



Cocos2d-x Lua API 开发环境搭建

在开始详细介绍 Cocos2d-x 引擎的 Lua API 之前,有必要介绍开发环境。本章我们会介绍 Cocos2d-x Lua API 开发环境搭建和 IDE 开发工具的使用。

3.1 搭建环境

在开发、编译、发布和运行 Cocos2d-x 游戏过程中都会用一些工具,这些工具需要搭建环境。有两种方法可以搭建 Cocos2d-x 开发环境:一种是使用 Cocos2d-x 开发包,自己手动配置环境;另一种是使用 Cocos 引擎自动配置环境。第二种方式更适合初学者,本书重点介绍第二种方式搭建开发环境。

3.1.1 Cocos 引擎下载安装

Cocos 开发团队提供了一个集成工具——Cocos 引擎,下载地址为 <http://www.cocos.com/download/>,如图 3-1 所示,单击 Cocos 引擎图标,在页面的下面可以选择操作系统版本,然后单击后面的“立即下载”按钮进行下载。

下载完成之后就可以双击安装文件进行安装了,Cocos 引擎安装成功之后,在安装目录(本例是 C:\Cocos)如图 3-2 所示,其中包含了 Cocos Studio(Cocos 场景设计工具)、cocos-simulator-bin(Cocos 模拟器)、frameworks(Cocos2d-x 框架)、templates(Cocos2d-x 模板)和 tools(基本的环境工具 ant 和 Python27)。而且 Cocos 引擎安装还设置和系统环境变量。

3.1.2 Cocos Framework 下载和安装

如果安装目录下没有 frameworks,说明这个 Cocos 引擎不包含 frameworks,我们需要在 Cocos 引擎中下载和安装 frameworks。运行 Cocos 引擎的界面如图 3-3 所示,我们可以在“商店”中找到 Cocos Framework,然后下载。下载完成之后可以在 Cocos 引擎中“下载”→“已下载”中找到 Cocos Framework 安装文件进行安装。



图 3-1 Cocos 引擎



图 3-2 Cocos 引擎目录



图 3-3 运行 Cocos 引擎界面

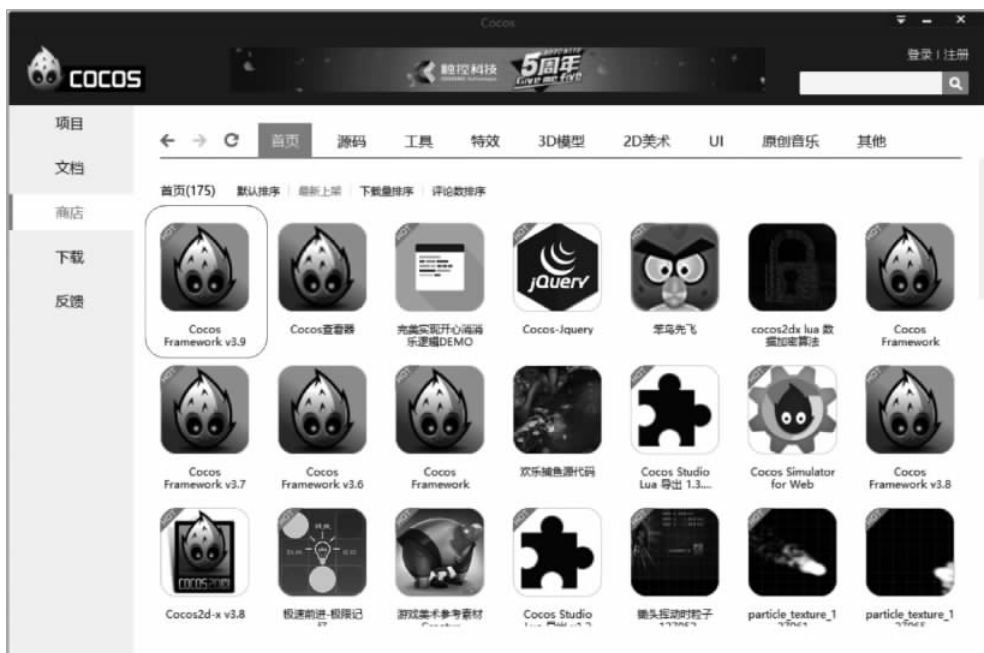


图 3-4 下载 frameworks 界面

3.2 Cocos Code IDE 开发工具

使用 Cocos2d-x Lua API 开发游戏的主要程序代码是 Lua 语言,因此凡是能够开发 Lua 语言工具都适用于 Cocos2d-x Lua API 游戏开发。本书推荐使用 Cocos Code IDE 工具。

