Methods and Planning 2

(Part of *Change2.pdf*)

Cohort 1, Team 4:

Robert Watts
Adam Wiegand
Josh Hall
Travis Gaffney
Xiaoyu Zou
Bogdan Lescinschi

Software Engineering Methods

We will continue to use the same methods used in the first part of the project, as these methods proved to be very effective.

Paired programming allowed multiple team members to contribute to the project, allowing us to complete the project before the deadline.

Using object orientated programming allowed team members to effectively add to other members work, by making it easier to understand the previous code.

Development and collaboration tools

We will continue to use the same tools used in the first part of the project.

Discord made team communications, live team chats, and material sharing easy. Also, the integration of GitHub allowed automatic notifications for changes to the repository.

GitHub made integrating changes to the game to be easily uploaded to the repository, and was usable in a code editor when used on desktop.

As the map has already been created, we have no further need for Tiled Map Editor.

We will continue to use Libgdx as it was also used by team 3^3.

We will continue to use Java as it was also used by team 3^3.

We will continue to use Google Docs as it allowed us to store all of our written documents and diagrams in a communal storage area. It will also allow us easy access to the deliverables from the first part of the project.

We will continue to use Photoshop as it was easy to use, and have already paid to use it.

We additionally used GDXpacker to change the texture atlas the game uses so we could update the game’s sprites.

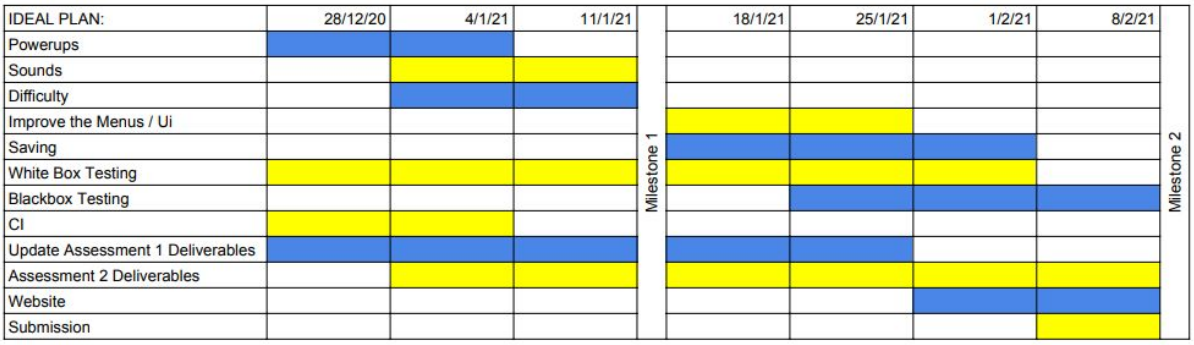
We will continue to have weekly meetings as scheduled by the computer science department, but we will have extra meetings allowing us to more often update each other on our progress. As we already know the strengths of the team members we will keep the team split of 3 groups of 2 the same, allowing us to properly complete the new tasks. As the new tasks require new and more complicated code, it is likely that some tasks will need to be considered by more team members in order to complete them, resulting in more inter-group collaboration. Continuing to be critical of each other's work will continue to allow us to make a more meaningful project, and allow everyone to collaborate on as much of the project as they can.

For the first part of the project, we followed a linear plan which gave every team member different tasks as we progressed through the project, assigned based on each team member's individual skills. This was more efficient than splitting each section equally between each member as not everyone had the skills for every task. If a team member couldn't work on any task from each section, they would go back and clean up the work from the previous section. This method allowed us to make steady progress through the project, and it allowed us more time to recheck our work to ensure we hadn't made any mistakes. We will continue to use this method in the second part of the project.

We will continue to use the same critical path: Requirements → Design → Implementation → Documentation → Testing / Feedback. Using this model ensured that we didn't start working on a new section without finishing the previous one, which ensures that we don't focus too much time on 1 part of the project.

We will continue to use a Gantt chart to plan responsibilities. It was easy to modify and read, and we were able to update it every week to keep the team up to date on the progress of all of our tasks.

