Gitlab has a desktop application with a GUI, but personally I find using a terminal is a lot easier. How to Get Started is on next page. If you need any help, just text me and I’ll help out!

Useful Commands  
-Most of these are Linux commands, so they will work there too!

pwd : print your current location within the file system.

ls : print contents of the current directory  
Useful variants:  
ls (filepath): print contents of a different directory, not the one you’re currently in.   
 Ex: ls ISUGameJam2020/sprites/bullets : print contents of the directory named “bullets”  
ls -l or ls -l (filepath): print extra info about the files   
ls -a or ls -a (filepath): print all contents of the directory, even normally hidden stuff

cd (targetDirectory): move to another directory. “.” is a shortcut for “the directory you’re currently in” and “..“ is a shortcut for “the directory containing the directory you’re currently in” (so one directory above. The parent directory.) Push Tab twice to auto-fill a name and cycle through to the one you want.

mkdir (newDirectory): create a new directory

clear : Clears your terminal so it looks nice and clean again.

rm : remove a file, but *don’t use this it’s really powerful and easy to delete more than you want just use a different method*

git status : Prints a lot of helpful info. Just type it every so often and you’ll see.

git add (filename) : Tells Git “next time I make a commit, add this file to the commit”. Use git add -A to add all changed files.

git pull : Overwrite your local copy with the most up-to-date version of the files in the repo. **Warning:** git pull overwrites your local copy with the repo copy, so any uncommited changes will be deleted!

git commit : Creates a permanent snapshot of the local files you have used git add on. **Does not** add your local files to the repo.

git push : Adds all the commits you’ve created to the collection of commits on the repo.

How To Get Started:  
- Download Git Bash from this link: <https://gitforwindows.org/> This is what you’ll use to push and pull from Git.  
  
- Start Git Bash. Use the commands above to navigate to where you want your local copy of the repository to be stored. I recommend making a directory specifically for local copies of repos, in case you use more repos in the future (you can use mkdir or something else to make the directory, it doesn’t matter). Don’t make a directory named ISUGameJam2020 - Git will do that automatically.

-Enter the command **git clone https://github.com/WilliamLavelle/ISUGameJam2020.git**

Normal Usage:

-Before you start editing files, start Git Bash and navigate to the ISUGameJam2020 directory. Enter the command **git** **pull** to make sure your local copy is up-to-date with the repo. **Warning: git** **pull** overwrites your local copy with the repo copy, so any uncommited changes will be deleted!

-Edit the files in your local copy however you want. You can close Git Bash if you want.

-When you’re done, start Git Bash and navigate to the ISUGameJam2020 directory.

-Enter the command **git** **add** **(filepath)** for every file you’ve changed, or **git add -A** to add every edited file at once.

-Enter the command **git commit**. A file opens in the terminal. Press the I key to enter Insert Mode. Type a message describing this commit at the bottom of the file (with no # in front of the message). Press Esc to get out of Insert Mode. Type **:wq** to save and quit, or **:q** to quit without saving.   
**Note:** This alone does not make your change appear in the repo where others can see them! You can make multiple commits between synchronizing yourself with the repo. You have to use **git push** to make all your commits appear on the repo at once.

-Enter the command **git push**. Venmo William $1, and you’re done. (Don’t forget to do all of this step - it won’t work otherwise)