

# Github, GitKraken & Unity

Bennett Foddy, NYU Game Center

Adapted from Robert Yang's SourceTree cheat sheet

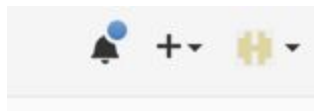
## 1. Unity Editor

- Create or open a Unity project, and at least save an empty scene in the assets folder. Git only cares about a folder if it has at least 1 file in it.
- Go to Edit->Project Settings->Editor, and set 'Version Control Mode' to 'Visible Meta Files'. Meta files are little files that Unity uses to store your import settings for each file in your project.




## 2. Github.com (In your web browser)

- Go to GitHub, create an account. Click the + icon to add a new Repo in the upper right of the interface



- Give your repo a useful name, like 'MyCoolGameProject', not an automatic suggestion.
- On the New Repo Page, enter the name of the project
- Click the little 'add .gitignore' button and select Unity. This will tell Git not to add Unity files which are temporary or specific to your particular computer.
- Click the 'Create' button.

Owner:  bfod / Repository name: MyCoolGame2 ✓

Great repository names are short and memorable. Need inspiration? How about [didactic-system](#).

Description (optional)

☒ Public  
Anyone can see this repository. You choose who can commit.

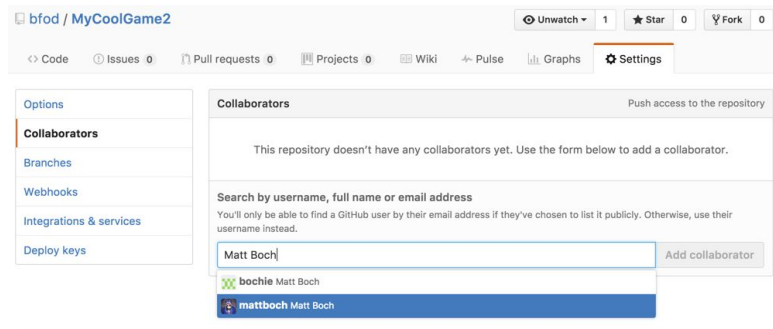
☐ Private  
You choose who can see and commit to this repository.

☐ Initialize this repository with a README  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **Unity** | Add a license: **None** ⓘ

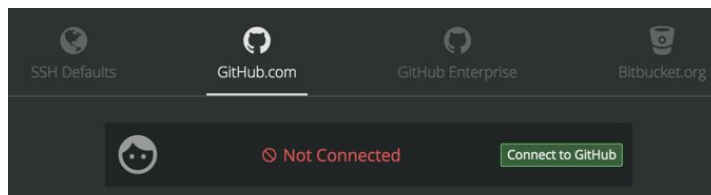
**Create repository**

- f. From the repository screen, go to 'Settings' so you can add any collaborators you want to work with.



### 3. GitKraken

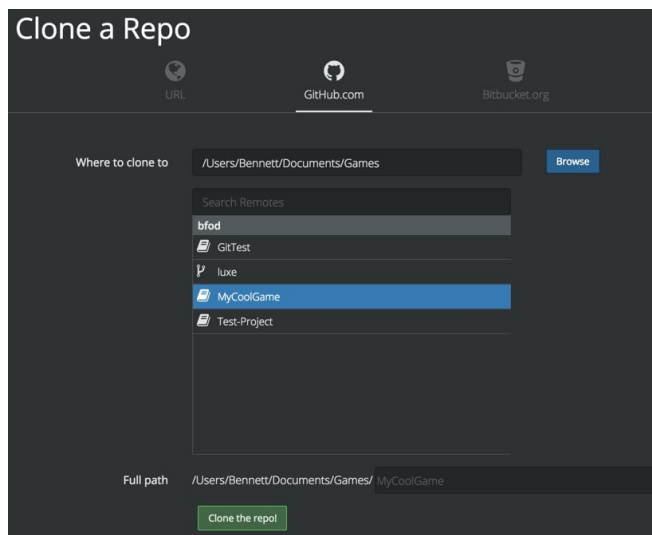
- Download and install GitKraken: <https://www.gitkraken.com/>  
The program is free and the (required) registration is free too.
- Launch GitKraken, it will ask you to create a free profile. You can either do that or sign in with your Github credentials.
- If you're using a GitKraken profile, go to the **Authentication** tab in the Preferences (GitKraken->Preferences on Mac, Edit->Preferences on Windows) and click 'Github'.  
Now click 'Connect to Github'



- Enter your details for your Github account, and it should hook everything up for you.

