还介绍了一些有力的工具用来辅助声音设计工作。多轨事件是 FMOD 事件的精华功能，动态效果、将游戏中的数据与声音动态挂钩等工作都是依赖于多轨事件实现的。

第 8 章介绍了 4 个综合实例的详细实现过程。这 4 个实例都是传统音效制作工作中难以实现的效果，本章将一一实现。

附录介绍了书中前面章节未曾提及的功能和属性，包括快捷键表、命令行参数表和 FAQ 等。作为知识罗列，不再给出具体的实例。

本书读者对象

- ☐ 音频设计师；
- ☐ 游戏制作人和策划人；
- ☐ 游戏开发人员；
- ☐ 音频制作人和爱好者；
- ☐ 游戏音频爱好者。

编著者

前 言

FMOD 音频引擎系统已经是全球使用最多的游戏音频引擎系统。其中,FMOD Designer 为 FMOD 引擎体系中的音频设计模块,它面向的是音频制作者群体。当前,在国内的游戏开发者群体中,FMOD 也逐渐被作为了首选。但国内目前还没有一本相关的图书出版,无论是游戏音频设计师还是爱好者,都急需一本能够指导他们快速上手的书。本书便是基于这个原因而写。

本书以 FMOD Designer 系统为基础,重点介绍音频引擎的基础知识、使用音频引擎进行声音设计的工作流程及以音频引擎为基础的声音制作思路等内容。适合广大音频设计师、游戏策划人员、音乐制作者、游戏开发人员及广大游戏音频设计爱好者阅读。

本书特色

1. 注重概念的引入和思路的讲解

本书对所有涉及的概念都做了较详细的讲解,并且对于解决问题的思路以实例为指导进行了讲解。这对于没有接触过这套工具的读者有非常大的帮助。

2. 图解教学,简单直观

本书对很多不容易理解的过程和概念提供了大量的示意图进行讲解,一目了然、简单直观,可以加深读者对知识的理解,大大节省阅读时间。

3. 实例丰富,注重实践

本书围绕学以致用教学目标,给出了针对每个知识点的小例子及 4 个综合实例,读者通过演练这些例子可以提高自己的实际设计能力,达到较好的学习效果。

4. 语言通俗,风格活泼

本书尽量用最浅显易懂的语言进行讲解,大大降低了读者的学习难度。

5. 需求导向,针对性强

本书讲解时尽可能地用新技术解决已有难题,或者用新技术创作新效果,引导读者先思考后学习,从而形成设计创意的思路,以激发读者兴趣,更有针对性地学习。

6. 脉络清晰,一目了然

本书内容从最开始的 FMOD 工具获取,到基本知识点的介绍,再到较为复杂的对象和

目 录

第 1 章	初识 FMOD 及其开发套件	1
1.1	基本概念介绍	1
1.1.1	引言	1
1.1.2	什么是音频引擎	3
1.1.3	FMOD Ex	4
1.1.4	什么是 FMOD Designer	4
1.2	如何获取	5
1.2.1	下载	5
1.2.2	安装	7
1.3	平台支持	9
1.4	为什么要使用 FMOD/音频引擎	9
1.5	本章小结	11
第 2 章	快速开始	12
2.1	操作界面前置知识	12
2.1.1	重要术语一览	12
2.1.2	通过操作界面来进行工作	14
2.1.3	概览和 workflows	14
2.2	认识操作界面	14
2.2.1	事件视图	15
2.2.2	声音定义视图	19
2.2.3	库视图	20
2.2.4	音乐视图	21
2.3	本章小结	22
第 3 章	通用组件的使用	23
3.1	FMOD Designer 2010 中的通用组件	23
3.1.1	定义你的工作区	23
3.1.2	属性	25
3.1.3	调节设置属性值	25
3.1.4	下拉菜单	27
3.1.5	值域推子	27
3.1.6	不可用的属性	28
3.1.7	属性取值限制	28

