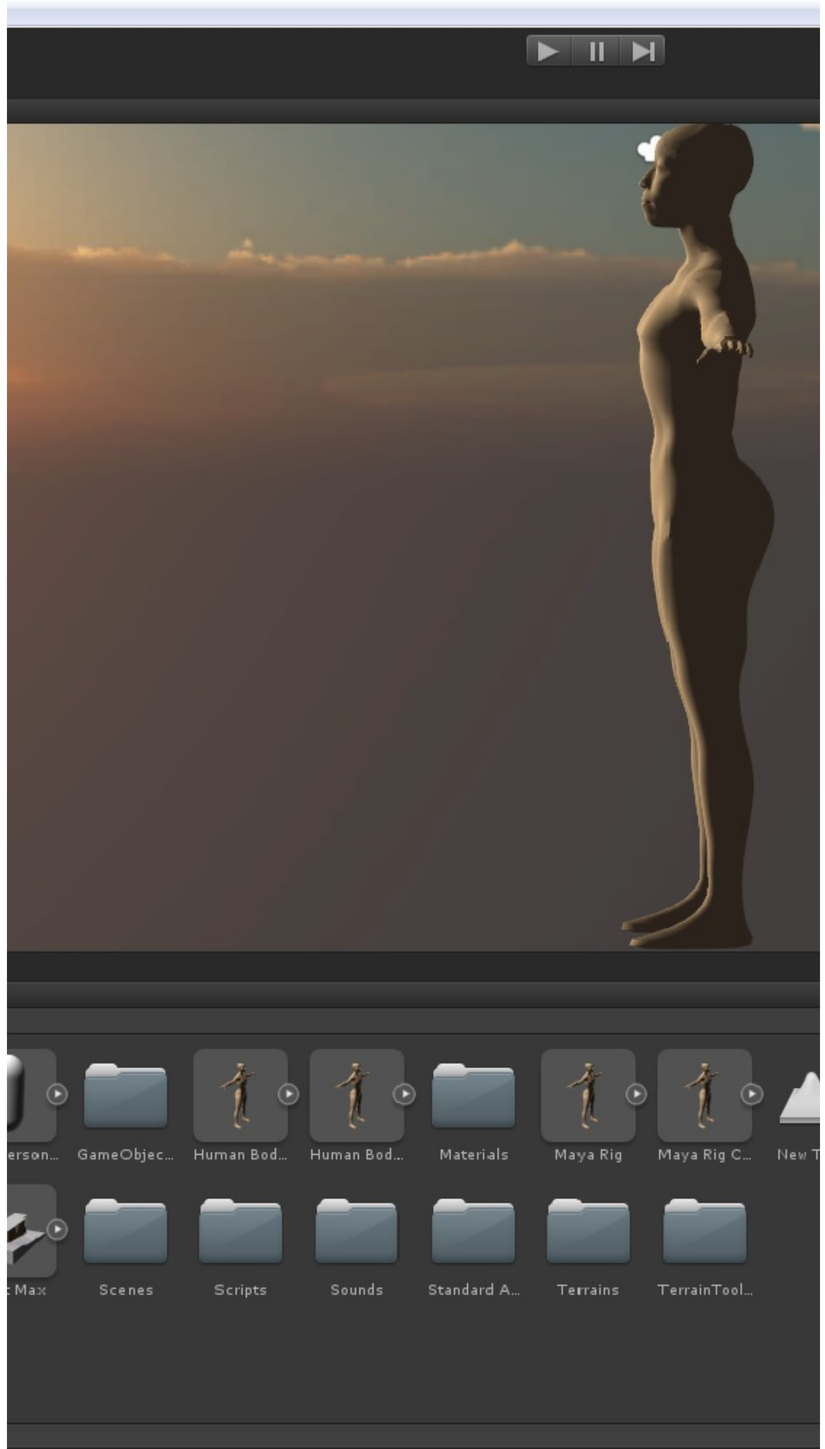
# UNITY 5.X
# &
# GIT

*How to Git and workflows*



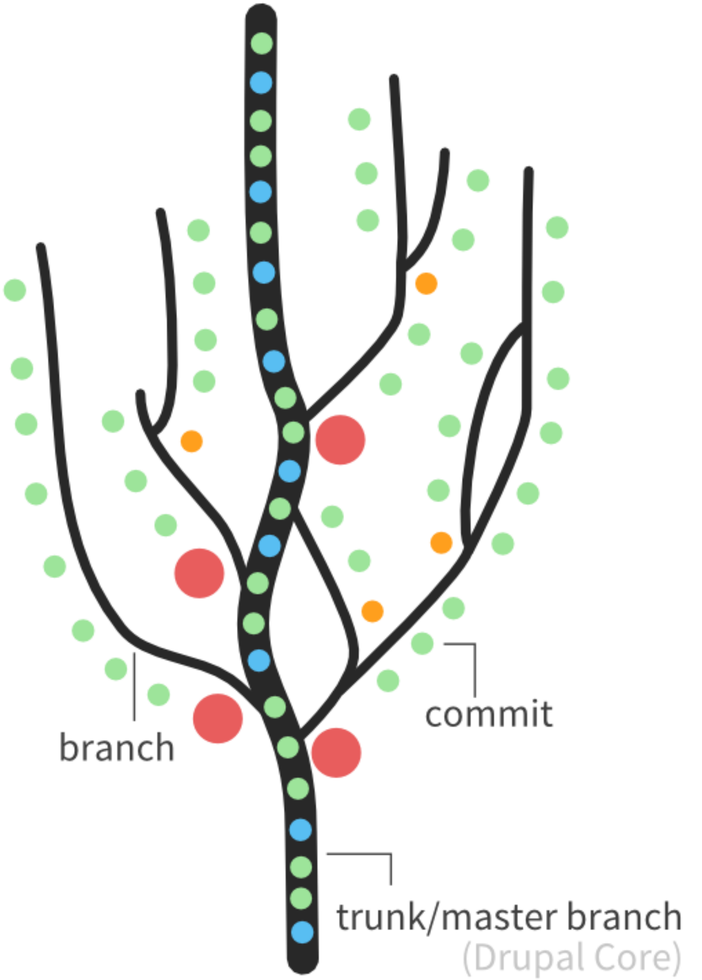

[studica.com]

# WHAT IS GIT

- A system to control versions of a project
- Enables multiple people to work on files without affecting each other
- Does this by using branches. Branches are independent of each other
- Pull-requests for review and approval before implementation


# WHAT IS A BRANCH

- Each branch in a repository represents an independent development path
- Most other Git functions operate using the branch in your working copy
- Branches are used for any change that you want to be implemented into the game. Whether it is a major feature or a single bug fix, every change requires a different branch that most pass through a review process before it is fully implemented.



# ON YOUR COMPUTER

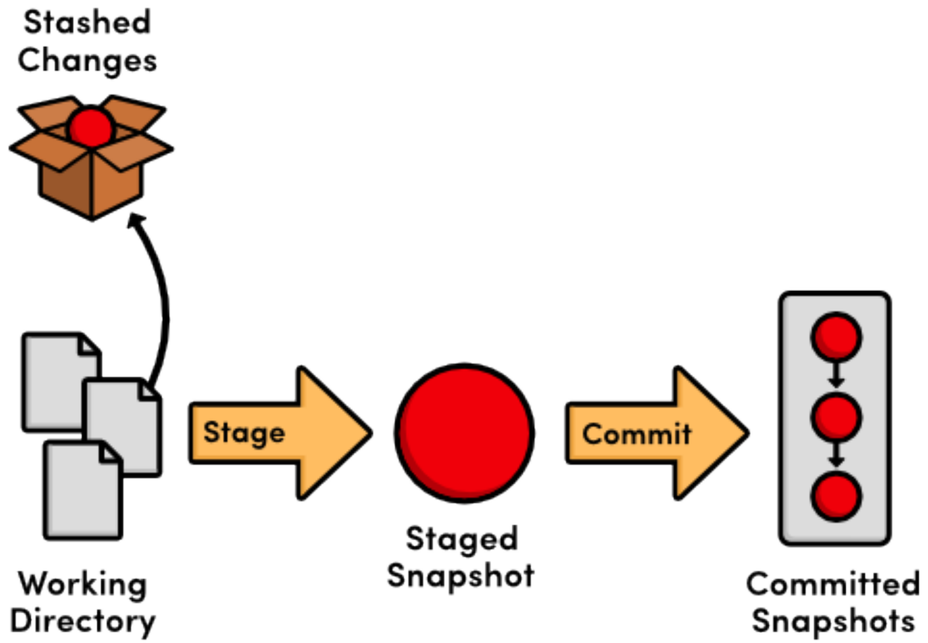
- Working copy is the repository directory on your computer and all its contents
- Git switches between branches by deleting your working copy and replacing it with the selected branch
- If you have uncommitted changes Git will not allow you to switch branches (stashes are a work around)
- Tracked files are files that are currently being used on the remote. Files can be on your computer that aren't actually in the repository

