从零开始写一个武侠冒险游戏-0-开发框架Codea简介

- 作者:FreeBlues
- 修订记录
  - 2016.06.21 初稿完成.
  - 2016.08.03 增加对 `XCode` 项目文件的说明.

概述

本游戏全程使用一款运行于 `iPad` 上的开发工具类 `APP` -- `Codea` 来开发, `Codea` 是一款 `轻量级` + `全功能` 的开发工具, 它既是一个:

- `运行时库-Runtime`

也是一个:

- `框架-Frameworks`

还是一个:

- `集成开发环境-IDE`

更是一个:

- `调试服务器-debugSever`, 可通过浏览器来云端调试代码.

它还是 `Lua` 语言的一种扩展, 本质上它就是一个运行于 `iPad` 上可以动态加载运行 `Lua` 代码的应用程序.

它实现了一个非常易于使用的编程框架(类似于 `Processing` 和 `OpenFrameWorks`):

- `setup()` 函数, 负责初始化工作, 程序启动时运行一次;
- `draw()` 函数, 负责屏幕绘制, `1` 秒钟刷新 `60` 次;
- `touched(touch)` 函数, 负责处理屏幕触摸;
- `keyboard(key)` 函数, 负责处理键盘操作;
- `collide(contact)` 函数, 负责处理物理模拟中的碰撞检测;
- `orientationchanged(newOrientation)` 函数, 负责检测屏幕位置(是否翻转)

用它提供的功能丰富的函数你可以在 `iPad` 上编写各种程序, 它提供了:

- `基本绘图函数`, 实现一些基本的绘图函数
- `高级绘图函数`, 支持 `OpenGL ES 2.0/3.0`, 直接编写 `shader` 代码, 即时查看运行结果;
- `Lua 语言的大多数函数`, 如 `table` 相关, `string` 相关, `os` 相关, 以及 `math` 相关的一些函

数, 还包括协程 `coroutine`;
- `Lua 的一些库`, 如 `socket`, `lpeg`, `json`
- `触控函数`, 负责处理屏幕触摸事件;
- `调试函数`, 负责处理程序调试工作;
- `物理模拟函数`, 封装 `box2D` 提供了相关的物理模拟函数;
- `动画函数`, 一个专门的动画类, 可以利用它来实现各种动画效果;
- `声音函数`, 负责处理播放声音以及生成各种音效;
- `显示函数`, 负责处理显示模式设置以及视频录制等等
- `矢量函数`, 提供了二维, 三维, 四维向量以及相关的各种操作;
- `传感器函数`, 负责处理 `GPS` 定位传感器和 `加速度` 传感器;
- `网络函数`, 提供了封装后的 `http.request()` 函数
- `存储函数`, 提供了各种存取函数, 用于存取游戏数据到 `iPad` 上.

它提供了一个 `IDE`, 可以 `编辑/调试/运行` 代码, 非常适合触屏操作. 也非常适合那些希望能随时随地拿起平板就能编程的开发者, 据本软件开发者 `Simen` 自述, 最初就是因为他想在平板电脑上随时运行一些代码片段, 于是萌生了这个想法, 结果一步步搞出这么一个超级方便很受欢迎的 `iPad` 开发工具来.

它可以把你编写的代码导出为一个 `XCode` 项目, 然后编译成一个真正的 `APP` 发布到 `APP Store`, 当然了, 需要你拥有一个开发者账号.

