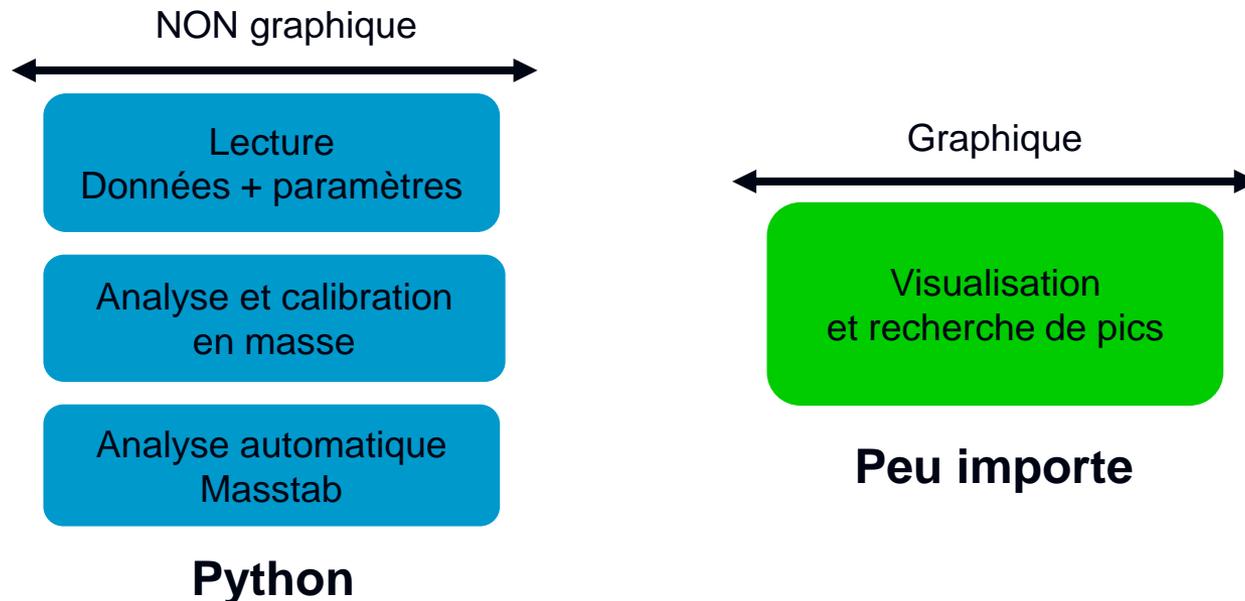


Retour d'expérience d'utilisation de PyQt4 – Python 3

Odile Cœur-Joly / projet PIRENEA

- Fonctionnalités de l'application
- Choix de développement
- Intégration IHM - Python
- Difficultés rencontrées
 - **Questions pour vous !!**
- Autres outils utilisés
 - Tests, performances, livraison, portabilité
 - **Encore des questions pour vous !!**

- Relire les données «historiques » de PIRENEA
- Calculer des spectres en masse à partir des signaux bruts
- Calibrer les spectres (fréquence – masse)
- Analyser graphiquement les spectres à la main
- Rechercher automatiquement des pics par masse : « masstab »



- Au sec....



- WinPython 3.4 (Windows XP, 7, 8.x)
 - Python 3.4
 - PyQt 4.8.6
- Eclipse Kepler
 - PyDev 3.9.2
 - Unittest + coverage
- IHM
 - Design automatique pour les widgets
 - Matplotlib
 - Zéro ligne à écrire en Qt ou C++

PIRANA

MassTab Viewer

Accuracy

Data Selector

Root: D:\PIRENEA_manips
 Year: 2006
 Month: 06
 Day: 07
 Folder: \PIRENEA_manips\2006\data_2006_06_07

Number	Acquis	Accum
001	A	00

Parameters

FILENAME:

Excitation

Dura. (ms)	Intens. (%)	Start (MHz)	End (MHz)
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Ejections

Dura. (ms)	Intens. (%)	Start (MHz)	End (MHz)
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Detection:

Analysis

Signal		
FILENAME: <input type="text"/>	Start Signal: <input type="text" value="0"/>	Stop Signal: <input type="text" value="0"/>
Step (µs): <input type="text" value="0.000"/>		
Max Points: <input type="text" value="0"/>		<input type="text" value="300000"/>
<input checked="" type="checkbox"/> Hann	<input type="checkbox"/> 1/2 Hann	<input type="checkbox"/> No Zero
	<input type="checkbox"/> Zero (1*N)	<input checked="" type="checkbox"/> Zero (2*N)

Mass Calibration	
Ref. mass (u):	<input type="text" value="300.09390"/>
Cycl. Freq. (kHz):	<input type="text" value="255.72700"/>
Mag. Freq (kHz):	<input type="text" value="0.001100"/>
Plot mass limits:	<input type="text" value="10.0"/> <input type="text" value="1000.0"/> <input type="checkbox"/> Hold

