GREY 2 Process Documentation

Notes

Documents are named x.y.z where:

- x is the current sprint
- y is the document number within that sprint
- z is the version number of that document

On the cover page for each sprint, product documents (user stories, use cases, etc.) are shaded blue.

The team did not fully implement Scrum until Sunday, 07/11/21. On this day the team decided to have 'Friday-to-Friday' sprints. We therefore consider sprint 1 to be 29/10/21-07/11/21 (9-10 days) and sprint 2 to be 07/11/21-12/11/21 (5-6 days). After this, sprints run for 7 days running from Friday to Friday until 17/12/21.

Contents

Sprint 1: 29/10/21-07/11/21

The focus of sprint 1 was twofold. First, it involved setting up an agile process, team norms, and tools for collaboration. Second, it involved game ideation, pitching, and determining which game concept to develop further.

Overview

What's supposed to happen this sprint

No.	Expected Start	Expected Finish	Objective
1	29/10/21	29/10/21	Initial team meeting
2	29/10/21	07/11/21	Conduct skills audit
3	29/10/21	30/10/21	Setup tools for collaboration
4	03/11/21	03/11/21	Gather requirements from customer meeting 1
5	03/11/21	05/11/21	Decide on game pitch to start to develop
6	07/11/21	07/11/21	Team's first daily scrum, sprint review, sprint retrospective, and sprint planning meeting

Review

What actually happened

No.	Started	Finished	Task	Doc
1	29/10/21	29/10/21	Initial team meeting held	1.1.1
2	29/10/21	30/10/21	Setup the following tools: WhatsApp, OneNote, SharePoint, MS Planner, MS Teams	1.2.1
3	30/10/21	06/11/21	Created 5 game pitches	1.3.1
4	03/11/21	03/11/21	Customer meeting with answers analysed	1.4.1
5	03/11/21	05/11/21	Skills audit created and completed	1.5.1
6	07/11/21	07/11/21	Game pitch voting form created and completed	1.6.1
7	07/11/21	07/11/21	End of sprint meeting held consisting of daily scrum, sprint review, sprint retrospective, sprint planning	1.7.1
8	Throughout		Backlog created and updated	1.8.1
9	Throughout		Exception handling	1.9.1

Initial Team meeting | 1.1.1

29/10/21, 13:00-14:30, 10W.5.01

Attendance

All present apart from CP due to surgery
Meeting led by TW
Minutes taken by IL

Minutes

1. Summarised project brief to ensure everyone had a shared understanding of the project.

2. Discussed background, relevant skills, and interests. Agreed to conduct a more detailed skills audit.

    a. QZ: engineering
    b. LW: music
    c. CP: physics
    d. DD: medicine
    e. NO: computer engineering
    f. LG: physics
    g. IL: maths
    h. TW: economics

3. Discussed aims for what each person wants to get out of this project. Shared aims were:

    a. Develop game development skills
    b. Experience following an agile process
    c. Produce a good game as a team

4. Decided to follow an agile process:

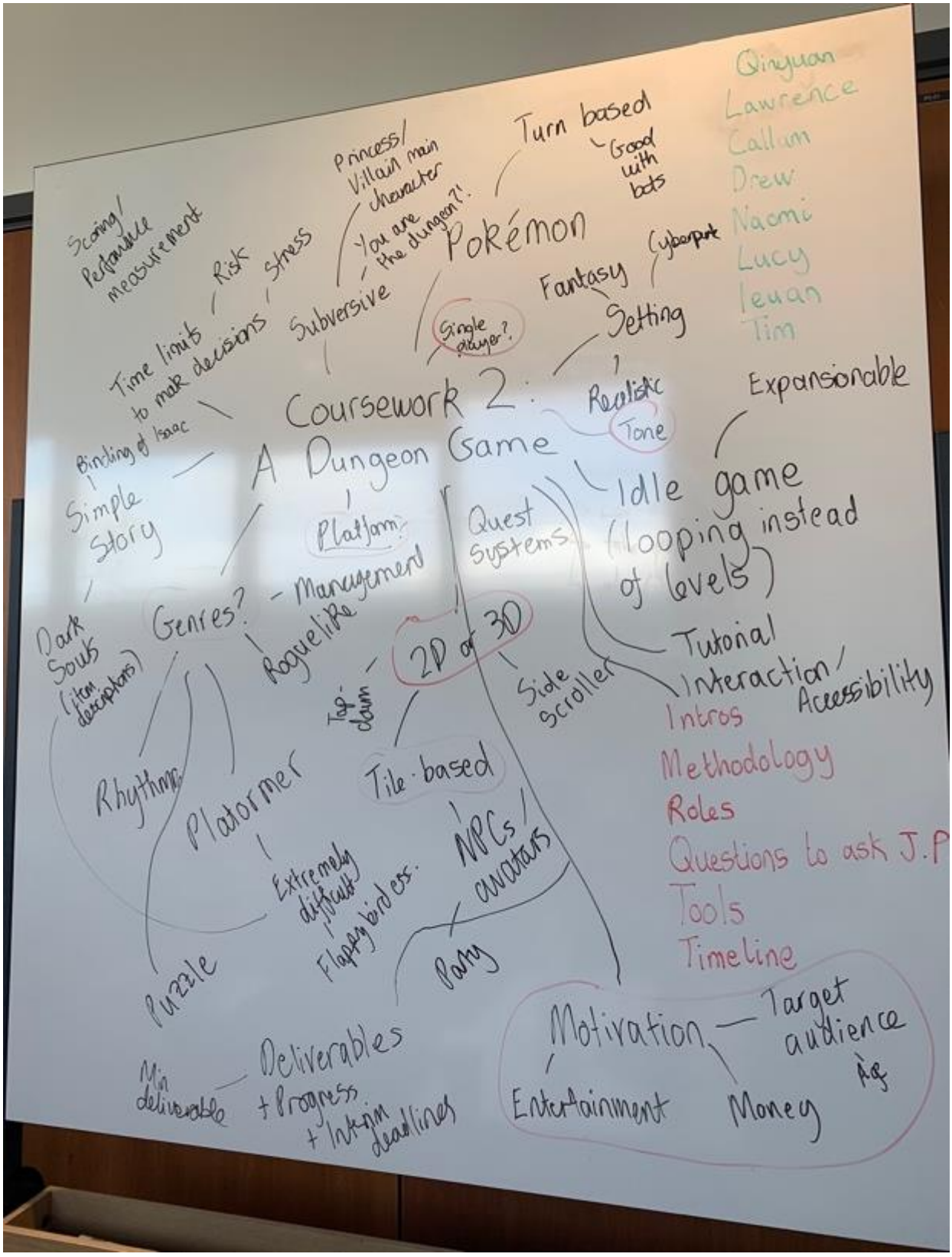
    a. Agreed to use Scrum with practices such as role rotation
    b. Agreed to use a Kanban board to organise tasks into backlogs
    c. Agreed to use practices from XP such as pair programming and refactoring

5. Decided the tools we will use are:

    a. Messaging: Whatsapp
    b. Collaboration: Microsoft Teams
    c. Files: Microsoft Sharepoint
    d. Kanban: Microsoft Planner
    e. Notes: Microsoft OneNote
    f. Git: Github

6. Briefly discussed game engines (Unity, RPG maker, and Unreal) and unanimously agreed Unity was the best option

7. Discussed game genres, features, and inspirations



Our brainstorm of game ideas

8. From our brainstorm, we identified aspects which could make development easier given our short timeframe:

    a. 2D
    b. Single player
    c. Tile-based
    d. Limited/simple story

9. We explored fundamental questions that would be important to ask the customer at our first meeting and found that these fell into 4 categories:

    a. Motivation for customer
    b. Game content
    c. Technical aspects
    d. Timeframe/deliverable questions

Task assignments

1. DD to inform CP about content of meeting as soon as possible

2. LW to create skills audit and share before the customer meeting

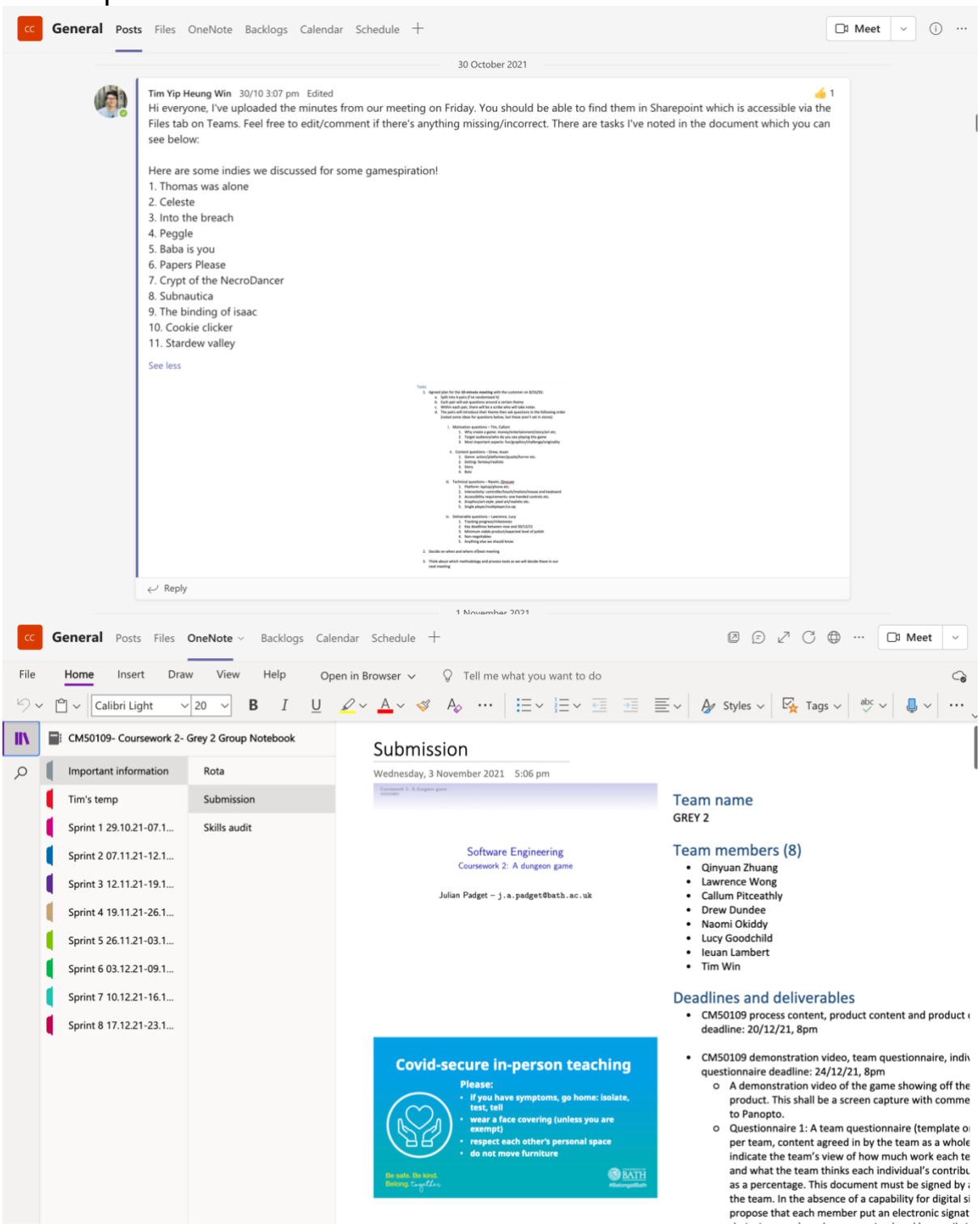
3. TW to set up Whatsapp, Teams, Sharepoint, Planner, and OneNote and integrate Sharepoint, Planner, and OneNote into Teams as soon as possible

4. Split into randomised pairs and prepare questions to ask customer around each category before the customer meeting and share these in OneNote

    a. TW, CP: motivation
    b. DD, IL: content
    c. NO, QZ: technical
    d. LW, LG: timeframe

5. ALL to think of game ideas before the customer meeting and share these in OneNote

Tools | 1.2.1



Setting up these tools and integrating them into Teams early helped us stay organised

Game pitches | 1.3.1

Inspirations

Games that would be feasible to develop to give us some inspiration for what we might pitch:

- Sports/Driving: Rocketleague
- Fighting: Nidhogg
- Puzzle: Stephen's Sausage Roll, The Witness, Baba Is You, Peggle, Puzzle Bobble, Candy Crush, Angry Birds
- Physics based: Getting Over It with Bennett Foddy, World of Goo, Super Monkey Ball
- 2D platformers: Sound Shapes, Celeste, Super Meat Boy, Thomas Was Alone
- Tile based platformer: Frogger
- Tile based tower defense: Plants vs. Zombies
- Tile based action: Crypt of the NecroDancer, Q*bert
- Top down action: Hotline Miami, The Binding of Isaac
- On rails: Pokémon Snap
- JRPG: Classic Final Fantasy
- Idle: Cookie Clicker
- Horror: 5 nights at Freddie's
- Endless runner/high score: Flappy Bird, Doodle Jump, Temple Run, Super Hexagon, Jetpack Joyride, Fruit Ninja
- 3D story: The Stanley Parable, The Unfinished Swan, Flower, Journey
- 3D shooter: Superhot
- Turning something mundane into a game: Paper's Please, Overcooked, Moving Out
- Party/microgame collection: WarioWare, Rhythm Heaven, NES remix, 1-2-Switch, Dumb Ways to Die, Mini Mix Mayhem, Brain Age

TW | Pitch 1: Microgames

A microgame collection with a story mode inspired by the WarioWare series.

Mechanics

- Select stages (dungeons) from a world map
- Each stage has a short story that progresses based on successful completion of microgames
- The collection of microgames to be completed are themed around the short story
- The short stories are told through static images
- The microgames provide a simple instruction like 'Get to the goal' and have very short time limits
- Control and graphics are simple
- Microgame goals include: take a picture, win this partially completed game of tic tac toe, win this game of rock paper scissors
- Microgames could be reused but be slightly different e.g. lose this game of rock paper scissors, different colours
- Tone is fun, wacky, and fast-paced
- You can get better scores the faster you complete the games and get a star rating based on how well you do
- You could have lives so that you can still progress even if you fail a microgame
- The games within each story are in a randomised order to keep you on your toes
- All microgames have the same simple art style to keep things consistent
- Some microgames could be competitive where you have to beat a bot or you could compete for a high-score against a bot

Benefits

- Lends itself well to agile development - each pair could create microgames in parallel
- Will always have something operational to show the customer

- Simple and original concept. Not too many games have these mechanics and other teams will likely do something else
- Tests quick thinking and reflexes, but is ultimately fun, accessible, and easy to learn
- Would work well on the web, or on mobile as controls and graphics will be simple

DD | Pitch 2: Darkest Dungeon

- 2D isometric game
- Party progressing through dungeon
- Move on grid co-ordinates
- Fixed branching points of path, direction chosen by player, possible time limited decision
- Some indication of relative risk and reward for these paths, with enemy encounter rate determined at this stage
- Success is determined by multiple metrics: distance travelled, bosses defeated, gold obtained etc.
- Can implement combat system or settle combat based on relative strength
- Items to restore health, buff party etc.
- Items, gold, and progress lost on death
- Can leave dungeon after each boss, recover, and spend gold
- Main loop is a balance of risk and reward, when to progress and when to leave.
- Could recruit "party leader" bots outside or inside dungeon, who can perform simulated or actual dungeon runs and act based on their attributes e.g. risk-averse.

DD | Pitch 3: Twin-stick shooter

- Single character progressing through multiple room dungeon
- Free movement in x-y, 'twin-stick' control, determining what direction faced and what direction moving separately
- Projectile based combat
- Enemies encountered in each room
- Upgrading or changing projectiles during the run through items
- Success of player determined by rooms cleared, bosses killed etc.
- Enemy characters as bots

NO | Pitch 4: Cards

- Board game type game
- Capture the core/castle style
- Play vs. bots or other players (pvp, pve)
- Draw a specific number of cards per turn
- At the end of everyone's turn, new cards are drawn
- Bots can replace players
- Benefits - a lot of players
- Cons - high learning curve

LW | Pitch 5: Autochess

This pitch was added on 05/11/21 (after the customer meeting)

- An Autochess army builder roguelike
- The player defeats rooms of enemies with a team of units they build
- The combat is not directly controlled by the player; instead, the two sides battle automatically
- After combat, the player collects rewards, rebuilds their team, and decides which room to go to next based on a difficulty hint
- Gameplay loop: build team, position team, start battle, collect rewards, buy more units to replace defeated ones, upgrade units, choose which room to go next, repeat
- Incorporates: risk and reward and strategic gameplay
- 2D isometric, pixel art style

Customer meeting | 1.4.1

03/11/21, 15:00-15:15, CB 5.13

Attendance

All present apart from CP due to surgery
Recording taken by IL
Minutes taken by IL

Prepared questions

TW, CP | Motivation questions

- Why create a game: money/entertainment/story/art
- Target audience: who do you see playing this game
- Important aspects: fun/graphics/challenge/originality

DD, IL | Content questions

- Genre: action/platformer/puzzle/horror
- Movement: 2D/3D/fixed grid/free movement
- Story: linear story/side quests?
- Bots: play against bots/play with bots?
- Single player/multiplayer/co-op/PvP/PvE

NO, QZ | Technical questions

- Platform: laptop/phone etc.
- Interactivity: controller/touch/motion/mouse and keyboard
- Accessibility requirements: one-handed controls/colour blind settings
- Art style: pixel art/realism

LW, LC | Deliverable questions

- Tracking progress/key milestones
- Minimum viable product/expected level of polish
- Non-negotiables
- AOB

Minutes

1. TW, DD, and NO pitched game ideas

2. Customer least intrigued by Twin-stick shooter as “it’s the first thing that people come up with”

3. There are nice things in the other 3 pitches

    a. Customer suggests the possibility of merging Microgames with one of the other ideas to help make it more novel
    b. Cards is likely incompatible with Darkest Dungeon
    c. One idea could form the basis of the game, but you could bring features from the other pitches to make it more original

4. TW asked which game aspects are important.

    a. Customer won’t impose technical demands regarding graphical fidelity. Good graphics are tricky, but you could do something with high quality static images for graphics
    b. Engagement and flow are important
    c. Replay value is important. Customer mentioned you don’t want to make a game that the player only plays once.

5. TW asked about target audience
    a. Customer replied that the target audience will come out from the features implemented
6. DD asked about favourite games and genres
    a. Customer likes games that require thinking about strategy/logic-based/puzzle
7. DD asked about fixed v. free movement
    a. Customer mentioned constrained movement can be part of the challenge of the game and can add to the cognitive reasoning load of the player
    b. It can be difficult to develop free movement
8. NO asked about which platform to develop for
    a. Customer showed strong enthusiasm for the idea of developing for the web
    b. If the game is browser based, it could be played on a laptop or phone if you can sort out resizing
9. DD asked about the involvement of bots
    a. Customer said bots can be used to populate a space
    b. It would be difficult to build a game a bot could play independently
10. JP said we should create a detailed storyboard for the next customer meeting

Analysis

1. Time spent in the lab before the customer meeting was spent discussing questions we would ask and our game pitches. Time spent in the lab after the customer meeting was spent reflecting on the meeting.
2. Given the lukewarm reaction to Twin-stick shooter, team unanimously agreed after the meeting to no longer consider this game idea.
3. Customer showed greatest enthusiasm for Darkest Dungeon as it requires strategy, logical thinking, and balancing risk and reward.
4. Key takeaways were that the customer strongly favours a game that is original, replayable, strategic, and web-based.
5. Customer's suggestion to combine game ideas would add complexity but may help make the game be more original.
6. Team decided to allow more time for additional game pitches (1.4.1). Each member will then rank the final list of game pitches.
7. Team agreed to meet after this poll on 07/11/21.

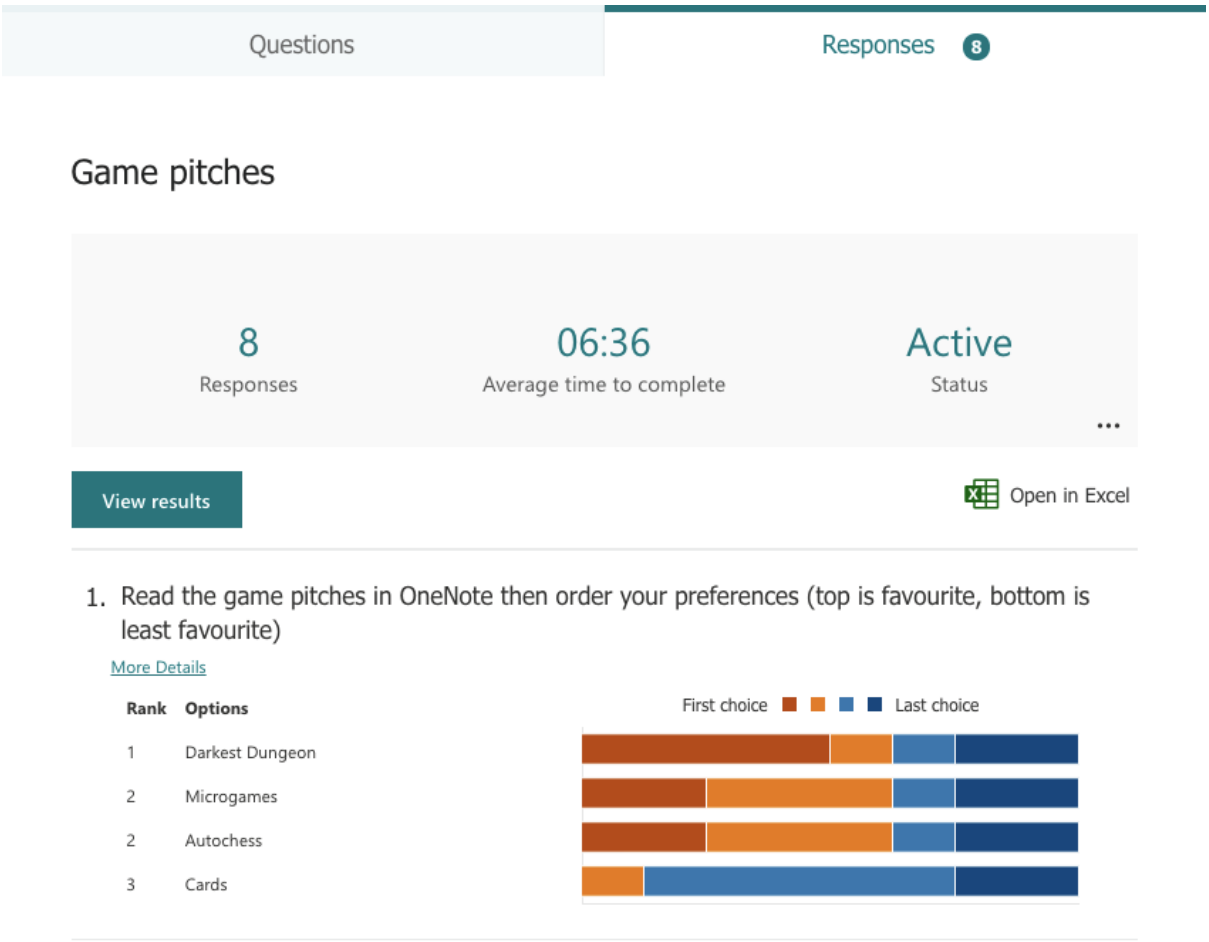
Skills audit | 1.5.1

Name	What was your first degree and was there anything you studied as part of your degree which would be relevant for this project?	Which of the following best describes you?	Do you have a hobby or interests relevant to the project (ie art/design, music, game design)	Is there anything else in particular you would be interested in learning / participating in?	Do you have any anything else to add?	Is there anything you have studied outside of your degree which is relevant for this project? (doesn't matter how closely related!)	Do you have any work experience that is related to this project? (doesn't matter how closely related!)	What areas are you most interested in?	Which areas do you have the most experience in?
Tim Win	BSc. Economics Maths Business and management	I Have no prior coding experience	I know a lot about game design, game journalism, and the gaming industry			-	Management consultant Project management Client management Change management Data analysis Risk management Deliverable creation	Programming;Agile process;Game design;Leadership;Testing;Presenting/talking with clients;Organisation;Art;Music/sound design;	Organisation;Presenting/talking with clients;Leadership;Agile process;Game design;Testing;Programming;Music/sound design;Art;
Andrew Dundee	Medicine.	I have some coding experience, but I am confident in my ability to learn and adapt to requirements of project	Game Design	Enjoy the design aspect of the game.	Passionate about games and the things that make games fun.	I play a lot of games.	No.	Game design;Programming;Presenting/talking with clients;Testing;Leadership;Agile process;Organisation;Art;Music/sound design;	Leadership;Presenting/talking with clients;Game design;Organisation;Programming;Art;Testing;Agile process;Music/sound design;
Callum Pitceathly	Physics, I've done 4 years of scientific computing in Python which included a fair amount of object oriented programming. Never done c# though.	I have some coding experience, but I am confident in my ability to learn and adapt to requirements of project	Not really, I'm a certified game design noob. I'm very into music so can suggest some cool background music for the game. Think it could be quite funny to have classical slay the spire sort of music for most of the game then some heavy trance or techno track for the boss fights.	Can't think of anything at this moment	Nothing particularly relavent sorry.	Not particularly, done a little bit of java.	Afraid not.	Programming;Testing;Music/sound design;Game design;Art;Agile process;Organisation;Leadership;Presenting/talking with clients;	Programming;Testing;Music/sound design;Leadership;Presenting/talking with clients;Organisation;Agile process;Game design;Art;
Naomi Adhiambo Okiddy	Computer engineering, the programming aspects	I have coding experience specific to the requirements of this project (C#/Unity), and am confident in my ability to work independently	game design, pixel art, creative writing	At this moment, no	no	how to use github	software development engineer internships	Art;Testing;Game design;Programming;Music/sound design;Agile process;Presenting/talking with clients;Organisation;Leadership;	Programming;Testing;Game design;Presenting/talking with clients;Leadership;Organisation;Agile process;Art;Music/sound design;
Ieuan Lambert	Maths	I have some coding experience, but I am confident in my ability to learn and adapt to requirements of project	nope	Nope	Nope	nope	Programmer for xassets	Programming;Testing;Game design;Art;Music/sound design;Agile process;Organisation;Leadership;Presenting/talking with clients;	Programming;Testing;Art;Game design;Music/sound design;Agile process;Organisation;Leadership;Presenting/talking with clients;
Zhuang Qinyuan	Electronics and engineering. I learned C and had wroten programs by C to control a robot. I also learned python to clean data, so I have some programming experience.	I have some coding experience, but I am confident in my ability to learn and adapt to requirements of project	programming part	programming part maybe.		I had used solidwork and blender to build simple 3D models.	Unfortunately not.	Programming;Testing;Game design;Agile process;Leadership;Art;Organisation;Music/sound design;Presenting/talking with clients;	Programming;Testing;Art;Leadership;Agile process;Game design;Organisation;Music/sound design;Presenting/talking with clients;
Lawrence Man Chun Wong	Digital Music and Sound Technology. Music production, editing, mixing, mastering.	I have some coding experience, but I am confident in my ability to learn and adapt to requirements of project	Music, gamer, game design	Coding, testing, Agile processes.		Web Development	Sound Engineer	Programming;Music/sound design;Game design;Agile process;Testing;Organisation;Leadership;Presenting/talking with clients;Art;	Programming;Music/sound design;Game design;Organisation;Agile process;Testing;Presenting/talking with clients;Leadership;Art;
Lucy Goodchild	Physics	I have some coding experience, but I am confident in my ability to learn and adapt to requirements of project	Interested in game design.	Anything physics-y we can implement in the game!		C family languages.	None :(except playing games :)	Game design;Programming;Music/sound design;Art;Testing;Organisation;Agile process;Leadership;Presenting/talking with clients;	Programming;Organisation;Agile process;Testing;Leadership;Game design;Presenting/talking with clients;Art;Music/sound design;

Although we're all on the MSc Computer Science course, we have varying skillsets, experience, and interests

Game pitch voting | 1.6.1

Results



Name	Read the game pitches in OneNote then order your preferences (top is favourite, bottom is least favourite)
Tim Yip Heung Win	Microgames;Autochess;Darkest Dungeon;Cards;
Ieuan Lambert	Darkest Dungeon;Microgames;Autochess;Cards;
Lawrence Man Chun Wong	Autochess;Microgames;Cards;Darkest Dungeon;
Andrew Dundee	Autochess;Darkest Dungeon;Cards;Microgames;
Zhuang Qinyuan	Darkest Dungeon;Microgames;Cards;Autochess;
Lucy Goodchild	Microgames;Autochess;Cards;Darkest Dungeon;
Callum Pitceathly	Darkest Dungeon;Autochess;Cards;Microgames;
Naomi Adhiambo Okiddy	Darkest Dungeon;Cards;Microgames;Autochess;

Commentary

1. All team members preferences were aggregated, and the results were shared. This helped promote collective ownership and transparent decision making.

2. The results were close but Darkest Dungeon was the first choice followed by Microgames and Autochess coming in joint second/third. Cards came in a clear fourth place.

3. DD noted that LG, TW, and LW had Darkest Dungeon as their 3rd/4th choice, and this formed a discussion point for our Sprint review (1.7.1).

End of sprint meeting | 1.7.1

07/11/21, 19:00-20:30, MS Teams

Attendance

All present
Meeting led by TW
Minutes taken by IL

Daily scrum and sprint review

As the last team meeting was the initial meeting, discussing work done since the last team meeting was the same as discussing the work done over the sprint, so the daily scrum and sprint review were combined.

Minutes

1. Game pitch voting had recently ended so the team were keen to discuss this first. Although not unanimous and close, Darkest Dungeon was the favourite

2. DD noted TW, LW, and LC had Darkest Dungeon as a lower choice and wanted to know if they were happy developing it

    a. LC lacked clarity regarding how the game worked as did not have familiarity with the game's inspiration
    b. DD clarified the map, branching paths, party system, and combat with examples.
    c. DD acknowledged the party system and combat could be swapped out with something different. The main hook was the branching paths with different types of nodes requiring the player to strategize and balance risk v. reward.
    d. All are happy to proceed with Darkest Dungeon/
    e. Acknowledgement from the team that many of the pitches were liked.
    f. LW added that the theme/tone of the game would be something we decide later but it could be the classic, stereotypical dark dungeon or cute and bright.

3. LW noted that Autochess could be used for the battles in the Darkest Dungeon game. The team agreed that this would combine the best elements from both ideas and make our game more original. LW explained how this would better meet the customer's requirements.

4. NO noted that Microgames could be included after the core game has been developed. These additional features could include: a microgame to get resources, or a microgame to get random items from a shop. Team agreed this would be a good idea for a stretch goal.

5. The team agreed to no longer consider Cards and to process with Darkest Dungeon x Autochess

6. TW explained the tools he setup

    a. SharePoint, OneNote, the Kanban board, and a shared Calendar have been added as tabs to MS Teams
    b. Explained what SharePoint is and how you could add the team's SharePoint to your file explorer for efficient access to the team's files
    c. Encouraged the team to feel free to post in the Teams group to keep others informed or share things others may find useful e.g., Unity tutorials
    d. Clarified OneNote is for draft or random other pieces of information such as plans for Storyboards, submission information, draft minutes etc. This will prevent our shared files becoming disorganised and everyone will be able to see when new pages are created

7. LW led the discussion of the skills audit (1.5.1):

a. Encouraged the team to look at the data on SharePoint if they have not already
b. Our team may have less technical skills in comparison to other teams as we are all on the MSc Computer Science course. Acknowledged that familiarity with Unity and C# may be challenging and time-consuming, but also that our technical skill limitations may encourage more creativity.
c. Coding experience and other skills/interested such as music, art, and design, will be helpful when dividing the team or knowing who to go to for advice

8. CP happy to meet everyone and has caught up with the assistance from DD and through MS Teams.

Sprint retrospective

1. The consensus was that the team synergised well and got off to a good start.

    a. The main reason was a long initial team meeting. This allowed sufficient time for introductions, ensuring a shared understanding of the coursework, brainstorming ideas, and deciding tasks.
    b. The second reason was proactive team members who set up tools for collaboration early. This ensured we could communicate regularly. CP was then able to catch up when he returned in week 2 (1.9.1).

2. We benefitted from the fact that the members in our team did not change after teams were confirmed on 27/10/21. This prevented confusion and meant that all members attended each meeting (except for CP who was injured).

3. The team expected the customer to be more specific and prescriptive in terms of what they wanted. The task is more open-ended than expected. We will need to focus on both responding to the customer's requirements and coming up with many ideas. We also will plan customer meetings around the time limit, so we go through everything essential (1.9.1).

4. Some game pitches could have shared with the rest of the team a day or two before the customer meeting. This would allow other team members to ask questions and build on these pitches.

5. Some team members know a lot about video games. It's important ensure team members with less knowledge have a clear understanding when explaining game pitches and referring to other games.

6. Allowing additional time for additional game pitches was a good idea as we ended up choosing one of these additional game pitches (1.3.1). This did however elongate the 'pilot' sprint we were running up until Sunday instead of Friday. If we had kept to Friday, we could have implemented 'Friday to Friday' sprints without sprint 2 being made shorter.

7. The team concluded that 'Friday to Friday' sprints should be implemented because:

    a. It matches the working week
    b. Provides buffer time after the customer meeting to reflect and adapt to the next sprint
    c. Extra time at the beginning of the sprint (at the weekend) allows more time for issues to be found
    d. Team members were more available on Friday for the longer end of sprint meetings.
    e. Although we don't have the full sprint development time before the client meeting, the benefits outweighed the costs.

8. In future sprints, all tasks need to be assigned with adequate information given. The task: 'Setting up GitHub' was not assigned and so no-one knew who was responsible for doing this.

9. Completed tasks remain in their ‘bucket’ in Microsoft Planner. We will reorganise our backlogs as Product backlog and Sprint backlog as per the Scrum methodology.

10. Sprint 1 documentation is partially completed as notes were thoroughly taken and stored in OneNote. These notes still need to be reformatted and organised as soon as possible so future sprint’s documentation is not also delayed.

11. IL’s decision to record meeting was very useful for remembering what was said. Additional minutes to capture non-verbal information could help too.

Sprint planning

1. QZ alerted the team that the ‘Submission Requirements’ document has recently been uploaded to Moodle

    a. QZ shared his screen. Went through document together to ensure we all understood the deliverables, deadlines, and assessment criteria.
    b. TW noted process content requirements were clear but had questions regarding whether we had to follow the ‘sprint-report-template.md’ exactly, and whether there were rules regarding document names.
    c. DD noted he may have future questions about certain product documents when he does them although the documentation for Sprint 1 should be simple.
    d. TW will add a page on OneNote with the coursework and submission requirements so it’s easily accessible

2. TW noted the focus of the next sprint is creating a storyboard, implementing scrum, and increasing familiarity with the tools we will use to create our game.

    a. TW confirmed no-one had set-up a GitHub for the team yet.
    b. QZ and NO said they have experience with GitHub and were willing to set this up.
    c. LW mentioned it would be best to use our University of Bath accounts and make a group rather than using our person GitHub accounts.
    d. QZ and NO not currently sure how to do this and how to link it to Unity but they are confident they can figure this out.

3. TW raised that it was important to fully define our process soon as there were only 7-8 sprints total. NO agreed that now we had our game concept and completed introductions, it was easier to start implementing more scrum elements.

    a. TW briefly outlined scrum master, product owner, developers, the product backlog, and sprint backlog
    b. TW suggested a shared schedule that kept track of team member’s role for the week, how long tasks take and descriptions of responsibilities for each role.
    c. TW suggested Kanban board is re-done to just have two buckets to represent the product backlog and sprint backlog and that items should be added by the appropriate person, detailed, time-bound, and assigned.

4. DD and LW were keen to create the storyboard. Each has a clear understanding of the Darkest Dungeon and Autochess parts of the game that they will focus on. DD will create a Word document in SharePoint which all team members are welcome to contribute to.

5. CP and LC mentioned we should start product documentation with user stories in Sprint 2.

6. NO, IL, QZ, and LC have had a brief look at Unity tutorials and strongly recommend that everyone start with the Unity Essentials Learning Pathway. TW mentioned if you complete any harder or specialised tutorials, or find you are interested in an area, let the team know

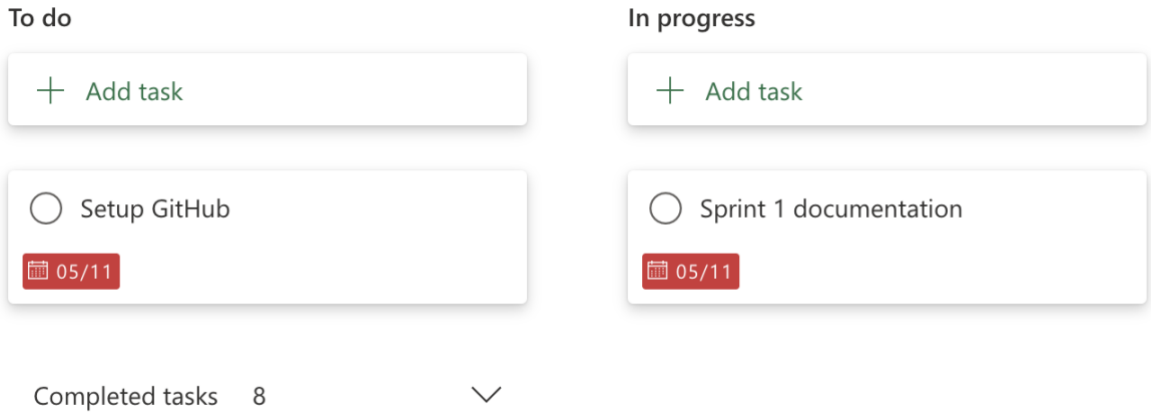
7. Arranging available times was relatively easy as we are all on the same course. The team decided to have 3 scrum meetings on Sunday, Tuesday (as it was before the client meeting), and Friday (before the end of sprint meetings).

Task assignments

1. TW to ask about documentation on 10/11/21
2. QZ and NO to create the GitHub by 09/11/21
3. TW to create a schedule by 09/11/21 and TW to update the Kanban board by 08/11/21
4. DD and LW to finish the storyboard by 10/11/21
5. CP and LC to create the first version of user stories by 12/11/21
6. ALL to download Unity and make progress on the Unity Essentials Learning Pathway
7. TW to add calendar events for meetings by 08/11/2

Backlog | 1.8.1

Snapshot at end of sprint



An early verion of our backlogs using Microsoft Planner

Commentary

1. Nearly all tasks in this sprint were completed.

2. Kanban board to be improved in the next sprint (see Sprint retrospective and Sprint planning in 1.7.1).

Exception handling | 1.9.1

Additional game pitches

1. Objective 5 for this sprint has been delayed.

2. This is because analysis of the customer meeting prompted the team to have additional game pitches based on their revealed preferences.

3. Although we will have to wait to re-pitch these to the customer, these further game pitches may be more aligned with what the customer wants (1.4.1).

4. It will also give more time for other team members to pitch their ideas.

5. The team has decided to allow further game pitches until COP 06/11/21.

6. Each member will then fill in a form asking them to rank the final list of game pitches on 07/11/21.

Known absence

1. DD alerted us to CP's absence this sprint during the initial team meeting.

2. Team agreed importance and priority of setting up collaboration tools such as a MS Teams group to ensure all information is disseminated to all team members in keeping with recommended Scrum practices.

Limited customer meeting time

1. Limited time in the customer meeting meant we were not able to ask all prepared questions.

2. We also did not have the time to ask follow-up questions based on what the customer was most enthused about.

3. Post-customer meeting analysis, it was determined that future meetings should be planned according to the 10-15 minutes.

    a. The most pertinent questions should be asked if time is limited.
    b. We should plan carefully what each meeting will cover and how to present concisely.

Lengths of sprints differ

1. The first 'pilot' sprint ended on Sunday even though we started work on Friday the week prior.

2. The sprint was elongated as we became accustomed to working together and to allow more time for additional game pitches.

3. Going forward the team agreed to keep sprints to the allocated timeframe and that this will be Friday-to-Friday.

Sprint 2: 07/11/21-12/11/21

The focus of sprint 2 was fleshing out our game concept further, training on the tools we'll use to develop our game, and properly implementing the scrum methodology.

