# 主要实现步骤

1. 将代码打包成jar包。
2. 整合资源文件
3. 将jar包打包成exe
4. 将jdk、资源文件、jar包转换后的exe三者再次打包成最终的exe。

# 准备软件

1. Idea：将代码打包成jar包（java形式的压缩包）
2. exe4j：将jar包转换成exe的工具。
3. innosetup：将游戏用到的图片，Java的运行环境和第二步打包的代码，组合成最终的安装程序。

**备注1：**exe4j和innosetup安装非常简单，傻瓜式下一步即可。

**备注2：**exe4j支持的JDK版本是8~11，其他版本的JDK不行，所以我也给大家提供了JDK11版本（这个JDK11在安装之后是不需要配置环境变量的，直接安装就行了）。

**备注3：**64位的操作系统用exe4j\_windows-x64\_6\_0\_2.exe

32位的操作系统用exe4j\_windows\_6\_0\_2.exe

**备注4：**打开我的电脑，右键空白处，点击属性，在弹出界面的中央就能查看自己电脑是64位的还是32位的。

# 软件下载地址

在《day18-API》--- 《资料》--- 《拼图游戏打包成exe》中

# 软件安装

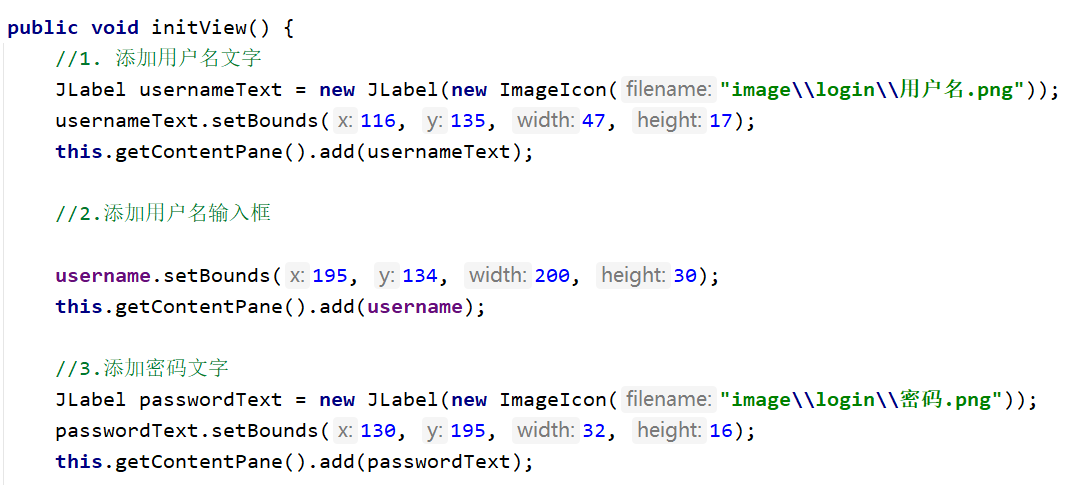
1. 确保idea已经安装完毕
2. 安装exe4j
3. 安装innosetup

傻瓜式安装，直接下一步即可，但是要记住安装路径。

# 代码改写：

代码中所有图片用到的路径需要把模块名删除，改写之后如下：

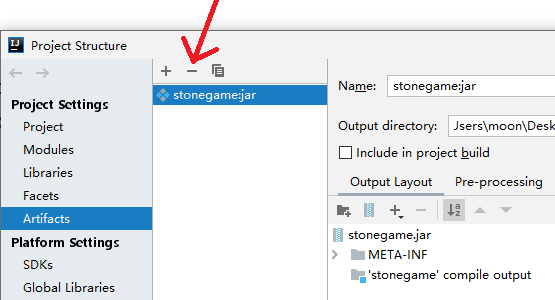
在不同的情况下，相对路径的相对位置是不一样的。



# 打包步骤：

## 第一步：代码打包成jar包

1. 点击File，再点Project Structure。
2. 点击Artifaces。
3. 下方图片箭头指向位置应为空白，如果有其他内容，可以选中之后，点击减号删除。



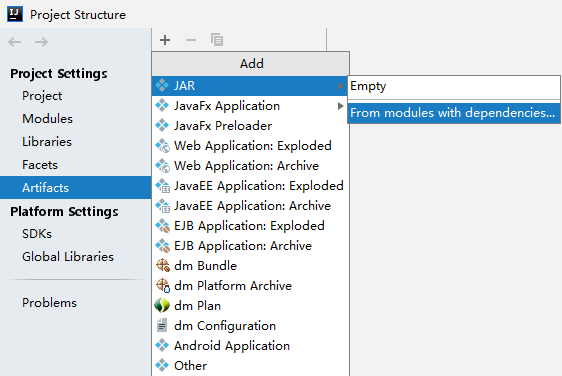
1. 选中左侧的Artifacts ，

点击中间上方位置的+

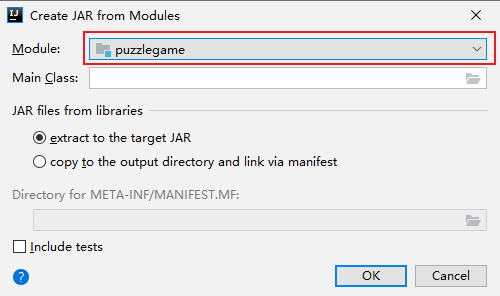
点击JAR

点击From modules with dependencies...

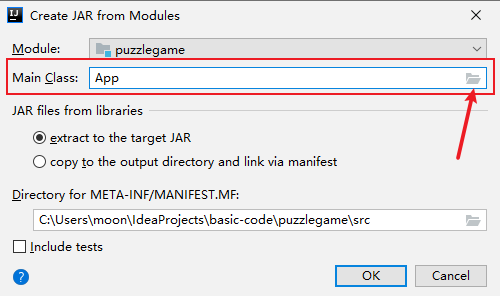
如下图所示：



1. Module：选择要打包的项目。

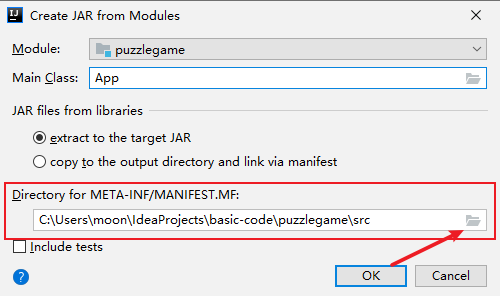


6，Main Class：点击红框最后的那个小图标，项目中是哪个main方法所在的类去启动项目，这里就选择哪个类。我的项目是由app里面的main方法启动的，所以我就选择App



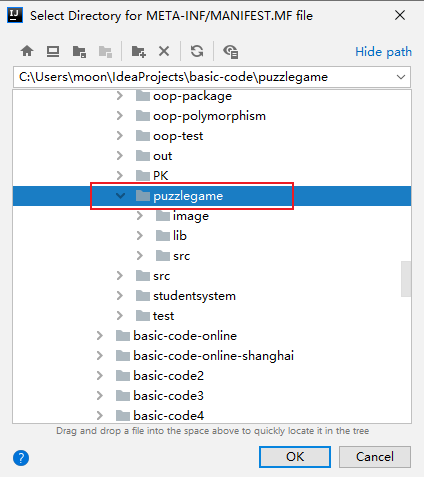
7，设置 META-INF/MANIFEST.MF。

点击箭头指向的图标进行设置。

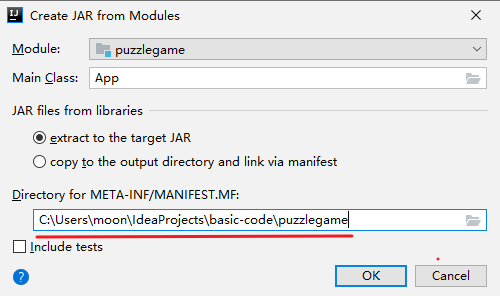


8，选择当前模块

点击OK



1. 此时这里的路径就是模块所在路径，点击OK。



1. 如果没有弹框报错，第十步可以跳过，直接看十一步。

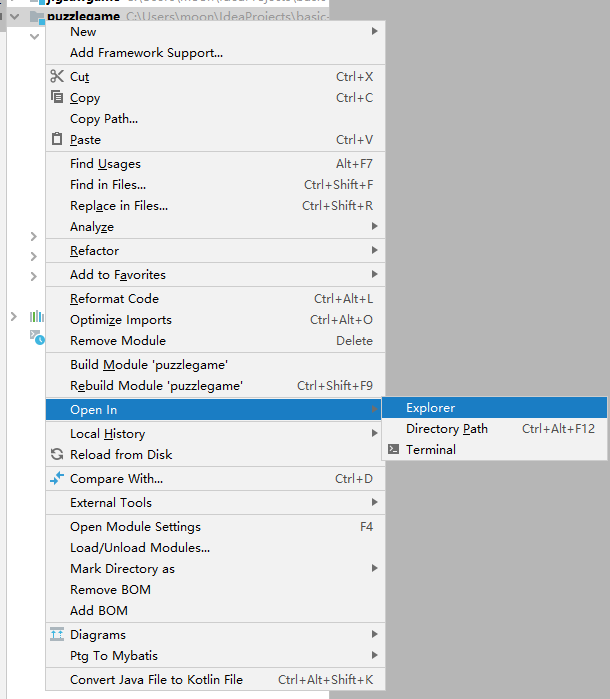
如果出现弹框报错，表示当前模块下已存在META-INF文件夹了。可以在本地找到已存在的META-INF文件夹，右键点击DELETE删除即可。

