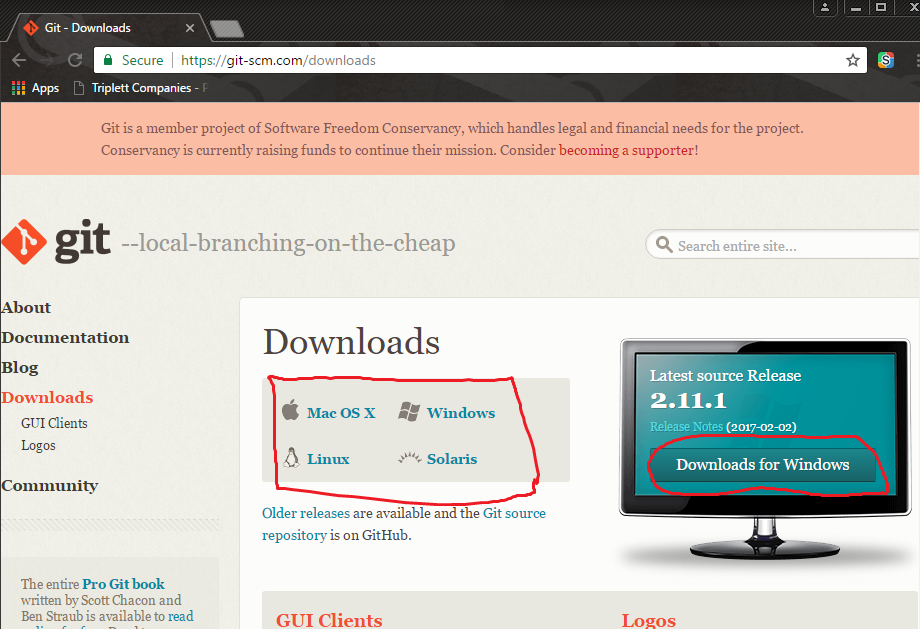
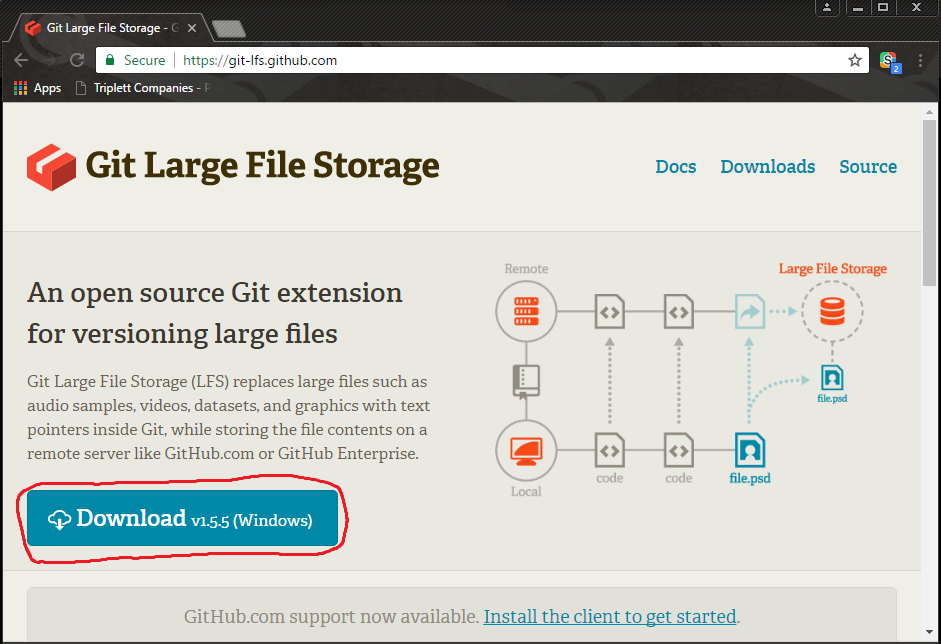
* **Installing Git-Bash**

1. Navigate to Git-scm.com/downloads and select what system you’re using and press *Download.*
2. Open the Downloaded File and start installing Git-Bash.
   1. On **GNU General Public License** page press *next.*
   2. Select where you want Git-Bash to install, press *next.*
   3. Select whatever components you need, the default components work just fine, press *next.*
   4. Press *next* on the **Select Start Menu Folder.**
   5. I typically select **Use Git from Git Bash Only,** select what you believe you you’ll use most and press *next.*
   6. Select **Checkout Windows-style, commit Unix-style line endings**, and press *next.*
   7. Select **Use MinTTY**, press *next.*
   8. On the **Configuring extra options**, it doesn’t matter what is selected, I use what’s selected at default and press *next.*
   9. Press *install*, and wait until it’s done, then press *finish*.