Overview

What's supposed to happen this sprint

No.	Expected Start	Expected Finish	Objective
1	07/11/21	10/11/21	Produce a storyboard which outlines our game
2	07/11/21	12/11/21	Make progress on the Unity Essentials Learning Pathway
3	07/11/21	09/10/21	Setup and learn GitHub
4	07/11/21	12/11/21	Write the first version of user stories
5	07/11/21	09/11/21	Create a schedule
6	07/11/21	10/11/21	Complete documentation for sprint 1
7	07/11/21	12/11/21	Start implementing scrum methodology

Reviews

What actually happened

No.	Started	Finished	Task	Doc
1	07/11/21	08/11/21	Created the schedule	2.1.1
3	09/11/21	09/11/21	Daily scrum held	2.2.1
2	07/11/21	10/11/21	Created the storyboard	2.3.1
4	10/11/21	10/11/21	Customer meeting with answers analysed	2.4.1
5	12/11/21	12/11/21	End of sprint meeting held consisting of daily scrum, sprint review, sprint retrospective, sprint planning	2.5.1
6	Throughout		Backlog updated	2.6.1
7	Throughout		Exception handing	2.7.1
9	Throughout		User stories with acceptance tests	2.8.1

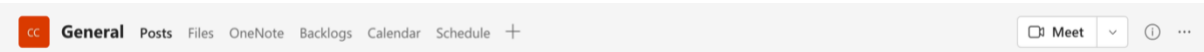
Schedule | 2.1.1

	A	B	C	D	E	F
1	**Date**	**08/11/21**	**09/11/21**	**10/11/21**	**11/11/21**	**12/11/21**
2	**Sprint**			**Sprint 2**		
3	**Qinyuan Zhuang**	Github				
4	**Lawrence Wong**	Storyboard				
5	**Callum Pitceathly**	User stories				
6	**Drew Dundee**	Product owner				
7	**Naomi Okiddy**	Github				
8	**Lucy Goodchild**	User stories				
9	**Ieuan Lambert**	Minutes				
10	**Tim Win**	Scurm master				
11	**Everyone**	Unity Essentials Learning Pathway				
12	**Daily scrum**		15:15 MS Teams			
13	**Daily scrum, sprint review, sprint retrospective, and sprint planning**					13:15 10W.5.01
14	**Customer meeting**			15:00 CB 5.13		
15						

TW thought Excel would be the simplest and easiest way to visualise a schedule

	A	B
1	**Roles**	**Artefacts**
2	**Scrum master**	**Product backlog**
3	1. Ensures Scrum understood and actioned	1. List of everything that might be needed in the product e.g. features, enhancements, and fixes
4	2. Facilitates Scrum events	2. Have the following attributes: description, order, time estimate, value
5	3. Responsible for solving issues development team have that are stopping progress	3. Updated as requirements change
6	4. Oversee documentation and ensure consistency	4. Higher ordered items are usually clearer, more detailed, and have more precise estimates
7	5. Book customer meetings	5. Items > 1 week are refined over time. When < 1 week, item is ready for selection during sprint planning
8	6. Takes minutes	
9		**Sprint backlog**
10	**Product owner**	1. Product backlog items selected for the sprint
11	1. Manage and refine product backlog	2. Items detailed with plan and definition of Done
12	2. Maximise value of development team	
13	3. Monitor team progress	**Other activities**
14	4. Leads meetings	**Backlogs (PO)**
15		*Friday 2:30-3:30pm*
16	**Development teams**	1. Refine product backlog
17	1. Self-organising - choose how best to accomplish your pair's sprint backlog items	2. Move items from product backlog to sprint backlog
18	2. Manage and refine sprint backlog	*Below was added on 19/11/21*
19	3. Deliver releasable increment of Done each sprint	**End of sprint code merge**
20	4. Determine what is accomplishable in future increments	1. By Friday noon
21	5. Ask product owner to change product backlog	2. All code pushed to master and combined
22	6. 1 member in pair produces diary for scrum master by end of sprint	3. If not working tell team before push
23		4. When committing add comments
24	**Meetings**	
25	**Daily scrum**	**Dev diaries**
26	*Monday 10:00-10:15* *Amended on 12/11/21*	Brief written information on what each dev team's done to SM/PO
27	*Wednesday 14:15-14:30* *Amended on 12/11/21*	
28	*Friday 13:15-13:25*	**Documentation**
29	1. What's been done in past 48hrs?	*Friday 2:30 - 3:30pm*
30	2. Were there any issues?	1. Download all branches as zip and put in sprint folder
31	3. Plan for next 48hrs? (not on Friday since we have sprint planning)	2. Merge all of the sprint's documentation into 1 pdf
32	4. Will there be any issues? (not on Friday since we have sprint planning)	
33		
34	**Sprint review**	
35	*Friday 13:25-13:45*	
36	1. Discuss sprint backlog (explain what's been done/not done)	
37	2. Demonstrate increment (code changes and additions) and answer questions	
38	3. Discuss product backlog	
39	4. Discuss likely completion date based on progress to date	
40	5. Review how requirements have changed	
41	6. Review what to do next	
42	7. Revise product backlog	
43		
44	**Sprint retrospective**	
45	*Friday 1:45-2:00 pm*	
46	1. How did the last sprint go?	
47	2. Identify and order improvements	
48	3. Plan to implement improvements	
49		
50	**Sprint planning**	
51	*Friday 2:00-2:30pm*	
52	1. Determine projected capacity and past performance	
53	2. Decide next increment goal	
54	3. Decide/confirm definitions of Done	
55	4. Define plan to achieve this goal and rota	
56		

A separate sheet was also added to outline the key points in our process



A tab was added to Teams so this file could be viewed without needing to open Excel

Daily scrum | 2.2.1

09/11/21, 15:15-15:30, Microsoft Teams

Attendance

All present
Meeting led by DD
Minutes taken by TW

Minutes

1. DD discussed his progress with the storyboard. He has created a 'flow chart' type chart with the different sections of the game that branch and go into more granular detail about how each element of the game works.

2. LW has created his own version of the storyboard which has pictures of similar games and text descriptions

3. Team briefly discussed what they expected the storyboard to look like. Some member thought it would look similar to a 6 panel comic strip but thought what DD and LW have created actually works better.

4. DD and LW will combine their work together into a integrated document

5. NO and QZ have setup our Github. We each need to register with our Bath email.

6. User stories team have not written anything yet, but they have done research on what user stories are and how they should be formatted. Explained they will continue work on this to have something by the end of the first sprint.

7. DD with assistance from LW will lead meeting with the customer tomorrow.

8. TW explained that he has completed the schedule and updated the backlogs. Ran out of time and this will take some time to cover so agreed this will be covered in the end of sprint meeting before actual development starts.

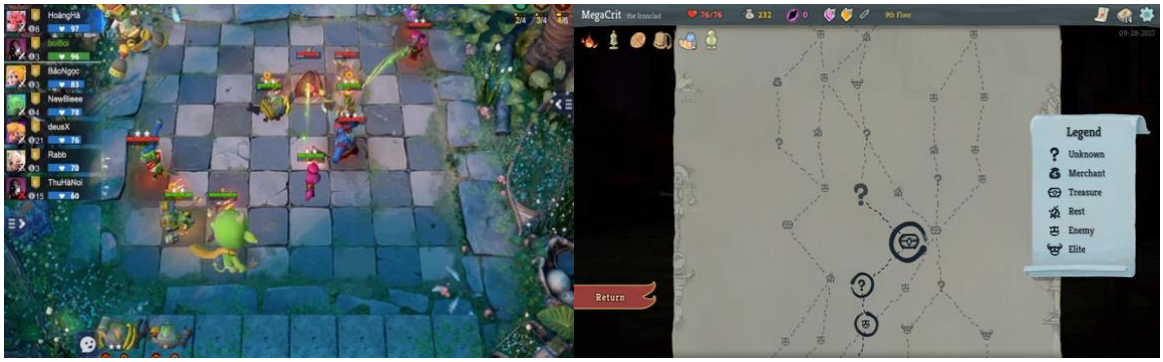
Storyboard | 2.3.1

Elevator pitch

You start a new game. You immediately see a tree-like map with branching paths that all lead to a big scary monster. There are safe and risky paths through the map that reward you according to the difficulty, so you must choose carefully. Along the paths you encounter enemies, bosses, vendors, and treasures. In each encounters, instead of directly controlling your characters, the battle is controlled by bots. What that means is as the player you choose your team composition, place them on the battle field strategically, and let the two sides fight it out themselves. The player defeats the enemy, collects rewards, upgrades their units, and prepares for the next fight.

The game combines the popular auto battler genre with the innovative branching path game mechanic resultng in original strategic gameplay that rewards knowledge and skill.

Influences



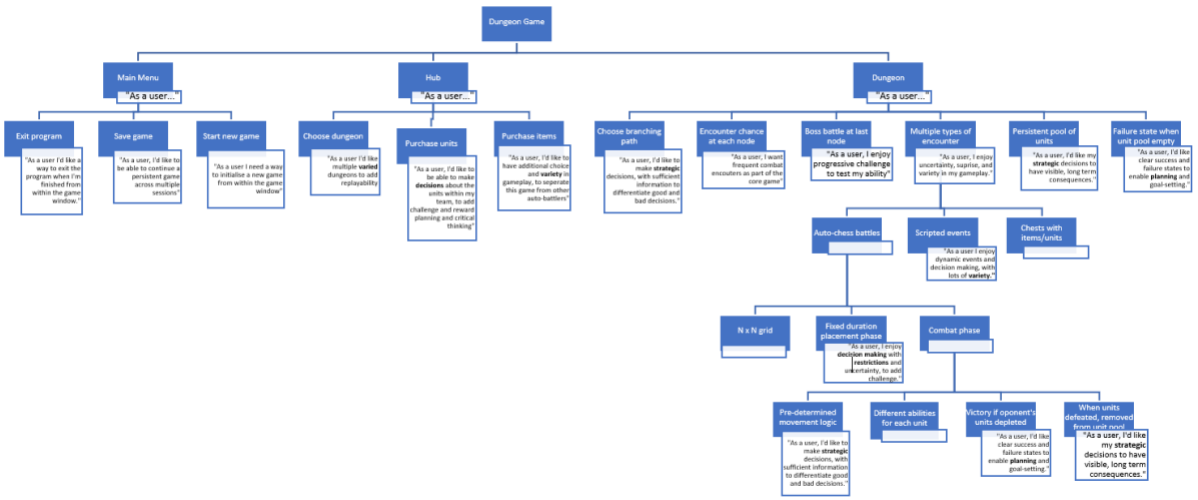
Auto Chess (left)

- Grid-based
- Battle favours careful planning, positioning, knowledge of you and your enemy's unit

Slay the Spire (right)

- Tree like map where each node is an encounter
- Branching paths provide risk vs. reward

Mechanics



Gameplay loop

- Explore map > choose next encounter > plan battle > position team > start battle > defeat enemy > upgrade units > repeat

Map

- A map with a tree like structure
- The player's goal is to reach the end of the map and defeat the end boss
- Nodes on the path will show some indication of relative risk and reward
- Minimal story but some basic story details could be sprinkled in-between battles

Battle

- The encounter starts with a stationary battle field
- The player must choose the unit they want to fight in the battle and place them strategically on the battle field
- The player hits 'start' and the units will fight automatically
- Each unit has its own way of attacking, range, speed, and behaviour
- Terrain on the grid may affect the battle
- The player cannot interfere with combat once the battle commences
- The player can change the outcome by building a well-rounded team and placing units in favourable positions
- Every room the player clears gives random rewards such as: currency to buy more units or upgrades, stat increases for their existing units, or new types of units
- Units follow RPG archetypes: warriors, tanks, ranged attackers, and healer

Customer meeting | 2.4.1

10/11/21, 15:00-15:15, CB 5.13

Attendance

All present
Recording taken by TW
Minutes taken by TW

Minutes

1. DD and LW explained the game step-by-step to customer from the perspective of the player

2. DD explained how we believed this game concept matches the themes we picked up during the initial discussion, namely - originality, strategic gameplay, and decisions with constraints/uncertainty. Customer agrees.

3. Customer happy that the autochess bot structure meets the requirements for bots

4. Customer likes the idea of using Unity and exporting using web.gl

5. Customer stresses the importance of making decisions with enough information to distinguish between good and bad decisions, and enough clear feedback to learn and improve as you play

6. Customer requests that by next meeting a basic prototype of either the dungeon map or the battle should be presentable. The team feel this is realistic.

7. Customer suggests that if minimum deliverables completed, to consider procedural generation of maps for replaybility, but otherwise pleased with idea in its current form

8. Customer mentioned that our idea a procedurally generated maps and battles was similar to a mechanic in Devil May Cry which we might want to look into if we are interested

9. JP notes that the sprint documentation should be finished as each sprint is completed

10. JP notes our user stories need acceptance tests so we have a definition of done and how the user story is achieved

Analysis

1. The tone of the customer meeting was positive particularly in regards to how original our game idea was (particularly as we knew the customer was asking lots of games from different teams to be built). This was good for morale.

2. Although the customer said to get an early prototype done for either the map or the battle, the team were very keen to make progress on both fronts. This is so there aren't too many people working on the same thing at the same time and many team members wanted to get stuck into game development.

3. TAs emphases that it would be a good idea to have a dedicated member of the team to integrate all of the documents being produced. We agreed and believed this should be the scrum master.

End of sprint meeting | 2.5.1

12/11/21, 13:15-14:30, 10W.5.01

Attendance

All present
Meeting led by DD
Minutes taken by TW

Daily scrum

1. TW had added a bit more detail about the scrum meetings and role responsibilities in the Schedule document

2. CP and LG have finished the first version of user stories and acceptance tests. The information about acceptance tests from the TAs was helpful.

3. Half of the team has made a start with learning Unity

Sprint review

1. TW explained the schedule which outlines the rota and meetings

2. TW explained the redesigned Kanban board to be used now that development will start

3. TW has started on the sprint 1 documentation and made some templates for it but it is still in development

4. NO and QZ outlined how the Github they setup works. There are separate branches that you work off so that separate teams don't corrupt each others' work. There is the pre-release branch which we merge our work into at the end of each sprint. When the pre-release branch is tested, we merge it into the master branch, delete the other branches and then base all of the next sprint's work off this master branch.

5. CP and LG showed the user stories and acceptance tests they made and explained how they should link to the more deteailed use cases which need to be created in the next sprint.

Sprint retrospective

1. Unanimously agreed that the virtual meeting on Sunday evening was not effective as team members' were tired and it was difficult to communicate. Team decided to try to avoid weekend, evening and virtual meetings where possible in the future.

2. The team also agreed that it seemed uneccessary to have the daily scrum on Tuesday, just before we would meet in the lab on Wednesday for the customer meeting. It would be more efficient to have the daily scrum just before the Wednesday customer meeting. This will therefore be rearranged in the future. TW to update the schedule.

3. We weren't all clear on what was meant by a storyboard so we could have discussed this in greater detail before the task was assigned. DD and LW agreed they could have made the initial draft together rather than combining their work after the daily scrum.

4. Some team members felt that following the scrum methodology rigidly does lead to some redundancy e.g. the daily scrum, review, and retorspective tended to have overlapping themes. Team agreed it was acceptable to have this overlap and that the rigidity of the distinct parts our end of scrum meeting made it difficult to know when to say what. Agreed to keep the agenda for our end of scrum meeting but make it more flexible i.e. we would timebox the meeting as a whole but not the individual parts within the meeting. Also agreed it would be helpful to add further info on how these individual parts differ in the schedule.

5. TW explained that although we have formal documentation, there is still utility in the shared Onenote for informal notetaking / drafts / unfiled notes

Sprint planning

1. The team agreed that we should rotate all of the roles starting from sprint 3 onwards. This was to keep things fair (so one person wouldn't have to be the scrum master every week), to ensure team members' had knowledge in many areas, and because we were all keen to learn and experience the different aspects of game development as this was primarily a learning exercise.

2. TW has created a Unity youtube playlist which he will share

3. As we did not all have the time to learn the basics of Unity, we agreed that this is something we should all continue to work on in the next sprint

4. TW had been doing a lot of the notetaking for sprints 1 and 2 so keen to be scrum master in sprint 3 and ensure the documentation for previous sprints was up-to-date

5. Following from the customer meeting, the team agreed that they would like to work on both the battling and the map simultaneously for the first prototype.

6. Team split into 3 pairs to allow for pair programming focused on the map, grid, and combat

7. DD keen to be product owner

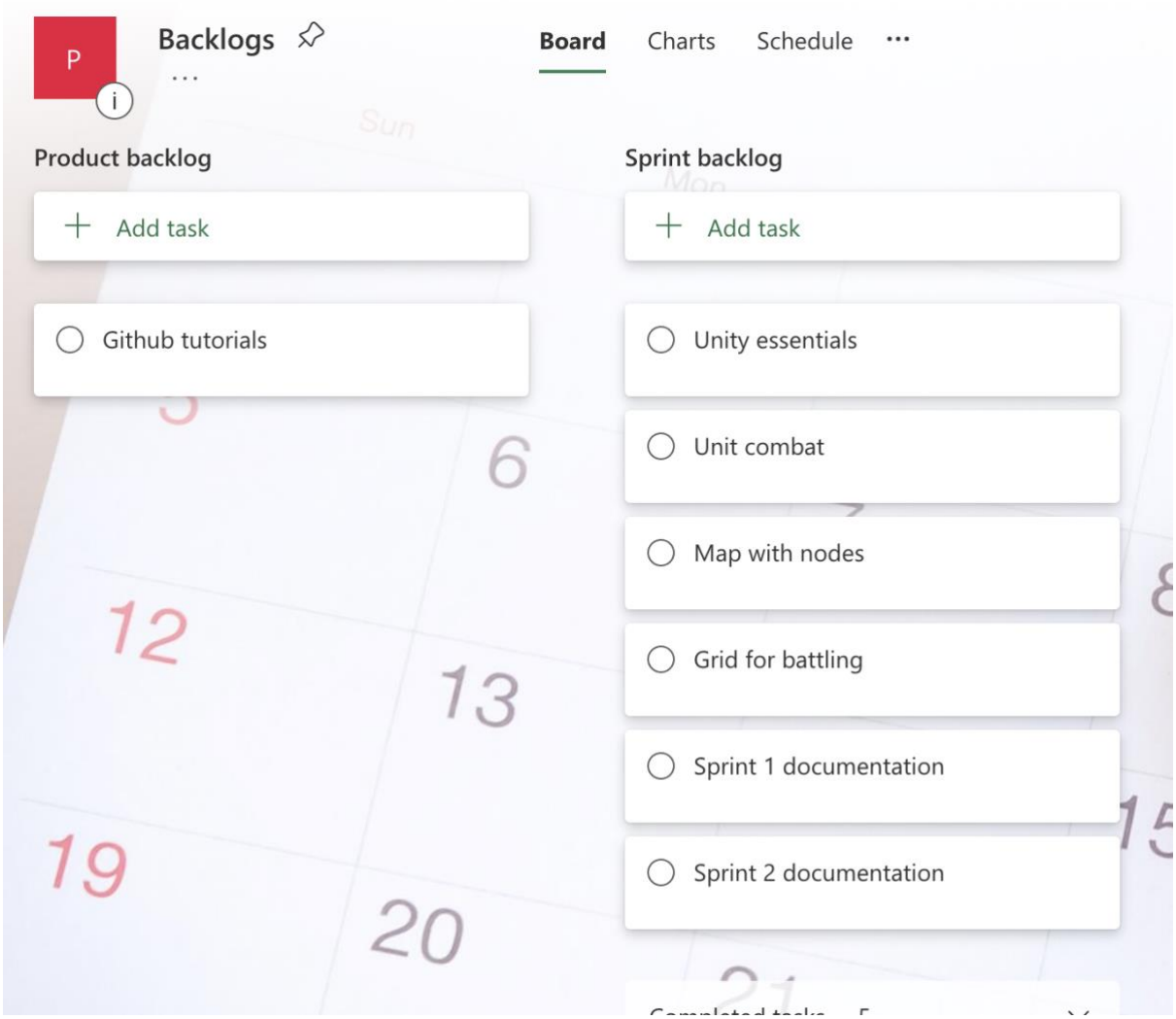
8. NO has found an open source pixel art program which she recommends and will share

Task assignments

	A	G	H	I	J	K	L	M
1	**Date**	**13/11/21**	**14/11/21**	**15/11/21**	**16/11/21**	**17/11/21**	**18/11/21**	**19/11/21**
2	**Sprint**				**Sprint 3**			
3	**Qinyuan Zhuang**	Map						
4	**Lawrence Wong**	Grid						
5	**Callum Pitceathly**	Combat						
6	**Drew Dundee**	Product owner						
7	**Naomi Okiddy**	Grid						
8	**Lucy Goodchild**	Combat						
9	**Ieuan Lambert**	Map						
10	**Tim Win**	Scrum master						
11	**Everyone**	Unity Essentials Learning Pathway						
12	**Daily scrum**			10:00 MS Teams		14:15 CB 5.13		
13	**Daily scrum, sprint review, sprint retrospective, and sprint planning**							13:15 10W.5.01
14	**Customer meeting**					15:00 CB 5.13		
15								

Backlog | 2.6.1

Snapshot at end of sprint



The new verion of our backlogs using Microsoft Planner

Commentary

1. Apart from everyone downloading and getting to grips with Unity and sprint 1 documentation, all other tasks for this sprint were completed

Exception handling | 2.7.1

Unity Essentials Learning Pathway

1. As this sprint was shorter, those with more tasks didn't have the time to download and get to grips with Unity yet

2. We tried to pair up the development teams so there was at least 1 person who had progressed further with the tutorials

3. This activity will have to continue into the next sprint. It is possible that because of this, learning more about Github was delayed which led to some issues with merging and version control in later sprints.

Completing sprint 1 documentation

1. This task wasn't assigned as we hadn't properly set up scrum and it we weren't as clear on the documentation requirements

2. The documentation is not yet complete, but it is started. This process will definitely get easier once the first sprint is done so we have a good basis for what the documents should contain which we can then replicate in future sprints.

3. Further adds to the importance of this start as there is a risk of neglecting this task and having and the documentation workload getting out of hand.

4. Proper documentation done in a timely manner also benefits the team as it ensure each team member knows what they're doing especially if they're absent, allows us to refer back to previous meetings, and enables us to verify we are building the correct features.

5. As the first sprint's documentation will take the longest, liberal use of the OneNote will be necessary to take notes for sprint 3 rather than waiting for the templates to be ready. This ensure we don't forget what was done or discussed.

User stories with acceptance tests | 2.8.1

User stories have been elucidated from the initial user stories on the slide "What the customer (thinks they) want(s)?" and from our first customer meeting.

US_01_1: Description: "As a user, I'd like a way to exit the program when I'm finished, from within the game window"
US_01_2: Acceptance Criteria: Accepted when quit game button implemented in main menu

US_02_1: Description: "As a user, I'd like to be able to continue a persistent game across multiple sessions"
US_02_2: Acceptance Criteria: Accepted when save game system implemented

US_03_1: Description: "As a user, I need a way to initialise a new game from within the game window"
US_03_2: Acceptance Criteria: Accepted when new game button implemented in main menu

US_04_1: Description: "As a user, I'd like multiple varied dungeons to add replayability"
US_04_2: Acceptance Criteria: Accepted when more than one dungeon implemented

US_05_1: Description: "As a user, I'd like to be able to make decisions about the units within my team, to add challenge and reward planning and critical thinking"
US_05_2: Acceptance Criteria: Accepted when able to select from multiple unit types during planning phase of combat

US_06_1: Description: "As a user, I'd like to have additional choice and variety in my gameplay, to separate this game from other auto-battlers"
US_06_2: Acceptance Criteria: Accepted when implement both persistent unit pools on main map, as well as being able to choose from branching paths

US_07_1: Description: "As a user, I'd like to make strategic decisions, with sufficient information to differentiate good and bad decisions"
US_07_2: Acceptance Criteria: Accepted when able to clearly see indication of reward and difficulty on each dungeon path.

US_08_1: Description: "As a user, I want frequent combat encounters as part of the core game"
US_08_2: Acceptance Criteria: Accepted when nodes in dungeon graph implemented, with encounters possible at each, including fully implemented auto-battling system

US_09_1: Description: "As a user, I enjoy progressive challenge to test my ability"
US_09_2: Acceptance Criteria: Accepted when persistent unit pool implemented, as well as progressively harder encounters culminating in "boss" encounter

US_10_1: Description: "As a user, I enjoy uncertainty, surprise, and variety in my gameplay"
US_10_2: Acceptance Criteria: Accepted when multiple encounter types implemented, with hidden possibility of each at given node

US_11_1: Description: "As a user, I'd like my strategic decisions to have visible, long-term consequences."
US_11_2: Acceptance Criteria: Accepted when persistent unit pool implemented, alongside choice of multiple branching paths

US_12_1: Description: "As a user, I enjoy dynamic events and decision making, with lots of variety"
US_12_2: Acceptance Criteria: Accepted when multiple encounter types implemented, with hidden possibility of each and given node

US_13_1: Description: “As a user, I enjoy decision making with restrictions and uncertainty to add challenge”
US_13_2: Acceptance Criteria: Accepted when planning phase implemented to combat with time restriction, alongside choice of branching paths

US_14_1: Description: “As a user, I’d like clear success and failure states to enable planning and goal setting”
US_14_2: Acceptance Criteria: Accepted when persistent pool of units implemented, with game-over occurring when depleted

US_15_1: Description: “As a user, I would like to understand the game’s UI/UX clearly, so they don’t interfere with gameplay”
US_15_2: Acceptance Criteria: Accepted when no elements of gameplay obscured behind UI but also all relevant information provided on UI

US_16_1: Description: “As a client, I would like the game to be accessible to as many people as possible so I can share it with friends/peers or monetise it in future”
US_16_2: Acceptance Criteria: Accepted when game is playable in browser across multiple devices

Sprint 3: 12/11/21-19/11/21

The focus of sprint 3 was creating the first prototype of our game.

Overview

What's supposed to happen this sprint

No.	Expected Start	Expected Finish	Objective
1	12/11/21	19/11/21	Build the first prototype of the combat scene
2	12/11/21	19/11/21	Build the first prototype of the map scene
3	12/11/21	19/11/21	Create the second version of our user stories with acceptance testing
4	12/11/21	19/11/21	Create the first version of our requirements use cases
5	12/11/21	19/11/21	Create the first version of our CRC cards
6	12/11/21	19/11/21	Create the first version of our design use cases
7	12/11/21	19/11/21	Create the first version of our user interface design

Reviews

What actually happened

No.	Started	Finished	Task	Doc
1	15/11/21	15/11/21	Daily scrum held	3.1.1
2	17/11/21	17/11/21	Daily scrum held	3.2.1
3	17/11/21	17/11/21	Customer meeting with answers analysed	3.3.1
4	19/11/21	19/11/21	End of sprint meeting held consisting of daily scrum, sprint review, sprint retrospective, sprint planning	3.4.1
5	Throughout		Backlog updated	3.5.1
6	Throughout		Exception handing	3.6.1
7	Throughout		User stories with acceptance tests	3.7.2
8	Throughout		Requirements use cases	3.8.1
9	Throughout		CRC cards	3.9.1
10	Throughout		Design use cases	3.10.1

Daily scrum | 3.1.1

15/11/21, 10:00-10:15, Microsoft Teams

Attendance

All present
Meeting led by DD
Minutes taken by TW

Minutes

1. LG and CP have made progress on the units that battle. They may assign proper sprites to them if there is time but otherwise will use a basic shape like a square

2. QZ and IL have found an open source map that is inspired by the one used in one of our main influences - Slay the Spire. It is very similar to what we want, but they do not understand how it works yet. It may be useful to show the customer how we intend the map to look like.

3. NO and LW have made progress with the board. They are currently working on how to make it so the squares are highlighted when the player hovers their mouse over them.

4. We will need to test how to integrate each pair's work together tomorrow and Wednesday morning

5. DD to coordinate the pairs' work. If we can't figure this out by the customer meeting, we can show him what everyone's been working on separately

6. Reminder from NO that each team should create and work on their own branch

Daily scrum | 3.2.1

17/11/21, 14:15-14:30, CB 5.13

Attendance

All present
Meeting led by DD virtually
Minutes taken by TW

Minutes

1. Not much progress made since the last meeting

2. We were unable to complete the merge so we should aim to do this by Friday

3. Development teams will continue work on their sections

4. TW has finished the documentation for sprint 1 so will start on sprint 2. There was a bit of confusion concerning the dates of the sprints as they were uneven lengths but this has now been resolved and it should be business as usual from here on out.

5. All agreed that it would be useful for each development pair to send the scrum master and product owner a brief 'dev diary' which outlines their work over the sprint before the end of sprint meeting which they can then go over in more detail in the sprint review. This would help with documentation and when we rotate roles.

Customer meeting | 3.3.1

17/11/21, 15:00-15:15, CB 5.13

Attendance

All present although DD joined virtually
Recording taken by TW
Minutes taken by TW

Minutes

1. QZ talked through the map using the version found on Github. Customer remarked it was a very pleasnt visualisation.

2. Customer commented that some parts could be shorter but have more difficult challenges.

3. IL explained how at the moment, we have ideas for 3 different nodes - a question mark for a mystery reward or battle, a chest, and a batlle.

4. LW presented an early version of the 8x8 grid which highlights where the mouse hovers

5. LG presented how the units would interact

6. Customer asked whether from the player's point of view, when the battle happens, they would see everything happening at the same time as we presented it this way. The customer said this would make things harder to learn from observing but this would add to the challenge. Alternatively, it might be an idea to allow the player to move through the turns step-by-step so they can learn how the units interact.

7. Customer seemed enthusiastic and said it was a really nice game conceptually and it doesn't feel like every other game he's seen

8. Customer said we have a lot to do noted that the simplest way we could make a playable game with a start, middle, and end, would be to have the simplest form of a map with the simplest node type. Then we can add things to this such as the function to move from the map scene to the battle scene and back again

9. JP asked us about documentation. TW replied saying we have sprint 1 complete with a documentation process in place. JP explained the importance of document numbers for cross referencing, and that after each sprint we should put it in a pdf then combine all sprints at the end.

Analysis

1. The customer's enthusiasm for our game concept further increased our confidence in the idea which was important especially for those with less knowledge of the autochess genre.

2. The replay function described by the customer would be desirable but potentially difficult to develop. Having the battle happen in real time would be easier and, as the customer mentioned, the challenge could come from this. Perhaps there would be ways to implement a function that achieves the same result but more simply such as a killfeed, damage numbers or a replay function. Added as US_18_1 (3.7.2).

3. Making the simplest working version of our game would be a more effective approach rather than trying to build complex features from scratch. This is something we may have forgotten when some development effort continued to go towards procedurally generated maps (see sprint planning 3.4.1).

End of sprint meeting | 3.4.1

19/11/21, 13:15-14:30, 10W.5.01

Attendance

All present although DD and CP joined virtually
Meeting led by TW
Minutes taken by TW

Daily scrum

1. TW briefly went through the documentation process. He explained the document naming system and where the templates were located.

2. NO, LW and IL, QZ have finished their work for this sprint and pushed this to their branch.

3. LG has just finished reviewing CP's code which they have pushed to their branch.

Sprint review

1. Each development pair sent TW a 'dev diary' which outlines their work this sprint.

2. NO met up with LW over the weekend and built the board and basic highlight. They found this difficult as they hard coded the grid. Once they discovered how to use tilemaps, it became a lot easier.

3. LW created a base unit class in which other units (players and enemies) can inherit methods from. Methods are setBaseHealth - which set the base health of the enemy or player unit, and takeDamage which takes away an inputted health amount from player/enemy unit. LW wrote an engagement function which scans for enemy units and checks how close they are. When within a certain range it will output damage to the unit within range. This is based on an unfinished function for checking where the closest enemy unit is.

4. QZ found the open source map over the weekend. He then worked to understand how to use it and updated it from to work with the current version of Unity. He then made the map project a package so everyone can easily download and import it.

Sprint retrospective

1. Team agreed we've now settled into a status quo. The number of meetings is good and the pace and quantity of work is manageable currently.

2. CP is still learning Unity but received a lot of support from LG. Remarked that pair programming was extremely effective as a development and learning technique. After the first pair programming session it became easier to split up and make further developments.

3. Everyone was working off different branches some of which became quite messy so it became slightly confusing. Some people were using Github Desktop while others used a terminal window. Some features are easier to do with the desktop app, but sometimes you can only use the terminal window e.g. to fix conflicts.

4. The merge didn't happen until Friday as team member's were not all comfortable merging their work in case they made errors. LW explained changed could be rollbacked. TW point out FM's post on Moodle on 18/11/21 with further information about learning git which the team thought we would be useful to review.

5. Time boxing for meetings hasn't been strictly enforced. Sometimes it is difficult to do so e.g. if people are late for the daily scrum. Because we are all still learning some things will take longer

than we expect at the beginning like merging branches. Agreed that while timeboxing is important, the scrum master won't enforce it as strictly until later sprints.

6. IL had questions about writing tests for our code. NO suggested writing some tests in natual language before developing, and then writing some tests to see whether your code works based on this.

Sprint planning

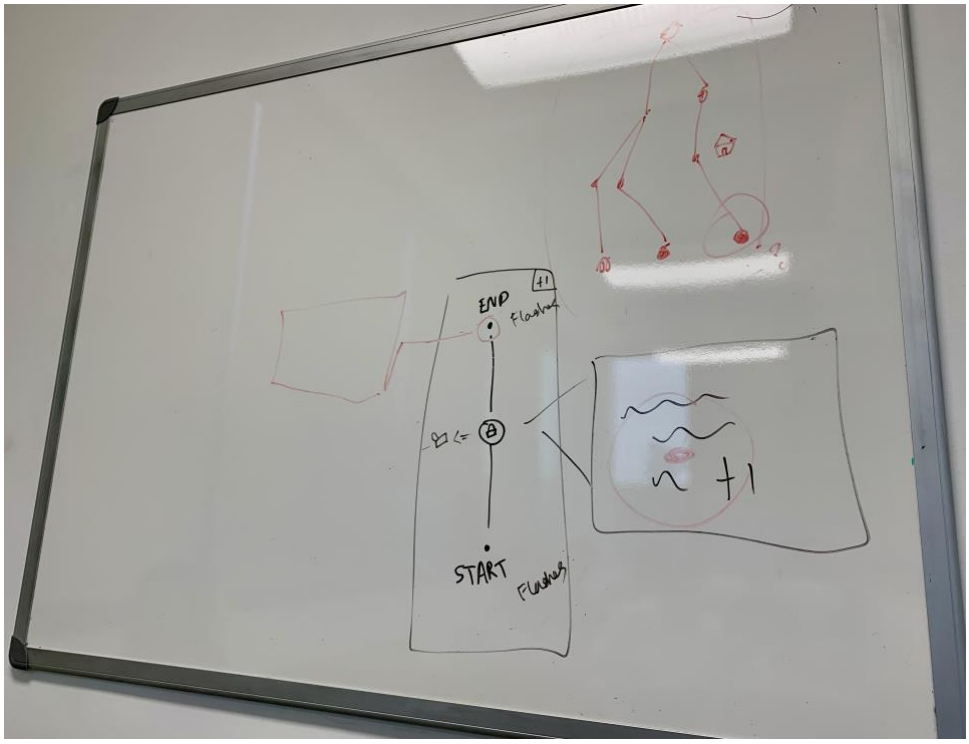
1. Merging everyone code was delayed so this was dealt with first as a priority. LW completed the merge and there were no conflicts. The old branches were then downloaded and deleted. LW went over how the process is to first merge to the pre-release branch, check this works, and then merge this into the master branch. The next sprint's branches will then be based on this master branch. Although you can approve your own pull request, it's best practice to request another reviewer. LW stressed that commits should be made with comments and be named meaningfully.

2. Discussed capacity. CP has more capacity this sprint as he's now recovered. IL does not have capacity this weekend

3. Team agreed that full role rotation is difficult because there's a lot to learn each sprint and some team member's are more attached to their work now that they've started / have developed the skills to now make much more progress. Other team members were very keen to rotate into new roles. Agreed compromise of staggered pair rotations where 1 team member in each pair swaps role. The remaining pair member does a handover to the new pair member to get them up to speed.

4. Discussed the minimum possible version of our game as per the customer meeting. This would be works on by 2 pairs while the other pair continues work on the battle scene. The first pair will work on the map (nodes, connectors), while the other pair works on the UI (pop ups, unit counter). We will then try to integrate the battle scene with the map scene by having the end node be a battle node that forces a scene change. This represents the boss battle at the end of the path.

5. Stretch goals are to have flashing nodes to indicate which one is active and for the path to be greyed out once it's been traversed.



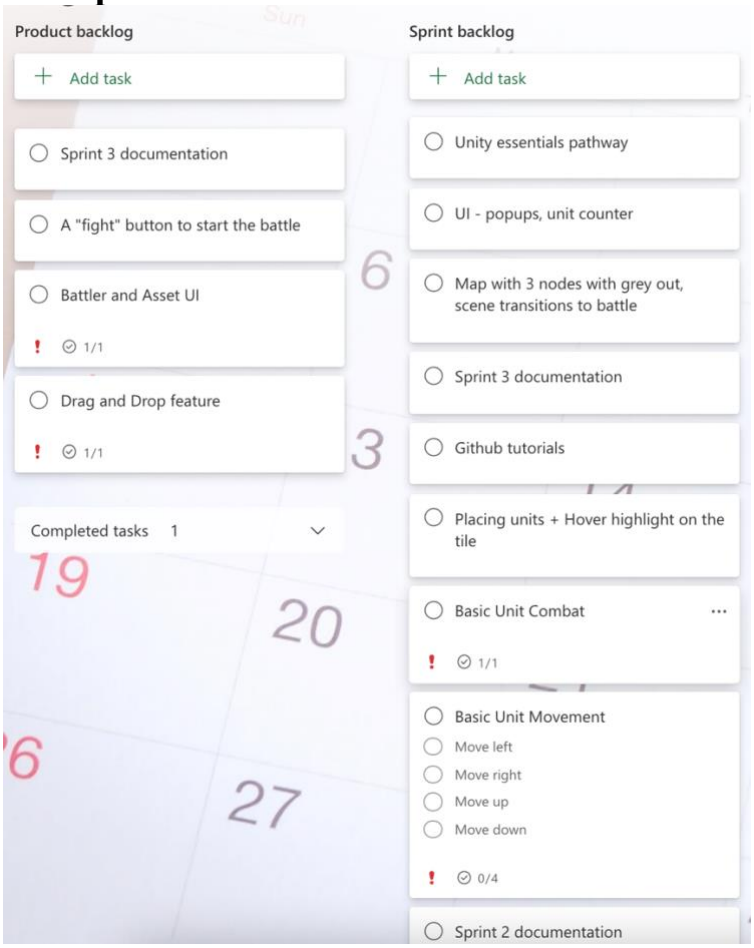
The simplest version of our original map for us to develop next sprint

Task assignments

	A	N	O	P	Q	R	S	T
1	**Date**	**20/11/21**	**21/11/21**	**22/11/21**	**23/11/21**	**24/11/21**	**25/11/21**	**26/11/21**
2	**Sprint**	**Sprint 4**						
3	**Qinyuan Zhuang**	Battle						
4	**Lawrence Wong**	Battle						
5	**Callum Pitceathly**	UI						
6	**Drew Dundee**	UI						
7	**Naomi Okiddy**	Product owner						
8	**Lucy Goodchild**	Scrum master						
9	**Ieuan Lambert**	Map						
10	**Tim Win**	Map						
11	**Everyone**	Learn Github						
12	**Daily scrum**			10:00 MS Teams		14:15 CB 5.13		
13	**Daily scrum, sprint review, sprint retrospective, and sprint planning**							13:15 NH 2.17a
14	**Customer meeting**					15:30 CB 5.13		
15								

Backlog | 3.5.1

Snapshot at end of sprint



Commentary

1. The tasks set out this sprint were completed with the exception of the unity essentials pathway and sprint 2 documentation

2. Product owner and development teams may need to flesh out some of these tasks further and although slightly arduous, fill out the fields such as deadline, assignments etc, so it's clear to everyone how development is progressing

3. There is a bit of a lag with the end of sprint documentation as it takes some time for the end of sprint meeting to be written up. TW will have to manage writing up sprints 2 and 3 while working on the map so was supported by LG in sprint 4 with this.