查找方式如下：

右键模块点击open in

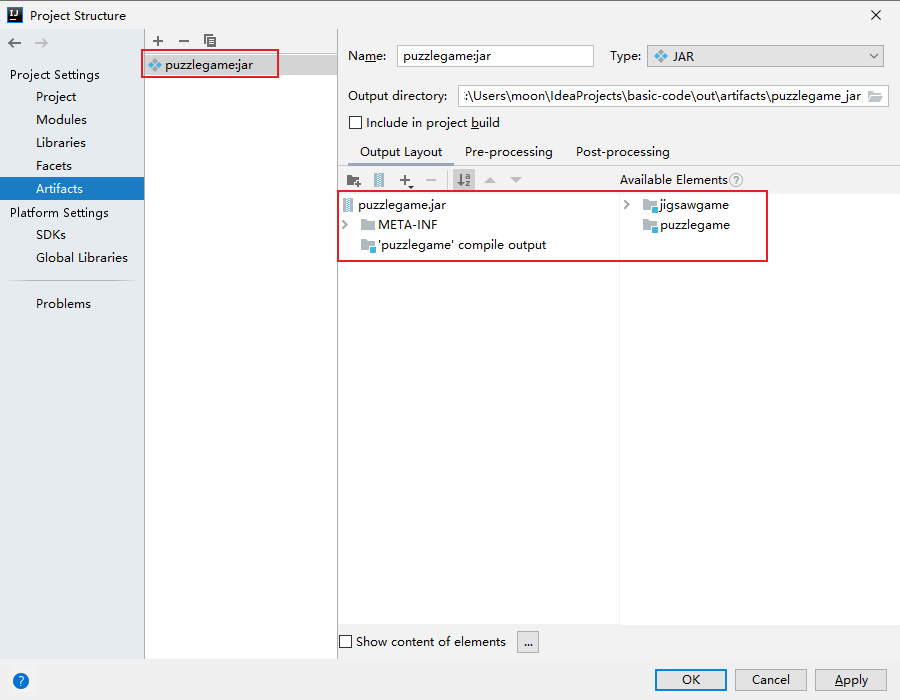
点击Explorer

在模块里面找到META-INF文件删除即可。



1. 如果没有报错，就出现一个puzzlegame：jar的提示。

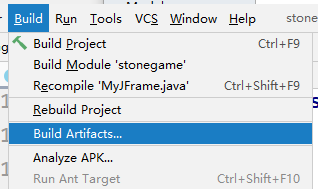
点击右下角的OK。



1. 在idea主界面上方，点击Build里面的Build Artifacts。

如果是灰色的不能按，请确定在上面第一步~第九步有没有配置好jar包的信息。

如果已经配置好了jar的信息，此处就可以正常点击。

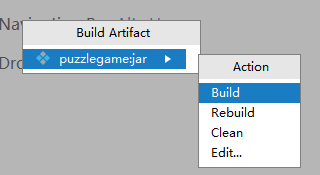


1. 在主界面正中央。

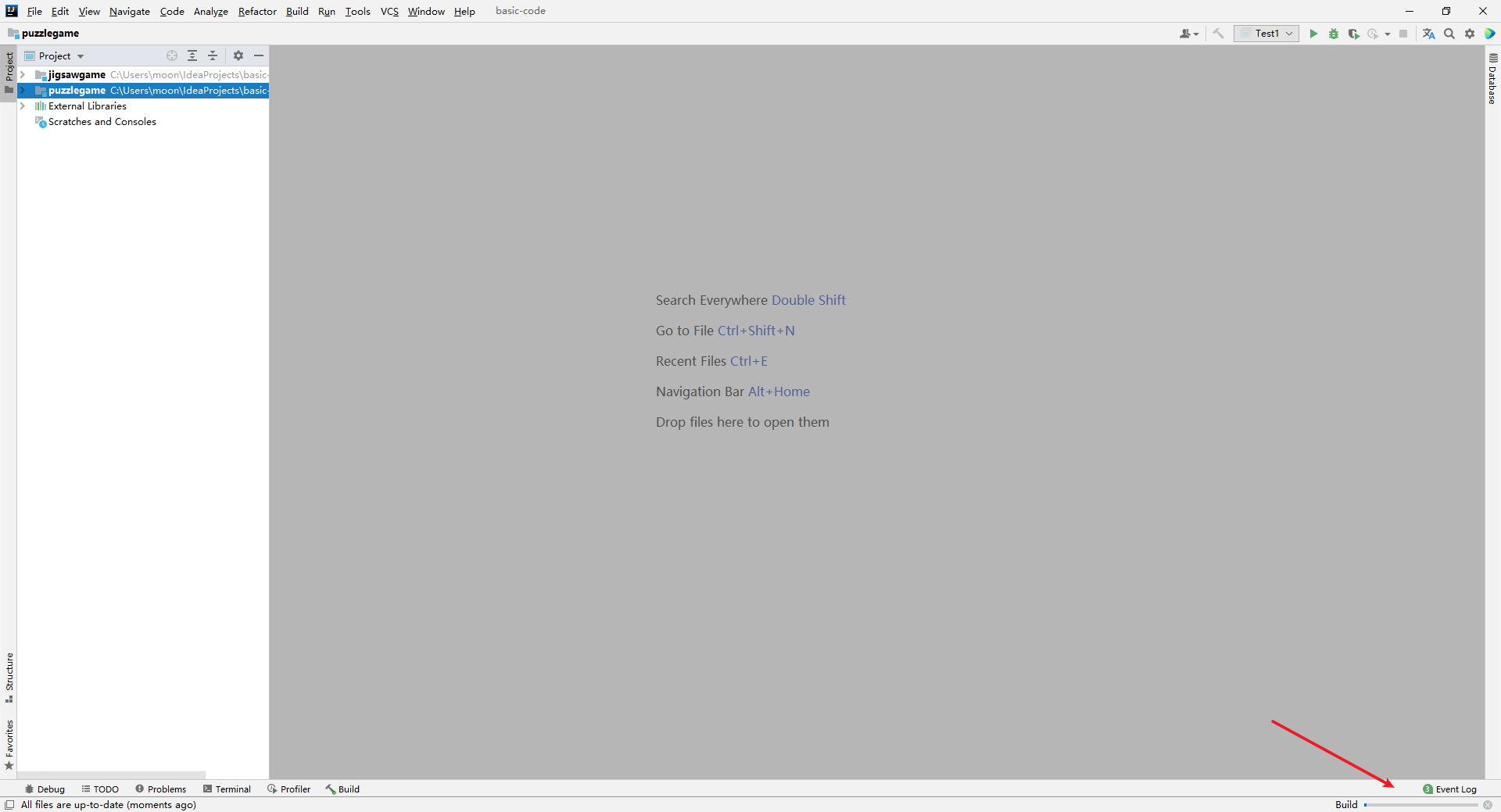
选择puzzlegame:jar。

再选择Build。

idea会帮我们生成jar包。



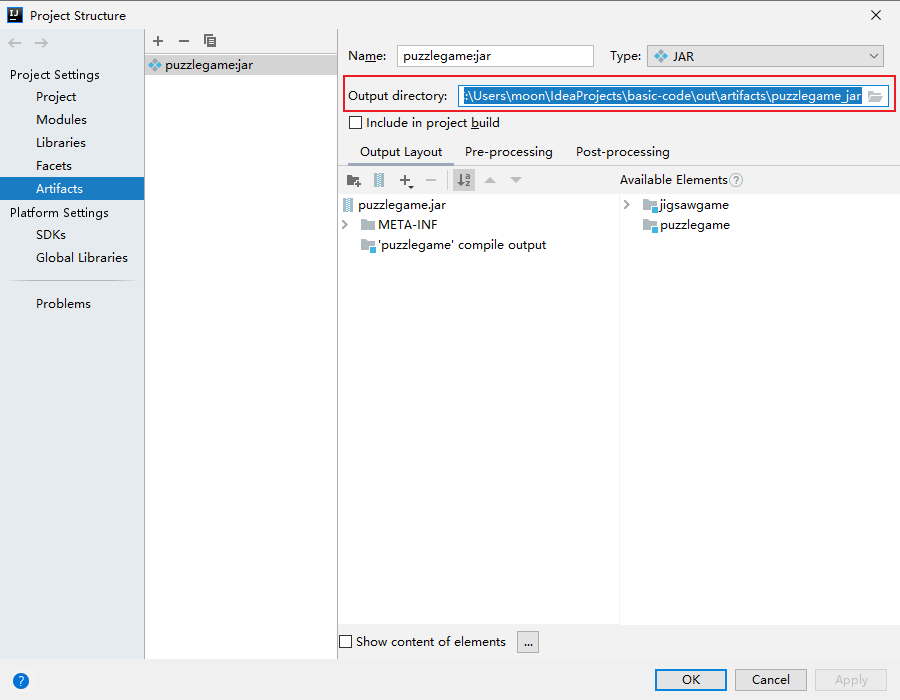
1. 等页面右下角的进度条结束，jar就已经生成完毕。



1. 点击File，再点Project Structure。

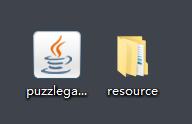
点击Artifaces。

在下图红色框中的路径下，可以找到生成好的jar包。



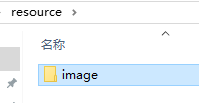
## 第二步：整合资源文件

1. 将第一步创建好的jar包拷贝到桌面上。
2. 在桌面上新建一个文件夹resource。



1. 将《拼图游戏》中的image文件夹粘贴到resource文件夹当中。

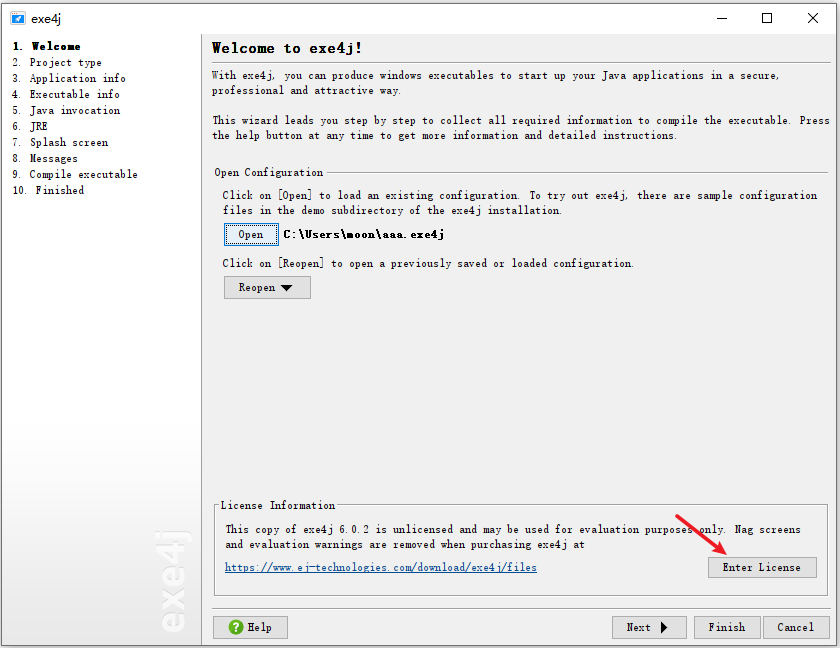
此时在桌面中的resource文件夹下的image文件夹下，就会看到游戏里面用到的所有图片。



