目录

[1.基本功能描述 1](#_Toc328517437)

[2.设计思路 2](#_Toc328517438)

[3.软件设计 5](#_Toc328517439)

[3.1设计步骤 5](#_Toc328517440)

[3.2界面设计 10](#_Toc328517441)

[3.3关键功能的实现 12](#_Toc328517442)

[4.结论与心得体会 13](#_Toc328517443)

[5.参考文献 15](#_Toc328517444)

[6.思考题 15](#_Toc328517445)

[7.附录 16](#_Toc328517446)

[7.1调试报告 16](#_Toc328517447)

[7.2测试结果 17](#_Toc328517448)

[7.3关键源代码 24](#_Toc328517449)

幸运52

# 1.基本功能描述

1）首先进入程序界面，除了查看“最高纪录”按钮可用以外，其他按钮均为不可

用，要使用程序则必须输入自己的姓名，然后按“提交”按钮，若姓名为空就提交，则此次提交无效，弹出对话框“请输入您的姓名”，当输入姓名正确以后，此时游戏设置区域的两个按钮“练习一下”，“进入现场”变为可用。否则所有按钮始终不可用。

2）“练习一下”与“进入现场”按钮的区别。这两个按钮是互斥的，当点击“练习一下”按钮后，“进入现场”按钮会立即失效；同理，点击“进入现场”则“练习一下”按钮立即失效。也就是说，每一次游戏只允许用户使用一种模式，而不允许中途更换模式。

当使用“练习一下”时，可以有较多的选择，“上一件商品”，“下一件商品”，“重新开始”，“确定”，“清除”，“开始竞猜”按钮均变为可用。此时用户可以点击“开始竞猜”按钮进行答题，同时进度条开始计时。程序设定时间为60秒，必须在60秒的时间内，对当前的商品输入正确的值，否则，超过时间程序会自动退出。若不知当前商品的价格，可以选择“上一件”，“下一件”来更换商品进行竞猜。同时会显示当前商品的编号和名称，以供用户参考。

当使用“进入现场”时，此时对用户会有更严的限制。此时只有“开始竞猜”，“确定”，“清除”按钮可用，其他按钮变为不可用。当按下“开始答题”按钮时，开始计时，进度条规定时间仍然为60秒。此时用户只有答对题目按下“确定”按钮才可以进入下一题。并且，和练习模式一样，每一件商品最多只允许6次猜测。次数大于6次或者时间超过60秒，程序都会自动退出。

3）“最高纪录”按钮，每一次按下都会弹出提示对话框，显示程序到目前为止，最高纪录的保持者的姓名和答对题目的个数。

4）“确定”按钮，是功能较多的按钮，当估价框为空的时候，如果按“确定”按钮，此时会提示用户“请您输入估价”但是为空的提交，程序并不将其当作一次竞猜，所以空提交，不影响真实竞猜的次数。

每一次“确定”程序都会将估价与商品的真实价格相比较，如果价格过高，则弹出对话框提示“您输入的价格太高了”，价格过低则提示“您输入的估价太低了”，如果估价正确则提示“恭喜您，答对了！”同时会进入下一件商品的竞猜，当前商品的竞猜次数会归零。

同时每按一次“确定”按钮，程序会将当前用户的记录与历史记录的“最高记录”进行一次比较，如果超过了历史最好成绩则将当前成绩进行存档。每一次确定都会实时进行存档，始终保持存入最高纪录。即使时间没到用户就退出程序，也会将最高纪录进行刷新。

# 2.设计思路

程序设计的思路按照程序要实现的功能来进行。

首先利用MFC类向导制作对话框的界面，按照自己预先规划的样式进行界面的设计。例如，按钮的个数，位置；编辑框的个数，位置，只读属性；静态提示的文字位置等等。其次，给每一个控件修改ID以便于自己的识别，控件的属性要按自己的思路修改。做好界面以后，开始按照程序流程图来编写程序。一是要，规划需要哪些功能的函数，每一个函数如何编写，实现该功能的函数是否为MFC类库中的函数；二是要，分清楚函数之间的调用关系，以及函数的带参数情况。三是要，区分哪些是基本功能要优先实现，哪些是附加功能，在基本功能之后实现。

具体设计要根据自己所画的流程图来进行，流程图要尽可能的详细，这样在实施起来就会清晰而方便。由流程图先写出主干的代码，实现功能需要什么样的函数，变量。变量的类型以及函数具体如何实现要事先根据功能查阅资料，然后动手写代码。每一次只为程序添加一种功能进去，这样代码会少很多，出现错误更容易查找。实现一种功能以后在依法炮制，添加下一个功能。直至所有预先要求的功能都加完为止。

