**程序名称：24点小游戏**

**开发环境：VC6.0**

**开发语言：C++**

**开发人：周子聪（914106840745）**

声明：本程序除自动生成代码外，所有函数，界面，图片均由属原创。

由于ios工程与MFC工程还是有一定的相似度，并且开始写之前花了大量时间准备，因此所遇大的问题比较少，常见问题就是不知道如何调用一些控件的函数，参考百度都能解决。

值得一提的问题有：

问题1：start按钮的消息响应函数（也就是随机获得四张牌的消息响应函数）不能在别处（四张牌无法计算得到24点时点击MessageBox中的确定按钮时，此轮得到或失去分数点击确定进行下一轮时）调用。

解决方法：将该消息响应函数作为该类的普通成员函数声明实现。（参考百度）

问题2：随机给四个按钮控件设置四张扑克牌的位图，已导入1~52张位图，但问题是这52张位图的ID值IDB\_BITMAP1~IDB\_BITMAP52并不是连续的，而是如下：

#define IDB\_BITMAP1 130

#define IDB\_BITMAP2 131

#define IDB\_BITMAP3 135

#define IDB\_BITMAP4 136

#define IDB\_BITMAP5 137

#define IDB\_BITMAP6 138

#define IDB\_BITMAP7 139

#define IDB\_BITMAP8 140

#define IDB\_BITMAP9 141

#define IDB\_BITMAP10 142

#define IDB\_BITMAP11 143

#define IDB\_BITMAP12 144

#define IDB\_BITMAP13 145

#define IDB\_BITMAP14 146

#define IDB\_BITMAP15 147

#define IDB\_BITMAP16 148

#define IDB\_BITMAP17 149

#define IDB\_BITMAP18 150

#define IDB\_BITMAP19 151

#define IDB\_BITMAP20 152

#define IDB\_BITMAP21 153

#define IDB\_BITMAP22 154

#define IDB\_BITMAP23 155

#define IDB\_BITMAP24 156

#define IDB\_BITMAP25 157

#define IDB\_BITMAP26 158

#define IDB\_BITMAP27 159

#define IDB\_BITMAP28 160

#define IDB\_BITMAP29 161

#define IDB\_BITMAP30 162

#define IDB\_BITMAP31 163

#define IDB\_BITMAP32 164

#define IDB\_BITMAP33 165

#define IDB\_BITMAP34 166

#define IDB\_BITMAP35 167

#define IDB\_BITMAP36 168

#define IDB\_BITMAP37 169

#define IDB\_BITMAP38 170

#define IDB\_BITMAP39 172

#define IDB\_BITMAP40 173

#define IDB\_BITMAP41 174

#define IDB\_BITMAP42 175

#define IDB\_BITMAP43 176

#define IDB\_BITMAP44 177

#define IDB\_BITMAP45 178

#define IDB\_BITMAP46 179

#define IDB\_BITMAP47 180

#define IDB\_BITMAP48 181

#define IDB\_BITMAP53 186

#define IDB\_BITMAP54 187

#define IDB\_BITMAP55 191

#define IDB\_BITMAP49 193

#define IDB\_BITMAP50 194

#define IDB\_BITMAP51 195

#define IDB\_BITMAP52 196

可见其中不少ID值跳跃了，开始想着手动更改这些ID值，便百度是否可行，结果有人这么做失败了，并且大家都说MFC中对具体的ID值进行操作是很蠢的。

想了很久才决定采用比较麻烦但不蠢的方法，定义一个int数组idb\_bitmap[52]重新给位图编号，如下：

idb\_bitmap[0]=IDB\_BITMAP1;

idb\_bitmap[1]=IDB\_BITMAP2;

idb\_bitmap[2]=IDB\_BITMAP3;

idb\_bitmap[3]=IDB\_BITMAP4;

idb\_bitmap[4]=IDB\_BITMAP5;

idb\_bitmap[5]=IDB\_BITMAP6;

idb\_bitmap[6]=IDB\_BITMAP7;

idb\_bitmap[7]=IDB\_BITMAP8;

idb\_bitmap[8]=IDB\_BITMAP9;

idb\_bitmap[9]=IDB\_BITMAP10;

