Basic Github guide

1. What is Github?

- Github is what makes using Git easier for version control stuffs. (You can Google yourself if you want to know more about Git).

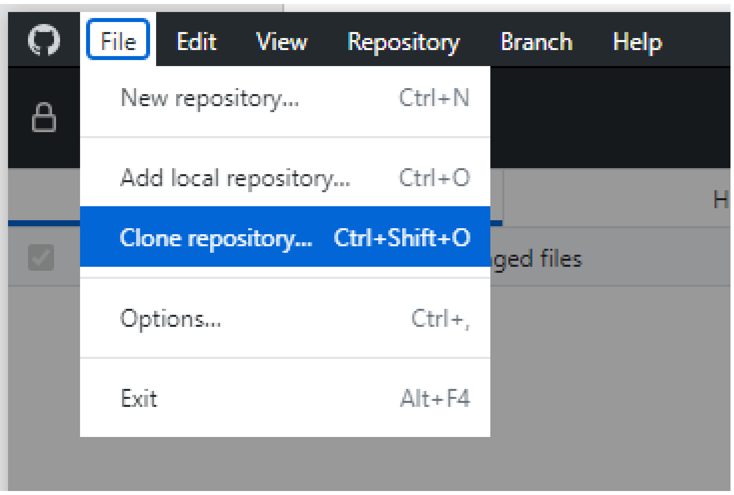
- Basically, Github store the most updated version of our project (known as repository). And we can clone the project to your computer to access it locally. If you then make changes to your local project, you can commit your changes through Github, and when everyone can get a hold of your changes on their local project.

2. Basic uses of Github.

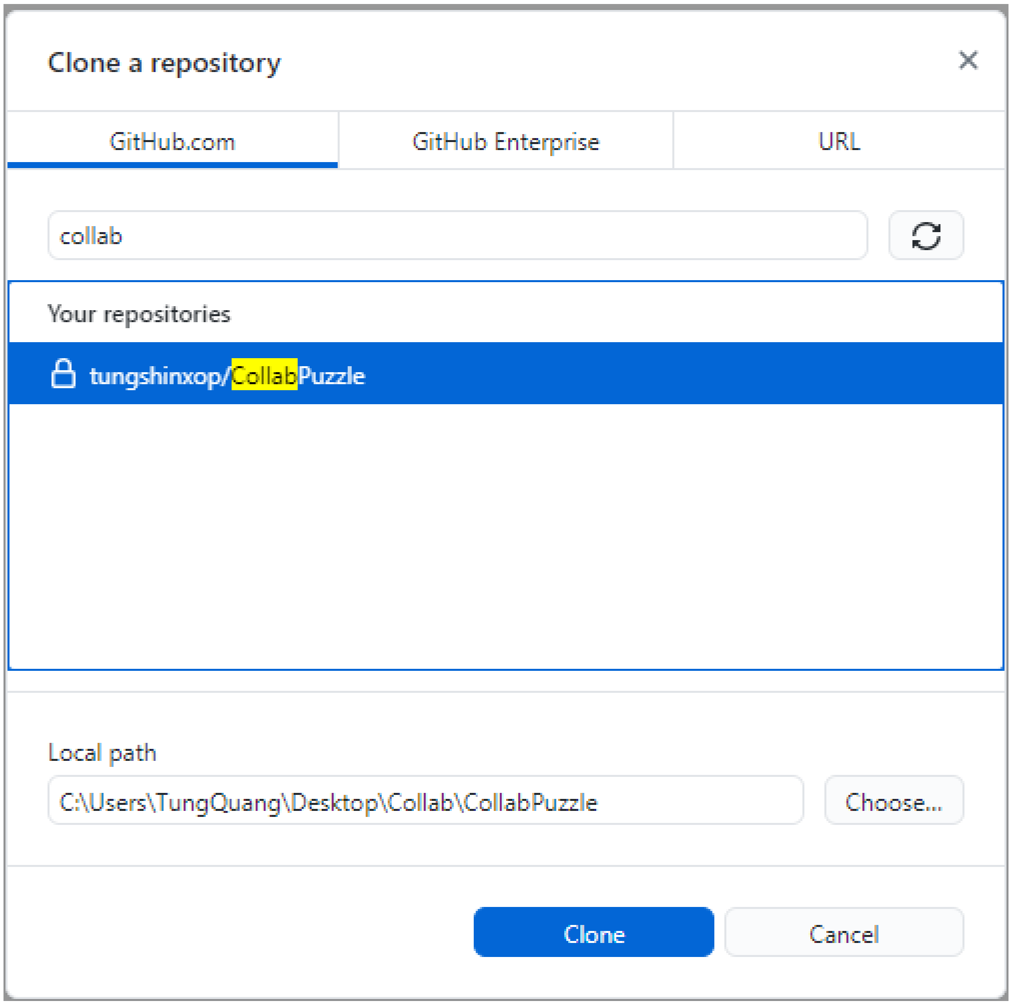
2.1. Clone project to your computer.

