

**2019年春季学期  
计算机学院《软件构造》课程**

**Lab 6实验报告**

|  |  |
| --- | --- |
| 姓名 | 林之浩 |
| 学号 | 1170300817 |
| 班号 | 1703008 |
| 电子邮件 | 630073498@qq.com |
| 手机号码 | 18065053516 |

**目录**

[1 实验目标概述 1](#_Toc1393927)

[2 实验环境配置 1](#_Toc1393928)

[3 实验过程 2](#_Toc1393929)

[3.1 ADT设计方案 2](#_Toc1393930)

[3.2 Monkey线程的run()的执行流程图 4](#_Toc1393931)

[3.3 至少两种“梯子选择”策略的设计与实现方案 5](#_Toc1393932)

[3.3.1 策略1 5](#_Toc1393933)

[3.3.2 策略2 5](#_Toc1393934)

[3.3.3 策略3（可选） 6](#_Toc1393935)

[3.4 “猴子生成器”MonkeyGenerator 6](#_Toc1393936)

[3.5 如何确保threadsafe？ 6](#_Toc1393937)

[3.6 系统吞吐率和公平性的度量方案 7](#_Toc1393938)

[3.7 输出方案设计 8](#_Toc1393939)

[3.8 猴子过河模拟器v1 10](#_Toc1393940)

[3.8.1 参数如何初始化 10](#_Toc1393941)

[3.8.2 使用Strategy模式为每只猴子随机选择决策策略 11](#_Toc1393942)

[3.9 猴子过河模拟器v2 12](#_Toc1393943)

[3.9.1 对比分析：固定其他参数，选择不同的决策策略 12](#_Toc1393944)

[3.9.2 对比分析：变化某个参数，固定其他参数 14](#_Toc1393945)

[3.9.3 分析：吞吐率是否与各参数/决策策略有相关性？ 17](#_Toc1393946)

[3.9.4 压力测试结果与分析 17](#_Toc1393947)

[4 实验进度记录 19](#_Toc1393948)

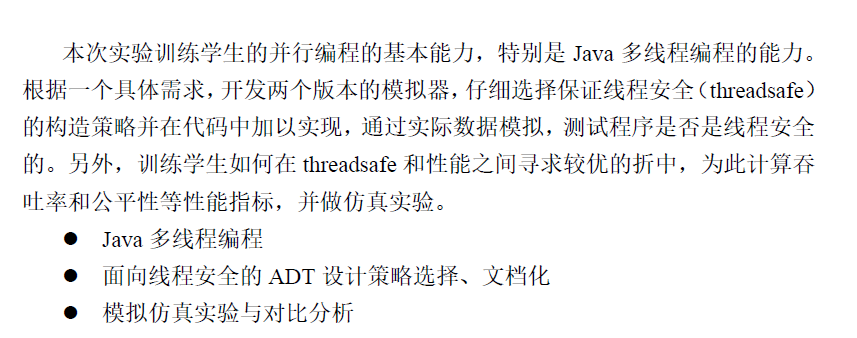
[5 实验过程中遇到的困难与解决途径 22](#_Toc1393949)

[6 实验过程中收获的经验、教训、感想 22](#_Toc1393950)

[6.1 实验过程中收获的经验和教训 22](#_Toc1393951)

[6.2 针对以下方面的感受 22](#_Toc1393952)

