**少数派问题博弈模拟器**

**设计规格说明书（面向对象方法）**

**小组成员：熊倩 黄力**

**2018年6月**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文档修改情况记录 | | | | |
| 版本 | 修改状态 | 修改日期 | 修改摘要 | 修改人 |
| 1.0 | 1,2,3,4,5 | 2018-6-3 | 创建 | 熊倩 |
| 1.1 | 7 | 2018-6-20 | 创建附录 | 熊倩 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

[**1.** **引言** 4](#_Toc193080911)

[1.1 项目名称 4](#_Toc193080912)

[1.2 编写目的 4](#_Toc193080913)

[1.3 项目背景 4](#_Toc193080914)

[1.4 名词解释 4](#_Toc193080915)

[1.5 参考资料 5](#_Toc193080916)

[2. 任务概述 5](#_Toc193080917)

[2.1 项目总体目标 5](#_Toc193080918)

[2.2 运行环境 5](#_Toc193080919)

[2.3 开发环境 5](#_Toc193080921)

[3. 总体结构设计 6](#_Toc193080917)

[3.1 系统架构 6](#_Toc193080918)

[3.2 架构描述 6](#_Toc193080919)

[**4.** **对象类设计** 7](#_Toc193080930)

[4.1 Strgy类 7](#_Toc193080931)

[4.1.1 类描述 7](#_Toc193080932)

[4.1.2 类成员 7](#_Toc193080933)

[4.1.3类成员函数 7](#_Toc193080934)

[4.2 Agent类 7](#_Toc193080931)

[4.2.1 类描述 7](#_Toc193080932)

[4.2.2 类成员 7](#_Toc193080933)

[4.2.3类成员函数 7](#_Toc193080934)

[4.3 World类 7](#_Toc193080931)

[4.3.1 类描述 8](#_Toc193080932)

[4.3.2 类成员 8](#_Toc193080933)

[4.3.3类成员函数 8](#_Toc193080934)

[4.4 UML类图 8](#_Toc193080931)

[5. 模块功能设计 9](#_Toc193080930)

[5.1 用户不参与博弈模块 9](#_Toc193080931)

[5.1.1 模块功能描述 9](#_Toc193080932)

[5.1.2 参数说明 9](#_Toc193080933)

[5.1.3 用户界面 9](#_Toc193080935)

[5.1.4 流程逻辑 9](#_Toc193080936)

[5.2 用户参与博弈模块 9](#_Toc193080931)

[5.2.1 模块功能描述 9](#_Toc193080932)

[5.2.2 参数说明 10](#_Toc193080933)

[5.2.3 用户界面 10](#_Toc193080935)

[5.2.4 流程逻辑 10](#_Toc193080936)

[5.3 用户查看博弈状态模块 10](#_Toc193080931)

[5.3.1 模块功能描述 10](#_Toc193080932)

[5.3.2 参数说明 10](#_Toc193080933)

[5.3.3 用户界面 10](#_Toc193080935)

[5.3.4 流程逻辑 11](#_Toc193080936)

[**6.** **需要改进的地方** 11](#_Toc193080926)

[**7.** 附录 11](#_Toc193080917)

[**7**.1 界面图 11](#_Toc193080918)

[**7.2** 活动图 12](#_Toc193080918)

[**7.5** **用况图** 13](#_Toc193080919)

[**7.5** **状态机图** 13](#_Toc193080919)

[**7.5** **顺序图** 14](#_Toc193080919)

