**“拈游戏”教学游戏程序**

**设计说明文档**

学 院： 计算机与软件学院 专 业： 计算机科学

班 级： 13级计科1班 学 号：2013150064

学生姓名： 刘沛奇 指导教师： 杨烜

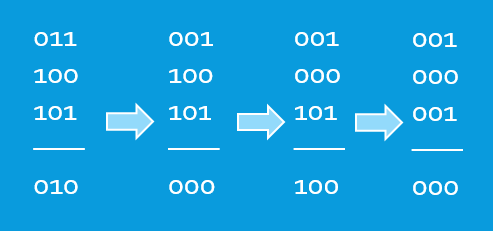
1. **游戏介绍**



“传统版拈游戏”

有若干堆硬币，每堆有若干枚。两人轮流从其中的一堆中至少取出一枚硬币，最多可以把一堆取完。取走整个游戏最后一枚硬币的人为胜者。

1. **减可变规模算法与传统拈游戏**



对于多堆的拈游戏，20世纪初哈佛的数学家才发现了它的制胜策略，通过计算每堆硬币数的二进制异或值，从而得出最佳取出的硬币数

例如当前有三堆硬币，数量是3,4,5，异或值是010，我们要让这个异或值变成0，所以最佳策略就是“从第一行中取出2枚”

如果对方从第二行取走4枚，那么当前异或值将变为100

我们要让异或值变成0，那么就应该从第三行也取走4枚硬币

现在只剩下2枚，答案已经很明显了

1. **传统拈游戏最佳策略的证明**

拈游戏这种博弈问题，最重要的是寻找必败态。这个必败态的的意思就是，这样一种局面摆在面前的话先手必败。其严格定义如下：1.无法进行任何移动的局面是必败态；2.可以移动到必败态的局面是非必败态；3.在必败态做的所有操作的结果都是非必败态。这个还是很好理解的吧，就是自己处在非必败态上总能移动到必败态把必败态留给对方，而对方处在必败态的话总是只能移动到非必败态，把非必败态留给自己，然后自己继续虐对方。

而对于传统拈游戏，局面是必败态当且仅当所有堆硬币的数量都异或起来结果为0，即a1^a2^...^an=0！为了证明之，我们只要证明它满足上述必败态的三条性质即可。

第一个命题显然，最终局面只有一个，就是全0，异或仍然是0。

第二个命题，对于某个局面(a1,a2,...,an)，若a1^a2^...^an!=0，一定存在某个合法的移动，将ai改变成ai'后满足a1^a2^...^ai'^...^an=0。不妨设a1^a2^...^an=k，则一定存在某个ai，它的二进制表示在k的最高位上是1（否则k的最高位那个1是怎么得到的）。这时ai^k<ai一定成立。则我们可以将ai改变成ai'=ai^k，此时a1^a2^...^ai'^...^an=a1^a2^...^an^k=0。

第三个命题，对于某个局面(a1,a2,...,an)，若a1^a2^...^an=0，一定不存在某个合法的移动，将ai改变成ai'后满足a1^a2^...^ai'^...^an=0。因为异或运算满足消去率，由a1^a2^...^an=a1^a2^...^ai'^...^an可以得到ai=ai'。所以将ai改变成ai'不是一个合法的移动。证毕。

1. **传统拈游戏最佳策略的算法表示**

//算法function(num[],n)

//输入：数组num（存储每堆硬币数量），堆数n

//输出：最佳策略的堆号ln和对应的数量cn，如果没有最佳策略则输出ln=-1,cn=-1

result=num[0] xor num[1] xor ... num[n-1]

if result==0

ln=-1

cn=-1

else

//找出result二进制最高位的位数high

high=0

temp=result

while temp!=0 do

temp=temp/2

high++

for i=0 to n-1 do

if num[i]二进制从左往右第high位==1

ln=i

cn=num[i]- ( num[i] xor result )