# GIT COMMANDS

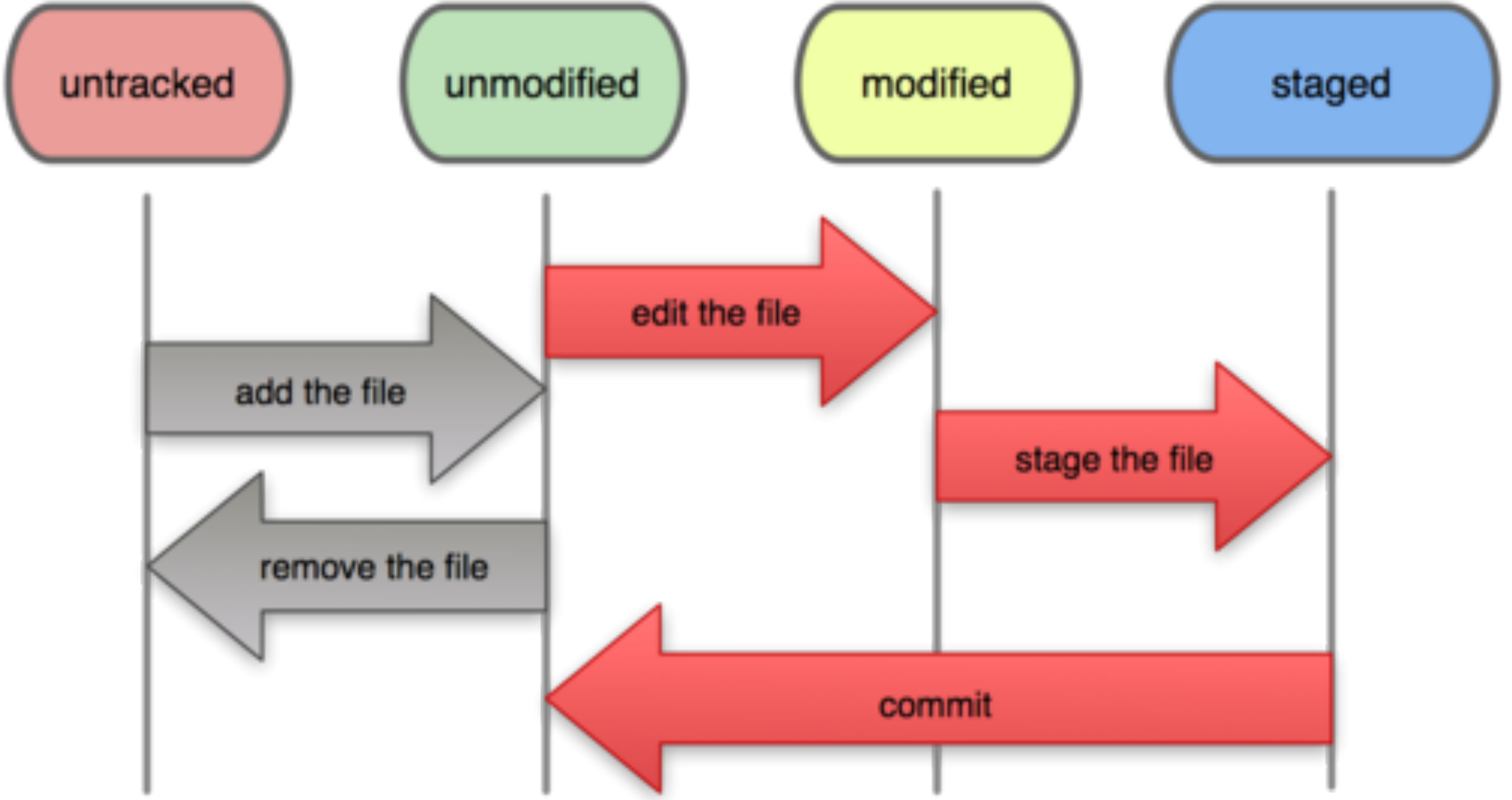
- Commit - The equivalent of a save. Commits will save the staged files with a comment. This will not be sent to the remote until a push.
- Push - Sends changes of a branch to the remote.
- Pull - Replaces your out of date working copy with the latest version of it from the remote
- Checkout - Adds a previously unknown branch to the working copy
- Fetch - Checks the remote to see if any of the branches on your working copy are out of date

# GIT COMMANDS

- Stash - A temporary storage of the working copy. Useful for switching branches of applying changes from one branch to another as it doesn't require a commit.
- Reset - Deletes your working copy and replaces it with the remote of the branch

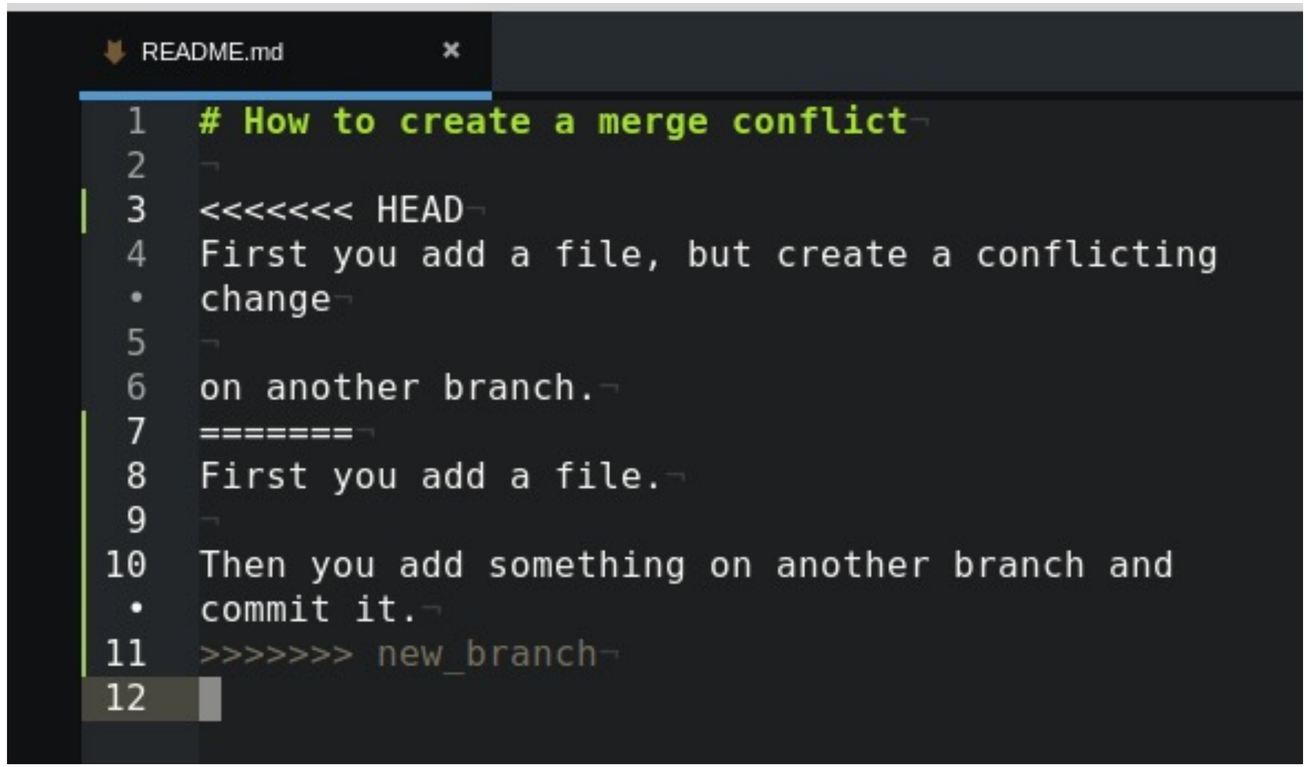


# File Status Lifecycle



# MERGE

- Merging is used to merge one branch in to another. This will essentially combine both branches and their functionality.
- Merging can create conflicts when the same lines data in a file has been edited by you and edited on the targeted merge branch.

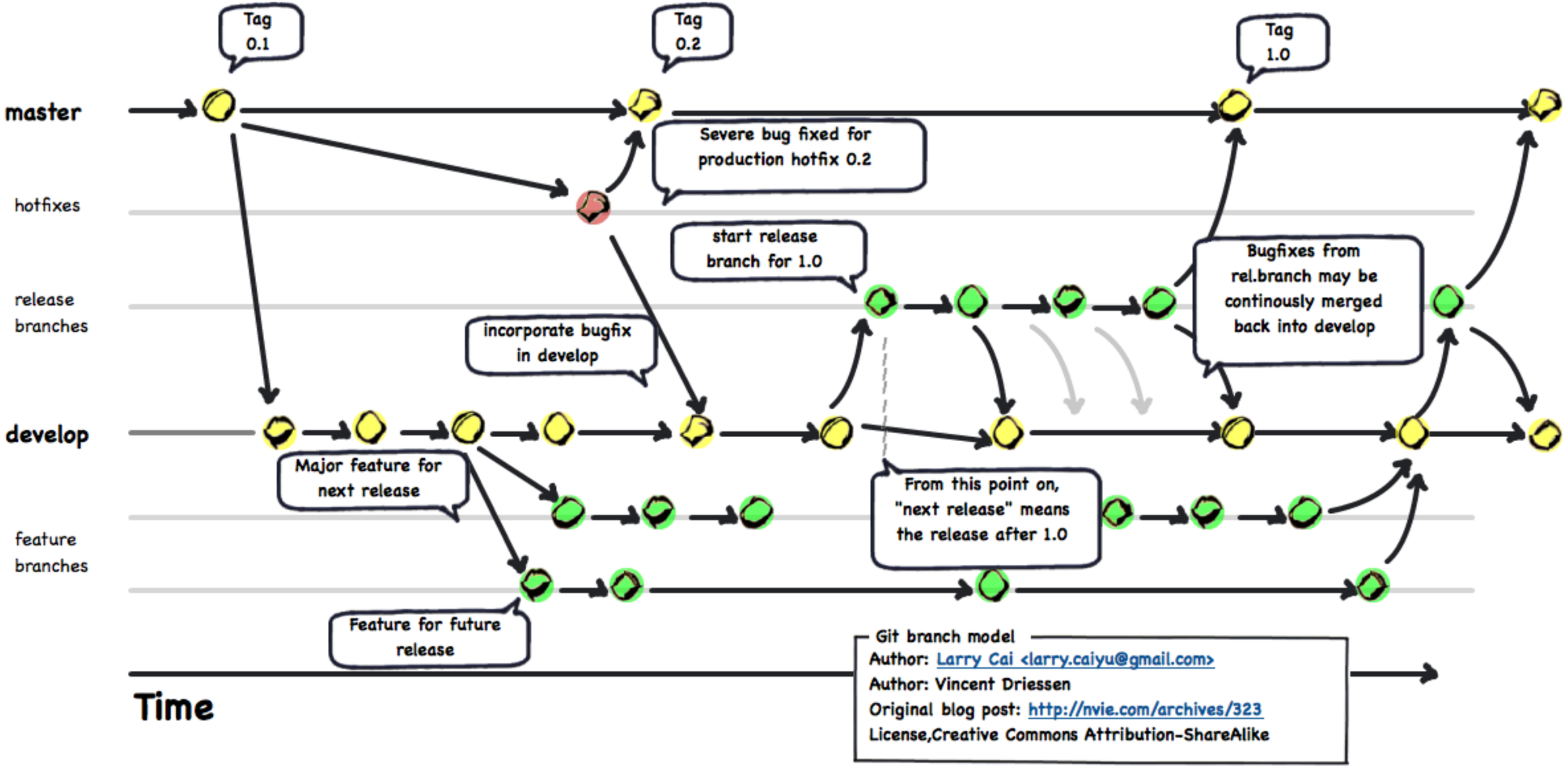


# BRANCHES

- master - The branch that represents the version of the project which is released. It will be tagged with a version for every commit made. Only directly editable by and admin.
- develop - The branch where all the features combine and represents what will be eventually turned into a release. Only admins can edit it directly but developers can through pull-requests.
- feature/feature-name - Branch which represents edits that's want to be implemented into the develop branch. Editable by everyone on the project. This is where most of the development actually happens.

# BRANCHES

- release-1.13.1 - The intermediate branch between the develop and master branches. Used to prepare the project for release by fixing bugs and building artifacts.
- hotfix - Branch that becomes active if a patch needs to be issued before the next release. Copies the master and applies the fixes before merging back in with a new version tag.