**1 引言**

**1.1 项目名称**

本项目的名称为《少数派问题博弈模拟器》。

**1.2 编写目的**

本设计说明书是开发少数派问题博弈模拟器的主要依据，描述程序系统结构，并进行程序设计，为开发人员提供工作基准文件，并对后续阶段的工作起指导作用。

预期读者为软件开发人员（课程项目小组成员：熊倩、黄力）、软件评审人员（课程教师、助教）以及其他相关人员。

**1.3 项目背景**

本项目是课程《软件工程》的期末PJ，开发者为熊倩和黄力，用户为课程老师及其他将使用本软件的用户，该软件的运行环境为Windows系统。

**1.4 名词解释**

**少数派博弈问题：**

少数派博弈问题是1997年由弗赖堡大学的张翼成教授与他的学生Damien Challet共同提出的一种动态博弈问题。与El Farol酒吧问题类似，少数派博弈问题是归纳推理和有界理性的一个例子。在该问题中，参与者们在每轮博弈的过程中都有两种选择，且他们在做决策时无法获知其他参与者本轮做出的决策，而只能基于对有限的过往博弈结果做归纳、分析和推理做出选择。每一轮将参与者按照他们所做的选择分为两方，人数较少的一方获得本轮博弈的胜利。同时本轮博弈结果也将参与到“反馈循环”中影响之后有限轮数的博弈。在一定轮数的博弈之后，参与者们能够每一轮做的决策能够趋于稳定状态。在少数派博弈模拟器中，归纳、分析和推理使用具体的策略集来代替。

**博弈模拟器：**

博弈模拟器是一个计算机软件，其能够模拟少数派博弈的具体运行过程。

**环境：**

博弈模拟器有一个博弈环境，该环境由固定个博弈者及相关全局参数组成。

**博弈者：**

博弈者是环境中进行博弈的实体。

**策略：**

每个博弈者有固定个数个策略，每个策略根据不同的情况给出是否去酒吧的建议，并且有一个分值代表其预测能力的好坏。

**迭代步数：**

环境中所有博弈者进行博弈的次数。

**博弈者数量：**

环境中博弈者的个数。

**记忆长度：**

每个博弈者所能记得的博弈历史结果的长度。

**策略个数：**

每个博弈者所拥有的策略的个数。

**1.5 参考资料**

[1]软件工程 钱乐秋 赵文耘 牛军钰编著 清华大学出版社 2016年9月

**2 任务概述**

**2.1 项目总体目标**

本项目将模拟少数派问题的博弈过程，并能让用户参与博弈过程，查看博弈状态以及博弈的可视化结果。

**2.2 运行环境**

运行环境：Windows系统。

**2.3 开发环境**

开发环境：Windows系统，PyCharm IDE，使用python 3.6语言。

**3 总体结构设计**

**3.1 系统架构**

如下图所示：



**3.2 架构描述**

软件由两大部分组成：对象类和软件界面。

对象类有三个类：

1. Strgy类，即策略类。
2. Agent类，即博弈者类。
3. World类，即环境类。

软件界面有三个模块功能：

1.博弈模拟器的用户不参与博弈，用户指定博弈模拟器的迭代步数、博弈者数量、记忆长度和策略个数，然后博弈模拟器会根据参数迭代至指定步数，并输出可视化结果，结果为png图片。

2.博弈模拟器的用户参与博弈，用户指定博弈模拟器的博弈者数量（该数量包含用户扮演的博弈者）、记忆长度和策略个数，然后用户可以进行任意次迭代或者选择终止博弈。如果用户进行博弈迭代，用户需要指定其扮演的博弈者的行为，待在家里或者去酒吧；如果用户终止博弈，博弈模拟器会根据博弈的历史记录，输出可视化结果，结果为png图片。

3.用户查看博弈状态，博弈状态共有三种：i号博弈者的信息，即其拥有的策略；用户所扮演的博弈者的信息博弈；全局信息，即每轮迭代的人员差分。模拟器会将博弈状态输出到显示框中。

软件界面的模块功能通过使用对象类的成员函数来实现。

**4 对象类设计**

**4.1 Strgy类**

**4.1.1 类描述**

Strgy类，即策略类，包含策略所代表的数据结构，行使策略和更新策略的成员函数。

**4.1.2 类成员**

Strgy类有三个成员：

score：表示本策略已获得的得分。

rules：表示本策略将如何选择是否去酒吧。每种博弈历史记录有其对应的数值，策略的位数和情况的总数相同，每位是0或1。本策略将根据当前情况的数值所对应的位选择是否去酒吧。