## 第三步：将jar包打包成exe

1. 双击打开安装好的exe4j.exe，注册软件。

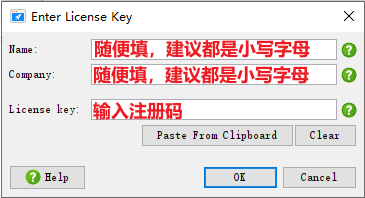
点击下面的Enter License



1. 输入用户名，公司名和注册码后点击ok

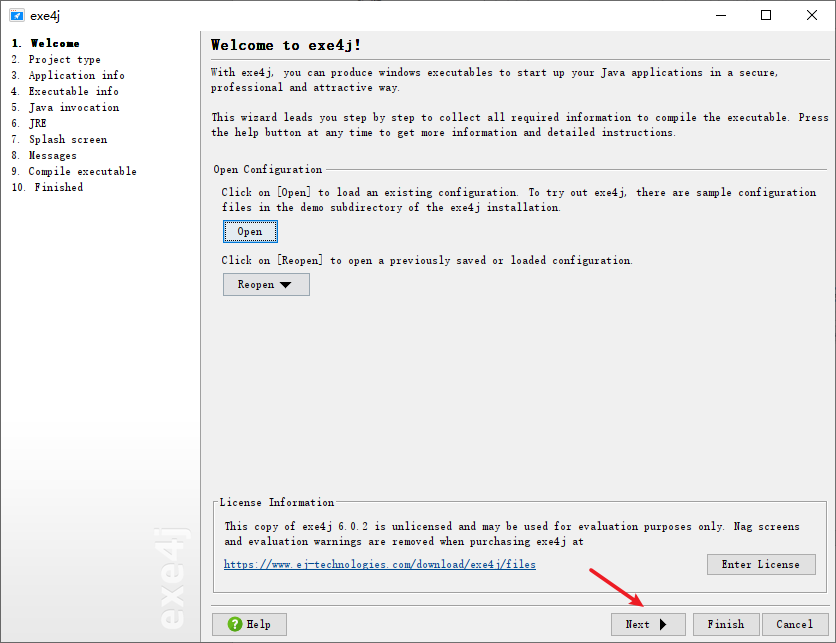
用户名，公司名随便填，最好都是小写字母。

注册码：L-g782dn2d-1f1yqxx1rv1sqd



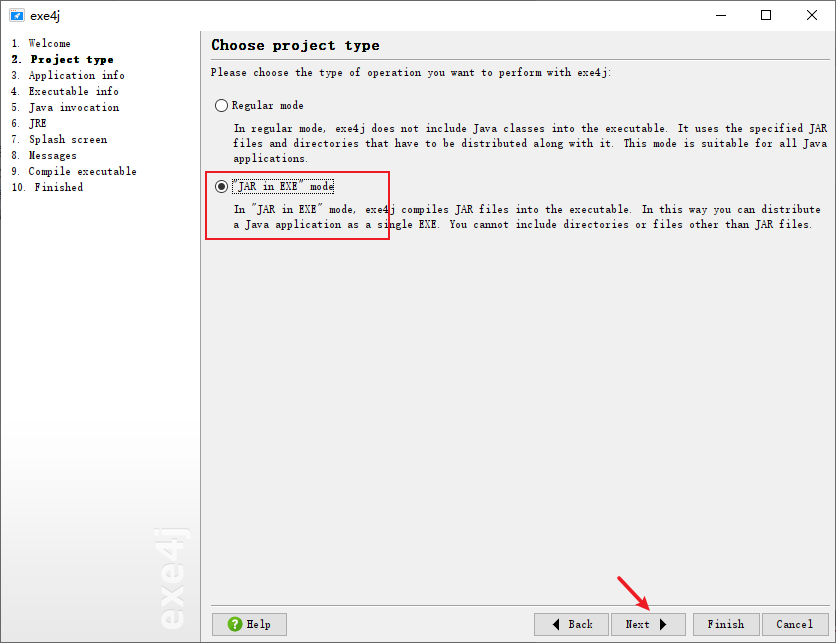
1. 注册完毕之后点击右下角的next

备注：exe4j.exe只识别JDK8到JDK11，如果安装时出现弹框报错，请检查JDK版本，资料文件夹中给大家提供了JDK11，有需要可以安装。（黑马程序员阿玮老师出品）



1. 选择JAVA转EXE。

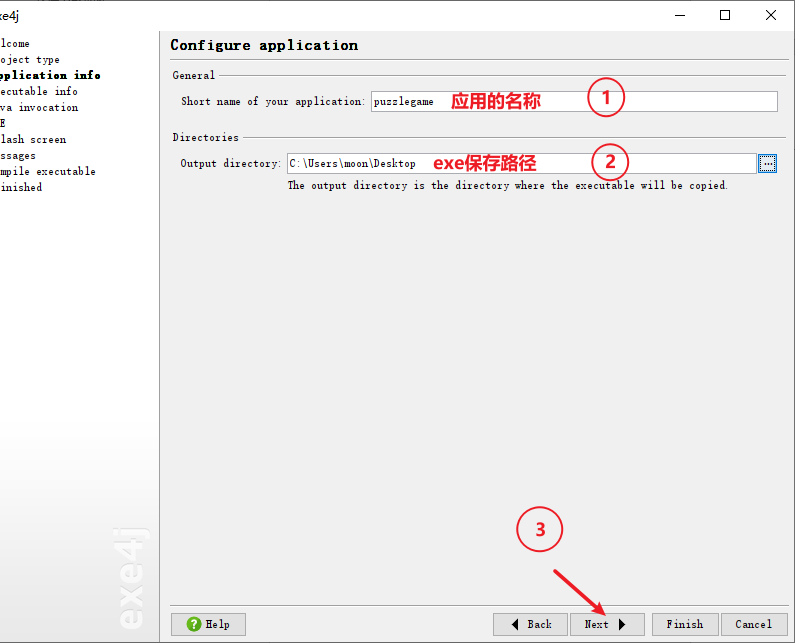
点击右下角的next



1. 输入名称puzzlegame

输出保存exe的路径，建议选择到桌面上。

点击右下角的next



1. 选择以图形界面的形式启动游戏

输入应用名称，puzzlegame

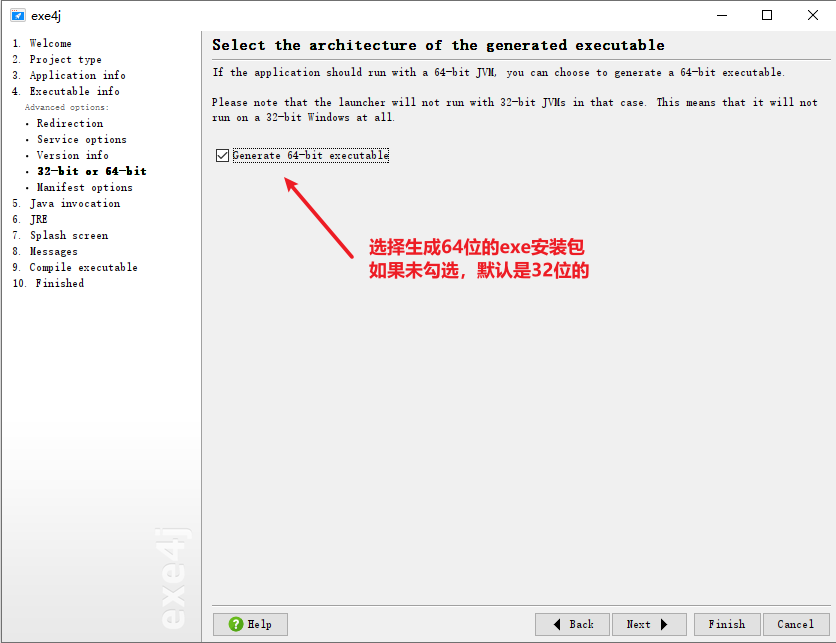
点击高级设置Advanced Options，选择32-bit or 64-bit