3.2.1 安装 Cocos Code IDE

Cocos Code IDE 是 Cocos2d-x 团队开发的,目前的版本是 2.0.0Beta,用于开发 Cocos2d-x Lua API 和 Cocos2d-x Lua API 开发工具,Cocos Code IDE 是基于 IntelliJ IDEA 平台的,IntelliJ IDEA 也是 JetBrains 公司开发,需要安装 JDK 或 JRE,JDK 是 Java 开发工具包,JRE 是 Java 运行环境。JDK 下载和安装可以参考 2.1.1 节。

1. Cocos Code IDE 完整安装

完整的安装包含了 IntelliJ IDEA 和 Cocos Code IDE 插件,Windows 下载地址为 <http://www.cocos2d-x.org/filedown/cocos-code-ide-2.0.0-beta.exe>; Mac OS X 下载地址是 <http://www.cocos2d-x.org/filedown/cocos-code-ide-2.0.0-beta.dmg>。这种方式安装比较简单,不需要过多介绍了。

2. Cocos Code IDE 插件安装

插件安装就是自己下载 Cocos Code IDE 插件和 IntelliJ IDEA,然后在 IntelliJ IDEA 中配置 Cocos Code IDE 插件。IntelliJ IDEA 的下载地址为 <https://www.jetbrains.com/idea/download/>。如图 3-5 所示,我们可以选择 Ultimate 和 Community 版本,Ultimate 版本是收费的,Community 版本是免费的。对于 Cocos2d-x Lua API 开发,Community 版本就够用了。

提示 IntelliJ IDEA 目前是 15.0.1 版本,这个版本调试模式下有一些问题,笔者推荐使用 14.0 版本,老版本可以通过页面下的“previous versions”超链接进入下载页面。

Cocos Code IDE 插件下载地址是 <http://www.cocos2d-x.org/filedown/cocos-intellij-plugin-2.0.0-beta.zip>,插件没有平台之分,无论是 Windows、Mac OS X 还是 Linux 都是通用的。

下载完成之后,需要在 IntelliJ IDEA 中配置插件,启动 IntelliJ IDEA 后可以看到欢迎界面,如图 3-6 所示,在欢迎界面中单击 Configure→Plugins,弹出如图 3-7 所示的插件安装对话框,单击 Install plugin from disk 按钮从磁盘中安装插件,安装完成单击 OK 按钮关闭对话框。