count：表示本策略被使用的次数。

**4.1.3 类成员函数**

Strgy类有两个成员函数：

act(self, state)：策略的行使将根据情况state及策略自身给出是否去酒吧的建议，策略的使用次数加一。

update(self, v)：给策略的得分score加上分值v对策略进行更新。

**4.2 Agent类**

**4.2.1 类描述**

Agent类，即博弈者类，包含博弈者所拥有的策略以及当前的行为，根据策略作出决定和更新自己所有策略的成员函数。

**4.2.2 类成员**

Strgies：本博弈者的所有策略。

action：本博弈者的当前行为，+1表示去酒吧，-1表示待在家里。

**4.2.3 类成员函数**

act(self, state)：本博弈者根据自己拥有的所有策略来作出决定，是否去酒吧。 本博弈者将选用得分最高的策略，若有多个策略得分相同且最高，则随机选择策略。

update(self, w)：本博弈者根据当前获胜行为来更新自己的所有策略，若策略与获胜行为一致则得分加一，反之则减一。

**4.3 World类**

**4.3.1 类描述**

World类，即环境类，包含博弈模拟器的各项参数，全局信息和全部博弈者。

**4.3.2 类成员**

T：博弈模拟器一次运行的迭代步数。

N：博弈者数量。

m：记忆长度。

s：博弈者拥有的策略个数。

Agents：博弈者全体。

W：博弈的获胜行为的历史记录。

A ：博弈的人员差分的历史记录，人员差分即一次博弈中博弈者全体行为的数值之和。

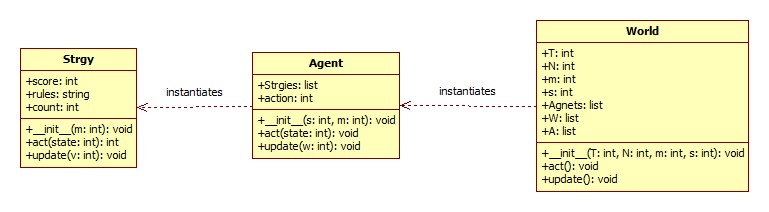
**4.3.3 类成员函数**

act(self)：博弈模拟器迭代T步，每次迭代中，所有博弈者根据获胜行为的历史记录作出决定，并产生此轮获胜行为，然后模拟器的环境进行更新，所有博弈者自身进行更新。

update(self)：模拟器的环境进行更新：更新获胜行为的历史记录和人员差分的历史记录。

**4.4 UML类图**

UML类图如下所示：



**5 模块功能设计**

**5.1 用户不参与博弈模块**

**5.1.1 模块功能描述**

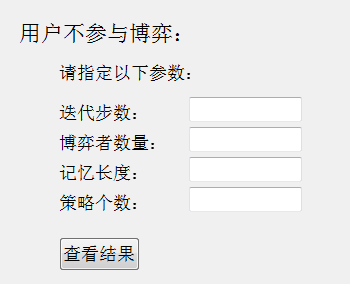
在用户不参与博弈模块中，用户指定博弈模拟器的迭代步数、博弈者数量、记忆长度和策略个数，然后博弈模拟器会根据参数迭代至指定步数，并输出可视化结果，结果为png图片。

**5.1.2 参数说明**

可输入的参数为迭代步数T、博弈者数量N、记忆长度m和策略个数s。行为参数有：“查看结果”。

**5.1.3用户界面**

用户界面如下图所示：



**5.1.4流程逻辑**

在此模块中，用户首先指定博弈模拟器的参数，然后博弈模拟器会根据参数迭代到指定的步数，并通过迭代的历史记录生成可视化结果。

**5.2 用户参与博弈模块**

**5.2.1 模块功能描述**

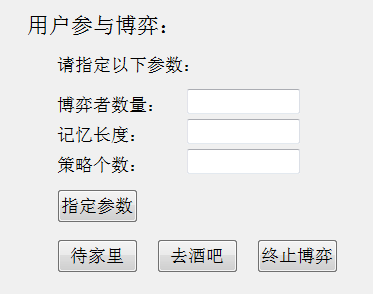
在用户参与博弈模块中，用户首先指定博弈模拟器的博弈者数量（该数量包含用户扮演的博弈者）、记忆长度和策略个数，然后用户可以进行任意次迭代或者选择终止博弈。如果用户进行博弈迭代，用户需要指定其扮演的博弈者的行为，待在家里或者去酒吧；如果用户终止博弈，博弈模拟器会根据博弈的历史记录，输出可视化结果，结果为png图片。

