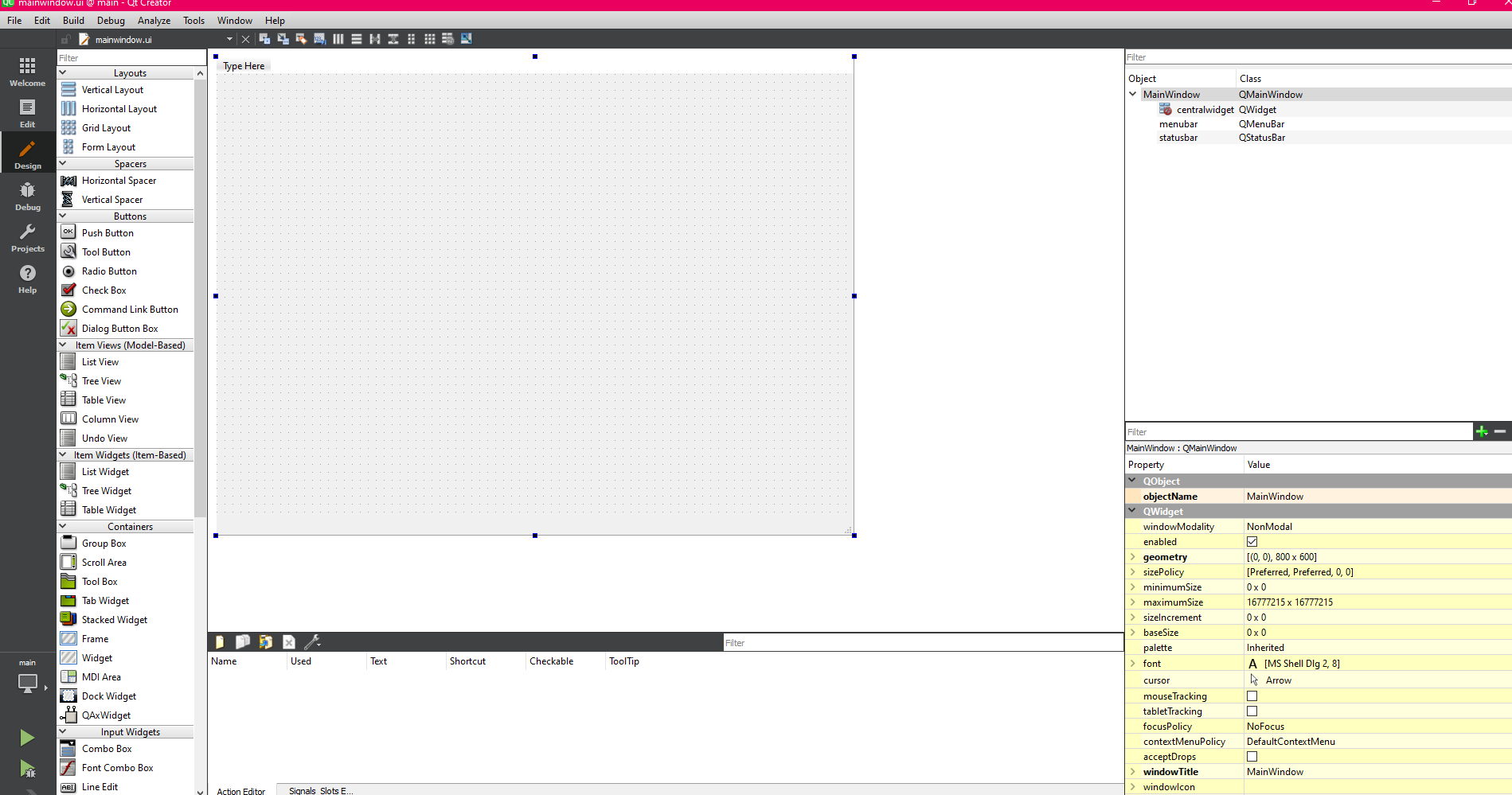
**GUI Development using Qt - C++ Walkthrough**

**Background:** Qt (“Cute”) is a software program used primarily for the development of Graphic User Interfaces that allow for easy use.

