## 4.1图形化界面（张君保编写）

### 4.1.1详细设计思想

图形化界面使用easyx绘制，其中，输出界面使用fillrectangle()函数绘制一个白色矩形，输出使用outtextxy()和drawtext()函数完成，每次输出完重新绘制一个白色矩形，实现清空输出区域的效果；按钮的制作同样使用fillrectangle()绘制一个矩形作为按钮，然后通过while(true)不断捕捉鼠标位置，并且通过显示屏幕长宽定位每个按钮的坐标，检测到鼠标在特定区域按下左键后，按钮对应单击数自增一，单击数是奇数执行对应按钮的功能，如果是偶数则执行初始化界面函数，依次实现点一下打开，再点一次关闭的效果，在树的图形化中，使用递归的思想，先给定树的根节点的坐标以及每次递归坐标向左右上下的偏移量，绘制出根节点后递归调用画树函数，绘制出左子树和右子树

开始

检测鼠标位置和单击情况

是否在第2个区域单击

是否在第1个区域单击

是否在第3个区域单击

是否在第4个区域单击

绘制初始化图形界面

FindNum++

弹窗要求输入要插入的内容

退出

AllNum++

按钮一单机次数=0，按钮二单击次数=0

FindNum%2==0?

Y

绘制查询菜单

N

AllNum%2==0

Y

绘制树

在第几个查询菜单单击

绘制选择菜单

按出生年月日查找

插入

1

按死亡年月日

按职业

按配偶姓名

3

2

4

5

N

N

判断在第几个选择菜单单击

查找父母

查找祖先

查找兄弟

查找孩子

查找后代

个人信息

1

2

3

4

5

6

输出相关信息

结束

/\*画圈\*/

1. void draw(int x, int y, char \*c) {
2. circle(x, y, 10);
3. outtextxy(x - 5, y - 5, \*c); //输出元素
4. }
5. /\*递归画树\*/
6. void imctree(person \* tree, int x, int y)
7. {
8. if (tree != NULL)
9. draw(x, y, tree->record.name);
10. if (tree->child != NULL) {
11. SuoJin++;
12. line(x - 14, y + 14, x - Left, y + 50);
13. Sleep(100);
14. imctree(tree->child, x - Left, y + 70);
15. }
16. if (tree->brother != NULL)
17. {
18. SuoJin++;
19. line(x + 14, y + 14, x + Left, y + 50);
20. Sleep(100);
21. imctree(tree->brother, x + Left, y + 70);
22. }

642

643 SuoJin--;

644

645 }

646

647 /\*初始化图形界面\*/

648

1. void initImg()
2. {

651

1. IMAGE imgbk;
2. loadimage(&imgbk, \_T("E:/vs2019/数据结构课设/数据结构课设/10.jpg"), 1280, 660);
3. putimage(0, 0, &imgbk);//显示图片

655

1. //loadimage(NULL, \_T("10.jpg"), width, height); /\*图片填充\*/
2. //fillrectangle(Frame, 60, width - 100, height - 200); /\*绘制输出区域\*/
3. IMAGE img;
4. loadimage(&img, \_T("E:/vs2019/数据结构课设/数据结构课设/11.jpg"), 1080, 400);
5. putimage(100, 60, &img); //显示图片

661

662 }

663

664 /\*绘制按钮\*/

665

1. void DrawButtle()
2. {
3. LOGFONT f;
4. gettextstyle(&f);
5. f.lfHeight = 35;
6. settextcolor(BLACK);
7. int left = Frame + Spacing / 2;
8. int top = height - 200 + Buttom;
9. int i;
10. char s0[15] = "查找";
11. char s1[15] = "插入";
12. char s2[15] = "退出";
13. char s3[15] = "查看";
14. char \*s[] = { s0,s1,s2,s3};
15. for (i = 0; i < buttleNum; i++)
16. {
17. fillrectangle(left, top, left + buttleWidth, top + buttleHeight);
18. outtextxy(left + 85, top + 6, s[i]);
19. left = left + buttleWidth + Spacing;
20. }
21. }

687

688 /\*绘制查询菜单\*/

689

1. void queryMenu()
2. {
3. int i;
4. char s0[15] = "根据姓名查询";
5. char s1[15] = "根据出生日期";
6. char s2[15] = "根据逝世日期";
7. char s3[15] = "根据职业查询";
8. char s4[30] = "根据配偶查询";
9. char \*s[] = { s0,s1,s2,s3,s4 };
10. int g;
11. LOGFONT f;
12. gettextstyle(&f); // 获取当前字体设置
13. f.lfHeight = 25;
14. settextstyle(&f);
15. int menuLeft = Frame + Spacing / 2; /\*定义菜单按钮初始化位置\*/
16. int MenuTop = height - 2\*Frame + Buttom + buttleHeight + 20; /\*菜单初始化高度\*/