**5.2.2 参数说明**

可输入的参数为博弈者数量N、记忆长度m、策略个数s。行为参数有：“指定参数”，“待家里”，“去酒吧”和“终止博弈”。

**5.2.3 用户界面**

用户界面如下图所示：



**5.2.4流程逻辑**

用户首先指定相关参数，然后手动进行一轮一轮的博弈，当用户选择终止博弈后，博弈模拟器会根据博弈历史输出可视化结果。

**5.3 用户查看博弈状态模块**

**5.3.1 模块功能描述**

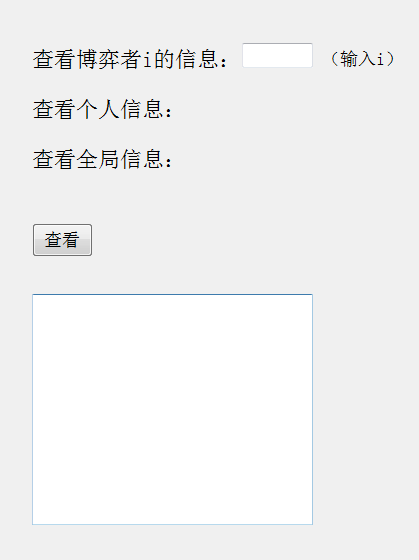
在用户查看博弈状态模块中，用户可以查看三种博弈状态：i号博弈者的信息，即其拥有的策略；用户所扮演的博弈者的信息博弈；全局信息，即每轮迭代的人员差分。模拟器会将博弈状态输出到显示框中。

**5.3.2 参数说明**

可输入的参数为：博弈者号码i。行为参数有：“查看”。

**5.3.3 用户界面**

用户界面如下图所示：



**5.3.4 流程逻辑**

用户首先指定要查看哪种信息，然后博弈模拟器会将指定信息输出到显示框中。

**6 需要改进的地方**

1.用户查看博弈信息时，可以用更生动的方式如图片等代替文本，来显示信息。

2.输出可视化结果时，可以从更多的角度输出博弈的结果和分析。

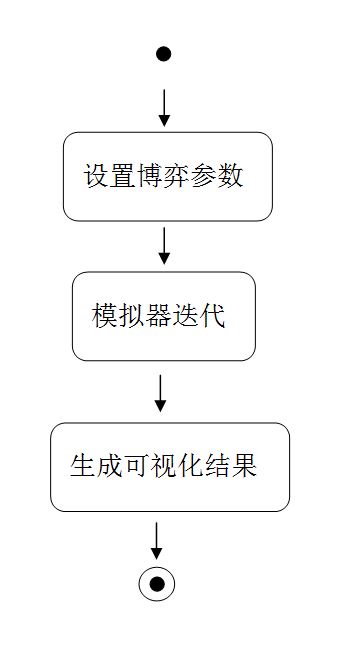
**7 附录**

7.1 [界面图](#_Toc193080918)

