Welcome to Week 4 - Intro to Git

These brief tutorials are designed to equip you with the tools you need to immediately start creating and updating “Git” repositories with Bitbucket or Github.

Bitbucket is similar to Github or Gitlab in the sense that it allows you to store your project files to a source control system.

Do not be too discouraged if this process is confusing at first. Source control often appears to be way more complicated than it actually is. That is why we are only covering the basics needed to get started.

Table of Contents

Part 1 - Intro to Git

Git allows you to save your projects to a repository, which functions similarly to a list of time-marked save points in a video game. The main feature is version control, which allows you to access and download previous version of your project in case something breaks.

The following setup pages will help you get started using git repositories for EAC projects and beyond.

What is a Git Repository?

- https://www.geeksforgeeks.org/what-is-a-git-repository/

Part 2 - Setup Bitbucket or Github

2.1 Bitbucket Setup

- https://eac-ualr.atlassian.net/wiki/pages/resumedraft.action?draftId=3070886055
  If you are not using Bitbucket for this tutorial, set up a GitHub account instead: https://docs.github.com/en/get-started/quickstart/set-up-git

2.2 Github Setup

- If you do not have access to Bitbucket, follow the Github Setup instructions.

Part 3 - Setup Sourcetree or Git Bash

3.1 Sourcetree Setup

- Checkout https://eac-ualr.atlassian.net/wiki/spaces/360Intro/pages/3070755147 to get started with Sourcetree.

3.2 Git Bash Setup

- Checkout https://eac-ualr.atlassian.net/wiki/spaces/360Intro/pages/3070885925 to get started with Git Bash.

Part 4 - Commit, Branch, Push, and Pull

Here is a list of common terminology that you will need to understand to start using Git.

4.1 Commit

- In the Commit stage, select the project files that you would like to add to the repository. Then add a detailed comment that describes the changes being made to the overall project.

4.2 Branch

- A branch is used to split a project into two different versions that can be updated and changed separately from one-another.
- Branching a project is useful for testing and reviewing new features before they are implemented into the working project.
- Making a branch is an excellent way to save controller and build settings when deploying to various platforms. (i.e. WebGL branch, Oculus Quest branch, iOS branch, etc)

4.3 Push

- When you are confident about the changes you are committing, it is time to use Push.
- Be sure to double check which branch of the project you are building to.
  Do not push test features into the main branch when adding to a team project. If you do you're going to have a bad time. It is possible to revert changes after they are pushed to a repository, but it can be very time consuming.

4.4 Pull

- The pull function allows you to download changes made to a remote repository, without needing to re-download the entire project.
- It is a good practice to review the changes and check for errors before pulling a project.
  If project changes do not work immediately after pulling them, clone a fresh version of the repository, save it in a brand new folder in your directory that is easy to identify (date-projectname-version), then download the updated project.

Once you have setup Bitbucket and Sourcetree successfully, complete the following steps.

1. Create a new repository with a unique project name.
a. Remember to create your repository in Bitbucket first, then clone the new repository to your computer.
b. Once the cloned repository folder is saved in your computer, place the files for your project into that cloned repository folder.
2. Commit and “Push” one file into the repository. This can be a text file, an image, or some code.
a. This is an exercise, so keep the file size small.
3. Share and clone a repository with a class mate by sharing file access in Bitbucket.
a. Feel free to ask EAC members outside of your intro class. We are here to help!