How to use GitHub + Unity + SourceTree, page 1

Match each term with its definition.

- VCS
- Git
- GitHub
- Repository (“repo”)
- Commit
- Push
- Pull
- Git Ignore
- Local
- Remote
- Clone
- Stage

1. _____________________: a free website / server where you can host Git repos

2. _____________________: “Version Control System”... like Git, Subversion, Mercurial, etc.

3. _____________________: this action downloads committed files, from remote to local

4. _____________________: this text file tells Git to ignore other files (you don’t need every file)

5. _____________________: this action uploads committed files, from local to remote

6. _____________________: like a snapshot of your project’s files at a specific point in time

7. _____________________: like a folder that remembers how it changes over time

8. _____________________: this action must be done to a file, before committing the file

9. _____________________: a name for the current computer that you are using

10. _____________________: a name for someone else’s computer / GitHub

11. _____________________: this action downloads an exact copy of the repo

12. _____________________: a type of version control system invented in 2005 to make Linux

How to use GitHub + Unity + SourceTree, page 2

Let's try using SourceTree for managing our Git repo. *If you use other tools like GitKraken or Rider's VCS integration, it's OK, you can use SourceTree with it too.* ***It's very common to use more than one Git program, and it usually won't cause any problems.***

1. Create a Unity project and make sure it has at least one file somewhere in /Assets/

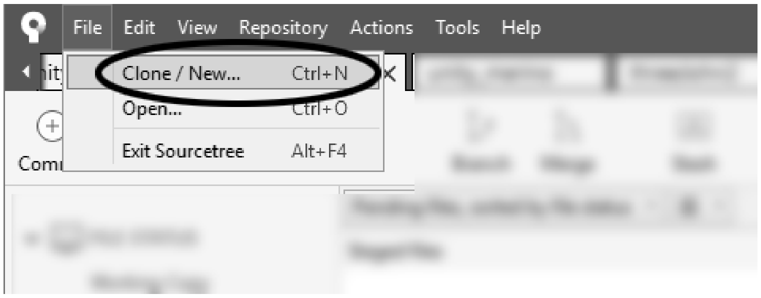
- If you want to try this on an existing project folder, then BACKUP THAT PROJECT FIRST, in case you destroy everything. *(Don't worry, that won't happen though!!)*

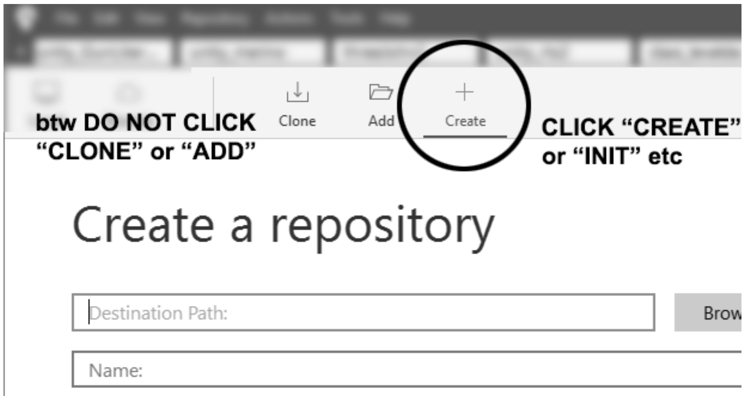
2. Download and install and open SourceTree.

- Go to sourcetreeapp.com and click "Download"
- When you install on your computer:
    - You must register a free Atlassian account. Sorry.
    - When asked, give SourceTree your GitHub account login and password.
        - **MacOS users, THIS IS VERY IMPORTANT: make sure you select "Basic" authentication mode**, don't use the OAuth mode, it's buggy.
- Open SourceTree.

3. In SourceTree, create / initialize a new Git repo inside your Unity project folder.

- In the top menu bar, click File > Create / New, and then click the "Create" (or "Init") button.
- "Destination Path" = your Unity project folder. This is the folder that has /Assets/ and /ProjectSettings/, etc.
- "Name" = whatever you want to call it
- Leave repo type as "Git"
- **Make sure you disable "Create Repository On Account**", and most of the time it doesn't even work anyway.





btw DO NOT CLICK "CLONE" or "ADD"

CLICK "CREATE" or "INIT" etc

Also: Keep in mind that Windows and MacOS versions look different. Sorry.

How to use GitHub + Unity + SourceTree, page 3

4. Setup the .gitignore file (VERY IMPORTANT)

- The most important things to ignore are /Library/ folder and /Temp/ folder.
- However, it's easiest to just copy and paste a pre-made .gitignore file.
- Go to https://github.com/NYUGameCenter/Unity-Git-Config/blob/master/.gitignore in your browser, and copy and paste the text file there.
- Then, to access your gitignore file from SourceTree:
    - In the SourceTree menu bar at the top, go to Repository > Repository Settings
    - Click the "Advanced" tab
    - Near the top, you will see "repository-specific ignore list"... click "Edit" button.
        - You can edit the .gitignore in Notepad (Windows) or TextEdit (MacOS)
    - Paste the contents from the URL above.
    - Save the file and close it.