break

**主要操作：比较**

**T(n)worst=n**

**T(n)平均=(1+n)/2**

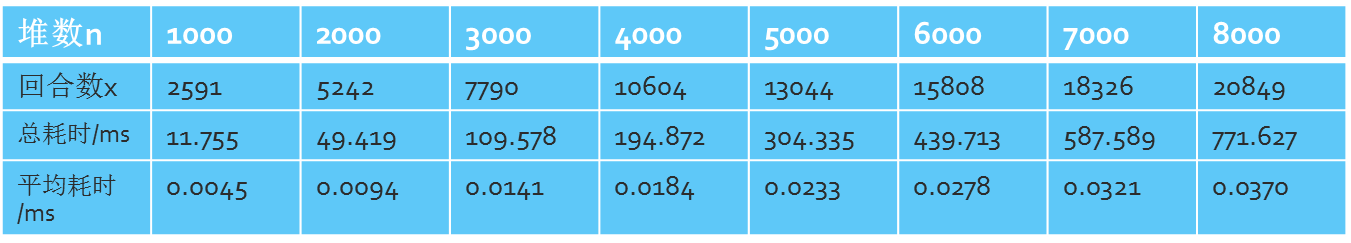
**效率属于Ɵ(n),是线性效率**

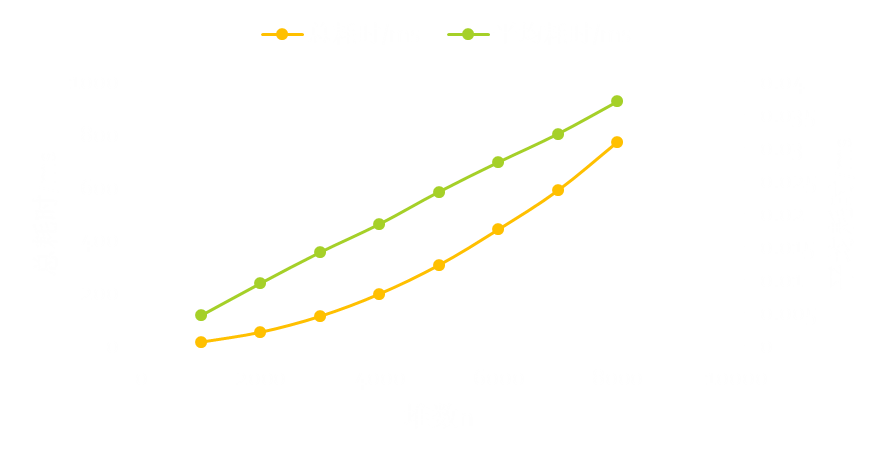
**总时间C(n)=xT(n) x为我方的总回合数，由于x的值很难估计，所以很难算出C(n)的准确效率**

1. **传统拈游戏时间统计与分析**

由于x不好估计，我们通过模拟堆数n=1000~8000（间隔1000），每堆数量随机为1~100的传统拈游戏。对手默认随机

测试每次游戏的回合数x，总耗时，最后算出平均时间t，理论上t应该线性增长





结论：平均时间符合线性效率

1. **游戏目录说明**

* 文件夹“battle存放对战模式素材，begin存放开始界面，bg存放背景，face存放头像，font存放字体文件，other存放其他素材，say存放对话文件，title存放标题图片，visual存放人物立绘”
* 其他dll为游戏运行时需要外链的库文件
* 开始游戏.exe为游戏入口程序

1. **游戏程序设计**

游戏本体采用纯C++编写，绘制图像的部分使用第三方开源的c++图形库“SDL”。

由于游戏演示流程复杂，单纯的图形库难以满足需求。我对”SDL图形库”的底层结构体和函数进行二次封装，将游戏中需要用到的及部分组件：图像、对话、文字三者封装为类SurfaceClass、MessageClass、StringSurface，并封装好显示用的相关方法。同时由于游戏出现了硬币组这个组件，我通过基础组件封装了类CoinSurface用于描述硬币堆。之后我将场景的绘制封装为工具类Drawable，方便统一绘制场景。最后是实现各个游戏中的方法，用于初始化、刷新、暂停、等待等。

而核心的必胜策略算法，我在游戏中封装为方法calculate。

我的游戏基本显示方法，就是用类Drawable不断更新绘制每个场景的各个图片组件SurfaceClass文字组件StringSurface。同时通过事件队列event响应用户操作，更新界面。