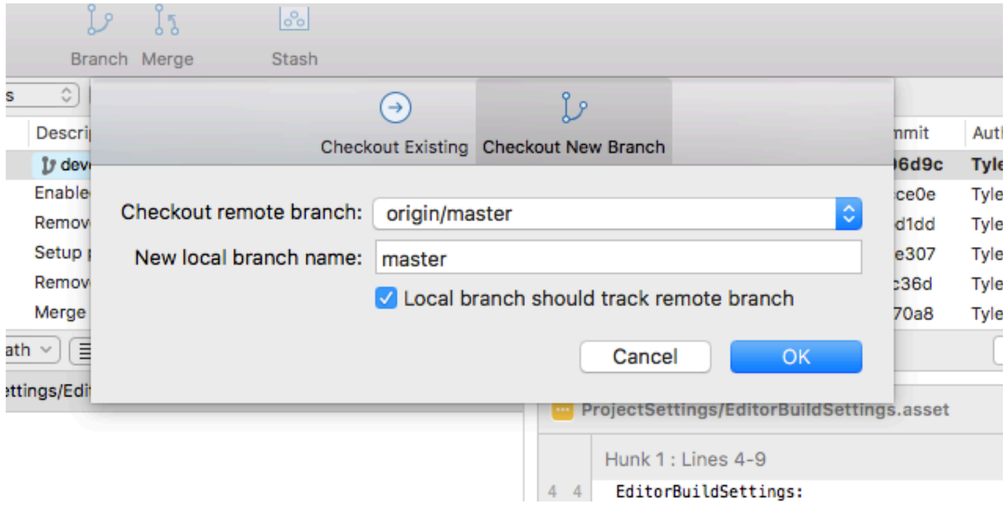

Tag 0.1
Tag 0.2
Tag 1.0
master
Severe bug fixed for production hotfix 0.2
hotfixes
start release branch for 1.0
release branches
Bugfixes from rel.branch may be continously merged back into develop
incorporate bugfix in develop
develop
Major feature for next release
From this point on, "next release" means the release after 1.0
feature branches
Feature for future release
Time
Git branch model
Author: Larry Cai <larry.caiyu@gmail.com>
Author: Vincent Driessen
Original blog post: http://nvie.com/archives/323
License,Creative Commons Attribution-ShareAlike

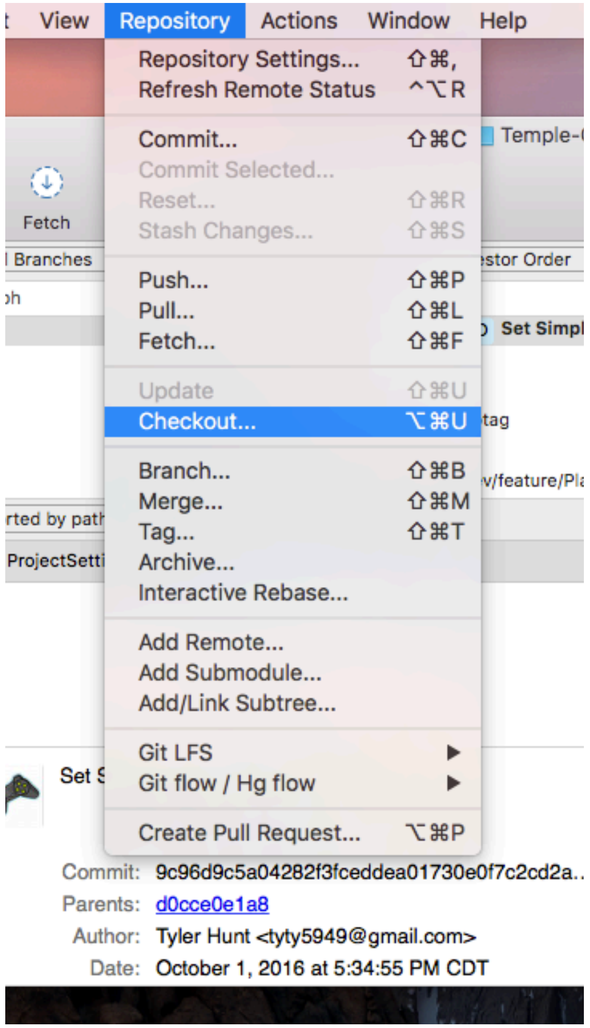
# WORKFLOW

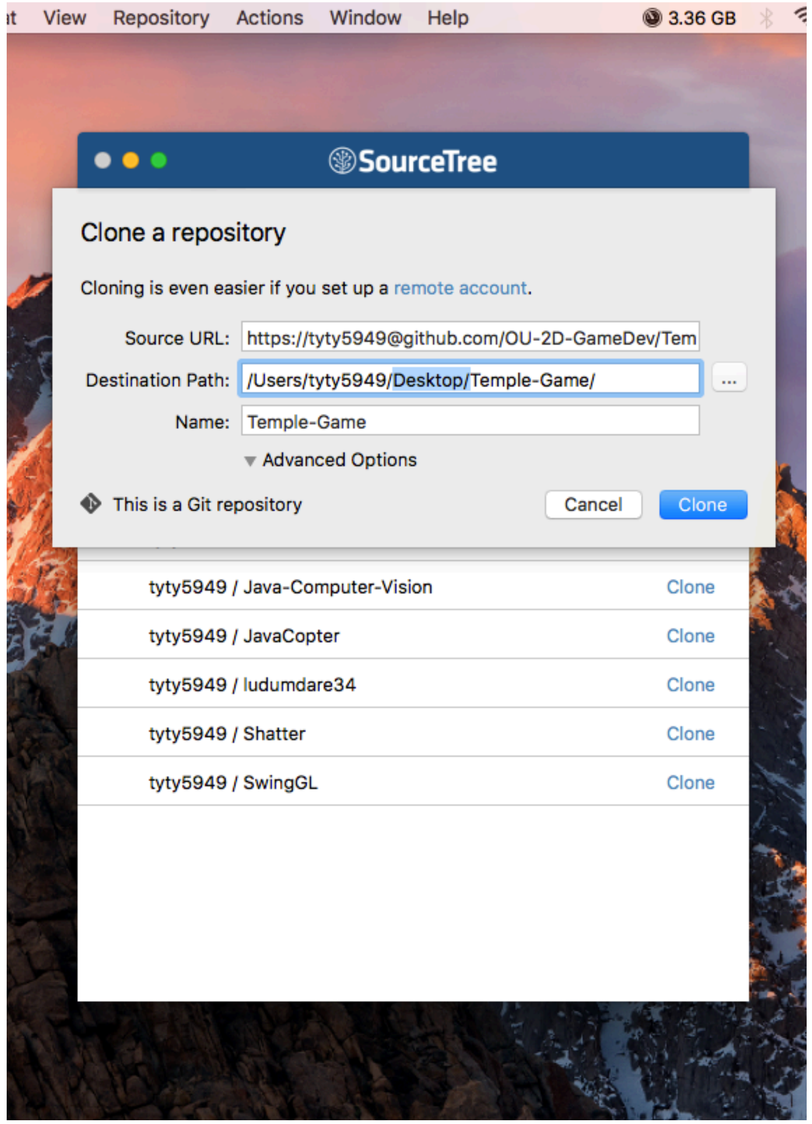
1. Clone into repository
2. Initialize Git Flow
3. Create feature branch
4. Make changes
5. Commit and push changes
6. (When ready to add feature to game)
7. Merge from develop and fix conflicts
8. Create pull-request

# CLONING IN SOURCE TREE

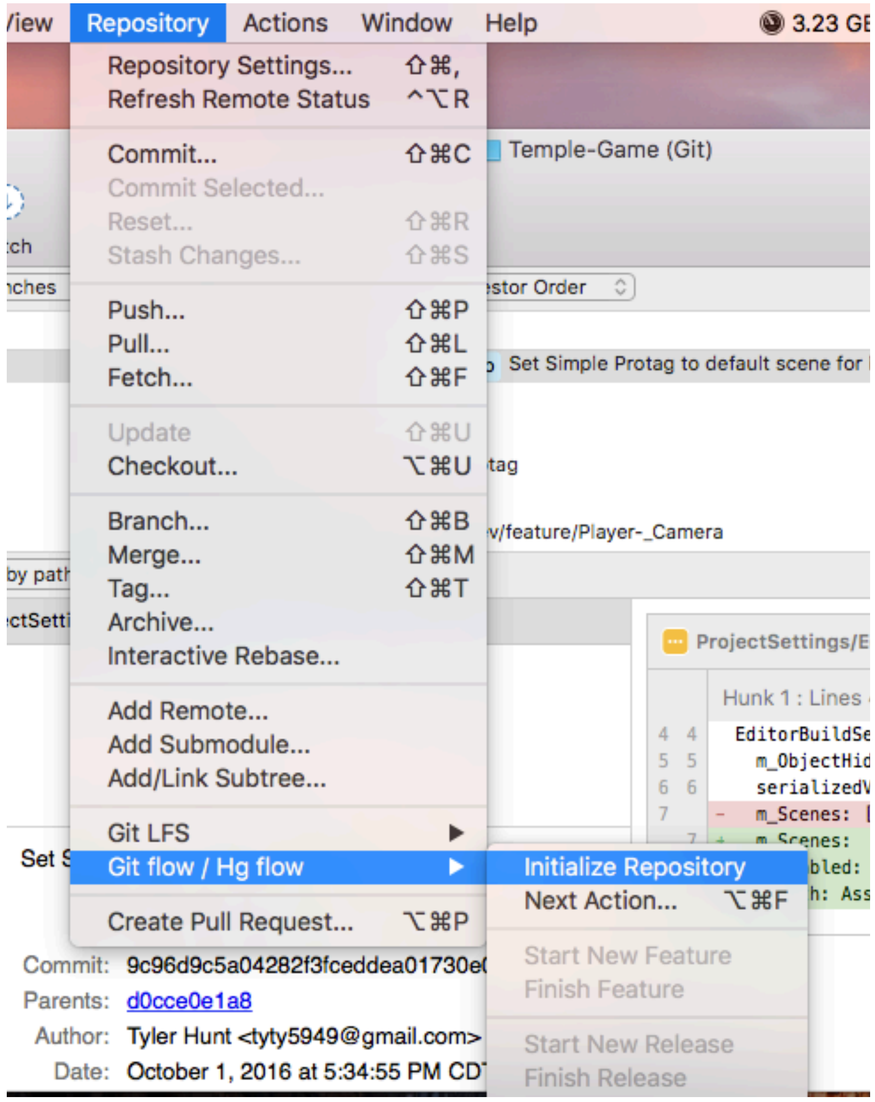
1. Clone the repository (Note: The directory)
2. Checkout master

