Tekkotsu 4.0.1 Setup Tutorial

Tekkotsu 4.0.1 needs to be run in either Ubuntu or Mac OS. Currently, Ubuntu 14.04 LTS is the latest version that is compatible with the framework.

This tutorial shows how to set up Tekkotsu 4.0.1 in Ubuntu 14.04 LTS operating system.

1. For those who'd like to avoid the hassle of setting up the Ubuntu system and Tekkotsu framework, first download *http://home.cc.umanitoba.ca/~youngj/files/TekkotsuUbuntuImage.zip*

It includes a raw disk image of Ubuntu 14.04 LTS with a working version of Tekkotsu 4.0.1.

Next, follow Step 8 to restore the disk image in virtual machine software.

2. For those who prefer to install everything from scratch, start from Step 1.

3. For those who already have Ubuntu 14.04 LTS installed, follow the instructions starting at Step 2.

Step 1: Install Ubuntu 14.04 LTS

Install Ubuntu 14.04 LTS on a virtual machine:

1. Open VirtualBox and select New.

2. Choose your guest OS (e.g. Ubuntu) and architecture (32 vs. 64-bit), and set your Base Memory (RAM)

3. Click next until it shows the VM storage size. Put how much space you need depending on your storage and finish the wizard by clicking the create button.

4. On VirtualBox main window, select START and pick your MEDIA SOURCE. In your case, select the .iso for the system you want to install.

5. Finish the installation as normal install.

6. Remove your installation .iso from the virtual optical disk drive before restarting the VM.

Please refer to *https://askubuntu.com/questions/142549/how-to-install-ubuntu-on-virtualbox* for more details.

Install Ubuntu 14.04 LTS on your computer to create a dual boot system:

1. Download an Ubuntu Linux ISO file from the Ubuntu download page *http://releases.ubuntu.com/14.04/*. If you are installing on a 64-bit machine you must use the 64-bit Desktop edition; for a 32-bit machine, use the 32-bit Desktop edition.

2. Boot your machine from the install CD or thumb drive. If you're using an ASUS netbook, hit the Escape key (not F2) during boot, and tell the BIOS to boot off the thumb drive. Go through the normal installation process; you don't need to modify any of the defaults. If you're building a "standard" Tekkotsu robot, specify the username "Tekkotsu User", the userid "user", and any password you like. Reboot the machine after installation completes.

3. Apply system software updates by typing the following commands in a terminal:

```
sudo -s
apt-get update
update-manager
```

Ask the update manager to update all packages. This will take a while.

Step 2: Basic Setup for Ubuntu

1. Install some packages.

```
sudo -s
apt-get update
apt-get -y remove brltty openjdk-6-*
apt-get -y install cvs build-essential openjdk-7-jdk libjinput-java
apt-get -y install sox libasound2-dev libjpeg-dev libpng12-dev libreadline-dev
libxml2-dev libncurses5-dev
```

2. You also need to install Java3D in order to use Tekkotsu's 3D SketchGUI.

```
cd /tmp
if [ `uname -m` == "x86_64" ]; then
  wget http://download.java.net/media/java3d/builds/release/1.5.2/j3d-1_5_2-linux-
amd64.zip
else
  wget  http://download.java.net/media/java3d/builds/release/1.5.2/j3d-1_5_2-linux-
i586.zip
fi
```

```
unzip j3d*zip
rm -f j3d*zip
cd j3d*
unzip j3d-jre.zip
sudo cp -p lib/*/libj3dcore-ogl.so /usr/lib
```

3.. Perl and Ruby packages: Perl package installation (required for Tekkotsu state machine compiler): hit Enter once to accept all offered defaults. If you mess this up, you can reset CPAN by deleting the file /etc/perl/CPAN/Config.pm or the directory ~/.cpan.

```
perl -MCPAN -e "install Parse::RecDescent"
# Select "sudo" when warned about not having privileges to write Perl libraries
```

Ruby package installation (required for the WorldBuilder tool):

```
apt-get -y install ruby
gem install plist treetop
```

4. Set up .profile, .bashrc, and cvs files:

```
su user
cd
wget http://Chiara-Robot.org/download/dot-profile
mv dot-profile .profile
wget http://Chiara-Robot.org/download/dot-bashrc
mv dot-bashrc .bashrc
wget http://Chiara-Robot.org/download/dot-cvsrc
mv dot-cvsrc .cvsrc
cat /dev/null >~/.cvspass
source .profile
source .bashrc
mkdir -p ~/.config/gtk-3.0
cd ~/.config/gtk-3.0
wget http://Chiara-Robot.org/download/gtk.css
```

5. Raise the system memory lock limit (eliminates annoying error message when running Tekkotsu).

```
sudo -s
echo @user - memlock 1024 >>/etc/security/limits.conf
```

For more info, please refer to

http://wiki.tekkotsu.org/index.php/Ubuntu_install_instructions

Troubleshoot: If you choose to create a dual boot system or install Ubuntu on a flash drive, you may encounter a network issue where your Wi-Fi can’t be enabled.