SDL介绍：

1. SDL（Simple DirectMedia Layer）是一套开放源代码的跨平台多媒体开发库，使用C语言写成。SDL提供了数种控制图像、声音、输出入的函数，让开发者只要用相同或是相似的代码就可以开发出跨多个平台（Linux、Windows、Mac OS X等）的应用软件。目前SDL多用于开发游戏、模拟器、媒体播放器等多媒体应用领域。
2. **SDL（Simple DirectMedia Layer）**被广泛的用于许多著名的游戏。最著名的游戏是赢得Linux组游戏开发大奖的 文明：权利的召唤（Civilization: Call To Power）。
3. 官方网址：[www.lib](http://www.libsdl.org/)**[sdl](http://www.libsdl.org/)**[.org/](http://www.libsdl.org/)
4. 中文教程：<http://tjumyk.github.io/sdl-tutorial-cn/>

***文件sdlClass.h***

该文件存放了所有的类声明、实现和所有方法的声明与实现

* 图像显示部分
* class SurfaceClass;//基本图像类

功能：存储并绘制一张图片

方法：

1. 构造方法SurfaceClass(const string &path, signed \_\_int16 x = 0, signed \_\_int16 y = 0)//通过图片路径和位移参数设置图像信息
2. void setNewPic(const string &path)//设置新图像
3. void setOffset(signed \_\_int16 x, signed \_\_int16 y)//设置新位移
4. void toScreen()//显示图像到屏幕
5. void gradient(SurfaceClass & aPic,int ms=1000)//产生透明渐变效果过渡当前图像到下一个图像
6. void copyScreen()//获取整个屏幕内容到该类对象
7. void setAlpha(int num)//设置透明度

* class MessageClass; //对话类

功能：按照一定的速度显示对话内容

方法：

1. 构造方法MessageClass(int x,int y,const string &fontName, int fontSize = 20,

unsigned \_\_int8 red = 0, unsigned \_\_int8 green = 0, unsigned \_\_int8 blue = 0)这里可以设置对话的位置，字体，字号，颜色

1. void setFont(const string &fontName, int fontSize) //设置新字体
2. void setColor(unsigned \_\_int8 red, unsigned \_\_int8 green, unsigned \_\_int8 blue)//设置新颜色
3. void drawMessage(string\* str, int lineNum)//一次性绘制信息
4. void readMessage(string\* str,int lineNum)//按照一定的对话速度显示对话

* class Drawable;//绘制方法类

功能：通过向量存储一个场景里面的所有SurfaceClass，以便一次性绘制

方法：

1. void push(SurfaceClass\* aPic)//将SurfaceClass指针加入其中
2. void cleanAll()//清空所有组件
3. void drawAll()//绘制所有组件

* class CoinSurface;//硬币类，是基本图像类的复合类

功能：显示并操作硬币的消失

方法：

1. 构造方法CoinSurface(const string& path, signed \_\_int16 x, signed \_\_int16 y,int xspace,int yspace)//传入硬币的路径，偏移量，间距
2. void setCoinsNum(int coinsNum[], int line)//设置硬币数量
3. void reset()//重置硬币状态
4. void destoryCoins(int lineNo, int coinsNum)//显示硬币消失的渐变动画
5. void pushToDrawable()//将所有硬币对象放置到绘图区

* class StringSurface;//文本类，继承SurfaceClass

功能：显示文本（非对话，主要用在对战模式的各种信息更新）

方法：

1. 构造方法StringSurface(const string &fontName, int fontSize = 20, signed \_\_int16 x = 0, signed \_\_int16 y = 0,

unsigned \_\_int8 red = 0, unsigned \_\_int8 green = 0, unsigned \_\_int8 blue = 0)//设置字体，字号，偏移量，颜色

1. void setFont(const string &fontName, int fontSize)//设置字体
2. void setColor(unsigned \_\_int8 red, unsigned \_\_int8 green, unsigned \_\_int8 blue)//设置颜色
3. void setStr(const string& str)//更改文本内容

* extern SDL\_Surface\* screen;//屏幕本体

功能：显示主窗口

* extern Drawable drawable;//全局绘制类对象

功能：Drawable类的一个实例，用于绘制画面

* extern SDL\_Event event;//事件队列

功能：处理键盘鼠标事件

* inline void initScreen(int width, int height, const string &title, const string &icoPath);//屏幕初始化方法