706

1. for (i = 0; i < MenuNum; i++)
2. {
3. fillrectangle(menuLeft, MenuTop, menuLeft + MenuWidth, MenuTop + MenuHeight);
4. outtextxy(menuLeft + 15, MenuTop + 6, s[i]);
5. menuLeft = menuLeft + MenuWidth + MenuSpacing;
6. }
7. }

714

715 /\*插入\*/

716

1. info addinfo(info \* newPeople)
2. {
3. bool ifaddName;
4. bool ifaddDataBirth;
5. bool ifaddPlaceBirth;
6. bool ifaddSex;
7. bool ifaddProfession;
8. bool ifaddSpouseName;
9. bool ifaddparent;
10. bool ifgetfather;

727

1. ifgetfather= InputBox(getfather, 20, "", "请输入你要添加的成员的父亲", "查询",

300);

1. sscanf(getfather, "%s", findfather); /\*findfather 是根据用户输入的父亲姓名，用来作为findname()的参数找到要插入的人的父亲节点\*/

730

1. /\*输入要插入的节点信息\*/
2. newPeople = (info\*)malloc(sizeof(info));
3. if (ifgetfather)
4. {
5. ifaddName = InputBox(addName, 20, "", "请输入你要添加的成员的姓名", "查询",

300);

1. sscanf(addName, "%s", newPeople->name); /\*获取用户输入的字符串\*/
2. }
3. if (ifaddName)
4. {
5. ifaddDataBirth = InputBox(addData\_birth, 20, "", "请输入出生日期", "查询", 300);
6. sscanf(addData\_birth, "%s", newPeople->data\_birth); /\*获取用户输入的字符串

\*/

1. }
2. if (ifaddDataBirth)
3. {
4. ifaddPlaceBirth= InputBox(addPlace\_birth, 20, "", "请输入出生地", "查询", 300);
5. sscanf(addPlace\_birth, "%s", newPeople->place\_birth); /\*获取用户输入的字符串\*/
6. }
7. if (ifaddPlaceBirth)
8. {
9. ifaddSex= InputBox(addSex, 20, "", "请输入性别", "查询", 300);
10. sscanf(addSex, "%d", newPeople->sex); /\*获取用户输入的字符串\*/
11. }
12. if (ifaddSex)
13. {
14. ifaddProfession= InputBox(addProfession, 20, "", "请输入职业", "查询", 300);
15. sscanf(addProfession, "%s", newPeople->profession); /\*获取用户输入的字符串

\*/

1. }
2. if (ifaddProfession)
3. {
4. ifaddSpouseName = InputBox(addSpouse\_name, 20, "", "请输入配偶", "查询", 300);
5. sscanf(addSpouse\_name, "%s", newPeople->spouse\_name); /\*获取用户输入的字符串\*/
6. }
7. if (ifaddSpouseName)
8. {
9. ifaddparent = InputBox(addParent\_name, 20, "", "请输入父母姓名", "查询",

300);

1. sscanf(addParent\_name, "%s", newPeople->parent\_name); /\*获取用户输入的字符串\*/
2. }
3. if (ifaddparent)
4. {

770

1. LOGFONT f;
2. gettextstyle(&f);
3. f.lfHeight =35;
4. settextcolor(BLACK);
5. setbkmode(TRANSPARENT);
6. \_tcscpy\_s(f.lfFaceName, \_T("黑体"));
7. settextstyle(&f);
8. char add[] = "新成员信息录入成功\n欢迎新成员加入！！";
9. RECT r = { Frame + 15, 75,width - 135, height - 215 };
10. drawtext(add, &r, DT\_CENTER | DT\_VCENTER);
11. }
12. return \*newPeople;
13. }