3.1.8	通用属性	29
3.1.9	事件 (Events)	31
3.1.10	事件组 (Event Group)	32
3.1.11	事件的基本处理	36
3.1.12	事件模板	39
3.1.13	事件批量编辑器	44
3.1.14	基础多轨事件	49
3.2	本章小结	50
第 4 章	简单事件	51
4.1	引入	51
4.2	简单事件	52
4.3	简单事件实例教程 1: 建立一个鸟鸣氛围事件	58
4.3.1	步骤 1: 创建一个单声音文件的简单事件	59
4.3.2	步骤 2: 向简单事件中添加一个单独的音频文件	60
4.3.3	步骤 3: 为简单事件添加音高随机化效果	61
4.3.4	步骤 4: 为简单事件添加音量衰减变化	63
4.3.5	步骤 5: 向简单事件中添加更多的声音文件	64
4.3.6	定制播放规则	65
4.3.7	步骤 6: 定义简单事件中粒子模式各项取值	68
4.3.8	关于粒子总数	69
4.4	简单事件实例教程 2	70
4.4.1	用简单事件制作更多有趣的东西——使用一个单声音文件 可以创造出多少内容	70
4.4.2	步骤 1: 建立一个单声音的简单事件	72
4.4.3	步骤 2: 向简单事件中添加一个单独的音频文件	72
4.4.4	步骤 3: 设置简单事件的粒子数值	73
4.4.5	步骤 4: 调节简单事件的总体音高	74
4.4.6	步骤 5: 定义简单事件的淡入淡出时间	75
4.4.7	步骤 6: 将简单事件设置为 3D 模式	76
4.4.8	步骤 7: 定义 3D 随机化数值	78
4.4.9	其他事项	79
4.5	简单事件编辑器属性表	79
4.6	本章小结	83
第 5 章	声音定义	84
5.1	声音定义概念介绍	84
5.1.1	打开声音定义界面	85
5.1.2	创建声音定义文件夹	86
5.1.3	创建声音定义	87
5.1.4	选择声音定义	88

5.1.5	将音频文件关联到声音定义中	90
5.1.6	从声音定义中移除声音文件	92
5.1.7	声音定义浏览器：其他附加功能	93
5.1.8	播放列表（Playlists）	96
5.1.9	调节播放列表中文件项的播放比重	97
5.2	声音定义实例教程 1：有限的音频创造丰富的变化	98
5.2.1	步骤 1：建立工程	99
5.2.2	步骤 2：创建声音定义文件夹结构	100
5.2.3	步骤 3：将声音文件关联到声音定义	100
5.2.4	步骤 4：监听第一个例子	102
5.2.5	步骤 5：应用音高随机化	103
5.2.6	步骤 6：应用音量随机化	105
5.2.7	其他事项	107
5.3	声音定义实例教程 2：创造可以永久生成的声音	108
5.3.1	步骤 1：建立和设置工程	108
5.3.2	步骤 2：建立声音定义目录结构	109
5.3.3	步骤 3：将声音文件关联到声音定义当中	110
5.3.4	步骤 4：试听我们的第一个例子	113
5.3.5	步骤 5：设置刷新时间的初始值	113
5.3.6	步骤 6：设置最大复音数属性	115
5.3.7	步骤 7：应用音高随机化	116
5.3.8	步骤 8：应用音量随机化	117
5.3.9	其他事项	118
5.3.10	更进一步的操作	119
5.4	声音定义实例教程 3：创建蛙鸣声、水流声以构建完整环境氛围	121
5.4.1	Sound Def Tutorial 2 事件结构介绍	123
5.4.2	步骤 1：构建 Frogs 声音定义	126
5.4.3	步骤 2：设置 Frogs 声音定义的刷新时间和最大复音数	128
5.4.4	步骤 3：应用音量和音高随机化到 Frogs 声音定义	131
5.4.5	步骤 4：设置 3D 位置随机化	133
5.4.6	步骤 5：构建 Waves 声音定义	137
5.4.7	步骤 6：设置 Waves 声音定义的刷新时间和最大复音数	138
5.4.8	步骤 7：应用音高随机化到 Waves 声音定义	139
5.4.9	步骤 8：设置 3D 位置随机化	141
5.4.10	步骤 9：建立使用 Crickets Single Cycle、Frogs 和 Waves 的多轨事件	142
5.5	声音定义属性归纳	147
5.6	本章小结	150
第 6 章	多轨事件	151
6.1	多轨事件简介	151

