Install environment

PySide2 installation on windows :

* Install Python : version 3. 9.1 (min 2.6)
* Ajouter les variables d’environnement de python : aller dans Panneau de configuration\Système et sécurité\Système, puis Paramètres Système Avancés, et ajouter à la variable Path le chemin vers l’installation de Python (emplacement de Python.exe, on peut le retrouver en faisant « where Python » dans un terminal).
* On peut aussi ajouter à la variable d’environnement Path le chemin vers pip pour installer Pyside2.
* Ensuite on peut simplement installer Pyside2 avec : pip install Pyside2.

Comprendre Pyside2 VS Qt

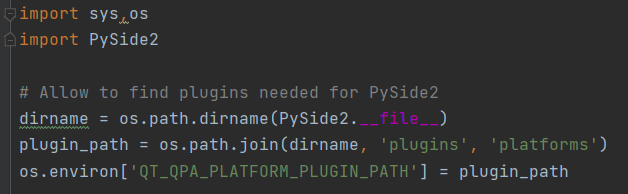
* Pyside2 semble proposer une alternative à la configuration de Qt
* Without going too much into the details of the different license models involved, if you want to develop a commercial application, PyQt requires you to pay fees for a commercial license, while the LGPL license of PySide permits application in commercial projects.

Setup PyCharm :

* On créer un projet, avec un fichier main pour notre application.
* Dans build, on ajoute une configuration de build : il faut au moins spécifier le fichier main de l’application et l’interpreteur à utiliser (la version de Python par exemple).
* Si aucun interpreter n’est configurer, il est possible d’ajouter une configuration dans File > Settings > Project : […] > Python Interpreter.

Gestion des dépendances PySide2

* Comme pour Qt, il faut gérer un certain nombre de dépendance pour que les exécutables fonctionnes.
* Avec Qt il y a plusieurs solutions : aller chercher les dll présent dans C:/path/qt/version/bin et les mettre à côté de l’exécutable, ajouter un fichier qt.conf avec l’exécutable pour indiquer le chemin vers les dlls (et dossier plugins), ajouter la variable d’environnement QT\_PLUGIN\_PATH qui pointe vers le dossier plugins (je pense).
* Pour Pyside2 il y a plusieurs solutions également.
* L’une est d’ajouter ses éléments pour laisser l’os retrouver les dépendances dans le fichier main :



Les différentes parties du framework :

* QWidget : Contient des éléments visuels déjà implémenter, que l’on peut utiliser tel quel, paramétrer ou hériter pour créer des éléments personnaliser. Cela comprend un très grand nombre de widget : des boutons, des grilles, des layouts, des calendiers, des sliders, etc.
* QGui : Contient dans outils qui permettent de gérer des élements relatif à la GUI. Cela contient des outils pour gérer les icones, les fonts, les images, les fichiers, les sons, etc.
* QCore : Contient des fonctionnalités Core liés à la machine et la configuration de Qt. On y retrouve des outils relatifs au temps, des enum de qt, etc.

Documentation :

* Pour la documentation de Pyside2, j’ai l’impression que l’on peut se baser sur la doc de PyQt.

Liaison avec un base de données :

* Pour jouer avec des données en ligne, il suffit d’utiliser une lib permettant de faire des requête (en SQL par exemple), puis injecter les données dans les éléments PySide.