Lightbulb's Tutorials / Tips / Tricks

A collection of tutorials, tips, and tricks.

- Putting unity projects in Github for EZ sharing & backups

Putting unity projects in Github for EZ sharing & backups

1. Putting Unity projects in Github

Github is a great way to freely revision control your projects (i.e. if you ever break stuff, you can just revert back to an old version of the project), as well as share with other people for collaborative work.

2. What is Github/git?

2.1 Git

Git is free and open source software for distributed version control: tracking changes in any set of files, usually used for coordinating work among
programmers collaboratively developing source code during software development. However by default it's just a command line program that is not very friendly to use, and you'd need to provide your own storage for it.

2.2 Github

Github is a free to use service & website that provides free (and paid) hosting services for git projects, as well as a friendly web interface (github.com) among other things. Github does have limitations though, primarily:

1. No project can have a total size of greater than 100GB
2. No individual file can be greater than 100MB
3. No single push/commit can be greater than 2GB (just have to break up very large commits into smaller chunks)

However these are pretty generous, and albeit mildly painful to setup at first for an existing project - the benefits far outweigh the one time setup IMO. If you have a single file that exceeds 100MB, you can just add that file and/or directory into the `.gitignore` file.

3. How To

Prerequisites

- Create an account on Github.com
- Download a git client. These instructions will cover TortoiseGit for Windows, but any git client will work
  - Download from https://tortoisegit.org/
  - Keep default options for everything
  - At somepoint it will ask you to locate the 'git.exe in PATH' (or something very similar) and it will be empty (unless you already have git installed). There is a link to download git for windows. Download that and install it, once done that field should be populated in the TortoiseGit installer.
  - Finish the installation

Upload an existing project to Github (First time setup)

1. Create a repository on github.com
2. Clone that empty repository to your computer (right click in a folder, and copy the GIT url from github into the TortoiseGit "Clone repository" option
3. Copy all your project files into that folder. (Do NOT include a .git folder if copying from an old git project, but DO include the `.gitignore` file (listed below if you don't have). do NOT include the Assets folder at first, unless its <2GB)
4. Using TortoiseGit and right click on the repository parent folder and, ADD the files, then COMMIT, then PUSH

If your Assets folder was larger than 2GB, then:
5. make the Assets folder in your new project/repository
6. copy LESS than 2GB worth of stuff from inside the original Assets folder into the Assets folder in your new github repository/folder
7. repeat 4-7 until the entire project is pushed to github

Push changes to Github (This is done on a regular basis)

1. Using tortoise git, ADD the files, then COMMIT, then PUSH. Just make sure you don't push more than 2GB at one time (which is very rare for regular development) or any single file that's over 100MB

Want to share your project? You can add collaborators on github.com easily - now multiple people can work on a project.

4. Notes & FAQ's

I get errors on uploads, what do?

Broke something and can't upload? I've primarily only ever had two types of errors: Timeouts/Slow network **OR** you'll get an obvious error message in the console saying you uploaded a file too large or the push was too large.

File/Commit/Push was too large:

My simple low-tech instructions for fixing it:
1. Redownload your project from github into a new folder
2. Transfer all your files _except_ the .git folder to the new project
3. Remove any files you just added that are >100MB and/or reduce the size of the commit to <2GB (or whatever other issue there is)
4. Try uploading again

Timeout/network error/ remote end hung up

Although I don't have those errors on hand, they were all fixed by googling them. Typically involves running a command or two on the git command line to increase timeouts and/or change which version of HTTP its using.

5. .gitignore

The '.gitignore' file is very important. In any project before uploading your files for the first time, make sure this file is present, has the following contents, its named exactly ".gitignore" and is placed in the root directory of your project. You may need to "Show hidden files" in Windows explorer to view files that start with a "."

Note: This mainly ignores your gigantic Library folder to save a lot of space and speed up uploads. Only downside is on a fresh download of a project from git, it will take a little time to rebuild this folder.

```
# This .gitignore file should be placed at the root of your Unity project directory
# Get latest from https://github.com/github/gitignore/blob/master/Unity.gitignore
/[Ll]ibrary/
/[Tt]emp/
/[Oo]bj/
/[Bb]uild/
/[Bb]uilds/
/[Ll]ogs/
/[Uu]ser[Ss]ettings/
# MemoryCaptures can get excessive in size.
# They also could contain extremely sensitive data
/[Mm]emoryCaptures/
# Asset meta data should only be ignored when the corresponding asset is also ignored
!/[Aa]ssets/**/*.meta
# Uncomment this line if you wish to ignore the asset store tools plugin
# /[Aa]ssets/AssetStoreTools*
# Autogenerated Jetbrains Rider plugin
/[Aa]ssets/Plugins/Editor/JetBrains*
# Visual Studio cache directory
.vs/
# Gradle cache directory
.gradle/
```

```
# Autogenerated VS/MD/Consulo solution and project files
ExportedObj/
.consulo/
*.csproj
*.unityproj
*.sln
*.suo
*.tmp
*.user
*.userprefs
*.pidb
*.booproj
*.svd
*.pdb
*.mdb
*.opendb
*.VC.db
# Unity3D generated meta files
*.pidb.meta
*.pdb.meta
*.mdb.meta
# Unity3D generated file on crash reports
sysinfo.txt
# Builds
*.apk
*.aab
*.unitypackage
# Crashlytics generated file
crashlytics-build.properties
# Packed Addressables
/[Aa]ssets/[Aa]ddressable[Aa]ssets[Dd]ata/*/*.bin*
# Temporary auto-generated Android Assets
/[Aa]ssets/[Ss]treamingAssets/aa.meta
/[Aa]ssets/[Ss]treamingAssets/aa/*
Assets/UdonSharp/UdonSharpSettings.asset
/[Aa]ssets/Editor/x64/Bakery/
/[Aa]ssets/PIANO\ HOUSE\ ARCHVIZ/HOUSE/LightingData.asset
/[Aa]ssets/PIANO\ HOUSE\ ARCHVIZ/Models/Common\
Textures/Wood_wall/boards_MetallicSmoothness.tga
```