# 实验目标概述



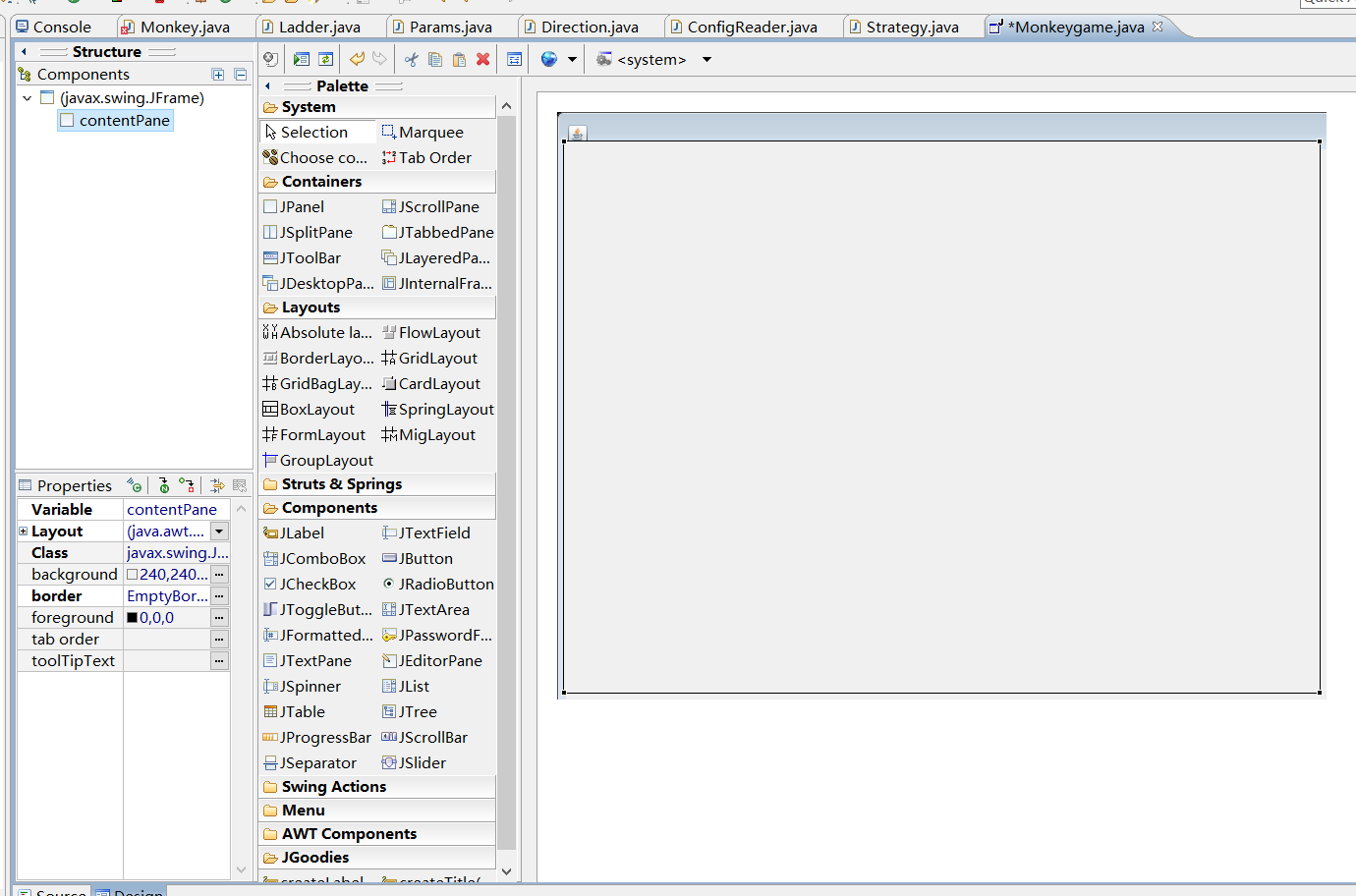
# 实验环境配置

简要陈述你配置本次实验所需环境的过程，必要时可以给出屏幕截图。

特别是要记录配置过程中遇到的问题和困难，以及如何解决的。

在<https://www.eclipse.org/windowbuilder/>下载了gui插件windowbuilder。

安装。用来可视化设计gui

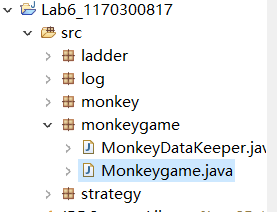


在这里给出你的GitHub Lab6仓库的URL地址（Lab6-学号）。

<https://github.com/ComputerScienceHIT/Lab6-1170300817>

# 实验过程

运行请选择monkeygame包下的Monkeygame.java文件



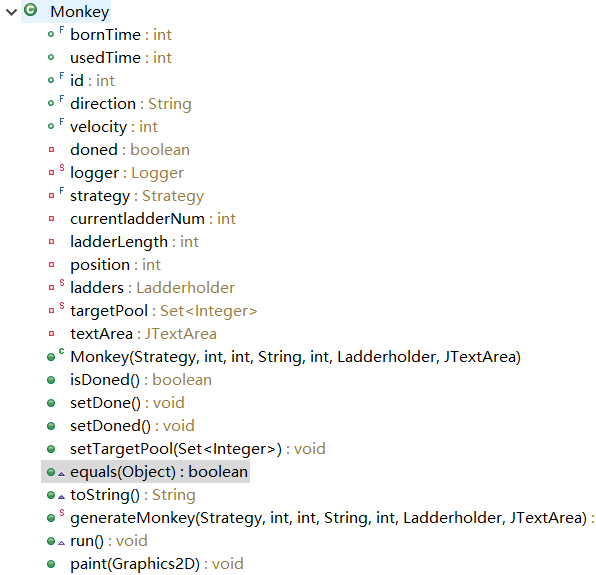
## ADT设计方案

设计了哪些ADT、各自的作用、属性、方法；

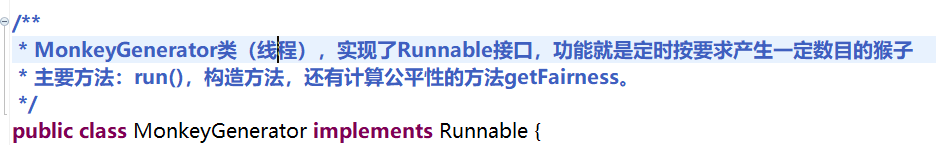
首先是猴子类：



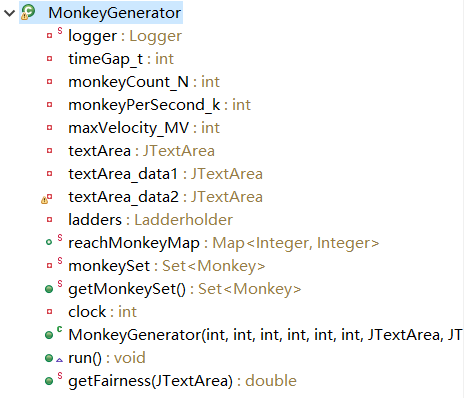
方法和属性：



其次是MonkeyGenerator类：

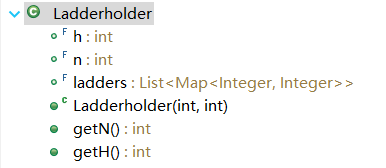


方法和属性：



梯子集合类：



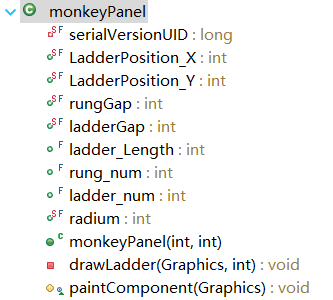


动画画板类：



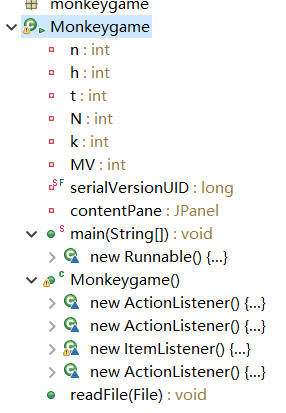
方法和属性：

主要是一些绘图要用的参数



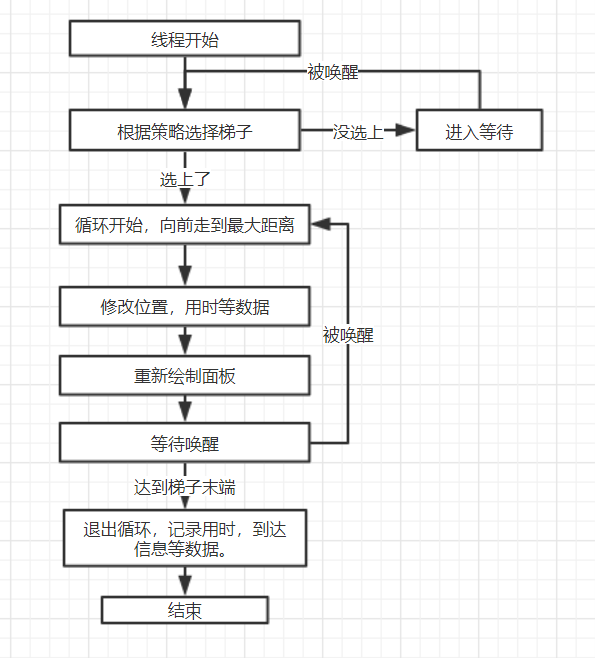
GUI实现：





## Monkey线程的run()的执行流程图

这里无需考虑具体采用的梯子选择策略。



## 至少两种“梯子选择”策略的设计与实现方案

### 策略1 只选择为空的梯子

很简单，如果不是空的就返回0继续等待。

### 策略2 选择空的，没有空的选择与自己同向速度最快的

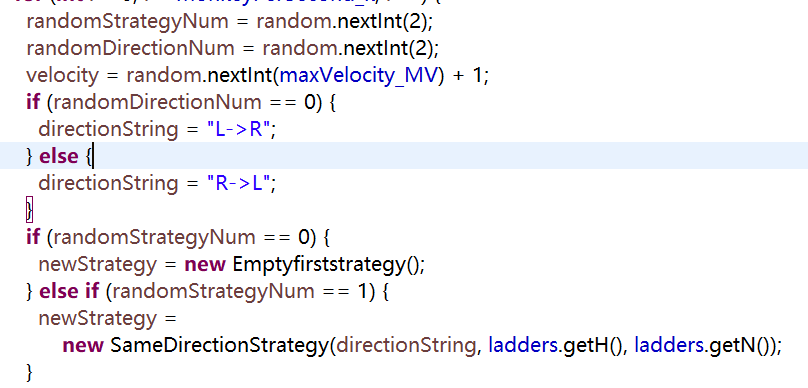
首先观察有没有空的梯子，没有的话开始记录当前梯子上的物体的位置，下一次被唤醒时，再次记录位置，从而就能得到每个梯子上面各个猴子的速度，之后选出最大的速度值就行了。

### 策略3（可选）

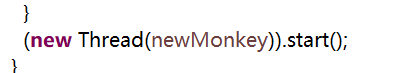
## “猴子生成器”MonkeyGenerator

如何设计和实现。

每隔一段时间生成一定数目的猴子，如果是随机生成，利用Random类生成随机数，决定猴子的策略和方向



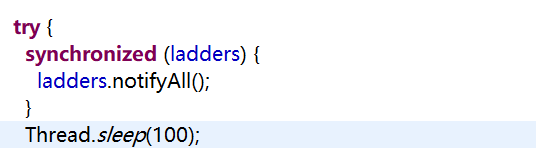
生成每一个猴子之后，启动猴子线程：



之后进入循环：

每隔一定的秒数，可以是1000ms，为了加快运行可以调成100ms之类

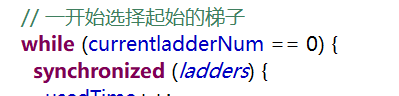
执行一次notifyall：



这样就会唤醒所有在等待的monkey线程，从而达到每秒执行一次行动的目的。

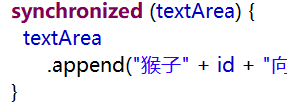
## 如何确保threadsafe？

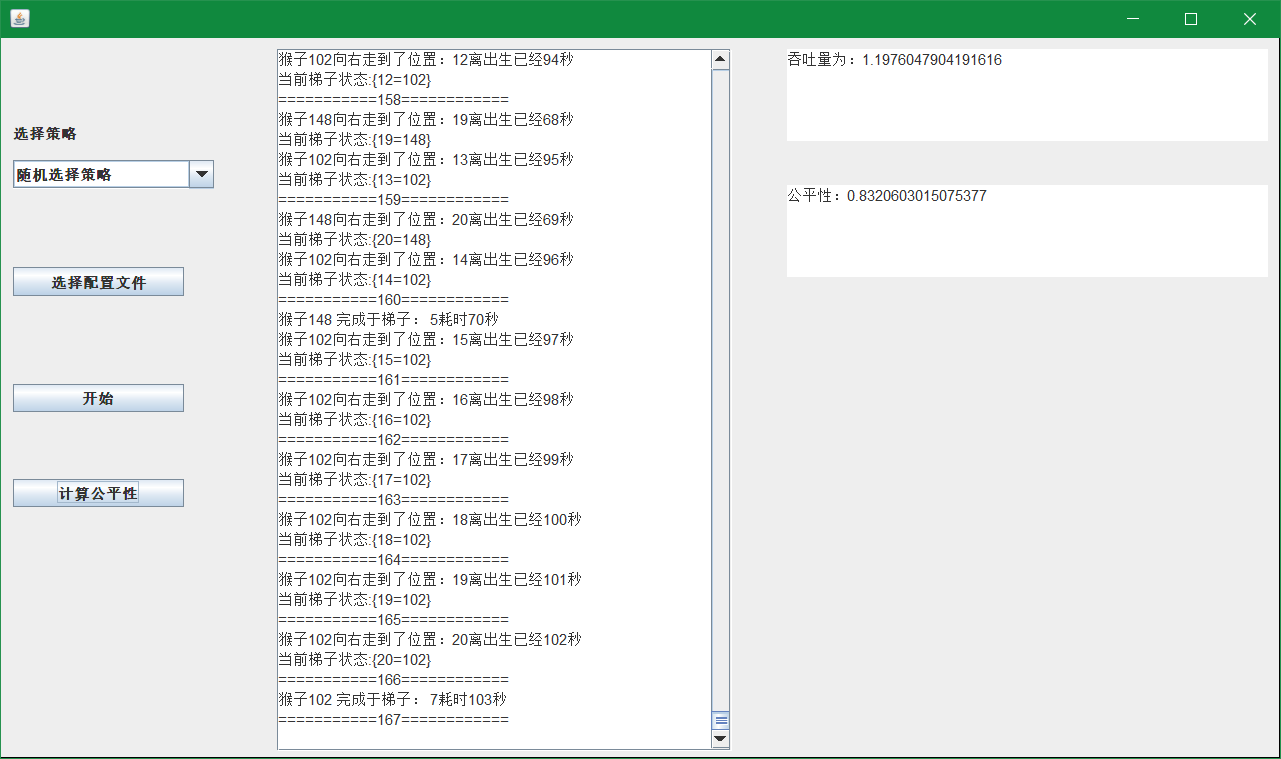
1.每个猴子线程在选择梯子，向前走的时候，总之是要修改记录位置信息的map的时候，都要在synchronized (ladders) {块之中，这样避免猴子在决策或者是移动的时候，梯子上的猴子信息又改变了，造成错误。