- **First step:** Install Github Desktop.

- **Second step:** Go to File -> Clone repository. (File is on the top left corner of the Github Desktop window).



- **Third step:** Choose CollabPuzzle repository and choose where to store it.



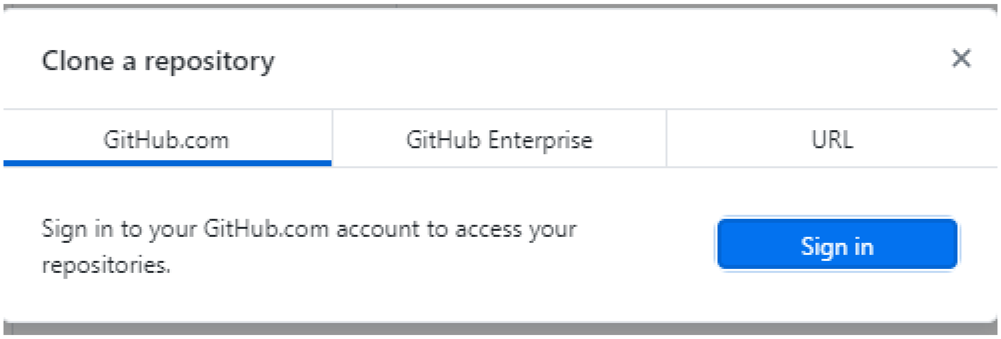
collab

tungshinxop/CollabPuzzle

Local path

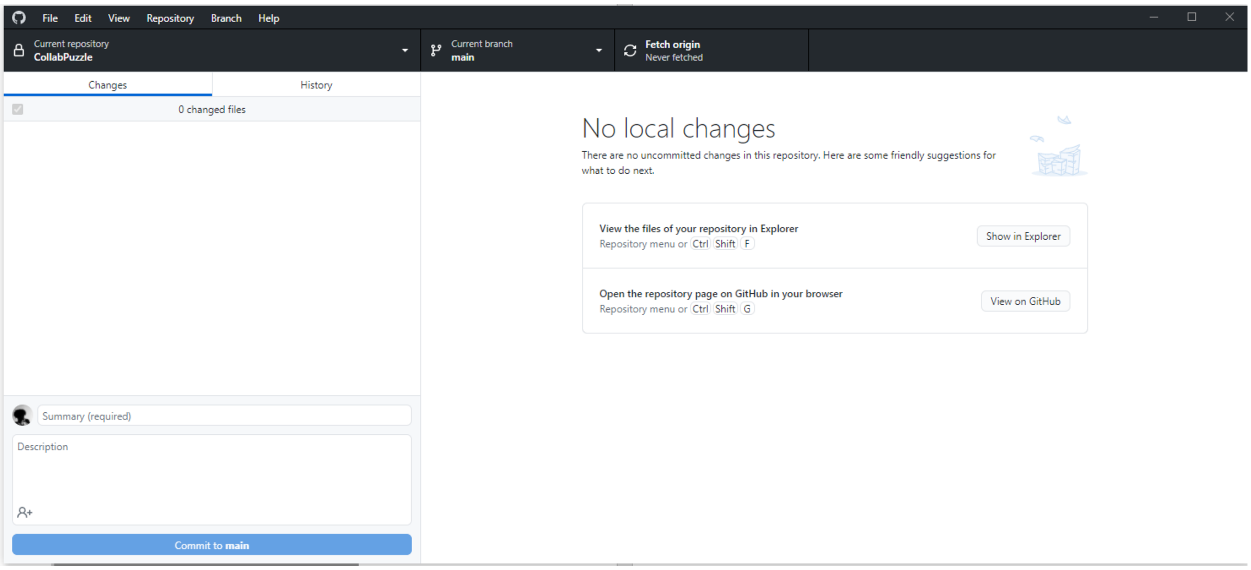
C:\Users\TungQuang\Desktop\Collab\CollabPuzzle