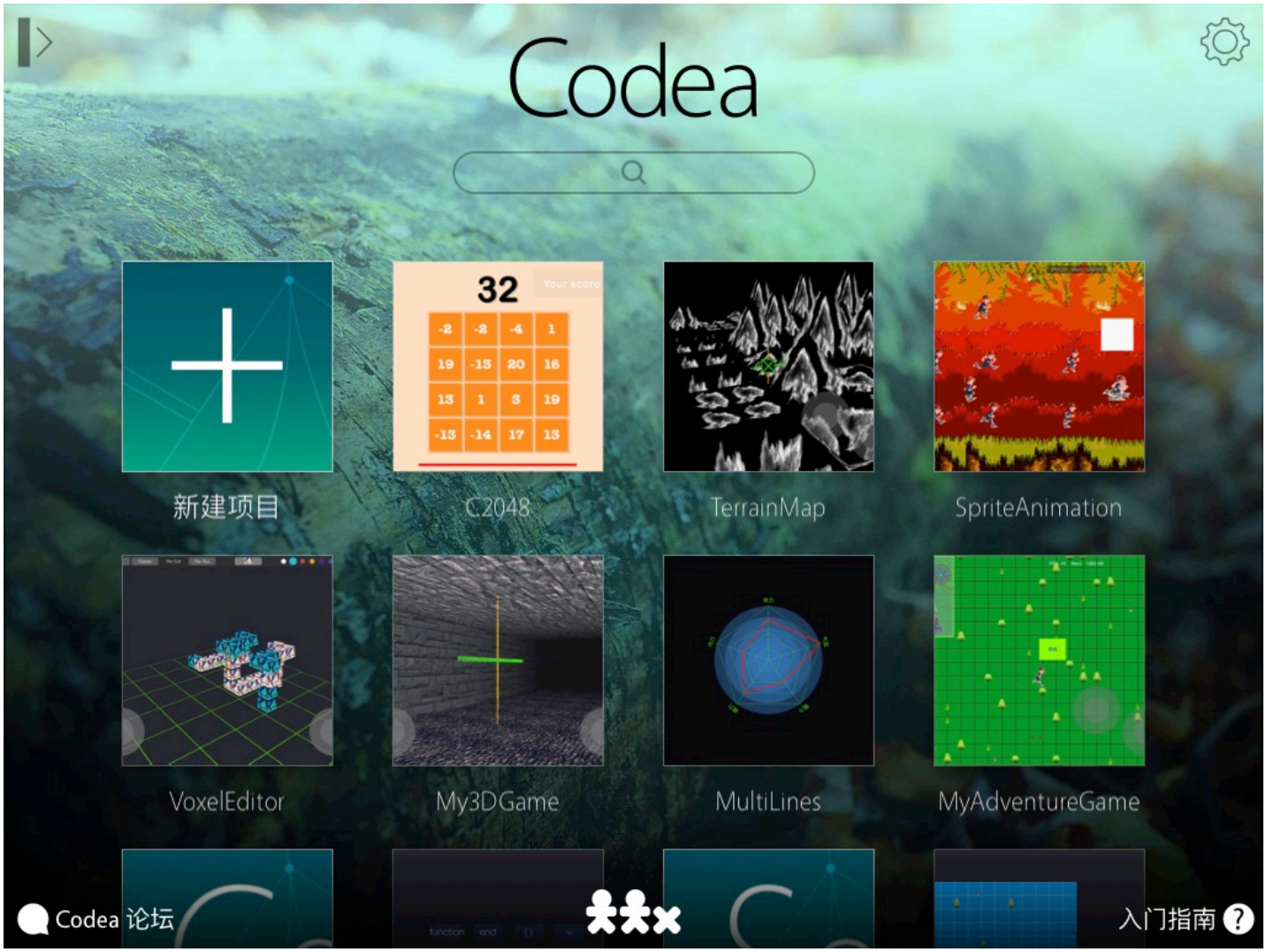
我用过很多 `iPad` 上的编程工具, 最喜欢的一款是 `Codea`, 而且为了更好地发挥它的作用, 专门去认真学了 `Lua`.

接下来我们会对 `Codea` 做一个介绍, 让读者对我们后续即将使用的开发工具有一个初步了解.