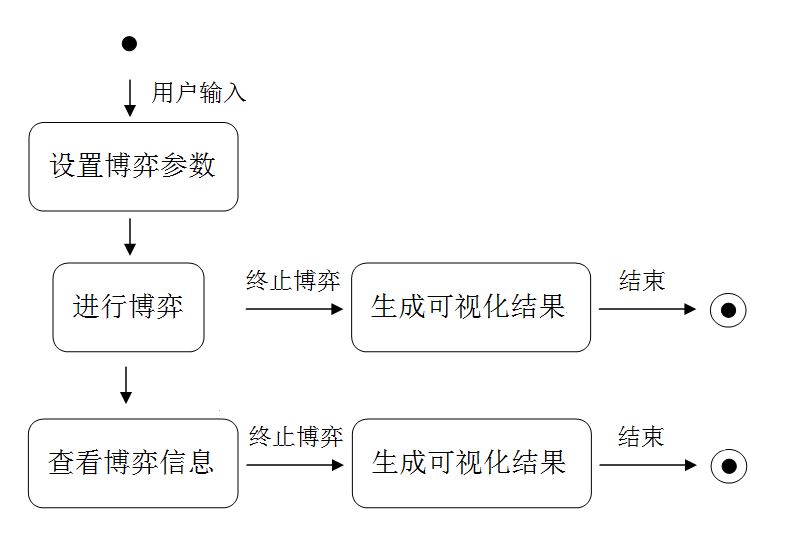


7.1软件整体界面图

7.2[活动图](#_Toc193080918)

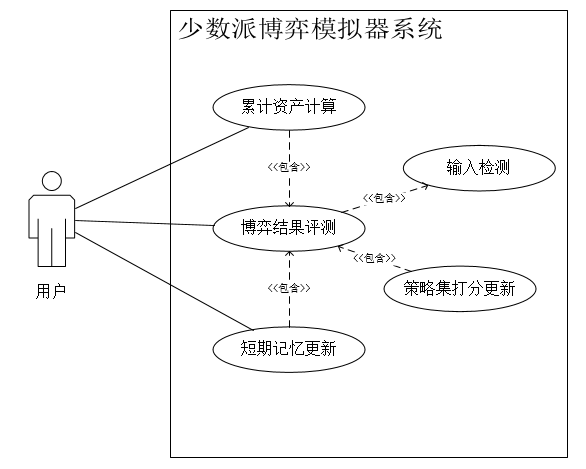


7.2.1 用户不参与博弈的活动图A



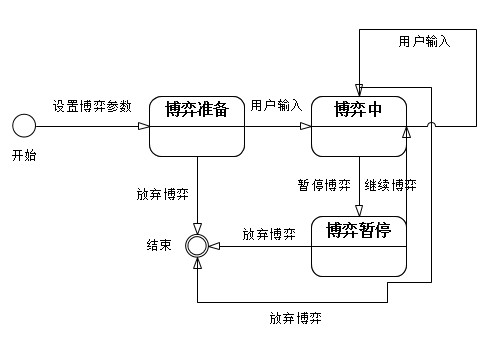
7.2.1 用户参与博弈的活动图B

7.3用况图



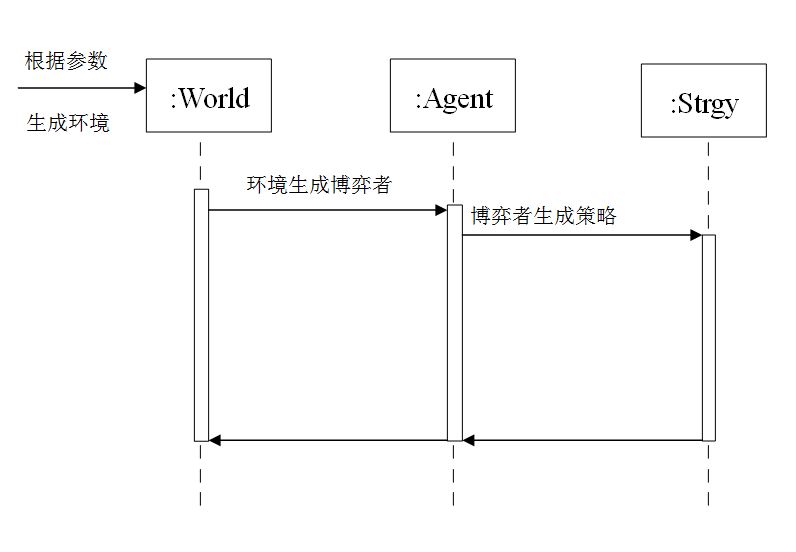
7.3 程序用况图

7.4[状态机图](#_Toc193080918)



7.4 程序状态机图

7.5顺序图



7.5 程序顺序图