

# 程考经验交流

罗必成 | CGCAD | THSS

2016.01.12

# 考什么

- C++基础语法
- STL库的使用（提高编码速度和正确率）
- MFC图形界面开发（纷繁复杂的API，重点掌握几种模式）
- 数据结构与算法？性能效率？（不考察算法，性能好坏不影响评价）
- 面向对象？设计模式？（不考察软件工程）
- 怎么简单怎么来（目标是通过）

# 考试要求

- 考试环境：Visual Studio 2012
- 考试时长：3 hours
- 2 \* CMD + 1 \* MFC
- 尽早检查开发环境，如有必要尽早申请换机器

# 兵来将挡， 水来土掩

- 编程技巧
  - CMD类型题目编程技巧
  - MFC类型题目编程技巧
- Visual Studio调试技巧
- 备考计划

# 编程技巧

- CMD类型题目
- 完成 > 性能 > 代码可读性 > 注释 （不要觉得自己做不出来就狂加注释）
- C++11 + STL = 事半功倍

# 编程技巧

- CMD类型题目 | 常见考察
- 整数越界、大整数类

```
typedef long long ll;  
typedef unsigned long long ull;  
ll a; scanf("%I64d", &a);  
ull b; scanf("%I64u", &b);
```

# 编程技巧

- CMD类型题目 | 常见考察
- 整数越界、大整数类

```
struct BigInteger {  
    vector<int> s;  
    BigInteger(long long num = 0);  
    BigInteger operator=(long long num);  
    BigInteger operator=(const string &str);  
    //参考白皮书P124  
}
```

# 编程技巧

- CMD类型题目 | 常见考察
- 字符串操作

`char[]` Or `std::string`

- 处理字符串内部的逻辑建议用`char[]`，处理多个字符串间的逻辑建议用`std::string`



# 编程技巧

- CMD类型题目 | 常见考察
- 字符串操作 | 字符串内部逻辑
- 单个字符的判断与转换

```
int isalpha(char); int isdigit(char);
```

```
int isspace(char);
```

```
char tolower(char); char toupper(char);
```

# 编程技巧

- CMD类型题目 | 常见考察
- 字符串操作 | 字符串内部逻辑
- 字符串与数值的互相转换

```
int atoi(const char*);
```

```
long atol(const char*);
```

```
long long atoll(const char*);
```

```
double atof(const char*);
```

```
sprintf(str, "%d", num);
```

# 编程技巧

- CMD类型题目 | 常见考察
- 字符串操作 | 字符串内部逻辑
- 字符串的拷贝拼接与判断

```
strcpy(char *dest, const char *src);
```

```
strcat(char *dest, const char *src);
```

```
size_t strlen(const char *str);
```

```
int strcmp(const char *lhs, const char *rhs);
```

# 编程技巧

- CMD类型题目 | 常见考察
- 字符串操作 | 多个字符串逻辑
- 字符串与数值的互相转换

```
int stoi(const string&); //等同于atoi(str.c_str())
```

```
long stol(const string&);
```

```
long long stoll(const string&);
```

```
float stof(const string&);
```

```
double stod(const string&);
```

```
string to_string(int/long/float/double...);
```

# 编程技巧

- CMD类型题目 | 常见考察
- 字符串操作 | 多个字符串逻辑
- 文件逐行读取

```
ifstream infile("input.txt");  
string line;  
while(getline(infile, line))  
{  
    //do something here  
}
```

# 编程技巧

- CMD类型题目 | 常见考察
- 字符串操作 | 多个字符串逻辑
- 字符串分割（按单个字符）

```
stringstream ss(str);  
string item;  
while(getline(ss, line, ','))  
{  
    //do something here  
}
```

# 编程技巧

- CMD类型题目 | 常见考察
- 字符串操作 | 多个字符串逻辑
- 字符串分割（按字符串）

```
str += delim;
int i = 0; size_t pos = str.find(delim);
while(pos != string::npos)
{
    string item = str.substr(i, pos - i);
    //do something here
    i = ++pos;
    pos = str.find(delim, pos);
}
```

# 编程技巧

- CMD类型题目 | 常见考察
- 集合类型

`vector, deque` (一端延长、两端延长)

`stack, queue` (先进后出, 先进先出)

`priority_queue` (堆)

