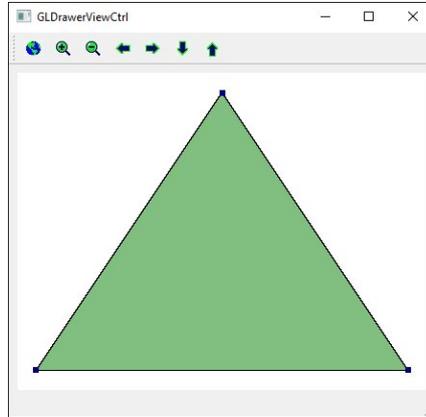


2º Trabalho: Visualização bidimensional com Qt e OpenGL

*Implementação de um programa gráfico simples para controle de visualização bidimensional*



Complemente o programa fornecido na *homepage* da disciplina para o segundo trabalho: arquivo `civ2802-241-trab2.zip`

Este programa em Python utiliza sistema de interface Qt e o sistema gráfico OpenGL.

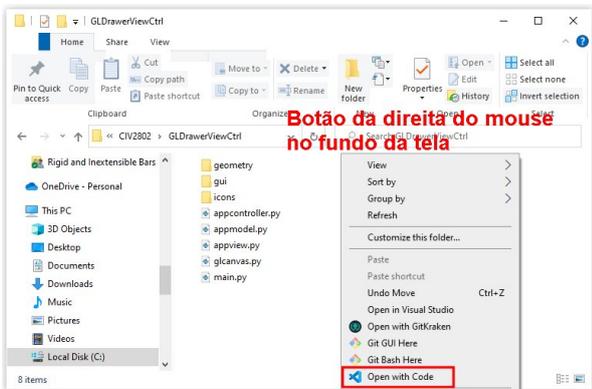
O entendimento dos arquivos e código do programa incompleto fornecido é parte do trabalho.

**Solicitado:**

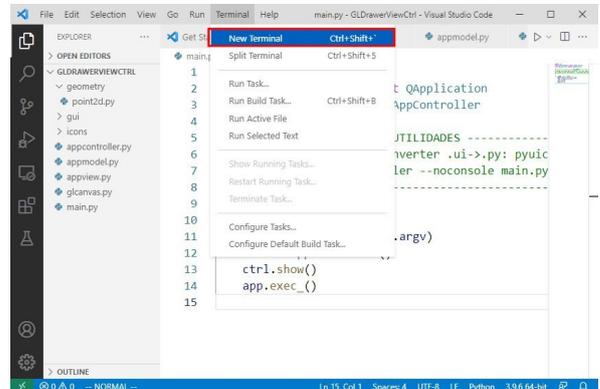
1. Seguindo o roteiro mostrado a partir da próxima página, criar a interface gráfica do programa utilizando o aplicativo Qt Designer. A interface deve conter uma barra de ferramentas (*tool bar*) superior com as seguintes ações:
  - Fit*: ajusta a imagem do objeto no canvas do programa.
  - Zoom in*: amplia a imagem do objeto no canvas.
  - Zoom out*: reduz a imagem do objeto no canvas.
  - Pan left*: move a imagem do objeto para a esquerda.
  - Pan right*: move a imagem do objeto para a direita.
  - Pan down*: move a imagem do objeto para baixo.
  - Pan up*: move a imagem do objeto para cima.Obrigatoriamente devem ser criados novos ícones (imagens) para essas ações. O arquivo criado com o Qt Designer deve ter o nome **myapp.ui**. Colocar esse arquivo em uma pasta com nome **gui**. Utilizando o comando `pyuic5 myapp.ui -o myapp.py`, criar o arquivo **myapp.py** na mesma pasta.
2. Completar as linhas de código no arquivo `glcanvas.py` que estão indicadas pelos comentários `##### COMPLETE HERE: GLCANVAS_XX #####`.
3. Enviar via e-mail em um arquivo `.zip` com os arquivos **myapp.ui**, **myapp.py**, e **glcanvas.py**, que correspondem à solução do trabalho, e uma pasta **icons** com os arquivos de imagens dos ícones criados para as ações de interface.
4. Enviar via e-mail um arquivo texto em PDF mostrando apenas os trechos de código que foram inseridos no arquivo `glcanvas.py`.

