OpenCPN 航路部分主要涉及这几个类:RoutePoint、Route、MyConfig。

一． 航路创建

点击功能按钮进入航路模式，在 ChartCanvas 类的左键单击事件处理中，判断如果为航路模式，进入创建航路的处理代码。

大体流程如下：

1. 本次单击如果为新建航路

```
if ( parent_frame->nRoute_State == 1 )
{
        m_pMouseRoute = new Route();
        pRouteList->Append ( m_pMouseRoute );
        r_rband.x = x;
        r_rband.y = y;
}
```

2. 如果单击选取的位置有可以重用的航路点，弹出提示框供选择是否重用。

2.1 判断是否鼠标单击点选择半径范围内存在点。

RoutePoint *pNearbyPoint = pWayPointMan->GetNearbyWaypoint(rlat, rlon, nearby_radius_meters);

GetNearbyWaypoint 函数： 遍历所有航路点，如果存在在范围内的点，返回，否则返回 NULL。代码如下：

```
wxRoutePointListNode *node = m_pWayPointList->GetFirst();
while(node)
{
        RoutePoint *pr = node->GetData();

        double a = lat - pr->m_lat;
        double b = lon - pr->m_lon;
        double l = sqrt((a*a) + (b*b));

        if((l * 60. * 1852.) < radius_meters)
                return pr;

        node = node->GetNext();
}
return NULL;
```

3. 根据选择重用和不重用以不同方式添加点，设置 nRoute_State

```
parent_frame->nRoute_State++;
```

刷新显示：

```
Refresh ( false );
```

二． 航路绘制

ChartCanvas::OnPaint --> DrawOverlayObjects --> DrawAllRoutesInBBox --> ...
--> Route::Draw

1. 航路点绘制

WayPointman 构造中创建一些了的 wxImage 对象，使用下面的宏：

```
#define MAKEICONARRAYS(key, xpm_ptr, description)\
 pmarkiconImage = new wxImage(( const char **)xpm_ptr);\
 ProcessIcon(pmarkiconImage, _T(key), _T(description));\
 delete pmarkiconImage;
```

其中，航路点默认为

MAKEICONARRAYS("diamond", diamond, "Diamond")

激活的航路点:

MAKEICONARRAYS("activepoint", activepoint, "Active WP")

在 OpenCPN 中海图上其他点元素也是以此方式绘制。

在 ProcessIcon 函数中根据不同颜色模式获取 Icon 定义。

加载用户定义 Icon，貌似某个版本后的扩展功能。

```
wxString UserIconPath = g_PrivateDataDir;    //g_SData_Locn;
wxChar sep = wxFileName::GetPathSeparator();
if (UserIconPath.Last() != sep)
        UserIconPath.Append(sep);
UserIconPath.Append(_T("UserIcons"));

if(wxDir::Exists(UserIconPath))
{
        wxArrayString FileList;

        wxDir dir(UserIconPath);
        int n_files = dir.GetAllFiles(UserIconPath, &FileList);

        for(int ifile=0 ; ifile < n_files ; ifile++)
        {
                wxString name = FileList.Item(ifile);

                wxFileName fn(name);
                wxString iconname = fn.GetName();
                wxBitmap icon1;

                if (icon1.LoadFile(name,wxBITMAP_TYPE_XPM))
                {
                        wxImage *iconImage = new wxImage;
                        *iconImage = icon1.ConvertToImage();
                        ProcessIcon(iconImage,iconname,iconname);
                        delete iconImage;
                }
```

```
}
```

2. Leg line 绘制 --> RenderSegment

三． 数据保存和恢复

在创建航路、修改等操作执行之后，需要将相应数据的变更保存，使得修改不仅仅对当前有效，这些功能由 MyConfig 承担。

主要成员函数包括：

```
virtual bool AddNewRoute(Route *pr, int ConfigRouteNum = -1);
virtual bool UpdateRoute(Route *pr);
virtual bool DeleteConfigRoute(Route *pr);

virtual bool AddNewWayPoint(RoutePoint *pWP, int ConfigRouteNum = -1);
virtual bool UpdateWayPoint(RoutePoint *pWP);
virtual bool DeleteWayPoint(RoutePoint *pWP);
```

