Connecting to GitHub

1. Create an account at [www.GitHub.com](http://www.GitHub.com)
2. Download SourceTree at <http://www.sourcetreeapp.com/>
3. After installing SourceTree, start the program and log in with your GitHub information. (Allow it to create a global gitignore file).
4. Go to Tools -> Options. In the general tab, enter your full name and Email address. (This step is incredibly important)
5. Click on the Clone/New button in the top-left corner. Copy this URL and paste it into the Source Path/URL box in the Clone Repository tab: https://github.com/Friedchiken777/GAM23TowerD.git. Set the destination path to where you want the files to go, then click the “Clone” button at the bottom.

Note: If you are getting a message saying that something is not a valid destination path (it is buggy sometimes), Create an Empty Folder, name it what you want, select that path. If it still doesn’t work, open up the directory, delete that folder, hit cancel, and hit “Ok”. It should work after that. It’s a terrible workaround, but it does work.

Using the project in Unity

With the repository cloned, the files in the repository will be in the directory you set the destination path to. Open Unity and open the project folder of the repository you downloaded. If you already have a project open, File -> Open Project -> Open Other -> Find the repository that you cloned.

Committing a change to the project

After you have made changes in the project and want to commit those changes to the repository, open up SourceTree and click the “Commit” button. In this tab, you will see which files have been added and which ones have been changed. If any files have been changed from what the repository currently has, you will see the differences on the right-hand side if you click them on the bottom portion. If you are okay with those changes, click the checkbox next to the file. Once you confirm which files you are committing, click on the comment box at the bottom, and be verbose about what you are committing. What changes have been made? What files have been added? After your comment, make sure you note who it is from (e.g. “- Anthony R.”).

NOTE: Only commit to the repository if there are no errors with anything you are committing. Don’t make others handle your errors! In addition, make sure before you commit, you “Fetch” and “Pull” any changes to make sure none of your changes or additions is overlapping with someone else’s. (Please see the next section)

“Pulling”, “Fetching”, and “Pushing”

Others who have access to the repository may be committing changes while you are working on your part. If you have made no changes before the last time you worked on the project, make sure you click the “Fetch” button at the top of SourceTree. You will then see all of the changes that have been made to the project since the last time you “Pulled.” You can see any changes that have been made to existing files and if any new files have been added. Then, click the “Pull” button, and your files will be updated.

As stated in the previous section, if you have been working on something and want to commit, make sure you “Fetch” first to see any changes. If a file has been changed locally and you try to “Pull” it will throw an error telling you to commit or stash your changes. If you commit your changes and then pull, it will merge the files together and you must go through them and make sure any changes are dealt with. Make sure you communicate with someone else before changing their code to work with yours if there is a problem (For art assets, if everyone is working on their own art, there should be no commit problems, unless something is named the same as something else). Once everything works and there are no errors, commit, make sure there is nothing else to pull, and then “Push” the changes. “Pushing” the changes will update the repository on the site and it will be a change that everyone can then see.

NOTE: Never, under any circumstance, must you ever push a project that has an error in it.

Special Note for Artists:

Because there is a limited amount of data we can store on the repository (1 gig), make sure anything you push is something that is going to be a part of the game. For example, do not put concept art or models in the Assets folder. If it is a rough model/texture for something that is going to be updated and used, that is okay to put in. If the art ends up taking more space than intended, we will use another way to handle assets.

When organizing your art, make sure there are folders and sub-folders so everything is as organized as possible. For example, in the Assets folder, Create a “Models” folder, then “EnemyModels”, and “OutdoorEnemies” folders, if those are appropriate outcomes for how enemies will be. We need to make sure everything is as organized as possible from the start so nothing is out of place later.