



计算机综合课程设计项目报告

COMPUTER INTEGRATED CURRICULUM DESIGN PROJECT REPORT

设计题目: 校园导航系统

组长姓名: 钟源 学号: 04022212

组员姓名: 王坤 学号: 04022725

黄蕾 学号: 04022611

许睿哲 学号: 04222131

左贤玮 学号: 04022115

报告人: 钟源 学号: 04022212

一、内容简介

(一) 课题介绍

小组经过全员讨论后，计划以东南大学九龙湖校区为地图，利用所学的 C++ 与 MFC 知识，设计出简易但功能完全的校园导航系统。比如可以实现“校园内建筑间经过道路的最佳路径”、“结合建筑物及内部设立的院系进行信息展示”、“校园旅游规划”、“地图的缩放”等等....小组将以图文并茂、音频等多方位形式呈现导航系统。

主要功能有：

- ① 建筑信息的建立:根据“定位”按钮法或坐标法的建筑名,查询到建筑的编号,然后在建筑上标注蓝点，作为对用户的提示；
- ② 建立道路节点，并标注黄点；
- ③ 校园导航的实现；
- ④ 目的地线路查询的实现：在查询设置的道路节点以及建筑物基础上，利用最短路径算法（即弗洛伊德算法），计算出经过道路的最短路径，将距离显示在弹出的对话框，再利用绘制路径函数，将路径显示在窗口中；
- ⑤ 旅游线路查询的实现：以最先点选的道路节点作为起始点，算出 A 点到其他各个旅游点(B、C、D、E 等)的最短路径，然后求出各条最短路径中最短的一条路径，将该旅游点作为第二个旅游景点，并将该点标记已访问过，然后以该点作为起点，继续进行如上的步骤，选出下一个未访问过的旅游点，然后将其标记已访问过，不断进行下去，直至全部所选节点都已经访问。最后得出以依次访问的旅游点数组,再根据数组中的顺序，将旅游线路绘制出来；

- ⑥ 辅助功能的实现：通过点击按钮与移动光标，可以实现地图放大、移动、全屏和建筑间距离量测等。

(二) 任务需求分析

1.目标

完成题目全部要求，实现校园内建筑的信息查询，并在已有要求之上设计美观简洁的展示页面。

具体体现在：

- 1) 通过地图显示建筑，鼠标移动至建筑图标上展示建筑信息；
- 2) 项目尽可能包括东大元素；
- 3) 简化、美化页面，使得页面清晰易懂；
- 4) 实现经过不同建筑的最短路径算法；
- 5) 实现地图的放大缩小；
- 6) 实现图片和音乐的插入。

2.特点

引导界面：“展示介绍”按钮，“校园导航”按钮，“播放”与“停止”按钮，“最大化”与“最小化”按钮，允许自由播放或停止音乐。

展示界面：展示九龙湖校区三维俯视图，图上标注建筑物，当鼠标移至对应建筑物上时，会显示该建筑物的简介。

导航界面：展示九龙湖二维俯视图，建筑标上蓝点，道路标上黄点，可以缩放俯视图，放大后的地图可拖动。导航分为两种，一种为经过最短路程达到目的地，选择起点和目的地后将显示路程距离；一种为可选多个目的地，系统将制定经过所选建筑物的最短路线，并将显示途中所经过的建筑物以及各建筑物间的路程。两种导航均可选择地图上所有的建筑物。