以 UpdateRoute 为例说明。

```
bool MyConfig::UpdateRoute ( Route *pr )
{
    if (pr->m_bIsInLayer) return true;

    if ( pr->m_bIsTrack )
    {
        if (!m_bIsImporting)
        {
            GpxTrkElement * trk = ::CreateGPXTrk( pr );
            trk->SetSimpleExtension(wxString(_T("opencpn:action")),
wxString(_T("update")));
            GpxRootElement * rt = (GpxRootElement *)
m_pNavObjectChangesSet->RootElement();
            rt->AddTrack(trk);
            StoreNavObjChanges();
        }
        return false;
    }

    wxString str_buf;

//  Build the Group Name
    wxString t ( _T ( "/Routes/RouteDefn" ) );
    str_buf.Printf ( _T ( "%d" ), pr->m_ConfigRouteNum );
    t.Append ( str_buf );
```

```
            DeleteGroup ( t );
            if (!m_bIsImporting)
            {
                  GpxRteElement * rte = ::CreateGPXRte( pr );
                  rte->SetSimpleExtension(wxString(_T("opencpn:action")),
wxString(_T("update")));
                  GpxRootElement * rt = (GpxRootElement *)
m_pNavObjectChangesSet->RootElement();
                  rt->AddRoute(rte);
                  StoreNavObjChanges();
            }
          return true;
      }
```

可以看出 MyConfig 类中实际是封装了 Gpx... 类对于 xml 文件的操作。下面介绍 Gpx... 。

对应航路中的航路点和航路，分别有 GpxWptElement 和 GpxRteElement 类。他们都继承与 TiXmlElement。

GpxDocument 继承与 TiXmlDocument。

TinyXml 是一个轻量级、方便实用的 xml 操作库，TiXmlDocument 代表 xml 文档本身，负责打开、保存修改到实际文件；TiXmlElement 代表 xml 结构中的一个节点，是数据携带者。即，在 OpenCPN 中，使用 GpxDocment 和 各*Element 完成航路对应 xml 的操作。

以 GpxWptElement 说明：

```
GpxWptElement::GpxWptElement(char *waypoint_type, double lat, double lon, double ele,
wxDateTime * time,
                  double magvar, double geoidheight, const wxString &name, const wxString
&cmt,
                  const wxString &desc, const wxString &src, ListOfGpxLinks *links, const
wxString &sym, const wxString &type,
                  GpxFixType fixtype, int sat, double hdop, double vdop, double pdop,
                  double ageofgpsdata, int dgpsid, GpxExtensionsElement *extensions) :
TiXmlElement(waypoint_type)
      {
            SetAttribute("lat", wxString::Format(_T("%.9f"), lat).ToUTF8());
            SetAttribute("lon", wxString::Format(_T("%.9f"), lon).ToUTF8());

            if (ele != 0)
                  SetProperty(wxString(_T("ele")), wxString::Format(_T("%f"), ele));

            if (time)
                  if (time->IsValid())
                        SetProperty(wxString(_T("time")),
time->FormatISODate().Append(_T("T")).Append(time->FormatISOTime()).Append(_T("Z")));
            if (magvar != 0)
```

```
                SetProperty(wxString(_T("magvar")), wxString::Format(_T("%f"), magvar));
        if (geoidheight != -1)
                SetProperty(wxString(_T("geoidheight")),        wxString::Format(_T("%f"),
geoidheight));
        if (!name.IsEmpty())
                SetProperty(wxString(_T("name")), name);
        if (!cmt.IsEmpty())
                SetProperty(wxString(_T("cmt")), cmt);
        if (!desc.IsEmpty())
                SetProperty(wxString(_T("desc")), desc);

      ......


        if (dgpsid != -1)
                SetProperty(wxString(_T("dgpsid")), wxString::Format(_T("%u"), dgpsid));
        if (extensions)
                LinkEndChild(extensions);
}
```

构造函数中设置了该 xml 的各属性，最后一句

```
LinkEndChild(extensions);
```

是将该节点设为 Route 所在节点的子节点。