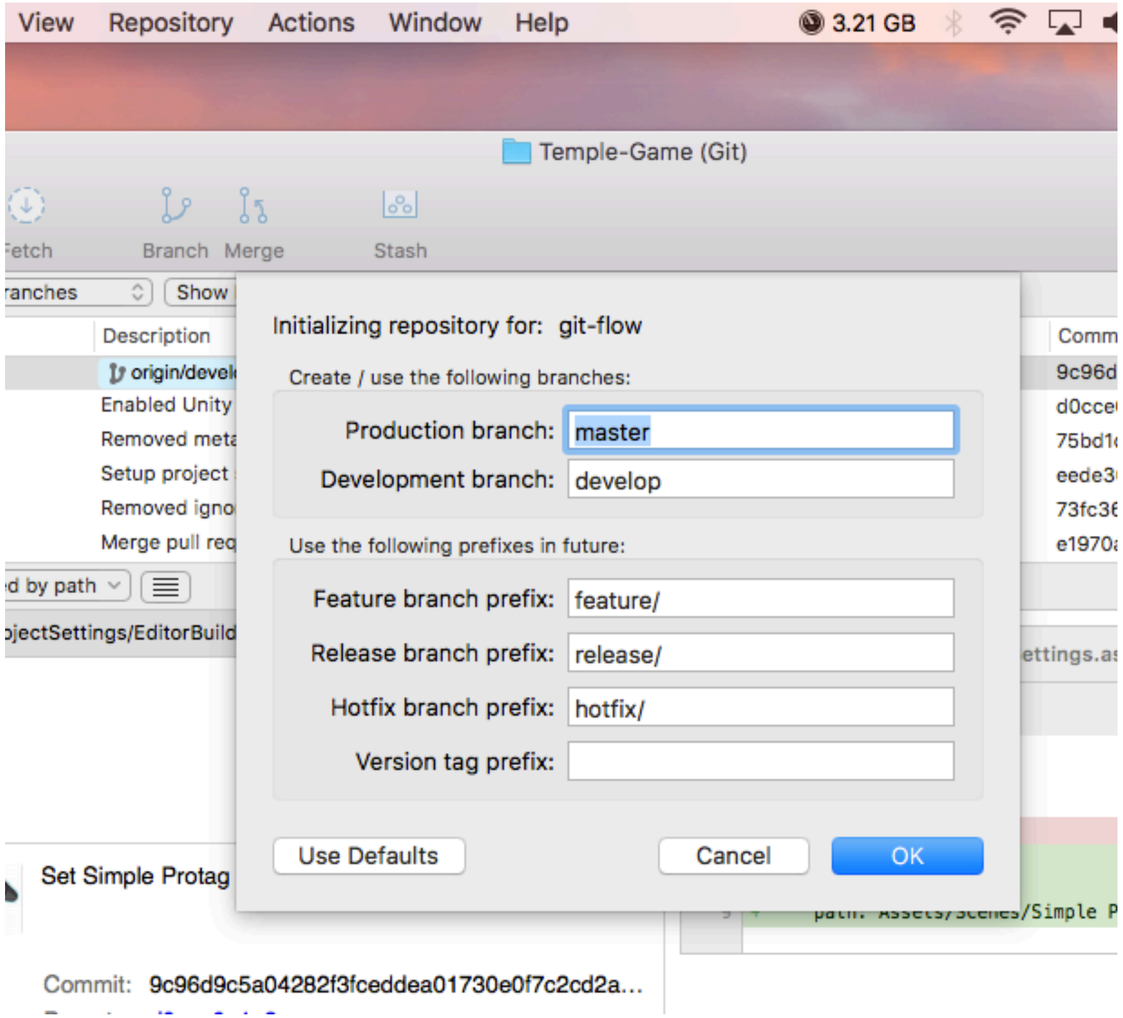


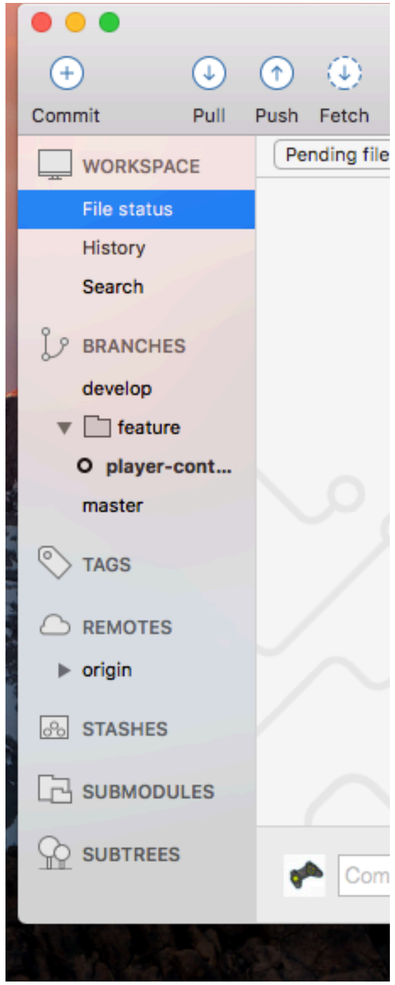
# INITIALIZE GITFLOW IN SOURCE TREE

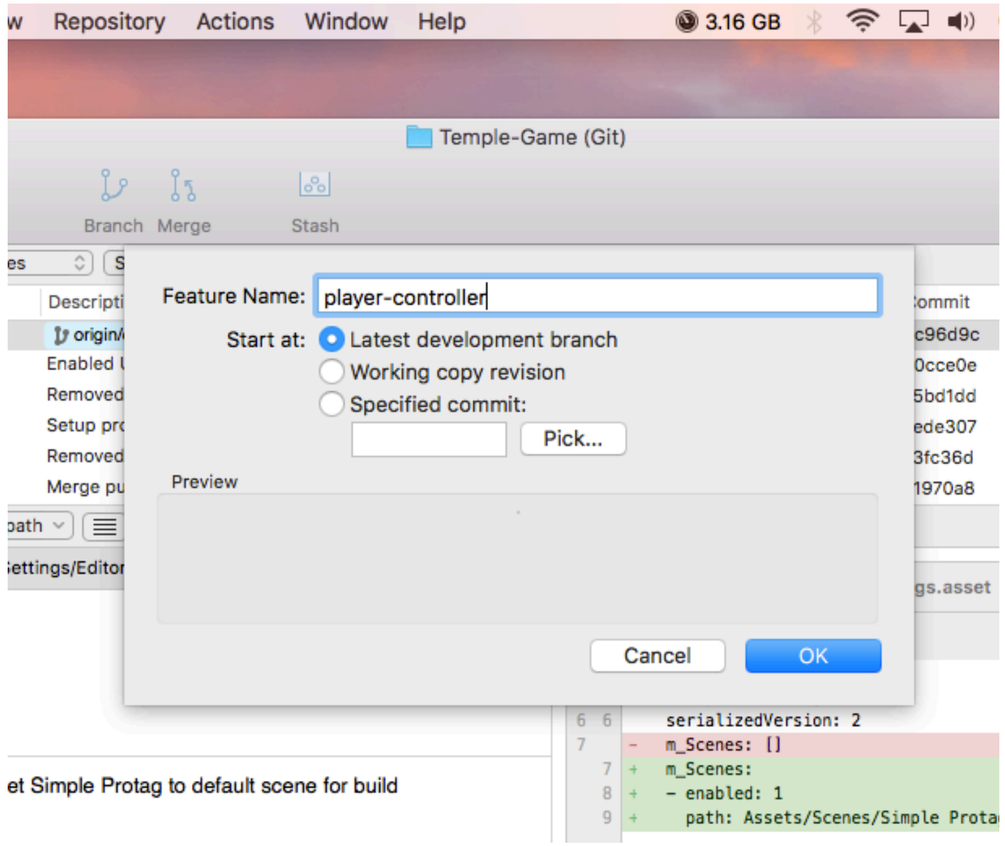


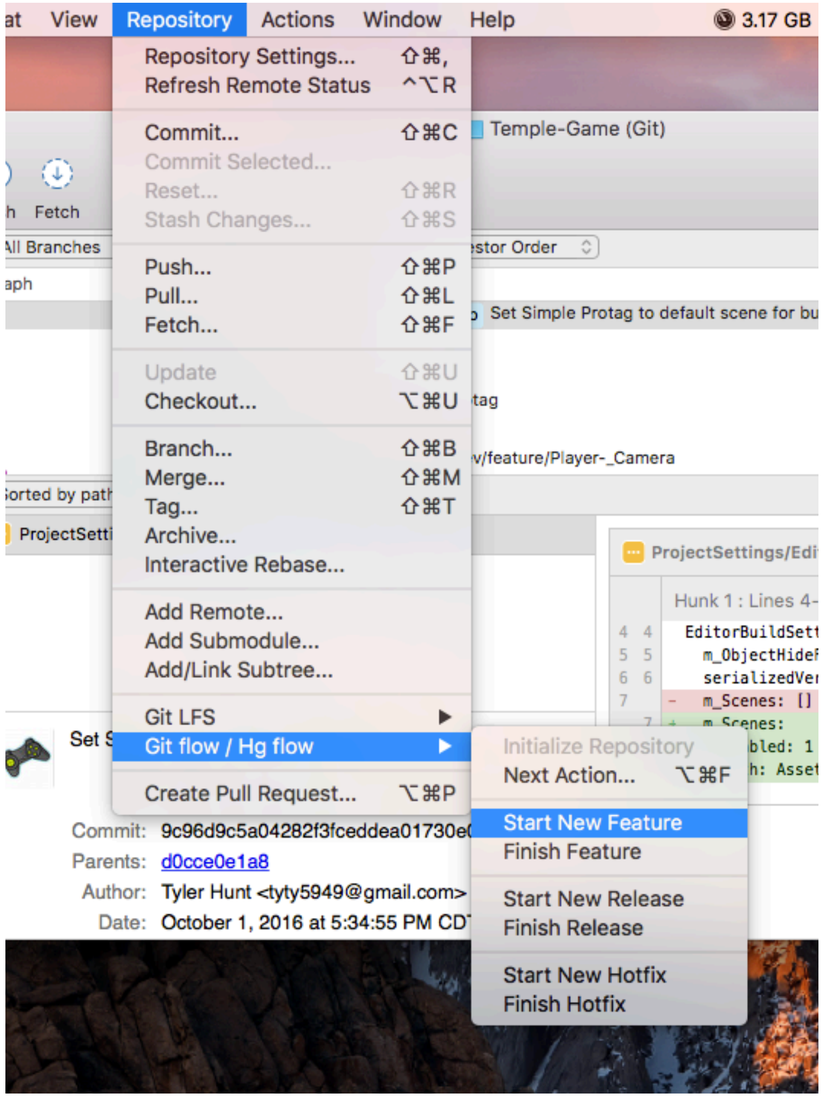



# CREATE BRANCH IN SOURCE TREE

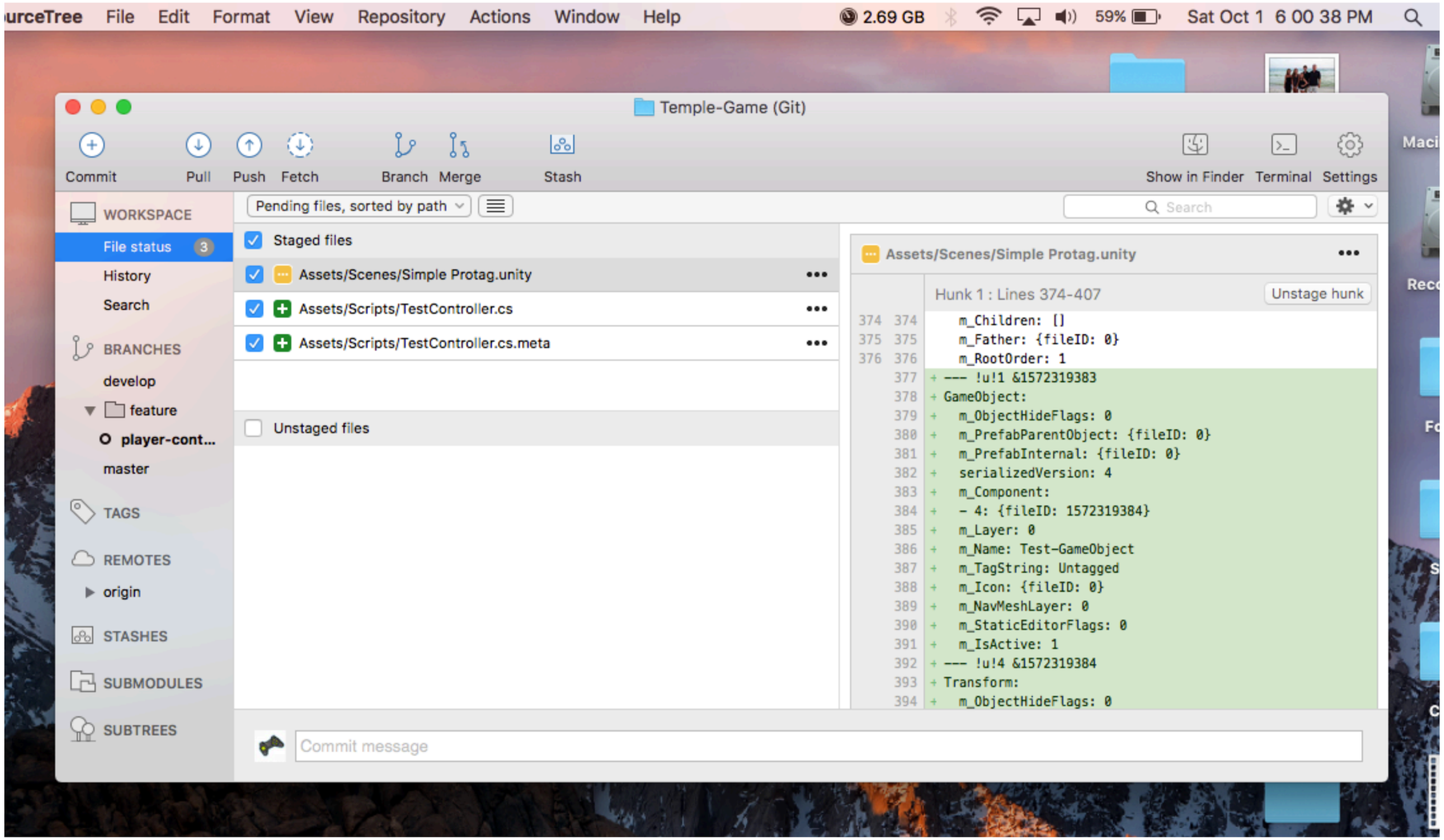
