Team Zero- New User Document

Authors: Wesley Adams, Adeshkumar Naik

Source Control

GitLab:

Team Zero Uses Gitlab for managing its software. The git repo hosted on gitlab that contains the team zero software Source code is named csc340ltp. It can be found here:

https://gitlab.mcs.sdsmt.edu/7442182/csc340ltp

You should be able to login to your SDSMT Gitlab account and have developer access to the Active Directory of the project to gain access for development. Some branches in Gitlab are restricted by group membership and you may need to ask for permissions when required to do your job.

After logging in follow the instructions set provided in the Wiki of the Gitlab software.

Development Setup:

The Team Zero uses specific development platform. The current development platform the Team Zero Software uses Linux and OpenGL. The Communications team of Team Zero has put up instructions for extra tools and setups.

Set up instructions:

1) The team prefers using Windows Linux Sub System for Software Development. You are welcome to use your own VMware or have a own Ubuntu OS running.
If you are planning on using the Windows Linux Subsystem that the team members use then following is the Instructions for getting you to setup Subsystems.

1) Open the Windows PowerShell as an administrator.

a. Click the Windows Button or Icon in the bottom left corner of the screen.

b. Type in PowerShell to search for the windows powershell.

c. Right click the PowerShell icon and select run as 'administrator'.

2) Enable the Windows Sub-system for Linux

a. Copy the following command and paste it into the powershell and hit enter:

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```

b. After the configuration hit 'Y' to restart your computer

3) Open the Microsoft Store and install Ubuntu 18.04 (It is free, if you see a price it is the wrong one)

a. Click the windows button or icon in the bottom left corner of the screen.

b. Type in Store in the search and select the Microsoft Store

c. In the top right search bar type in Ubuntu 18.04

d. Scroll down to apps and select Ubuntu 18.04

Note: it is free do not select any option that has a price.

e. Click on the get button

f. That will download and install the initial part of Ubuntu 18.04

4) Finish Installing Ubuntu

a. Click the windows button or icon in the bottom left corner of the screen.

b. Type in Ubuntu and select the icon, or click the icon from the recently added list

c. This will open a terminal window that begins finishing the install of Ubuntu

d. The terminal will then prompt you for a user name and password. It is recommend to set these up and not directly use the root user.

2) You might be required to setup certain necessary packages which might be missing. The following are the instructions for the same:

1) If Ubuntu is closed, open a terminal window from the windows start menu.

Note: This should leave you sitting at a command prompt that looks similar to jon@T-SMD1026564:~$. This will be your Home directory

2) Enter the following text to update your currently installed programs on Ubuntu:

a. sudo apt-get update   <- Pulls update list

b. sudo apt-get upgrade   <- Performs update list

c. sudo apt-get dist-upgrade   <- Updates packages that were held back

d. sudo apt autoremove   <- Deletes old files that were not removed after updates

3) Install missing software.

a. if you didn't just run update and upgrade, run sudo apt-get update

b. Run the following commands

sudo apt-get install gdb freeglut3-dev gcc g++ git   <- Installs   necessary tools for development *See below for descriptions : test

Note: For full Linux installations some of these will already be installed

wget -P ~ git.io/.gdbinit   <- Adds a graphic interface to gdb for debugging

Note: The following command is not for full installations of Linux

echo "export DISPLAY=:0" >> ~/.bashrc   <- First step to allow windows users to have a program display

Note: If you are using Linux and typed in the above command edit the ./bashrc file and remove it from bottom of list

Windows Users Only

4. Install Xming

a. Download the file at this link. (It will start automatically after 5 seconds)

https://sourceforge.net/projects/xming/files/latest/download

b. Run the file either from where you saved it or the download completion notice

c. You can leave everything as the default

d. Run xming from the windows start menu.

Note: After this step you should have a little X in the notifications area or in the box when you click on the up arrow in the bottom right corner.

When you mouseover the X it should say: Xming Server:0.0

You should now be ready to develop in linux using OpenGL.

