Rapid-I vision measuring System Project

Internship activity - <Day 36>

Author:Yeswanthram R

**Part program:**

**Abort error:**

void RWrapper::RW\_PartProgram::Abort\_PartProgram()

{

try

{

MAINDllOBJECT->changeHandler(RapidEnums::RAPIDHANDLERTYPE::DEFAULT\_HANDLER);

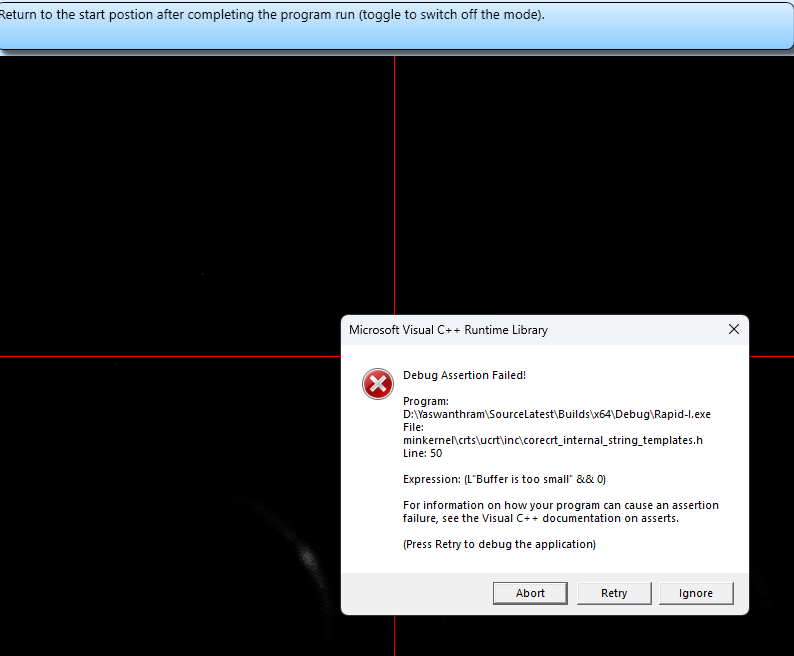
RWrapper::RW\_PartProgram::MYINSTANCE()->PartProgram\_Stop();

RunIsnotValid::raise();

}

catch(Exception^ ex){ RWrapper::RW\_MainInterface::MYINSTANCE()->WriteErrorLog("WRCNC0012", ex); }

}



**Camera Properties:**

void RCadApp::cameraFormatDialog()

{

try

{

if (CurrentCamera() == 3)

{

CameraClose();

bool succeeded = Initialise(NULL, 1, 102);

if (succeeded)

{

MAINDllOBJECT->CurrentCameraType = CurrentCamera();

if (HELPEROBJECT->CamSizeRatio == 1)

{

succeeded = SetVideoWindow(currCamWidth, currCamHeight, MAINDllOBJECT->VideoFrameRate(), true, this->CameraDistortionFactor, this->CamXOffSet, this->CamYOffSet);

}

else //We have to compress a larger video stream to show in a smaller area. So we call SetVideoWindowEx

{

succeeded = SetVideoWindowEx(currCamWidth, currCamHeight, 0, 0, currCamWidth / HELPEROBJECT->CamSizeRatio, currCamHeight / HELPEROBJECT->CamSizeRatio,

MAINDllOBJECT->VideoFrameRate(), true, this->CameraDistortionFactor, this->CamXOffSet, this->CamYOffSet, true);

}

if (succeeded)

{

//if (CurrentMahcineType == RapidEnums::RAPIDMACHINETYPE::HORIZONTAL || CurrentMahcineType == RapidEnums::RAPIDMACHINETYPE::HORIZONTAL\_DSP)

// FlipVertical();

Preview();

//Set\_ImBufferMode(true);

CameraConnected(true);

this->SetRCam3\_DigitalGain(this->RCam3\_D\_GainSettings);

MAINDllOBJECT->CameraReConnected(); // Reactivate lighting

Sleep(100);

}

else

CameraConnected(false);

}

else

CameraConnected(false);

}

else

GetCameraFormatDialog();

}

catch(...){ MAINDllOBJECT->SetAndRaiseErrorMessage("RCAD1202", \_\_FILE\_\_, \_\_FUNCSIG\_\_);}

}

****

**UCS point creation:**

void RWrapper::RW\_PartProgram::SetOffsetForRelativePoint(double offset\_X, double offset\_Y, double offset\_Z)

{

try

{

for(RC\_ITER Item\_ucs = MAINDllOBJECT->getUCSList().getList().begin(); Item\_ucs != MAINDllOBJECT->getUCSList().getList().end(); Item\_ucs++)