2.在向通知面板输出通知的时候，由于是每个线程单独执行写入操作

JTextArea.append，不是线程安全的，也需要保证线程安全，否则信息会出现错乱

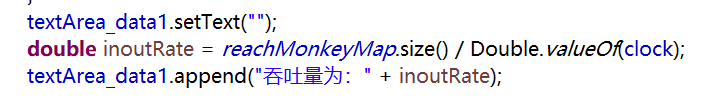




## 系统吞吐率和公平性的度量方案

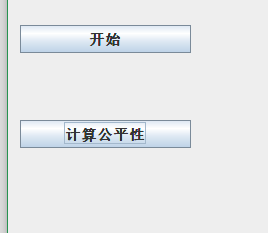
1. 吞吐率：

吞吐率的计算是随时刷新的，直接把记录到达猴子的map的大小除以时间



1. 公平性

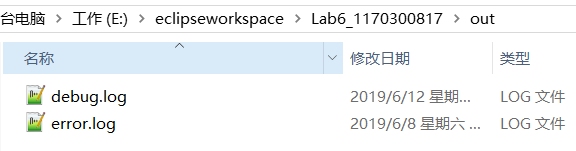
公平性的计算可以在执行结束后，点击计算公平性按钮进行，主要思路就是利用monkey对象中的两个属性：bornTime和usedTime，bornTime+usedTime就是结束时间，以此计算。