5. Setup your GitHub repo. You need to setup your git repo's twin on GitHub, so you can upload your files to it.

- Open GitHub.com in your browser and login
- Click the "+" button in the top right of the page, and "Create a New Repository"
- Type in a good descriptive name, ignore their bad twee name suggestions
- IMPORTANT: **DO NOT INITIALIZE THE REPO WITH ANY FILES**
    - DO NOT ADD A GITIGNORE
    - DO NOT ADD A README
    - DO NOT PASS GO, DO NOT COLLECT $200
    - In fact, after you type in a name, DON'T EDIT ANYTHING ELSE, just click the big green "Create" button at the bottom.
    - (Why? We already created / initialized a repo in step 3.)
- After you create the repo, it should be empty.
- **Copy the "clone URL" in the webpage,** it should start with "https://" and end with ".git"

6. Connect your local repo to your remote repo in SourceTree.

- In the SourceTree menu bar at the top, go to Repository > Repository Settings
- Click the "Add" button to add a remote
    - "Remote name" = traditionally, we use the name "origin"
    - "URL/Path" = paste the URL you copied from GitHub.com (see step 5)

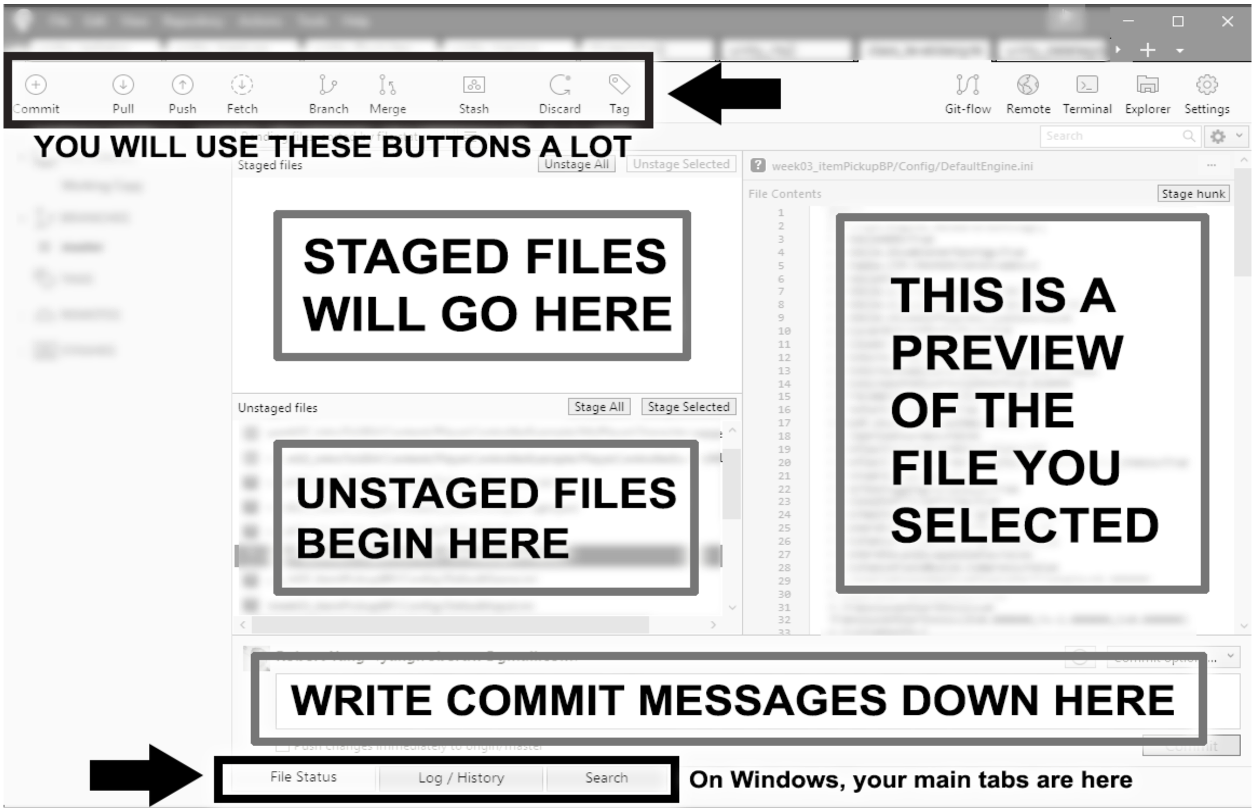
7. OK, you finished setting up your local and remote repo!

Good job. Now you need to make a commit, and push that commit to GitHub. (next page)

How to use GitHub + Unity + SourceTree, page 4

Let's practice your typical daily workflow:

- **click "Commit" button in top-left** to open the File Status commit screen
- **<u>stage files</u>;** select unstaged files and click "Stage Selected" (or click "Stage All")
- **write descriptive <u>commit message</u>** at bottom,
- **click <u>"Commit"</u> button in bottom-right** to commit staged files
- **click <u>"Push"</u> button at the top** to upload committed files to GitHub



YOU WILL USE THESE BUTTONS A LOT

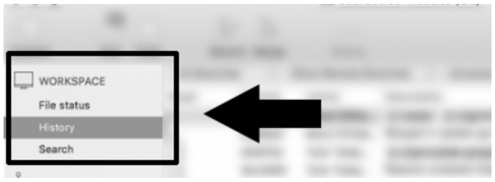
STAGED FILES WILL GO HERE

THIS IS A PREVIEW OF THE FILE YOU SELECTED

UNSTAGED FILES BEGIN HERE

WRITE COMMIT MESSAGES DOWN HERE

On Windows, your main tabs are here



NOTE: on MacOS, your main view tabs are at the top of the left sidebar, under "Workspace"

+ IMPORTANT WARNING ABOUT STAGING / UNSTAGING:

- If you want to unstage a file, then right-click the staged file and select "Unstage"
- **DO NOT select "Remove"**... this means "remove this file from tracking", and if the file isn't tracked, then that actually means **"remove this file from existence"**