Method Selection and Planning

Group 27 BlackCatStudios
Jack Vickers, Hubert Solecki, Jack Hinton, Sam Toner, Felix Seanor, Azzam Amirul Bahri

Taking over from Group 30 Triple 10
Kelvin Chen, Amy Cross, Amber Gange, Robin Graham, Riko Puusepp, Labib Zabaneh

Our Chosen Software Engineering Methods and Tools

Software Development Approach

Triple 10 followed a waterfall model, due to the clearly defined steps that ensured they fully completed each phase and had a clear end goal that they could work towards. Being the requirements specified in assessment 1's criteria. BlackCatStudios have decided to modify this to what we have used and experienced in assessment 1; we will be taking over the project and implementing the spiral software development approach. We decided to move away from the Waterfall Model since it didn't provide us with the flexibility that we desired with our project. In the previous assessment, we were happy with how the Spiral Method provided us the flexibility to jump between tasks and work packages, dealing with both risks and deadlines efficiently. We again took a risk-driven approach that had iterations in its scrum sprints between final prototypes that allowed us to easily implement continuous integration and versions as well as practice better teamwork through the scrum method. This will allow us to complete stages of the project and its deliverables in stages/sprints, allowing us to jump between work packages and communicate effectively. This would allow us to also complete individual sprints within our development groups where necessary to provide a more flexible approach to the project's requirements and work packages. Our first iterations consisted of understanding the code base fully ensuring that we could accurately and reliably lay out our requirements for the next assessment and development phase while improvements to the documentation of the previous assessment were being improved. This included the identification and mitigation of existing and new risks to the assessment that were again planned and accounted for. This was prior to implementing testing environments and continuous integration.

Software Engineering Methods

BlackCat Studios intends to change the working method, instead using the scrum method. It was unclear what method Triple10 was using since it was not stated in their documentation, but we assumed that they used the Scrum Method based on cohort conversations. We're choosing this method and model because it worked well in our previous assessment and allowed us to get a lot of work done in a short period of time while letting people in the project know what's happening in other segments. This time we are planning on splitting the team into two "dev teams". One team will be focusing on initially overhauling the code base and implementing new features. The other team will initially focus on updating the documentation, setting up unit tests and implementing continuous integration. This team will create unit tests for the assessment 1 content and beyond as the implementation team develops new features. We have regular team meetings to see how other members are doing and within each dev team. This allows us to do weekly sprints of work, allowing us to maintain progress while remaining flexible between tasks and group member requirements and schedules. BlackCatStudios also considered different methodologies such as Extreme Programming and Crystal. We acknowledged that even though we were going to use the Scrum Methodology, Extreme Programming would also be extremely useful later in the project if we were short of time in terms of testing and development. Its test-driven approach would be particularly useful in the closing stages of the project where if bugs were found they could be easily fixed and code could be adjusted and implemented rapidly. If any risks were encountered or we were short on time, BlackCatStudios was prepared to switch to Extreme Programming in case of an emergency. When we considered the Crystal Methodology, we understood that each adaptation of the model is bespoke to the specific project. While this would allow us to adapt the model specifically to our needs for the assessment, we did not want to waste time adapting and developing a method for our team, we did not want to waste time completing this process when we already had experience using scrums and enjoyed the flexibility of the process that suited our team perfectly anyway. As a result, even though we had considered using different methodologies, we decided to stick with our well-known scrum method to continue our efficiency and workflow.

Development, Collaboration Tools and Software Selection

Triple10 used GitHub to implement the project as it allowed for easy collaboration between team members and provides a secure backup of their files rather than a personal machine. It is also a well-known system, so documentation of features can easily be found.
Their team has also used the more general collaboration tools of discord and google drive to converse and collaborate on the documentation. These programs are designed to aid collaboration between group members through group chats/calls and shared folders that each group member can edit.

By using these different applications, it allows Triple10 to follow the waterfall methodology by keeping each phase separated (for example, they exclusively uploaded files involved in the implementation to GitHub and had individual files for each stage in google drive) Some alternative applications they considered were GitLab, Bitbucket, and Microsoft word; however, they felt that they could find better support for GitHub and that collaboration would be easier in google docs than in Microsoft.

BlackCat Studios intends to keep using the methods Triple10 used during their project for consistency and that it closely lines up to how they worked in the past.

While BlackCatStudios will be using similar tools as Team Triple10. We wanted to highlight alternatives that could have been considered and mention why the current ones are the best for the job.

- We considered using the Eclipse IDE as it is widely used in industry and by developers. However we decided to use IntelliJ IDEA because of its strong java syntax highlighting and intelligent systems. Allowing us to debug and adapt faster. It is also lightweight and has superb git features.
- JMonkey engine was a contender. However it doesn't have a strong community behind it and the documentation wasn't very structured. Making LibGDX the clear choice due to being used in many games and having good documentation.
- We chose github pages to host our public face of our game. This is because it can be hosted on the same repository as our game.

