Continuous Integration

yorkpirates2 (game) repository

Continuous integration occurs in two ways in our game repository

1. If a pull request is made, the `test workflow is run
   - This is to ensure that the act of merging the implementation branch (different for each part of the game being worked on) and the main branch, particularly the resolution of conflicts, has not caused any of the game to break.
2. If a pull request is **successfully** merged to main, the `test` , `build` and `documentation` work flows are all run.
   - This generates the test report, jar file and documentation pages to put on the website and the new build is put to main.

`test` workflow

1. Builds the project using gradle.
2. The workflow will test the project, but continue if it errors. This will generate a test report.
3. The workflow will then deploy the test report to the website2 repository.
4. Finally, the workflow will run the tests again, but fail upon error.

This ensures that if there are any errors in testing, it still generates a test report for us to see, and also alerts us to the error during the pull request. This means that we can then fix the issue before the game is published to main

`build` workflow

1. **Builds** the project using Gradle
2. Uses the dist command to **generate a jar** file
3. **Deploys** the jar file to the website2 repository

This allows us to automatically place the latest jar onto the website.

`documentation`

1. **Builds** the project using Gradle.
2. Uses the javadoc command to **generate** documentation as html.
3. Then **deploys** the documentation to the website2 repository.

This means that all of the javaDocs in our code are viewable on the website to allow for future developers to inspect but also to allow the rest of the team to view them easily

documentation2 repository

If a pull request or push to the main branch in the documentation repository is made, `deploy` is ran.

`deploy` workflow

Deploys the markdown files to the website2 repository. This allows us to keep the websites documentation files up to date.

website2 repository

All continuous integration in this repository occurs on a push to the main branch. `pages-build-deployment` will always be ran, however `convert` will only be ran if the push includes markdown files in the `/markdowns` directory.

`convert` workflow

1. Converts all of the files in the `/markdowns` directory into pdf files.
2. Commits the pdf files to the `/pdfs` directory in the repository.

This allows our documentation to be presented in both markdown and pdf format.

`pages-build-deployment` workflow

1. Builds the jekyll website.
2. Deploys the built site to GitHub pages.

This allows our website to constantly be up to date with all the latest files.