厦門大學

Management System Simulation Report

A Coin-Flipping Game

Group Members： Zhemian Zhao 17420182200850
Yifei Zhu 21620152203091

Tutor： Dr. Di Xu
Dr. Wei Zhang

2021.04.25

Catalogue

1.Problem Description

1.1 Problem Background

We need to play a game in which we repeatedly flip an unbiased coin until the difference between the number of heads tossed and the number of tails tossed is three. We are required to pay￥1 for each flip of the coin, but we receive￥8 at the end of each play of the game. We are not allowed to quit during a play of the game. Thus we win money if the number of flips required is fewer than eight, but we lose money if more than eight flips are required.

1.2 The Simulation Variables

Table 1.1 Input variables

Input		
Data Type	**Name**	**Description**
int	*m_cost*	The cost for each flip of the coin
int	*m_condition*	The condition for the game to stop
float	*m_revenue*	The revenue of per round
int	*m_replication*	The times of replication

Table 1.2 Output variables

Output		
Data Type	**Name**	**Description**
float	*m_flips*	The expected number of flips
float	*m_profit*	The expect revenue of the game

2.Model

2.1 The Probability Distribution

The probability distribution for the outcome of a flip is that the probability of a head is ½ and the probability of a tail is ½. Before we use computer to simulate the game, we should formally construct a full-fledged simulation model. The stochastic system being simulated is the successive flipping of the coin for a play of the game.

2.2 The Simulated Clock

The *simulation clock* records the number of (simulated) flips t that have occurred so far.

2.3 The State of System

The information about the system that defines its current status, i.e., the state of the system, is

$$N(t) = \textit{number of heads minus number of tails after t flips.}$$

The *events* that change the state of the system are the flipping of a head or the flipping of a tail.

2.4 The State Transition Mechanism

The *event generating mechanism* is the generation of a random digit. The *state transition mechanism* is to set

$$\begin{aligned} N(t) &= N(t-1)+1 \quad \textit{if flip t is a head} \\ N(t) &= N(t-1)-1 \quad \textit{if flip t is a tail} \end{aligned}$$

The simulated game then ends at the first value of t where N(t)=±3, where the resulting sampling observation for the simulated experiment is 8-t, the amount won (positive or negative) for that play of the game.

3.Flow Chart

3.1Code

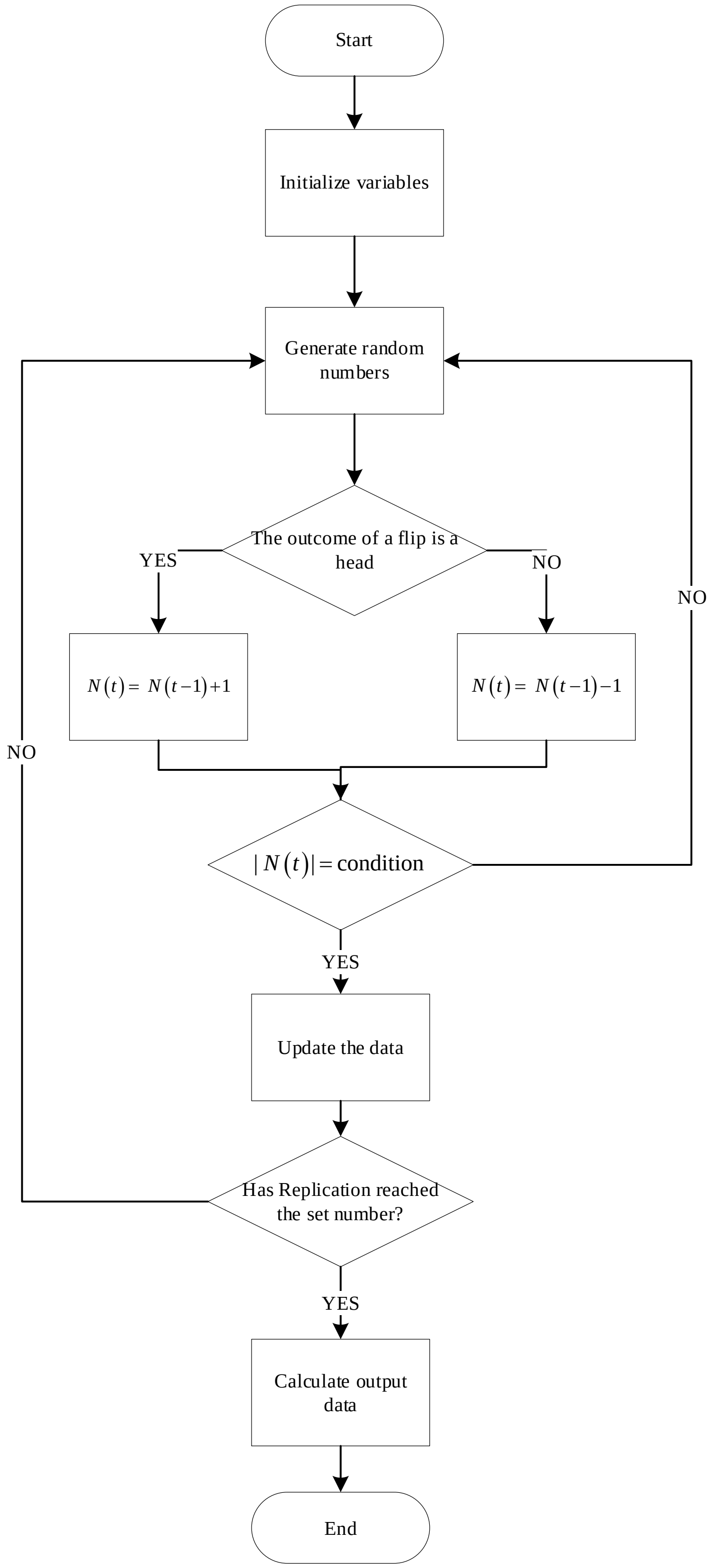
```
void Coin::OnBnClickedButton1()
{
   UpdateData(1);
   int condition = 0;
   int flips = 0;
   float replication = 0;
```

```
        float sum_flips = 0;
        while(replication<=m_replication)//replication 不达到设定的次数就需要一直重复试验
        {
            do
            {
                float randmon_number = lcgrand(1);//产生随机数
                //正反面
                flips += 1;
                if (randmon_number < 0.5)
                    condition += 1;
                else
                    condition -= 1;
            } while (abs(condition) != m_conditon);

            replication += 1;//每做完一次+1（进行计数）
            sum_flips =sum_flips+flips;//将一次实验的结果进行存储
            condition = 0;
            flips = 0;//将 flips 和 condition 归 0，准备进入下一次循环
        }
        m_flips = sum_flips/m_replication;
        m_profit = m_revenue - m_flips*m_cost;
        UpdateData(0);
    }
```