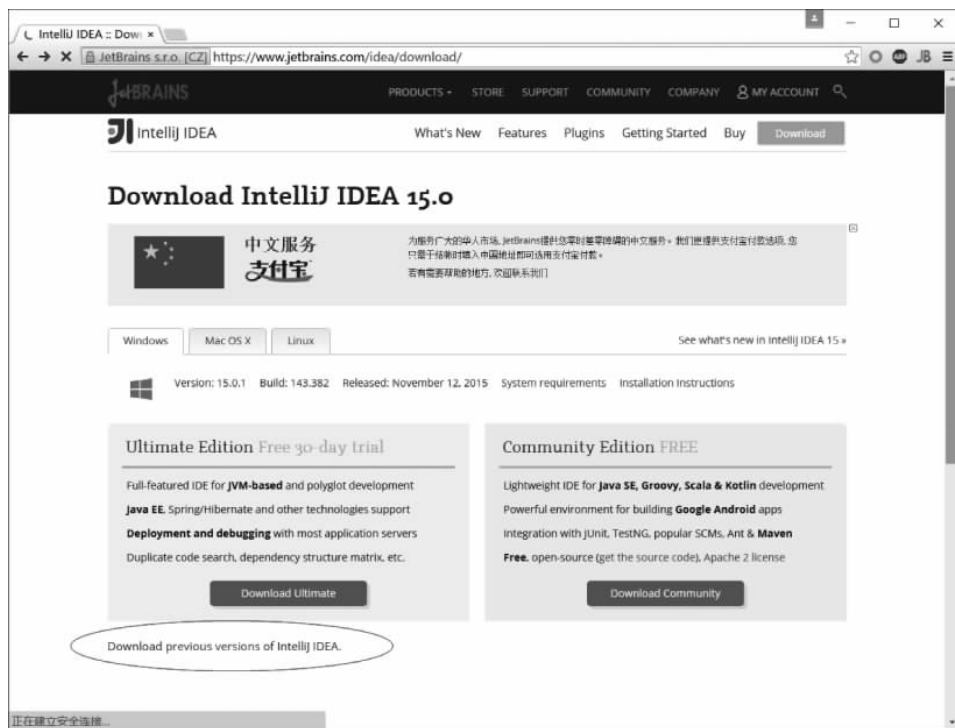


图 3-5 IntelliJ IDEA 下载页面



图 3-6 配置 Cocos Code IDE 插件

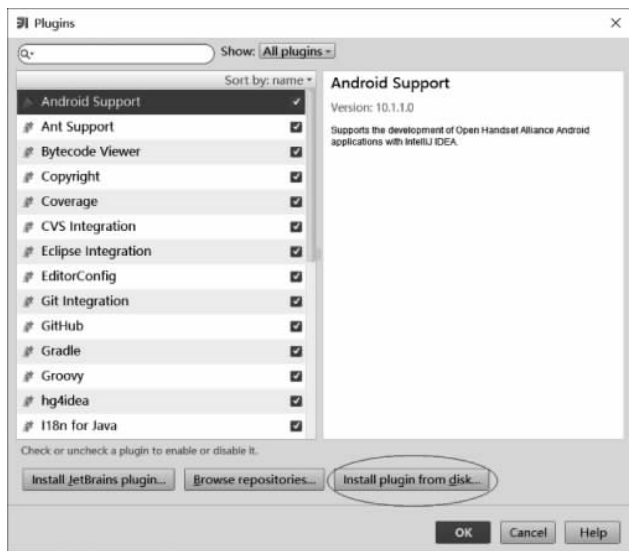


图 3-7 插件安装对话框

3.2.2 配置 Cocos Code IDE

Cocos Code IDE 工具安装完成后,我们还需要配置该工具。可以通过在欢迎界面中单击 Configure→Settings,或通过单击菜单 File→Settings,弹出如图 3-8 所示的 IntelliJ IDEA 配置对话框,在 Other Settings→Cocos Framework 配置 Cocos Code IDE 插件。有两种模式(Framework Mode 和 Engine Mode)可以配置插件。使用 Cocos Framework 可以配置 Framework Mode 模式,这种模式下运行配置比较简单;如果不使用 Cocos Framework 而是 Cocos 开发包,可以选中 Engine Mode 模式,其中 Lua 需要 Cocos2d-x 开发包,JavaScript 需要 Cocos2d-x JS 开发包,但是这种 Engine Mode 模式在 Cocos2d-x 3.7 之后不再使用了,在 Cocos2d-x 3.7 之后 Cocos2d-x 和 Cocos2d-JS 合并了,合并之后的目录结构有很大的改变。

Framework Mode 下还需要配置 Simulator(模拟器),我们可以在图 3-8 所示的界面中 Simulator 后面的 Browse 按钮找到模拟器安装目录,如果安装了 Cocos 引擎,则在 Cocos 引擎的安装目录下会有 cocos-simulator-bin 目录,这就是模拟器安装目录,图 3-9 是笔者根据自己情况设置的模拟器。

注意 使用 Cocos 引擎提供的 Win32 模拟器时,程序无法进行日志输出,这是由于这个版本的 Win32 模拟器工具在编译时候是 release 模式。解决办法是找到 Win32 模拟器源程序进行 debug 编译。我们可以在 < cocos2d-x 开发包 > \tools\simulator\frameworks\runtime-src\proj.win32 目录找到 Win32 模拟器程序代码,我们需要用 Visual Studio 工具打开并进行 debug 编译。另外,也可以直接使用笔者编译好的模拟器,模拟器可以在本书工具包中找到,工具包下载地址请参考本书前言。