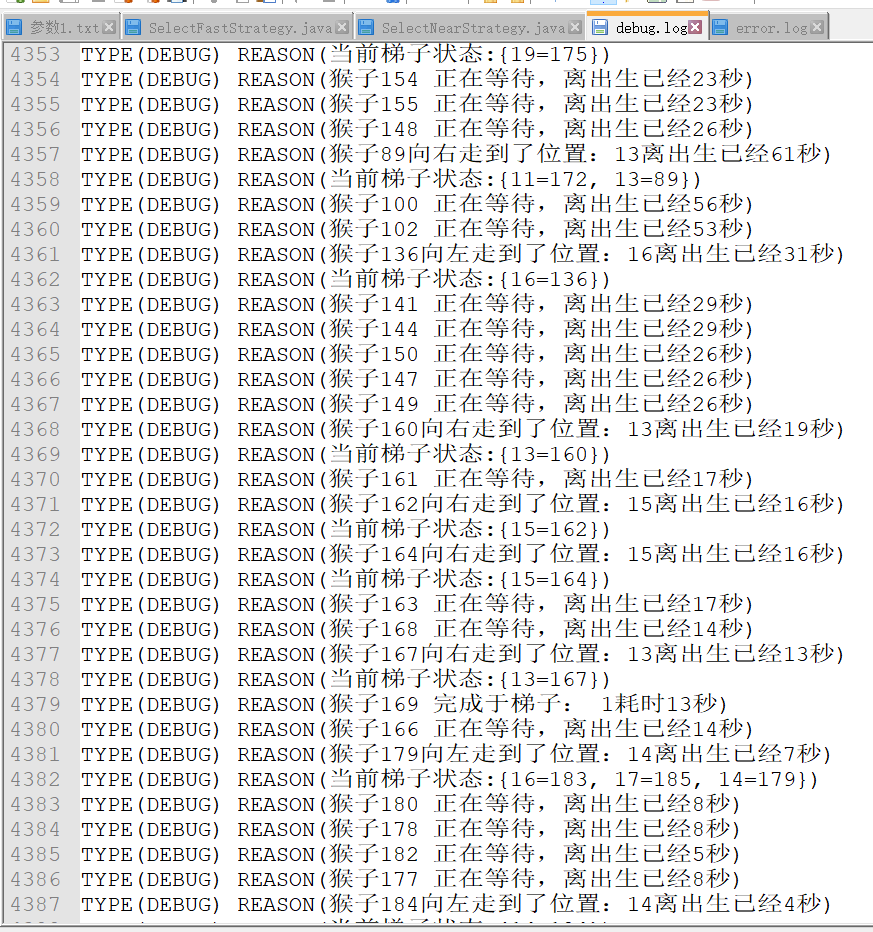
## 输出方案设计

日志：

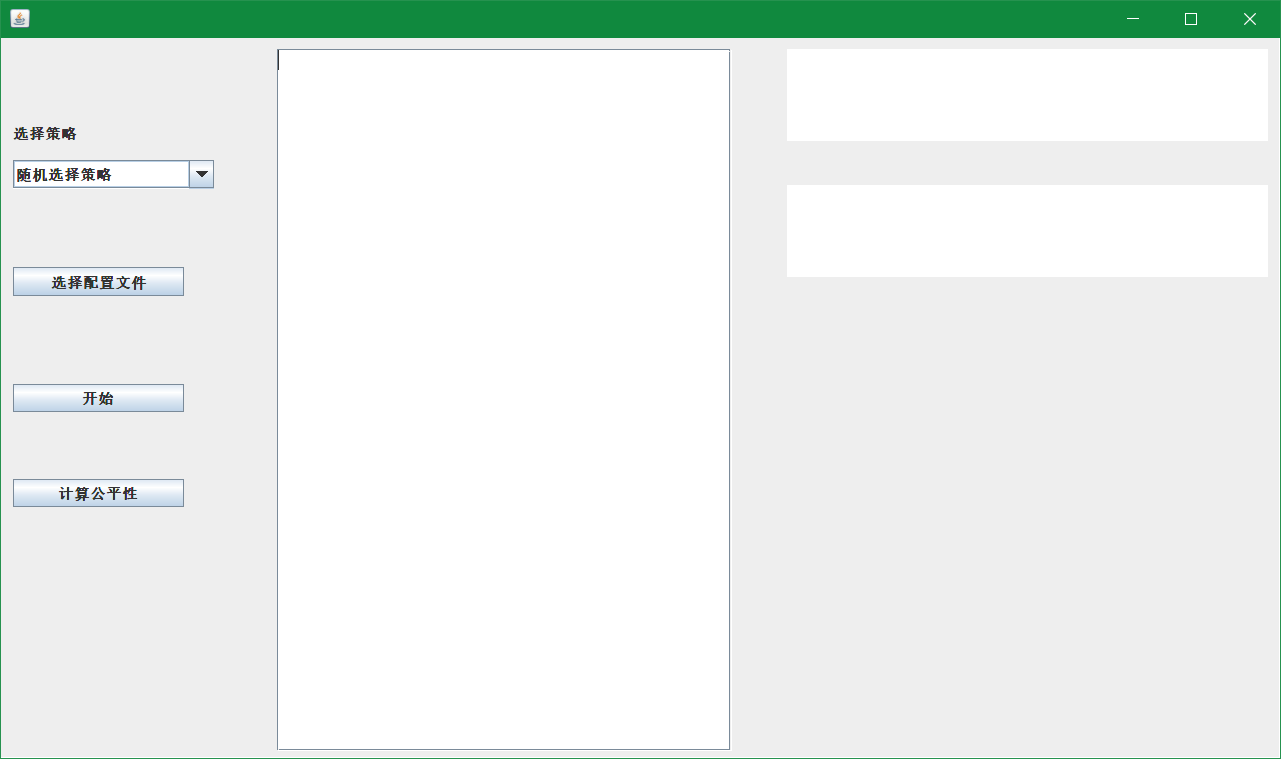
日志输出在Lab6\_1170300817\out下的log文件里



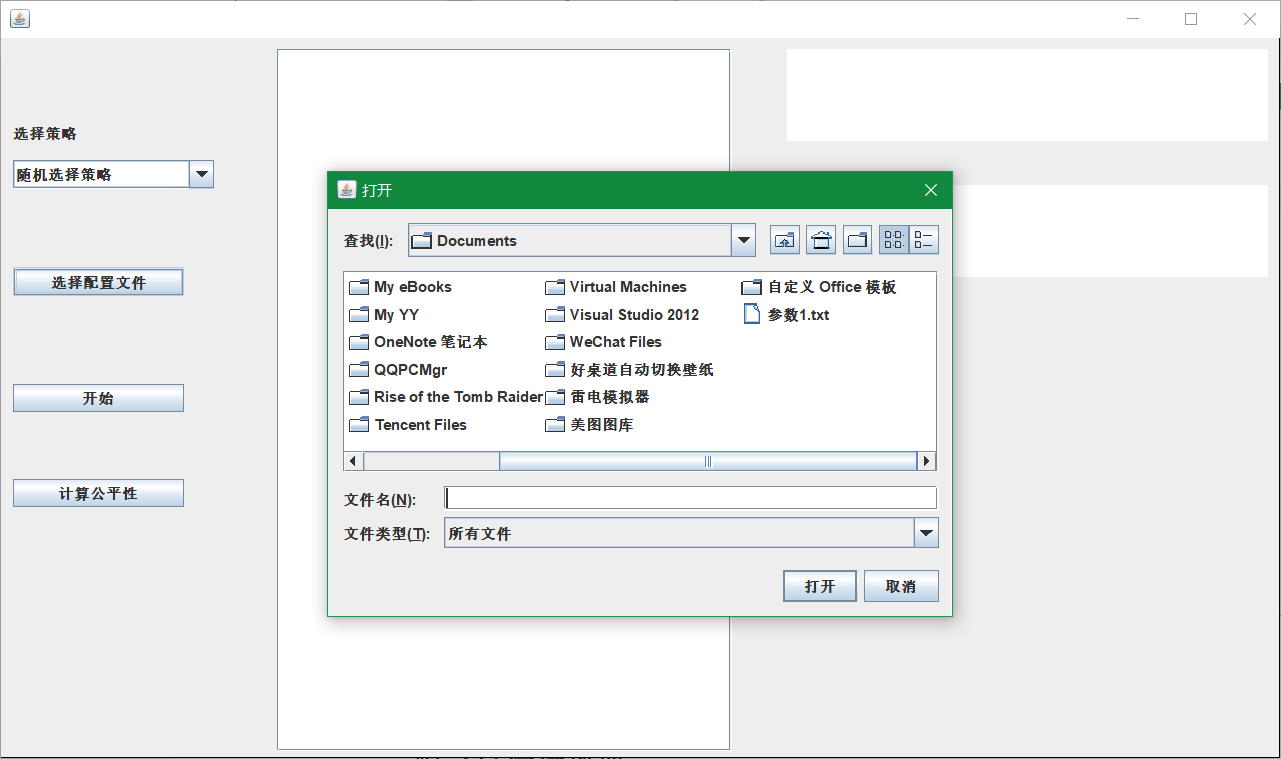
如：



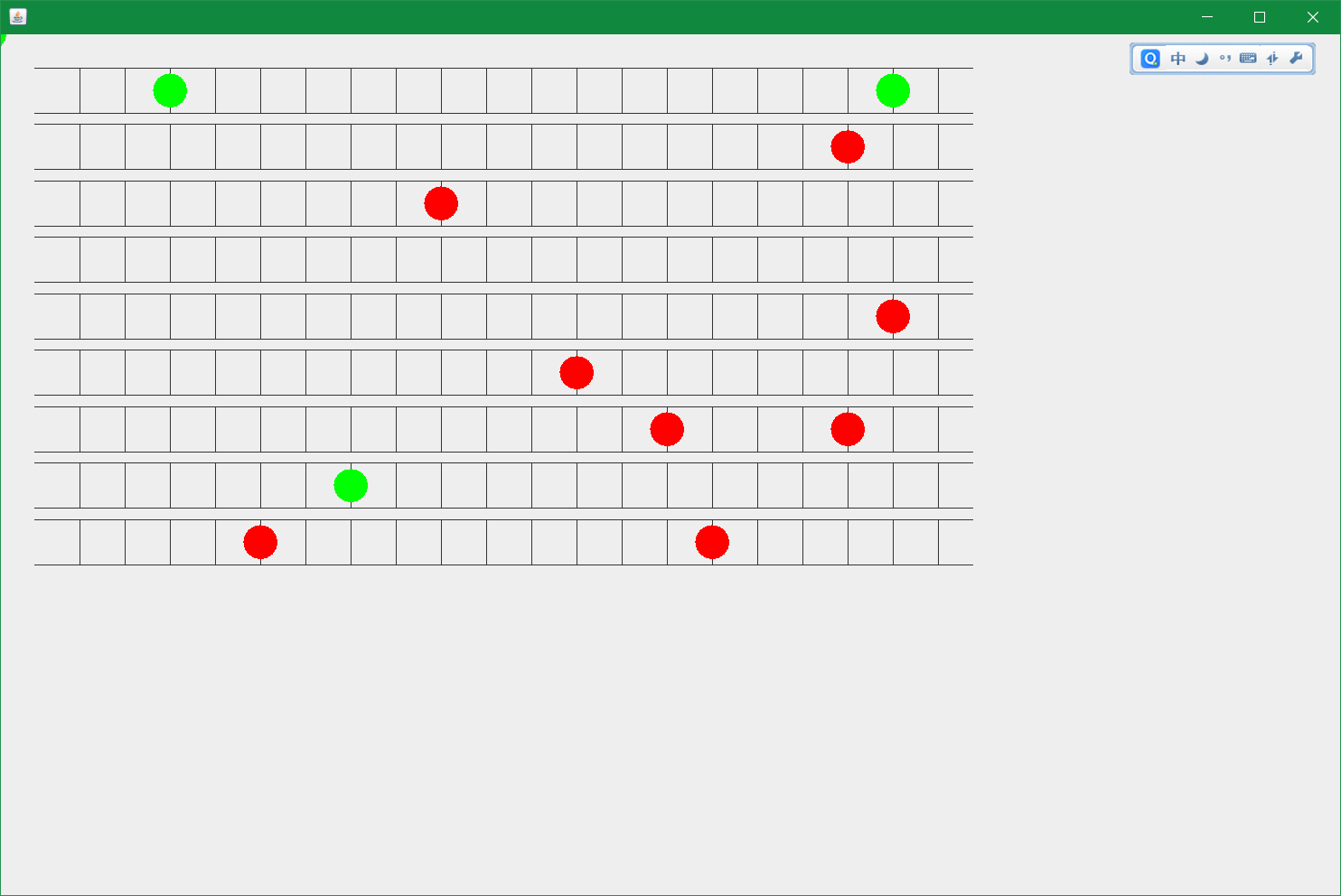
GUI：界面：



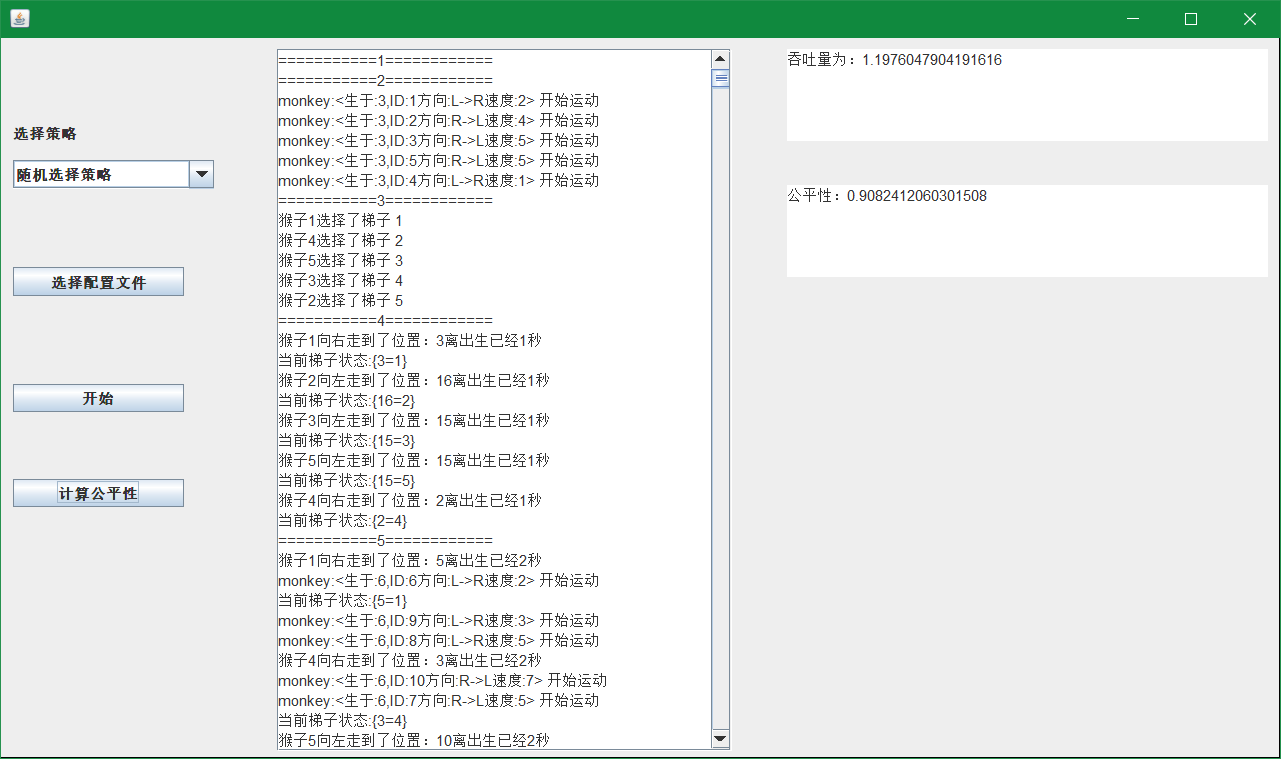
可以选择配置文件：



选择之后点击开始，就会跳出动画界面：



同时界面上也会输出日志：

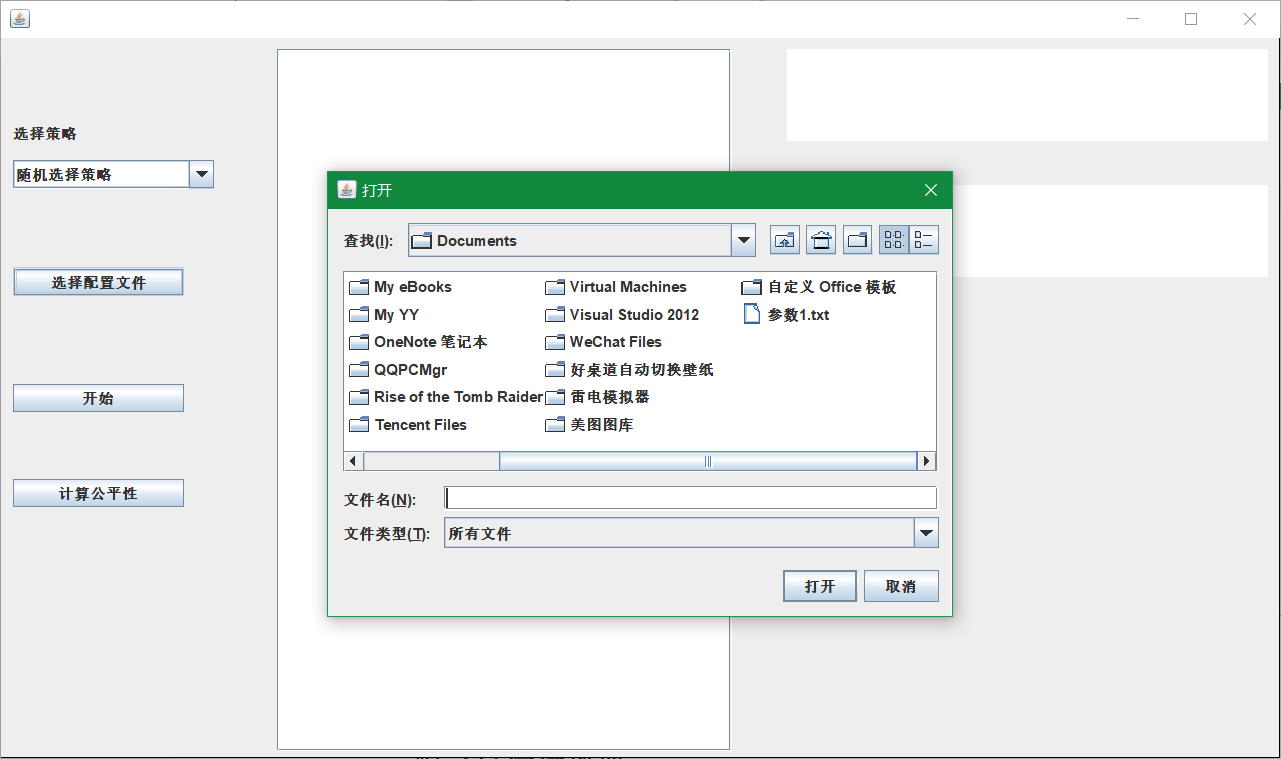


最后点击计算公平性，输出公平性数据

