

资深手游开发工程师倾情奉献，独家揭秘数十项开发技巧
跟随资深工程师探索接近商业项目的手游开发全过程



Game Design and Develop

游戏设计与开发



libGDX

移动游戏开发 从入门到精通

黄俊东 著



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



游戏设计与开发



libGDX

移动游戏开发

从入门到精通

黄俊东 著

人民邮电出版社
北京

图书在版编目 (CIP) 数据

libGDX移动游戏开发从入门到精通 / 黄俊东著. --
北京: 人民邮电出版社, 2016.1
ISBN 978-7-115-40799-3

I. ①1… II. ①黄… III. ①移动终端—游戏程序—
程序设计 IV. ①TN929.53②TP311.5

中国版本图书馆CIP数据核字(2015)第277540号

内 容 提 要

随着移动互联网的快速发展,移动端的游戏开发已经成为发展迅速、市场潜力巨大、前景诱人的开发方向。作为一种优秀的游戏引擎,libGDX让更多梦想加入游戏开发的人圆梦。本书全面介绍了使用libGDX进行游戏开发时所需要用到的知识,其中包括libGDX的详细介绍、libGDX的UI编程、libGDX的配套开发工具、libGDX中所遇到的一些问题的解决方案。

本书并不局限于介绍libGDX编程的各种理论知识,而是从“项目驱动”的角度来讲授理论,全书给出了丰富的实例,这些示范性的实例既可帮助读者更好地理解各知识点在实际开发中的应用,也可供读者在实际开发时作为参考。本书最后还提供了两个实用的案例:捡金币项目和2048项目,具有很高的参考价值。

本书实用性强,内容详细,不仅适合有一定经验的开发者,对于那些没有经验的开发者来说,也是一本很好的入门指南。

-
- ◆ 著 黄俊东
责任编辑 陈冀康
责任印制 张佳莹 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 19
字数: 387千字
印数: 1—2500册
- 2016年1月第1版
2016年1月河北第1次印刷
-

定价: 49.00元

读者服务热线: (010)81055410 印装质量热线: (010)81055316
反盗版热线: (010)81055315

写作初衷

从创新工场-涂鸦移动离职以后，我一直想把自己所学到的东西总结一下，以帮助更多想走上或者准备走上游戏开发这条道路上的人。很庆幸地接到了出版社的约稿，便愉快地决定要把自己一路走来所学到的知识做一个系统的总结，并尝试用一种更容易理解的方式把它们讲出来，以让更多的读者少走一些弯路，可以在尽可能短的时间内学到尽可能多的知识。

当我刚开始学 libGDX 游戏引擎的时候，对于很多自学的人来说，我应该是很幸运的了。因为当时身边有很多经验很丰富的同事可以请教。虽然这样，依然感觉到 libGDX 的教程少之又少，学习起来困难重重。整个过程大部分都是以一个初学者的水平一点一点从那晦涩难懂的英文 libGDX 官方教程中学习。所以当自己有能力去做些什么的时候，我很乐意用自己微薄的能力去为后来的学习者做些什么。这就是我写这本书的初衷。

作为一个优秀的游戏引擎，libGDX 不知不觉也走过了好几个年头。从一开始的一个不怎么起眼的小项目发展到现在拥有众多使用者的项目，从第一个 bug 不少的版本发展到较为稳定的 0.9x 版本，再到后来的 1.x 版本，开发者们能够真真切切地感受到 libGDX 的每一步成长。

作为一个游戏引擎，libGDX 支持使用 Java 语言进行游戏开发，这让众多爱好游戏开发，但又苦于使用门槛过高的其他语言进行游戏开发的开发者看到了希望。不仅如此，libGDX 还兼容多种平台，包括 Android、iOS、Html5、PC。作为一个游戏引擎，libGDX 让使用 Android 原生进行开发的开发者们从 SurfaceView 中解脱出来，开发效率得到极大的提高。

本书特色

1. 内容实用、详略得当，符合初学者的认知规律

本书内容涵盖了使用 libGDX 进行游戏开发所必须掌握的知识，其中包

包括但不限于：libGDX 的架构、libGDX UI 编程、libGDX 配套的开发工具、libGDX 在一些问题上的解决方案。从内容结构上非常注重知识的实用性和可操作性。必须掌握的细节处绝不吝惜笔墨、甚至细致到每一次的鼠标单击；仅需要大致了解的内容绝不铺张浪费纸张、整体结构的描述提纲挈领。这样的安排注重了只介绍初学阶段必备知识的深入了解，需要大致了解的知识也能够有所认识，这种由浅入深、循序渐进地讲授遵循和尊重初学者对知识的认知规律。