`set, unordered_set` (自平衡二叉搜索树、哈希表)

`map, unordered_map` (自平衡二叉搜索树、哈希表)

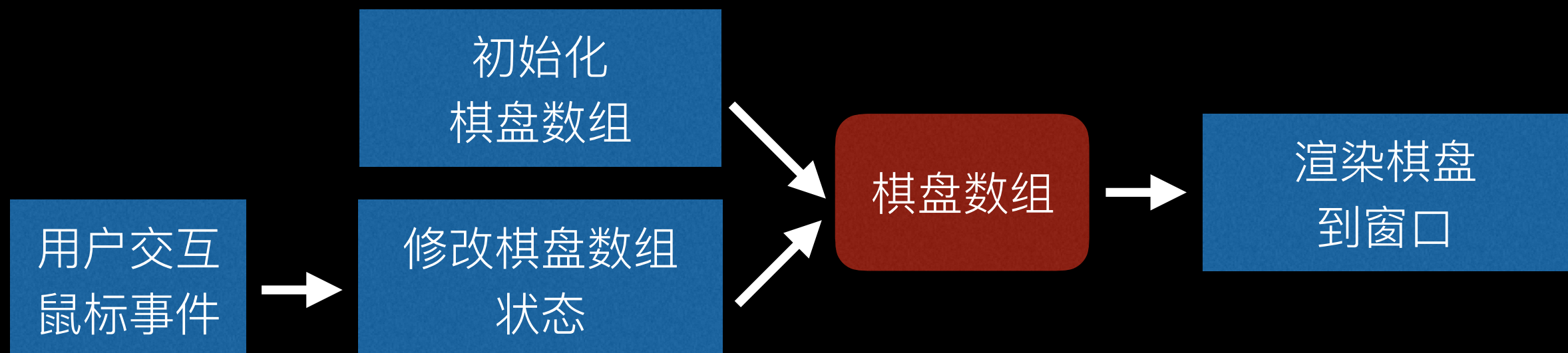


# 编程技巧

- MFC类型题目
- 有界面就有分！（不要纠结于细节）
- 各种棋类游戏、XX管理系统等
  - 基于单文档（SDI）或者对话框（推荐）
  - 熟悉MFC消息处理机制，正确添加事件处理程序
  - 在`OnDraw/OnPaint`函数中实现绘制代码
  - 熟悉`CString/std::string/char*`之间的转换
  - 读写文件、对象序列化