## 猴子过河模拟器v1

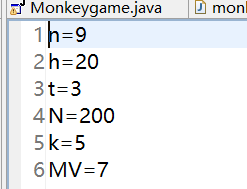
### 参数如何初始化

利用JFileChooser实现：

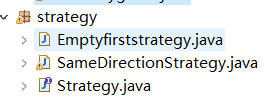


选择文件之后，用正则表达式解析

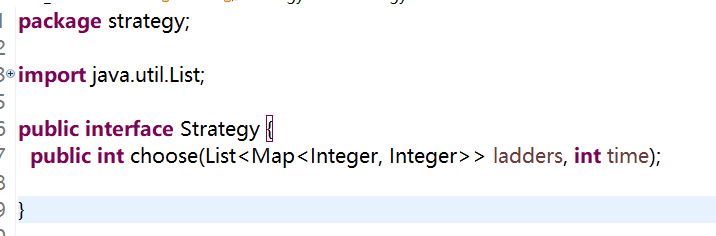
参数文件示例：



### 使用Strategy模式为每只猴子选择决策策略



类似之前的读写策略，也是先设计一个Strategy接口，



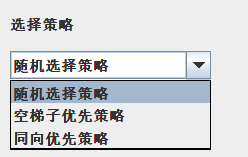
之后分别实现每个策略的这个choose方法。

## 猴子过河模拟器v2

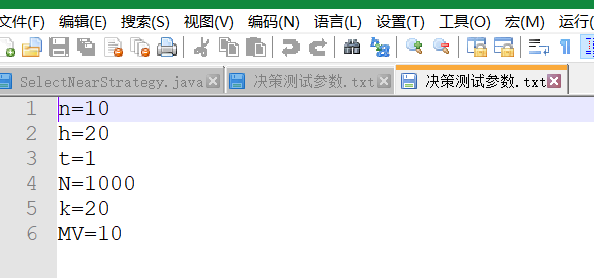
在不同参数设置和不同“梯子选择”模式下的“吞吐率”和“公平性”实验结果及其对比分析。