2. 实例驱动，特别适合初学者阅读

本书每介绍一个基本知识点，都会通过一个难度适中的实例来帮助读者加深对该知识点的理解，从而能够举一反三。而且，每一个实例作者都会配以详细的注释，帮助读者更好地理解这个实例中每一行代码的意义，从而达到“知其然”，更要“知其所以然”的目标。

3. 项目接近真实的商业开发

本书中所介绍的两个项目存在于真实的商业开发中。其中捡金币的游戏模型应用广泛。2014 年百度在圣诞节期间的一些推广活动就是嵌入到这种模式的游戏实现的。而其中的 2048 游戏更是在各大应用商店中广受好评。

本书内容及体系结构

第 1 章 libGDX 简介

本章对 libGDX 游戏引擎进行了较为详细的介绍。包括 libGDX 的历史、功能、特点以及 libGDX 都有哪些配套的开发工具。让读者对 libGDX 有一个感性的认识，为后面的章节打下基础。

第 2 章 libGDX 的架构分析

本章详细介绍了 libGDX 的生命周期、模块组成、libGDX 中可以使用的输出日志的方式以及一些其他的辅助工具。通过本章的学习，读者对使用 libGDX 进行游戏开发时所编写的代码架构有了较为直观的认知，清楚每一个生命周期函数中都应该完成哪些操作。

第 3 章 数据存取

本章详细介绍了使用 libGDX 进行游戏开发时所需要用到的一些游戏的数据存取的一些方式。这一章知识的讲解不局限于 libGDX，也详细介绍了 Android 中的操作方式。因为在开发的时候，这两种经常会配合着使用。

第 4 章 libGDX UI 中的编程

本章详细介绍了使用 libGDX 来进行 UI 编程时所需要的知识。其中包括对 SpriteBatch、BitmapFont 基础类的详细介绍。这些知识有利于更好地理解和掌握本书第 5 章的内容。毕竟本章的知识其实是第 5 章所介绍到的内容的底层实现。

第 5 章 libGDX 中的常见 UI 组件

本章是站在控件的高度去讲解 UI 编程。其中所涉及的控件包括 Image、Button、Slider、

SplitPane 等常见又实用的控件。熟练掌握这些控件的使用,能让我们的开发更有效率,也让我们能够编写出符合我们项目需求的控件。

第 6 章 libGDX 中的事件处理与音效

本章详细介绍了 libGDX 中的事件处理与音效。事件处理是整个游戏的基础,它能让游戏从一个状态切换到另外一个状态。而音效更是能为一个游戏加分不少。通过本章的学习,读者将掌握更上一层楼的技能与技巧。

第 7 章 资源加载器、游戏界面的跳转以及物理引擎

本章详细介绍了资源加载器、游戏界面的跳转以及 libGDX 中的物理引擎。资源加载器所对应的异步加载方式常常被运用到大型游戏中。游戏界面的跳转是一个游戏中状态切换自然而然的效果。而物理引擎则在体育类游戏中随处可见。

第 8 章 libGDX 中的 UI 编辑器

本章详细介绍了如何让 CocosStudio 这个 UI 编辑器来为 libGDX 服务,从而弥补 libGDX 在 UI 编辑器上的不足。通过本章的学习能够让读者比较轻松地处理游戏元素较多的情况。

第 9 章 小游戏实战——捡金币

本章详细介绍了如何使用本书中所介绍到的知识去开发出一个具有较强趣味性的游戏——捡金币,该游戏模式被广泛应用各类的网页推广中。

第 10 章 休闲类游戏——2048

本章详细介绍了一个成熟的休闲类游戏——2048 的开发。首先分析了整个项目的组成,让读者对整个游戏的架构有一个清晰的认识。接下来把这个游戏中所涉及的每一部分都在详细的介绍中实现,并在最后提出了一些优化与改进的方向。

本书读者对象

- ◇ libGDX 初级及中级开发者,了解 Java 语言和游戏开发的读者。
- ◇ libGDX 零基础,但具有一定的 SurfaceView 开发经验的工程师。
- ◇ 游戏开发零基础,但爱好游戏开发的工程师。
- ◇ 爱好游戏开发的大学生。