{

UCS\* Cucs = (UCS\*)(\*Item\_ucs).second;

for(RC\_ITER Item\_shape = Cucs->getShapes().getList().begin(); Item\_shape != Cucs->getShapes().getList().end(); Item\_shape++)

{

Shape\* Cshape = (Shape\*)(\*Item\_shape).second;

if(Cshape->Normalshape())

{

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

{

((RPoint\*)Cshape)->SetOffsets(offset\_X, offset\_Y, offset\_Z);

break;

}

}

}

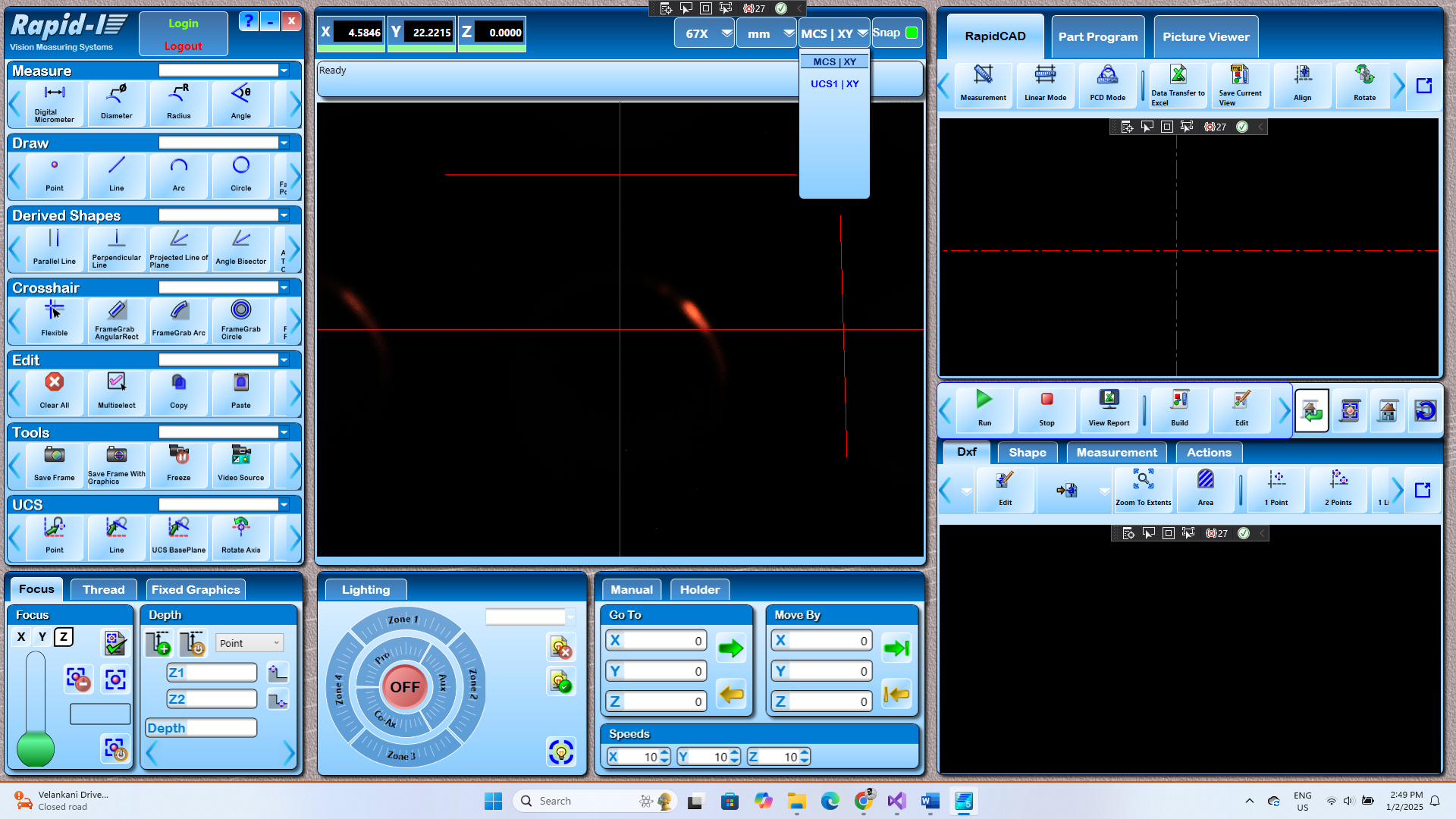
}

}

catch(Exception^ ex){ RWrapper::RW\_MainInterface::MYINSTANCE()->WriteErrorLog("WRPP0027", ex); }