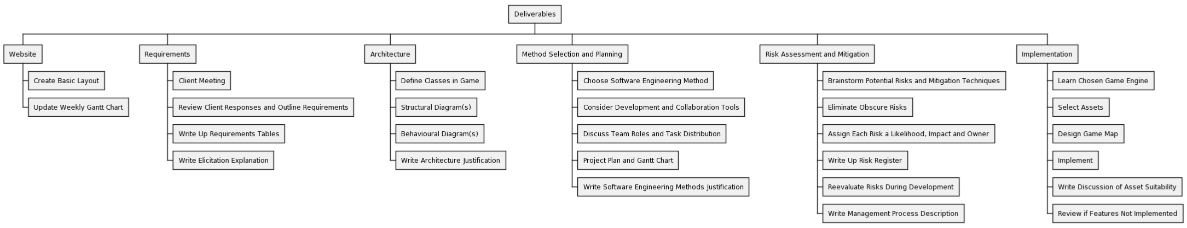
- We chose to use MKDocs for designing the website as its fast and easy to use, while creating professional looking websites. The alternative was using HTML however none of our group members were proficient at this so it didn't seem worth spending time learning, It would also take far longer to produce the same quality as MKDocs. While the previous team elected to do this, we instead converted their files to markdown and used MKDocs to make website development easier. We continued to use their css stylesheet so that the website kept a similar look and feel.
- The most up-to-date markdown documents for the website are stored in the docs folder of the main branch. When the MKDocs files are built, the build files are pushed to the gh-pages branch. This is the branch which the website is deployed from.
- We continue to use PlantUML.com for the creation of sequence, state, class, package and use case diagrams. We noticed that Triple10 also used PlantUML consistently and since we were comfortable with this approach we decided to continue to adopt it into this extended project also. As well as this, for CRC cards, we continued to use Echeung.me since we are familiar with the website through our experience with assessment 1. Triple10 did not have many/any CRC cards in their assessment; since we found them useful to build from in our first assessment we decided to again use the website to help us in our development, especially in the implementation of new features and understanding the taken over code base. BlackCatStudios considered using the website draw.io to carry out all these diagrams as it was easy to use to draw shapes and flow diagrams; however, it lacked automatic layouts which PlantUML.com surpassed it in. We also wanted to keep the same format we had as with Triple10's documentation so we decided to move forward with these websites.
- For communication, BlackCatStudios used Discord since it was what we used in our previous assessment. It is unclear what Triple10 used, but we are assuming that they followed this approach also from general conversations around the cohort. We considered different software such as Slack and social media, however, we found that Discord was the perfect middle-ground between a strictly professional-seeming platform and pure social media. It allowed us to isolate our work in the form of different channels in servers and communicate effectively, without it feeling too formal.
- The game was developed as in the previous assessment on Windows laptops. BlackCatStudios assumed that this was also the case for Triple10 due to the general preferences of the cohort. BlackCatStudios also made sure that even though we were developing on Windows, that we would test and ensure that the game runs on MacOS and Linux systems also; this was mentioned in our requirements as necessary.

Approach to team organisation

Team Triple 10

To organise their team, Triple10 created a Gantt chart to plan their project and when they would focus on each development section. They also created a Team meeting logbook to record which deliverables each team member worked on that week so that the group shared the work evenly. The risks that were outlined were also shared between team members so that they could fully monitor for any problems in our project. They would often have external team meetings to ensure they kept in line with the Gantt chart and collaborated through any issues.

Below is a work breakdown diagram splitting up the main sections of the first part to the project:


Deliverables
Website
Create Basic Layout
Update Weekly Gantt Chart
Requirements
Client Meeting
Review Client Responses and Outline Requirements
Write Up Requirements Tables
Write Elicitation Explanation
Architecture
Define Classes in Game
Structural Diagram(s)
Behavioural Diagram(s)
Write Architecture Justification
Method Selection and Planning
Choose Software Engineering Method
Consider Development and Collaboration Tools
Discuss Team Roles and Task Distribution
Project Plan and Gantt Chart
Write Software Engineering Methods Justification
Risk Assessment and Mitigation
Brainstorm Potential Risks and Mitigation Techniques
Eliminate Obscure Risks
Assign Each Risk a Likelihood, Impact and Owner
Write Up Risk Register
Reevaluate Risks During Development
Write Management Process Description
Implementation
Learn Chosen Game Engine
Select Assets
Design Game Map
Implement
Write Discussion of Asset Suitability
Review if Features Not Implemented

Team BlackCatStudios

BlackCat Studios will be following a similar path, creating Gantt charts and updating them weekly to adjust to any changes in the schedule. Regular in person meetings will also be held, to cover any issues members might have with the work, update the website and delegate work to members. We have also created a minutes document to keep track of what has been done in each of these meetings and which members were present. Risks have also been outlined and shared, with responsibility to prevent/fix issues designated to each member.

We nominated Hubert as the team leader again due to him being an effective leader in the last assessment and having a clear vision on how to finish this project in time. Felix was elected as the development lead due to his advanced experience in Java and constructing game engines. He would be able to use this experience to vet code and spot errors quickly in code. Jack Vickers was nominated as the testing lead as he showed enthusiasm towards it but also was quite proficient at picking up and understanding code. He was assisted by Hubert Soleki and Azzam Bahri. While everyone else contributed to testing where possible. Hubert took on risk manager due to him also being the project manager, however everyone was equally responsible for managing their own risks and reporting on them. To prevent risks affecting the project's timeline.
Following the bus method we always made sure 2-3 people understood parts of the project and were able to take it on, in case of catastrophe.