In general, there are three files we are concerned with in regards to the GUI: .pro, .cpp and .ui. Of course, depending on the desired functions, your header files and source files are also of great importance. The .py file has the code portion of the GUI which can be used in the main Python Source code. The .ui QDesigner form allows you to directly design the layout of your GUI with the use of drag and drop widgets. Once the .ui file is created, it must be converted directly into code via video link **3.** The GUI itself can be firstly coded in the .cpp file or be designed (layout only) in the .ui form then converted. Once converted, the actual design of the functionality portion of the GUI begins.

**QDesigner/ .ui form (below):**

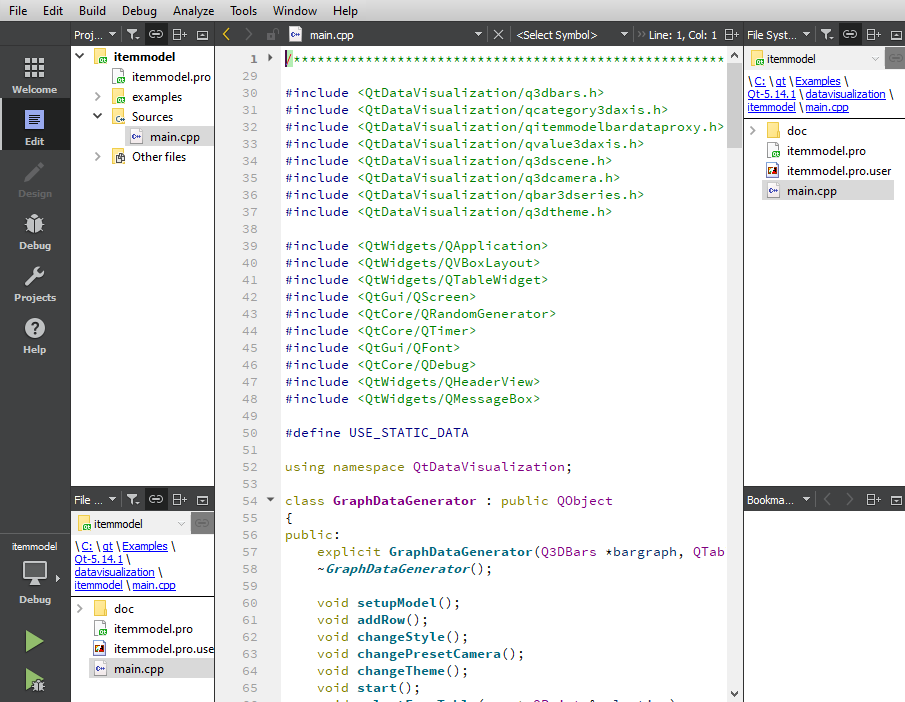
**Left:** Widgets **Middle:** Canvas layout **Top Right:** Object Identifier **Bottom Right:** Property Manager



In the .ui form, you can assign buttons, windows, tables and other widgets to your canvas. By dragging and dropping a widget onto the canvas, an object related to that widget is then created and shown in the property manager when said widget is selected. Be sure to identify the specific object names for all GUI components as they must be known when designing the actual functions of the GUI related to the desired functions of the project.

