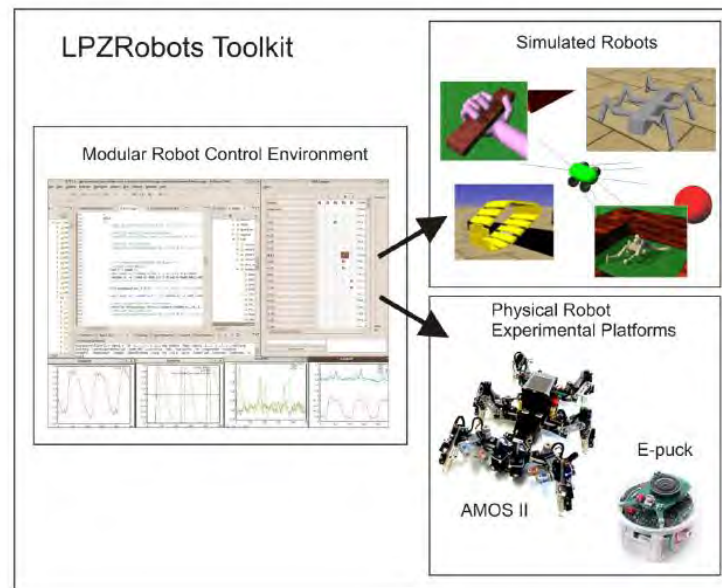


Installation: Gorobots_edu & Lpzrobots (C++)

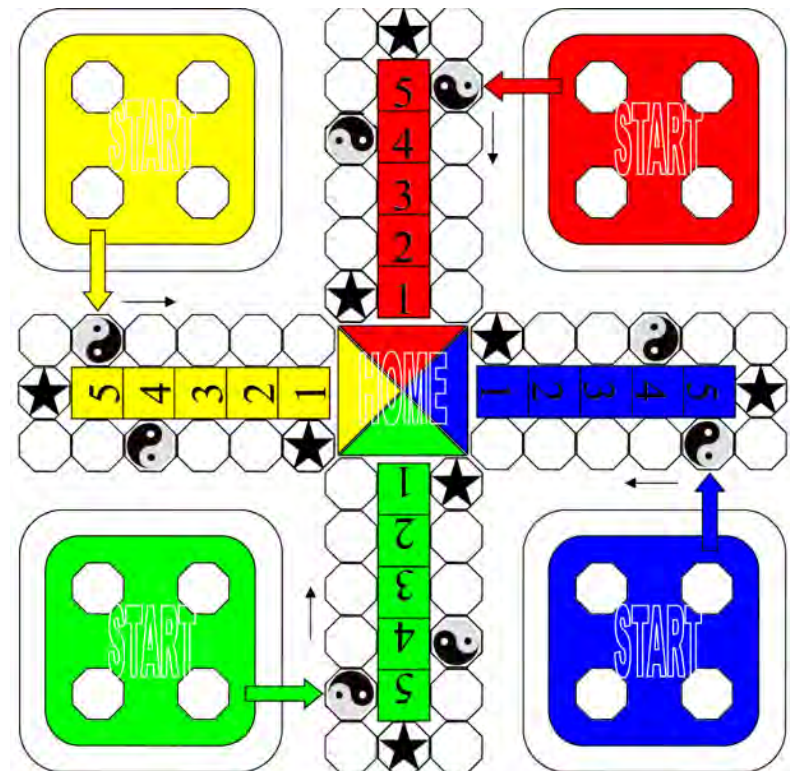
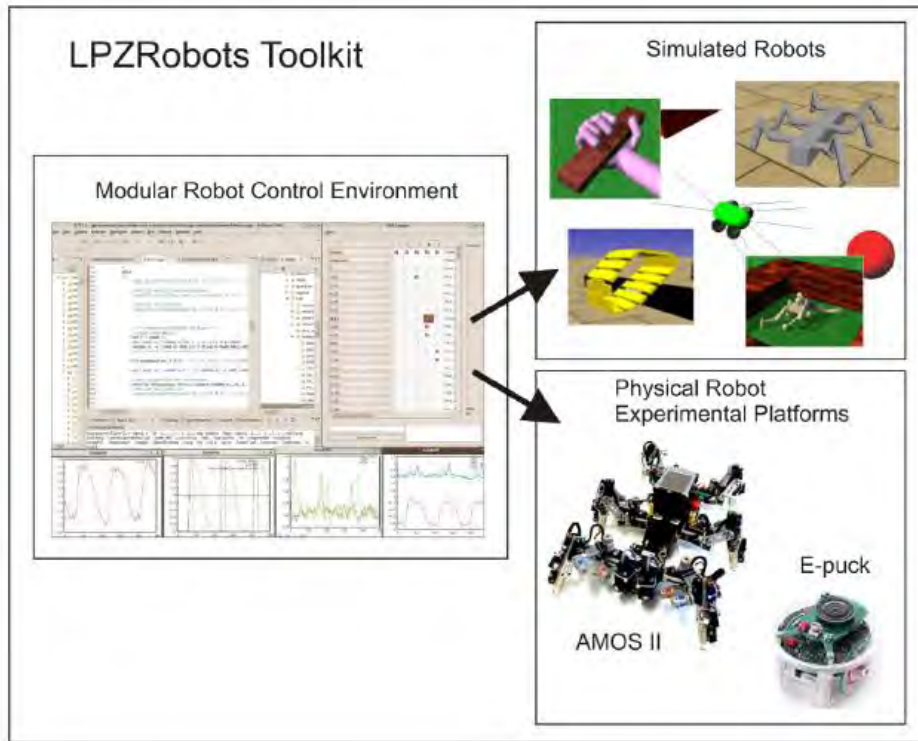


<http://manoonpong.com/MOROCO/manual.pdf>

Lpzrobots has been developed by Research Network for Self-Organization of Robot Behavior in Leipzig, Göttingen and Edinburgh

Exercises

- 1) Robot simulation (C++, gorobots_edu)
- 2) Ludo game (Java)



Step by step

Please follow the installation guide from

http://manoonpong.com/MOROCO/lpz_guide.txt

<https://github.com/pmanoonpong>

Step by step

Import the project settings file, Compile Lpzrobots

```
----- Compile LpzRobots -----  
Do you want to compile LpzRobots now? [y/n] : y  
  
-----make LpzRobots -----  
cd /home/poma/workspace/pmanoonpong-lpzrobots-fork  
make all  
Makefile:7: Makefile.conf: No such file or directory  
Generating Makefile.conf (configuration makefile)  
You can change your preference by editing Makefile.conf  
or just delete it and run make again  
Where do you want to install the simulator?  
Please use either /usr, /usr/local or you home directory  
unless you know what you are doing. (no trailing slash)  
e.g. (/home/yourlogin) (don' use ~): [/usr/local] /home/poma
```

“Your login name”

Step by step

If you cannot compile Lpzrobots → may be you have to add this path in .bashrc, otherwise skip this! → “export PATH=/home/your login/bin:\$PATH”

```
# definitions for lpzrobots
export CPATH="$HOME/include"
export LIBRARY_PATH="$HOME/lib"
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$HOME/lib:/usr/lib/osgPlugins-3.0.1
export PATH=${PATH}:$HOME/bin
export PATH=/home/poma/bin:$PATH
```

Step by step

```
Installation type (user or development):  
  Choose user (u) if you are a user and only program your own simulations (default)  
  Choose devel (d) if you develop the simulator  
Our choice (u/d): [d] d
```

Type "d" as developer

```
Check your settings:  
  Installation to /home/poma  
  (d) development installation  
All right? [y/N] y
```

Step by step

**** Done, you can go and compile your simulations ****

```
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-lpzrobots-fork/ode_robots/simulations/nimm2_arena$ make
```

#go to simulations directory

/workspace/yourlogin-lpzrobots-fork\ode_robots\simulations\nimm2_arena\ make and then ./start

Step by step

- If you have an error about '**libode_dbl.so.1**'
- This is because when you compile lpz-robot to some local directory. A quick fix involves making a symbolic link (with root privilege) that points to the file in your systems /lib/ library.
- for an example:

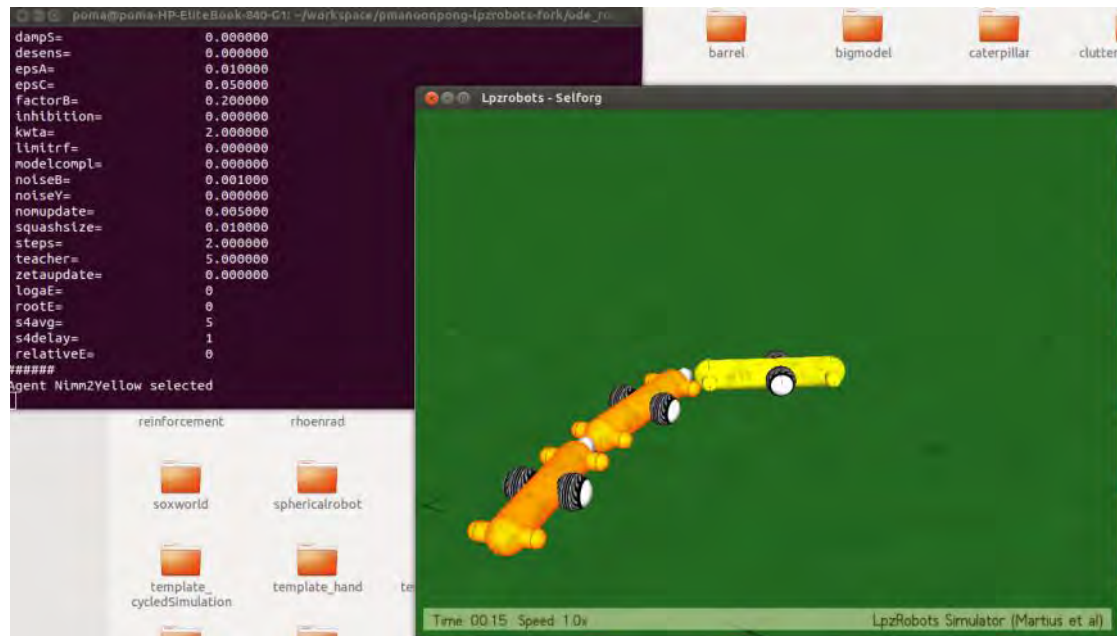
```
sudo ln -sf home/YOUR USER NAME/lpzrobots/opende/ode/src/.libs/libode_dbl.so.1 /lib/libode_dbl.so.1
```


Step by step

**** Done, you can go and compile your simulations ****

```
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-lpzrobots-fork/ode_robots/simulations/nimm2_arena$ make
```

You should see this!



Using Lpzrobots

- > after start simulation , press 1 to fixed camera view
- > Ctrl r = record movie
- > .\start -f = record log file
- > .\start = start program
- > .\start -g 1 = display GUI

Check out a remote branch in git

If you would like to checkout the branch groups, you can simply say this in recent versions of Git:

```
>> git fetch origin
```

```
>> git branch -v -a
```

This will show all branches

```
>> git checkout -b stable/ lpzrobots_ai2_class
```

Or >> git checkout -b origin/ lpzrobots_ai2_class

This will automatically track origin/groups from a local branch groups. E.g.,

```
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-lpzrobots-fork$ git  
fetch origin
```

Password for 'https://p.manoonpong@git.assembla.com':

```
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-lpzrobots-fork$ git  
checkout -b origin/ lpzrobots_ai2_class
```

Problem of Displaying GUI logger

- Please install "*gnuplot-x11*"
- *sudo apt-get install gnuplot-x11*

Problem of Displaying GUI logger

- guillogger not working (not compiling).
- When installing lpz-robots, there is a small change you need to make before you running "make all".
- Edit lpzrobots/guillogger/configure.sh
Change "qmake" into "qmake-qt4".
- This will allow guillogger to compile.
Now you can "make all" from the lpzrobots directory.

Installation

Ubuntu and Lpzrobot simulation

Install Ubuntu 18.04

- Download Ubuntu software

Ubuntu 18.04 choose alternative downloads on the download website

<https://releases.ubuntu.com/bionic/>

Select an image

Ubuntu is distributed on three types of images described below.

Desktop image

The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of image is what most people will want to use. You will need at least 1024MiB of RAM to install from this image.

[64-bit PC \(AMD64\) desktop image](#)
Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

Server install image

The server install image allows you to install Ubuntu permanently on a computer for use as a server. It will not install a graphical user interface.

[64-bit PC \(AMD64\) server install image](#)
Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

A full list of available files, including BitTorrent files, can be found below.
If you need help burning these images to disk, see the [Image Burning Guide](#).

Name	Last modified	Size	Description
Parent Directory	-	-	-
64-bit PC (AMD64) desktop image	2018-04-11 14:00	2.6GB	64-bit PC (AMD64) desktop image

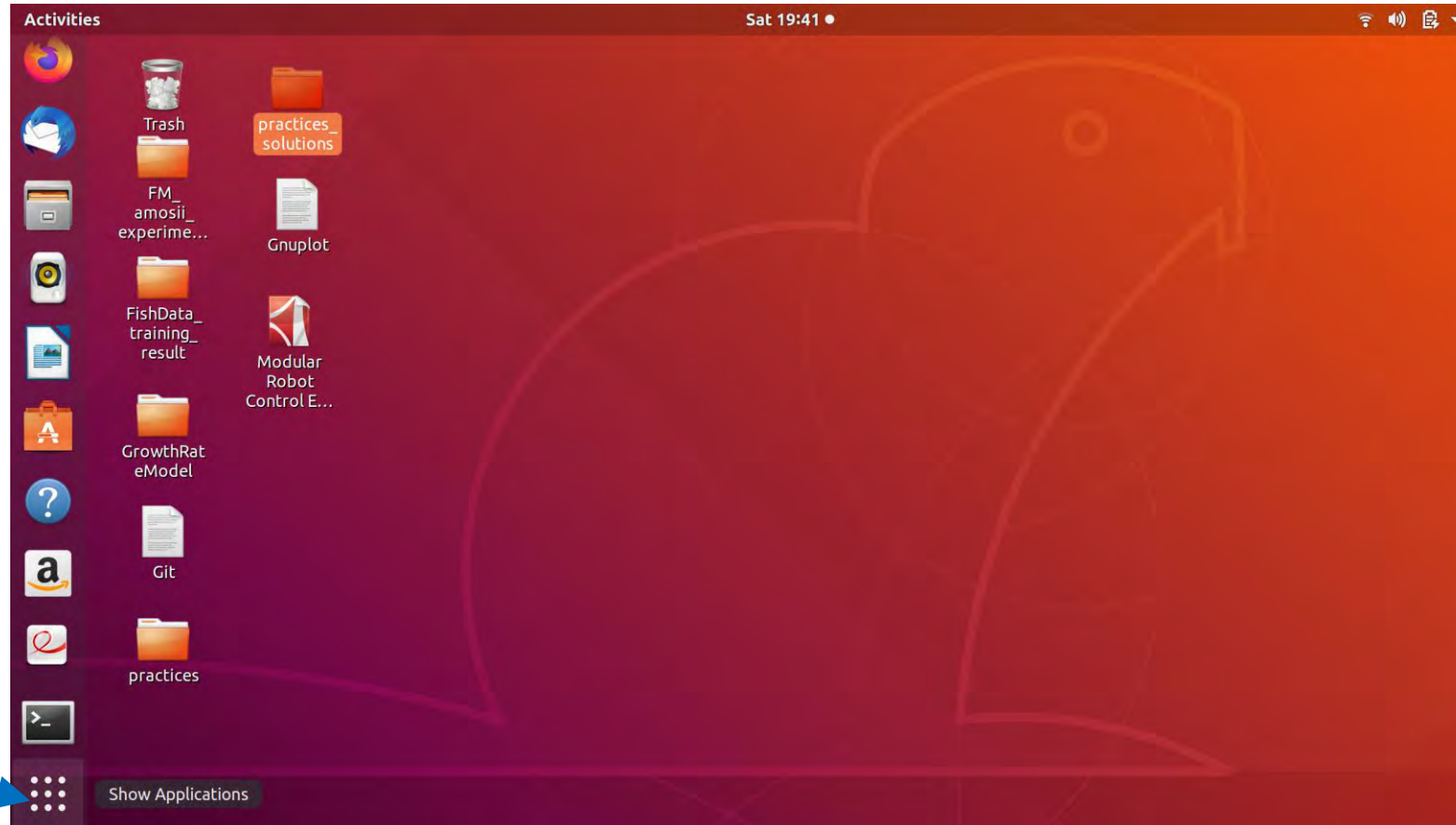
Install Ubuntu 18.04

- After you downloaded the software, please install the Ubuntu 18.04
- You can also see the link below how to install Ubuntu 18.04 step by step alongside With Windows 10 in Dual Boot
- <https://www.itzgeek.com/how-tos/linux/ubuntu-how-tos/how-to-install-ubuntu-18-04-alongside-with-windows-10-or-8-in-dual-boot.html>

After installing Ubuntu 18.04

- Please open Ubuntu and open “terminal”
- To open terminal

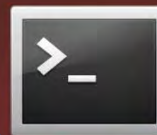
Click this



Type → terminal

terminal

Click this



Terminal



Ubuntu Software
12 more

- Terminator Multiple **terminals** in one window
- MATE Terminal Use the command line
- Termit Lightweight **terminal** emulator
- Xfce Terminal **Terminal** Emulator
- terminatorX terminatorX is a realtime audio synthesizer



Trash

FM_
amosii_
experime...FishData_
training_
resultGrowthRat
eModel

Git



practices

practices_
solutions

Gnuplot

Modular
Robot
Control E...

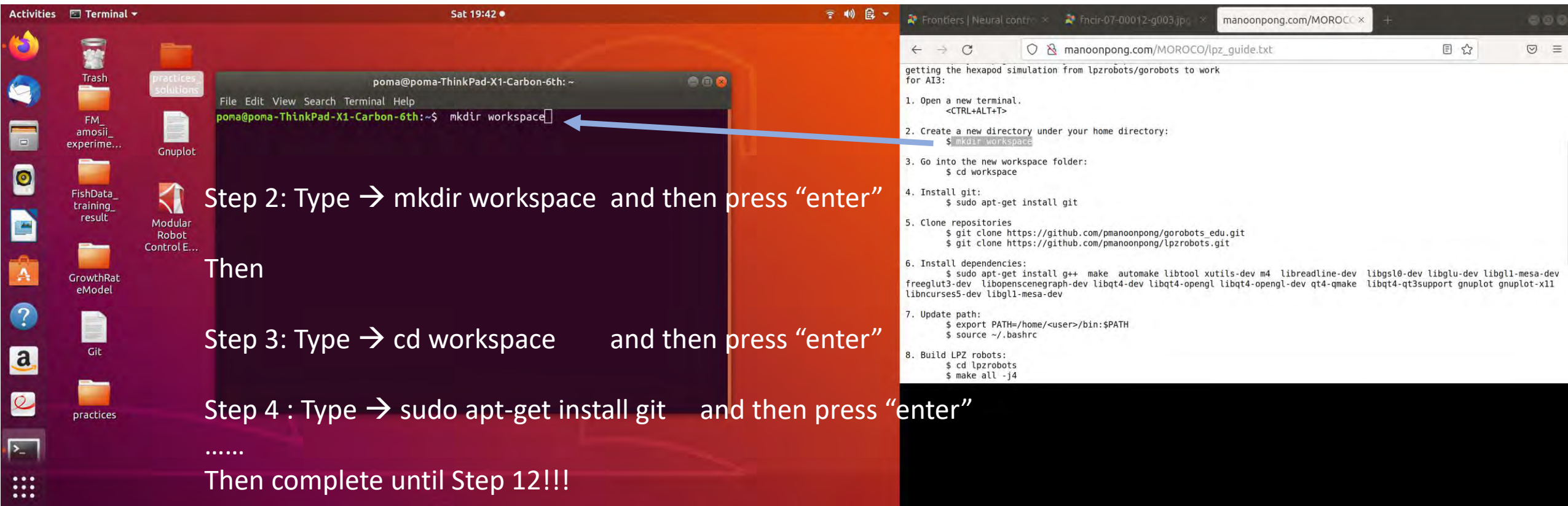
```
poma@poma-ThinkPad-X1-Carbon-6th: ~  
File Edit View Search Terminal Help  
poma@poma-ThinkPad-X1-Carbon-6th:~$
```

You should now see terminal

Installing Lpzrobots

Follow the instruction from this link

http://manoonpong.com/MOROCO/lpz_guide.txt



Sat 19:42

Step 2: Type → `mkdir workspace` and then press “enter”

Then

Step 3: Type → `cd workspace` and then press “enter”

Step 4 : Type → `sudo apt-get install git` and then press “enter”

.....

Then complete until Step 12!!!

manoonpong.com/MOROCO/lpz_guide.txt

getting the hexapod simulation from lpzrobots/gorobots to work for AI3:

1. Open a new terminal.
 `<CTRL+ALT+T>`
2. Create a new directory under your home directory:
 `$ mkdir workspace`
3. Go into the new workspace folder:
 `$ cd workspace`
4. Install git:
 `$ sudo apt-get install git`
5. Clone repositories
 `$ git clone https://github.com/pmanoonpong/gorobots_edu.git`
 `$ git clone https://github.com/pmanoonpong/lpzrobots.git`
6. Install dependencies:
 `$ sudo apt-get install g++ make automake libtool xutils-dev m4 libreadline-dev libgsl0-dev libglu-dev libgl1-mesa-dev
freeglut3-dev libopencscenagraph-dev libqt4-dev libqt4-opengl libqt4-opengl-dev qt4-qmake libqt4-qt3support gnuplot gnuplot-x11
libncurses5-dev libgl1-mesa-dev`
7. Update path:
 `$ export PATH=/home/<user>/bin:$PATH`
 `$ source ~/.bashrc`
8. Build LPZ robots:
 `$ cd lpzrobots`
 `$ make all -j4`

Problem and Solution

- Install tutorial:(初学者强烈建议虚拟机， 初学者强烈建议虚拟机， 初学者强烈建议虚拟机)

- 1 Vmware

- 2 Ubuntu安装：<https://www.bilibili.com/video/BV1zt411G7Vn?p=2>

(看完再装， 看完再装， 看完再装)

(Ubuntu安装完成之后， 虚拟机 / 虚拟机设置 / 硬件 / CD/DVD(SATA) 处的连接改为使用物理驱动器， 退出ISO镜像文件；)

(Ubuntu安装完成之后， 虚拟机 / 虚拟机设置 / 硬件 / CD/DVD(SATA) 处的连接改为使用物理驱动器， 退出ISO镜像文件；)

(Ubuntu安装完成之后， 虚拟机 / 虚拟机设置 / 硬件 / CD/DVD(SATA) 处的连接改为使用物理驱动器， 退出ISO镜像文件；)

(最好安装Vmware tools， 最好安装Vmware tools， 最好安装Vmware tools)

3 http://manoonpong.com/MOROCCO/lpz_guide.txt

Problem and Solution

- Problem1: Install the Ubuntu failed;
- Solution: Download other edition ISO Ubuntu, e.g. Ubuntu18.04.01, Ubuntu18.04.04;
- Problem2: Network is not good.Solution:
- Try your phone hotspot. Or choose the best server. And maybe need to wait minuites.
- Problem3: Install git failed
- Before the install git, try \$sudo apt-get update
- Problem4: User is the name before "@". My computer's user is "dongyi".
- Solution: Find your computer's user.
- Problem5: Be careful about your folder's name. Is it "workspace"?
- Solution: If you use other folder name, replace all "workspace" in the tutorial codes with your folder name.
- Problem6: make sure your step8 make successfully
- Solution: If you failed in the first time, you can repeat this step for sencond time.

Problem and Solution

```
make[1]: Entering directory '/home/yanbin/workspace/gorobots_edu/practices/amosii'
make[1]: ode_robots-config: Command not found
make[1]: selforg-config: Command not found
make[1]: selforg-config: Command not found
make[1]: ode_robots-config: Command not found
make[1]: ode_robots-config: Command not found
make[1]: selforg-config: Command not found
make[1]: selforg-config: Command not found
make[1]: ode_robots-config: Command not found
g++ -Wall -pipe -Wno-deprecated -I -I. -I. -c -o main.o main.cpp
main.cpp:26:10: fatal error: ode_robots/simulation.h: No such file or directory
#include <ode_robots/simulation.h>
          ^
compilation terminated.
<builtin>: recipe for target 'main.o' failed
make[1]: *** [main.o] Error 1
make[1]: Leaving directory '/home/yanbin/workspace/gorobots_edu/practices/amosii'
Makefile:53: recipe for target 'normal' failed
make: *** [normal] Error 2
yanbin@yanbin-ThinkPad-X13-Gen-2i:~/workspace/gorobots_edu/practices/amosii$
```

This means that: you did not successfully compile Lpzrobots in the step 8

→ Build LPZ robots: `$ cd lpzrobots $ make all -j4`

Solution: redo "make all"


Problem and Solution

- 1>when I install lpz, run step8"make all -j4", i didnot notice some error during compile process
- 2>i continued the installation, still follow the step to the end, it shows the robot, no error.
- 3>then i turn to task1, make clean, then the generated executable file change, cannot make successfully again, shows error
- 4>the solution is turn to step8"make all", i think it show "go to simulation" such notice means it compiled successfully







Make sure you use the right “Makefile”

🔗 master ▾ [lpzrobots](#) / [ode_robots](#) / [simulations](#) / [template_amosii](#) /

This branch is 22 commits ahead, 28 commits behind georgmartius:master.

 georgmartius again gitignore

..

 .gitignore	again gitignore
 Makefile.conf	amosII definitions moved to ode_robot/robots
 main.cpp	added rubber_feet usage to Amos II template simulation
 tripodgait18dof.cpp	removed obsolete function insertCVSInfo
 tripodgait18dof.h	wrapped lines in tripodgait18dof.h to fit into 80 characters
 Makefile	

Copy Makefile



pmanoonpong Update PR mechanism

..



Complete_control_solution.h

Update practice for adaptive locomotion control course



Makefile.conf

Update practice amosii



Task1_CPG_control_empty.h

Update practice amosii



Task1_CPG_control_solution.h

Update practice for adaptive locomotion control course



Task2_CPG_VRN_control_empty.h

Update practice amosii



Task2_CPG_VRN_control_solution.h

Update practice for adaptive locomotion control course



Task3_CPG_VRN_MRC_control_empty.h

Update practice for adaptive locomotion control course



Task3_CPG_VRN_MRC_control_solution.h

Update practice for adaptive locomotion control course



Task4_CPG_VRN_MRC_ICO_control_empty.h

Update practice for adaptive locomotion control course



Task4_CPG_VRN_MRC_ICO_control_solution.h

Update practice for adaptive locomotion control course



Task5_CPG_VRN_MRC_ICO_Hormone_control_empty.h

Update practice for adaptive locomotion control course



Task5_CPG_VRN_MRC_ICO_Hormone_control_solution.h

Update practice for adaptive locomotion control course



Task6_CPG_VRN_MRC_ICO_DecoupledCPGs_control_emp...

Update practice for adaptive locomotion control course



Task6_CPG_VRN_MRC_ICO_DecoupledCPGs_control_solu...

Update PR mechanism



emptycontroller.h

Update practice for adaptive locomotion control course



main.cpp

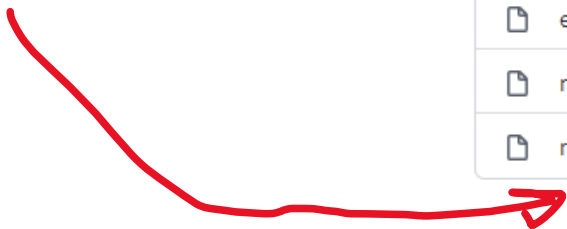
Update practice for adaptive locomotion control course



rough6.ppm

Update practice amosii

Put it here!!

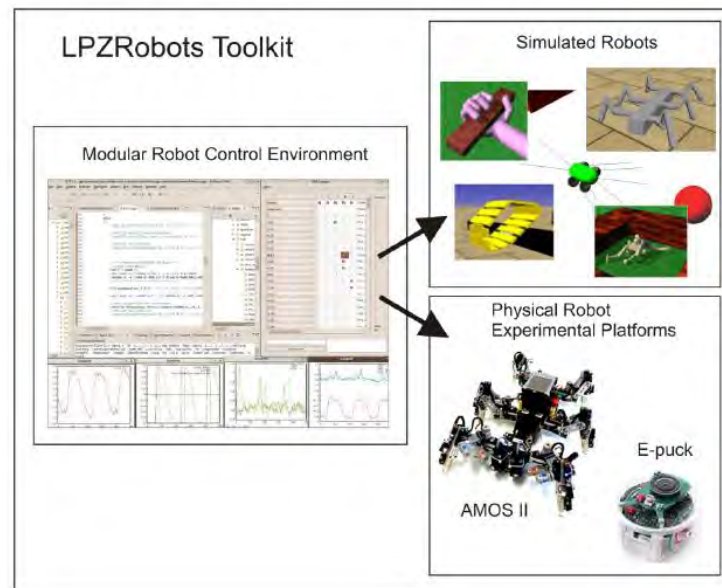


If you have Ubuntu 20

- https://github.com/TheWorldOfCode/AI2_docker

Appendix

Installation: Gorobots_edu & Lpzrobots (C++)

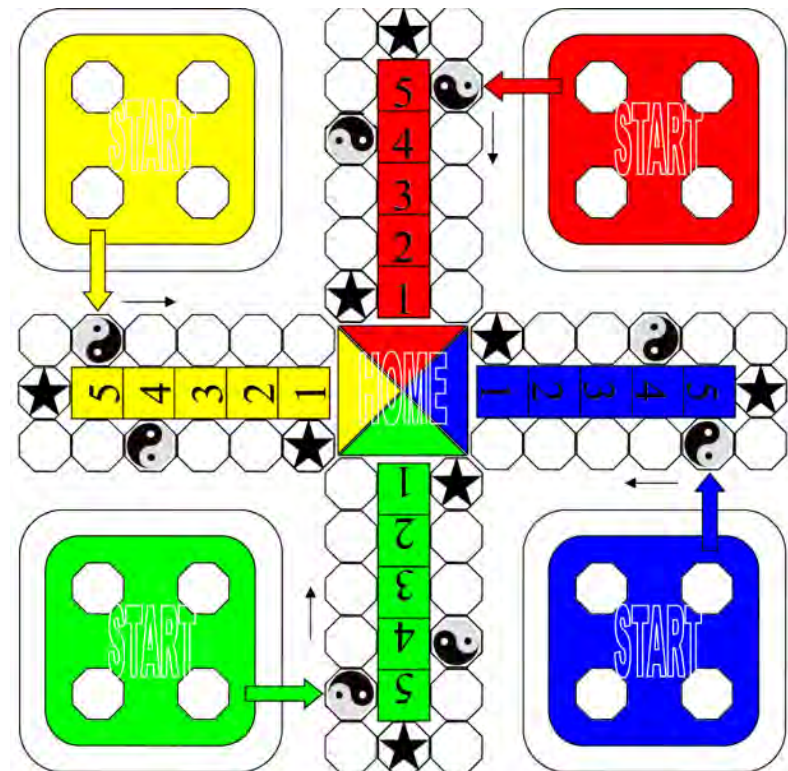
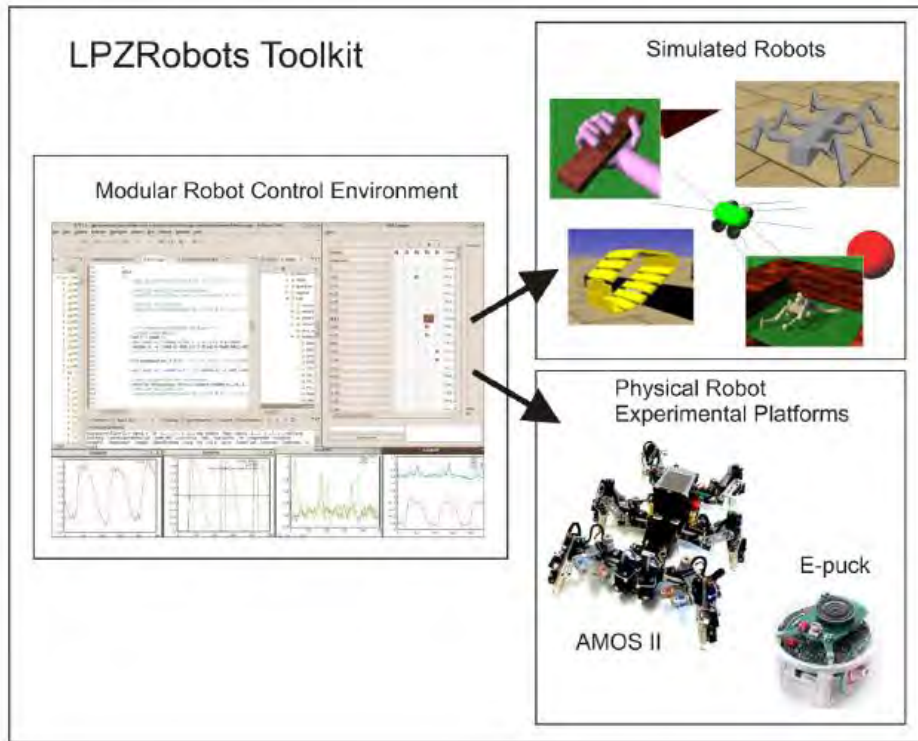


<http://manoonpong.com/MOROCO/manual.pdf>

Lpzrobots has been developed by Research Network for Self-Organization of Robot Behavior in Leipzig, Göttingen and Edinburgh

Exercises

- 1) Robot simulation (C++, gorobots_edu)
- 2) Ludo game (Java)



Step by step

Please follow the installation guide from

http://manoonpong.com/MOROCO/lpz_guide.txt

<https://github.com/pmanoonpong>

Step by step

Import the project settings file, Compile Lpzrobots

```
----- Compile LpzRobots -----  
Do you want to compile LpzRobots now? [y/n] : y  
  
-----make LpzRobots -----  
cd /home/poma/workspace/pmanoonpong-lpzrobots-fork  
make all  
Makefile:7: Makefile.conf: No such file or directory  
Generating Makefile.conf (configuration makefile)  
You can change your preference by editing Makefile.conf  
or just delete it and run make again  
Where do you want to install the simulator?  
Please use either /usr, /usr/local or you home directory  
unless you know what you are doing. (no trailing slash)  
e.g. (/home/yourlogin) (don' use ~): [/usr/local] /home/poma
```

“Your login name”

Step by step

If you cannot compile Lpzrobots → may be you have to add this path in .bashrc, otherwise skip this! → “export PATH=/home/your login/bin:\$PATH”

```
# definitions for lpzrobots
export CPATH="$HOME/include"
export LIBRARY_PATH="$HOME/lib"
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$HOME/lib:/usr/lib/osgPlugins-3.0.1
export PATH=${PATH}:$HOME/bin
export PATH=/home/poma/bin:$PATH
```

Step by step

```
Installation type (user or development):  
  Choose user (u) if you are a user and only program your own simulations (default)  
  Choose devel (d) if you develop the simulator  
Our choice (u/d): [d] d
```

Type "d" as developer

```
Check your settings:  
  Installation to /home/poma  
  (d) development installation  
All right? [y/N] y
```

Step by step

**** Done, you can go and compile your simulations ****

```
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-lpzrobots-fork/ode_robots/simulations/nimm2_arena$ make
```

#go to simulations directory

/workspace/yourlogin-lpzrobots-fork\ode_robots\simulations\nimm2_arena\ make and then ./start

Step by step

- If you have an error about '**libode_dbl.so.1**'
- This is because when you compile lpz-robot to some local directory. A quick fix involves making a symbolic link (with root privilege) that points to the file in your systems /lib/ library.
- for an example:

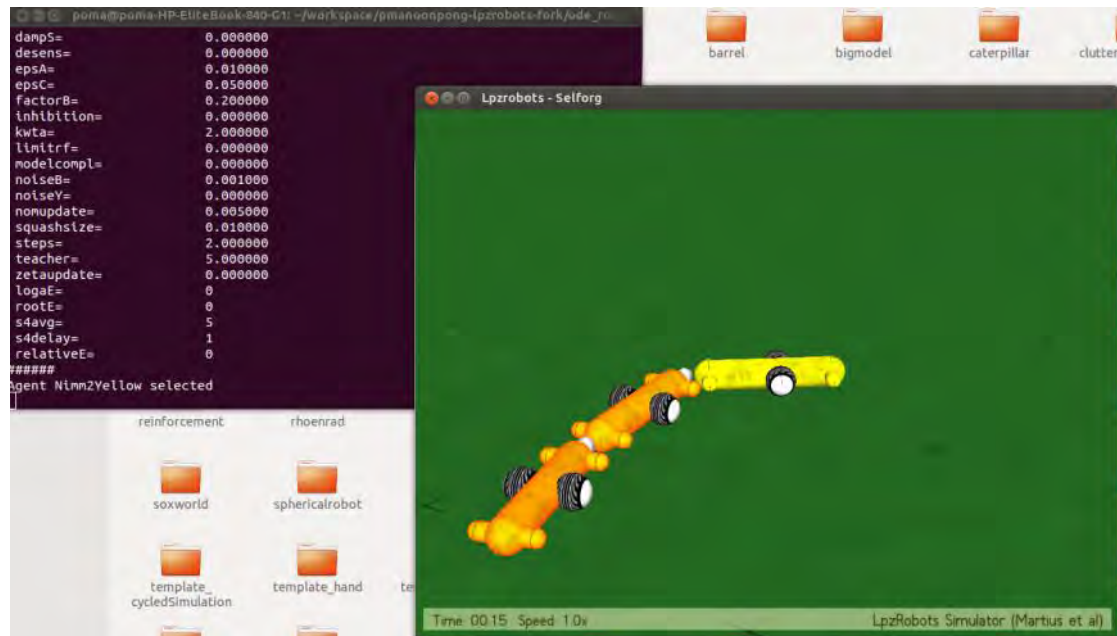
```
sudo ln -sf home/YOUR USER NAME/lpzrobots/opende/ode/src/.libs/libode_dbl.so.1 /lib/libode_dbl.so.1
```

Step by step

**** Done, you can go and compile your simulations ****

```
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-lpzrobots-fork/ode_robots/simulations/nimm2_arena$ make
```

You should see this!



Using Lpzrobots

- > after start simulation , press 1 to fixed camera view
- > Ctrl r = record movie
- > .\start -f = record log file
- > .\start = start program
- > .\start -g 1 = display GUI

Check out a remote branch in git

If you would like to checkout the branch groups, you can simply say this in recent versions of Git:

```
>> git fetch origin
```

```
>> git branch -v -a
```

This will show all branches

```
>> git checkout -b stable/ lpzrobots_ai2_class
```

Or >> git checkout -b origin/ lpzrobots_ai2_class

This will automatically track origin/groups from a local branch groups. E.g.,

```
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-lpzrobots-fork$ git  
fetch origin
```

Password for 'https://p.manoonpong@git.assembla.com':

```
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-lpzrobots-fork$ git  
checkout -b origin/ lpzrobots_ai2_class
```

Problem of Displaying GUI logger

- Please install "*gnuplot-x11*"
- *sudo apt-get install gnuplot-x11*

Problem of Displaying GUI logger

- guillogger not working (not compiling).
- When installing lpz-robots, there is a small change you need to make before you running "make all".
- Edit lpzrobots/guillogger/configure.sh
Change "qmake" into "qmake-qt4".
- This will allow guillogger to compile.
Now you can "make all" from the lpzrobots directory.