Exception handling | 3.6.1

Virtual meeting disruptions

1. DD joined Wednesday and Friday meetings virtually as he was waiting for Covid test results. CP also joined the Friday meeting virtually.

2. DD explained there was lots of background noise in the customer meeting so having the recording was helpful

3. DD and CP found it difficult to hear and participate in the end of sprint meeting. The room was unusually loud and busy. We would have also benefitted from a large screen display in addition to a whiteboard so we could do code reviews / LW could have demonstrated a live branch merge.

4. In future the scrum master will book a room for the team meeting.

5. We should continue doing dev diaries as they helped guide the sprint review especially during the disruptive meeting.

User stories with acceptance tests | 3.7.2

US_01_1: Description: "As a user, I'd like a way to exit the program when I'm finished, from within the game window"
US_01_2: Acceptance Criteria: Accepted when quit game button implemented in main menu

US_02_1: Description: "As a user, I'd like to be able to continue a persistent game across multiple sessions"
US_02_2: Acceptance Criteria: Accepted when save game system implemented

US_03_1: Description: "As a user, I need a way to initialise a new game from within the game window"
US_03_2: Acceptance Criteria: Accepted when new game button implemented in main menu

US_04_1: Description: "As a user, I'd like multiple varied dungeons to add replayability"
US_04_2: Acceptance Criteria: Accepted when more than one dungeon implemented

US_05_1: Description: "As a user, I'd like to be able to make decisions about the units within my team, to add challenge and reward planning and critical thinking"
US_05_2: Acceptance Criteria: Accepted when able to select from multiple unit types during planning phase of combat

US_06_1: Description: "As a user, I'd like to have additional choice and variety in my gameplay, to separate this game from other auto-battlers"
US_06_2: Acceptance Criteria: Accepted when implement both persistent unit pools on main map, as well as being able to choose from branching paths

US_07_1: Description: "As a user, I'd like to make strategic decisions, with sufficient information to differentiate good and bad decisions"
US_07_2: Acceptance Criteria: Accepted when able to clearly see indication of reward and difficulty on each dungeon path.

US_08_1: Description: "As a user, I want frequent combat encounters as part of the core game"
US_08_2: Acceptance Criteria: Accepted when nodes in dungeon graph implemented, with encounters possible at each, including fully implemented auto-battling system

US_09_1: Description: "As a user, I enjoy progressive challenge to test my ability"
US_09_2: Acceptance Criteria: Accepted when persistent unit pool implemented, as well as progressively harder encounters culminating in "boss" encounter

US_10_1: Description: "As a user, I enjoy uncertainty, surprise, and variety in my gameplay"
US_10_2: Acceptance Criteria: Accepted when multiple encounter types implemented, with hidden possibility of each at given node

US_11_1: Description: "As a user, I'd like my strategic decisions to have visible, long-term consequences."
US_11_2: Acceptance Criteria: Accepted when persistent unit pool implemented, alongside choice of multiple branching paths

US_12_1: Description: "As a user, I enjoy dynamic events and decision making, with lots of variety"
US_12_2: Acceptance Criteria: Accepted when multiple encounter types implemented, with hidden possibility of each and given node

US_13_1: Description: "As a user, I enjoy decision making with restrictions and uncertainty to add challenge"

US_13_2: Acceptance Criteria: Accepted when planning phase implemented to combat with time restriction, alongside choice of branching paths

US_14_1: Description: “As a user, I’d like clear success and failure states to enable planning and goal setting”
US_14_2: Acceptance Criteria: Accepted when persistent pool of units implemented, with game-over occurring when depleted

US_15_1: Description: “As a user, I would like to understand the game’s UI/UX clearly, so they don’t interfere with gameplay”
US_15_2: Acceptance Criteria: Accepted when no elements of gameplay obscured behind UI but also all relevant information provided on UI

US_16_1: Description: “As a client, I would like the game to be accessible to as many people as possible so I can share it with friends/peers or monetise it in future”
US_16_2: Acceptance Criteria: Accepted when game is playable in browser across multiple devices

US_17_1: Description: “As a user, I would like to clearly see each action being taken sequentially rather than simultaneously, to ensure I can learn and adapt.”
US_17_2: Acceptance Criteria: Accepted when actions within battle are made and displayed sequentially with delay between.

US_18_1: Description: “As a user, I would like to be able to review the sequence of events after a “battle” in ensure I can learn and adapt.”
US_18_2: Acceptance Criteria: Accepted when user is able to either re-watch battle, or view some representation of the events that took place afterwards

Requirements use cases | 3.8.1

UC_01:
Use Case: Load Game
Scope: Main menu
Level: User Goal
Context: Users can load a game progression by selecting the “load game” option from the main menu
Users are unable to select a “load game” option from the main menu
Frequency of occurrence: Any time the player enters the programme or wishes to load a game from the map scene
Open issues:

How to ensure a failsafe option to save game prompt in case player wishes to return to original game progression?

UC_02:
Use Case: Exit game
Scope: Main menu
Level: User Goal
Context: Users can exit the game by selecting the “exit game” option from the main menu
Users are unable to select “exit game” option from the main menu
Frequency of occurrence: Any time the player wishes to start exit the game.
Open issues:

How to ensure a failsafe option to save game prompt in case player forgets to save their progress before quitting the programme.

UC_03:
Use Case: Save Game
Scope: N/A
Level: User Goal
Context: Users can save a game by selecting the “save game” option from the main menu
Users are unable to select the “save game” option from the main menu
Frequency of occurrence: Any time the player wishes to save a game.
Open issues:

How do we store saves?
How many saves can be stored by the game?
How many characters can be used in naming the save?
Can we implement a failsafe in case two saves are given the same name?

UC_04:
Use Case: Main menu
Scope: N/A
Level: Subfunction
Context: Users can decide to save load or exit the game by entering the main menu
Users are unable to save load or exit the game as they cannot access the main menu
Frequency of occurrence: Any time the player enters the game or wishes to save, load or exit the game.
Open issues:

Do we display this as a popup or a scene?

UC_05:
Use Case: Popup displayed
Scope: N/A
Level: Subfunction
Context: Users are informed of a consequence when they make a game action.

Users fail to be informed of the consequence of a game action taken
Frequency of occurrence: Any time the player makes a game action.
Open issues:
How do we ensure consistent alignment of popups any time they are instantiated?
How do we set the informational text of each popup to be dependent on the action taken?
How do we ensure the background is still visible when a popup is triggered?

UC_06:
Use Case: Enter treasure node
Scope: N/A
Level: Subfunction
Context: Upon entering a treasure node, users are rewarded with extra units.
Upon entering a treasure node, users fail to be rewarded with extra units.
Frequency of occurrence: Any time the player enters a treasure node.
Open issues:
How is the number of units rewarded determined?
How do we link this to the update unit pool functionality?

UC_07:
Use Case: Welcome node
Scope: N/A
Level: Subfunction
Context: Users can select the welcome node to initiate progression through the map, and are informed of their objective.
Users can select the welcome node to initiate progression through the map but aren't informed of their objective
Frequency of occurrence: Any time the player enters the map scene of a new game.
Open issues:

UC_08:
Use Case: New Arena node
Scope: N/A
Level: Subfunction
Context: Users can select a subsequent new arena node and are informed they are progressing to a new arena.
Users can select a subsequent new arena node but aren't informed they're progressing to a new arena.
Frequency of occurrence: Any time the player can progress to a new arena.
Open issues:

UC_09:
Use Case: Game completion node
Scope: N/A
Level: Subfunction
Context: Users can select the game completion node to complete the game and are informed they have won.
Users can select the game completion node to complete the game but aren't informed they have won.
Frequency of occurrence: When the player has completed the final battle encounter of the game.
Open issues:

UC_10:
Use Case: Update unit pool
Scope: N/A

Level: User goal
Context: The number of units of the unit count and unit pool can be updated.
The number of units of the unit count and unit pool cannot be updated.
Frequency of occurrence: Every time the player is awarded units.
Open issues:
How do we ensure there is no conflict between the unit count number and the unit pool contents?

UC_11:
Use Case: Enter battle node
Scope: N/A
Level: Subfunction
Context: Users can select a battle node which initiates the grid scene.
Users can select a battle node but the grid scene fails to be initialised.
Frequency of occurrence: When the player has a battle node adjacent to their current node.
Open issues:
How do we trigger grid scene initialisation.
How do we ensure the grid scene initialised is the correct encounter for the current node (ie enemy difficulty etc) ?

UC_12:
Use Case: Enemy unit health scaling
Scope: N/A
Level: User goal
Context: Users can select the game completion node to complete the game and are informed they have won.
Users can select the game completion node to complete the game but aren't informed they have won.
Frequency of occurrence: When the player has completed the final battle encounter of the game.
Open issues:
How is scaling factor per node determined?

UC_13:
Use Case: Highlight tile
Scope: N/A
Level: Sub function
Context: When a user moves their cursor over a tile in the grid scene the tile is highlighted
When a user moves their cursor over a tile in the grid scene the tile remains unchanged.
Frequency of occurrence: Any time the user is in the grid scene and moves their cursor over a tile.
Open issues:
How do we identify which tile the cursor is over?
What colour should the tile be highlighted?

UC_14:
Use Case: Select highlighted tile
Scope: N/A
Level: User goal
Context: User can select a highlighted tile and the programme recognises the action
Users cannot select the highlighted tile
Frequency of occurrence: Whenever the cursor is above a tile and it is highlighted.
Open issues:
How do we ensure selection of the tile is linked to a distinct action?

UC_15:
Use Case: Place units on the grid

Scope: N/A
Level: User goal
Context: User can select a unit and place it on a grid tile
User can select a unit but cannot place it on a grid tile
Frequency of occurrence: Whenever the user is in the unit placement sequence.
Open issues:
How do we select a unit?
How do we ensure a unit is placed when the tile is selected?
How do we ensure the placed unit is deducted from the unit pool?

UC_16:
Use Case: Unit constant damage output
Scope: N/A
Level: Sub function
Context: During battle sequence, units will output damage consistently which can then be accessed by opposing units.
During battle sequence, units will output damage consistently but output can't be accessed by opposing units.
Frequency of occurrence: Every time a fight sequence is initiated, for every unit.
Open issues:
How do we determine damage output and frequency?
When should damage output be initiated?
How is each unit's damage output accessed by opposing units?

UC_17:
Use Case: Unit take damage
Level: Sub function
Context: During battle sequence, units will take damage if they are within range of an opposing unit
During battle sequence, units are unable to take damage when in range of an opposing unit.
Frequency of occurrence: Whenever a unit is within range of an enemy unit.
Open issues:
How do we determine when a unit is within range of another?
How does a unit access an opposing unit's damage output?
How is health deducted from the unit's health points variable?

UC_18:
Use Case: Unit death
Level: Sub function
Context: During battle sequence, when a unit's health subceeds zero it will be eliminated from the battle
During battle sequence, when a unit's health subceeds zero it fails to be eliminated from the battle
Frequency of occurrence: Whenever a unit's health subceeds zero.
Open issues:
How to we eliminate an object from the battle?
How do we give a unit a "dead" attribute?

UC_19:
Use Case: Play game in browser
Level: Summary
Context: Users can access the game through a web browser
Users cannot access the game through a web browser
Frequency of occurrence: Dependent on User
Open issues:
What hosting service do we upload the game to?

How do we make the game as accessible as possible?
What devices will be able to run the game online?

CRC cards | 3.9.1

GridManager **References – UC_03**	
RESPONSIBILITIES: **Responsible for creating and containing a data structure containing all Tiles of grid and their positions** **Responsible for making this data available publicly to other functions**	COLLABORATORS: Tiles BaseUnit Renderer

Tiles **References - UC_03**	
RESPONSIBILITIES: **Responsible for containing sprite to be rendered on game window to represent a tile** **Responsible for listening for mouse click and then altering sprite to indicate selection of tile** **Responsible for listening for mouseover and then altering sprite to indicate the tile is highlighted**	COLLABORATORS: Tiles BaseUnit Renderer

Renderer **References – UC_01, UC_02, UC_03, UC_04**	
RESPONSIBILITIES: **Responsible for drawing all active sprites onto the game window**	COLLABORATORS: Tiles BaseUnit Dungeon_Tree

Dungeon_Tree **References - UC_04**	
RESPONSIBILITIES: **Responsible for containing recursive tree structure of other trees and relationships between trees** **Responsible for containing sprite representation of Nodes as well as connections between Nodes** **Responsible for listening for mouse click then altering sprite of node clicked to indicate it is selected** **Responsible for containing the current node position of the party** **Responsible for ensuring directly related distal nodes are the only nodes which are able to be selected on click** **Responsible for providing method to change current position of party to newly selected node**	COLLABORATORS: Renderer

BaseUnit **References – US_05_1, US_05_2, US_08_1, US_08_2, UC_03**	
RESPONSIBILITIES: **Responsible for containing and updating unit position** **Responsible for having method available for other functions/classes to change Unit position** **Responsible for containing and updating health, attack speed, and damage** **Responsible for having method available for other functions/classes to "attack"surrounding units**	COLLABORATORS: Tiles GridManager Renderer

Design use cases | 3.10.1

UC_01

UC_01_1: **Use Case**: Load game
UC_01_2: **Author**: DD
UC_01_3: **Date**: 15/11/21
UC_01_4: **Purpose**: Initiate new game from within main menu
UC_01_5: **Overview**: Starts when User presses load game button on main menu screen. On button press the player should be prompted to select a game save with one of the options being a new game. The selected game instance should begin, with scene either in dungeon map or in hub depending on program progress.
UC_01_6: **Cross References**: US_03_1, US_03_2
UC_01_7: **Actors**: User
UC_01_8: **Pre-conditions**:

UC_01_Pre-1: Program should be in main menu scene
UC_01_Pre-2: load game button must be "listening" for click event with prepared new game and load game functions actioned on-click

UC_01_9: **Post Conditions**:

UC_01_Post_1: Scene should be changed to a dungeon map with the respective progression selected

UC_01_10: **Main flow of events**:

User Starts the game and enters the main menu scene.
User correctly selects the load game button which prompts the user to select a save.
User selects a save and is transferred to the dungeon scene with map progression corresponding to the respective save selected.

UC_01_11: **Alternative flow of events**:

If user clicks outside of area of button actively listening for input, no action taken.
User selects the load game button and selects a save but is transferred to a new dungeon map scene with the incorrect progression loaded.
In the case of a failure of this process with user unable to begin new game instance, product not viable, if not rectified would necessitate removing main meu scene.

UC_01_12: **Testing**:

In testing this function, ensure that on mouse click, the new game button calls the new game function, and that the scene then changes appropriately to the hub/dungeon depending on the stage of development.

UC_02

UC_02_1: **Use Case**: Exit Game
UC_02_2: **Author**: DD
UC_02_3: **Date**: 15/11/21
UC_02_4: **Purpose**: End game process from within game window
UC_02_5: **Overview**: Starts when User presses quit game button on main menu screen. Button should use built in Unity application.quit() method to end game process.
UC_02_6: **Cross References**: US_01_1, US_01_2
UC_02_7: **Actors**: User
UC_02_8: **Pre-conditions**:
UC_02_Pre-1: Program should be in main menu scene
UC_02_Pre-2: Exit button must be "listening" for click event with prepared quit game function actioned on-click
UC_02_9: **Post Conditions**:
UC_02_Post_1: The program and all related processes must end.

UC_02_10: **Main flow of events**:

User enters the main menu scene and correctly selects the exit game button.

The programme successfully calls the application.quit() method and is terminated.

UC_02_11: **Alternative flow** of events:

If user clicks outside of area of button actively listening for input, no action taken.

In the case of a failure of this process the User can instead use system services to close the game, through the keyboard alt-F4 shortcut or through the windows task manager.

UC_02_ 12: **Testing**: When testing, ensure that on mouse click, the exit game button calls the exit game function.

UC_03

UC_03_1: **Use Case**: Save Game
UC_03_2: **Author**: CP
UC_03_3: **Date**: 15/11/21
UC_03_4: **Purpose**: Save game progress from within game window
UC_03_5: **Overview**: Starts when User presses save game button on main menu screen. This should save the player's progress through the map where the user has the option to save the game under a name.
UC_03_6: **Cross References**: US_02_1, US_02_2
UC_03_7: **Actors**: User
UC_03_8: **Pre-conditions**:
UC_03_Pre-1: Program should be in main menu scene
UC_03_Pre-2: Save button must be "listening" for click event with prepared save game function actioned on-click
UC_03_9: **Post Conditions**:
UC_03_Post_1: The program saves and stores current game progress under the player's name.
UC_03_10: **Main flow**:

When the player is in the main menu and decides they want to save their progress they select the save game button and the player is prompted to enter their name.

The player enters their name and selects okay.

The player sees their name listed under saved games.

UC_02_11: **Alternative flow** of events:

The player selects the save game button but is not prompted to enter their name.

After the player enters their name, they cannot see it listed under saved games.

UC_02_12: **Testing**:
When testing ensure the game is saved with the correct name entered and when the game is re-opened the user is at the same map position with the same number of troops as when they left the game.

UC_04

UC_04_1: **Use Case**: Main Menu
UC_04_2: **Author**: CP
UC_04_3: **Date**: 15/11/21
UC_04_4: **Purpose**: Interface for saving and exiting the game
UC_04_5: **Overview**: The user should be able to select a main menu option from the map scene from which they can choose to load a game, save or exit the game. User should also be met by the main menu upon entry to the game.
UC_04_6: **Cross References**: US_02_1, US_02_2
UC_04_7: **Actors**: User
UC_04_8: **Pre-conditions**:
UC_04_Pre-1: Program should be in map scene or have just been run
UC_04_9: **Post Conditions**:

UC_04_Post_1: The program displays the main menu with all its options.
UC_04_10: **Main flow**:

When the player is in the map scene, they select the main menu option, triggering the main menu screen.
The main menu is displayed with options to load game, save game or exit game
When the player enters the game they are met by the main menu scene displaying options to load game, save game or exit game

UC_04_11: **Alternative flow** of events:

The player selects the main menu button but the main menu scene is not triggered.
After the player enters the game they are not met by the main menu screen and instead transferred directly to the dungeon map scene.

UC_04_12: **Testing**:
When testing ensure the main menu screen is triggered upon selecting the main menu button. Ensure that upon entry to the game the player is met by the main menu screen.

UC_05
UC_05_1: **Use Case**: Popup displayed
UC_05_2: **Author**: CP
UC_05_3: **Date**: 15/11/21
UC_05_4: **Purpose**: Inform player of a game consequence.
UC_05_5: **Overview**: When the user takes an action and a consequence is reached, they should be informed of the resulting consequence.
UC_05_6: **Cross References**: US_07_01, US_07_02, US_15_01, US_15_02.
UC_05_7: **Actors**: User interface.
UC_05_8: **Pre-conditions**:

UC_05_Pre-1: Player has made an action in the game.
UC_05_Pre-2: The action taken has a consequence.

UC_05_9: **Post Conditions**:

UC_04_Post_1: The player understands the consequence of the action taken.

UC_05_10: **Main flow**:

The player makes an action in the game (e.g. enters a treasure node)
The popup is displayed clearly on the scene, with the background still visible
The player understands the consequence of the action taken (e.g. has been awarded 4 units).

UC_05_11: **Alternative flow** of events:

The player makes an action in the game (e.g. enters a treasure node)
The popup is not displayed clearly on the scene.
The popup is displayed clearly on the scene; however, the background is not visible.
The popup is displayed clearly on the scene but the player does not understand the consequence of the action taken.

UC_05_12: **Testing**:
When testing ensure that the popup is correctly aligned in the centre of the screen and the background scene is still visible to the user. Ensure that a situationally required message can be displayed dependent on the action triggering the popup.

UC_06
UC_06_1: **Use Case**: Enter treasure node.
UC_06_2: **Author**: CP
UC_06_3: **Date**: 15/11/21
UC_06_4: **Purpose**: Informs user of their treasure reward units
UC_06_5: **Overview**: When a user selects a treasure node they should be greeted by a popup informing them of how many troops they have gained as a reward.

UC_06_6: **Cross References**: US_07_1, US_07_2, US_09_1, US_09_2, US_10_1, US_10_2,
UC_06_7: **Actors**: User, User interface
UC_06_8: **Pre-conditions**:
UC_06_Pre-1: Programme should be on map scene
UC_06_Pre-2: Treasure node is adjacent to current node.
UC_06_9: **Post Conditions**:
UC_06_Post_1: New occupied node should now be the treasure node
UC_06_Post_2: A dismissible popup should now displayed on the screen informing the user of troop reward
UC_06_Post_3: Unit pool should be updated to reflect additional troops gained.
UC_06_10: **Main flow**:
User sees a treasure node adjacent to current node and selects it.
After the treasure node is selected, a dismissible popup is displayed informing the user of their reward.
After the popup is dismissed the unit pool is updated according to the number of troops awarded.
UC_06_11: **Alternative flow** of events:
User sees a treasure node adjacent to current node, but cannot select it.
After selecting the treasure node, the programme fails to display a popup message.
After selecting a treasure node, a popup is displayed showing the chest reward, but the user is unable to dismiss it.
After the popup is dismissed, the unit pool is not updated to show the extra units awarded.
UC_06_12: **Testing**:
When testing, ensure selection of the treasure node is recognised by the programme and this can be used to trigger a popup. Ensure the popup can successfully be dismissed and that the popup management system updates the unit manager to reflect the additional units gained so that this is displayed in the unit pool.

UC_07
UC_07_1: **Use Case**: Welcome node
UC_07_2: **Author**: CP
UC_07_3: **Date**: 15/11/21
UC_07_4: **Purpose**: Introduce the player to the game.
UC_07_5: **Overview**: Upon entering the game the player should be able to select the starting node where a welcome message is displayed, welcoming the player to the game and informing them of the point they must reach to complete it.
UC_07_6: **Cross References**: US_03_1, US_03_2
UC_07_7: **Actors**: User, User Interface
UC_07_8: **Pre-conditions**:
UC_07_Pre-1: The game has been started and the map scene initiated.
UC_07_Pre-2: No nodes have been clicked and activated yet.
UC_07_9: **Post Conditions**:
UC_07_Post_1: A dismissible popup should be displayed welcoming the player and informing them of the game endpoint (star)
UC_07_10: **Main flow**:
The player enters the game and clicks on the first node on the map.
A dismissible popup is displayed welcoming the player to the game and informing them they must reach the star node at the end of the map.
Popup is dismissed and welcome node is now darkened indicating it is the user's current position on the map.
UC_07_11: **Alternative flow** of events:

The player enters the game and clicks on the first node.
A dismissible popup fails to be displayed to welcome the player or inform them of the objective.
A dismissible popup is created but it is not visible on the screen and therefore cannot be dismissed.
Popup is dismissed but the welcome node is still selectable indicating the user hasn't taken position at the welcome node yet.

UC_07_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct welcome message. Ensure node is darkened and no longer selectable after dismissal.

UC_08

UC_08_1: **Use Case**: New arena node
UC_08_2: **Author**: CP
UC_08_3: **Date**: 15/11/21
UC_08_4: **Purpose**: Transfers the player to the first node of a new arena and informs them they've completed the previous arena.
UC_08_5: **Overview**: When a player have taken a path through an arena and made it to the end, they should be able to select a new arena node which informs them they've completed the previous arena. The player can then choose between different paths through the new arena.
UC_08_6: **Cross References**: US_06_1, US_06_2
UC_08_7: **Actors**: User
UC_08_8: **Pre-conditions**:
UC_08_Pre-1: Player has reached and completed one of the final nodes of their current arena.
UC_08_Pre-2:
UC_08_9: **Post Conditions**:
UC_08_Post_1: Player has the option to select from different starting nodes through the new arena with different paths and encounters.
UC_08_10: **Main flow**:
The player selects the new arena node.
A popup is displayed informing the player that they can now enter a new arena.
The player is transferred to the new arena node and can now select a path through the new arena.
UC_08_11: **Alternative flow** of events:
The player is unable to select the new arena node
The player selects the new arena node but a popup informing the player they can now enter a new arena fails to be displayed.
The player fails to be transferred to the new arena node and therefore cannot select a path through the new arena.
UC_08_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct enter new arena message displayed. Ensure node is darkened and no longer selectable after dismissal.

UC_09

UC_09_1: **Use Case**: Game completion node
UC_09_2: **Author**: CP
UC_09_3: **Date**: 15/11/21
UC_09_4: **Purpose**: Initiates and informs user of completion of the game.
UC_09_5: **Overview**: There should be a final node on the map, whose reaching is the objective of the game. Upon entry, the player should be informed they have completed the game and congratulated.

UC_09_6: **Cross References**: US_01_1, US_01_2
UC_09_7: **Actors**: User
UC_09_8: **Pre-conditions**:
UC_09_Pre-1: The player has successfully completed the last battle node of the game.
UC_09_Pre-2: The player has returned to the map scene.
UC_09_9: **Post Conditions**:
UC_09_Post_1: The game is complete and a message is shown informing the user they have won.
UC_09_10: **Main flow**:
After completing the final battle encounter the player has returned to the map scene.
The player selects the final node on the map.
A dismissible popup is initiated, with a message congratulating the user for winning the game.
UC_09_11: **Alternative flow** of events:
After completing the final battle encounter the player has returned to the map scene.
The player selects the final node on the map.
A dismissible popup fails to be initiated and the player is not informed they have won the game.
UC_09_12: **Testing:**
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct congratulatory message displayed. Ensure node is darkened and no longer selectable after dismissal.

UC_10
UC_10_1: **Use Case**: Update unit pool
UC_10_2: **Author**: CP
UC_10_3: **Date**: 15/11/21
UC_10_4: **Purpose**: Reflects additional troops awarded to the player
UC_10_5: **Overview**: When a player is awarded units from a treasure node, the unit pool should be updated to show the new total number of units.
UC_10_6: **Cross References**: US_06_1, US_06_2
UC_10_7: **Actors**: User
UC_10_8: **Pre-conditions**:
UC_10_Pre-1: Player has selected a treasure node.
UC_10_Pre-2: Unit reward has been determined.
UC_10_9: **Post Conditions**:
UC_10_Post_1: Unit pool should reflect the units awarded on top of the units in possession preceding node selection
UC_10_10: **Main flow**:
User selects a treasure node and is informed of unit reward.
Unit pool is updated to show the sum of the player's previous number of units and the number of units awarded.
UC_10_11: **Alternative flow** of events:
User selects a treasure node and is informed of unit reward.
Unit pool is not updated and remains the same as before node entry.
Unit pool is updated; however the new value is not the sum of the player's previous number of units and the number of units awarded.
UC_10_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct unit award message displayed. Ensure that the unit count and unit pool are updated to reflect the extra troops rewarded.

UC_11

UC_11_1: **Use Case**: Enter battle node
UC_11_2: **Author**: CP
UC_11_3: **Date**: 15/11/21
UC_11_4: **Purpose**: Allow user to enter a battle node to progress through the map.
UC_11_5: **Overview**: When the user is in the map scene and the adjacent node is a battle node, they should be able to select the node and be transferred to the respective level's grid scene.
UC_11_6: **Cross References**: US_04_1, US_04_2, US_08_1, US_08_2, US_09_1, US_09_2, US_10_1, US_10_2
UC_11_7: **Actors**: User
UC_11_8: **Pre-conditions**:
UC_11_Pre-1: Programme is on map scene.
UC_11_Pre-2: These is a battle node adjacent to the current node.
UC_11_9: **Post Conditions**:
UC_11_Post_1: The player is transferred to grid scene
UC_11_10: **Main flow**:
The player is currently at a node with a battle node adjacent.
The player selects the battle node.
Upon selecting the battle node, the player is transferred to the grid scene.
UC_11_11: **Alternative flow** of events:
The player selects the battle node; however, they are unable to select the battle node.
The player selects the battle node; however, they are not transferred to the grid scene.
UC_11_12: Testing:
When testing ensure that the grid scene is loaded successfully with the unit pool of the unit placement scene reflecting the number of units in the unit count of the map scene.

UC_12

UC_12_1: **Use Case**: Enemy unit health scaling
UC_12_2: **Author**: CP
UC_12_3: **Date**: 15/11/21
UC_12_4: **Purpose**: To add difficulty to battle encounters as the player progresses through the game
UC_12_5: **Overview**: As the player progresses through the game, the health of enemy units should increase at each battle encounter.
UC_12_6: **Cross References**: US_09_1, US_09_2, US_12_1, US_12_2
UC_12_7: **Actors**: User
UC_12_8: **Pre-conditions**:
UC_12_Pre-1: Player has completed at least one battle encounter.
UC_12_9: **Post Conditions**:
UC_12_Post_1: The enemy units encountered at the subsequent battle node have more health points than those of the previous node.
UC_12_10: **Main flow**:
The player completes a battle encounter.
The player enters a subsequent battle node and encounters enemy units with one more health point than those of the previous node.
UC_12_11: **Alternative flow** of events:
The player completes a battle encounter.
The player enters a subsequent battle node; however, the enemy units encountered have the same amount of or less health points than those of the previous node.
UC_12_12: **Testing**:
When testing ensure that the enemy units in each subsequent battle node have one extra health point.

UC_13

UC_13_1: **Use Case**: Highlight tile
UC_13_2: **Author**: CP
UC_13_3: **Date**: 15/11/21
UC_13_4: **Purpose**: Highlights the tile which the user's cursor is interacting with at its current position.
UC_13_5: **Overview**: During the grid scene, the player should be able to see which tile their cursor is interacting with according to its position, where the current tile is highlighted
UC_13_6: **Cross References**: US_05_1, US_05_2, US_13_1, US_13_2
UC_13_7: **Actors**: User
UC_13_8: **Pre-conditions**:
UC_13_Pre-1: The programme should be in the grid scene, and the board displayed
UC_13_9: **Post Conditions**:
UC_13_Post_1: When the cursor moves over a tile, the tile is highlighted.
UC_13_10: **Main flow**:
During the grid scene the player moves the cursor over a specific tile.
The tile the cursor is positioned over is highlighted.
UC_13_11: **Alternative flow** of events:
During the grid scene the player moves the cursor over a specific tile.
The tile the cursor is positioned over is highlighted along with one or more other tiles which the cursor is not positioned over.
The tile the cursor is positioned over is not highlighted; however one or more other tiles which the cursor is not positioned over are highlighted.
The tile the cursor is positioned over is not highlighted nor is any other tile on the board.
UC_13_12: **Testing**:
When testing ensure that only the tile the cursor is hovering over is highlighted and this is reflected as the cursor is moved around the grid.

UC_14
UC_14_1: **Use Case**: Select highlighted tile
UC_14_2: **Author**: CP
UC_14_3: **Date**: 15/11/21
UC_14_4: **Purpose**: To enable action to be performed upon selection of a tile.
UC_14_5: **Overview**: During the grid scene, the player should be able to select a highlighted tile which the cursor is positioned over and the programme should recognise this selection enabling an action to be performed.
UC_14_6: **Cross References**: US_08_01, US_08_02, US_05_1, US_05_2, US_13_1, US_13_2
UC_14_7: **Actors**: User
UC_14_8: **Pre-conditions**:
UC_14_Pre-1: The programme should be in the grid scene
UC_14_Pre-2: The user should have the cursor positioned over the highlighted tile which they wish to select.
UC_14_9: **Post Conditions**:
UC_14_Post_1: The tile is selected, turning grey and the action is recognised by the programme.
UC_14_10: **Main flow**:
During the grid scene the player moves the cursor over a specific tile, which is highlighted.
The cursor changes to a hand symbol letting the player know its clickable,
The player selects the tile, which turns grey, meaning the programme recognises the action.
UC_14_11: **Alternative flow** of events:
During the grid scene the player moves the cursor over a specific tile, which is highlighted
The cursor remains unchanged, so they do not know its clickable.

The cursor changes to a hand symbol, letting the player know its clickable, the player selects the tile; However, the tile does not change colour, meaning the programme does not recognise the action.

UC_14_12: **Testing**:
When testing ensure that it is only the highlighted tile which is selected and the programme recognises the action.

UC_15

UC_15_1: **Use Case**: Place units on grid
UC_15_2: **Author**: DD
UC_15_3: **Date**: 15/11/21
UC_15_4: **Purpose**: Allow user to place units on grid before battle
UC_15_5: **Overview**: Starts during placement phase of auto-battler encounter. Player should be able to select from persistent pool of units, visible on the lower edge of the screen in a relevant UI object, and these can then be placed in empty hexes on the board on the lower half of the game board.
UC_15_6: **Cross References**: US_08_01, US_08_02, US_11_01, US_11_02, US_14_01, US_14_01, US_15_01, US_15_02
UC_15_7: **Actors**: User
UC_15_8: **Pre-conditions**:
UC_15_Pre-1: Program should be within encounter scene
UC_15_Pre-2: Current state of encounter should be planning
UC_15_Pre-3: User must have units available to place
UC_15_Pre-4: At least one square on board should not be full
UC_15_Pre-5: Current units placed must be less than maximum placeable units
UC_15_9: **Post Conditions**:
UC_15_Post_1: Selected unit should be placed on the game board
UC_15_Post_2: The Square on the board the unit now occupies should no longer be empty
UC_15_Post_3: One copy of the selected unit should be removed from the unit pool temporarily
UC_15_10: **Main flow of events**:
Player Enters the battle node initiating the placement phase of the battle encounter.
Player select a unit from their unit pool.
Unit is placed on one of the empty hexes on the board.
UC_15_11: **Alternative flow** of events:
User tries to place unit on non-empty square, explanatory error message is shown
User tries to place unit from empty pool of units, explanatory error message is shown
UC_15_12: Testing:
When testing, ensure unit clicked becomes selected. Also ensure when subsequently clicking on tile that a new unit of the selected type is instantiated, with a position equal to the selected tile.

UC_16

UC_16_1: **Use Case**: Unit constant damage output
UC_16_2: **Author**: CP
UC_16_3: **Date**: 15/11/21
UC_16_4: **Purpose**: Allows units to deal damage to opposing units.
UC_16_5: **Overview**: Units consistently output damage with amount and rate determined by the unit type.
UC_16_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2
UC_16_7: **Actors**: User
UC_16_8: **Pre-conditions**:
UC_16_Pre-1: Units are placed on the board

UC_16_Pre-2: Fight sequence is initiated

UC_16_9: **Post Conditions**:

UC_16_Post_1: Damage is consistently output while the unit is not dead.

UC_16_10: **Main flow**:

The fight sequence is initiated and all units on the board begin outputting damage.

Damage is consistently output until the unit is dead.

UC_16_11: **Alternative flow** of events:

The fight sequence is initiated; however, some or all the units do not begin outputting damage.

Damage is consistently output by each unit, even after the unit is dead.

UC_16_12: **Testing**:

When testing ensure the unit damage output is accessible to other units so that they can use it to take damage.

UC_17

UC_17_1: **Use Case**: Unit take damage

UC_17_2: **Author**: CP

UC_17_3: **Date**: 15/11/21

UC_17_4: **Purpose**: Allows units to take damage from other units

UC_17_5: **Overview**: When a unit is in range of an opposing unit, it should be dealt damage corresponding to the opposing unit's damage output.

UC_17_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2

UC_17_7: **Actors**: User

UC_17_8: **Pre-conditions**:

UC_17_Pre-1: The unit has been engaged by an opposing unit.

UC_17_Pre-2: The unit is in range of the opposing unit.

UC_17_9: **Post Conditions**:

UC_17_Post_1: The unit health is consistently reduced according to the opposing unit's damage output.

UC_17_10: **Main flow**:

The unit comes into range of an opposing unit which has engaged it.

The opposing unit begins consistently dealing damage to the unit while it is in range.

The health variable of the unit is consistently reduced according to the opposing unit's damage output.

UC_17_11: **Alternative flow** of events:

The unit comes into range of the opposing unit which has engaged it.

The opposing unit fails to deal damage to the unit

The opposing unit

The health attribute of a unit engaged in a 1v1 battle subceeds zero.

The dead Boolean of the unit object fails to be set to true.

UC_17_12: **Testing**:

When testing ensure the damage output of the unit dealing the damage is accessible to the one taking it. Ensure a corresponding amount of health points are deducted from the unit's health variable.

UC_18

UC_18_1: **Use Case**: Unit death

UC_18_2: **Author**: CP

UC_18_3: **Date**: 15/11/21

UC_18_4: **Purpose**: Criteria for the elimination of a unit from a battle.

UC_18_5: **Overview**: When the health of a unit has reached or subceeded zero, the unit is given the dead attribute, which can then be used to disengage the opposing unit and identify unit objects which need to be destroyed.
UC_18_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2
UC_18_7: **Actors**: User
UC_18_8: **Pre-conditions**:
UC_18_Pre-1: The unit in question has been engaged in a 1v1 battle.
UC_18_Pre-2: The health of the unit is less than or equal to zero.
UC_18_9: **Post Conditions**:
UC_17_Post_1: The unit is given the dead attribute which is a Boolean set to true.
UC_18_10: **Main flow**:
The health attribute of a unit engaged in a 1v1 battle subceeds zero.
The "dead" Boolean of the unit object is set to true.
The "dead" unit object is destroyed, eliminating it from the battle
UC_18_11: **Alternative flow** of events:
The health attribute of a unit engaged in a 1v1 battle subceeds zero.
The dead Boolean of the unit object fails to be set to true.
The dead Boolean of the object is set to true but it is not destroyed, failing to eliminate it from the game.
UC_18_12: **Testing**:
When testing ensure that the unit object is successfully destroyed upon its dead Boolean being set to true

UC_19
UC_19_1: **Use Case**: Play game in browser
UC_19_2: **Author**: CP
UC_19_3: **Date**: 15/11/21
UC_19_4: **Purpose**: Allow players to play game in a browser across multiple devices.
UC_19_5: **Overview**: The game should be playable in a browser.
UC_19_6: **Cross References**: US_16_1, US_16_2
UC_19_7: **Actors**: User
UC_19_8: **Pre-conditions**:
UC_19_Pre-1: The game is fully functional.
UC_19_9: **Post Conditions**:
UC_19_Post_1: The game is playable on a web browser.
UC_19_10: **Main flow**:
The game is uploaded to unity free hosting service and embedded in a web page.
The game is playable online
UC_19_11: **Alternative flow** of events:
The game fails to be uploaded to unity free hosting service
UC_19_12: **Testing**:
When testing ensure that the game is playable across a range of devices on a range of web browsers.

Sprint 4: 19/11/21-26/11/21

The focus of sprint 4 was creating the simplest form of our game with a start, middle, and end.