### 4. Setting up your project to work with GitKraken on the local computer

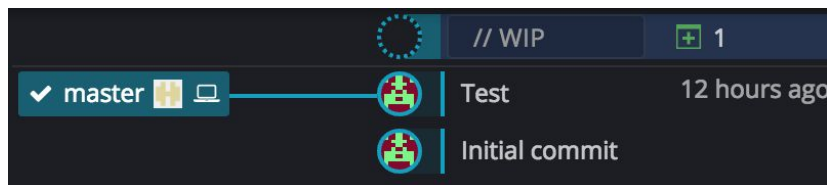
- What we want to do is to **Clone** the remote repo into a local folder on your computer
- In GitKraken, go File->Clone Repo (not INIT repo) and choose the Github.com tab



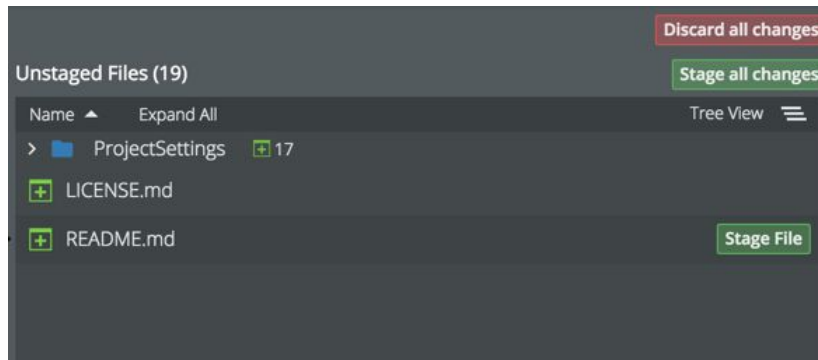
- c. Tell it where to clone to - this should be your local folder where you keep all your various Unity projects.
- d. Select the remote repo we created in step one
- e. Click “Clone the repo”
- f. You now have a local, version-controlled repo folder with a connection to the remote repo. But you need your project to go in there! Right now it’s in another folder.
- g. Go ahead and put all the files from your project folder in the new local repo folder. You’re good to go ahead with commits and pushing. You can delete your old project folder if you like.

*Note: there is another way to do all this, which is to create a local repo in your existing project folder, and then link it to the Github remote. I find that way more painful and error-prone. The other advantage to this method is: it is the same method whether you own the repo or are collaborating with someone else.*

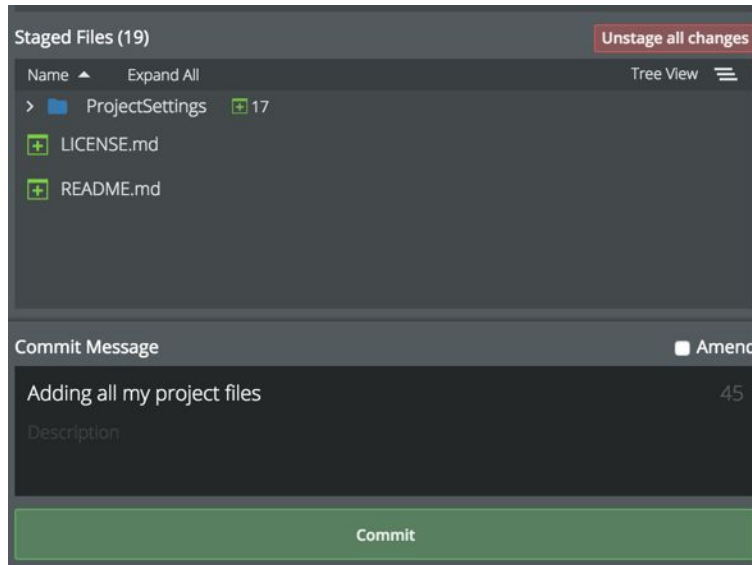
- h. In GitKraken, go View->Reload to make sure it picks up all these new files. Whenever Gitkraken detects changed or added files, it will make a ‘WIP’ area in the list of changes. If it’s not already selected, click it to select it so you can commit the changes:



- i. We want to add everything to the remote repo, so go ‘Stage all changes’. This makes all the files move down into the ‘Staged’ area where we can commit them to the local repo



- j. Type in a commit message **MAKE SURE IT IS A SENSIBLE MESSAGE WITH USEFUL INFORMATION, NEVER TYPE IN AN EMPTY OR NONSENSE COMMIT MESSAGE** and press Commit.

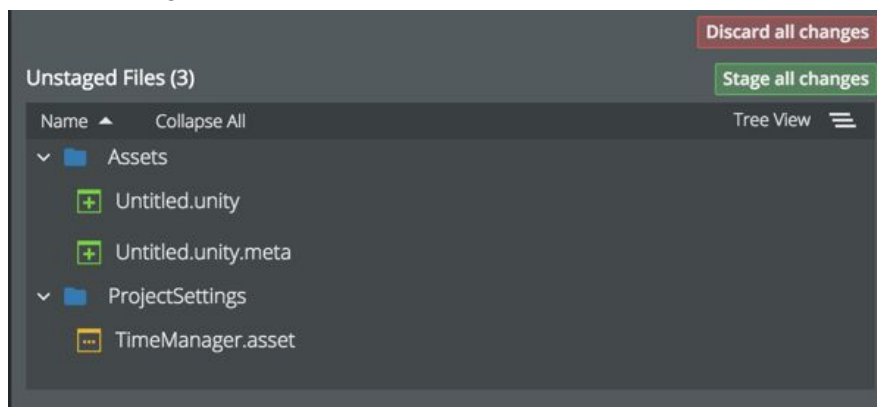


- k. Now that the local repo is up to date, we can press 'Push' to put it up to the remote.



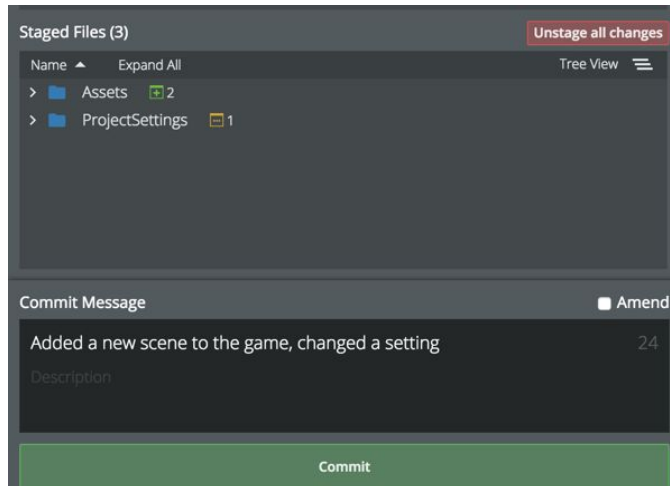
## 5. Working with GitKraken and Unity

- a. Do some work in Unity. Edit files, save your scene, etc.  
b. Modified files appear with orange icons in the 'Unstaged files' area. Newly added files appear with green icons.

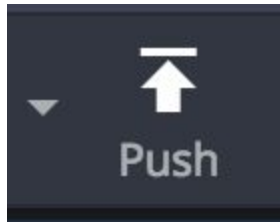


- c. You can either 'stage all', or stage them one by one if you only want to update certain files. Perhaps you've agreed with your coworker that you'll work on the sprites and they'll work on the scene file - in that case don't stage the scene file.  
d. **Make sure you stage any meta files to go with files you're staging** or you won't get their import settings across.

- e. Type in a sensible commit message and commit your changes



- f. Push your changes to the remote repo by pressing Push. Now they're backed up to the cloud!

