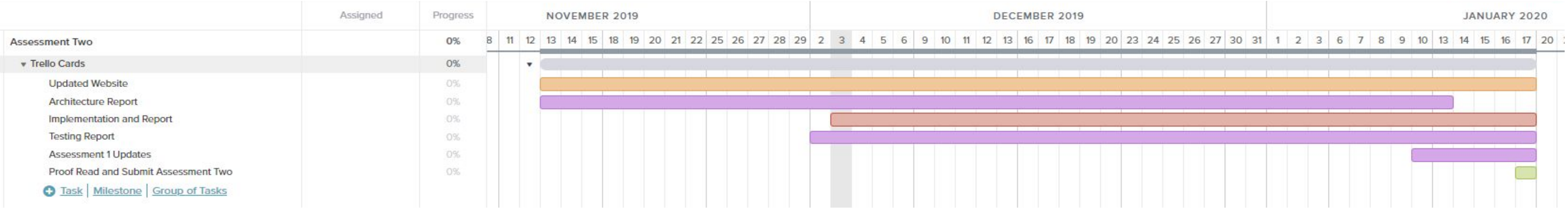


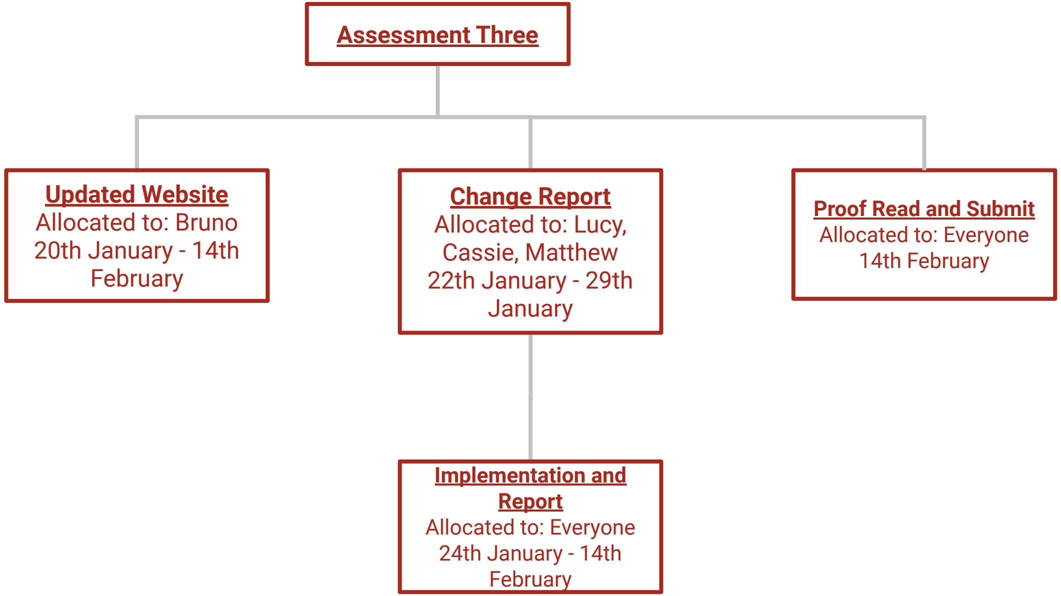
	Assigned	Progress
Assessment Two		0%
▾ Trello Cards		0%
Updated Website		0%
Architecture Report		0%
Implementation and Report		0%
Testing Report		0%
Assessment 1 Updates		0%
Proof Read and Submit Assessment Two		0%

⊕ Task | Milestone | Group of Tasks

Originally we had planned to try and get the implementation done before we all went home for Christmas. This is because implementation is a lot easier when we can all meet in person and work as a team. It is a lot harder to understand other people’s code and develop a game together effectively when we can’t meet up. As the meetings went on, it seemed as though we wouldn’t be able to achieve this. This is because we had to spend a lot of time learning new things such as using git properly learning the capabilities of our chosen game engine. We have therefore extended the finish date for the implementation to after we come back in January. This allows us to work on the development over Christmas, and hopefully finish it. The finish date for implementation is now the 13th as this means we still have a week before the deadline to go over the code each of us have done and debug and refactor it if needed.