# 编程技巧

- MFC类型题目 | 棋类游戏
- 棋类游戏的棋盘可以抽象为一个二维数组
- 不同的棋子表示数组中不同的值



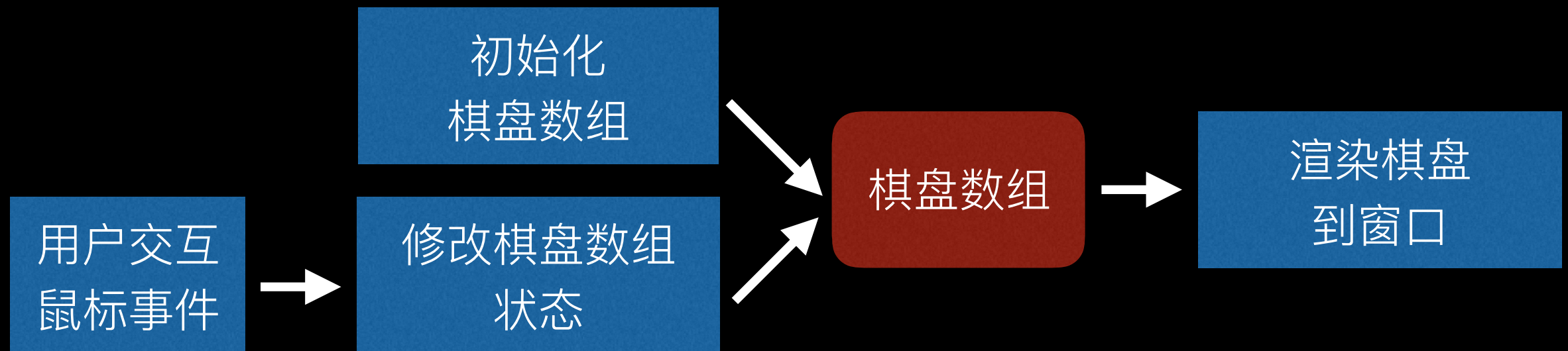
# 编程技巧

- MFC类型题目 | 棋类游戏

OnInitDialog

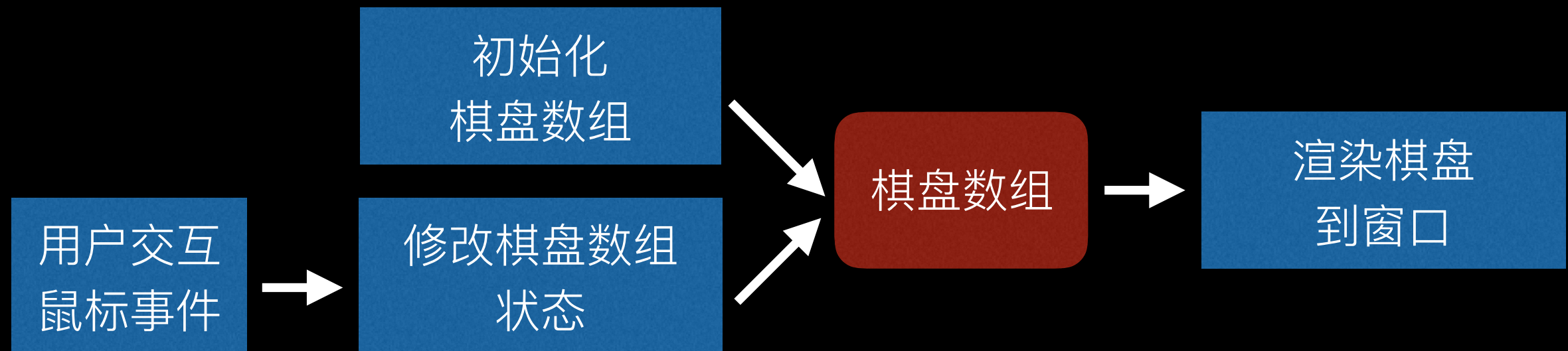
CXXDlg构造函数

CXXDlg.h对话框数据



# 编程技巧

- MFC类型题目 | 棋类游戏



`OnLButtonUp`

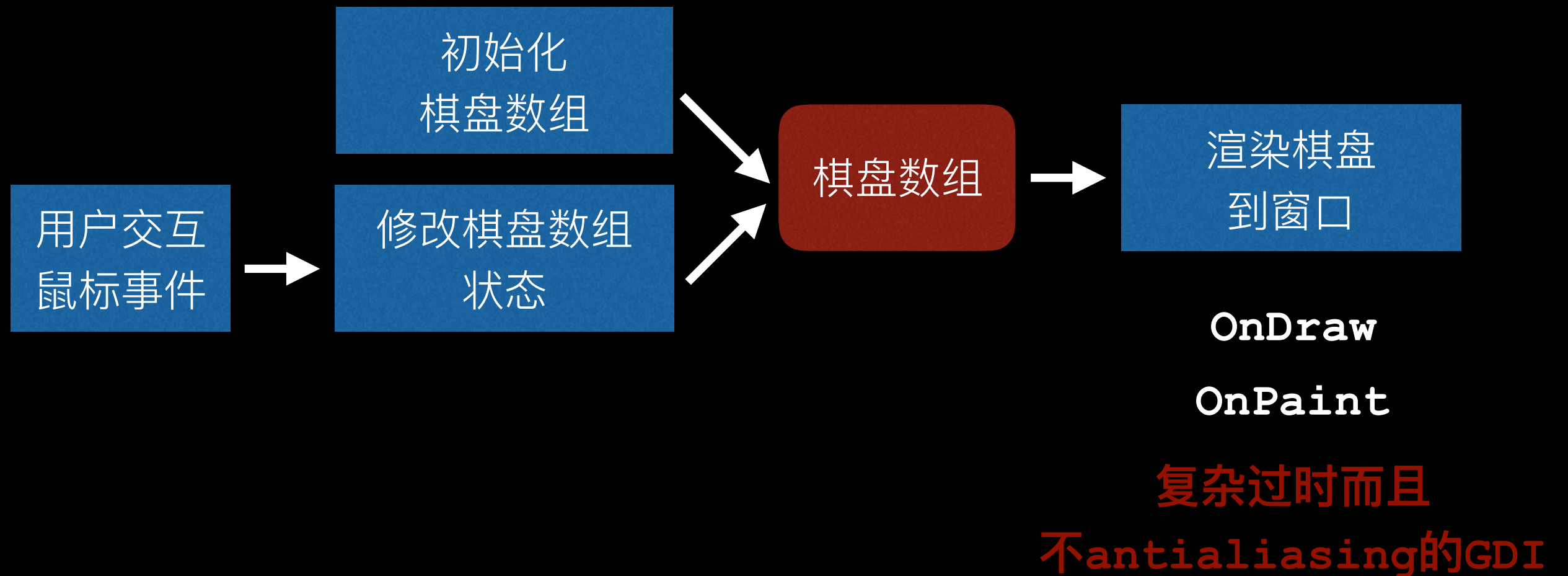