3.功能规定

启动程序调试后，出现引导窗口，此时系统默认不播放音乐，用户可以手动

单击“播放”按钮播放音乐，也可以单击“停止”按钮停止播放。

单击“展示介绍”按钮后，系统弹出“介绍界面”窗口，此时引导窗口不可操作。

在“展示介绍”中，系统通过捕捉鼠标在窗口内的位置，获取消息，从而判断是否展示信息以及展示何种信息。

单击关闭按钮，停止展示介绍，回到引导界面，此时引导窗口可见可操作。

单击“校园导航”按钮，进入“导航界面”窗口，此时引导窗口可见不可操作。

在“导航界面”中，用户可以自由缩放地图，可以自由选择地图上所标注的起点和目的地，系统会通过最短路径算法为用户计算出所需的最短路径。

4.性能规定

- 1) 实用性：能较好模拟校园地图，给出有价值的参考路线。
- 2) 简易性：只有校园建筑展示和导航系统，导航系统只分为最短路程和合理观光路线，操作简单，结果易懂。
- 3) 丰富性：具有音乐，图文并茂，可以良好的展示建筑以及路线。
- 4) 可读性：代码尽量简单易懂。

（三）程序功能设计说明

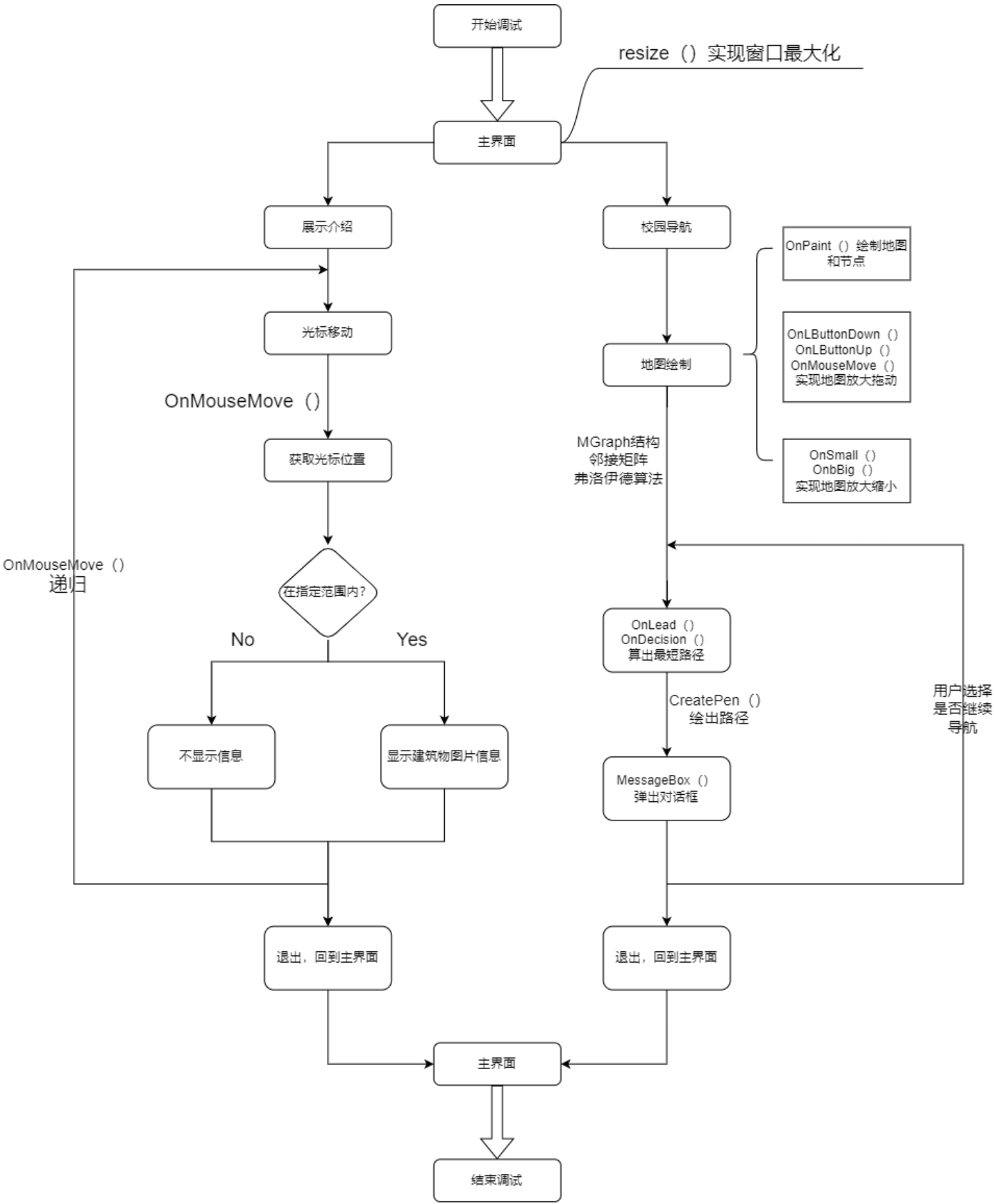
启动程序，进入主界面。单击“播放”按钮播放音乐，“停止”按钮停止音乐。

单击“展示介绍”按钮进入介绍界面，移动鼠标至对应建筑可以查看对应信息简介。关闭展示界面，回到主界面。

单击“校园导航”按钮进入导航界面，导航界面可以放大缩小展示的地图。

选择出发地点和目的地，可以搜索最短路径；选择现所在地点和目的地，可以制定观光路线。系统会根据选择显示路程以及沿途建筑。

(四) 程序模块框图



二、完成情况

(一) 对象设计和算法设计描述

1.设计独立功能对话框：新增设计两个对话框资源，即：IDD_Introduction, IDD_Guidance, 并创建两个相应的对话框类 IntroductionDlg 类,CGuidanceDlg 类来实现两块功能——东大地图显示介绍和校园导航。

2.对于地图显示介绍：插入一个 Picture Control 控件显示地图，设计一个 OnMouseMove()函数，递归调用呈现出实时跟随鼠标光标移动，显示相应建筑的图片介绍的效果。

部分代码如下：

```
void IntroductionDlg::OnMouseMove(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default