Peak Detection	
Peak Height:	<input type="text" value="0.040"/>
Start Mass:	<input type="text" value="290.0"/>
Peak Distance:	<input type="text" value="50"/> <input type="text" value="0"/>
End Mass:	<input type="text" value="310.0"/>
<input type="checkbox"/> Auto Update	<input type="button" value="Update Plots"/>

MassTab Selector

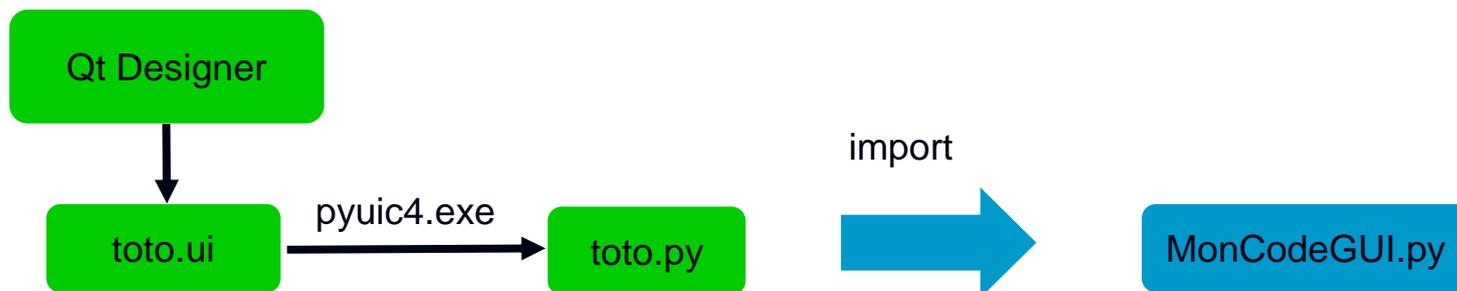
- 290
- 291
- 292
- 293
- 294
- 295
- 296
- 297
- 298
- 299
- 300
- 301
- 302
- 303
- 304
- 305
- 306
- 307
- 308
- 309
- 599
- 600
- 601

Signal

Signal

- WinPython = Python + modules scientifiques regroupés dans un installateur prêt à l'emploi
 - SIP (bindings Python en C++) non nécessaire
 - Compatibilité des modules déjà testée, pas de compilation
 - Aucune variable d'environnement nécessaire (désinstallation)
 - Plusieurs versions de WinPython peuvent cohabiter
 - Consoles python, IPython + Qt designer configurés
 - Ajout de packages supplémentaires via la console Python ou via un package manager (wppm)
 - ➔ Pas le choix de la version de PyQt associée à Python !
 - ➔ ex : PyQt5 prévue en 2015

- Principe des IHM avec PyQt
 - Design dans Qt Designer (clickodrome de course)
 - ➔ **génération d'un fichier XML (.ui) décrivant l'IHM**
 - Transformation du « .ui » en code Python « .py » (automatique)
 - ➔ **on ne touche pas à une ligne du code Python généré**
 - Import du module généré dans son code



The screenshot shows the Qt Designer application window with the following components:

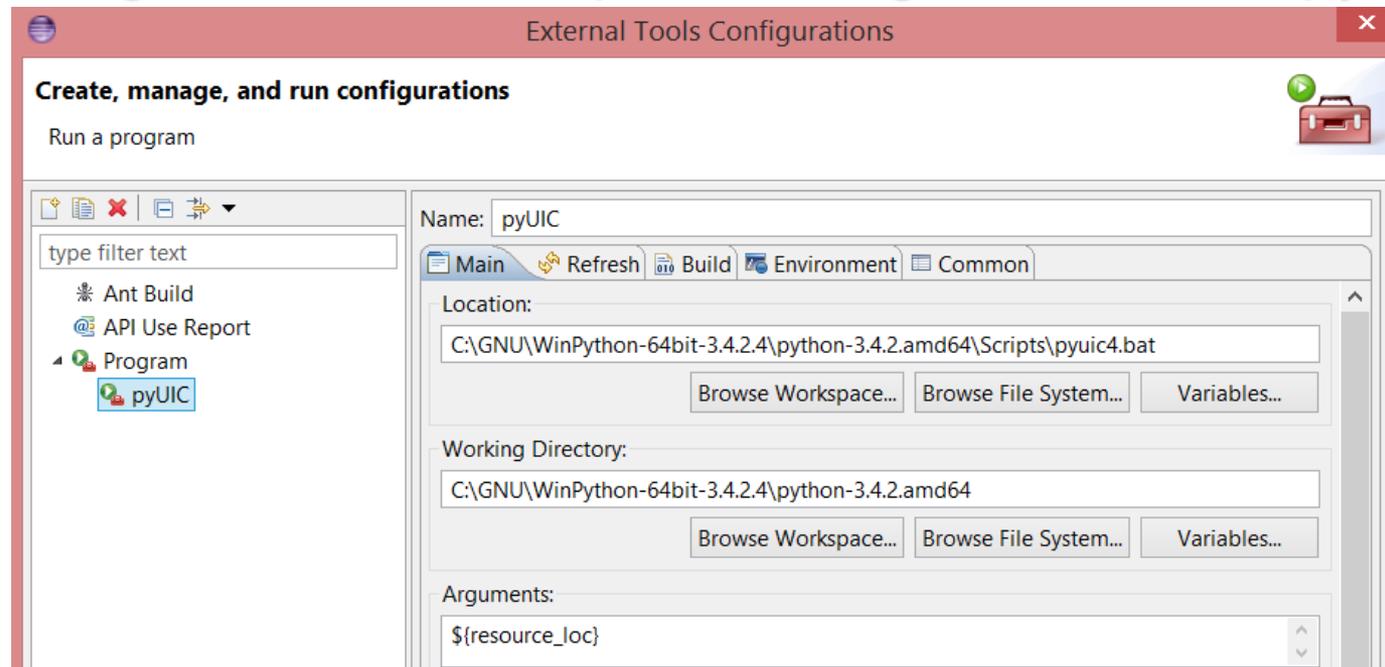
- Menu Bar:** Fichier, Edit, Formulaire, Affichage, Configuration, Fenêtre, Aide
- Boîte de widget (Widget Box):**
 - Layouts: Vertical Layout, Horizontal Layout, Grid Layout, Form Layout
 - Spacers: Horizontal Spacer, Vertical Spacer
 - Buttons: Push Button, Tool Button, Radio Button, Check Box, Command Link Button, Button Box
 - Item Views (Model-Based): List View, Tree View, Table View, Column View
 - Item Widgets (Item-Based): List Widget, Tree Widget, Table Widget
 - Containers: Group Box
- Data Selector - data_se Dialog:**
 - Root: D:\PIRENEA_manips
 - Year, Month, Day: dropdown menus
 - Folder: text input field
 - Number, Acquis, Accum: dropdown menus
 - Start Analysis: button
- Parameters - parameters Dialog:**
 - FILENAME: text input field
 - Excitation: text input field
 - Dura. (ms), Intens. (%), Start (MHz), End (MHz): dropdown menus
 - Ejections: dropdown menu
 - Dura. (ms), Intens. (%), Start (MHz), End (MHz): dropdown menus
 - Detection: dropdown menu
- Inspecteur d'objet (Object Inspector):**
 - Objet: DockWidget_Parameters
 - dockWidgetContents
 - groupBox_Detect
 - comboBox_DetectNam
- Éditeur de propriétés (Property Editor):**
 - Propriété: enabled, geometry
 - Valeur: checked, [(0, 0), 330 x...
 - Sub-properties: X (0), Y (0), Largeur (330)
- Explorateur de ressources (Resource Explorer):**
 - Filter: <resource ro...
 - Buttons: Éditeur de..., Éd..., Explorateur...

- En ligne de commande

```

C:\WINDOWS\system32\cmd.exe
d:\>
d:\>
d:\>pyuic4 -o toto.py toto.ui
  
```

- Intégration dans Eclipse : configuration « run pyuic »



- Nommage dans Qt

```
from PyQt4 import QtCore
from PyQt4 import QtGui
from toto import Ui_MainWindow

class MonCodeGUI(QtGui.QMainWindow):

    """ constructor """
    def __init__(self):
        super(MonCodeGUI, self).__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)

if __name__ == '__main__':
    import sys

    app = QtGui.QApplication(sys.argv)
    win = MonCodeGUI()
    win.show()
    sys.exit(app.exec_())
```

- Fenêtre principale

- Qt designer
 - Fenêtres indépendantes (type DockWidget)
 - Intégration de matplotlib comme widget
 - ➔ **widgets « promus »**
- Gestion des événements entre fenêtres
 - Constructeurs des widgets depuis une seule classe MainWindow
 - ➔ **initialisation des widgets fils non standard**
 - Communication entre widgets indépendants
 - ➔ **pyqtSignal pour générer des événements**

- Tests : unittest + coverage
 - ➔ **Tester la partie graphique ? Comment ?**
- Analyseur de code : Pylint, autopep8
- Performances : time.time()
 - ➔ **pas trop réussi avec « profile » : qui l'utilise ?**
- Distributions
 - Windows 7/8 : py2exe ➔ de 200 ko à 200 Mo
 - OS X : problème Python3 PyQt4
 - ➔ **VM Mountain Lion (Python2) ou VM Yosemite (PyQt5)**
 - ➔ **Existe-t-il une sorte de WinPython pour OS X ?**

- Restez cool...