`OnLButtonDown`

`OnMouseMove`

`OnBnClickedXXXX`

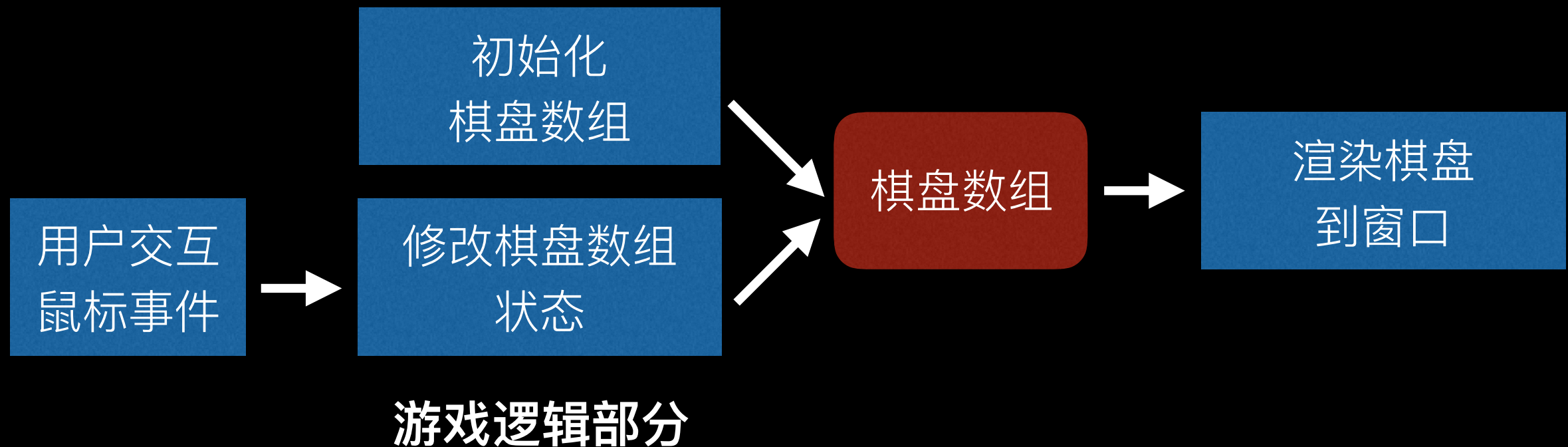
# 编程技巧

- MFC类型题目 | 棋类游戏



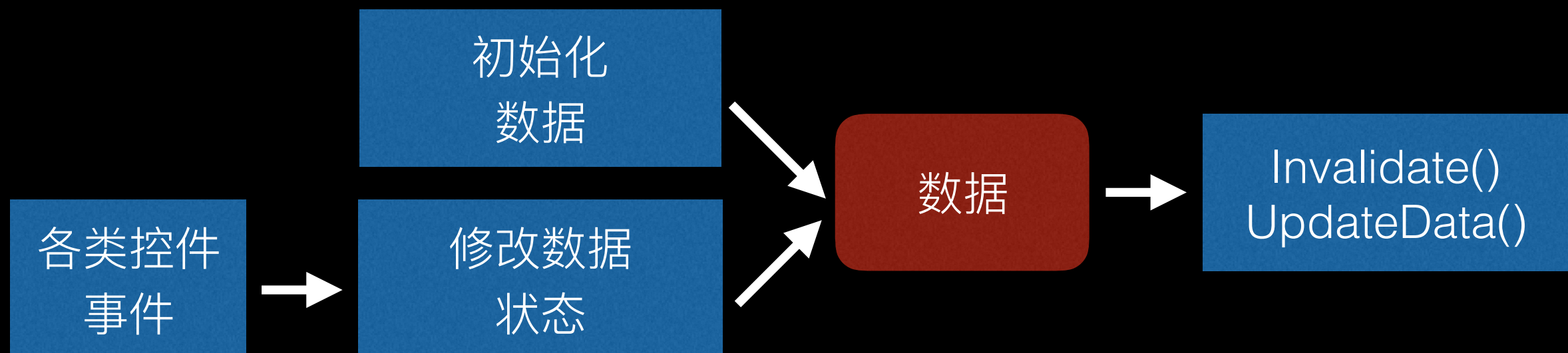
# 编程技巧

- MFC类型题目 | 棋类游戏



# 编程技巧

- MFC类型题目 | 管理系统
- 管理系统与棋类游戏类似
- 不同之处在于，不存在渲染绘制部分，但是控件事件更多更繁杂



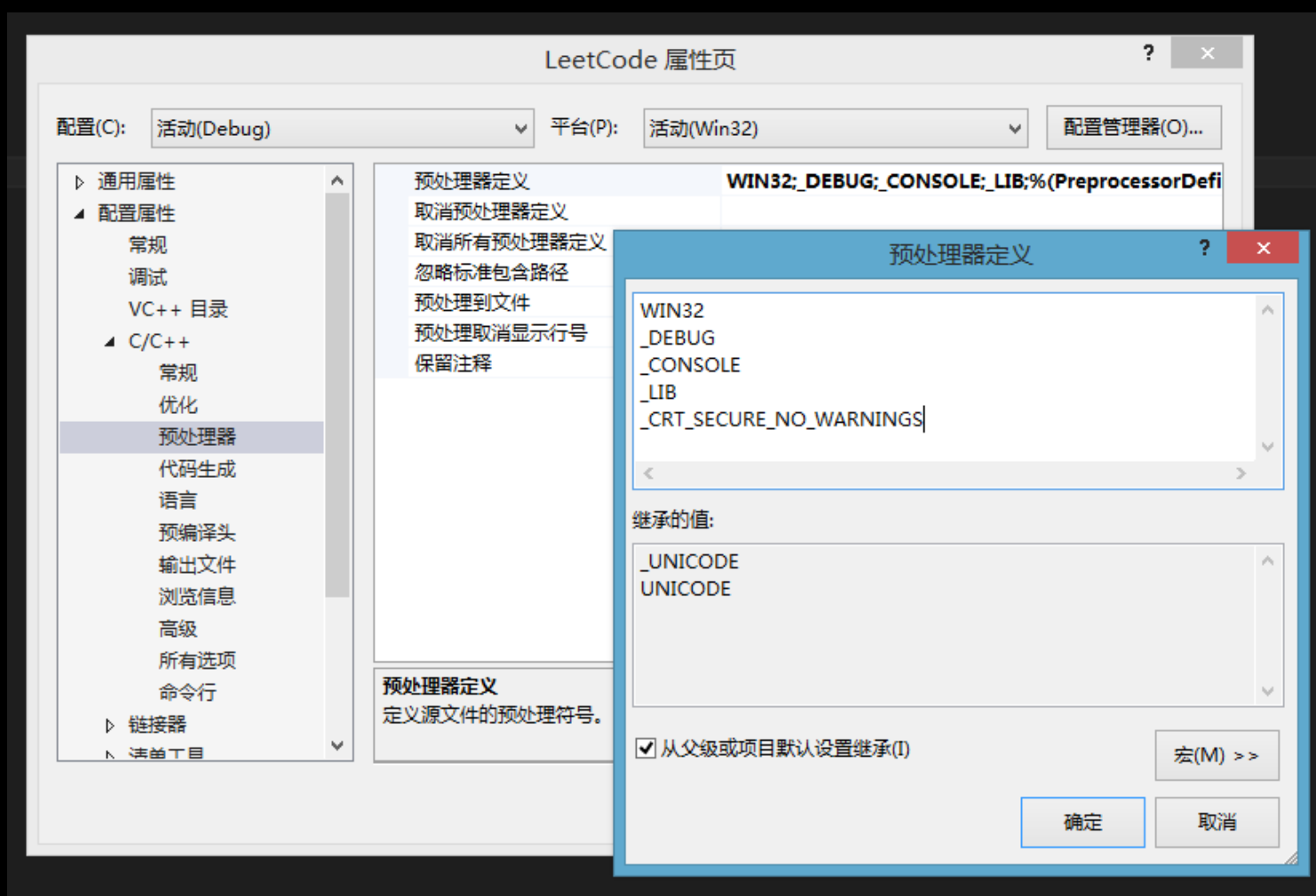
