Robot SimGUI Walkthrough:

This walkthrough will explain how to set up and use the SimGUI.

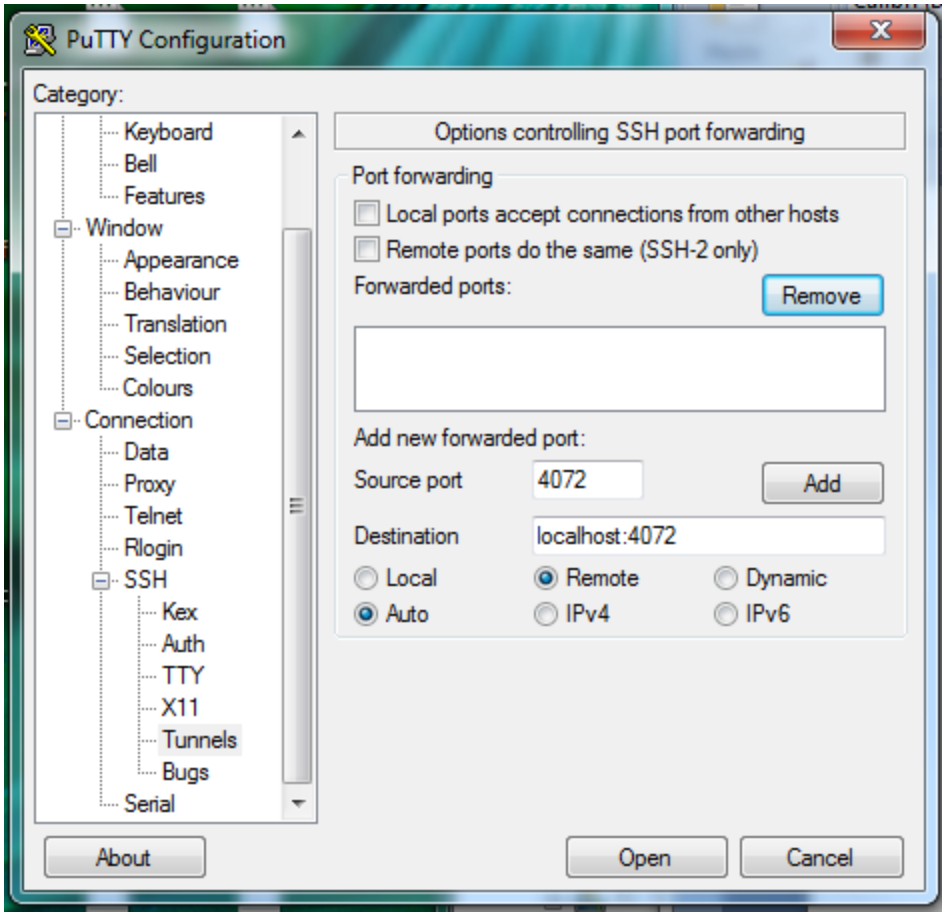
You will need:

1. RobotBrains on popeye
    - You must have written the RobotBrains you wish to test in order to run the GUI. As per the spec, these must be standalone executables that perform simple algorithms for deciding robot movements.
2. Piperproxy on popeye
    - This program acts as piper when using the gui. It stitches everything together. It should be placed in the file as your robot executables on popeye.
3. Proxy on popeye
    - This program acts as a relay between piper and the GUI. This program should also be placed in the file as your robot executables on popeye.
4. Java installed on your computer
    - The GUI is written in java, and you must therefore have the JVM installed on your computer. You must also be able to execute Java programs from the command line.
5. GUIJar.jar on your computer
6. A config file on your computer
    - The GUI's first step is to ask you for the location of a config file on your computer to use.
7. The ability to reverse-forward a port from popeye to your computer.
    - If you are using putty, there is an option to do this, and this will be discussed shortly.
    - If you are using SSH from a Linux terminal, ask Wayne for detailed instructions on how to do this.

