

gui_unifiée/class_Fenetre.py

```
import sys

#from PyQt5.QtCore import *
#from PyQt5.QtGui import *
#from PyQt5.QtWidgets import *
from PyQt5.QtCore import QSize
from PyQt5.QtWidgets import QApplication, QTabWidget, QWidget, QMessageBox

from class_TabGraphique3D import TabGraphique3D
from class_TabMilleFeuille3D import TabMilleFeuille3D
from class_TabMilleFeuilleIRM import TabMilleFeuilleIRM
from class_TabAffichageCoupes import TabAffichageCoupes
from class_TabVTK import TabVTK
from class_TabAide import TabAide
from class_TabBienvenue import TabBienvenue

from parametres import DISABLE_IRM
from class_Parametres import Parametres # Ne sert que si est exécuté
séparemment
from parametres_graph3D_pour_demo import grapheDeDemonstration # Ne sert
que si est exécuté séparemment

"""
Classe Fenetre, hérite de la classe QTabWidget (Et plus QWidget vu qu'on
veut faire des onglets)
Cette classe permet de gérer la fenêtre Qt avec onglets (Appel de la
procédure "addTab()")
@author Maylis et Amaury
"""
class Fenetre(QTabWidget) :
    """
    Constructeur
    """
    def __init__(self, objParams, parent=None) :
        super(Fenetre, self).__init__(parent) # Appel du constructeur de
        QTabWidget

        self.objParams = objParams

        # Taille minimale de la fenêtre, en pixels
        self.setMinimumSize( QSize(400, 400) )

        msgBox = QMessageBox()
        msgBox.setText("Voulez-vous activer l'affichage des VTKs ?\n(Il y
aura alors un temps de chargement plus long)")
        msgBox.setWindowTitle("Information")
        msgBox.setStandardButtons(QMessageBox.Yes | QMessageBox.No)
        returnValue = msgBox.exec()

        # Création des onglets de la fenêtre
        self.onglet1 = QWidget()
        self.onglet2 = QWidget()
        if not DISABLE_IRM : self.onglet3 = QWidget()
        self.onglet4 = QWidget()
        if returnValue == QMessageBox.Yes : self.onglet5 = QWidget()
        self.onglet6 = QWidget()
        self.onglet7 = QWidget()
```

```

# Ajout des onglets à la fenêtre
self.addTab( self.onglet7, "Bienvenue" )
self.addTab( self.onglet4, "Vues Coupes" )
self.addTab( self.onglet1, "Trajectoires" )
self.addTab( self.onglet2, "Vue Mille-feuilles" )
if not DISABLE_IRM : self.addTab( self.onglet3, "Vue IRM" )
if returnValue == QMessageBox.Yes : self.addTab( self.onglet5, "Vue
3D" )

self.addTab( self.onglet6, "Aide" )

# Création et enregistrement de l'objet de l'onglet du graphique 3D
# (Permet de l'appeler depuis n'importe où via l'objet Parametres)
self.objParams.TabGraphique3D = TabGraphique3D( objParams =
self.objParams )

# Remplissage des onglets en créant les grilles
self.onglet7.setLayout( TabBienvenue(objParams=self.objParams))
self.onglet4.setLayout( TabAffichageCoupes( objParams =
self.objParams ) )
self.onglet1.setLayout( self.objParams.TabGraphique3D )
self.onglet2.setLayout( TabMilleFeuille3D( objParams =
self.objParams ) )
if not DISABLE_IRM : self.onglet3.setLayout( TabMilleFeuilleIRM(
objParams = self.objParams ) )
if returnValue == QMessageBox.Yes : self.onglet5.setLayout( TabVTK(
objParams = self.objParams ) )
self.onglet6.setLayout( TabAide( objParams = self.objParams, noVTK
= (returnValue == QMessageBox.No) ) )

"""
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 = Fenetre( Parametres() ) # Crée un objet de type Fenetre
    fenetre.setWindowTitle("WoopWoop 3D (MODE DÉMONSTRATION)") # Définit le
nom de la fenêtre
    fenetre.show() # Affiche la fenêtre
    application.exec_() # Attendre que tout ce qui est en cours soit
exécuté

```