致谢

感谢南京邮电大学物联网学院院长孙知信教授在我编写本书时所给予的鼓励,也感谢给予过我很多帮助的同事和朋友们。

本书在编写过程中,我虽然具备一些开发经验,但是水平有限,书中不免有错误之处,望读者批评指正。

第 1 章 libGDX 简介 1

1.1 libGDX 介绍 1

1.1.1 libGDX 的特点 1

1.1.2 libGDX 中的工具 3

1.1.3 libGDX 的历史 5

1.1.4 我的 libGDX 版本 5

1.2 搭建 libGDX 开发环境 5

1.2.1 libGDX 下载 6

1.2.2 Android 项目的新建 7

1.2.3 libGDX 的安装 9

1.3 开始第一个 libGDX 游戏 11

1.3.1 第一个 HelloWorld 程序的编写 11

1.3.2 在 Android 设备上调试 15

1.3.3 桌面调试环境的搭建 17

1.3.4 HelloWorld 程序的简单分析 22

1.4 运行 libGDX 的官方 Demo 23

1.4.1 libGDX 的官网 23

1.4.2 Demo 的下载演示 24

1.5 其他的一些准备工作 25

1.5.1 关联 Android 源码 25

1.5.2 libGDX 源码的关联 28

1.5.3 直接打开 .jar 文件的配置 30

第 2 章 libGDX 的架构分析 34

2.1 生命周期 34

2.1.1 libGDX 中生命周期的图示 34

2.1.2 生命周期中各个函数的分析 35

2.1.3 实例：通过例子深刻理解生命
周期 35

2.2 模块 38

2.2.1 libGDX 模块组成的图示 38

2.2.2 libGDX 模块的分析 39

2.3 运行环境的相关查询 40

2.3.1 获取目前的运行环境 40

2.3.2 获取内存的消耗情况 41

2.4 日志工具 41

2.4.1 Android 中的日志等级及输出方法 41

2.4.2 libGDX 中的日志等级及输出方法 42

2.4.3 一种方便的输出日志的方法 44

第 3 章 数据存储 45

3.1 简介 45

3.2 Android 中 SharedPreferences 的使用 45

3.2.1 获得一个 SharedPreferences 对象 46

3.2.2 从 SharedPreferences 获取数据 46

3.2.3 往 SharedPreferences 保存数据 46

3.2.4 实例：使用 Android 中的
SharedPreferences 来实现
存取数据 47

3.3 libGDX 对 SharedPreferences 的支持 49

3.3.1 获得一个 SharedPreferences 数据 49

3.3.2 从 SharedPreferences 获取数据 49

3.3.3 往 SharedPreferences 保存数据 49

3.3.4 实例：使用 libGDX 对 SharedPreferences 的支持 来实现数据的存取.....	50	3.6.7 Android 中使用 FastJson 来解析 JSON 形式的数据.....	81
3.4 Android 中操作 XML 形式的数据.....	51	3.6.8 Android 中使用 FastJson 来生成 JSON 形式的数据.....	83
3.4.1 XML 的定义.....	51	3.6.9 以上 3 种操作 JSON 格式数据的 方法的简单比较.....	83
3.4.2 XML 的优缺点.....	52	3.7 libGDX 中对 JSON 格式数据的支持.....	83
3.4.3 Android 中使用 DOM 来解析 XML 形式的数据.....	52	3.7.1 使用 libGDX 中的 API 生成 JSON 形式的数据.....	83
3.4.4 Android 中使用 DOM 来生成 XML 形式的数据.....	55	3.7.2 使用 libGDX 中的 API 解析 JSON 形式的数据.....	86
3.4.5 Android 中使用 SAX 来解析 XML 形式的数据.....	58	3.8 XML 与 JSON 的简单比较.....	86
3.4.6 Android 中使用 SAX 来生成 XML 形式的数据.....	61	3.8.1 关于轻量级和重量级.....	86
3.4.7 Android 中使用 PULL 来解析 XML 形式的数据.....	64	3.8.2 关于数据格式编码及解析难度.....	87
3.4.8 Android 中使用 PULL 来生成 XML 形式的数据.....	67	3.9 libGDX 中的文件处理.....	87
3.5 libGDX 对 XML 的支持.....	70	3.9.1 几种文件存储路径.....	88
3.5.1 使用 libGDX 中的接口将生成 XML 形式的数据.....	70	3.9.2 检查存储路径是否可用.....	88
3.5.2 使用 libGDX 中的接口解析 XML 形式的数据.....	72	3.9.3 获得一个 FileHandle 实例.....	89
3.6 Android 中操作 JSON 格式的数据.....	74	3.9.4 对 FileHandle 的常见操作.....	89
3.6.1 JSON 的定义.....	74	3.9.5 读取文件.....	90
3.6.2 JSON 的优缺点.....	74	3.9.6 写文件.....	90
3.6.3 Android 中 JSON 的原始解析 方法.....	75	3.9.7 使用 FileHandle 来删除、拷贝、 移动、重命名文件.....	90
3.6.4 Android 中 JSON 的原始生成 方法.....	78	第 4 章 libGDX 中的 UI 编程.....	92
3.6.5 Android 中使用 Gson 来解析 JSON 形式的数据.....	79	4.1 绘制的基础——SpriteBatch 类的介绍.....	92
3.6.6 Android 中使用 Gson 来生成 JSON 形式的数据.....	80	4.1.1 SpriteBatch 的功能.....	92
		4.1.2 基本 API.....	92
		4.2 中文字符的显示——hiero 工具的 介绍与基本使用.....	92
		4.2.1 hiero 工具的下载.....	93
		4.2.2 hiero 的功能界面剖析.....	93
		4.2.3 利用 hiero 生成程序中要用的 字体文件.....	94