界面

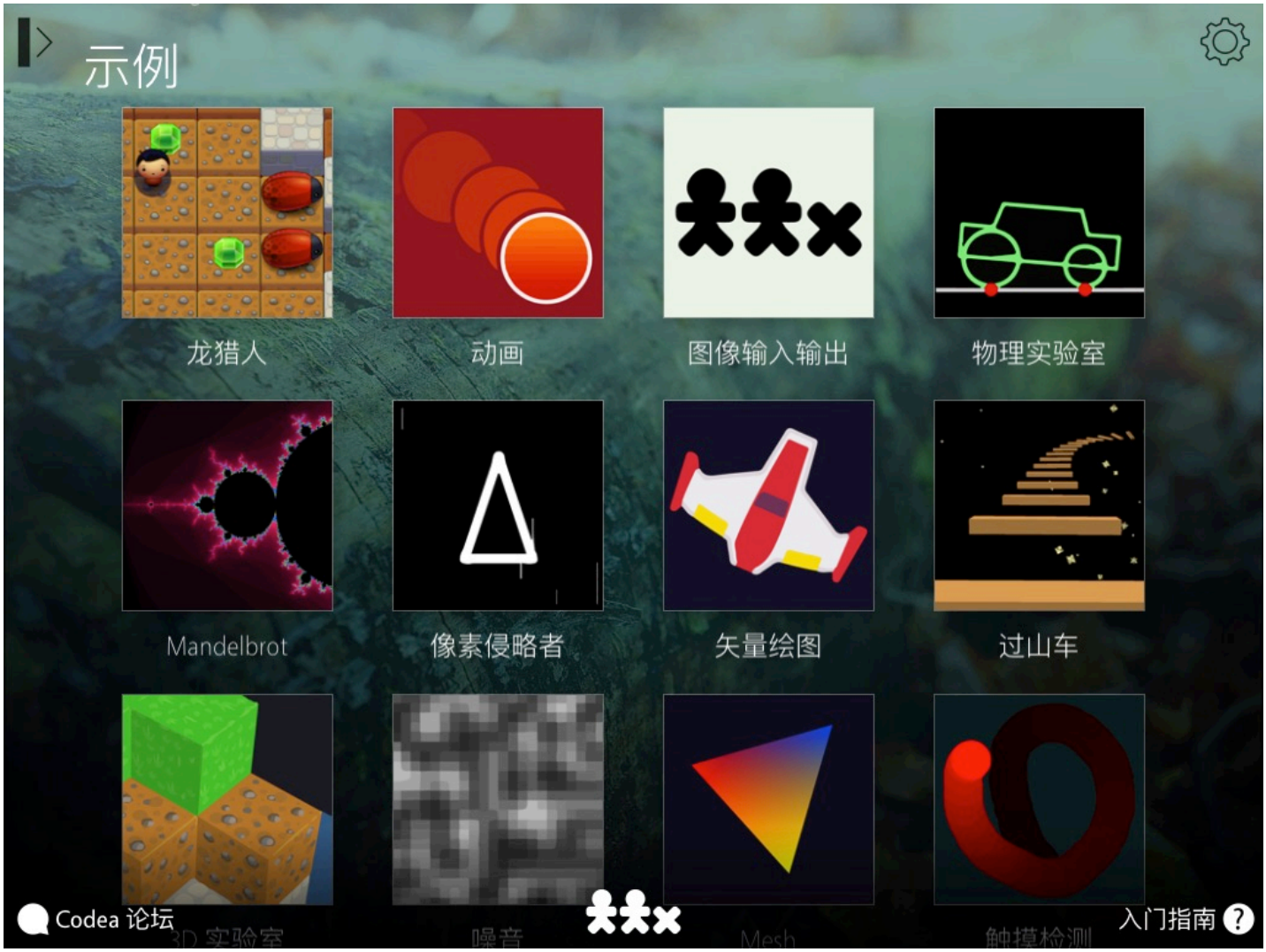
启动进入

启动 `Codea` 后进入主界面, 如下:



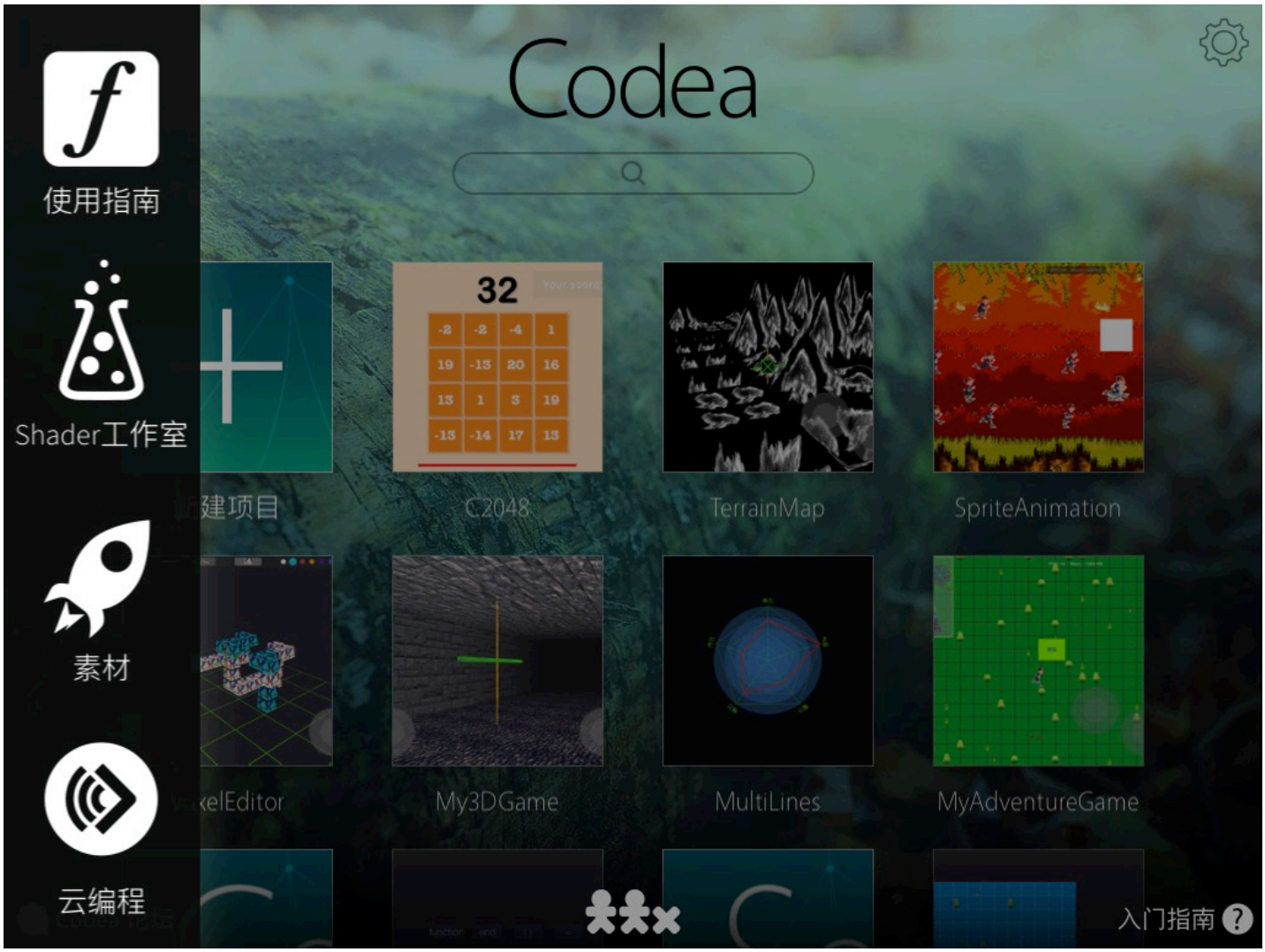
示例

`Codea` 自带了一些示例程序, 如下:



