

# Lab6

助教：钱宜嘉([19302010065@fudan.edu.cn](mailto:19302010065@fudan.edu.cn))

助教：邢惠锋([21110240051@m.fudan.edu.cn](mailto:21110240051@m.fudan.edu.cn))

助教：史磊([21210240314@m.fudan.edu.cn](mailto:21210240314@m.fudan.edu.cn))

## 背景说明

结合教学内容安排，本次 Lab 将为此前的 Karel 机器人系列添加图形界面。本次 Lab 工作量较大，请务必及早开始，避免迟交扣分。

## 目标

1. 完成 JavaFx 环境配置
2. 实现用户友好的界面交互

## JavaFx 安装与配置

首先确认自己的 JDK 版本在 11 及以上，否则后续步骤中可能会报错，再到 [JavaFx 下载网站](#) 下载对应版本的 JavaFx。

在 [JavaFx 配置教程](#) 中，有详细的安装步骤，根据自己的 IDE 选择红框中对应的板块，在子版块中选择 **Non-modular from IDE**，并按步骤操作，即可完成配置。其他板块可以参考，但是实际配置中一般不会用到。

### JavaFX

#### Getting Started with JavaFX

Introduction

Install Java

Run HelloWorld using JavaFX

Run HelloWorld via Maven

Run HelloWorld via Gradle

Runtime images

JavaFX and IntelliJ

- Non-modular from IDE
- Non-modular with Maven
- Non-modular with Gradle
- Modular from IDE
- Modular with Maven
- Modular with Gradle

JavaFX and NetBeans

JavaFX and Eclipse

JavaFX and Visual Studio Code

Next Steps

#### JavaFX and IntelliJ IDEA

This section explains how to create a JavaFX application in IntelliJ IDEA. JavaFX 15.0.1 and IntelliJ IDEA 2020.1 were used for the IDE screenshots.

Download an appropriate JDK for your operating system and set `JAVA_HOME` to the JDK directory. Refer to [Install Java](#) section for more information.

You can create a JavaFX modular or non-modular project and use the IDE tools, Maven or Gradle build tools.

#### Non-modular projects

##### IDE

Follow these steps to create a JavaFX non-modular project and use the IDE tools to build it and run it. Alternatively, you can download a similar project from [here](#).

Download the appropriate [JavaFX SDK](#) for your operating system and unzip it to a desired location, for instance `/Users/your-user/Downloads/javafx-sdk-11`.