# Roteiro para criação da interface do programa 'GLDrawerViewCtrl' usando o aplicativo Qt Designer

## 1. Abra o Visual Studio Code

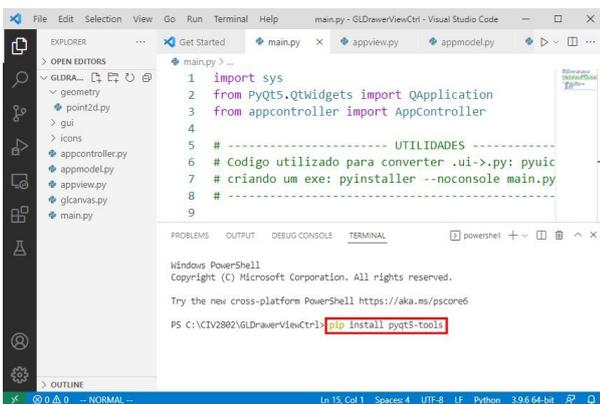


## 2. Abra um Terminal no Visual Studio Code



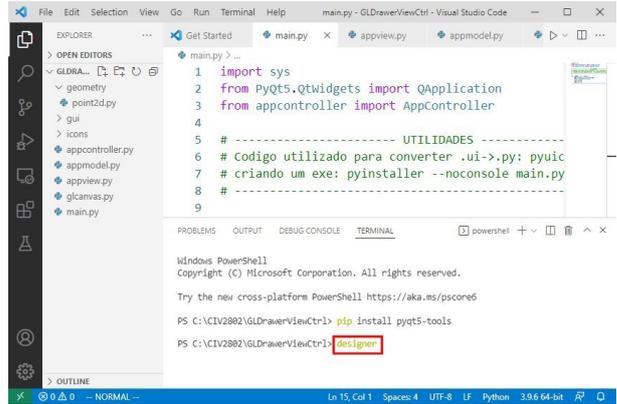
## 3. Instale o pacote PyQt5-tools

No Visual Studio Code:  
`pip install PyQt5-tools`

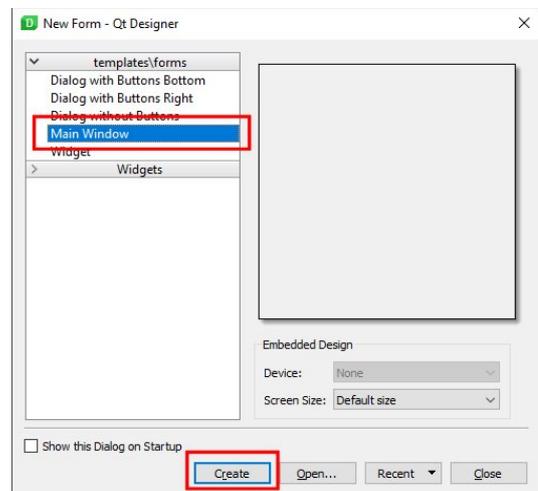
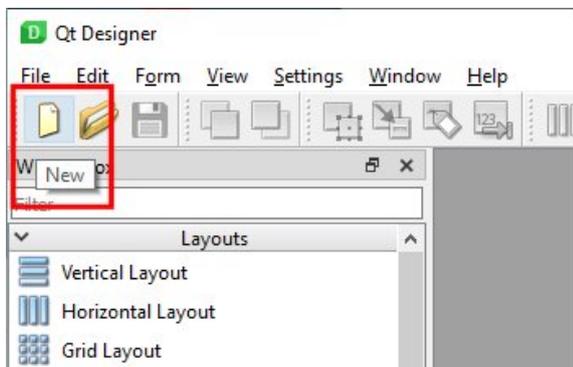