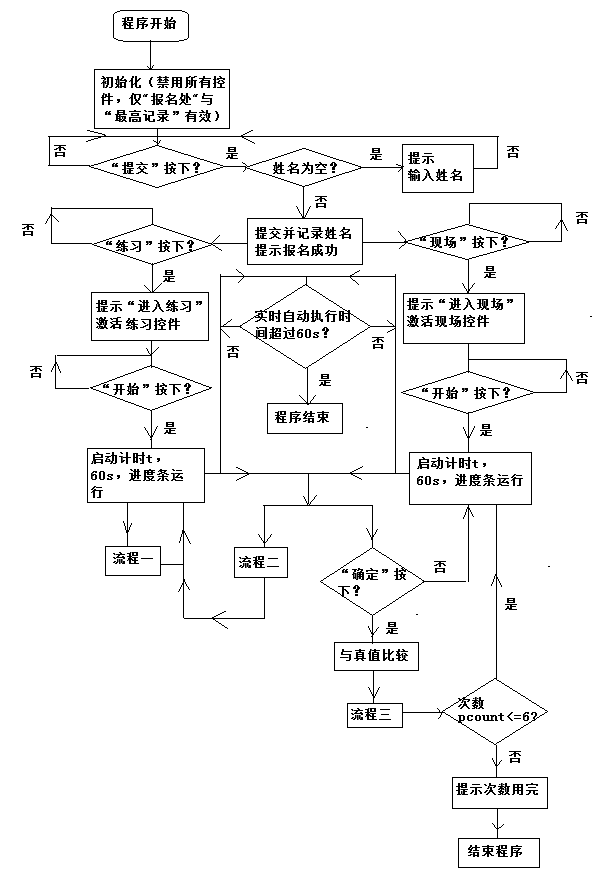


图1程序总体结构流程图

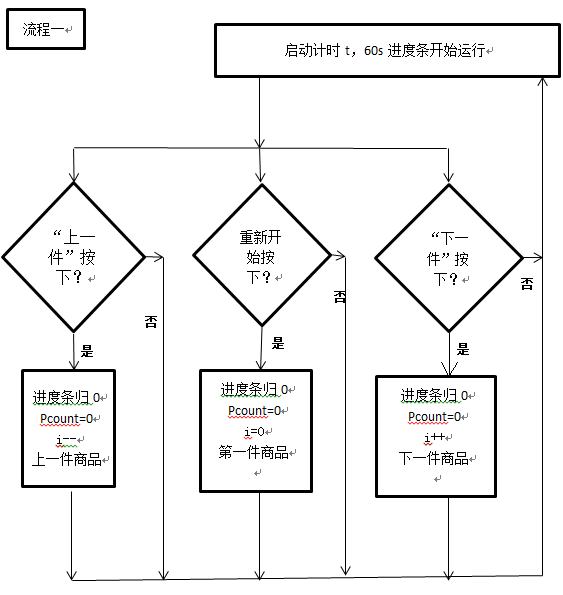


图2流程一的结构

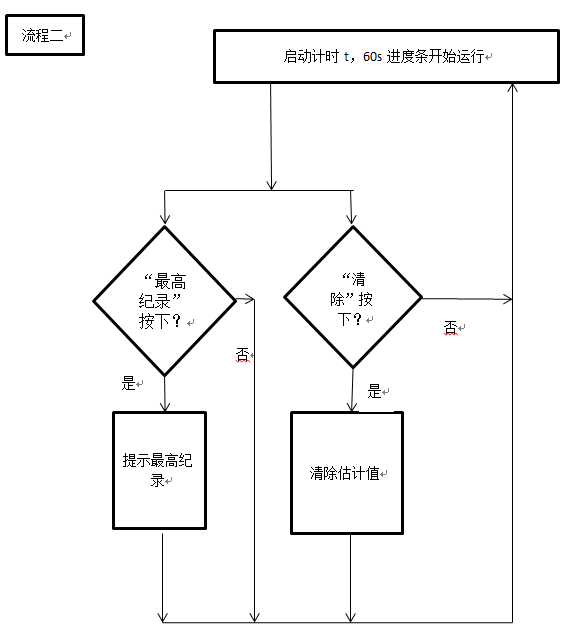


图3流程二的结构

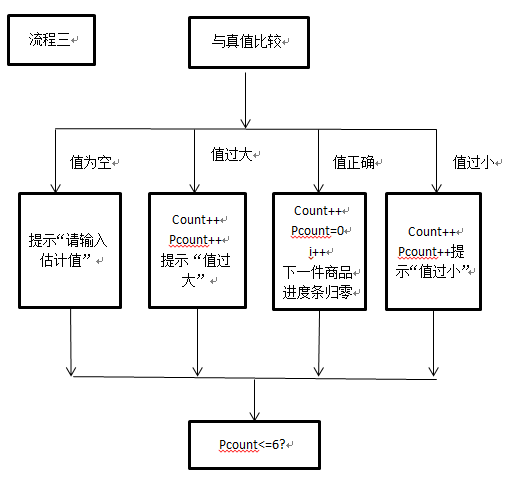


图4流程三的结构

# 3.软件设计

## 3.1设计步骤

1）在File菜单下选择NEW，创建MFC AppWizard(exe)工程，键入的工程名汪少锋，在应用程序类型中选择基本对话框，其他属性直接选用默认，单击确定完成对话框的创建。最后弹出新建工程信息对话框，该对话框列出了关于新建的应用程序项目文件的相关信息。单击确定，应用程序向导所有的工作就全部完成了，此时向导为用户生成了一个可执行的应用程序框架。



图5新建工程

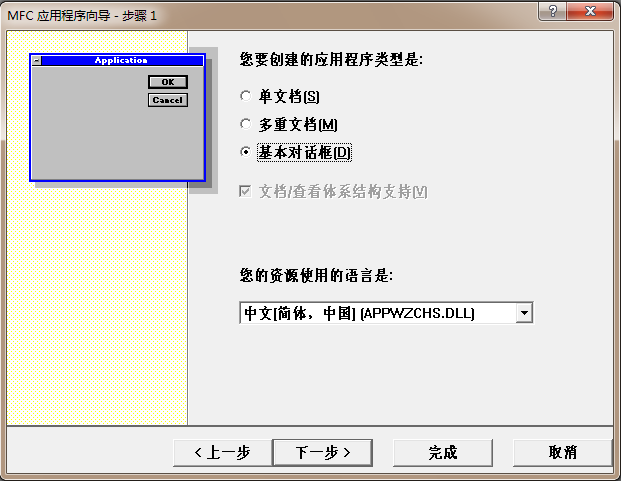


图6选择基于对话框，其他默认，按“完成”

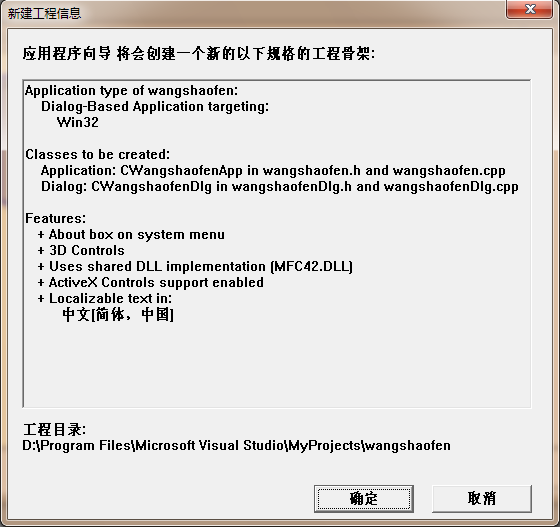


图7单击“确定”完成工程新建

2）初始化界面，在对话框中添加所需的控件如按钮，文本，编辑框，菜单，Picture控件，进度条；放置在相应的位置上。根据需要对其属性进行修改，添加对应的控件变量，并建立类向导，在相应的地方添加响应函数。