Overview

What's supposed to happen this sprint

No.	Expected Start	Expected Finish	Objective
1	19/11/21	26/11/21	Create a simple node map
2	19/11/21	26/11/21	Continue developing the grid
3	19/11/21	26/11/21	Add units to the grid
4	19/11/21	26/11/21	Create a pop-up UI
5	19/11/21	26/11/21	Merge branches on GitHub
6	19/11/21	26/11/21	Daily scrum, sprint review, sprint retrospective, sprint planning held

Reviews

What actually happened

No.	Started	Finished	Task	Doc
1	22/11/21	22/11/21	Daily scrum	4.1.1
2	24/11/21	30/10/21	Daily scrum	4.2.1
3	24/11/21	24/11/21	Customer meeting with answers analysed	4.3.1
4	26/11/21	26/11/21	Daily scrum, sprint review, sprint retrospective, sprint planning	4.4.1
5	Throughout		Backlog updated	4.5.1
6	Throughout		Exception handling	4.6.1
7	Throughout		User stories with acceptance tests	4.7.3
8	Throughout		Requirements use cases	4.8.2
9	Throughout		CRC cards	4.9.2
10	Throughout		Design use cases	4.10.2
11	Throughout		User interface design	4.11.1

Daily scrum | 4.1.1

22/11/21, 10:00-10:15, Microsoft Teams

Attendance

All present
Meeting led by LG
Minutes taken by LG

Minutes

1. The team at the last daily scrum has been split into 3 development teams

    a. IL and TW working on the node map,
    b. LW and QZ on the grid and adding units
    c. DD and CP on the pilot pop-up UI

2. There are no current updates on the progression of the node map from the last daily scrum meeting.

3. LW has finished the grid but is struggling to integrate the units branch code into theirs

    a. LG notes that they have not updated the unit branch repository with their latest code and will do that as soon as possible
    b. LW and QZ have organised a pair programming session for tomorrow (23/11/21)

4. DD and CP have started the pilot pop-up UI and it is mostly completed for the Customer meeting on Wednesday (24/11/21)

    a. CP notes that they may struggle to decide where to put the trigger

5. LW notes that we have not done a merge of all the branches to the master repository, but it is agreed by all team members that this will be delayed until at least Wednesday (24/11/21)

6. NO is to meet which each development pair before the Custome meeting to discuss progress and any present issues

7. Each development pair from Sprint 3 is to write up a diary of what they did for the documentation of Sprint 3

8. All agreed to stick to the current version of Unity and to ignore the new update

Daily scrum | 4.2.1

24/11/21, 14:15-14:30, CB 5.13

Attendance

All present
Meeting led by LG
Minutes taken by LG

Minutes

1. IL and TW have completed their simple node map.

    a. The background is taken from Reddit and can be replaced
    b. When the nodes are clicked on, they change colour to grey
    c. When the treasure chest node is clicked on, it takes the user to a new scene with a back arrow, which then takes the user back to the node map

2. DD and CP have been working on the pilot UI pop-up on the GitHub map

    a. A system of pop-ups link to nodes with a controller class
    b. There was a miscommunication with which map was to be used to build the pop-up on
    c. NO resolves that on Friday (26/11/21), the maps will be merged into one
    d. The team is unsure on whether the GitHub map code is free to use and whether to start the node map from scratch or not, this is not resolved

3. LW and QZ have finished putting units on the grid. The units spawn on pre-determined tiles in 3 different rotations

    a. Stretch goal is a drag-and-drop feature

4. DD and CP note that their pair programming session worked well and that they will continue their pair programming practise to generalise their code for the pop-up

5. TW and IL agree to meet for a pair programming session before Friday (26/11/21) to finish the simple start-end node map

6. Team agree to keep working on the current version of the map they have previously, and all versions will be merged on Friday (26/11/21)

Customer meeting | 4.3.1

17/11/21, 15:30-15:45, CB 5.13

Attendance

All present
Recording taken by LG
Minutes taken by LG

Minutes

1. TW shows the basic node map and demonstrates its functionality, JP is neutral

2. LW presents the grid, including the 3 random formations the units can be placed in

    b. JP cannot completely tell which side is the players/enemies, but LW states this will be obvious when the sprites are not all identical rocks
    c. The 3 random formations on the units are “cool”

3. It is decided that the GitHub node map can be used for the game, with the final look being a simpler map with the same functionality

4. JP fully understands how to play the game from start to finish

5. The goal of Sprint 5 is decided

    d. All parts of the game are to be fully integrated
    e. The units can be moved – possibly a drag and drop feature
    f. A stretch goal is to have the units interacting/attacking

6. The priority for the next sprint is to have one level of the game working, in which we can build upon if times allows

7. JP suggests the team does game-testing with non-team members

    g. JP will produce an umbrella ethics form the team can use
    h. The team is to research a validated survey form which we can adapt and run it by him for approval for next week

8. The documentation for Sprint 3 needs to be completed and JP emphasises not to leave the product documentation to the last few weeks

Analysis

1. Time spent in the lab before the customer meeting was spent updating the team on each of the development teams progress as well as training for the Scrum Owner and Product Owner handovers.

2. Given the good reaction to the using the GitHub node map for help with the functionality, it is decided the team will use a hybrid of the simple and GitHub map but with our own sprites/images

3. Key takeaways were that the Customer wants a fully working integrated game for next Wednesday (01/12/21)

4. JP’s suggestion that next week's goal of a fully integrated game will take a lot of work, but the team is happy to put in the effort next week to reach this goal

5. Team agreed to merge each of the branches of the GitHub repository on Friday (26/11/21)

6. Team agreed that for additional levels of the game could include a treasure chest which would give the player additional units as well as implementing boss battles and rest areas

End of sprint meeting | 4.4.1

26/11/21, 13:15-14:30, NH 2.17a

Attendance

All present although CP joined virtually
Meeting led by LG
Minutes taken by LG

Daily scrum

1. All development teams demonstrated their current progress of the game on the TV and re-capped what part of the game they were working on

    a. IL and TW were working on the node map
    b. LW and QZ on the grid and adding units to the grid
    c. DD and CP on the pilot pop-up UI

2. TW and IL have completed their sprint goal and made good progress on the new simple node map

    a. When a curser hovers over a node, it switches to a hand symbol to let the player know it is clickable
    b. When the node is clicked, it turns grey to signal it has been clicked on
    c. When the treasure node is clicked, it takes the player to a new scene
    d. IL notes that it is hard to keep colours of the nodes saved
    e. The quality of the code can be improved, but the node map is fully functioning

3. Merging of the UI branch from the old GitHub node map to the new simple node map has been completed

4. DD and CP have added a UI manager to manage the UI pop-up

    a. They have also added a tracker to count the number of units on the board
    b. When the treasure chest node is clicked on, there is now a pop-up with a dismiss button
    c. The tracker is not linked to increase when the pop-up gifts the player a unit

5. NO, the product owner checks that the node map is in a class. It is confirmed it is

6. All team are to make sure that they are commenting all code for readability

Sprint review

1. The consensus was that the whole team completed their assigned sprint goals

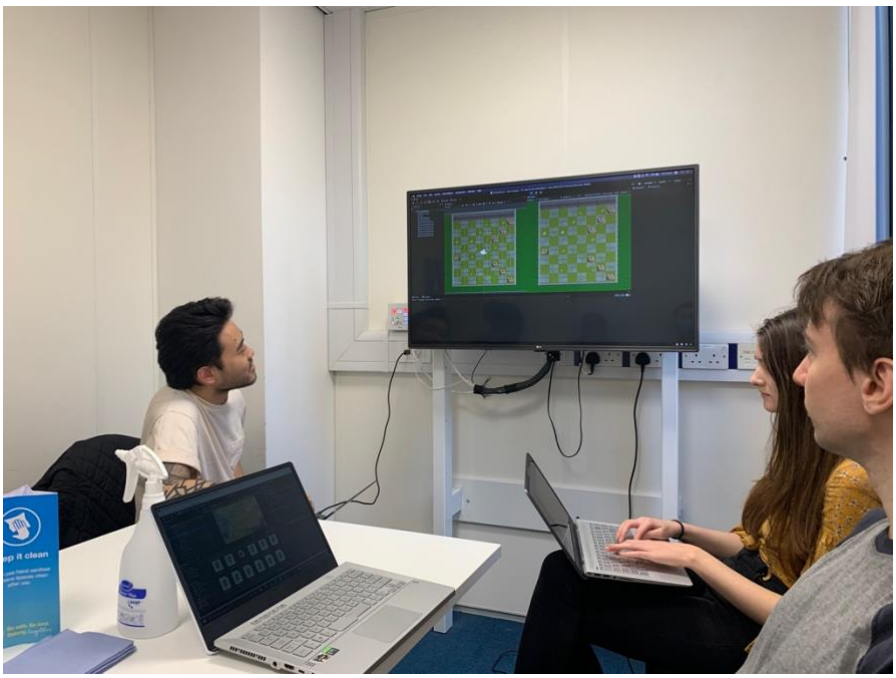
    a. TW and IL have completed implementing a basic node map which included clicking on a treasure map node and taking the player to a new scene
    b. DD and CP have completed creating a pop-up UI
    c. LW created 6 different formations that the units can battle in

2. The following code changes and additions have been made:

    a. A UI manager has been added to the code base to implement and manage the UI pop-up
    b. All of team are to make sure that variable names are in camel case
    c. The old branches in the archive folder have been cleaned up in the MS teams file section

3. Units moving step wise and with the same speed is agreed to be the simplest requirement for completion of the game

4. There are no changes to user stories or use cases, NO is to look through these for this sprint and will add any if needed



Booking a room facilitated a more successful end of sprint meeting compared to the sprint prior

Sprint retrospective

1. Everyone on the team agrees that the assigned tasks each sprint are of reasonable difficulty and of the right amount of quantity

2. All of team are to make sure that they are checking the sprint backlog regularly and checking off any tasks they have completed

3. The developer diaries, added in Sprint 3, are agreed to be very helpful for the Scrum Master and Product Owner when writing up the documentation and are to be continued for the upcoming sprints

4. The last sprint is agreed by all the team to be a success, all development teams were able to complete their assigned tasks and progress of the game is progressing well

5. Handover of Scrum Master and Product Owner is tricky to do over the weekend, to improve this the handover will be changed to the Monday instead of the Friday when the next Sprint starts

Sprint planning

1. Projected capacity of the team of Sprint 5 is less than Sprint 4 due to numerous coursework due during the week

2. The next increment goal is decided as following:

    a. Create a drag-and-drop feature and a battler UI for different units
    b. Further develop the map
    c. Create a scene change from map to grid

Task assignments

	A	U	V	W	X	Y	Z	AA
1	**Date**	**27/11/21**	**28/11/21**	**29/11/21**	**30/11/21**	**01/12/21**	**02/12/21**	**03/12/21**
2	**Sprint**				**Sprint 5**			
3	**Qinyuan Zhuang**	UI						
4	**Lawrence Wong**	Product owner						
5	**Callum Pitceathly**	Battle						
6	**Drew Dundee**	Battle						
7	**Naomi Okiddy**	Scrum master						
8	**Lucy Goodchild**	UI						
9	**Ieuan Lambert**	Map						
10	**Tim Win**	Map						
11	**Everyone**							
12	**Daily scrum**			10:00 MS Teams		14:15 MS Teams		
13	**Daily scrum, sprint review, sprint retrospective, and sprint planning**							13:15 NH 2.17a
14	**Customer meeting**					Cancelled		
15								

Backlog | 4.5.1

Complete backlog tasks

1. Placing units on the grid
2. Create a simple node map
3. Pop-up UI
4. Attach map to treasure chest node
5. Treasure chest node content
6. Merging branches

New backlog tasks

1. Place units and hover highlight on tile
2. Drag and drop feature
3. Battler and Asset UI
4. Scene Changing: Map to Grid

Current backlog

1. Basis Unit Movement
2. Main menu
3. Map randomiser
4. Change the tilesets based on difficulty
5. Main menu music
6. Recording feature
7. Sound effects

Exception handling | 4.6.1

Multiple maps

1. During this sprint we had 2 development teams working on 2 different node maps, due to miscommunication

2. The current status of this situation is 2 new features: a new simple node map and a pop-up UI built upon a more complicated node map sourced from GitHub

3. The actions taken were team agreed in daily scrum on (26/11/21) that there would be more communication during the sprint on what each development team are working on and had completed

4. The simple node map and pop-up UI can be merged together

User stories with acceptance tests | 4.7.3

US_01_1: Description: "As a user, I'd like a way to exit the program when I'm finished, from within the game window"
US_01_2: Acceptance Criteria: Accepted when quit game button implemented in main menu

US_02_1: Description: "As a user, I'd like to be able to continue a persistent game across multiple sessions"
US_02_2: Acceptance Criteria: Accepted when save game system implemented

US_03_1: Description: "As a user, I need a way to initialise a new game from within the game window"
US_03_2: Acceptance Criteria: Accepted when new game button implemented in main menu

US_04_1: Description: "As a user, I'd like multiple varied dungeons to add replayability"
US_04_2: Acceptance Criteria: Accepted when more than one dungeon implemented

US_05_1: Description: "As a user, I'd like to be able to make decisions about the units within my team, to add challenge and reward planning and critical thinking"
US_05_2: Acceptance Criteria: Accepted when able to select from multiple unit types during planning phase of combat

US_06_1: Description: "As a user, I'd like to have additional choice and variety in my gameplay, to separate this game from other auto-battlers"
US_06_2: Acceptance Criteria: Accepted when implement both persistent unit pools on main map, as well as being able to choose from branching paths

US_07_1: Description: "As a user, I'd like to make strategic decisions, with sufficient information to differentiate good and bad decisions"
US_07_2: Acceptance Criteria: Accepted when able to clearly see indication of reward and difficulty on each dungeon path.

US_08_1: Description: "As a user, I want frequent combat encounters as part of the core game"
US_08_2: Acceptance Criteria: Accepted when nodes in dungeon graph implemented, with encounters possible at each, including fully implemented auto-battling system

US_09_1: Description: "As a user, I enjoy progressive challenge to test my ability"
US_09_2: Acceptance Criteria: Accepted when persistent unit pool implemented, as well as progressively harder encounters culminating in "boss" encounter

US_10_1: Description: "As a user, I enjoy uncertainty, surprise, and variety in my gameplay"
US_10_2: Acceptance Criteria: Accepted when multiple encounter types implemented, with hidden possibility of each at given node

US_11_1: Description: "As a user, I'd like my strategic decisions to have visible, long-term consequences."
US_11_2: Acceptance Criteria: Accepted when persistent unit pool implemented, alongside choice of multiple branching paths

US_12_1: Description: "As a user, I enjoy dynamic events and decision making, with lots of variety"
US_12_2: Acceptance Criteria: Accepted when multiple encounter types implemented, with hidden possibility of each and given node

US_13_1: Description: “As a user, I enjoy decision making with restrictions and uncertainty to add challenge”
US_13_2: Acceptance Criteria: Accepted when planning phase implemented to combat with time restriction, alongside choice of branching paths

US_14_1: Description: “As a user, I’d like clear success and failure states to enable planning and goal setting”
US_14_2: Acceptance Criteria: Accepted when persistent pool of units implemented, with game-over occurring when depleted

US_15_1: Description: “As a user, I would like to understand the game’s UI/UX clearly, so they don’t interfere with gameplay”
US_15_2: Acceptance Criteria: Accepted when no elements of gameplay obscured behind UI but also all relevant information provided on UI

US_16_1: Description: “As a client, I would like the game to be accessible to as many people as possible so I can share it with friends/peers or monetise it in future”
US_16_2: Acceptance Criteria: Accepted when game is playable in browser across multiple devices

US_17_1: Description: “As a user, I would like to clearly see each action being taken sequentially rather than simultaneously, to ensure I can learn and adapt.”
US_17_2: Acceptance Criteria: Accepted when actions within battle are made and displayed sequentially with delay between.

US_18_1: Description: “As a user, I would like to be able to review the sequence of events after a “battle” in ensure I can learn and adapt.”
US_18_2: Acceptance Criteria: Accepted when user is able to either re-watch battle, or view some representation of the events that took place afterwards

US_19_1: Description: “As a user, I would like to make decisions in which route I should take on the game map rather than follow the one set route.”
US_19_2: Acceptance Criteria: Accepted when game map forks to different battle nodes of varying difficulty

US_20_1: Description: “As a user, I would like a way to dismiss pop-ups easily and clearly.
US_20_2: Acceptance Criteria: Accepted when user is able to click on a dismiss button and the pop-up closes

Requirements use cases | 4.8.2

UC_01:
Use Case: Load Game
Level: User Goal
Context: Users can load a game progression by selecting the “load game” option from the main menu
Users are unable to select a “load game” option from the main menu
Frequency of occurrence: Any time the player enters the programme or wishes to load a game from the map scene
Open issues:

How to ensure a failsafe option to save game prompt in case player wishes to return to original game progression?

UC_02:
Use Case: Exit game
Level: User Goal
Context: Users can exit the game by selecting the “exit game” option from the main menu
Users are unable to select “exit game” option from the main menu
Frequency of occurrence: Any time the player wishes to start exit the game.
Open issues:

How to ensure a failsafe option to save game prompt in case player forgets to save their progress before quitting the programme.

UC_03:
Use Case: Save Game
Level: User Goal
Context: Users can save a game by selecting the “save game” option from the main menu
Users are unable to select the “save game” option from the main menu
Frequency of occurrence: Any time the player wishes to save a game.
Open issues:

How do we store saves?
How many saves can be stored by the game?
How many characters can be used in naming the save?
Can we implement a failsafe in case two saves are given the same name?

UC_04:
Use Case: Main menu
Level: Subfunction
Context: Users can decide to save load or exit the game by entering the main menu
Users are unable to save load or exit the game as they cannot access the main menu
Frequency of occurrence: Any time the player enters the game or wishes to save, load or exit the game.
Open issues:

Do we display this as a popup or a scene?

UC_05: (COMPLETE)
Use Case: Popup displayed
Level: Subfunction
Context: Users are informed of a consequence when they make a game action.
Users fail to be informed of the consequence of a game action taken
Frequency of occurrence: Any time the player makes a game action.
Open issues:

How do we ensure consistent alignment of popups any time they are instantiated? (RESOLVED)

How do we set the informational text of each popup to be dependent on the action taken? (RESOLVED)

How do we ensure the background is still visible when a popup is triggered? (RESOLVED)

UC_06: (COMPLETE)
Use Case: Enter treasure node
Level: Subfunction
Context: Upon entering a treasure node, users are rewarded with extra units.

Upon entering a treasure node, users fail to be rewarded with extra units.
Frequency of occurrence: Any time the player enters a treasure node.
Open issues:

How is the number of units rewarded determined? (RESOLVED)

How do we link this to the update unit pool functionality? (RESOLVED)

UC_07:
Use Case: Welcome node
Level: Subfunction
Context: Users can select the welcome node to initiate progression through the map, and are informed of their objective.

Users can select the welcome node to initiate progression through the map but aren't informed of their objective
Frequency of occurrence: Any time the player enters the map scene of a new game.
Open issues:

UC_08:
Use Case: New Arena node
Level: Subfunction
Context: Users can select a subsequent new arena node and are informed they are progressing to a new arena.

Users can select a subsequent new arena node but aren't informed they're progressing to a new arena.
Frequency of occurrence: Any time the player can progress to a new arena.
Open issues:

UC_09:
Use Case: Game completion node
Level: Subfunction
Context: Users can select the game completion node to complete the game and are informed they have won.

Users can select the game completion node to complete the game but aren't informed they have won.
Frequency of occurrence: When the player has completed the final battle encounter of the game.
Open issues:

UC_10: (COMPLETE)
Use Case: Update unit pool
Level: User goal
Context: The number of units of the unit count and unit pool can be updated.

The number of units of the unit count and unit pool cannot be updated.
Frequency of occurrence: Every time the player is awarded units.
Open issues:

How do we ensure there is no conflict between the unit count number and the unit pool contents? (RESOLVED)

UC_11:
Use Case: Enter battle node
Level: Subfunction
Context: Users can select a battle node which initiates the grid scene.
Users can select a battle node but the grid scene fails to be initialised.
Frequency of occurrence: When the player has a battle node adjacent to their current node.
Open issues:
How do we trigger grid scene initialisation. (RESOLVED)
How do we ensure the grid scene initialised is the correct encounter for the current node (ie enemy difficulty etc) ?

UC_12:
Use Case: Enemy unit health scaling
Level: User goal
Context: Users can select the game completion node to complete the game and are informed they have won.
Users can select the game completion node to complete the game but aren't informed they have won.
Frequency of occurrence: When the player has completed the final battle encounter of the game.
Open issues:
How is scaling factor per node determined?

UC_13:
Use Case: Highlight tile (COMPLETE)
Level: Sub function
Context: When a user moves their cursor over a tile in the grid scene the tile is highlighted
When a user moves their cursor over a tile in the grid scene the tile remains unchanged.
Frequency of occurrence: Any time the user is in the grid scene and moves their cursor over a tile.
Open issues:
How do we identify which tile the cursor is over? (RESOLVED)
What colour should the tile be highlighted? (RESOLVED)

UC_14:
Use Case: Select highlighted tile (COMPLETE)
Level: User goal
Context: User can select a highlighted tile and the programme recognises the action
Users cannot select the highlighted tile
Frequency of occurrence: Whenever the cursor is above a tile and it is highlighted.
Open issues:
How do we ensure selection of the tile is linked to a distinct action? (RESOLVED)

UC_15:
Use Case: Place units on the grid
Level: User goal
Context: User can select a unit and place it on a grid tile
User can select a unit but cannot place it on a grid tile
Frequency of occurrence: Whenever the user is in the unit placement sequence.
Open issues:
How do we select a unit?
How do we ensure a unit is placed when the tile is selected?
How do we ensure the placed unit is deducted from the unit pool?

UC_16:
Use Case: Unit constant damage output
Level: Sub function

Context: During battle sequence, units will output damage consistently which can then be accessed by opposing units.

During battle sequence, units will output damage consistently but output can't be accessed by opposing units.

Frequency of occurrence: Every time a fight sequence is initiated, for every unit.

Open issues:

How do we determine damage output and frequency?

When should damage output be initiated?

How is each unit's damage output accessed by opposing units?

UC_17:

Use Case: Unit take damage

Level: Sub function

Context: During battle sequence, units will take damage if they are within range of an opposing unit

During battle sequence, units are unable to take damage when in range of an opposing unit.

Frequency of occurrence: Whenever a unit is within range of an enemy unit.

Open issues:

How do we determine when a unit is within range of another?

How does a unit access an opposing unit's damage output?

How is health deducted from the unit's health points variable?

UC_18:

Use Case: Unit death

Level: Sub function

Context: During battle sequence, when a unit's health subceeds zero it will be eliminated from the battle

During battle sequence, when a unit's health subceeds zero it fails to be eliminated from the battle

Frequency of occurrence: Whenever a unit's health subceeds zero.

Open issues:

How to we eliminate an object from the battle?

How do we give a unit a "dead" attribute?

UC_19:

Use Case: Play game in browser

Level: Summary

Context: Users can access the game through a web browser

Users cannot access the game through a web browser

Frequency of occurrence: Dependent on User

Open issues:

What hosting service do we upload the game to?

How do we make the game as accessible as possible?

What devices will be able to run the game online?

UC_20: (COMPLETE)

Use Case: Move on map

Level: User goal

Context: Users can select an adjacent node and progress through the map

Users are unable to select adjacent nodes.

Frequency of occurrence: Any time the user is in the map scene.

Open issues:

How do we show the current location of the user on the map? (RESOLVED)

UC_21: (COMPLETE)

Use Case: Return to map from grid

Level: User goal
Context: The user can select a button to return to the map scene from the grid scene.
The user is unable to return to the map scene from the grid scene.
Frequency of occurrence: Whenever the user wishes to return to the map from the grid scene.
Open issues:
How to we ensure the player is returned to the node they exited? (RESOLVED)

CRC cards | 4.9.2

GridManager **References – UC_03**	
RESPONSIBILITIES: **Responsible for creating and containing a data structure containing all Tiles of grid and their positions** **Responsible for making this data available publicly to other functions**	COLLABORATORS: Tiles BaseUnit Renderer

Tiles **References - UC_03**	
RESPONSIBILITIES: **Responsible for containing sprite to be rendered on game window to represent a tile** **Responsible for listening for mouse click and then altering sprite to indicate selection of tile** **Responsible for listening for mouseover and then altering sprite to indicate the tile is highlighted**	COLLABORATORS: Tiles BaseUnit Renderer

Renderer **References – UC_01, UC_02, UC_03, UC_04**	
RESPONSIBILITIES: **Responsible for drawing all active sprites onto the game window**	COLLABORATORS: Tiles BaseUnit Dungeon_Tree

Dungeon_Tree **References - UC_04**	
RESPONSIBILITIES: **Responsible for containing recursive tree structure of other trees and relationships between trees** **Responsible for containing sprite representation of Nodes as well as connections between Nodes** **Responsible for listening for mouse click then altering sprite of node clicked to indicate it is selected** **Responsible for containing the current node position of the party** **Responsible for ensuring directly related distal nodes are the only nodes which are able to be selected on click** **Responsible for providing method to change current position of party to newly selected node**	COLLABORATORS: Renderer

BaseUnit **References – US_05_1, US_05_2, US_08_1, US_08_2, UC_03**	
RESPONSIBILITIES: **Responsible for containing and updating unit position** **Responsible for having method available for other functions/classes to change Unit position** **Responsible for containing and updating health, attack speed, and damage** **Responsible for having method available for other functions/classes to "attack"surrounding units**	COLLABORATORS: Tiles GridManager Renderer

UIManager	
RESPONSIBILITIES: Responsible for providing pop ups on certain dungeon levels Responsible for providing item counter Responsible for providing asset counter	COLLABORATORS: MapManager

MapManager	
RESPONSIBILITIES: Responsible to generating the map Responsible for saving the current state of the map Responsible for saving the player's progress	COLLABORATORS: UIManager

Design use cases | 4.10.2

UC_01

UC_01_1: **Use Case**: Load game
UC_01_2: **Author**: DD
UC_01_3: **Date**: 15/11/21
UC_01_4: **Purpose**: Initiate new game from within main menu
UC_01_5: **Overview**: Starts when User presses load game button on main menu screen. On button press the player should be prompted to select a game save with one of the options being a new game. The selected game instance should begin, with scene either in dungeon map or in hub depending on program progress.
UC_01_6: **Cross References**: US_03_1, US_03_2
UC_01_7: **Actors**: User
UC_01_8: **Pre-conditions**:
UC_01_Pre-1: Program should be in main menu scene
UC_01_Pre-2: load game button must be "listening" for click event with prepared new game and load game functions actioned on-click
UC_01_9: **Post Conditions**:
UC_01_Post_1: Scene should be changed to a dungeon map with the respective progression selected
UC_01_10: **Main flow of events**:
User Starts the game and enters the main menu scene.
User correctly selects the load game button which prompts the user to select a save.
User selects a save and is transferred to the dungeon scene with map progression corresponding to the respective save selected.
UC_01_11: **Alternative flow of events**:
If user clicks outside of area of button actively listening for input, no action taken.
User selects the load game button and selects a save but is transferred to a new dungeon map scene with the incorrect progression loaded.
In the case of a failure of this process with user unable to begin new game instance, product not viable, if not rectified would necessitate removing main meu scene.
UC_01_12: **Testing**:
In testing this function, ensure that on mouse click, the new game button calls the new game function, and that the scene then changes appropriately to the hub/dungeon depending on the stage of development.

UC_02

UC_02_1: **Use Case**: Exit Game
UC_02_2: **Author**: DD
UC_02_3: **Date**: 15/11/21
UC_02_4: **Purpose**: End game process from within game window
UC_02_5: **Overview**: Starts when User presses quit game button on main menu screen. Button should use built in Unity application.quit() method to end game process.
UC_02_6: **Cross References**: US_01_1, US_01_2
UC_02_7: **Actors**: User
UC_02_8: **Pre-conditions**:
UC_02_Pre-1: Program should be in main menu scene
UC_02_Pre-2: Exit button must be "listening" for click event with prepared quit game function actioned on-click
UC_02_9: **Post Conditions**:
UC_02_Post_1: The program and all related processes must end.

UC_02_10: **Main flow of events**:

User enters the main menu scene and correctly selects the exit game button.

The programme successfully calls the application.quit() method and is terminated.

UC_02_11: **Alternative flow** of events:

If user clicks outside of area of button actively listening for input, no action taken.

In the case of a failure of this process the User can instead use system services to close the game, through the keyboard alt-F4 shortcut or through the windows task manager.

UC_02_ 12: **Testing**: When testing, ensure that on mouse click, the exit game button calls the exit game function.

UC_03

UC_03_1: **Use Case**: Save Game
UC_03_2: **Author**: CP
UC_03_3: **Date**: 15/11/21
UC_03_4: **Purpose**: Save game progress from within game window
UC_03_5: **Overview**: Starts when User presses save game button on main menu screen. This should save the player's progress through the map where the user has the option to save the game under a name.
UC_03_6: **Cross References**: US_02_1, US_02_2
UC_03_7: **Actors**: User
UC_03_8: **Pre-conditions**:
UC_03_Pre-1: Program should be in main menu scene
UC_03_Pre-2: Save button must be "listening" for click event with prepared save game function actioned on-click
UC_03_9: **Post Conditions**:
UC_03_Post_1: The program saves and stores current game progress under the player's name.
UC_03_10: **Main flow**:

When the player is in the main menu and decides they want to save their progress they select the save game button and the player is prompted to enter their name.

The player enters their name and selects okay.

The player sees their name listed under saved games.

UC_02_11: **Alternative flow** of events:

The player selects the save game button but is not prompted to enter their name.

After the player enters their name, they cannot see it listed under saved games.

UC_02_12: **Testing**:
When testing ensure the game is saved with the correct name entered and when the game is re-opened the user is at the same map position with the same number of troops as when they left the game.

UC_04

UC_04_1: **Use Case**: Main Menu
UC_04_2: **Author**: CP
UC_04_3: **Date**: 15/11/21
UC_04_4: **Purpose**: Interface for saving and exiting the game
UC_04_5: **Overview**: The user should be able to select a main menu option from the map scene from which they can choose to load a game, save or exit the game. User should also be met by the main menu upon entry to the game.
UC_04_6: **Cross References**: US_02_1, US_02_2
UC_04_7: **Actors**: User
UC_04_8: **Pre-conditions**:
UC_04_Pre-1: Program should be in map scene or have just been run
UC_04_9: **Post Conditions**:

UC_04_Post_1: The program displays the main menu with all its options.
UC_04_10: **Main flow**:

When the player is in the map scene, they select the main menu option, triggering the main menu screen.
The main menu is displayed with options to load game, save game or exit game
When the player enters the game they are met by the main menu scene displaying options to load game, save game or exit game

UC_04_11: **Alternative flow** of events:

The player selects the main menu button but the main menu scene is not triggered.
After the player enters the game they are not met by the main menu screen and instead transferred directly to the dungeon map scene.

UC_04_12: **Testing**:
When testing ensure the main menu screen is triggered upon selecting the main menu button. Ensure that upon entry to the game the player is met by the main menu screen.

UC_05
UC_05_1: **Use Case**: Popup displayed
UC_05_2: **Author**: CP
UC_05_3: **Date**: 15/11/21
UC_05_4: **Purpose**: Inform player of a game consequence.
UC_05_5: **Overview**: When the user takes an action and a consequence is reached, they should be informed of the resulting consequence.
UC_05_6: **Cross References**: US_07_01, US_07_02, US_15_01, US_15_02.
UC_05_7: **Actors**: User interface.
UC_05_8: **Pre-conditions**:

UC_05_Pre-1: Player has made an action in the game.
UC_05_Pre-2: The action taken has a consequence.

UC_05_9: **Post Conditions**:

UC_04_Post_1: The player understands the consequence of the action taken.

UC_05_10: **Main flow**:

The player makes an action in the game (e.g. enters a treasure node)
The popup is displayed clearly on the scene, with the background still visible
The player understands the consequence of the action taken (e.g. has been awarded 4 units).

UC_05_11: **Alternative flow** of events:

The player makes an action in the game (e.g. enters a treasure node)
The popup is not displayed clearly on the scene.
The popup is displayed clearly on the scene; however, the background is not visible.
The popup is displayed clearly on the scene but the player does not understand the consequence of the action taken.

UC_05_12: **Testing**:
When testing ensure that the popup is correctly aligned in the centre of the screen and the background scene is still visible to the user. Ensure that a situationally required message can be displayed dependent on the action triggering the popup.

UC_06
UC_06_1: **Use Case**: Enter treasure node.
UC_06_2: **Author**: CP
UC_06_3: **Date**: 15/11/21
UC_06_4: **Purpose**: Informs user of their treasure reward units
UC_06_5: **Overview**: When a user selects a treasure node they should be greeted by a popup informing them of how many troops they have gained as a reward.

UC_06_6: **Cross References**: US_07_1, US_07_2, US_09_1, US_09_2, US_10_1, US_10_2,
UC_06_7: **Actors**: User, User interface
UC_06_8: **Pre-conditions**:
UC_06_Pre-1: Programme should be on map scene
UC_06_Pre-2: Treasure node is adjacent to current node.
UC_06_9: **Post Conditions**:
UC_06_Post_1: New occupied node should now be the treasure node
UC_06_Post_2: A dismissible popup should now displayed on the screen informing the user of troop reward
UC_06_Post_3: Unit pool should be updated to reflect additional troops gained.
UC_06_10: **Main flow**:
User sees a treasure node adjacent to current node and selects it.
After the treasure node is selected, a dismissible popup is displayed informing the user of their reward.
After the popup is dismissed the unit pool is updated according to the number of troops awarded.
UC_06_11: **Alternative flow** of events:
User sees a treasure node adjacent to current node, but cannot select it.
After selecting the treasure node, the programme fails to display a popup message.
After selecting a treasure node, a popup is displayed showing the chest reward, but the user is unable to dismiss it.
After the popup is dismissed, the unit pool is not updated to show the extra units awarded.
UC_06_12: **Testing**:
When testing, ensure selection of the treasure node is recognised by the programme and this can be used to trigger a popup. Ensure the popup can successfully be dismissed and that the popup management system updates the unit manager to reflect the additional units gained so that this is displayed in the unit pool.

UC_07
UC_07_1: **Use Case**: Welcome node
UC_07_2: **Author**: CP
UC_07_3: **Date**: 15/11/21
UC_07_4: **Purpose**: Introduce the player to the game.
UC_07_5: **Overview**: Upon entering the game the player should be able to select the starting node where a welcome message is displayed, welcoming the player to the game and informing them of the point they must reach to complete it.
UC_07_6: **Cross References**: US_03_1, US_03_2
UC_07_7: **Actors**: User, User Interface
UC_07_8: **Pre-conditions**:
UC_07_Pre-1: The game has been started and the map scene initiated.
UC_07_Pre-2: No nodes have been clicked and activated yet.
UC_07_9: **Post Conditions**:
UC_07_Post_1: A dismissible popup should be displayed welcoming the player and informing them of the game endpoint (star)
UC_07_10: **Main flow**:
The player enters the game and clicks on the first node on the map.
A dismissible popup is displayed welcoming the player to the game and informing them they must reach the star node at the end of the map.
Popup is dismissed and welcome node is now darkened indicating it is the user's current position on the map.
UC_07_11: **Alternative flow** of events:

The player enters the game and clicks on the first node.
A dismissible popup fails to be displayed to welcome the player or inform them of the objective.
A dismissible popup is created but it is not visible on the screen and therefore cannot be dismissed.
Popup is dismissed but the welcome node is still selectable indicating the user hasn't taken position at the welcome node yet.

UC_07_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct welcome message. Ensure node is darkened and no longer selectable after dismissal.

UC_08

UC_08_1: **Use Case**: New arena node
UC_08_2: **Author**: CP
UC_08_3: **Date**: 15/11/21
UC_08_4: **Purpose**: Transfers the player to the first node of a new arena and informs them they've completed the previous arena.
UC_08_5: **Overview**: When a player have taken a path through an arena and made it to the end, they should be able to select a new arena node which informs them they've completed the previous arena. The player can then choose between different paths through the new arena.
UC_08_6: **Cross References**: US_06_1, US_06_2
UC_08_7: **Actors**: User
UC_08_8: **Pre-conditions**:
UC_08_Pre-1: Player has reached and completed one of the final nodes of their current arena.
UC_08_Pre-2:
UC_08_9: **Post Conditions**:
UC_08_Post_1: Player has the option to select from different starting nodes through the new arena with different paths and encounters.
UC_08_10: **Main flow**:
The player selects the new arena node.
A popup is displayed informing the player that they can now enter a new arena.
The player is transferred to the new arena node and can now select a path through the new arena.
UC_08_11: **Alternative flow** of events:
The player is unable to select the new arena node
The player selects the new arena node but a popup informing the player they can now enter a new arena fails to be displayed.
The player fails to be transferred to the new arena node and therefore cannot select a path through the new arena.
UC_08_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct enter new arena message displayed. Ensure node is darkened and no longer selectable after dismissal.

UC_09

UC_09_1: **Use Case**: Game completion node
UC_09_2: **Author**: CP
UC_09_3: **Date**: 15/11/21
UC_09_4: **Purpose**: Initiates and informs user of completion of the game.
UC_09_5: **Overview**: There should be a final node on the map, whose reaching is the objective of the game. Upon entry, the player should be informed they have completed the game and congratulated.

UC_09_6: **Cross References**: US_01_1, US_01_2
UC_09_7: **Actors**: User
UC_09_8: **Pre-conditions**:
UC_09_Pre-1: The player has successfully completed the last battle node of the game.
UC_09_Pre-2: The player has returned to the map scene.
UC_09_9: **Post Conditions**:
UC_09_Post_1: The game is complete and a message is shown informing the user they have won.
UC_09_10: **Main flow**:
After completing the final battle encounter the player has returned to the map scene.
The player selects the final node on the map.
A dismissible popup is initiated, with a message congratulating the user for winning the game.
UC_09_11: **Alternative flow** of events:
After completing the final battle encounter the player has returned to the map scene.
The player selects the final node on the map.
A dismissible popup fails to be initiated and the player is not informed they have won the game.
UC_09_12: **Testing:**
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct congratulatory message displayed. Ensure node is darkened and no longer selectable after dismissal.

UC_10
UC_10_1: **Use Case**: Update unit pool
UC_10_2: **Author**: CP
UC_10_3: **Date**: 15/11/21
UC_10_4: **Purpose**: Reflects additional troops awarded to the player
UC_10_5: **Overview**: When a player is awarded units from a treasure node, the unit pool should be updated to show the new total number of units.
UC_10_6: **Cross References**: US_06_1, US_06_2
UC_10_7: **Actors**: User
UC_10_8: **Pre-conditions**:
UC_10_Pre-1: Player has selected a treasure node.
UC_10_Pre-2: Unit reward has been determined.
UC_10_9: **Post Conditions**:
UC_10_Post_1: Unit pool should reflect the units awarded on top of the units in possession preceding node selection
UC_10_10: **Main flow**:
User selects a treasure node and is informed of unit reward.
Unit pool is updated to show the sum of the player's previous number of units and the number of units awarded.
UC_10_11: **Alternative flow** of events:
User selects a treasure node and is informed of unit reward.
Unit pool is not updated and remains the same as before node entry.
Unit pool is updated; however the new value is not the sum of the player's previous number of units and the number of units awarded.
UC_10_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct unit award message displayed. Ensure that the unit count and unit pool are updated to reflect the extra troops rewarded.

UC_11

UC_11_1: **Use Case**: Enter battle node
UC_11_2: **Author**: CP
UC_11_3: **Date**: 15/11/21
UC_11_4: **Purpose**: Allow user to enter a battle node to progress through the map.
UC_11_5: **Overview**: When the user is in the map scene and the adjacent node is a battle node, they should be able to select the node and be transferred to the respective level's grid scene.
UC_11_6: **Cross References**: US_04_1, US_04_2, US_08_1, US_08_2, US_09_1, US_09_2, US_10_1, US_10_2
UC_11_7: **Actors**: User
UC_11_8: **Pre-conditions**:
UC_11_Pre-1: Programme is on map scene.
UC_11_Pre-2: These is a battle node adjacent to the current node.
UC_11_9: **Post Conditions**:
UC_11_Post_1: The player is transferred to grid scene
UC_11_10: **Main flow**:
The player is currently at a node with a battle node adjacent.
The player selects the battle node.
Upon selecting the battle node, the player is transferred to the grid scene.
UC_11_11: **Alternative flow** of events:
The player selects the battle node; however, they are unable to select the battle node.
The player selects the battle node; however, they are not transferred to the grid scene.
UC_11_12: Testing:
When testing ensure that the grid scene is loaded successfully with the unit pool of the unit placement scene reflecting the number of units in the unit count of the map scene.

UC_12

UC_12_1: **Use Case**: Enemy unit health scaling
UC_12_2: **Author**: CP
UC_12_3: **Date**: 15/11/21
UC_12_4: **Purpose**: To add difficulty to battle encounters as the player progresses through the game
UC_12_5: **Overview**: As the player progresses through the game, the health of enemy units should increase at each battle encounter.
UC_12_6: **Cross References**: US_09_1, US_09_2, US_12_1, US_12_2
UC_12_7: **Actors**: User
UC_12_8: **Pre-conditions**:
UC_12_Pre-1: Player has completed at least one battle encounter.
UC_12_9: **Post Conditions**:
UC_12_Post_1: The enemy units encountered at the subsequent battle node have more health points than those of the previous node.
UC_12_10: **Main flow**:
The player completes a battle encounter.
The player enters a subsequent battle node and encounters enemy units with one more health point than those of the previous node.
UC_12_11: **Alternative flow** of events:
The player completes a battle encounter.
The player enters a subsequent battle node; however, the enemy units encountered have the same amount of or less health points than those of the previous node.
UC_12_12: **Testing**:
When testing ensure that the enemy units in each subsequent battle node have one extra health point.