4.3 文本信息的显示——BitmapFont 类的 基本使用	95	4.8.3 利用 TexturePacker 进行合图	109
4.3.1 BitmapFont 的功能	95	4.9 libGDX 中的合图 (2) ——TextureAtlas 类的介绍与基本使用	111
4.3.2 基本 API	95	4.9.1 功能	111
4.3.3 实例: 使用 BitmapFont 来显示 一段文本信息	95	4.9.2 基本 API	111
4.4 自定义的中文字体的显示 ——TTF 字库的介绍与使用	97	4.9.3 实例: 使用 TextureAtlas 后的 图片的使用	111
4.4.1 功能	97	4.10 自定义动画——Animation 类的介绍 与基本使用	113
4.4.2 基本 API	97	4.10.1 功能	113
4.4.3 使用优美字体前的准备	98	4.10.2 基本 API	113
4.4.4 实例: 显示一段帅气的文本	98	4.10.3 实例: 奔跑的小人	114
4.5 图片的显示 (1) ——Texture 类的介绍 及基本使用	100	4.11 libGDX 中的粒子编辑器 (1) ——Particle-Editor 的介绍与基本使用	116
4.5.1 功能	100	4.11.1 下载	116
4.5.2 基本 API	100	4.11.2 功能界面剖析	117
4.5.3 实例: 使用 SpriteBatch 和 Texture 在屏幕中显示一张图片	101	4.11.3 利用粒子编辑器生成相应文件	118
4.6 图片的显示 (2) ——TextureRegion 类的介绍与基本使用	103	4.12 libGDX 中的粒子编辑器 (2) ——ParticleEffect 类的介绍与基本 使用	119
4.6.1 功能	103	4.12.1 功能	119
4.6.2 基本 API	103	4.12.2 基本 API	119
4.6.3 实例: 使用 SpriteBatch 和 TextureRegion 在屏幕中显示 一张图片	104	4.12.3 实例: 类似于微信的烟火效果	119
4.7 图片的显示 (3) ——Sprite 类的介绍 与基本使用	105	第 5 章 libGDX 中的常见 UI 组件	122
4.7.1 功能	105	5.1 演员与舞台的简单介绍	122
4.7.2 基本 API	105	5.2 文本信息的显示——Label、LabelStyle 类 的介绍与基本使用	122
4.7.3 实例: 旋转的精灵	106	5.2.1 功能	122
4.8 libGDX 中的合图 (1) ——合图工具 TexturePacker 的介绍与基本使用	107	5.2.2 基本 API	122
4.8.1 TexturePacker 的下载	108	5.2.3 实例: 用 Label 显示一段文本 信息	123
4.8.2 TexturePacker 功能界面剖析	108	5.3 图片的显示——Image 类的介绍与 基本使用	124

5.3.1 功能	124	5.8.5 组合动画——顺序动画与并行 动画	145
5.3.2 基本 API	125	5.8.6 实例: Tween 动画的综合例子	146
5.3.3 实例: 实现一个“旋转的飞机”	125	5.9 按钮——Button 类的介绍与基本使用	148
5.4 Image 生成的另外一种方式——Pixmap 类的介绍与基本使用	126	5.9.1 功能	148
5.4.1 功能	127	5.9.2 基本 API	148
5.4.2 基本 API	127	5.9.3 实例: 使用 Button 来控制一个 loading 图片	149
5.4.3 实例: 使用 Pixmap 来画多种 几何图形	127	5.10 游戏摇杆——TouchPad 的介绍与 基本使用	153
5.4.4 实例: 使用 Pixmap 来生成 涂抹功能	129	5.10.1 功能	153
5.4.5 Pixmap 在使用时需要注意的 问题——白屏问题	133	5.10.2 基本 API	154
5.5 图片翻转的两种方式	134	5.10.3 实例: 使用游戏摇杆控制 游戏元素	155
5.5.1 图片翻转的第一种方式—— 使用 TextureRegion 的 flip()	134	5.11 滑块——Slider 的介绍与基本使用	159
5.5.2 图片翻转的第二种方式—— 通过 SpriteBatch 的 draw() 方法	135	5.11.1 功能	159
5.6 滚动条——ScrollPane 的介绍与基本 使用	137	5.11.2 基本 API	160
5.6.1 功能	137	5.11.3 实例: 使用滑块 Slider 来控制 游戏元素的缩放	161
5.6.2 基本 API	137	5.12 分割窗口——SplitPane 的介绍与 基本使用	165
5.6.3 实例: 使用 ScrollPane 实现一个 选关界面	137	5.12.1 功能	166
5.7 游戏中的背景移动	139	5.12.2 基本 API	166
5.7.1 原理分析	139	5.12.3 实例: 使用分割窗口 SplitPane 来实现一个分割窗口	167
5.7.2 实现天天打飞机中的“背景 移动”	141	5.13 复选框——CheckBox 的介绍与基本 使用	170
5.8 演员的常见动画的介绍与基本使用	143	5.13.1 功能	171
5.8.1 旋转动画——rotateTo 与 rotateBy	144	5.13.2 基本 API	171
5.8.2 位移动画——moveTo 与 moveBy	144	5.13.3 实例: 使用复选框 CheckBox 来控制画面的切换	172
5.8.3 透明度渐变动画——fadeIn() 与 fadeOut()	145	5.14 列表——List 的介绍与基本使用	176
5.8.4 缩放动画——scaleTo() 与 scaleBy()	145	5.14.1 功能	176
		5.14.2 基本 API	177