1. 勾选Generate 64-bit executable

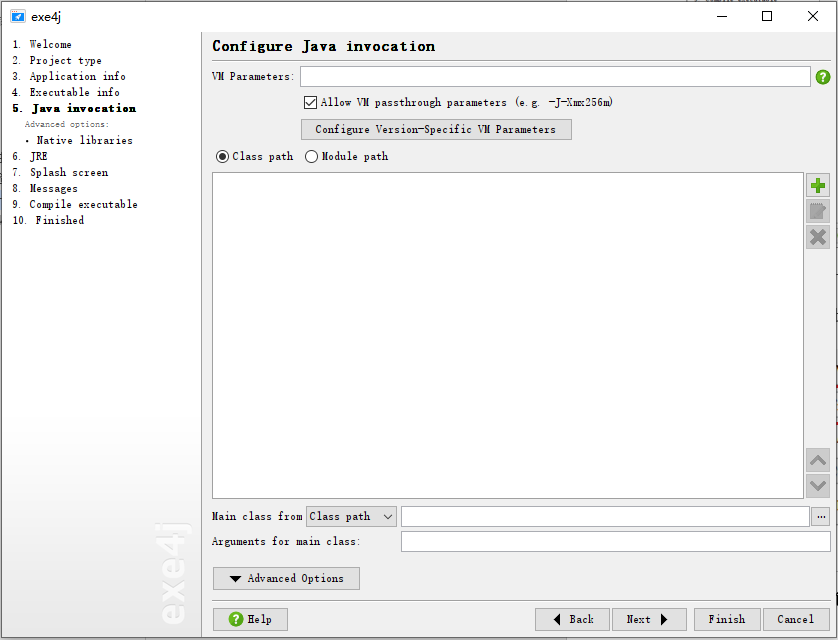
表示要生成64位的exe安装包，如果未勾选默认生成32位的安装包。

点击右下角next

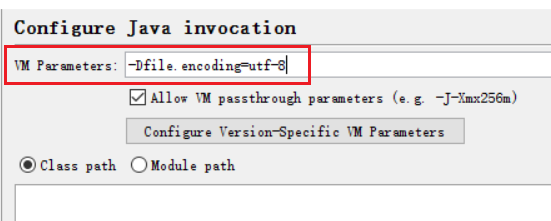


1. 然后一直下一步，一直出现如下界面

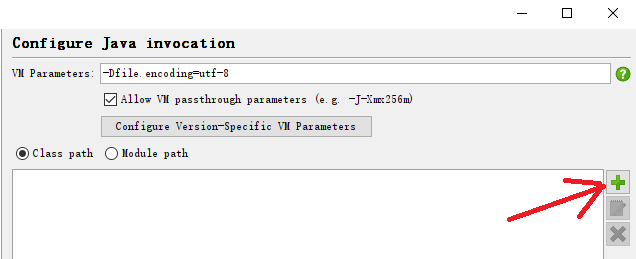
在本界面开始jar包并以及配置启动类。（黑马程序员阿玮老师出品）



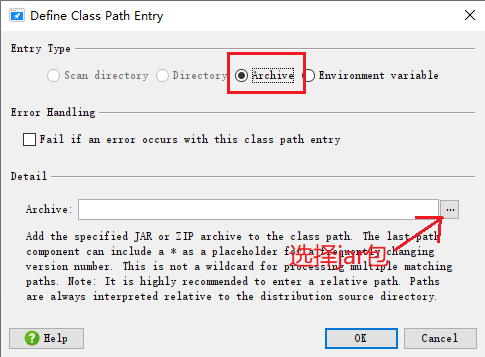
1. 第一行，VM参数配置的地方加上：-Dfile.encoding=utf-8



