QR Code Tactical Decision Aid

Quick Start

Mike Bailey

Moves Institute

Naval Postgraduate School

11 Oct 2017

##### Deployment

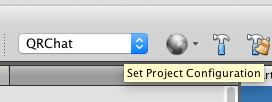
A normal "deployment" package of this software is a zip file. Such a file is produced after a project clean-build and is placed in **target/Sailor.zip**. This file does not go into the Git repository. However, the contents of this file are placed in **currentDistribution/SailorZip**, which is there. To create the equivalent of Sailor.zip manually, zip-compress that directory.

##### Architecture

The architecture of this software is described in detail in Project.docx, which should reside next to this document in the file tree on disk and in the source repository. This document describes how to run various configurations from within the Netbeans IDE, which is the quickest way to get the software going without having to export a jar, etc.

The QR Code Tactical Decision Aid, or QRTDA, is modular. It contains various components which, when assembled, combine to perform overall tasks. Many configurations have been put together. Each is controlled by a configuration file.

The configurations are normally used in one of two ways:

1. Through the Set Project Configuration selector at the top of the Netbeans IDE window:
2. Through passing the name of a configuration file to the application when it is launched from the command line. This method is described in Project.docx, not here.

To launch the application in its various forms, set the desired configuration through the selector, the run the application in the normal way within Netbeans.

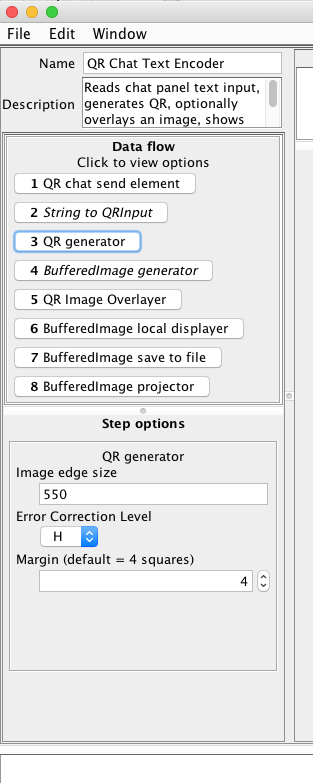
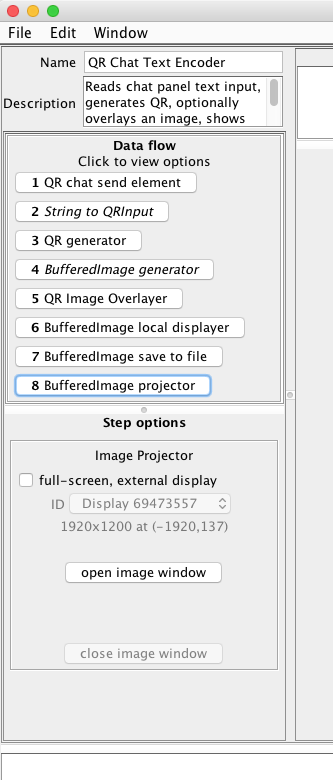
Some of the different configuration are listed here.