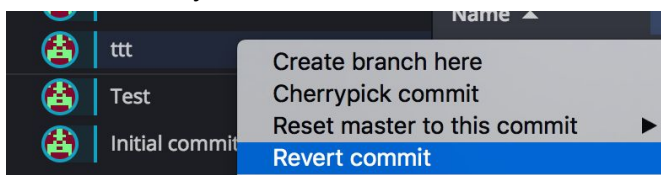


- g. Repeat steps a-f forever, **every time you make a meaningful change that doesn't break the project**. Your teammates will get mad at you if you push broken changes, but if you wait more than a few hours between commits you're setting yourself up for a world of pain.

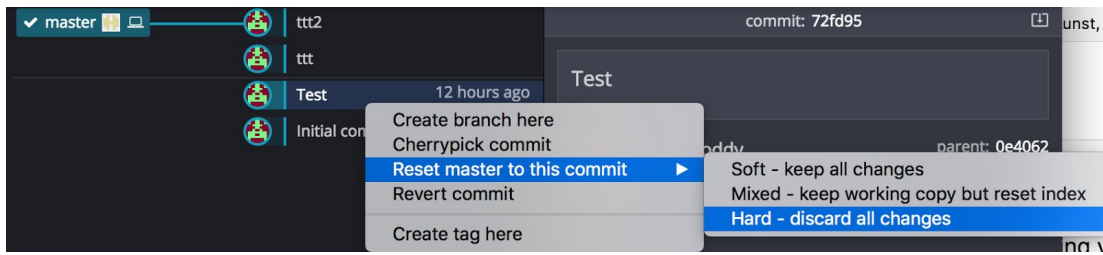
**QUESTION:** What if I screwed up my project and I want to go back to an older version?

**ANSWER: First**, you can go through your old commits and just click on the files in them to look at them, copy or paste the code you want.

**Second**, if you have a particular commit you want to roll back, you can right-click it and revert that one commit. That will work fine so long as it doesn't conflict with any subsequent changes you made (for example, if you modified the reverted files in a subsequent commit, that'll cause a conflict, which you'll have to resolve. That's out of scope for this document.)



**Finally**, if you want to roll everything back to an earlier commit, you can right-click it and choose 'Reset', deciding whether you want to keep all your existing changes (so you can make a new commit) or stomp all your changes with a 'hard reset'.



**WARNING WARNING WARNING:** resetting to an older revision will overwrite your current changes, even if you committed them! You will never be able to get them back. BE CAREFUL

**QUESTION:** What happens if I try to push and it tells me to pull first?

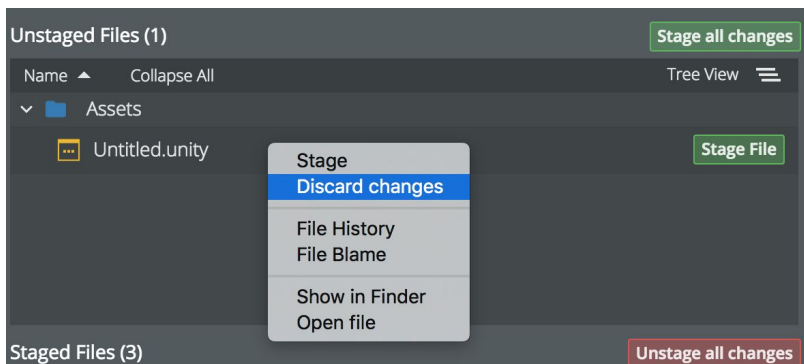
**ANSWER:** That means someone else on your team pushed before you (this won't ever happen

unless you're collaborating). You have to pull first. Press the 'Pull' button.



**QUESTION:** So what if I pull and it turns out my teammate changed a file that I was working on?

**ANSWER:** This is called a **merge conflict**. If it's a text file (like a code file) you can resolve this by choosing which lines of each version to keep. But many game files are binary files, which means you need to just decide whose version you're going to keep. If you want to keep your teammate's version, you right-click your unstaged file and go 'discard changes'.



**QUESTION:** What do all these other buttons do?

**ANSWER:** Most of them break your repo and make your life a total misery