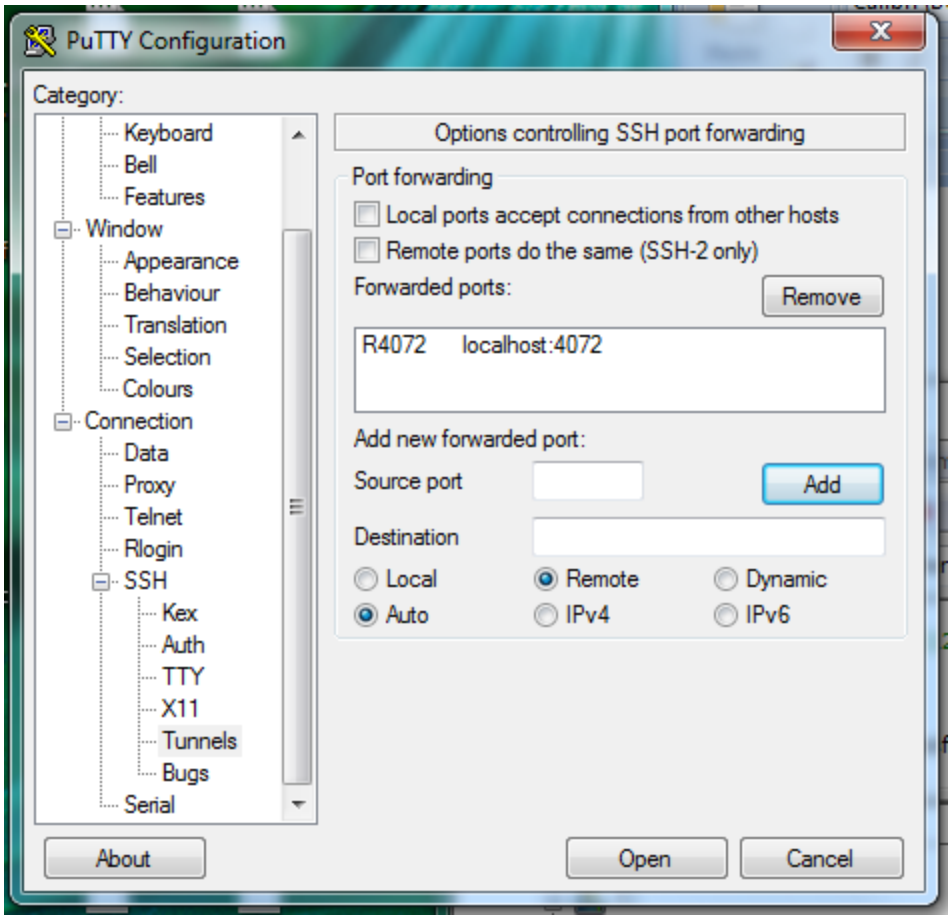
The first step is to remote into popeye with a port reverse-forwarded from popeye to your pc. To do this, you must first select the port you want to use. Usually, any port above 5000 up till 65535 will work; if you pick one that doesn't, you'll get an error message, and you'll just have to pick a different port.

To forward the port in putty, in the left-hand side menu, expand the SSH section, and select Tunnels. You will need to:
Type into Source port the port you want to use. Select remote.

Type into Destination the word localhost, followed by a : and the port you used. For example, to forward port 4072 from popeye to your computer, it would look like this:



Click Add. You should now have a forwarded port, and should look like the following:



Now log into Popeye as usual. When logged in, you can check the forwarded status. To do this, left-click the icon in the upper left corner, and select event log. If you were successful, there will be a line in there that now reads
Attempting to forward remote port to localhost:4072
Forwarded port opened successfully

Once you have port forwarding set up, you are ready to run.
The steps for execution are as follows:

1. Copy proxy and proxypiper into the same directory on popeye as your robot brains.
2. On your home computer, open the command prompt, navigate to where you have the file GUIJar.jar, and type:
   java –jar GUIJar.jar <numturns> <port>
   For example, to run it with 100 turns and use port 4072, you would say:
   java –jar GUIJar.jar 100 4072
   Note that nothing will display until you run the bots, but this is expected.
3. On popeye, execute proxypiper, with proxy as the first argument instead of your sim. An example usage:
   ./proxypiper ./proxy ./sentry ./pilladdict 4072

would run proxy with two bots on port 4072.

4. Now the GUI should display. Select the config file you want, and commence the simulation.

Some important things to note:

- On popeye, proxy will print out whatever it receives from both the sim and the bots, so you can see what messages are being passed where.
- The Sim gets really flaky if you don't meet some of the spec requirements. For example, make sure you give piperproxy the same number of bots as are in the config file you choose, or it will break in a hurry.
- The GUI imposes an additional requirement to the spec: Only files with board sizes of 35x35, the maximum allowed, work with the GUI. It will not allow a file of smaller size.
- The GUI executes all actions of the Robot. However, it doesn't display shooting or probing in any way, other than removing the required energy, and passing probe info on the next turn. It does execute these moves, though, so if your bot probes it will get what the spec says on the next turn.
- The Sim doesn't indicate if a move is illegal, it simply doesn't execute it. Proxy does attempt to catch some of these errors, but if they slip through, the Sim just won't execute it. Examples might be a negative parameter for a probe or shoot; Proxy won't catch that, and the Sim will just ignore that move for that bot.
- The Sim doesn't indicate that a bot has died, other than that it is removed from the Sim board, and won't receive any messages until the terminate message at the end.
- Lastly, please note that it is entirely possible that the GUI may have bugs. If you notice behavior that doesn't match the spec or seems incorrect, please bring it to the TAs' attention. Erroneous behavior in this GUI will not act as a justification for erroneous behavior in a submission. The GUI should be used as a tool, not a crutch.