**.cpp File screen (below):**

**Left:** Project File Selection **Middle:** Editor **Right:** Project Directory

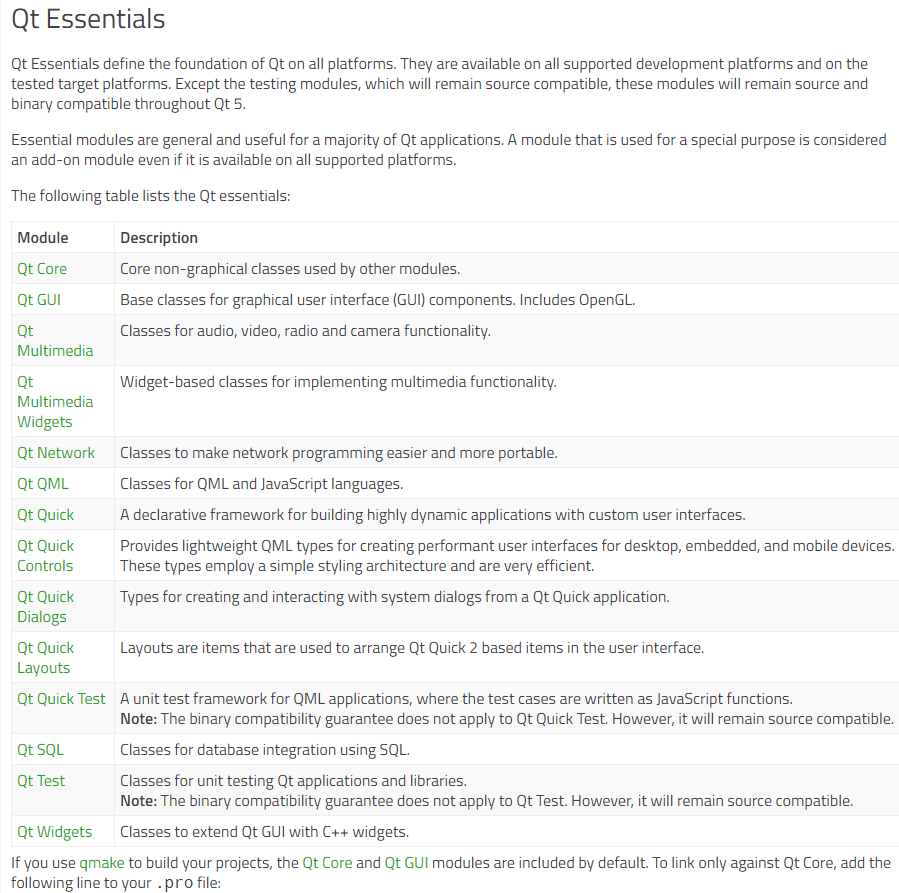
The .cpp file is the main source file (code) for the C++ GUI. All widgets will appear here as functions and can only be used when the correct modules are imported. Once the .ui form is completed, the .cpp file will be populated with all the functions that represent the widgets on the .ui file (GUI layout). The .cpp file is like any other source file, so insert all code related to the functionality requirements of the GUI here.

**.pro File screen (not shown):** Not a concern for now since we are not developing a standalone application.

**GUI Requirements:**

* **Real-time data feed of Coordinate data**
* **Populate real-time data into Table widget**
* **Activate/Deactivate main code using button widgets**
* **Save data files to specific file directories**
* **Implement Graph widgets**

**Modules**



**GUI Main Functions**

1. **Reading.CSV File containing Coordinate Data and populating Table:** 
   1. Import QTCore to use buttons and their connectivity to functions via signal slots architecture.

**Qt- Resources:**

1. **General C++ Reference- Homepage:**[**https://doc.qt.io/qt-5/reference-overview.html**](https://doc.qt.io/qt-5/reference-overview.html)
2. **Q&A Forum:** [**https://forum.qt.io/**](https://urldefense.com/v3/__https:/forum.qt.io/__;!!DPL28cuj!03WU1LqFFNjKjwWM0jQBHqu6g4vpEzSfMaJ2kdrqrt8pTI7AYGp-iA11EbUBYmAoDv-L$)
3. **Signal and Slots Documentation:**

<https://doc.qt.io/qt-5/signalsandslots.html>

**Video Sources:**

1. **Open Source Options Youtube page - QCustomPlot:** [**https://www.youtube.com/watch?v=5ENvNtYhoPM**](https://www.youtube.com/watch?v=5ENvNtYhoPM)