The final plan for the project was split so that the majority of our time would be spent on implementation, as the project was quite ambitious. Requirements, design, and testing were given a week each, and Implementation was given about 3 weeks. While the sections turned to take a day to two more than we anticipated, it didn't have a drastic effect on our schedule. As for the second part of the project, we don't need to create the game from scratch, we will likely not need to spend as much time on the implementation. This will allow us to give each section of the project two weeks, which will give us extra time if some sections take longer, and more time at the end to go over our work. This will also allow us extra time to familiarise ourselves with the other teams code and documentation.



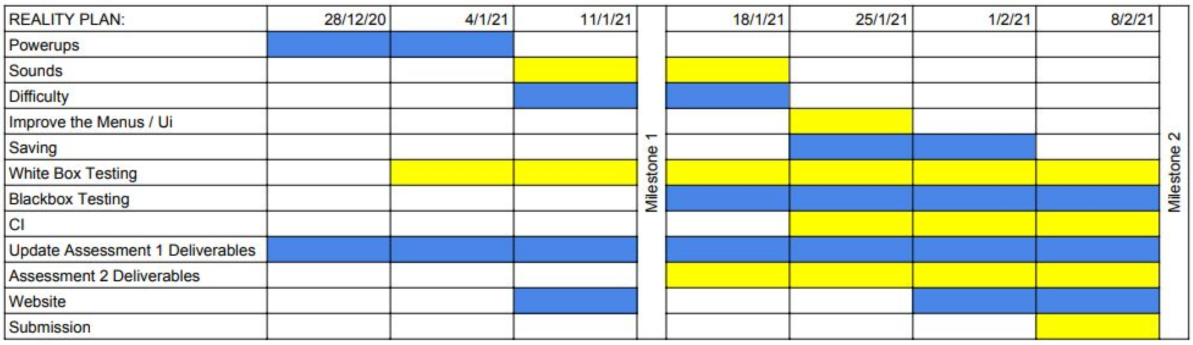
IDEAL PLAN:	28/12/20	4/1/21	11/1/21		18/1/21	25/1/21	1/2/21	8/2/21	
Powerups	🟦	🟦		Milestone 1					Milestone 2
Sounds		🟨	🟨						
Difficulty		🟦	🟦						
Improve the Menus / Ui					🟨	🟨			
Saving					🟦	🟦	🟦		
White Box Testing	🟨	🟨	🟨		🟨	🟨	🟨		
Blackbox Testing						🟦	🟦	🟦	
CI	🟨	🟨							
Update Assessment 1 Deliverables	🟦	🟦	🟦		🟦	🟦			
Assessment 2 Deliverables		🟨	🟨		🟨	🟨	🟨	🟨	
Website							🟦	🟦	
Submission								🟨	

In reality we weren't able to follow this plan exactly due to difficulties with various sections.
Testing was much harder to set up then we expected, delaying the writing of tests and setting up CI (which required testing to be in place).

CI took longer to implement as we encountered difficulties in setting it up correctly, we initially did it incorrectly and had to redo it.

Getting difficulty and sound working didn't take longer than expected, but we started on them later than expected (mainly due to delays in testing and updating deliverables).
UI and menu improvements took less time than expected, as once we had changed one part we knew how the menu/Ui worked and could thus add subsequent parts much quicker than expected.

We expected implementing saving would be difficult and thus time consuming, while implementing it was difficult we managed to complete it in two weeks, allowing us to catch up with the progress we expected to make on coding changes in our ideal plan.
We did work on the website earlier then we planned to, this is because the dependencies of over sections weren't complete, and thus those sections couldn't be worked on. So while waiting for the dependencies to be completed some effort was spent on the website.



REALITY PLAN:	28/12/20	4/1/21	11/1/21	Milestone 1	18/1/21	25/1/21	1/2/21	8/2/21	Milestone 2
Powerups	🟦	🟦							
Sounds			🟨		🟨				
Difficulty			🟦		🟦				
Improve the Menus / Ui						🟨			
Saving						🟦	🟦		
White Box Testing		🟨	🟨		🟨	🟨	🟨	🟨	
Blackbox Testing					🟦	🟦	🟦	🟦	
CI						🟨	🟨	🟨	
Update Assessment 1 Deliverables	🟦	🟦	🟦		🟦	🟦	🟦	🟦	
Assessment 2 Deliverables					🟨	🟨	🟨	🟨	
Website			🟦				🟦	🟦	
Submission								🟨	