工具介绍

LWJGL是一个底层API.与其说它像一个引擎,不如说他更像一个OpenGL的一个包装,以下简单的介绍几款JAVA研发所需的软件

工具包推荐:

* Intellij Idea
* Maven

基础工具

JDK

一个Java程序员在入手到一个新电脑后,第一步是什么?

是下载英雄联盟吗? 还是下载Steam?

都不是.

Java程序员的第一步,是寻找合适版本的JDK

,各位读者可以在Bing上直接搜索JDK,找到并下载.

构建工具

Maven

Maven是最最简单的Java构建工具.

想要学会使用Maven,你需要掌握以下知识

* 英文2级
* 会用搜索引擎
* 看得懂XML

Gradle

Gradle是一个令新手头疼的Java构建工具.

想要学会它,你需要掌握以下知识.

* Groovy
* Android
* Maven
* 插件编写

集成开发环境(IDE)

Notepad (笔者极力推荐)

每一台电脑自带的编译器,是分布最”广”的集成”开发环境”, 没有之一.

缺点

* 未集成编译功能

优点

* 免安装
  + 每一台电脑自带该软件,免去了电脑小白安装的烦恼.
* 支持类型多
  + 不论什么文件,都可以用它打开
* UI简洁
  + 简介的UI和永远不会飙红的特点,让众多IDE都望尘莫及

IntelliJ Idea (笔者推荐)

最好的Java开发工具,没有之一.

其创作团队JetBrain还研发了Kotlin.

什么是Kotlin?

被安卓官方任命为官方研发语言的编程语言.

VsCode(Atom)

VsCode于Atom都属于Notepad的扩展版本,不太建议小白使用在此不做过多的介绍.

Eclipse

最常见的Java IDE.

Netbean

冷门,但最”时尚”的IDE,它可以确保你的代码能够跟得上潮流,不免为于编程小白面前装逼的一大利器.

Maven配置下载

<https://www.lwjgl.org/customize>

操作方法

这样子,你就可以得到适合你的Maven的配置文件了



基本概念

以下代码均为伪代码

代码地址: <https://www.lwjgl.org/guide>

初始化代码

* 以下知识涉及OpenGL(GLFW),请自行学习,谢谢.
* 新手,直接把初始化代码照抄就可以了.

初始化代码的作用,就是给你画一个框框,就想打开”画图”这个软件一样,

|  |
| --- |
| //对于高手来说,这个是设置输出流的,你可以把一些Log4j写进去  GLFWErrorCallback.createPrint(System.err).set();  //初始化OpenGL  //准备打开图画软件  if ( !glfwInit() )  return;  //配置OpenGL  //设置这个图画软件能不能修改大小,能不能被你看得见之类的.  glfwDefaultWindowHints(); // optional, the current window hints are already the default  glfwWindowHint(GLFW\_VISIBLE, GLFW\_FALSE); // the window will stay hidden after creation  glfwWindowHint(GLFW\_RESIZABLE, GLFW\_TRUE); // the window will be resizable  //创建窗口(返回值为窗口Id(Long window)  //设置这个框框的大小,和名字  window = glfwCreateWindow(300, 300, "Hello World!", NULL, NULL);  //监听键盘输入,  //没错,这里就是检测键盘输入的地方了. 划重点  glfwSetKeyCallback(window, (window, key, scancode, action, mods) ->{});  // 看看你的桌面一共有多大  GLFWVidMode vidmode = glfwGetVideoMode(glfwGetPrimaryMonitor());  //运用数学的能力计算居中位置  Double center\_hieight=(vidmode.height-300)/2  Double center\_width=;  //把你的画板,放到屏幕中间  glfwSetWindowPos(window,(vidmode.width() - pWidth.get(0)) / 2,(vidmode.height() - pHeight.get(0)) / 2;  // 这个非常重要,就是告诉电脑,我们现在需要用的就是这个画板.  glfwMakeContextCurrent(window);  //新手不用管,这么设置就对了  // Enable v-sync  glfwSwapInterval(1);  //最后一步,让电脑把画板展现出来  glfwShowWindow(window);  } //恭喜你,成功的创建了一共窗口~ |

循环代码

* 循环代码,是接在初始化代码之后的
* 以后你要画画什么的,都写在循环代码里就可以了

循环,就是在窗口出现和游戏关闭之前,一直用到的东西.

|  |
| --- |
| // 创建画布,你可以开始画画咯  GL.createCapabilities();  // 背景颜色,就是你画画的纸是什么颜色.3  glClearColor(1.0f, 0.0f, 0.0f, 0.0f);  //真正的循环开始了,快去画画吧.  while ( !glfwWindowShouldClose(window) ) {  //刷新缓存  //你在画完的时候,如果不换一张纸,那么画就不好看了(重叠)  //所以说,我们是不是要换一张纸,不是么  glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);  glfwSwapBuffers(window); // swap the color buffers  //召唤所有事件  //键盘\鼠标的移动, 都是在这里处理的  //简单的说,你如果不用这个方法,你就只可以干看着.  glfwPollEvents();  } |

运行代码

|  |
| --- |
| public void run() {  //初始化GLFW  //为你打开图画软件  init();  //自由的作画  loop();  //作画完毕  //关闭窗口  glfwFreeCallbacks(window);  glfwDestroyWindow(window);  //关闭GLFW  glfwTerminate();  glfwSetErrorCallback(null).free();  } |