Coding Standards

Format Standards:

ClassesAreNamedLikeThis
IInterfacesHaveAnIInThem
methodsAreNamedLikeThis()
variablesAreNamedLikeThis
m_memberVariablesAreNamedLikeThis
NamespacesAreDefinedLikeThis
ConstructorsShouldFollowClassName
signalsAndSlotsAreNamedLikeThis()
widgetItemNames should follow camelCase and be descriptive to their function

Indenting & Whitespace:

Curly brackets should include a line space, with one being on the method name line
Any code within curly brackets should be indented one tab higher than the curly braces

```
returnType someMethodName() {
        /*Method Definition*/
}
```

Variables:

Variable names should present a clear message as to what it is and what it affects. There should not be any single letter variables except in iterations like for loops. All variables should be camelCase. Only declare variables in the most limited scope possible. Member variables should start with m_. Member variables should be protected or private. Initialize all local variables at declaration. Initialize all member variables in the class constructor. Try to use class initialization lists rather than in the constructor.

Methods:

Methods returning bool should begin with ‘is’. Methods that search for some sort of data should start with ‘find’.

Coding Conventions:

Duplicate code should be removed / refactored
All class and variable names should be descriptive but not too long
Member variables should be protected or private

Comments:

Comments should always be used for method descriptions in a doxygen format. Variable and method names should be a sufficient replacement to commenting. Only very difficult to understand code should be commented. Comments should be reviewed by at least one other team member to ensure quality commenting is used.

Files:

Header files’ #ifndef are formatted like this: FILENAME_H. All header and source files are placed in “Headers” and “Sources” folders respectively. Each class should have its own description and authors. Each file should only have one class defined within it.

Etc:

Each separated group should only modify part of the file that are relevant to their task, if they were to modify part of the code that will affect other tasks as well it should be discussed first.