

# HOW TO PUT A UNITY PROJECT ON GITHUB, WITH SOURCETREE

by Robert Yang!!!

## 1. UNITY EDITOR:

a) Create a Unity project, and save an empty scene in the Assets folder

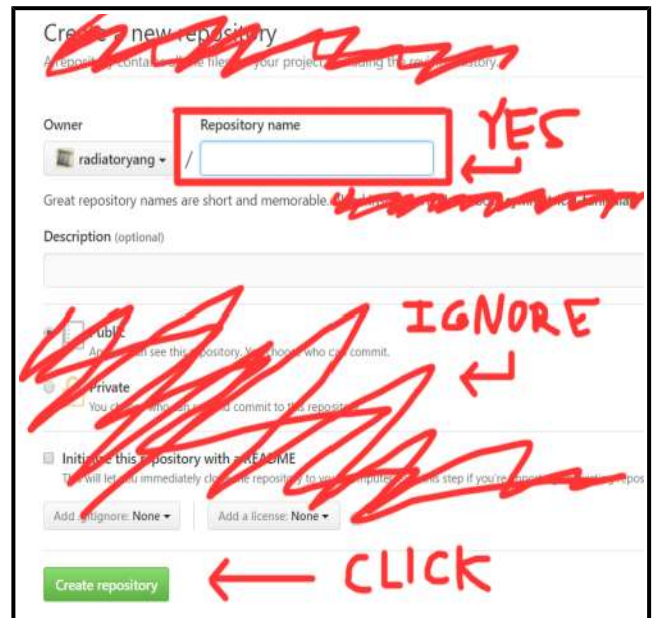
- Git only cares about folders if it has at least 1 file in it

## 2. GITHUB.COM (IN YOUR WEB BROWSER):

a) go to GitHub, create a New Repo (click + icon in the upper-right)

- give your repo a useful name, like “GameProject” or something, don’t use GitHub’s terrible stupid suggestions

- on the New Repo page, **\*ONLY\* type in a repo name and LEAVE THE REST AS DEFAULT** then at the bottom of the page, click “Create” button





**b) after you create it, copy the “Clone URL” to clipboard**  
 the clone URL will be in a textbox on the webpage, starts with HTTPS and ends with .GIT like: https://github.com/YourName/RepoName.git

### 3. SOURCETREE:

**a) Download and install SourceTree**

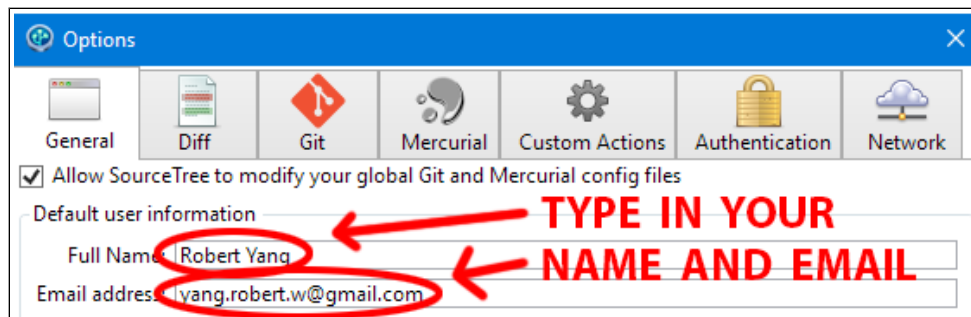
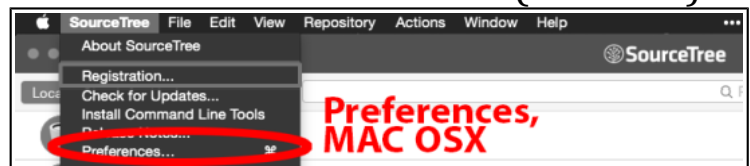
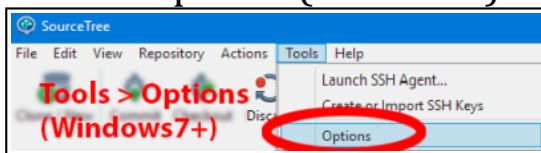
<https://www.sourcetreeapp.com/>

*the program is free and registration is free*

- if asked, say YES to installing Git
- if asked, say NO to installing a global .gitignore
- if asked, say NO to setting up an SSH key
- if asked, give it your GitHub login and password

**b) add your “Default user information” in options**

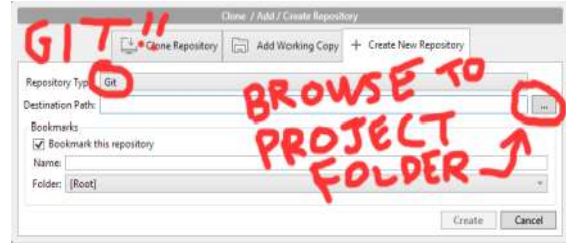
Tools > Options (Windows) or SourceTree > Preferences (Mac OSX)



c) In the menu bar, go to **File > Clone / New Repo**, or something like that, they change it all the time

d) select **“Create New Repo” / “Create Local Repo”**

- make sure **Repository Type** is set to **“Git”**
- Destination Path = browse to your Unity project folder (folder that has Assets, ProjectSettings, etc.)
- click Create