UC_13

UC_13_1: **Use Case**: Highlight tile

UC_13_2: **Author**: CP
UC_13_3: **Date**: 15/11/21
UC_13_4: **Purpose**: Highlights the tile which the user's cursor is interacting with at its current position.
UC_13_5: **Overview**: During the grid scene, the player should be able to see which tile their cursor is interacting with according to its position, where the current tile is highlighted
UC_13_6: **Cross References**: US_05_1, US_05_2, US_13_1, US_13_2
UC_13_7: **Actors**: User
UC_13_8: **Pre-conditions**:
UC_13_Pre-1: The programme should be in the grid scene, and the board displayed
UC_13_9: **Post Conditions**:
UC_13_Post_1: When the cursor moves over a tile, the tile is highlighted.
UC_13_10: **Main flow**:
During the grid scene the player moves the cursor over a specific tile.
The tile the cursor is positioned over is highlighted.
UC_13_11: **Alternative flow** of events:
During the grid scene the player moves the cursor over a specific tile.
The tile the cursor is positioned over is highlighted along with one or more other tiles which the cursor is not positioned over.
The tile the cursor is positioned over is not highlighted; however one or more other tiles which the cursor is not positioned over are highlighted.
The tile the cursor is positioned over is not highlighted nor is any other tile on the board.
UC_13_12: **Testing**:
When testing ensure that only the tile the cursor is hovering over is highlighted and this is reflected as the cursor is moved around the grid.

UC_14
UC_14_1: **Use Case**: Select highlighted tile
UC_14_2: **Author**: CP
UC_14_3: **Date**: 15/11/21
UC_14_4: **Purpose**: To enable action to be performed upon selection of a tile.
UC_14_5: **Overview**: During the grid scene, the player should be able to select a highlighted tile which the cursor is positioned over and the programme should recognise this selection enabling an action to be performed.
UC_14_6: **Cross References**: US_08_01, US_08_02, US_05_1, US_05_2, US_13_1, US_13_2
UC_14_7: **Actors**: User
UC_14_8: **Pre-conditions**:
UC_14_Pre-1: The programme should be in the grid scene
UC_14_Pre-2: The user should have the cursor positioned over the highlighted tile which they wish to select.
UC_14_9: **Post Conditions**:
UC_14_Post_1: The tile is selected, turning grey and the action is recognised by the programme.
UC_14_10: **Main flow**:
During the grid scene the player moves the cursor over a specific tile, which is highlighted.
The cursor changes to a hand symbol letting the player know its clickable,
The player selects the tile, which turns grey, meaning the programme recognises the action.
UC_14_11: **Alternative flow** of events:
During the grid scene the player moves the cursor over a specific tile, which is highlighted
The cursor remains unchanged, so they do not know its clickable.

The cursor changes to a hand symbol, letting the player know its clickable, the player selects the tile; However, the tile does not change colour, meaning the programme does not recognise the action.

UC_14_12: **Testing**:
When testing ensure that it is only the highlighted tile which is selected and the programme recognises the action.

UC_15

UC_15_1: **Use Case**: Place units on grid
UC_15_2: **Author**: DD
UC_15_3: **Date**: 15/11/21
UC_15_4: **Purpose**: Allow user to place units on grid before battle
UC_15_5: **Overview**: Starts during placement phase of auto-battler encounter. Player should be able to select from persistent pool of units, visible on the lower edge of the screen in a relevant UI object, and these can then be placed in empty hexes on the board on the lower half of the game board.
UC_15_6: **Cross References**: US_08_01, US_08_02, US_11_01, US_11_02, US_14_01, US_14_01, US_15_01, US_15_02
UC_15_7: **Actors**: User
UC_15_8: **Pre-conditions**:
UC_15_Pre-1: Program should be within encounter scene
UC_15_Pre-2: Current state of encounter should be planning
UC_15_Pre-3: User must have units available to place
UC_15_Pre-4: At least one square on board should not be full
UC_15_Pre-5: Current units placed must be less than maximum placeable units
UC_15_9: **Post Conditions**:
UC_15_Post_1: Selected unit should be placed on the game board
UC_15_Post_2: The Square on the board the unit now occupies should no longer be empty
UC_15_Post_3: One copy of the selected unit should be removed from the unit pool temporarily
UC_15_10: **Main flow of events**:
Player Enters the battle node initiating the placement phase of the battle encounter.
Player select a unit from their unit pool.
Unit is placed on one of the empty hexes on the board.
UC_15_11: **Alternative flow** of events:
User tries to place unit on non-empty square, explanatory error message is shown
User tries to place unit from empty pool of units, explanatory error message is shown
UC_15_12: Testing:
When testing, ensure unit clicked becomes selected. Also ensure when subsequently clicking on tile that a new unit of the selected type is instantiated, with a position equal to the selected tile.

UC_16

UC_16_1: **Use Case**: Unit constant damage output
UC_16_2: **Author**: CP
UC_16_3: **Date**: 15/11/21
UC_16_4: **Purpose**: Allows units to deal damage to opposing units.
UC_16_5: **Overview**: Units consistently output damage with amount and rate determined by the unit type.
UC_16_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2
UC_16_7: **Actors**: User
UC_16_8: **Pre-conditions**:
UC_16_Pre-1: Units are placed on the board

UC_16_Pre-2: Fight sequence is initiated

UC_16_9: **Post Conditions**:

UC_16_Post_1: Damage is consistently output while the unit is not dead.

UC_16_10: **Main flow**:

The fight sequence is initiated and all units on the board begin outputting damage.

Damage is consistently output until the unit is dead.

UC_16_11: **Alternative flow** of events:

The fight sequence is initiated; however, some or all the units do not begin outputting damage.

Damage is consistently output by each unit, even after the unit is dead.

UC_16_12: **Testing**:

When testing ensure the unit damage output is accessible to other units so that they can use it to take damage.

UC_17

UC_17_1: **Use Case**: Unit take damage

UC_17_2: **Author**: CP

UC_17_3: **Date**: 15/11/21

UC_17_4: **Purpose**: Allows units to take damage from other units

UC_17_5: **Overview**: When a unit is in range of an opposing unit, it should be dealt damage corresponding to the opposing unit's damage output.

UC_17_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2

UC_17_7: **Actors**: User

UC_17_8: **Pre-conditions**:

UC_17_Pre-1: The unit has been engaged by an opposing unit.

UC_17_Pre-2: The unit is in range of the opposing unit.

UC_17_9: **Post Conditions**:

UC_17_Post_1: The unit health is consistently reduced according to the opposing unit's damage output.

UC_17_10: **Main flow**:

The unit comes into range of an opposing unit which has engaged it.

The opposing unit begins consistently dealing damage to the unit while it is in range.

The health variable of the unit is consistently reduced according to the opposing unit's damage output.

UC_17_11: **Alternative flow** of events:

The unit comes into range of the opposing unit which has engaged it.

The opposing unit fails to deal damage to the unit

The opposing unit

The health attribute of a unit engaged in a 1v1 battle subceeds zero.

The dead Boolean of the unit object fails to be set to true.

UC_17_12: **Testing**:

When testing ensure the damage output of the unit dealing the damage is accessible to the one taking it. Ensure a corresponding amount of health points are deducted from the unit's health variable.

UC_18

UC_18_1: **Use Case**: Unit death

UC_18_2: **Author**: CP

UC_18_3: **Date**: 15/11/21

UC_18_4: **Purpose**: Criteria for the elimination of a unit from a battle.

UC_18_5: **Overview**: When the health of a unit has reached or subceeded zero, the unit is given the dead attribute, which can then be used to disengage the opposing unit and identify unit objects which need to be destroyed.
UC_18_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2
UC_18_7: **Actors**: User
UC_18_8: **Pre-conditions**:
UC_18_Pre-1: The unit in question has been engaged in a 1v1 battle.
UC_18_Pre-2: The health of the unit is less than or equal to zero.
UC_18_9: **Post Conditions**:
UC_17_Post_1: The unit is given the dead attribute which is a Boolean set to true.
UC_18_10: **Main flow**:
The health attribute of a unit engaged in a 1v1 battle subceeds zero.
The "dead" Boolean of the unit object is set to true.
The "dead" unit object is destroyed, eliminating it from the battle
UC_18_11: **Alternative flow** of events:
The health attribute of a unit engaged in a 1v1 battle subceeds zero.
The dead Boolean of the unit object fails to be set to true.
The dead Boolean of the object is set to true but it is not destroyed, failing to eliminate it from the game.
UC_18_12: **Testing**:
When testing ensure that the unit object is successfully destroyed upon its dead Boolean being set to true

UC_19
UC_19_1: **Use Case**: Play game in browser
UC_19_2: **Author**: CP
UC_19_3: **Date**: 15/11/21
UC_19_4: **Purpose**: Allow players to play game in a browser across multiple devices.
UC_19_5: **Overview**: The game should be playable in a browser.
UC_19_6: **Cross References**: US_16_1, US_16_2
UC_19_7: **Actors**: User
UC_19_8: **Pre-conditions**:
UC_19_Pre-1: The game is fully functional.
UC_19_9: **Post Conditions**:
UC_19_Post_1: The game is playable on a web browser.
UC_19_10: **Main flow**:
The game is uploaded to unity free hosting service and embedded in a web page.
The game is playable online
UC_19_11: **Alternative flow** of events:
The game fails to be uploaded to unity free hosting service
UC_19_12: **Testing**:
When testing ensure that the game is playable across a range of devices on a range of web browsers.

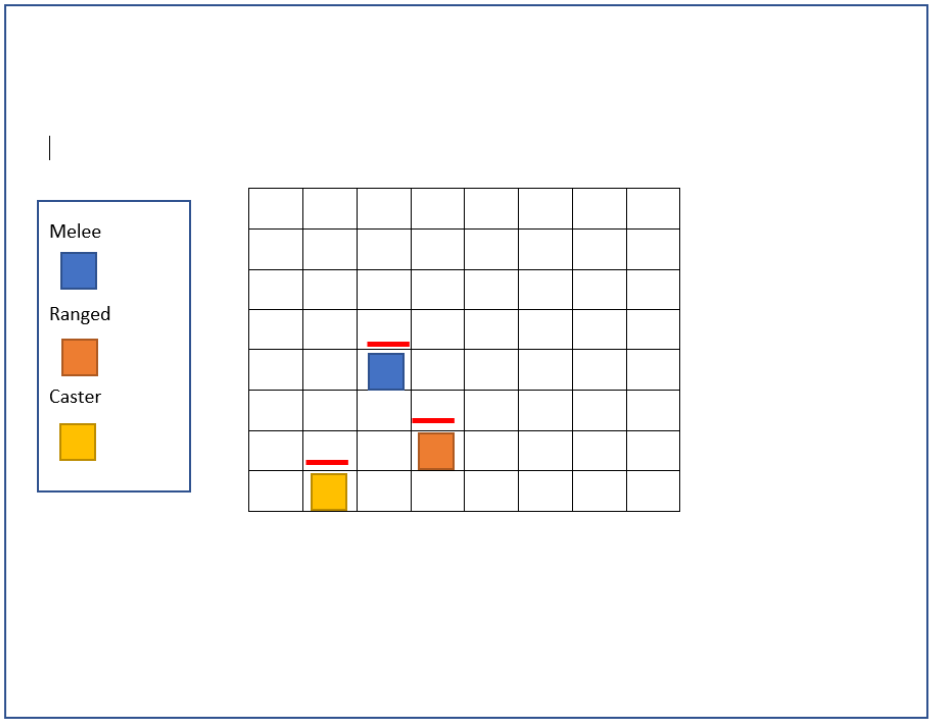
UC_20
UC_20_1: **Use Case**: Move on Map
UC_20_2: **Author**: DD
UC_20_3: **Date**: 22/11/21
UC_20_4: **Purpose**: Allow user to change current location to adjacent node to progress through map
UC_20_5: **Overview**: When user is in the map scene, they should be able to select any of the adjacent nodes on the map, and their current position should then be changed to this node. This should also be visually indicated on the scene

UC_20_6: **Cross References**: US_05_1, US_05_2, US_06_1, US_06_2, US_07_1, US_07_2, US_13_1, US_13_2, US_19_1, US_19_2
UC_20_7: **Actors**: User
UC_20_8: **Pre-conditions**:
UC_20_Pre-1: Program should be in map scene
UC_20_Pre-2: Current party position should not be at end node
UC_20_9: **Post Conditions**:
UC_20_Post_1: Current party position should change to selected node
UC_20_Post_2: The visual indicator for party position should change to selected node
UC_20_Post_3: Event for new node position should be triggered
UC_20_10: **Main flow**:
UC_20_11: **Alternative flow** of events:
If a user selects a node proximal in the tree, the party position should remain at the current node
If a user selects the currently occupied node, the party position should remain at the current node
UC_20_12: **Testing**: When testing, ensure only nodes distal and adjacent to active nodes are selectable. Also ensure that when selectable nodes are clicked, the party position then changes to this current node.

UC_21
UC_21_1: **Use Case**: Dismiss Popup
UC_21_2: **Author**: CP
UC_21_3: **Date**: 22/11/21
UC_21_4: **Purpose**: Allow user to Dismiss a popup from the screen.
UC_21_5: **Overview**: When a popup is displayed, the user should be able to easily select a dismiss button to close it.
UC_21_6: **Cross References**: US_15_1, US_15_2, US_20_1, US_20_2
UC_21_7: **Actors**: User
UC_21_8: **Pre-conditions**:
UC_04_Pre-1: A dismissible popup is currently in display on the current scene.
UC_21_9: **Post Conditions**:
UC_04_Post_1: Popup is dismissed from the scene.
UC_20_10: **Main flow**:
The user successfully selects the dismiss button.
The popup is closed.
UC_21_11: **Alternative flow** of events:
The user fails to select the dismiss button.
The user successfully selects the dismiss button; however, the popup isn't closed
UC_21_12: **Testing**:
When testing ensure that selecting the dismiss option of the popup successfully destroys the popup object.

User interface design | 4.11.1

Auto-Battler UI



Elements

Unit Sprites - References **US_05_1, US_05_2, US_08_1, US_08_2, UC_03, Units, Renderer**

8x8 grid comprised of Tiles **US_08_01, US_08_02, US_11_01, US_11_02, US_14_01, US_14_01, US_15_01, US_15_02, UC_03, Tiles, Renderer, Grid Manager**

Unit selector on left side of window **US_5_1, US_5_2, US_8_1, US_8_2, US_13_1, US_13_2, US_15_1, US_15_2, UC_03, Renderer, Units**

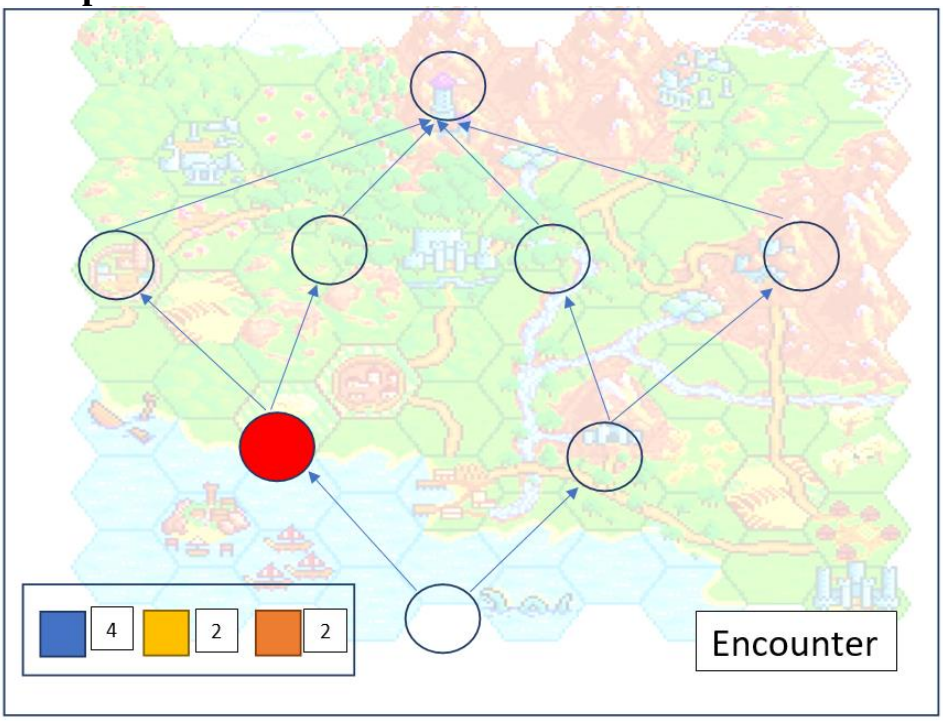
Should indicate types available, and how many of each type remaining

Appropriate methods implemented to enable clicking on unit type to select, and then clicking on Tile to place

Health bars above unit sprites **US_15_1, US_15_2**

Method should resize bar depending on current health

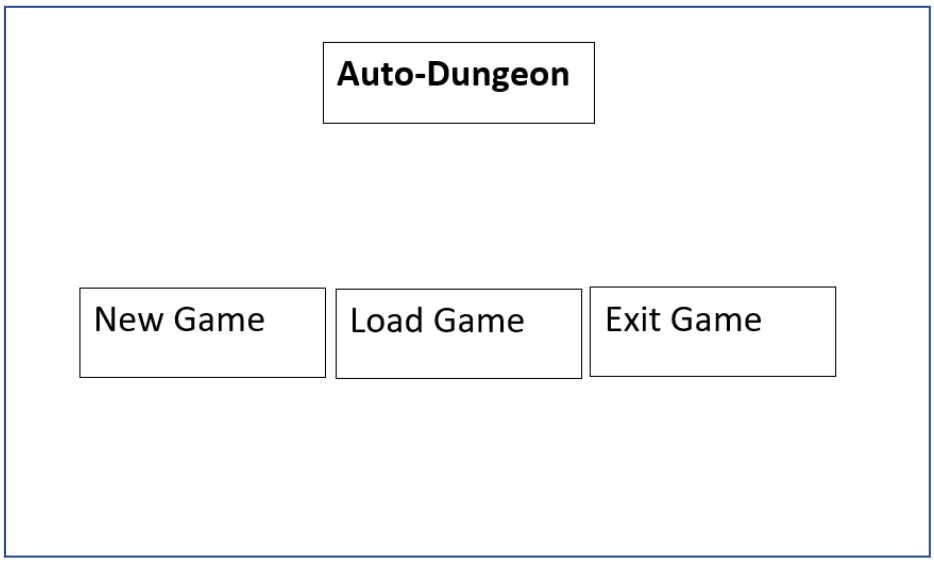
Dungeon Map UI



Elements

Tree Nodes **US_06_1, US_06_2, US_07_1, US_07_2, UC_04, Dungeon_Tree**
Should have method to change colour to indicate node currently occupied by party
Should have method to determine which encounter and then change scene
Unit's remaining indicator **US_07_1, US_07_2, US_09_1, US_09_2, US_11_1, US_11_2, US_14_1, US_14_2, US_15_1, US_15_2, Dungeon_Tree, Units**
Should visually show the number and composition of remaining units to aid decision making
Encounter button to enter encounter at current Node
Should have visual "overworld" map to help in telling underlying game story
Encounter difficulty should increase as game progresses

Main Menu UI



Elements
New Game Button **US_03_1, US_03_2, UC_01**
Should have on-click method ready to start new game method
Load Game Button **US_02_1, US_02_2**
Should have on-click method ready to start load game method
Method should load available save game, display error message if no available save game
Exit Game Button **US_01_1, US_01_2, UC_02**
Should close game window and end all associated processes

Sprint 5: 27/11/21-03/12/21

The focus of sprint 5 was integrating our game together and further developing both our map and battle scenes.

Overview

What's supposed to happen this sprint

No.	Expected Start	Expected Finish	Objective
1	26/11/21	01/12/21	Clean up and improve the map
2	26/11/21	03/12/21	Create drag and drop functionality
3	26/11/21	03/12/21	Create battle functionality
4	26/11/21	08/12/21	Integrate into single game
5	26/11/21	01/12/21	Prep for game testing with participants
6	26/11/21	01/12/21	Complete documentation for sprint 4

Reviews

What actually happened

No.	Started	Finished	Task	Doc
1	29/11/21	29/11/21	Daily scrum	5.1.1
2	01/12/21	01/12/21	Daily scrum	5.2.1
3	03/12/21	03/12/21	Daily scrum, sprint review, sprint retrospective, sprint planning	5.3.1
5	Throughout		Backlog updated	5.4.1
6	Throughout		Exception handling	5.5.1
7	Throughout		User stories with acceptance tests	5.6.4
8	Throughout		Requirements use cases	5.7.3
9	Throughout		CRC cards	5.8.3
10	Throughout		Design use cases	5.9.3
11	Throughout		User interface design	5.10.3

Daily scrum | 5.1.1

29/11/21, 10:00-10:15, Microsoft Teams

Attendance

All present except CP
Meeting led by NO
Minutes taken by NO

Minutes

1. Within the last 48 hours:

    a. DD added some new templates to the grid scene where drag and drop would be and added some new assets to make the UI more pleasing eye. Stitched the 2 scenes together to make it less buggy
    b. TW create a new branch for map
    c. QZ got the units to move around the board and added a find target mechanism. Will add it to a new branch

2. Issues that arose within the last 48 hours:

    a. Figuring out how to make it work post integration
    b. Managing all the different workloads

3. Within the next 48 hours:

    a. LW will keep up with project documentation
    b. LG will deal with sprint 4 documentation
    c. DD and TW will meet sometime before Wednesday to work out the map
    d. LG, TW, IL to look at backlog for what to do next
    e. Grid squad to meet up and split up work

4. Foreseeable issues within the next 48 hours are:

    a. LW might not be able to come in for customer meeting
    b. Clearing up who is working on what inside the grid

Daily scrum | 5.2.1

01/12/21, 14:15-14:30, Microsoft Teams

Attendance

All present except IL
Meeting led by NO
Minutes taken by NO

Minutes

1. Within the last 48 hours:

    a. DD pushed some things to the branches. Drag and drop and highlight, and unified the unit counts, letting the numbers be carried over from map to grid and grid to map
    b. DD and TW met last night and did some work on the map
    c. TW changed the design of the map to use Unity's Tile map to create custom maps, map still has previous functionality
    d. CP has been sick so hasn't been able to do anything
    e. QZ added buttons for battling but some bugs appeared. Player unit does not move when fight is initiated.

2. Issues that arose within the last 48 hours:

    a. Minor bugs with the mouse and the sprite (clicking 24 pixels to one side)

3. Within the next 48 hours:

    a. TW wants to animate the icon on the map to show it's the one being chosen
    b. Battle scenario to be created
    c. Getting both sets of bots moving and fighting and disappearing when dead
    d. Finish off documentation
    e. QZ, CP, DD to meet up

4. No issues foreseen

End of sprint meeting | 5.3.1

03/12/21, 13:15-14:30, NH 2.17a

Attendance

Everyone present
Minutes taken by NO
Meeting led by NO

Daily scrum

1. Within the last 48 hours:
    a. DD has added the full drag and drop feature to the game
2. Issues that arose within the last 48 hours:
    a. Accidentally, DD broke the GitHub, but TW helped him fix it
    b. Change the yellow highlight to a more aesthetically pleasing one

Sprint review

1. The team felt that there were a lot of external deadlines this week that reduced productivity
2. TW would have liked to add an animation to the map icons and to have heard about Tilemap earlier
3. DD found it fine but struggled to understand the grid code in the beginning
4. CP could not complete his tasks
5. LW kept up with PB and added tasks as he felt was needed
6. TW would like to add something to let users know about the difficulty of the level
7. TW also mentions we need to make a tutorial
8. The team discussed that there was currently no end state to the battle, that declared victory or loss, and supplied the user with the means to return to the map.
    a. The replay functionality was also brought up during discussion
9. The team felt that the game was lacking in story and that it would be nice to add in some story elements
10. The lack of a customer meeting did reduce the feelings of urgency, but the team did discuss the possibility of game testing with outsider participants:
    a. The customer had approved the survey chosen
    b. The team felt it best to push testing to Sprint 7 due to backlog overflow into sprint 6
11. LW touched bases with everyone
12. The team also discussed potential additions to the backlog:
    a. End screen can show the reward gotten from winning the level/game
    b. Implement star levels where the number of stars corresponding with how well the level was played
    c. Ledger that records what units did

d. Use the unity asset store for free assets to improve current sprites and switch out inappropriate sprites
e. Increase difficulty the further into the map the player goes

Sprint retrospective

1. The team agreed that there was not enough focus on priorities, and we are at risk of falling behind on our progress.

2. The team tried to integrate all git branches together but ran into issues

    a. Instead of GitHub, use unity's collaboration tool

3. GitHub was not as helpful as we thought it was going to be

    a. Caused all the problems we were hoping to avoid

Sprint planning

1. Within the next 48 hours, the following will be addressed as needed:

    a. Battling UI
    b. Turn based attack function

2. Foreseeable issues within the next 48 hours (about 2 days) are:

    a. CW deadlines

3. DD will fix issues with git

4. The team all agreed upon a list of priorities for next week:

    a. Have them dealing damage adding onto QZ fight button
    b. Finding the right sprites for the units
    c. End state (success and failure)
    d. HP bars
    e. Removing units after killing
    f. Briefing document
    g. Main menu

Task assignments

	A	AB	AC	AD	AE	AF	AG	AH
1	**Date**	**04/12/21**	**05/12/21**	**06/12/21**	**07/12/21**	**08/12/21**	**09/12/21**	**10/12/21**
2	**Sprint**	**Sprint 6**						
3	**Qinyuan Zhuang**	End state and fight button (lead developer on the battle side of things so go to him for help)						
4	**Lawrence Wong**	HP bars and removing uits						
5	**Callum Pitceathly**	Dealing damage						
6	**Drew Dundee**	Dealing damage						
7	**Naomi Okiddy**	HP bars and removing uits						
8	**Lucy Goodchild**	Product owner						
9	**Ieuan Lambert**	Scrum master						
10	**Tim Win**	UI and sprites						
11	**Everyone**							
12	**Daily scrum**			10:00 MS Teams		14:15 CB 5.13		
13	**Daily scrum, sprint review, sprint retrospective, and sprint planning**							13:15 NH 2.17a
14	**Customer meeting**					14:45 CB 5.13		
15								

Backlog | 5.4.1

Complete backlog tasks

1. Placing units and hover highlight on the tile
2. Basic unit movement
3. Drag and drop feature
4. Battler and Asset UI
5. Integration Issues
6. Scene Changing: Map to Grid

New backlog tasks

1. HP bars for units
2. A ‘Fight’ button to start the battle
3. Get appropriate assets and sprites
4. Ranged units
5. End screen after battle
6. Kill feed
7. Pop-up for battle nodes

Current backlog

1. Main menu
2. Map randomiser
3. Change the tilesets based on difficulty
4. Main menu music
5. Recording feature
6. Sound effects

Exception handling | 5.6.1

Integration issues

1. The integration problems were unexpected as we had expected that using Github would reduce these issues instead. The problems were resolved within 48 hours with manual integration being performed by DD.

2. Minor bugs with the mouse sprite and with it clicking approx. 24 pixels to one side. Resolved within 48 hours by DD. Resolved by fixing mouse position vector and camera to world position vector.

3. A push to the Github broke the repository. Resolved within 48 hours by DD and TW. Repository was restored to previous state.

Customer meeting cancelled

1. The customer meeting did not take place this week due to Industrial action.

2. Coursework workload from other modules also increased.

3. Both factors meant the team deprioritised work related to the sprint.

User stories with acceptance tests | 5.7.4

US_01_1: Description: "As a user, I'd like a way to exit the program when I'm finished, from within the game window"
US_01_2: Acceptance Criteria: Accepted when quit game button implemented in main menu

US_02_1: Description: "As a user, I'd like to be able to continue a persistent game across multiple sessions"
US_02_2: Acceptance Criteria: Accepted when save game system implemented

US_03_1: Description: "As a user, I need a way to initialise a new game from within the game window"
US_03_2: Acceptance Criteria: Accepted when new game button implemented in main menu

US_04_1: Description: "As a user, I'd like multiple varied dungeons to add replayability"
US_04_2: Acceptance Criteria: Accepted when more than one dungeon implemented

US_05_1: Description: "As a user, I'd like to be able to make decisions about the units within my team, to add challenge and reward planning and critical thinking"
US_05_2: Acceptance Criteria: Accepted when able to select from multiple unit types during planning phase of combat

US_06_1: Description: "As a user, I'd like to have additional choice and variety in my gameplay, to separate this game from other auto-battlers"
US_06_2: Acceptance Criteria: Accepted when implement both persistent unit pools on main map, as well as being able to choose from branching paths

US_07_1: Description: "As a user, I'd like to make strategic decisions, with sufficient information to differentiate good and bad decisions"
US_07_2: Acceptance Criteria: Accepted when able to clearly see indication of reward and difficulty on each dungeon path.

US_08_1: Description: "As a user, I want frequent combat encounters as part of the core game"
US_08_2: Acceptance Criteria: Accepted when nodes in dungeon graph implemented, with encounters possible at each, including fully implemented auto-battling system

US_09_1: Description: "As a user, I enjoy progressive challenge to test my ability"
US_09_2: Acceptance Criteria: Accepted when persistent unit pool implemented, as well as progressively harder encounters culminating in "boss" encounter

US_10_1: Description: "As a user, I enjoy uncertainty, surprise, and variety in my gameplay"
US_10_2: Acceptance Criteria: Accepted when multiple encounter types implemented, with hidden possibility of each at given node

US_11_1: Description: "As a user, I'd like my strategic decisions to have visible, long-term consequences."
US_11_2: Acceptance Criteria: Accepted when persistent unit pool implemented, alongside choice of multiple branching paths

US_12_1: Description: "As a user, I enjoy dynamic events and decision making, with lots of variety"

US_12_2: Acceptance Criteria: Accepted when multiple encounter types implemented, with hidden possibility of each and given node

US_13_1: Description: “As a user, I enjoy decision making with restrictions and uncertainty to add challenge”
US_13_2: Acceptance Criteria: Accepted when planning phase implemented to combat with time restriction, alongside choice of branching paths

US_14_1: Description: “As a user, I’d like clear success and failure states to enable planning and goal setting”
US_14_2: Acceptance Criteria: Accepted when persistent pool of units implemented, with game-over occurring when depleted

US_15_1: Description: “As a user, I would like to understand the game’s UI/UX clearly, so they don’t interfere with gameplay”
US_15_2: Acceptance Criteria: Accepted when no elements of gameplay obscured behind UI but also all relevant information provided on UI

US_16_1: Description: “As a client, I would like the game to be accessible to as many people as possible so I can share it with friends/peers or monetise it in future”
US_16_2: Acceptance Criteria: Accepted when game is playable in browser across multiple devices

US_17_1: Description: “As a user, I would like to clearly see each action being taken sequentially rather than simultaneously, to ensure I can learn and adapt.”
US_17_2: Acceptance Criteria: Accepted when actions within battle are made and displayed sequentially with delay between.

US_18_1: Description: “As a user, I would like to be able to review the sequence of events after a “battle” in ensure I can learn and adapt.”
US_18_2: Acceptance Criteria: Accepted when user is able to either re-watch battle, or view some representation of the events that took place afterwards

US_19_1: Description: “As a user, I would like to make decisions in which route I should take on the game map rather than follow the one, set route.”
US_19_2: Acceptance Criteria: Accepted when game map forks to different battle nodes of varying difficulty

US_20_1: Description: “As a user, I would like a way to dismiss pop-ups easily and clearly.
US_20_2: Acceptance Criteria: Accepted when user is able to click on a dismiss button and the pop-up closes

US_21_1: Description: “As a user, I would like a way to form strategic formations with my units.”
US_21_2: Description: Acceptance Criteria: Accepted when drag and drop feature implemented

Requirements use cases | 5.8.3

UC_01:
Use Case: Load Game
Scope: Main menu
Level: User Goal
Context: Users can load a game progression by selecting the “load game” option from the main menu
Users are unable to select a “load game” option from the main menu
Frequency of occurrence: Any time the player enters the programme or wishes to load a game from the map scene
Open issues:

How to ensure a failsafe option to save game prompt in case player wishes to return to original game progression?

UC_02:
Use Case: Exit game
Scope: Main menu
Level: User Goal
Context: Users can exit the game by selecting the “exit game” option from the main menu
Users are unable to select “exit game” option from the main menu
Frequency of occurrence: Any time the player wishes to start exit the game.
Open issues:

How to ensure a failsafe option to save game prompt in case player forgets to save their progress before quitting the programme.

UC_03:
Use Case: Save Game
Scope: N/A
Level: User Goal
Context: Users can save a game by selecting the “save game” option from the main menu
Users are unable to select the “save game” option from the main menu
Frequency of occurrence: Any time the player wishes to save a game.
Open issues:

How do we store saves?
How many saves can be stored by the game?
How many characters can be used in naming the save?
Can we implement a failsafe in case two saves are given the same name?

UC_04:
Use Case: Main menu
Scope: N/A
Level: Subfunction
Context: Users can decide to save load or exit the game by entering the main menu
Users are unable to save load or exit the game as they cannot access the main menu
Frequency of occurrence: Any time the player enters the game or wishes to save, load or exit the game.
Open issues:

Do we display this as a popup or a scene?

UC_05: (COMPLETE)
Use Case: Popup displayed
Scope: N/A
Level: Subfunction
Context: Users are informed of a consequence when they make a game action.

Users fail to be informed of the consequence of a game action taken
Frequency of occurrence: Any time the player makes a game action.
Open issues:
How do we ensure consistent alignment of popups any time they are instantiated? (RESOLVED)
How do we set the informational text of each popup to be dependent on the action taken? (RESOLVED)
How do we ensure the background is still visible when a popup is triggered? (RESOLVED)

UC_06: (COMPLETE)
Use Case: Enter treasure node
Scope: N/A
Level: Subfunction
Context: Upon entering a treasure node, users are rewarded with extra units.
Upon entering a treasure node, users fail to be rewarded with extra units.
Frequency of occurrence: Any time the player enters a treasure node.
Open issues:
How is the number of units rewarded determined? (RESOLVED)
How do we link this to the update unit pool functionality? (RESOLVED)

UC_07: (COMPLETE)
Use Case: Welcome node
Scope: N/A
Level: Subfunction
Context: Users can select the welcome node to initiate progression through the map, and are informed of their objective.
Users can select the welcome node to initiate progression through the map but aren't informed of their objective
Frequency of occurrence: Any time the player enters the map scene of a new game.
Open issues:

UC_08:
Use Case: New Arena node
Scope: N/A
Level: Subfunction
Context: Users can select a subsequent new arena node and are informed they are progressing to a new arena.
Users can select a subsequent new arena node but aren't informed they're progressing to a new arena.
Frequency of occurrence: Any time the player can progress to a new arena.
Open issues:

UC_09:
Use Case: Game completion node
Scope: N/A
Level: Subfunction
Context: Users can select the game completion node to complete the game and are informed they have won.
Users can select the game completion node to complete the game but aren't informed they have won.
Frequency of occurrence: When the player has completed the final battle encounter of the game.
Open issues:

UC_10: (COMPLETE)
Use Case: Update unit pool

Scope: N/A
Level: User goal
Context: The number of units of the unit count and unit pool can be updated.
The number of units of the unit count and unit pool cannot be updated.
Frequency of occurrence: Every time the player is awarded units.
Open issues:
How do we ensure there is no conflict between the unit count number and the unit pool contents? (RESOLVED)

UC_11: (COMPLETE)
Use Case: Enter battle node
Scope: N/A
Level: Subfunction
Context: Users can select a battle node which initiates the grid scene.
Users can select a battle node but the grid scene fails to be initialised.
Frequency of occurrence: When the player has a battle node adjacent to their current node.
Open issues:
How do we trigger grid scene initialisation. (RESOLVED)
How do we ensure the grid scene initialised is the correct encounter for the current node (ie enemy difficulty etc) ? (RESOLVED)

UC_12:
Use Case: Enemy unit health scaling
Scope: N/A
Level: User goal
Context: Users can select the game completion node to complete the game and are informed they have won.
Users can select the game completion node to complete the game but aren't informed they have won.
Frequency of occurrence: When the player has completed the final battle encounter of the game.
Open issues:
How is scaling factor per node determined?

UC_13:
Use Case: Highlight tile (COMPLETE)
Scope: N/A
Level: Sub function
Context: When a user moves their cursor over a tile in the grid scene the tile is highlighted
When a user moves their cursor over a tile in the grid scene the tile remains unchanged.
Frequency of occurrence: Any time the user is in the grid scene and moves their cursor over a tile.
Open issues:
How do we identify which tile the cursor is over? (RESOLVED)
What colour should the tile be highlighted? (RESOLVED)

UC_14:
Use Case: Select highlighted tile (COMPLETE)
Scope: N/A
Level: User goal
Context: User can select a highlighted tile and the programme recognises the action
Users cannot select the highlighted tile
Frequency of occurrence: Whenever the cursor is above a tile and it is highlighted.
Open issues:
How do we ensure selection of the tile is linked to a distinct action? (RESOLVED)

UC_15: (COMPLETE)

Use Case: Place units on the grid
Scope: N/A
Level: User goal
Context: User can select a unit and place it on a grid tile
User can select a unit but cannot place it on a grid tile
Frequency of occurrence: Whenever the user is in the unit placement sequence.
Open issues:
How do we select a unit? (RESOLVED)
How do we ensure a unit is placed when the tile is selected? (RESOLVED)
How do we ensure the placed unit is deducted from the unit pool? (RESOLVED)

UC_16:
Use Case: Unit constant damage output
Scope: N/A
Level: Sub function
Context: During battle sequence, units will output damage consistently which can then be accessed by opposing units.
During battle sequence, units will output damage consistently but output can't be accessed by opposing units.
Frequency of occurrence: Every time a fight sequence is initiated, for every unit.
Open issues:
How do we determine damage output and frequency? (RESOLVED)
When should damage output be initiated? (RESOLVED)
How is each unit's damage output accessed by opposing units?

UC_17:
Use Case: Unit take damage
Level: Sub function
Context: During battle sequence, units will take damage if they are within range of an opposing unit
During battle sequence, units are unable to take damage when in range of an opposing unit.
Frequency of occurrence: Whenever a unit is within range of an enemy unit.
Open issues:
How do we determine when a unit is within range of another?
How does a unit access an opposing unit's damage output?
How is health deducted from the unit's health points variable?

UC_18:
Use Case: Unit death
Level: Sub function
Context: During battle sequence, when a unit's health subceeds zero it will be eliminated from the battle
During battle sequence, when a unit's health subceeds zero it fails to be eliminated from the battle
Frequency of occurrence: Whenever a unit's health subceeds zero.
Open issues:
How to we eliminate an object from the battle?
How do we give a unit a "dead" attribute?

UC_19:
Use Case: Play game in browser
Level: Summary
Context: Users can access the game through a web browser
Users cannot access the game through a web browser
Frequency of occurrence: Dependent on User
Open issues:

What hosting service do we upload the game to?
How do we make the game as accessible as possible?
What devices will be able to run the game online?

UC_20: (COMPLETE)
Use Case: Move on map
Level: User goal
Context: Users can select an adjacent node and progress through the map
Users are unable to select adjacent nodes.
Frequency of occurrence: Any time the user is in the map scene.
Open issues:
How do we show the current location of the user on the map? (RESOLVED)

UC_21: (COMPLETE)
Use Case: Return to map from grid
Level: User goal
Context: The user can select a button to return to the map scene from the grid scene.
The user is unable to return to the map scene from the grid scene.
Frequency of occurrence: Whenever the user wishes to return to the map from the grid scene.
Open issues:
How to we ensure the player is returned to the node they exited? (RESOLVED)

UC_22: (COMPLETE)
Use Case: Dismiss popup
Level: User goal
Context: Users are able to select a dismissal button which closes the popup
Users are able to select a dismissal button but the popup fails to close
Frequency of occurrence: Whenever a unit's health subceeds zero.
Open issues:
How to we destroy the popup object? (RESOLVED)
How do we change the dismissal button text depending on the trigger scenario? (RESOLVED)
Can we add a function to the dismissal button when required (e.g. a "try again" button for a battle lost popup)? (RESOLVED)

UC_23:
Use Case: Fight sequence initiation
Level: Sub function
Context: Users are able to initiate a fight sequence, spawning the enemy units, upon clicking a fight button
Users are unable to initiate a fight sequence upon clicking a fight button
Frequency of occurrence: When the user is satisfied with their unit placement for the battle encounter.
Open issues:
How do we initialise the fight sequence?
How to we initialise the enemy units? (RESOLVED)
How do we give the enemy units a random formation each time?