5.14.3 实例：使用列表 List 来显示 内容	178	7.1.1 功能	199
5.15 文本输入框——TextField 的介绍与 基本使用	181	7.1.2 基本 API	200
5.15.1 功能	181	7.1.3 实例：游戏中的 loading 界面 第一部分	200
5.15.2 基本 API	181	7.1.4 实例：游戏中的 loading 界面 第二部分	203
5.15.3 实例：使用文本输入框 TextField 来实现文本信息的输入	183	7.2 游戏界面的跳转	204
第 6 章 libGDX 中的事件处理与音效	188	7.2.1 第一种方式——隐藏其中一个， 显示另外一个	204
6.1 InputProcessor 的介绍与基本使用	188	7.2.2 第二种方式——setScreen()	206
6.1.1 InputMultiplexer 的介绍与基本 使用	189	7.3 libGDX 中的物理引擎——Box2D 基础	209
6.1.2 实例：使用按钮来操控滚动条	190	7.3.1 基础类概览	209
6.2 振动器的介绍与使用	192	7.3.2 基础类的详细介绍	210
6.2.1 准备工作——添加相应的权限	192	7.3.3 实例：下降的小球与砖块	213
6.2.2 基本 API	193	第 8 章 libGDX 中的 UI 编辑器	216
6.2.3 实例：按按钮触发振动	193	8.1 让 CocosStudio 成为 libGDX 的 UI 编辑器	216
6.3 back 键和 menu 键的捕捉	194	8.2 CocosStudio 的 UI 编辑器的基本 使用	217
6.4 短音频文件的处理——Sound 类的介绍 与基本使用	195	8.2.1 新建项目	217
6.4.1 功能	195	8.2.2 调整屏幕大小	218
6.4.2 基本 API	195	8.2.3 导入资源	219
6.4.3 实例：给按钮添加单击声音	195	8.2.4 摆放素材	219
6.5 长音频文件的处理——Music 类的介绍 与基本使用	196	8.2.5 设置元素属性	220
6.5.1 功能	197	8.2.6 修改元素的名字	221
6.5.2 基本 API	197	8.2.7 保存项目	221
6.5.3 实例：单击某一个按钮播放音乐	197	8.3 解析 CocosStudio 的 UI 文件	221
第 7 章 资源加载器、游戏界面的跳转以及 物理引擎	199	8.3.1 找到数据文件	221
7.1 资源加载器——AssetManager 类的介绍 与基本使用	199	8.3.2 简单分析数据文件	222
		8.3.3 编写代码解析 JSON 文件	223
		8.3.4 CocosStudio 解析工具类的基本 用法	228

