

AS226472

Git It Together – An Introduction to Version Control

John Pierson
Parallax Team, Inc

Learning Objectives

- **Gain** an introductory understanding of version control processes.
- **Understand** key Git terms.
- **Learn** how to manage your files using version control.
- **Learn** how to actively contribute to open-source projects.

Description

Tool customization, automation, data mining, and design iteration exploration are a few examples of why you should learn to code in the architecture, engineering, and construction (AEC) industry. Coding includes the use of text-based languages such as Python or C#, or visual programming environments such as Dynamo and Grasshopper. With the use of these tools come additional methodologies that we need to learn, such as version control or Git. You've probably heard of this process through GitHub, which is a version control platform (Dynamo is hosted on GitHub). Essentially, version control lets you manage revisions regarding your code. The problem with Git (version control) is that this process isn't introduced in a way that makes the most sense to a Revit user. That is where this class comes in: We'll introduce the process and key terms and explain why version control matters to you. After this class, attendees will have an introductory understanding of how to use Git and will be able to manage their files using a version control process.

Speaker



John Pierson is a Design Technology Specialist at Parallax Team, a full-service Implementation firm. Specializing in the creation of custom tools and workflows for the AEC industry, John has become well known for his Dynamo-based solutions. Currently, John manages several Dynamo packages that are available to the Dynamo community. This includes Rhythm, Bang! and Duct Tape totaling over 45,000 downloads.

In addition to the creation of automated workflows and custom tools, John is a Revit certified professional for all disciplines. As a Revit certified professional, he provides custom content creation, project-based modelling, and Revit training. John is an avid contributor to the Dynamo community, being a moderator on the Dynamo forum and a contributor to the Revit Forum. John is also a top speaker at many events including, Autodesk University, BiLT Conferences and various user groups everywhere. Find him at sixtysecondrevit.com, [@60secondrevit](https://twitter.com/@60secondrevit) or on parallaxteam.com

Overview

Git is a [free and open source](#) distributed version control system designed to handle everything from small to very large projects with speed and efficiency. (<https://git-scm.com/>) Essentially Git allows us to manage changes in files in a very efficient way.

So, why should we in the AEC industry care about this?

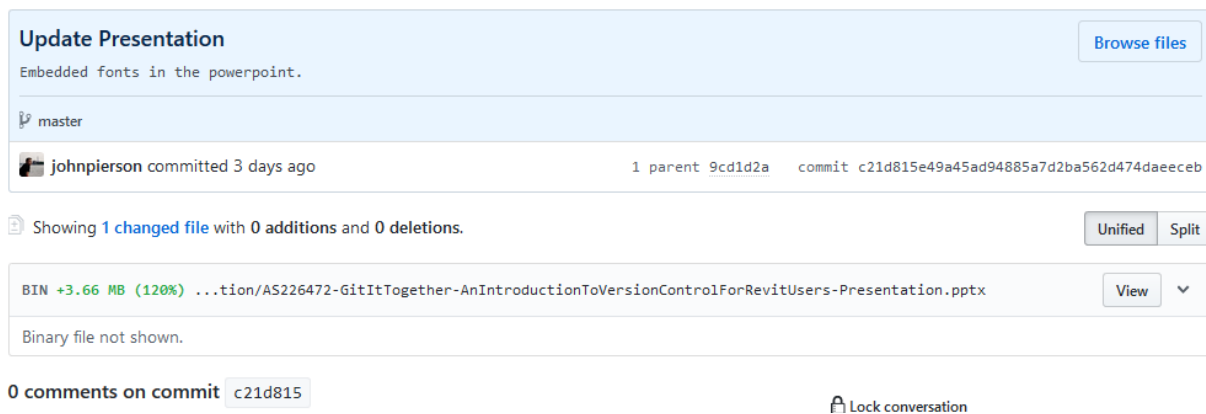
1. Sometimes we are not so great at our own version control.
2. We have all lost work at one time or another.
3. AU and similar conferences are turning out more and more coders every year. Along with coding we need to learn how to manage our precious scripts and code.
4. Git is another skill that spans between industries.