1. 点击右侧绿色的+，添加jar包

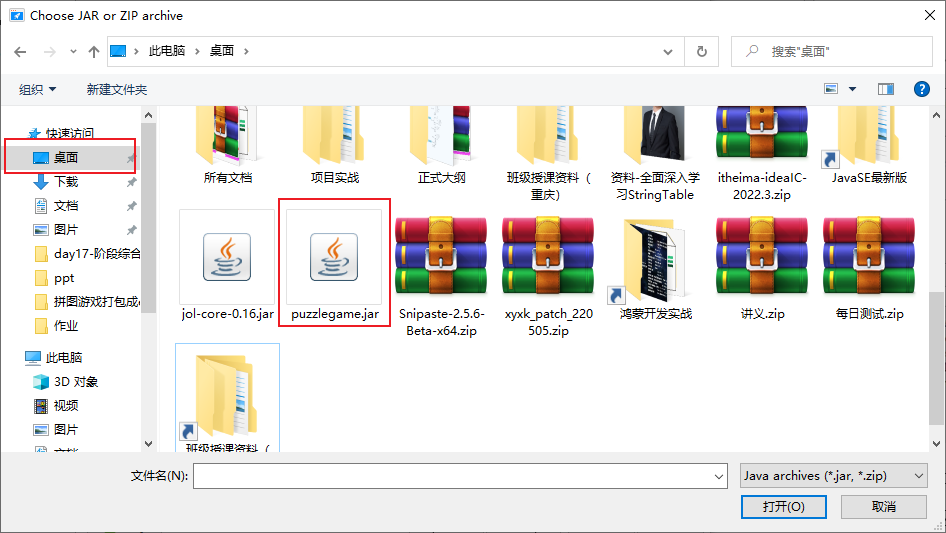


1. 添加jar包

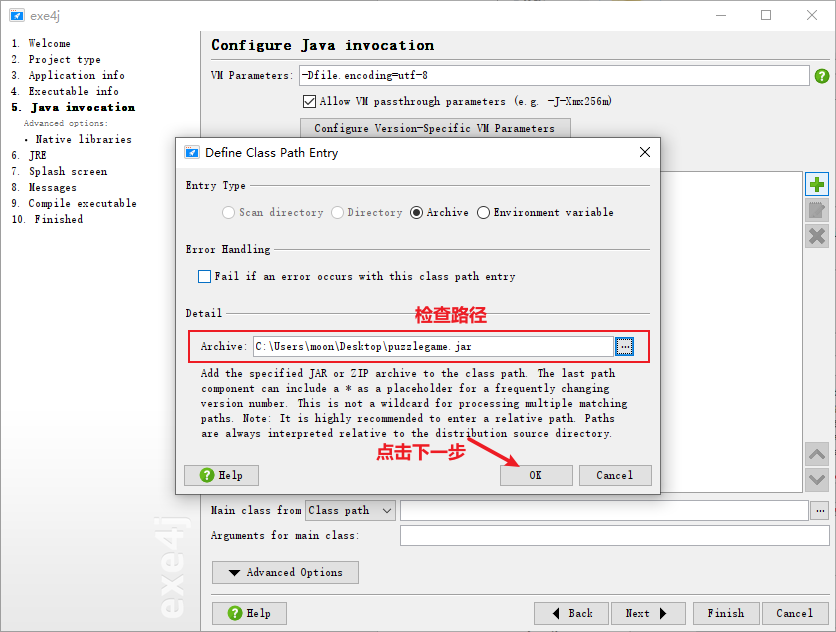


1. 选择桌面上的puzzlegame.jar。

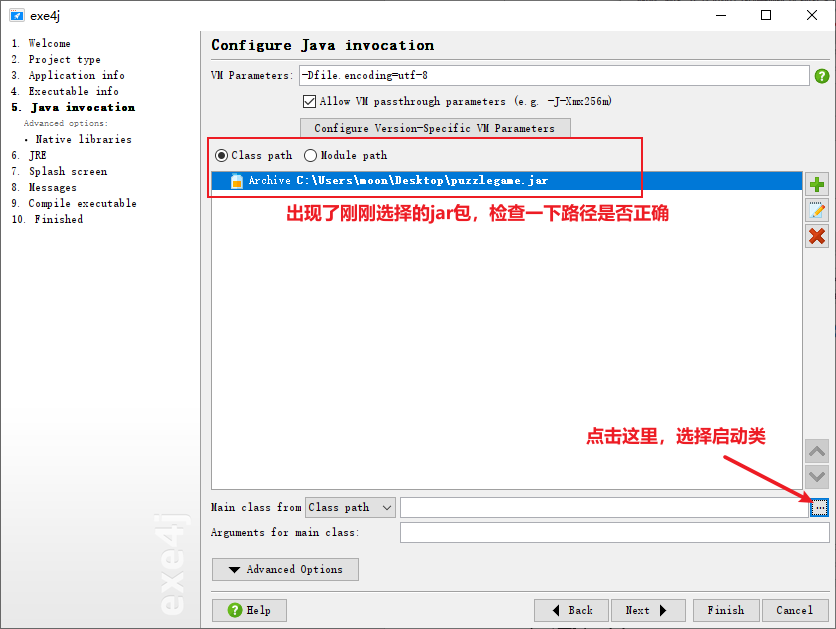
点击下面的打开。（黑马程序员阿玮老师出品）



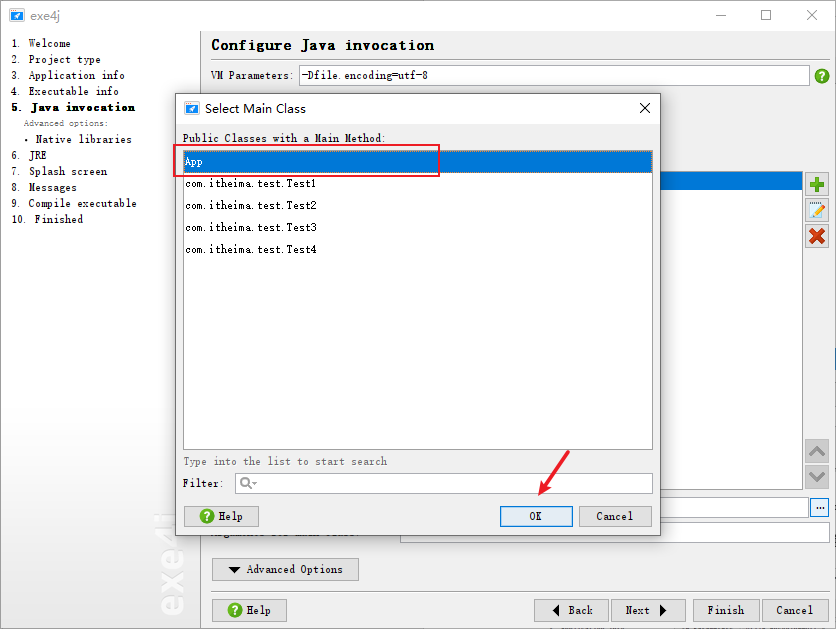
1. 检查路径，如果无误点击右下角的OK



1. 选择项目启动类

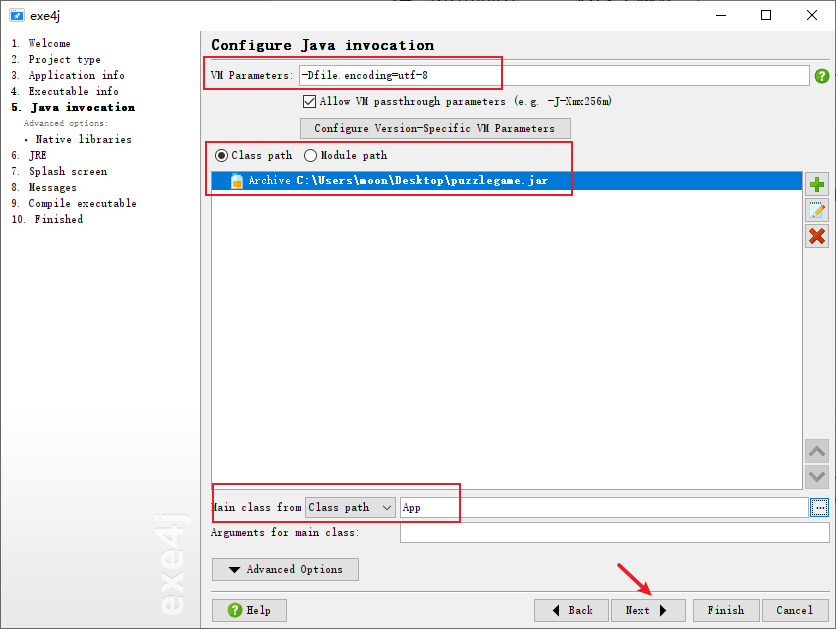


1. 因为程序主入口main方法写在App类中，所有选择app，并点击OK。



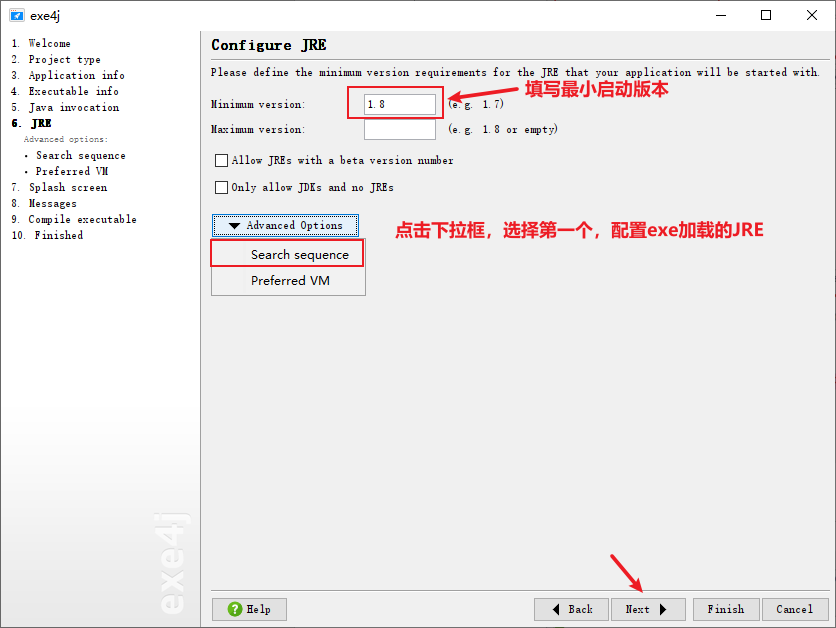
1. 本页面中，一共修改了三处。

三处全部操作完毕。点击右下角的next

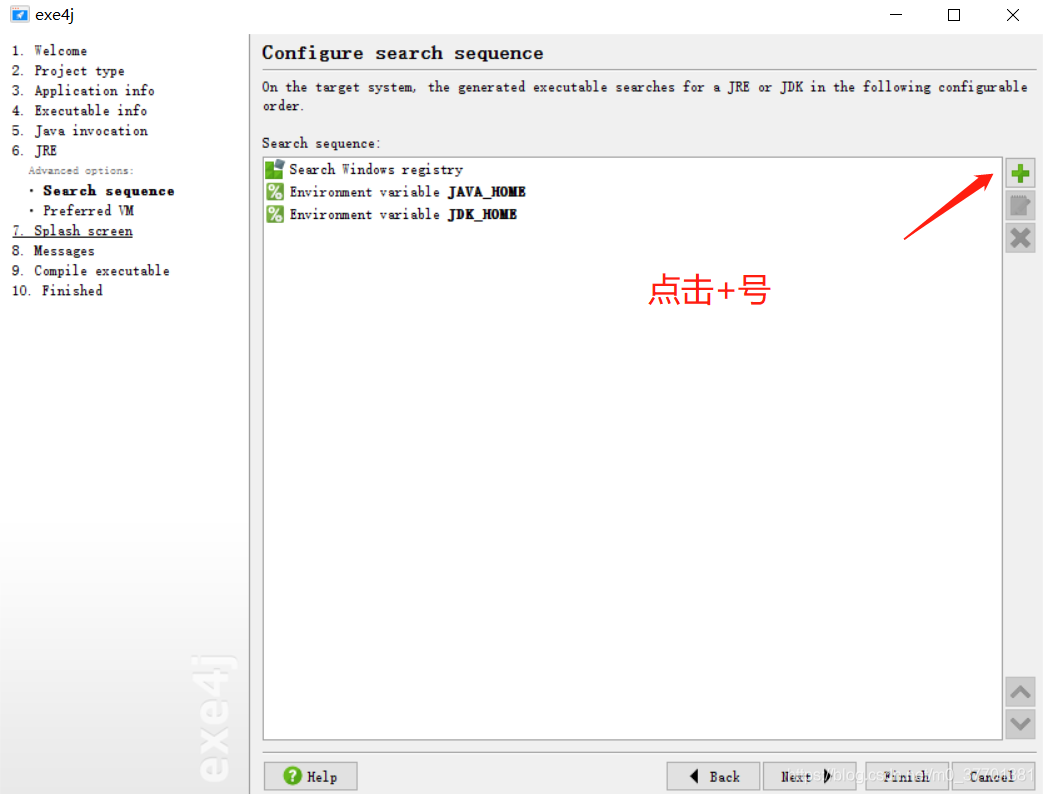


1. 填写最小启动的JDK版本。输入1.8

配置exe加载的JDK，选择第一个。



1. 点击+

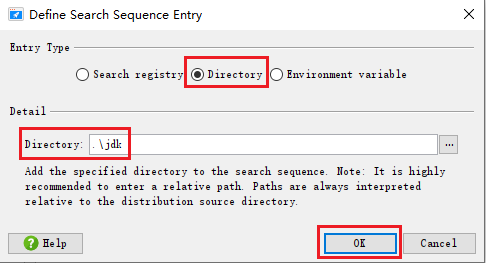


1. 选择Directory

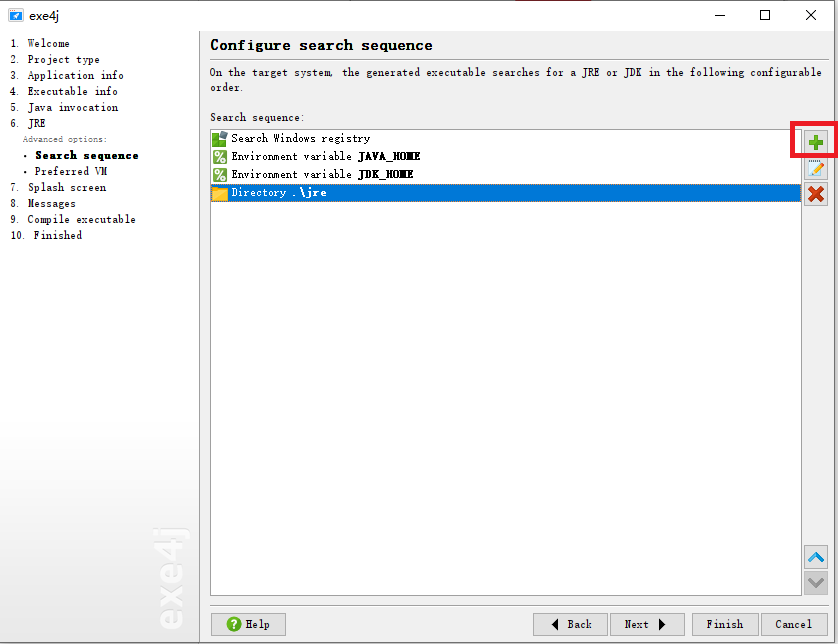
下面输入.\jdk

**（注意：输入点杠jdk，都要是英文状态下输入）**

点击OK



1. 再次点击+

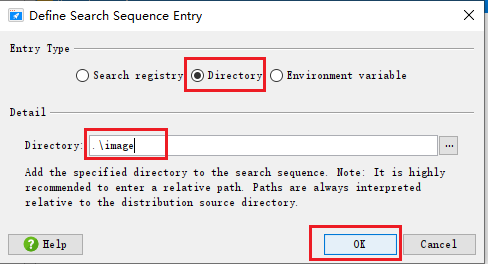