8.3.5 让 CocosStudio UI 编辑器的数据 在 libGDX 中完美运行.....	229	10.2.1 游戏的策划	247
8.4 CocosStudio 生成的 JSON 文件中的 隐患及解决方案.....	231	10.2.2 Android 平台下游戏的准备工作	247
8.4.1 隐患分析	231	10.3 游戏的架构	248
8.4.2 解决方案	231	10.3.1 游戏中用到的主要技术.....	248
第 9 章 小游戏实战——捡金币	232	10.3.2 各类的简要介绍	248
9.1 游戏的感性认识	232	10.3.3 游戏的框架简介	249
9.1.1 游戏运行效果图	232	10.3.4 游戏中核心逻辑的几个要点.....	250
9.1.2 组成分析	233	10.4 游戏程序入口的设计与实现.....	251
9.2 金币、背景的实现的原理分析.....	233	10.4.1 MainActivity 的实现	251
9.3 游戏成就的原理分析及游戏中的 碰撞检测	233	10.4.2 MyGame 类的实现	252
9.3.1 检测的简单介绍	234	10.5 游戏主界面 GameScreen 的设计与 实现	253
9.3.2 游戏成就的原理分析.....	234	10.5.1 游戏主界面的框架设计.....	253
9.4 成功界面与失败界面的原理分析.....	234	10.5.2 游戏主界面 GameScreen 的 实现	257
9.5 剩余时间的原理分析.....	235	10.6 游戏中所涉及的各个逻辑体.....	270
9.6 代码实现	235	10.6.1 资源管理类 Assets 的设计	270
9.6.1 金币(Jinbi) 逻辑体的实现	235	10.6.2 资源管理类 Assets 的实现	271
9.6.2 工具类(Utils)的实现	237	10.6.3 游戏中的常量类 Contants	272
9.6.3 背景(BgImage)逻辑体的实现	237	10.6.4 游戏中分数逻辑体 Score 类的 设计与实现	273
9.6.4 游戏主逻辑体 MyGame.....	239	10.6.5 每种分值对应的颜色的逻辑体 Square 的设计	274
9.6.5 Android 与 libGDX 交互类 MainActivity 的编写	244	10.6.6 Square 类的实现	275
第 10 章 休闲类游戏——2048	245	10.6.7 分数格子逻辑体 Cell 类的设计	279
10.1 2048 的由来及功能概述.....	245	10.6.8 分数格子逻辑体 Cell 类的实现	280
10.1.1 2048 的由来	245	10.6.9 数据存储与恢复逻辑体 Prefs 的设计	285
10.1.2 功能简介	245	10.6.10 数据存储与恢复逻辑体 Prefs 的实现	285
10.2 游戏的策划及准备工作.....	247	10.7 游戏的优化与改进	289

第 1 章 libGDX 简介

目前，手游在整个游戏的市场份额上所占的比例越来越大。腾讯系的手游更是一度占领了苹果 App Store 前十名超半数的席位。现在就让我们走进 libGDX 这个游戏引擎的世界，开发出一款属于自己的游戏。

1.1 libGDX 介绍

在本节中，主要简单介绍一下 libGDX 这个游戏引擎，包括其特点、配套的开发工具等，让读者对 libGDX 这个游戏引擎有一个感性的认识。

1.1.1 libGDX 的特点

libGDX 是一个跨平台的 2D/3D 游戏开发框架，它由 Java/C/C++ 语言编写而成。它基于 Apache LicenseVersion 2.0 协议，对商业使用和非商业使用均免费，代码托管于 Github 中，最新版本为 1.5.4。开源的游戏框架并不少见，但是 libGDX 的优势是相当明显的，主要是体现在以下几个方面：

1. 强兼容性

libGDX 兼容多种平台系统(Windows、Linux、Max OS X、Java Applet、Javascript/WebGL)，包括移动系统 Android (1.5 版本+) 和 iOS。在 Android 阵营和相关市场日益扩大的情况下，对于 Android 平台的支持使 libGDX 的使用者越来越多。另外，基于 RoboVM 可以实现 iOS 兼容。

极强的兼容性为调试和开发提供了便利。你可以使用 Android 上面同样的代码在桌面 PC 上面编写、测试、调试你的应用（也就是说，你不需要打开 Android 模拟器，就可以调试用 libGDX 引擎写的游戏应用）。它剥离了常见的 Windows 之间/ Linux 应用程序和一个 Android 应用程序的区别。一般的开发过程是尽可能地停留在桌面 PC 上，同时周期性地检查你的当

前代码是否仍然能在 Android 上运行。

2. 高效性

libGDX 主要是用 Java 写的，其中也掺杂了一些 C/C++ 代码，这些代码是为了处理一些对性能要求很高的操作，比如物理引擎或者音频处理。作为用户，你只需要关注 Java 的封装就可以了，它已经把所有的本地代码封装好了。相比于其他 android 游戏引擎，libGDX 的效率优势十分明显。

3. 架构清晰

libGDX 的主要构成，如图 1.1 所示。

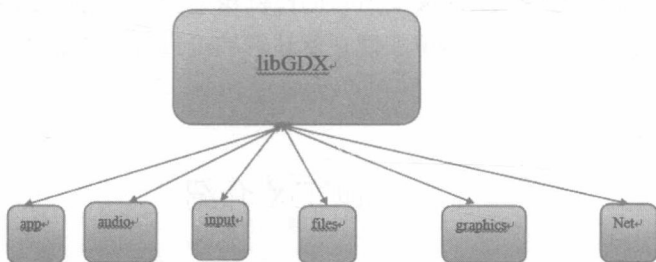


图 1.1 libGDX 的主要构成

libGDX 由 audio、files、graphics、math、physics、scenes、utils 这些主要模块所组成，它们对应了 libGDX 中的常用函数、接口、音频操作、输入/输出、文件操作，2D/3D 渲染及 libGDX 绘图相关运算、网络模块。