Installation: Eclipse editor & compiler

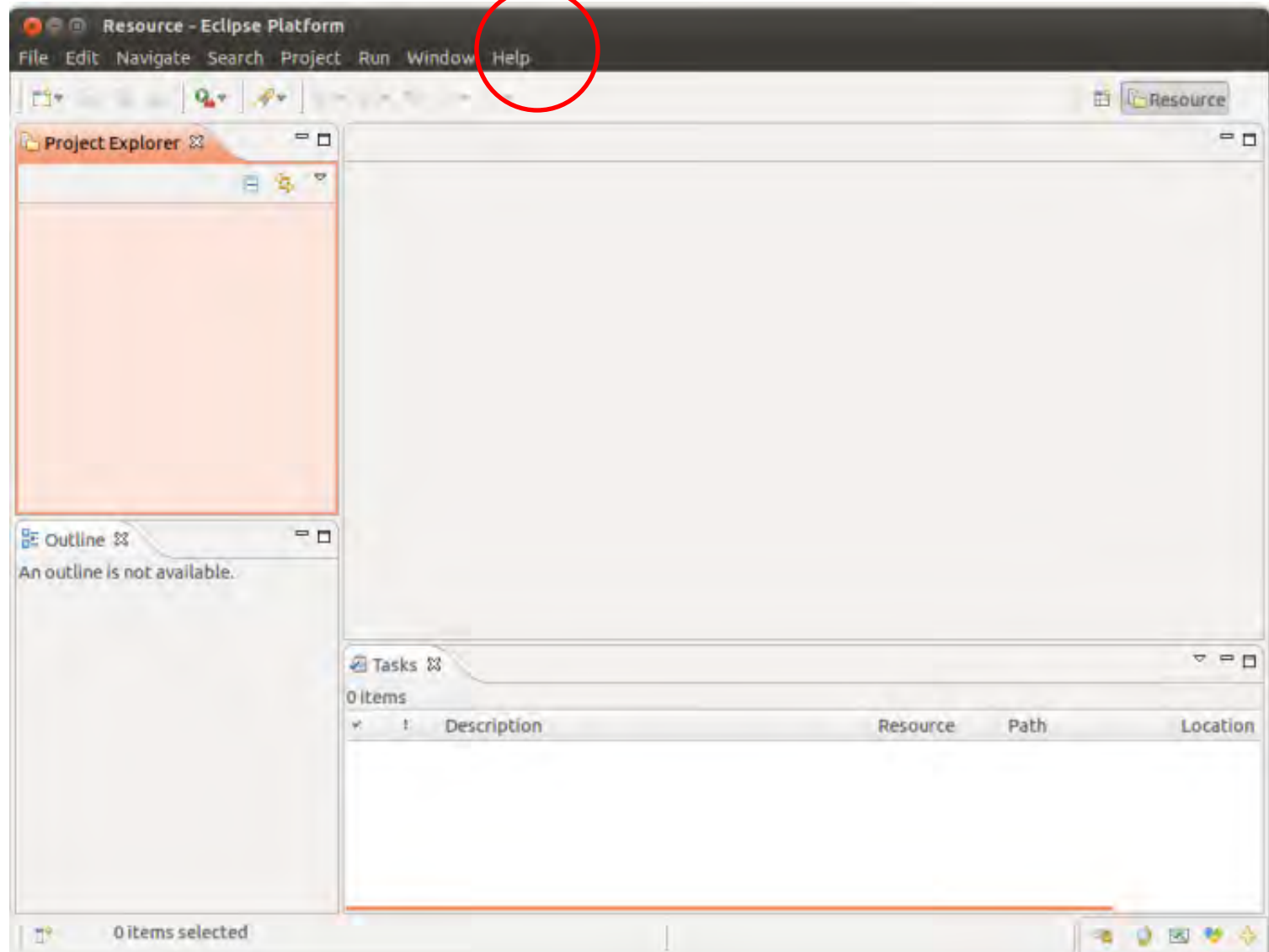


Step by step

7) Setup editor, e.g., Eclipse → Install Eclipse from Ubuntu Software Center. If you already installed, then just run Eclipse

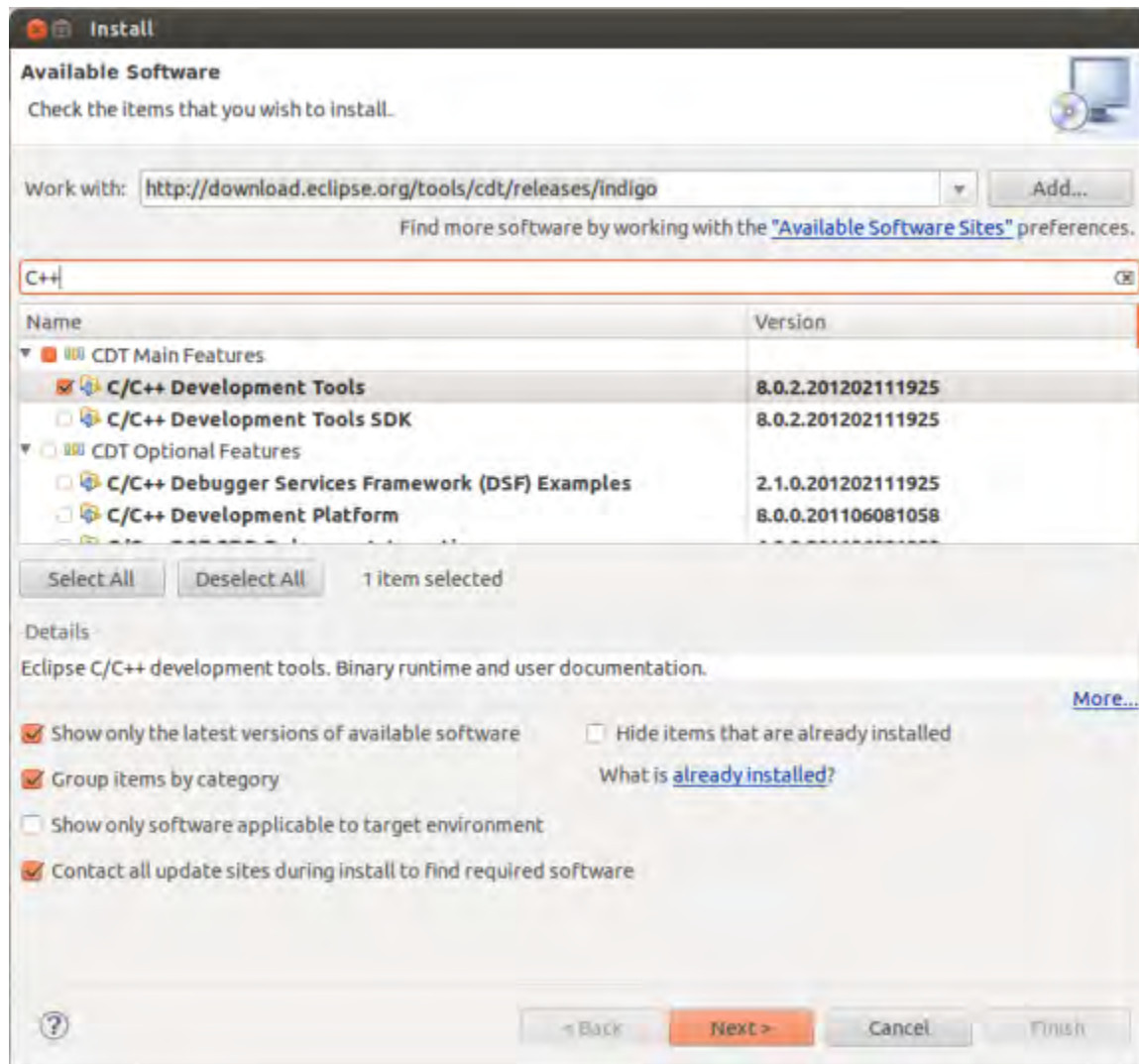


Step by step



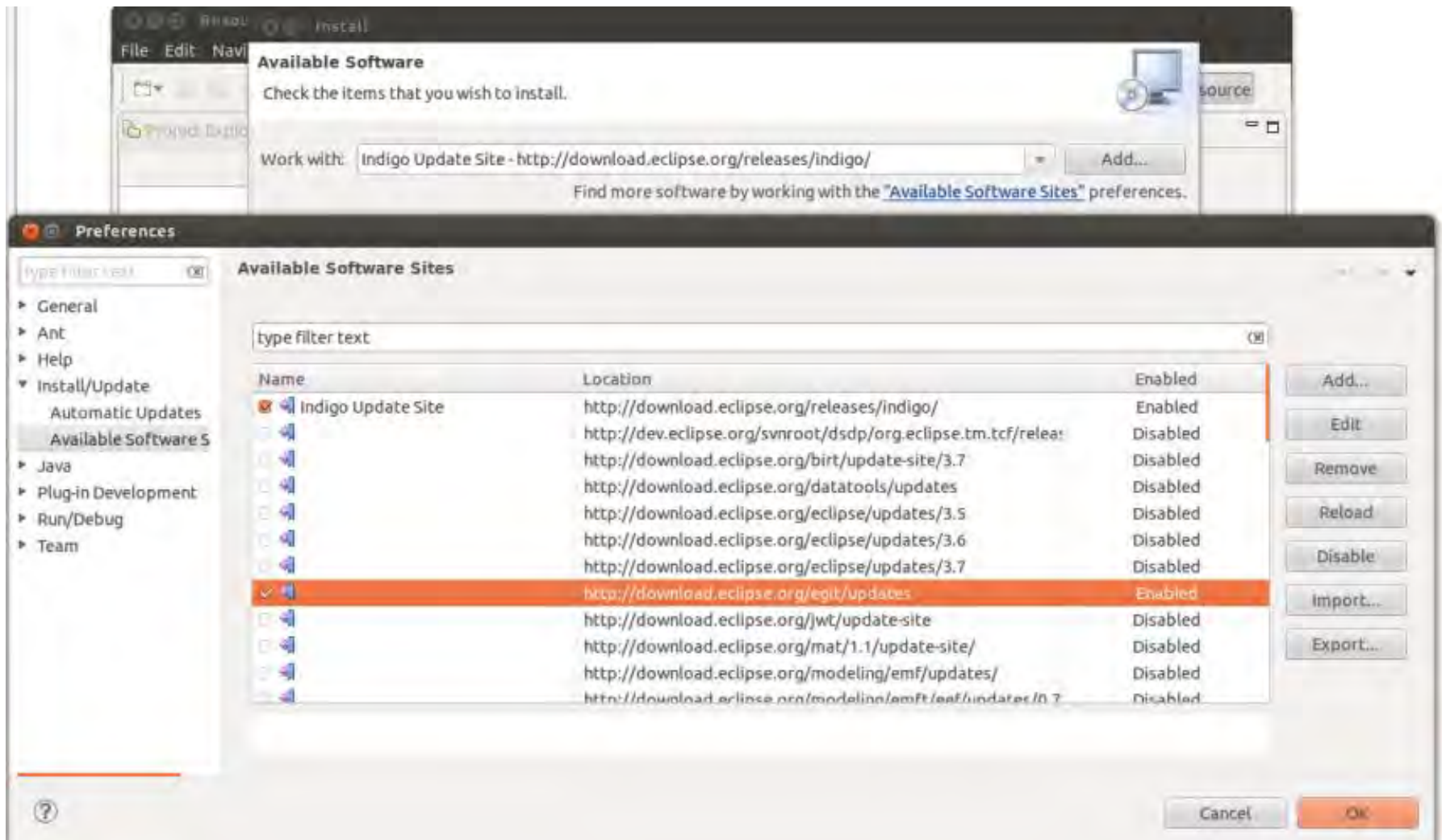
Step by step

Help>> Install New Software>> Add C++ Development Tools



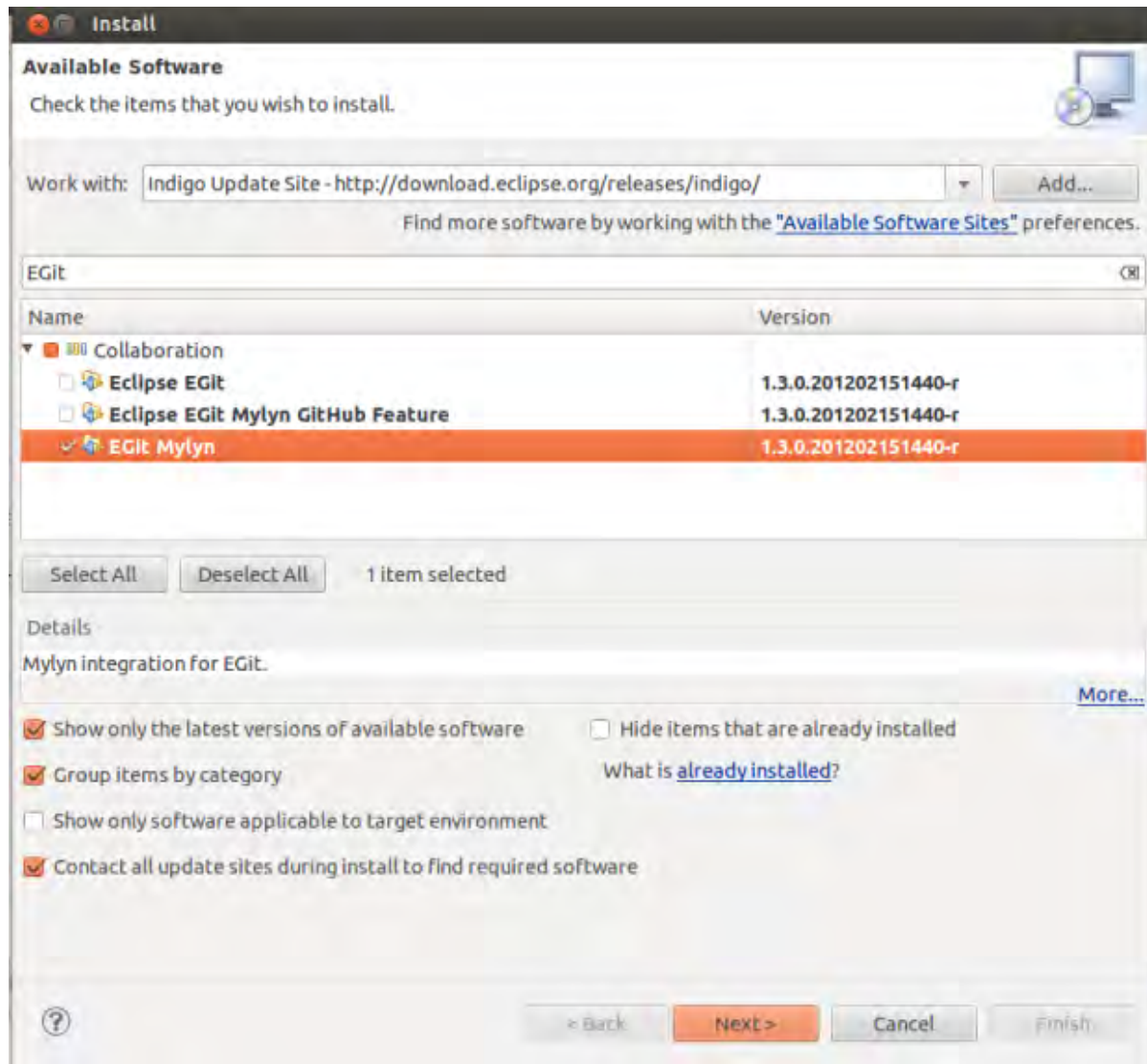
Step by step

Help>> Install New Software>> Add EGit



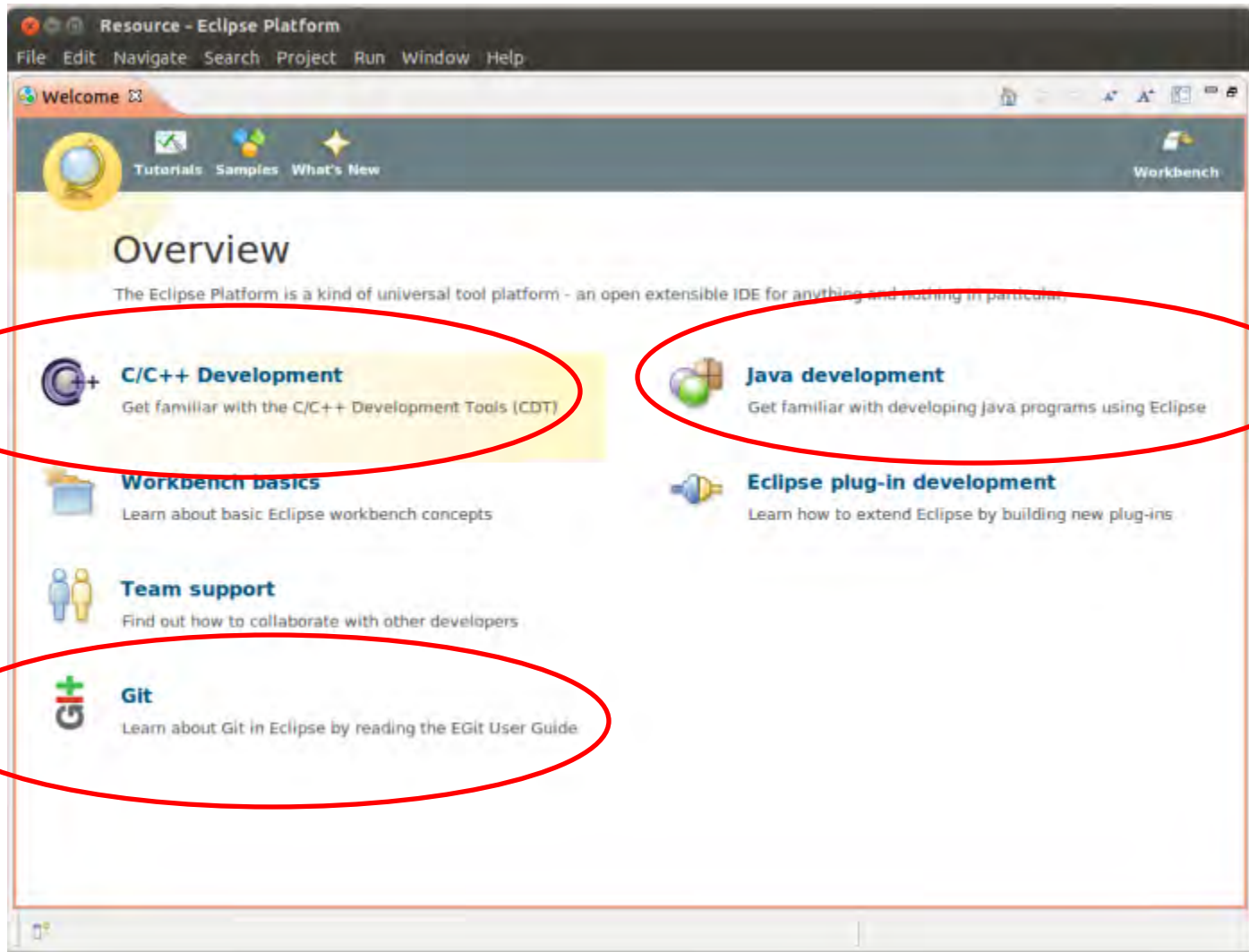
Step by step

Help>> Install New Software>> Add EGit



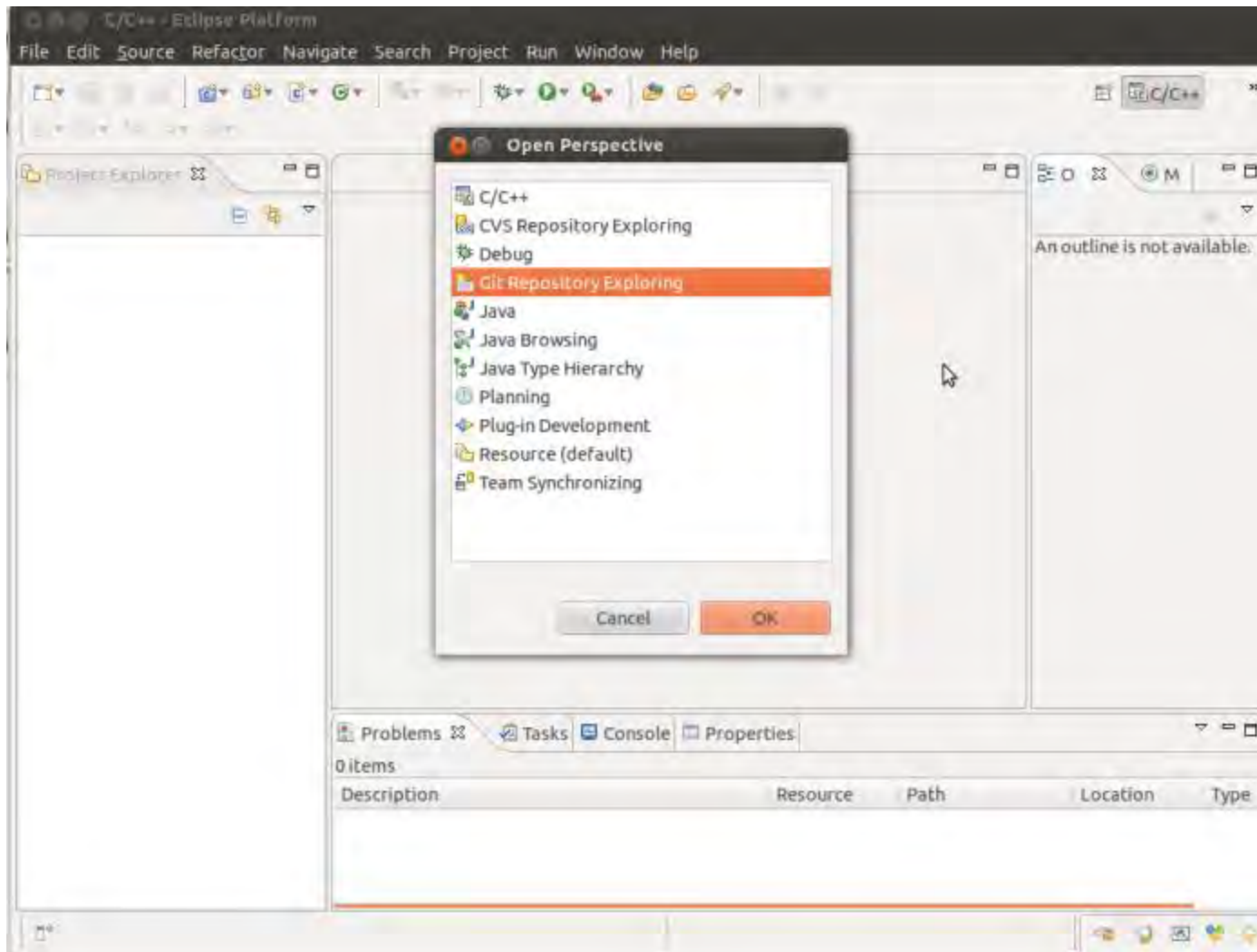
Step by step

You should see this after restart Eclipse



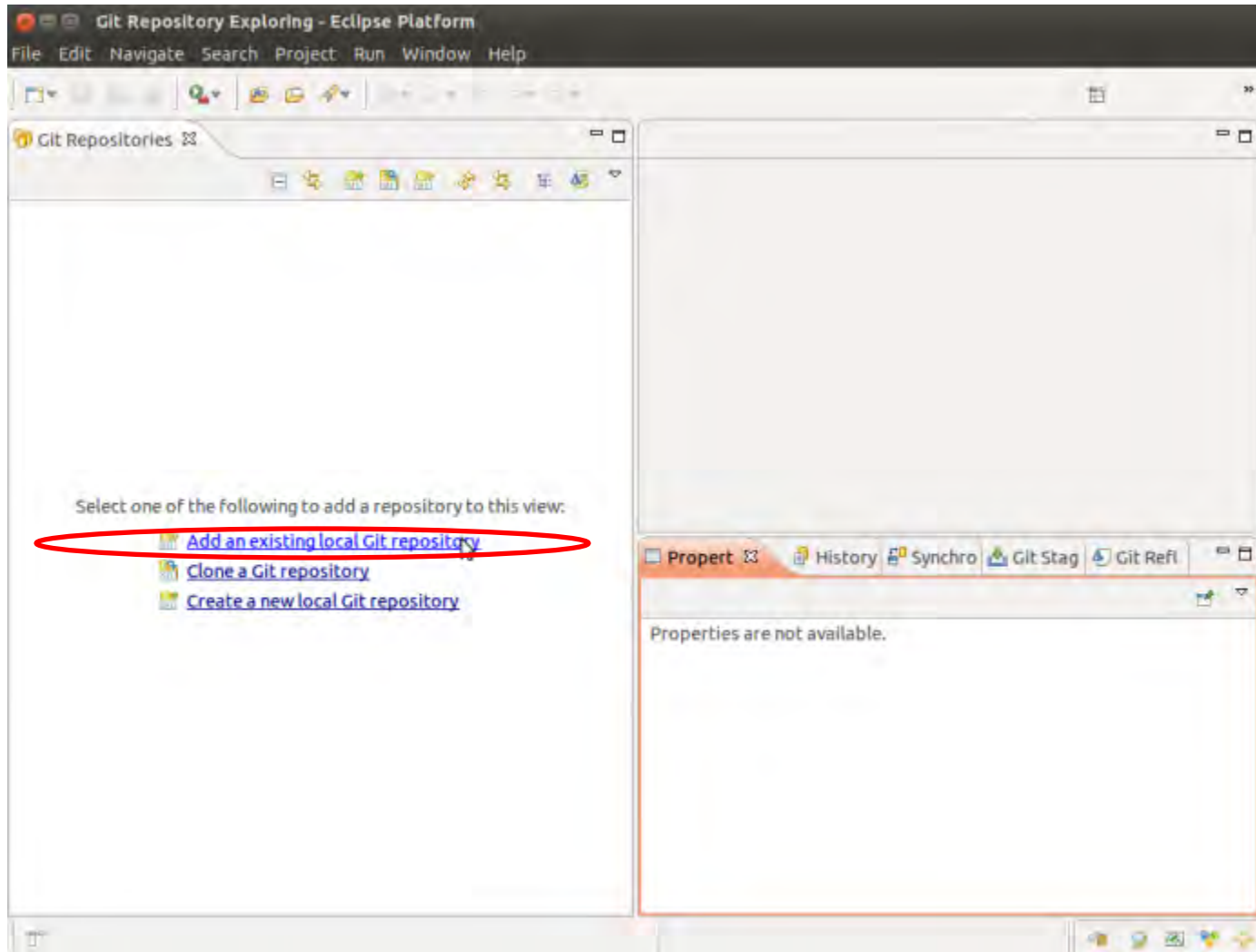
Step by step

8) Importing the Repositories into Eclipse

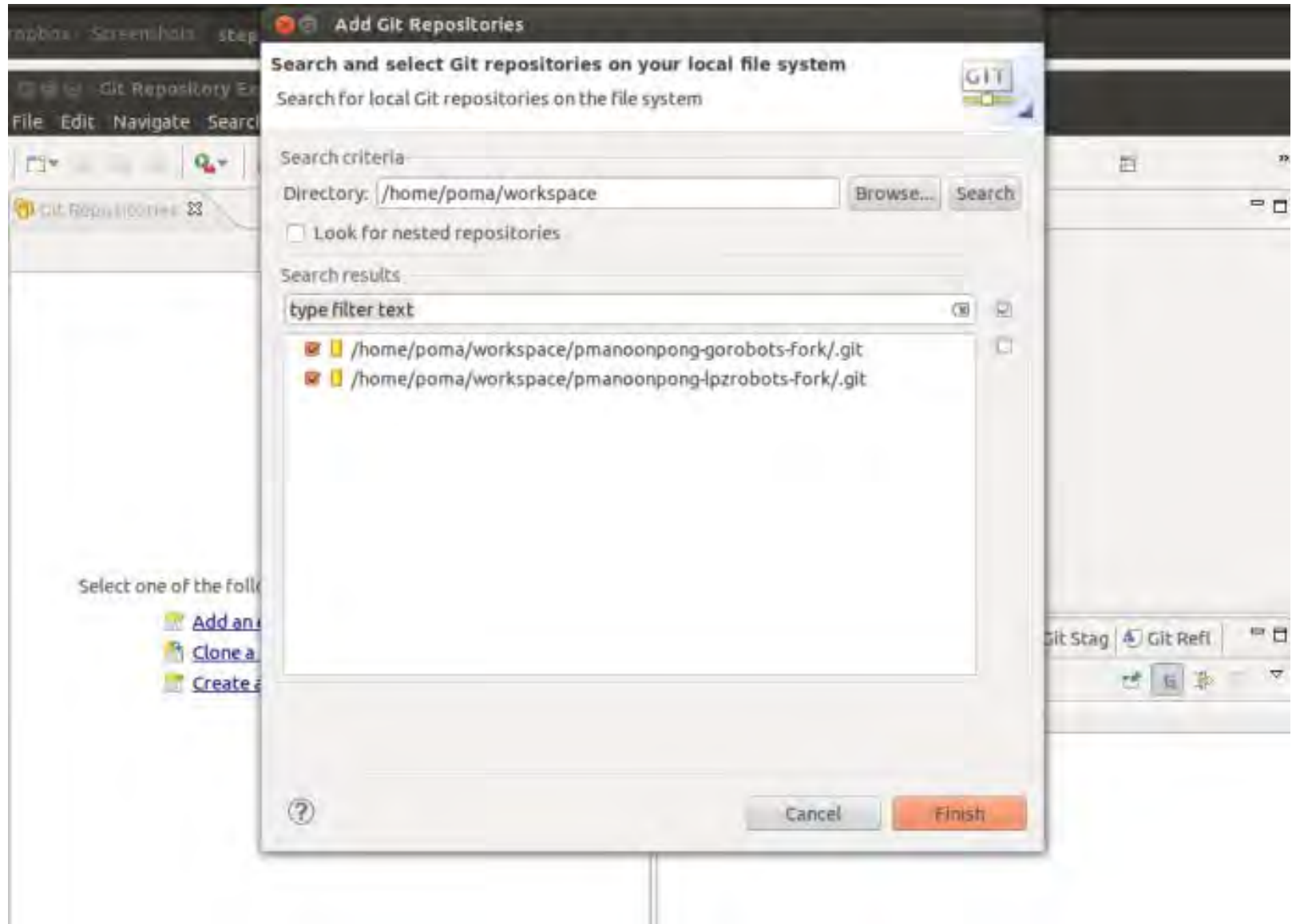


Step by step

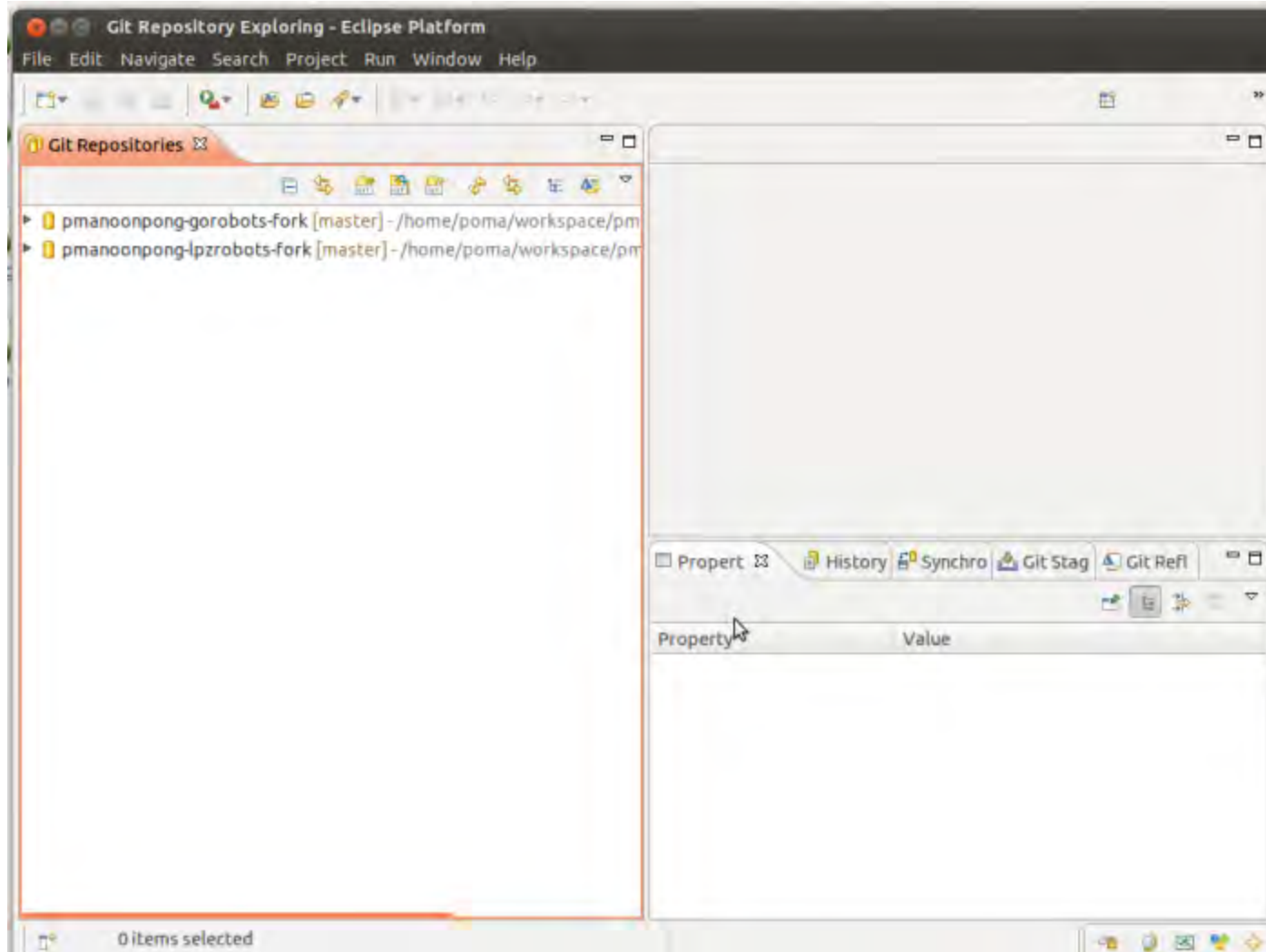
8) Importing the Repositories into Eclipse