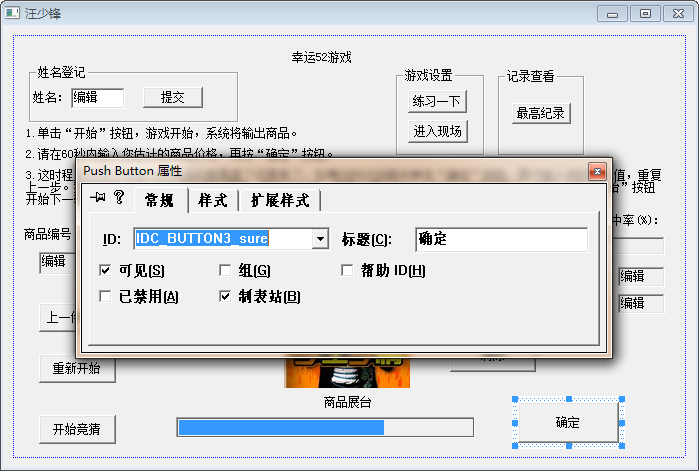


图8添加控件并设定属性

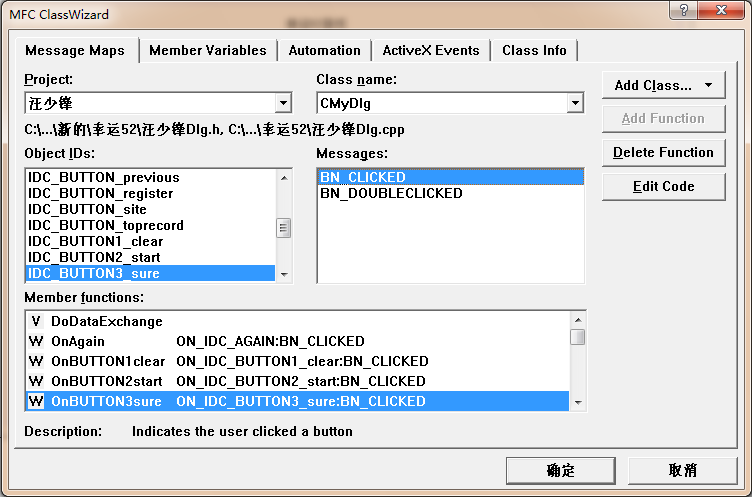


图9利用类向导添加消息的响应函数

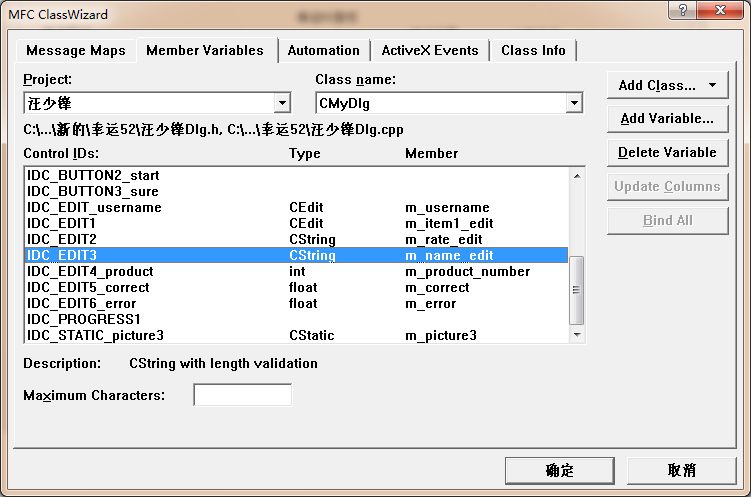


图10利用类向导添加成员变量

3）制作菜单并为相应的菜单项设定ID和变量以及响应函数，部分菜单具体功能与主界面相同，可以直接将ID设为与界面控件一致即可实现相同功能，不用另外添加代码。

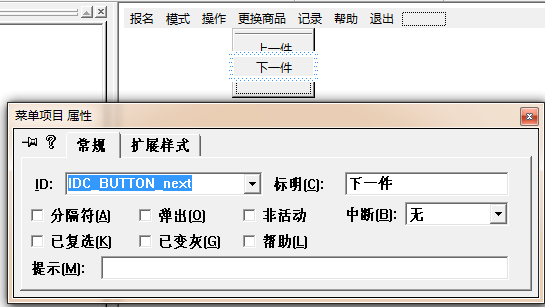


图11为菜单项设定ID

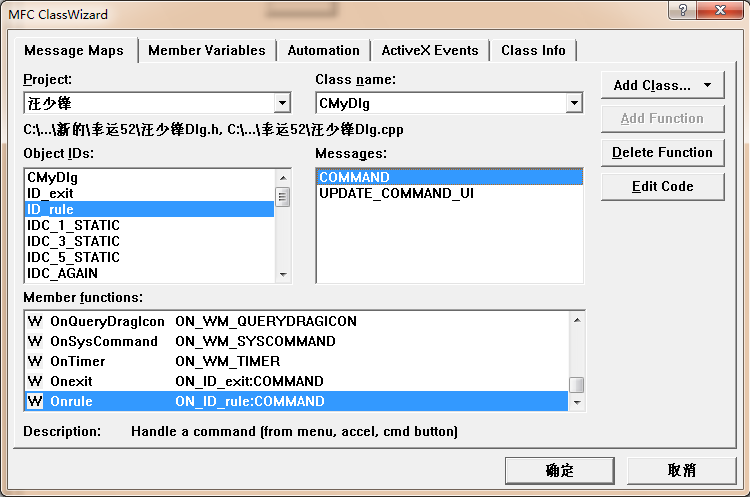


图12为菜单项添加响应函数

4）在实现文件.cpp中添加实现功能所需要的各种函数，并添加相应的代码，同时在头文件中添加对各种函数及变量的声明。

5）对完成后的程序进行调试，一定会遇到错误，仔细查找，修改其中的错误，直到程序运行不再报错，并能够实现所要求的各项功能。遇到不懂的报错可以在网上搜索相应的知识，可以帮助分析。

## 3.2界面设计

1）初始化界面如下所示：

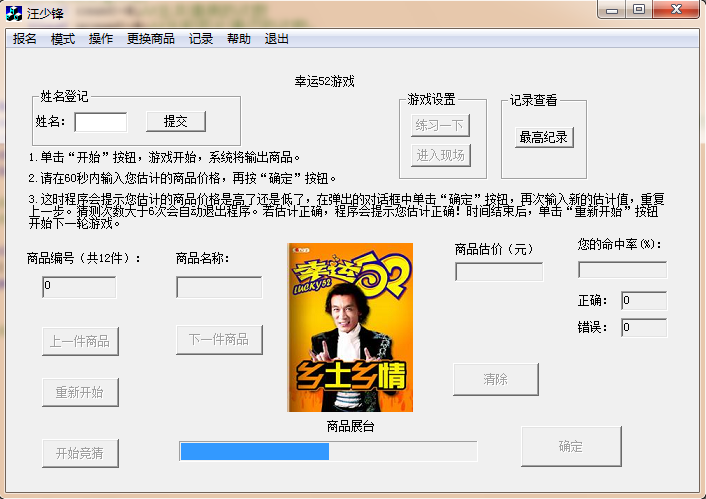


图13主界面初始显示图

2）主界面控件设置如下：

表1 主界面对话框控件表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控件名称 | 控件类型 | 属性(ID) | 对应变量类型 | 对应函数名 | 对应变量名 |
| 提交 | 按钮 | IDC\_BUTTON\_register |  | OnBUTTONregister（） |  |
| 练习一下 | 按钮 | IDC\_BUTTON\_practice |  | OnBUTTONpractice() |  |
| 进入现场 | 按钮 | IDC\_BUTTON\_site |  | OnBUTTONsite() |  |
| 最高纪录 | 按钮 | IDC\_BUTTON\_toprecord |  | OnBUTTONrecord() |  |
| 上一件商品 | 按钮 | IDC\_BUTTON\_previous |  | OnBUTTONprevious() |  |
| 下一件商品 | 按钮 | IDC\_BUTTON\_next |  | OnBUTTONnext() |  |
| 重新开始 | 按钮 | IDC\_AGAIN |  | Onagain() |  |

续表 表1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 开始竞猜 | 按钮 | IDC\_BUTTON2\_start |  | Onstart() |  |
| 清除 | 按钮 | IDC\_BUTTON\_clear |  | OnBUTTONclear() |  |
| 确定 | 按钮 | IDC\_BUTTON3\_sure |  | OnBUTTON3sure() |  |
| 正确次数 | 只读编辑框 | IDC\_EDIT5\_correct | float |  | m\_correct |
| 错误次数 | 只读编辑框 | IDC\_EDIT6\_error | float |  | m\_error |
| 命中率 | 只读编辑框 | IDC\_EDIT2 | CString |  | m\_rate\_edit |
| 商品展台 | PICTURE | IDC\_STATIC\_picture3 | 位图 |  |  |
| 耗时60秒显示 | 进度条（平滑） | IDC\_PROGRESS1 |  |  |  |
| 商品名称 | 只读编辑框 | IDC\_EDIT3 | CString |  | m\_name\_edit |
| 商品编号 | 只读编辑框 | IDC\_EDIT4\_product | int |  | m\_product\_number |
| 商品估价 | 编辑框 | IDC\_EDIT1 | CEdit |  | m\_item1\_edit |
| 姓名 | 编辑框 | IDC\_EDIT\_username | CEdit |  | m\_username |

表2 主界面菜单项列表

|  |  |  |
| --- | --- | --- |
| 菜单名称 | 菜单ID | 菜单函数名称 |
| 报名 | IDC\_BUTTON\_register | OnBUTTONregister（） |
| 上一件商品 | IDC\_BUTTON\_previous | OnBUTTONprevious() |
| 下一件商品 | IDC\_BUTTON\_next | OnBUTTONnext() |
| 重新开始 | IDC\_AGAIN | Onagain() |
| 开始竞猜 | IDC\_BUTTON2\_start | Onstart() |
| 确定 | IDC\_BUTTON3\_sure | OnBUTTON3sure() |
| 清除 | IDC\_BUTTON\_clear | OnBUTTONclear() |
| 练习一下 | IDC\_BUTTON\_practice | OnBUTTONpractice() |
| 进入现场 | IDC\_BUTTON\_site | OnBUTTONsite() |
| 最高纪录 | IDC\_BUTTON\_toprecord | OnBUTTONrecord() |
| 帮助 | ID\_rule | Onrule() |
| 退出 | ID\_exit | Onexit() |

3）程序的菜单制作

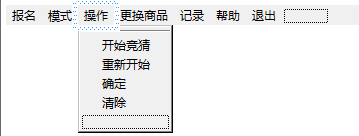


图14菜单制作

## 3.3关键功能的实现

1）输入焦点的控制：当开始运行程序时，设置焦点使得光标在姓名编辑框内，在Onpaint（）函数中利用语句GetDlgItem(IDC\_EDIT\_username)->SetFocus()实现此功能。按下“开始竞猜”按钮后，应该设置焦点使得光标停留在在商品估价输入框内，在Onstart（）函数中采用 GetDlgItem(IDC\_EDIT1)->SetFocus()实现。这样每一次按下“确定”按钮都会将焦点设定在估价输入编辑框内。

2）设计初始界面：首先生成一个基于对话框的基本框架，再添加控件例如按钮，编辑框，位图，进度条等，并设置属性，然后为相关控件连接变量，为功能控件添加函数。

3）随机显示某一种商品：首先引入12张位图资源；然后在对话框类的声明文件中定义2个数组，分别存储商品的名称、价格；接着在对话框类的构造函数中为名称与价格数组初始化，并采用LoadImage( )实现位图的加载来完成图片数组的初始化，并使同一商品在2个数组中对应的下标相同；程序初始时选中第一件商品，以后每一次猜中则更换为下一件商品，当猜完12件商品以后，会返回第一件商品，然后在进行猜测。

