$Q|SI\rangle$ Cheat Sheet

Shusen Liu mailto:shusen.liu88@gmail.com

1 Setup

Choose a suitable setup file to download.

1. `QSI_Online_Setup` is a small setup file that uses NuGet to retrieve all the required Visual Studio libraries. It also includes installation packages of Visual Studio 2017 and MATLAB Runtime 2017 on the condition that you do NOT have installed VS 2017 and MATLAB currently installed. These packages require an internet connection to complete installation.

2. `QSI_Offline_Basic_Setup` is a medium-sized setup file equipped with all the essential Visual Studio libraries. It also includes inductive packages of Visual Studio 2017 and MATLAB Runtime 2017 on the condition that you do NOT have Visual Studio 2017 and MATLAB currently installed. These packages require an internet connection to complete installation.

3. `QSI_Offline_Full_Setup` is a huge setup file equipped with all the essential VS libraries and Visual Studio 2017 Full version and MATLAB Runtime 2017. The packages do not require an internet.

Quick Reference Table:

Internet	VS& MATLAB	Choice
Good	Installed&Not Installed	`QSI_Online_Setup`
Average	Installed & Not Installed	`QSI_Offline_Basic_Setup`
None	Installed	`QSI_Offline_Basic_Setup`
None	Not Installed	`QSI_Offline_Full_Setup`

2 Folder Structure

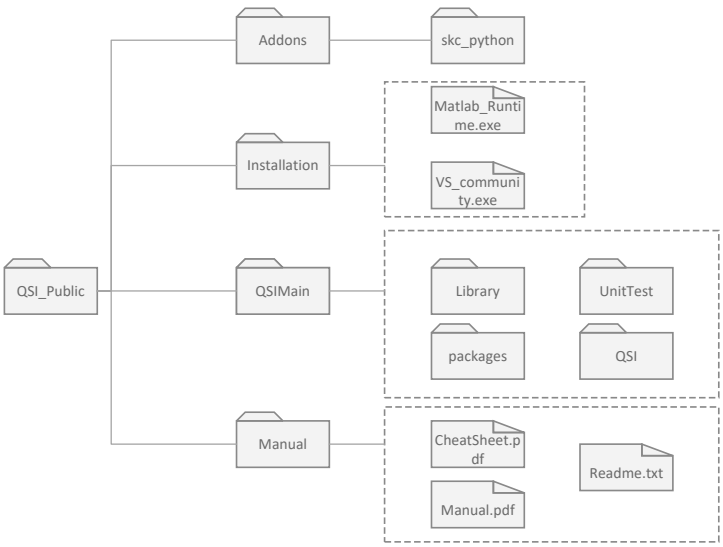

Figure 1: Installation Folder Structure

After installation, a folder called QSI_Public is created on your desktop by default. Figure 1 shows the folder structure.

1. The `Addons` folder contains advanced features, such as the Solovay-Kitaev decomposition component.

2. The `Installation` folder contains two essential software installation packages. Install them as needed.

3. The `QSIMain` folder contains the main programming environment files. `UnitTest`, which includes prepared set of examples; and `QSI`, which is your working folder.

4. The `Manual` folder contains the technical manuals.

3 Start

Note that the examples are not provided in a unified pure second generation language. Some code has been programmed using the engine layer (basic) language for testing and debugging reasons. These code segments are only for study and preview purposes.

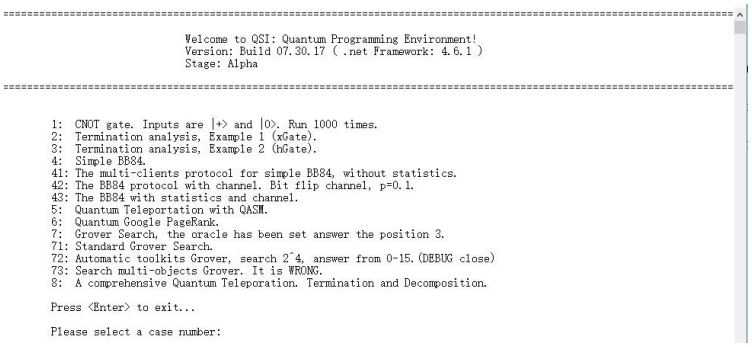

Figure 2: Welcome to $Q|SI\rangle$

1. Ensure the system environment has been installed correctly. The main program, Visual Studio and MATLAB Runtime should be installed following the procedure in the Programming Manual.

2. Menu → Debug → Start Debugging (or press F5).

3. Wait a few seconds. (It may take more than 2 minutes to build the project the first time it is executed, depending on your hardware.)

4. Eureka! You should see a display similar to Figure 2

There are two projects in the environment: QSI and UnitTest.

Project	Purpose	Use
QSI	User coding area	Write your quantum code in Test.cs. Write your classical control code in Program.cs
UnitTest	Examples	Study exercises and languages

4 Examples

The examples can be found in the UnitTest folder. All the examples can be traced to their corresponding source files.

Example	Function	Explanation
CNOT gates	Generate the Bell state	Uses quantum language to construct the algorithm
xGate	Termination analysis	xGate, termination
hGate	Termination analysis	hGate, almost sure termination
BB84	Protocol simulation	Finds the behavior of BB84 in quantum channels
Teleportation	Using QIf & QWhile	Study of the most powerful structures
Grover	Engine Layer Usage	Uses the engine layer to enrich flexible programming

5 Key Functions

You should start with QSI default and UnitTest. You may need other advanced features to customize the environment. Please use QSI as a template and configure it with classical codes. The key functions are provided in Section 3 of the Manual.pdf; they can be used to adjust your program.