*NOTE: If you are not sign in (your tab does not look like above but below), just press the "Sign in" button to sign in and repeat this step.



Sign in to your GitHub.com account to access your repositories.

- **Fourth step:** Press the blue "Clone" button and we are finished. The folder for project would be stored in the path that you use for the third step.



Current repository
CollabPuzzle

Current branch
main

Fetch origin
Never fetched

0 changed files

No local changes

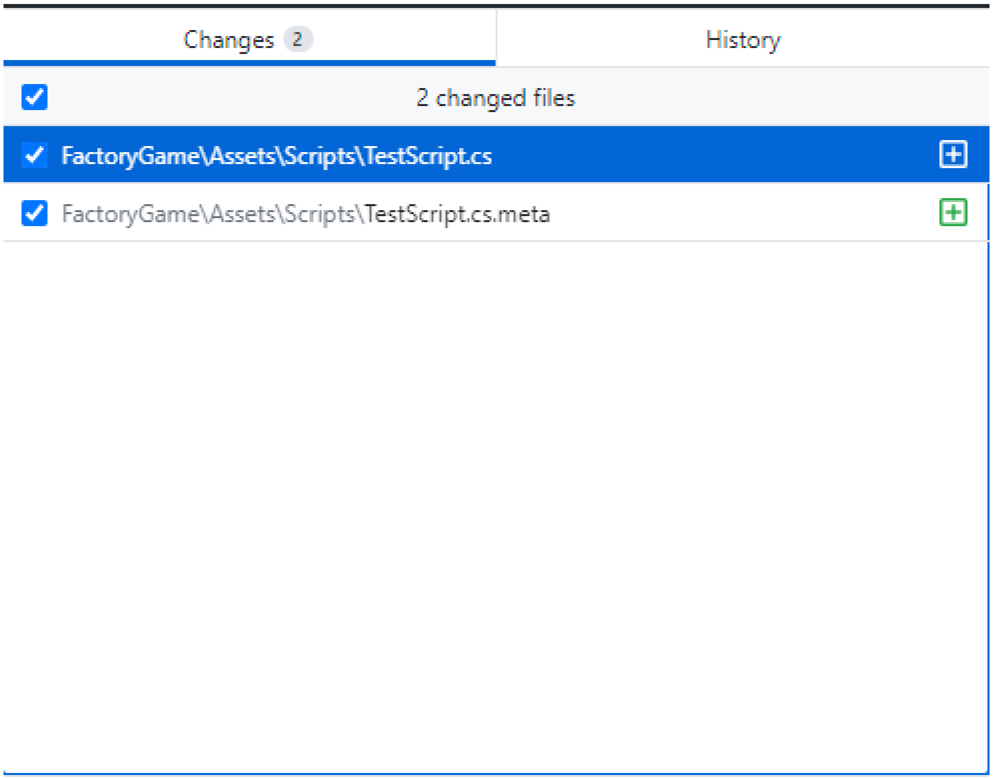
There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

View the files of your repository in Explorer
Repository menu or Ctrl Shift F

Open the repository page on GitHub in your browser
Repository menu or Ctrl Shift G

2.2. Making changes.

- Any changes that you made to the project will be shown in the Changes section in Github Desktop.