4）计时：游戏开始时调用SetTimer( ) 函数启动计时器并主动调用OnTimer(UINT nIDEvent) 函数对进度条进行设置，其中SetTime（）函数第一个参数表示计时器的ID号，第二个参数表示时间步长，以毫秒为单位，本程序设定1000ms为步长，每隔步长时间就发送一个WM\_TIME消息；收到此消息，程序会自动执行OnTime（）函数，每执行一次此函数就会让进度条前进一步；设定进度条共有60步，则程序对每一件商品竞猜所规定的时间为60s；超过60s时，程序会自动结束。

5）对估价的验算与正确次数、错误次数的统计：取用户输入估价与真实价格进行比较，若估价与真实价格不符弹出提示消息框“值过大”或“值过小”，并使总共所猜次数count加1，当前商品所猜次数pcount加1，总共猜错次数errocount加1；再比较pcount与6的大小，pcount≥6即估价次数已达到六次便自动退出程序；若估价与真实价格相等，则提示竞猜正确，并使总共所猜次数count加1，将商品信息更换为下一件商品的信息，同时把当前商品所猜次数pcount清零。

6）对历史最高纪录的处理：历史最高纪录不单单只是程序本次运行的最高纪录，应该是程序以往运行与本次运行共同的最高纪录，若在程序中设定两个变量用来存储最多猜中次数与姓名是实现不了此功能的。为此，在工程文件夹中创建两个文本文档分别为username.txt与correct.txt，分别存储最高纪录的保持者的姓名和所猜对商品的个数。每轮游戏开始时读取username.txt与correct.txt中内容并且存入record（字串存储姓名）与cs（数值型存储次数）两个变量当中，与本次游戏者的记录Tempname与（count-pcount）做比较，其中count-pcount表示总共猜对的商品的个数。若（count-pcount）>cs则表示当前记录超过以往的记录，那么此时就会将当前游戏者的名字Tempname和次数（count-pcount）记录进入文本文件当中，替换原有记录成为最高纪录的保持者。

优点：每一次确定按钮被按下，都会将当前游戏者的记录与原始记录进行比较，决定是否替换，这样适时地保存记录是有好处的。如果发生突发状况，比如断电，人为退出，记录都会被记录下来而不会发生丢失。

7）查看最高纪录：当用户查看最高纪录时，分别从username.txt和correct.txt中读取最高纪录与姓名，并在消息框中显示。

8）重新开始游戏：当用户重新开始游戏时，此时OnAGAIN（）函数内部会将商品下标i清零，同时调用Onstart（）函数，会将当前商品换为第一张，进度条从零开始。可以重新进行游戏。

# 4.结论与心得体会

这次课程设计总体上对我来说并不那么简单，首先是距离学习C++实在是过的时间太久了，很多知识都忘记的差不多了，并且期间穿插有必修课的考试。一边准备考试，一边准备课设确实让课设在时间上打了折扣。考试结束后便全力投入到课设当中了，最终也做出了幸运52这个小游戏。

但是整个过程中的实际操作确实是对我的一种考验。之前学习过C++语言，对MFC编程的印象还是比较深的，但是经过课设以后，无疑更加加深了印象。让我再一次体会到了VC++的强大与复杂，感觉到自己学习C++就好像在大海里挣扎，这门语言真的好复杂。系统提供的各种类，各种函数真是多如牛毛。尽管我只用了很少的一些函数，但是这几天在网上搜索资料，对C++有了进一步的了解。最强的感觉就是，自己真的好菜，好水。不得已，只得要用什么功能就在网上搜索对应的资料。这样才感觉有了一点眉目，要不然真的是只能望洋兴叹了。

之所以选择幸运52来做，主要原因可能就是觉得比较这个题目比较新颖。最开始也不知道幸运52是干嘛的。于是到网上搜索，才对游戏有了了解。在动手做程序之前，首先就是看书了，花了一部分时间将以前学习C++的教材看了一下，主要是看MFC编程部分，将最基本的控件使用方法看完，比如编辑框，按钮，picture控件，菜单。其他的控件因为时间关系，并且用到的可能性不大，便忽略了。但是这些知识都是非常有用的，有机会的话还是要多学一学，毕竟在我看来VC++非常强大。

做MFC程序，首先就是做界面了，主体是一个对话框。我是先在纸上画出了界面的布局，然后照着预先的布局做的界面，添加各种各样的控件，每添加一个控件我就会在布局的纸上相应控件的位置上标上自己更改以后控件的ID号，这样既便于认识出控件是代表哪个，又不至于以后弄混淆了无法差错，同样的方法，修改了控件的属性，对应的函数以及变量名，目的就是为了自己能更好地识别每一个控件，防止自己犯低级错误。当然，之所以这样做，也是由于以前犯过弄混淆ID和变量对应不好的低级错误。

我先规划做出最基本的功能，然后再一步一步添加其他的功能。在我看来，难点一就是位图的动态加载，因为以前自己没有动手做过这个。于是到网上去找资料，发现了LoadImage（）函数用来加载位图，于是仔细研究一番就用了这个函数。难点二在于最高纪录的存取，因为有C语言的基础，而VC6.0可以编译C于是文件的记录writedata（）函数用C实现，由于用C写得读数据函数readdata（）函数无效，于是读函数用C++实现。

难点三是控制焦点，因为以前没听说过，但是经过学习，发现这个其实实现比较简单。

最后我实现了最基本的功能以后，又添加了几个按钮来实现其他一些功能，比方说设定两种游戏模式，将不需要输入值的编辑框设定为只读，将只需要输入一次的编辑框在输入之后让其变为只读。让只需要按下一次的按钮在按下后立即失效变为灰色，另外，为主面板添加了12个菜单项。如此一来，程序就更丰富了；其实之前是想把所有人的游戏记录都记录在一个文本文件中的，包括姓名和次数；也尝试着来编程实现，但是后来运行程序太复杂，并且错误难找，并且时间已经不够了。于是果断放弃尝试，改用更简单的方式，只记录一个人的数据，并且将姓名和次数分开存放。

不管怎样还是做出东西来了，尽管没什么含金量，但至少全是自己的心血。课设这几天感觉比上课累多了，也更有挑战一些。先不说别的，对C++的了解更多一些了，对MFC的控件添加流程也更了解了，对于用到的函数也更了解了。最最重要的是自己对于常见的一些调试错误能更好地找出问题了。总之，收获还是很多的。

# 5.参考文献

[1]揣锦华.面向对象程序设计与VC++实践[M].西安：西安电子科技大学出版社,2006.

[2]马德骏，等.C语言程序设计（第2版）[M].北京：科学出版社，2009.

[3]方可燕.Visual C++ 6.0实战与精通[M].北京：清华大学出版社，2000.

# 6.思考题

1） 说明保存记录文件信息的方法，考虑是否还能用其他方法实现？

答：在工程文件夹中创建两个文本文档分别为username.txt与correct.txt，分别存储最高纪录的姓名与猜对次数。每轮游戏计时开始时读取username.txt与correct.txt中内容与本轮正确次数(count-errocount)比较，若大于其中内容，便将当前游戏者的姓名与猜对次数替代原有的username.txt与correct.txt中的内容。这种比较与替换函数在“确定”按钮每按下一次就进行一次。可以适时地记录最高纪录，即使时间没到或者突然断电同样可以将当前的最高纪录保持下来，不论玩家是否自愿退出游戏都没有关系。

2） 说明商品名称和商品图片、商品价格一一对应的关键部分？

答：在对话框类的声明文件中定义2个数组，分别存储商品的名称和价格，接着在引入资源时候按照商品顺序引入位图资源，并且逐个命名位图ID号。利用switch-case结构调用LoadImage（）函数来逐个调用位图，使得数组的下标与case语句中的位图一一对应即可。对话框类的构造函数中为商品名称与价格数组初始化，并采用LoadImage()实现位图的加载来完成图片数组的初始化，并使同一商品在2个数组中对应的下标相同。这样，便将商品名称和商品图片、商品价格一一对应起来。

3） 如何在游戏未开始时显示空白位图？

答：在位图资源中插入一张空白位图，再将“图像”选为该空白位图的属性ID，初始化函数中使用LoadImage（）函数调用该空白位图。

# 7.附录

## 7.1调试报告

1）丢失分号的语法错误：例如error C2143: syntax error : missing ';' before '}'在错误出现的地方查看分号丢失的位置并添加上。