Git is Good at a lot of things.

Here are some of the best features of using Git.

Snapshots

In Git, versions are saved as a “snapshot”. Snapshots allow you to return to the code or file base at that state on-demand. On Github these are known as commits.



The screenshot shows a GitHub commit page for a presentation. The commit title is "Update Presentation" and the description is "Embedded fonts in the powerpoint." The commit was made by johnpierson 3 days ago. The commit message is "Showing 1 changed file with 0 additions and 0 deletions." The file being changed is "AS226472-GitItTogether-AnIntroductionToVersionControlForRevitUsers-Presentation.pptx", which is a binary file (+3.66 MB, 120% change). The file is not shown in the diff view. There are 0 comments on the commit.

Sample Commit from this Class' Repo

Viewing Differences

Since one of the major focuses of Git is to enable collaboration, you can review differences in files in amazing ways.

Showing 1 changed file with 5 additions and 3 deletions. Unified Sp

```

8 ■■■■ README.md <> View
... .. @@ -1,11 +1,13 @@
1 - # AU2018-GitItTogether
2 - ![Image](_resources\Images\homeImage.png)
3 + ![Image](_resources/Images/homeImage.png)
4 - # What is this?
5 + # Where am I?
6   This repo contains the files for the presentation, "Git it Together", that will be demonstrated at [Autodesk University
7   2018](https://autodeskuniversity.smarteventscld.com/connect/sessionDetail.wv?SESSION_ID=226472&
8   tclass=popup#.W9dy01NZ3_Q.twitter). Since this presentation is about getting started on Github, it seemed fitting to build
9   it using the same methodologies.
10  # Who?
11  Presenter - John Pierson of [ParallaxTeam](http://www.parallaxteam.com/) and [sixtysecondreelit.com]
12  (http://sixtysecondreelit.com/).
13  # License
14  <a rel="license" href="http://creativecommons.org/licenses/by-nc-nd/4.0/"></a><br />This work is licensed under
16  a <a rel="license" href="http://creativecommons.org/licenses/by-nc-nd/4.0/">Creative Commons Attribution-NonCommercial-
17  NoDerivatives 4.0 International License</a>.
18  +
19  + # Directory
20  + _to fill out with directory info when complete_

```

Sample Diff Generated for an Update of a repo's README

Giving Back to the Community

With Github you can manage your project in two ways. This includes out in the open and hidden. With Github and a free account, by default your projects will be open, (more on that later). Since our projects are potentially out in the open as well, we must worry about licensing. Thankfully there are a lot of options out there. For this presentation I went with a pretty simple one from [creativecommons](https://creativecommons.org/).

- Help choosing an Open Source license: <https://choosealicense.com/>
- Creative Commons Licenses: <https://creativecommons.org/>

For a project, a license is saved simply as "LICENSE" in the root directory.

So, why is this not more common in AEC?

When I first learned about Github and Git in general, I immediately wondered why this was not something we were all familiar with in the AEC industry; version control, cloud backups, multiple collaborators sounded like a dream come true! But, I immediately learned there was a bit more to it than simply uploading files. There are quite a few new terms to learn and other items to keep an eye out for.

Key Terminology

As always, along with a new process comes some new terminology to learn. Below are the most common items you are going to run into with Git.

- **Command Line:** Interface for text-based input for Git. (not how this class is teaching this)
- **Repository/Repo:** Storage space where your projects live. (like a Revit central file)
- **Version Control:** Instead of manually saving multiple versions, Git saves “snapshots” of every point in time for the life of the project. (like Revit’s backup folder, but better.)
- **Commit:** This is the beautiful part. A commit is you pushing your changes to Git. Each time you commit to the repo, Git takes a snapshot to give you a checkpoint to go back to if needed. (like a sync with central, but also way better)
- **Branch:** This allows people to work on their own version of the project before pushing the changes back to the main project. (like a Revit local file)
- **Fork:** Your very own copy of a project separate from the source project with the option to push your changes to the source if you want to.
- **Pull Request:** A way of submitting contributions to an open project. This is done by making changes on your fork and generating the pull request (PR).
- **.gitignore:** When you are managing a project with version control, there are going to be files you don’t want to “sync” to the repo. Rather than removing these files from the local repo we can tell GitHub to ignore them by extension!