6.1.1	打开事件界面	151
6.1.2	选中事件	152
6.1.3	创建事件组	153
6.1.4	创建多轨事件	153
6.2	将声音定义添加到多轨事件中	155
6.2.1	打开声音定义实例属性面板	157
6.2.2	重新调整声音定义实例的大小	158
6.2.3	重新调整声音定义的位置	158
6.3	事件参数	159
6.3.1	事件参数举例	160
6.3.2	自动参数	162
6.3.3	添加自动参数	162
6.4	在多轨事件中添加层	163
6.4.1	在多轨事件中添加层效果器	163
6.4.2	添加效果器自动化包络线的调节点	167
6.4.3	更改效果器自动化包络线的曲线类型	167
6.4.4	移动效果器自动化包络线的调节点	168
6.5	本章小结	170
第 7 章	多轨事件进阶篇	171
7.1	声音定义实例	171
7.1.1	声音定义实例属性	172
7.1.2	声音定义实例属性表	174
7.2	事件参数应用	182
7.2.1	事件参数属性框	183
7.2.2	事件参数属性表	185
7.2.3	速度型参数	190
7.2.4	搜索型参数	194
7.3	事件层和效果器	198
7.3.1	事件层	198
7.3.2	事件层属性	198
7.3.3	效果器	201
7.3.4	效果器使用指南	202
7.3.5	效果器功能介绍	205
7.3.6	属性自动化效果器 (Property Automation Effects)	205
7.3.7	FMOD DSP 效果器	214
7.4	多轨事件属性详解	223
7.5	工程工具	229
7.5.1	工程操作	229
7.5.2	混响定义 (Reverb Defs)	233
7.5.3	事件监听	236

7.6	本章小结	240
第 8 章	实例应用	241
8.1	赛车引擎	241
8.1.1	创建事件结构	241
8.1.2	构建声音定义	242
8.1.3	应用自动变调功能	243
8.1.4	建立空载 (Offload) 层	244
8.1.5	应用音量包络	245
8.1.6	监听事件	246
8.2	模拟距离	246
8.2.1	建立事件结构	247
8.2.2	添加滑离效果	248
8.2.3	监听事件	250
8.3	开枪音效	251
8.3.1	建立事件结构	252
8.3.2	建立声音定义	253
8.3.3	监听事件	253
8.4	子弹时间	254
8.4.1	构造事件结构	255
8.4.2	建立声音定义和声音定义实例	255
8.4.3	设定参数与声音定义实例属性	256
8.4.4	添加效果器及包络线设置	258
8.5	本章小结	259
附录 A	其他属性和功能	260
A.1	其他属性表	260
A.1.1	库界面属性表	260
A.1.2	混响效果的属性表	262
A.1.3	工程属性表	263
A.2	工具参考指南	263
A.2.1	FMOD 引擎设计师 (Engine Designer)	264
A.2.2	3D 监听 (Audition 3D)	264
A.2.3	环绕声场 (Surround Pan)	266
A.2.4	锥形声场设计师 (Cone Designer)	267
A.3	常见问题解答	267
A.4	快捷键指南	269
A.4.1	事件界面	269
A.4.2	事件编辑器页面	270
A.4.3	声音定义界面	271
A.4.4	库界面	271
A.5	命令行 (Command Line) 的作用	271

第 1 章 初识 FMOD 及其开发套件

1.1 基本概念介绍

1.1.1 引言

游戏已经伴随着时代进步，融入到人们的日常生活当中。技术的进步已经使得游戏的载体有了巨大的变化，从最早运行在如同科研设备一样的 PDP-1 游戏机（也许称它为巨型电脑比较好）上的人类史上的第一个电子游戏（图 1-1），到现在地铁、咖啡馆里随处可见的手机游戏（图 1-2），游戏无论是外观、载体还是用户体验，都有了长足的进步。