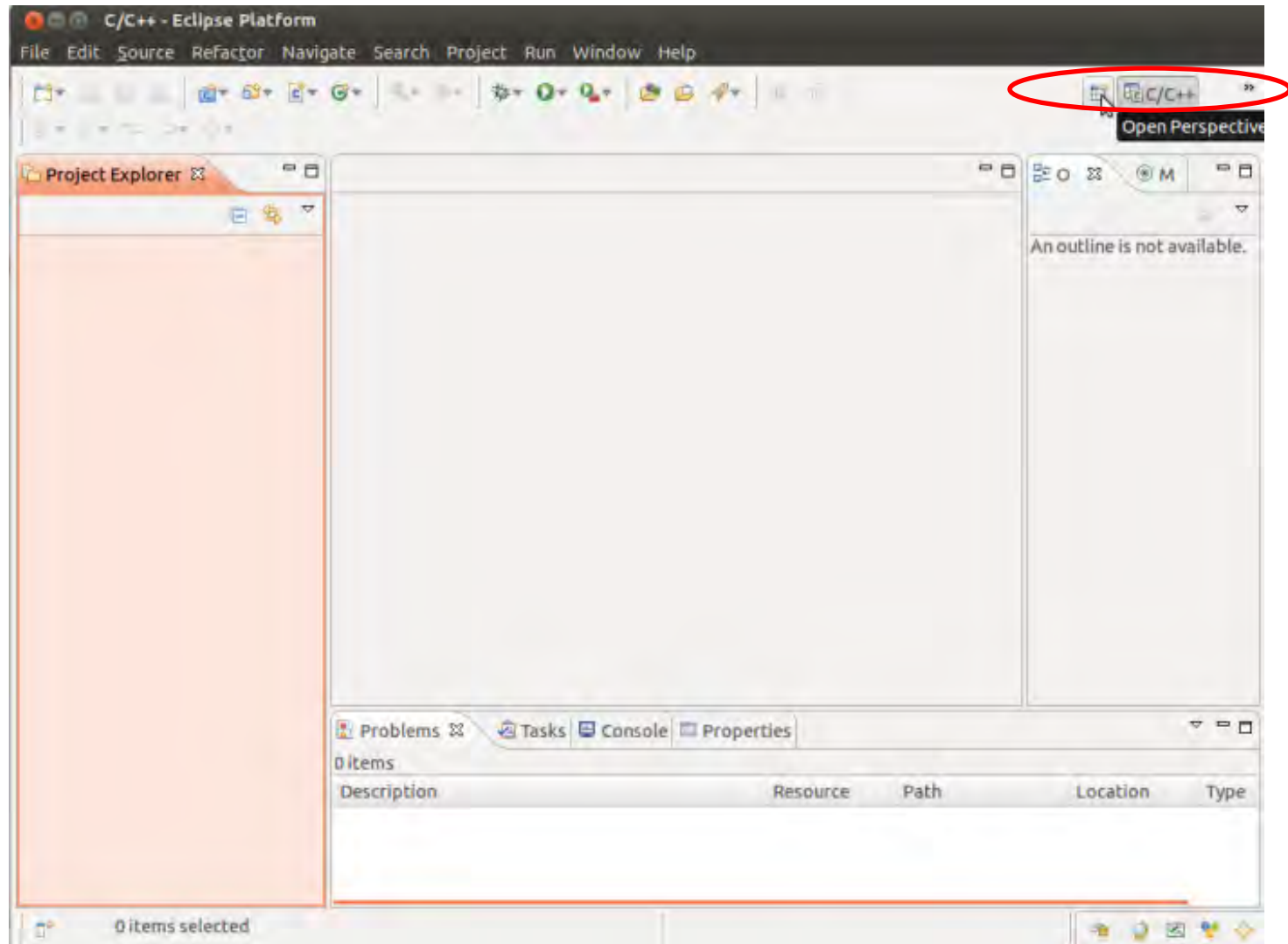
Step by step



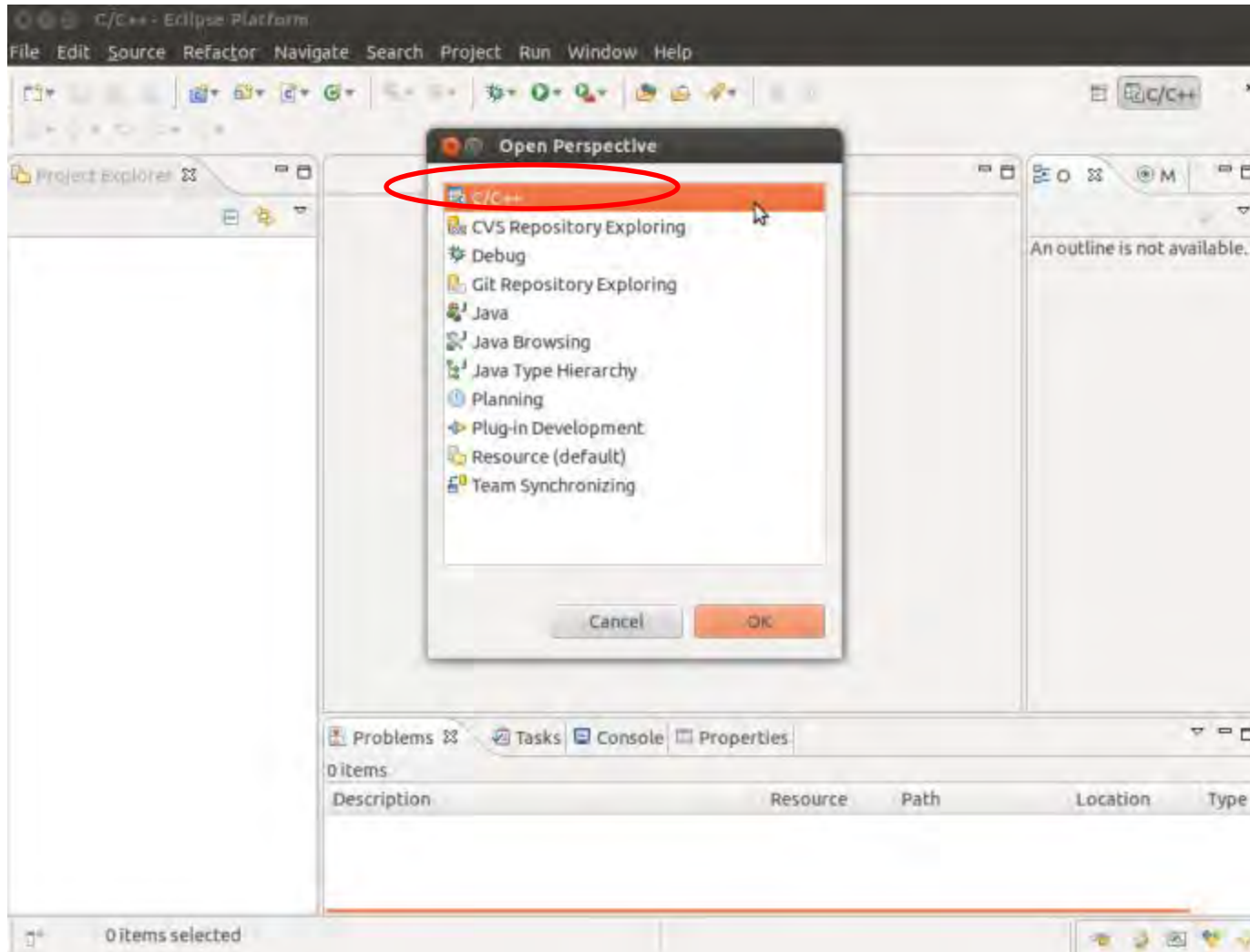
Step by step



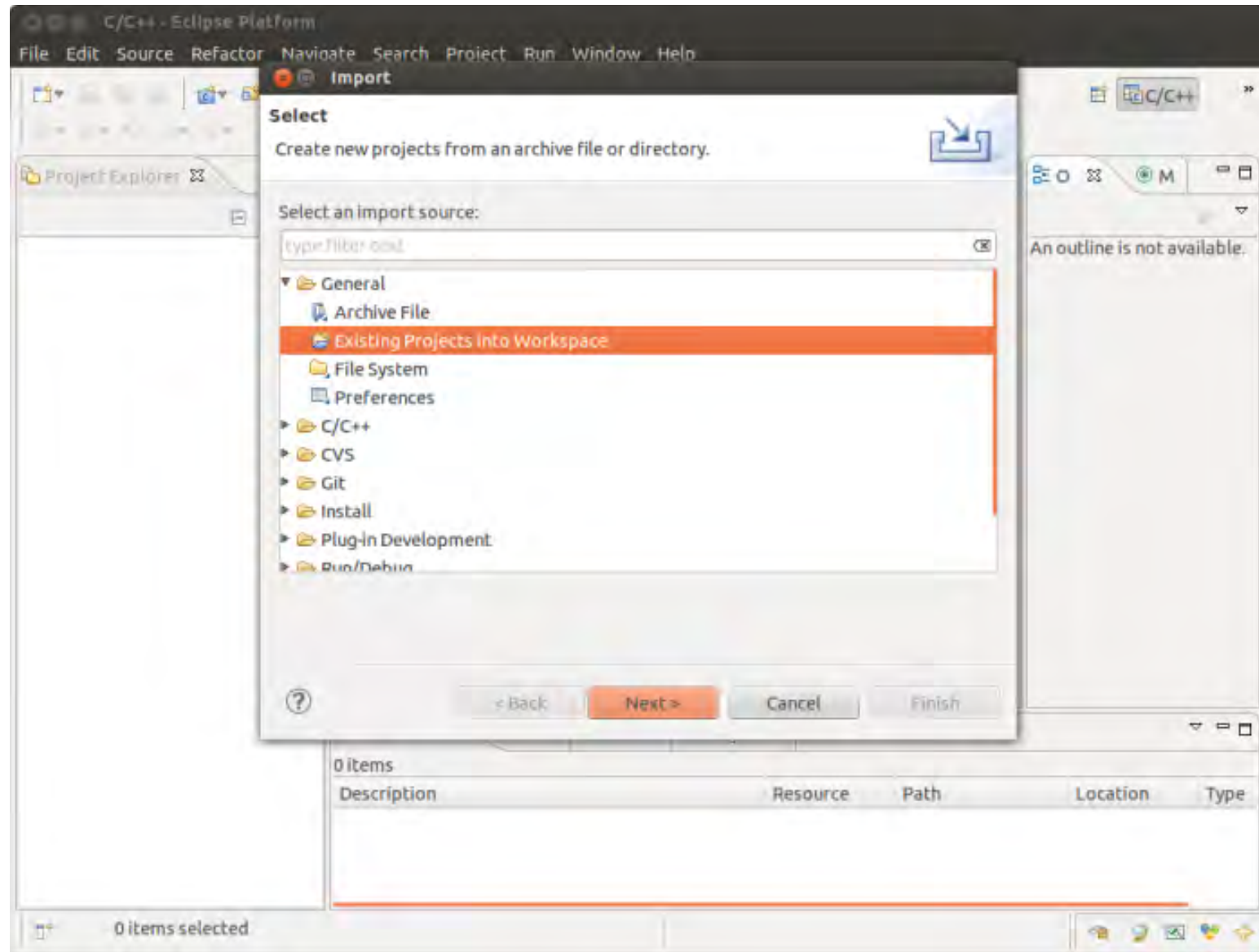
Step by step



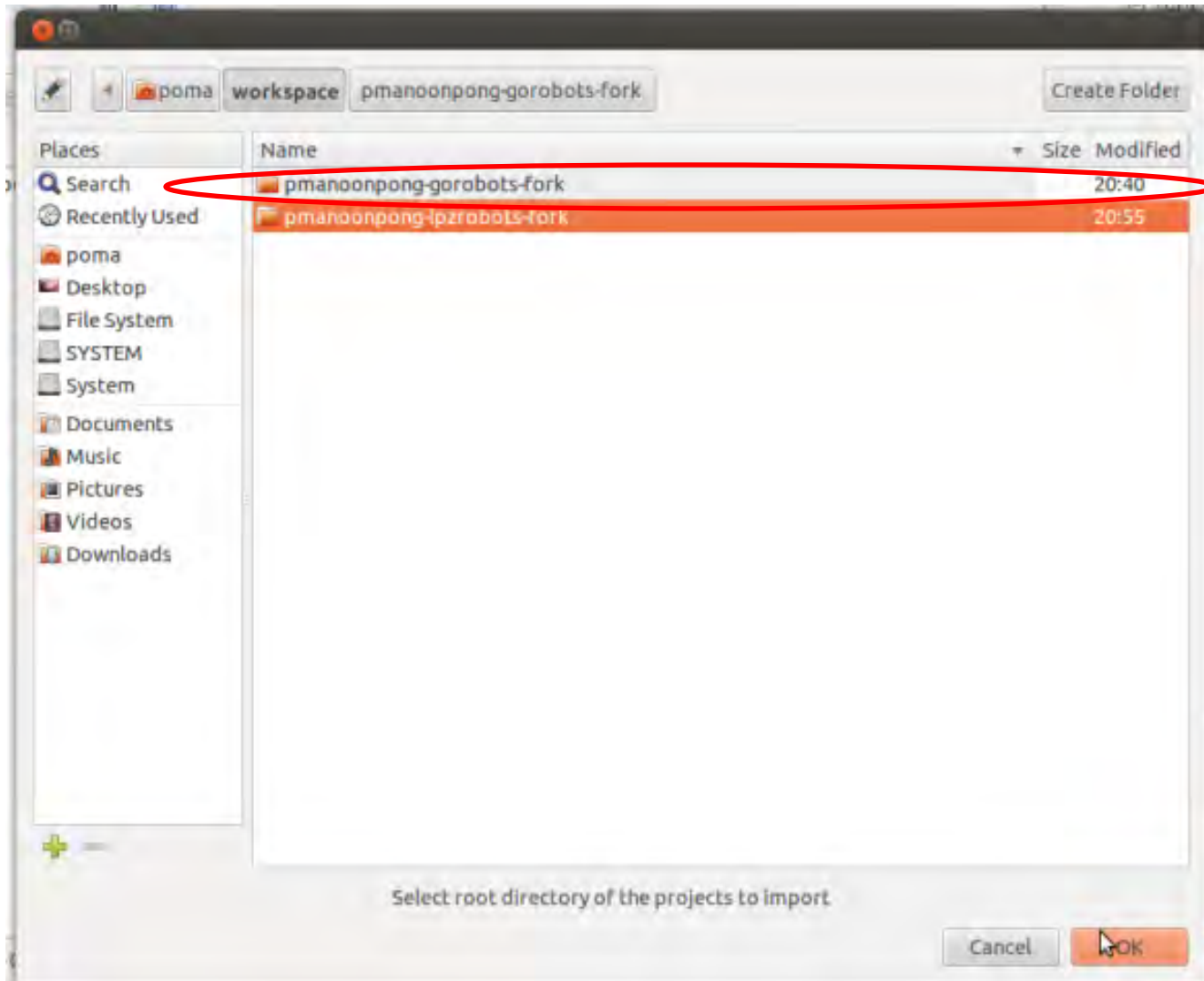
Step by step



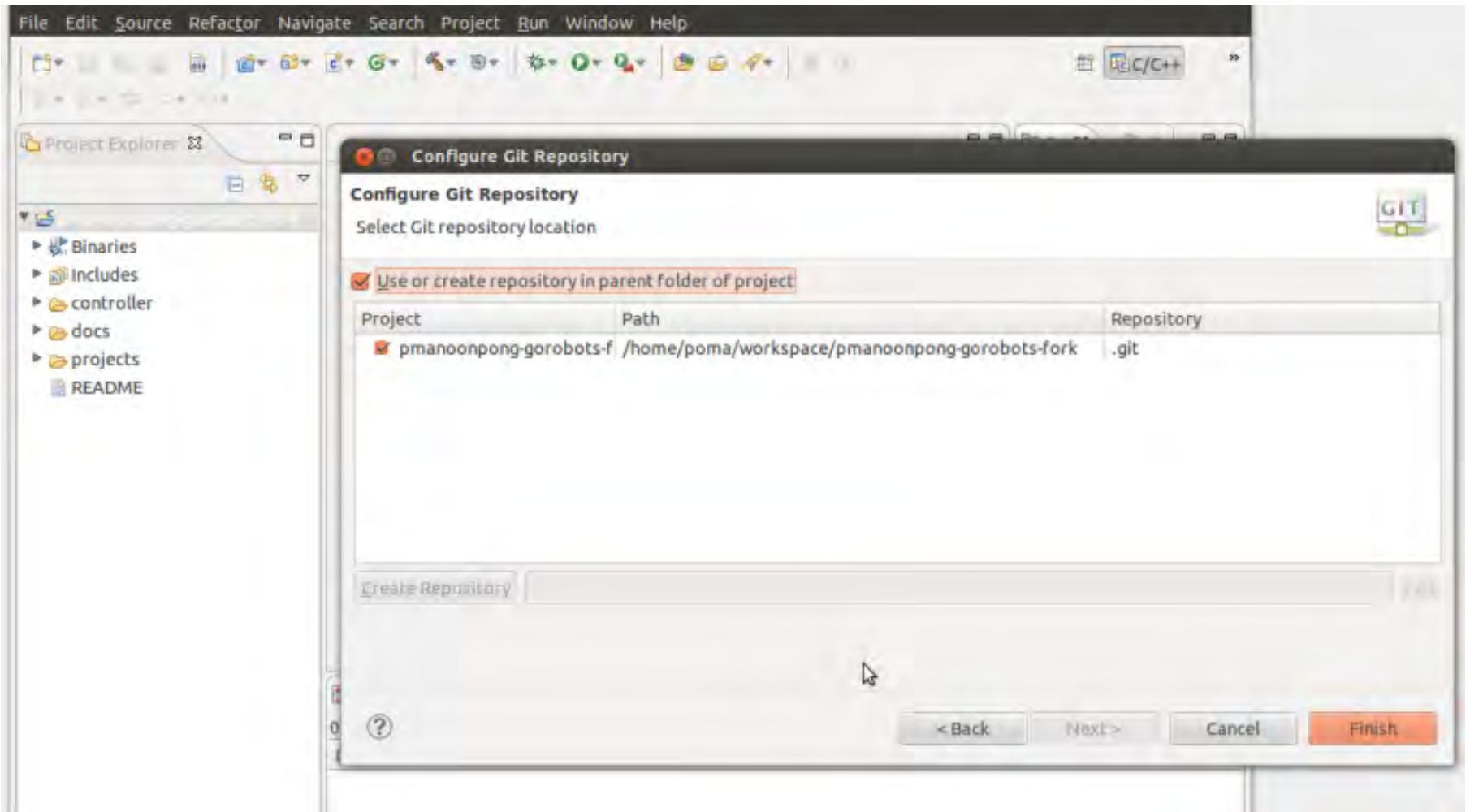
Step by step



Step by step



Step by step



Problems with GIT

Setting up Repositories in Eclipse

In some cases, the instructions on how to set up the GIT-repositories within Eclipse did not work.

Here is a different approach, that worked for me:

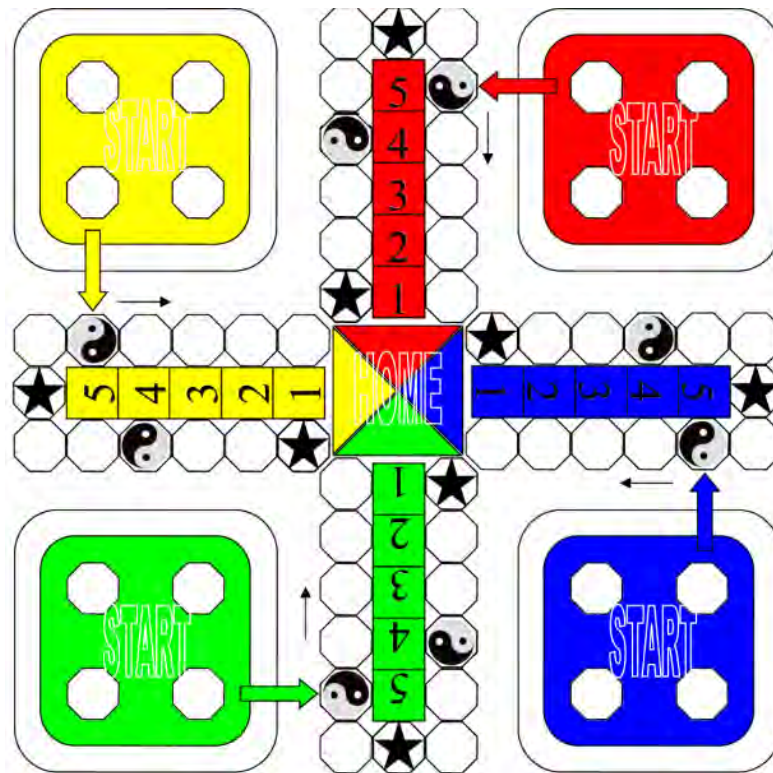
1. Import the repositories into the GIT-view of Eclipse, just as described before
2. Instead of importing over the GIT-View, you now go onto [File → Import → General → Existing Projects into Workspace](#) and you then choose the two repositories
3. After Eclipse has imported the files, you can [right-click](#) on the Project, and then select [Team → Share](#)
4. Now, just select [GIT](#) and the two GIT-repository-adresses should appear
5. [Apply](#)

Add other repositories!

- `git remote add name https://your
user@git.assembla.com/repository name`

```
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote  
add stable https://p.manoonpong@git.assembla.com/gorobots.git  
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote  
add gorobots_edu https://p.manoonpong@git.assembla.com/gorobots_edu.git  
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ cd ..  
poma@poma-HP-EliteBook-840-G1:~/workspace$ cd pmanoonpong-gorobots-fork/  
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote  
add xiaofeng https://p.manoonpong@git.assembla.com/lenonxiong-gorobots.git  
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote  
add sakya https://p.manoonpong@git.assembla.com/sakyad-gorobots-fork.git  
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote  
add robotdemo https://p.manoonpong@git.assembla.com/RobotDemo-gorobots-fork.git  
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote  
add ren https://p.manoonpong@git.assembla.com/sqrgj-gorobots-fork.git  
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote  
add eduard https://p.manoonpong@git.assembla.com/eduardg-gorobots-fork.git  
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote  
add dennis https://p.manoonpong@git.assembla.com/degoltschmidt-gorobots-fork.git
```

Java project: LUDO game



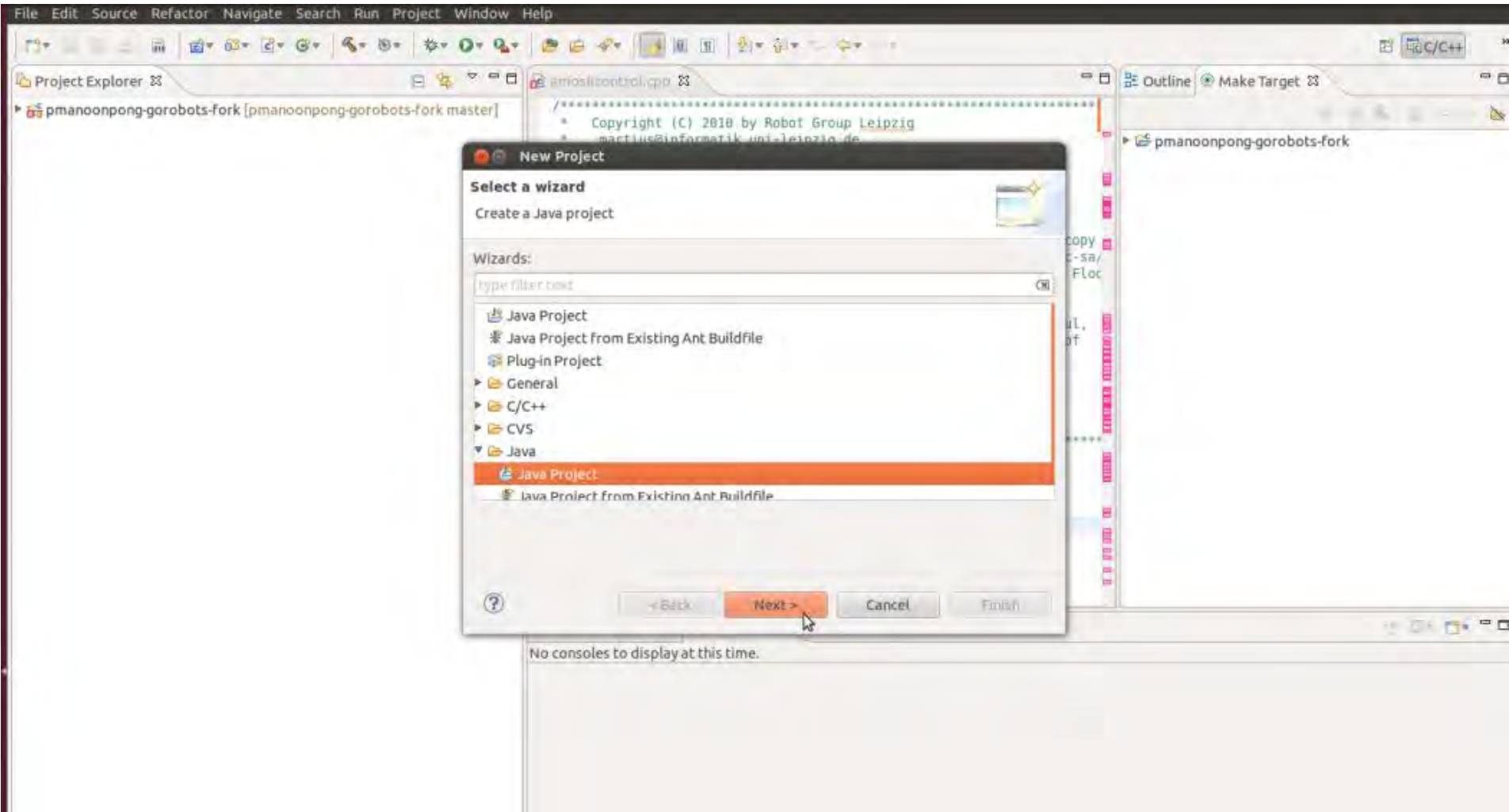
Step by step

9) Importing the LUDO project into Eclipse

→ Extract workspacejava in home directory:
:~/workspacejava/LUDO

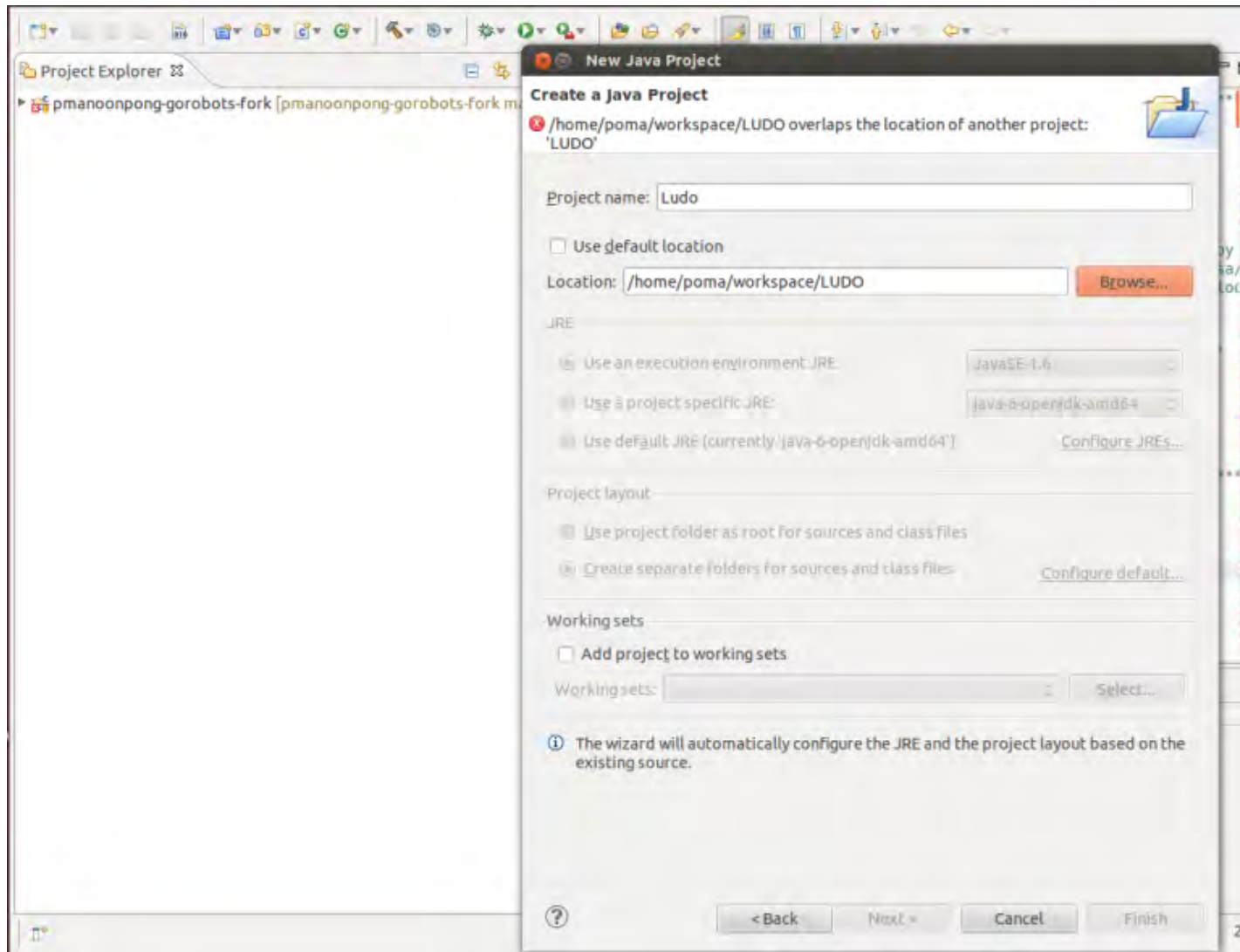
Step by step

File>>New Project>> Java project

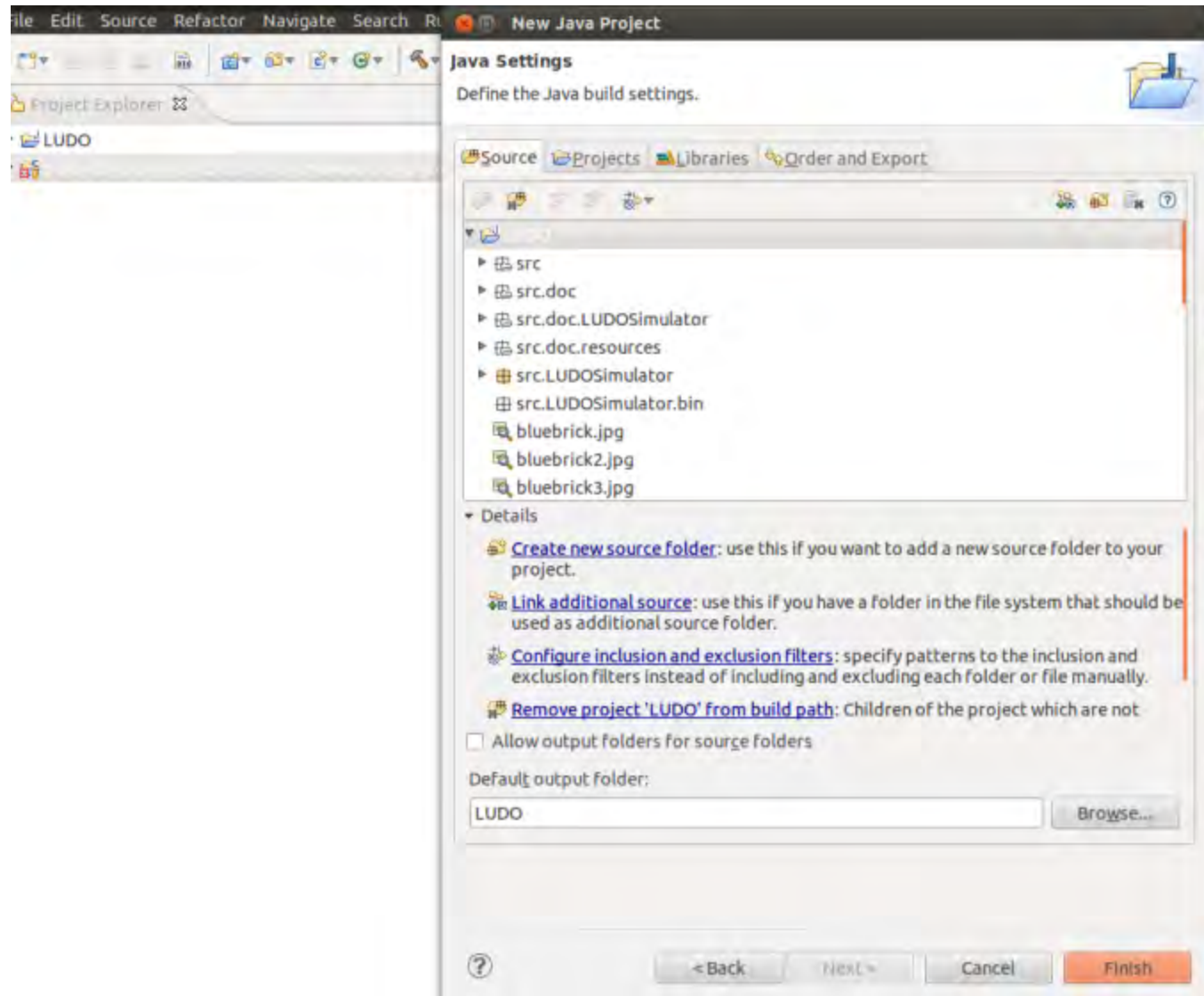


Step by step

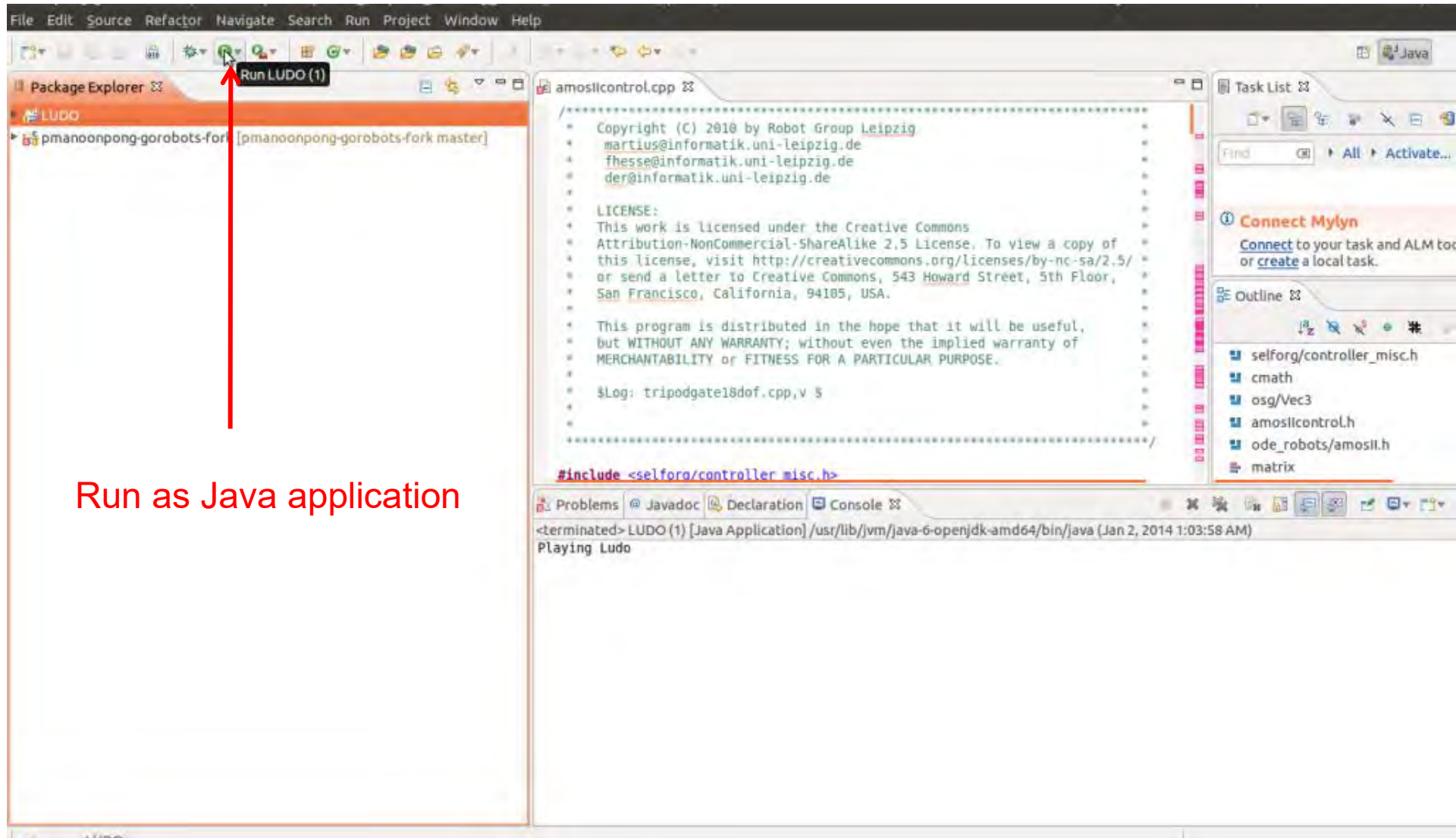
Browse to /home/your login/workspace/LUDO



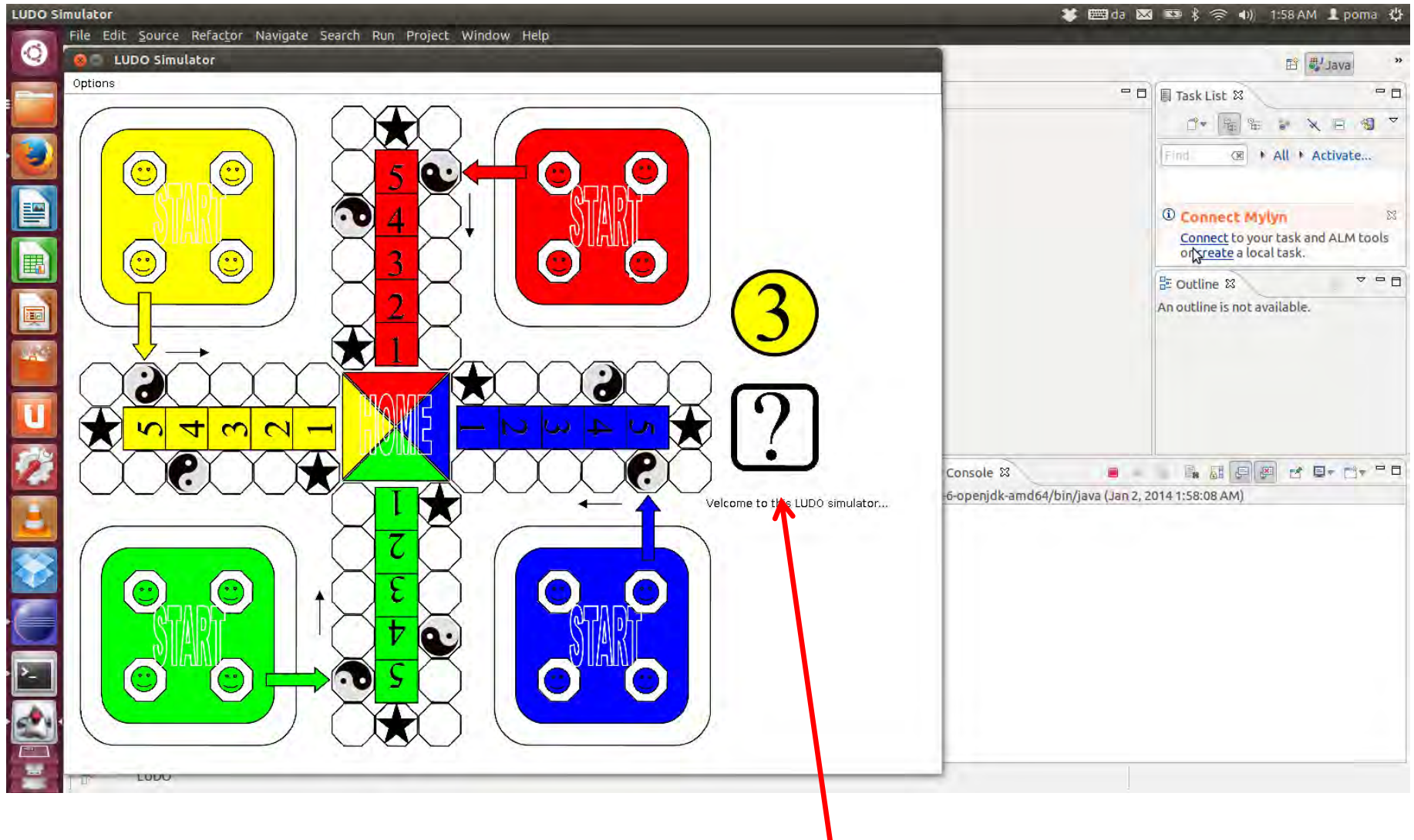
Step by step



Step by step



Step by step



Click die to start the game

For a 64 bit PC

You might need to update gcc

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test
```

```
sudo apt-get update sudo apt-get install gcc-4.8
```

```
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.8 50
```

Upgrade your Ubuntu system to avoid system freeze

```
apt-get dist-upgrade
```

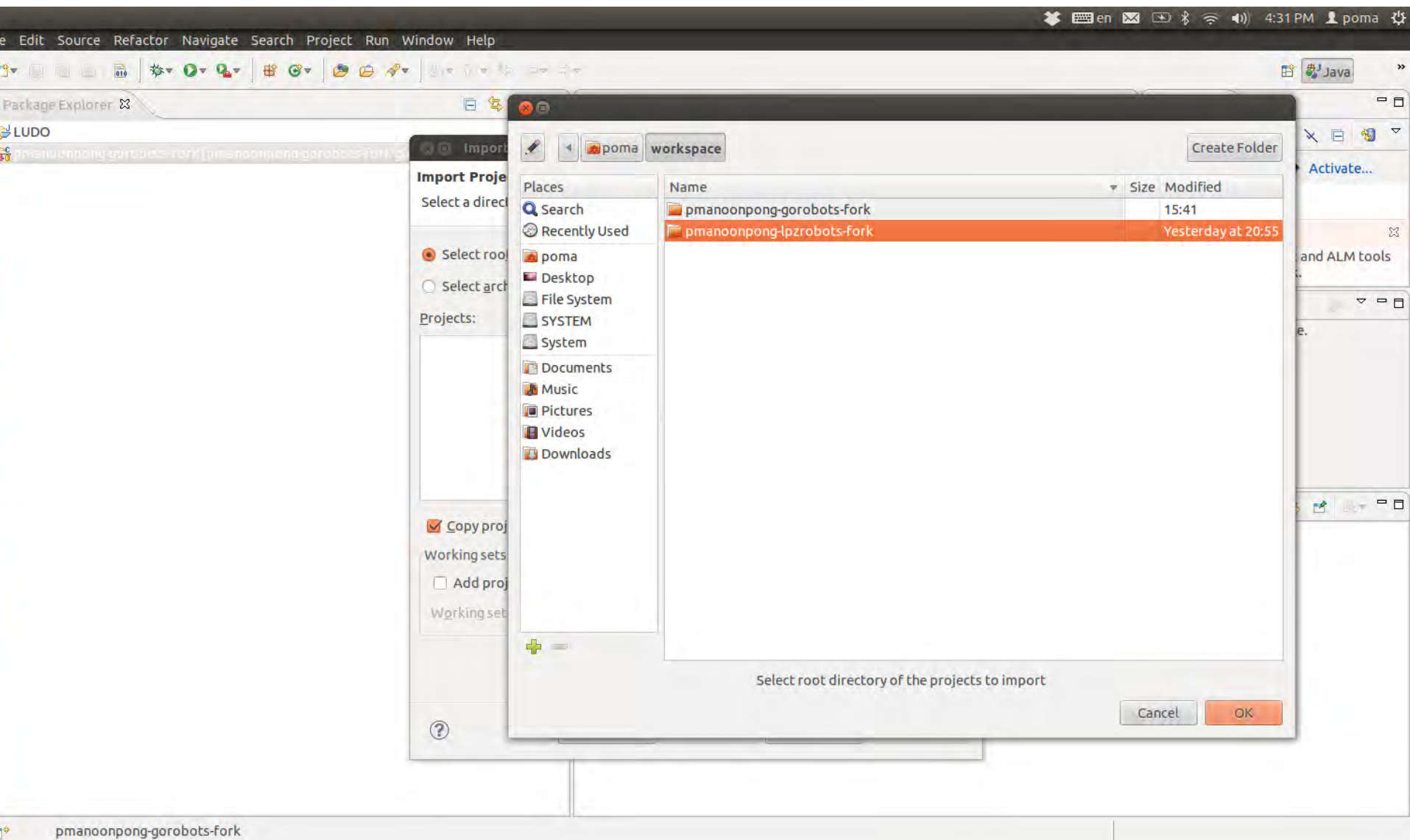
Such a problem might occur when compiling a program