Please try blacklist Acer_wmi by tying: `sudo tee /etc/modprobe.d/blacklist-acer-wmi.conf <<< "blacklist acer_wmi"` in the terminal

Note: Since Ubuntu 14.04 LTS no longer has security updates, it's not recommended to connect it to any network because of the potential cyber attacks.

The best way to get the files you need is through a shared folder between your host and guest system.

1. Download the files on a secured host OS.
2. Drag them to the shared folder.
3. Open it from the guest OS (Ubuntu in this case).

Step 3: Create a shared folder between the host and guest systems

1. Make a folder to be shared on your Windows 10 host operating system.

2. Make sure that you have an empty optical drive for your virtual machine. In order to do that, open the virtual machine Setting. Under Storage tab, check if the Controller: IDE is empty. Clear it out if it's not.

3. Boot your Ubuntu virtual machine. Go to Devices tab and click Insert Guest Additions CD image.
4. Open a terminal and install the Guest Additions CD.
5. We make a directory to mount the Guest Additions CD. The Guest Additions CD contains a file system type that we need to use.

```
sudo mkdir /media/cdrom
```

6. Now, we can mount the Guest Additions CD to the */media/cdrom* folder.

```
sudo mount /dev/cdrom /media/cdrom
mount: block device /dev/sr0 is write-protected, mounting read-only
```

7. We should install necessary packages if they have not already been installed.

```
sudo apt-get install make gcc linux-headers-$(uname -r)
```

8.Now, we can install and run the Guest Additions CD.

```
sudo /media/cdrom/VBoxLinuxAdditions.run
```

9. Click on the Devices tab and click Shared Folder Settings… and integrate your target shared folder.

Hit the add shared folder button. Click on the Folder Path list and find the shared folder on your host system. I checked off Read-only, Auto-mount, and Make Permanent. Hit OK.

10. Create a shared folder on Ubuntu virtual machine and mount the shared folder.

You make a directory on the Ubuntu virtual machine that will act as Ubuntu's shared folder.

```
mkdir ~/shared_folder_name
```

We mount the *shared* folder on your host system to the *~/shared_folder_name* folder.

```
sudo mount -t vboxsf shared ~/shared_folder_name
```

shared was the name of your host system shared folder, and I called the Ubuntu shared folder, ~/shared_folder_name, to differentiate between the two.
For more info, please go to *https://www.slothparadise.com/create-shared-folder-virtualbox/*

Note: If you prefer to delete VirtualBox Guest Additions, first click *Insert Guest Additions CD image* under *Devices* tab in *VirtualBox,* then find its directory in */media/usr/* and enter `sudo sh /VBoxLinuxAdditions.run uninstall`

Troubleshoot: If you have previously set up a shared folder but it doesn't work after you shut down the virtual machine, re-run `sudo mount -t vboxsf shared ~/shared-folder` to mount your shared folder where *shared-folder* is your shared folder's name. The problem does happen to me every time I shut down the virtual machine.

Troubleshoot: If you encounter error: `Could not mount the media/drive 'Program Files\Oracle\VirtualBox\VBoxGuestAdditions.iso' (VERR_PDM_MEDIA_LOCKED).`, you need to make sure that you have an empty optical drive for your virtual machine. Go to the setting of virtual machine, open *Storage* tab, click on the dropdown beside *Optical Drive* tag, and remove the current content on the drive.

Step 4: Installing OPEN-R SDK

In order to compile any code with Tekkotsu for the AIBO, you must first install OPEN-R and build a MIPS cross-compiler.

There are two major versions: OPEN-R SDK with GCC 3.3.6 and OPEN-R SDK with GCC 5.4.

The former works with Tekkotsu 4.0.1 and the latter is compatible with Tekkotsu 5.1. Currently, only version 4.0.1 works with AIBO.