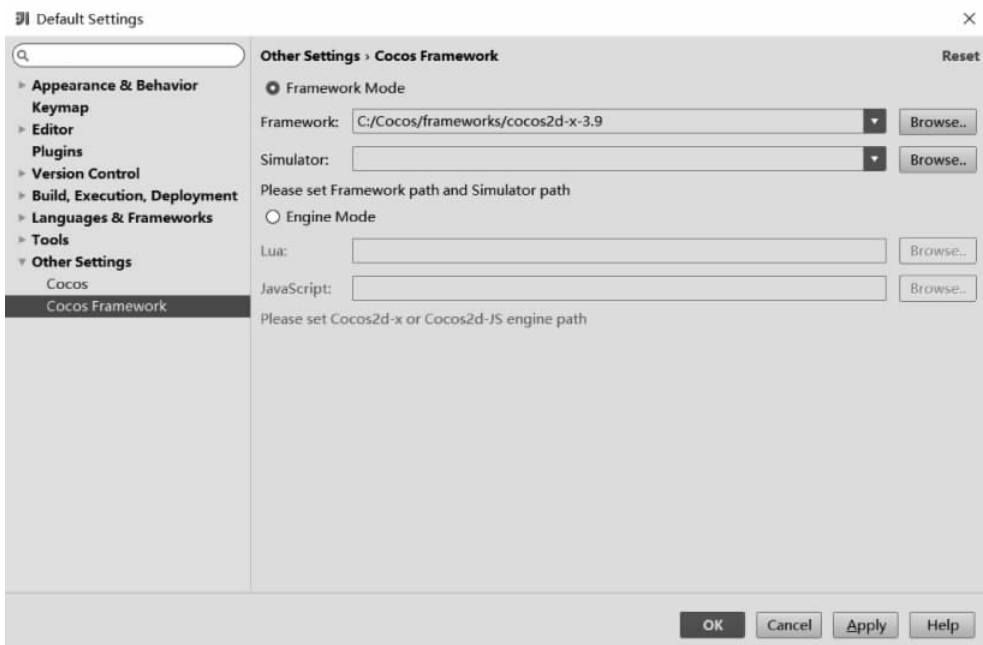


图 3-8 配置 Cocos Code IDE 工具

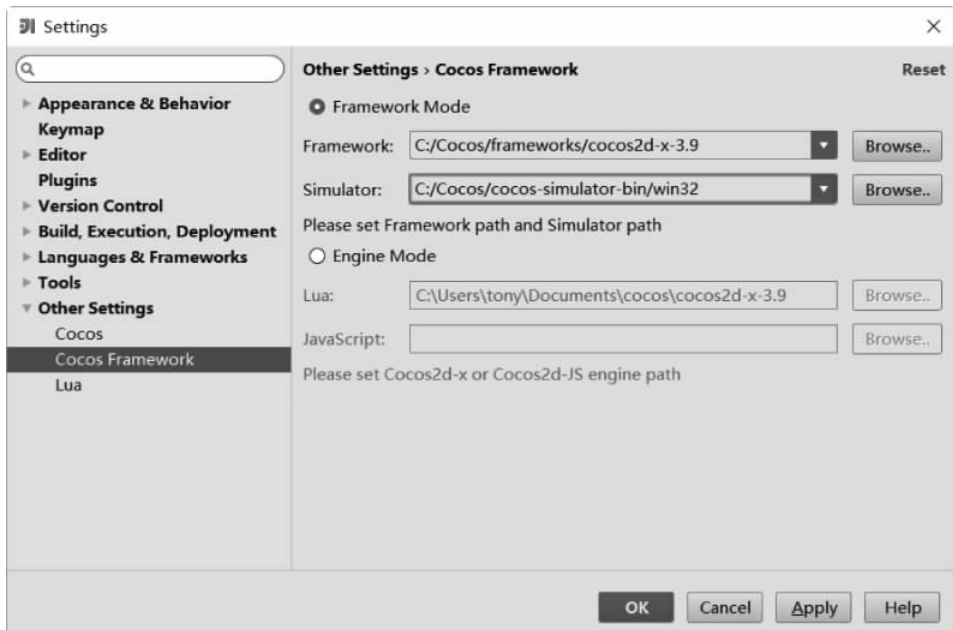


图 3-9 配置模拟器

有时候,虽然模拟器设好了,但是还会出现无法找到模拟器的现象,我们可以重新设置模拟器环境。如图 3-10 所示,单击工具栏中的 HelloLua 按钮,在弹出菜单中选择 Edit Configurations,然后弹出如图 3-11 所示的对话框,在 LuaBinding Target Platform→Windows 7/ Windows 8 的 Simulator path 中输入 