3.3.5 Finalize the MVP

As tempting as it might be to jump right into the next set of features, we should wrap up the MVP we defined earlier. That means we're done with the `feature/initial-game` branch, so it's time to commit and push our changes:

1. Use `git status` and `git add` to stage the current changes.

2. Use `git commit -m "start and end game functions"` to commit these changes.

3. Use `git push origin feature/initial-game` to push the branch to GitHub.

4. Use `git checkout develop` to switch branches.

5. Use `git merge feature/initial-game` to merge the MVP into `develop`.

6. Use `git push origin develop` to push the updated version of `develop` to GitHub.

Keep in mind that the `develop` branch represents our development environment. If we wanted to deploy these changes to GitHub Pages, for example, we'd need to merge them into `master`. Now that we have an MVP, this is a good time to update `master`. This way, the judges will still have something to look at even if we don't finish any other features.

To update `master`, follow these Git commands:

```
git checkout master
git merge develop
git push origin master
```

Now close the corresponding GitHub issue and celebrate!

With the MVP done, we can start knocking out some of our feature improvements, starting with the shop. These nice-to-haves can be the most fun to build, but it was important to have the groundwork in place first. That's why GitHub issues are so valuable. They keep us on track so that we don't skip critical setup in favor of going straight to the "wow factor."

Because we're working on a new feature, our Git process should follow suit:

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/shop` to create and switch to a new branch.