2）未定义变量的错误：例如：error C2065: 'h' : undeclared identifier。在调试的过程中会出现某些变量未定义，导致这种错误有可能是在对控件建立类向导的时候没有添加变量，也有可能是在添加代码的时候变量名输错导致的；对变量建立类向导的时候添加变量或者改正错误的变量名就能改正这种错误。

3）未添加消息响应函数的逻辑错误：在类的源文件中添加代码之后，运行的过程中代码并没有被应用，错误原因是没有添加相应的消息函数；在类向导中添加相应的消息函数之后，代码就是可用的了。

4）头文件错误：例如：fatal error C1010: unexpected end of file while looking for precompiled header directive添加了打开文件的相关代码后，编译显示相关语句没有定义不能识别：这是由于忘了加头文件；加上头文件后编译成功。

5）开始下一个商品的猜价的时候“当前错误次数”没有清0：估价正确后没有给当前猜测次数赋值为0；在估价正确时添加pcount=0;语句后正确。

6）点击“重新开始”按钮，不能重新游戏： “重新开始”按钮并没有对进度条的初值进行设置，时间结束后再点击“重新开始”计时仍为达到最大值，对此在它的响应函数里面对进度条设置初值为0，并且将当前商品更换为第一件商品，当前猜测次数pcount清零，再调用CLiudanDlg::OnBegin()函数，便可重新开始游戏。

7）在第一次编译程序时要首先编译StdAfx.cpp文件，将头文件包含进去，否则第一次编译就会报错，例如：Cannot open precompiled header file: 'Debug/汪少锋.pch': No such file or directory

## 7.2测试结果

1）游戏开始时必须输入游戏者的姓名并提交，否者无法进入游戏；若姓名为空则提示“请输入您的姓名”，直到姓名不为空方可进入游戏。游戏初始其他控件均被失效，仅有“提交”按钮和“最高纪录”按钮可用。提交姓名以后，报名处的控件失效。

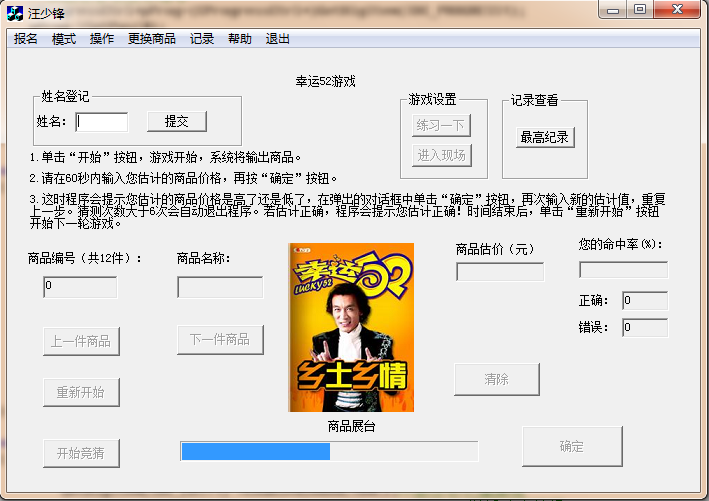


图15初始化主界面显示图

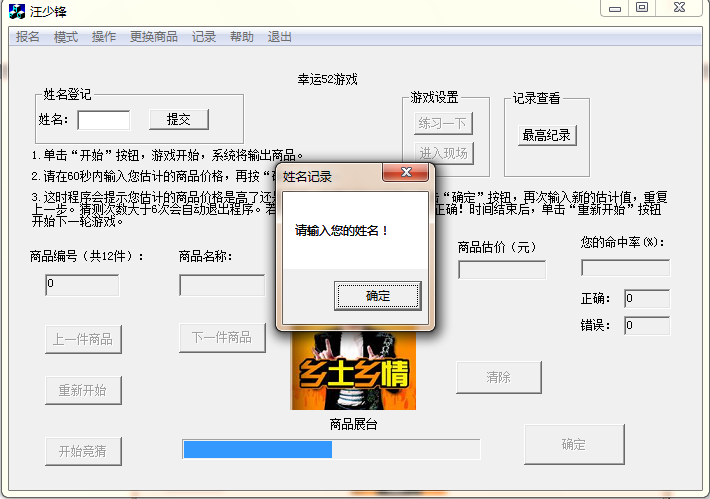


图16提交空姓名会提示

2）提交姓名以后，用户可以选择“练习一下”或者“进入现场”两种游戏模式。点击会给以提示并激活相应的按钮与编辑框，同时输出第一张商品的图片。例如进入“练习一下”模式。

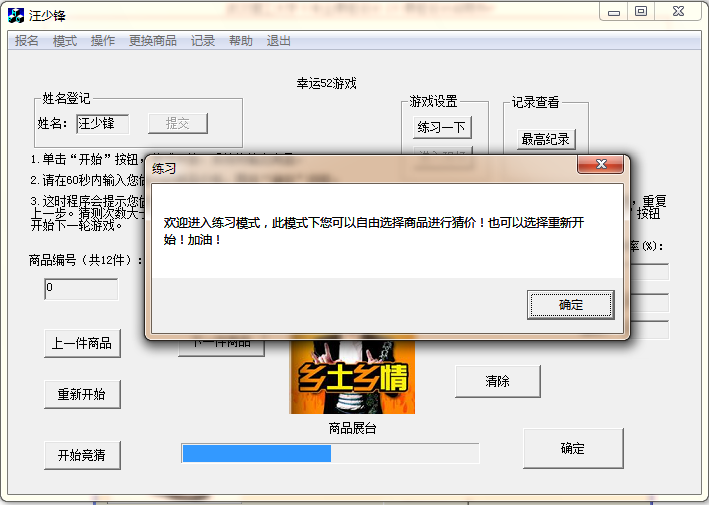


图17未输入商品估价显示图

3）进入练习模式可以更换当前的商品为“上一件”或“下一件”，也可以选择重新开始。按下“开始竞猜”按钮，光标聚焦在估价编辑框中，可以输入价格，若价格为空则给与提示，并不计数；不为空则对价格进行比较并给与提示。