### 对比分析：固定其他参数，选择不同的决策策略

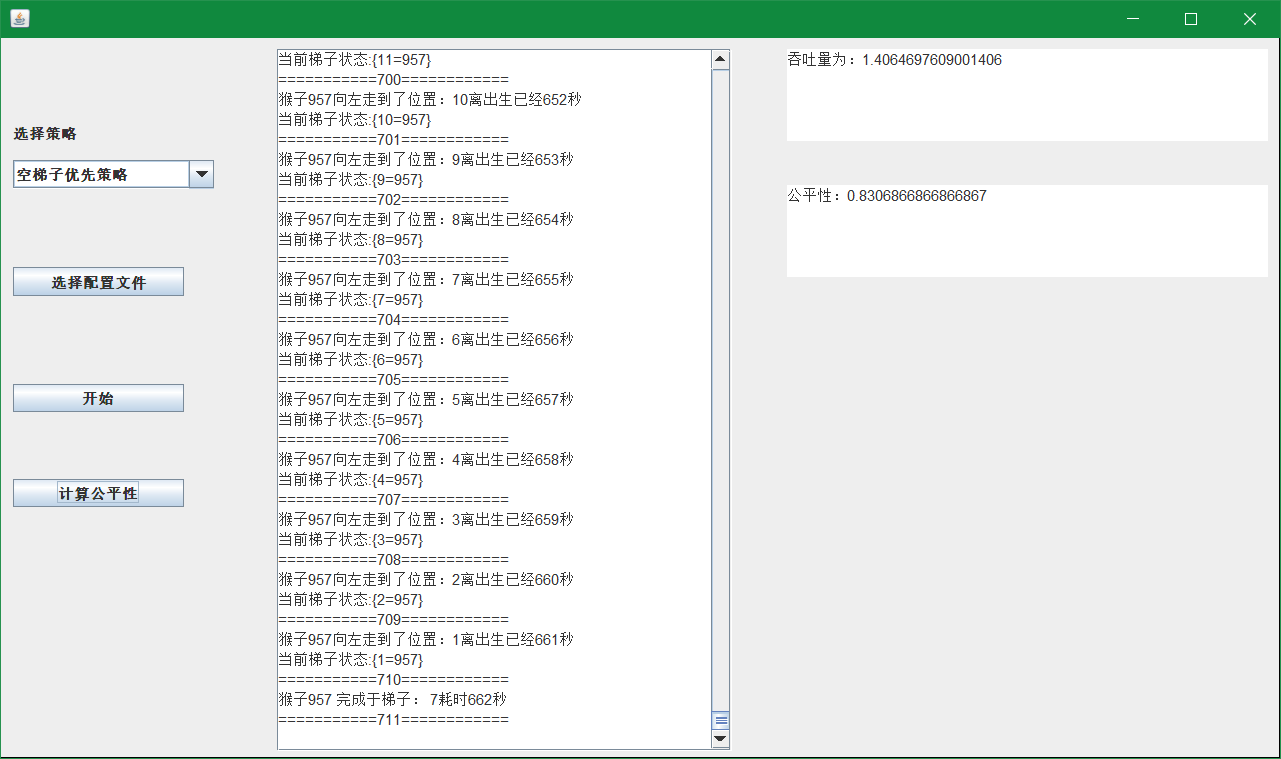
在版本2中，可以选择固定的策略。



先固定下参数为：

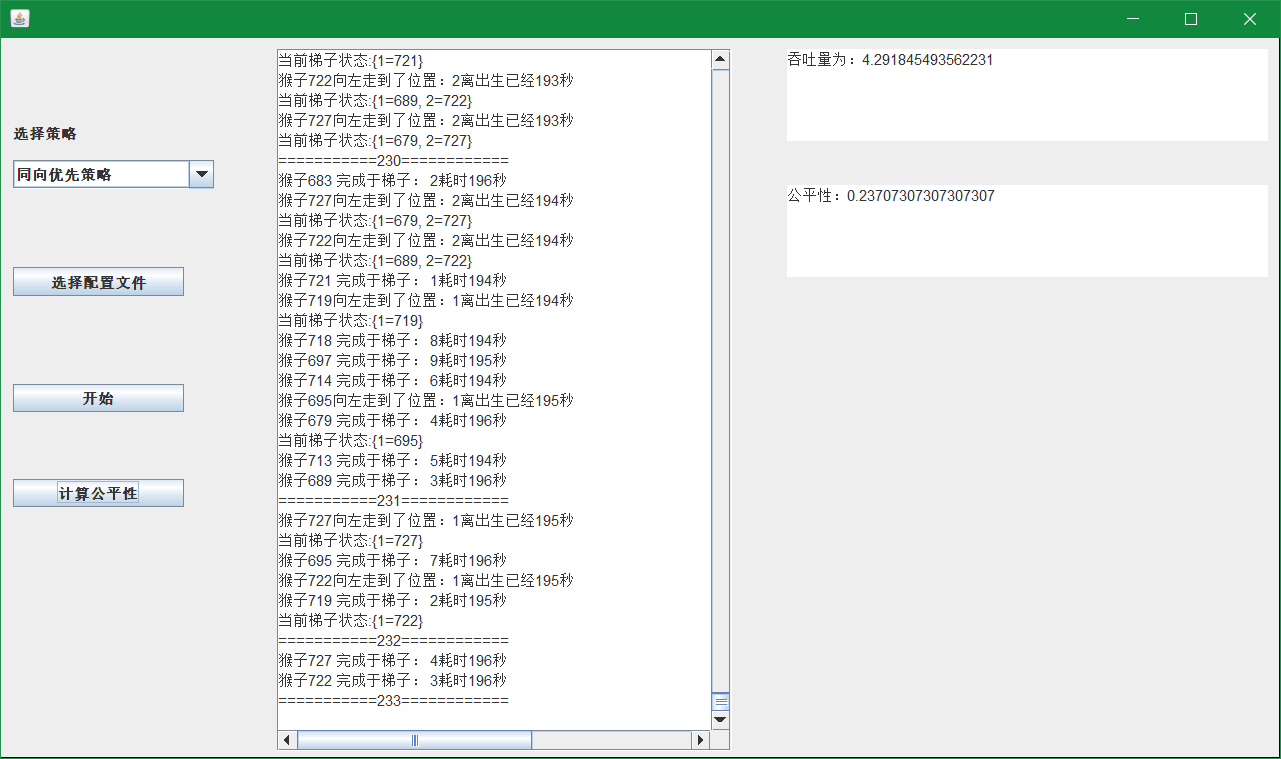


空梯子优先策略耗时：



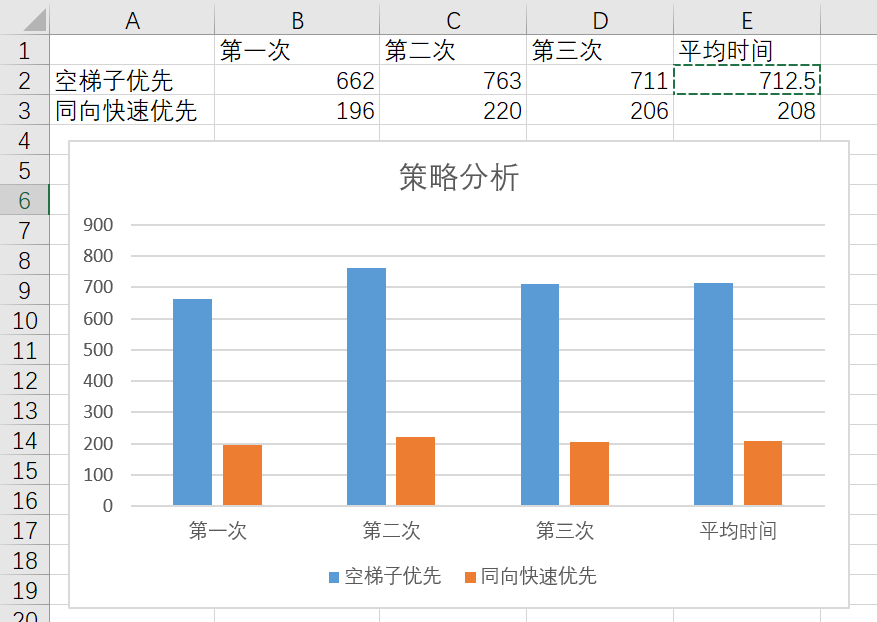
一共是662秒

同向优先策略耗时：



一共耗时196秒

反复试验三次：

、

可见，由于第一个选择空的策略过于简单，第二个策略选择复杂得多，虽然需要额外的观察时间，但是选择同向快速的梯子仅从原理上就可知道会碾压第一种策略。

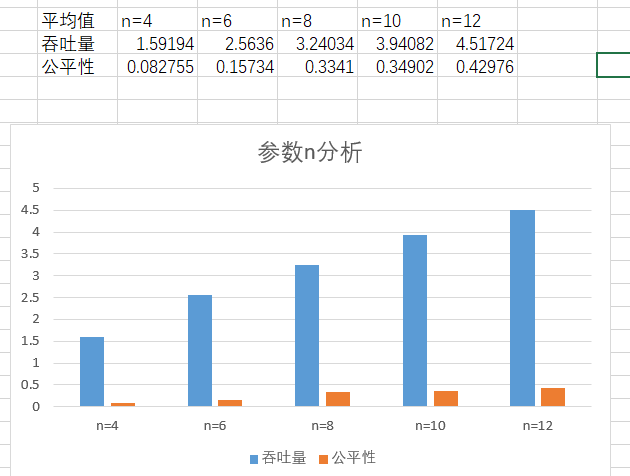
### 对比分析：变化某个参数，固定其他参数

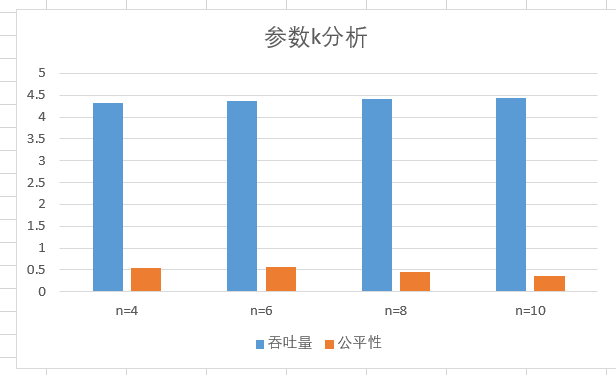
选择第二个策略分析：

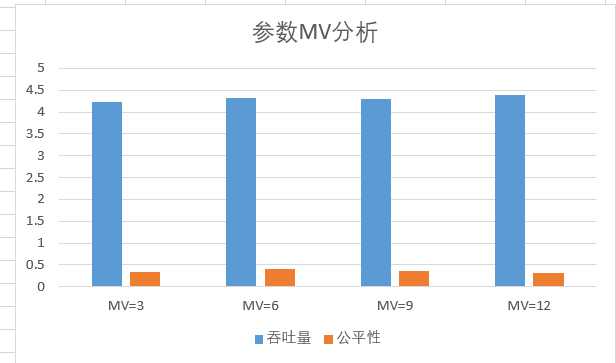
分别改变不同参数：