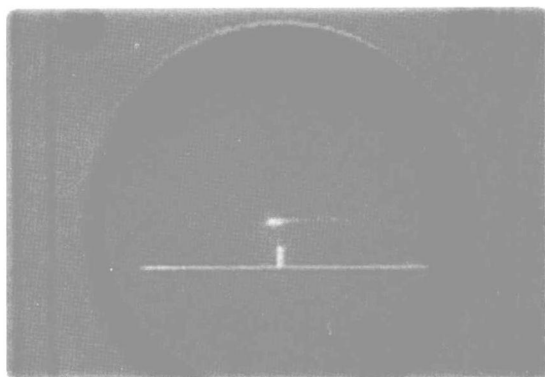


图 1-1 运行在 PDP-1 上的 spacewar



图 1-2 手机游戏

但是似乎游戏的一些关键架构，从一开始就固定了并且很难再发生变化。比如它们都是人为设置的谜题，都需要玩家来进行一些操作而解决这些问题，它们都会通过视觉和听觉的方式来向玩家反馈你的操作是否有效。实际上从 1969 年开始才使用阴极射线管这种示波器一样的粗暴方式让玩家得到视觉反馈，也尝试过使用一些昂贵而笨重的设备为玩家提供声音（一个 Journey 的早期游戏，其音频设备跟现在的一台街机一样大）。而游戏技术的进步，则就是围绕构造更有趣的谜题，让玩家可以进行的操作更刺激，反馈给玩家的信息更能调动玩家的情绪而进行。本书想要讨论的，就是游戏音频上的技术进步以及现在科技所提供的顶峰的游戏音频设计技术。

我们可以做一个简单的归纳并给出一些相应的演示：最早的时期，也就是游戏机这个概念刚诞生不久之后，人们已经开始尝试使用各种模拟设备来为设计出来的游戏添加声音。但是由于设备昂贵、体积过大乃至连台式机主机都无法容纳这些庞大设备，于是只有寥寥数款游戏提供了声音的功能。显然使用模拟设备的方式提供音乐不是明智的行为，这个时

期的游戏音乐也没有构成有规模的设计，于是工程师们就寄希望于新的数字的方式能够解决这一状况。

很快，随着大规模以及超大规模集成电路的高速发展，集成芯片和振荡器被广泛地应用于电子设备中，随着合成器概念的出现，游戏音乐骤然之间开了花。以《吃豆人》（Pac-Man）、《太空入侵者》（Space-Invader）为代表的游戏拉开了游戏音频时代的序幕，如图 1-3、图 1-4 所示。大量地使用芯片来播放声音是这个时期的游戏音频的重要特征，最关键的核心则是，这个时期的音乐几乎都是通过编程实现的。



图 1-3 Pac-Man 吃豆人-Namco



图 1-4 Space Invaders 太空入侵者-Taito

因为知识领域和专业技能训练的限制，在这个时期，受过专业音乐训练的艺术工作者并没有大量地介入实现游戏音乐这个部分的工作，绝大部分游戏音乐都是将已有的音乐名作以编程的方式编写到游戏中，而音效则是各种波形模拟生成的音效。

不久，开发者们就意识到这种做法的局限性。人们毋庸置疑地需要更好的声音反馈，需要更能调动情绪的音乐，也需要更能反馈游戏内状况的音效。数字化合成和采样的概念被逐渐引入到游戏音频制作的工作中，人们可以直接使用预先编程到芯片中的采样和编码来合成声音，而无需使用模拟设备手工合成各种波形，这就极大地开阔了人们对于游戏音频制作的思路和方法，这对后来 MIDI 标准的诞生也是具有极其重要意义的。此期间，诸如 yamaha 等厂商都把自己的声音芯片和 DAC（数模转换器）芯片集成到游戏主机中，乃至提供各种 DSP 功能的著名的 Sony SPC700 芯片也诞生了。其中最著名的游戏主机当属超级任天堂游戏机——SNES，如图 1-5 所示。超任上脍炙人口游戏的非常丰富，这里就不一一举例。