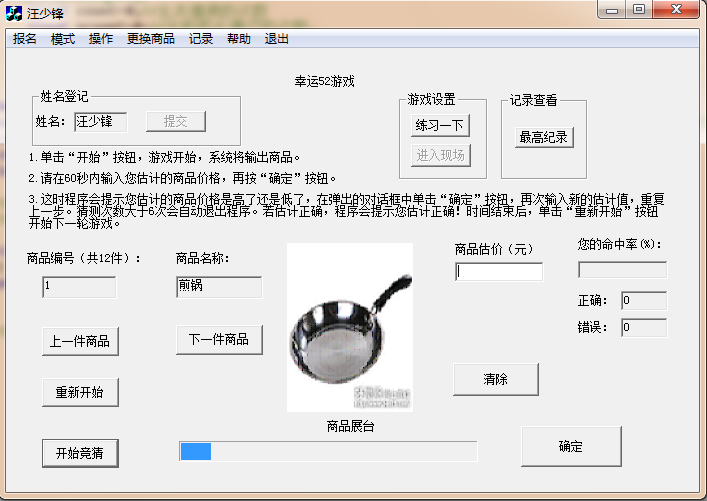


图18选择“开始竞猜”第一件商品

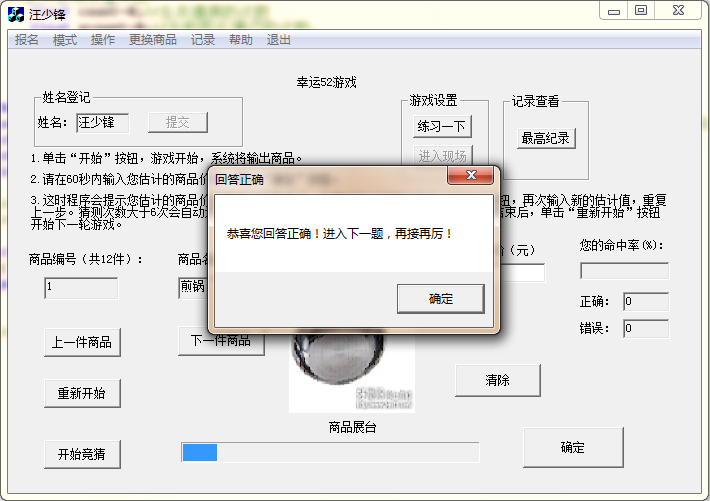


图19竞猜正确



图20选择下一件商品



图21选择上一件商品

4）输入估价，点击“确定”，如果估价偏大则提示“估计过大”，估价过小则提示估价过小。



图22 估价过高显示图

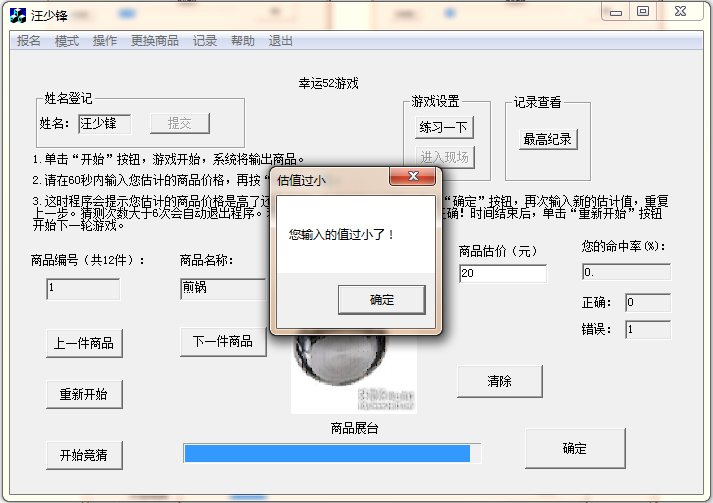


图23提示估价过低

5）当用户选择“现场模式”时，只有答对了题目才可以进入下一件商品的竞猜，并且不允许重新开始。

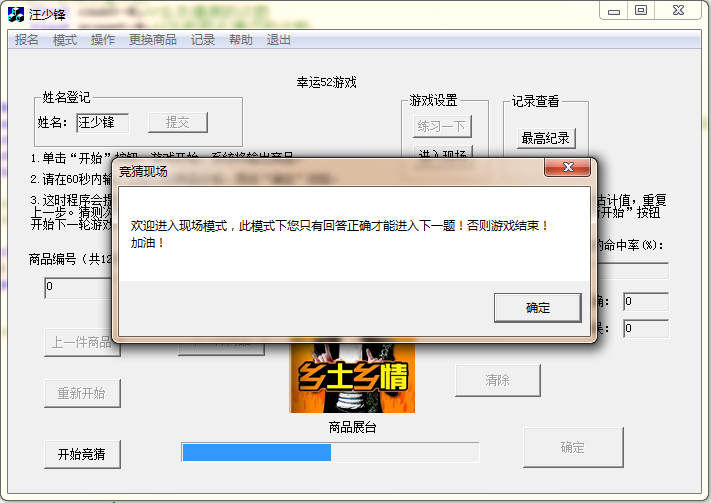


图24 进入现场模式

6）如果在60秒时间内没有猜对价格，则计时结束，自动弹出消息框，显示“时间到！竞猜结束！”。

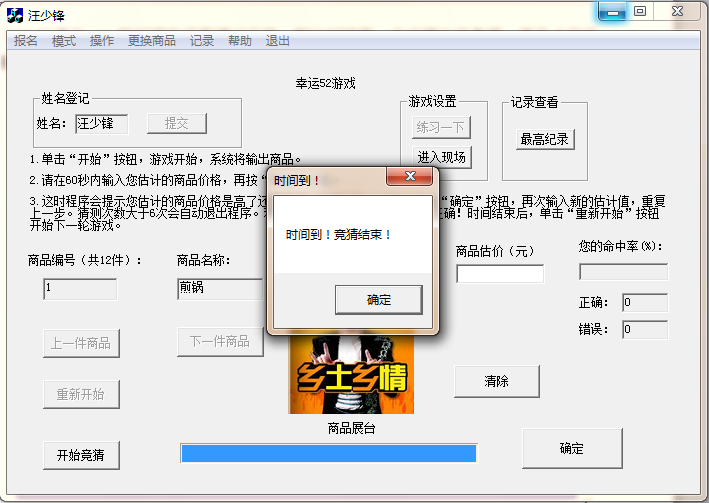


图25计时结束显示图

7）当估价次数超过6次时，自动弹出消息框，显示“对不起，您没有机会了，超过了次数限制！”，点击“确定”后，自动退出本程序。

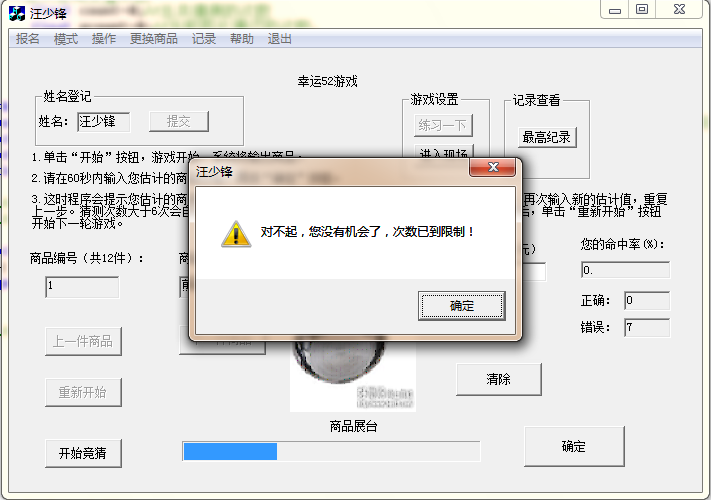


图26次数超过设定上限即6次

8）无论本次游戏是否打破历史纪录，都可以查看最高的历史记录，点击“最高纪录”即可看到最高纪录。



图27打破纪录后点击“最高记录”弹出的对话框显示图

## 7.3关键源代码

1）定义全局变量

//定义全局变量次数

float count=0;//总共猜测的次数

float pcount=0;//当前图片猜过的次数；

float errocount=0;//总共猜错的次数

float rate=0;//命中率

int i=0;//作为信息数组的下标

#define k 6//每一张图片可供估价的次数

unsigned long guesstime=60000,guessstep=1000;

//进度条60步，每一步为1秒，即设定time为60s

//步长为1秒，60000为计时器的ID

char Tempname[10];//用于保存当前用户名

char record[10];//记录中的用户名

int cs;//记录中的次数

2）初始化数组

CMyDlg::CMyDlg(CWnd\* pParent /\*=NULL\*/)

: CDialog(CMyDlg::IDD, pParent)

{

//给名称+价格+图片赋给初值

name[0]="煎锅";

name[1]="篮球";

name[2]="牛仔裤";

name[3]="诺基亚手机";

name[4]="女靴";

name[5]="扇子";

name[6]="水晶耳环";

name[7]="网球拍";

name[8]="铂金戒指";

name[9]="压力锅";

name[10]="音乐盒";

name[11]="数码相机";

price[0]=50;//每一张图片对应的价格

price[1]=128;

price[2]=280;

price[3]=2200;

price[4]=350;

price[5]=30;

price[6]=3800;

price[7]=180;

price[8]=11900;

price[9]=238;

price[10]=125;

price[11]=2050;

}

3）构造函数进行初始化

BOOL CMyDlg::OnInitDialog()

{

//添加进度条初始化代码

CProgressCtrl\*pProg=(CProgressCtrl\*)GetDlgItem(IDC\_PROGRESS1);

pProg->SetRange(0,60);

pProg->SetPos(30);