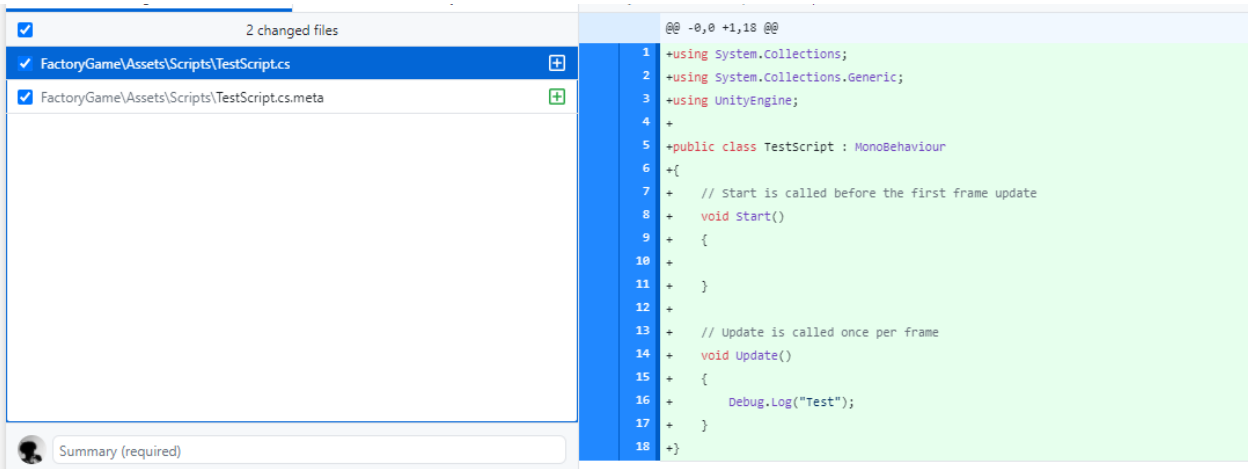


☑ 2 changed files

☑ FactoryGame\Assets\Scripts\TestScript.cs

☑ FactoryGame\Assets\Scripts\TestScript.cs.meta

- If the changes are script (with the .cs), you can easily select any of the change and see what have been modified.



☑ 2 changed files

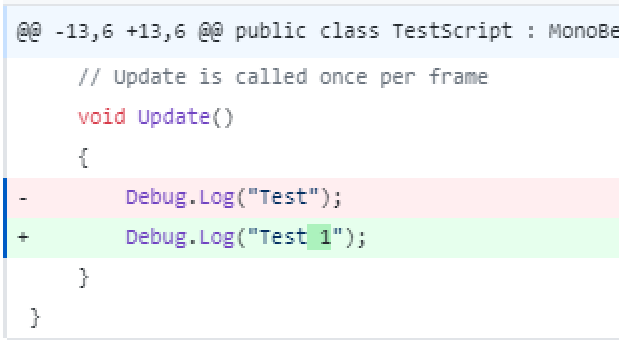
☑ FactoryGame\Assets\Scripts\TestScript.cs

☑ FactoryGame\Assets\Scripts\TestScript.cs.meta

Summary (required)

```
@@ -0,0 +1,18 @@
+using System.Collections;
+using System.Collections.Generic;
+using UnityEngine;
+
+public class TestScript : MonoBehaviour
+{
+    // Start is called before the first frame update
+    void Start()
+    {
+
+    }
+
+    // Update is called once per frame
+    void Update()
+    {
+        Debug.Log("Test");
+    }
+}
```