左侧菜单栏

点开左侧菜单栏, 出现 `4` 个选项

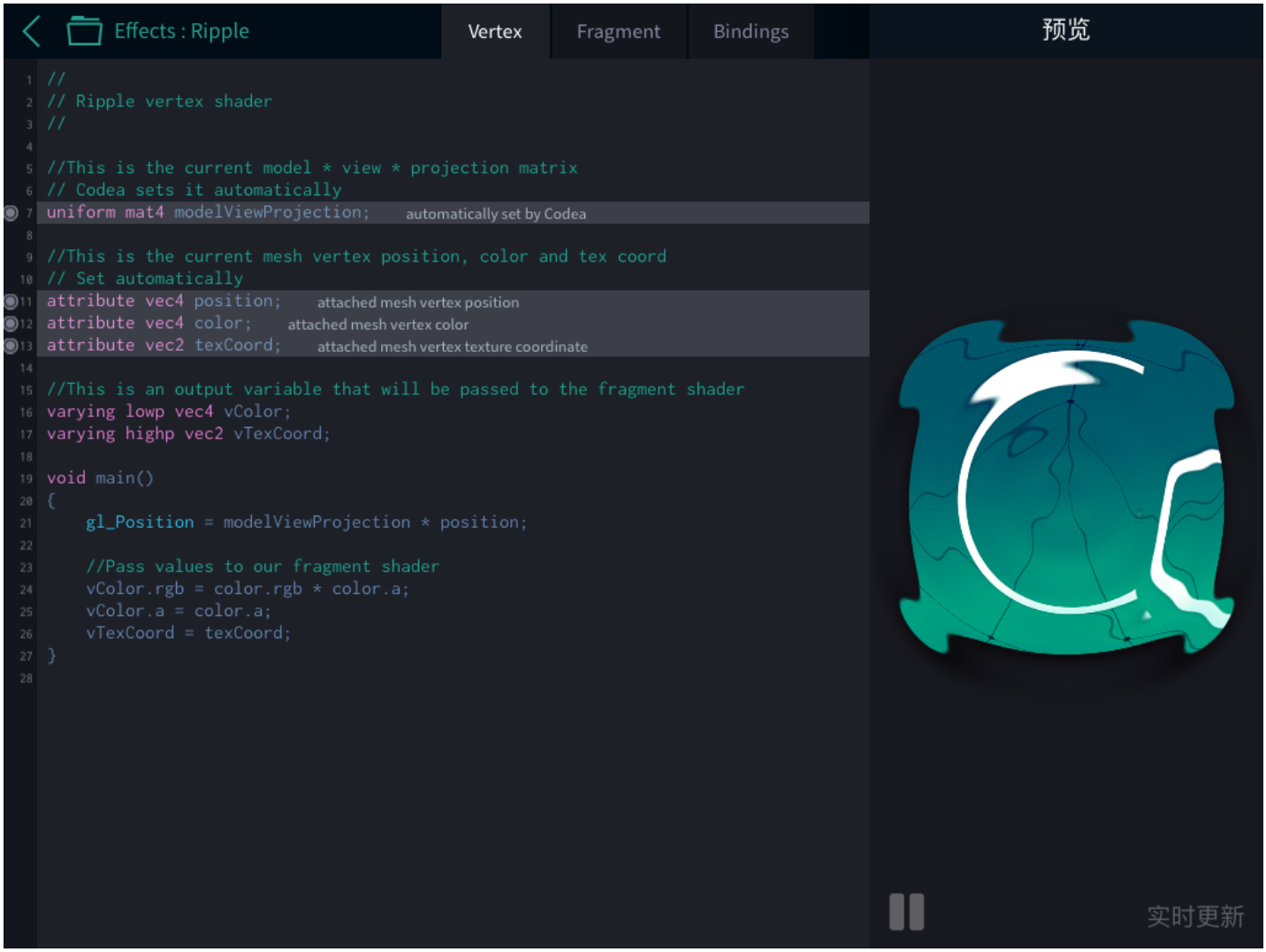


内置函数手册

在 `Codea` 中内置了全部的函数说明, 非常方便在编程时随手查阅

