Gitkraken is like the supply depot for the noahs ark that is GitHub. In order to use the 2 compatibly however one has to align the computer and internet in just such a way that they can find and track one another and keep data safe from the flood. Once this is done however there is a fragile harmony that if well balanced, will allow the user to “push and pull” files to and from any computer that has access to both Gitkraken and GitHub via the internet. I hope I can adequately explain the process of syncing computer and site as I understand it.

GitHub is an open source distribution system that allows for file sharing between users and different computers. This works through a series of repositories and Remotes to allow the same user to access content via a different computer. Git Kraken is, like I said above, the depot for that system that we are using. In order to get content from the computer to GitHub one must first create an account on both the site and the program. next the user has to authenticate their computer by generating and syncing an SSH code. Upon authentication the user is ready to set up and clone a Repository. Cloning a repo is a way to sync your computer to Github. A repo on GitHub is the place you set up to keep and distribute all of your files for a specific project. Cloning your repo with GitKraken allows you to have access to that repo on your local machine via copying the repository and creating a pathway between the supply and the Ark of GitHub. Once that is done the user can begin to communicate and transport files (happy animals) back and forth between the two (two by two).

For those of us in the class this means we can “push” our Maya and Unity projects from our computer and access them from any other computer that has GitKraken. We can also access other files that have been forked to us. In order to do this the user has to go through a few steps. first the project must be saved within a folder where it will show up as a repo for both GitKraken and GitHub. If the authentication process worked in the steps previously then this should already be set up and the user just has to make sure they save their scenes to that file. next the user has to “Stage” the files they want to go to GitHub. this means they are loading the files into the transport to send to the Ark while the rain starts. They write a comment labeling the launch and hit “commit”. This means the transport is only waiting for clearance in order to head into the rain towards the Arl. The user then hits “Push” and this sends the files to the big ol boat. The moment of truth has arrived, refresh the GitHub page and the files should show up in the appropriate Repository ready to be pulled from another source. That process is basically the steps of pushing only in reverse. Allowing GitHub to release the files onto a different computer.