Install Open-R SDK with GCC 5.4

Install git first by typing `sudo apt-get install git`, then type in `git clone git://git.code.sf.net/p/aiboplus/aibo-sdk` to download the repository.

Extract "*aibo-sdk-gcc5*" and enter the directory. Specify the directory for installing open-r by typing `sudo cmake . -DCMAKE_INSTALL_PREFIX=/usr/local/OPEN_R_SDK`

Compile the file with "*sudo make*"

If console shows the following error that stops compiling the code:

```
checking for the correct version of gmp.h... no
configure: error: Building GCC requires GMP 4.2+, MPFR 2.3.1+ and MPC 0.8.0+.
```

Please download "*gcc-5.4.0.tar.gz*" from https://ftp.gnu.org/gnu/gcc/gcc-5.4.0/ and put it in directory /aibo-sdk-gcc5/openr-gcc5

Type the following commands to install gcc 5.4.0' prerequisites:

```
tar xzf gcc-5.4.0.tar.gz
cd gcc-5.4.0
sudo ./contrib/download_prerequisites
```

If you encounter error `include could not find load file: /usr/lib/x86_64-linux-gnu/cmake/Qt5/Qt5Config.cmake" or "include could not find load file: /usr/lib/x86_64-linux-gnu/cmake/Qt5Core/Qt5CoreConfig.cmake"`, install Qt5 by typing *sudo apt install qt5-default*

"*newlib-1.15.0.tar.gz*" is missing in the directory. You need to download it from *http://tekkotsu.org/openr-install.html and put it in /aibo-sdk-gcc5/openr-gcc5*

Finish the installation with "*sudo make*".

For more information, please reference:
http://aiboplus.sourceforge.net/docs/openr_sdk_guide.pdf

Install Open-R SDK with GCC 3.3.6

Ubuntu users: You will need to:

```
sudo apt-get install flex gettext patch bison texinfo make g++
```

- build-aibo-toolchain-3.3.6-r3.sh
- gcc-3.3.6.tar.bz2 (23MB, do not unzip, mirrors)
- binutils-2.15.tar.bz2 (11MB, do not unzip, mirrors)
- newlib-1.15.0.tar.gz (10MB, do not unzip, mirror)
- OPEN_R_SDK-1.1.5-r5.tar.gz (12MB, do not unzip)

1. Download the files mentioned above from *http://tekkotsu.org/openr-install.html* and place them together into a temporary directory.
2. The build script will install the toolchain into /usr/local/OPEN_R_SDK. If you prefer a different location, edit build-aibo-toolchain-3.3.6-r3.sh.

    - If you change the installation directory, you will need to assign the path to the OPENRSDK_ROOT environment variable so Tekkotsu's build scripts will be able to find it later.

3. Run build-aibo-toolchain-3.3.6-r3.sh

    - You may need to 'chmod a+x build-aibo-toolchain-3.3.6-r3.sh' first.
    - You may also need to run the script as root, or otherwise create the OPEN_R_SDK directory and ensure your user account write access.

4. Your target directory (i.e. /usr/local/OPEN_R_SDK) should now contain:
   OPEN_R/ include/ lib/ mipsel-linux/
   bin/ info/ man/ share/

For more info, please refer to: *http://tekkotsu.org/openr-install.html*

Troubleshoot: if you encounter 'Running the script needs root access.' When try to run the script, execute *sudo ./build-aibo-toolchain-3.3.6-r3.sh*

Troubleshoot: You might encounter the following error message when running the script:

```
../../../binutils-2.23.2/bfd/doc/bfd.texinfo:325: unknown command `colophon'
../../../binutils-2.23.2/bfd/doc/bfd.texinfo:336: unknown command `cygnus'
WARNING: `makeinfo' is missing on your system.  You should only need it if you
modified a `.texi' or `.texinfo' file, or any other file indirectly affecting the
aspect of the manual.
```

It is telling you that Binutils isn't properly installed. The reason is one of its dependency texinfo has a higher version than what we need.

