Setting up de project

# Easiest way:

### Getting the project:

Go to: <https://github.com/YVaraug/SpaceShooter>

Press: 

Select download zip, and extract were preferred.

### Getting Unity:

Go to: <https://store.unity.com/es/download?ref=personal>

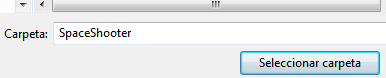
Download and execute installer for your system

### Opening the project:

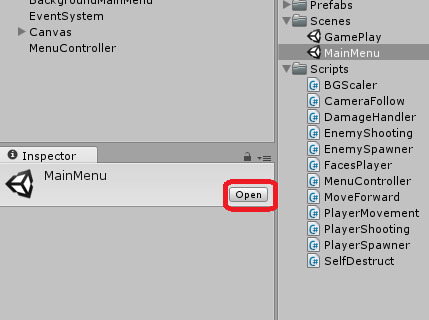
Execute Unity: 

Open directory were you extracted de zip:





In unity open Main Menu Scene:

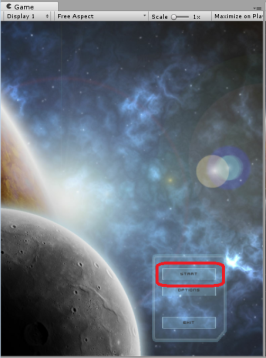


### Playing:

Press Play:



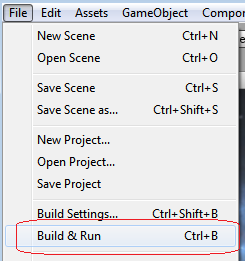
And … Start playing in game window



…. Stop playing



### Creating an executable:



…..

# More Advanced Way ( using Git version control ):

All the same but we get the project with Git instead of ZIP, so we can collaborate with others.

### Get Git Client:

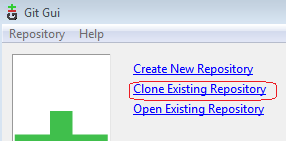
Go to: <https://git-scm.com/downloads>

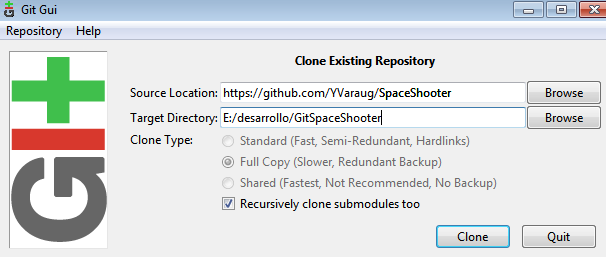
Execute your download.

### Getting the Project with Git:

Execute Git: 

Get the project:





Target directory must not exist.

The rest is the same (get unity, execute with unity,..)

# Even More Advanced ( collaborating with Git):

### Functionality:

If you have the Git Client, you can get or make changes that everyone can see, at the same time. Git will control there are no collisions between changes, and you can always chose if you want to get the newest changes or keep with your own version.

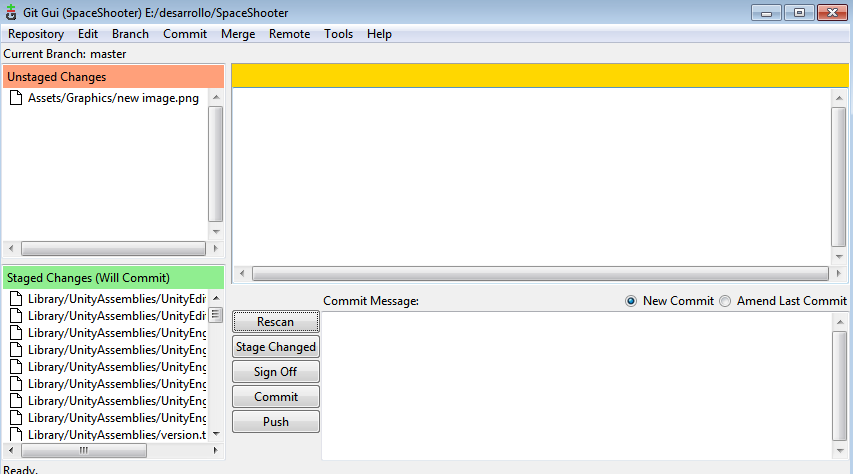
### Main view:

Local Remote

Workspace -🡪 Staged -🡪 Commit --Push🡪 Everyone´s Pushed Changes

### Operating:

You work in your normal directory which is the workspace with unity or your editor, when you think some small current changes are good enough, you go to git and get the affected files staged, and then make a commit with observations telling what the change is meant to do.



Press **Rescan** tosee which files have changed.

Select (press icon) on top (unstaged) files of the new functionality (normally all); if you want to select all press **Stage Changed** .

Write what the changes and functionality is in the Commit Message window and press **Commit**.

If you want to share the changes with everyone press **Push**, and everyone will be able to get the new changes.

If you want to get new changes (pushed by others) in the remote server go to **Remote>Fetch from>origin**.

Then if you want to include this changes do **Merge>Local Merge**. And press the **Merge** button.

If there are any conflicts that git can’t resolve he will tell you.

As this is a very common practice there is a shortcut called **pull** which fetches and merges in one command.

Although this is a command line instruction so you need to **Repository>Git Bash** to introduce the command.

If you want to include especial commands in de GUI you can from **Tools/Add**.

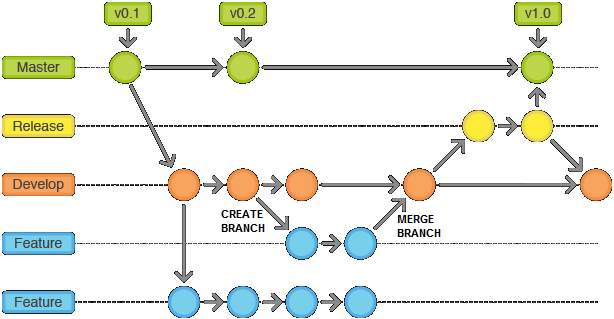
For most useful commands goto: <https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf>

Best way to keep things clean (using branches)

The best way to keep things clean is to use branches. Branches are conceptually complete independent projects (although they share the main trunk files).

In this way each person can work in different branches and/or there can be different branches for different parts of the project.

Also branches like a tree come from common places so they share common files.



But we can merge branches so we can add the new feature to the main trunk/ branch.

In this case you would fetch and merge from your main trunk and not from origin/master.

Then some supervisor can eventually merge your main trunk to origin.

More advanced:

Want to see a simplified flow video: <https://www.youtube.com/watch?v=-N4Cghw0l2Q>

You like advanced: <http://www.cheat-sheets.org/saved-copy/git-cheat-sheet.pdf>

Want resume and best practices: https://www.git-tower.com/blog/git-cheat-sheet/

Have a problem, go to: <http://justinhileman.info/article/git-pretty/git-pretty.png>

Even more advanced: read the manual at <https://git-scm.com/doc>