1. 选择Directory

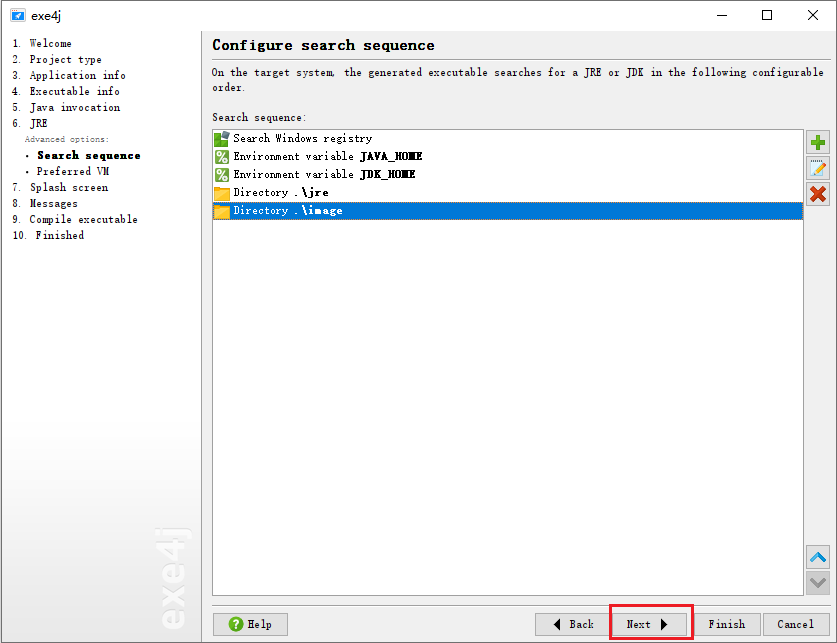
下面输入.\image

**（输入点杠image，都要是英文状态下输入）**

点击OK

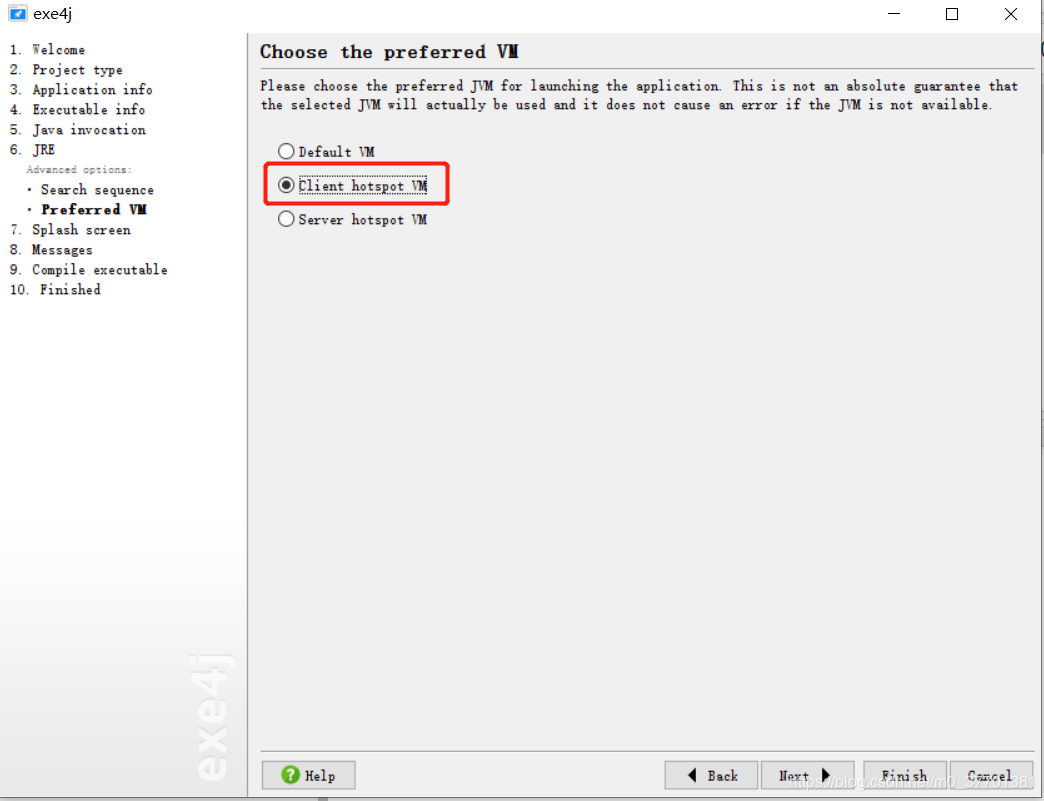


1. 点击右下角的next



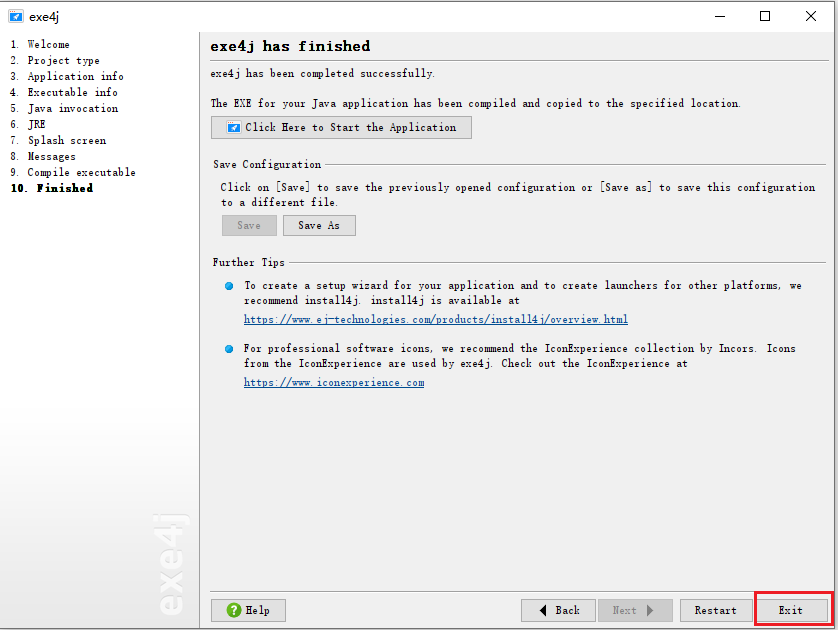
1. 选择Client VM

点击右下角的next

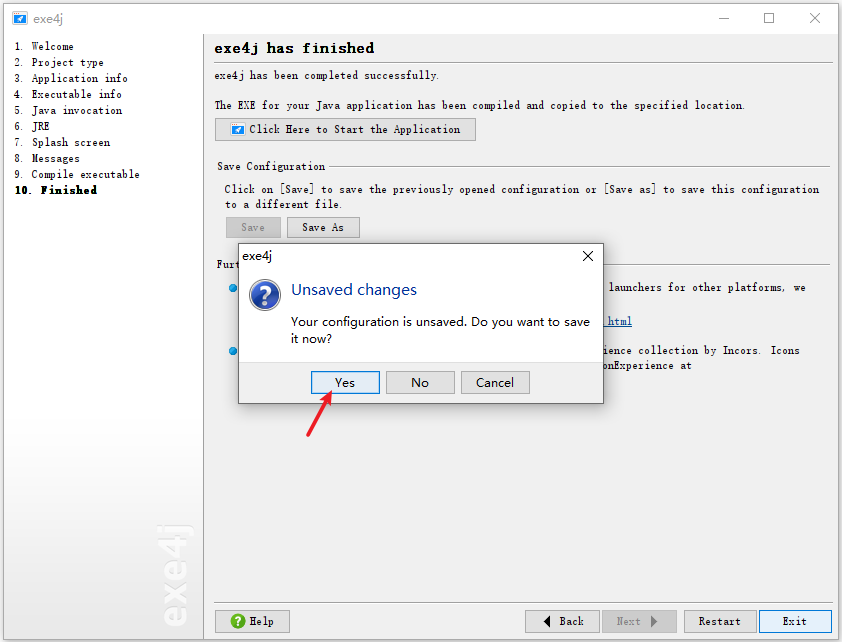


1. 然后一直下一步，最终出现如下界面

点击右下角的Exit退出。



1. 点击Exit后，会提升是否需要保存刚刚的配置信息，可以点击Yes，并选择一个路径进行保存。



1. 如果第三步选择的exe保存的路径是桌面，那么在桌面上

就能看到生成的puzzlegame.exe文件了。

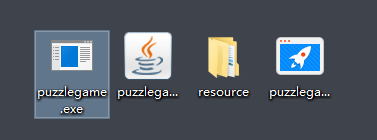
四个文件分别为：

左一：刚刚用jar生成的exe文件。

左二：idea生成的jar包

左三：游戏用到的资源图片

左四：刚刚用exe4J设置完毕之后保存的信息。



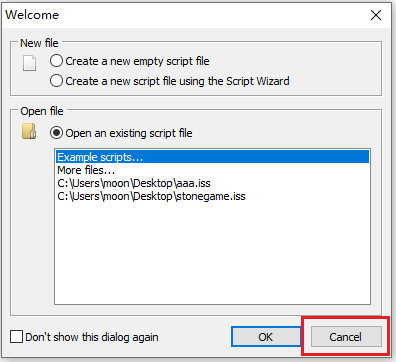
## 第四步：将jdk、资源文件、jar包转换后的exe三者再次打包成最终的exe。

刚刚，我们仅仅是把java代码变成了exe。下面我们要把游戏中依赖的资源文件，也就是使用到的所有图片，还有JDK三者再次打包成最终的exe，这样在没有jdk电脑环境下也能运行。