1. Remove the current texinfo via apt-get (I'm not sure if this is required, but I did it just to be safe).
2. Download the texinfo source:

```
wget http://ftp.gnu.org/gnu/texinfo/texinfo-4.13.tar.gz
```

3. Extract the source files:

```
gzip -dc < texinfo-4.13.tar.gz | tar -xf -
cd texinfo-4.13
```

4. Configure, build and install:

```
./configure
make
sudo make install
```

For more info, please visit: *https://askubuntu.com/questions/491184/texinfo-downgrade*

Troubleshoot: if you need to reinstall Open-R SDK, delete relevant folders with "*bld-*" in their names before running "*sudo ./build-aibo-toolchain-3.3.6-r3.sh*"

Step 5: How to build and run your first Tekkotsu program:

It's a good practice to copy the *Tekkotsu/project* directory to make multiple independent "projects". (You can change the name of the directory for each copy, but do not remove the original project directory.)

Run "*make*" in your project directory to build the default robot target listed in *project/Environment.conf*

Run Tekkotsu UI: `/usr/local/Tekkotsu/tools/mon/ControllerGUI localhost`

For more info, please visit *https://www.cs.cmu.edu/~dst/Tekkotsu/Tutorial/behaviors.shtml*

Tip: if your computer has a multi-core CPU, you can fully utilize it by appending “*-jx*” after “*sudo make*”, replaces x with the number of cores you have. For example, “*-j4*” means four cores will be utilized when compiling.

Tip: If you have trouble modifying *Tekkotsu/project/Environment.conf*, you might want to open it with “*sudo*”.

Troubleshoot: If you encounter the error message `cc1plus: error: unrecognized command line option "-std=c++11"`, it’s because the GCC version in OPEN-R is too old. Please install Tekkotsu 4.0.1 provided on *http://www.tekkotsu.org/downloads.html*. Remember to change the directory name (originally *Tekkotsu 4.0.1*) to *Tekkotsu* after unzip the file and copy it to */usr/local* directory.

Step 6: Mirage Setup (Optional)

Mirage is a simulation environment which can host multiple agents in a virtual world.

Mirage is only available in the source code of Tekkotsu 5.1, but not 4.0.1. And so far, I haven’t found a way to manually install it.

1. Install hardware graphics drivers.

If you are running Ubuntu on a Mac using an emulator, skip this step and proceed to step 2. If you are running on a machine using the Intel GMA 3600 graphics chip set, you may not be able to run Mirage due to lack of a fully functioning device driver.

Ubuntu installs by default with the Mesa indirect software renderer, but OGRE3D requires the hardware driver specific to the graphics board being used. To check whether you have the hardware graphics driver installed on your workstation or laptop, type:

```
sudo apt-get install mesa-utils
glxinfo | head
```

If the vendor string is "ATI' or "nVidia" then you're using the hardware driver and can skip the rest of this step. If the "server glx vendor string" is "SGI", then you're using the

Mesa renderer and may need to find and install the hardware driver for your board. However, some Intel and nVidia graphics boards do work acceptably with Mesa.

To find the manufacturer and model number of your graphics board, do:

```
lspci | grep VGA
```

Try turning on the hardware graphics driver by going to System > Administration > Hardware Drivers.

If you don't have a suitable hardware graphics driver, you may be able to install one using jockey-gtk:

```
sudo apt-get install jockey-gtk
jockey-gtk
```

2. Install library packages Ogre depends on:

```
sudo -s
apt-get update
apt-get -y install libfreeimage-dev libxrandr-dev libgl1-mesa-dev libglu1-mesa-dev
```

Install packages for building various additional libraries

```
apt-get -y install subversion libtool automake autoconf libxaw7-dev cmake
```

3. Install OIS, the Object-Oriented Input System.

```
apt-get -y install libois-dev
```

4. Install FreeType.

```
apt-get -y install libfreetype6-dev
```

The following must be done on Ubuntu 14.04 systems for compatibility with OGRE 1.7.4:

```
cd /usr/include/freetype2

ln -s . freetype
```

5. Install the OGRE3D rendering system version 1.7.4:

```
cd /opt
wget http://downloads.sourceforge.net/project/ogre/ogre/1.7/ogre_src_v1-7-4.tar.bz2
tar xjf ogre_src_v1-7-4.tar.bz2
cd ogre_src_v1-7-4
mkdir build
cd build
cmake .. -DOGRE_BUILD_SAMPLES=OFF
make -j4
make install
```

Note: if you get an "internal compiler error" with gcc 4.8, just type "make" again.

You can obtain other Ogre releases by going to the OGRE3D source download page at *https://sourceforge.net/projects/ogre/files/* and clicking on the version you want.

