Continuous Integration

Team 24

Joe Cambridge
Joss Davis
Emily Dennison
Louise Evenden
Amber Hemsley
Josh McWilliam
Lianyu Zhao

Methods and approaches

Frequent integration
We decided to continuously integrate the code with the main branch in gitHub. This helps prevent merge conflicts as well as making it easier for people to work simultaneously with other members of the team due to small sections of code being pushed frequently. It also helps everyone working on the code keep track of what has been done and what each person is working on.

Prioritising errors
In order to prevent errors being lost amongst other code we decided to prioritise fixing them as soon as they came up even if that means delaying other code sections. This makes it easier to understand what caused the error so it can be dealt with quickly. It also makes sure the code always runs, preventing testing delays.

Code styles and conventions
Our implementation team followed the code style created by the previous team. This was done to ensure that code was consistent and followed the same naming conventions. This improves readability of the code which helps to ensure that other team members could take over the code easily and understand the structure.

Continuous frequent testing
We decided to write unit tests intermittently as we added new features to the game. This allows us to run our tests on every change as they are pushed to make sure that no new features affect any of the previously implemented features. It also helps to keep track of which sections of the code have been tested to ensure maximum coverage. Testing features using automated tests as they are added also helps to make sure they work as expected in all cases avoiding human error if we were to test the features manually. It also makes it a lot faster than repeating manual tests. We thought this was appropriate because each member is then able to create new and useful pieces of code knowing that there were no bugs created.

Regular task assignment
We decided to review task assignments at each meeting to make sure we all know who is working on each section. This will prevent crossover and help avoid merge conflicts as each person is working on something different.

Continuous Integration Infrastructure

Frequent integration
We developed on a new branch, core-dev_1. During development, the members would frequently push their commits onto this development branch. Once the feature that they implemented was completed and tested, the development branch was merged with the main branch. Merge conflicts were then resolved in development branches and the tests were rerun again in order to make sure that no existing tests were broken. After completing this, the development branch was merged again with the main branch and was considered successfully implemented.

Prioritising errors
We made sure not to push code containing errors wherever possible meaning most issues were fixed before being added to the main branch.

Code styles and conventions
Variables and functions were named using camelcase where the first letter is lowercase and classes were named in camelcase with a capital letter at the beginning. This follows java convention as well as staying consistent with the previous team's work.

Continuous frequent testing
Unit tests were implemented alongside the main code and we made sure the tests covered all the major features added. We used a testing document to keep track of the tests we needed and marked them complete as they were added. This helped make sure we didn't miss anything in our testing as we aimed for as much test coverage as possible. We made sure to run all the tests each time code changes were pushed to the main branch.

Regular task assignment
Minutes taken at each meeting included notes on what each of us had completed since the last meeting and what we were assigned next. These documents were all on our shared drive so we could easily check what everyone was working on. Regular meetings also allowed us to reassign sections that people were struggling on to keep up progress. We also kept in contact between meetings allowing us to update assignments at any time.