//禁用确定按钮和编辑框的代码

{ GetDlgItem(IDC\_EDIT1)->EnableWindow(FALSE);//禁用估价编辑框

GetDlgItem(IDC\_BUTTON3\_sure)->EnableWindow(FALSE);//禁用确定按钮

GetDlgItem(IDC\_BUTTON1\_clear)->EnableWindow(FALSE);//禁用清除按钮

GetDlgItem(IDC\_AGAIN)->EnableWindow(FALSE);//禁用重新开始按钮

GetDlgItem(IDC\_BUTTON\_next)->EnableWindow(FALSE);//禁用下一件商品按钮

GetDlgItem(IDC\_BUTTON\_previous)->EnableWindow(FALSE);//禁用上一件商品按钮

GetDlgItem(IDC\_BUTTON2\_start)->EnableWindow(FALSE);//禁用开始按钮

GetDlgItem(IDC\_BUTTON\_practice)->EnableWindow(FALSE);//禁用练习按钮

GetDlgItem(IDC\_BUTTON\_site)->EnableWindow(FALSE);//禁用现场按钮

}

//初始化要读取文本文件里的数据

readdata();

}

4）程序初始时聚焦到“姓名”编辑框

void CMyDlg::OnPaint()

{

GetDlgItem(IDC\_EDIT\_username)->SetFocus();//程序最开始的时候聚焦到姓名编辑框

}

5）对“开始”按钮的响应函数

void CMyDlg::OnBUTTON2start()

{

CProgressCtrl\*pProg=(CProgressCtrl\*)GetDlgItem(IDC\_PROGRESS1); //开始按钮使得进度条归零

//设置指向进度条的指针

pProg->SetPos(0);//进度条初值为零

SetTimer(guesstime,guessstep,NULL);//启动计时器，自动调用OnTimer（）函数

//初始化商品信息

m\_name\_edit=(LPCTSTR)name[1];

UpdateData(FALSE);//更新数据的显示

GetDlgItem(IDC\_EDIT1)->SetFocus();//开始按钮使得聚焦到估价编辑框

//让按钮和编辑框在“开始”按钮后变为可用

{GetDlgItem(IDC\_EDIT1)->EnableWindow(TRUE);//激活估价编辑框

GetDlgItem(IDC\_BUTTON3\_sure)->EnableWindow(TRUE);//激活确定按钮

GetDlgItem(IDC\_BUTTON1\_clear)->EnableWindow(TRUE);}//激活清除按钮

dspbmp(0);//初始化显示第一张图

dspname(0);//初始化显示第一个名字

}

6）显示商品名称和编号的函数

void CMyDlg::dspname(int i)//显示名称和商品编号

{

m\_name\_edit=(LPCTSTR)name[i];//将字符强制转换为CString类型

m\_product\_number=i+1;//产品编号从1开始，产品名字从0数组开始

UpdateData(FALSE);//更新

}

7）显示位图的函数

void CMyDlg::dspbmp(int i)//显示位图，第i副图片

{

int z;

z=i+1;//显示位图

CBitmap hbmp;//创建位图对象

HBITMAP hbitmap;//创建位图句柄

CStatic \*pStaic;//将pStatic指向要显示的地方

pStaic=(CStatic\*)GetDlgItem(IDC\_STATIC\_picture3);//将指针指向Picture控件

switch(z)

{//从资源文件中加载

//使用LoadImage（）函数加载图片，使用MAKEINTRESOURCE时加入图片ID号码即可

case1:hbitmap=(HBITMAP)::LoadImage(::AfxGetInstanceHandle(), MAKEINTRESOURCE(IDB\_BITMAP13), IMAGE\_BITMAP, 0,0,LR\_CREATEDIBSECTION);break;

case2:hbitmap=(HBITMAP)::LoadImage(::AfxGetInstanceHandle(), MAKEINTRESOURCE(IDB\_BITMAP14), IMAGE\_BITMAP, 0,0,LR\_CREATEDIBSECTION);break;

case 3:hbitmap = (HBITMAP)::LoadImage(::AfxGetInstanceHandle(), MAKEINTRESOURCE(IDB\_BITMAP15), IMAGE\_BITMAP, 0,0,LR\_CREATEDIBSECTION);break;

case 4:hbitmap = (HBITMAP)::LoadImage(::AfxGetInstanceHandle(), MAKEINTRESOURCE(IDB\_BITMAP16), IMAGE\_BITMAP, 0,0,LR\_CREATEDIBSECTION);break;

case 5:hbitmap = (HBITMAP)::LoadImage(::AfxGetInstanceHandle(), MAKEINTRESOURCE(IDB\_BITMAP17), IMAGE\_BITMAP, 0,0,LR\_CREATEDIBSECTION);break;

case 6:hbitmap = (HBITMAP)::LoadImage(::AfxGetInstanceHandle(), MAKEINTRESOURCE(IDB\_BITMAP18), IMAGE\_BITMAP, 0,0,LR\_CREATEDIBSECTION);break;

case 7:hbitmap = (HBITMAP)::LoadImage(::AfxGetInstanceHandle(), MAKEINTRESOURCE(IDB\_BITMAP19), IMAGE\_BITMAP, 0,0,LR\_CREATEDIBSECTION);break;

case 8:hbitmap = (HBITMAP)::LoadImage(::AfxGetInstanceHandle(), MAKEINTRESOURCE(IDB\_BITMAP20), IMAGE\_BITMAP, 0,0,LR\_CREATEDIBSECTION);break;

case 9:hbitmap = (HBITMAP)::LoadImage(::AfxGetInstanceHandle(), MAKEINTRESOURCE(IDB\_BITMAP21), IMAGE\_BITMAP, 0,0,LR\_CREATEDIBSECTION);break;

case 10:hbitmap = (HBITMAP)::LoadImage(::AfxGetInstanceHandle(), MAKEINTRESOURCE(IDB\_BITMAP22), IMAGE\_BITMAP, 0,0,LR\_CREATEDIBSECTION);break;

case 11:hbitmap = (HBITMAP)::LoadImage(::AfxGetInstanceHandle(), MAKEINTRESOURCE(IDB\_BITMAP23), IMAGE\_BITMAP, 0,0,LR\_CREATEDIBSECTION);break;

case 12:hbitmap = (HBITMAP)::LoadImage(::AfxGetInstanceHandle(), MAKEINTRESOURCE(IDB\_BITMAP24), IMAGE\_BITMAP, 0,0,LR\_CREATEDIBSECTION);break;

}

hbmp.Attach(hbitmap);//获取图片格式

BITMAP bm;//建立位图对象

hbmp.GetBitmap(&bm);

CDC dcMem;//建立设备环境类的对象

dcMem.CreateCompatibleDC(GetDC());//创建兼容设备环境，GetDC（）函数用于获取设备环境

CBitmap \*poldBitmap=(CBitmap\*)dcMem.SelectObject(hbmp);//指针指向位图

CRect lRect;//作图区域对象

pStaic->GetClientRect(&lRect);//获得客户区域

//显示位图 pStaic->GetDC()->StretchBlt(lRect.left ,lRect.top ,lRect.Width(),lRect.Height(), &dcMem,0 ,0,bm.bmWidth,bm.bmHeight,SRCCOPY);

dcMem.SelectObject(&poldBitmap);//将位图选入设备环境

}

8）“确定”按钮响应函数

void CMyDlg::OnBUTTON3sure()

{

char cstritem[10],cBuffer[50];

double dfitem;

m\_item1\_edit.GetWindowText(cstritem,10);//获取估价编辑框的10个字符转换为double型

dfitem=atof((LPCTSTR)cstritem);//将字符转换为数值float型

if(pcount>k)//次数超过，程序结束

{AfxMessageBox("对不起，您没有机会了，次数已到限制！");//头Afx表示对话框有警示牌

OnOK();}

if(dfitem==0){MessageBox("请输入估价！","输入估价",MB\_OK);} if(dfitem>price[i]){count++;pcount++;errocount++;if(pcount<=k)MessageBox("您输入的值过大了！","估值过大",MB\_OK);}

if((dfitem<price[i])&&(dfitem!=0)){count++;pcount++;errocount++;if(pcount<=k)MessageBox("您输入的值过小了！","估值过小",MB\_OK);}

if(dfitem==price[i]){count++;if(pcount<=k)MessageBox("恭喜您回答正确！进入下一题，再接再厉！","回答正确",MB\_OK);pcount=0;OnBUTTONnext();OnBUTTON1clear();}

//计算命中率

rate=(float)((count-errocount)/count)\*100;