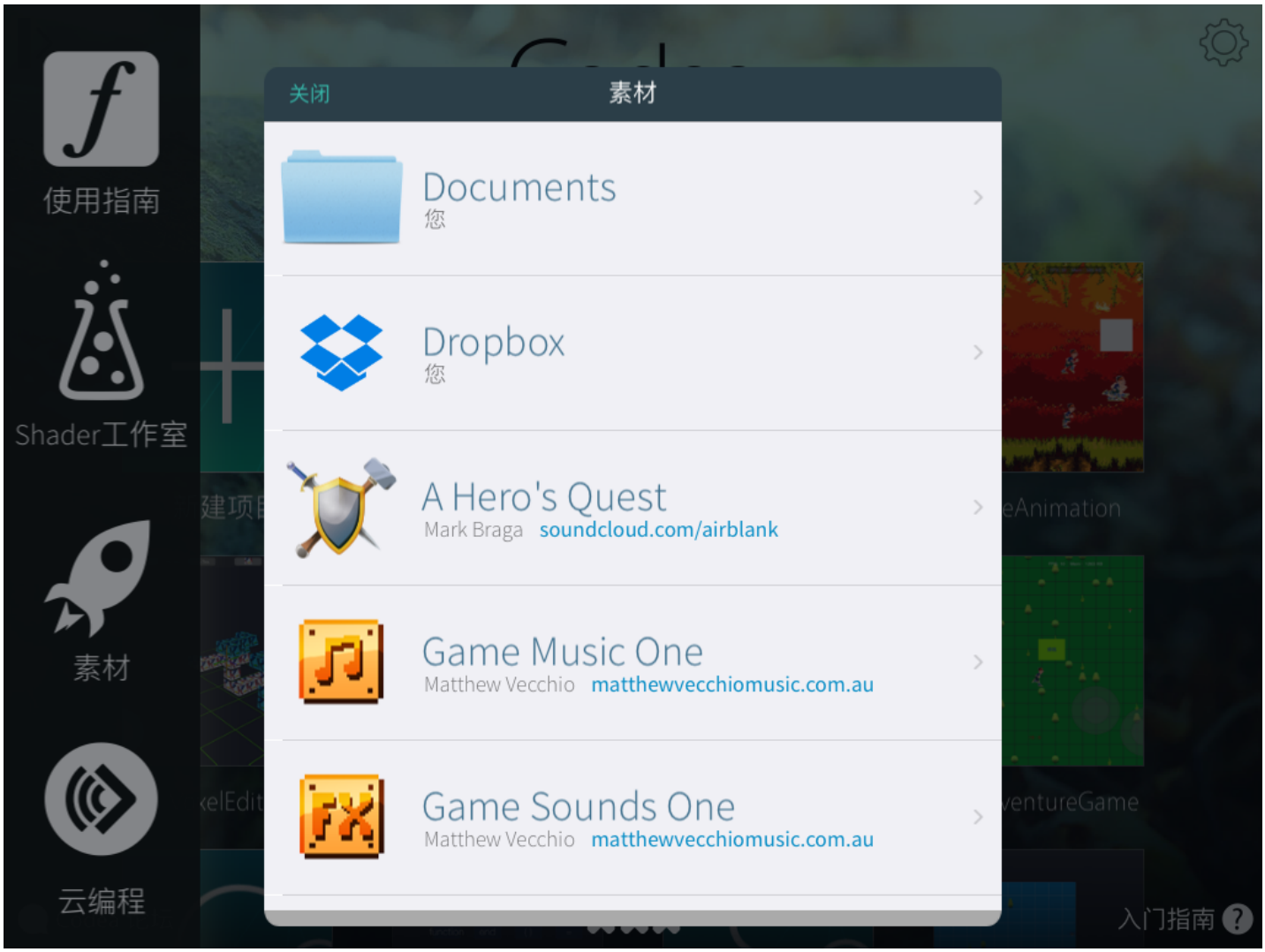
shader实验室

左侧菜单栏的第二项是一个 `shader` 实验室, 你可以在里面调试自己的 `shader` 代码



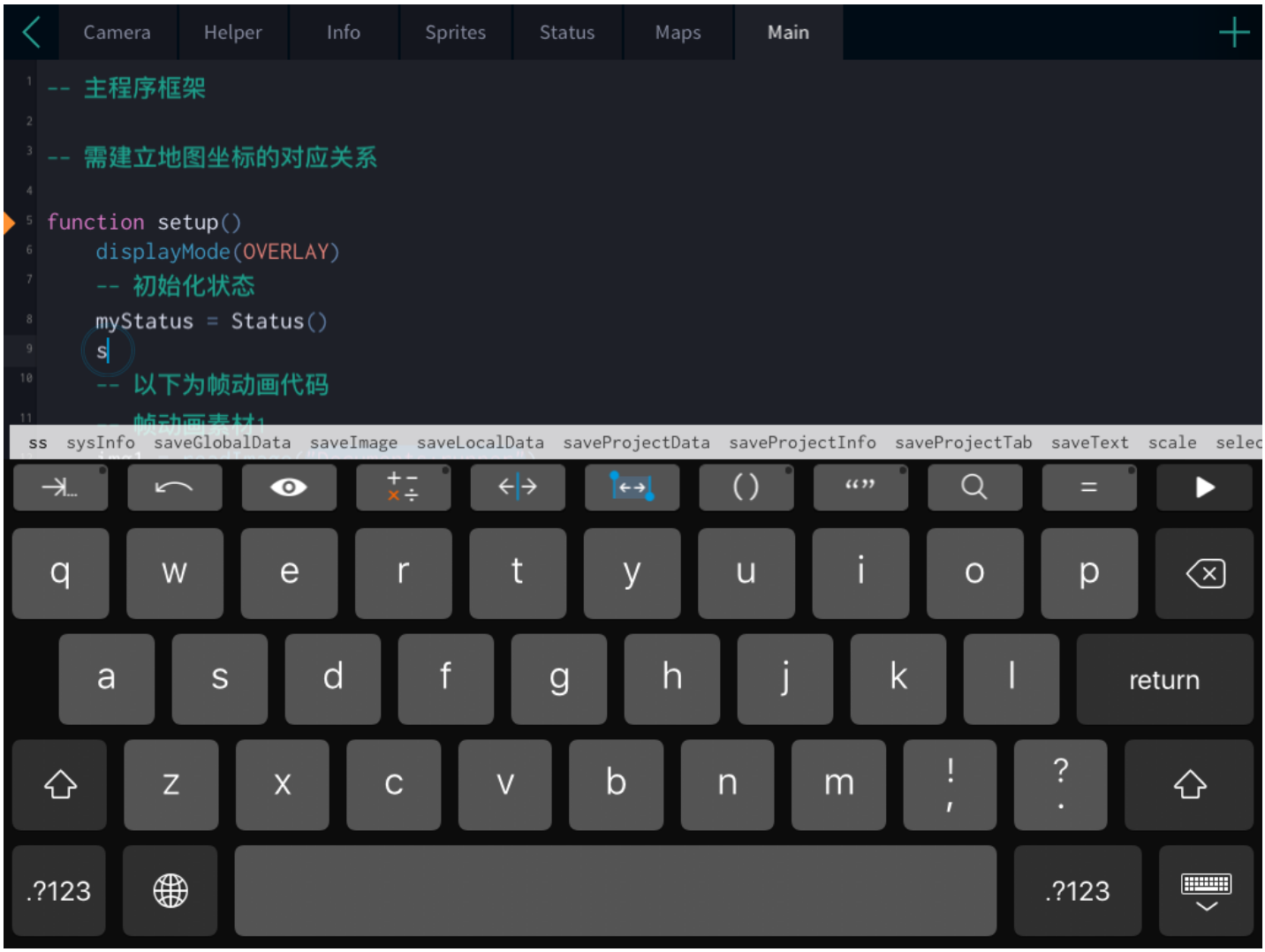