idb\_bitmap[10]=IDB\_BITMAP11;

idb\_bitmap[11]=IDB\_BITMAP12;

idb\_bitmap[12]=IDB\_BITMAP13;//梅花1~K

idb\_bitmap[13]=IDB\_BITMAP14;

idb\_bitmap[14]=IDB\_BITMAP15;

idb\_bitmap[15]=IDB\_BITMAP16;

idb\_bitmap[16]=IDB\_BITMAP17;

idb\_bitmap[17]=IDB\_BITMAP18;

idb\_bitmap[18]=IDB\_BITMAP19;

idb\_bitmap[19]=IDB\_BITMAP20;

idb\_bitmap[20]=IDB\_BITMAP21;

idb\_bitmap[21]=IDB\_BITMAP22;

idb\_bitmap[22]=IDB\_BITMAP23;

idb\_bitmap[23]=IDB\_BITMAP24;

idb\_bitmap[24]=IDB\_BITMAP25;

idb\_bitmap[25]=IDB\_BITMAP26;//方块1~K

idb\_bitmap[26]=IDB\_BITMAP27;

idb\_bitmap[27]=IDB\_BITMAP28;

idb\_bitmap[28]=IDB\_BITMAP29;

idb\_bitmap[29]=IDB\_BITMAP30;

idb\_bitmap[30]=IDB\_BITMAP31;

idb\_bitmap[31]=IDB\_BITMAP32;

idb\_bitmap[32]=IDB\_BITMAP33;

idb\_bitmap[33]=IDB\_BITMAP34;

idb\_bitmap[34]=IDB\_BITMAP35;

idb\_bitmap[35]=IDB\_BITMAP36;

idb\_bitmap[36]=IDB\_BITMAP37;

idb\_bitmap[37]=IDB\_BITMAP38;

idb\_bitmap[38]=IDB\_BITMAP39;//红桃1~K

idb\_bitmap[39]=IDB\_BITMAP40;

idb\_bitmap[40]=IDB\_BITMAP41;

idb\_bitmap[41]=IDB\_BITMAP42;

idb\_bitmap[42]=IDB\_BITMAP43;

idb\_bitmap[43]=IDB\_BITMAP44;

idb\_bitmap[44]=IDB\_BITMAP45;

idb\_bitmap[45]=IDB\_BITMAP46;

idb\_bitmap[46]=IDB\_BITMAP47;

idb\_bitmap[47]=IDB\_BITMAP48;

idb\_bitmap[48]=IDB\_BITMAP49;

idb\_bitmap[49]=IDB\_BITMAP50;

idb\_bitmap[50]=IDB\_BITMAP51;

idb\_bitmap[51]=IDB\_BITMAP52;//黑桃1~K

然后采用如下方法随机设置按钮的位图

srand((unsigned)time(NULL));

int index0=rand()%52;

int index1=rand()%52;

int index2=rand()%52;

int index3=rand()%52;

m\_card1.SetBitmap(LoadBitmap(AfxGetInstanceHandle(),MAKEINTRESOURCE(idb\_bitmap[index0])));

m\_card2.SetBitmap(LoadBitmap(AfxGetInstanceHandle(),MAKEINTRESOURCE(idb\_bitmap[index1])));

m\_card3.SetBitmap(LoadBitmap(AfxGetInstanceHandle(),MAKEINTRESOURCE(idb\_bitmap[index2])));

m\_card4.SetBitmap(LoadBitmap(AfxGetInstanceHandle(),MAKEINTRESOURCE(idb\_bitmap[index3])));

心得与体会：由于该程序设计是有ios app改过来的，所以部分函数操作都类似，编写代码时间大概4，5小时，但是除上课老师讲的知识外自己也花了许多时间了解MFC的基本概念。

通过此次MFC 24点小程序的修改加深了对c++的认识，将许多之间有所遗忘的知识点捡起来并且拓展了不少知识，比如命名空间，消息的发送接收，程序的编译过程（顺序等）。

此外说明一下，原先我写过ios版本的24点游戏，相比之下这个MFC版本少了游戏计时，发牌等动画，防止明显错误输入这些东西。自然我知道MFC可以写的更好，只是时间不多，要复习电路，希望老师能谅解。

周子聪 914106840745