深圳大学考试答题纸

(以论文、报告等形式考核专用)  
二○ 一四 ～二○ 一五 学年度第 二 学期

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程编号 | 1500770003 | | 课程名称 | | 算法设计与分析 | | 主讲教师 | | 杨烜 | 评分 |  |
| 学 号 | 2013150064 | | 姓名 | 刘沛奇 | | 专业年级 | | 计算机科学与技术13级1班 | | | |
|  | | | | | | | | | | | |
| 教师评语： | | | | | | | | | | | |
| 题目： | | “拈游戏”算法教学游戏设计 | | | | | | | |  | |

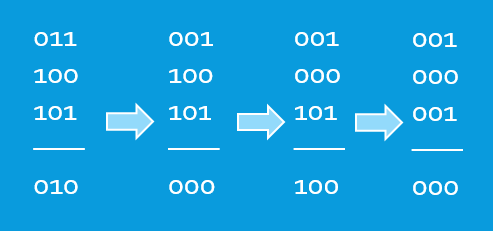
1. **游戏介绍**



“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**

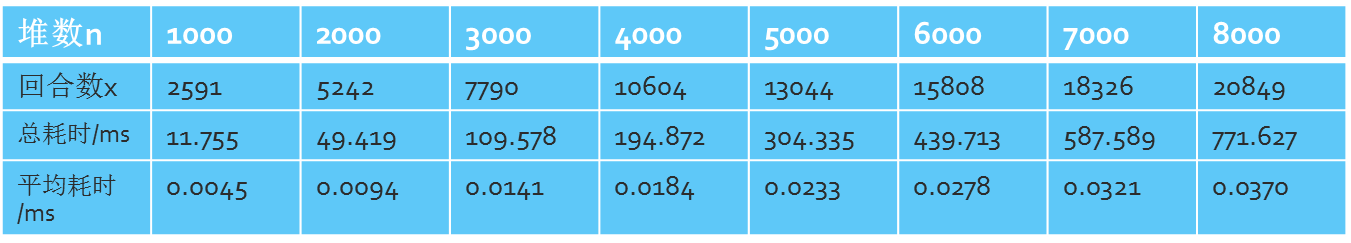
**效率属于Ɵ(n),是线性效率**

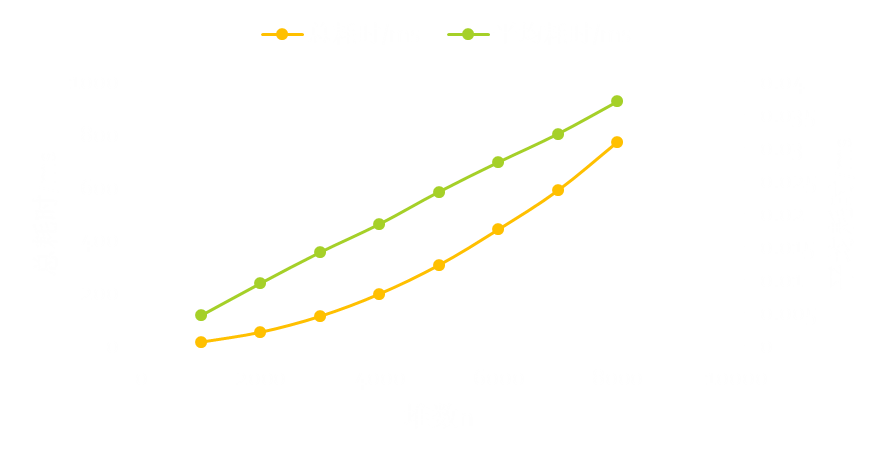
**总时间C(n)=xT(n) x为我方的总回合数，由于x的值很难估计，所以很难算出C(n)的准确效率**

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**sdl**.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

1. **游戏使用说明**
2. **文件夹介绍**

* *计时专用源代码(无图像界面)*

这里面有两份源代码，对应的是PPT中效率分析时使用的程序。由于计算效率需要计算时间，此时图像显示会消耗时间，故专门分离出核心代码进行测试。

* *游戏程序源代码(图形界面)*

带有图形界面的游戏程序的源代码，main.cpp为主程序代码，sdlClass.h是我封装的用于显示图像图像的类，还有各类函数。由于调用了第三方函数库”SDL”，如果您想要把这份源代码在您的本地编译，请阅读“编译我的代码请看.docx”。