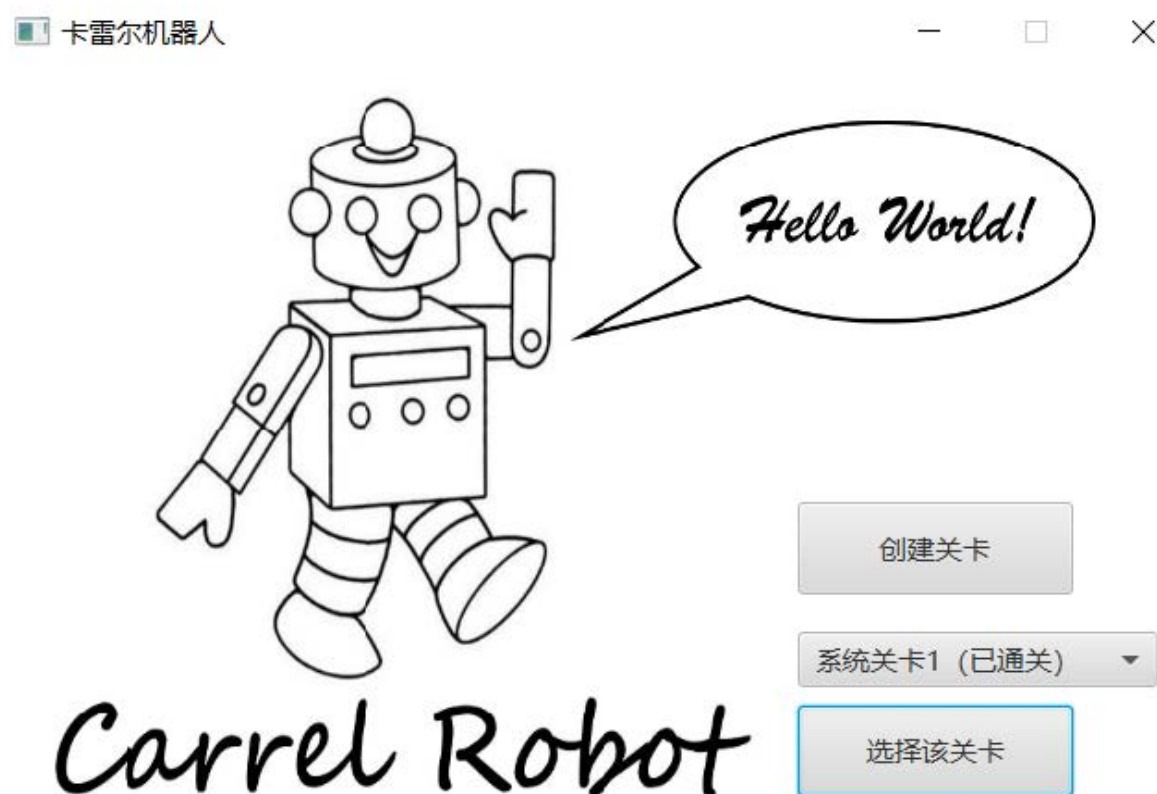
1. Create a JavaFX project

## 基础功能

所有的界面以用户友好为前提，并尽量保持美观，但是对美观没有过高要求。以下的图片仅供参考，不应全盘照抄。

### 欢迎界面

开始界面中有两个按钮和一个下拉框：`创建关卡`对应之前的 `NEW MAP`；下拉框中可以选择系统内置的三个关卡，对应之前的 STAGE 1 至 STAGE 3，所有用户自定义的关卡也需要在其中展示，每个关卡都应有一个对应的状态，如未游玩、已游玩、已通关；点击 `选择该关卡` 按钮可以进入下拉框中显示的关卡进行游玩，进入未游玩或已通关的关卡，应该从头开始，进入已游玩但未通关的关卡，需要从上次存档处开始。