```
./start: symbol lookup error: /usr/lib/libgsl.so.0: undefined symbol:  
cblas_dnrm2
```

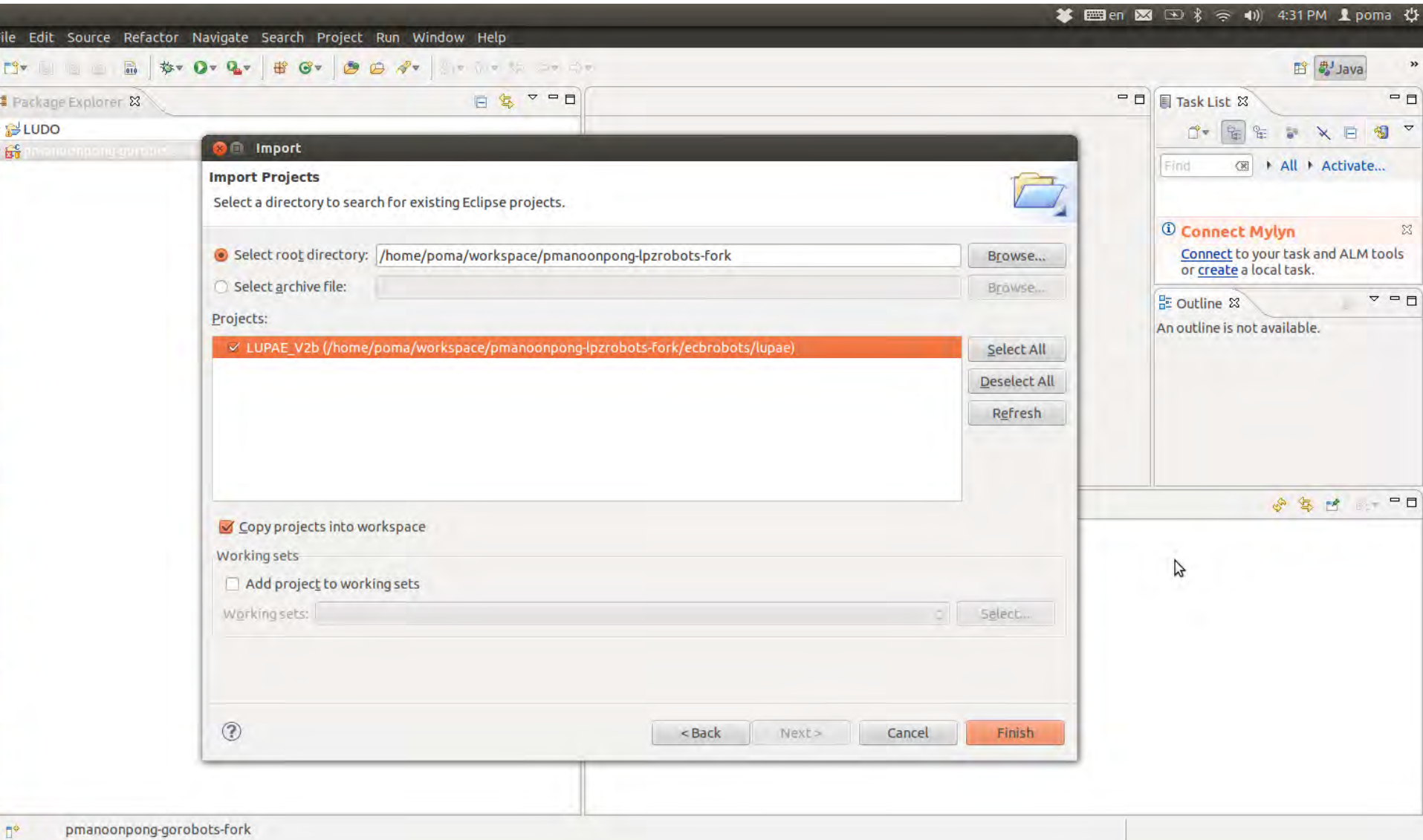
solve by adding this in Makefile.conf