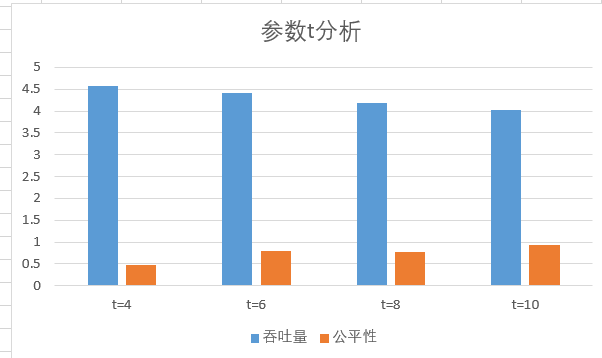
为了加速测试，降低了一些猴子数目

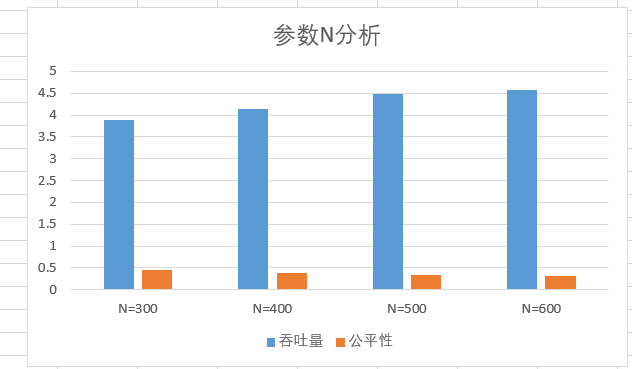




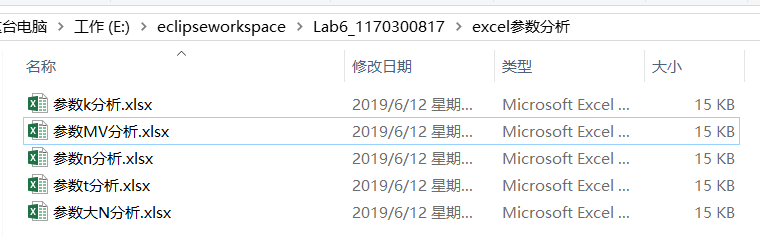








详细实验信息参见：



总体上看：参数n与吞吐率，公平性都是正相关。

参数t增大时，吞吐量下降，公平性上升

参数N增大时，吞吐量上升，公平性下降

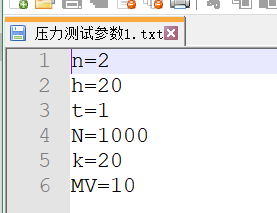
参数MV和k增大时，吞吐量公平性看不出明显变化

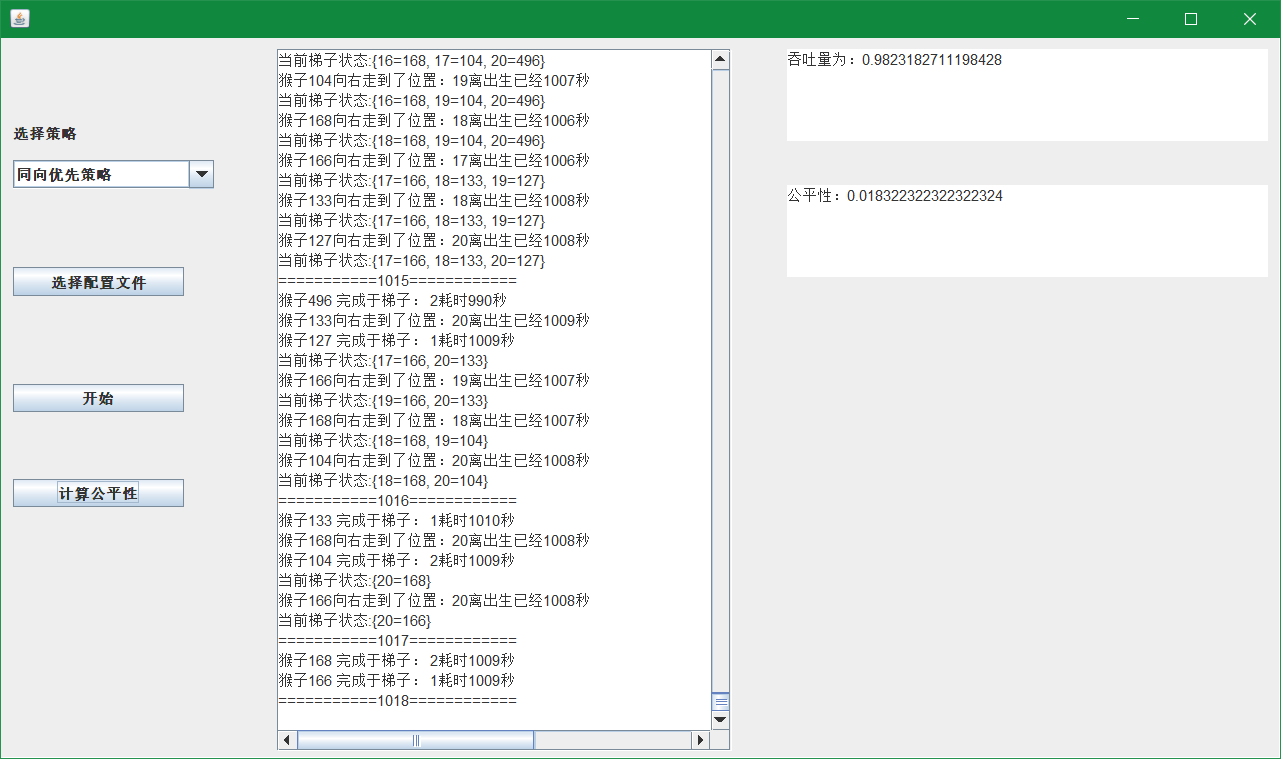
### 分析：吞吐率是否与各参数/决策策略有相关性？

有的，分析结果如上，从理论上也不难推测，更优的策略会带来更多的并行，会有更好的吞吐率。对于梯子数n，也不难推测，梯子更多，过河自然更快，而间隔时间越长，出现可以安排猴子却没有猴子生成的情况也越多，吞吐率下降也不难理解。

