1. PLAOnlineMapOverview 加载网络参考地图

**keyPoint1:在菜单中添加一个子项**

void PLAOnlineMapOverviewPlugin::initGui()

{

if(m\_pQGisIface == 0){return;}

m\_pAction = new QAction(m\_pQGisIface);

m\_pAction->setIcon(QIcon(tr("Resources/OnlinemapDownload.png")));

m\_pAction->setText("网络地图参考");

/\*QMenu\* layerMenu = m\_pQGisIface->layerMenu();

if(layerMenu != 0){layerMenu->addAction(m\_pAction);}\*/

m\_pQGisIface->viewMenu()->addAction(m\_pAction); //viewMenu表示“视图”菜单

connect(m\_pAction, SIGNAL(triggered()), this, SLOT(onOnlineMapOverview()));

}

**keyPoint2:卸载在菜单中添加的菜单项**

void PLAOnlineMapOverviewPlugin::unload()

{

m\_pQGisIface->viewMenu()->removeAction(m\_pAction);

/\*QMenu\* layerMenu = m\_pQGisIface->layerMenu();

if(layerMenu != 0){layerMenu->removeAction(m\_pAction);}\*/

delete m\_pAction;

if(m\_pOverviewDlg != 0)

delete m\_pOverviewDlg;

m\_pOverviewDlg = 0;

}

1. PLABlockSmoothTool 地块光滑处理工具

**keyPoint1:在工具栏中添加一个子项**

void PLABlockSmoothTool::initGui()

{

if(mQGisIface == 0)

{

return;

}

if ( mAction == 0 )

{

mAction = new QAction(mQGisIface->mainWindow());

mAction->setText(tr("地块光滑处理工具"));

mAction->setIcon(QIcon(tr(":/blocksmooth/Resources/simplifymerge.png")));

connect(mAction, SIGNAL(triggered()), this, SLOT(onActionTriggered()));

mQGisIface->blockEditToolbar()->addAction(mAction);

}

}

**keyPoint2:卸载在菜单中添加的菜单项**

void PLABlockSmoothTool::unload()

{

if(mAction)

{

disconnect(mAction, SIGNAL(triggered()), this, SLOT(onActionTriggered()));

mQGisIface->blockEditToolbar()->removeAction(mAction);

delete mAction;

mAction = 0;

}

}

KeyPoint3：QListWidgetItem存储自定义数据

//将数据添加进list

void PLABlockSmoothDialog::addFromLayer(QgsVectorLayer\* layer)

{

QListWidgetItem \*item = new QListWidgetItem(ui.lsLayers);

item->setData(Qt::UserRole, QVariant((qlonglong)layer));

item->setText(layer->name());

ui.lsLayers->addItem(item);

item->setCheckState(Qt::Checked);

}

//从list中将数据读取出来

QList<QgsVectorLayer\*> inputLayers;

for (int i = 0; i < ui.lsLayers->count(); i++)

{

if(ui.lsLayers->item(i)->checkState() != Qt::Checked)

{

continue;

}

QgsVectorLayer\* vlayer =(QgsVectorLayer\*)ui.lsLayers->item(i)->data(Qt::UserRole).toLongLong();

if(vlayer == 0)

{

continue;

}

inputLayers.append(vlayer);

}