## 4. Abra o aplicativo Qt Designer

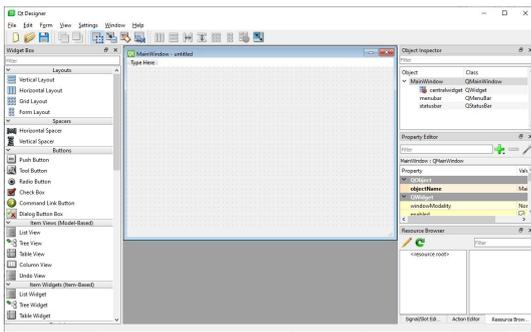
Antes adicione essa pasta no "path" do usuário:  
`C:\Users\\AppData\Local\Programs\Python\Python39\Lib\site-packages\qt5_applications\Qt\bin`



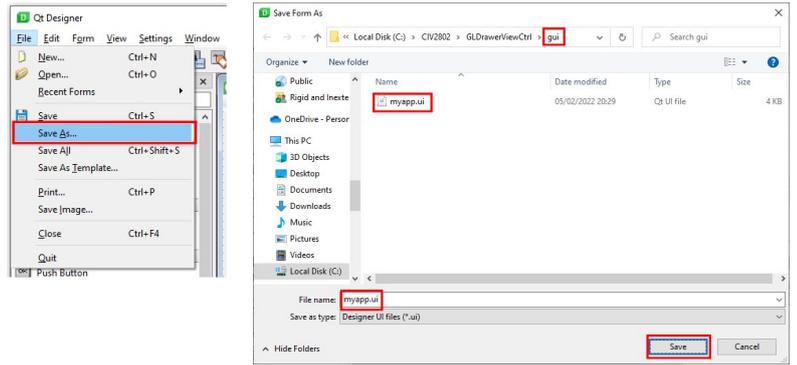
## 5. Execute o programa Qt Designer e crie uma nova Qt Main Window



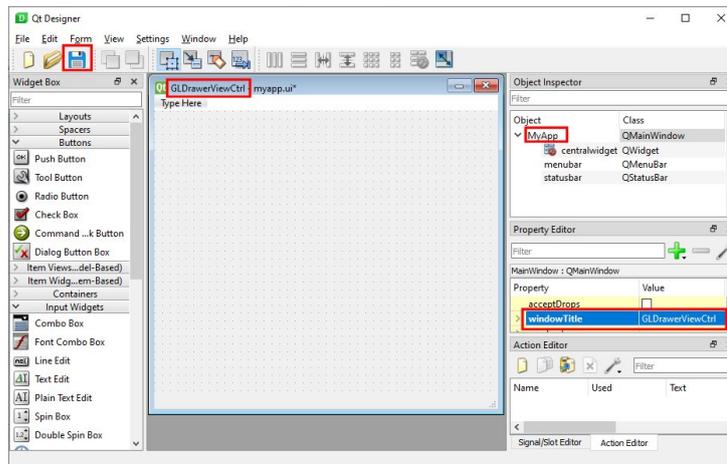
## 6. Pronto para criar interface gráfica