* *游戏文件可执行程序(带图像界面)*

里面包含了游戏的所有资源文件，外部链接的运行库，和可执行文件exe，启动游戏时请执行该文件夹中的“开始游戏.exe”

1. **游戏介绍**

**全程请尽量不要狂点屏幕，慢慢来一步步点，否则可能会失去响应，万一失去响应了不要紧张，稍微等一下就好了。**

* 开始界面



这里有三个入口：

1. 教学模式：进行拈游戏的介绍说明，并通过简单的例子展示拈游戏的必胜策略和规模如何减小
2. 对战模式：6回合的人机对战，让玩家充分体会和学习拈游戏的玩法
3. 退出游戏：程序终止并退出

* 教学模式
* 序章：

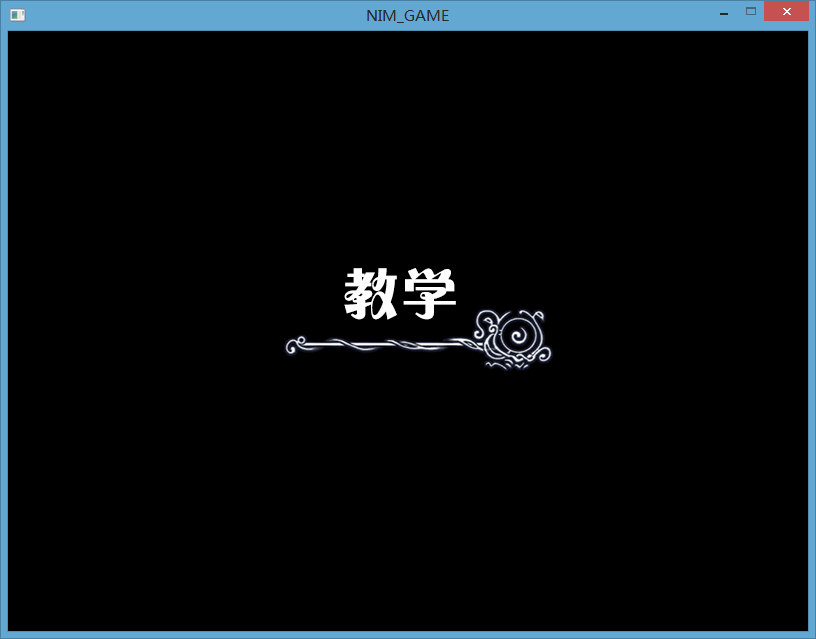




这一部分主要介绍作者为何想制作这个游戏。

**整个教学模式里玩家都通过单击屏幕或者按空格或者回车推进对话的进行**

* 教学





这里会自动演示拈游戏的流程无需单击



**当游戏发展到与电脑测试对战“单堆拈游戏”时，玩家通过按键盘1~7输入个数来控制每次需要拿走的个数。按其它按键无效**



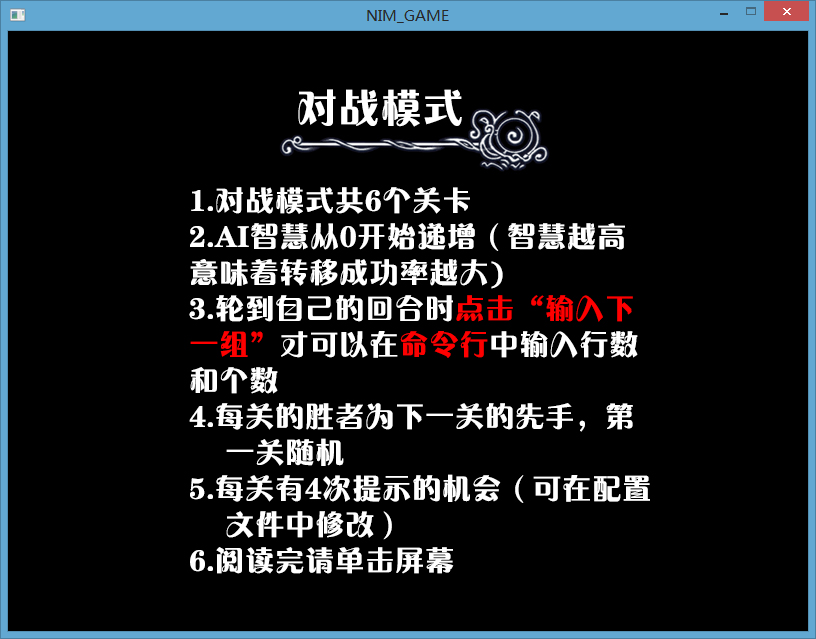
剩下均为教学内容，可浏览至教学结束





教学模式结束后会自动进入对战模式

* 对战模式