C:\Cocos\cocos-simulator-bin\win32\ Simulator.exe,C:\Cocos 是笔者的 Cocos 引擎安装目录,读者可以根据自己的情况设置模拟器目录。设置完成之后单击 OK 按钮就可以了。

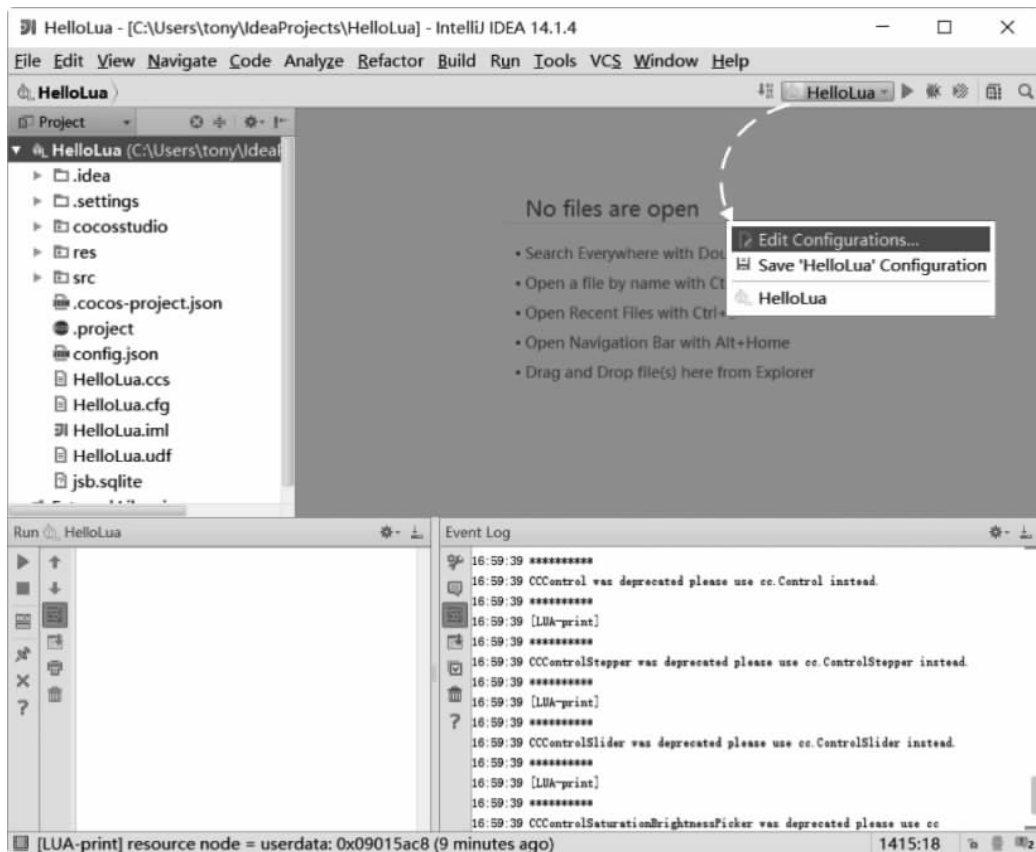


图 3-10 模拟器不存在

3.2.3 使用 Cocos Code IDE 调试功能

熟悉 Cocos Code IDE 调试功能是很有必要的,Cocos Code IDE 是基于具 IntelliJ IDEA 平台的,调试功能是由 IntelliJ IDEA 提供的。如果熟悉 IntelliJ IDEA 相关软件,对于 Cocos Code IDE 调试就不陌生了。