```
“ LIBS = -Wl,--no-as-needed -lgsl -lgslcblas -lm “
```

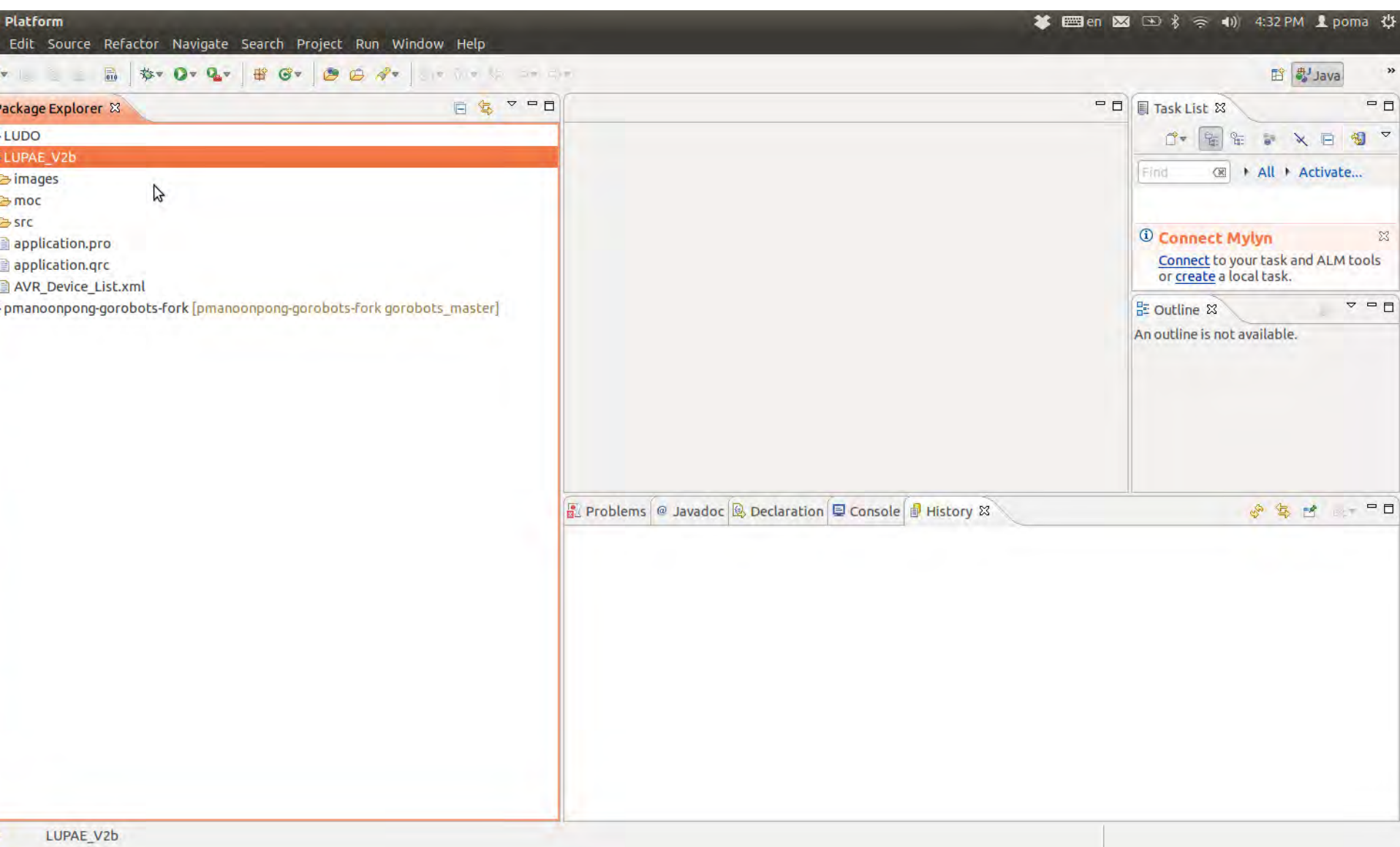
Problem



Problem

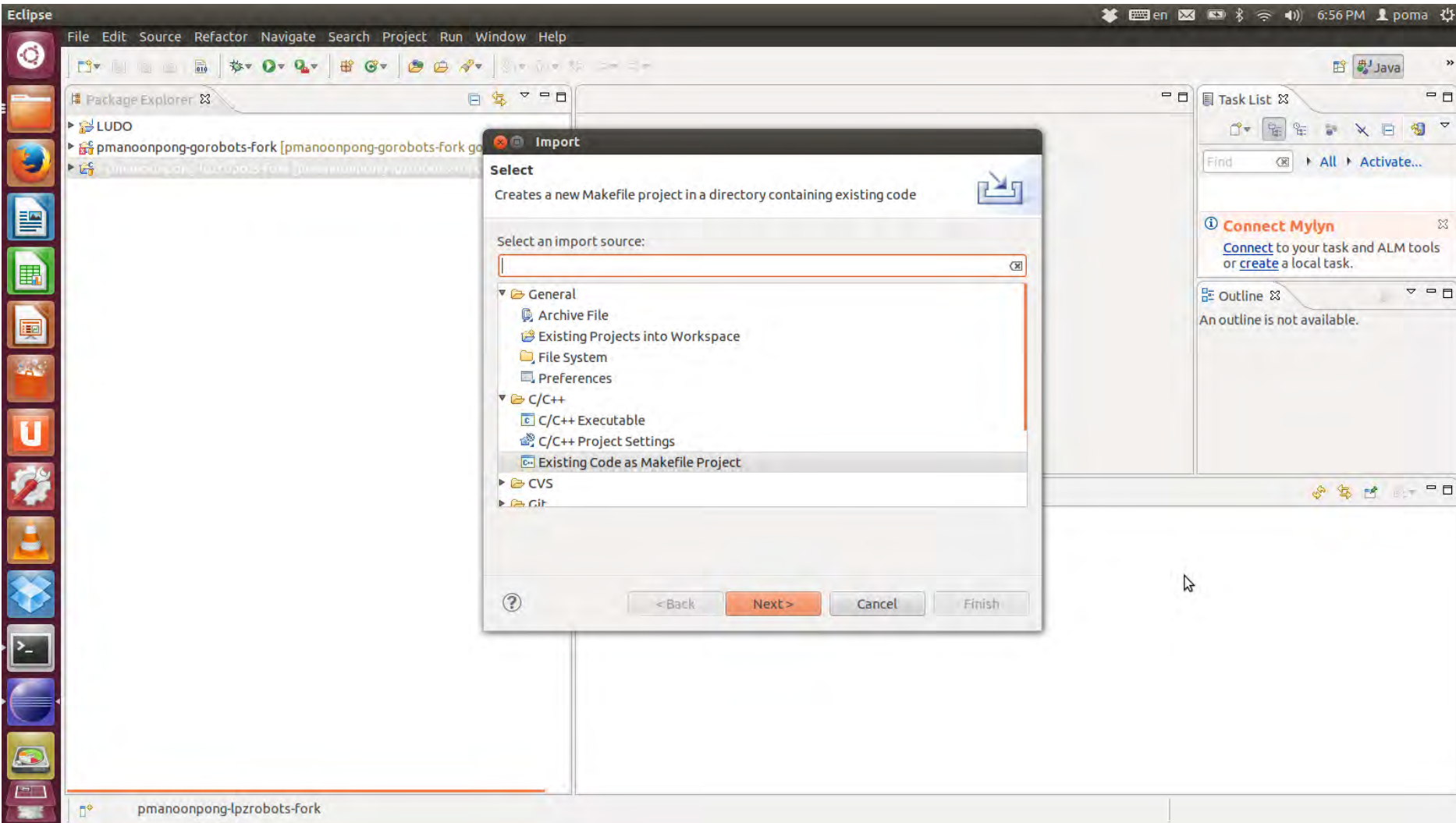


Problem



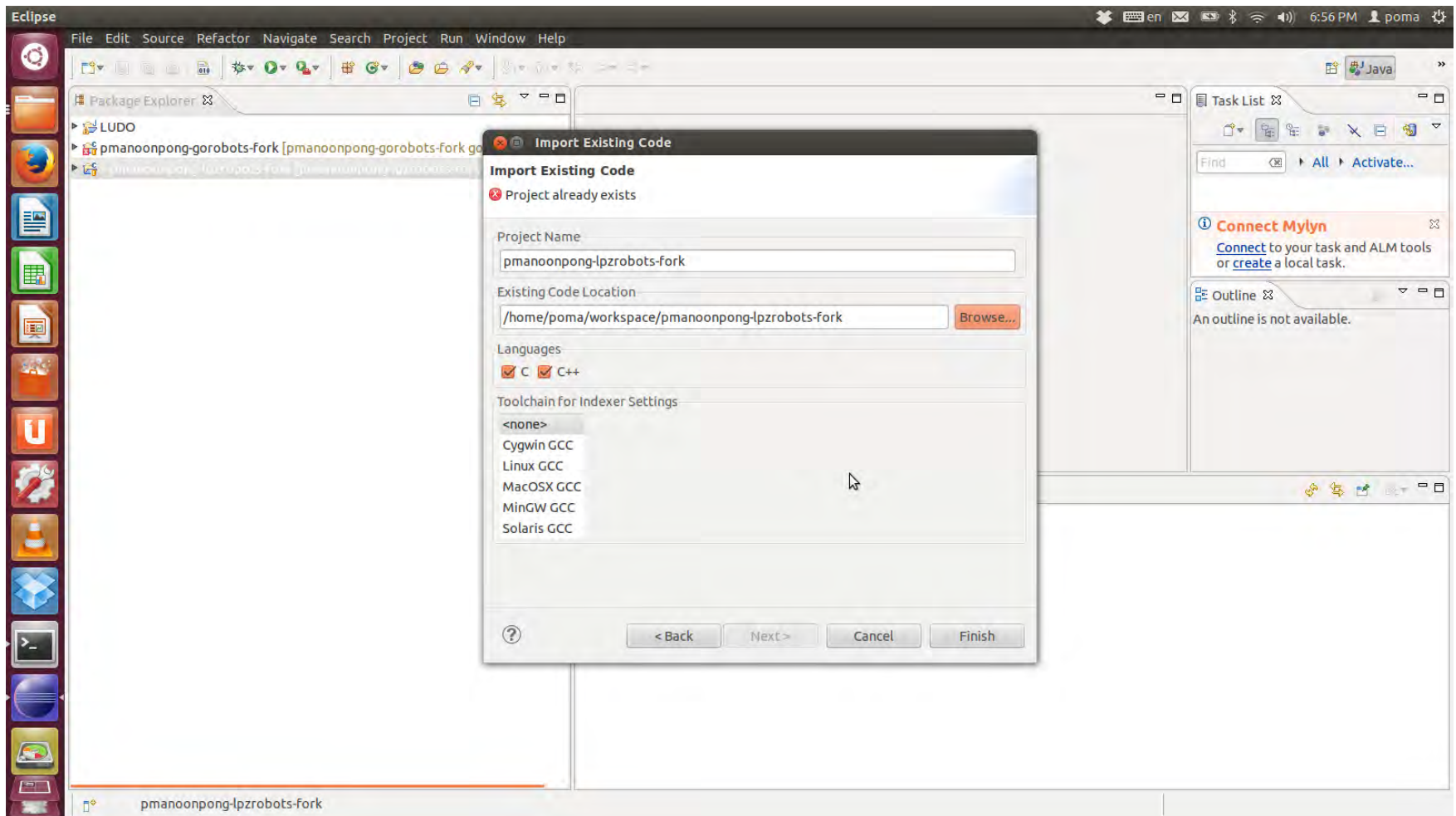
Solution

- File->Import->C/C++->Existing Code as Makefile Project.

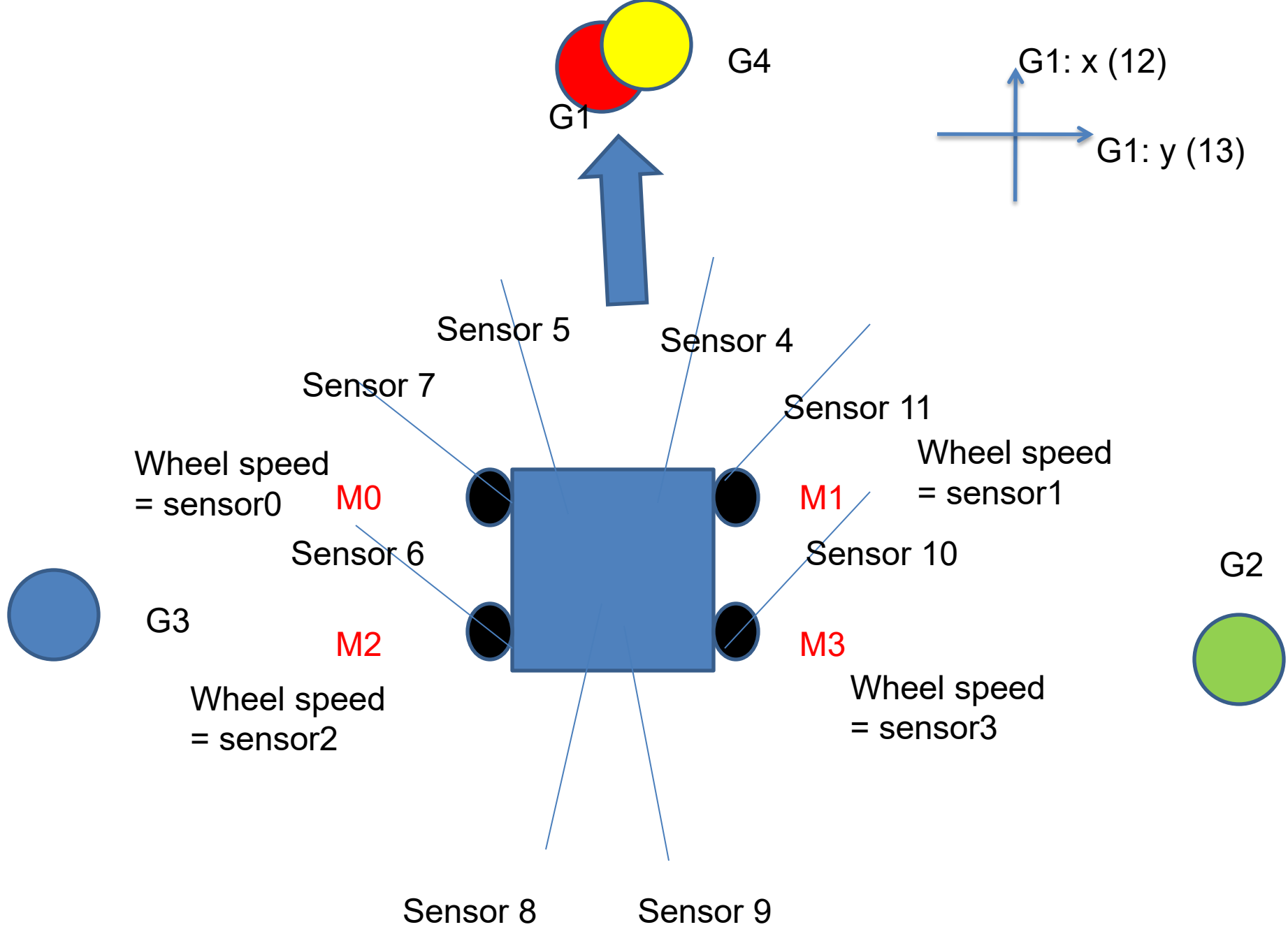


Solution

- File->Import->C/C++->Existing Code as Makefile Project.



Nimm4 system



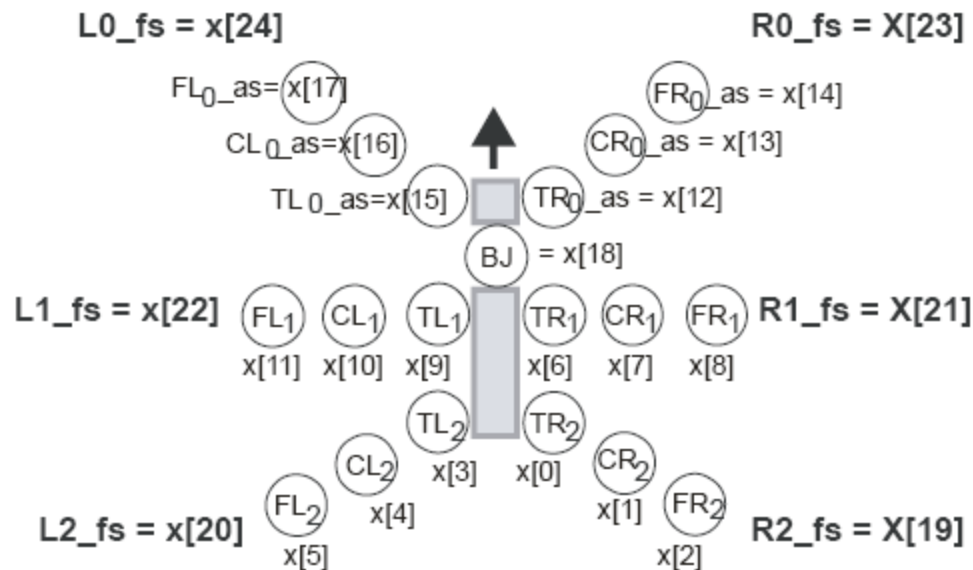
Hexapod system

AMOS simulation

Sensors:

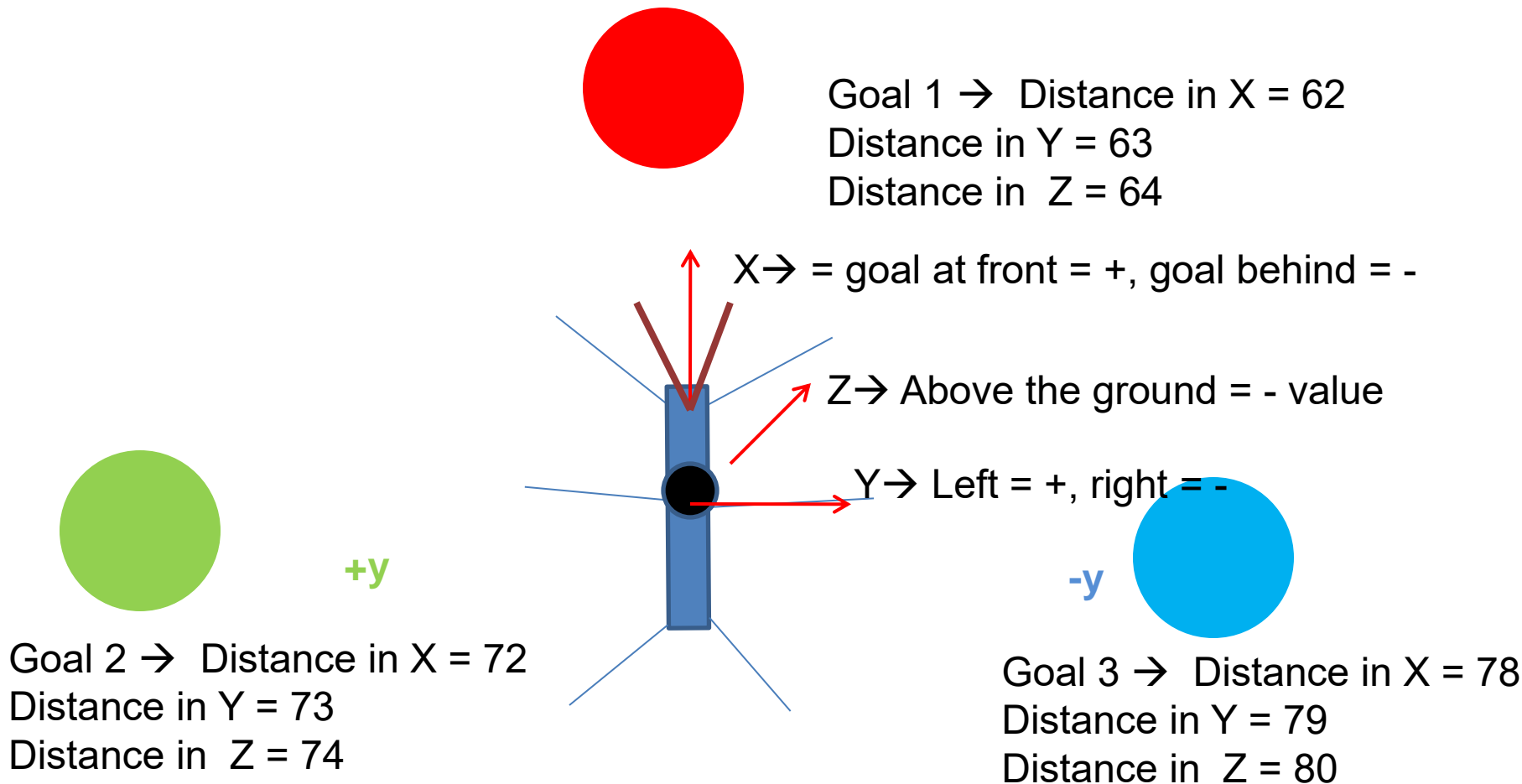
`_as` = angle sensors = $[-1, \dots, 1]$; -1 = backward, down, flex; +1 = forward, up, extend

`_fs` = foot sensors = $[0, \dots, 1]$; 0 = swing (off ground); +1 = stance (touch ground)

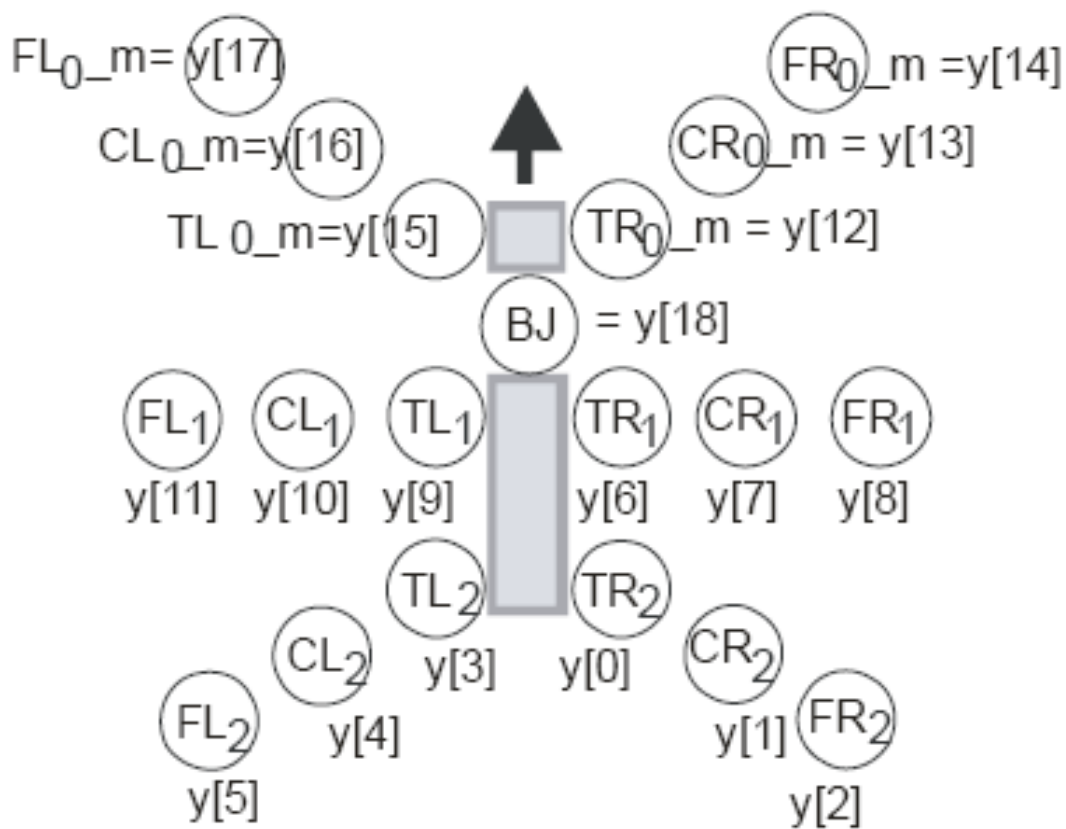


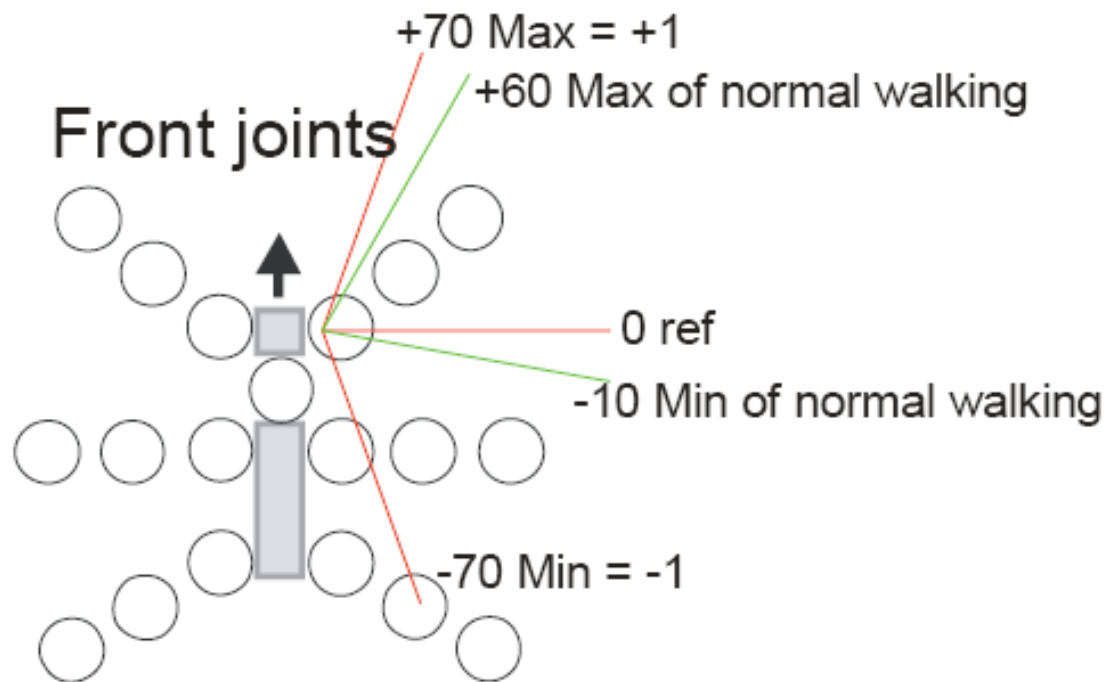
Sensors

- Goal detection sensors

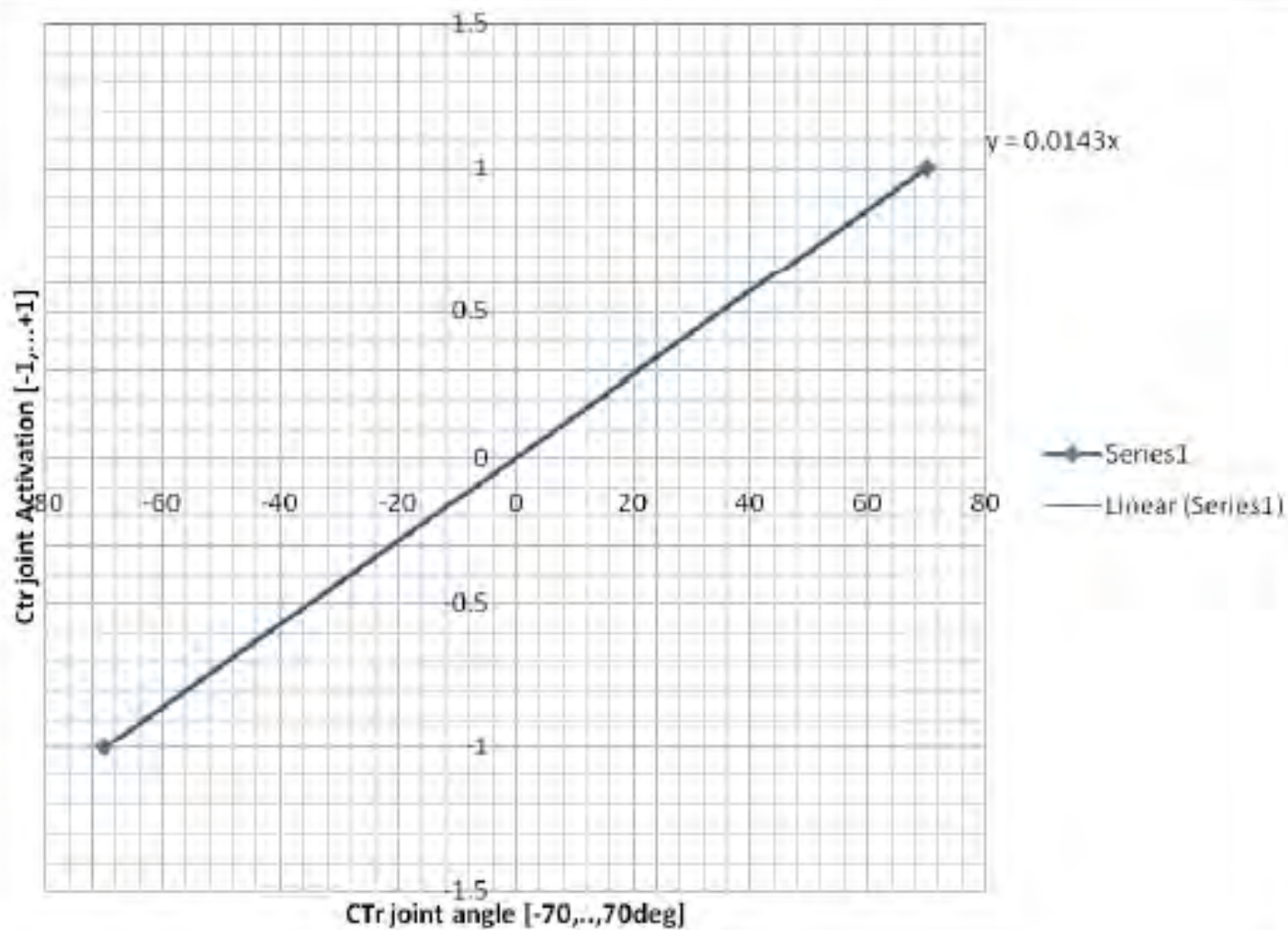


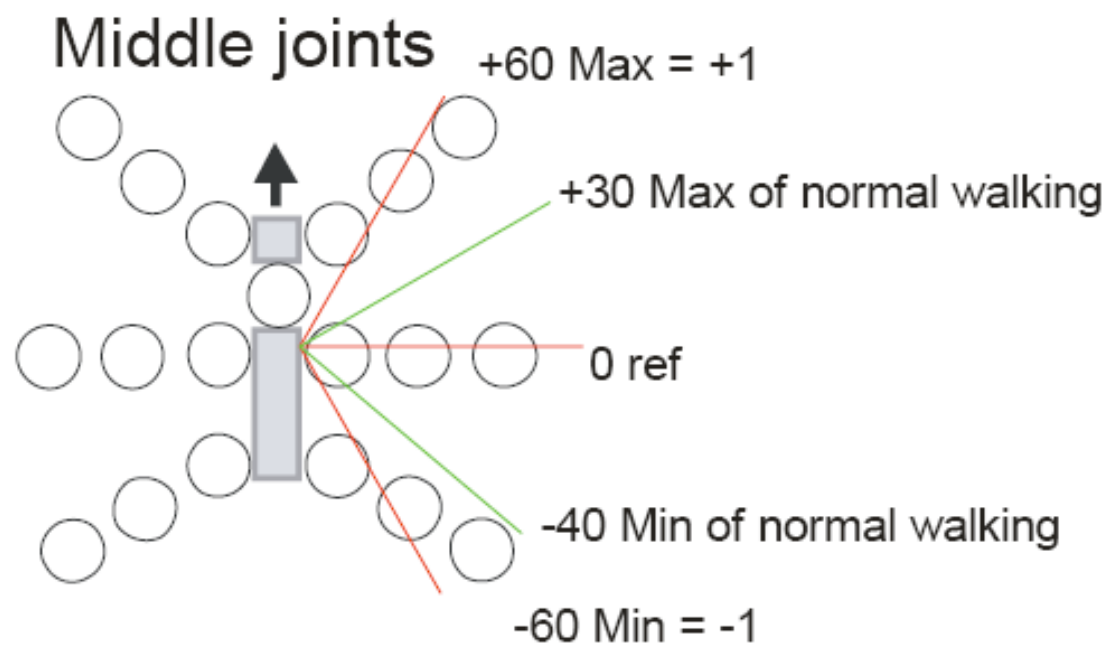
Motors:



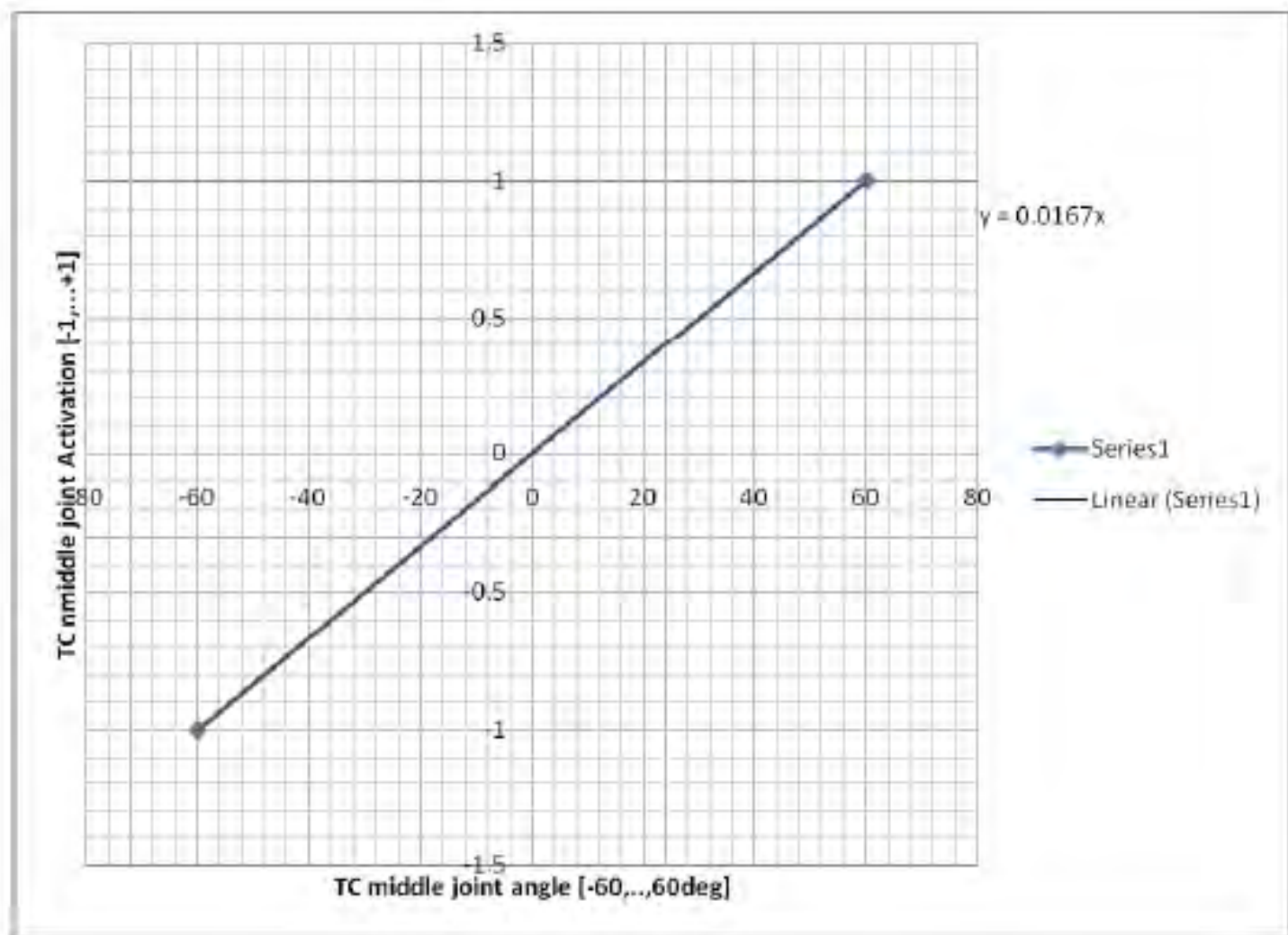


Deg	Activation
-70	-1.001
-60	-0.858
-50	-0.715
-40	-0.572
-30	-0.429
-20	-0.286
-10	-0.143
0	0
10	0.143
20	0.286
30	0.429
40	0.572
50	0.715
60	0.858
70	1.001

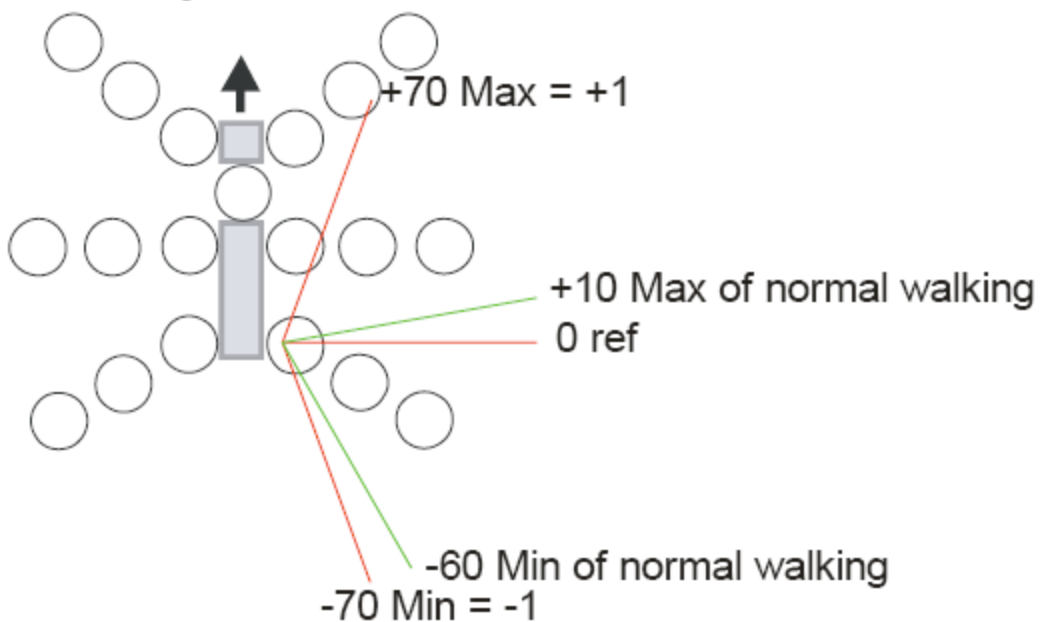




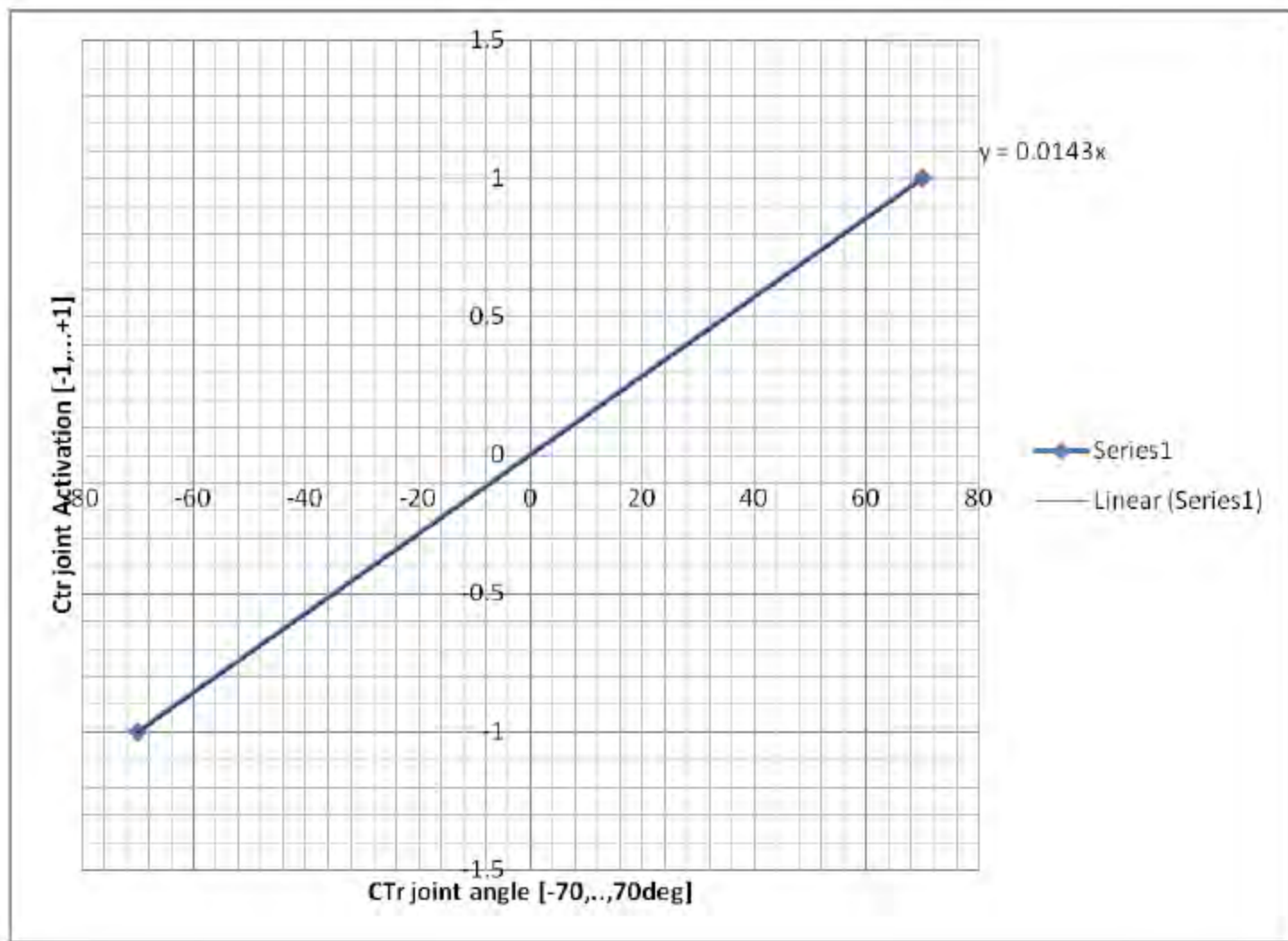
Deg	Activation
-60	-1.002
-50	-0.835
-40	-0.668
-30	-0.501
-20	-0.334
-10	-0.167
0	0
10	0.167
20	0.334
30	0.501
40	0.668
50	0.835
60	1.002

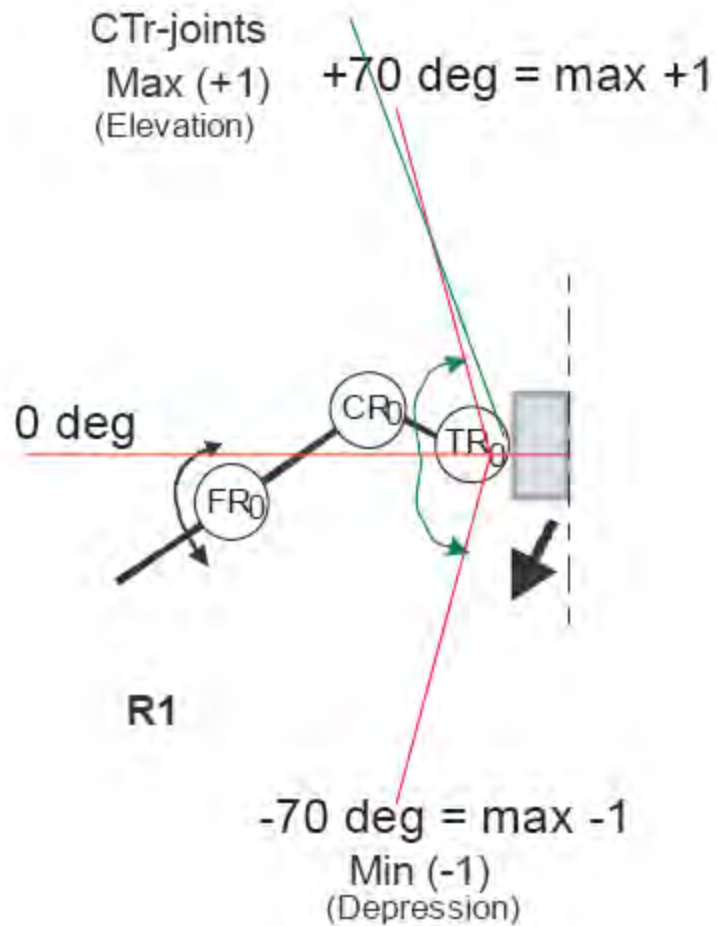


Hind joints

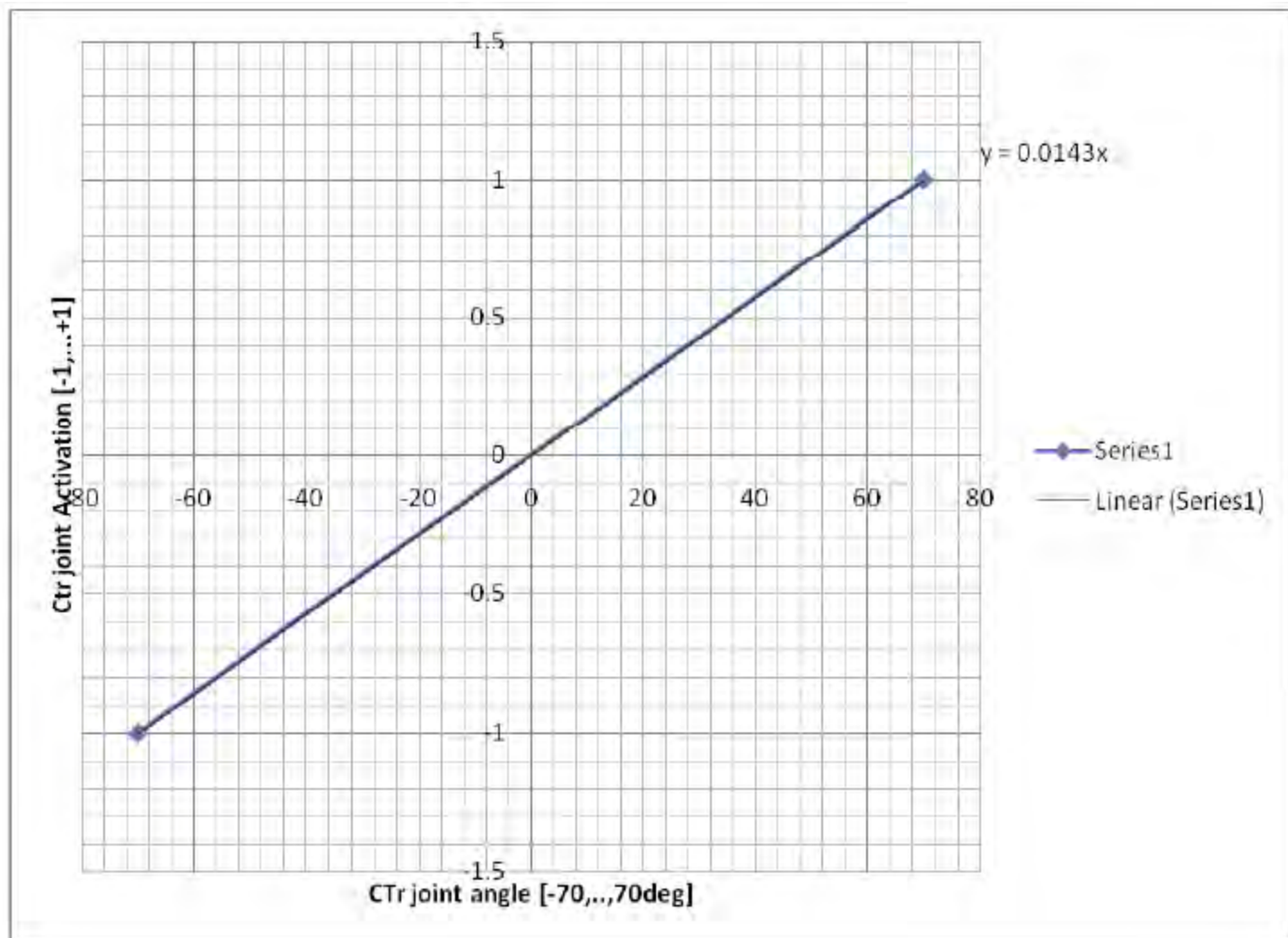


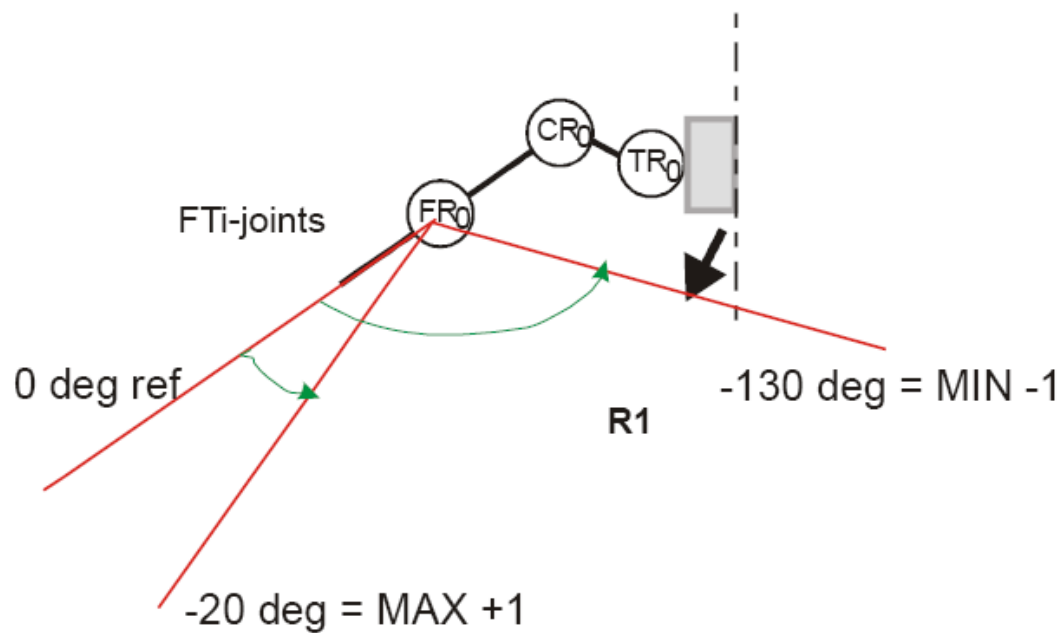
Deg	Activation
-70	-1.001
-60	-0.858
-50	-0.715
-40	-0.572
-30	-0.429
-20	-0.286
-10	-0.143
0	0
10	0.143
20	0.286
30	0.429
40	0.572
50	0.715
60	0.858
70	1.001



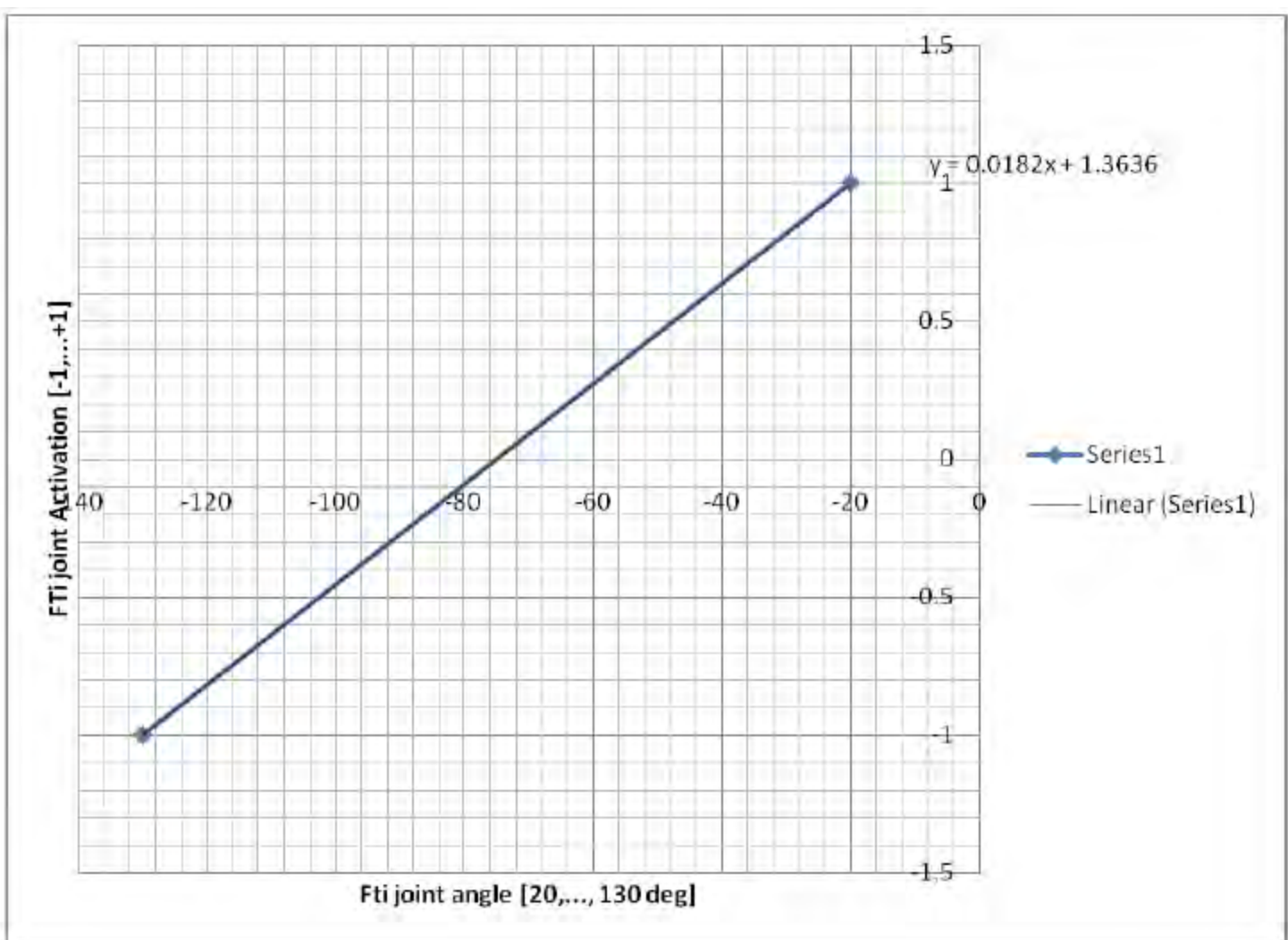


Deg	Activation
-70	-1.001
-60	-0.858
-50	-0.715
-40	-0.572
-30	-0.429
-20	-0.286
-10	-0.143
0	0
10	0.143
20	0.286
30	0.429
40	0.572
50	0.715
60	0.858
70	1.001





Activati on	
-20	0.9996
-30	0.8176
-40	0.6356
-50	0.4536
-60	0.2716
-70	0.0896
-80	-0.0924
-90	-0.2744
-100	-0.4564
-110	-0.6384
-120	-0.8204
-130	-1.0024



Installation: Eclipse editor & compiler

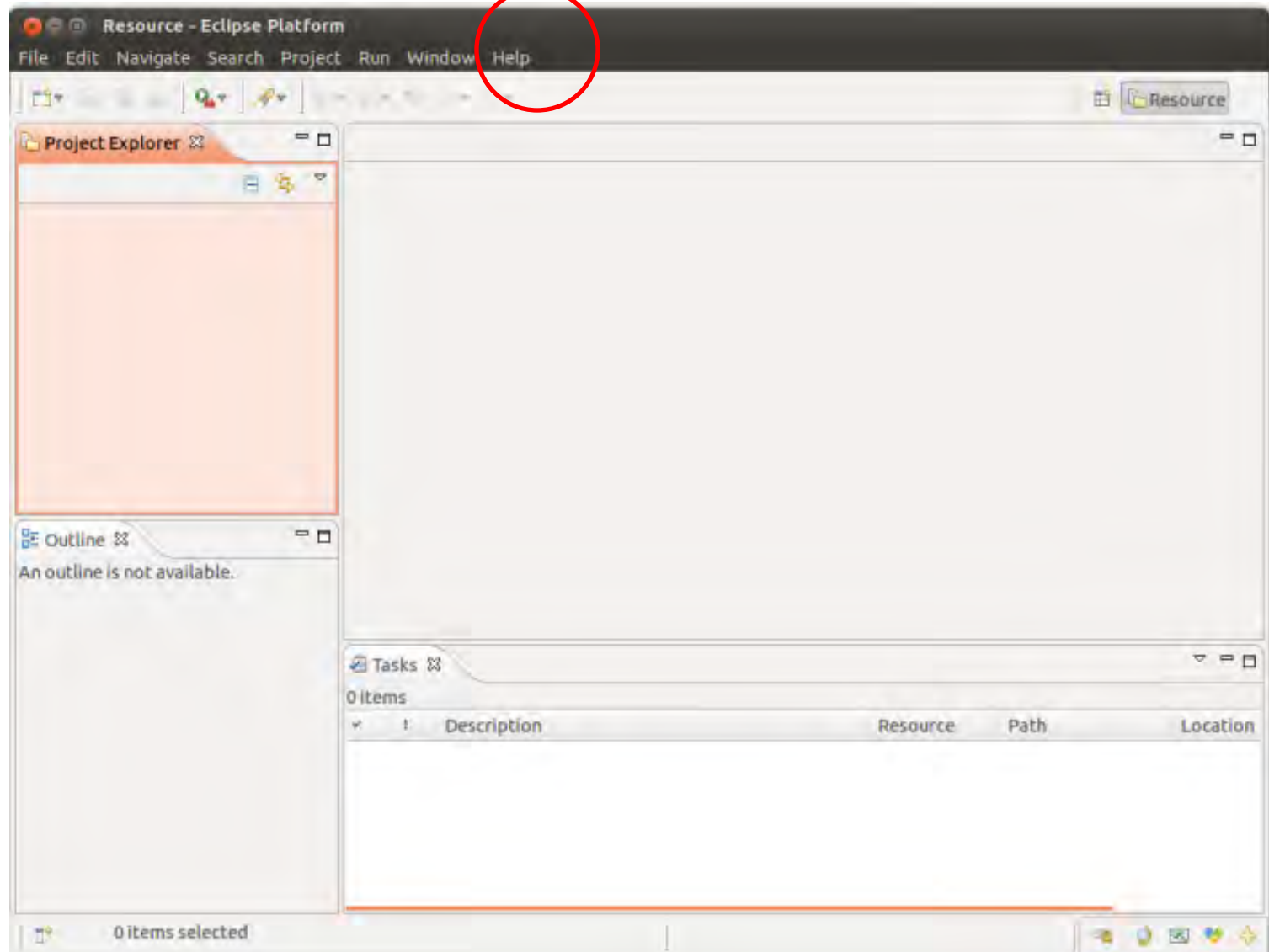


Step by step

7) Setup editor, e.g., Eclipse → Install Eclipse from Ubuntu Software Center. If you already installed, then just run Eclipse

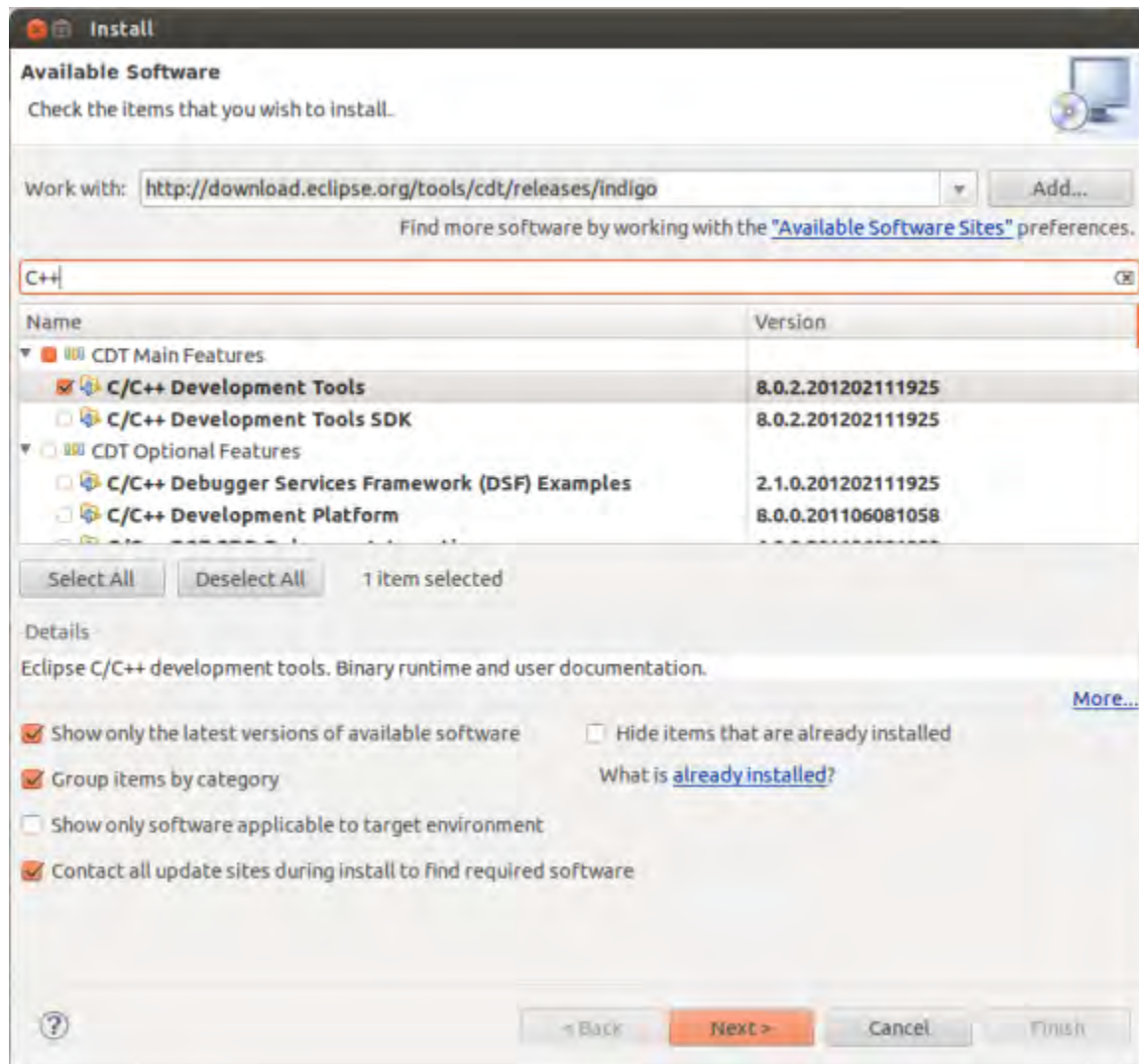


Step by step



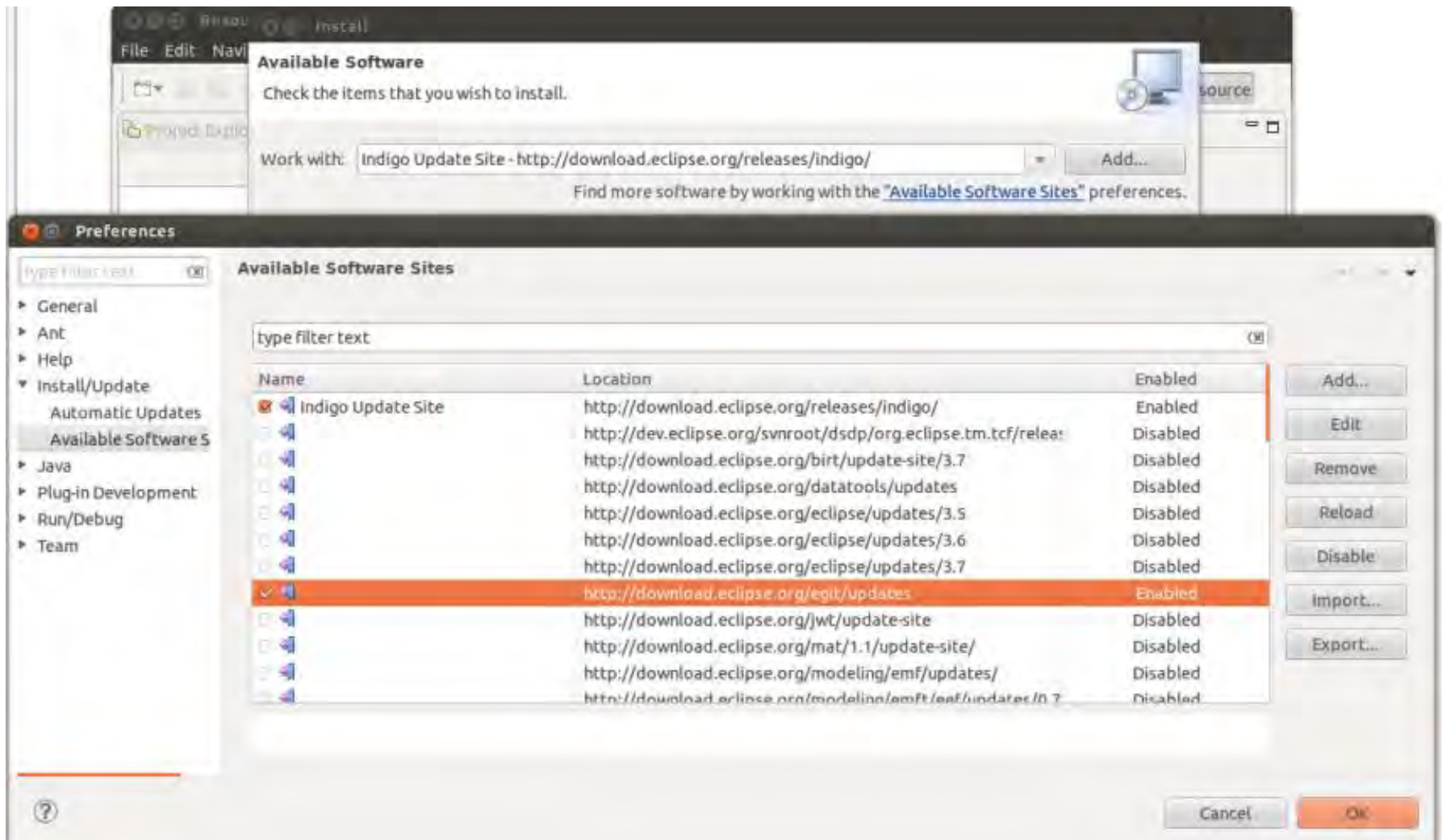
Step by step

Help>> Install New Software>> Add C++ Development Tools



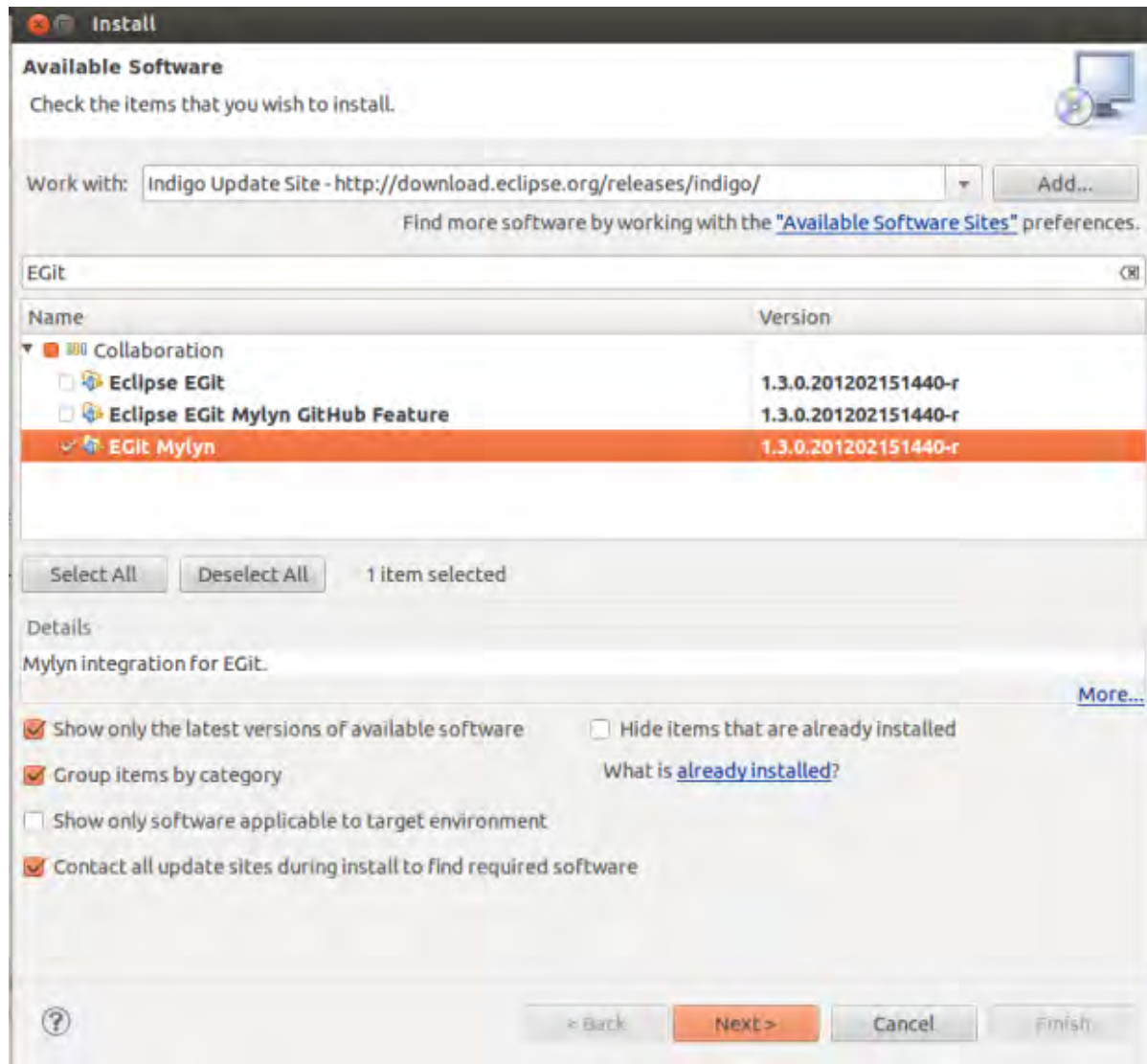
Step by step

Help>> Install New Software>> Add EGit



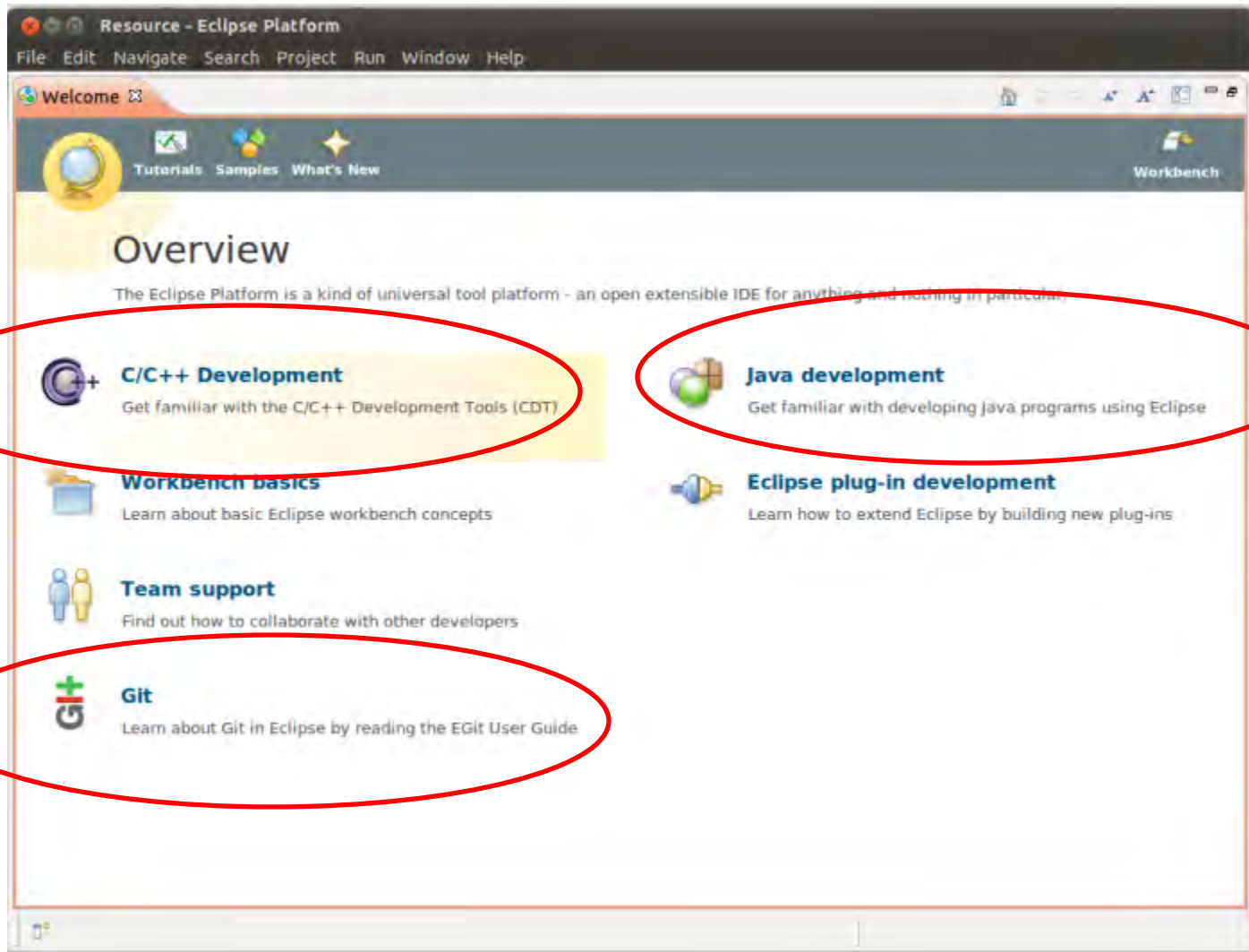
Step by step

Help>> Install New Software>> Add EGit



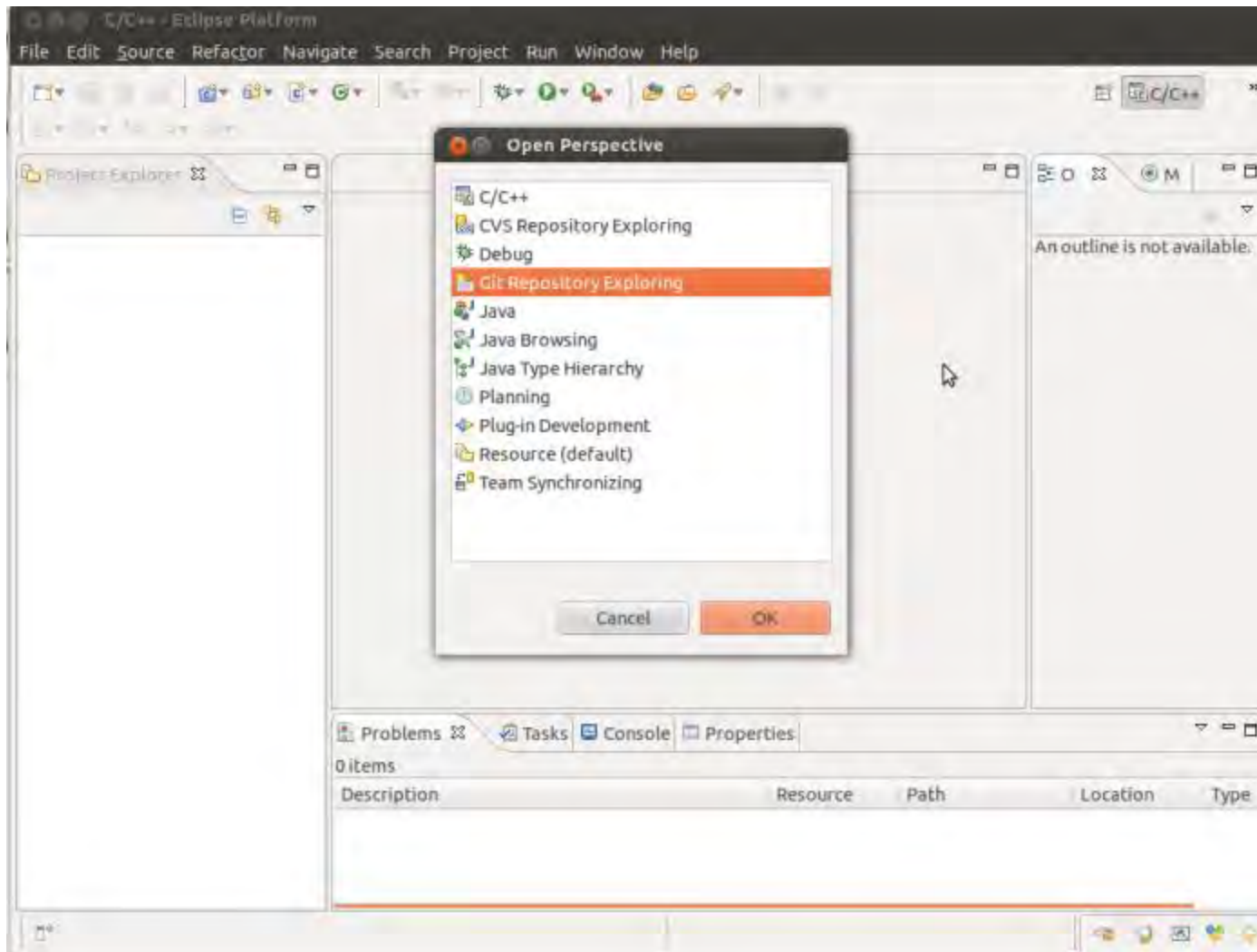
Step by step

You should see this after restart Eclipse



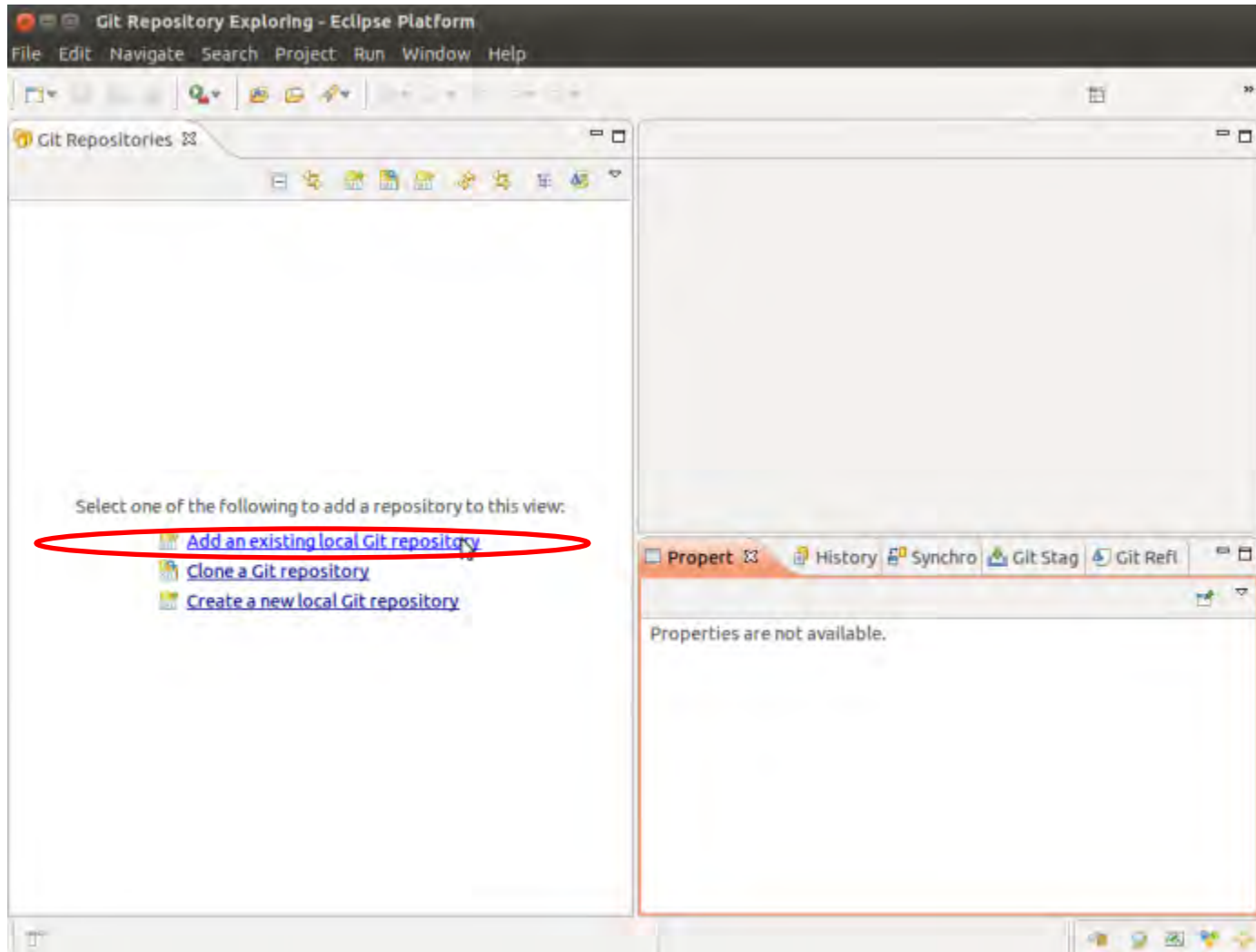
Step by step

8) Importing the Repositories into Eclipse

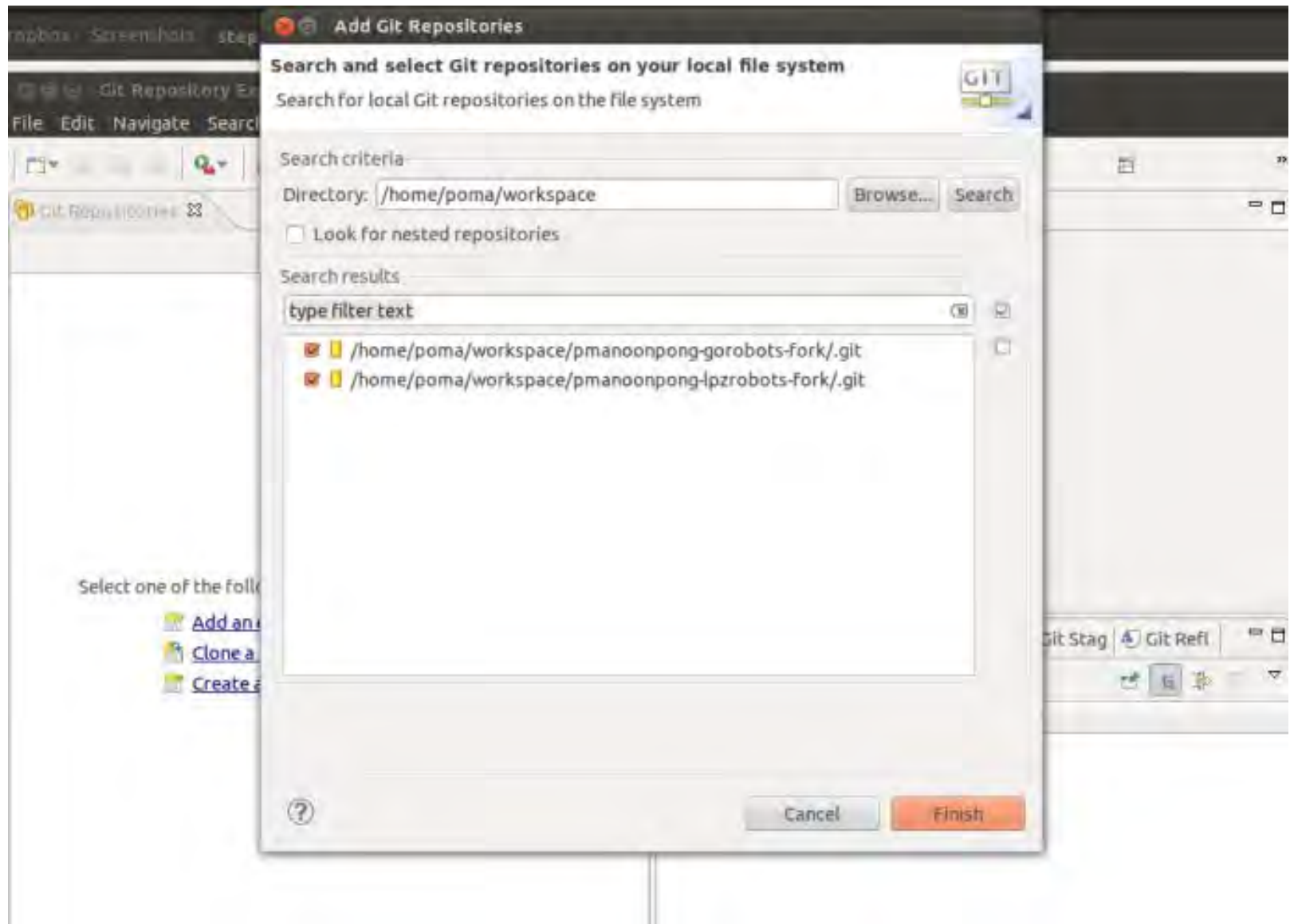


Step by step

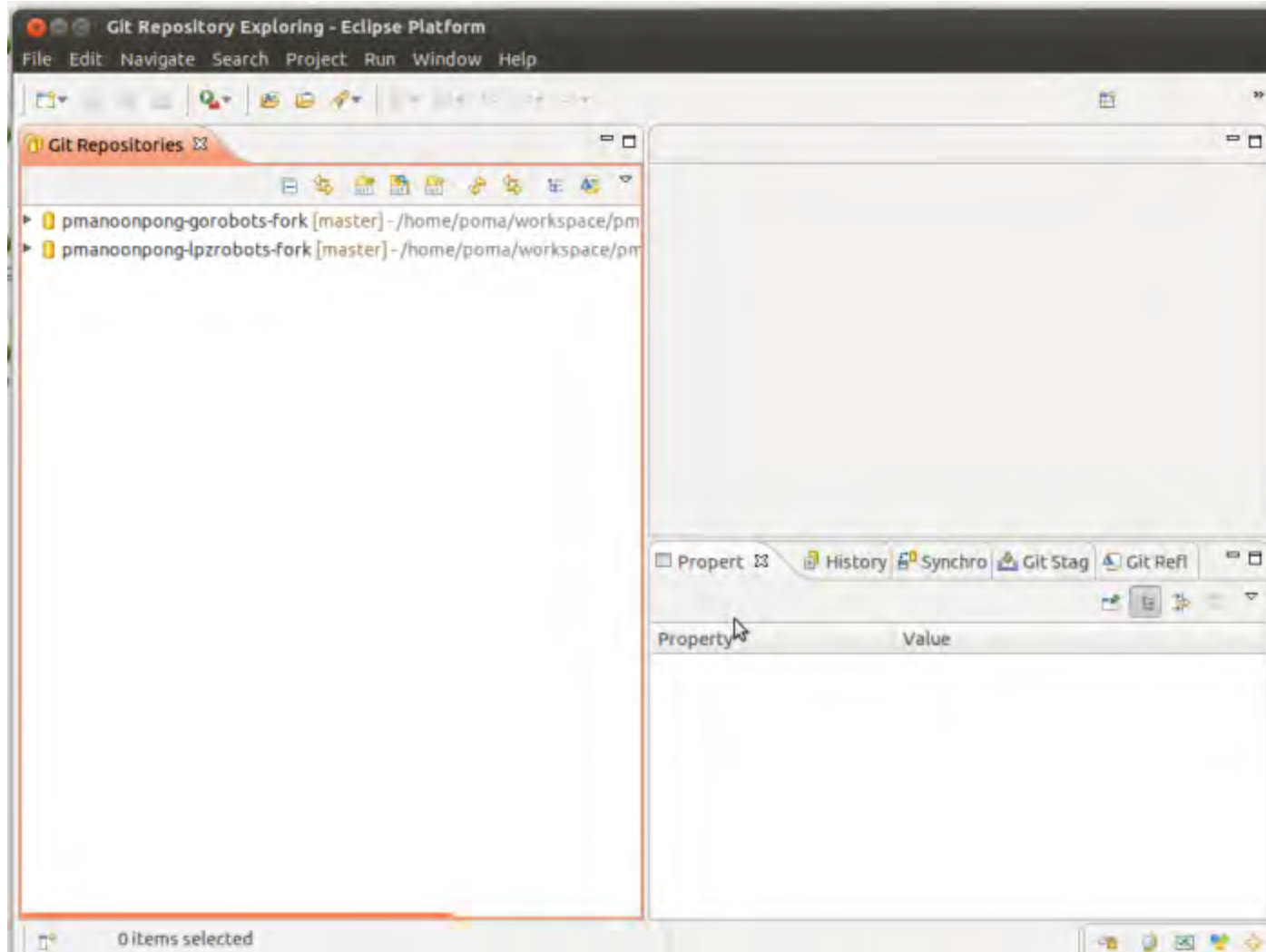
8) Importing the Repositories into Eclipse



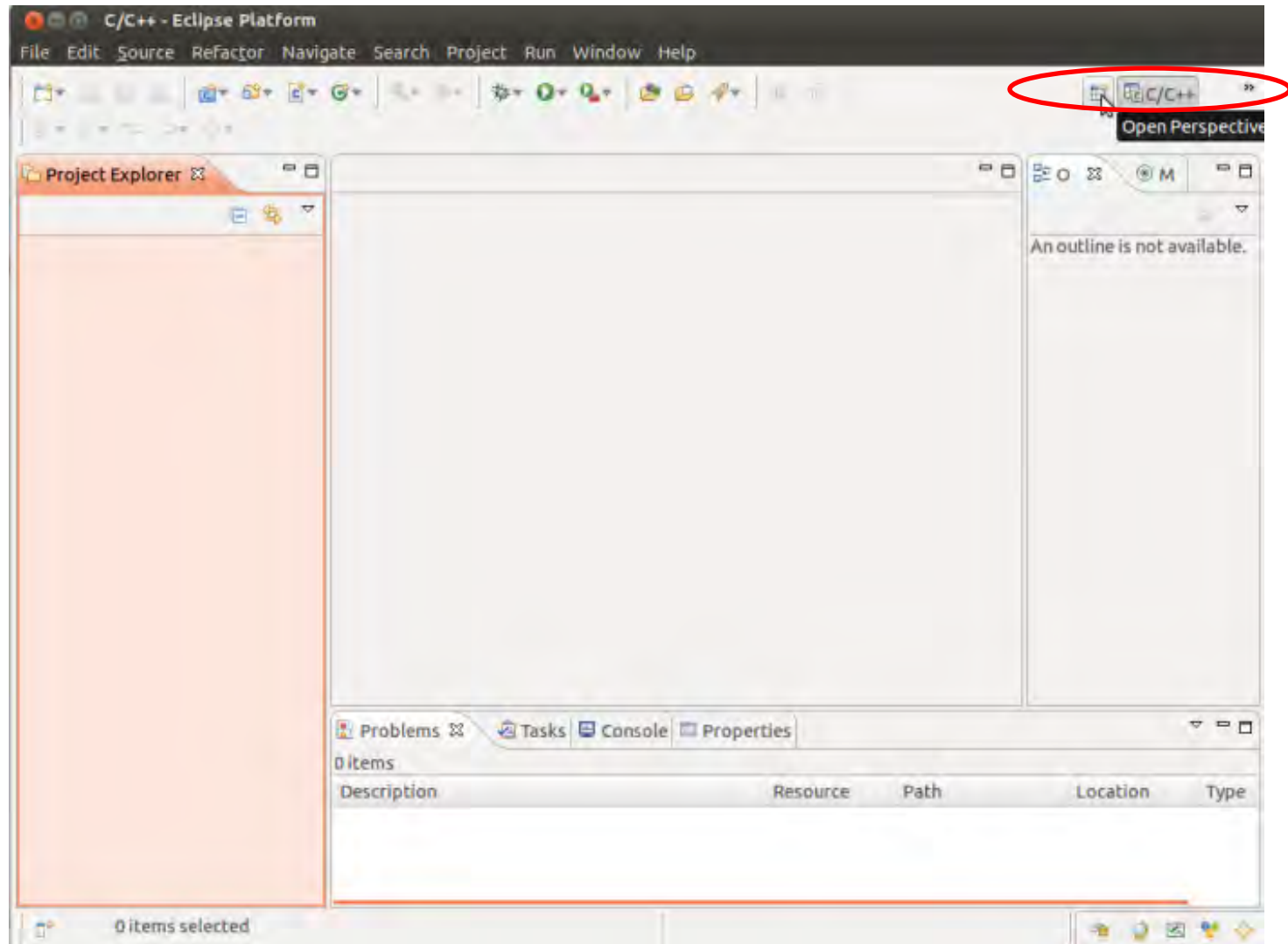
Step by step



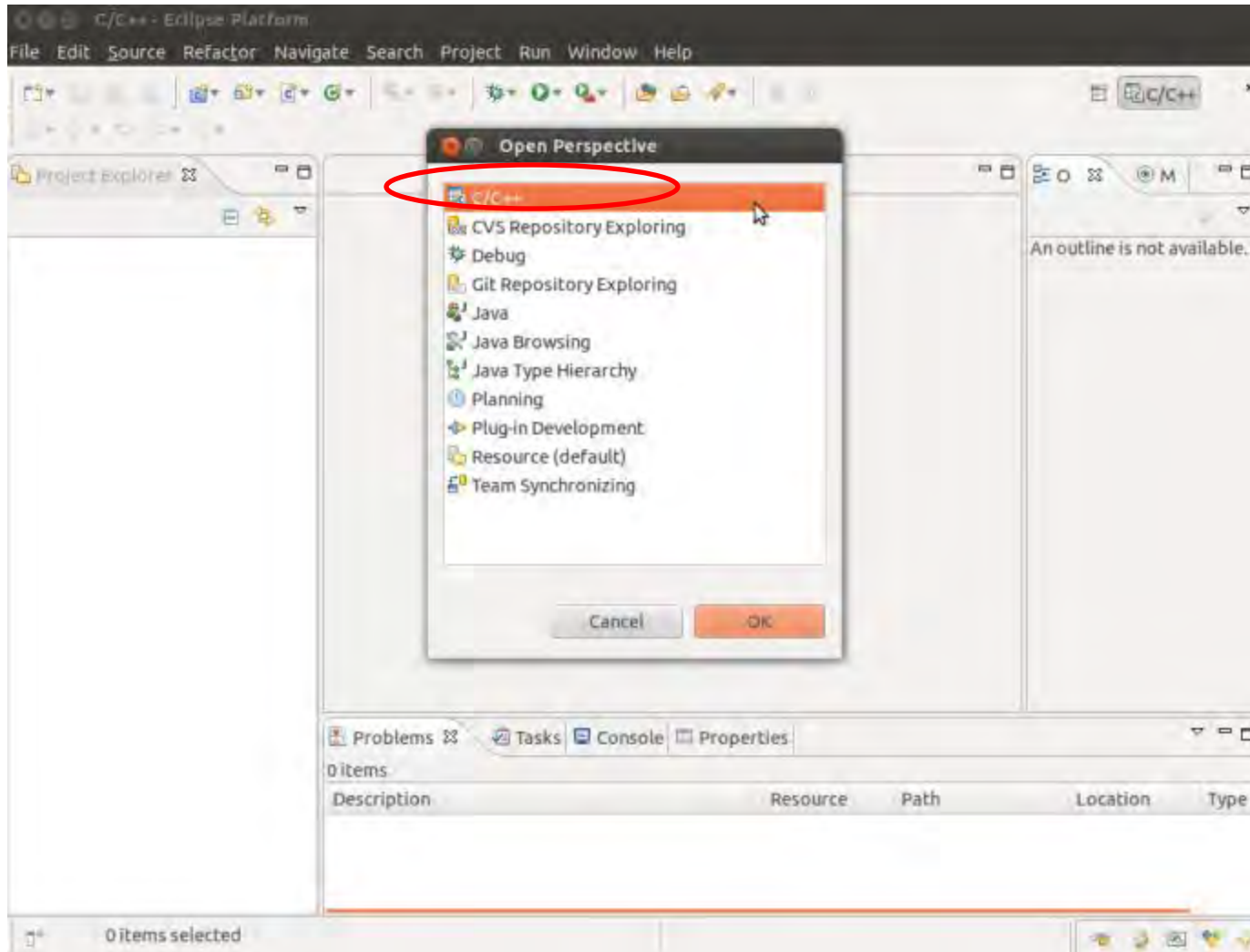
Step by step



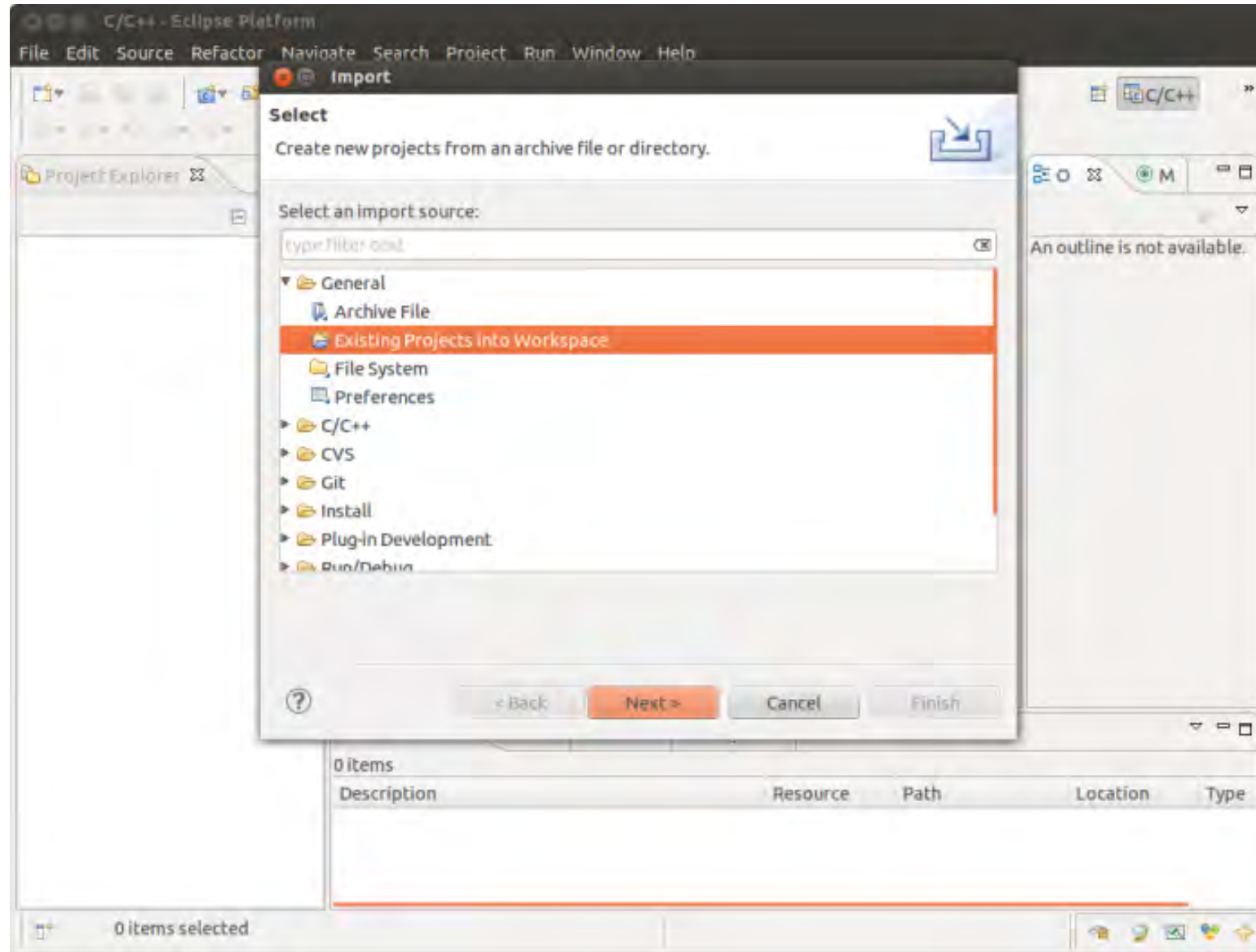
Step by step



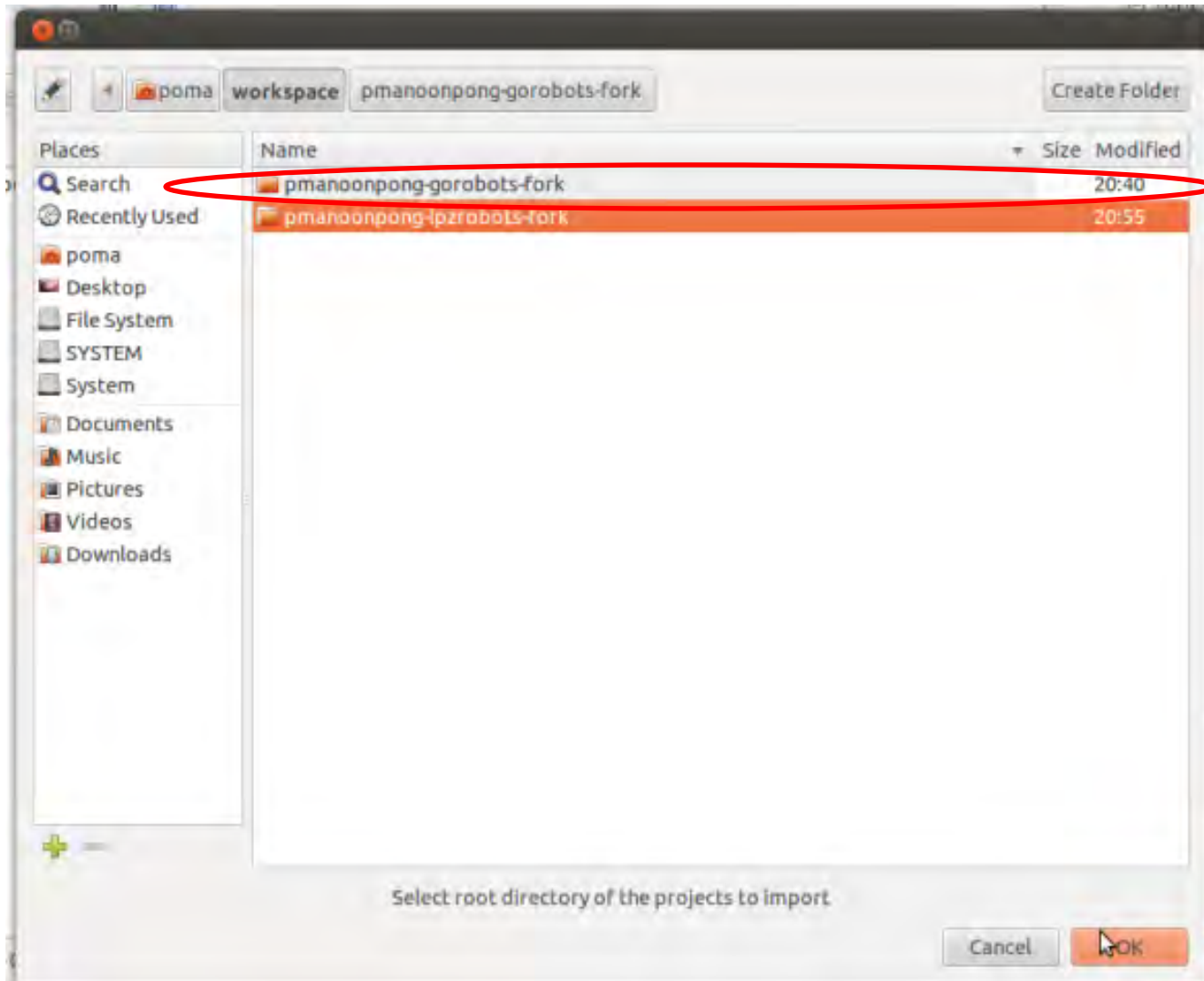
Step by step



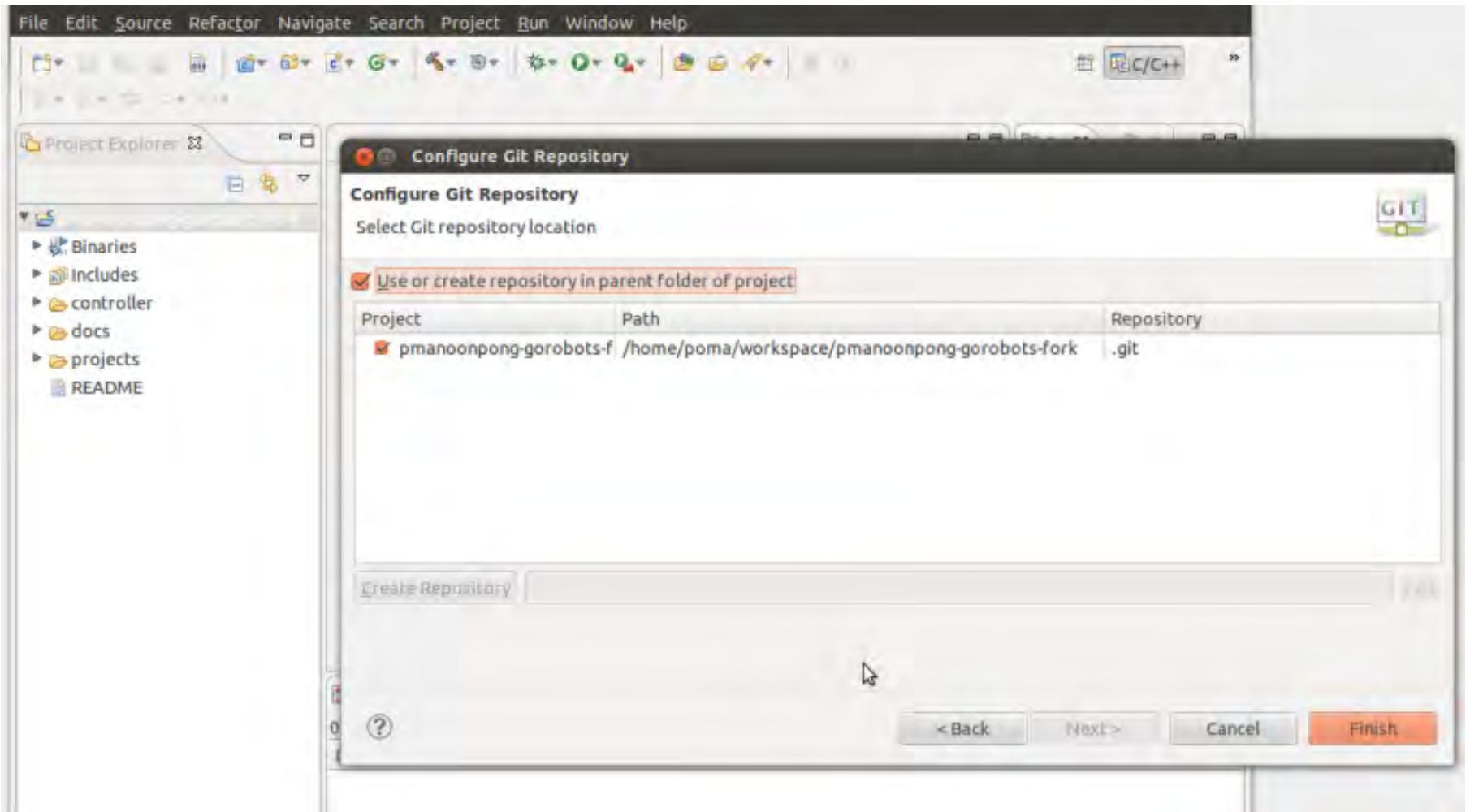
Step by step



Step by step



Step by step



Problems with GIT

Setting up Repositories in Eclipse

In some cases, the instructions on how to set up the GIT-repositories within Eclipse did not work.

Here is a different approach, that worked for me:

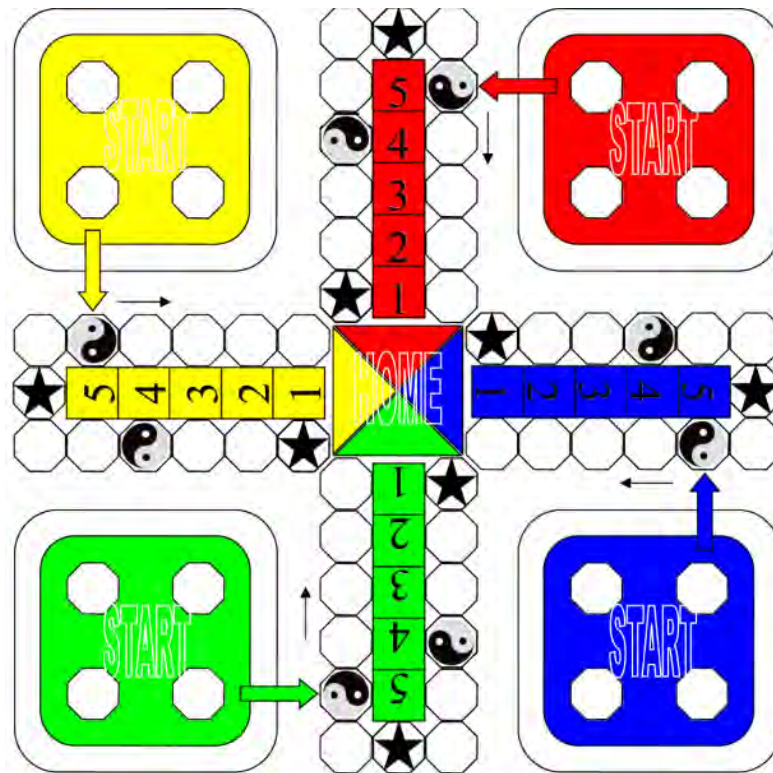
1. Import the repositories into the GIT-view of Eclipse, just as described before
2. Instead of importing over the GIT-View, you now go onto [File → Import → General → Existing Projects into Workspace](#) and you then choose the two repositories
3. After Eclipse has imported the files, you can [right-click](#) on the Project, and then select [Team → Share](#)
4. Now, just select [GIT](#) and the two GIT-repository-adresses should appear
5. [Apply](#)

Add other repositories!

- `git remote add name https://your
user@git.assembla.com/repository name`