784

1. /\* 绘制关系菜单\*/
2. void relationshipMenu()
3. {
4. int g;
5. LOGFONT f;
6. gettextstyle(&f); // 获取当前字体设置
7. f.lfHeight = 25;
8. settextcolor(BLACK);
9. setbkmode(TRANSPARENT);
10. \_tcscpy\_s(f.lfFaceName, \_T("黑体")); // 设置字体为“黑体”
11. settextstyle(&f);
12. char r0[15] = "查找父母";
13. char r1[15] = "查找祖先";
14. char r2[15] = "查找兄弟";
15. char r3[15] = "查找孩子";
16. char r4[15] = "查找后代";
17. char r5[15] = "个人信息";
18. char \*p[] = {r0,r1,r2 ,r3,r4,r5};
19. int relaMenuLeft = Frame + (width - 2 \* Frame) / 3; /\*定义关系菜单左侧坐标\*/
20. int relaMenuRight = relaMenuLeft + relationMenuWidth; /\*定义关系菜单右侧左边\*/
21. int relaMenuTop = 63; /\*定义第一个关系按钮顶部坐标\*/
22. int relaMenuButton = relaMenuTop + relationMenuHeight; /\*定义关系按钮底部坐标\*/
23. fillrectangle(Frame, 60, width - 100, height - 2 \* Frame); /\*清空输出区域\*/
24. for ( g = 0; g < relationMenuNum; g++)
25. {
26. setfillcolor(RGB(72, 81, 81));
27. fillrectangle(relaMenuLeft, relaMenuTop, relaMenuRight, relaMenuButton);
28. outtextxy(relaMenuLeft+120, relaMenuTop + 26, p[g]);
29. //setfillcolor(RGB(255, 255, 255));
30. relaMenuTop = relaMenuTop + relationMenuHeight+3;
31. relaMenuButton = relaMenuTop + relationMenuHeight;
32. }
33. setfillcolor(RGB(255,255,255 ));
34. }

819

820

821

822 /\* 鼠标事件\*/

823

1. char \* GetMouse(char out[500] )
2. {
3. person \* t;
4. char file[20] = "02.txt";
5. FILE \*fp = fopen(file, "r");
6. t = create\_family(fp);
7. MOUSEMSG mousemsg; /\* 定义鼠标消息\*/
8. while (true)
9. {
10. mousemsg = GetMouseMsg(); /\*获取一条鼠标消息\*/
11. int x, y;
12. bool mklButton;
13. x = mousemsg.x; /\*获取鼠标当前x坐标\*/
14. y = mousemsg.y; /\*获取鼠标当前y坐标\*/
15. mklButton = mousemsg.mkLButton; /\*获取鼠标当前左键是否按下\*/

839

840

1. if ((y > height - 2 \* Frame + Buttom) && (y < height - 2 \* Frame + Buttom + buttleHeight))
2. {
3. /\*查询\*/
4. if ((x > Frame + Spacing / 2) && (x < Frame + Spacing / 2 + buttleWidth)

&& mklButton) /\*鼠标在第一个按钮上单击左键时，激发鼠标事件 1\*/

1. {
2. fillrectangle(Frame, 60, width - 100, height - 2 \* Frame); /\*清空输出区域\*/
3. MouseFindNum++;
4. if (MouseFindNum % 2 != 0)
5. {
6. queryMenu();
7. }
8. else /\*再次点击时重新绘制界面，关闭菜单\*/
9. {
10. fillrectangle(Frame, 60, width - 100, height - 2 \* Frame); /\*清空输出区域\*/
11. initImg();
12. DrawButtle();
13. GetMouse(out);
14. }
15. }
16. /\*插入\*/
17. else if ((x > (Frame + Spacing / 2) + buttleWidth + Spacing) && (x <

(Frame + Spacing / 2) + 2 \* buttleWidth + Spacing) && mklButton)

1. {
2. fillrectangle(Frame, 60, width - 100, height - 2 \* Frame); /\*清空输出区域\*/
3. info \*newone= (info\*)malloc(sizeof(info));
4. newone = &addinfo(newpeople); /\*newonw就是要插入的节点信息\*/
5. FlushMouseMsgBuffer(); /\*清空鼠标缓存区\*/
6. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
7. \* \*
8. \* 插入的代码 \*
9. \* 变量名：newone \* 871 \* \*
10. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
11. }
12. /\*退出\*/
13. else if ((x > (Frame + Spacing / 2) + 2 \* (buttleWidth + Spacing)) && (x < (Frame + Spacing / 2) + 2 \* (buttleWidth + Spacing) + buttleWidth) && mklButton)
14. {
15. exit(1);
16. }
17. /\*画树\*/
18. else if ((x > (Frame + Spacing / 2) + 3 \* buttleWidth + Spacing) && (x < (Frame + Spacing / 2) + 3 \* (buttleWidth + Spacing) + buttleWidth) && mklButton)
19. {
20. LOGFONT f;
21. gettextstyle(&f);
22. f.lfHeight = 15;
23. settextcolor(BLUE);
24. setbkmode(TRANSPARENT);
25. \_tcscpy\_s(f.lfFaceName, \_T("黑体"));
26. settextstyle(&f);
27. setfillcolor(RGB(0,0,0));
28. fillrectangle(0, 60, width, height - 2 \* Frame);
29. MouseTreeNum++;
30. setlinecolor(BLUE);
31. if (MouseTreeNum % 2 != 0)
32. {
33. imctree(t, 640, 60+10+5);
34. setfillcolor(RGB(255, 255, 255));
35. }
36. else
37. {
38. setfillcolor(RGB(255, 255, 255));
39. setlinecolor(RGB(255, 255, 255));
40. fillrectangle(Frame, 60, width - 100, height - 2 \* Frame); /\*清空输出区域\*/
41. initImg();
42. DrawButtle();
43. GetMouse(out);
44. }
45. }
46. }
47. /\* 如果单机次数为奇数，说明菜单栏弹出，判断鼠标单击位置\*/
48. if (MouseFindNum % 2 != 0)
49. {
50. LOGFONT f;
51. gettextstyle(&f); // 获取当前字体设置
52. f.lfHeight = 25;
53. settextcolor(BLACK);
54. setbkmode(TRANSPARENT);
55. \_tcscpy\_s(f.lfFaceName, \_T("黑体")); // 设置字体为“黑体”
56. settextstyle(&f);

