Rapid-I vision measuring System Project

Internship activity - <Day 36>

Author:Yeswanthram R

**Clear All Properties:**

void UCS::clearAll()

{

try

{

this->getShapes().deleteAll();

this->getDimensions().deleteAll();

this->getIPManager().getIsectPts()->deleteAll();

this->getShapes().getSelected().clearAll();

this->getDimensions().getSelected().clearAll();

this->RelatedUCSChangeAction->clearAll();

this->PPshapesCollection\_Original->deleteAll();

this->PPshapesCollection\_ForPath->deleteAll();

this->PPMeasureCollection\_Original->deleteAll();

this->PPMeasureCollection\_ForPath->deleteAll();

}

catch(...){ MAINDllOBJECT->SetAndRaiseErrorMessage("RUCS0004", \_\_FILE\_\_, \_\_FUNCSIG\_\_); }

}



**UCS projection type properties:**

void UCS::SetUCSProjectionType(int i)

{

try

{

UCSProjectionType(i);

if(this->shapes.count() >= 3)

{

if(i == 0) //XY

{

X\_AxisLine->setModifiedName("X Axis");

Y\_AxisLine->setModifiedName("Y Axis");

Z\_AxisLine->setModifiedName("Z Axis");

}

else if(i == 1) //YZ

{

X\_AxisLine->setModifiedName("Y Axis");

Y\_AxisLine->setModifiedName("Z Axis");

Z\_AxisLine->setModifiedName("X Axis");

}

else if(i == 2) //XZ

{

X\_AxisLine->setModifiedName("X Axis");

Y\_AxisLine->setModifiedName("Z Axis");

Z\_AxisLine->setModifiedName("Y Axis");

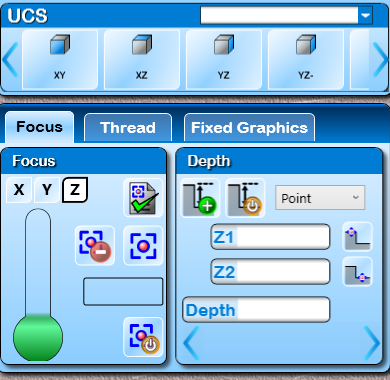
}

}

}

catch(...){ MAINDllOBJECT->SetAndRaiseErrorMessage("RUCS0005", \_\_FILE\_\_, \_\_FUNCSIG\_\_); }

}



**Un DO Action:**

void DeleteShapesAction::undo()

{

try

{

RCollectionBase& rshapes = MAINDllOBJECT->getShapesList();

MAINDllOBJECT->getShapesList().clearSelection();

for (RC\_ITER i = delshapes->getList().begin(); i!=delshapes->getList().end(); i++)

{

ShapeWithList\* Csh = (ShapeWithList\*)((\*i).second);

Csh->IsValid(true);

rshapes.addItem(Csh);

for(list<BaseItem\*>::iterator shapeiterator = Csh->getParents().begin();shapeiterator != Csh->getParents().end(); shapeiterator++)

{

((Shape\*)(\*shapeiterator))->addChild(Csh);

}

for each(BaseItem\* act in Csh->PointAtionList)

((AddPointAction\*)act)->ActionStatus(true);

if(Csh->selected())

MAINDllOBJECT->getSelectedShapesList().addItem(Csh, false);

for(list<BaseItem\*>::iterator shapeiterator = Csh->getChild().begin();shapeiterator != Csh->getChild().end(); shapeiterator++)

{

ShapeWithList\* Cshape = (ShapeWithList\*)(\*shapeiterator);

if(Cshape->ShapeType == RapidEnums::SHAPETYPE::CIRCLE)

{

if(((Circle\*)Cshape)->CircleType == RapidEnums::CIRCLETYPE::PCDCIRCLE)

{

((PCDCircle\*)Cshape)->AddParentShapes(Csh);

continue;

}

}

}

if(((Shape\*)((\*i).second))->ShapeType == RapidEnums::SHAPETYPE::DEPTHSHAPE)

FOCUSCALCOBJECT->DepthDatumList.push\_back((\*i).second->getId());

}



**Redo Action:**

bool DeleteShapesAction::redo()

{

try

{

RCollectionBase& selshapes = MAINDllOBJECT->getSelectedShapesList();

RCollectionBase& rshapes = MAINDllOBJECT->getShapesList();

while (selshapes.count() != 0)

{

RC\_ITER i = selshapes.getList().begin();

ShapeWithList \*CurrentShape = (ShapeWithList\*)((\*i).second);

CurrentShape->IsValid(false);

for each(BaseItem\* act in CurrentShape->PointAtionList)

((AddPointAction\*)act)->ActionStatus(false);

deleteRelatedShape((Shape\*)((\*i).second));

if(CurrentShape->ShapeType == RapidEnums::SHAPETYPE::DEPTHSHAPE)

FOCUSCALCOBJECT->DepthDatumList.remove(CurrentShape->getId());

delshapes->addItem((CurrentShape), false);

rshapes.removeItem(CurrentShape);

}

for(RC\_ITER It = delshapes->getList().begin(); It != delshapes->getList().end(); It++)

{

ShapeWithList \*csh = (ShapeWithList\*)(It->second);

for each(BaseItem\* act in csh->PointAtionList)

MAINDllOBJECT->getActionsHistoryList().notifyRemove(act);

if(csh->CurrentShapeAddAction != NULL)

MAINDllOBJECT->getActionsHistoryList().notifyRemove(csh->CurrentShapeAddAction);

}

for(RC\_ITER It = delMeasurement->getList().begin(); It != delMeasurement->getList().end(); It++)

{

DimBase \*cdim = (DimBase\*)(It->second);

if(cdim->CurrentDimAddAction != NULL)

MAINDllOBJECT->getActionsHistoryList().notifyRemove(cdim->CurrentDimAddAction);

}

return true;

}

catch(...){ MAINDllOBJECT->SetAndRaiseErrorMessage("ACTDSH0004", \_\_FILE\_\_, \_\_FUNCSIG\_\_); return false; }

}

**Set snap for ucs:**

//Initialise the snap point manager....//

void UCS::init()

{

try

{

UCSangle(0); UCSMode(0); UCSIntercept(0); UCSAngle\_B2S = 0;

UCSPoint.set(0, 0, 0); UCSPt\_B2S.set(0, 0, 0);

KeepGraphicsStraightMode(false);

currentDroX(0); currentDroY(0); currentDroZ(0); currentDroR(0);

RMATH2DOBJECT->LoadIdentityMatrix(&transform[0], 3);

isptmgr = new SnapPtManager(&shapes);

RelatedUCSChangeAction = new RCollectionBase(false);

PPshapesCollection\_Original = new RCollectionBase(false);

PPshapesCollection\_ForPath = new RCollectionBase(false);

PPMeasureCollection\_Original = new RCollectionBase(false);

PPMeasureCollection\_ForPath = new RCollectionBase(false);

AxisLine = NULL; CenterPt = NULL;

AddingFirstTime(true);

ParentUcsId(0);

ChildUcsId(0);

MarkingShape = NULL;

}

catch(...){ MAINDllOBJECT->SetAndRaiseErrorMessage("RUCS0003", \_\_FILE\_\_, \_\_FUNCSIG\_\_); }

}