```
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote
add stable https://p.manoonpong@git.assembla.com/gorobots.git
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote
add gorobots_edu https://p.manoonpong@git.assembla.com/gorobots_edu.git
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ cd ..
poma@poma-HP-EliteBook-840-G1:~/workspace$ cd pmanoonpong-gorobots-fork/
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote
add xiaofeng https://p.manoonpong@git.assembla.com/lenonxiong-gorobots.git
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote
add sakya https://p.manoonpong@git.assembla.com/sakyad-gorobots-fork.git
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote
add robotdemo https://p.manoonpong@git.assembla.com/RobotDemo-gorobots-fork.git
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote
add ren https://p.manoonpong@git.assembla.com/sqrgj-gorobots-fork.git
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote
add eduard https://p.manoonpong@git.assembla.com/eduardg-gorobots-fork.git
poma@poma-HP-EliteBook-840-G1:~/workspace/pmanoonpong-gorobots-fork$ git remote
add dennis https://p.manoonpong@git.assembla.com/degoltschmidt-gorobots-fork.git
```

Java project: LUDO game



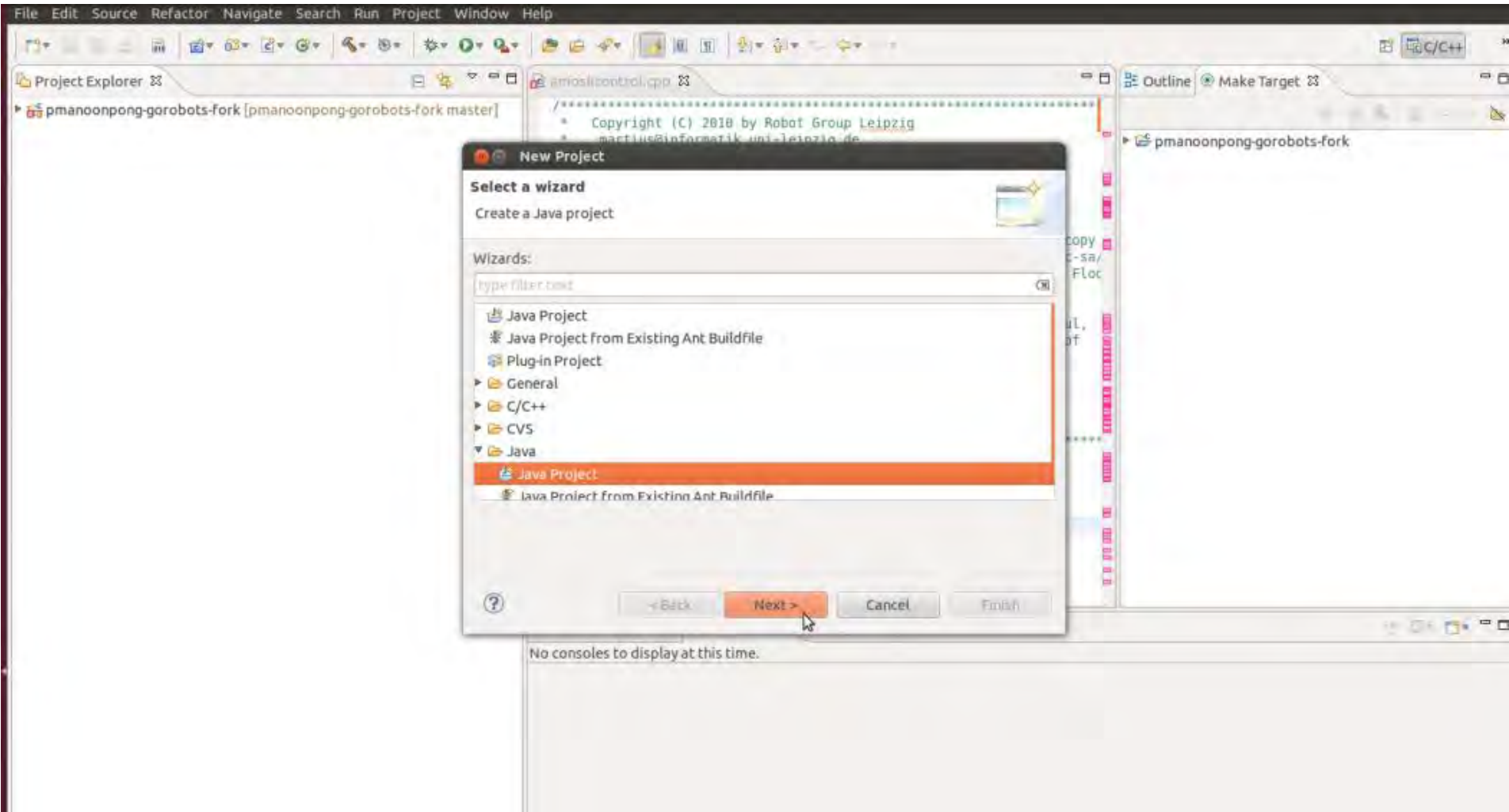
Step by step

9) Importing the LUDO project into Eclipse

→ Extract workspacejava in home directory:
:~/workspacejava/LUDO

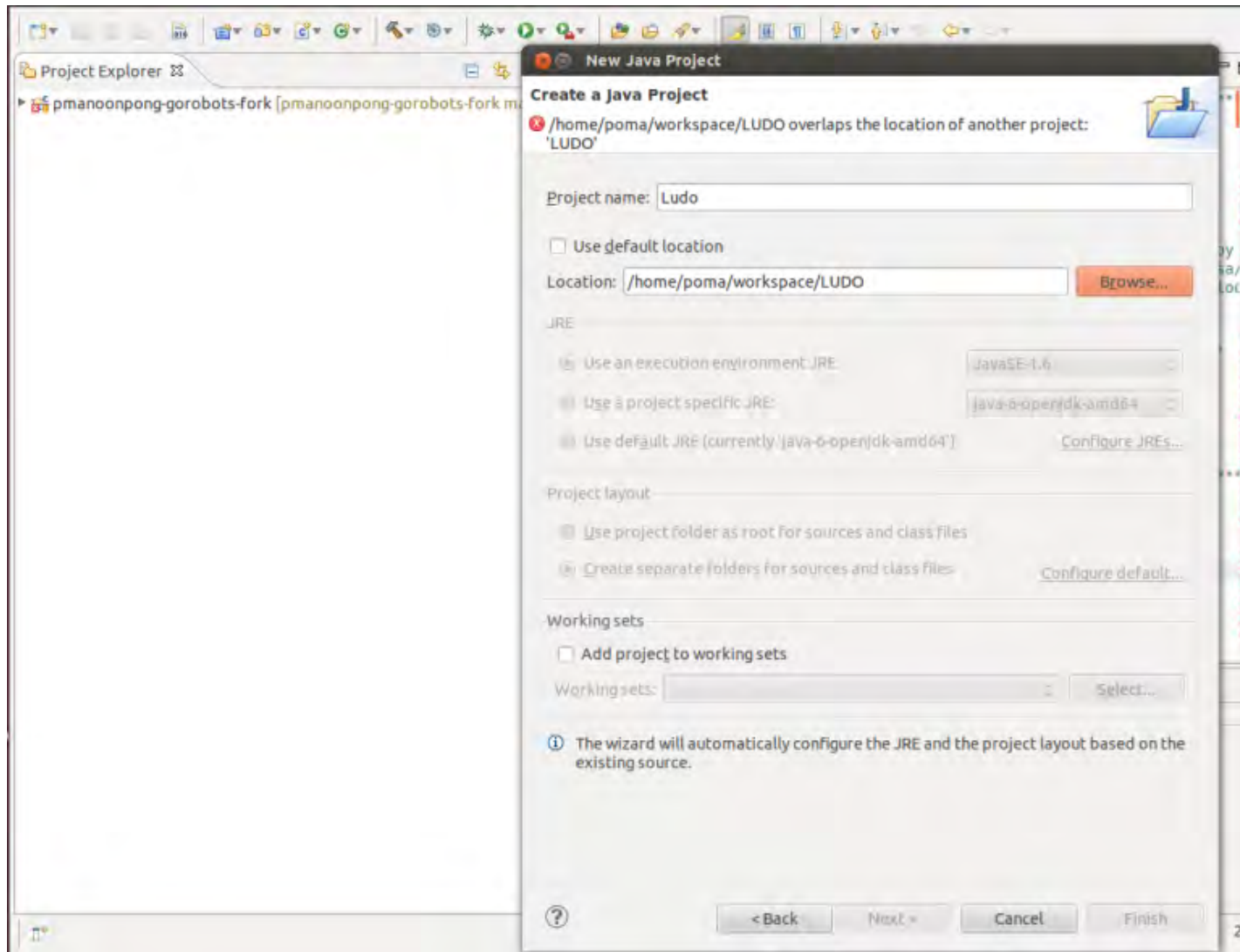
Step by step

File>>New Project>> Java project

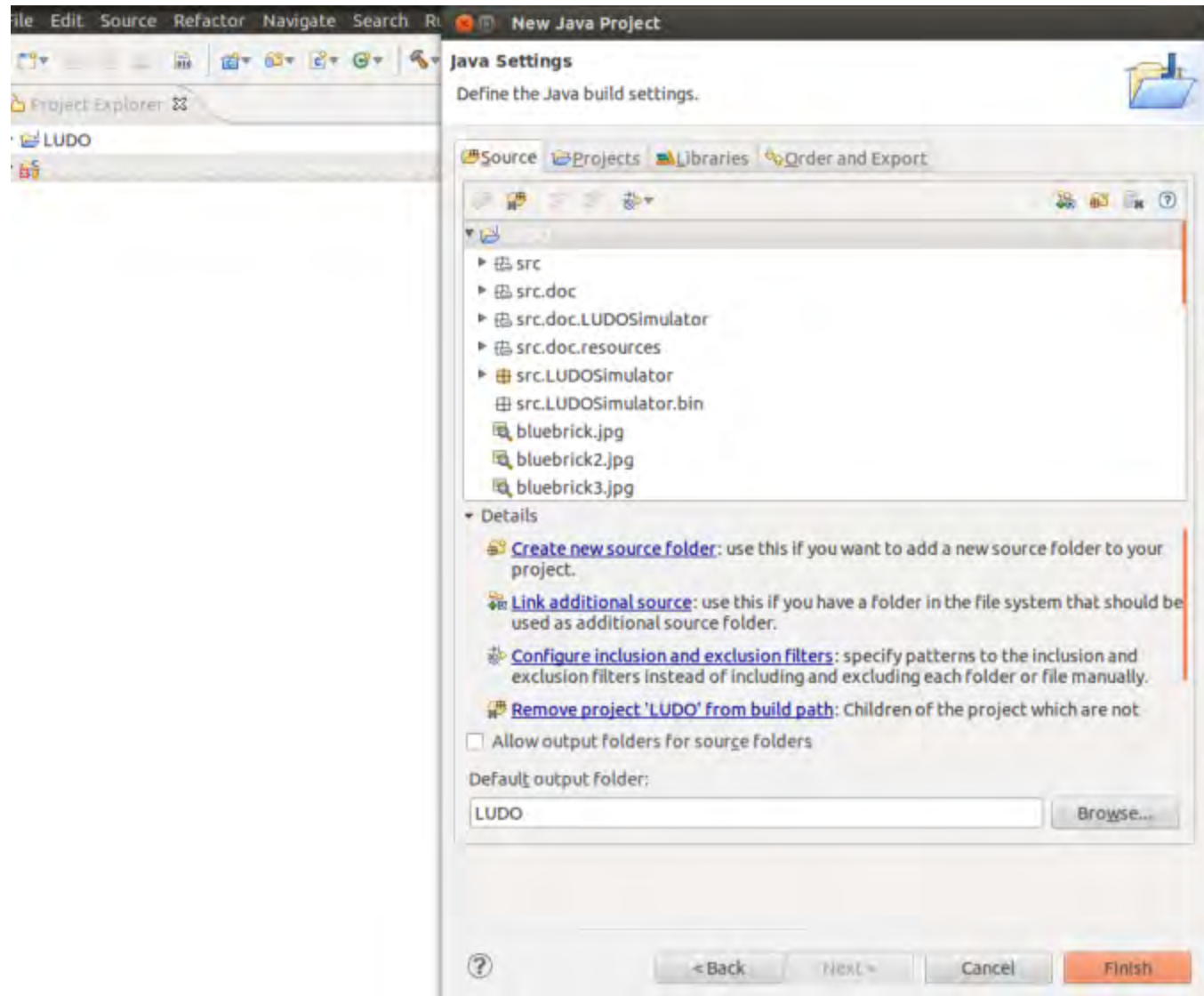


Step by step

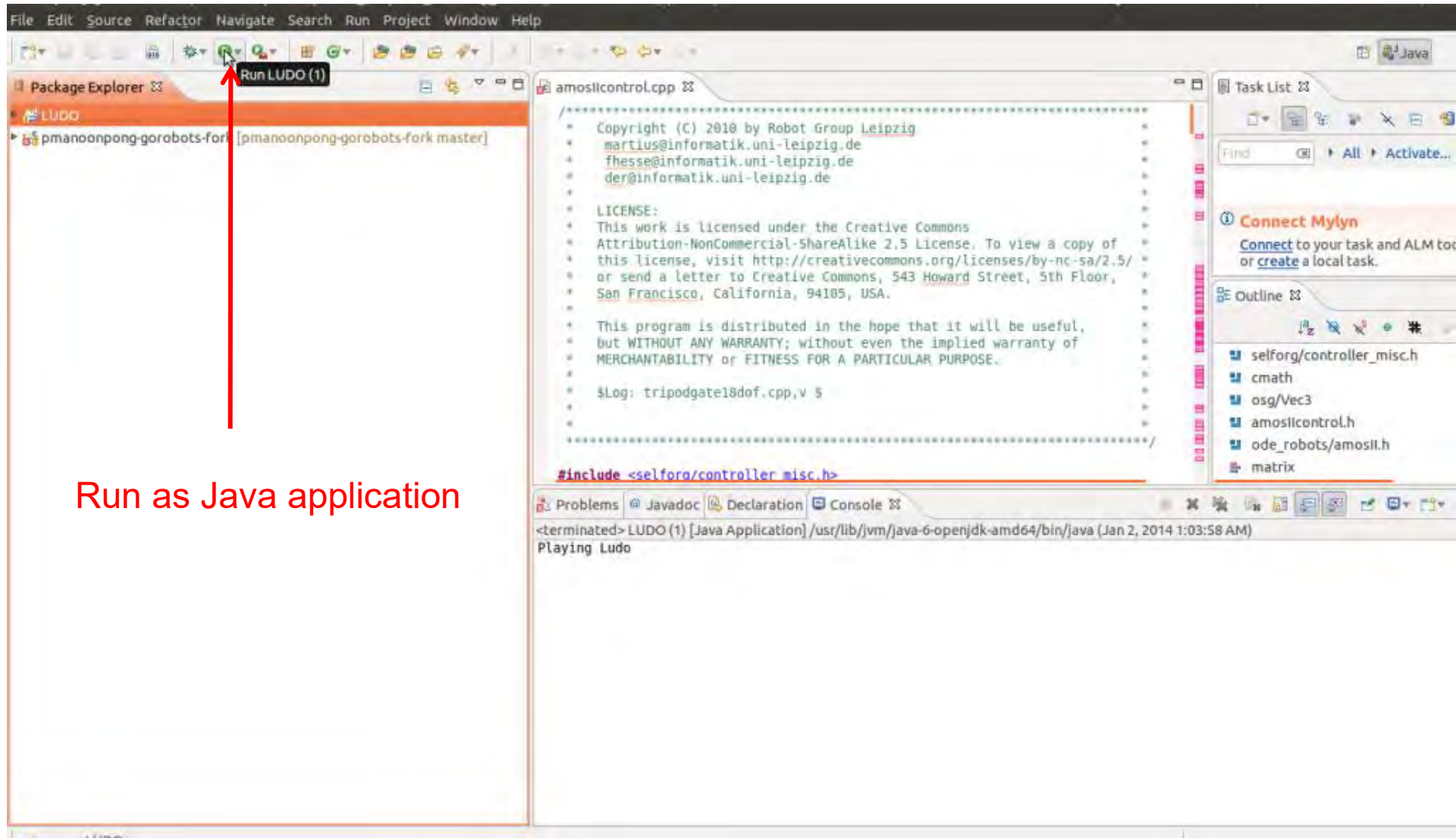
Browse to /home/your login/workspace/LUDO



Step by step

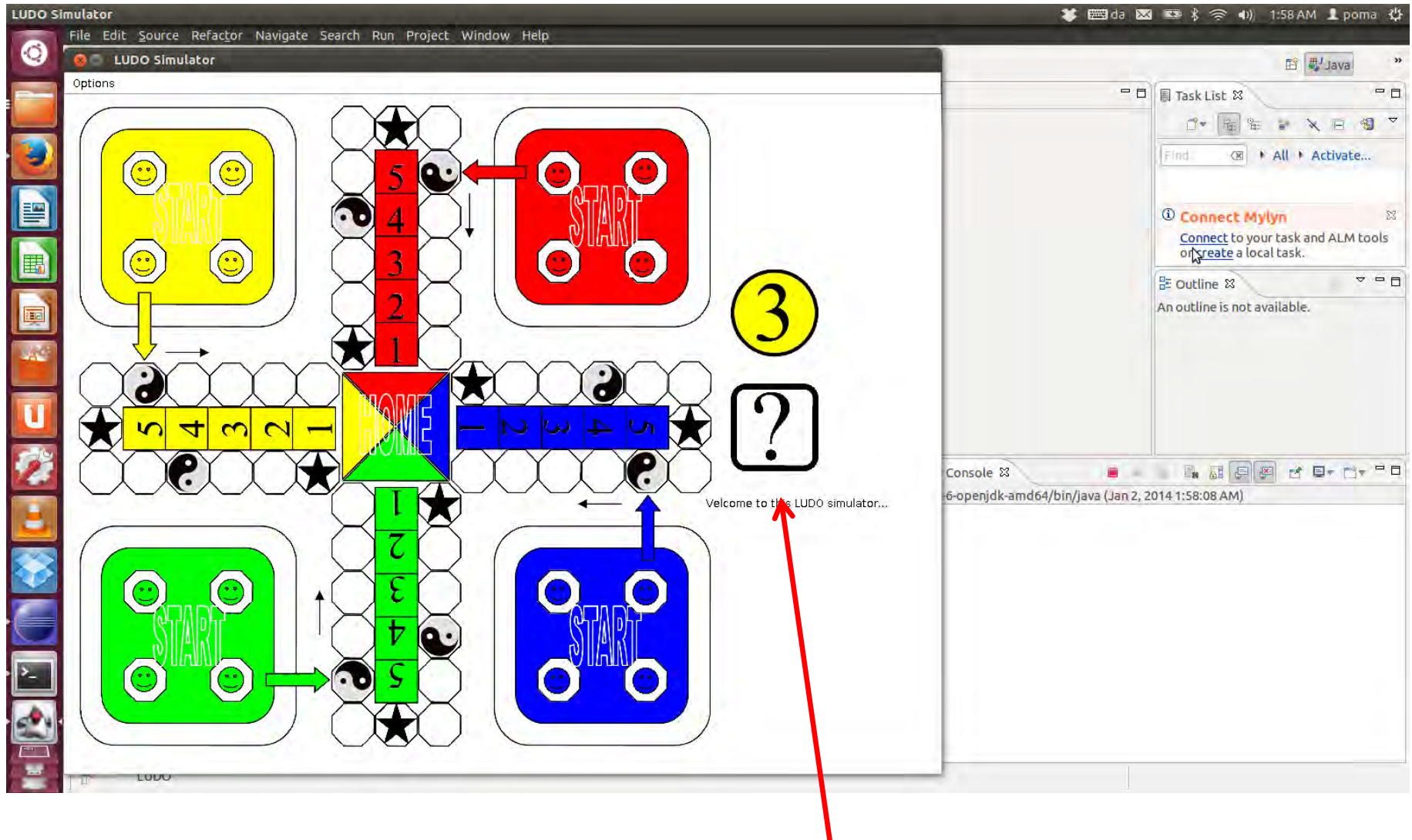


Step by step



Run as Java application

Step by step



Click die to start the game

For a 64 bit PC

You might need to update gcc

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test
```

```
sudo apt-get update sudo apt-get install gcc-4.8
```

```
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.8 50
```