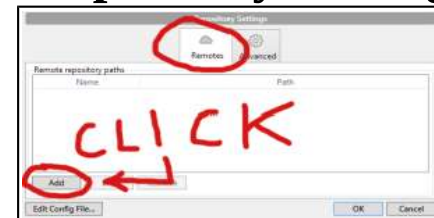


e) if SourceTree doesn't pop-up a new window already, then double-click on the new repo to open it

## LINKING YOUR PROJECT TO GITHUB:

f) in the menu bar, go to **Repository > Repository Settings**

- in Remotes tab, click ADD button to ADD A REMOTE



- name: **“origin”**  
(on Windows, just tick **“default remote”** checkbox)
- URL Path: paste the **“https://...”** URL from GitHub (step 2b)



- SourceTree will finish the bottom half automatically

## **TELLING GIT TO IGNORE BAD FILES:**

**g) in Repository > Repository Settings, go to “Advanced”**

**h) in Repository-Specific Git Ignore List click “EDIT” to open the .gitignore in Notepad or TextEdit or something**

**i) google “Unity gitignore”, click the first link, copy and paste the main text into your file, and save the file**

# THE GIT IGNORE TEXT WILL LOOK LIKE THIS:

```
/[Ll]ibrary/  
/[Tt]emp/  
/[Oo]bj/  
/[Bb]uild/  
/[Bb]uilds/  
/Assets/AssetStoreTools*
```

# Autogenerated VS/MD/Consulo solution and project files

```
ExportedObj/  
.consulo/  
*.csproj  
*.unityproj  
*.sln  
*.suo  
*.tmp  
*.user  
*.userprefs  
*.pidb  
*.booproj  
*.svd
```

# Unity3D generated meta files

```
*.pidb.meta
```

# Unity3D Generated File On Crash Reports

```
sysinfo.txt
```

# Builds

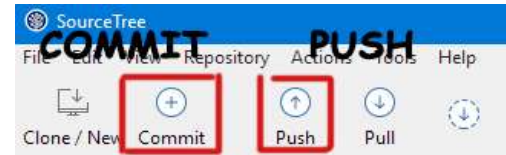
```
*.apk  
*.unitypackage
```

**YOU ARE NOW DONE SETTING UP! GOOD JOB!**  
**NOW IT'S TIME TO TRY TO USE THIS... (turn page)**

# UNITY + SOURCETREE + GITHUB EVERY DAY:

4. **Do some work in Unity.** Edit files, save your scene, etc.

5. **Open your repo in SourceTree.**  
Click “Commit” near the top.



“commit” = a snapshot, in time, of all the files in your project

“stage” = staged files will be in the commit

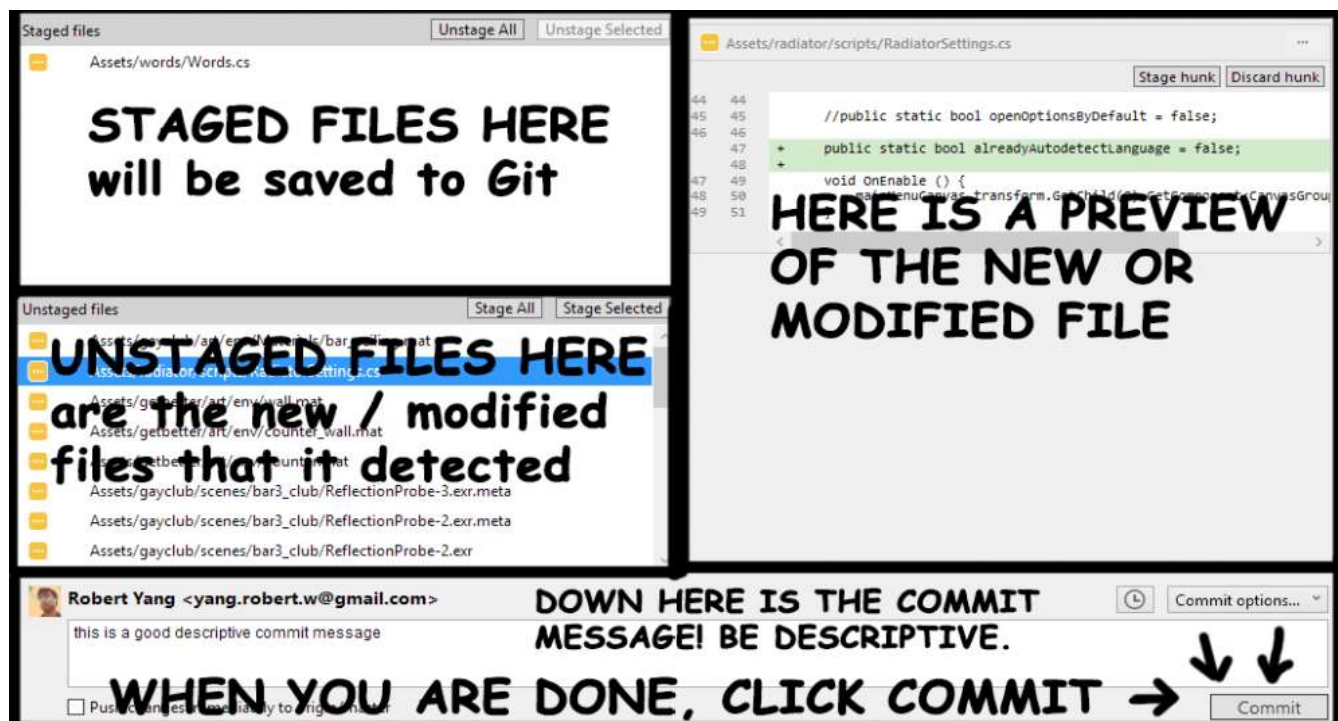
6. **Stage the files you want to remember / save / backup.**