它提供了便于设计游戏的清晰架构，如图 1.2 所示。

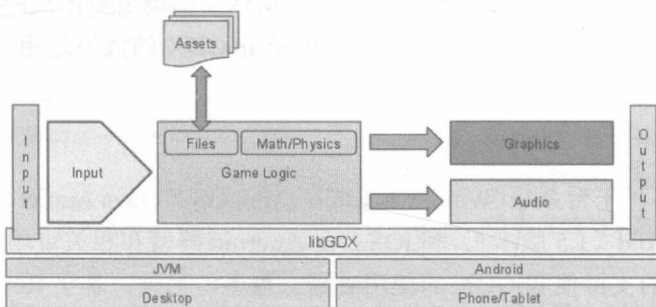


图 1.2 libGDX 的架构

4. 封装

libGDX 对于物理引擎的封装也是让人惊讶的。它使用 jni 封装了 box2d 的 C++ 版本，使得其运行效率比其他同级的物理引擎如 jbox2d 更快。现在流行的几个包含物理引擎的

Android 游戏引擎（如 Andengine、Rokon 等），几乎都在用 libGDX 所封装的物理引擎。如果你的游戏（特别是针对 Android 平台的）准备使用物理引擎，请优先考虑 libGDX。

1.1.2 libGDX 中的工具

libGDX 具有较为完善的配套的开发工具，如粒子编辑器（Particle editor）、文字生成工具（Hiero bitmap font generator）、图片合并工具（Texture packer）等。这些所涉及的工具都会在以后的章节中陆续介绍。现在主要是贴出其软件界面，让大家先对其有一个直观的印象。

1. 粒子编辑器（Particle editor）

粒子编辑器一个很明显的作用就是可以给游戏的画面添加更炫丽的效果。无论是大型游戏还是中小型游戏，基本都会有“每日登录”这一模块，而这一模块中，就经常使用到粒子效果，如图 1.3 所示。

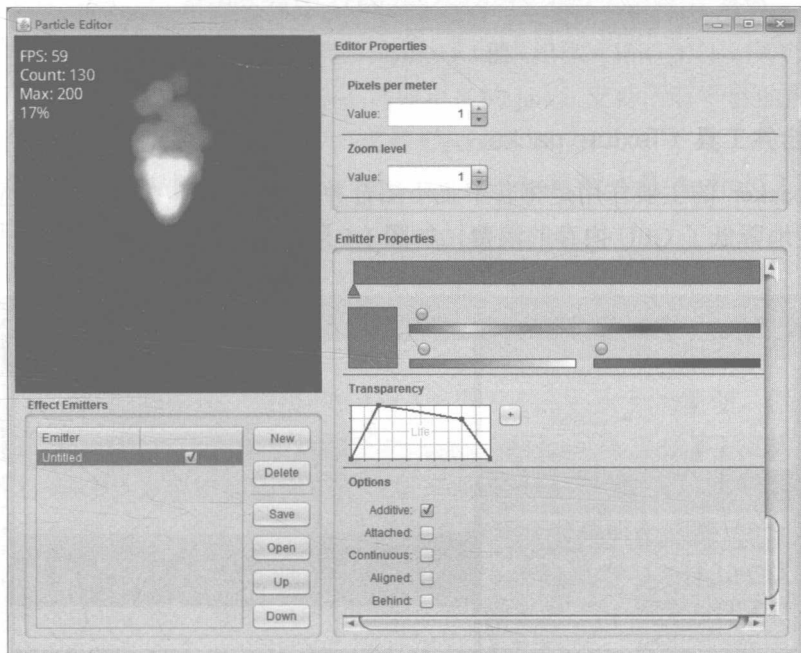


图 1.3 粒子编辑器的界面

2. 文字生成工具（Hiero bitmap font generator）

借助 Hiero bitmap font generator，可以生成游戏中需要用到的字体文件。在显示游戏中的文字信息时，通常需要用该字体文件。文字生成工具如图 1.4 所示。