- Don’t use spaces, all lowercase



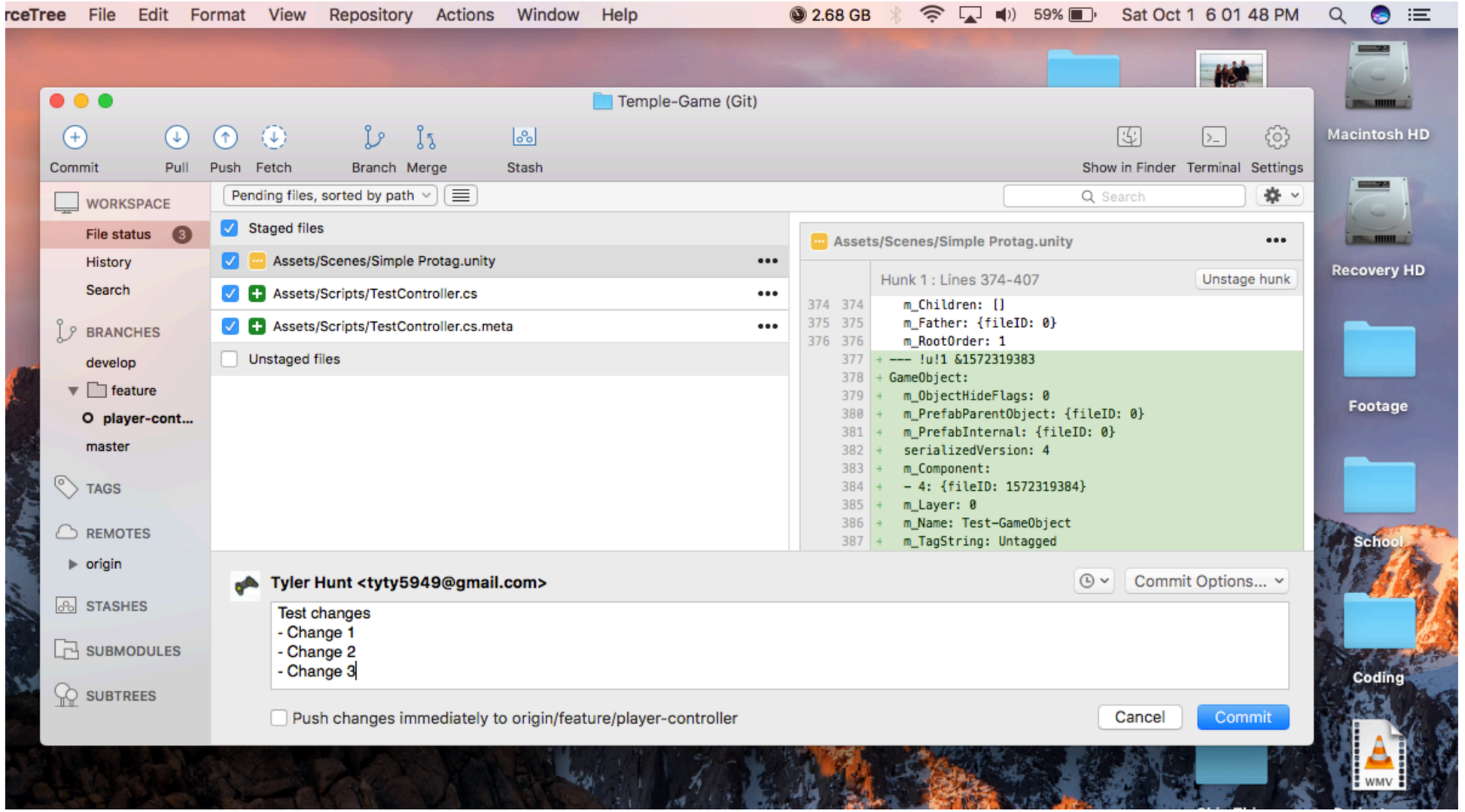



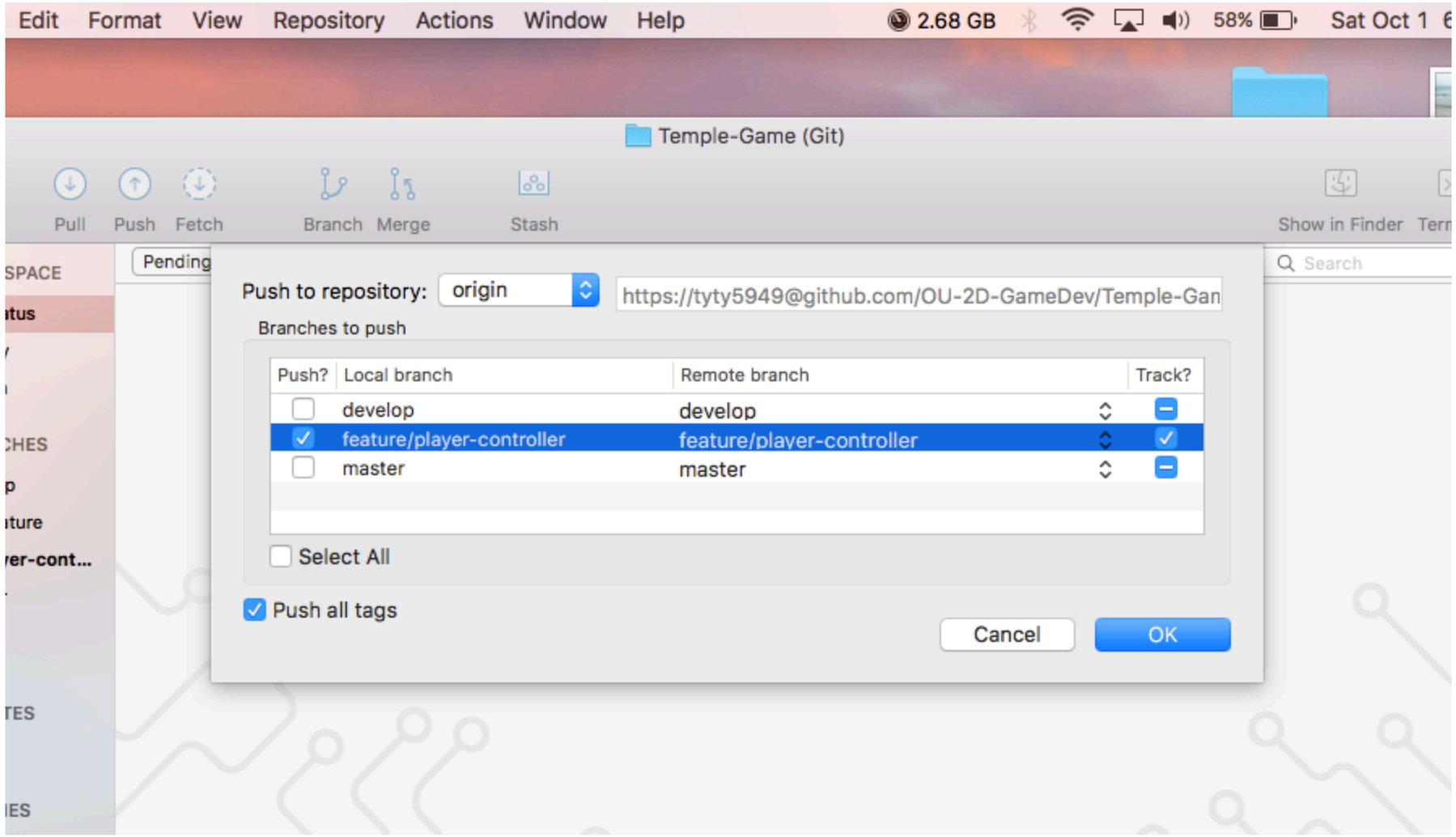
# STAGE CHANGES IN SOURCE TREE



# COMMIT IN SOURCE TREE

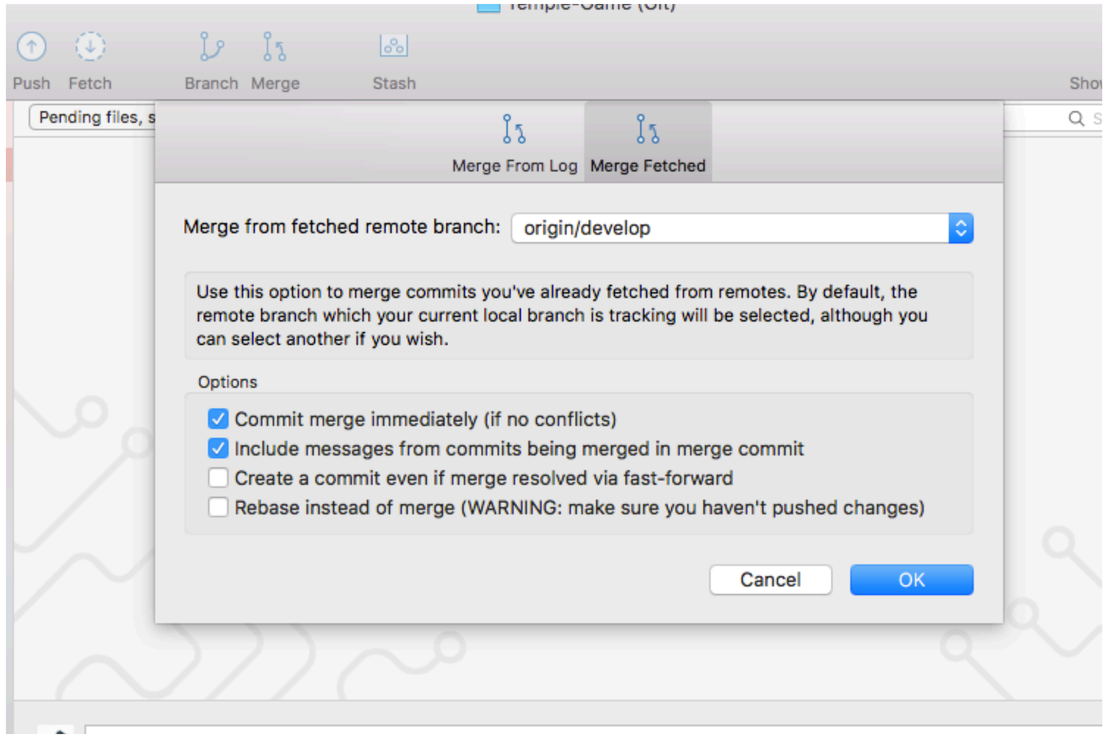


# PUSH IS SOURCE TREE



# MERGE FROM DEVELOP

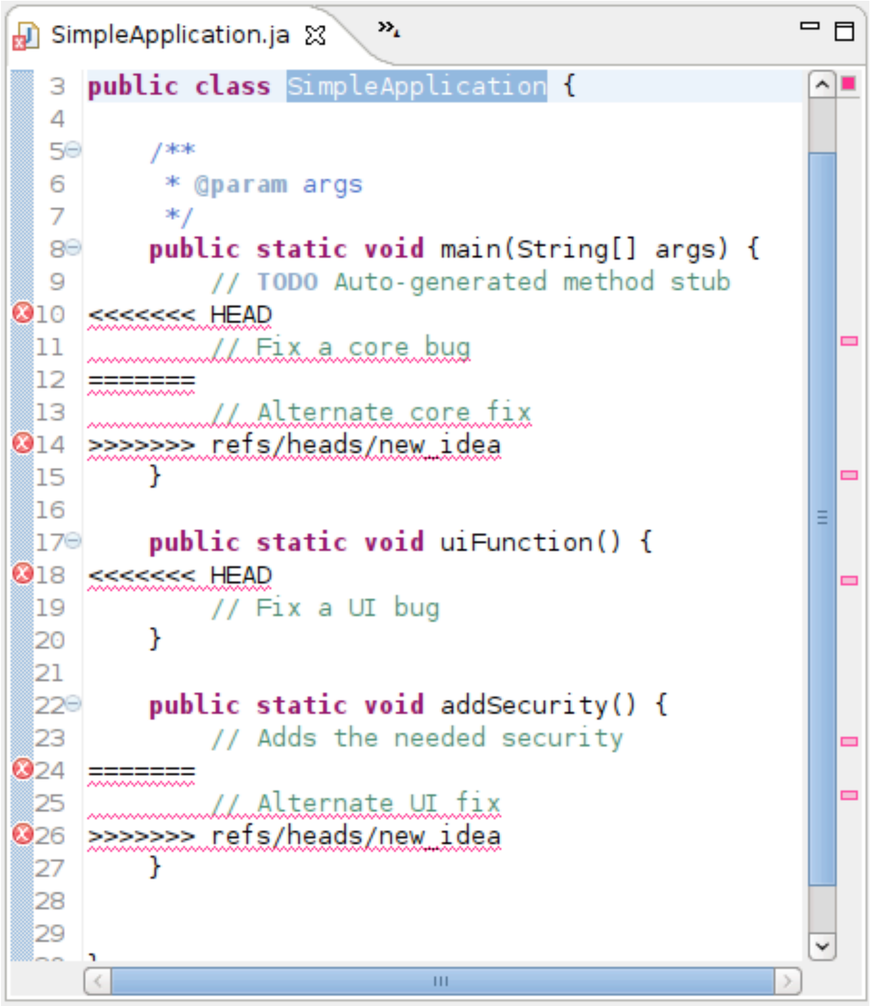
