Change Report

Group 27 - BlackCat Studios
Jack Vickers, Hubert Solecki, Jack Hinton, Sam Toner, Felix Seanor, Azzam Amirul Bahri

To begin our process of taking over Triple10's project, we as a group went through all of the Assessment 1 deliverables done by team Triple10. For the documents that were submitted before, we had a look at the ones that match the requirements that are suitable to move forward with Assessment 2.

- For the tools used for the changes, we've used Google Docs to edit each deliverable and to keep track of every change in the Change Log document. This gives us the ability to see all edits that have been made to the documents by each team member. We made sure that every new change has been peer reviewed as quality control to ensure the work done will be the best quality. We also cross checked any new implementations with the requirements to ensure we have fulfilled all the requirements.
    - To ensure we caught everything, we made a record of every change we made in a document and added the most notable changes to the change log.
- We have also made a Gantt chart to track the progress of the project for assessment 2 as well as for the changes.
- Each week we wrote minutes for the team meetings that had taken place so that we could keep track of what happened during these.
- After each Assessment 1 deliverables were edited, we peer reviewed the document and if we felt like we were missing any information, other team members had a look at them.
- We also have commented on the code on things that have been changed and peer reviewed them and finally put a date on them after we have finalised the bit of the code. In the comments in the code we have mentioned which sections were written by our team and which were written by team Triple 10
- We also used Discord to communicate with each other during term time for online meetings if team members can not do in-person meetings and also during the Easter break to go over the Assessment 1 deliverables to see which documents need to be changed.
- We mostly have done conventional in-person meetings which we found as the best method to review the deliverables and determine which documents need to be changed as well as doing peer reviews after any changes that have been made because the communication is direct and any edits can be done straight away.

Changes Made

Requirements

Original: https://jv1ck3rs.github.io/Triple10-X-BlackCatStudios/Req1.pdf
Updated: https://jv1ck3rs.github.io/Triple10-X-BlackCatStudios/pdfs/Req2.pdf
The updated requirements document written by team BlackCatStudios can be found at

- Updated Team Triple 10's requirements to suit the new assessment and introduced new and old requirements.
- Removed **NFR_CONTROL_ SCHEME** (1/3/2023 - Sam Toner) as we believed it was unnecessary.
- Changed **NFR_SPEED** to **NFR_MEMORY** (22/2/2023 - Sam Toner) as runtime didn't make sense as a requirement.
- Added requirements from assessment 1 that we believed were missed, E.G
    - **UR_COLLECT_ITEM** (22/2/2023 - Azzam)
    - **UR_INTERACTION** (22/2/2023 - Jack Vickers)

  This has been done so that we cover as many requirements as are necessary for the game.
- Added new requirements from assessment 2, E.G
    - **UR_REPUTATION** (22/2/2023 - Azzam)
    - **UR_GAMEMODE** (22/2/2023 - Jack Vickers)
- Included all requirements mentioned in the assessment brief as well as the hidden additional requirements **UR_POWERUP** (22/2/2023 - Azzam)**, UR_SAVE_GAME** (22/2/2023 - Jack Vickers)**, UR_DIFFICULTY** (22/2/2023 - Sam Toner)**, NFR_SAVE_GAME** (22/2/2023 - Sam Toner)
- Added categories to the requirements tables to make it easier to see which requirements apply to which aspects of the game.
- Wrote about how we researched requirements. Most of the research methods are on the requirements document.
- Changed **NFR_TELEPORT_POWERUP** to **NFR_MEGA_FOOD_POWERUP** as we decided teleportation would cause a major imbalance in the game (23/3/2023 - Sam Toner, Jack Vickers)

Architecture
The original architecture document written by team Triple10 can be found at
Original: **https://jv1ck3rs.github.io/Triple10-X-BlackCatStudios/Arch1.pdf**
New: https://jv1ck3rs.github.io/Triple10-X-BlackCatStudios/pdfs/Arch2.pdf
The updated architecture document written by team BlackCatStudios can be found at
Insert link

(10/2/2023 - Felix, Sam, Jack Vickers)
We took over Team Triple 10s project, due to their aesthetic art style and map design. When we started we intended to follow brownfield development practices. However after fully exploring their code, we discovered that it would be incredibly difficult to do so. We decided to overhaul core elements and functionalities of the game that would allow us to rapidly extend and enhance the game. While we tried to reuse the previous group's code as much as possible (see comments in the games source files), many elements had to be completely ripped out to make space for more functional and object oriented code. We decided to overhaul these core systems, chefs, rendering, and the food & stations system. Allowing us to fulfil old and new requirements such as **UR_ENDLESS_END**, **UR_POWERUPS** and **UR_CUSTOMER_ ORDER**.

