1.0 Lesson - Github and Xcode Projects

Paul Solt - Paul@SuperEasyApps.com
SuperEasyApps.com

1.1 Lecture - Github and Xcode Projects

Github is a service that uses git to track changes to your digital files, like code, Xcode projects, user interface files, and images. It is useful for sharing code with others online, as well as providing a way to track working versions of your iPhone app.

- 1.1 Lecture - Github and Xcode Projects
- 1.1.1 Lesson PDF - Github and Xcode Projects
- 1.2 Tutorial - Create your Github account and Download the Mac App
- 1.3 Tutorial - Create and Download your Github Project
- 1.4 Tutorial - Make and Commit Changes with Github
- 1.5 Tutorial - Simple Github Workflow on Master
- 1.6 Tutorial - Git and Xcode Projects for Teams
- 1.7 Bug Fix - How to Undo Changes With Github
- 1.8 Tutorial - Complex Github Branch Workflow for Teams
- 1.9 Bug Fix - How to Resolve Conflicts with Xcode and Github
- 1.10 Homework - Share your Github Project
- 1.11 More Advanced Git and Github Resources

1.2 Tutorial - Create Your Github Account and Download the Mac App

You will create your personal Github account where you can share code with others and track your own personal projects.

Create a Github account and download the Mac app for github.

Links

- http://github.com
- Github Mac App

1.3 Tutorial - Create and Download Your Github Project

Make your Github projects online at <github.com> and download them to your Mac using the Github link from the website:

1. Create a new Github repository
2. Add a README (markdown)
3. Add a .gitignore file for Swift
4. Download your github project from github

Links

- http://github.com

1.4 Tutorial - Make and Commit Changes With Github

You will learn about making changes and tracking changes using the Github app.

1. Create a new Xcode project and save it in your Github project directory.
2. Add a print statement to ViewController.swift:

```
print("hello!")
```

3. Commit the change with Github app.
4. Open your LaunchScreen.storyboard.
5. Commit the changes with the Github app:
   1. Sometimes Xcode will update UI files when you look at them if they're an older version.

6. Open your Main.storyboard and add some UI.
7. Commit the changes.
8. Push your changes to your public code repository with the Sync button (auto-sync option).

Links

- https://github.com/PaulSolt/Github-Xcode-Project

1.5 Tutorial - Simple Github Workflow on Master

Github is easiest to learn when using your own projects, and before you join a group project. Since you are a beginner, I recommend the simple Github workflow until you become more comfortable with the git version control software.

1. Always sync first:
    1. Important in team projects or if you have multiple computers (laptop and desktop).
2. Make changes.
3. Commit often when you changed with descriptive messages.
4. Sync your commits to Github.

1.6 Tutorial - Git and Xcode Projects for Teams

When working on a team, you need to communicate about who is making changes to specific files. This is especially important with Xcode project file changes (i.e.: adding files, changing settings), and Storyboard user interface file changes:

1. To prevent merge conflicts, you need to pick one person at a time to make changes:
    1. Only one person is allowed to add files to a shared project (swift, storyboard, .xib, config files, images, etc.).
    2. Commit the changes.
    3. Inform the rest of the team to pull changes (sync).
2. You have to communicate via chat/text/email when things are changing so the rest of the team can get the latest Xcode project and files.
3. To prevent merge conflicts with your storyboard files, only one person should be editing them at a time.

1.7 Bug Fix - How to Undo Changes With Git

You can use git to undo changes before you sync them with your online code repository.

Github App

Disable Auto Sync with Commits
`Edit > Automatically Sync after Committing`

Undo Local Changes
1. Right click on file in current changes (before commit)
2. Choose `Discard Changes`

Revert Commit
1. Click on change (reverse order)
2. Click on Cog button
3. Revert this Commit

Terminal - Command Line (Use Spotlight)

1. Undo the last commit

   ```
   git revert HEAD
   ```

2. Undo changes to a single file

   ```
   git checkout filename
   ```

3. Discard all local changes (Warning: use sparingly–uncommitted work is deleted; prefer revert)

   ```
   git reset HEAD --hard
   ```

1.8 Tutorial - Github Branch Pull Request Workflow for Teams

In a bigger team environment, you should adopt the Github Branch Workflow using branches. This flow is more complex and requires more steps in a specific order:

1. Always sync first.
2. Create a new branch (no spaces in name).
3. Switch to the branch.
4. Make changes to your project.
5. Merge any changes from the master branch.
6. Create a `Pull request`.
7. On github.com, a project owner (you or one of your team members) needs to Merge the pull request (if there are no merge conflicts):
    1. Pull requests on Github enable discussion and your team can review your code before pulling in your changes.
    2. This is good for code quality, but it requires more steps and overhead.
    3. Communication is key to making progress.
    4. You still need to follow the rules about only one person modifying the Xcode project, Storyboard files, or adding new files to an Xcode project.

1.9 Bug Fix - How to Resolve Conflicts With Storyboard Files and Xcode

1. Make sure you resolve, commit, and sync your changes.
2. If you are on a team, have everyone else sync to pull the latest changes before making more changes.

Avoid Conflicts

Always Sync before making changes to the project, UI, or code.

Option 1: Revert Changes

Learn how to fix common problems when your code and storyboard files have conflicts. You can use both the Github and Xcode apps (source control view) to fix these issues:

1. Write down what you changed.
2. Revert the changes that you caused to conflict.
3. Sync.

4. Re-apply the changes from your notes.

Option 2: Merge Changes

1. Open Xcode and use the Versions Editor (Left/Right arrows).
2. Storyboard files must be opened as a code file to resolve conflicts:
    1. Right click Storyboard file
    2. Open As
    3. Source Code
3. Remove the conflicts between the markers:

```
>>>>>>> HEAD
// your current stuff here ...
======
// the other code from someone else ...
// pick one or cherry pick ... this get's super complex quickly
<<<<<< master
```

4. Add any removed code from the Current Revision to your BASE Revisions.
5. Open Terminal to mark the conflict as resolved and commit it.
    1. On Mac Github app right-click on the project name > Open in Terminal
6. Check the status of conflicts:

```
git status
```

7. Mark each file resolved by adding it back to git:

```
git add filepath/filename.storyboard
```

or add all files using the wild card symbol (easier):

```
git add *
```

8. Commit the changes:

```
git commit -m "Merged changes with XYZ and fixed issues"
```

9. Sync or create a Pull Request to finish.
0. Everyone on the team needs to do a Sync or pull to get the updated changes.

Links

- Undoing Git Changes - altassian.com

1.10 Homework - Share Your Github Project

Sharing your code via a Github project is the easiest way to share code with other people. They can always get your latest version instead of an old .zip file.

Homework

1. Create and share your Github Project:
    1. Copy your first iPhone app from Lesson 3 into your project folder.
    2. Commit it.
    3. Sync it.
2. Share your github account and project on the community forum.

Collaborate

1. Collaborating with another person on a shared github project:
    1. Add collaborators on the settings page.

Homework

- [Chapter 1] 1.10 Homework - Share Your Github Project

1.11 More Advanced Git and Github Resources

There are a lot of resources on git and Github. You now have a base level of understanding and can learn more about it using the following links:

1. https://help.github.com
2. Git Book
3. Git Cheat Sheet
4. GitHub Pull Request Flow
5. Try Git