}





**UCS Properties:**

else if(buffer == L"UCS")

{

if(isMCS)

isMCS = false;

else

{

UCS\* ucs = new UCS(\_T("pUCS"));

ifile >> (\*ucs); UCS& cucs = MAINDllOBJECT->getCurrentUCS();

ucs->getWindow(0).init(cucs.getWindow(0));

ucs->getWindow(1).init(cucs.getWindow(1));

ucs->getWindow(2).init(cucs.getWindow(2));

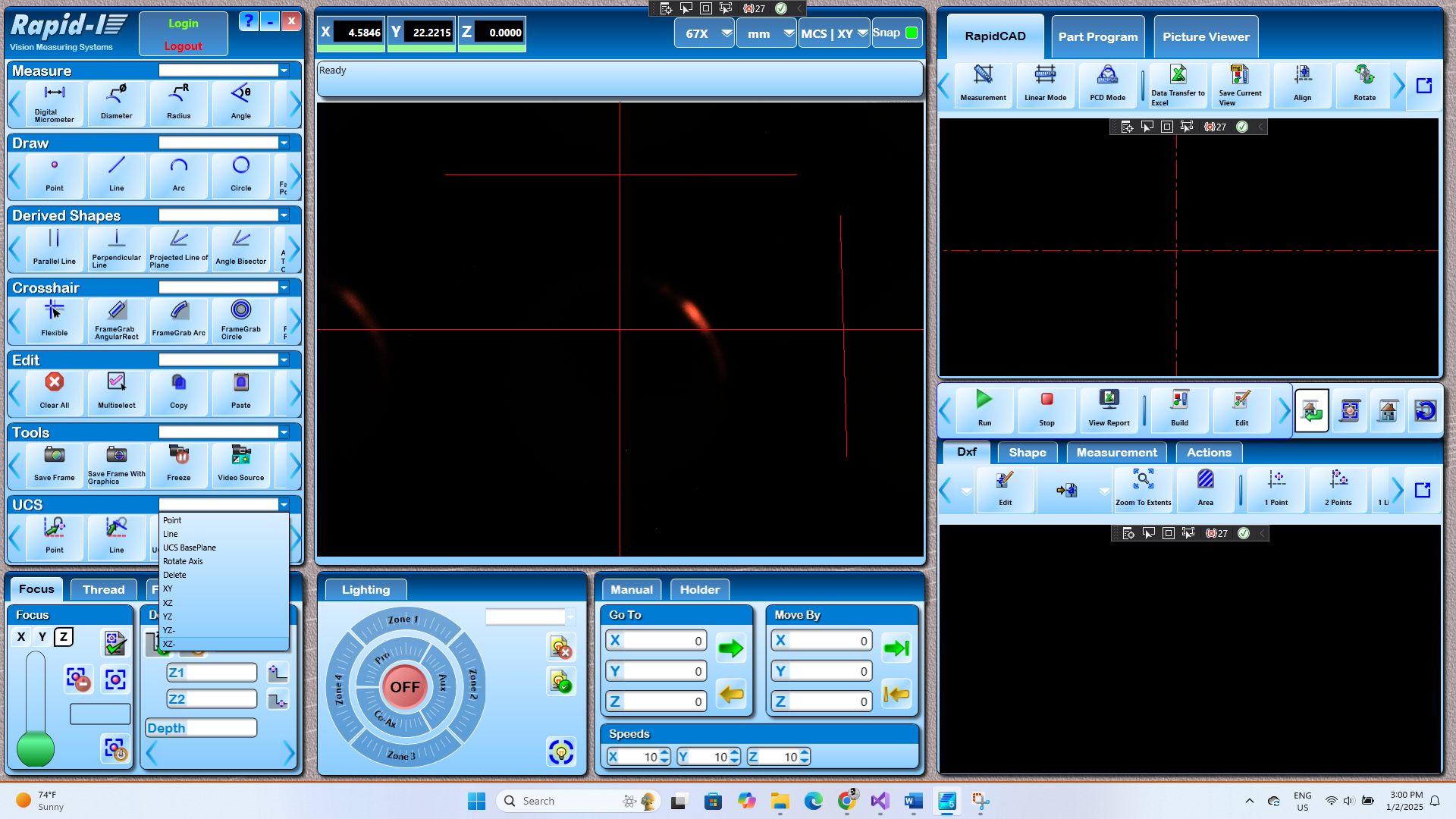
MAINDllOBJECT->AddingUCSModeFlag = true;

AddUCSAction::addUCS(ucs);

MAINDllOBJECT->AddingUCSModeFlag = false;

**}**

**}**

****