To maintain code quality and cleanliness we did frequent code reviews from group members. This helped all members to understand critical elements of code. Allowing tests to be written quickly by members of the group who didn't personally write the code. A side effect of this was a reduction of bugs.
Similar to the code reviews, we regularly held meetings to review documentation and any other work that needed completing. This helped to not only mitigate the risks mentioned in the risk assessment but also prevent many from happening. This also helped to ensure everyone was aware of the current point in development we were at.

Role	**Name**
Team Lead	Hubert Solecki
Team Lead Backup	Felix Seanor
DevOps Lead	Felix Seanor
DevOps Backup	Sam Toner, Jack Hinton
Change log	Azzam Bahri
Change log backup	Hubert Solecki, Jack Vickers
Risks Lead	Hubert Solecki
Risks Backup	Jack Vickres, Azzam Bahri

Website Lead	Jack Vickers
Testing Lead	Jack Vickers
Testing Backup	Azzam Bahri, Hubert Solecki
Architecture Lead	Jack Hinton
Architecture Backup	Sam Toner, Felix Seanor
Gantt Chart and planning	Hubert Soleki
Gantt Chart and Planning Backup	Jack Vickers, Felix Seanor

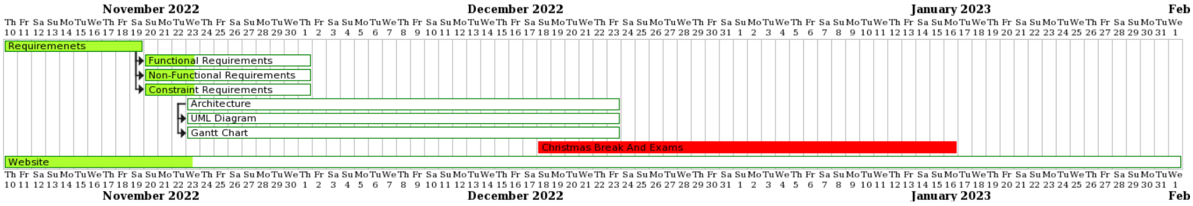
Change log is for subsequent documents except for architecture. Due it needing to be done by the development team.

Minor Roles:

- Secretary - Jack Vickers. In charge of minutes with Hubert Solecki as a backup
- Timeline Manager - Felix Seanor. In charge of code and documentation being done in time.

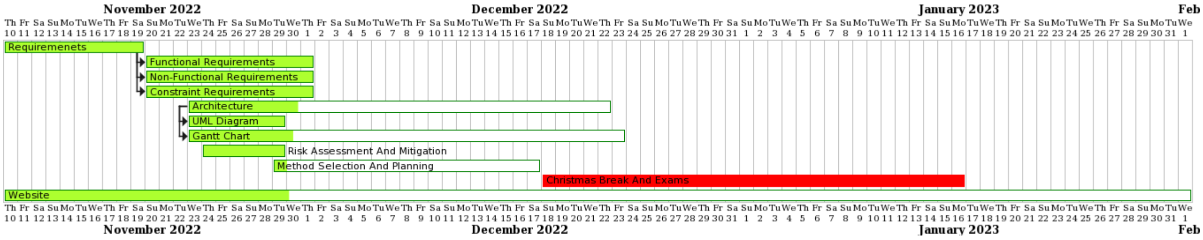
Systematic Plan - Triple10 - Assessment 1

23/11/2022



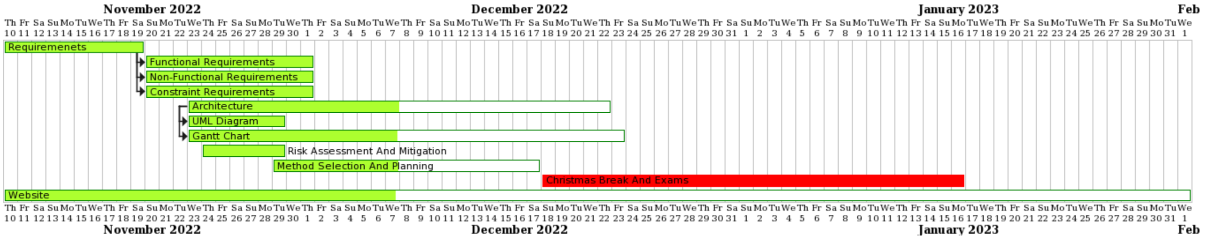
This was our first week and the first iteration of our gantt chart, we had finished our client meeting and we had started to organise our requirements specification. Here, our plan was to focus on the requirements and then start transitioning to the Architectural part. We did not start working on the risk assessment and mitigation, we had a general idea on the method we were gonna use but nothing final.