    int x = point.x;
    int y = point.y;
    CString str;
    str.Format(_T("%d,%d"), x, y);
    CDC* pDC = GetDC();
    pDC->TextOut(0, 0, str);
    ReleaseDC(pDC);
    if (point.x > 300 && point.x < 340 && point.y > 380 && point.y < 420 ) {
        picture.SetWindowPos(NULL, point.x + 1, point.y + 1, 0, 0, SWP_NOZORDER |
SWP_NOSIZE);
        CStatic* pPic = (CStatic*)GetDlgItem(IDC_Picture);
        CBitmap bitmap;
        bitmap.LoadBitmap(IDB_BITMAP1);
        pPic->SetBitmap((HBITMAP)bitmap.GetSafeHandle());
        //computer software
        HCURSOR hCur = LoadCursor( NULL , IDC_HAND );
        ::SetCursor(hCur);
    }
    .....
    CDialog::OnMouseMove(nFlags, point);
}
```

3.对于校园导航：插入一个 Picture Control 控件显示卫星地图，设计若干个 Button 控件，Text 控件，Combo-Box 控件，Listbox 控件以及对应的控件变量来实现信息的交互。

4.对于校园导航：编写 OnPaint () ，画出道路节点（黄点），建筑物点（蓝点）。

蓝点部分代码如下：

```
CBrush brs(RGB(0,0,255));
CPen pen(PS_SOLID,1,RGB(0,0,255));
pDC->SelectObject(pen);
pDC->SelectObject(brs);
pDC->Ellipse(443.60073801,155.82725832,453.60073801,165.82725832);
```