(24/2/2023 - Felix, Jack Hinton)
The biggest change we made to the architecture was changing from a monolithic architecture where all the logic was in one file to an OOP-style architecture using dependency injection and polymorphism to give functionality to objects. We did this in as it would allow us to rapidly expand the game in the direction we wanted to take us with little overhead. To efficiently implement this we used a custom ECS system where each object in the game would be represented by a GameObject. Each GameObject has a renderable image, a position and a list of scripts (segments of code defining logic based on the GameObject). These scripts are updated every frame, and before rendering to allow for complicated logic to take place. This made the requirement **FR_CHANGE_CHEF** incredibly simple and straightforward to implement. But also a more complicated requirement e.g. **UR_CUSTOMER_ ORDER**, possible to implement.

(23/3/2023 - Jack Hinton)
The item and workstation feature overhaul was mainly due to their monolithic architectural style that used consonant spatial references rather than object oriented references, we decided that it would be easiest to introduce a whole new system instead of extending the system. As we wanted to allow for complex recipes involving time steps with an interaction step mixed in. As well as new stations. Knowing our plans for the game we decided that it would be entirely impractical to extend their current system, leaving no choice but to overhaul it completely from the ground up. This overhaul was to ensure **UR_PREP** was met correctly. As it allowed us to design the complex logic surrounding recipes and cooking.

(23/3/2023 - Felix, Jack Hinton, Jack Vickers, Sam Toner)
Finally the rendering system was one of the larger changes as it had to change from rendering a static map and a few predefined movable objects, to countless objects that could

be defined and destroyed during runtime. We did this as we realised due to the static map rendering, this wouldn't work with our ECS system, as we would have had to have a predefined notion of what objects we wanted to render, when and where. To do this we incorporated rendering into our ECS system using the render() procedure. This uses our renderable abstraction to LibGDX image system. Allowing for both sprites and TextureRegions to define how our objects look like. Finally GameObjects are stored by a GameObjectManager, and requested on render by a RenderingManager and sorted by their layer to create a sufficient top down render. We also implemented a texture dictionary that stores references to every image loaded in, and prevents the same image from being loaded in more than once. This was to fulfil **NFR_MEMORY**, as it prevents major memory leaks.

(23/3/2023 - Jack Hinton, Felix)
Because of all these changes, Team Triple 10s previous most architectural diagrams were completely invalid and couldn't be simply updated. Thus we had to create new diagrams as seen in the architectural file. We were able to only modify the scenario flow state as our games were designed to be very similar in functionality. For endless assume that n = infinity and can only end when the player runs out of reputation. We were also able to modify the behaviour diagram to include assembly stations. We left their reasoning about why it was used as it was a good summary of why the behaviour didn't change throughout the design process. Nothing needed to be added to it.

(23/3/2023 - Felix, Jack Hinton)
We added in a revamped design process section as their documentation didn't reflect our design process and we wanted to include some of our decisions making and changes we made throughout our part of the project. Since only small segments of their code remained, it didn't feel right to keep much of their design process that didn't match up with our process. Parts of their code such as reflections on how their classes expanded through coding the design. We expanded this section as it reflected our experiences and we wanted to show how testing also affected us.

(23/3/2023 - Felix, Jack Hinton, Sam Toner)
We also made changes to the paragraph about how the classes reflected our requirements, as the classes had changed, but fulfilled mostly the same requirements.

(23/3/2023 - Felix)
Added in paragraphs explaining our initial design concepts and what makes our design unique.

(24/3/2023 - Jack Hinton, Felix, Sam Toner)
We added CRC cards in order to show our planning.
We also added in use case diagrams to understand early on how we wanted to game to operate and the user to interact with it. This allowed us to understand what we needed to achieve with development. We decided to reuse some of our use case diagrams from our past assessment to maintain simplicity and easy of interaction from assessment 1, as fundamentally the systems are not that different, instead editing some of the scenarios and postconditions to adjust for new factors.

(25/3/2023 - Jack Hinton, Felix, Sam Toner)

We created our first iteration of the package diagram, we simplified it in line with the feedback from the previous assessment. We also planned some sequence diagrams in order to understand how our program would work, and streamline development.

(2/5/2023 - Jack Hinton, Felix, Sam Toner)
We looked through our implementation and tried to make an updated package diagram, however this was not possible due to the large number of dependencies and packages as we wanted to keep it readable and simple. So therefore has been omitted.