30/11/2022



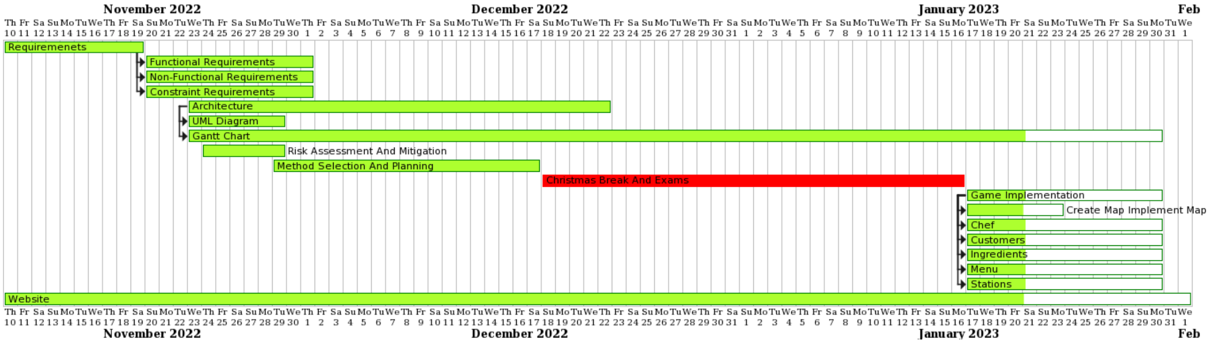
This week, we finished the requirements document and we had started to work on the architecture, as seen above we were making progress with the gantt chart and we had finished the UML diagram, we also started working on the Method that we are gonna adopt, our plan was to finish everything other than the the implementation of the game and its documentation until after the Exam week, we knew it was gonna be tight with the deadlines but that decision was made by our team as a whole, we believed we could make it work

07/12/2022



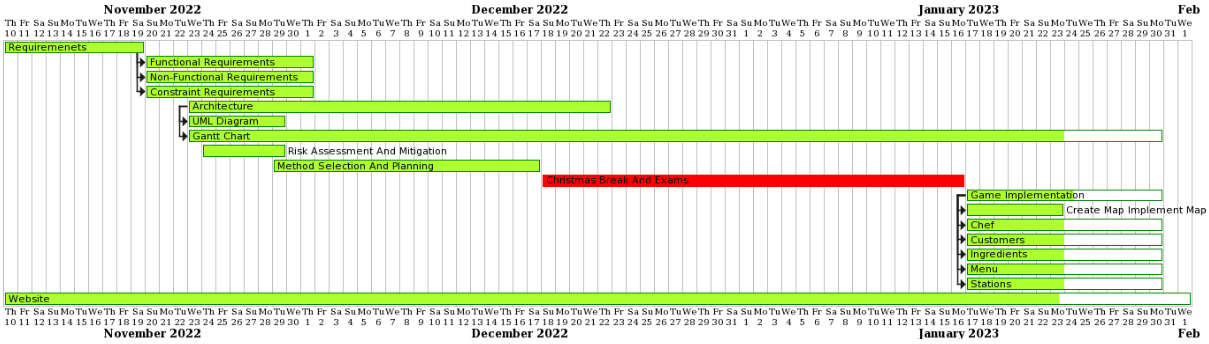
This week, we were still still working on the architecture and we started designing the map and we continued the research for the method selection and planning and we stuck with the plan which is finishing everything except the implementation of the game until after the Christmas break and the exam week

20/01/2023



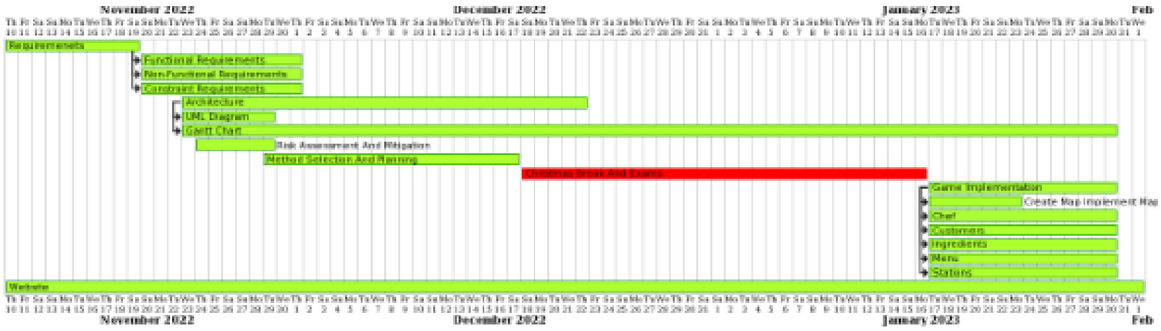
This week, we had finished the method selection and planning and the architecture, we started to implement the game, we split up the game implementation to allow us to work in parallel, we were behind schedule on the implementation of the game as we had some problems with fully understanding libGDX but as all other parts were mostly done, we were able to use the full effort of the team which allowed us to pick up the pace and get right back on schedule

25/01/2023



This week, we were in the later stages of the game implementation, we continued the work on it, and we also started to review all the previous work that had been done on the requirements, architecture, risk assessment and the method selection and planning.