5.对于校园导航：设计 OnLButtonDown () ， OnLButtonUp () ， OnMouseMove () 实现放大时的拖动。

具体代码如下：

```
void CGuidanceDlg::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    if (point.x>15 && point.x<(15+551) && point.y>15 && point.y<(15+673))
        m_lButtonDown=true;
    else
        m_lButtonDown=false;
    m_orignalPoint=point;
    m_mapTempLocation=m_mapLocation;
    CDialog::OnLButtonDown(nFlags, point);
}

void CGuidanceDlg::OnLButtonUp(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    m_lButtonDown=false;
    m_destinationPoint=point;
    m_mapLocation.x=m_mapTempLocation.x-(m_destinationPoint.x-m_orignalPoint.x);
    m_mapLocation.y=m_mapTempLocation.y-(m_destinationPoint.y-m_orignalPoint.y);
    CDialog::OnLButtonUp(nFlags, point);
}

void CGuidanceDlg::OnMouseMove(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
```



```

if (point.x<15 || point.x>(15+551) || point.y<15 || point.y>(15+673))
    m_lButtonDown=false;
int changesToPointX=0;
int changesToPointY=0;
CDC *pDC=GetDC();
CBitmap BmpMem;
CDC MemDC;
MemDC.CreateCompatibleDC(pDC);
if(m_size==1 && m_lButtonDown==true && m_mapLocation.x>0 && m_mapLocation.y>0
    && m_mapLocation.x<(1682-673) && m_mapLocation.y <(1377-551)){
    if (m_size==0)
        BmpMem.Attach((HBITMAP)::LoadImage(NULL,"small.bmp",IMAGE_BITMAP,
        0,0,LR_LOADFROMFILE));
    else
        BmpMem.Attach((HBITMAP)::LoadImage(NULL,"big.bmp",IMAGE_BITMAP,
        0,0,LR_LOADFROMFILE));
    changesToPointX=point.x-m_currentPoint.x;
    changesToPointY=point.y-m_currentPoint.y;
    m_mapLocation.x-=changesToPointX;
    m_mapLocation.y-=changesToPointY;
    MemDC.SelectObject(&BmpMem);
    pDC->BitBlt(15,15,673,551,&MemDC, m_mapLocation.x,
        m_mapLocation.y,SRCCOPY);
    BmpMem.DeleteObject();
    MemDC.DeleteDC();
    ReleaseDC(pDC);
    m_currentPoint=point;
    CDialog::OnMouseMove(nFlags, point);
}

```

6.实现最短路径算法：巧妙运用邻接矩阵，弗洛伊德算法以及新增的结构来实现。

邻接矩阵是一种表示图的方式，其中使用一个二维数组来表示图中顶点之间的连接关系。弗洛伊德算法是一种用于解决带权图中任意两点之间最短路径的算法。它通过动态规划的方式逐步求解最短路径。具体步骤如下：

- 1) 初始化一个二维数组 D，用于记录任意两点之间的最短路径长度。
- 2) 遍历图中的每一对顶点，初始化 D 数组中的值为两个顶点之间的边的权重。

- 3) 通过遍历每个顶点 k，将顶点 k 作为中间节点，更新 D 数组中的值。
- 4) 重复以上步骤直到完成所有顶点的遍历。
- 5) 最终得到的 D 数组即为任意两点之间的最短路径长度数组。

相关代码如下：

```
typedef struct { // 结构
    float edges[MAXV][MAXV];
    int a,b,n;
    VertexType vexs[MAXV];
}MGraph;

void Floyd(MGraph g,bool flag) { // 弗洛伊德算法
    INT path[MAXV][MAXV];
    int i,j,k;
    for(i=0;i<66;i++) {
        for(j=0;j<66;j++) {
            AA[i][j]=g.edges[i][j];
            path[i][j]=-1;
        }
    }
    for(k=0;k<66;k++) {
        for(i=0;i<66;i++) {
            for(j=0;j<66;j++) {
                if(AA[i][j]>AA[i][k]+AA[k][j]) {
                    AA[i][j]=AA[i][k]+AA[k][j];
                    path[i][j]=k;
                }
            }
        }
    }
    if(flag)Dispath1(AA,path,g.a,g.b);
}
```