Resources

Along with the recording of this class, there are some magnificent examples and resources available. Below we list these items for your review. In addition to these resources, the Github for this class contains video demos of all the processes outlined in the live presentation.

Markdown

Markdown is a lightweight and easy-to-use syntax for styling all forms of writing on the GitHub platform. The links below help you get started with markdown. (also, Slack uses some of these, so you are learning it for more than one reason!)

- <https://guides.github.com/features/mastering-markdown/>
- <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

Sample Repos

Here are some cool repos to check out as you dive into this further! (The Dynamo one is cool to check out because you can see what is coming soon if you dive deep enough)

- The repo for this class - <https://github.com/johnpierson/AU2018-GitItTogether>
- Hello world tutorial - <https://guides.github.com/activities/hello-world/>
- Dynamo Repo - <https://github.com/DynamoDS/Dynamo>

Popular Options for Using Git

Git is the software at the heart of many platforms. This section aims to give a summary of some of the most popular ones available.

Github (<https://github.com/>)

Arguably the most well-known of the bunch. Github started in 2007 and offers the ability to create as many public repos you would like. Github is a great option because they offer a magnificent desktop application which eliminates the need to mess with the command line. One of the most notable aspects of Github is, their terms of service page, <https://help.github.com/articles/github-terms-of-service/>. Within it they state that the materials you upload remain yours. Section D-3 of their TOS states, “You retain ownership of and responsibility for Your Content.” Pretty neat!

Pros:

Desktop App for Easy use

Cons:

Limited to public repos for free

Bitbucket (<https://bitbucket.org/product>)

Bitbucket is a cloud-based version control system from Atlassian. It is very similar to Github in the way that it provided this functionality using Git. With Bitbucket there are several different account times at different tiers. For their “free” account you are allowed 5 users and unlimited public and private repositories. Unfortunately, Bitbucket does not have a desktop application like Github does. However, if you are familiar with visual studio, their integration is very nice!

Pros:

Unlimited Public and Private Repos

Cons:

Lacks a desktop App for Managing Repos

GitLab (<https://about.gitlab.com/>)

GitLab is a platform that I am admittedly less familiar with. Their goal is to be a “one-stop shop” of sorts for the entire development lifecycle. A lot of times when you are just using version control with other platforms, you are working to tie it into other software as well. At GitLab, they seem to want to change that. Side note: The main reason I have ever heard of GitLab was the loads of users saying they were moving their projects to their platform when [Microsoft purchased Github](#).

Next Steps

If you are feeling more comfortable with the idea of working in a platform like Github, you can begin to contribute to other open source projects. A great one that was recently started is the [Dynamo Python Repo, by Sol Amour](#).

I also wanted to add that preparing an AU presentation using Github has been a joy. I was able to see my presentation as it grew and not worry about losing any work. Along with this functionality I was able to build a live example of how to do this stuff. In a way, I hope we can all begin to realize that Github or version control is not just for coders.

Conclusion

Well, there we have it. A virtual “bootcamp” if you will of getting started with version control. I hope that all this information helps you all along your professional career paths. By the time I successfully started using version control for the Rhythm Dynamo package, I had lost a few nodes and realized I was doing something wrong. I hope that you begin to implement some of this in your daily work and save time by working in an efficient manner. Thanks for checking out this class and for the support!



John Pierson

Design Technology Specialist

PARALLAX TEAM

johnPierson@parallaxTeam.com

sixtysecondrevit@gmail.com

@60secondrevit