第一幕为对战模式的规则介绍，**其中提示机会的个数，在“\游戏文件可执行程序(带图像界面)\battle\config.txt”中修改即可，默认为4**

对战界面：



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
6. **左边中间有一个“输入下一组”按钮，每次轮到自己时，需要单击一次这个按钮，然后才可以在命令行中输入自己要取走的堆数和个数，当输入不合法时会提示**



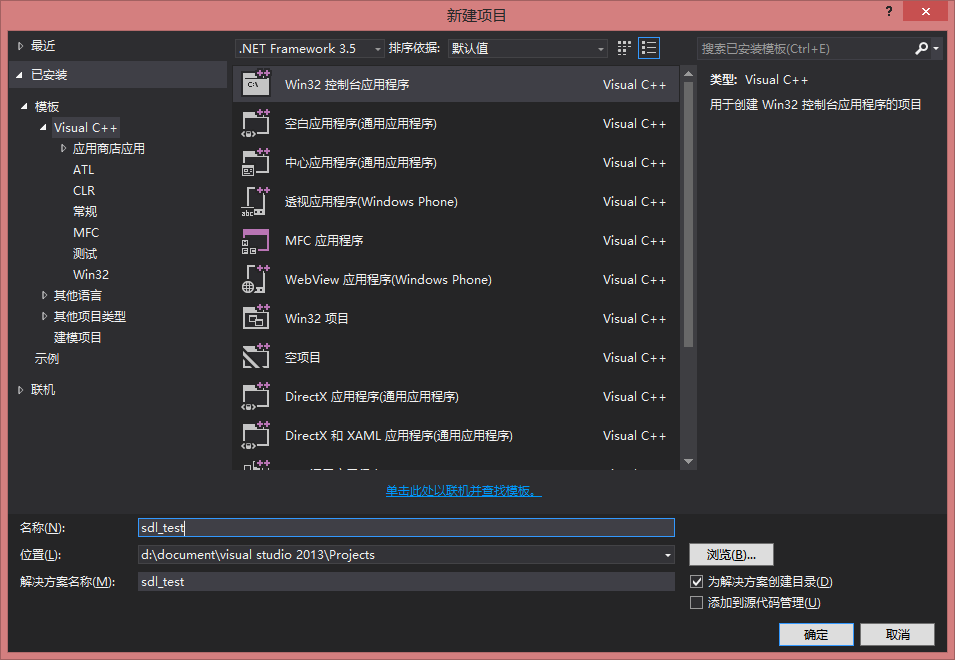
每轮比赛结束后会进入下一关卡，总共六个关卡，每关胜利后会成为下一个关卡的先手，并奖励关卡数\*10的分数。在6关完全结束后会给出总分，游戏结束。

1. **代码编译方法**

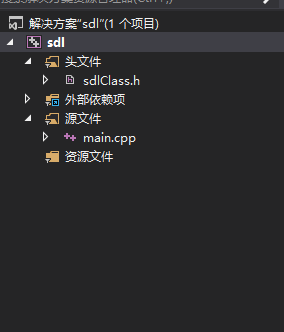
SDL是第三方的图形库，编译前你需要安装这套库。

官方建议是安装在较高版本的VC上，低版本是否兼容不清楚，我自己使用的是VS2013，这里我以VS2013为例来做说明。所需要的文件都在“编译时需要的文件”这个目录里。

1. 打开vs2013安装目录里的“C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\include”文件夹，将“SDL”文件夹拷贝到里面
2. 在vs2013中新建一个C++控制台应用程序并设置为空项目