自带素材库

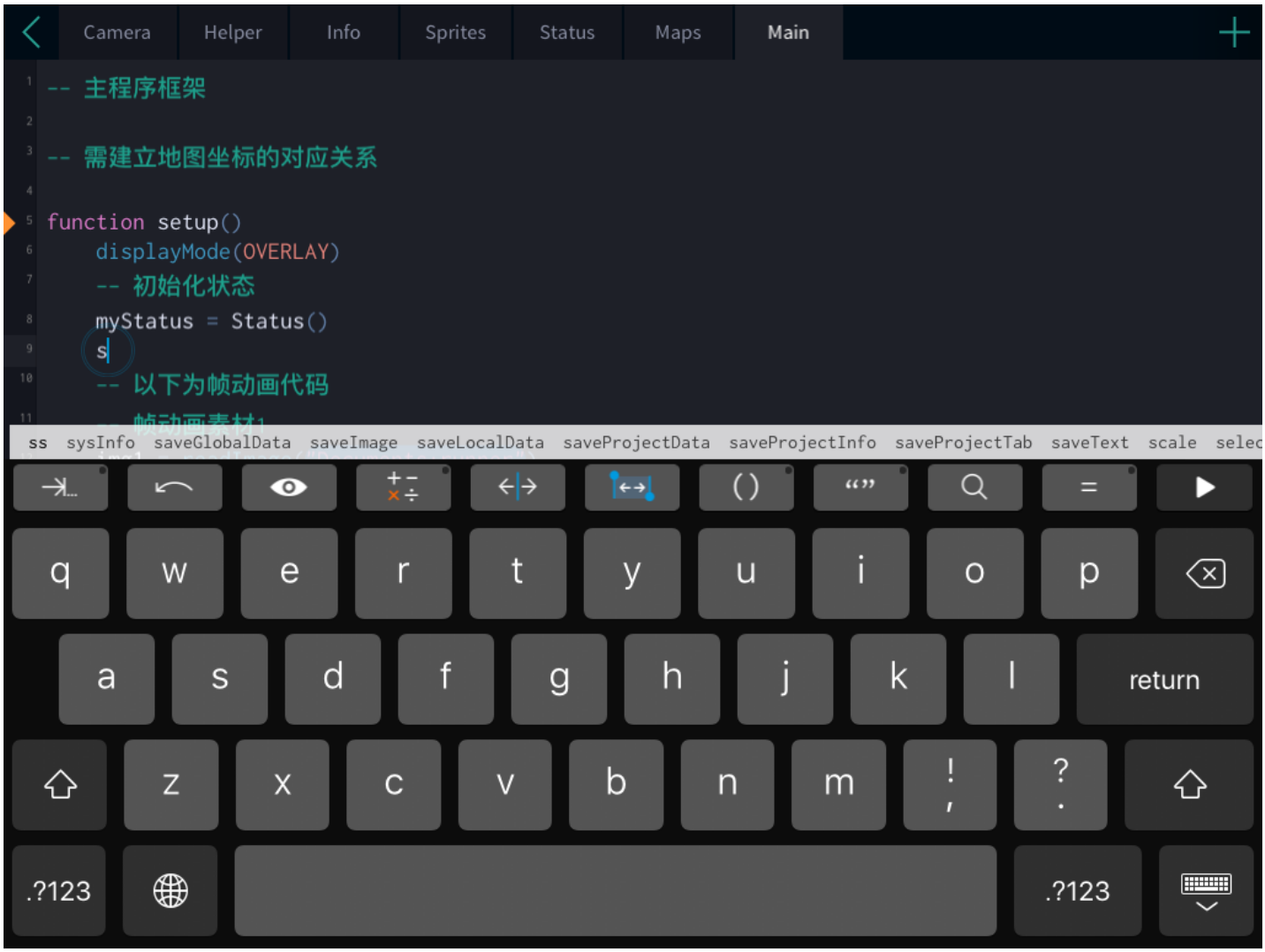
左侧菜单栏第三项是 `Codea` 自带的素材库, 有一些可以免费使用的图片素材和音乐音效素材, 还有一些 `shader` 模板, 用户也可以自行添加素材到素材库