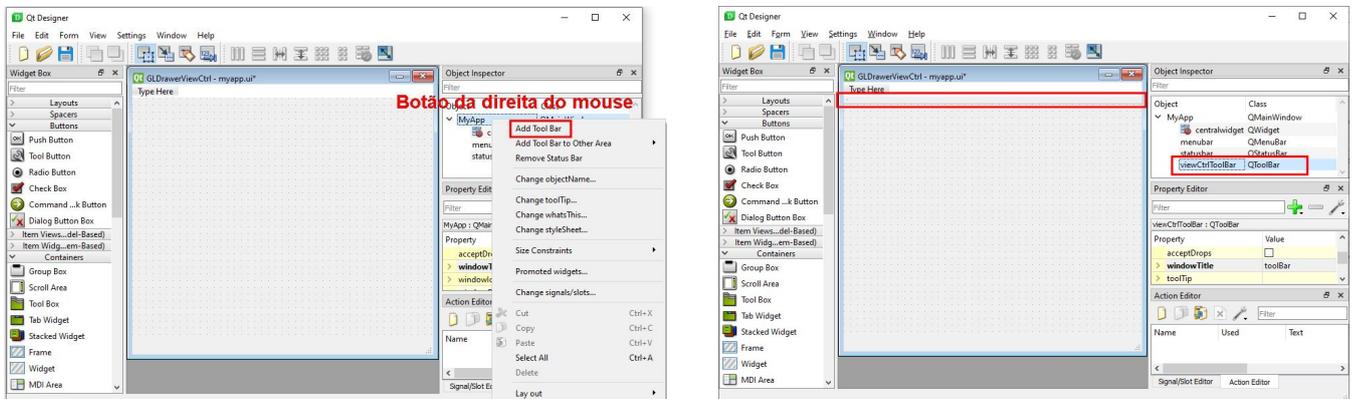
## 7. Salve arquivo com nome 'myapp.ui'



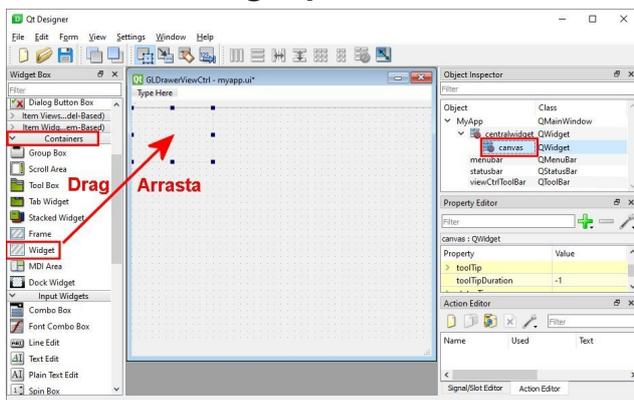
## 8. Muda o nome do objeto da janela principal e muda o seu título (salve)



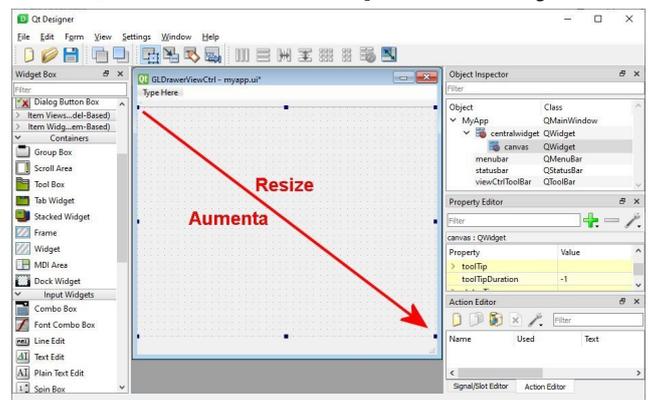
## 9. Adicione uma barra de ferramentas (tool bar) com nome 'viewCtrlToolBar'