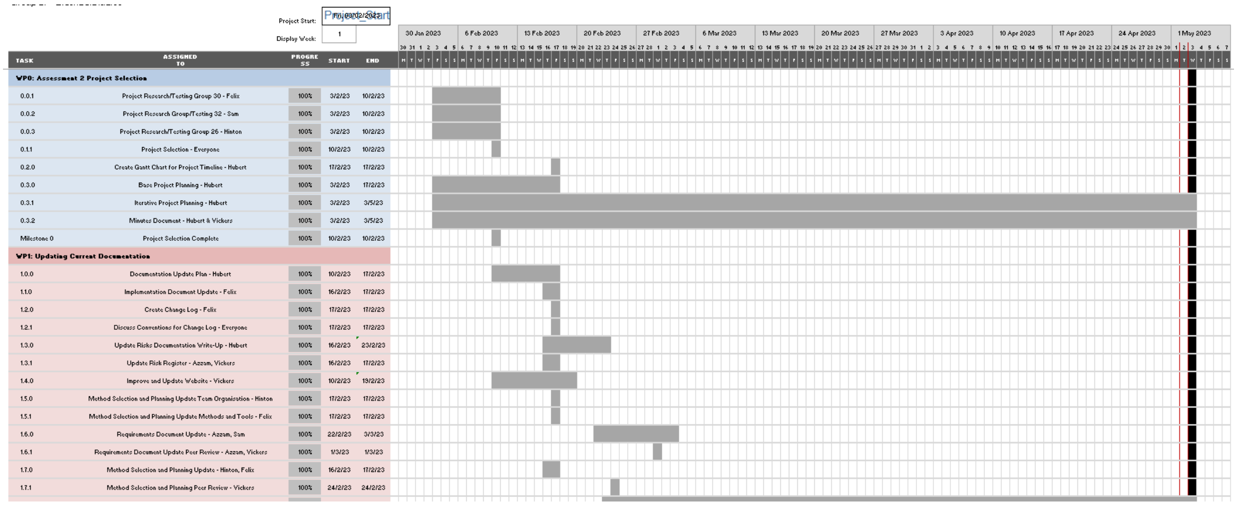
30/01/2023



Systematic Plan - BlackCatStudios - Assessment 2

We added all the dynamic risks into gant chart on the website as well as how our project plan dynamically changed according to these risks and separate circumstances. See our website for these images.