如果我们想调试程序,可以设置断点,如图 3-12 所示,单击行号后面位置,设置断点。在调试模式下运行,程序运行到断点所在行挂起,如图 3-13 所示。

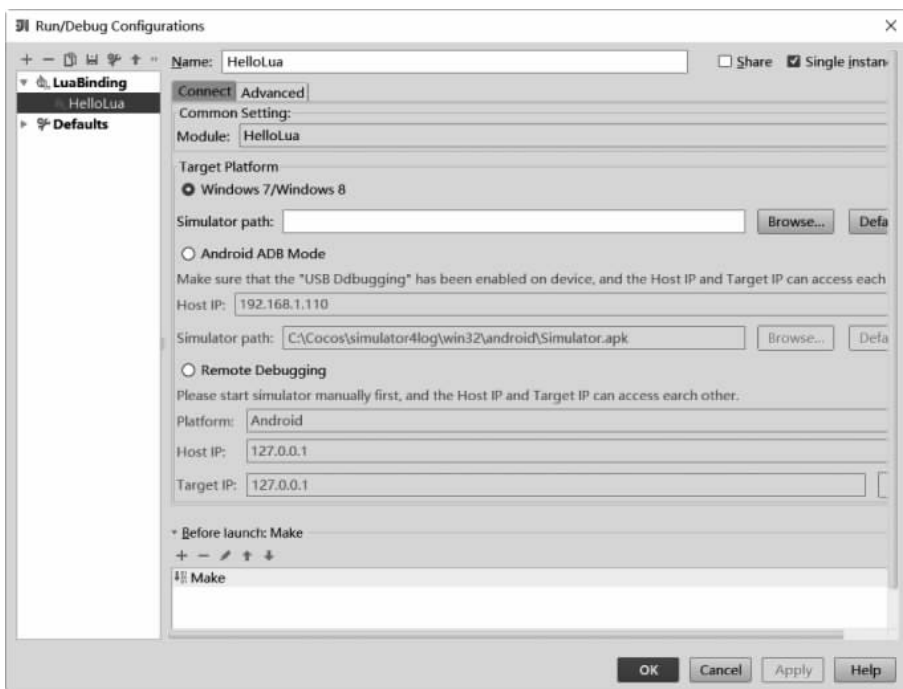


图 3-11 设置模拟器

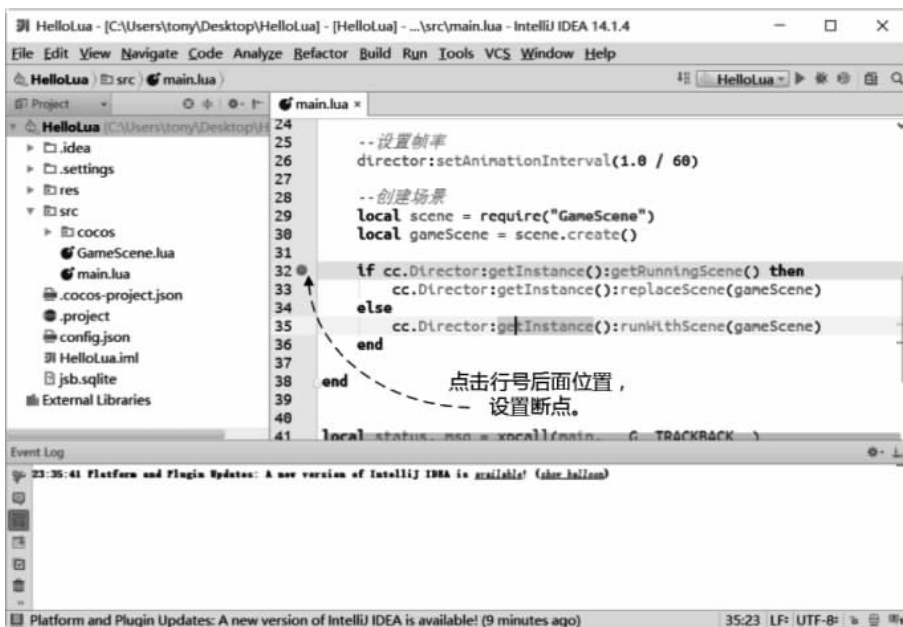


图 3-12 设置断点

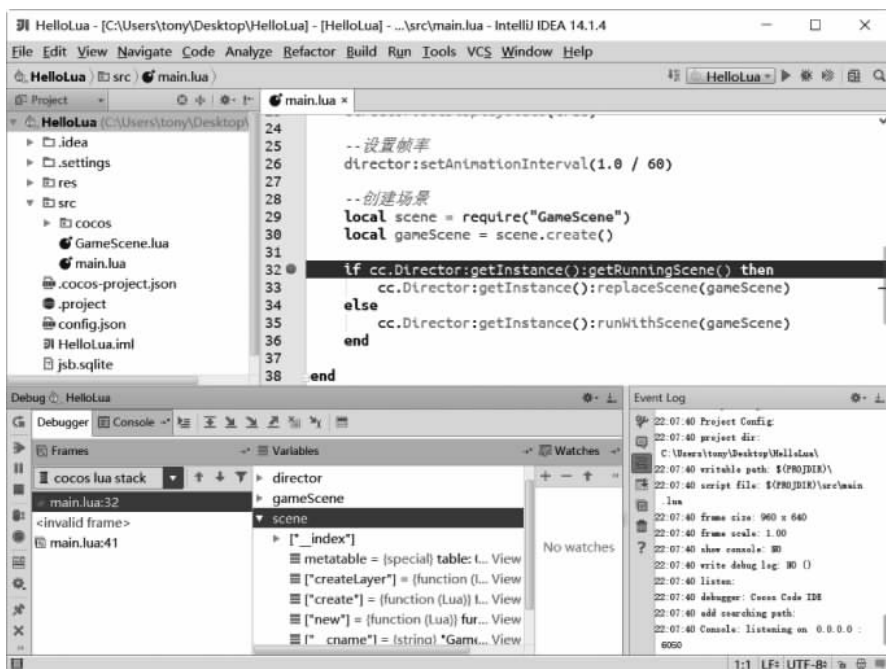


图 3-13 运行到断点挂起