1，打开inno setup

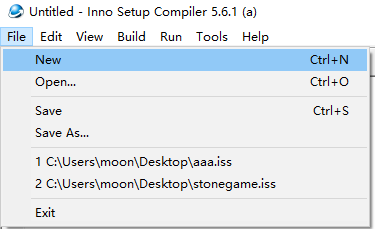


2，在欢迎页面点击右下角的关闭



1. 点击左上角的File

再点击NEW

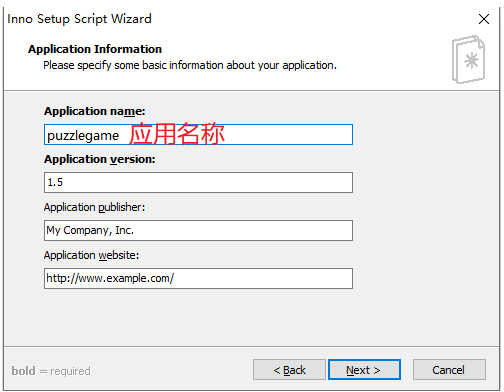


1. 点击next

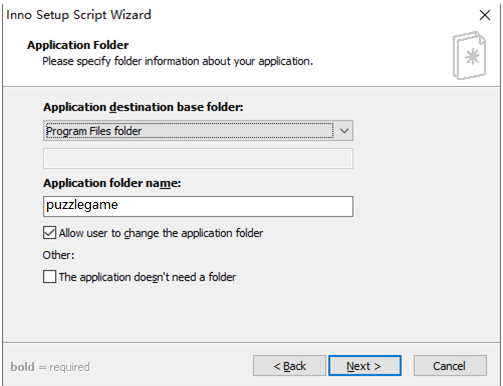


1. 输入应用名称puzzlegame

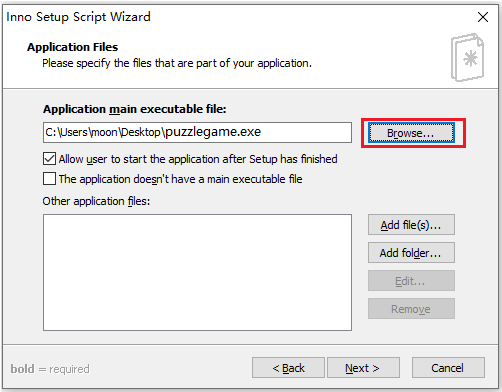
点击next



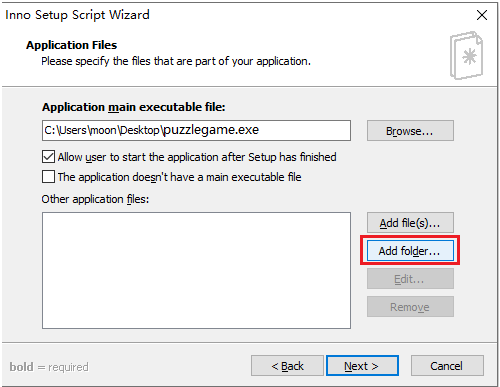
1. 不修改任何东西，直接点击next



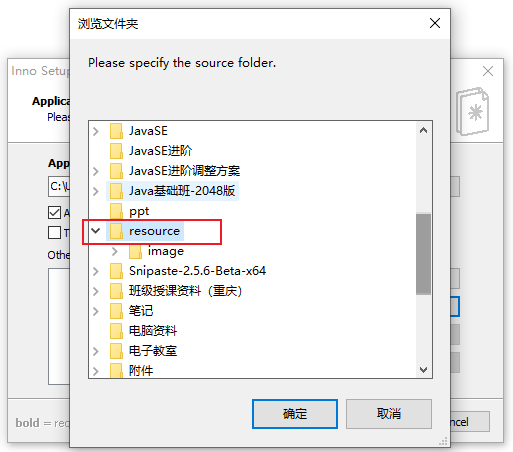
1. 点击这里，选择桌面上已经生成好的puzzlegame.exe



1. 点击Add folder

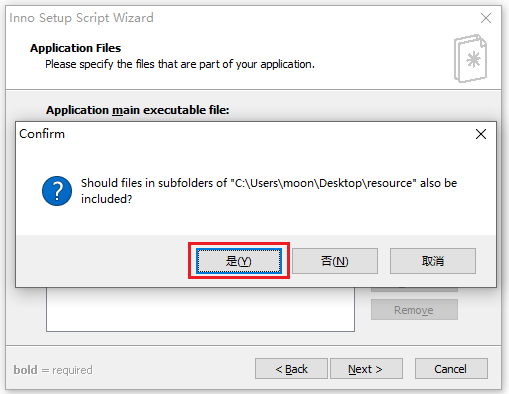


1. 选择桌面的resource，再点击确定。

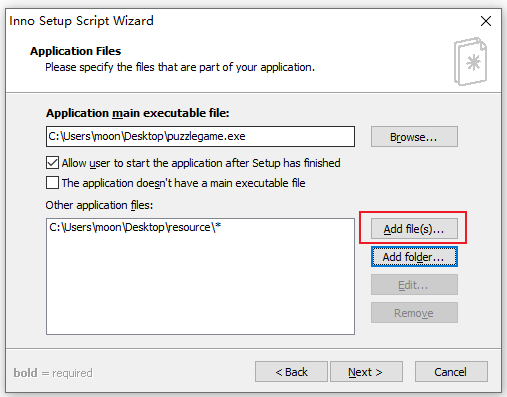


1. 如果出现下面弹框，则点击是。

如果没有出现也没有任何关系。

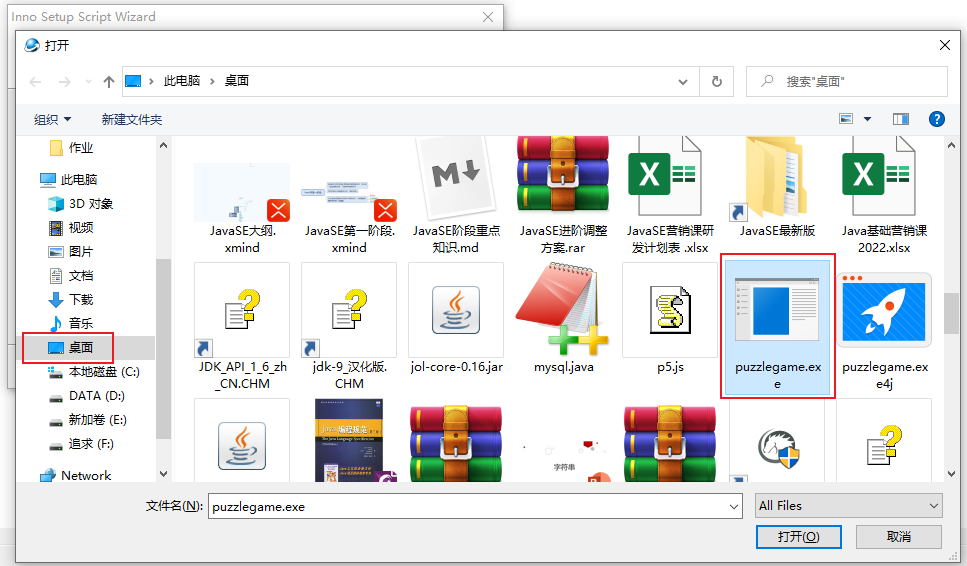


1. 再点击 Add file(s)…



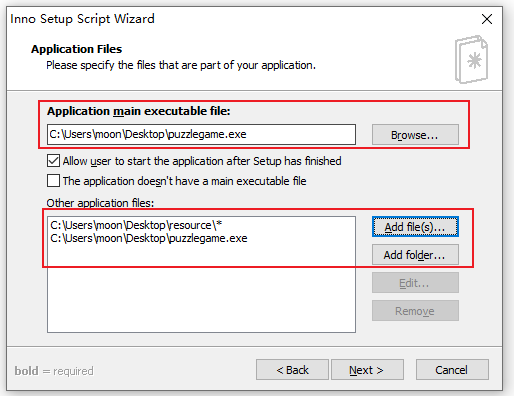
1. 选中桌面的puzzlegame.exe，再次添加一次。

点击下面的打开。

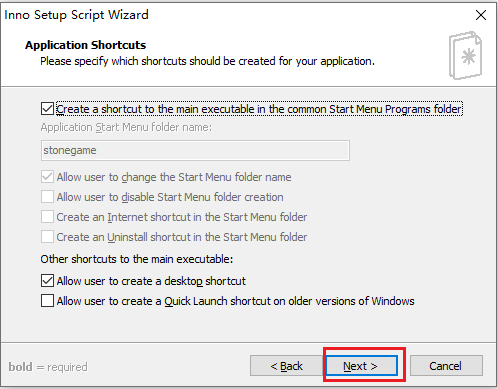


1. 在本页面中一共设置了三处地方。

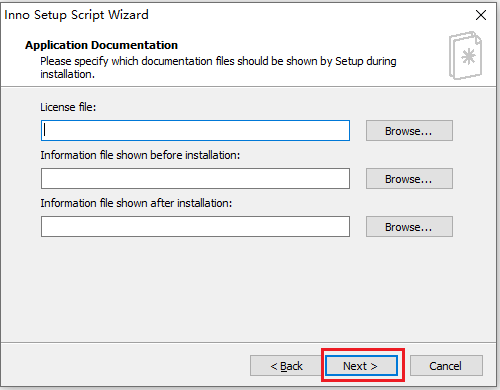
全部设置完毕，点击next。



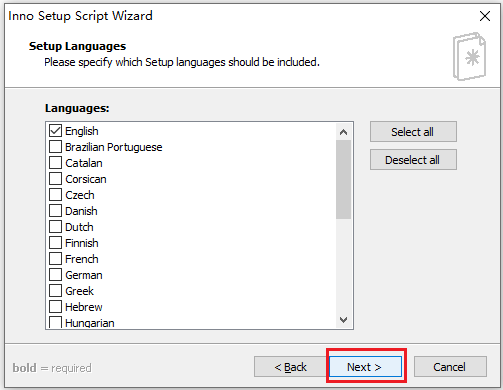
1. 默认不用选择，点击next



1. 默认不用选择，点击next



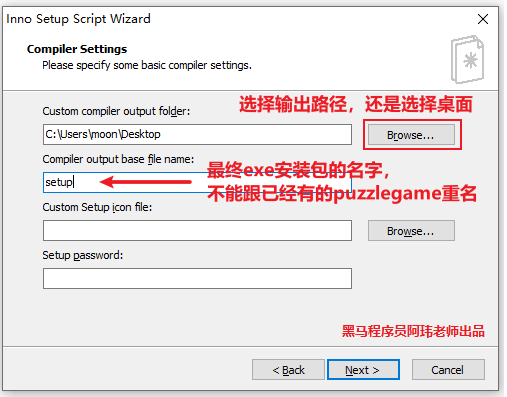
1. 选择语言，还是默认，点击next



1. 选择输出路径，还是选择桌面。

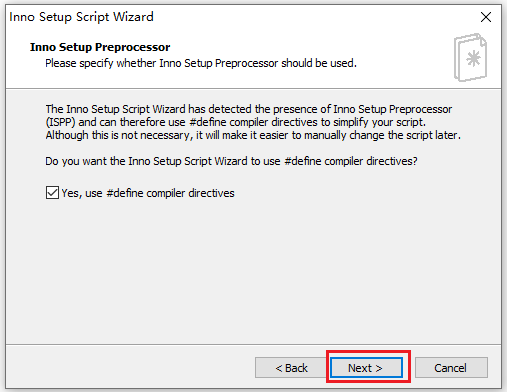
输入最终安装包的名字，不能跟已有的puzzlegame重名。

所以我写setup，再点击右下角next

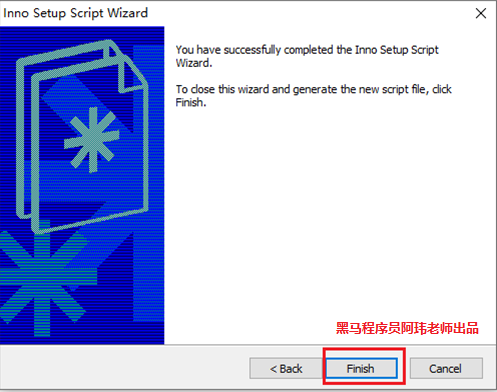


1. 默认点击next

有部分同学电脑不显示这一步，也没有关系。

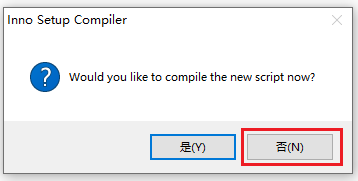


1. 完成，点击finish



1. 配置到最后一步了，脚本文件，到这里会弹出问你是否马

