

gui_unifiée/class_TabVTK.py

```
import os
import sys

#from PyQt5.QtCore import *
#from PyQt5.QtGui import *
#from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtWidgets import QApplication, QWidget, QGridLayout, QFrame,
QVBoxLayout, QScrollBar, QLabel, QGroupBox, QHBoxLayout

#import vtk
from vtk import vtkNamedColors, vtkRenderer, vtkPolyDataReader,
vtkPolyDataMapper, vtkActor
from vtk.qt.QVTKRenderWindowInteractor import QVTKRenderWindowInteractor

from class_Parametres import Parametres # Ne sert que si est exécuté
séparemment

"""
Classe TabVTK, hérite de la classe QGridLayout, c'est donc une grille
Cette classe représente le contenu d'une fenêtre PyQt
Elle peut donc aussi être utilisée comme un onglet dans une fenêtre
Basé sur ce script :
https://stackoverflow.com/questions/48105646/embedding-vtk-object-in-pyqt5-window
@author Amaury
"""
class TabVTK(QGridLayout) :
    """
    Constructeur, crée le contenu de l'onglet
    """
    def __init__(self, objParams, parent=None) :
        super(TabVTK, self).__init__(parent) # Appel du constructeur de
        QGridLayout

        self.objParams = objParams

        """ Divers """
        # Défilement temporel
        self.barreDeScrollTemps = QScrollBar(Qt.Horizontal)
        self.barreDeScrollTemps.setMaximum(
self.objParams.nombreInstantsTemporels() )
        self.barreDeScrollTemps.valueChanged.connect( self.dessinerVTK )
        self.addWidget( self.barreDeScrollTemps, 2, 1 )

        """ VTK """
        self.colors = vtkNamedColors() # Pour pouvoir mettre des couleurs

        self.frame = QFrame()
        self.vl = QVBoxLayout()
        self.vtkWidget = QVTKRenderWindowInteractor(self.frame)
        self.vl.addWidget(self.vtkWidget)

        # Ajout Maylis
        vertical_layout = QVBoxLayout()
        group_box=QGroupBox("Positions courantes des barres de scroll")
```

```

horizontal_layout = QHBoxLayout()
group_box.setLayout(horizontal_layout)
self.valeur_temps = QLabel("Temps : 0")
horizontal_layout.addWidget(self.valeur_temps)
vertical_layout.addWidget(group_box)
vertical_layout.addWidget(self.frame, stretch=2)
self.addLayout(vertical_layout,1,1)
# Fin Ajout Maylis

self.ren = vtkRenderer()
self.vtkWidget.GetRenderWindow().AddRenderer(self.ren)
self.iren = self.vtkWidget.GetRenderWindow().GetInteractor()

#self.actor = vtk.vtkActor()

self.actors = []
for instantTemporel in range(
int(self.objParams.nombreInstantsTemporels()) + 1 ) :
    fichierVTK = self.objParams.genererURLdesVTK( instantTemporel )

    if not os.path.isfile( fichierVTK ) :
        print ( "[Erreur TabVTK] " + fichierVTK + " n'existe pas !"
)

    else :
        print ( "[Info TabVTK] Chargement : " + fichierVTK)

        # Create source
        # Source :
https://lorensen.github.io/VTKExamples/site/Python/IO/ReadVTP/
        reader = vtkPolyDataReader()
        reader.SetFileName(fichierVTK)
        reader.Update()

        # Create a mapper
        mapper = vtkPolyDataMapper()
        mapper.SetInputConnection(reader.GetOutputPort())

        # Create an actor
        self.actors.append( vtkActor() )
        self.actors[instantTemporel].SetMapper(mapper)

self.actors[instantTemporel].GetProperty().SetColor(self.colors.GetColor3d(
'Tan')) # Couleur de l'objet 3D

self.actorPrecedent = self.barreDeScrollTemps.value()
self.dessinerVTK(0)

self.ren.SetBackground(self.colors.GetColor3d('White')) # Couleur
du fond
#     self.ren.ResetCamera()

self.frame.setLayout(self.vl)
#     self.addWidget(self.frame, 1, 1)

self.iren.Initialize()
self.iren.Start()

"""

```

```

Gère le dessin et les changements du VTK
"""
def dessinerVTK(self, value) :
    if self.actors != [] :
        instantTemporel = self.barreDeScrollTemps.value()
        print ( "[Info TabVTK] Affichage : " +
self.objParams.genererURLdesVTK( instantTemporel ) )

        # Ajout Maylis
        self.valeur_temps.setText("Temps : " + str(instantTemporel))
        # Fin Ajout Maylis

        self.ren.RemoveActor( self.actors[self.actorPrecedent] )
        self.ren.AddActor( self.actors[instantTemporel] )
        self.actorPrecedent = instantTemporel
        self.iren.Initialize()

"""
Code principal pour démonstration
"""
# Si on est le script principal
# Cela permet de ne pas exécuter ce bloc de codes lorsque ce script est
importé par un autre
# Source : https://stackoverflow.com/questions/419163/what-does-if-name-main-do
if __name__ == '__main__' :
    application = QApplication(sys.argv) # Crée un objet de type
QApplication (Doit être fait avant la fenêtre)
    fenetre = QWidget() # Crée un objet de type QWidget
    fenetre.setWindowTitle("MODE DÉMONSTRATION") # Définit le nom de la
fenêtre
    fenetre.setLayout( TabVTK( Parametres() ) )
    fenetre.show() # Affiche la fenêtre
    application.exec_() # Attendre que tout ce qui est en cours soit
exécuté

```