* **Installing Git-LFS**

1. Navigate to git-lfs.github.com and press *Download.*
2. Open the Downloaded File and start installing Git-LFS
   1. Accept the License Agreement and press *next.*
   2. Select where you want Git-LFS to install and press *next.*
   3. Git-LFS will install, press *finish*.

* C:\Users\bradr\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Gitbashicon.png**Setting up Git-Bash and GitHub**

1. Open the Git-Bash terminal.
2. Type ***cd ~***  to make sure you are in the home directory.
3. Type ***git lfs install***, you should see **Git LFS initialized.**
4. Type ***git config --global user.name “your username”*** and ***git config --global user.email “your email”***
5. Type ***ssh-keygen -t rsa -b 4096 -C “your email”***
6. The command above will first ask where you want to save the key, just press enter and it will save at the default location. Then it will ask for a password, use one you will remember.
7. Type ***eval $(ssh-agent -s)*** and it will give you an **Agent pid.**
8. Type ***ssh-add*** and enter the password you used above.
9. Type ***clip < ~/.ssh/id\_rsa.pub*** (this will copy your key to your clipboard)
10. In a browser go to *github.com* and sign into your account
11. In the upper-right corner click your profile photo, then click **Settings.**
12. In the user settings sidebar, click **SSH and GPG keys.**
13. Click **New SSH key** or **Add SSH key.**
14. In the **Title** field, add a descriptive label for the new key. For example, if you’re using a personal laptop, you might call this key “*My Personal Laptop”*
15. In the **Key** field, paste the contents of step 9. (it should start with *ssh-rsa* and end with your email)
16. Click **Add SSH key.**
17. If prompted, confirm your GitHub password.

* **Cloning MetaBlast**

1. Open the Git-Bash Terminal (see **Extra Notes**)
2. Navigate to where you want MetaBlast. (Example ***cd ~/Documents***)
3. Type ***git lfs clone git@github.com:MetaBlastTeam/MetaBlast.git*** , this will take around 30minutes to an hour and you shouldn’t be prompted with any passwords during any of this, if you are then something above went wrong or you didn’t do the **Extra Notes**.
4. Once MetaBlast has successfully been cloned, open it in whatever version of Unity you would like. It will import assets and recreate the **.meta** files. This will take about 30 minutes to an hour.
5. Once done close out of Unity completely and reopen it. Then you should be good to go.

* **Extra Notes**

1. Every time you open a new Git-Bash terminal run the following commands:
   1. ***eval $(ssh-agent -s)***
   2. ***ssh-add***
   3. You will be prompted to enter a password, enter the password you used for **Setting up Git-Bash and GitHub** step **6.**
   4. If you don’t do this every time you open a new Git-Bash terminal you will get all sorts of problems with Git-LFS and such. To automate this do the following:
      1. Type ***cd ~***
      2. Type ***touch .bashrc***
      3. Type ***notepad .bashrc***
      4. This will open notepad, inside notepad type ***eval $(ssh-agent -s)*** press enter and type ***ssh-add***
      5. Save the notepad document and exit out of notepad and Git-Bash
      6. Now every time you open a new Git-Bash terminal you will be prompted for a password.
      7. You can always edit the **.bashrc** file. For example under ***ssh-add***  you can add ***cd ~/document/metablast***  to have Git-Bash prompt for a password and put you in the home directory of metablast without having to do this every time by hand.
2. Now you should be good to go!

* **Extra Extra Notes**

1. Make branches in git while doing ongoing work. For example, If you’re working in Leaf\_Exterior level make a leafExterior branch to work in to commit and push fixes to.
   1. ***git branch leafExterior*** this creates the leafExterior branch.
   2. ***git checkout leafExterior*** this checks out the leafExterior branch where all your work that you have previously done inleafExterior is held and keeps your work away from other people’s/branches work.
2. Make sure you’re committing often
   1. ***git status*** this shows you what has been changed (red) and what has been added but not committed (green).
   2. ***git add .*** or ***git add [file(s) that are red in git status]*** this adds the files where they wait to be committed.
   3. ***git commit -m “Short and sweetly explain what was done”*** this creates a git commit with a description
   4. ***git push origin [branch name]*** this pushes your commit(s) to GitHub.