- use [SHIFT+LEFT CLICK] to select multiple files, then click “Stage Selected”... or click “Stage All”

- to unstage a file, \*\*\* **DO NOT CLICK “REMOVE”** \*\*\* THAT MAY **“REMOVE THE FILE FROM EXISTENCE”**... instead, select the staged file then go to “Actions > Unstage From Index”

7. **Fill-in a Commit Message, and finish the commit.**

good descriptive commit message about what's in this commit (“new menu icons”, “fixed bug #402”)



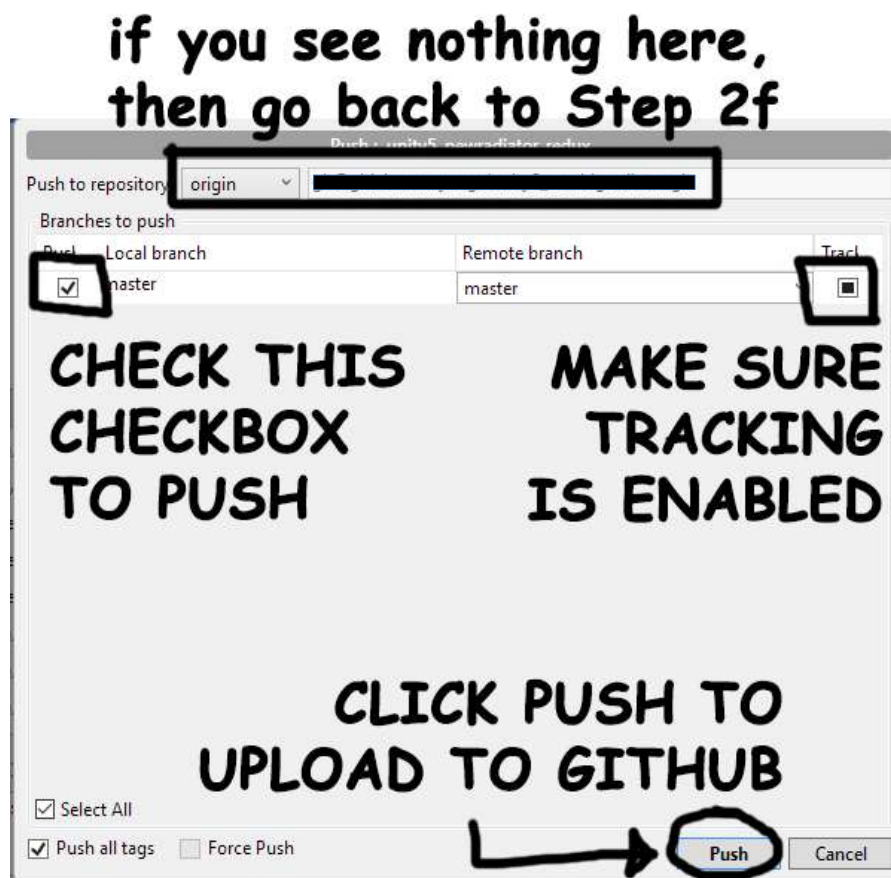
**8. PUSH your commits to GitHub.** Click “Push” near top.

*“pushing” = uploading to the remote*

*“remote” = GitHub’s server*

*“local” = your laptop*

**9. In the Push dialog,** (a) check the “Push” checkbox and (b) enable tracking. Then push.



**LAST STEP: MAKE SURE IT WORKED?**

**10. Open your GitHub repo page in a browser,** verify that it worked. You should see all your files on GitHub now.

**11. Repeat Steps 4-10 every day, forever, until you die**