3) Clone Repository:

Prepare directories

a. In your Linux Terminal create a directory to store the project in.

```
mkdir -p ~/CSC_340/Tanks
```

b. This will create two directories from your home directory to a directory called tanks.

c. change directory to the new directory.

```
cd ~/CSC_340/Tanks
```

Begin Cloning

a. If you have not setup SSH access to the school's getlab:

```
git clone https://gitlab.mcs.sdsmt.edu/7442182/csc340ltp.git
```

b. If you have setup SSH access to he school's getlab:

```
git clone git@gitlab.mcs.sdsmt.edu:7442182/csc340ltp.git
```

Change directories to repository

a. Right away

```
cd CSC340ltp
```

b. Any other time you want to develop

```
cd ~/CSC_340/Tanks
```

4) Enabling SSH

https://help.github.com/articles/connecting-to-github-with-ssh/

Branching

Team Zero has a specific way of making its team members write Production Level Code. You are requested to follow the Team Zeros method.

Do all of your work while writing any Production Code on one of the specific branches available. Make sure to create an Issue on Gitlab and have it assigned to you if you are going to work on it. Also assign it the particular Milestone that you want to get it done for.

Once you have done the work on your local repo push it to particular remote repo. When you do a push please include the issue for which the change or the production code is for by including #issue-number.

Code Review

Code reviews is one of the most important standard coding practices that Team Zero follows for it's development process. During the Code Review One of the team members will have a milestone opened up and create the issues which will be primarily assigned to team members who haven't written code for that particular branch or files. The main aim behind Team Zero's Code Reviews is to spread the knowledge of the codebase and finding bugs or optimization in the existing Code Base.

All the things pointed out during the code review are to be discussed in Discord Chat by talking to the team members responsible for writing the code reviewed. The code review can be viewed in the design documentation

Permissions:

In General, the security philosophy is "we want you to have access to everything you need to effectively do your job, but nothing else"

Issue Management:

Team Zero uses Gitlab Issue Management feature to manage our backlog of our work. We use this Issue Management Feature to plan our new features and track bugs etc. Anyone who creates an issue is expected to make a clear description of what he wants the assignee do.

Defect Tracking:

You are encouraged to track bugs and defects found in all phases of development process, not only with respect to the Code Base. If code is working correctly but if you feel we can optimize or if there is a defect in the process then you can create an Issue on GitLab and discuss about it in the General Team Zero Chat Channel.
This means that you can create a new issue against the following lifecycle phases: Requirements, Design, Implementation, Test

Tests – Continuous Integration

The Team Zero Software has built in Tests that run whenever you make a push to Gitlab. Always use the Compile Tests through your terminal on your local repo before making any push to Gitlab.

After you find all tests passing on your local repo make a git commit. Once you push a particular commit The Gitlab runners will run a bunch of coverage tests. These tests are in a pipeline. Following is a breakdown of the tests run.

1) Build Test
2) Prog Test
3) Error Test
4) Catch Test
5) Coverage Test

These tests run for about two minutes for every commit until it fails one of the test in the above sequence. So please do have patience..!!

You will also receive an email on your school email ID for every Gitlab Pipeline failed. You can look at the place where the fail build and narrow down your search on your mistake/bug.

First Time Build:

This project requires freeglut3-dev to be installed on the linux device. From the linux terminal this can be accomplished with the following command:

```
sudo apt-get install freeglut3-dev
```

The project is currently set to use a makefile to automate the building process. After cloning the repository

a. mkdir tank

b. cd tank

c. git clone https://gitlab.mcs.sdsmt.edu/7442182/csc340ltp.git

   or git clone git@gitlab.mcs.sdsmt.edu:7442182/csc340ltp.git

d. cd csc340ltp

e. make

Hint: use make help to see the functionality of our makefile.

Open Source Software:

Key things to remember:
We want to maximize our work not done. Thus we make use of Open Source Software to reduce Development Costs wherever it makes sense. If you are working on the project and you feel we should be using an addition software package or any other tool then please contact the Devops Team.
Following are the Open Source Software that Team Zero uses.
1) Ubuntu 18.04 (Subsystem available on Windows App Store)

2) Git- Source Control Management Software

3) Gitlab- A software Development Server that provides git repo management along with issue management, wiki, etc.

Each of these has a specific license. Sometimes it may contain other components under multiple licenses.