7.绘制路径： 在 OnLead() ， OnDecision()中调用 Floyd()以及画笔程序完成最短路径的绘制以及路径长度显示窗口的弹出。

相关代码如下：

```
void CGuidanceDlg::OnLead()
{
    UpdateData(TRUE);
    int i,j;
```

```

        CString out;
        Invalidate();
        UpdateWindow();
        m=1;
        MGraph g;
        for (i=0;i<65;i++){
            for (j=0;j<65;j++){
                g.edges[i][j]=seu[i][j];
            }
        }
        int m=m_original.GetCurSel();
        int n=m_destination.GetCurSel();
        g.a=m;
        g.b=n;
        Floyd(g,1);
        i=0;j=1;
        Float xx[65]={.....};
        float yy[65]={.....};
        CClientDC dc(this);
        CPen penNew;
        penNew.CreatePen(PS_SOLID, 4, RGB(220,20,60));
        CPen* Old = dc.SelectObject(&penNew);
        CString ss;
        ss.Format("%d",(int)(AA[g.a][g.b]*2.631));
        out=out+name[g.a]+"->" +name[g.b]+":"+ss+"米 × \n";
        while(way[i]!=999&&way[j]!=999)
        {
            m=way[i];
            n=way[j];
            dc.MoveTo(xx[m],yy[m]);
            dc.LineTo(xx[n],yy[n]);
            i++;j++;
        }
        MessageBox(out);
        UpdateData(FALSE);
    }
}

```

8.放大与缩小：利用 OnSmall(),OnBig() 实现卫星地图的放大与缩小转换.

相关代码如下：

```

void CGuidanceDlg::OnSmall()
{

```

```

// TODO: Add your control notification handler code here
m_size=0;
CDC *pDC=GetDC();
CBitmap BmpMem;
CDC MemDC;
MemDC.CreateCompatibleDC(pDC);
    if (m_size == 0)
    {
        CStatic* pPic = (CStatic*)GetDlgItem(IDC_Picture);
        CBitmap bitmap;
        bitmap.LoadBitmap(IDB_BITMAP_SMALL);
        pPic->SetBitmap((HBITMAP)bitmap.GetSafeHandle());
        pPic->ShowWindow(SW_SHOW);
    }
    else
    {
        CStatic* pPicture = (CStatic*)GetDlgItem(IDC_Picture);
        pPicture->ShowWindow(SW_HIDE);
        BmpMem.Attach((HBITMAP)LoadImage(AfxGetInstanceHandle(),
            MAKEINTRESOURCE(IDB_BITMAP_BIG),
            IMAGE_BITMAP,
            0, 0,
            LR_CREATEDIBSECTION));
    }
    MemDC.SelectObject(&BmpMem);
    pDC->BitBlt(15,15,673,551,&MemDC,0,0,SRCCOPY);
//.....省略画点函数
    BmpMem.DeleteObject();
    MemDC.DeleteDC();
    ReleaseDC(pDC);
}
void CGuidanceDlg::OnBig()
{
    // TODO: Add your control notification handler code here
    m_size=1;
    CDC *pDC=GetDC();
    CBitmap BmpMem;
    CDC MemDC;
    MemDC.CreateCompatibleDC(pDC);
    if (m_size == 0)
    {

```

```

        CStatic* pPic = (CStatic*)GetDlgItem(IDC_Picture);
        CBitmap bitmap;
        bitmap.LoadBitmap(IDB_BITMAP_SMALL);
        pPic->SetBitmap((HBITMAP)bitmap.GetSafeHandle());
        pPic->ShowWindow(SW_SHOW);
    }
    else
    {
        CStatic* pPicture = (CStatic*)GetDlgItem(IDC_Picture);
        pPicture->ShowWindow(SW_HIDE);
        BmpMem.Attach((HBITMAP)LoadImage(AfxGetInstanceHandle(),
            MAKEINTRESOURCE(IDB_BITMAP_BIG),
            IMAGE_BITMAP,
            0, 0,
            LR_CREATEDIBSECTION));
    }
    MemDC.SelectObject(&BmpMem);
    pDC->BitBlt(15,15,673,551,&MemDC,m_mapLocation.x,m_mapLocation.y,SRCCOPY);
}

```

9.窗口的最大化与最小化：利用 OnSize()实现主对话框的窗口最大化最小化

功能以及背景图片随窗口大小改变的优化设计.