The date to start the testing report has also been moved to be earlier. This is because we realised we could start to plan how we are going to test the game before it is finished being implemented. We also realised it is important to write tests as we are implementing, to make sure we cover everything. If we waited till the implementation was finished for this sprint before we started writing tests then there is a chance we could miss something and therefore leave some bugs in our game.

The start date for the assessment one updates document has also been brought sooner because we realised we could start to make notes of any changes that are made throughout the whole assessment window. It makes it easier to remember to include everything, and also means the workload we have after the implementation is reduced. As we only have a small amount of time when returning from Christmas break before the deadline, we don’t want to have a large amount of work to complete, therefore we decided to start this now and work on it throughout.


Assessment Three
Updated Website
Allocated to: Bruno
20th January - 14th February
Change Report
Allocated to: Lucy, Cassie, Matthew
22th January - 29th January
Proof Read and Submit
Allocated to: Everyone
14th February
Implementation and Report
Allocated to: Everyone
24th January - 14th February

The updated website branch will consist of making sure the links on our website show the most recent stuff. This includes the game, the testing material, all out documentation and the user manual.

The change report will include our explanation of our approach to change management as well as any changes we have made to the product, with justification.

We will explain how we have met requirements and how we have implemented our architecture in the implementation report, as well as continuing with development and implementing the game further.

Starting the second assessment, we thought it would be beneficial to review the tools we were using now that is was time to start implementation. Originally we had decided to use Trello for our organisation of tasks as well as for our implementation backlog. Upon planning the backlog, we discovered a tool called GitKraken. This is a tool similar to Trello however it is integrated with GitHub. We looked into it and noticed that the Glo Boards had a nearly identical concept and interface as Trello, but with the addition that any lists and cards created would be added and synced with the issues board on GitHub. We also noticed we could assign GitHub users to the tasks. This seemed very ideal and efficient, due to these features, and therefore we decided to use this for our implementation rather than Trello. We created lists for Backlog, In Progress and Completed, as we had originally done on Trello. We are continuing to use Trello for the general organisation of documentation, as the lists were already created in great detail and we had made use of the feature of TeamGantt.

Another new idea we came up with was creating a new repository on GitHub which would act as a test repository. We thought this would be a good idea to utilise whilst we all learnt not only how to develop a game in Java but also how to use git effectively. We didn't want any errors caused by our learning process to destroy the game we had created so far. We therefore added a list to GitKraken, and made it clear which tasks were being implemented on the test repository and which were on the final repository for the game. As things become fully implemented and working on test, we transfer them over to the final one.

Thirdly, something which isn't an entire change nor a new addition, we have all decided to use IntelliJ as our IDE. Originally we were all going to use our preferred one, whether they were the same or different to each other, however as we were setting the project up we found it was easier for us all to use the same one so that we could all help each other out if we ran into issues. IntelliJ was the easiest one for us to set the project up in as it was fairly intuitive and simple to use. It also has a lot of efficient tools built in, such as generating constructors, getters and setters automatically, which makes our lives easier as programmers.

Previously we said that we had a minimum of two group meetings per week. Due to the larger workload in the implementation stage of the project, and the fact that we can't have face to face meetings over the Christmas period, we have decided to increase that number to three. Three meetings a week, minimum, means that we can get a lot of the implementation done effectively as a team while we can still meet up in person. This will be a harder task when we are all at home for Christmas as we all programs in very different ways and we will all have individual plans for the festivities. We don't have a large amount of time between coming back in January and the deadline for the assessment, so we wanted to get a very large percentage of the programming done before we all leave, for these reasons.