*NOTE: Red is removed, Green is added.



```
@@ -13,6 +13,6 @@ public class TestScript : MonoBe
     // Update is called once per frame
     void Update()
     {
-        Debug.Log("Test");
+        Debug.Log("Test 1");
     }
 }
```

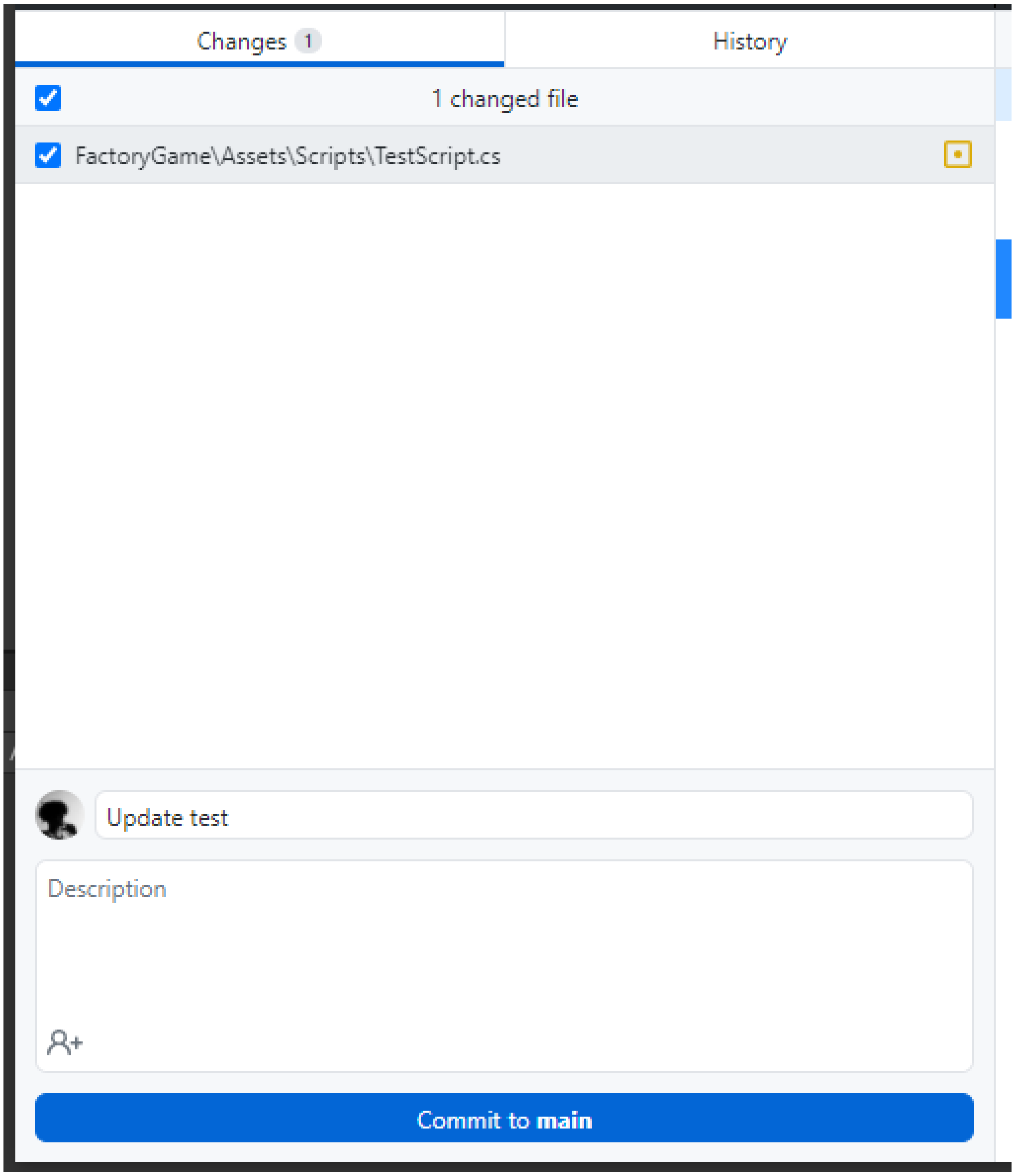
2.3. Pushing changes.