UC_24: (COMPLETE)
Use Case: Path-finder/ unit-engagement
Level: Sub function
Context: Units are able to locate the closest opposing unit, lock onto them and plot a course to meet them
Units are able to locate the closest opposing unit but unable to plot a course to meet them.
Frequency of occurrence: Constant during the fight sequence

Open issues:

How do we locate the closest opposing unit? (RESOLVED)

How is the path determined, specifically when there are many units on the board? (RESOLVED)

UC_25:

Use Case: 1v1 fight sequence

Level: Sub function

Context: When engaged units are in range they will deal damage to each other until one of them is eliminated.

When engaged units are in range they fail to deal damage to each other.

Frequency of occurrence: Any time two engaged units are within range.

Open issues:

How do we ensure after a unit has defeated another it searches for a new target?

CRC cards | 5.9.3

GridManager **References – UC_03**	
RESPONSIBILITIES: **Responsible for creating and containing a data structure containing all Tiles of grid and their positions** **Responsible for making this data available publicly to other functions** **Responsible for making units move around on the grid using a pathfinding algorithm**	COLLABORATORS: GridGraph BaseUnit Renderer

GridGraph **References – UC_03**	
RESPONSIBILITIES: **Define the logic of the graph structure underlying the tileset** **Assign a node to each tile** **Contain the pathfinding algorithm**	COLLABORATORS: GridManager

Renderer **References – UC_01, UC_02, UC_03, UC_04**	
RESPONSIBILITIES: **Responsible for drawing all active sprites onto the game window**	COLLABORATORS: BaseUnit Dungeon_Tree

Dungeon_Tree **References - UC_04**	
RESPONSIBILITIES: **Responsible for containing recursive tree structure of other trees and relationships between trees** **Responsible for containing sprite representation of Nodes as well as connections between Nodes** **Responsible for listening for mouse click then altering sprite of node clicked to indicate it is selected** **Responsible for containing the current node position of the party** **Responsible for ensuring directly related distal nodes are the only nodes which are able to be selected on click** **Responsible for providing method to change current position of party to newly selected node**	COLLABORATORS: Renderer

UnitManager **References – US_05_1, US_05_2, US_08_1, US_08_2, UC_03**	
RESPONSIBILITIES: **Responsible for containing and updating unit position** **Responsible for having method available for other functions/classes to change Unit position** **Responsible for containing and updating health, attack speed, and damage** **Responsible for having method available for other functions/classes to "attack" surrounding units**	COLLABORATORS: GridManager Renderer

UnitSeletor **References – UC_03**	
RESPONSIBILITIES: **Responsible for handling drag-and-drop action for all units** **Responsible for passing the drag-and-drop position to other classes/functions** **Responsible for updating the units count when drag-and-drop actions are performed**	COLLABORATORS: GameManager GridManager UnitManager

UIManager **Reference – UC_01, UC_04**	
RESPONSIBILITIES: Responsible for providing pop ups on certain dungeon levels Responsible for providing item counter Responsible for providing asset counter	COLLABORATORS: MapManager

MapManager **Reference – UC_04, US_11_1, US_12_1**	
RESPONSIBILITIES: Responsible to generating the map Responsible for saving the current state of the map Responsible for saving the player's progress	COLLABORATORS: UIManager

HoverHighlight **References – UC_03**	
RESPONSIBILITIES: **Highlight the tile which the cursor is hovering over**	COLLABORATORS: GridManager GridGraph

Design use cases | 5.10.3

UC_01

UC_01_1: **Use Case**: Load game
UC_01_2: **Author**: DD
UC_01_3: **Date**: 15/11/21
UC_01_4: **Purpose**: Initiate new game from within main menu
UC_01_5: **Overview**: Starts when User presses load game button on main menu screen. On button press the player should be prompted to select a game save with one of the options being a new game. The selected game instance should begin, with scene either in dungeon map or in hub depending on program progress.
UC_01_6: **Cross References**: US_03_1, US_03_2
UC_01_7: **Actors**: User
UC_01_8: **Pre-conditions**:
UC_01_Pre-1: Program should be in main menu scene
UC_01_Pre-2: load game button must be "listening" for click event with prepared new game and load game functions actioned on-click
UC_01_9: **Post Conditions**:
UC_01_Post_1: Scene should be changed to a dungeon map with the respective progression selected
UC_01_10: **Main flow of events**:
User Starts the game and enters the main menu scene.
User correctly selects the load game button which prompts the user to select a save.
User selects a save and is transferred to the dungeon scene with map progression corresponding to the respective save selected.
UC_01_11: **Alternative flow of events**:
If user clicks outside of area of button actively listening for input, no action taken.
User selects the load game button and selects a save but is transferred to a new dungeon map scene with the incorrect progression loaded.
In the case of a failure of this process with user unable to begin new game instance, product not viable, if not rectified would necessitate removing main meu scene.
UC_01_12: **Testing**:
In testing this function, ensure that on mouse click, the new game button calls the new game function, and that the scene then changes appropriately to the hub/dungeon depending on the stage of development.

UC_02

UC_02_1: **Use Case**: Exit Game
UC_02_2: **Author**: DD
UC_02_3: **Date**: 15/11/21
UC_02_4: **Purpose**: End game process from within game window
UC_02_5: **Overview**: Starts when User presses quit game button on main menu screen. Button should use built in Unity application.quit() method to end game process.
UC_02_6: **Cross References**: US_01_1, US_01_2
UC_02_7: **Actors**: User
UC_02_8: **Pre-conditions**:
UC_02_Pre-1: Program should be in main menu scene
UC_02_Pre-2: Exit button must be "listening" for click event with prepared quit game function actioned on-click
UC_02_9: **Post Conditions**:
UC_02_Post_1: The program and all related processes must end.

UC_02_10: **Main flow of events**:

User enters the main menu scene and correctly selects the exit game button.

The programme successfully calls the application.quit() method and is terminated.

UC_02_11: **Alternative flow** of events:

If user clicks outside of area of button actively listening for input, no action taken.

In the case of a failure of this process the User can instead use system services to close the game, through the keyboard alt-F4 shortcut or through the windows task manager.

UC_02_ 12: **Testing**: When testing, ensure that on mouse click, the exit game button calls the exit game function.

UC_03

UC_03_1: **Use Case**: Save Game

UC_03_2: **Author**: CP

UC_03_3: **Date**: 15/11/21

UC_03_4: **Purpose**: Save game progress from within game window

UC_03_5: **Overview**: Starts when User presses save game button on main menu screen. This should save the player's progress through the map where the user has the option to save the game under a name.

UC_03_6: **Cross References**: US_02_1, US_02_2

UC_03_7: **Actors**: User

UC_03_8: **Pre-conditions**:

UC_03_Pre-1: Program should be in main menu scene

UC_03_Pre-2: Save button must be "listening" for click event with prepared save game function actioned on-click

UC_03_9: **Post Conditions**:

UC_03_Post_1: The program saves and stores current game progress under the player's name.

UC_03_10: **Main flow**:

When the player is in the main menu and decides they want to save their progress they select the save game button and the player is prompted to enter their name.

The player enters their name and selects okay.

The player sees their name listed under saved games.

UC_02_11: **Alternative flow** of events:

The player selects the save game button but is not prompted to enter their name.

After the player enters their name, they cannot see it listed under saved games.

UC_02_12: **Testing**:

When testing ensure the game is saved with the correct name entered and when the game is re-opened the user is at the same map position with the same number of troops as when they left the game.

UC_04

UC_04_1: **Use Case**: Main Menu

UC_04_2: **Author**: CP

UC_04_3: **Date**: 15/11/21

UC_04_4: **Purpose**: Interface for saving and exiting the game

UC_04_5: **Overview**: The user should be able to select a main menu option from the map scene from which they can choose to load a game, save or exit the game. User should also be met by the main menu upon entry to the game.

UC_04_6: **Cross References**: US_02_1, US_02_2

UC_04_7: **Actors**: User

UC_04_8: **Pre-conditions**:

UC_04_Pre-1: Program should be in map scene or have just been run

UC_04_9: **Post Conditions**:

UC_04_Post_1: The program displays the main menu with all its options.
UC_04_10: **Main flow**:

When the player is in the map scene, they select the main menu option, triggering the main menu screen.
The main menu is displayed with options to load game, save game or exit game
When the player enters the game they are met by the main menu scene displaying options to load game, save game or exit game

UC_04_11: **Alternative flow** of events:

The player selects the main menu button but the main menu scene is not triggered.
After the player enters the game they are not met by the main menu screen and instead transferred directly to the dungeon map scene.

UC_04_12: **Testing**:
When testing ensure the main menu screen is triggered upon selecting the main menu button. Ensure that upon entry to the game the player is met by the main menu screen.

UC_05
UC_05_1: **Use Case**: Popup displayed
UC_05_2: **Author**: CP
UC_05_3: **Date**: 15/11/21
UC_05_4: **Purpose**: Inform player of a game consequence.
UC_05_5: **Overview**: When the user takes an action and a consequence is reached, they should be informed of the resulting consequence.
UC_05_6: **Cross References**: US_07_01, US_07_02, US_15_01, US_15_02.
UC_05_7: **Actors**: User interface.
UC_05_8: **Pre-conditions**:

UC_05_Pre-1: Player has made an action in the game.
UC_05_Pre-2: The action taken has a consequence.

UC_05_9: **Post Conditions**:

UC_04_Post_1: The player understands the consequence of the action taken.

UC_05_10: **Main flow**:

The player makes an action in the game (e.g. enters a treasure node)
The popup is displayed clearly on the scene, with the background still visible
The player understands the consequence of the action taken (e.g. has been awarded 4 units).

UC_05_11: **Alternative flow** of events:

The player makes an action in the game (e.g. enters a treasure node)
The popup is not displayed clearly on the scene.
The popup is displayed clearly on the scene; however, the background is not visible.
The popup is displayed clearly on the scene but the player does not understand the consequence of the action taken.

UC_05_12: **Testing**:
When testing ensure that the popup is correctly aligned in the centre of the screen and the background scene is still visible to the user. Ensure that a situationally required message can be displayed dependent on the action triggering the popup.

UC_06
UC_06_1: **Use Case**: Enter treasure node.
UC_06_2: **Author**: CP
UC_06_3: **Date**: 15/11/21
UC_06_4: **Purpose**: Informs user of their treasure reward units
UC_06_5: **Overview**: When a user selects a treasure node they should be greeted by a popup informing them of how many troops they have gained as a reward.

UC_06_6: **Cross References**: US_07_1, US_07_2, US_09_1, US_09_2, US_10_1, US_10_2,
UC_06_7: **Actors**: User, User interface
UC_06_8: **Pre-conditions**:
UC_06_Pre-1: Programme should be on map scene
UC_06_Pre-2: Treasure node is adjacent to current node.
UC_06_9: **Post Conditions**:
UC_06_Post_1: New occupied node should now be the treasure node
UC_06_Post_2: A dismissible popup should now displayed on the screen informing the user of troop reward
UC_06_Post_3: Unit pool should be updated to reflect additional troops gained.
UC_06_10: **Main flow**:
User sees a treasure node adjacent to current node and selects it.
After the treasure node is selected, a dismissible popup is displayed informing the user of their reward.
After the popup is dismissed the unit pool is updated according to the number of troops awarded.
UC_06_11: **Alternative flow** of events:
User sees a treasure node adjacent to current node, but cannot select it.
After selecting the treasure node, the programme fails to display a popup message.
After selecting a treasure node, a popup is displayed showing the chest reward, but the user is unable to dismiss it.
After the popup is dismissed, the unit pool is not updated to show the extra units awarded.
UC_06_12: **Testing**:
When testing, ensure selection of the treasure node is recognised by the programme and this can be used to trigger a popup. Ensure the popup can successfully be dismissed and that the popup management system updates the unit manager to reflect the additional units gained so that this is displayed in the unit pool.

UC_07
UC_07_1: **Use Case**: Welcome node
UC_07_2: **Author**: CP
UC_07_3: **Date**: 15/11/21
UC_07_4: **Purpose**: Introduce the player to the game.
UC_07_5: **Overview**: Upon entering the game the player should be able to select the starting node where a welcome message is displayed, welcoming the player to the game and informing them of the point they must reach to complete it.
UC_07_6: **Cross References**: US_03_1, US_03_2
UC_07_7: **Actors**: User, User Interface
UC_07_8: **Pre-conditions**:
UC_07_Pre-1: The game has been started and the map scene initiated.
UC_07_Pre-2: No nodes have been clicked and activated yet.
UC_07_9: **Post Conditions**:
UC_07_Post_1: A dismissible popup should be displayed welcoming the player and informing them of the game endpoint (star)
UC_07_10: **Main flow**:
The player enters the game and clicks on the first node on the map.
A dismissible popup is displayed welcoming the player to the game and informing them they must reach the star node at the end of the map.
Popup is dismissed and welcome node is now darkened indicating it is the user's current position on the map.
UC_07_11: **Alternative flow** of events:

The player enters the game and clicks on the first node.
A dismissible popup fails to be displayed to welcome the player or inform them of the objective.
A dismissible popup is created but it is not visible on the screen and therefore cannot be dismissed.
Popup is dismissed but the welcome node is still selectable indicating the user hasn't taken position at the welcome node yet.

UC_07_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct welcome message. Ensure node is darkened and no longer selectable after dismissal.

UC_08

UC_08_1: **Use Case**: New arena node
UC_08_2: **Author**: CP
UC_08_3: **Date**: 15/11/21
UC_08_4: **Purpose**: Transfers the player to the first node of a new arena and informs them they've completed the previous arena.
UC_08_5: **Overview**: When a player have taken a path through an arena and made it to the end, they should be able to select a new arena node which informs them they've completed the previous arena. The player can then choose between different paths through the new arena.
UC_08_6: **Cross References**: US_06_1, US_06_2
UC_08_7: **Actors**: User
UC_08_8: **Pre-conditions**:
UC_08_Pre-1: Player has reached and completed one of the final nodes of their current arena.
UC_08_Pre-2:
UC_08_9: **Post Conditions**:
UC_08_Post_1: Player has the option to select from different starting nodes through the new arena with different paths and encounters.
UC_08_10: **Main flow**:
The player selects the new arena node.
A popup is displayed informing the player that they can now enter a new arena.
The player is transferred to the new arena node and can now select a path through the new arena.
UC_08_11: **Alternative flow** of events:
The player is unable to select the new arena node
The player selects the new arena node but a popup informing the player they can now enter a new arena fails to be displayed.
The player fails to be transferred to the new arena node and therefore cannot select a path through the new arena.
UC_08_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct enter new arena message displayed. Ensure node is darkened and no longer selectable after dismissal.

UC_09

UC_09_1: **Use Case**: Game completion node
UC_09_2: **Author**: CP
UC_09_3: **Date**: 15/11/21
UC_09_4: **Purpose**: Initiates and informs user of completion of the game.
UC_09_5: **Overview**: There should be a final node on the map, whose reaching is the objective of the game. Upon entry, the player should be informed they have completed the game and congratulated.

UC_09_6: **Cross References**: US_01_1, US_01_2
UC_09_7: **Actors**: User
UC_09_8: **Pre-conditions**:
UC_09_Pre-1: The player has successfully completed the last battle node of the game.
UC_09_Pre-2: The player has returned to the map scene.
UC_09_9: **Post Conditions**:
UC_09_Post_1: The game is complete and a message is shown informing the user they have won.
UC_09_10: **Main flow**:
After completing the final battle encounter the player has returned to the map scene.
The player selects the final node on the map.
A dismissible popup is initiated, with a message congratulating the user for winning the game.
UC_09_11: **Alternative flow** of events:
After completing the final battle encounter the player has returned to the map scene.
The player selects the final node on the map.
A dismissible popup fails to be initiated and the player is not informed they have won the game.
UC_09_12: **Testing:**
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct congratulatory message displayed. Ensure node is darkened and no longer selectable after dismissal.

UC_10
UC_10_1: **Use Case**: Update unit pool
UC_10_2: **Author**: CP
UC_10_3: **Date**: 15/11/21
UC_10_4: **Purpose**: Reflects additional troops awarded to the player
UC_10_5: **Overview**: When a player is awarded units from a treasure node, the unit pool should be updated to show the new total number of units.
UC_10_6: **Cross References**: US_06_1, US_06_2
UC_10_7: **Actors**: User
UC_10_8: **Pre-conditions**:
UC_10_Pre-1: Player has selected a treasure node.
UC_10_Pre-2: Unit reward has been determined.
UC_10_9: **Post Conditions**:
UC_10_Post_1: Unit pool should reflect the units awarded on top of the units in possession preceding node selection
UC_10_10: **Main flow**:
User selects a treasure node and is informed of unit reward.
Unit pool is updated to show the sum of the player's previous number of units and the number of units awarded.
UC_10_11: **Alternative flow** of events:
User selects a treasure node and is informed of unit reward.
Unit pool is not updated and remains the same as before node entry.
Unit pool is updated; however the new value is not the sum of the player's previous number of units and the number of units awarded.
UC_10_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct unit award message displayed. Ensure that the unit count and unit pool are updated to reflect the extra troops rewarded.

UC_11

UC_11_1: **Use Case**: Enter battle node
UC_11_2: **Author**: CP
UC_11_3: **Date**: 15/11/21
UC_11_4: **Purpose**: Allow user to enter a battle node to progress through the map.
UC_11_5: **Overview**: When the user is in the map scene and the adjacent node is a battle node, they should be able to select the node and be transferred to the respective level's grid scene.
UC_11_6: **Cross References**: US_04_1, US_04_2, US_08_1, US_08_2, US_09_1, US_09_2, US_10_1, US_10_2
UC_11_7: **Actors**: User
UC_11_8: **Pre-conditions**:
UC_11_Pre-1: Programme is on map scene.
UC_11_Pre-2: These is a battle node adjacent to the current node.
UC_11_9: **Post Conditions**:
UC_11_Post_1: The player is transferred to grid scene
UC_11_10: **Main flow**:
The player is currently at a node with a battle node adjacent.
The player selects the battle node.
Upon selecting the battle node, the player is transferred to the grid scene.
UC_11_11: **Alternative flow** of events:
The player selects the battle node; however, they are unable to select the battle node.
The player selects the battle node; however, they are not transferred to the grid scene.
UC_11_12: Testing:
When testing ensure that the grid scene is loaded successfully with the unit pool of the unit placement scene reflecting the number of units in the unit count of the map scene.

UC_12

UC_12_1: **Use Case**: Enemy unit health scaling
UC_12_2: **Author**: CP
UC_12_3: **Date**: 15/11/21
UC_12_4: **Purpose**: To add difficulty to battle encounters as the player progresses through the game
UC_12_5: **Overview**: As the player progresses through the game, the health of enemy units should increase at each battle encounter.
UC_12_6: **Cross References**: US_09_1, US_09_2, US_12_1, US_12_2
UC_12_7: **Actors**: User
UC_12_8: **Pre-conditions**:
UC_12_Pre-1: Player has completed at least one battle encounter.
UC_12_9: **Post Conditions**:
UC_12_Post_1: The enemy units encountered at the subsequent battle node have more health points than those of the previous node.
UC_12_10: **Main flow**:
The player completes a battle encounter.
The player enters a subsequent battle node and encounters enemy units with one more health point than those of the previous node.
UC_12_11: **Alternative flow** of events:
The player completes a battle encounter.
The player enters a subsequent battle node; however, the enemy units encountered have the same amount of or less health points than those of the previous node.
UC_12_12: **Testing**:
When testing ensure that the enemy units in each subsequent battle node have one extra health point.

UC_13

UC_13_1: **Use Case**: Highlight tile
UC_13_2: **Author**: CP
UC_13_3: **Date**: 15/11/21
UC_13_4: **Purpose**: Highlights the tile which the user's cursor is interacting with at its current position.
UC_13_5: **Overview**: During the grid scene, the player should be able to see which tile their cursor is interacting with according to its position, where the current tile is highlighted
UC_13_6: **Cross References**: US_05_1, US_05_2, US_13_1, US_13_2
UC_13_7: **Actors**: User
UC_13_8: **Pre-conditions**:
UC_13_Pre-1: The programme should be in the grid scene, and the board displayed
UC_13_9: **Post Conditions**:
UC_13_Post_1: When the cursor moves over a tile, the tile is highlighted.
UC_13_10: **Main flow**:
During the grid scene the player moves the cursor over a specific tile.
The tile the cursor is positioned over is highlighted.
UC_13_11: **Alternative flow** of events:
During the grid scene the player moves the cursor over a specific tile.
The tile the cursor is positioned over is highlighted along with one or more other tiles which the cursor is not positioned over.
The tile the cursor is positioned over is not highlighted; however one or more other tiles which the cursor is not positioned over are highlighted.
The tile the cursor is positioned over is not highlighted nor is any other tile on the board.
UC_13_12: **Testing**:
When testing ensure that only the tile the cursor is hovering over is highlighted and this is reflected as the cursor is moved around the grid.

UC_14

UC_14_1: **Use Case**: Select highlighted tile
UC_14_2: **Author**: CP
UC_14_3: **Date**: 15/11/21
UC_14_4: **Purpose**: To enable action to be performed upon selection of a tile.
UC_14_5: **Overview**: During the grid scene, the player should be able to select a highlighted tile which the cursor is positioned over and the programme should recognise this selection enabling an action to be performed.
UC_14_6: **Cross References**: US_08_01, US_08_02, US_05_1, US_05_2, US_13_1, US_13_2
UC_14_7: **Actors**: User
UC_14_8: **Pre-conditions**:
UC_14_Pre-1: The programme should be in the grid scene
UC_14_Pre-2: The user should have the cursor positioned over the highlighted tile which they wish to select.
UC_14_9: **Post Conditions**:
UC_14_Post_1: The tile is selected, turning grey and the action is recognised by the programme.
UC_14_10: **Main flow**:
During the grid scene the player moves the cursor over a specific tile, which is highlighted.
The cursor changes to a hand symbol letting the player know its clickable,
The player selects the tile, which turns grey, meaning the programme recognises the action.
UC_14_11: **Alternative flow** of events:
During the grid scene the player moves the cursor over a specific tile, which is highlighted

The cursor remains unchanged, so they do not know its clickable.
The cursor changes to a hand symbol, letting the player know its clickable, the player selects the tile; However, the tile does not change colour, meaning the programme does not recognise the action.

UC_14_12: **Testing**:
When testing ensure that it is only the highlighted tile which is selected and the programme recognises the action.

UC_15

UC_15_1: **Use Case**: Place units on grid
UC_15_2: **Author**: DD
UC_15_3: **Date**: 15/11/21
UC_15_4: **Purpose**: Allow user to place units on grid before battle
UC_15_5: **Overview**: Starts during placement phase of auto-battler encounter. Player should be able to select from persistent pool of units, visible on the lower edge of the screen in a relevant UI object, and these can then be placed in empty hexes on the board on the lower half of the game board.
UC_15_6: **Cross References**: US_08_01, US_08_02, US_11_01, US_11_02, US_14_01, US_14_01, US_15_01, US_15_02
UC_15_7: **Actors**: User
UC_15_8: **Pre-conditions**:
UC_15_Pre-1: Program should be within encounter scene
UC_15_Pre-2: Current state of encounter should be planning
UC_15_Pre-3: User must have units available to place
UC_15_Pre-4: At least one square on board should not be full
UC_15_Pre-5: Current units placed must be less than maximum placeable units
UC_15_9: **Post Conditions**:
UC_15_Post_1: Selected unit should be placed on the game board
UC_15_Post_2: The Square on the board the unit now occupies should no longer be empty
UC_15_Post_3: One copy of the selected unit should be removed from the unit pool temporarily
UC_15_10: **Main flow of events**:
Player Enters the battle node initiating the placement phase of the battle encounter.
Player select a unit from their unit pool.
Unit is placed on one of the empty hexes on the board.
UC_15_11: **Alternative flow** of events:
User tries to place unit on non-empty square, explanatory error message is shown
User tries to place unit from empty pool of units, explanatory error message is shown
UC_15_12: Testing:
When testing, ensure unit clicked becomes selected. Also ensure when subsequently clicking on tile that a new unit of the selected type is instantiated, with a position equal to the selected tile.

UC_16

UC_16_1: **Use Case**: Unit constant damage output
UC_16_2: **Author**: CP
UC_16_3: **Date**: 15/11/21
UC_16_4: **Purpose**: Allows units to deal damage to opposing units.
UC_16_5: **Overview**: Units consistently output damage with amount and rate determined by the unit type.
UC_16_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2
UC_16_7: **Actors**: User
UC_16_8: **Pre-conditions**:

UC_16_Pre-1: Units are placed on the board
UC_16_Pre-2: Fight sequence is initiated

UC_16_9: **Post Conditions**:
UC_16_Post_1: Damage is consistently output while the unit is not dead.

UC_16_10: **Main flow**:
The fight sequence is initiated and all units on the board begin outputting damage.
Damage is consistently output until the unit is dead.

UC_16_11: **Alternative flow** of events:
The fight sequence is initiated; however, some or all the units do not begin outputting damage.
Damage is consistently output by each unit, even after the unit is dead.

UC_16_12: **Testing**:
When testing ensure the unit damage output is accessible to other units so that they can use it to take damage.

UC_17

UC_17_1: **Use Case**: Unit take damage
UC_17_2: **Author**: CP
UC_17_3: **Date**: 15/11/21
UC_17_4: **Purpose**: Allows units to take damage from other units
UC_17_5: **Overview**: When a unit is in range of an opposing unit, it should be dealt damage corresponding to the opposing unit's damage output.
UC_17_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2
UC_17_7: **Actors**: User
UC_17_8: **Pre-conditions**:
UC_17_Pre-1: The unit has been engaged by an opposing unit.
UC_17_Pre-2: The unit is in range of the opposing unit.

UC_17_9: **Post Conditions**:
UC_17_Post_1: The unit health is consistently reduced according to the opposing unit's damage output.

UC_17_10: **Main flow**:
The unit comes into range of an opposing unit which has engaged it.
The opposing unit begins consistently dealing damage to the unit while it is in range.
The health variable of the unit is consistently reduced according to the opposing unit's damage output.

UC_17_11: **Alternative flow** of events:
The unit comes into range of the opposing unit which has engaged it.
The opposing unit fails to deal damage to the unit
The opposing unit
The health attribute of a unit engaged in a 1v1 battle subceeds zero.
The dead Boolean of the unit object fails to be set to true.

UC_17_12: **Testing**:
When testing ensure the damage output of the unit dealing the damage is accessible to the one taking it. Ensure a corresponding amount of health points are deducted from the unit's health variable.

UC_18

UC_18_1: **Use Case**: Unit death
UC_18_2: **Author**: CP
UC_18_3: **Date**: 15/11/21
UC_18_4: **Purpose**: Criteria for the elimination of a unit from a battle.

UC_18_5: **Overview**: When the health of a unit has reached or subceeded zero, the unit is given the dead attribute, which can then be used to disengage the opposing unit and identify unit objects which need to be destroyed.
UC_18_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2
UC_18_7: **Actors**: User
UC_18_8: **Pre-conditions**:
UC_18_Pre-1: The unit in question has been engaged in a 1v1 battle.
UC_18_Pre-2: The health of the unit is less than or equal to zero.
UC_18_9: **Post Conditions**:
UC_17_Post_1: The unit is given the dead attribute which is a Boolean set to true.
UC_18_10: **Main flow**:
The health attribute of a unit engaged in a 1v1 battle subceeds zero.
The "dead" Boolean of the unit object is set to true.
The "dead" unit object is destroyed, eliminating it from the battle
UC_18_11: **Alternative flow** of events:
The health attribute of a unit engaged in a 1v1 battle subceeds zero.
The dead Boolean of the unit object fails to be set to true.
The dead Boolean of the object is set to true but it is not destroyed, failing to eliminate it from the game.
UC_18_12: **Testing**:
When testing ensure that the unit object is successfully destroyed upon its dead Boolean being set to true

UC_19
UC_19_1: **Use Case**: Play game in browser
UC_19_2: **Author**: CP
UC_19_3: **Date**: 15/11/21
UC_19_4: **Purpose**: Allow players to play game in a browser across multiple devices.
UC_19_5: **Overview**: The game should be playable in a browser.
UC_19_6: **Cross References**: US_16_1, US_16_2
UC_19_7: **Actors**: User
UC_19_8: **Pre-conditions**:
UC_19_Pre-1: The game is fully functional.
UC_19_9: **Post Conditions**:
UC_19_Post_1: The game is playable on a web browser.
UC_19_10: **Main flow**:
The game is uploaded to unity free hosting service and embedded in a web page.
The game is playable online
UC_19_11: **Alternative flow** of events:
The game fails to be uploaded to unity free hosting service
UC_19_12: **Testing**:
When testing ensure that the game is playable across a range of devices on a range of web browsers.

UC_20
UC_20_1: **Use Case**: Move on Map
UC_20_2: **Author**: DD
UC_20_3: **Date**: 22/11/21
UC_20_4: **Purpose**: Allow user to change current location to adjacent node to progress through map
UC_20_5: **Overview**: When user is in the map scene, they should be able to select any of the adjacent nodes on the map, and their current position should then be changed to this node. This should also be visually indicated on the scene

UC_20_6: **Cross References**: US_05_1, US_05_2, US_06_1, US_06_2, US_07_1, US_07_2, US_13_1, US_13_2, US_19_1, US_19_2
UC_20_7: **Actors**: User
UC_20_8: **Pre-conditions**:
UC_20_Pre-1: Program should be in map scene
UC_20_Pre-2: Current party position should not be at end node
UC_20_9: **Post Conditions**:
UC_20_Post_1: Current party position should change to selected node
UC_20_Post_2: The visual indicator for party position should change to selected node
UC_20_Post_3: Event for new node position should be triggered
UC_20_10: **Main flow**:
UC_20_11: **Alternative flow** of events:
If a user selects a node proximal in the tree, the party position should remain at the current node
If a user selects the currently occupied node, the party position should remain at the current node
UC_20_12: **Testing**: When testing, ensure only nodes distal and adjacent to active nodes are selectable. Also ensure that when selectable nodes are clicked, the party position then changes to this current node.

UC_21
UC_21_1: **Use Case**: Dismiss Popup
UC_21_2: **Author**: CP
UC_21_3: **Date**: 22/11/21
UC_21_4: **Purpose**: Allow user to Dismiss a popup from the screen.
UC_21_5: **Overview**: When a popup is displayed, the user should be able to easily select a dismiss button to close it.
UC_21_6: **Cross References**: US_15_1, US_15_2, US_20_1, US_20_2
UC_21_7: **Actors**: User
UC_21_8: **Pre-conditions**:
UC_04_Pre-1: A dismissible popup is currently in display on the current scene.
UC_21_9: **Post Conditions**:
UC_04_Post_1: Popup is dismissed from the scene.
UC_20_10: **Main flow**:
The user successfully selects the dismiss button.
The popup is closed.
UC_21_11: **Alternative flow** of events:
The user fails to select the dismiss button.
The user successfully selects the dismiss button; however, the popup isn't closed
UC_21_12: **Testing**:
When testing ensure that selecting the dismiss option of the popup successfully destroys the popup object.

UC_22
UC_22_1: **Use Case**: Fight sequence initiation
UC_22_2: **Author**: CP
UC_22_3: **Date**: 29/11/21
UC_22_4: **Purpose**: Positions enemy units in a random formation, giving variety to battle gameplay, and initiates fight sequence.
UC_22_5: **Overview**: When a user selects the fight button, the respective enemy units for that level should be positioned in a random formation and the fight sequence should be initiated.

UC_22_6: **Cross References**: US_08_1, US_08_2, US_21_1: , US_21_1:2, US_22_1, US_22_2
UC_22_7: **Actors**: User, units
UC_22_8: **Pre-conditions**:
UC_22_Pre-1: Programme is in grid scene with all player units placed
UC_22_Pre-2: fight button has been selected.
UC_22_9: **Post Conditions**:
UC_22_Post_1: All enemy units are placed on the board in a random formation.
UC_22_Post_3: Fight sequence is initiated.
UC_22_10: **Main flow**:
After the player is satisfied with their unit placement, they select the fight button on the grid scene.
The enemy units for the respective level are placed in a random formation on the board.
The fight sequence is initiated and the battle begins.
Battle continues until one side's units have all been eliminated.
UC_22_11: **Alternative flow** of events:
After the player is satisfied with their unit placement, they select the fight button on the grid scene.
The enemy units for the respective level are placed in a non-random formation on the board, which is the same every time.
The enemy units are placed on the board in a random formation; however, the fight sequence fails to be initiated and the battle does not begin.
The fight sequence is successfully initiated however one or more units fail to engage an opponent.
All units engage an opponent however one or more units fail to lock onto a new opponent after their opponent has been destroyed. This could lead to units stopping fight before one side has been eliminated.
UC_05_12: **Testing**:
When testing ensure all units are activated and engage an opponent. Ensure that units keep fighting unit all the units of one team are eliminated. Ensure units only deal damage to opposing units. Ensure after destroying an opponent, units are successfully able to lock onto a new opponent.

UC_23
UC_23_1: **Use Case**: Path-finder/ unit engagement
UC_23_2: **Author**: CP
UC_23_3: **Date**: 29/11/21
UC_23_4: **Purpose**: Engages troops in combat
UC_23_5: **Overview**: When a battle is initiated, both user and enemy units will search for the closest opposing unit and lock onto them. The units will then move towards each other until they are within attacking range at which point they will stop and begin fighting. If a unit successfully wins, it then locks onto a new opponent and the process is repeated.
UC_23_6: **Cross References**: US_08_1, US_08_2, US_21_1: , US_21_1:2, US_22_1, US_22_2
UC_23_7: **Actors**: Units
UC_23_8: **Pre-conditions**:
UC_23_Pre-1: User has placed all units on the board and initiated the fight sequence.
UC_23_Pre-2: Enemy units have successfully been deployed on the board upon initiation of fight sequence.
UC_23_9: **Post Conditions**:
UC_23_Post_1: All units on the board have successfully locked onto an opposing unit
UC_23_Post_2: All units have managed to reach and engage their opposing unit in battle.
UC_23_10: **Main flow**:

Fight sequence is initiated, all units on the board successfully lock onto an opposing unit and begin moving towards them.

All units successfully reach their respective opposing units and are able to engage them in battle.

The unit which wins the battle then locks onto a new opponent and the process is repeated.

UC_23_11: **Alternative flow** of events:

Fight sequence is initiated; however, some or all units are unable to successfully lock onto an opposing unit.

All units successfully lock onto an opposing unit; however, some or all units are unable to successfully reach their respective opposing unit.

All units successfully reach their respective opposing unit; however, some or all units fail to engage their respective opposing unit in battle.

All units successfully engage their opposing unit in battle; however, if they win their battle, they fail to engage a new unit and become inactive on the board.

UC_23_12: **Testing**:

When testing ensure each unit locks onto its closest opposing unit. Ensure after

UC_24

UC_24_1: **Use Case**: 1v1 fighting sequence

UC_24_2: **Author**: CP

UC_24_3: **Date**: 29/11/21

UC_24_4: **Purpose**: Process for engaged units to fight one another.

UC_24_5: **Overview**: When locked on units reach each other and are in range, they begin to deal damage to each other, according to their attributes, until one defeats the other.

UC_24_6: **Cross References**: US_08_1, US_08_2, US_21_1: , US_21_1:2, US_22_1, US_22_2

UC_24_7: **Actors**: User

UC_24_8: **Pre-conditions**:

UC_24_Pre-1: All units have been placed and fight sequence has been initiated.

UC_24_Pre-2: Pathfinding has successfully brought two opposing units together so that they're within attack range of each other.

UC_24_9: **Post Conditions**:

UC_24_Post_1: Units have successfully engaged in battle and one has been destroyed.

UC_24_10: **Main flow**:

Two engaged units are within range at which point they begin dealing damage to each other.

Each unit deals a certain amount of damage to the other at a constant rate. These variables are determined by the unit type.

Once the health of one of the units is less than or equal to zero, the unit object is destroyed and the opposing unit has won the battle.

UC_24_11: **Alternative flow** of events:

Two engaged units are within range of each other; however, on or both the units fail to deal damage to the other.

Once the health of one of the units is less than or equal to zero, the unit object fails to be destroyed. The units stay engaged in battle and keep dealing damage to each other.

UC_24_12: **Testing**:

When testing ensure that both units, when in range, deal damage to each other. Ensure that as soon as one unit's health points subceeds zero, it is destroyed. Ensure the correct damage output and frequencies of units are received by the opposing unit.

UC_25

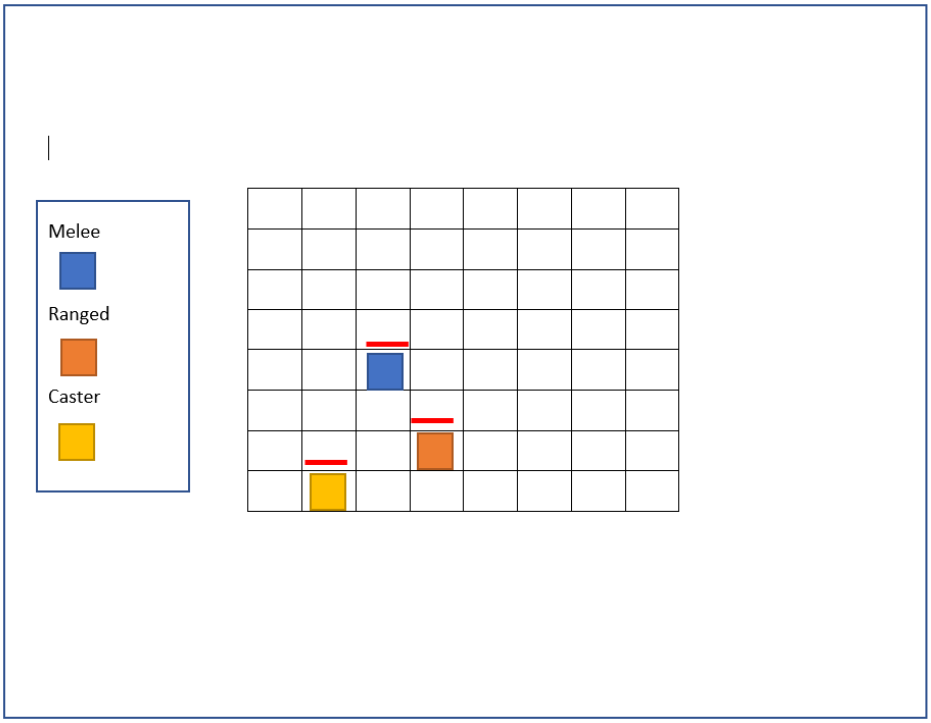
UC_07_1: **Use Case**: Return to map button.

UC_25_2: **Author**: CP

UC_25_3: **Date**: 29/11/21
UC_25_4: **Purpose**: Allows the player to exit the battle sequence and go back to the grid scene.
UC_25_5: **Overview**: When the player is in the grid scene, they should be able to return to the map by selecting a return button at any point the grid scene sequences.
UC_25_6: **Cross References**: US_04_1, US_04_2, US_10_1, US_10_2
UC_25_7: **Actors**: User, grid manager
UC_25_8: **Pre-conditions**:
UC_04_Pre-1: Player is currently in the grid scene
UC_25_9: **Post Conditions**:
UC_04_Post_1: Player is returned to the map scene positioned at the node they exited.
UC_25_10: **Main flow**:
Player is in the grid scene and selects the return to map button.
Player is returned to the map and they are positioned at the node they exited.
UC_25_11: **Alternative flow** of events:
Player is in grid scene and fails to select the return to map button.
Player selects the return to map button; however is not successfully transferred to the map scene.
The player returns to the map without completing the battle encounter; however, they are no longer positioned at the node they exited.
UC_25_12: **Testing**:
When testing ensure that upon selecting the return to map button the user is transferred to the map scene and positioned at the node they exited if they did not complete the battle and the node they entered if they did.