编辑界面

编辑界面自带联想输入, 非常方便直接在 `Codea` 上输入代码


Camera
Helper
Info
Sprites
Status
Maps
Main
-- 主程序框架
-- 需建立地图坐标的对应关系
function setup()
displayMode(OVERLAY)
-- 初始化状态
myStatus = Status()
s
-- 以下为帧动画代码
ss sysInfo saveGlobalData saveImage saveLocalData saveProjectData saveProjectInfo saveProjectTab saveText scale
q w e r t y u i o p
a s d f g h j k l return
z x c v b n m
.?123
.?123



程序框架说明

`Codea` 的运行机制是这样的:

- `setup()` 只在程序启动时执行一次
- `draw()` 在程序执行完 `setup()` 后反复循环执行, 每秒执行 `60` 次
- `touched()` 跟 `draw()` 类似, 也是反复循环执行

简单说, 就是类似于这样的一个程序结构:

```
setup()

while true do
    ...
    draw()
    touched(touch)
    ...
end
```

如果读者有过使用 `Processing` 或 `OpenFrameWorks` 的经验, 就比较熟悉这种框架了, 这种框架的优点就是结构简单易懂, 流程非常清晰, 容易上手, 而且功能也不弱, 反而 `Cocos2d-x` 的那种又是场景, 又是导演的框架比较麻烦.

如果你愿意, 也很容易在它的基础上自己搞一个 `MVC` 架构出来, 如果你想实现更复杂的流程控制, 可以通过 `Lua` 的 `coroutine` 自行扩展, 也可以参考本教程提供的用 `coroutine` 实现的 `Threads` 类.

主要函数说明

`Codea` 的函数大多数都支持可变参数, 根据输入实参的个数决定对应哪些形参, 好像在面向对象中叫多态. 下面在每种函数中选择两个常用的函数

基本绘图函数

- background( )
  - 语法
  - background(gray)
  - background(gray, alpha)
  - background(red, green, blue)
  - background(red, green, blue, alpha)
  - background(color)
- rect( )
  - 语法
  - rect(x, y, width, height)
- sprite(name, x, y)
  - 语法
  - sprite(name, x, y)
  - sprite(name, x, y, width)
  - sprite(name, x, y, width, height)
  - sprite(image, x, y)
  - sprite(image, x, y, width)
  - sprite(image, x, y, width, height)

高级绘图函数

提供 `shader` 和 `mesh`

支持 `OpenGL ES 2.0/3.0`

Lua 语言

`Lua` 中的 `表` , `时间` , `字符串` 和 `数学运算` 以及一部分 `os` 函数

触控函数

负责处理屏幕触摸事件;

调试函数

负责处理程序调试工作;

物理模拟函数

封装 `box2D` 提供了相关的物理模拟函数;

动画函数

一个专门的动画类, 可以利用它来实现各种动画效果;

声音函数

负责处理播放声音以及生成各种音效;

显示函数

负责处理显示模式设置以及视频录制等等

矢量函数

提供了二维, 三维, 四维向量以及相关的各种操作;

传感器函数

负责处理 `GPS` 定位传感器和 `加速度` 传感器;

网络函数

提供了封装后的 `http.request()` 函数

存储函数

提供了各种存取函数, 用于存取游戏数据到 `iPad` 上.

其他在 iPad 上运行的开发工具

除了 `Codea`, 还有很多可以运行在 `iPad` 上的编程工具, 比如 `Processing`, `TechBASIC` 等等, 它们各有所长.

苹果在最近的 `WWDC` 上宣布 `iOS 10` 会提供一款名为 `Swift Playground` 的免费开发工具, 看来随着平板电脑性能的提升, 用平板编写代码已经是趋势所向.