- If you need a feature that has already been merged into develop, or to test if your patch is compatible with the branch



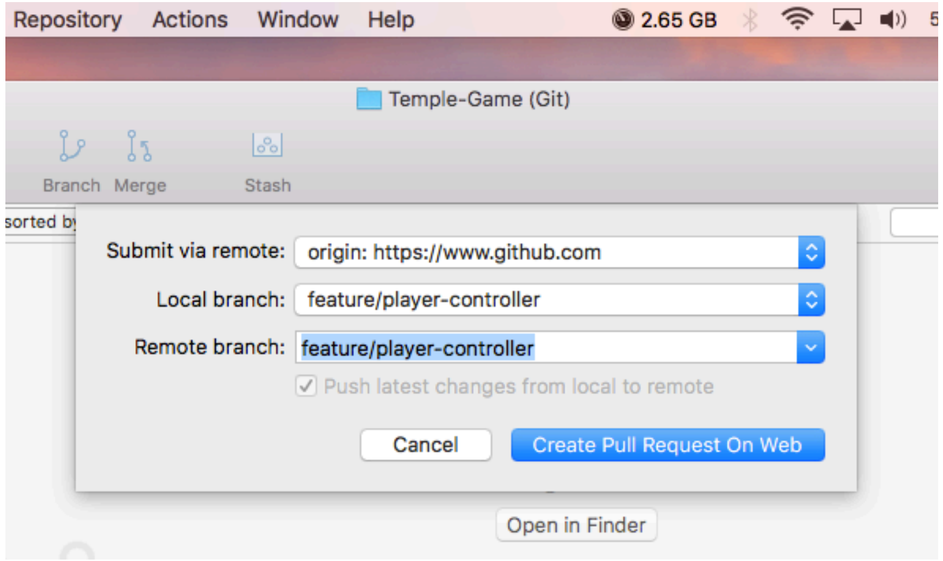

- Conflicts will have to be resolved before the merge can be committed. The merge changes WILL be reflected in your working copy.

