

1 Séance 1

1.1 Etudier la structure de base d'une application PyQt6 (sans évènement)

1.2 Intégrer de nouveaux éléments graphiques

1.3 Utiliser des signaux pour gérer des évènements

1.4 Ouvrir une image et l'afficher

1.5 Afficher un graphique (utilisation de *PyQtGraph*)

Temps conseillé : 60 min

A partir du fichier **graph_gui.py**

- Lancer l'application et visualiser le résultat. Chargez les données du fichier "**data.csv**"
- En utilisant la classe **RangeWidget**, faites apparaître sur le menu principal un panneau de réglage de la fenêtre graphique.
- Par défaut, les zones de texte ne sont pas éditables, même lorsque des données sont chargées. Modifiez le code pour activer les zones de texte une fois des données chargées.
(Indice : il faut modifier la méthode **action_signal()** de la classe **RangeWidget**)
- Maintenant, les zones de textes sont éditables mais elles n'ont pas d'effet sur le graphe. Définissez une nouvelle méthode "**set_lim**" dans la classe **RangeWidget** permettant de mettre à jour les limites des axes du graphique pour les valeurs indiquées dans les zones de texte. Ensuite, il faut que cette méthode soit appelée à chaque fois que les zones de texte sont modifiées.
(Indice : utilisation de la méthode **setRange** et du signal **QLineEdit::textChanged**)
- En utilisant la classe **QLineEdit**, ajoutez dans le menu principal une zone de texte qui permet de modifier le titre du graphique.
- Enfin, ajoutez dans le menu principal un bouton permettant d'enregistrer le graphe au format .jpg. On pourra utiliser les deux lignes de code suivante :
exporter = exporters.ImageExporter(self.plot_graph.plotItem)
exporter.export(self.title + '.png')