我们在 Debugger 中的 Variables 中查看变量,从中可以看到变量的值,在 Debugger 窗口中有很多调试工具栏按钮,这些按钮的含义说明如图 3-14 所示。

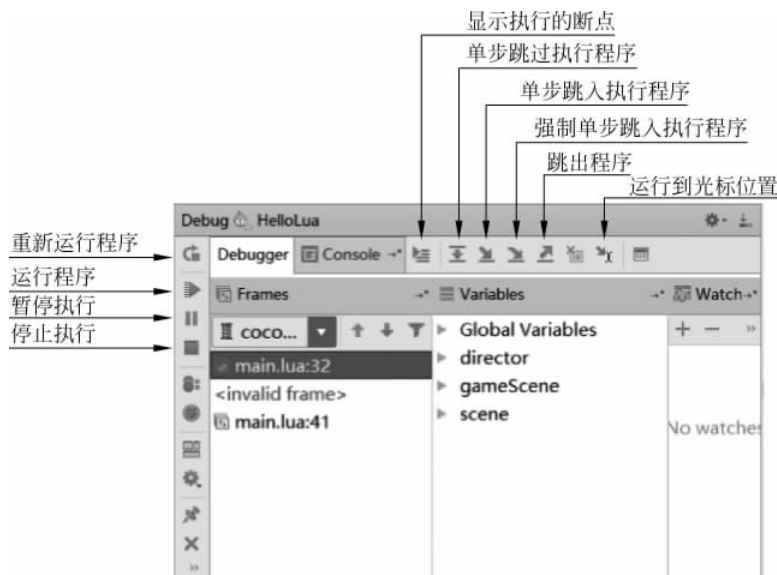


图 3-14 调试工具栏按钮

本章小结

通过对本章的学习,读者可以了解 Cocos2d-x Lua API 开发环境的搭建,其中包括 Cocos 引擎工具和 Cocos Code IDE 工具的安装和配置。