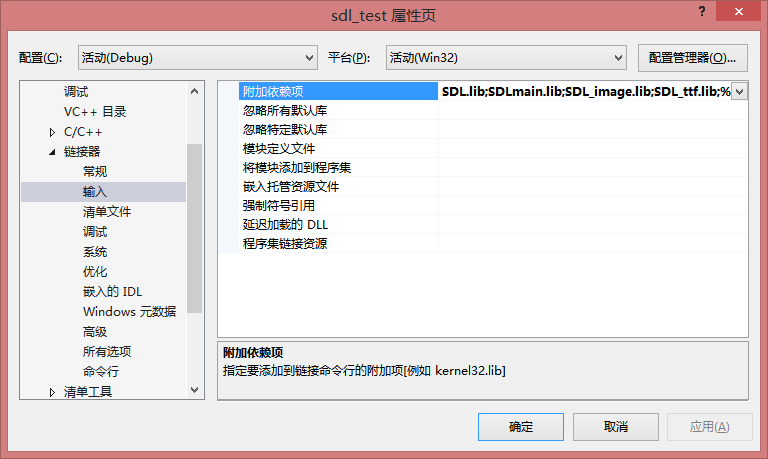
1. 头文件中新建一个文件叫“sdlClass.h”,源文件中新建一个文件叫做”main.cpp”，将源文件中对应的代码拷贝进去编译一次，此时会编译失败。
2. 打开工程的项目设置-链接器-输入-附加依赖项-编辑，填写