919

1. if (y > (height - 200 + Buttom + buttleHeight + 20) && y < (height - 200

+ Buttom + buttleHeight + 20 + MenuHeight))

1. {
2. LOGFONT f;
3. gettextstyle(&f); // 获取当前字体设置
4. f.lfHeight = 25; // 设置字体高度
5. settextcolor(BLACK);
6. setbkmode(TRANSPARENT);
7. \_tcscpy\_s(f.lfFaceName, \_T("黑体")); // 设置字体为“黑体”
8. settextstyle(&f);
9. /\*根据姓名查找\*/
10. if (x > (Frame + Spacing / 2) && x < (Frame + Spacing / 2 + MenuWidth) && mklButton)
11. {
12. ifgetName=InputBox(name, 20, "", "请输入你要查询的姓名", "查询",

300); /\*用于以对话框形式获取用户输入\*/

1. sscanf(name, "%s", getInfo); /\*获取用户输入的字符串\*/

934

1. RECT r = { Frame + 15, 75,width - 135, height - 215 };
2. FlushMouseMsgBuffer(); /\*清空鼠标缓存区\*/
3. }
4. /\*根据出生年月查找\*/
5. else if (x > (Frame + Spacing / 2) + MenuWidth + MenuSpacing && x <

(Frame + Spacing / 2) + 2 \* MenuWidth + MenuSpacing && mklButton)

1. {
2. InputBox(year, 20, "", "请输入你要查询的出生年月", "查询", 300,

0, false); /\*用于以对话框形式获取用户输入\*/

1. sscanf(year, "%s", getInfo);
2. findbirth(t, year);
3. FlushMouseMsgBuffer(); /\*清空鼠标缓存区\*/
4. }
5. /\*根据逝世日期查找\*/
6. else if (x > (Frame + Spacing / 2) + 2 \* MenuWidth + 2\*MenuSpacing && x < (Frame + Spacing / 2) + 3 \* MenuWidth +2\* MenuSpacing && mklButton)
7. {
8. InputBox(finddeath, 20, "", "请输入你要查询的逝世年月", "查询",

300, 0, false); /\*用于以对话框形式获取用户输入\*/

1. sscanf(finddeath, "%s", getInfo);
2. fdeathdata(t,finddeath);
3. FlushMouseMsgBuffer(); /\*清空鼠标缓存区\*/
4. }
5. /\*根据职业查找\*/
6. else if (x > (Frame + Spacing / 2) + 3\*MenuWidth +3\* MenuSpacing && x < (Frame + Spacing / 2) + 4 \* MenuWidth + 3 \* MenuSpacing && mklButton)
7. {
8. InputBox(findprofe, 20, "", "请输入你要查询的职业", "查询", 300,

0, false); /\*用于以对话框形式获取用户输入\*/

1. sscanf(findprofe, "%s", getInfo);
2. findpro(t, findprofe);
3. FlushMouseMsgBuffer(); /\*清空鼠标缓存区\*/
4. }
5. /\*根据配偶姓名查找\*/
6. else if (x > (Frame + Spacing / 2) + 4 \* MenuWidth +4 \* MenuSpacing && x < (Frame + Spacing / 2) + 5 \* MenuWidth + 4\*MenuSpacing && mklButton)
7. {
8. InputBox(findspouse, 20, "", "请输入你要查询的逝世年月", "查询",