相关代码如下：

```

BOOL CWholeDlg::OnInitDialog()
{
    CRect rect;
    GetClientRect(&rect);
    Old.x = rect.right - rect.left;
    Old.y = rect.bottom - rect.top;
    m_oldWidth = rect.Width();
    m_oldHeight = -rect.Height();
    return TRUE; // return TRUE unless you set the focus to a control
}

void CWholeDlg::OnSize(UINT nType, int cx, int cy)
{
    CDialog::OnSize(nType, cx, cy);
    if (nType == SIZE_RESTORED || nType == SIZE_MAXIMIZED)
    {
        resize();
    }
}

```

```

    Invalidate(FALSE);
}

void CWholeDlg::resize()
{
    float fsp[2];
    POINT Newp;
    CRect recta;
    GetClientRect(&recta);
    Newp.x = recta.right - recta.left;
    Newp.y = recta.bottom - recta.top;
    fsp[0] = (float)Newp.x / Old.x;
    fsp[1] = (float)Newp.y / Old.y;
    CRect Rect;
    int woc;
    CPoint OldTLPPoint, TLPPoint;
    CPoint OldBRPoint, BRPoint;
    HWND hwndChild = ::GetWindow(m_hWnd, GW_CHILD);
    while (hwndChild)
    {
        woc = ::GetDlgCtrlID(hwndChild);
        GetDlgItem(woc)->GetWindowRect(Rect);
        ScreenToClient(Rect);
        OldTLPPoint = Rect.TopLeft();
        TLPPoint.x = long(OldTLPPoint.x * fsp[0]);
        TLPPoint.y = long(OldTLPPoint.y * fsp[1]);
        OldBRPoint = Rect.BottomRight();
        BRPoint.x = long(OldBRPoint.x * fsp[0]);
        BRPoint.y = long(OldBRPoint.y * fsp[1]);
        Rect.SetRect(TLPPoint, BRPoint);
        GetDlgItem(woc)->MoveWindow(Rect, TRUE);
        hwndChild = ::GetWindow(hwndChild, GW_HWNDNEXT);
    }
    Old = Newp;
}

```

10.音乐的插入:

相关代码如下:

```

void CWholeDlg::OnButton3()
{
    if(music){
        PlaySound(NULL,NULL,SND_FILENAME);
    }
}