如何使用本教程中的源码

你有两种方式来运行本教程的示例, 一种是下载源码, 直接在 `iPad` 上通过 `Codea` 来运行, 另一种是下载对应的 `XCode` 项目文件包, 然后在 `XCode` 中把它编译为一个 `ipa` 应用, 在把它加载到模拟器上运行.

直接在 iPad 上用 Codea 加载代码

如果你有 `iPad`, 并且购买了 `Codea`, 那么你可以打开 `Codea`, 新建一个项目, 把例程代码拷贝进去, 同时要把例程中用到的图像资源下载拷贝到 `Codea` 的素材库目录下, 一般是 `Documents`, 然后就可以运行了.

这种方法最简单, 也最方便, 唯一的不便之处是需要预先手动下载保存好图片素材(不过后续我会写几个自动下载保存图片的函数来提供更方便的使用)

通过 XCode 的 iOS 模拟器来运行

如果你没有 `iPad`, 或者虽然有 `iPad`, 但是暂时还不准备花 `15` 美元购买一份 `Codea`, 那么你还可以通过模拟器来运行.

这就需要你有一台安装了 `XCode` 的 `Mac` 电脑, 或者你有一个安装了 `XCode` 的 `OSX` 虚拟机, 你只要把每章对应的 `XCode` 项目文件包下载回去, 然后在 `XCode` 中打开它, 编译运行, 就可以在模拟器中看到运行结果了.

而且你也可以在 `XCode` 中修改代码, 重新编译查看效果.

XCode项目文件夹结构

项目文件夹结构如下:

```
Air:Write-A-Adventure-Game-From-Zero admin$ tree
.
MyAdventureGame
├── Assets
|   ├── ...
├── Libs
|   ├── ...
├── MyAdventureGame
|   ├──...
├── MyAdventureGame.codea
|   ├──...
├── MyAdventureGame.xcodeproj
|   ├──...
└── libversion
```

其中我们编写的 `Lua` 脚本放在 `MyAdventureGame.codea` 目录下, 以 `main.lua` 命名, 如下:

```
Air:Write-A-Adventure-Game-From-Zero admin$ tree ./MyAdventureGame/MyAdventureGame
.codea
./MyAdventureGame/MyAdventureGame.codea
├── Info.plist
├── Main.lua
├── Sprites.lua
└── Status.lua

0 directories, 4 files
Air:Write-A-Adventure-Game-From-Zero admin$
```

Codea的运行时库文件

- 注意: 在用 `XCode` 编译项目时, 需要用到 `Codea` 的两个库文件: `libcodea.a` 和 `libtools.a`.

当前版本 `2.3.2`, `XCode` 会自动从 `Codea` 官网下载, 不过貌似会被屏蔽, 再加上这两个文件稍微大了点(一个 `20` M, 一个 `50` M), 就没有放在 `github` 上, 改为放到百度网盘上Codea库文件下载, 下载回来后解压得到名为 `libcodea.a` 和 `libtools.a` 的两个文件, 将其拷贝到项目文件夹的 `Libs` 目录下即可.

参考资源

官网函数手册
官方wiki
官网论坛
中文函数手册

所有章节链接

Github项目地址

Github项目地址, 源代码放在 `src/` 目录下, 图片素材放在 `assets/` 目录下, `XCode` 项目文件放在 `MyAdventureGame` 目录下, 整个项目文件结构如下:

```
Air:Write-A-Adventure-Game-From-Zero admin$ tree
.
├── MyAdventureGame
|   ├── Assets
|   |   ├── ...
|   ├── Libs
|   |   ├── ...
|   ├── MyAdventureGame
|   |   ├──...
|   ├── MyAdventureGame.codea
|   |   ├──...
|   ├── MyAdventureGame.xcodeproj
|   |   ├──...
|   └── libversion
├── README.md
├── Vim 列编辑功能详细讲解.md
├── assets
|   ├── ...
|   └── runner.png
├── src
|   ├── c01.lua
|   ├── c02.lua
|   ├── c03.lua
|   ├── c04.lua
|   ├── c05.lua
|   ├── c06-01.lua
|   ├── c06-02.lua
|   ├── c06-03.lua
|   └── c06.lua
├── 从零开始写一个武侠冒险游戏-0-开发框架Codea简介.md
├── 从零开始写一个武侠冒险游戏-1-状态原型.md
├── 从零开始写一个武侠冒险游戏-2-帧动画.md
├── 从零开始写一个武侠冒险游戏-3-地图生成.md
├── 从零开始写一个武侠冒险游戏-4-第一次整合.md
├── 从零开始写一个武侠冒险游戏-5-使用协程.md
├── 从零开始写一个武侠冒险游戏-6-用GPU提升性能(1).md
├── 从零开始写一个武侠冒险游戏-6-用GPU提升性能(2).md
└── 从零开始写一个武侠冒险游戏-6-用GPU提升性能(3).md

2 directories, 26 files
Air:Write-A-Adventure-Game-From-Zero admin$
```