Upgrade your Ubuntu system to avoid system freeze

```
apt-get dist-upgrade
```

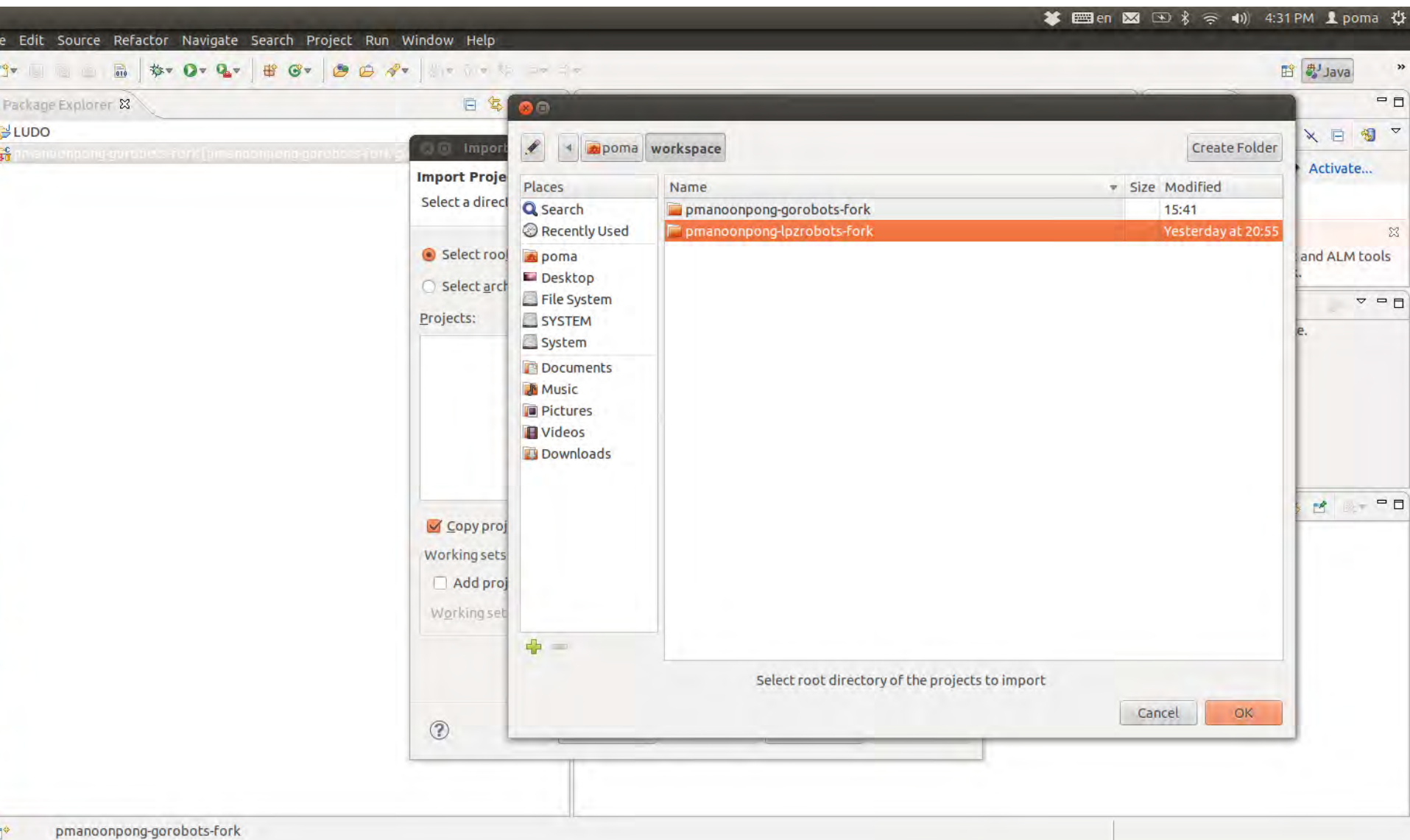
Such a problem might occur when compiling a program

```
./start: symbol lookup error: /usr/lib/libgsl.so.0: undefined symbol:  
cblas_dnrm2
```

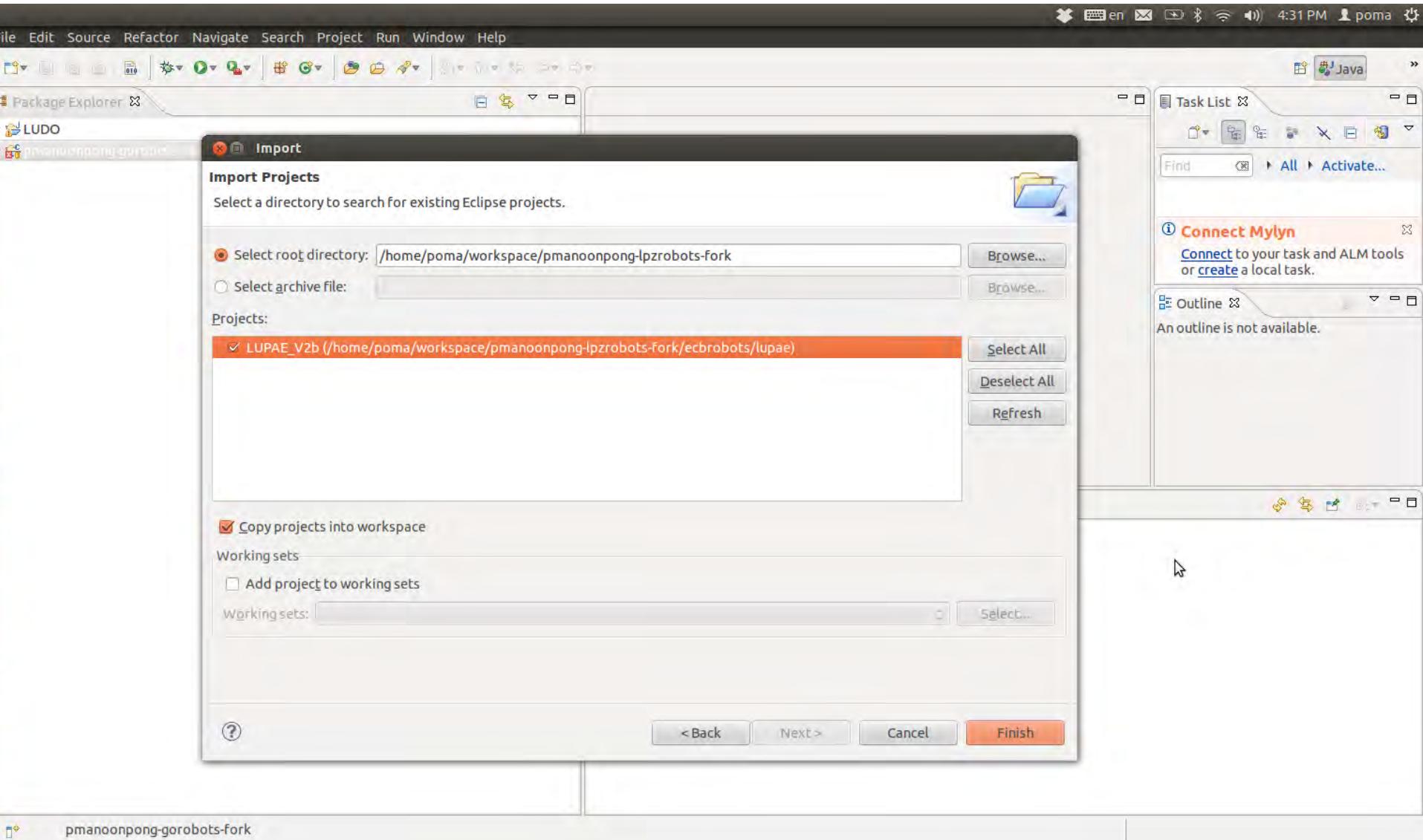
solve by adding this in Makefile.conf

```
“ LIBS = -Wl,--no-as-needed -lgsl -lgslcblas -lm “
```

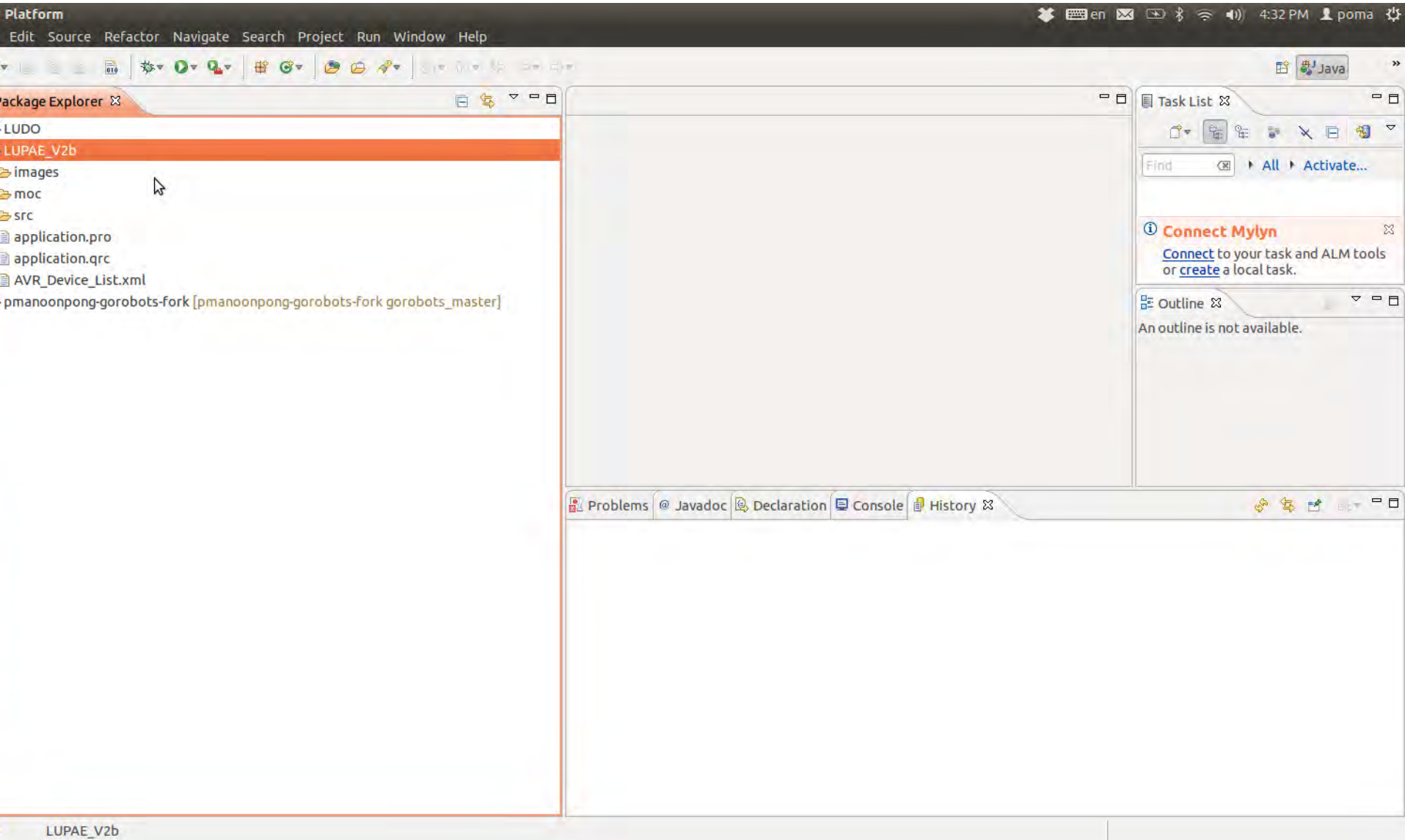
Problem



Problem

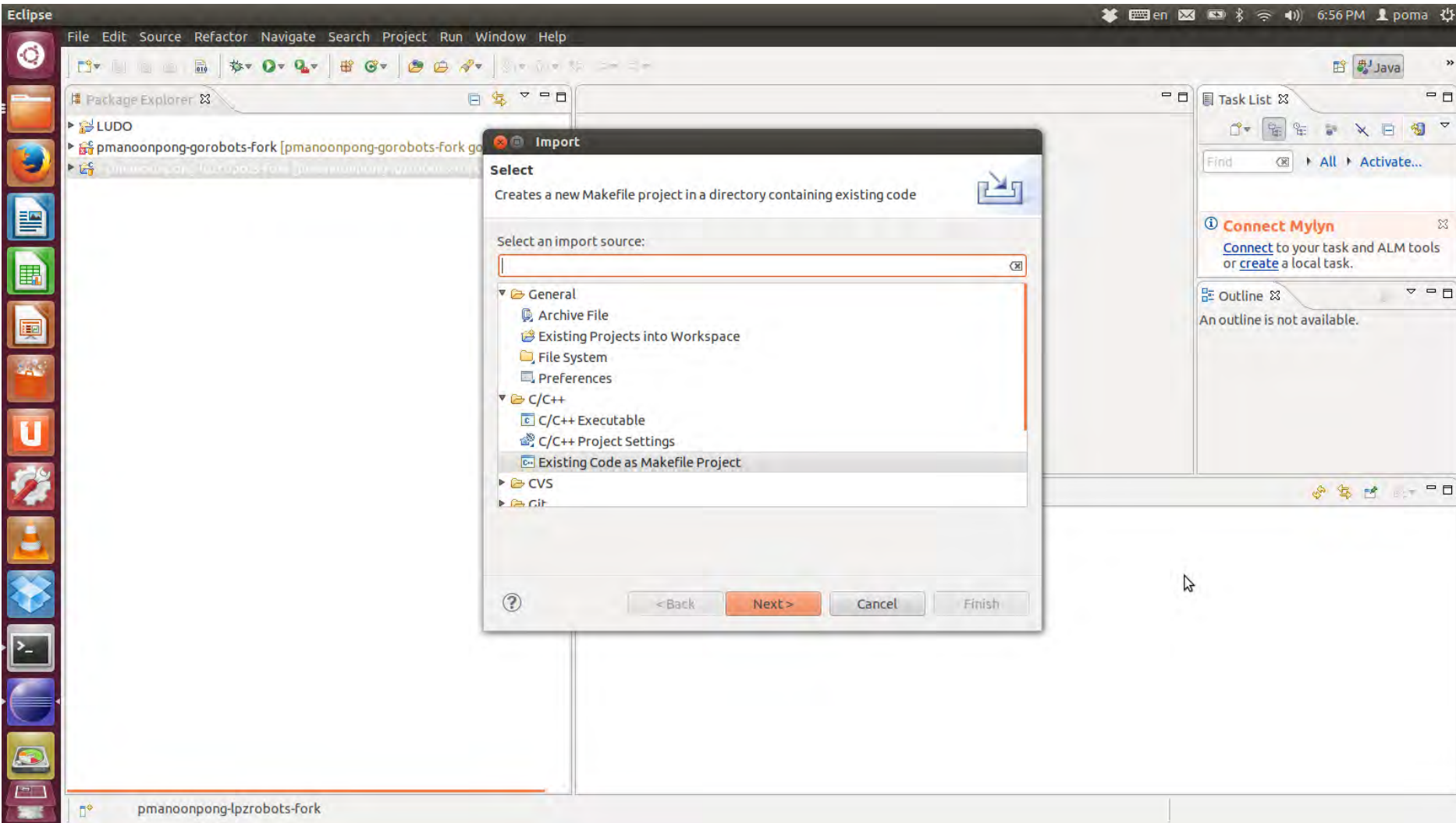


Problem



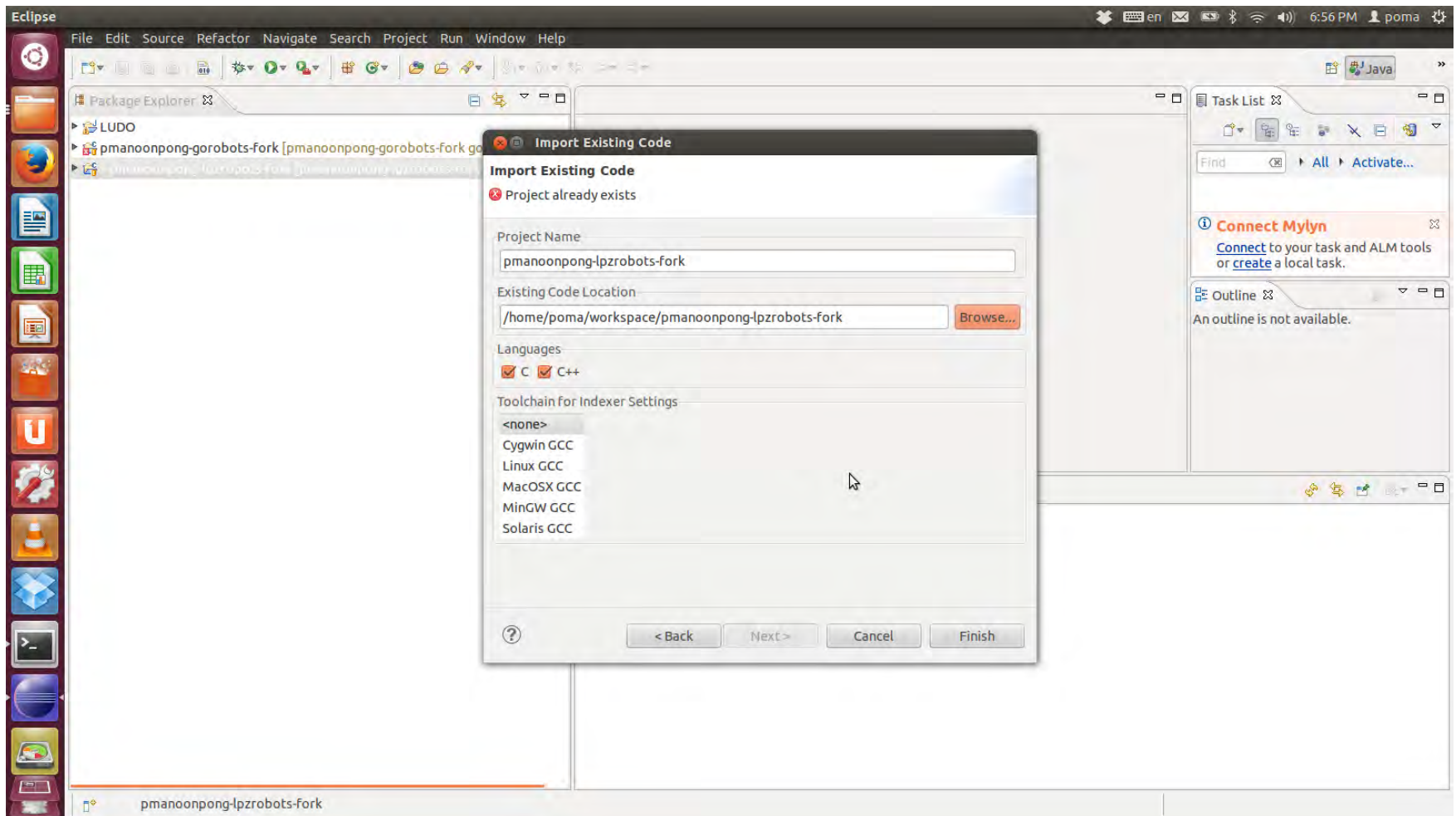
Solution

- File->Import->C/C++->Existing Code as Makefile Project.