300, 0, false); /\*用于以对话框形式获取用户输入\*/

1. sscanf(findspouse, "%s", getInfo);
2. findspous(t, findspouse);
3. FlushMouseMsgBuffer(); /\*清空鼠标缓存区\*/
4. }
5. }
6. }
7. if (ifgetName) /\* 如果输入要查询的姓名后单击确定，则输出关系菜单供选择\*/
8. {

974

1. relationshipMenu();
2. ifgetName = FALSE; /\*初始化，避免重复查找不输入\*/
3. Sleep(60);
4. MOUSEMSG mousemsg1; /\* 定义鼠标消息\*/
5. person \*str;
6. str = findname(t, name);
7. while (true)
8. {
9. mousemsg1 = GetMouseMsg(); /\*获取一条鼠标消息\*/
10. int x1, y1;
11. bool mklButton1;
12. bool ifgetGener;
13. x1 = mousemsg1.x; /\*获取鼠标当前x坐标\*/
14. y1 = mousemsg1.y; /\*获取鼠标当前y坐标\*/
15. mklButton1 = mousemsg1.mkLButton; /\*获取鼠标当前左键是否按下\*/

990

1. if (x1 > Frame + (width - 2 \* Frame) / 3 && x1 < Frame + width - 2 \* Frame / 3 + relationMenuWidth) /\*判断鼠标是否在关系菜单的宽度区间

\*/

1. {
2. if ((y1 > 63) && (y1 < (63 + relationMenuHeight)) && mklButton1) /\*单击第一个按钮查询父母\*/
3. {
4. fillrectangle(Frame, 60, width - 100, height - 2 \* Frame); /\*清空输出区域\*/
5. parents(t, str);
6. break;
7. }
8. else if ((y1 > (63 + relationMenuHeight + 3)) && (y1 < (63 + 2 \* relationMenuHeight + 3)) && mklButton1) /\*单击第二个按钮，查询祖先\*/
9. {
10. fillrectangle(Frame, 60, width - 100, height - 2 \* Frame); /\*清空输出区域\*/
11. ancestor(t, str);
12. break;
13. }
14. else if ((y1 > (63 + 2 \* relationMenuHeight + 6)) && (y1 < (63 + 3 \* relationMenuHeight + 6)) && mklButton1) /\*单击第三个按钮，

查询兄弟\*/

1. {
2. fillrectangle(Frame, 60, width - 100, height - 2 \* Frame); /\*清空输出区域\*/
3. brother(t, str);
4. break;
5. }
6. else if ((y1 > (63 + 3 \* relationMenuHeight + 9)) && (y1 < (63 + 4 \* relationMenuHeight + 9)) && mklButton1) /\*单击第四个按钮，

查询孩子\*/

1. {
2. fillrectangle(Frame, 60, width - 100, height - 2 \* Frame); /\*清空输出区域\*/
3. children1(str);
4. break;
5. }
6. else if ((y1 > (63 + 4 \* relationMenuHeight + 12)) && (y1 < (63 +
   1. \* relationMenuHeight + 12)) && mklButton1) /\*单击第五个按

钮，查询后代\*/

1. {
2. fillrectangle(Frame, 60, width - 100, height - 2 \* Frame); /\*清空输出区域\*/
3. ifgetGener = InputBox(getGeneration, 20, "", "您要查询第几代", "查询", 300);
4. sscanf(getGeneration, "%c",gener);
5. if (ifgetGener)
6. {
7. person \*find\_name = findname(t, name);
8. children(find\_name, gener);
9. }
10. break;
11. }
12. else if ((y1 > (63 + 5 \* relationMenuHeight + 12)) && (y1 < (63 +
    1. \* relationMenuHeight + 12)) && mklButton1) /\*单击第6个按

钮，查看个人信息\*/

1. {
2. fillrectangle(Frame, 60, width - 100, height - 2 \* Frame); /\*清空输出区域\*/

1032

1. display(str);
2. break;
3. }
4. }
5. }
6. }
7. }
8. return getInfo;
9. }

1042

1043

1. void title()
2. {
3. LOGFONT f;
4. gettextstyle(&f); // 获取当前字体设置
5. f.lfHeight = 35; // 设置字体高度为 12
6. settextcolor(RGB(72,81,81));
7. setbkmode(TRANSPARENT);
8. \_tcscpy\_s(f.lfFaceName, \_T("黑体")); // 设置字体为“黑体”
9. settextstyle(&f);
10. char s[] = "家庭族谱查询";
11. outtextxy(15,15, s);
12. }

1056