- Pushing changes is referred to the action of updating your changes to repository and everyone can pull your changes.

- Remember, it is wiser to push very often, pushing small chunks of work to avoid any problems that may result in loss of work.

- How to push:

    - **First step:** Comment your summary (mandatory) and description (optional) of your changes.

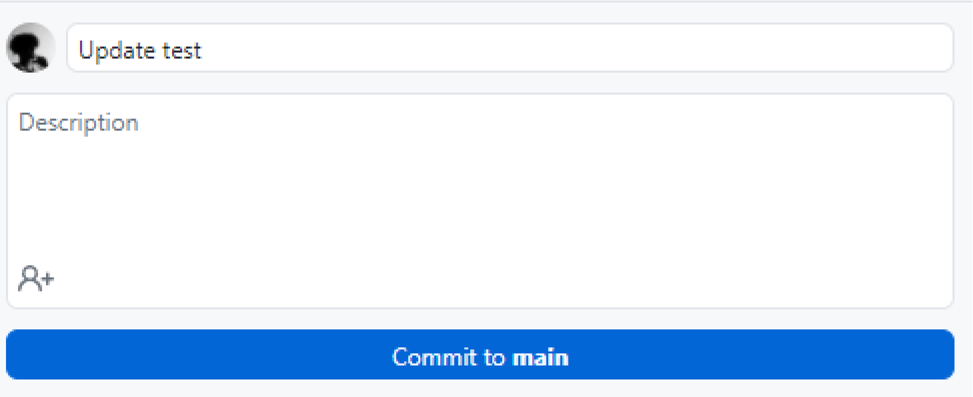


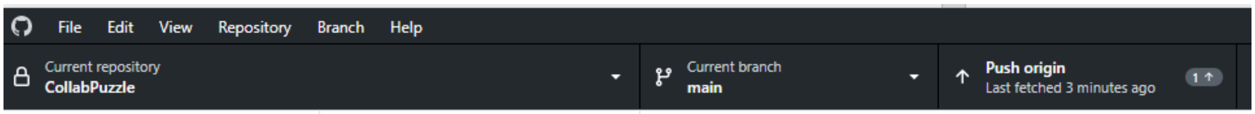
☑ FactoryGame\Assets\Scripts\TestScript.cs

Update test

- **Second step:** REMEMBER TO PULL BEFORE PUSHING TO AVOID MERGE CONFLICTS. See how to pull in section 2.4

- **Third step:** Press “Commit to main” and the press push origin and now your changes are live.


Update test



Current repository CollabPuzzle

Current branch main

Push origin
Last fetched 3 minutes ago

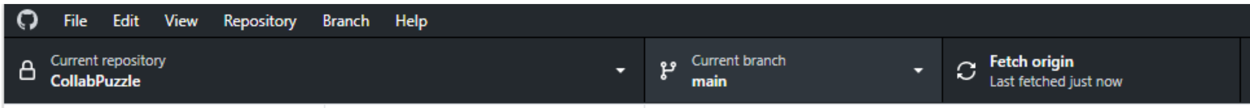
2.4. Pulling changes

- Pulling changes is referred to the action of “installing” other members’ changes of the project to your own local project.

- Remember, it is wiser to start pulling when start to work on a new stuff (e.g. script or level), or at least, **REMEMBER TO PULL BEFORE PUSH.**

- How to pull:

  - **First step:** Press Fetch Origin.