3.2Flow Chart

Figure 3.1 Flow Chart



4.Output Analysis

4.1 Monte Carlo Method (statistical test method)

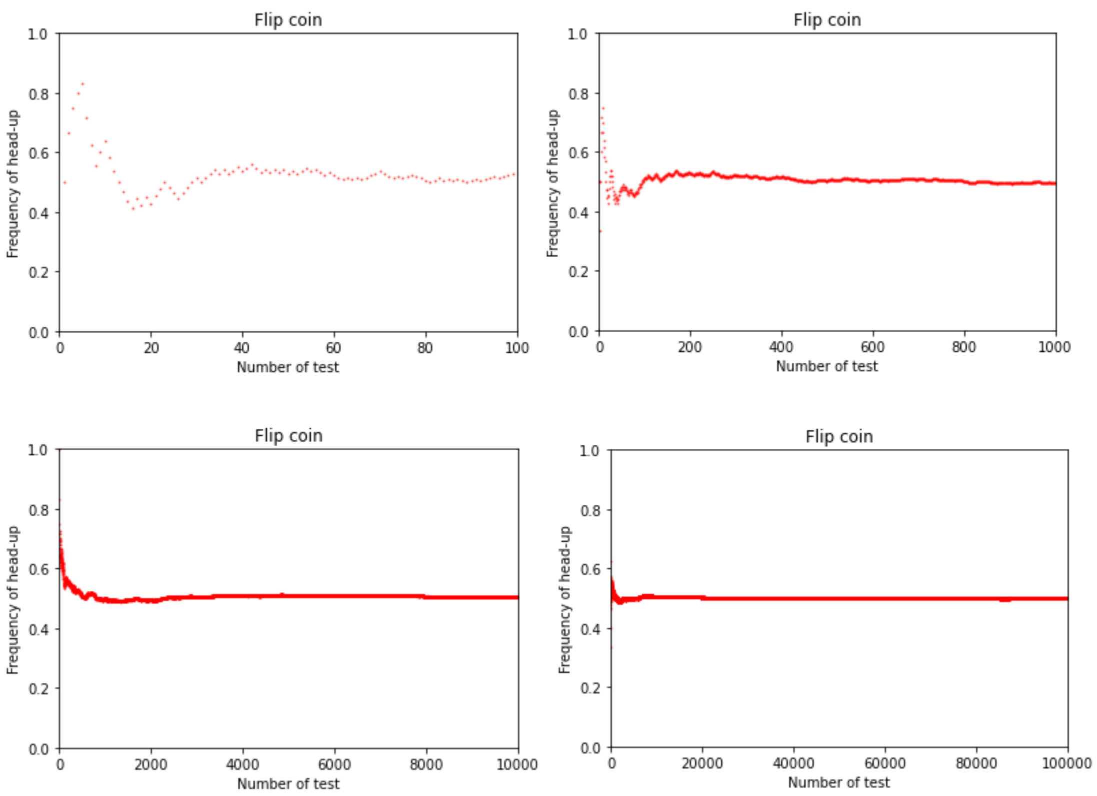
4.1.1 Basic Train of Thought

When solving the problem is the probability of a random event, or is a random variable of expectation, through some kind of "experiment" method, frequency of such events to estimate the probability of the random events, or get some of the numerical characteristics of the random variable, and use it as a solution of the problem. We can call it "probability approximated by frequency."

