CI2

Enigma / Team 29

Michael Beal
Lucas Solomon
Thomas Auburn
Ryan Early
Inna Strogonova
Daniel Pearce

Summary

For our continuous integration, we forked group 32's game repository on GitHub. This preserved their git repo's history and the contributors as well, allowing us to seamlessly continue with their work. In addition, we decided to continue using the previous group's GitHub workflow for our continuous integration with some minor modifications. everytime a push was made to a branch, the workflow tested to see if the project would build correctly on a virtual machine hosted by github. When other tests were created and added (such as unit tests), those too were run automatically each time if possible.

As part of the continuous integration, it was required for members of the team involved in coding to (ideally) upload their changes as soon as possible, rather than working locally and not committing anything to the repository for long periods of time. This was so that everyone had access to an up to date version of the repository and ensured there wouldn't be as many conflicts while coding.

A record of all commits and results of the build attempts and tests are on github in the actions tab, which can allow for an overview of progress on the project as well as to shorten down what possible changes are causing problems when they arise. A badge has also been added to the README to quickly and easily see if the current build passed or if there's an issue that requires further attention.

This approach was decided as the previous group's system worked fine and there was no point completely redoing the workflow when it could simply be expanded upon when and where necessary. It gave a quick indication if there were any problems when commiting and what could be the potential issue(s) if anything went wrong. It was decided to create the continuous integration solely in GitHub without 3rd party tools to keep everything centralised and simple to use and edit.

Report

The workflow for the project is located in .github/workflows as gradle.yml. Whenever a push or pull is made to one of the specified branches in the file, the workflow is run. When new branches are made, they are added to the push list.

```yaml
on:
  push:
    branches: [ main, mostly_broken, changing_saving, CI, remaking_tutorial]
  pull_request:
    branches: [ main]
```

The workflow builds the product on a virtual machine (in our case one running ubuntu) hosted by GitHub. After setting up java and gradle execute permissions, the virtual machine attempts to build the game, run the tests and then will publish the results of said tests. The publishing of the tests was done using the “Publish Unit Test Results” GitHub action.

```yaml
- name: Publish Unit Test Results
  uses: EnricoMi/publish-unit-test-result-action@v1.8
  if: always()
  with:
    files: tests/build/test-results/**/*.xml
```

The result of the last commit (and therefore the current state of the branch) can be seen from the badge in the README. More specifics on the result and results of previous commits or those of other branches can be seen in the actions tab. Clicking on a specific workflow run will allow the user to see the result of each step of the workflow run as well as the results of the tests, and which tests have failed.