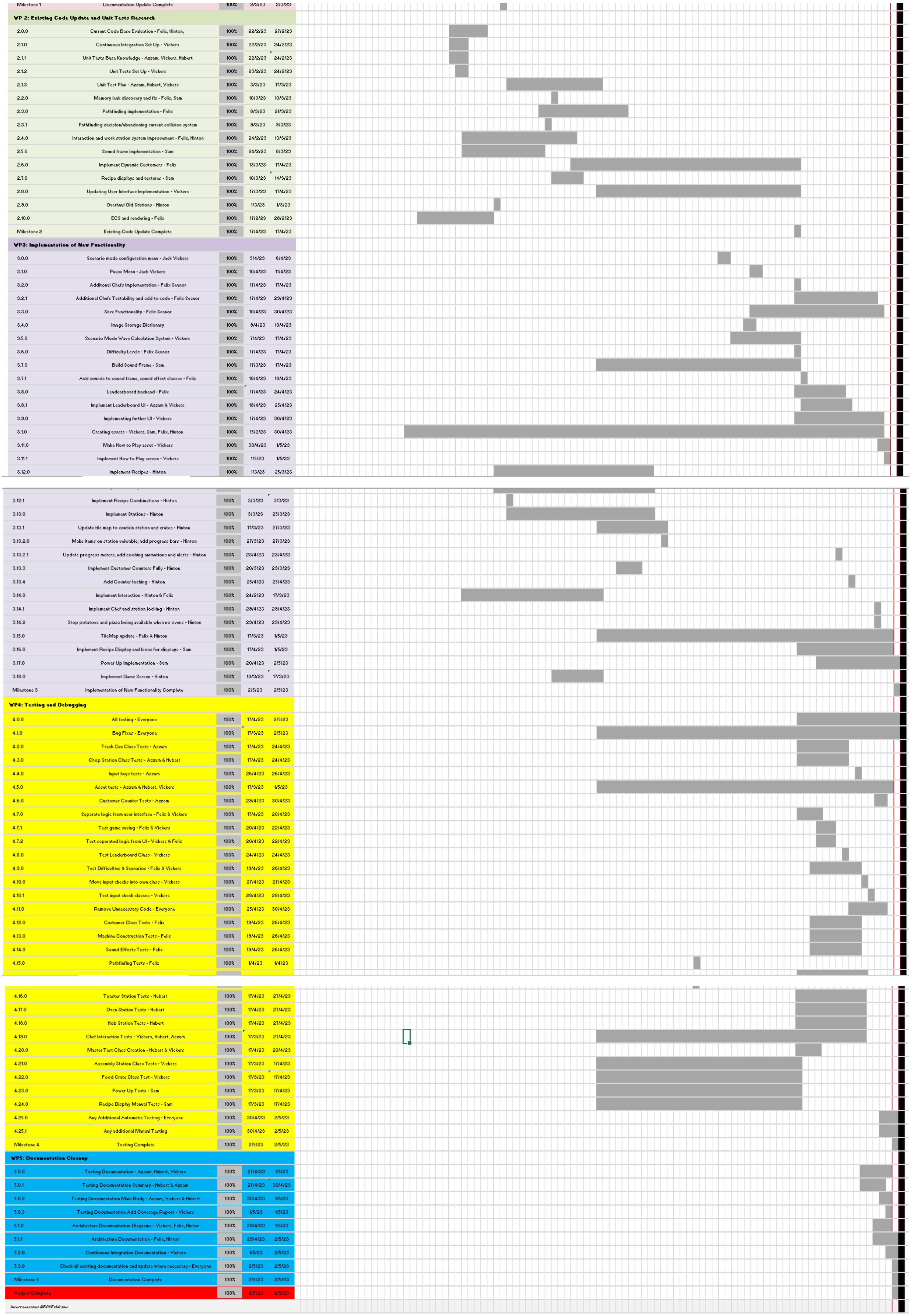
For the full gantt graph see below



Project Start: Fri, 03/02/2023 Project_Start

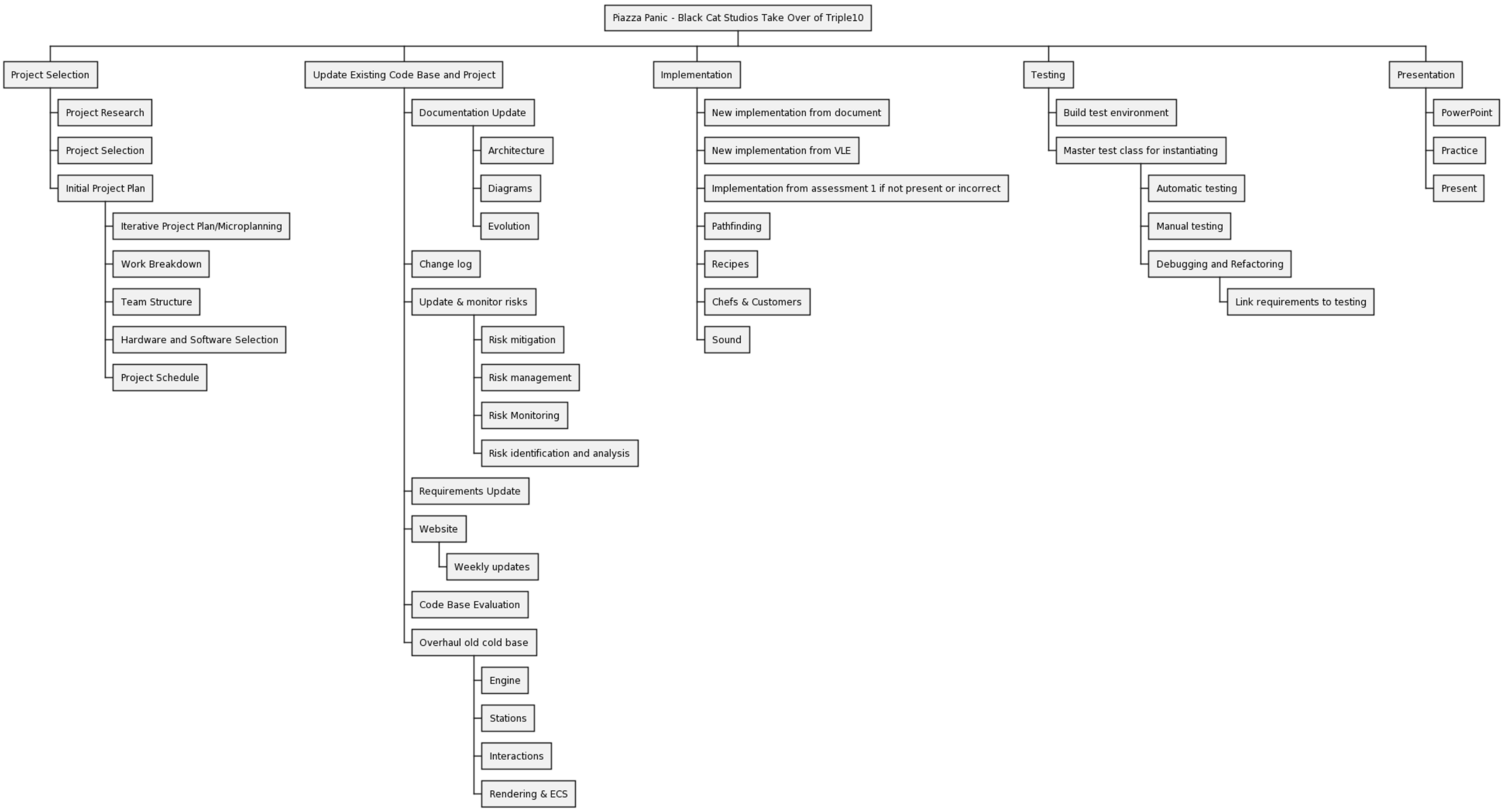
Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
WP0: Assessment 2 Project Selection				
0.0.1	Project Research/Testing Group 30 - Felix	100%	3/2/23	10/2/23
0.0.2	Project Research Group/Testing 32 - Sam	100%	3/2/23	10/2/23
0.0.3	Project Research/Testing Group 26 - Hinton	100%	3/2/23	10/2/23
0.1.1	Project Selection - Everyone	100%	10/2/23	10/2/23
0.2.0	Create Gantt Chart for Project Timeline - Hubert	100%	17/2/23	17/2/23
0.3.0	Base Project Planning - Hubert	100%	3/2/23	17/2/23
0.3.1	Iterative Project Planning - Hubert	100%	3/2/23	3/5/23
0.3.2	Minutes Document - Hubert & Vickers	100%	3/2/23	3/5/23
Milestone 0	Project Selection Complete	100%	10/2/23	10/2/23
WP1: Updating Current Documentation				
1.0.0	Documentation Update Plan - Hubert	100%	10/2/23	17/2/23
1.1.0	Implementation Document Update - Felix	100%	16/2/23	17/2/23
1.2.0	Create Change Log - Felix	100%	17/2/23	17/2/23
1.2.1	Discuss Conventions for Change Log - Everyone	100%	17/2/23	17/2/23
1.3.0	Update Risks Documentation Write-Up - Hubert	100%	16/2/23	23/2/23
1.3.1	Update Risk Register - Azzam, Vickers	100%	16/2/23	17/2/23
1.4.0	Improve and Update Website - Vickers	100%	10/2/23	19/2/23
1.5.0	Method Selection and Planning Update Team Organisation - Hinton	100%	17/2/23	17/2/23
1.5.1	Method Selection and Planning Update Methods and Tools - Felix	100%	17/2/23	17/2/23
1.6.0	Requirements Document Update - Azzam, Sam	100%	22/2/23	3/3/23
1.6.1	Requirements Document Update Peer Review - Azzam, Vickers	100%	1/3/23	1/3/23
1.7.0	Method Selection and Planning Update - Hinton, Felix	100%	16/2/23	17/2/23
1.7.1	Method Selection and Planning Peer Review - Vickers	100%	24/2/23	24/2/23



