Detailed Instructions to install the all the 3 Lunar Lander project:

**A. asteroid-lander:**

1. Fork the github repository. Go to the following url, <https://github.com/bravelittlescientist>. Here you will find the the link to the aestroid-lander. Fork and clone the repo locally on your System. Incase, you are not sure what fork and clone is, it's the right time to learn about them. Follow the link: <https://help.github.com/articles/fork-a-repo>.
2. Once the repository is copied. Follow the instruction listed below,

|  |
| --- |
| **Step 1:** To Install PodSixNet  $ cd your/path/to/asteroid-lander/PodSixNet-78  $ sudo python setup.py install  **Step 2:** Make sure the top-level project directory is on the Python Path. Some IDEs (Eclipse) will do this for you, but in Ubuntu you may need the following command:  $ export PYTHONPATH=$PYTHONPATH:/path/to/asteroid-lander/  **Step 3:** Install Python dependencies  $ [pip,easy\_install] jsonpickle  $ [pip,easy\_install] pygame # You need to install a bunch of headers for this to work, see the pygame site.  **Caveat:**  *Use the Synaptic Manager on Linux to install the same. There should be something similar for Mac(Homebrew ...) and Windows too. Best friend Google should help. if you encounter any problem. Shoot a mail and we will figure it out.*  **Step 4:** Make starting scripts executable  $ cd /path/to/asteroid-lander/game  $ chmod +x startClientHelperScript.sh  $ chmod +x startServerHelperScript.sh  Getting Started: Run Server and Clients!  To run a server:  $ cd path/to/asteroid-lander/game  $ ./startServerHelperScript.sh localhost:12345  Open a new tab to run two clients, each using startClientHelperScript.sh, using the same host and port!  $ cd path/to/steroid-lander/game  $ ./startClientHelperScript.sh localhost:12345  Each window will open a tab! Use space to start the game, click a mineral button in the bottom right, and use arrow-keys to control your ship!  **Known Issues**  Issue 1: Python Path (Seen on Ubuntu)  If you try to run the game and it is not on your PythonPath, it will report an error like this:  ImportError: No module named game.Constants  The solution is to add the asteroid-lander top-level project directory to your Python Path.  $ export PYTHONPATH=$PYTHONPATH:/path/to/asteroid-lander/ |

**B. SuperAsteroid Melee:**

1. This project was developed using nodejs. Should be fun to extend it.
2. Follow the step on this link for installation of nodejs, npm and socket.io needed for the project, <http://rawkes.com/articles/creating-a-real-time-multiplayer-game-with-websockets-and-node.html>
3. Search for any occurrence of '<http://169.234.20.35:8000/socket.io/socket.io.js>' and replace it with '<http://localhost:8000/socket.io/socket.io.js>'( replace local-host to any-other ip, if you wish to use a server of your choice). Change should only be in the file

*yourpath/UpdatedCode/public/index.html*

* + cd to yourpath/SuperAsteroidMelee(Chan)/UpdatedCode/
  + type node game.js and press enter. This will start the server(socket.io)
  + Open a browser, would suggest using Chrome or Firefox as they have in-built javascript debugger as a part of their dev tools.
  + Type yourpath/SuperAsteroidMelee(Chan)/UpdatedCode/public/index.html in the opened browser and press enter. This should connect the client to the server.
  + Start enjoying the game.

**C. Java based Lunar Lander:**

Would suggest to use, eclipse for this project. However command line would work fine too. Instruction below are wrt to eclipse.

1. Import the project into the eclipse workspace as an existing project.
2. Resolve the build dependencies by adding all the jars.

* Right click on the imported project and click on 'Build Path' → 'Configure Build Path' → Go to libraries tab and add all the jars namely *1. kryonet-2.18-all 2. kryonet-debug-2.18-all 3. logging\_system(extract the zip file and all the jars under the lib folders) 4. commons-io-2.4\*.jar.*
* This should resolve all the dependencies.
* Now there is a server and client module. Start the server code using the vm flag '*-Djava.net.preferIPv4Stack=true*'. Similarly, start the client module. Just to be clear, the Server and client module are in different folders.

3. You are all set to go. Enjoy the game and extend it.