4.1.2 Example – Flip Coin Experiment

As the number of flips increases form 0 to 20 to 200 to 2000 to 100000,the frequency of head-up is basically close to the true probability of 0.5.

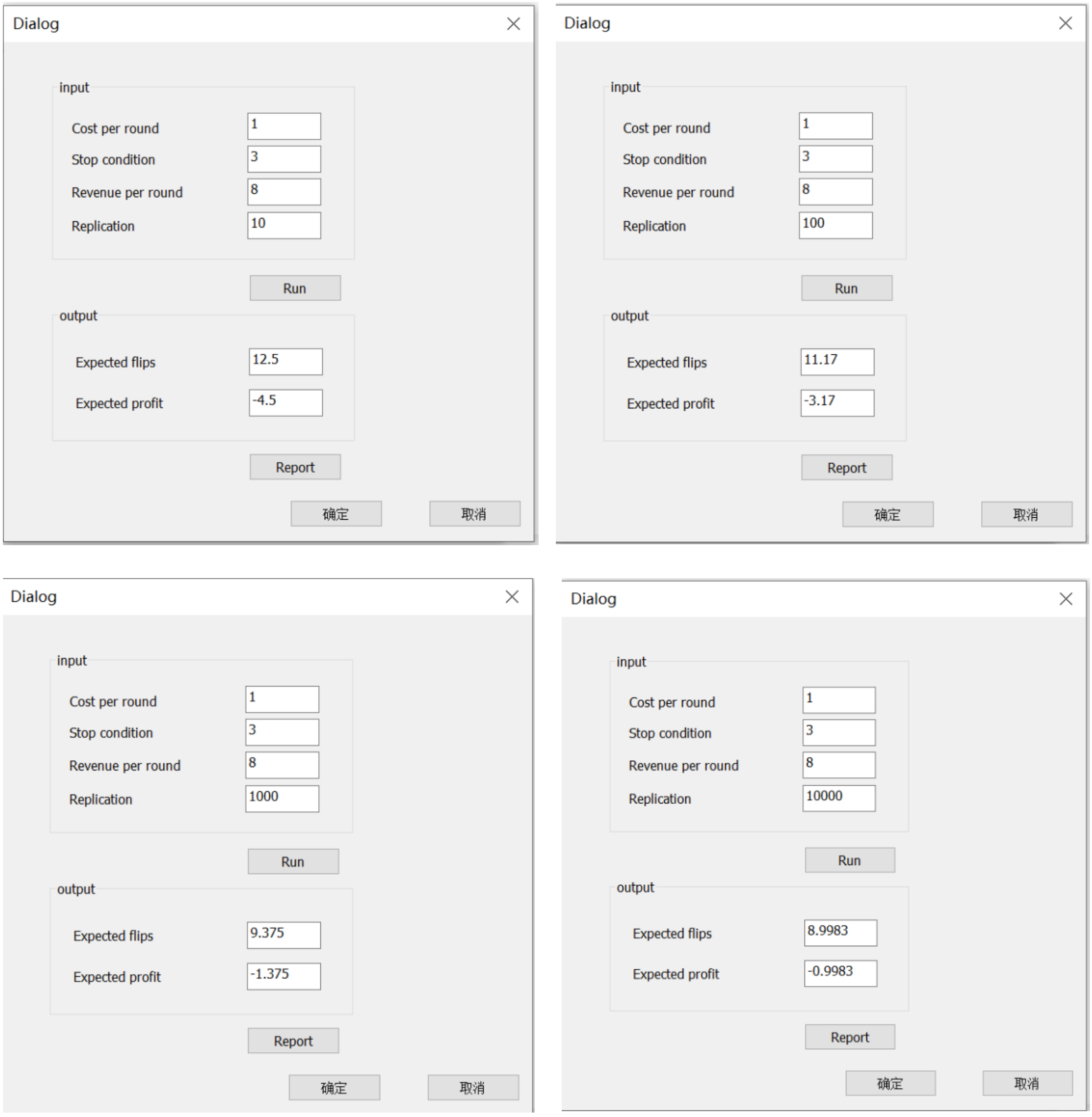
Figure 4.1 Coin toss experiment under different numbers



4.2 Output analysis

We can get the following slightly different results when Replication= 10,100,1000,10000:

Figure 4.2 Program dialog interface



As you can see, with the increase in the number of experiments, the number of coin flips and the return tend to be stable. For rational players, this game will not be played. For businesses, playing this game will be profitable in the long term.

5. Summary

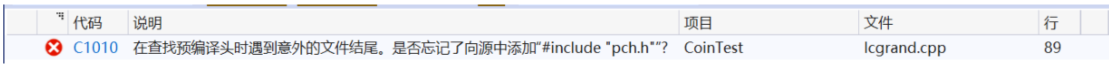
5.1 Experiments need to be done more to increase proficiency.

5.2 Communicate more if you do not understand the operation.

Even if both of us install the same version of software, there may not always report the same errors. So,we should learn to analyze the error notifications.

Such as:

Figure 5.1 Error instructions



The computer didn't have the "#include "stdafx.h" bug that the teacher mentioned in class, So, when I started doing it with the prescribed order, I found more bugs.