# 编程技巧

- Visual Studio调试技巧
  - 关闭CRT SECURE警告
  - 条件断点
  - Watch 数组



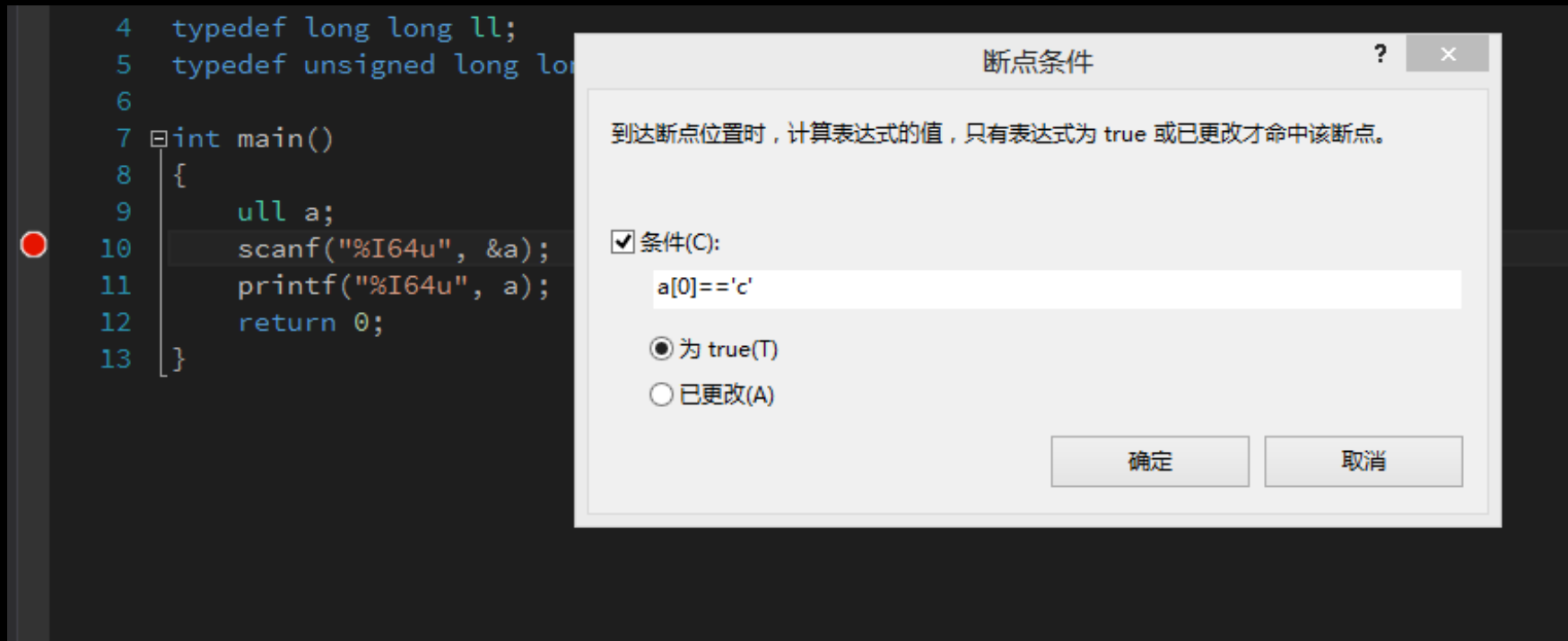
# 编程技巧

- Visual Studio调试技巧
- 关闭CRT SECURE警告



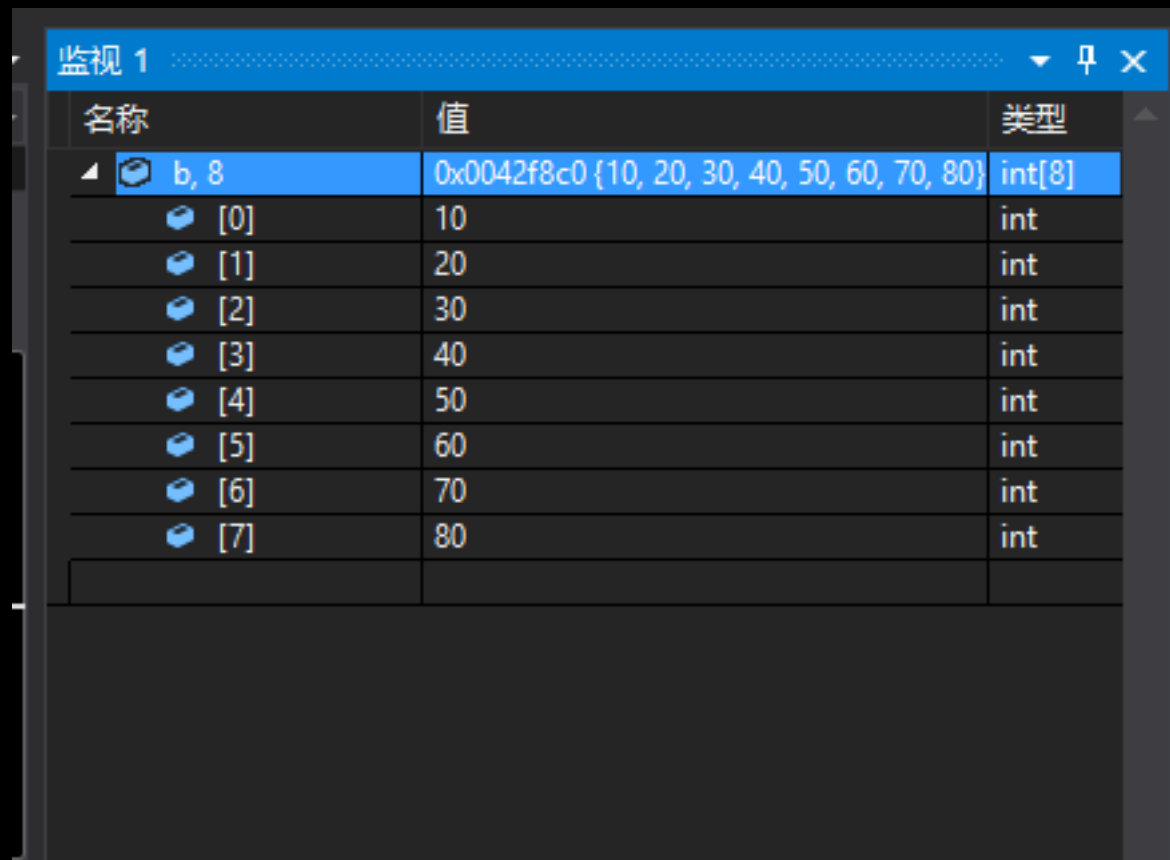
# 编程技巧

- Visual Studio调试技巧
- 条件断点



# 编程技巧

- Visual Studio调试技巧
  - Watch数组



The screenshot shows the 'Watch' window in Visual Studio, titled '监视 1'. It displays a table with three columns: '名称' (Name), '值' (Value), and '类型' (Type). The first row is expanded, showing an array named 'b' of type 'int[8]'. The array contains eight elements, indexed from [0] to [7], with values ranging from 10 to 80 in increments of 10. The memory address '0x0042f8c0' is also shown.

名称	值	类型
b, 8	0x0042f8c0 {10, 20, 30, 40, 50, 60, 70, 80}	int[8]
[0]	10	int
[1]	20	int
[2]	30	int
[3]	40	int
[4]	50	int
[5]	60	int
[6]	70	int
[7]	80	int

# 备考计划

- 实践比听咱们的讲座更有意义
- 把样题刷一遍
- MFC相关API是遗忘重灾区，多练是关键
- 很难一遍AC，提高debug能力