上编译，选择否，先把脚本写好再自己编译。

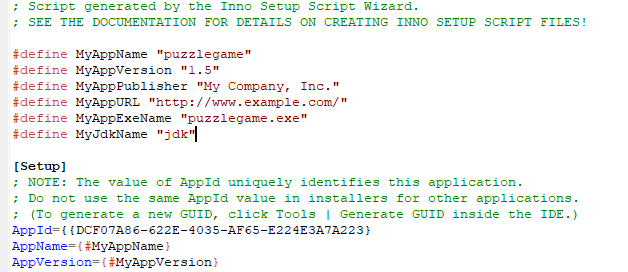


1. 上面红色箭头处添加一行脚本。

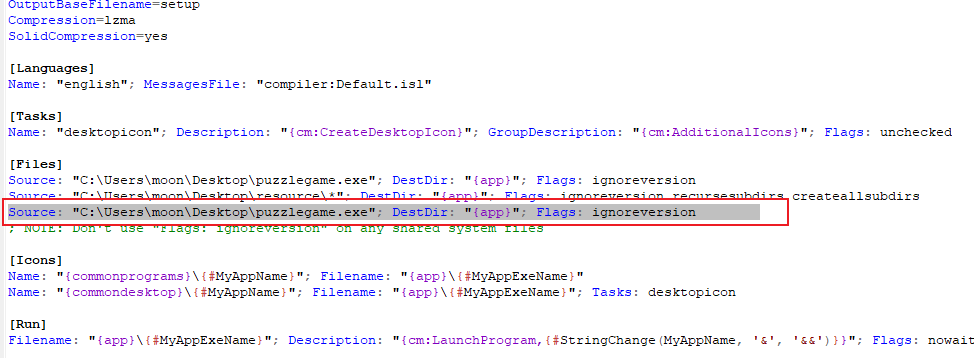
#define MyJdkName "jdk"

添加前：

1. 添加完毕之后，如下图所示



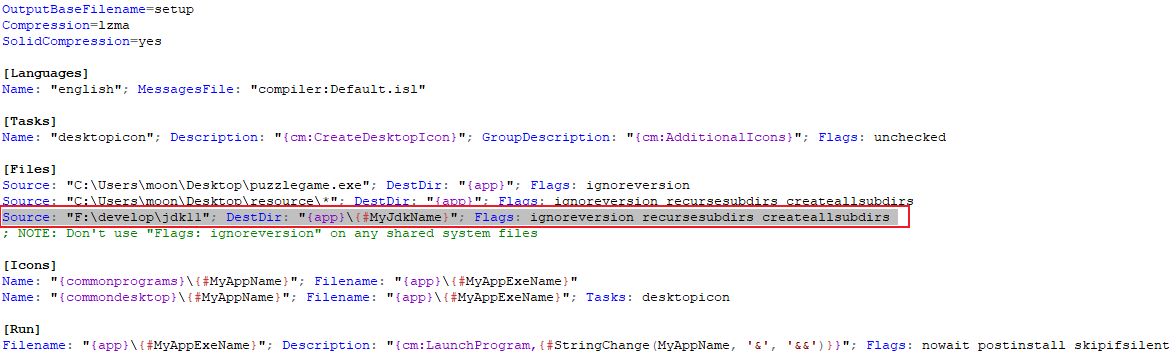
1. 往下拉，把有红色框起来的这一行删掉



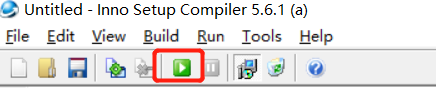
1. 在上一步删除位置添加一段行的文字

Source: "自己本地JDK路径\\*"; DestDir: "{app}\{#MyJdkName}"; Flags: ignoreversion recursesubdirs createallsubdirs

**千万注意：这段文字不要直接复制。根据自己电脑中JDK的安装位置进行修改，添加完毕之后如下图**

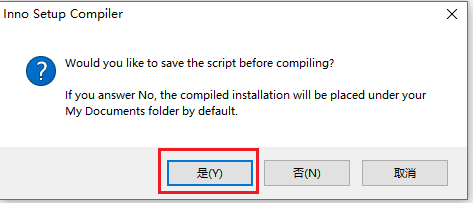


1. 点击上方的绿色按钮开始编译。

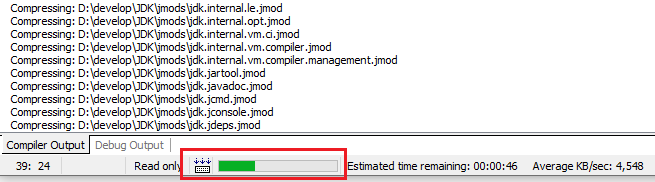


1. 此时会询问，是否需要保存。

可以点击是，选择一个位置保存一下刚刚修改之后的结果。



1. 然后等待绿色滚动条结束



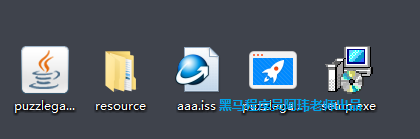
1. 当绿色滚动条结束后，会自动安装setup.exe文件。

此时可以点击否，先不安装。

在桌面上，会多了一个setup.exe文件和一个后缀名为iss的文件

setup.exe：打包成功的游戏安装包。

iss文件：就是刚刚设置的脚本文件。



1. 现在就可以把这个exe文件发给你的好基友了，他的电脑

上不需要安装JDK，直接双击这个安装包就可以玩游戏了。

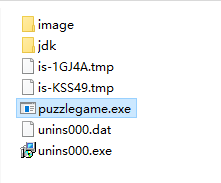
在安装的时候可以选择安装路径。

还可以在桌面生成快捷方式。

# 注意点：

1. 安装完毕之后，可以到安装目录去找puzzlegame.exe

双击就可以玩游戏了。刚刚桌面上的5个文件可以全部删除。以后双击puzzlegame.exe就可以玩游戏了。



1. 如果游戏运行之后，有效果但是无图片，或者部分图片无

法加载。如下图所示：此时背景图片无法加载。

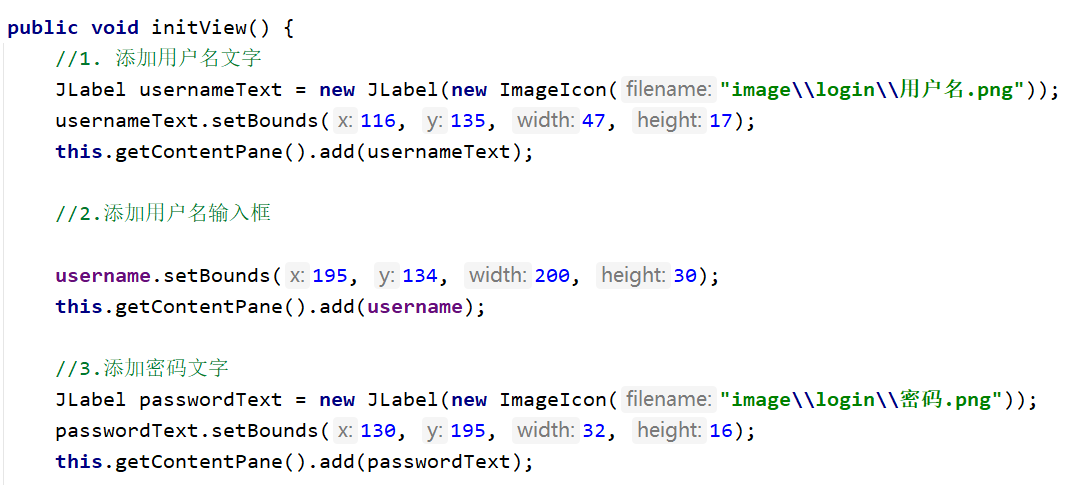


1. 修改步骤一：

检查安装目录下image文件夹中是否有该图片。如果没有将缺少的图片粘贴进去重新运行游戏。（黑马程序员阿玮老师出品）

修改步骤二：

检查代码，图片的路径需要把模块名删掉。



4，代码修改后，需要重新生成exe安装包，重新安装游戏。