    Current repository CollabPuzzle

    Current branch main

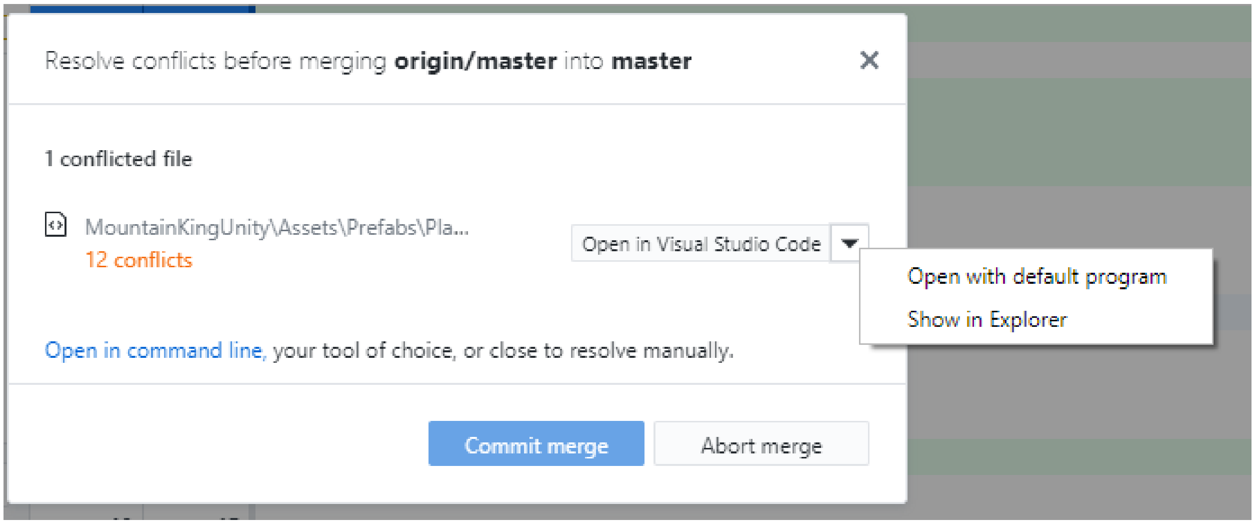
    Fetch origin
    Last fetched just now

  - **Second step:** If there are any changes that is not present in your local project, the button will be changed into Pull Origin, you can then proceed to press that button to perform pulling. (If the button does not change to Pull Origin => no update, no need to pull).

3. Merge conflicts and what to do when encounter merge conflicts.

- Merge conflicts happen when two changes are present in one version of the project. Maybe two people are updating one script at a time.

- Merge conflicts mean that Github does not know which changes to use.



Resolve conflicts before merging **origin/master** into **master**

1 conflicted file

MountainKingUnity\Assets\Prefabs\Pla...

12 conflicts

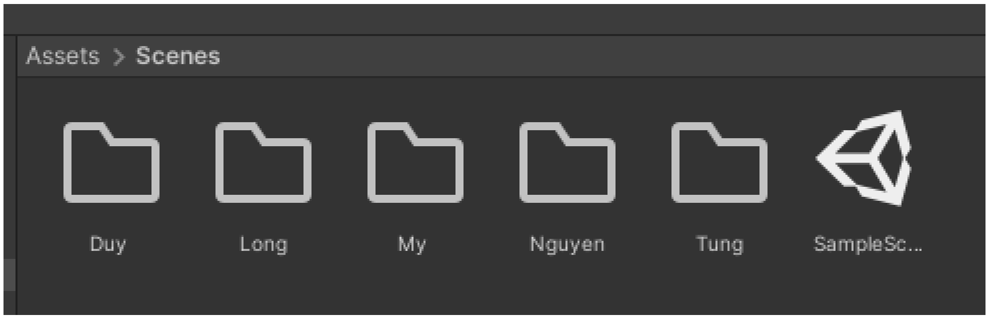
Open in command line, your tool of choice, or close to resolve manually.

- So, if you happen to encounter merge conflicts, notify your Seniors and we will figure the solutions out, don't be panic.

4. Apply

- What you should be doing now is to clone the project and create your own scene (e.g named Tung01) and push your changes.

- Your scene should be put inside the folder with your name.



Assets > Scenes
Duy
Long
My
Nguyen
Tung
SampleSc...