```

```

music=0;

GetDlgItem(IDC_BUTTON3)->SetWindowText("播放");

}

else{

PlaySound(MAKEINTRESOURCE(IDR_WAVE1),AfxGetResourceHandle(),SND_ASYNC|SND_RESOURCE|SND_NODEFAULT|SND_LOOP);

music=1;

GetDlgItem(IDC_BUTTON3)->SetWindowText("停止");

} }

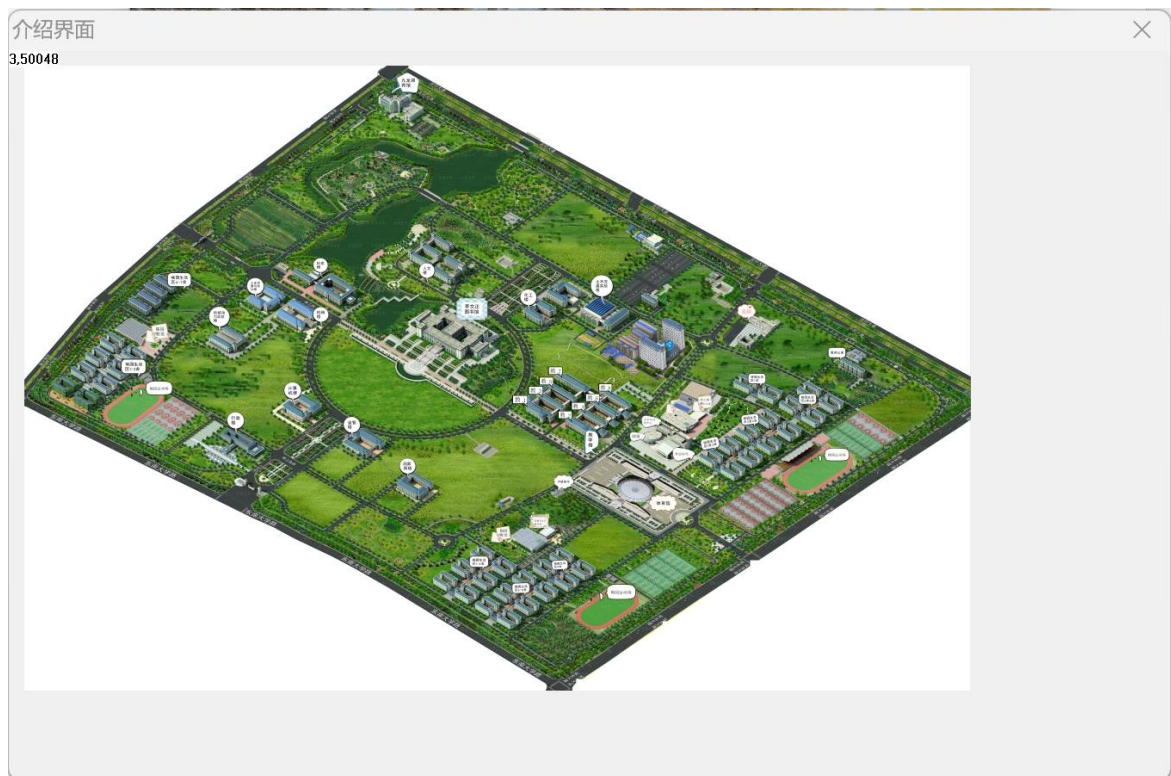
```

