

# 高级语言程序设计

## 实验报告

南开大学 信安法

姓名：刘迪乘

学号：2312810

班级：0926

2024年5月13日

## 目录

高级语言程序设计大作业实验报告 .....	2
一. 作业题目 .....	2
二. 开发软件 .....	2
三. 课题要求 .....	2
四. 主要流程 .....	2
1. 整体流程 .....	2
2. 算法或公式 .....	2
(1) <code>player</code> 类设计 .....	2
(2) <code>ui</code> 类设计 .....	2
(3) <code>ZAI</code> 类设计 .....	2
(4) <code>game</code> 类 .....	2
(5) <code>ZAI</code> 决策算法 .....	3
五. 单元测试 .....	3
六. 收获 .....	4
七. 反思 .....	4

## 一、作业题目

Liar Dice 游戏的人机博弈

## 二、开发软件

Visual Studio 2019

Easyx库

## 三、课题要求

- 1) 面相对象
- 2) 单元测试
- 3) 模型部分
- 4) 验证

## 四、主要流程

### 1.整体流程

首先构建场景之间的框架。游戏中首先实现了startUI，来负责游戏的开始页面，并设置有两个选项分别为游戏开始选项以及help选项用于介绍游戏规则。在游戏内容中则呈现了牌桌骰子，猜测的记录，以及玩家进行决策的必要按键，结束后对手的骰子的呈现。最后通过弹窗展示胜负结果

### 2.算法或公式

- 1) player类的实现

实现了玩家选取点数，选择guess或者liar的功能

- 2) ZAI类的实现

实现了ZAI策略的算法

- 3) UI类的实现

包括StartUI、游戏界面的实现，以及结束时双方骰子结果的展示和公布胜负的弹窗

- 4) game类

实现了整体游戏的逻辑

## 实现方法

- a、Easyx图形库的使用实现界面的展示
- b、基于概率论知识实现AI的策略
- c、鼠标事件的实现

## AI策略的详细解释:

我们首先考虑一个朴素的情况, 此时对手给出猜测[4,1], 我们手上的骰子点数为[1,1,2,3,5] 那么一个很简单的想法就是我们计算对手手中拥有两个1的概率, 同时我们知道对手点数为1的骰子个数服从二项式分布, 由此我们计算出  $P = 1 - \binom{0}{5}(\frac{1}{6})^0 * (\frac{5}{6})^5 - \binom{1}{5}(\frac{1}{6})^1 * (\frac{5}{6})^4 \approx 0.8$

那么显然这是个概率很大的事件, 我们应该选择加大筹码

那么我们该如何计算选择的筹码呢, 一个很暴力的思想油然而生:

我们可以枚举所有合法的猜测, 然后选出对手喊liar的概率最小也就是对手按上面方法计算得到的概率最大的猜测

此时我们面临两个问题:

1.上面的方法中我们是知道自己的骰子的分布的, 但是此时我们并不知道对手骰子的分布, 可是对手向我们提供了一个信息也就是他的guess[r,d], 那么我们由此根据贝叶斯的思想, 选择对手做出这个决策先验概率最大的分布。这实际上是很难算出来的因此我们可以采取如下的猜测方式:

分两种情况考虑, 第一种  $d = 1$ ,

那么此时我们认为对方至少拥有

$$\begin{cases} r - d & r > m \\ r & r \leq m \end{cases}$$

个点数为1的骰子(其中m为我们拥有的点数为1的骰子个数), 则剩下的骰子则平摊给其他点数的骰子, 此时我们就得到了对手的maybeDice, 从而计算每个guess 的概率

第二种情况, 若  $d \neq 1$ , 那么我们此时就要考虑点数为1的骰子对结果的影响, 依据高中知识我们可以知道对手有0个或者1个点数为1的骰子的概率是最大的, 那么我们认为对手拥有1个点数为1的骰子, 然后再按照case1计算出骰子的分布即可

2.这个概率的计算, 因为我们发现我们的猜测是点数为x的骰子数量至少为y的, 用二项式分布这是不好计算的。我们可以利用中心极限定理将其转化为标准正态分布计算

我们知道点数为r的骰子 ( $r \neq 1$ ) 的个数服从二项式分布  $X \sim B(5, \frac{1}{3})$  (以为点数为1的骰子可以当做任意点数的骰子) 我们有:

$$E(x) = \mu = np = \frac{5}{3}$$

$$D(X) = \sigma^2 = npq = \frac{10}{3}$$

$$\text{那么有: } \frac{X - \mu}{\sigma} \sim N(0, 1)$$

因此我们只需要通过标准正态分布的CDF即可计算概率

结合这两个我们可以计算出每个合法猜测的prob, 从中选出概率最大的作为本次的猜测, 如果最大的概率都小于我们设定的阈值, 我们就选择liar

## 五、测试

对本游戏的AI的性能进行了测试, 发现可以维持高于50%的胜率

## 六、收获

- 1.学习了Easyx中图形化知识
- 2.对类与对象有了更加深刻的理解
- 3.将学会的概率论知识进行了实践

## 七、反思

游戏的UI界面可以更加精美，结束可以加上动画，AI策略猜测对方骰子分布可以利用更加准确的贝叶斯估计