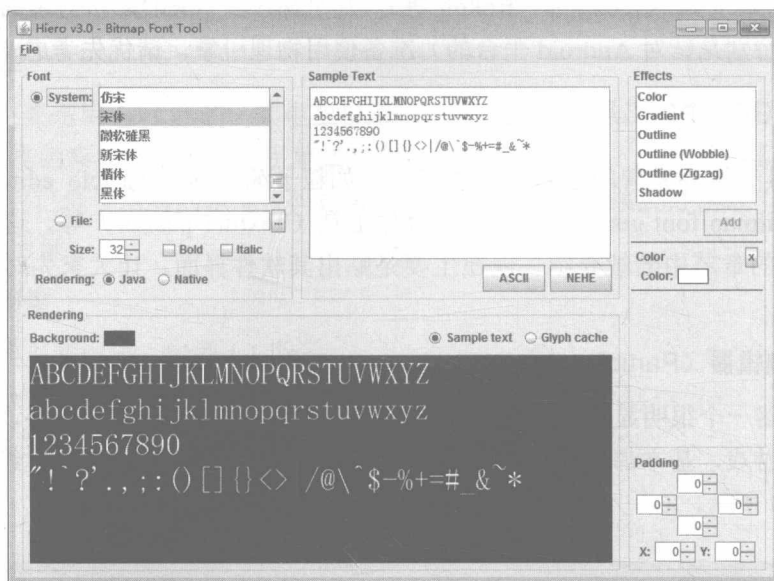


图 1.4 Hiero 的界面

3. 图片合并工具 (Texture packer)

Texture packer 的主要作用是将若干张小图合并为一张大图。这样在方便资源管理的同时,也尽可能地降低了 GPU 内存的浪费,如图 1.5 所示。

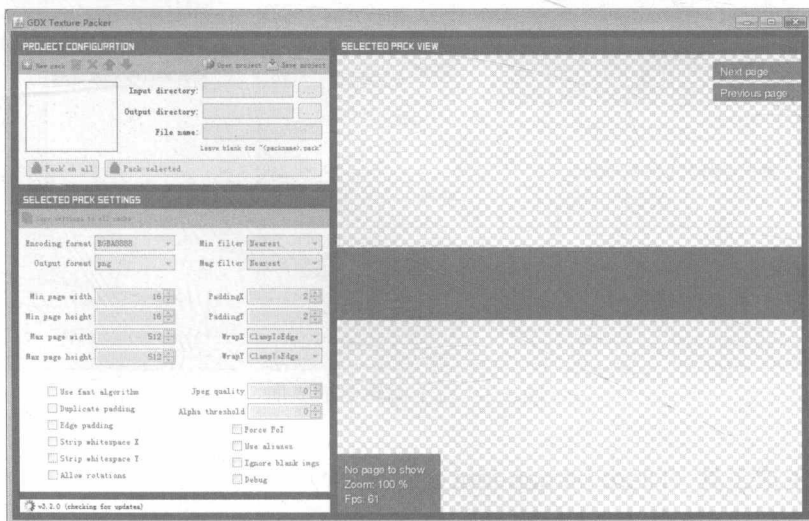


图 1.5 Texture packer 合并工具

1.1.3 libGDX 的历史

2009 年中期，项目开始，初始名称为 AFX。

2010 年 3 月 6 日，项目开源，代码托管在 Google Code 上。

2012 年 5 月，市场占有率超过 1.24%，超过了当时的 Unity、AndEngine、Cocos2D、Corona 和 Marmelade。

2012 年，Google Ingress 使用了 libGDX 项目。

2013 年，添加 3D API 支持。

2014 年 4 月 20 日，1.0 正式版发布。

此后 libGDX 版本的更新比较频繁，截止到 2015 年 3 月 24 日。libGDX 已经更新到了 1.5.4 版本。libGDX 的作者之一 Mario Zechner，也是《Beginning Android Games》一书的作者之一。

1.1.4 我的 libGDX 版本

这本书主要围绕 libGDX 0.9.8 来编写。与 0.9.8 版本相比，libGDX 1.x 具有以下特点。

1. 更新了基于 Gradle 项目的安装，不需要更多的 jars，支持所有平台的简单打包。
2. 移除了对 OpenGL ES 1.x 的支持，支持 OpenGL ES 3.0。
3. 移除了大量 Android 后端的集群，要求最低 Android 版本是 Android 2.2。
4. Box2D 转移到了扩展中。
5. 大幅度的改进了 Scene2D，比如 viewport enhancement。
6. 更新了 libGDX repo、网站和 wiki，移除了旧的 setup UI。
7. 移除了音频和图像扩展和 demo 都放在了仓库中。

可以看出与 0.9.8 版本相比，使用 libGDX 1.x 开发起来将会更加简便。但是其最致命的缺点是移除了对 OpenGL 低版本的支持而不是选择兼容 OpenGL 低版本，这就导致市面上相当一部分使用 OpenGL 1.x 手机的用户将无法运行使用 libGDX 1.x 开发出来的游戏。而这一部分用户的丢失所带来的收益上的巨大损失是很多游戏开发商所无法接受的。所以，大部分的游戏开发商还是选择 libGDX 0.9.8 来进行游戏开发。这也是本书选择 libGDX 0.9.8 来进行讲解的主要原因。与 libGDX 0.9.8 相比，libGDX 1.x 的主要 API 的改动较小，它们之间的大部分内容都是类似的。

1.2 搭建 libGDX 开发环境

在本节中，我们将介绍搭建 libGDX 的开发环境。本节是大家使用 libGDX 进行游戏开