User interface design | 5.11.2

Auto-Battler UI



Elements

Unit Sprites - References **US_05_1, US_05_2, US_08_1, US_08_2, UC_03, Units, Renderer**

8x8 grid comprised of Tiles **US_08_01, US_08_02, US_11_01, US_11_02, US_14_01, US_14_01, US_15_01, US_15_02, UC_03, Tiles, Renderer, Grid Manager**

Unit selector on left side of window **US_5_1, US_5_2, US_8_1, US_8_2, US_13_1, US_13_2, US_15_1, US_15_2, UC_03, Renderer, Units**

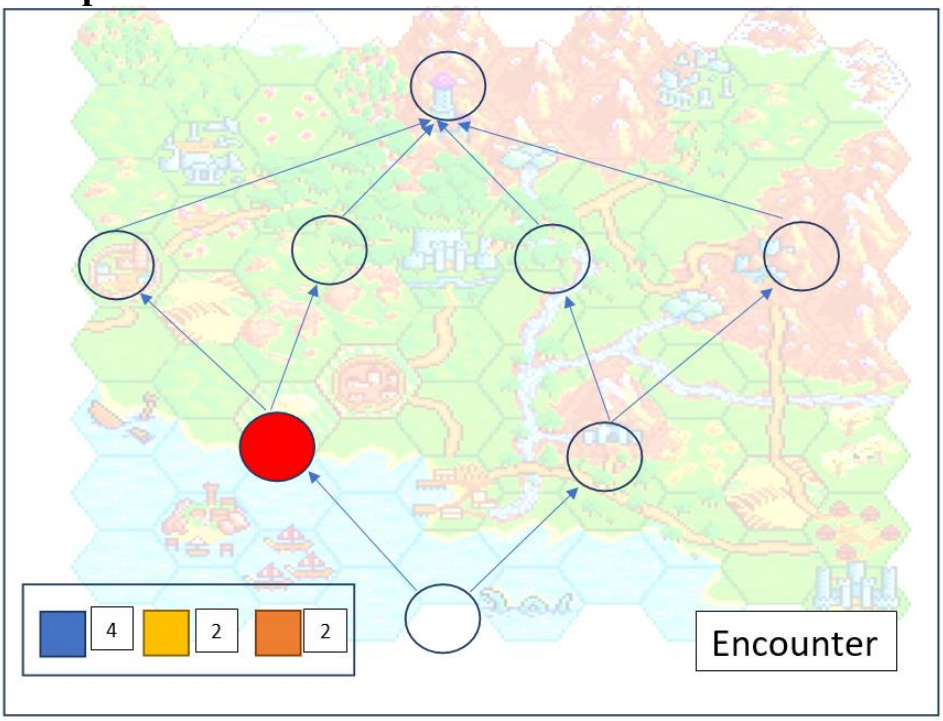
Should indicate types available, and how many of each type remaining

Appropriate methods implemented to enable clicking on unit type to select, and then clicking on Tile to place

Health bars above unit sprites **US_15_1, US_15_2**

Method should resize bar depending on current health

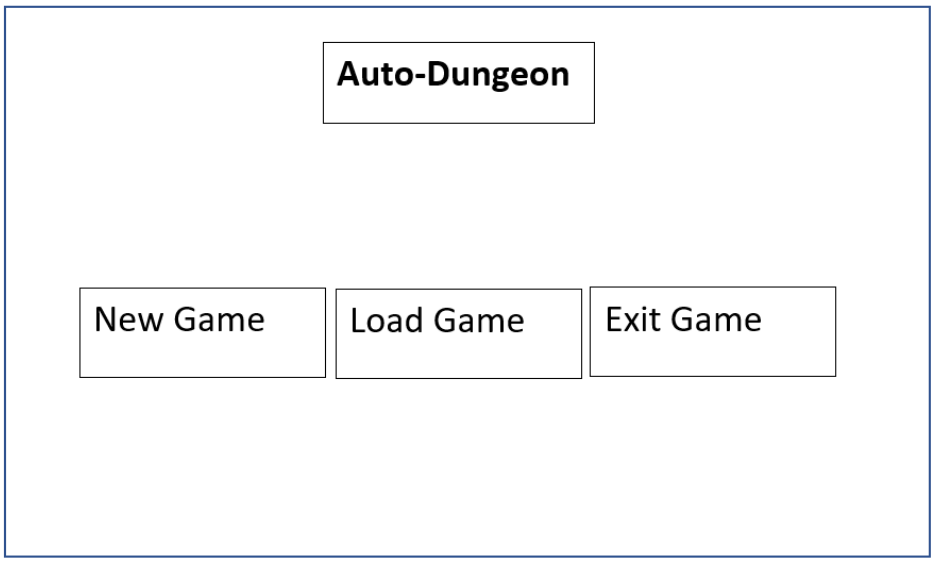
Dungeon Map UI



Elements

Tree Nodes **US_06_1, US_06_2, US_07_1, US_07_2, UC_04, Dungeon_Tree**
Should have method to change colour to indicate node currently occupied by party
Should have method to determine which encounter and then change scene
Unit's remaining indicator **US_07_1, US_07_2, US_09_1, US_09_2, US_11_1, US_11_2, US_14_1, US_14_2, US_15_1, US_15_2, Dungeon_Tree, Units**
Should visually show the number and composition of remaining units to aid decision making
Encounter button to enter encounter at current Node
Should have visual "overworld" map to help in telling underlying game story
Encounter difficulty should increase as game progresses

Main Menu UI



Elements

New Game Button **US_03_1, US_03_2, UC_01**
Should have on-click method ready to start new game method
Load Game Button **US_02_1, US_02_2**
Should have on-click method ready to start load game method
Method should load available save game, display error message if no available save game
Exit Game Button **US_01_1, US_01_2, UC_02**
Should close game window and end all associated processes

Sprint 6: 04/12/21-10/12/21

The focus of sprint 6 was further developing the combat gameplay.

Overview

What's supposed to happen this sprint

No.	Expected Start	Expected Finish	Objective
1	06/12/21	10/12/21	Add global light and point lights to increase the atmosphere of the map
2	06/12/21	10/12/21	Add range units
3	06/12/21	10/12/21	Add sprites for enemy and player units, including range and melee units. Potentially have animations for the sprites
4	06/12/21	13/12/21	Make game changes to target selection. Finds a random target in range and deals constant damage
5	06/12/21	13/12/21	Added a menu
6	06/12/21	13/12/21	Health bar for both enemy and player units
7	06/12/21	13/12/21	Added win and loss popups within the unit grid

Reviews

What actually happened

No.	Started	Finished	Task	Doc
1	06/12/21	06/12/21	Daily scrum	6.1.1
2	08/12/21	08/12/21	Daily scrum	6.2.1
	08/12/21	08/12/21	Customer meeting	6.3.1
3	10/12/21	10/12/21	Daily scrum, sprint review, sprint retrospective, sprint planning	6.4.1
5	Throughout		Backlog updated	6.5.1
6	Throughout		Exception handling	6.6.1
7	Throughout		User evaluation	6.7.1
8	Throughout		User stories with acceptance tests	6.8.5
9	Throughout		Requirements use cases	6.9.4
10	Throughout		CRC cards	6.10.4
11	Throughout		Design use cases	6.11.4
12	Throughout		User interface design	6.12.3

Daily scrum | 6.1.1

06/12/21, 13:00-13:15, Microsoft Teams

Attendance

All present except CP
Meeting led by IL
Minutes taken by IL

Minutes

1. Within the last 48 hours:

    a. DD and TW improvements on map
    b. CP focused on implementing grid UI elements including finalizing unit selection element.
    c. DD added additional information pop-ups to the map scene
    d. QZ got the units to move around the board and added a find target mechanism. Will add it to a new branch

2. Issues that arose within the last 48 hours:

    a. GitHub integration merging errors between 3-4 branches
    b. Managing workloads with current coursework's

3. Within the next 48 hours:

    a. Whole team working on project documentation
    b. CP and DD adding range units
    c. Tweaks with drag and drop
    d. LG, TW, IL to look at backlog for what to do next

4. Foreseeable issues within the next 48 hours are:

    a. LW might not be able to come in for customer meeting
    b. Clearing up documentation and adding it within a single file

Daily scrum | 6.2.1

08/12/21, 14:15-14:30, CB 5.13

Attendance

All present except IL
Meeting led by NO
Minutes taken by NO

Minutes

1. Within the last 48 hours:

    a. Added range units
    b. DD and TW met and added improvement to the map
    c. Added extra nodes to the map
    d. Health bar for enemy and player units
    e. Bugs within battle scenario

2. Issues that arose within the last 48 hours:

    a. Switching roles to much

3. Within the next 48 hours:

    a. Possible animations of units
    b. Work on documentation

4. A lot of coursework deadlines due

Customer meeting | 6.3.1

08/11/21, 14:45-15:00, CB 5.13

Attendance

All present
Recording taken by TW
Minutes taken by IL

Minutes

1. Fully functional map and fight scenes
2. Simple map design with the ability of the player to zoom out giving an overview of the map, allowing them to visualise better during gameplay
3. Added lighting to nodes
4. Updated grid with placement of units
5. Two types of units melee and range (maybe too strong at the moment)
6. Units are able to deal 1 damage constantly throughout the fight, will change these numbers depending on unit type
7. Units share the same sprite place holder, this will be changed
8. GitHub merging can be complicated depending on how many branches have been worked on
9. Should never commit a merge unless everything in a specific branch is working
10. Future commits should be fully functional
11. Group swapping roles too often causing reduced workflow
12. Viable product – 2 different units, one map that can be extended, better to have few things done well than a lot of features half working

Analysis

1. Fully functional game with different units with different attributes, nice simple map that can possibly be extended
2. Hard and easy enemy encounters shown with different icons, one with skull and crossbones and the other with skull
3. GitHub commits should be fully functional as causing meetings to take longer than the allocated time
4. Suggestion that we slowed down the switching role procedure as it was slowing our progress and causing reduced flow

End of sprint meeting | 6.4.1

10/12/21, 13:15-14:30, NH 2.17a

Attendance

Everyone present
Minutes taken by IL
Meeting led by IL

Daily scrum

1. Within the last 48 hours:

    a. TW extra work on map added lighting with global and point light to increase atmosphere and added lighting next node
    b. CP DD added new sprites for enemy and player units, made changes to target selection. Added damage and range units with a range of 5 tiles
    c. NO looked at adding a menu

2. Issues that arose within the last 48 hours:

    a. IL not able to come in diabetic issues
    b. Getting slowed down as switching roles and switching team members on different parts of the product

Sprint review

1. TW – extra work on the map, added lighting (global light and point lights to make it more atmospheric lighting up adjacent nodes

2. TW wanting to – reset camera on different scenes locking on to certain scenes, work out bugs with the nodes turning grey. Adding difficulty on the paths

3. CP and DD - Added new sprites for enemy and player units, made changes to target selection, finds random target and in range and does constant damage, added ranged units with range 5 tiles. Sprites can have potential animations.

4. NO – looked into getting a menu

5. LW – health bar to show up

6. QZ- Added win and loss popups within the grid.

7. Discuss sprint backlog. Completed:

    a. Basic unit combat
    b. Fight button
    c. Appropriate assets and sprites
    d. Battler asset and UI
    e. End screen

Sprint retrospective

1. Issues - IL had diabetic issues

2. Progress slowed down as strikes too place disputing the flow of the scrum, and the amount of coursework.

3. Plan for improvements - keeping things simple with in code to offer a deliverable game

4. Problems with merging with git. We should merge more often than once a week.

5. Meeting on Wednesday TA's and Julian, getting slowed down as switching roles and switching parts of the product, getting redundancies etc. Maybe stick to the same roles for the next two weeks.

Sprint planning

1. Getting the game merged together and working correctly

2. Fixing bugs within map, and combat.

3. Tim merging his branch with the win states

4. Naomi and Qinyuang's branches are merged and work together, need to merge Callum, Drew and Tim's sections into the main branch.

5. Getting game fully merged together

6. Fixing bugs within map and combat

7. Merge TW branch with win states

8. NO QZ branches need to be merged with CP DD branch and TW branch

Task assignments

	A	AI	AJ	AK	AL	AM	AN	AO
1	**Date**	**11/12/21**	**12/12/21**	**13/12/21**	**14/12/21**	**15/12/21**	**16/12/21**	**17/12/21**
2	**Sprint**				**Sprint 7**			
3	**Qinyuan Zhuang**	Scrum master						
4	**Lawrence Wong**	CRC cards						
5	**Callum Pitceathly**	Use cases						
6	**Drew Dundee**	User interface design						
7	**Naomi Okiddy**	Maintenance guide						
8	**Lucy Goodchild**	User stories						
9	**Ieuan Lambert**	User manual and sprint 6 documentation						
10	**Tim Win**	Collating everything						
11	**Everyone**							
12	**Daily scrum**			13:00 MS Teams		14:15 CB 5.13		
13	**Daily scrum, sprint review, sprint retrospective, and sprint planning**							13:15 MS Teams
14	**Customer meeting**					10:30 CB 5.13		
15								

Backlog | 6.5.1

Complete backlog tasks

1. Map improvements such as extra nodes, atmosphere and icons
2. End screen
3. Hp bars for units
4. Fight button to start battle
5. Assets and sprites for units
6. Added range units
7. Battler asset and UI

New backlog tasks

1. Merge game into a usable web application
2. Fixing bugs within map and combat screen
3. Fixing path finding
4. Adding music into different environments
5. Map randomiser

Current backlog

1. Main menu
2. Menu music
3. Recording feature/kill feed
4. Sound effects
5. Change the tile set based on difficulty

Exception handling | 6.6.1

Integration issues

1. Github integration merging errors having to merge Naomi and Qinyuang's branch with Callum's, Drew's and Tim's branches and then merging them to main.

2. Visual bugs within the map solved by setting layers.

3. Errors with back arrow setting current node to complete fixed by changing state of current node

4. Ieuan Lambert diabetic issues rest of team present on Wednesday

User evaluation | 6.7.1

As part of our development process, we had planned from an early stage to seek potential end user input in order to guide our iterative process.

Multiple delays were unfortunately imposed however, as we had not accounted for the necessity of seeking University level ethical approval for this part of the project.
We were thankfully provided with an ethical approval for this work (as a result of our supervisor kindly submitting the pre-requisite ethical implications forms to the relevant departments.)

Unfortunately, this delay, as well as the pace of development and our teams multiple parallel priorities, meant that we were unable to get the surveys to users prior to delivering the product and documentation. After discussion with our primary customer, and review of the delivered product, we felt that we more than met the customers requirements regardless, but felt that we could deliver further value by creating and document a process for seeking user feedback for potential future use.

As part of this documentation, we have then included our adapted Guess-18 survey (a survey tool validated for the purposes of user feedback for games Validation of the GUESS-18: A Short Version of the Game User Experience Satisfaction Scale (GUESS) – Google Research), our participant information leaflet, and have provided a link to the completed ethical implications form (see below) to guide completion of similar work.

https://drive.google.com/file/d/1cS2Cer5t2XK7geetrEvq8n8wfT_udr8K/view?usp=sharing

PARTICIPANT INFORMATION SHEET

Auto-Dungeon - CM50109

Name of Researcher: Andrew Dundee
Contact details of Researcher: --\----\---\---

Name of Supervisor: Dr Julian Padget
Contact details of Supervisor: --\----\---\---

1. What is the purpose of the survey:

The purpose of this survey is to give our team actionable player feedback for our game, to help us improve and meet the requirements of our players.

2. Why have I been selected to take part? [or Who can be a participant?]

We would love for anyone with the time and the ability to take part. All feedback is welcomed and appreciated.

3. Do I have to take part?

It is completely up to you to decide if you would like to participate. Before you decide to take part we will describe the project and go through this information sheet with you. If you agree to take part, we will then ask you to sign a consent form. However if at any time you decide you no longer wish to take part in this project you are free to withdraw, without giving a reason.

4. What will I be asked to do?

-

You will be provided with a URL link to the website currently hosting our game. A link to the user manual will then be available in the game description to guide you through the process. We would ask that you spend at least 30 minutes playing the game. Afterwards, we would appreciate if you could complete the provided survey, to provide feedback and help us make the best version of our game.

5. What are the exclusion criteria?
(are there reasons why I should not take part)?

Anyone with a laptop/pc capable of opening a modern browser will be able to take part in this survey.

6. What are the possible benefits of taking part?

There are no direct benefits of taking part in the project. However, the information that you and other participants provide in this project will help us to make the best and most enjoyable version of our game we can.

7. What are the possible disadvantages and risks of taking part?

There are no disadvantages to you taking part in the project. If the questionnaire asks a question that you do not want to answer for any reason, you can choose not to answer.

8. Will my participation involve any discomfort or embarrassment?

We do not expect you to feel any discomfort or embarrassment if you take part in the project. If for any reason, as a result of our product or survey you do feel uncomfortable or upset at any time, you can choose to withdraw from the process without further explanation.

9. Who will have access to the information that I provide?

Only the research team will have access to information that you provide. All records will be treated as confidential.

10. What will happen to the data collected and results of the project?

No personal, identifiable data will be collected or stored at any stage of this process. No identifiable information should be provided on the returned survey.

After the project has finished, we will also provide participants with a summary of the project results if they would like that. This summary will not include any identifiable information and will show the overall findings of the project.

11. Who has reviewed the project?

This project has been reviewed through the EIRA1 process within the Department of Computer Science.

12. How can I withdraw from the project?

You may withdraw from participating in the process at any stage without the need for explanation or further contact. If you have returned your survey results already and wish for them to be withdrawn, please contact an identified researcher within two weeks of your participation, although you may need to provide some information pertaining to your survey responses to enable us to identify your survey results.

13. University of Bath privacy notice

The University of Bath privacy notice can be found here: https://www.bath.ac.uk/corporate-information/university-of-bath-privacy-notice-for-research-participants/.

14. What happens if there is a problem?

If you have a concern about any aspect of the project you should ask to speak to the researchers who will do their best to answer any questions.

15. If I require further information who should I contact and how?

Thank you for expressing an interest in participating in this project. Please do not hesitate to get in touch with us if you would like some more information.

Name of Researcher: Andrew Dundee
Contact details of Researcher: --\----\---\---

Name of Supervisor: Dr Julian Padget
Contact details of Supervisor: --\----\---\---

User stories with acceptance tests | 6.8.5

US_01_1: Description: "As a user, I'd like a way to exit the program when I'm finished, from within the game window"
US_01_2: Acceptance Criteria: Accepted when quit game button implemented in main menu

US_02_1: Description: "As a user, I'd like to be able to continue a persistent game across multiple sessions"
US_02_2: Acceptance Criteria: Accepted when save game system implemented

US_03_1: Description: "As a user, I need a way to initialise a new game from within the game window"
US_03_2: Acceptance Criteria: Accepted when new game button implemented in main menu

US_04_1: Description: "As a user, I'd like multiple varied dungeons to add replayability"
US_04_2: Acceptance Criteria: Accepted when more than one dungeon implemented

US_05_1: Description: "As a user, I'd like to be able to make decisions about the units within my team, to add challenge and reward planning and critical thinking"
US_05_2: Acceptance Criteria: Accepted when able to select from multiple unit types during planning phase of combat

US_06_1: Description: "As a user, I'd like to have additional choice and variety in my gameplay, to separate this game from other auto-battlers"
US_06_2: Acceptance Criteria: Accepted when implement both persistent unit pools on main map, as well as being able to choose from branching paths

US_07_1: Description: "As a user, I'd like to make strategic decisions, with sufficient information to differentiate good and bad decisions"
US_07_2: Acceptance Criteria: Accepted when able to clearly see indication of reward and difficulty on each dungeon path.

US_08_1: Description: "As a user, I want frequent combat encounters as part of the core game"
US_08_2: Acceptance Criteria: Accepted when nodes in dungeon graph implemented, with encounters possible at each, including fully implemented auto-battling system

US_09_1: Description: "As a user, I enjoy progressive challenge to test my ability"
US_09_2: Acceptance Criteria: Accepted when persistent unit pool implemented, as well as progressively harder encounters culminating in "boss" encounter

US_10_1: Description: "As a user, I enjoy uncertainty, surprise, and variety in my gameplay"
US_10_2: Acceptance Criteria: Accepted when multiple encounter types implemented, with hidden possibility of each at given node

US_11_1: Description: "As a user, I'd like my strategic decisions to have visible, long-term consequences."
US_11_2: Acceptance Criteria: Accepted when persistent unit pool implemented, alongside choice of multiple branching paths

US_12_1: Description: "As a user, I enjoy dynamic events and decision making, with lots of variety"

US_12_2: Acceptance Criteria: Accepted when multiple encounter types implemented, with hidden possibility of each and given node

US_13_1: Description: “As a user, I enjoy decision making with restrictions and uncertainty to add challenge”
US_13_2: Acceptance Criteria: Accepted when planning phase implemented to combat with time restriction, alongside choice of branching paths

US_14_1: Description: “As a user, I’d like clear success and failure states to enable planning and goal setting”
US_14_2: Acceptance Criteria: Accepted when persistent pool of units implemented, with game-over occurring when depleted

US_15_1: Description: “As a user, I would like to understand the game’s UI/UX clearly, so they don’t interfere with gameplay”
US_15_2: Acceptance Criteria: Accepted when no elements of gameplay obscured behind UI but also all relevant information provided on UI

US_16_1: Description: “As a client, I would like the game to be accessible to as many people as possible so I can share it with friends/peers or monetise it in future”
US_16_2: Acceptance Criteria: Accepted when game is playable in browser across multiple devices

US_17_1: Description: “As a user, I would like to clearly see each action being taken sequentially rather than simultaneously, to ensure I can learn and adapt.”
US_17_2: Acceptance Criteria: Accepted when actions within battle are made and displayed sequentially with delay between.

US_18_1: Description: “As a user, I would like to be able to review the sequence of events after a “battle” in ensure I can learn and adapt.”
US_18_2: Acceptance Criteria: Accepted when user is able to either re-watch battle, or view some representation of the events that took place afterwards

US_19_1: Description: “As a user, I would like to make decisions in which route I should take on the game map rather than follow the one, set route.”
US_19_2: Acceptance Criteria: Accepted when game map forks to different battle nodes of varying difficulty

US_20_1: Description: “As a user, I would like a way to dismiss pop-ups easily and clearly.
US_20_2: Acceptance Criteria: Accepted when user is able to click on a dismiss button and the pop-up closes

US_21_1: Description: “As a user, I would like a way to form strategic formations with my units.”
US_21_2: Description: Acceptance Criteria: Accepted when drag and drop feature implemented

US_22_1: Description: “As a user, I would like to see the health of my units and enemy units depleting in real time.”
US_22_1: Acceptance Criteria: Accepted when health bar, that is a slider that decreases with decreasing health of units is implemented

US_23_1: Description: “As a user, I’d like to be told whether I have won a battle or lost a battle.”

US_23_2: Acceptance Criteria: Accepted when pop-up UI implemented which displays a “You won” or “You lost” message

Requirements use cases | 6.9.4

UC_01:
Use Case: Load Game
Scope: Main menu
Level: User Goal
Context: Users can load a game progression by selecting the “load game” option from the main menu
Users are unable to select a “load game” option from the main menu
Frequency of occurrence: Any time the player enters the programme or wishes to load a game from the map scene
Open issues:

How to ensure a failsafe option to save game prompt in case player wishes to return to original game progression?

UC_02:
Use Case: Exit game
Scope: Main menu
Level: User Goal
Context: Users can exit the game by selecting the “exit game” option from the main menu
Users are unable to select “exit game” option from the main menu
Frequency of occurrence: Any time the player wishes to start exit the game.
Open issues:

How to ensure a failsafe option to save game prompt in case player forgets to save their progress before quitting the programme.

UC_03:
Use Case: Save Game
Scope: N/A
Level: User Goal
Context: Users can save a game by selecting the “save game” option from the main menu
Users are unable to select the “save game” option from the main menu
Frequency of occurrence: Any time the player wishes to save a game.
Open issues:

How do we store saves?
How many saves can be stored by the game?
How many characters can be used in naming the save?
Can we implement a failsafe in case two saves are given the same name?

UC_04:
Use Case: Main menu
Scope: N/A
Level: Subfunction
Context: Users can decide to save load or exit the game by entering the main menu
Users are unable to save load or exit the game as they cannot access the main menu
Frequency of occurrence: Any time the player enters the game or wishes to save, load or exit the game.
Open issues:

Do we display this as a popup or a scene?

UC_05: (COMPLETE)
Use Case: Popup displayed
Scope: N/A
Level: Subfunction
Context: Users are informed of a consequence when they make a game action.

Users fail to be informed of the consequence of a game action taken
Frequency of occurrence: Any time the player makes a game action.
Open issues:
How do we ensure consistent alignment of popups any time they are instantiated? (RESOLVED)
How do we set the informational text of each popup to be dependent on the action taken? (RESOLVED)
How do we ensure the background is still visible when a popup is triggered? (RESOLVED)

UC_06: (COMPLETE)
Use Case: Enter treasure node
Scope: N/A
Level: Subfunction
Context: Upon entering a treasure node, users are rewarded with extra units.
Upon entering a treasure node, users fail to be rewarded with extra units.
Frequency of occurrence: Any time the player enters a treasure node.
Open issues:
How is the number of units rewarded determined? (RESOLVED)
How do we link this to the update unit pool functionality? (RESOLVED)

UC_07: (COMPLETE)
Use Case: Welcome node
Scope: N/A
Level: Subfunction
Context: Users can select the welcome node to initiate progression through the map, and are informed of their objective.
Users can select the welcome node to initiate progression through the map but aren't informed of their objective
Frequency of occurrence: Any time the player enters the map scene of a new game.
Open issues:

UC_08: (COMPLETE)
Use Case: New Arena node
Scope: N/A
Level: Subfunction
Context: Users can select a subsequent new arena node and are informed they are progressing to a new arena.
Users can select a subsequent new arena node but aren't informed they're progressing to a new arena.
Frequency of occurrence: Any time the player can progress to a new arena.
Open issues:

UC_09: (COMPLETE)
Use Case: Game completion node
Scope: N/A
Level: Subfunction
Context: Users can select the game completion node to complete the game and are informed they have won.
Users can select the game completion node to complete the game but aren't informed they have won.
Frequency of occurrence: When the player has completed the final battle encounter of the game.
Open issues:

UC_10: (COMPLETE)
Use Case: Update unit pool

Scope: N/A
Level: User goal
Context: The number of units of the unit count and unit pool can be updated.
The number of units of the unit count and unit pool cannot be updated.
Frequency of occurrence: Every time the player is awarded units.
Open issues:
How do we ensure there is no conflict between the unit count number and the unit pool contents? (RESOLVED)

UC_11: (COMPLETE)
Use Case: Enter battle node
Scope: N/A
Level: Subfunction
Context: Users can select a battle node which initiates the grid scene.
Users can select a battle node but the grid scene fails to be initialised.
Frequency of occurrence: When the player has a battle node adjacent to their current node.
Open issues:
How do we trigger grid scene initialisation. (RESOLVED)
How do we ensure the grid scene initialised is the correct encounter for the current node (ie enemy difficulty etc) ? (RESOLVED)

UC_12:
Use Case: Enemy unit health scaling
Scope: N/A
Level: User goal
Context: Users can select the game completion node to complete the game and are informed they have won.
Users can select the game completion node to complete the game but aren't informed they have won.
Frequency of occurrence: When the player has completed the final battle encounter of the game.
Open issues:
How is scaling factor per node determined?

UC_13:
Use Case: Highlight tile (COMPLETE)
Scope: N/A
Level: Sub function
Context: When a user moves their cursor over a tile in the grid scene the tile is highlighted
When a user moves their cursor over a tile in the grid scene the tile remains unchanged.
Frequency of occurrence: Any time the user is in the grid scene and moves their cursor over a tile.
Open issues:
How do we identify which tile the cursor is over? (RESOLVED)
What colour should the tile be highlighted? (RESOLVED)

UC_14:
Use Case: Select highlighted tile (COMPLETE)
Scope: N/A
Level: User goal
Context: User can select a highlighted tile and the programme recognises the action
Users cannot select the highlighted tile
Frequency of occurrence: Whenever the cursor is above a tile and it is highlighted.
Open issues:
How do we ensure selection of the tile is linked to a distinct action? (RESOLVED)

UC_15: (COMPLETE)

Use Case: Place units on the grid
Scope: N/A
Level: User goal
Context: User can select a unit and place it on a grid tile
User can select a unit but cannot place it on a grid tile
Frequency of occurrence: Whenever the user is in the unit placement sequence.
Open issues:
How do we select a unit? (RESOLVED)
How do we ensure a unit is placed when the tile is selected? (RESOLVED)
How do we ensure the placed unit is deducted from the unit pool? (RESOLVED)

UC_16: (COMPLETE)
Use Case: Unit constant damage output
Scope: N/A
Level: Sub function
Context: During battle sequence, units will output damage consistently which can then be accessed by opposing units.
During battle sequence, units will output damage consistently but output can't be accessed by opposing units.
Frequency of occurrence: Every time a fight sequence is initiated, for every unit.
Open issues:
How do we determine damage output and frequency? (RESOLVED)
When should damage output be initiated? (RESOLVED)
How is each unit's damage output accessed by opposing units? (RESOLVED)

UC_17: (COMPLETE)
Use Case: Unit take damage
Level: Sub function
Context: During battle sequence, units will take damage if they are within range of an opposing unit
During battle sequence, units are unable to take damage when in range of an opposing unit.
Frequency of occurrence: Whenever a unit is within range of an enemy unit.
Open issues:
How do we determine when a unit is within range of another? (RESOLVED)
How does a unit access an opposing unit's damage output? (RESOLVED)
How is health deducted from the unit's health points variable? (RESOLVED)

UC_18: (COMPLETE)
Use Case: Unit death
Level: Sub function
Context: During battle sequence, when a unit's health subceeds zero it will be eliminated from the battle
During battle sequence, when a unit's health subceeds zero it fails to be eliminated from the battle
Frequency of occurrence: Whenever a unit's health subceeds zero.
Open issues:
How to we eliminate an object from the battle? (RESOLVED)
How do we give a unit a "dead" attribute? (RESOLVED)

UC_19:
Use Case: Play game in browser
Level: Summary
Context: Users can access the game through a web browser
Users cannot access the game through a web browser
Frequency of occurrence: Dependent on User
Open issues:

What hosting service do we upload the game to?
How do we make the game as accessible as possible?
What devices will be able to run the game online?

UC_20: (COMPLETE)
Use Case: Move on map
Level: User goal
Context: Users can select an adjacent node and progress through the map
Users are unable to select adjacent nodes.
Frequency of occurrence: Any time the user is in the map scene.
Open issues:
How do we show the current location of the user on the map? (RESOLVED)

UC_21: (COMPLETE)
Use Case: Return to map from grid
Level: User goal
Context: The user can select a button to return to the map scene from the grid scene.
The user is unable to return to the map scene from the grid scene.
Frequency of occurrence: Whenever the user wishes to return to the map from the grid scene.
Open issues:
How to we ensure the player is returned to the node they exited? (RESOLVED)

UC_22: (COMPLETE)
Use Case: Dismiss popup
Level: User goal
Context: Users are able to select a dismissal button which closes the popup
Users are able to select a dismissal button but the popup fails to close
Frequency of occurrence: Whenever a unit's health subceeds zero.
Open issues:
How to we destroy the popup object? (RESOLVED)
How do we change the dismissal button text depending on the trigger scenario? (RESOLVED)
Can we add a function to the dismissal button when required (e.g. a "try again" button for a battle lost popup)? (RESOLVED)

UC_23: (COMPLETE)
Use Case: Fight sequence initiation
Level: Sub function
Context: Users are able to initiate a fight sequence, spawning the enemy units, upon clicking a fight button
Users are unable to initiate a fight sequence upon clicking a fight button
Frequency of occurrence: When the user is satisfied with their unit placement for the battle encounter.
Open issues:
How do we initialise the fight sequence? (RESOLVED)
How to we initialise the enemy units? (RESOLVED)
How do we give the enemy units a random formation each time? (RESOLVED)

UC_24: (COMPLETE)
Use Case: Path-finder/ unit-engagement
Level: Sub function
Context: Units are able to locate the closest opposing unit, lock onto them and plot a course to meet them
Units are able to locate the closest opposing unit but unable to plot a course to meet them.
Frequency of occurrence: Constant during the fight sequence

Open issues:

How do we locate the closest opposing unit? (RESOLVED)

How is the path determined, specifically when there are many units on the board? (RESOLVED)

UC_25: (COMPLETE)
Use Case: 1v1 fight sequence
Level: Sub function
Context: When engaged units are in range they will deal damage to each other until one of them is eliminated.

When engaged units are in range they fail to deal damage to each other.
Frequency of occurrence: Any time two engaged units are within range.
Open issues:

How do we ensure after a unit has defeated another it searches for a new target? (RESOLVED)

UC_26: (COMPLETE)
Use Case: Battle won
Level: User goal
Context: Once all enemy units are eliminated from the board, the battle is terminated and the user is informed they have won.

Once all enemy units are eliminated from the board the battle fails to terminate.
Frequency of occurrence: Any time all enemy units have been eliminated during a fight sequence.
Open issues:

How do we identify when all enemy units have been destroyed? (RESOLVED)

How does the user return to the map scene once the battle is won? (RESOLVED)

UC_27: (COMPLETE)
Use Case: Battle lost
Level: Sub function
Context: When all a player's units are eliminated from the board, the battle is terminated and the user is informed they have lost the battle.

When all a player's units are eliminated from the board, the battle fails to be terminated.
Frequency of occurrence: Any time all player units have been eliminated during a fight sequence.
Open issues:

How do we identify when all player units have been destroyed? (RESOLVED)

How do we return to the unit placement sequence in order try the battle again? (RESOLVED)

UC_28: (COMPLETE)
Use Case: Deploy ranged units
Level: Sub function
Context: Users can deploy ranged units in support of their melee units

Users are unable to deploy ranged units.
Frequency of occurrence: Every time the user is in the unit placement sequence.
Open issues:

How will the pathfinding work with ranged units? (RESOLVED)

What is the optimal amount of damage they should be dealing? (RESOLVED)

How do we ensure that melee units are able to lock onto ranged units so that they are also under threat along with the melee troops? (RESOLVED)

What is the optimum ratio of ranged to melee units for a competitive battle? (RESOLVED)

Do we need to improve the enemy pathfinding system to ensure that they are engaged by enemy units? (RESOLVED)

UC_29: (COMPELTE)
Use Case: Deal damage to non-engaged targets in range.

Level: Sub function
Context: When a unit is engaged with an opposing unit, but the opposing unit is out of range, the unit will deal damage to another non-engaged units which are in range

When a unit is engaged with an opposing unit, but the opposing unit is out of range, the unit cannot deal damage to other non-engaged units which are in range
Frequency of occurrence: When the opposing unit of an engaged unit is out of range but there are non-engaged opposing units in range, constant during fight sequence.
Open issues:

How to we keep track of which units are in range? (RESOLVED)

How do we determine which enemy unit receives the damage if multiple are in range? (RESOLVED)

UC_30: (COMPLETE)
Use Case: Grid scene colour.
Level: Sub function
Context: The colour of the grid scene changes depending on which map tile the battle node is located on.

The colour of the grid scene does not change depending on which map tile the battle node is located on.
Frequency of occurrence: Every time a battle node is entered.
Open issues:

How do we determine what colour tile the node is on? (RESOLVED)

How do we change the colour of the grid scene depending on the tile colour? (RESOLVED)

UC_31:
Use Case: Unit health bar.
Level: Sub function
Context: During the fight sequence all units have a health bar displayed above their sprite which changes according to their current health points.
During the fight sequence all units have a health bar displayed above their sprite but it fails to change according to their health points.
Frequency of occurrence: Constant during every fight sequence, for every alive unit.
Open issues:

How do we link the health bar to the unit's health points and ensure it changes accordingly?

How do we display the health bar above the unit sprite?

How do we ensure the health bar moves with the unit sprite?

CRC cards | 6.10.4

GridManager **References – UC_03**	
RESPONSIBILITIES: **Responsible for creating and containing a data structure containing all Tiles of grid and their positions** **Responsible for making this data available publicly to other functions** **Responsible for making units move around on the grid using a pathfinding algorithm**	COLLABORATORS: GridGraph Tiles BaseUnit Renderer

GridGraph **References – UC_03**	
RESPONSIBILITIES: **Responsible for defining the logic of the graph structure underlying the board** **Responsible for assigning a node to each tile** **Responsible for implementing the pathfinding algorithm**	COLLABORATORS: GridManager Tiles

Renderer **References – UC_01, UC_02, UC_03, UC_04**	
RESPONSIBILITIES: **Responsible for drawing all active sprites onto the game window**	COLLABORATORS: Tiles BaseUnit Dungeon_Tree

Dungeon_Tree **References - UC_04**	
RESPONSIBILITIES: **Responsible for containing recursive tree structure of other trees and relationships between trees** **Responsible for containing sprite representation of Nodes as well as connections between Nodes** **Responsible for listening for mouse click then altering sprite of node clicked to indicate it is selected** **Responsible for containing the current node position of the party** **Responsible for ensuring directly related distal nodes are the only nodes which are able to be selected on click** **Responsible for providing method to change current position of party to newly selected node**	COLLABORATORS: Renderer

BaseUnit **References – US_05_1, US_05_2, US_08_1, US_08_2, UC_03**	
RESPONSIBILITIES: **Responsible for containing and updating unit position** **Responsible for having method available for other functions/classes to change Unit position** **Responsible for containing and updating health, attack speed, and damage** **Responsible for having method available for other functions/classes to "attack" surrounding units** **Responsible for removing the unit and freeing the node when a unit is dead** **Responsible for finding the nearest enemy unit and move towards it**	COLLABORATORS: UnitManager Renderer

UnitManager **References – US_05_1, US_05_2, US_08_1, US_08_2, UC_03**	
RESPONSIBILITIES: **Create new instances of the player's units** **Check whether the win or lose condition is met** **Place enemy units in random formations**	COLLABORATORS: BaseUnit GridManager Renderer

UnitSelector **References – UC_03**	
RESPONSIBILITIES: **Handle drag-and-drop action for all units** **Pass the drag-and-drop position to other classes/functions** **Update the units count when drag-and-drop actions are performed**	COLLABORATORS: GameManager GridManager UnitManager

UIManager **Reference – UC_01, UC_04**	
RESPONSIBILITIES: Responsible for providing pop ups on certain dungeon levels Responsible for providing item counter Responsible for providing asset counter	COLLABORATORS: MapManager

MapManager **Reference – UC_04**	
RESPONSIBILITIES: Responsible to generating the map Responsible for saving the current state of the map Responsible for saving the player's progress	COLLABORATORS: UIManager

HoverHighlight **References – UC_03**

RESPONSIBILITIES:	COLLABORATORS:
Highlight the tile which the cursor is hovering over	GridManager GridGraph

HealthBar **References – UC_03, US_05_1**	
RESPONSIBILITIES:	COLLABORATORS:
Responsible for displaying a health bar for each unit on the board **Responsible for taking in the max health and current health of each unit**	UnitManager

Popup **References – US_14_1**	
RESPONSIBILITIES:	COLLABORATORS:
Show different messages and rewards for when the battle is won or lost	UnitManager GridManager

ObjectClick **References – US_14_1**	
RESPONSIBILITIES:	COLLABORATORS:
Create different lightings for active and inactive nodes on the map	UnitManager MapManager

PlayerController **References – UC_04, US_07_1, US_15_1**	
RESPONSIBILITIES:	COLLABORATORS:
Enable the player to pan their camera around the map **Restrict the camera to be inside the map**	MapManager

Design use cases | 6.11.4

UC_01

UC_01_1: **Use Case**: Load game
UC_01_2: **Author**: DD
UC_01_3: **Date**: 15/11/21
UC_01_4: **Purpose**: Initiate new game from within main menu
UC_01_5: **Overview**: Starts when User presses load game button on main menu screen. On button press the player should be prompted to select a game save with one of the options being a new game. The selected game instance should begin, with scene either in dungeon map or in hub depending on program progress.
UC_01_6: **Cross References**: US_03_1, US_03_2
UC_01_7: **Actors**: User
UC_01_8: **Pre-conditions**:
UC_01_Pre-1: Program should be in main menu scene
UC_01_Pre-2: load game button must be "listening" for click event with prepared new game and load game functions actioned on-click
UC_01_9: **Post Conditions**:
UC_01_Post_1: Scene should be changed to a dungeon map with the respective progression selected
UC_01_10: **Main flow of events**:
User Starts the game and enters the main menu scene.
User correctly selects the load game button which prompts the user to select a save.
User selects a save and is transferred to the dungeon scene with map progression corresponding to the respective save selected.
UC_01_11: **Alternative flow of events**:
If user clicks outside of area of button actively listening for input, no action taken.
User selects the load game button and selects a save but is transferred to a new dungeon map scene with the incorrect progression loaded.
In the case of a failure of this process with user unable to begin new game instance, product not viable, if not rectified would necessitate removing main meu scene.
UC_01_12: **Testing**:
In testing this function, ensure that on mouse click, the new game button calls the new game function, and that the scene then changes appropriately to the hub/dungeon depending on the stage of development.