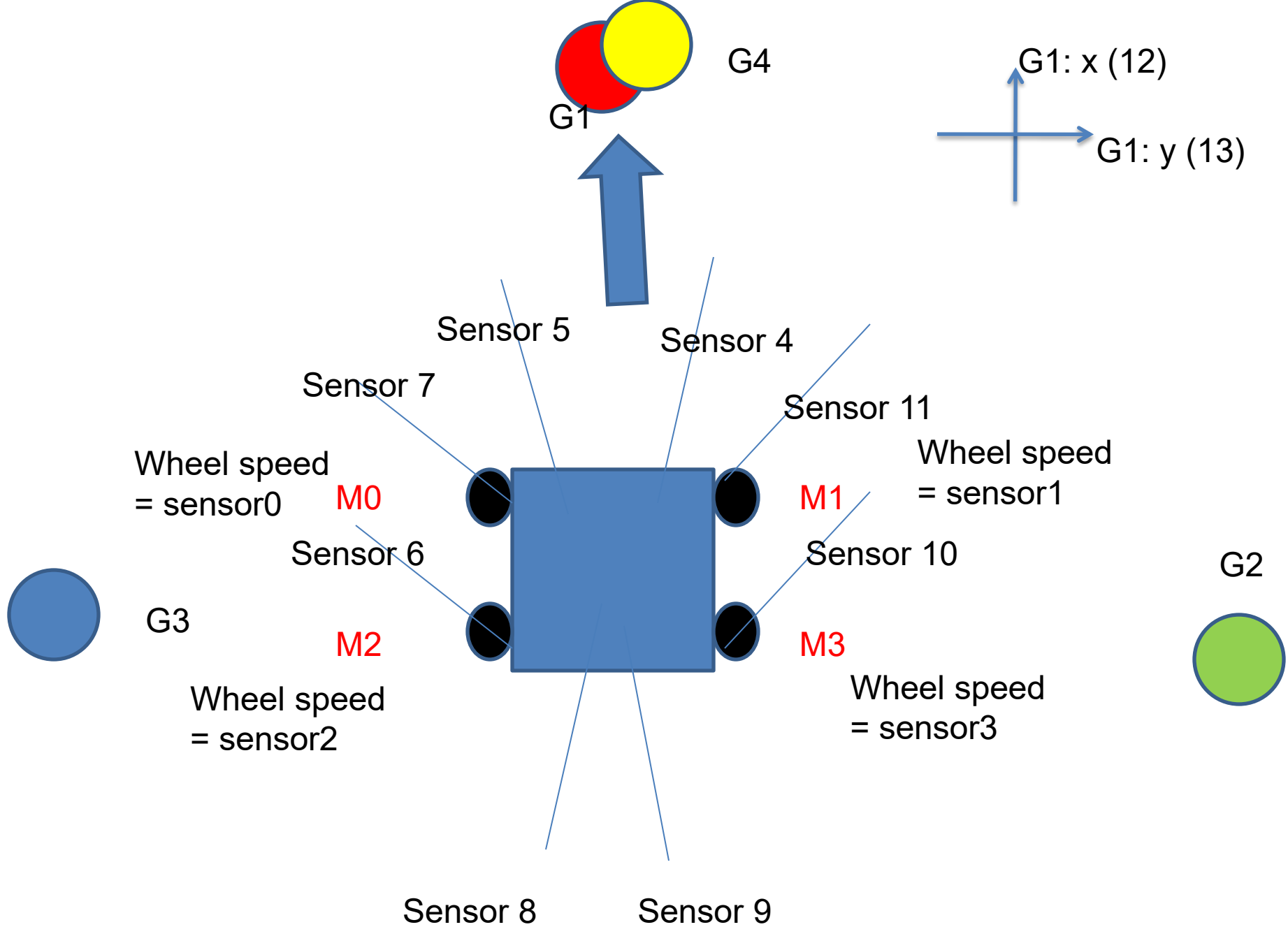


Solution

- File->Import->C/C++->Existing Code as Makefile Project.



Nimm4 system



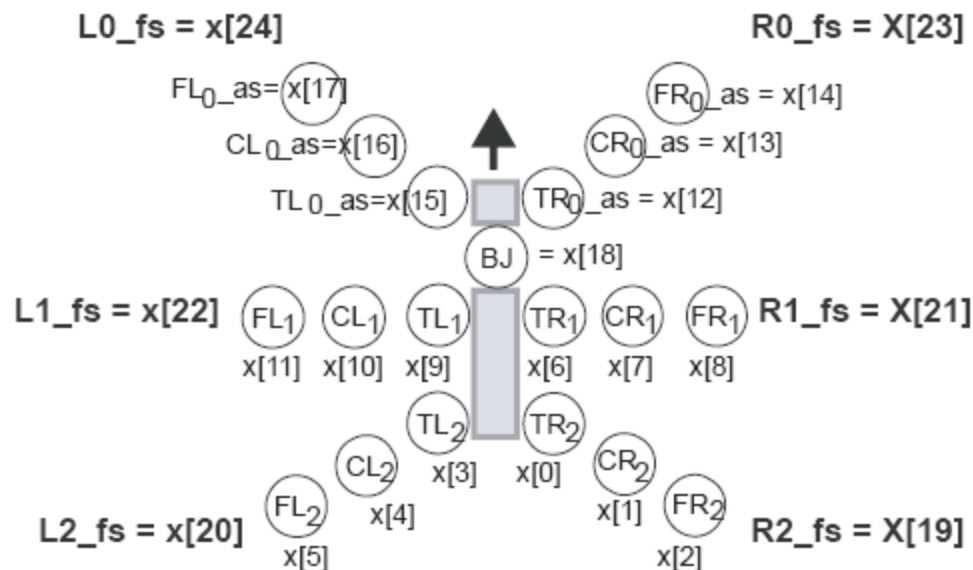
Hexapod system

AMOS simulation

Sensors:

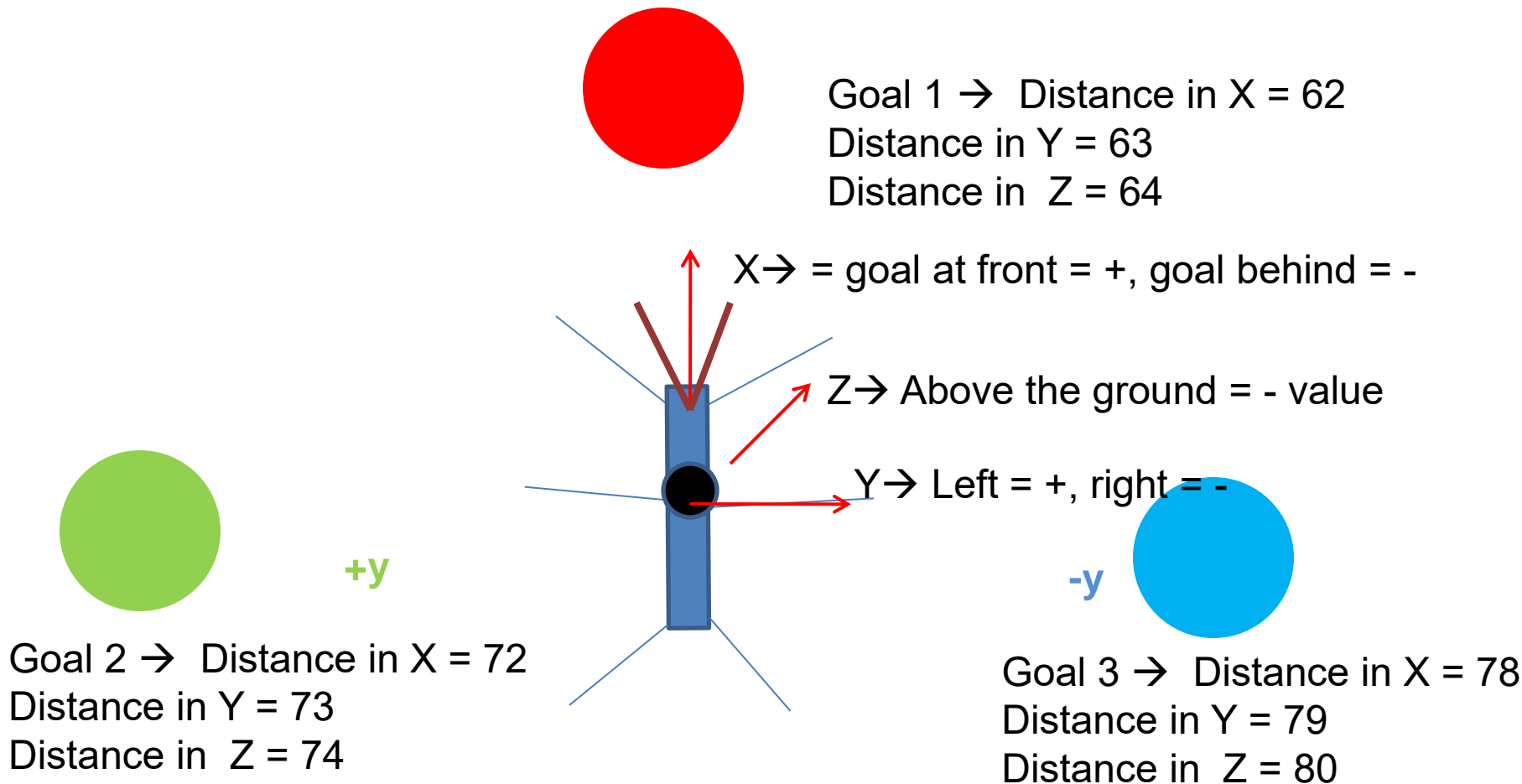
`_as` = angle sensors = $[-1, \dots, 1]$; -1 = backward, down, flex; +1 = forward, up, extend

`_fs` = foot sensors = $[0, \dots, 1]$; 0 = swing (off ground); +1 = stance (touch ground)

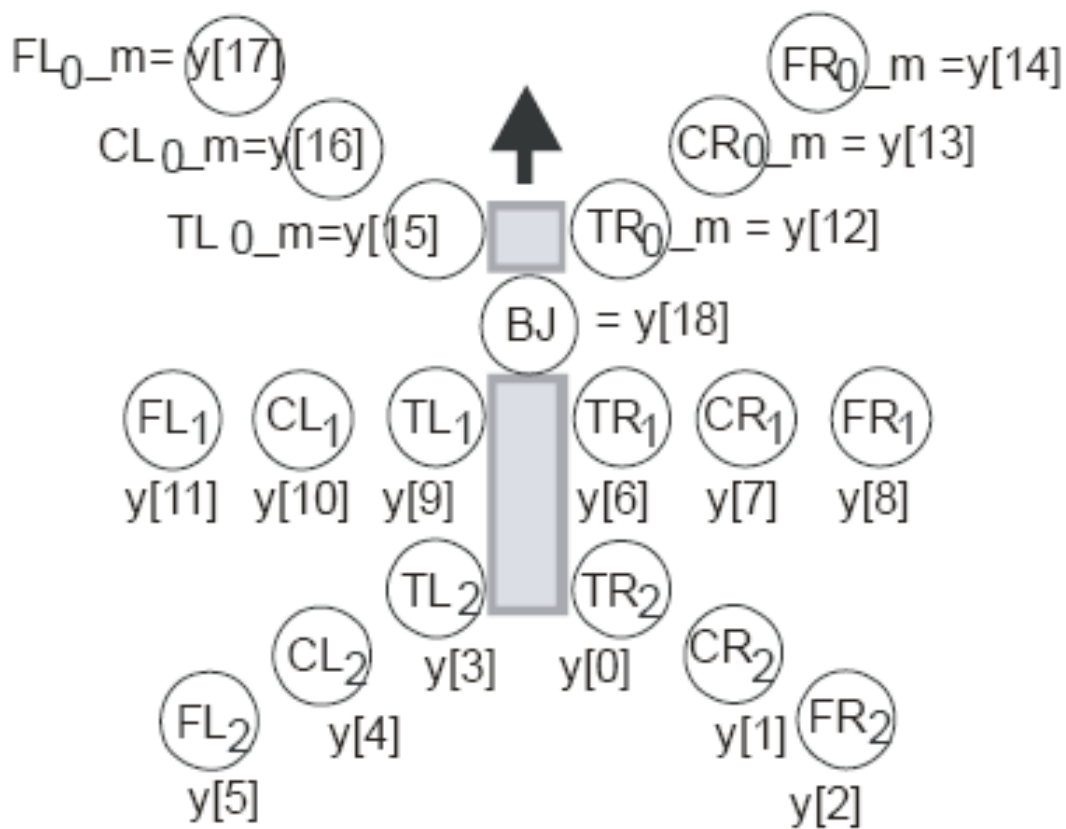


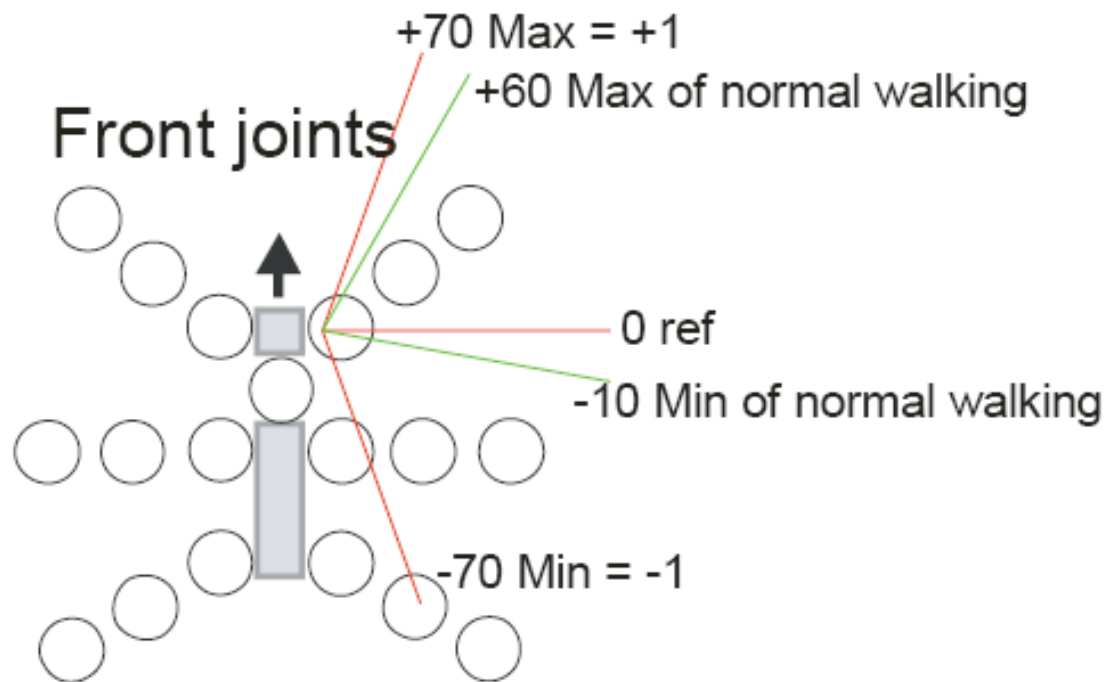
Sensors

- Goal detection sensors

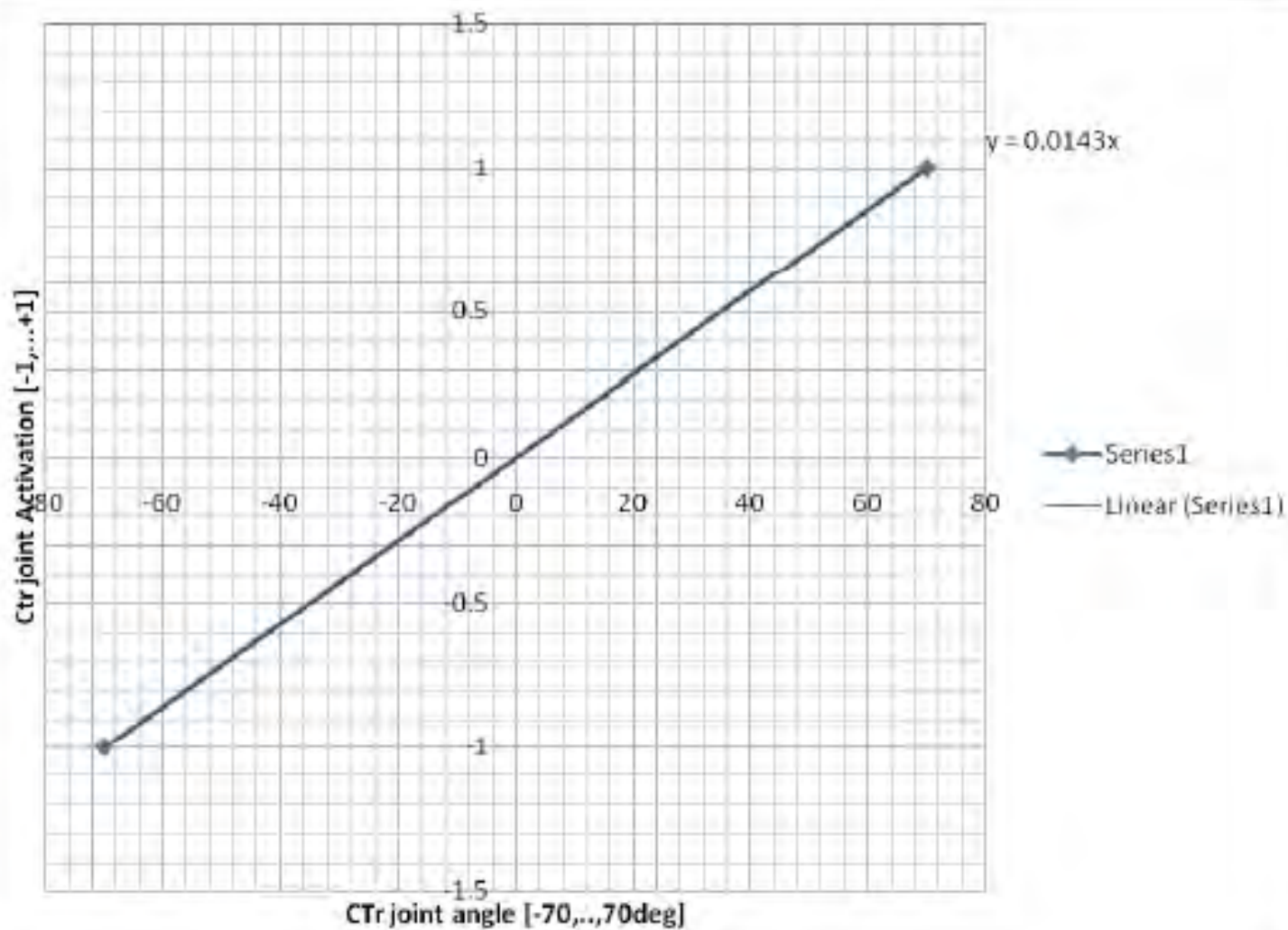


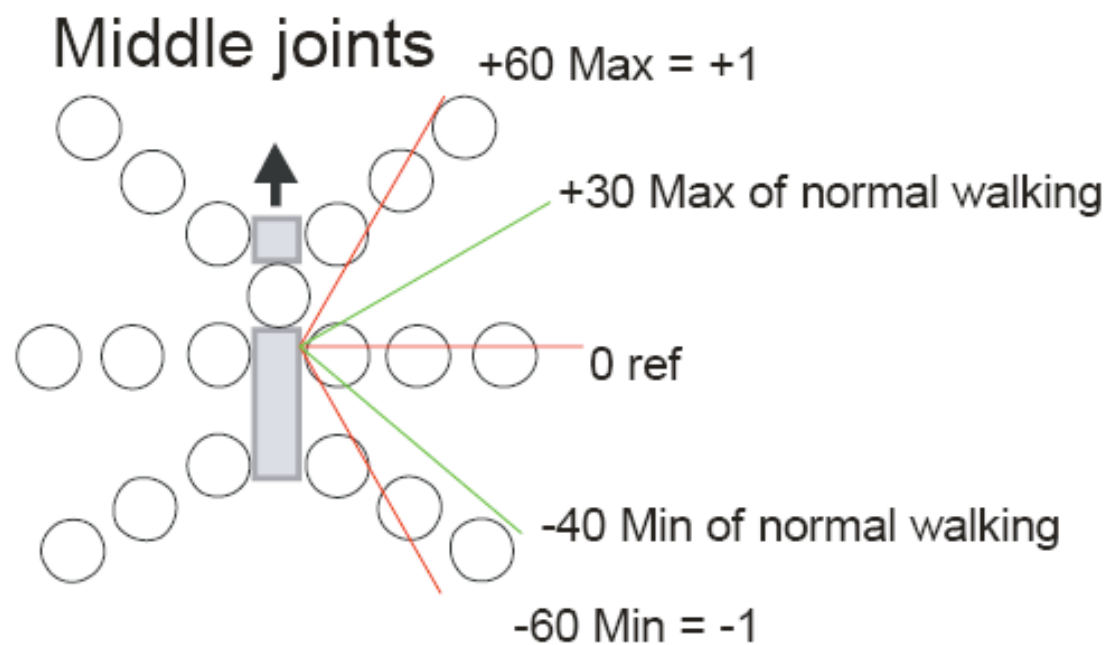
Motors:



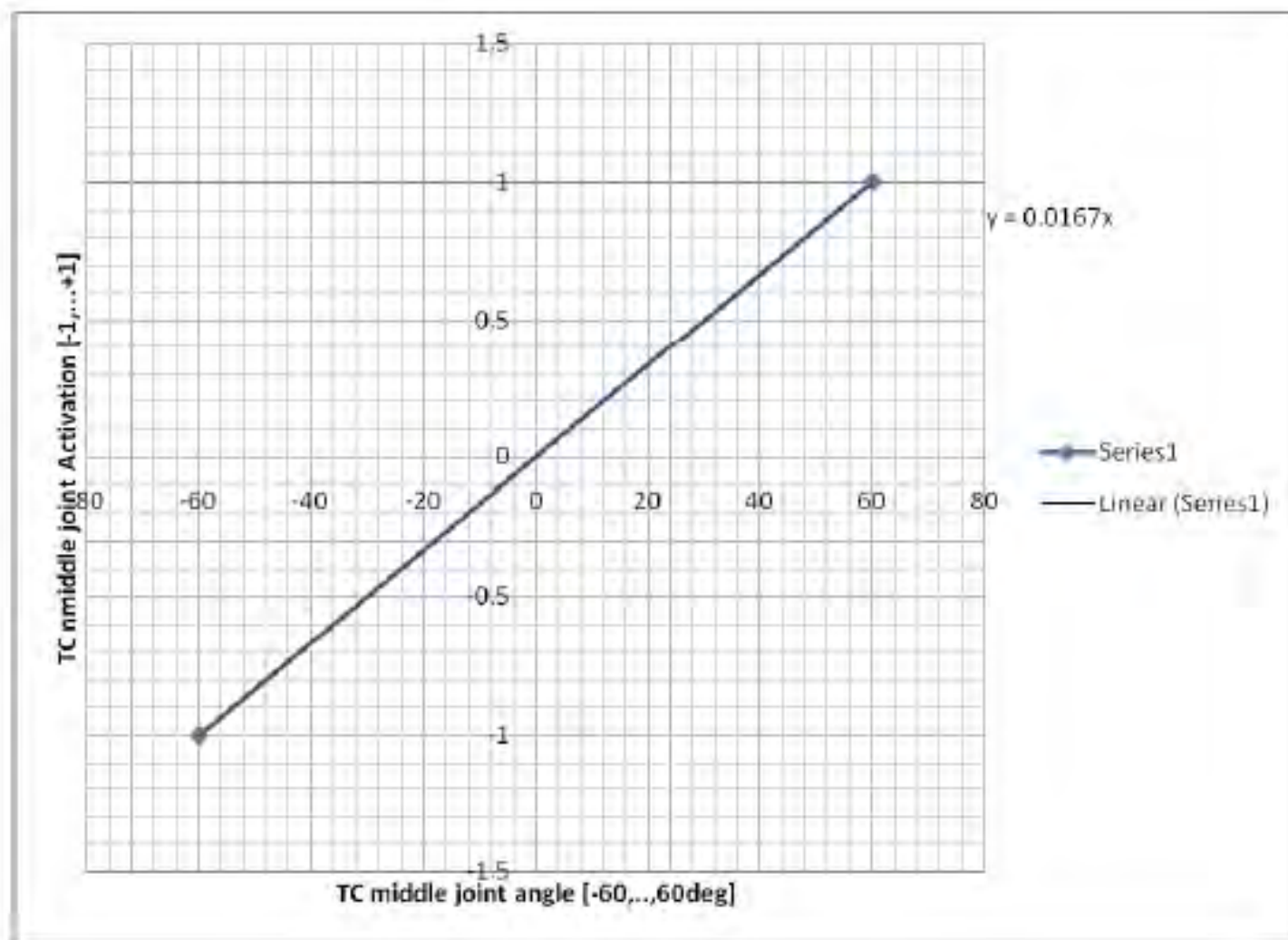


Deg	Activation
-70	-1.001
-60	-0.858
-50	-0.715
-40	-0.572
-30	-0.429
-20	-0.286
-10	-0.143
0	0
10	0.143
20	0.286
30	0.429
40	0.572
50	0.715
60	0.858
70	1.001

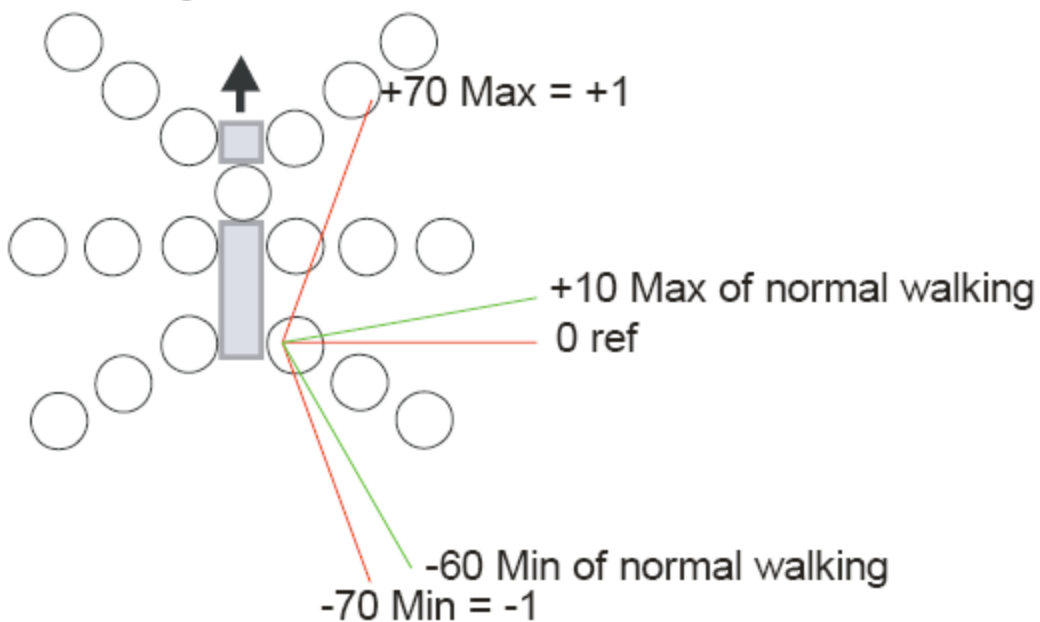




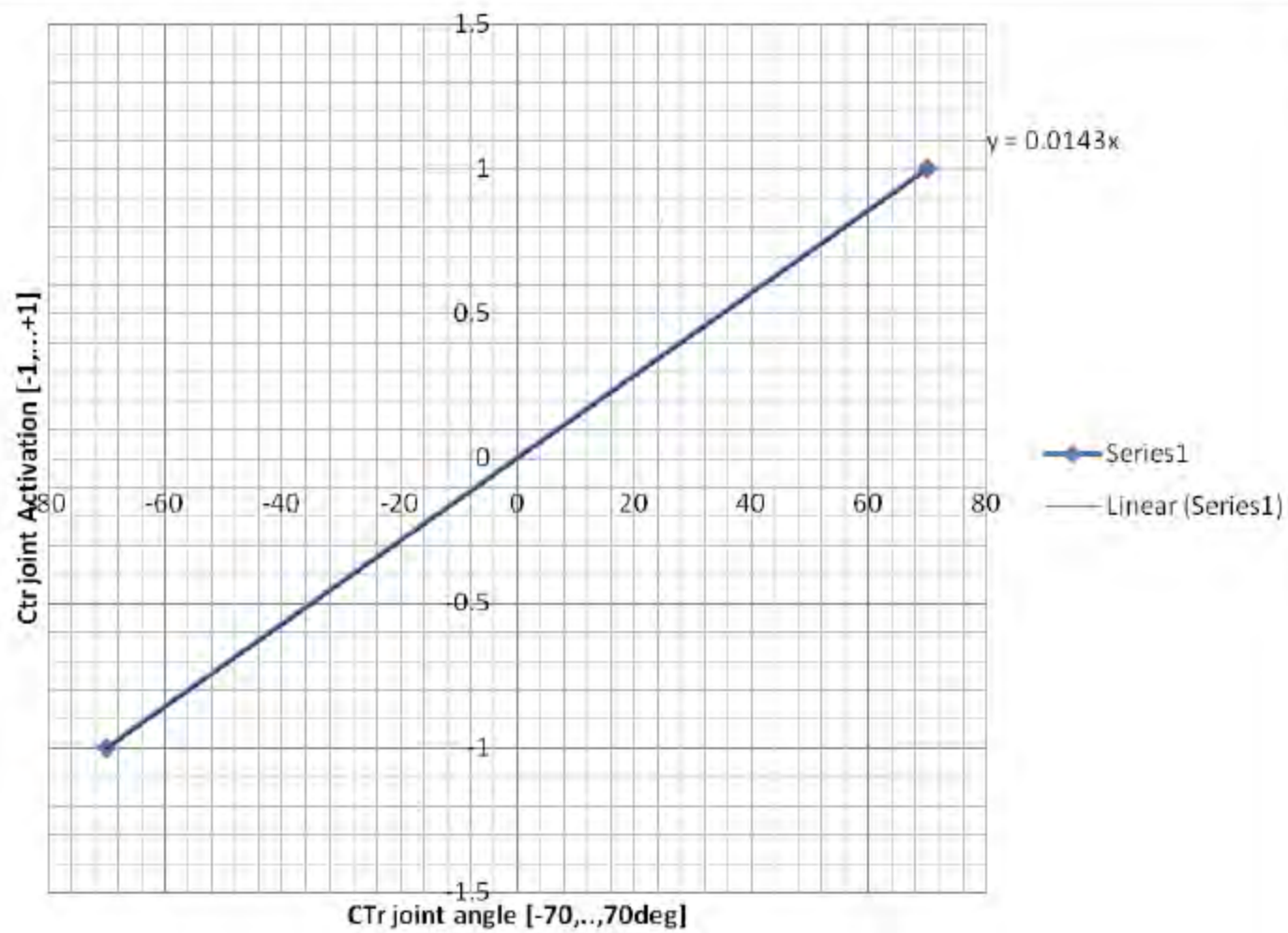
Deg	Activation
-60	-1.002
-50	-0.835
-40	-0.668
-30	-0.501
-20	-0.334
-10	-0.167
0	0
10	0.167
20	0.334
30	0.501
40	0.668
50	0.835
60	1.002

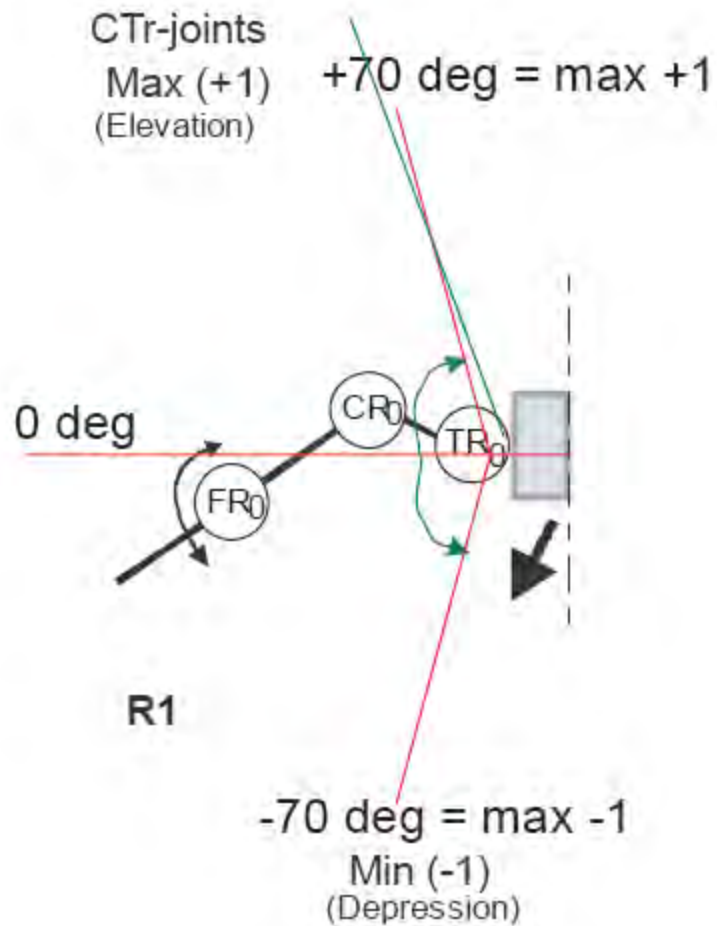


Hind joints

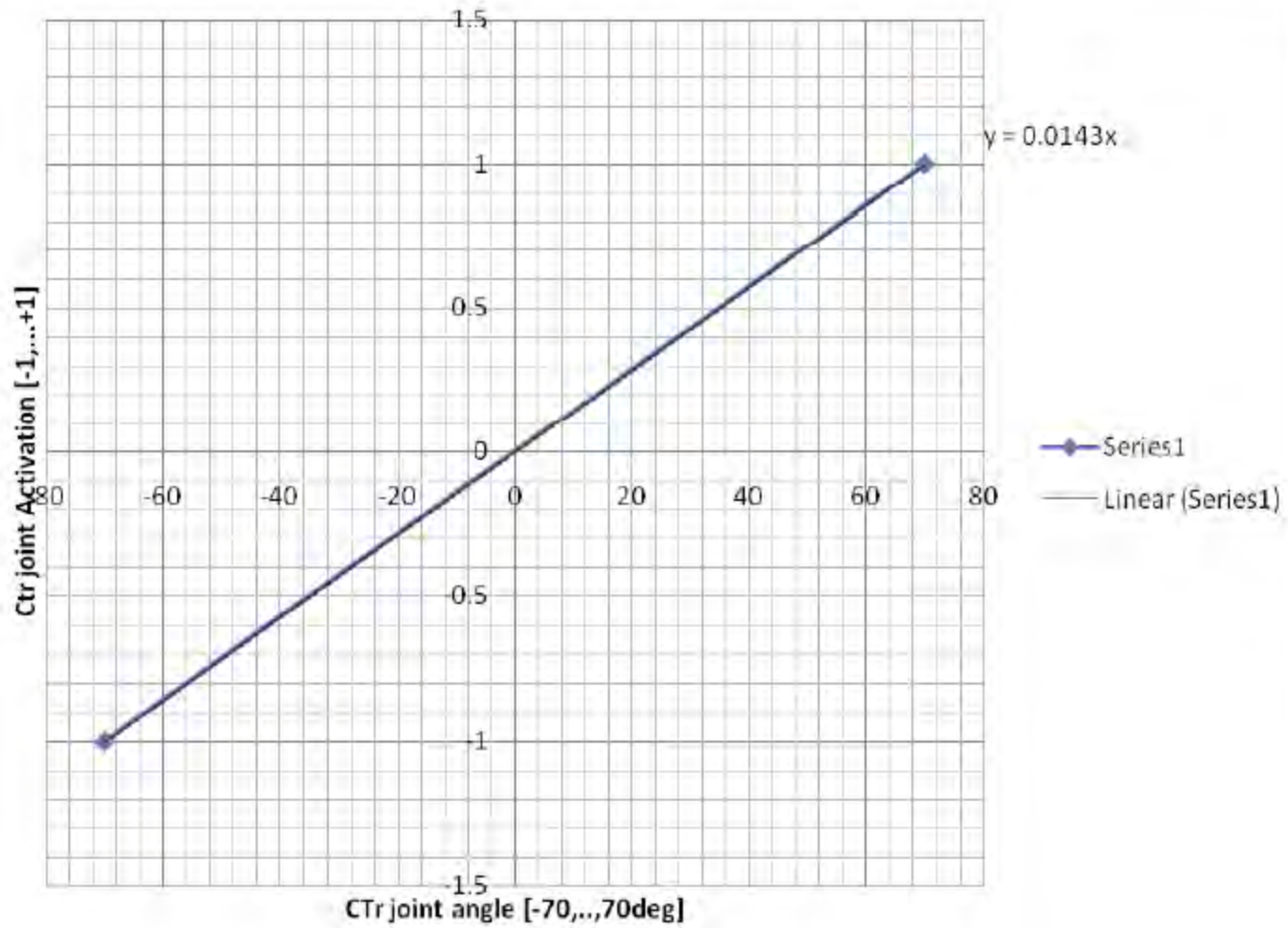


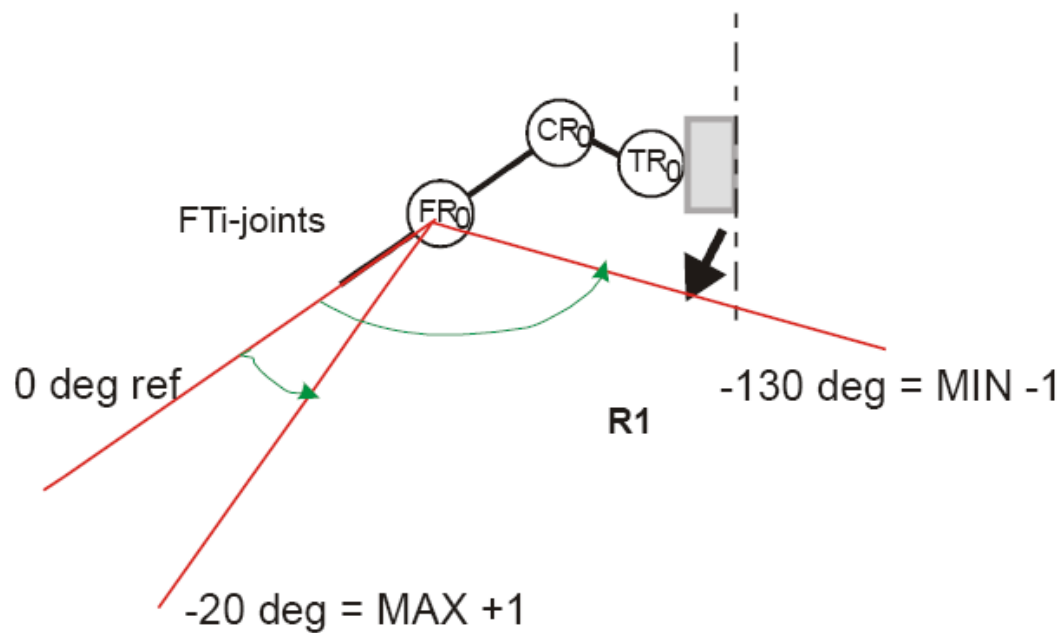
Deg	Activation
-70	-1.001
-60	-0.858
-50	-0.715
-40	-0.572
-30	-0.429
-20	-0.286
-10	-0.143
0	0
10	0.143
20	0.286
30	0.429
40	0.572
50	0.715
60	0.858
70	1.001





Deg	Activation
-70	-1.001
-60	-0.858
-50	-0.715
-40	-0.572
-30	-0.429
-20	-0.286
-10	-0.143
0	0
10	0.143
20	0.286
30	0.429
40	0.572
50	0.715
60	0.858
70	1.001





Activati on	
-20	0.9996
-30	0.8176
-40	0.6356
-50	0.4536
-60	0.2716
-70	0.0896
-80	-0.0924
-90	-0.2744
-100	-0.4564
-110	-0.6384
-120	-0.8204
-130	-1.0024

