One success we had was that the initial user interface was done very quickly, which allowed us to test all our features thoroughly as they were implemented. The images, and all other files following this except Java and FXML, were also split up into appropriately named folders. This made finding the needed files to edit much easier while implementing, and kept the directories tidier.

Sourcetree acted as a major drawback during the project, having clashes automatically fixed in an unsuitable way that we had to manually revert. It also caused data to be lost or not shown to some individuals during pushes.
There was also a long-lasting bug linked to placing the appropriate token doors that wasn't anticipated to require as much work as it initially did.

Ben Parker
Ben helped with the general organisation of the project, creating lists of jobs and their dependencies, as well as aiding in the design of the codes structure. He split global files from the source code, and created a custom files folder within that to store the custom levels. He also removed duplicate images from the new Images folder, and went through all source code amending this. He created player folders, and set it up to automatically create a uniquely named folder for every user profile to hold their saves in. He went on to set up a hashing system for the files, allowing them to have a user-inputted name visually and for file names, while also having unique level numbers for backend operations. He made Javadoc for many methods, and helped reduce redundancy, inefficiency and errors in several other parts of code throughout. He modified GameControllers map saving methods to allow them to be used for the editor, and converted image urls to two-character map representations. He went through the contributions report, fixing grammatical errors and improving sentence structures, before adding the successes and drawbacks. He also helped with some parts of the video plan to ensure everything was shown.

Chester Descallar

Chester helped in both back-end and front-end of the implementation. To begin with, he helped hugely in implementing parts of the LevelEditor class (front end), allowing for image views (the sprites that we have in the game) to be clicked, dragged, and dropped onto the gridpane built by Joseph. This means that a level can easily be created or an existing level can easily be edited. With this implementation - the rest of the group were able to add their pieces of code to the class to allow for error handling, such as Sean's wall enemy error handling and Petko's token door functionality. Chester also helped Joseph with creating other classes and methods to be able to save the created into a text file, by creating a SavedImage class which stores variables that are used to convert ImageViews to have their corresponding names (as opposed to their object names). Chester then helped the group during the last week, with the back-end of things such as ensuring that all teleporters were written correctly to the text files when a new level is created (ensuring that the links are correct) and fixing small errors and bugs. Chester communicated with everyone, ensuring that the group work was going smoothly, helping with the minutes along the way.

Hossein Mohammadi

Hossein contributed as a part doing Javadoc and enemy constructors. However, other group members contributed well and many necessary parts were covered. He tried to understand and participate in each part of group work, and helped as much as he could. Also, he tried to do some extra features to improve game modality. And finally, he checked if there are any forgotten parts or lingering mistakes to be resolved, such as coding conventions.

Marcin Kapcia

Marcin Kapcia set up contributed in meetings, alongside contributing to the design and implementation of the project. He kept track of the minutes and the contribution report during meetings. He aided the project progression further week by week, and recorded the meetings. During the implementation, Marcin focused on being able to resize the gridpane with a scrollbar, changing it from a gridpane to be inside a scrollpane. This made it become pannable with the objects inside. From there he moved over to loading existing files into map editor with Joseph. He created the FXML that allows users to choose between a level map or custom map for editing. He made sure the files were loaded correctly with correct attributes and were able to be saved, intracted, edited and played. He made sure what other people implementations worked with loading custom files to be edited. Marcin also has improved the visuals and the navigation of the application, as some FXMLs were not user friendly. Marcin tried to aid any issues people had and gave suggestions on how to resolve them, constantly being able to be contacted. With the design, Marcin wrote JavaDoc of methods and classes, and looked through the previous JavaDoc to see if any methods/classes were missing or needed editing.

Sean Coaker

Sean threw himself into the work from the start, by firstly getting to grips with the group's code and game (as he had joined from a different group) and tried to help the group as much as he could. He started by working on a delete function where he designed an FXML file and java class which allowed a user to delete a custom map which had been created. After that, he was assigned the task of writing code that when a user placed a Wall Enemy in the level editor, would check to see if the Wall Enemy had been placed next to a wall, then gave the user a Combo Box of options as starting directions for the user to choose. He then worked on a way to keep the integrity of this data, ensuring that all enemies' starting directions and coordinates were being saved to the file correctly and were being loaded in from the file correctly when editing the level. Sean then worked hard in the last week to fix any bugs that were arising in the code. Specifically, he was ensuring that custom levels could be played properly, cells on the map worked properly when being loaded in from custom levels, custom levels could be loaded and edited without needing a naming standard for user ease and that the user could save their progress when playing a custom map. He also tried to keep everyone motivated in the last week to try and get the work completed a couple of days before deadline to be able to give the group sufficient time to create a good video and polish up on the documentation. This included listing tasks that needed to be finished and setting timescales for getting the tasks completed. Sean finally helped with the motion side of the video, demonstrating how different functions in our implementation worked and tried to show them as clearly as possible.

Khaled Alamri
Khaled has worked on improving the default layout of the custom map when first created, by adding a border of wall cells by default, which scales with the desired map size. He tried to help with any issues that occurred in the implementation process by pointing to bugs that occurred, or by giving suggestions on how to resolve them. In the end, He also worked on polishing the Javadoc and adding missing comments in the code, and finally he worked on the video demo demonstrating the group's work.

Petko Kuzmanov
Petko has worked on every aspect of the assignment. He gave ideas about what menus were needed and how the overall interface would look. First, he divided the cells on the side into categories and added a menu for the size and title of new custom maps. Then he worked mostly on the cells that needed more than just to be clicked and placed. For example, he made the option for the user to choose how many tokens the token door requires and an option for the player to choose the direction of the straight line enemy. Also, he devised a way to link teleports. Furthermore, he thought of a way to limit the custom maps to having only one player cell. He went on from this to make the menu that allows the custom maps to be played, which helped us tremendously with the testing process. Throughout the assignment, he was looking for bugs and redundancies in the code.

Joseph Semgalawe
Joseph contributed to every aspect of the assignment. He made a sample design using fxml in scene builder to figure out how it would initially look. He then worked with Chester on figuring out how the drag and drop functionality would work. He presented what he had to the group to stir the initial development of the level editor. After his first presentation, he made some changes to the level editor sample to make the gridpane resize with the clicks of buttons(for initial testing) . In addition to that, he made the default map for a custom level to initially load up with preset ground images. He then made further implementations and refactoring to the event handling used to add and replace images on the gridpane. After his event handling implementation, he started work on saving the custom levels, with the help of Ben’s code as a template. Alongside Chester, he made additional changes to its functionality that helped with making the save work with slight ease(with the help of the custom SavedImage class that extended Images). He additionally worked with Marcin on the load functionality of the level editor, simultaneously working on the drag and drop functionality to work with files that are loaded - like border walls around the map being able to be replaced when a file is loaded. During the final stretch of the project, he assisted with the video demonstrating the group's work, concluding his contribution.