(2/5/2023 - Hubert Solecki, Felix Seanor)
We added a description of how we implemented difficulty levels in the game describing specifically what the user starts with and what is omitted or added in different difficulty levels to make that more difficult or easier.

Method Selection and Planning
Original: https://jv1ck3rs.github.io/Triple10-X-BlackCatStudios/Plan1.pdf
New: https://jv1ck3rs.github.io/Triple10-X-BlackCatStudios/pdfs/Plan2.pdf

The original method selection and planning document written by team Triple10 can be found at:

- We made changes to Team Organisation and Tools chosen to reflect the handover in ownership of the project. This was to ensure the workload was distributed evenly for every team member depending on strengths to make sure work can be done in an efficient manner. (17/2/2023 - Jack Hinton, Hubert)
- Changed references to team Triple10 to past tense to make it more clear that they are no longer working on the project. This was to not confuse the old deliverables with the latest version that suits our project. (17/2/2023 - Felix)
- Explained our intentions with organisation and tools in order to maintain a structure throughout the course of the assessment. (17/2/2023 - Hubert)
  - What we are following from their decisions and deviations we are making
  - How BlackCat studios will be organising and whos in charge of what
- Added in sections about why we use a tool and alternatives even though we are taking over from Triple10. We explained the chosen software engineering methodology and working modules fit our style of work. (17/2/2023 - Felix)
- Added a section about leading certain aspects of the project. We split the group into two teams; Dev team and Dev Testing team, where the leaders of each group would have the most knowledge of what they will be delivering and implementing and to monitor the work of their team members. (17/2/2023 - Hubert, Felix)
- Added our final gantt graph and a link to our gantt graph. We keep track of our changes and set a timeframe for tasks to be done. This was to ensure efficiency of the project. (02/05/2023 - Felix, Hubert)
- Added and edited evolution of our plan, methods, models, and software selection. Also argued why we changed or kept our selections the same compared to team Triple10's giving justifications and alternate considerations for our selections and why we didn't choose those. (02/03/2023 Hubert)
- Added diagram describing general plan of the project. The visualisation of the diagram makes it easier to show the workflow and the work distribution. (02/05/2023 - Hubert)
- We added more sophisticated evolution of the project plan and gantt chart on the website in the weekly updates section to improve on the feedback that we got from Assessment 1. This includes better risk mitigation contingencies and identification and the resulting updates to the project planning and the gantt chart. This also provided a better view of the project overtime. (2/5/2023 - Hubert)

Risks Assessment and Mitigation
Original:https://jv1ck3rs.github.io/Triple10-X-BlackCatStudios/Risk1.pdf
New: https://jv1ck3rs.github.io/Triple10-X-BlackCatStudios/pdfs/Risk2.pdf

- We added a new risk register for assessment 2 and modified the IDs of the risks in the original register which was made by Team Triple 10. The IDs of these risks are now of the form R1.X where X is the specific risk number. The IDs of risks in the new table are of the form R2.X. This naming convention ensures that there are no name conflicts between the two registers and allows for easy understanding between the group, as everyone is familiar with the naming conventions. (17/2/2023 - Hubert)
- Some of the risks in the new register have been inherited from Team Triple 10's assessment 1 risk register as they still apply for assessment 2. Ownership has been changed and assigned to members of our team (BlackCatStudios). (17/2/2023 - Hubert)
- Some of the risks from our assessment 1 risk register have been included in the new risk register because they also still apply for assessment 2. We made sure that the risks that we included are suitable for assessment 2 and can be mitigated if anything occurs. (17/2/2023 - Hubert)
- The likelihood and severity ratings of some inherited risks have been changed because we felt that their impact had changed. As an example, the likelihood of risk R2.26 has been changed from high to low because we all have decent experience using LibGDX now. (17/2/2023 - Hubert)
- The new risk register also includes new risks that we identified during assessment 2. These risks were identified at the start of assessment 2 when we encountered certain problems as well as reviewing feedback from assessment 1. The reason we made a separate register for assessment 2 and didn't add to the original one is so we can easily differentiate between the work done by Team Triple 10 and the work done by our team. (17/2/2023 - Hubert)
- We also completed a write-up of how we continued the risk assessment and mitigation document, carrying on from team Triple 10's work. It contained updates, continuations and anything changed from the original document combined with ours. This was also peer reviewed and changed where necessary. (23/2/2023 - Hubert)
- We have also added how we dynamically assess and deal with risks through our mitigation plan using weekly updates on the website (02/05/2023 - Hubert Solecki)