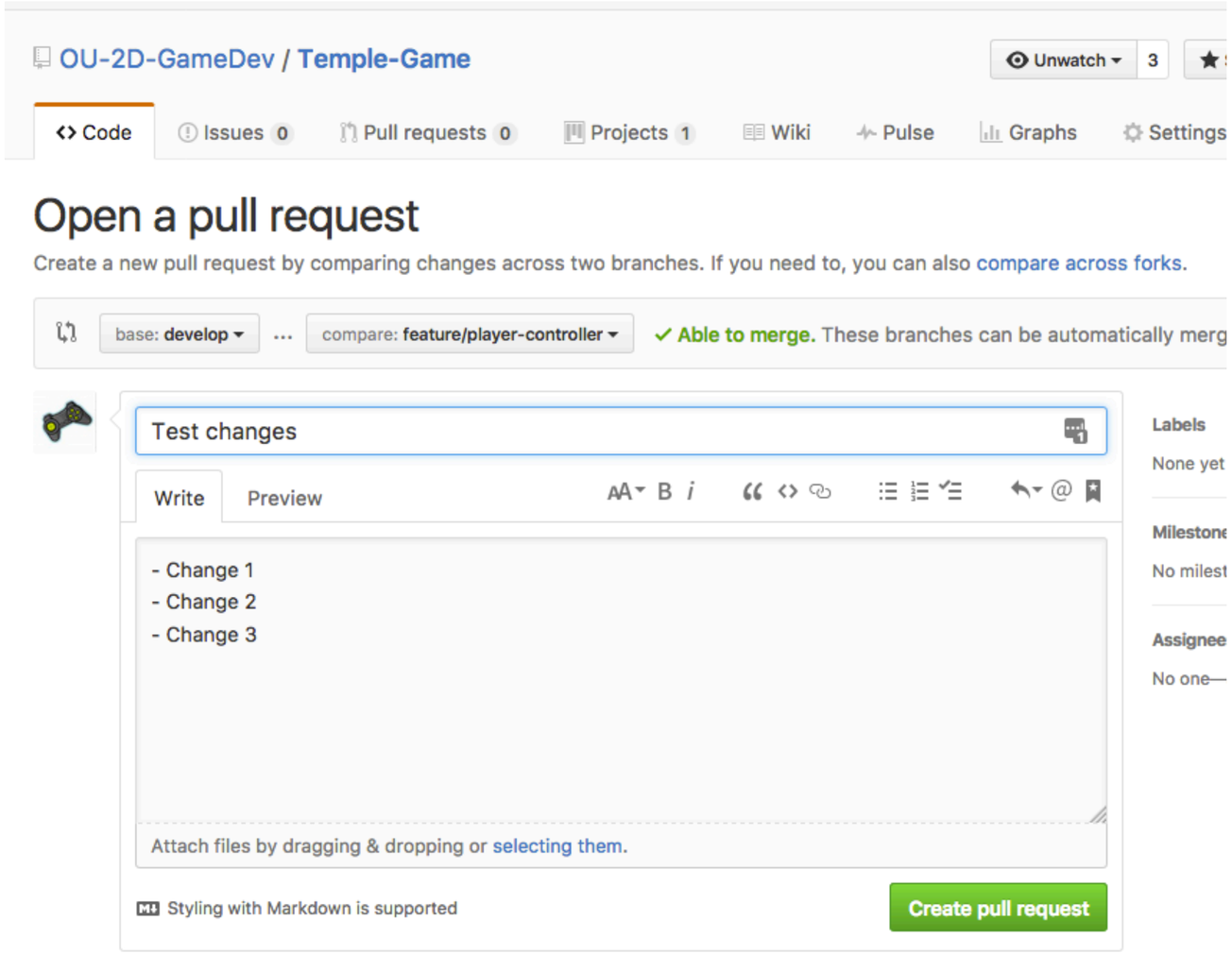
# FIXING CONFLICTS

- SourceTree will indicate what files have conflicts
- Manually edit files and choose the line of code to resolve the conflict
- Stage the edited files, then commit and push



# GITHUB PULL-REQUEST

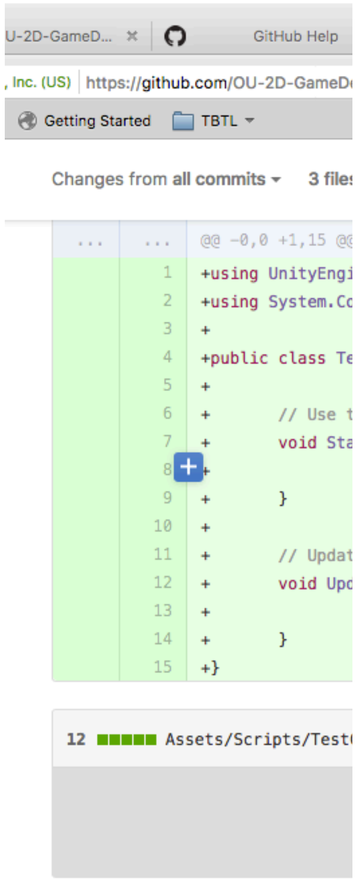


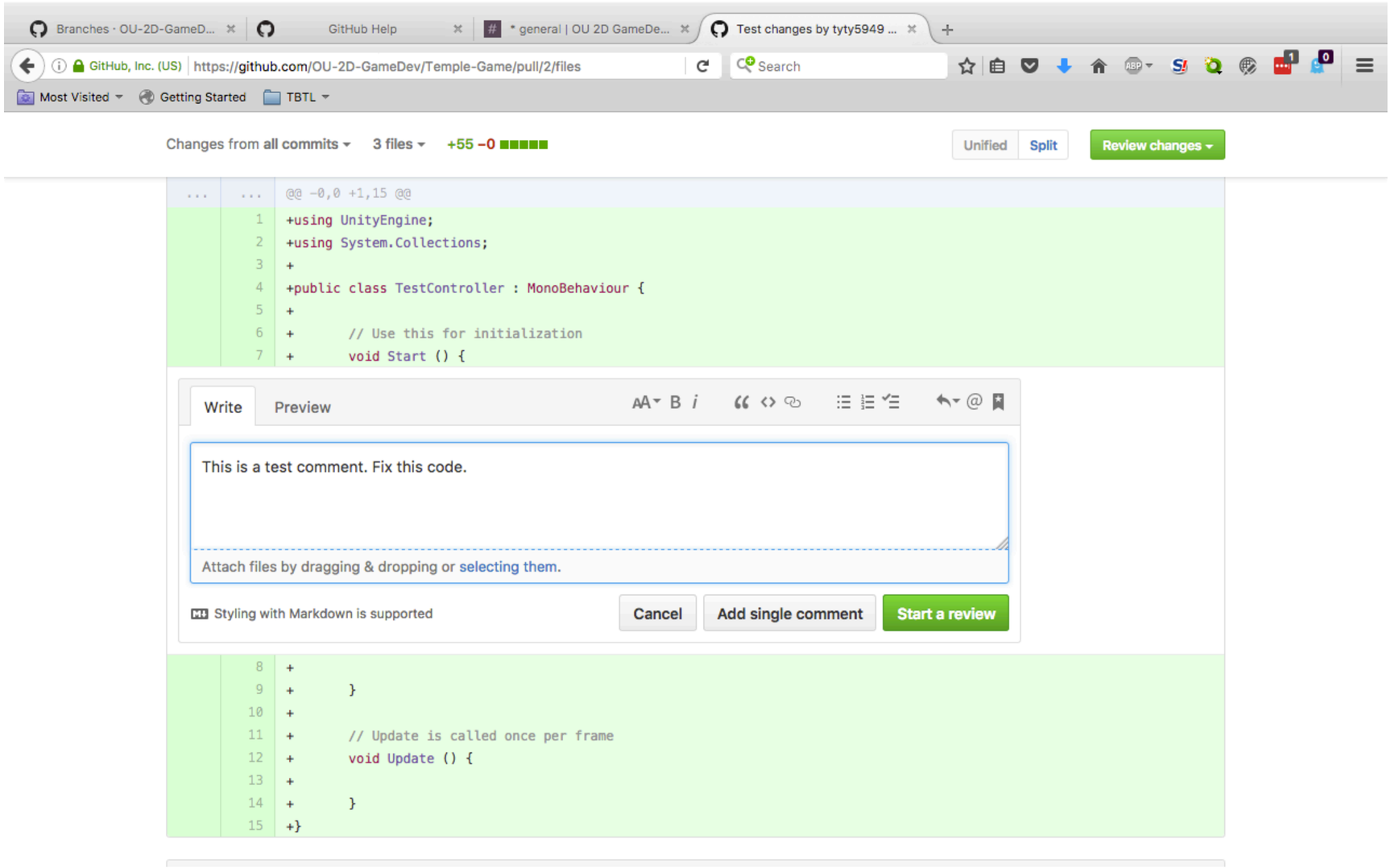



# PULL-REQUESTS

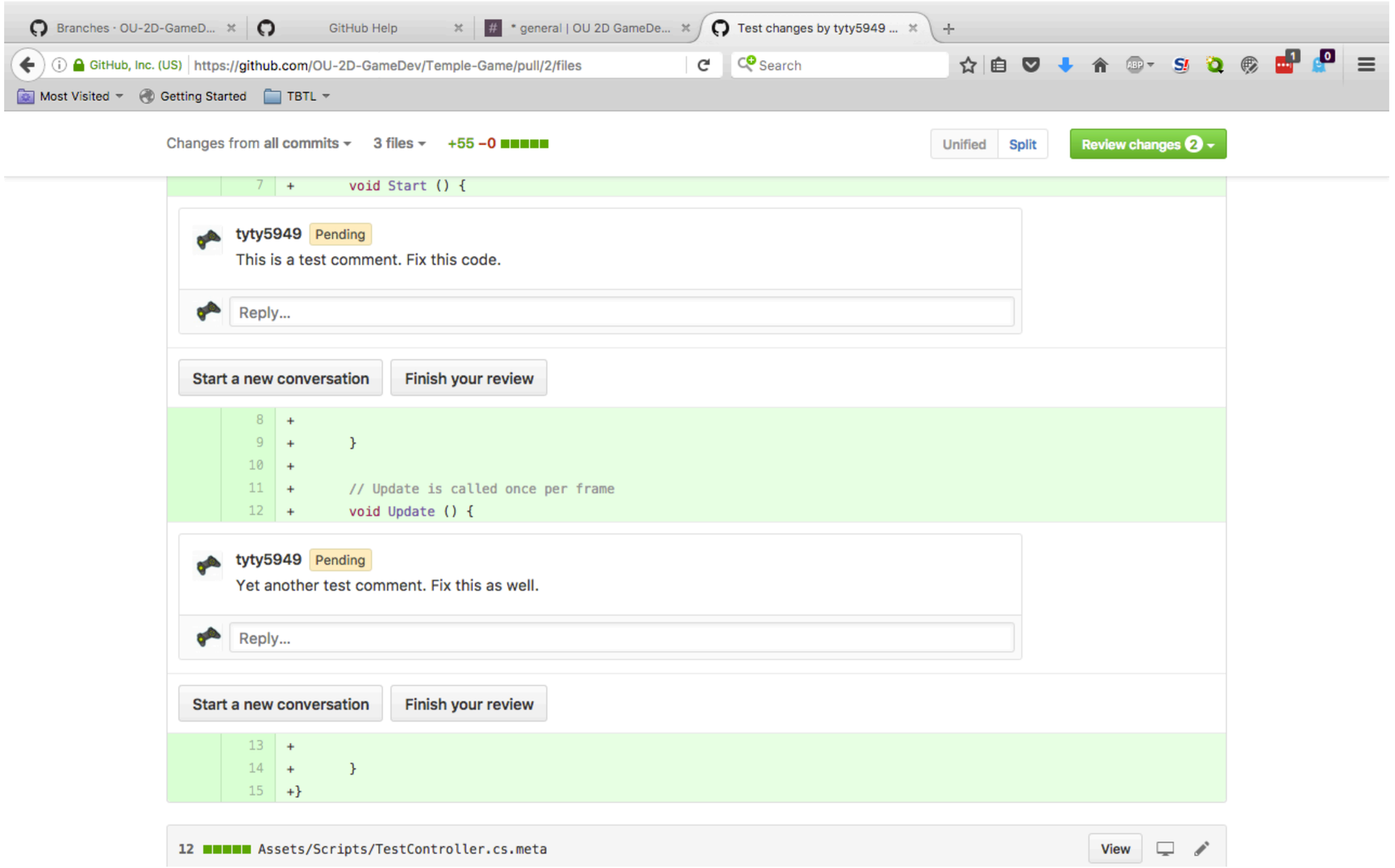
- Used to review and recommend changes before they go live on the develop branch
- At least one person must approve before the branch can be merged. IN THEORY, everyone reviews the code and either makes comments of approves.
- Notification via Slack that a pull-request has been made. As soon as possible, either comment, or approve the request.