### 压力测试结果与分析

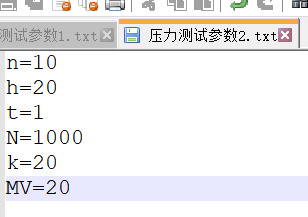
设计压力测试参数1：

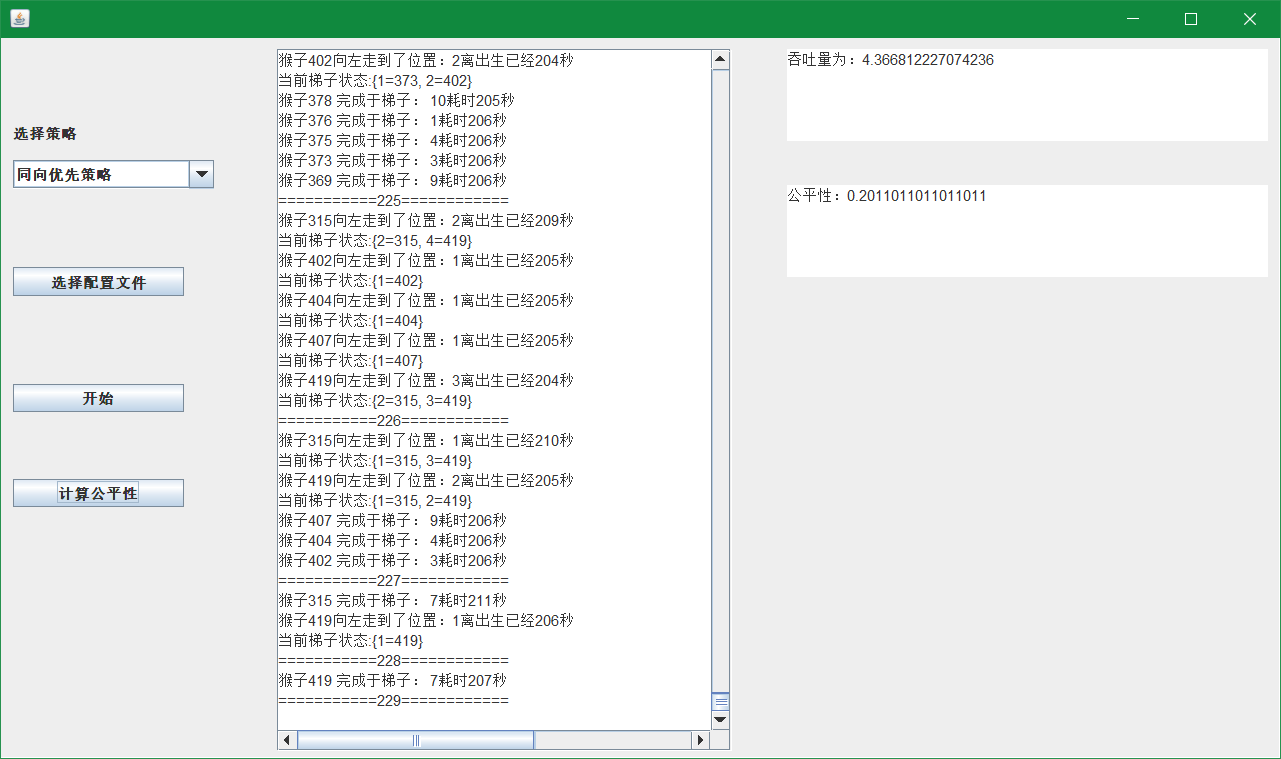




不难发现，在梯子数目很少的情况下，过河是非常慢的，而且由于并行策略，如果碰巧头两个上梯子的猴子都是从左往右，之后的也都是同向的猴子了，直到所有要去右边的猴子全部到达，去左边的才会上梯子。

设计压力测试参数2：

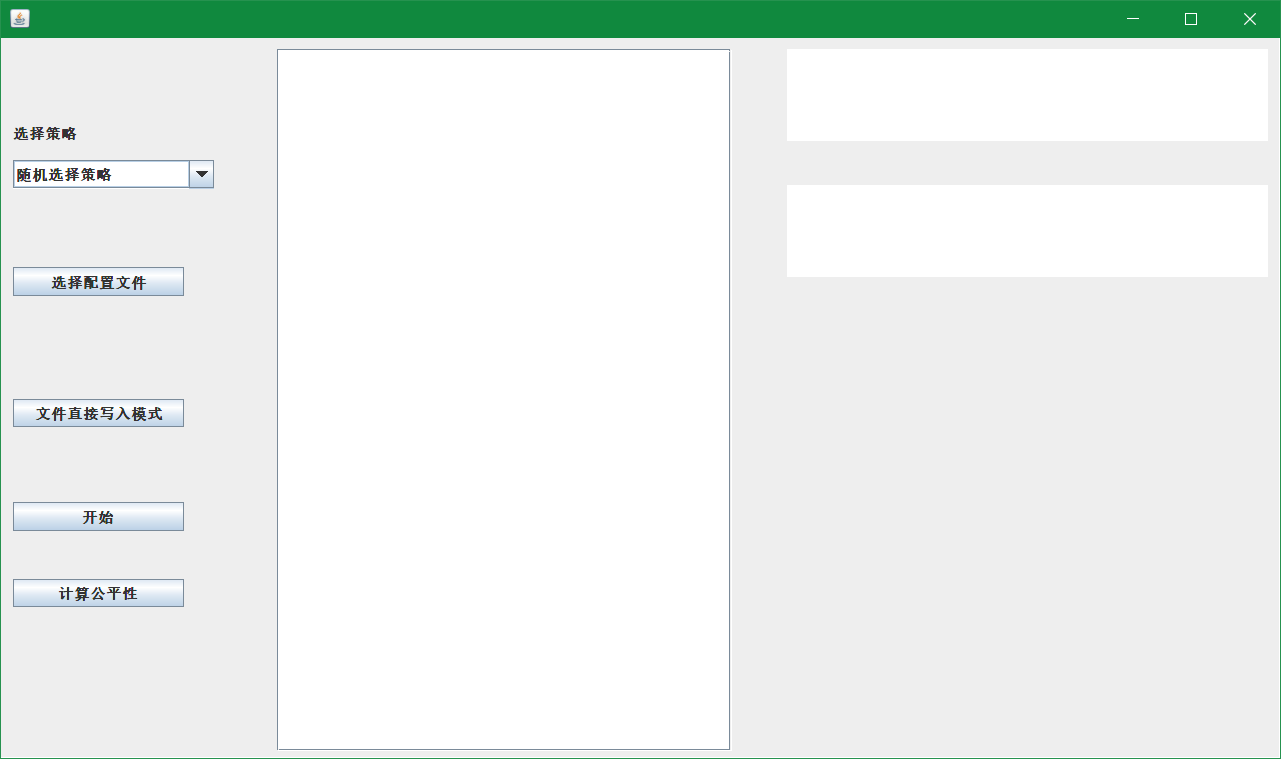
速度范围为0-20



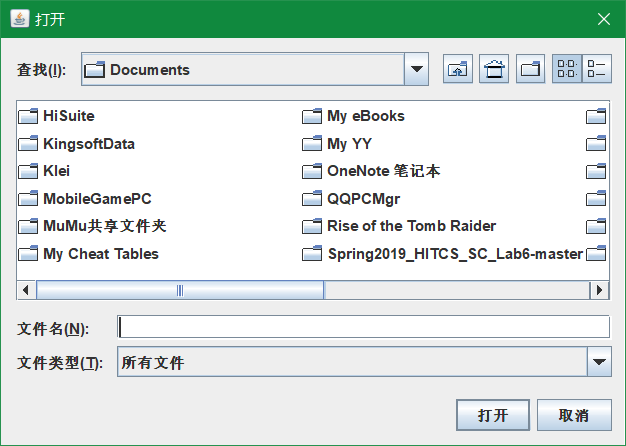
通过观察动画可以发现，速度范围很大时，常常会出现快的猴子被慢的猴子挡住的情况，（结合之前的分析也可以得出同样的结果）所以吞吐率的提升不是特别明显

## 猴子过河模拟器v3

介绍一下使用方法：



在界面上选择“文件直接写入模式”，之后选择文件



之后会直接开始”。

针对教师提供的三个文本文件，分别进行多次模拟，记录模拟结果。



# 实验进度记录

请使用表格方式记录你的进度情况，以超过半小时的连续编程时间为一行。

每次结束编程时，请向该表格中增加一行。不要事后胡乱填写。

不要嫌烦，该表格可帮助你汇总你在每个任务上付出的时间和精力，发现自己不擅长的任务，后续有意识的弥补。

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 计划任务 | 实际完成情况 |
| 6.8 | 10:00-22:00 | V1 | 未完成 |
| 6.9 | 10:00-22:00 | V1 | 未完成 |
| 6.10 | 19:00-22:00 | V1 | 未完成 |
| 6.12 | 19:00-22:00 | V2 | 完成 |
| 6.13 | 10:00-22:00 | V3 | 完成 |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的难点 | 解决途径 |
| 第一次写gui不会 | 在网上学习了Windowsbuilder的简单使用 |
|  |  |
|  |  |

# 实验过程中收获的经验、教训、感想

## 实验过程中收获的经验和教训

应该先把ppt看一遍再下手做，避免不必要的知识性错误。

## 针对以下方面的感受

1. 多线程程序比单线程程序复杂在哪里？你是否能体验到多线程程序在性能方面的改善？

复杂在有的bug不知道是什么地方出错，能。

1. 你采用了什么设计决策来保证threadsafe？如何做到在threadsafe和性能之间很好的折中？

加锁，需要的部分加锁，尽可能少。

1. 你在完成本实验过程中是否遇到过线程不安全的情况？你是如何改进的？

使用线程安全的数据结构或使用锁。

1. 关于本实验的工作量、难度、deadline。

都适度。

1. 到此为止你对《软件构造》课程的意见和建议。

获益匪浅。