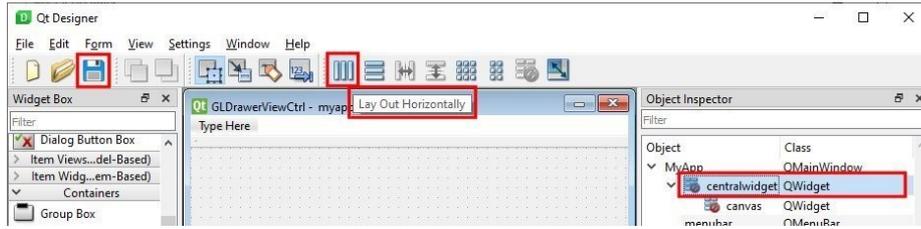
## 10. Crie um widget para o canvas



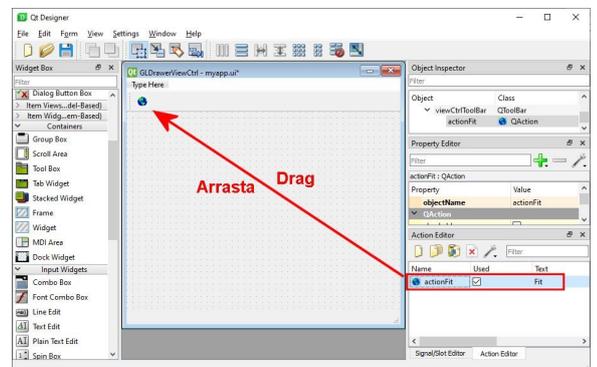
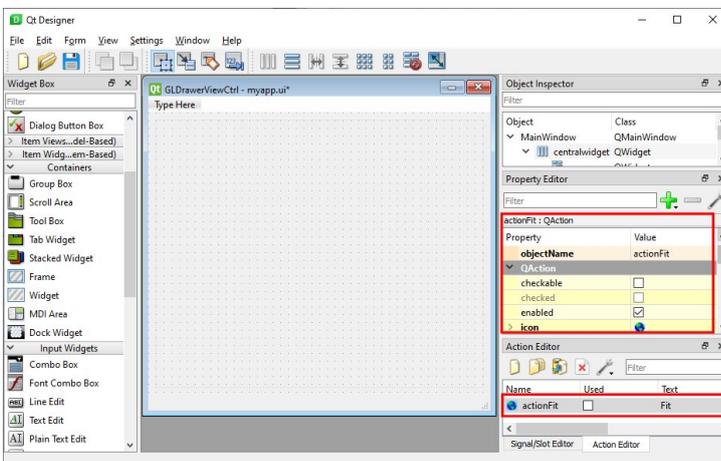
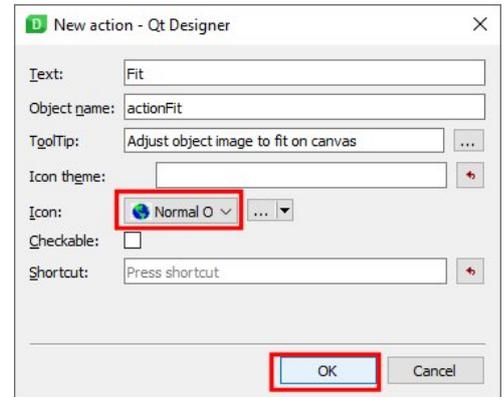
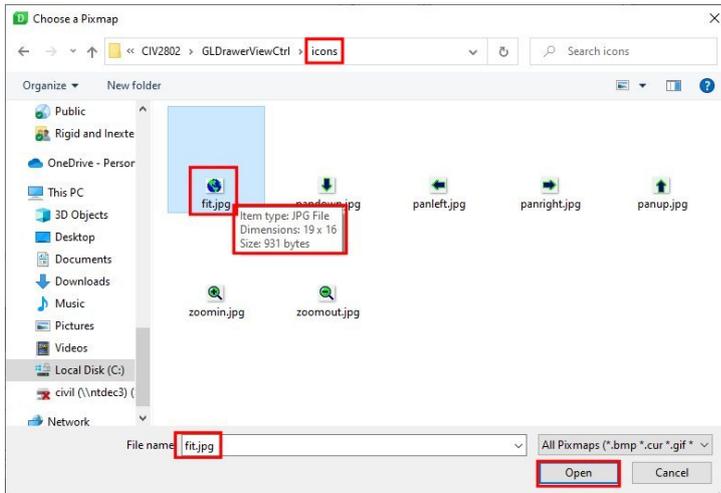
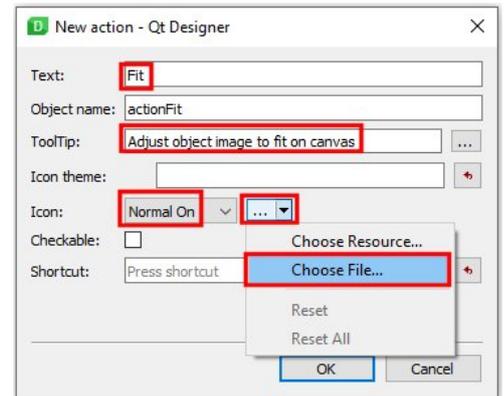
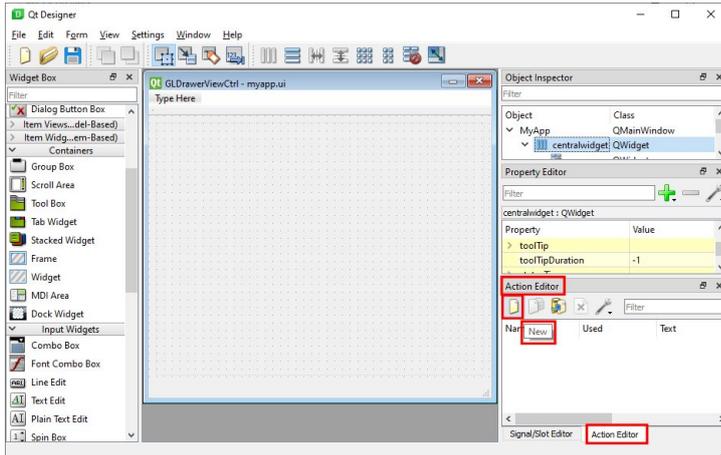
## 11. Faça o canvas ocupar toda a janela



