

3.4.5 Merge and Switch Branches

When you're happy with the state of the game, save your progress with Git. You won't be doing any more work related to this branch, so go ahead and merge it as part of the Git process:

1. Use `git status` to verify that the correct files were modified.
2. Use `git add -A` or `git add .` to stage any changed files.
3. Use `git commit -m "random health and attack values"` to commit your changes.
4. Use `git push origin feature/random` to push the branch to GitHub.
5. Use `git checkout develop` to switch to the `develop` branch.
6. Use `git merge feature/random` to merge the new feature into the `develop` branch.
7. Use `git push origin develop` to push the updated `develop` branch to GitHub.

Close the corresponding GitHub issue and move on to the next one. We'll treat this next "code quality" issue as a feature, too, so the Git steps will be

the same as before:

1. Use the `git branch` command to verify that you're in the `develop` branch. If not, switch to `develop` before proceeding (e.g., `git checkout develop`).
2. Use the command `git checkout -b feature/objects` to create and switch to the new branch.

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.