Continuous Integration

Group Number: 10

Team Name: Decassociation

Group Member Names:
Tom Broadbent
Poppy Fynes
Owen Lister
Michael Marples
Lucy Walsh

Continuous Integration Methodology

- We will make use of different branches to ensure that the commit history is kept clean and easy to understand.
    - This allows us to work on new features separately and helps keep the master branch free of bugs
    - This also allows us to go back and edit commits if needed, or revert particular branch workflows, rather than having to rollback other work at the same time.
- We will also implement different workflows and releases
- Our README will include several badges/shields that provide information for our CI/CD; such as test passing status and code coverage as well as commit history and licensing.
- By having shields, we can quickly establish the current state of CI on the repository, what other members have been working on(and how much), as well as any issues that have been raised.
- Each feature to be added or change, will have been delegated to a subsect of the group, allowing us to parallelise our workflows
    - This allows for verification as well as enabling us to add or remove members to tasks, to ensure work is completed on time and to the expected standard.

Continuous Integration Infrastructure

- For our continuous integration, we have used primarily GitHub actions.
- Using this, we have implemented multiple workflow pipelines, including development as well as website updates and also testing.
    - Development has its own branch, and allows us to track the progress of new game commits and raise issues separately to the website
        - Building the game at each change to master will highlight any issues preventing the game from building which we can address sooner
        - Automatically building the game on tagged releases saves us time from having to manually run the commands to build and upload the files.
    - The website also has its own workflow pipeline, allowing us to update the website and track commit progress separately.
    - Testing workflows will ensure that all code meets the required standard, so that once the push is complete, the game is downloadable and runs
- We have also implemented unit testing, which is what is deployed in the testing workflow. By creating tests we ensure that our code is functional and meets our predetermined standards, before being approved for release.
- To enable clear development pipelines and workflows, we have also implemented a variety of branches
    - Branching allowed us to separate our testing development from the main game for example, allowing our master branch to stay functional and to avoid any messy merges that may come as a result of not using branches
    - We have also implemented branch protection rules to keep master clean from bugs