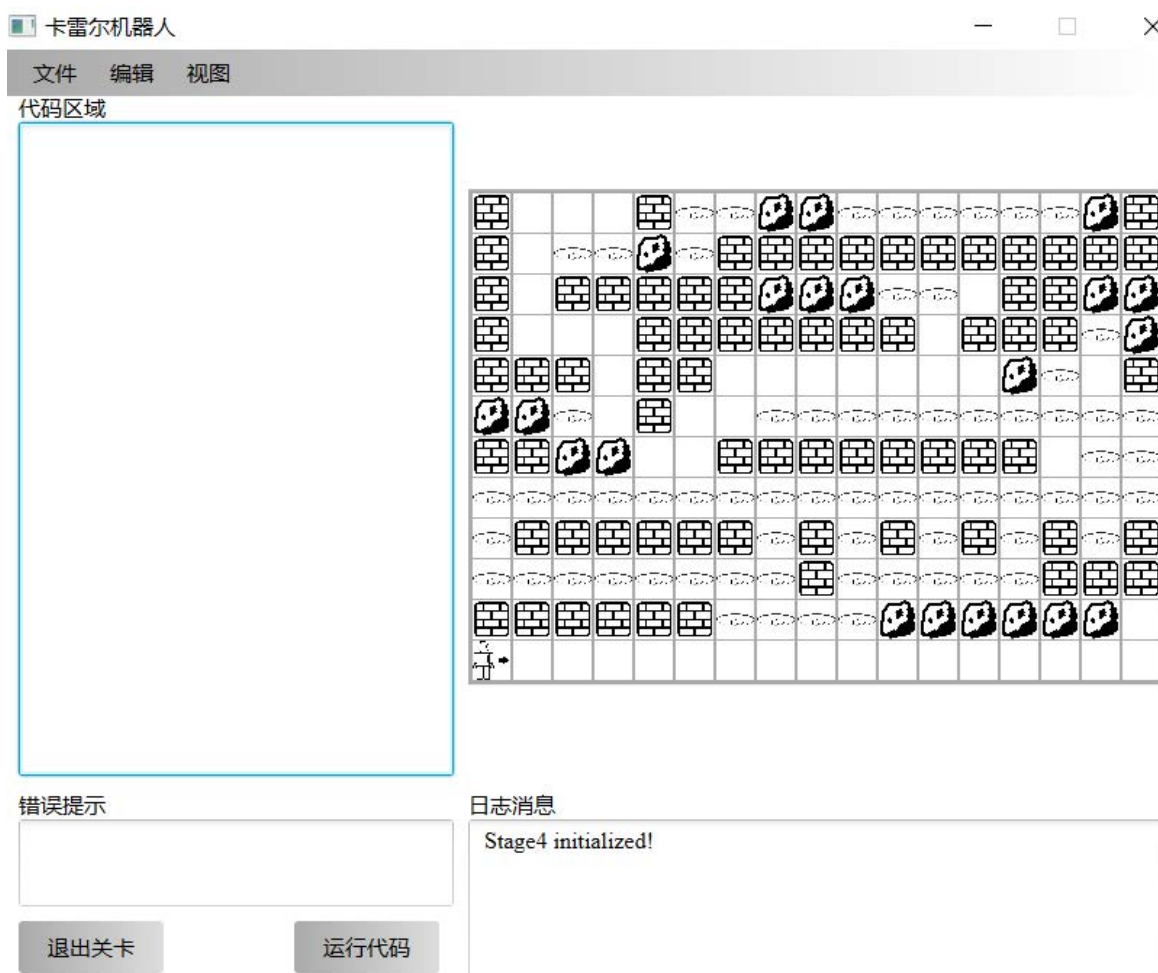
### 主界面

主界面中需要有以下六个区域：

1. 代码区域用于输入指令，对应之前的控制台输入，但是现在可以多行输入，而不是一次只能输入一行，每条指令都需要以分号结尾，具体指令的格式如下：

```
turnLeft();
move();
if (noRockPresent()) {
    move();
}
turnRight() {
    turnLeft();
    turnLeft();
    turnLeft();
}
turnRight();
```

2. 地图区域需要用相应的图片代替之前的地图标记，展示当前的地图状况，所有的地图标记都需要有对应的图片，包括上下左右四个方向的机器人、石头、墙壁、地面、陷阱、被填补的陷阱。
3. 错误提示对应之前的报错信息，如指令输入错误、机器人越界等。
4. 日志消息中打印正确的游戏信息，如关卡初始化完成、用户自定义方法已经定义成功、`showInformation()` 的结果、游戏胜利或失败等。
5. 菜单栏中需要提供切换关卡、存档、读档、游戏帮助功能。切换关卡时，如果用户没有存档，需要询问是否存档，如果不进行存档，关卡状态应与进入关卡前保持一致；存档需要保存当前的地图状态、代码区域的输入、日志和报错信息；读档需要还原存档中的所有信息，就像没有退出过一样；游戏帮助中需要展示游戏的游玩方法，如游戏中包含哪些指令，应该如何输入，游戏胜利或失败的条件，游戏目标等，游戏帮助应该在新的窗口中显示。
6. 功能按钮需要提供退出关卡和运行代码的功能，点击 `运行代码` 后，需要需要执行代码区域中的所有代码，执行完成后展示对应的错误提示和日志消息，运行代码不会清空代码区域、错误提示和日志消息，代码区域留给用户修改，之前的错误提示和日志消息需要一直保留，以便用户参考（类似于在 IDE 中写程序 debug）。



## 加分项

所有的加分项都不是必做的，有较高的自由度，具体的设计和操作逻辑都由自己创作，视完成情况给分，完成加分项可以补足基础功能中被扣除的分数，上限 20 分。

## 背景音乐与音效

所有按钮的点击需要有对应的音效，游戏时需要播放背景音乐，所有的音效和音乐都需要符合情境，不要做成恐怖游戏。更进一步，可以在游戏内调节音量大小，暂停播放，或切换不同的音乐。

## 登录界面

用户可以创建多个账号，有多个游戏进度。

## 快捷键

用户可以通过快捷键进行快速操作，如按 `ctrl+Enter` 执行指令等，系统自带的快捷键如 `ctrl+c` 等不算在内。

## 游戏时间统计

在界面中加入当前的游戏时间，存档时需要保存游戏时间，读档时还原，在开始界面选关时显示花了多久通关等信息。

## 其他合理的优化

优化可以是功能上的拓展，也可以是更用户友好的交互方式，但是不要在这部分花过多的时间，舍本逐末。

## 文档

本次 Lab 要求书写文档，文档内容包括但不限于：代码运行截图，类设计，程序实现思路，心得与体会。文档应当简洁明了，逻辑清晰，图片应当有对应的文字说明。文档不宜过长，不鼓励内卷，最终得分与字数多少无关。

## 提交与评分标准

本次 Lab 截止时间：**2023 年 6 月 11 日**。每逾期一天（向上取整，即逾期一分钟也视为一天），在最终成绩的基础上扣 5 分，最高扣 40 分。

提交内容为 `Lab6_学号_姓名.zip` 和 `Lab6_学号_姓名.pdf`，如 `Lab6_22302011234_张三.zip` 和 `Lab6_22302011234_张三.pdf`，上传至 eLearning。压缩包中应包含所有代码，pdf 文件为实验文档。上传文件命名不规范扣 2 分，例外：多次提交文件名相同的文件，eLearning 会自动重命名，在原来的文件名后面加数字，此类问题不属于文件命名不规范，不扣分。

本次 Lab 满分 100 分，得分点如下：

得分点	分值
图形界面正常运行（按钮点击、文本输入正确）	20
存档、读档、切换关卡正确	20
地图场景正确显示	10
错误提示正常	10
日志消息正常	10
界面美观（按钮对齐，字体统一，有一定的插图等，不需要非常精致）	10
文档与代码风格	20
加分项	20
所有前置 Lab 的功能均正常	达成不得分，未达成倒扣分，一个功能 5 分，最高扣 40 分