Milestone 1	Documentation Update Complete	100%	2/3/23	2/3/23
WP 2: Existing Code Update and Unit Tests Research				
2.0.0	Current Code Base Evaluation - Felix, Hinton,	100%	22/2/23	27/2/23
2.1.0	Continuous Integration Set Up - Vickers	100%	22/2/23	24/2/23
2.1.1	Unit Tests Base Knowledge - Azzam, Vickers, Hubert	100%	22/2/23	24/2/23
2.1.2	Unit Tests Set Up - Vickers	100%	23/2/23	24/2/23
2.1.3	Unit Test Plan - Azzam, Hubert, Vickers	100%	3/3/23	17/3/23
2.2.0	Memory leak discovery and fix - Felix, Sam	100%	10/3/23	10/3/23
2.3.0	Pathfinding implementation - Felix	100%	8/3/23	21/3/23
2.3.1	Pathfinding decision/abandoning current collision system	100%	9/3/23	9/3/23
2.4.0	Interaction and work station system improvement - Felix, Hinton	100%	24/2/23	13/3/23
2.5.0	Sound frame implementation - Sam	100%	24/2/23	8/3/23
2.6.0	Implement Dynamic Customers - Felix	100%	13/3/23	17/4/23
2.7.0	Recipe displays and textures - Sam	100%	10/3/23	14/3/23
2.8.0	Updating User Interface Implementation - Vickers	100%	17/3/23	17/4/23
2.9.0	Overhaul Old Stations - Hinton	100%	1/3/23	1/3/23
2.10.0	ECS and rendering - Felix	100%	17/2/23	28/2/23
Milestone 2	Existing Code Update Complete	100%	17/4/23	17/4/23
WP3: Implementation of New Functionality				
3.0.0	Scenario mode configuration menu - Jack Vickers	100%	5/4/23	6/4/23
3.1.0	Pause Menu - Jack Vickers	100%	10/4/23	11/4/23
3.2.0	Additonal Chefs Implementation - Felix Seanor	100%	17/4/23	17/4/23
3.2.1	Additional Chefs Testability and add to code - Felix Seanor	100%	17/4/23	29/4/23
3.3.0	Save Functionality - Felix Seanor	100%	10/4/23	30/4/23
3.4.0	Image Storage Dictionary	100%	9/4/23	10/4/23
3.5.0	Scenario Mode Wave Calculation System - Vickers	100%	7/4/23	17/4/23
3.6.0	Difficulty Levels - Felix Seanor	100%	17/4/23	17/4/23
3.7.0	Build Sound Frame - Sam	100%	17/3/23	17/4/23
3.7.1	Add sounds to sound frame, sound effect classes - Felix	100%	18/4/23	18/4/23
3.8.0	Leadearboard backend - Felix	100%	17/4/23	24/4/23
3.8.1	Implement Leaderboard UI - Azzam & Vickers	100%	18/4/23	25/4/23
3.9.0	Implementing further UI - Vickers	100%	17/4/23	30/4/23
3.1.0	Creating assets - Vickers, Sam, Felix, Hinton	100%	15/2/23	30/4/23
3.11.0	Make How to Play asset - Vickers	100%	30/4/23	1/5/23
3.11.1	Implement How to Play screen - Vickers	100%	1/5/23	1/5/23
3.12.0	Implement Recipes - Hinton	100%	1/3/23	25/3/23
3.12.1	Implement Recipe Combinations - Hinton	100%	3/3/23	3/3/23
3.13.0	Implement Stations - Hinton	100%	3/3/23	25/3/23
3.13.1	Update tile map to contain station and crates - Hinton	100%	17/3/23	27/3/23
3.13.2.0	Make items on station veiwable; add progress bars - Hinton	100%	27/3/23	27/3/23
3.13.2.1	Update progress meters, add cooking animations and alerts - Hinton	100%	23/4/23	23/4/23
3.13.3	Implement Customer Counters Fully - Hinton	100%	20/3/23	23/3/23
3.13.4	Add Counter locking - Hinton	100%	25/4/23	25/4/23
3.14.0	Implement Interaction - Hinton & Felix	100%	24/2/23	17/3/23
3.14.1	Implement Chef and station locking - Hinton	100%	29/4/23	29/4/23
3.14.2	Stop potatoes and pizza being available when no ovens - Hinton	100%	29/4/23	29/4/23
3.15.0	TileMap update - Felix & Hinton	100%	17/3/23	1/5/23
3.16.0	Implement Recipe Display and Icons for displays - Sam	100%	17/4/23	1/5/23
3.17.0	Power Up Implementation - Sam	100%	20/4/23	2/5/23
3.18.0	Implement Game Screen - Hinton	100%	10/3/23	17/3/23
Milestone 3	Implementation of New Functionality Complete	100%	2/5/23	2/5/23
WP4: Testing and Debugging				
4.0.0	All testing - Everyone	100%	17/4/23	2/5/23
4.1.0	Bug Fixes - Everyone	100%	17/3/23	2/5/23
4.2.0	Trash Can Class Tests - Azzam	100%	17/4/23	24/4/23
4.3.0	Chop Station Class Tests - Azzam & Hubert	100%	17/4/23	24/4/23
4.4.0	Input keys tests - Azzam	100%	26/4/23	26/4/23
4.5.0	Asset tests - Azzam & Hubert, Vickers	100%	17/3/23	1/5/23
4.6.0	Customer Counter Tests - Azzam	100%	29/4/23	30/4/23
4.7.0	Separate logic from user interface - Felix & Vickers	100%	17/4/23	20/4/23
4.7.1	Test game saving - Felix & Vickers	100%	20/4/23	22/4/23
4.7.2	Test seperated logic from UI - Vickers & Felix	100%	20/4/23	22/4/23
4.8.0	Test Leaderboard Class - Vickers	100%	24/4/23	24/4/23
4.9.0	Test Difficulties & Scenarios - Felix & Vickers	100%	19/4/23	26/4/23
4.10.0	Move input checks into own class - Vickers	100%	27/4/23	27/4/23
4.10.1	Test input check classes - Vickers	100%	28/4/23	28/4/23
4.11.0	Remove Unnecessary Code - Everyone	100%	25/4/23	30/4/23
4.12.0	Customer Class Tests - Felix	100%	19/4/23	26/4/23
4.13.0	Machine Construction Tests - Felix	100%	19/4/23	26/4/23
4.14.0	Sound Effects Tests - Felix	100%	19/4/23	26/4/23
4.15.0	Pathfinfing Tests - Felix	100%	1/4/23	1/4/23
4.16.0	Toaster Station Tests - Hubert	100%	17/4/23	27/4/23
4.17.0	Oven Station Tests - Hubert	100%	17/4/23	27/4/23
4.18.0	Hob Station Tests - Hubert	100%	17/4/23	27/4/23
4.19.0	Chef Interaction Tests - Vickers, Hubert, Azzam	100%	17/3/23	27/4/23
4.20.0	Master Test Class Creation - Hubert & Vickers	100%	17/4/23	20/4/23
4.21.0	Assembly Station Class Tests - Vickers	100%	17/3/23	17/4/23
4.22.0	Food Crate Class Test - Vickers	100%	17/3/23	17/4/23
4.23.0	Power Up Tests - Sam	100%	17/3/23	17/4/23
4.24.0	Recipe Display Manual Tests - Sam	100%	17/3/23	17/4/23
4.25.0	Any Additional Automatic Testing - Everyone	100%	30/4/23	2/5/23
4.25.1	Any additional Manual Testing	100%	30/4/23	2/5/23
Milestone 4	Testing Complete	100%	2/5/23	2/5/23
WP5: Documentation Cleanup				
5.0.0	Testing Documentation - Azzam, Hubert, Vickers	100%	27/4/23	1/5/23
5.0.1	Testing Documentation Summary - Hubert & Azzam	100%	27/4/23	30/4/23
5.0.2	Testing Documentation Main Body - Azzam, Vickers & Hubert	100%	30/4/23	1/5/23
5.0.3	Testing Documentation Add Coverage Report - Vickers	100%	1/5/23	1/5/23
5.1.0	Architecture Documentation Diagrams - Vickers, Felix, Hinton	100%	29/4/23	1/5/23
5.1.1	Architecture Documentation - Felix, Hinton	100%	29/4/23	2/5/23
5.2.0	Continuous Integration Documentation - Vickers	100%	1/5/23	2/5/23
5.3.0	Check all existing documentation and update where necessary - Everyone	100%	2/5/23	2/5/23
Milestone 5	Documentation Complete	100%	2/5/23	2/5/23
Porject Complete		100%	2/5/23	2/5/23
Insert new rows ABOVE this one				

Above is the final snapshot of the Gantt Chart for the project. Please zoom in to view it better or view it on our website at the following link (under ‘Final Snapshot Of The Gantt Chart’).



Above is the work breakdown diagram that shows the work packages that we were considering at the start of the project. This was useful in the development of the plan, the gantt chart and the way in which this gantt chart would be divided into work packages. We have seen a work package diagram in Triple10’s documentation and we have decided to add it since it was useful in the development of the plan in assessment 1. We have adjusted the diagram to best suit us in terms of assessment 2.