(二) 测试效果

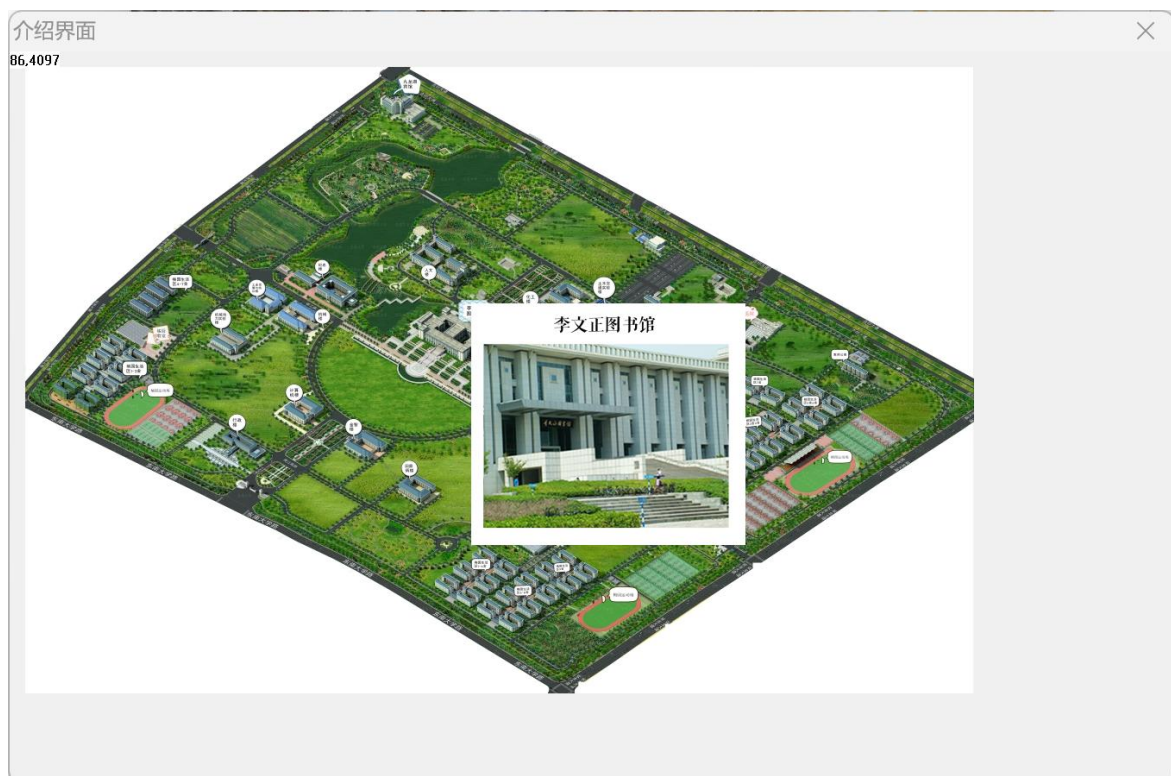
1.启动程序调试，系统进入引导界面：



2.单击“展示介绍”，进入介绍界面：



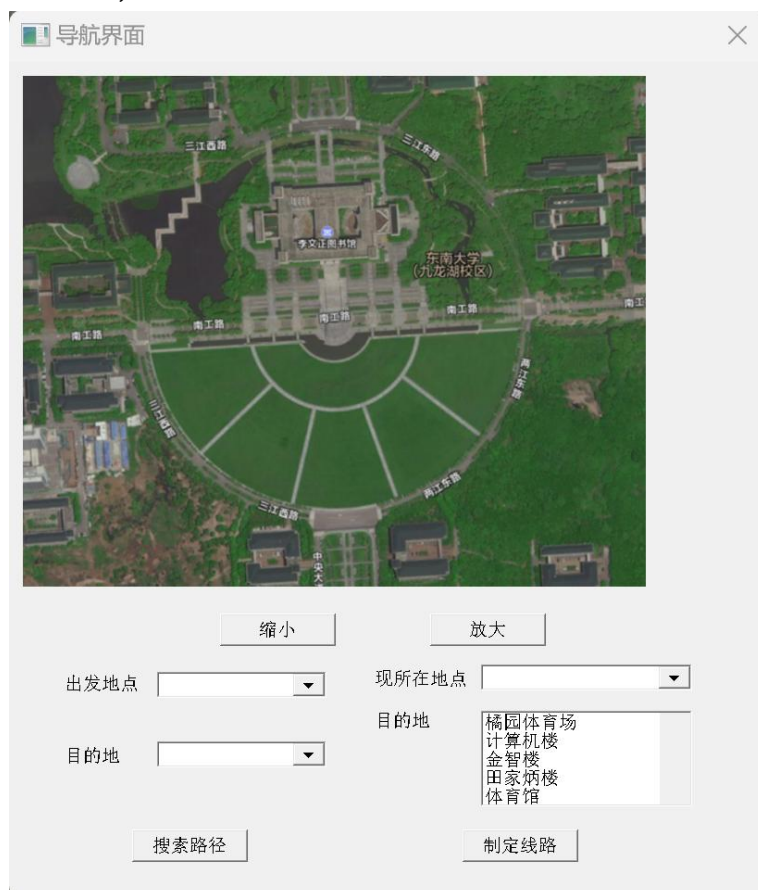
鼠标移动到相应位置，则有该建筑的介绍图片浮现：



3.退出展示，回到主页面，单机“校园导航”，进入导航界面：



可单击放大按钮，查看放大的地图并用鼠标拖拽：



可选择地点，进行导航：