**QR Chat**

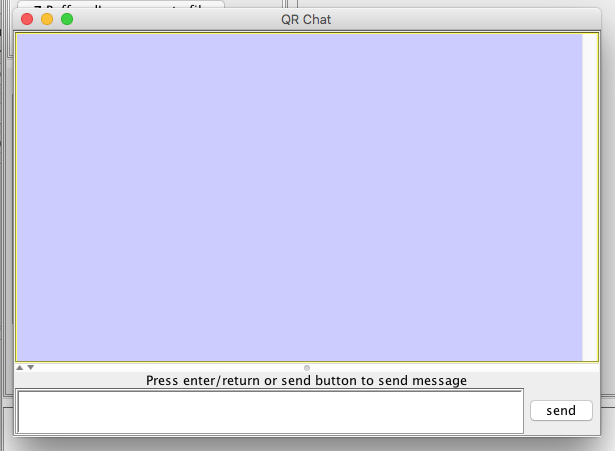
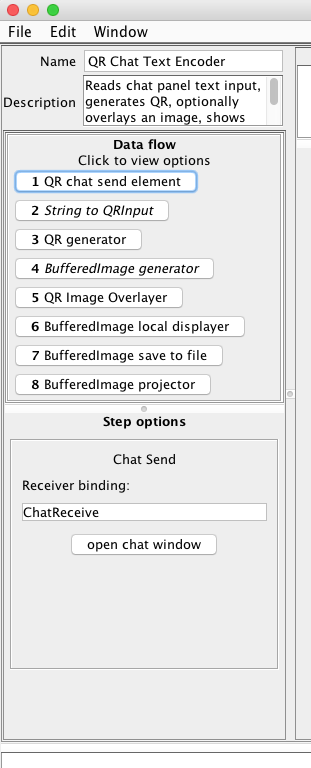
Opens 2 application windows: one to accept keyboard entry and display a corresponding QR image on a screen for remote reading; and one to read the images from the attached camera to decode incoming messages.

The first window, “QR Chat Text Encoder”, displays the 8 components that make up the send configuration. “BufferedImageProjector” is important to either open a draggable window (which can be moved to an external display) or use an external display in exclusive, full screen mode.

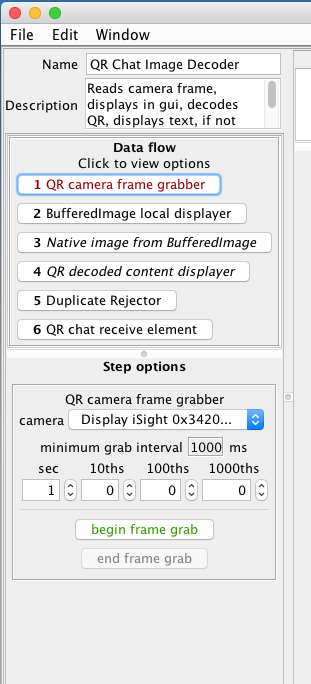
The “QR generator” component controls 3 QR image parameters.

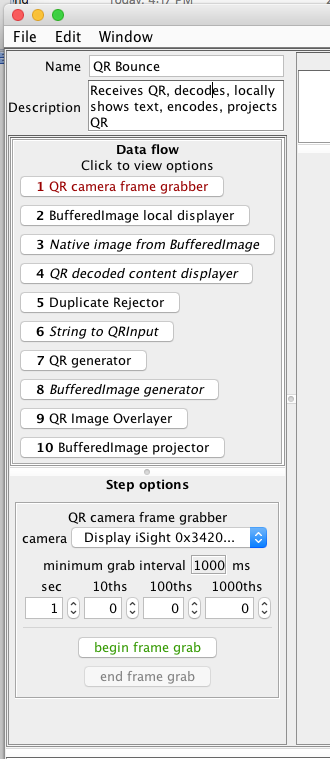


The “QR chat send element” allows you to open a separate chat window to enter text to be sent and display received text. The chat window looks like below:



The “QR Chat Image Decoder” window, the second half of the QR Chat application has an important button to begin “frame grabs” from the camera. You find it by clicking on the “QR camera frame grabber” component. You choose which attached camera to use and the sleep interval between frame grabs.



**QRBounce**

This is a configuration to simply receive camera input, decode it, re-encode it, then display the resulting image on the attached display. The parameters are similar to QRChat. Note: If two QRTDA instances are communicating and both are using QRBounce, there is no way to send the initial image. If there were, you would see the same data going back and forth. Typically one side would be a QRChat ThroughputTest.

**Other configurations**

With these examples, you should understand the general idea. Look in the configurations directory. Each of the choices from the Netbeans selector above typically maps to one of these (or exercises command line operation, described elsewhere).

All the configuration files refer to components located in the edu.nps.moves.qrtda.elements package in the source tree. They are not large files, and a quick reading of the source will let you know what they do.

End of document.