功能：调用底层SDL函数初始化窗口

* inline void closeScreen();//屏幕关闭方法
* inline void refreshScreen();//屏幕刷新方法
* inline void stopScreen(unsigned int ms);//屏幕暂停方法
* inline void waitToClick();//等待用户点击方法
* void readFileString(string\* strArray, ifstream& inFile, int num)//读取文件流中的字符串
* 字符串处理部分
* stringstream stream;//格式转换流
* void toString(string& str, int num)//将数组转换为字符串
* string getNumString(int num)//获取数字对应的字符串
* string getRoundString(int round)//获取回合数的字符串
* string getBinaryString(int num)//获取一个数的二进制字符串
* int getDecimalNum(const string& str)//获取一个字符串的二进制
* string getResultString(int num[],int lineNum)//获取二进制数组异或值
* void calculate(int num[], int lineNum,int& il,int& in)//计算最佳策略
* string getPromptString(int num[], int lineNum)//获取最佳策略提示字符串

通过SurfaceClass可以记录一个图像，通过多个SurfaceClass的堆叠，和逐帧的刷新，便可以产生游戏般的显示效果。通过MessageClass实现了如游戏中对话一样的显示效果。而StringSurface继承了SurfaceClass用于实现游戏中所有的文字效果。CoinSurface是一个复合类，通过产生SurfaceClass数组来实现硬币的产生与消失动画。最后Drawable将记录所有的SurfaceClass对象，负责每帧的刷新。而各式各样的字符串方法可以在对战中用于更新各项数据。

***文件main.cpp***

其中有两个入口函数，一个指向教学模式，一个指向对战模式。

在教学模式下，用户可以学习什么事“拈游戏”，“拈游戏”如何取胜，通过简单的交互了解这里面使用了什么算法等等。

而对战模式，通过6关的人机对战，感受“拈游戏”的过程，更深刻的理解“拈游戏”的算法经过，学会“拈游戏”的必胜策略。

* 教学模式

这个模式主要由对话和图片组成，使用到的类包括SurfaceClass和MessageClass。类Drawable存储所有的图片元素，负责每次切换画面时不断刷新。通过不断绘制对话，配合背景的动画演示效果和与用户的互动，让用户学会“拈游戏”



中间的简单互动环节有使用到硬币，这里便是使用了CoinSurface进行绘制

对话文本：

/say/input1.say

/say/input2.say

大家好，我是正义的伙伴小神弟弟。

也是这个游戏的作者。

前段时间，发生了一件事情，促使我想制作这个游戏。

我看到身边有男生经常和女同学玩一种取硬币的游戏，

而男生通过某种技巧，使得女生很难取胜。

从而赢得女生的掌声。

作为正义的伙伴的我有点瞧不起那些男生。

因为这个叫做“Nim”的游戏，是有必胜法则的。

……

* 对战模式



这一部分需要若干个SurfaceClass和StringSurface记录各项数据

1. 左上角显示关卡数和当前是轮到谁的回合
2. 中间为硬币显示面板，左边部分是行数和个数，右边是硬币的图示
3. 右边上部分是玩家信息，每次输入要取走的行数和个数后会在这里显示
4. 右边中间部分是提示面板，显示了必胜策略的计算过程。当轮到自己的回合时，通个单击“提示”按钮程序会调用最佳策略算法显示出当前的最佳策略。同个回合重复使用无效。
5. 右边下部分是AI信息面板，IQ代表AI可能采取最佳策略的概率。例如IQ0%时，AI采取最佳策略的概率是0%，也就是100%随机取值

当点击提示时，会调用calculate计算出最佳策略，而AI也会有一定概率使用calculate计算出的最佳策略。

每个关卡会将上一关的胜者的设置为先手，自己的回合时，程序将等待用户点击，如果点击“提示”将显示最佳策略。点击“输入下一组”则可以在命令行中输入堆数和个数。轮到AI时，会根据IQ算出概率，根据概率决定是否采取最佳策略，如果没有执行最佳策略则随机从一堆中拿走随机个硬币。直到硬币数归0。

在6关完全结束后会给出总分，游戏结束。

***文件传统拈游戏计时.cpp和单堆拈游戏计时.cpp***

由于效率不需要图形界面，所以单独分离出游戏流程和核心算法进行测试