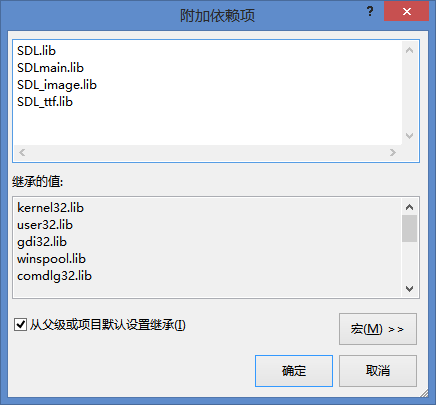
SDL.lib

SDLmain.lib

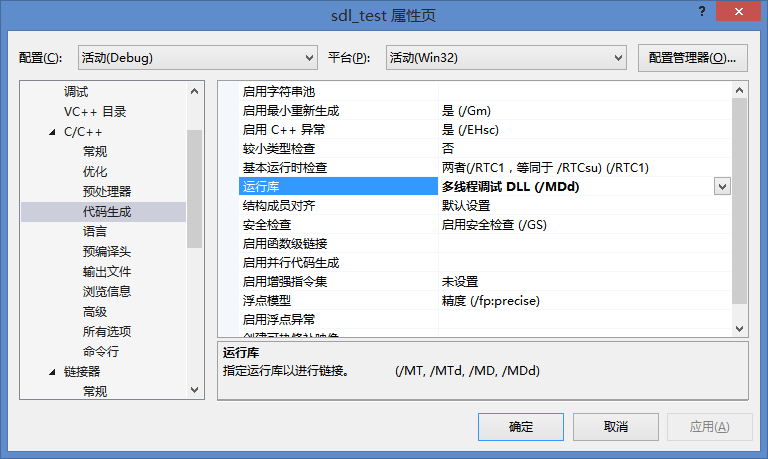
SDL\_image.lib

SDL\_ttf.lib

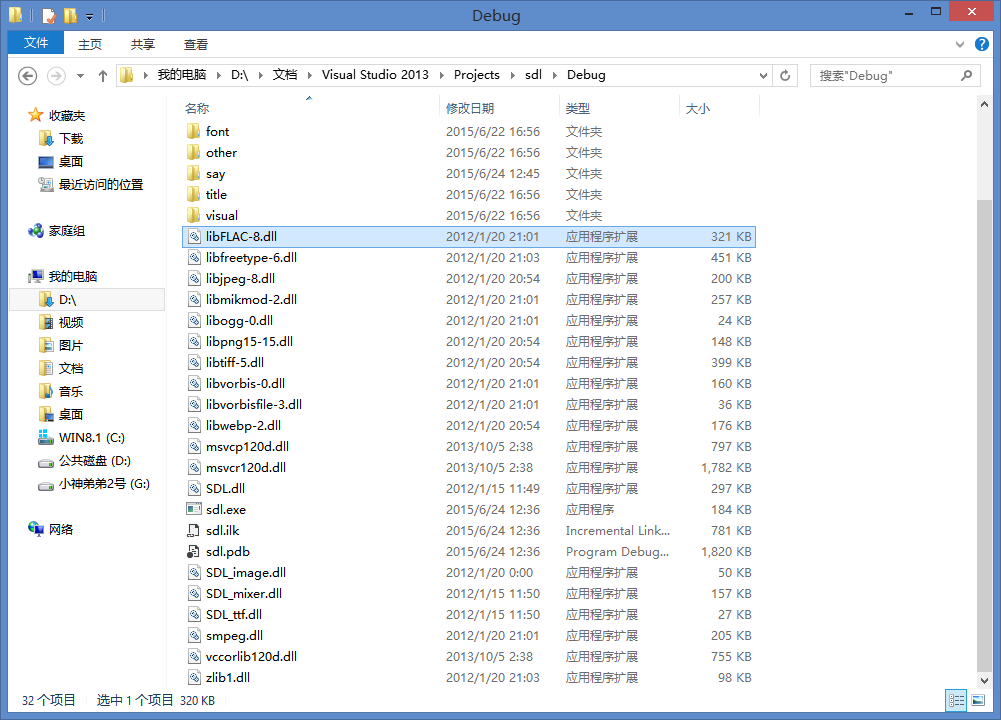




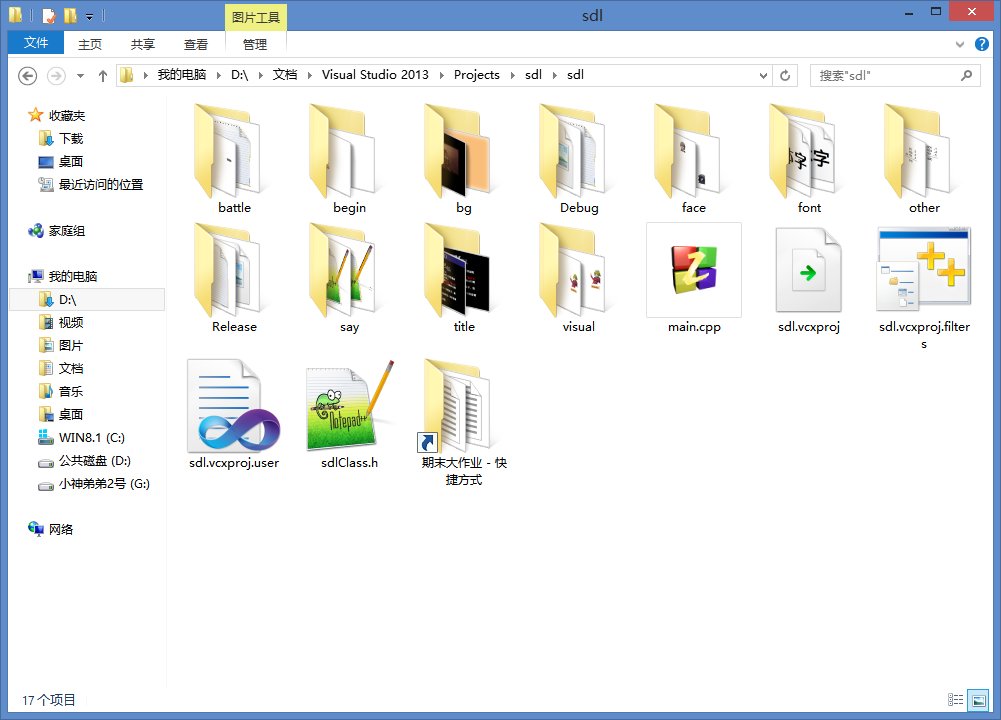
1. 打开工程的项目设置-代码生成-运行库，改为“多线程调试DLL(/MDd)”



1. 打开你的项目目录- Debug，将dll目录里面的东西全部拷贝进去



1. 打开项目目录-项目名称，这里是源代码的目录，将“资源”目录里面的全部东西拷贝进去



8.此时在回到工程中编译就可以运行了