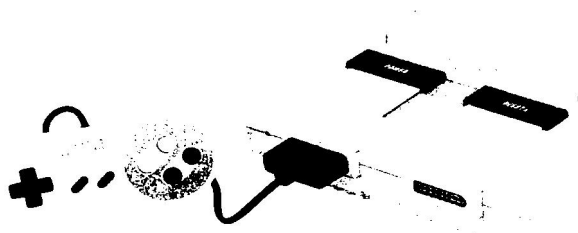


图 1-5 超任主机，自带音色芯片

从这个时代开始，音乐音频数据也就开始大量地使用外部数据，而不是通过编程的方式集成在芯片或者游戏代码里。专门以解码和提供 DSP 功能的芯片比如 Spc-700 也出现了。至此，游戏音频制作正式进入了程序员与音乐人分工合作的时代。这个思路也被广泛应用到当时爆发性增长的 PC 市场中。MIDI 标准在这时诞生了，虽然这种技术本来为游戏而生，但是却在音乐制作领域得到大量应用。游戏音频的实现方式逐渐演变成了一个新的模式：音乐人/音频设计师使用自己的设备制作合适的声响，而程序员们使用各种技术将它们结合到游戏内部。而之后的技术进化的主流方向，就变成了：如何制作更合适的声音，以及如何让它们与游戏结合得更自然，更能给玩家真实感，并且能激发玩家的情绪。

当然，本书的目的并不是为了讨论如何能在音乐上表达得更好，我们也不会去探讨录音、混音等声学上的基本知识概念。我们着重要讨论的就是上述提出的问题：如何制作更合适的声音，如何让它们与游戏结合得更自然，给玩家真实感，并且能激发玩家的情绪。实际上，自从音乐制作人开始直接接受游戏音频制作的工作以来，上述问题已经成为了这个领域第一重要的课题。杜比技术诞生了，软件合成器、采样器技术也诞生了，数字音频设备已经功能强大到无以复加，游戏的画面操作感也有了极大的提升，游戏性已经不可同日而语。于是将音频与游戏系统结合到一起的技术诞生了，它们被称为音频引擎。

音频引擎其实也并不是突然出现的，它结合了我们上述的多个时代中各种声音回放方式的种种特点，利用这个时代里强大的 CPU 运算能力与面向对象的开发方式进化而来。它的初期，也只是程序员们为了调用外部音频文件撰写的具有各种音频功能的功能函数库，逐渐演变下来，它们集成为了一个整体的系统。其中早期比较有代表性的，只能称其为接口不能称其为引擎的程序标准，包括 DirectMusic、OpenAL 等。这些程序接口的一大特点就是，你可以调用音乐人制作好的各种声音，如何去触发如何调用，这些规则必须有程序员全盘把握，在程序员没有声学经验的情况下，只能凭感觉为山洞里面的音效加上混响，为山谷里面的声音加上回声。而对于音频工作者就更头疼了，有些时候他们根本不知道自己设计的声音最后到底会怎样被使用。如果他们想控制声音最终使用的方式，他们就不得不去熟悉程序的开发技巧，这对于他们精于音乐或者声学上数十年的经验无疑是一种可怕的浪费。而真正意义上的音频引擎，是为了解决这一困境而诞生的，一个整体的解决方案，如同解决了美工和程序员们大麻烦的图形引擎及其图形引擎一般。我们将要介绍的，就是目前世界范围内使用最为广泛的音频引擎：FMOD 的一些相关技术知识。

1.1.2 什么是音频引擎

从技术上来说，到底我们怎么去解释音频引擎这一概念呢？我们以 FMOD 为例，来说一下一般音频引擎的结构。

首先整体上说，音频引擎就是一个可以在不同操作系统下回放不同格式编码的声音数据的音频调用函数库。通过音频引擎，你可以在游戏或者软件中实现各种音频功能。比如为声音增加 DSP，按你的期望组织播放声音等。而 FMOD 音频引擎包含 3 个部分：

- ❑ FMOD Ex：底层的音频引擎——提供直接代码库函数的各种 API，一般内嵌到游戏中。
- ❑ FMOD 事件系统：一个位于应用层的抽象概念，实际上就是你所建立的各种声音回放方式与规则的处理系统，Ex 通过事件系统的指挥来调度音频内容。事件实际