6. Install release 2531 (not the latest release) of the Bullet physics engine. Download it from the subversion repository and compile it by following the steps below.

```
cd /opt
svn checkout -r 2531 http://bullet.googlecode.com/svn/trunk/ bullet-280
cd bullet-280
cmake -DBUILD_SHARED_LIBS=ON -DBUILD_EXTRAS=OFF -DBUILD_DEMOS=OFF -DINSTALL_LIBS=ON
make -j4
sudo make install
```

7. Build Mirage. First check the file */usr/local/Tekkotsu/tools/mirage/Resources/plugins.cfg* and verify that *PluginFolder* is set to the Ogre library; by default, this is /usr/local/lib/OGRE. Then:

```
cd /usr/local/Tekkotsu/tools/mirage
sudo make -j4
```

For more details, please visit:

http://wiki.tekkotsu.org/index.php/Mirage_install_instructions

8. Run Mirage: `/usr/local/Tekkotsu/tools/mirage/mirage Mirage tictactoe.mirage`

Note: the SVN provided by the tutorial for downloading Bullet physics engine has expired. Download *“bullet-2.80-rev2531.tgz”* from the link here instead: *https://code.google.com/archive/p/bullet/downloads*

If you are running Ubuntu in a virtual machine, just ignore step 1 because VMs always use a virtualised VMware SVGA 2 video card. And it doesn’t allow you to install any hardware graphics driver mentioned in the tutorial.

Troubleshoot*:* When you install its dependencies, if the terminal shows “*Could not open lock file*”, please prepend “*sudo*” to your command.

For any further question, please visit the Tekkotsu discussion forum at *https://groups.yahoo.com/neo/groups/tekkotsu_dev/info*

Step 7: Create a Disk Image of Ubuntu (Optional)

If you ever need to create a backup for your Ubuntu system in the form of a raw disk image, please follow the instructions below:

dd is a low-level byte-for-byte copy utility that you can use to copy the whole system.

1. To list all the partitions on your drive, enter `lsblk`

On my virtual machine, the result is shown as following:

```
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0    20G  0 disk
├─sda1   8:1    0  17.6G  0 part
├─sda2   8:2    0     1K  0 part
├─sda5   8:5    0   2.5G  0 part [SWAP]
```

Adopted from *https://askubuntu.com/questions/425767/i-cant-tell-which-dev-sda-file-is-my-external-hard-drive-to-install-xubuntu-on*

2. Preparation

To take the advantage of high compression rate, you should overwrite the empty space with zeros:

```
## Most commands need sudo.
mount -o rw /dev/sda1 /mnt
dd if=/dev/zero of=/mnt/zero_file bs=32M
## Long wait here. Expect "no space left on device".
sync
rm /mnt/zero_file
umount /dev/sda1
```

Credit to Dr. Jim E. Young for the suggestion and finding *https://superuser.com/questions/1097210/clone-only-space-in-use-from-hard-disk*

Repeat this with /dev/sda2, /dev/sda3, ... except /dev/sda5

3. Copy

This method may be used if the target filesystem (where the image file will be saved) supports sparse files. To generate a sparse image file, invoke:

```
## dd probably needs sudo here.
dd if=/dev/sda of=~/shared-windows10/ubuntu.iso bs=512 conv=sparse
```

Note: There’s no status bar when *dd* clones your system. Please wait until final status is logged in the console.

Step 8: Restore your system (Optional)

In Linux system:

```
## dd probably needs sudo here.
dd if=~/target_path/ubuntu.iso of=/dev/sdb bs=32M
```

In Oracle VirtualBox:

1. Redirect into the VirtualBox directory and copy the disk image here.

2. Convert the image into a vdi file by entering `VBoxManage convertfromraw --format VDI [filename].img [filename].vdi` in the windows console and wait until it finishes.

3. When creating a new virtual machine, in the *Hard disk* step, select *Use an existing virtual hard disk file*, find and add the vdi file we just created and. Continue finishing the setup.

If you need to mount it to a flash drive, go to the setting for your virtual machine, select the *USB* tab, check the *Enable USB Controller*, and check the boxes in the sub option depending on your USB type. Restart your virtual machine and the flash drive should show up. Enter `sudo dd if=/dev/sda of=/media/user/NEW\ VOLUME/ubuntu.iso` to start cloning.

Troubleshoot: Files larger than 4GB can NOT be stored on a FAT32 volume. Thus, if your flash drive has a FAT32 file system, format it as NTFS as it will resolve this issue.