\_gcvt(rate,10,cBuffer);//将数据rate取出10位转换为字符存在cBuffer[]之中

m\_rate\_edit=(LPCTSTR)cBuffer;//将字符转换为CString类型

//刷新正确和错误的次数

m\_error=errocount;

m\_correct=count-errocount;//总共猜对的次数

record();//判定数据是否已成为最高纪录，若是则记录

GetDlgItem(IDC\_EDIT1)->SetFocus();//每一次确定后将焦点设置在估价编辑框

UpdateData(FALSE);

}

9）“清除”按钮的响应函数

void CMyDlg::OnBUTTON1clear()

{

m\_item1\_edit.SetSel(0,-1);//全选编辑框内容

m\_item1\_edit.ReplaceSel(" ");//在清除按钮中将编辑框里的数据清除

UpdateData(FALSE);

}

10）时间响应函数OnTimer()

void CMyDlg::OnTimer(UINT nIDEvent)

{

if(nIDEvent==guesstime)

{

CProgressCtrl\*pProg=(CProgressCtrl\*)GetDlgItem(IDC\_PROGRESS1);

pProg->SetPos(pProg->GetPos()+1);//进度条前进一步

if(pProg->GetPos()>=60)//进度条已经走完

{ KillTimer(nIDEvent);//终止计时器

MessageBox("时间到！竞猜结束！","时间到！",MB\_OK);

record();

OnOK();

}

}

record();//时间到则决定数据是否存入到文本当中

CDialog::OnTimer(nIDEvent);

}

11）“重新开始”按钮的响应函数

void CMyDlg::OnAgain() //重新开始

{

// TODO: Add your control notification handler code here

i=0;//将图片信息的下标清零，

OnBUTTON2start();//游戏从第一张图片重新开始

}

12）“下一件商品”按钮的响应函数

void CMyDlg::OnBUTTONnext() //显示下一张图的信息

{

i++;

//换图后要进度条清零

CProgressCtrl\*pProg=(CProgressCtrl\*)GetDlgItem(IDC\_PROGRESS1);

pProg->SetPos(0);

SetTimer(guesstime,guessstep,NULL);//设置时钟

pcount=0;//当前图片所猜测次数归零

if(i>=12){i=0;}//12张图片进行循环，每到第12张就变为第0张

if((i>=0)&&(i<=11))

{ dspbmp(i);//画图片

dspname(i); //显示名称

}

}

13）前一件商品的响应函数

void CMyDlg::OnBUTTONprevious() //更换图片到前一张

{

i--;

//换图后要进度条清零

CProgressCtrl\*pProg=(CProgressCtrl\*)GetDlgItem(IDC\_PROGRESS1);

pProg->SetPos(0);

//重新设定计时器

SetTimer(guesstime,guessstep,NULL);

pcount=0;//当前图片的所猜次数清零

if(i>=12){i=0;}//图片循环

if(i<0){i=11;}

if((i>=0)&&(i<=11))

{ dspbmp(i);//显示图片

dspname(i); //显示名称

}

}

14）“练习一下”按钮的响应函数

void CMyDlg::OnBUTTONpractice() //练习模式下，可以随意更换商品

{

GetDlgItem(IDC\_EDIT1)->EnableWindow(TRUE);//激活估价编辑框

GetDlgItem(IDC\_BUTTON3\_sure)->EnableWindow(TRUE);//激活确定按钮

GetDlgItem(IDC\_BUTTON1\_clear)->EnableWindow(TRUE);//激活清除按钮

GetDlgItem(IDC\_AGAIN)->EnableWindow(TRUE);//激活重新开始按钮

GetDlgItem(IDC\_BUTTON\_next)->EnableWindow(TRUE);//激活下一件商品按钮

GetDlgItem(IDC\_BUTTON\_previous)->EnableWindow(TRUE);//激活上一件商品按钮

GetDlgItem(IDC\_BUTTON2\_start)->EnableWindow(TRUE);//激活开始按钮

GetDlgItem(IDC\_BUTTON\_site)->EnableWindow(FALSE);//禁用现场按钮 MessageBox("欢迎进入练习模式，此模式下您可以自由选择商品进行猜价！也可以选择重新开始！加油！","练习",MB\_OK);

}

15）“进入现场”按钮的响应函数

void CMyDlg::OnBUTTONsite()//现场模式下，只能回答正确才能进入下一轮

{

GetDlgItem(IDC\_BUTTON2\_start)->EnableWindow(TRUE);//激活开始按钮

GetDlgItem(IDC\_AGAIN)->EnableWindow(FALSE);//禁用重新开始按钮

GetDlgItem(IDC\_BUTTON\_next)->EnableWindow(FALSE);//禁用下一件商品按钮

GetDlgItem(IDC\_BUTTON\_previous)->EnableWindow(FALSE);//禁用上一件商品按钮

GetDlgItem(IDC\_BUTTON\_practice)->EnableWindow(FALSE);//禁用练习按钮MessageBox("欢迎进入现场模式，此模式下您只有回答正确才能进入下一题！否则游戏结束！加油！","竞猜现场",MB\_OK);

}

16）“最高纪录”按钮的响应函数

void CMyDlg::OnBUTTONtoprecord() //用于查看最高的记录

{

FILE \*out;//指向文件的指针

out=fopen("correct.txt","r");//打开文件，指向文件头

char str1[10];//用于存贮纪录的次数

fgets(str1,4,out);//从文件中获取4个字符存到str1中

int s=atoi(str1);//将字符转换为整型数据

CString s1;//创建CString对象

s1.Format("%d",s);//将整型次数s转换为CString类型s1,任意数据类//型转换为字符串

out=fopen("username.txt","r");

char str2[20];//用于存贮记录的姓名str2

fgets(str2,20,out);

CString str;

str="最多猜中次数为："+s1+"次；"+"姓名:"+str2;

MessageBox(str,"最高纪录",MB\_OK);

}

17）姓名“提交”按钮的响应函数

void CMyDlg::OnBUTTONregister() //报名函数，用于记录用户名

{

m\_username.GetWindowText(Tempname,10);

if(strcmp(Tempname,"")==0)

{

MessageBox("请输入您的姓名！","姓名记录",MB\_OK);

}

else { //报名以后才可以选择模式

GetDlgItem(IDC\_BUTTON\_practice)->EnableWindow(TRUE);//激活练习按钮

GetDlgItem(IDC\_BUTTON\_site)->EnableWindow(TRUE);//激活现场按钮

//报名成功则禁用报名

GetDlgItem(IDC\_BUTTON\_register)->EnableWindow(FALSE);//禁用报名按钮

m\_username.SetReadOnly(TRUE);//报名成功后将姓名编辑框设为只读

}

}

18）读出数据函数readdata（）

void CMyDlg::readdata()//读出数据函数（读出名字和正确次数）

{

FILE \*out;

out=fopen("correct.txt","r");

char str1[10];//用于存贮纪录的次数

fgets(str1,4,out);

int s=atoi(str1);

cs=s;

CString s1;

s1.Format("%d",s);//将次数s转换为字符串s1

out=fopen("username.txt","r");

char str2[20];//用于存贮记录的姓名str2

fgets(str2,20,out);

}

19）数据比较与写入函数record（）

void CMyDlg::record()

{ if((count-errocount)>=cs){//已经刷新记录，记录姓名和次数

FILE \*fp;

fp=fopen("username.txt","w");

fprintf(fp,"%10s",Tempname);

fclose(fp);

fp=fopen("correct.txt","w");

fprintf(fp,"%f",count-errocount);

fclose(fp);

}

}

20）“帮助”菜单的响应函数Onrule（）

void CMyDlg::Onrule() //在帮助对话框中弹出规则提示

{

MessageBox("1.单击“开始”按钮，游戏开始，系统将输出商品。\n2.请在60秒内输入您估计的商品价格，再按“确定”按钮。\n3.这时程序会提示您估计的商品价格是高了还是低了，在弹出的对话框中单击“确定”按钮，再次输入新的估计值，重复上一步。猜测次数大于6次会自动退出程序。若估计正确，程序会提示您估计正确！时间结束后，单击“重新开始”按钮开始下一轮游戏。","游戏规则",MB\_OK);

}

21）“退出”菜单的响应函数Onexit（）

void CMyDlg::Onexit()

{

MessageBox("游戏结束！谢谢使用！","退出游戏",MB\_OK);

OnOK();

}