# COMMENTING

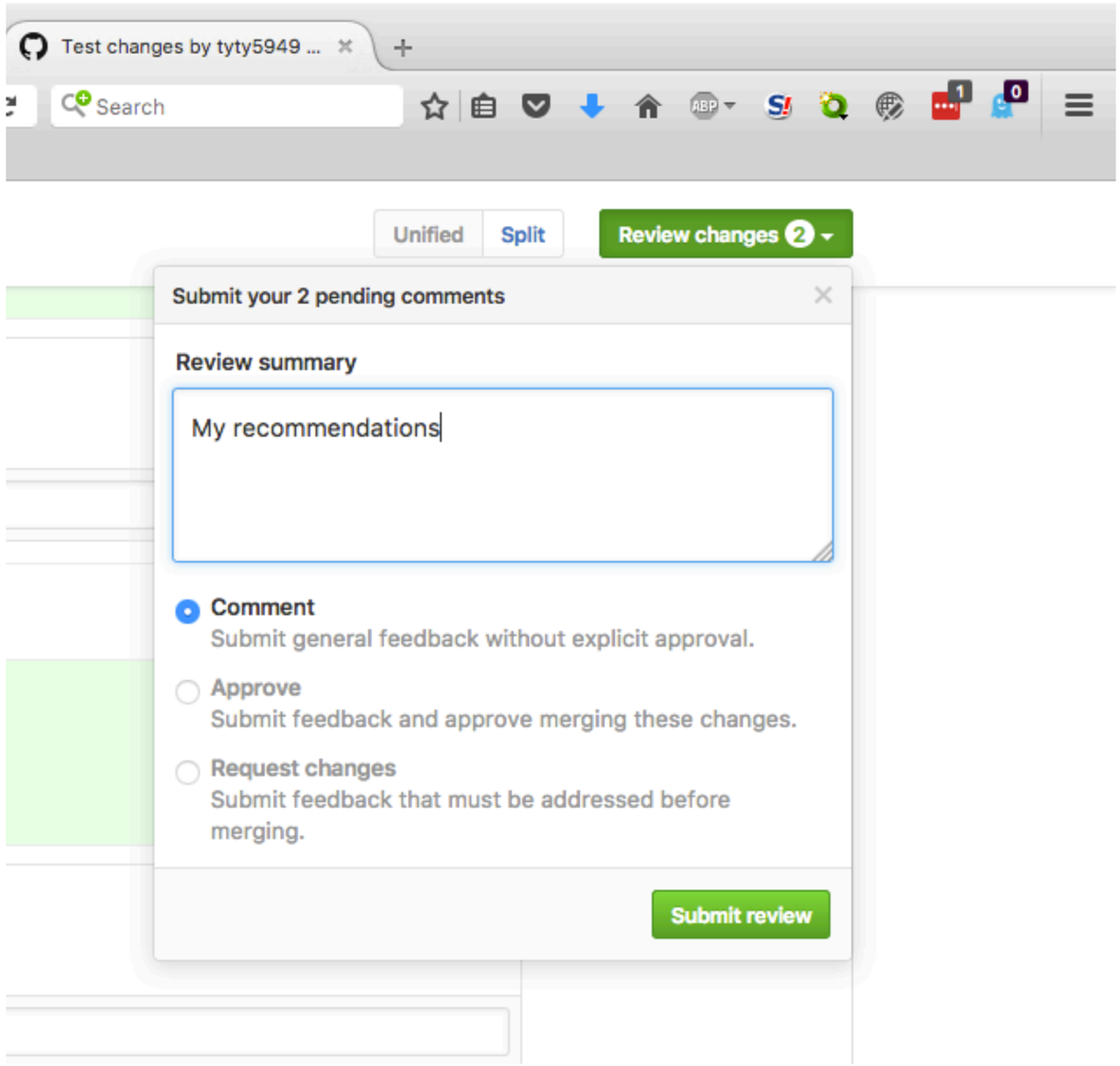

GitHub Help
https://github.com/OU-2D-GameD
Getting Started
TBTL
Changes from all commits
3 files
@@ -0,0 +1,15 @@
+using UnityEngi
+using System.Co
+public class Te
+ // Use t
+ void Sta
+ }
+ // Updat
+ void Upd
+ }
+}
12 Assets/Scripts/Test


Branches · OU-2D-GameD...
GitHub Help
* general | OU 2D GameDe...
Test changes by tyty5949 ...
GitHub, Inc. (US)
https://github.com/OU-2D-GameDev/Temple-Game/pull/2/files
Search
Most Visited
Getting Started
TBTL
Changes from all commits
3 files
+55 −0
Unified
Split
Review changes
@@ -0,0 +1,15 @@
+using UnityEngine;
+using System.Collections;
+
+public class TestController : MonoBehaviour {
+
+ // Use this for initialization
+ void Start () {
Write
Preview
This is a test comment. Fix this code.
Attach files by dragging & dropping or selecting them.
Styling with Markdown is supported
Cancel
Add single comment
Start a review
+
+ }
+
+ // Update is called once per frame
+ void Update () {
+
+ }
+}

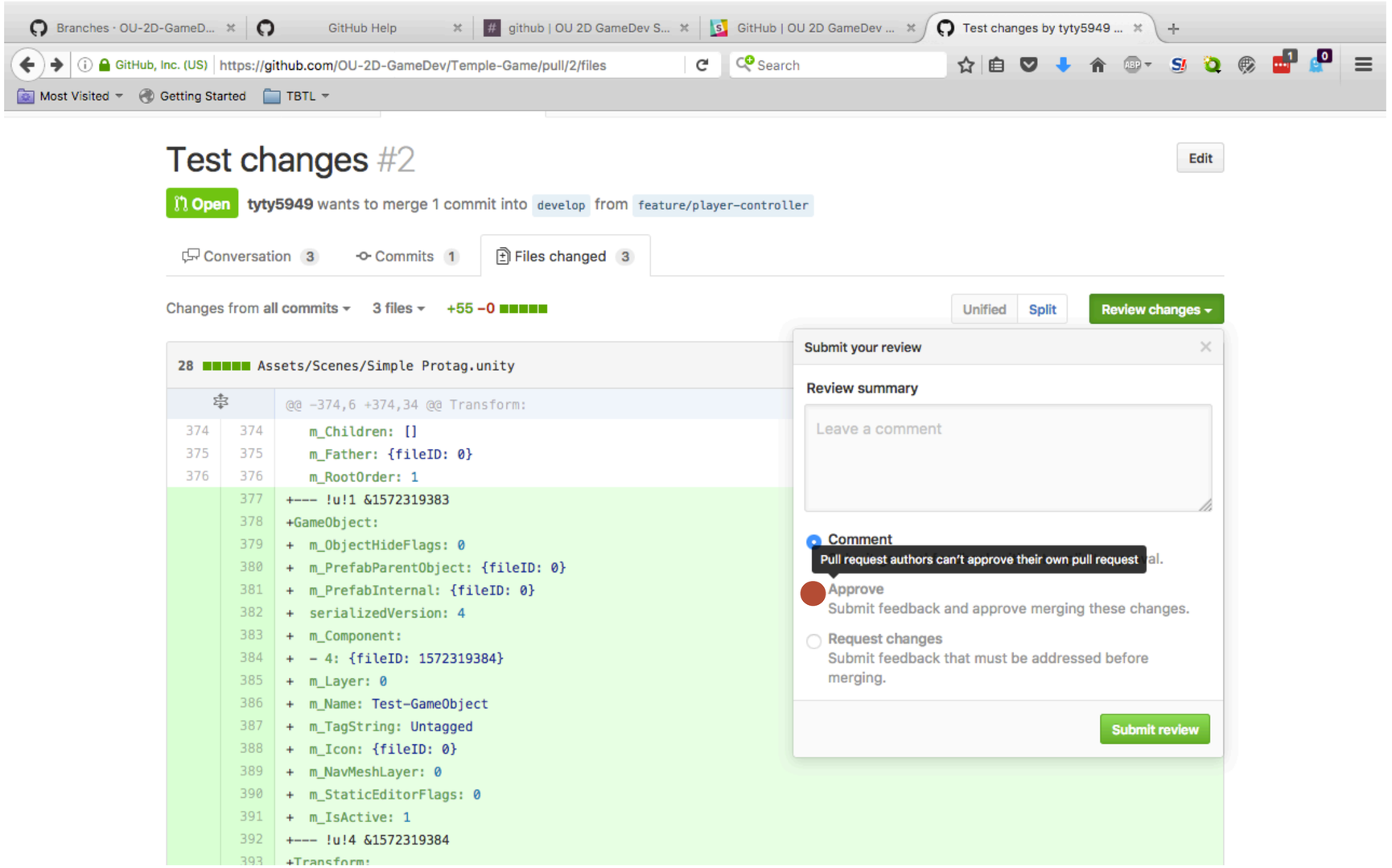
# COMMENTING


Branches · OU-2D-GameD...
GitHub Help
* general | OU 2D GameDe...
Test changes by tyty5949 ...
GitHub, Inc. (US) https://github.com/OU-2D-GameDev/Temple-Game/pull/2/files
Search
Most Visited
Getting Started
TBTL
Changes from all commits
3 files
+55 −0
Unified
Split
Review changes 2
7 + void Start () {
tyty5949 Pending
This is a test comment. Fix this code.
Reply...
Start a new conversation
Finish your review
8 +
9 + }
10 +
11 + // Update is called once per frame
12 + void Update () {
tyty5949 Pending
Yet another test comment. Fix this as well.
Reply...
Start a new conversation
Finish your review
13 +
14 + }
15 +}
12 Assets/Scripts/TestController.cs.meta
View

# COMMENTING


Test changes by tyty5949 ...
Search
Unified
Split
Review changes 2
Submit your 2 pending comments
Review summary
My recommendations
Comment
Submit general feedback without explicit approval.
Approve
Submit feedback and approve merging these changes.
Request changes
Submit feedback that must be addressed before merging.
Submit review

# APPROVING A PULL-REQUEST



# UNITY WORKFLOW

- Keep workspace organizes (folders, naming, readability)
- Write scripts and name things in a way which anyone will be able to understand what they do
- If necessary, create a new scene to test a feature. Try not to modify game levels just to test a feature.
- Scenes create for testing should have the same name as the branch of the feature
- If checking out a feature to test it, you will have to load the scene manually, it will NOT automatically load.