From my experience, exFAT is not supported by Ubuntu 14.04 LTS.

Note: When you format your flash drive as NTFS, it is NOT possible to write files on a Mac computer. Most Mac computers can read NTFS, but not write. And once the device is formatted as NTFS, you MUST use "Safely Remove Hardware" to remove your device. Adopted from *https://kb.sandisk.com/app/answers/detail/a_id/289/~/transferring-a-4gb-or-larger-file-to-a-usb-flash-drive-or-memory-card*

Frequently Used Ubuntu Commands

a. Remove a directory with its files: `sudo rm -rf directory`

b. Unzip a tar.gz file*:* `tar xvzf target.tar.gz`

c. If access permission is denied, run the command as super user by appending `sudo`

d. Rename a directory: `mv /home/user/oldname /home/user/newname`

e. Check your g++ and gcc version by typing `dpkg --list | grep compiler`

f. Remove package via package manager by entering `sudo apt-get remove <application_name>`

g. After uninstalling the package, to uninstall its dependencies that were automatically installed and are now no longer needed as dependencies changed, type `apt-get autoremove`

h. To shows you a package list sorted by size, enter `dpkg-query -Wf '${Installed-size}\t${Package}\n' | column -t`

i. To remove the account password, open the sudoers configuration file with `sudo visudo`.

Find the corresponding lines and change them as follows:

```
%admin ALL=(ALL) NOPASSWD:ALL
%sudo ALL=(ALL:ALL) NOPASSWD:ALL
```

Press `Ctrl+X` to exit the file and leave the file name unchanged when prompted.

Partially adopted from *https://askubuntu.com/questions/281074/can-i-set-my-user-account-to-have-no-password*

Note: If `Ctrl+X` doesn’t work, it’s probably because VirtualBox is using `Ctrl` as a default host key. With your virtual machine running, you can replace it by clicking the *Input* tab, then *Keyboard* followed by *Keyboard Settings*. Replace the *Host Key Combination* with your custom shortcut.

Note: If you want to reset a password, enter `passwd user -d`

Frequently Used Ubuntu Terms

Ubuntu live CD: A live CD can be used for a quick demo or test of Ubuntu without any changes to your machine. Adopted from *https://help.ubuntu.com/community/LiveCD*

Mounting: Mounting is the act of associating a storage device to a particular location in the directory tree. Adopted from *https://unix.stackexchange.com/questions/3192/what-is-meant-by-mounting-a-device-in-linux*

Boot Partition (also known as the system volume, system partition, and the boot volume): It is a computing term for disk partitions of a hard disk drive or solid-state drive that must exist and be properly configured for a computer to operate. Adopted from *https://en.wikipedia.org/wiki/System_partition_and_boot_partition*

Disk Image: A computer file containing the contents and structure of a disk volume or of an entire data storage device. Adopted from *https://en.wikipedia.org/wiki/Disk_image*

Mirror: A copy of programs available for download. Adopted from: *https://www.quora.com/What-do-mirrors-mean-in-Linux*

Root: Root (also known as superuser) access to run administrative tasks in Linux. Adopted from *https://www.wikihow.com/Become-Root-in-Ubuntu*

Sudo: a program for Unix-like computer operating systems that allows users to run programs with the security privileges of another user, by default the superuser. Adopted from *https://en.wikipedia.org/wiki/Sudo*

Apt*:* Ubuntu's Advanced Packaging Tool (APT) performing such functions as installation of new software packages, upgrade of existing software packages, and even upgrading the entire Ubuntu system. Adopted from *https://help.ubuntu.com/lts/serverguide/apt.html*

GNU nano: GNU nano is a text editor for Unix-like computing systems or operating environments using a command line interface. Adopted from *https://en.wikipedia.org/wiki/GNU_nano*

X Window System (X11, or simply X): A windowing system for bitmap displays, common on Unix-like operating systems. X provides the basic framework for a GUI environment: drawing and moving windows on the display device and interacting with a mouse and keyboard. Adopted from: *https://en.wikipedia.org/wiki/X_Window_System*

Unity (user interface): A graphical shell for the GNOME desktop environment originally developed by Canonical Ltd. for its Ubuntu operating system. Adopted from: *https://en.wikipedia.org/wiki/Unity_(user_interface)*

Toolchain: In software, a toolchain is a set of programming tools that is used to perform a complex software development task or to create a software product. Adopted from: *https://en.wikipedia.org/wiki/Toolchain*