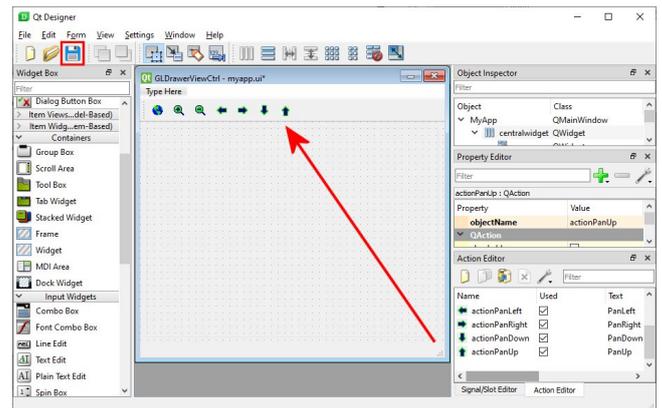
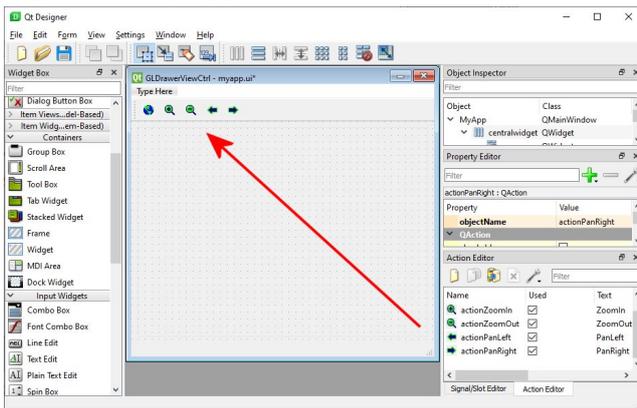
## 12. Force um layout horizontal para o centralWidget



## 13. Crie uma ação 'Fit' com ícone e insira na toolbar 'viewCtrlToolBar'



## 14. Crie as outras ações e insira na *tool bar* 'viewCtrlToolBar' e salve o arquivo 'myapp.ui'



## 15. Crie o arquivo 'myapp.py' na pasta 'gui' (não modifique esse arquivo)

cd gui

pyuic5 myapp.ui -o myapp.py