UC_02

UC_02_1: **Use Case**: Exit Game
UC_02_2: **Author**: DD
UC_02_3: **Date**: 15/11/21
UC_02_4: **Purpose**: End game process from within game window
UC_02_5: **Overview**: Starts when User presses quit game button on main menu screen. Button should use built in Unity application.quit() method to end game process.
UC_02_6: **Cross References**: US_01_1, US_01_2
UC_02_7: **Actors**: User
UC_02_8: **Pre-conditions**:
UC_02_Pre-1: Program should be in main menu scene
UC_02_Pre-2: Exit button must be "listening" for click event with prepared quit game function actioned on-click
UC_02_9: **Post Conditions**:
UC_02_Post_1: The program and all related processes must end.

UC_02_10: **Main flow of events**:

User enters the main menu scene and correctly selects the exit game button.

The programme successfully calls the application.quit() method and is terminated.

UC_02_11: **Alternative flow** of events:

If user clicks outside of area of button actively listening for input, no action taken.

In the case of a failure of this process the User can instead use system services to close the game, through the keyboard alt-F4 shortcut or through the windows task manager.

UC_02_ 12: **Testing**: When testing, ensure that on mouse click, the exit game button calls the exit game function.

UC_03

UC_03_1: **Use Case**: Save Game
UC_03_2: **Author**: CP
UC_03_3: **Date**: 15/11/21
UC_03_4: **Purpose**: Save game progress from within game window
UC_03_5: **Overview**: Starts when User presses save game button on main menu screen. This should save the player's progress through the map where the user has the option to save the game under a name.
UC_03_6: **Cross References**: US_02_1, US_02_2
UC_03_7: **Actors**: User
UC_03_8: **Pre-conditions**:
UC_03_Pre-1: Program should be in main menu scene
UC_03_Pre-2: Save button must be "listening" for click event with prepared save game function actioned on-click
UC_03_9: **Post Conditions**:
UC_03_Post_1: The program saves and stores current game progress under the player's name.
UC_03_10: **Main flow**:

When the player is in the main menu and decides they want to save their progress they select the save game button and the player is prompted to enter their name.

The player enters their name and selects okay.

The player sees their name listed under saved games.

UC_02_11: **Alternative flow** of events:

The player selects the save game button but is not prompted to enter their name.

After the player enters their name, they cannot see it listed under saved games.

UC_02_12: **Testing**:
When testing ensure the game is saved with the correct name entered and when the game is re-opened the user is at the same map position with the same number of troops as when they left the game.

UC_04

UC_04_1: **Use Case**: Main Menu
UC_04_2: **Author**: CP
UC_04_3: **Date**: 15/11/21
UC_04_4: **Purpose**: Interface for saving and exiting the game
UC_04_5: **Overview**: The user should be able to select a main menu option from the map scene from which they can choose to load a game, save or exit the game. User should also be met by the main menu upon entry to the game.
UC_04_6: **Cross References**: US_02_1, US_02_2
UC_04_7: **Actors**: User
UC_04_8: **Pre-conditions**:
UC_04_Pre-1: Program should be in map scene or have just been run
UC_04_9: **Post Conditions**:

UC_04_Post_1: The program displays the main menu with all its options.
UC_04_10: **Main flow**:

When the player is in the map scene, they select the main menu option, triggering the main menu screen.
The main menu is displayed with options to load game, save game or exit game
When the player enters the game they are met by the main menu scene displaying options to load game, save game or exit game

UC_04_11: **Alternative flow** of events:

The player selects the main menu button but the main menu scene is not triggered.
After the player enters the game they are not met by the main menu screen and instead transferred directly to the dungeon map scene.

UC_04_12: **Testing**:
When testing ensure the main menu screen is triggered upon selecting the main menu button. Ensure that upon entry to the game the player is met by the main menu screen.

UC_05
UC_05_1: **Use Case**: Popup displayed
UC_05_2: **Author**: CP
UC_05_3: **Date**: 15/11/21
UC_05_4: **Purpose**: Inform player of a game consequence.
UC_05_5: **Overview**: When the user takes an action and a consequence is reached, they should be informed of the resulting consequence.
UC_05_6: **Cross References**: US_07_01, US_07_02, US_15_01, US_15_02.
UC_05_7: **Actors**: User interface.
UC_05_8: **Pre-conditions**:

UC_05_Pre-1: Player has made an action in the game.
UC_05_Pre-2: The action taken has a consequence.

UC_05_9: **Post Conditions**:

UC_04_Post_1: The player understands the consequence of the action taken.

UC_05_10: **Main flow**:

The player makes an action in the game (e.g. enters a treasure node)
The popup is displayed clearly on the scene, with the background still visible
The player understands the consequence of the action taken (e.g. has been awarded 4 units).

UC_05_11: **Alternative flow** of events:

The player makes an action in the game (e.g. enters a treasure node)
The popup is not displayed clearly on the scene.
The popup is displayed clearly on the scene; however, the background is not visible.
The popup is displayed clearly on the scene but the player does not understand the consequence of the action taken.

UC_05_12: **Testing**:
When testing ensure that the popup is correctly aligned in the centre of the screen and the background scene is still visible to the user. Ensure that a situationally required message can be displayed dependent on the action triggering the popup.

UC_06
UC_06_1: **Use Case**: Enter treasure node.
UC_06_2: **Author**: CP
UC_06_3: **Date**: 15/11/21
UC_06_4: **Purpose**: Informs user of their treasure reward units
UC_06_5: **Overview**: When a user selects a treasure node they should be greeted by a popup informing them of how many troops they have gained as a reward.

UC_06_6: **Cross References**: US_07_1, US_07_2, US_09_1, US_09_2, US_10_1, US_10_2,
UC_06_7: **Actors**: User, User interface
UC_06_8: **Pre-conditions**:
UC_06_Pre-1: Programme should be on map scene
UC_06_Pre-2: Treasure node is adjacent to current node.
UC_06_9: **Post Conditions**:
UC_06_Post_1: New occupied node should now be the treasure node
UC_06_Post_2: A dismissible popup should now displayed on the screen informing the user of troop reward
UC_06_Post_3: Unit pool should be updated to reflect additional troops gained.
UC_06_10: **Main flow**:
User sees a treasure node adjacent to current node and selects it.
After the treasure node is selected, a dismissible popup is displayed informing the user of their reward.
After the popup is dismissed the unit pool is updated according to the number of troops awarded.
UC_06_11: **Alternative flow** of events:
User sees a treasure node adjacent to current node, but cannot select it.
After selecting the treasure node, the programme fails to display a popup message.
After selecting a treasure node, a popup is displayed showing the chest reward, but the user is unable to dismiss it.
After the popup is dismissed, the unit pool is not updated to show the extra units awarded.
UC_06_12: **Testing**:
When testing, ensure selection of the treasure node is recognised by the programme and this can be used to trigger a popup. Ensure the popup can successfully be dismissed and that the popup management system updates the unit manager to reflect the additional units gained so that this is displayed in the unit pool.

UC_07
UC_07_1: **Use Case**: Welcome node
UC_07_2: **Author**: CP
UC_07_3: **Date**: 15/11/21
UC_07_4: **Purpose**: Introduce the player to the game.
UC_07_5: **Overview**: Upon entering the game the player should be able to select the starting node where a welcome message is displayed, welcoming the player to the game and informing them of the point they must reach to complete it.
UC_07_6: **Cross References**: US_03_1, US_03_2
UC_07_7: **Actors**: User, User Interface
UC_07_8: **Pre-conditions**:
UC_07_Pre-1: The game has been started and the map scene initiated.
UC_07_Pre-2: No nodes have been clicked and activated yet.
UC_07_9: **Post Conditions**:
UC_07_Post_1: A dismissible popup should be displayed welcoming the player and informing them of the game endpoint (star)
UC_07_10: **Main flow**:
The player enters the game and clicks on the first node on the map.
A dismissible popup is displayed welcoming the player to the game and informing them they must reach the star node at the end of the map.
Popup is dismissed and welcome node is now darkened indicating it is the user's current position on the map.
UC_07_11: **Alternative flow** of events:

The player enters the game and clicks on the first node.
A dismissible popup fails to be displayed to welcome the player or inform them of the objective.
A dismissible popup is created but it is not visible on the screen and therefore cannot be dismissed.
Popup is dismissed but the welcome node is still selectable indicating the user hasn't taken position at the welcome node yet.

UC_07_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct welcome message. Ensure node is darkened and no longer selectable after dismissal.

UC_08

UC_08_1: **Use Case**: New arena node
UC_08_2: **Author**: CP
UC_08_3: **Date**: 15/11/21
UC_08_4: **Purpose**: Transfers the player to the first node of a new arena and informs them they've completed the previous arena.
UC_08_5: **Overview**: When a player have taken a path through an arena and made it to the end, they should be able to select a new arena node which informs them they've completed the previous arena. The player can then choose between different paths through the new arena.
UC_08_6: **Cross References**: US_06_1, US_06_2
UC_08_7: **Actors**: User
UC_08_8: **Pre-conditions**:
UC_08_Pre-1: Player has reached and completed one of the final nodes of their current arena.
UC_08_Pre-2:
UC_08_9: **Post Conditions**:
UC_08_Post_1: Player has the option to select from different starting nodes through the new arena with different paths and encounters.
UC_08_10: **Main flow**:
The player selects the new arena node.
A popup is displayed informing the player that they can now enter a new arena.
The player is transferred to the new arena node and can now select a path through the new arena.
UC_08_11: **Alternative flow** of events:
The player is unable to select the new arena node
The player selects the new arena node but a popup informing the player they can now enter a new arena fails to be displayed.
The player fails to be transferred to the new arena node and therefore cannot select a path through the new arena.
UC_08_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct enter new arena message displayed. Ensure node is darkened and no longer selectable after dismissal.

UC_09

UC_09_1: **Use Case**: Game completion node
UC_09_2: **Author**: CP
UC_09_3: **Date**: 15/11/21
UC_09_4: **Purpose**: Initiates and informs user of completion of the game.
UC_09_5: **Overview**: There should be a final node on the map, whose reaching is the objective of the game. Upon entry, the player should be informed they have completed the game and congratulated.

UC_09_6: **Cross References**: US_01_1, US_01_2
UC_09_7: **Actors**: User
UC_09_8: **Pre-conditions**:
UC_09_Pre-1: The player has successfully completed the last battle node of the game.
UC_09_Pre-2: The player has returned to the map scene.
UC_09_9: **Post Conditions**:
UC_09_Post_1: The game is complete and a message is shown informing the user they have won.
UC_09_10: **Main flow**:
After completing the final battle encounter the player has returned to the map scene.
The player selects the final node on the map.
A dismissible popup is initiated, with a message congratulating the user for winning the game.
UC_09_11: **Alternative flow** of events:
After completing the final battle encounter the player has returned to the map scene.
The player selects the final node on the map.
A dismissible popup fails to be initiated and the player is not informed they have won the game.
UC_09_12: **Testing:**
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct congratulatory message displayed. Ensure node is darkened and no longer selectable after dismissal.

UC_10
UC_10_1: **Use Case**: Update unit pool
UC_10_2: **Author**: CP
UC_10_3: **Date**: 15/11/21
UC_10_4: **Purpose**: Reflects additional troops awarded to the player
UC_10_5: **Overview**: When a player is awarded units from a treasure node, the unit pool should be updated to show the new total number of units.
UC_10_6: **Cross References**: US_06_1, US_06_2
UC_10_7: **Actors**: User
UC_10_8: **Pre-conditions**:
UC_10_Pre-1: Player has selected a treasure node.
UC_10_Pre-2: Unit reward has been determined.
UC_10_9: **Post Conditions**:
UC_10_Post_1: Unit pool should reflect the units awarded on top of the units in possession preceding node selection
UC_10_10: **Main flow**:
User selects a treasure node and is informed of unit reward.
Unit pool is updated to show the sum of the player's previous number of units and the number of units awarded.
UC_10_11: **Alternative flow** of events:
User selects a treasure node and is informed of unit reward.
Unit pool is not updated and remains the same as before node entry.
Unit pool is updated; however the new value is not the sum of the player's previous number of units and the number of units awarded.
UC_10_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct unit award message displayed. Ensure that the unit count and unit pool are updated to reflect the extra troops rewarded.

UC_11

UC_11_1: **Use Case**: Enter battle node
UC_11_2: **Author**: CP
UC_11_3: **Date**: 15/11/21
UC_11_4: **Purpose**: Allow user to enter a battle node to progress through the map.
UC_11_5: **Overview**: When the user is in the map scene and the adjacent node is a battle node, they should be able to select the node and be transferred to the respective level's grid scene.
UC_11_6: **Cross References**: US_04_1, US_04_2, US_08_1, US_08_2, US_09_1, US_09_2, US_10_1, US_10_2
UC_11_7: **Actors**: User
UC_11_8: **Pre-conditions**:
UC_11_Pre-1: Programme is on map scene.
UC_11_Pre-2: These is a battle node adjacent to the current node.
UC_11_9: **Post Conditions**:
UC_11_Post_1: The player is transferred to grid scene
UC_11_10: **Main flow**:
The player is currently at a node with a battle node adjacent.
The player selects the battle node.
Upon selecting the battle node, the player is transferred to the grid scene.
UC_11_11: **Alternative flow** of events:
The player selects the battle node; however, they are unable to select the battle node.
The player selects the battle node; however, they are not transferred to the grid scene.
UC_11_12: Testing:
When testing ensure that the grid scene is loaded successfully with the unit pool of the unit placement scene reflecting the number of units in the unit count of the map scene.

UC_12

UC_12_1: **Use Case**: Enemy unit health scaling
UC_12_2: **Author**: CP
UC_12_3: **Date**: 15/11/21
UC_12_4: **Purpose**: To add difficulty to battle encounters as the player progresses through the game
UC_12_5: **Overview**: As the player progresses through the game, the health of enemy units should increase at each battle encounter.
UC_12_6: **Cross References**: US_09_1, US_09_2, US_12_1, US_12_2
UC_12_7: **Actors**: User
UC_12_8: **Pre-conditions**:
UC_12_Pre-1: Player has completed at least one battle encounter.
UC_12_9: **Post Conditions**:
UC_12_Post_1: The enemy units encountered at the subsequent battle node have more health points than those of the previous node.
UC_12_10: **Main flow**:
The player completes a battle encounter.
The player enters a subsequent battle node and encounters enemy units with one more health point than those of the previous node.
UC_12_11: **Alternative flow** of events:
The player completes a battle encounter.
The player enters a subsequent battle node; however, the enemy units encountered have the same amount of or less health points than those of the previous node.
UC_12_12: **Testing**:
When testing ensure that the enemy units in each subsequent battle node have one extra health point.

UC_13

UC_13_1: **Use Case**: Highlight tile

UC_13_2: **Author**: CP
UC_13_3: **Date**: 15/11/21
UC_13_4: **Purpose**: Highlights the tile which the user's cursor is interacting with at its current position.
UC_13_5: **Overview**: During the grid scene, the player should be able to see which tile their cursor is interacting with according to its position, where the current tile is highlighted
UC_13_6: **Cross References**: US_05_1, US_05_2, US_13_1, US_13_2
UC_13_7: **Actors**: User
UC_13_8: **Pre-conditions**:
UC_13_Pre-1: The programme should be in the grid scene, and the board displayed
UC_13_9: **Post Conditions**:
UC_13_Post_1: When the cursor moves over a tile, the tile is highlighted.
UC_13_10: **Main flow**:
During the grid scene the player moves the cursor over a specific tile.
The tile the cursor is positioned over is highlighted.
UC_13_11: **Alternative flow** of events:
During the grid scene the player moves the cursor over a specific tile.
The tile the cursor is positioned over is highlighted along with one or more other tiles which the cursor is not positioned over.
The tile the cursor is positioned over is not highlighted; however one or more other tiles which the cursor is not positioned over are highlighted.
The tile the cursor is positioned over is not highlighted nor is any other tile on the board.
UC_13_12: **Testing**:
When testing ensure that only the tile the cursor is hovering over is highlighted and this is reflected as the cursor is moved around the grid.

UC_14
UC_14_1: **Use Case**: Select highlighted tile
UC_14_2: **Author**: CP
UC_14_3: **Date**: 15/11/21
UC_14_4: **Purpose**: To enable action to be performed upon selection of a tile.
UC_14_5: **Overview**: During the grid scene, the player should be able to select a highlighted tile which the cursor is positioned over and the programme should recognise this selection enabling an action to be performed.
UC_14_6: **Cross References**: US_08_01, US_08_02, US_05_1, US_05_2, US_13_1, US_13_2
UC_14_7: **Actors**: User
UC_14_8: **Pre-conditions**:
UC_14_Pre-1: The programme should be in the grid scene
UC_14_Pre-2: The user should have the cursor positioned over the highlighted tile which they wish to select.
UC_14_9: **Post Conditions**:
UC_14_Post_1: The tile is selected, turning grey and the action is recognised by the programme.
UC_14_10: **Main flow**:
During the grid scene the player moves the cursor over a specific tile, which is highlighted.
The cursor changes to a hand symbol letting the player know its clickable,
The player selects the tile, which turns grey, meaning the programme recognises the action.
UC_14_11: **Alternative flow** of events:
During the grid scene the player moves the cursor over a specific tile, which is highlighted
The cursor remains unchanged, so they do not know its clickable.

The cursor changes to a hand symbol, letting the player know its clickable, the player selects the tile; However, the tile does not change colour, meaning the programme does not recognise the action.

UC_14_12: **Testing**:
When testing ensure that it is only the highlighted tile which is selected and the programme recognises the action.

UC_15

UC_15_1: **Use Case**: Place units on grid
UC_15_2: **Author**: DD
UC_15_3: **Date**: 15/11/21
UC_15_4: **Purpose**: Allow user to place units on grid before battle
UC_15_5: **Overview**: Starts during placement phase of auto-battler encounter. Player should be able to select from persistent pool of units, visible on the lower edge of the screen in a relevant UI object, and these can then be placed in empty hexes on the board on the lower half of the game board.
UC_15_6: **Cross References**: US_08_01, US_08_02, US_11_01, US_11_02, US_14_01, US_14_01, US_15_01, US_15_02
UC_15_7: **Actors**: User
UC_15_8: **Pre-conditions**:
UC_15_Pre-1: Program should be within encounter scene
UC_15_Pre-2: Current state of encounter should be planning
UC_15_Pre-3: User must have units available to place
UC_15_Pre-4: At least one square on board should not be full
UC_15_Pre-5: Current units placed must be less than maximum placeable units
UC_15_9: **Post Conditions**:
UC_15_Post_1: Selected unit should be placed on the game board
UC_15_Post_2: The Square on the board the unit now occupies should no longer be empty
UC_15_Post_3: One copy of the selected unit should be removed from the unit pool temporarily
UC_15_10: **Main flow of events**:
Player Enters the battle node initiating the placement phase of the battle encounter.
Player select a unit from their unit pool.
Unit is placed on one of the empty hexes on the board.
UC_15_11: **Alternative flow** of events:
User tries to place unit on non-empty square, explanatory error message is shown
User tries to place unit from empty pool of units, explanatory error message is shown
UC_15_12: Testing:
When testing, ensure unit clicked becomes selected. Also ensure when subsequently clicking on tile that a new unit of the selected type is instantiated, with a position equal to the selected tile.

UC_16

UC_16_1: **Use Case**: Unit constant damage output
UC_16_2: **Author**: CP
UC_16_3: **Date**: 15/11/21
UC_16_4: **Purpose**: Allows units to deal damage to opposing units.
UC_16_5: **Overview**: Units consistently output damage with amount and rate determined by the unit type.
UC_16_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2
UC_16_7: **Actors**: User
UC_16_8: **Pre-conditions**:
UC_16_Pre-1: Units are placed on the board

UC_16_Pre-2: Fight sequence is initiated

UC_16_9: **Post Conditions**:

UC_16_Post_1: Damage is consistently output while the unit is not dead.

UC_16_10: **Main flow**:

The fight sequence is initiated and all units on the board begin outputting damage.

Damage is consistently output until the unit is dead.

UC_16_11: **Alternative flow** of events:

The fight sequence is initiated; however, some or all the units do not begin outputting damage.

Damage is consistently output by each unit, even after the unit is dead.

UC_16_12: **Testing**:

When testing ensure the unit damage output is accessible to other units so that they can use it to take damage.

UC_17

UC_17_1: **Use Case**: Unit take damage

UC_17_2: **Author**: CP

UC_17_3: **Date**: 15/11/21

UC_17_4: **Purpose**: Allows units to take damage from other units

UC_17_5: **Overview**: When a unit is in range of an opposing unit, it should be dealt damage corresponding to the opposing unit's damage output.

UC_17_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2

UC_17_7: **Actors**: User

UC_17_8: **Pre-conditions**:

UC_17_Pre-1: The unit has been engaged by an opposing unit.

UC_17_Pre-2: The unit is in range of the opposing unit.

UC_17_9: **Post Conditions**:

UC_17_Post_1: The unit health is consistently reduced according to the opposing unit's damage output.

UC_17_10: **Main flow**:

The unit comes into range of an opposing unit which has engaged it.

The opposing unit begins consistently dealing damage to the unit while it is in range.

The health variable of the unit is consistently reduced according to the opposing unit's damage output.

UC_17_11: **Alternative flow** of events:

The unit comes into range of the opposing unit which has engaged it.

The opposing unit fails to deal damage to the unit

The opposing unit

The health attribute of a unit engaged in a 1v1 battle subceeds zero.

The dead Boolean of the unit object fails to be set to true.

UC_17_12: **Testing**:

When testing ensure the damage output of the unit dealing the damage is accessible to the one taking it. Ensure a corresponding amount of health points are deducted from the unit's health variable.

UC_18

UC_18_1: **Use Case**: Unit death

UC_18_2: **Author**: CP

UC_18_3: **Date**: 15/11/21

UC_18_4: **Purpose**: Criteria for the elimination of a unit from a battle.

UC_18_5: **Overview**: When the health of a unit has reached or subceeded zero, the unit is given the dead attribute, which can then be used to disengage the opposing unit and identify unit objects which need to be destroyed.
UC_18_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2
UC_18_7: **Actors**: User
UC_18_8: **Pre-conditions**:
UC_18_Pre-1: The unit in question has been engaged in a 1v1 battle.
UC_18_Pre-2: The health of the unit is less than or equal to zero.
UC_18_9: **Post Conditions**:
UC_17_Post_1: The unit is given the dead attribute which is a Boolean set to true.
UC_18_10: **Main flow**:
The health attribute of a unit engaged in a 1v1 battle subceeds zero.
The "dead" Boolean of the unit object is set to true.
The "dead" unit object is destroyed, eliminating it from the battle
UC_18_11: **Alternative flow** of events:
The health attribute of a unit engaged in a 1v1 battle subceeds zero.
The dead Boolean of the unit object fails to be set to true.
The dead Boolean of the object is set to true but it is not destroyed, failing to eliminate it from the game.
UC_18_12: **Testing**:
When testing ensure that the unit object is successfully destroyed upon its dead Boolean being set to true

UC_19
UC_19_1: **Use Case**: Play game in browser
UC_19_2: **Author**: CP
UC_19_3: **Date**: 15/11/21
UC_19_4: **Purpose**: Allow players to play game in a browser across multiple devices.
UC_19_5: **Overview**: The game should be playable in a browser.
UC_19_6: **Cross References**: US_16_1, US_16_2
UC_19_7: **Actors**: User
UC_19_8: **Pre-conditions**:
UC_19_Pre-1: The game is fully functional.
UC_19_9: **Post Conditions**:
UC_19_Post_1: The game is playable on a web browser.
UC_19_10: **Main flow**:
The game is uploaded to unity free hosting service and embedded in a web page.
The game is playable online
UC_19_11: **Alternative flow** of events:
The game fails to be uploaded to unity free hosting service
UC_19_12: **Testing**:
When testing ensure that the game is playable across a range of devices on a range of web browsers.

UC_20
UC_20_1: **Use Case**: Move on Map
UC_20_2: **Author**: DD
UC_20_3: **Date**: 22/11/21
UC_20_4: **Purpose**: Allow user to change current location to adjacent node to progress through map
UC_20_5: **Overview**: When user is in the map scene, they should be able to select any of the adjacent nodes on the map, and their current position should then be changed to this node. This should also be visually indicated on the scene

UC_20_6: **Cross References**: US_05_1, US_05_2, US_06_1, US_06_2, US_07_1, US_07_2, US_13_1, US_13_2, US_19_1, US_19_2
UC_20_7: **Actors**: User
UC_20_8: **Pre-conditions**:
UC_20_Pre-1: Program should be in map scene
UC_20_Pre-2: Current party position should not be at end node
UC_20_9: **Post Conditions**:
UC_20_Post_1: Current party position should change to selected node
UC_20_Post_2: The visual indicator for party position should change to selected node
UC_20_Post_3: Event for new node position should be triggered
UC_20_10: **Main flow**:
UC_20_11: **Alternative flow** of events:
If a user selects a node proximal in the tree, the party position should remain at the current node
If a user selects the currently occupied node, the party position should remain at the current node
UC_20_12: **Testing**: When testing, ensure only nodes distal and adjacent to active nodes are selectable. Also ensure that when selectable nodes are clicked, the party position then changes to this current node.

UC_21
UC_21_1: **Use Case**: Dismiss Popup
UC_21_2: **Author**: CP
UC_21_3: **Date**: 22/11/21
UC_21_4: **Purpose**: Allow user to Dismiss a popup from the screen.
UC_21_5: **Overview**: When a popup is displayed, the user should be able to easily select a dismiss button to close it.
UC_21_6: **Cross References**: US_15_1, US_15_2, US_20_1, US_20_2
UC_21_7: **Actors**: User
UC_21_8: **Pre-conditions**:
UC_04_Pre-1: A dismissible popup is currently in display on the current scene.
UC_21_9: **Post Conditions**:
UC_04_Post_1: Popup is dismissed from the scene.
UC_20_10: **Main flow**:
The user successfully selects the dismiss button.
The popup is closed.
UC_21_11: **Alternative flow** of events:
The user fails to select the dismiss button.
The user successfully selects the dismiss button; however, the popup isn't closed
UC_21_12: **Testing**:
When testing ensure that selecting the dismiss option of the popup successfully destroys the popup object.

UC_22
UC_22_1: **Use Case**: Fight sequence initiation
UC_22_2: **Author**: CP
UC_22_3: **Date**: 29/11/21
UC_22_4: **Purpose**: Positions enemy units in a random formation, giving variety to battle gameplay, and initiates fight sequence.
UC_22_5: **Overview**: When a user selects the fight button, the respective enemy units for that level should be positioned in a random formation and the fight sequence should be initiated.

UC_22_6: **Cross References**: US_08_1, US_08_2, US_21_1: , US_21_1:2, US_22_1, US_22_2
UC_22_7: **Actors**: User, units
UC_22_8: **Pre-conditions**:
UC_22_Pre-1: Programme is in grid scene with all player units placed
UC_22_Pre-2: fight button has been selected.
UC_22_9: **Post Conditions**:
UC_22_Post_1: All enemy units are placed on the board in a random formation.
UC_22_Post_3: Fight sequence is initiated.
UC_22_10: **Main flow**:
After the player is satisfied with their unit placement, they select the fight button on the grid scene.
The enemy units for the respective level are placed in a random formation on the board.
The fight sequence is initiated and the battle begins.
Battle continues until one side's units have all been eliminated.
UC_22_11: **Alternative flow** of events:
After the player is satisfied with their unit placement, they select the fight button on the grid scene.
The enemy units for the respective level are placed in a non-random formation on the board, which is the same every time.
The enemy units are placed on the board in a random formation; however, the fight sequence fails to be initiated and the battle does not begin.
The fight sequence is successfully initiated however one or more units fail to engage an opponent.
All units engage an opponent however one or more units fail to lock onto a new opponent after their opponent has been destroyed. This could lead to units stopping fight before one side has been eliminated.
UC_05_12: **Testing**:
When testing ensure all units are activated and engage an opponent. Ensure that units keep fighting unit all the units of one team are eliminated. Ensure units only deal damage to opposing units. Ensure after destroying an opponent, units are successfully able to lock onto a new opponent.

UC_23
UC_23_1: **Use Case**: Path-finder/ unit engagement
UC_23_2: **Author**: CP
UC_23_3: **Date**: 29/11/21
UC_23_4: **Purpose**: Engages troops in combat
UC_23_5: **Overview**: When a battle is initiated, both user and enemy units will search for the closest opposing unit and lock onto them. The units will then move towards each other until they are within attacking range at which point they will stop and begin fighting. If a unit successfully wins, it then locks onto a new opponent and the process is repeated.
UC_23_6: **Cross References**: US_08_1, US_08_2, US_21_1: , US_21_1:2, US_22_1, US_22_2
UC_23_7: **Actors**: Units
UC_23_8: **Pre-conditions**:
UC_23_Pre-1: User has placed all units on the board and initiated the fight sequence.
UC_23_Pre-2: Enemy units have successfully been deployed on the board upon initiation of fight sequence.
UC_23_9: **Post Conditions**:
UC_23_Post_1: All units on the board have successfully locked onto an opposing unit
UC_23_Post_2: All units have managed to reach and engage their opposing unit in battle.
UC_23_10: **Main flow**:

Fight sequence is initiated, all units on the board successfully lock onto an opposing unit and begin moving towards them.

All units successfully reach their respective opposing units and are able to engage them in battle.

The unit which wins the battle then locks onto a new opponent and the process is repeated.

UC_23_11: **Alternative flow** of events:

Fight sequence is initiated; however, some or all units are unable to successfully lock onto an opposing unit.

All units successfully lock onto an opposing unit; however, some or all units are unable to successfully reach their respective opposing unit.

All units successfully reach their respective opposing unit; however, some or all units fail to engage their respective opposing unit in battle.

All units successfully engage their opposing unit in battle; however, if they win their battle, they fail to engage a new unit and become inactive on the board.

UC_23_12: **Testing**:

When testing ensure each unit locks onto its closest opposing unit. Ensure after

UC_24

UC_24_1: **Use Case**: 1v1 fighting sequence

UC_24_2: **Author**: CP

UC_24_3: **Date**: 29/11/21

UC_24_4: **Purpose**: Process for engaged units to fight one another.

UC_24_5: **Overview**: When locked on units reach each other and are in range, they begin to deal damage to each other, according to their attributes, until one defeats the other.

UC_24_6: **Cross References**: US_08_1, US_08_2, US_21_1: , US_21_1:2, US_22_1, US_22_2

UC_24_7: **Actors**: User

UC_24_8: **Pre-conditions**:

UC_24_Pre-1: All units have been placed and fight sequence has been initiated.

UC_24_Pre-2: Pathfinding has successfully brought two opposing units together so that they're within attack range of each other.

UC_24_9: **Post Conditions**:

UC_24_Post_1: Units have successfully engaged in battle and one has been destroyed.

UC_24_10: **Main flow**:

Two engaged units are within range at which point they begin dealing damage to each other.

Each unit deals a certain amount of damage to the other at a constant rate. These variables are determined by the unit type.

Once the health of one of the units is less than or equal to zero, the unit object is destroyed and the opposing unit has won the battle.

UC_24_11: **Alternative flow** of events:

Two engaged units are within range of each other; however, on or both the units fail to deal damage to the other.

Once the health of one of the units is less than or equal to zero, the unit object fails to be destroyed. The units stay engaged in battle and keep dealing damage to each other.

UC_24_12: **Testing**:

When testing ensure that both units, when in range, deal damage to each other. Ensure that as soon as one unit's health points subceeds zero, it is destroyed. Ensure the correct damage output and frequencies of units are received by the opposing unit.

UC_25

UC_07_1: **Use Case**: Return to map button.

UC_25_2: **Author**: CP

UC_25_3: **Date**: 29/11/21
UC_25_4: **Purpose**: Allows the player to exit the battle sequence and go back to the grid scene.
UC_25_5: **Overview**: When the player is in the grid scene, they should be able to return to the map by selecting a return button at any point the grid scene sequences.
UC_25_6: **Cross References**: US_04_1, US_04_2, US_10_1, US_10_2
UC_25_7: **Actors**: User, grid manager
UC_25_8: **Pre-conditions**:
UC_04_Pre-1: Player is currently in the grid scene
UC_25_9: **Post Conditions**:
UC_04_Post_1: Player is returned to the map scene positioned at the node they exited.
UC_25_10: **Main flow**:
Player is in the grid scene and selects the return to map button.
Player is returned to the map and they are positioned at the node they exited.
UC_25_11: **Alternative flow** of events:
Player is in grid scene and fails to select the return to map button.
Player selects the return to map button; however is not successfully transferred to the map scene.
The player returns to the map without completing the battle encounter; however, they are no longer positioned at the node they exited.
UC_25_12: **Testing**:
When testing ensure that upon selecting the return to map button the user is transferred to the map scene and positioned at the node they exited if they did not complete the battle and the node they entered if they did.

UC_26
UC_26_1: **Use Case**: Battle won
UC_26_2: **Author**: CP
UC_26_3: **Date**: 6/12/21
UC_26_4: **Purpose**: Allow user to progress forward in the game once they've won a battle encounter.
UC_26_5: **Overview**: When the user has initiated a fight sequence in a battle encounter and successfully eliminated all enemy units, they should be able to choose to return to the map scene and progress through the game.
UC_26_6: **Cross References**: US_23_1, US_23_2
UC_26_7: **Actors**: User
UC_26_8: **Pre-conditions**:
UC_26_Pre-1: Player has initiated a fight sequence.
UC_26_Pre-2: Player has successfully eliminated all enemy units.
UC_26_9: **Post Conditions**:
UC_26_Post_1: Player is returned to the map positioned at the node they entered.
UC_26_10: **Main flow**:
Player has successfully eliminated all enemy units.
Player is displayed a popup informing them the battle is won with a button to return to the map.
Player selects the button and is returned to the map positioned at the node they entered.
UC_25_11: **Alternative flow** of events:
Player initiates a fight sequence and successfully eliminates all enemy units.
Player is not displayed a popup informing them the battle is won with a button to return to the map.
Player selects the button; however they are not returned to the map scene.
Player selects the button and is returned to the map scene, however they are not returned to the node they entered.

UC_26_12: **Testing**:
When testing ensure all enemy units are destroyed before battle won popup is triggered. Ensure return to map button successfully returns the player to the map positioned at the battle node they entered.

UC_27

UC_27_1: **Use Case**: Battle lost
UC_27_2: **Author**: CP
UC_27_3: **Date**: 6/12/21
UC_27_4: **Purpose**: Allow user to replay a battle if they loose.
UC_27_5: **Overview**: When the user has initiated a fight sequence in a battle encounter and all their troops have been eliminated by the enemy units, they should be able to restart the battle and try again.
UC_27_6: **Cross References**: US_23_1, US_23_2
UC_27_7: **Actors**: User
UC_27_8: **Pre-conditions**:
UC_27_Pre-1: Player has initiated a fight sequence.
UC_27_Pre-2: Player has had all their units eliminated by the enemy units.
UC_27_9: **Post Conditions**:
UC_27_Post_1: The player returns to the troop placement sequence in the grid scene.
UC_27_10: **Main flow**:
The player initiates a fight sequence and all their units are eliminated by the enemy units.
The player is displayed a popup message informing they have lost the battle with an option to select and try again.
The player selects try again and is transferred back to the start of the unit placement sequence, with the same number of units they had preceding the battle.
The player can either replace their units and initiate a new fight sequence or return to the map.
UC_27_11: **Alternative flow** of events:
The player initiates a fight sequence and all their units are eliminated by the enemy units.
The player is not displayed a popup message informing them they have lost the battle with the option to select try again.
The player selects try again but is not transferred back to the start of the unit placement sequence.
The player is transferred back to the start of the unit placement sequence but no longer has the same number of units as they had preceding the battle.
UC_27_12: **Testing**:
When testing ensure that all player units have been destroyed before battle lost popup is triggered. Ensure the player is returned to the start of the unit placement sequence when the try again button is selected.

UC_28

UC_28_1: **Use Case**: Deploy ranged units
UC_28_2: **Author**: CP
UC_28_3: **Date**: 6/12/21
UC_28_4: **Purpose**: Gives the player gameplay and tactical variety by allowing them to deploy ranged units on the board to support their melee units.
UC_28_5: **Overview**: When the user is placing their units, they should be able to strategically place ranged units as tactical support for their melee units
UC_28_6: **Cross References**:
UC_28_7: **Actors**: User
UC_28_8: **Pre-conditions**:
UC_28_Pre-1: Player is in the grid scene in the unit placement phase.
UC_28_Pre-2: Player has one or more ranged units in their unit pool

UC_28_9: **Post Conditions**:

UC_28_Post_1: The player has strategically placed ranged units on the board to support their melee units.

UC_28_10: **Main flow**:

Once the troop placement scene is initiated, the player will see ranged units in their arsenal.

The player selects ranged units from their unit arsenal and places them where they see fit.

The player hits “fight” and the ranged units support the melee units by dealing ranged damage to enemy units.

The enemy units are able to detect the placed ranged units and will try to attack them along with the melee units.

UC_28_11: **Alternative flow** of events:

Once the troop placement scene is initiated, the player does not see ranged units in their arsenal.

After the player selects ranged units from their arsenal they are unable to place them on the board.

After the player hits “fight” the ranged units fail to deal damage to the enemy units from range.

The enemy units are unable to detect the placed ranged units and do not try to attack them.

UC_28_12: **Testing**:

When testing ensure pathfinding is adapted and functional for ranged units so that ranged units are able to engage opposing units and vice versa. Ensure that ranged units function correctly with the unit and grid managers. Ensure that Ranged units are able to deal and take damage successfully.

UC_29

UC_29_1: **Use Case**: Deal damage to non-engaged targets in range.

UC_29_2: **Author**: CP

UC_29_3: **Date**: 6/12/21

UC_29_4: **Purpose**: To enable melee units to damage targets other than the one they have engaged

UC_29_5: **Overview**: During the battle sequence, units should be able to damage opposing units which are in range but not necessarily engaged.

UC_29_6: **Cross References**:

UC_29_7: **Actors**: User

UC_29_8: **Pre-conditions**:

UC_29_Pre-1: Fight sequence has been initiated.

UC_29_Pre-2: One or more non-engaged opposing units are in range.

UC_29_9: **Post Conditions**:

UC_29_Post_1: Damage is dealt to a non-engaged unit.

UC_29_10: **Main flow**:

The player initiates the fight sequence.

One or more non-engaged opposing units come into range of a unit, while the engaged unit is out of range.

Damage is consistently dealt to a random non-engaged unit which is in range while the engaged unit is not in range.

UC_29_11: **Alternative flow** of events:

The player initiates the fight sequence.

No non-engaged enemy units come into range of a unit.

One or more enemy units come into range of a unit; however, the engaged unit is also in range so damage is consistently dealt to the engaged unit.

One or more enemy units come into range of a unit, the engaged unit is also in range; however the unit fails to deal damage to any of the opposing units in range.

One or more non-engaged opposing units come into range of a unit, while the engaged unit is out of range; however, the unit fails to deal damage to any of the opposing units in range.

UC_29_12: **Testing**:

When testing ensure that damage is randomly dealt to a target within range and this is only triggered when the engaged target is out of range. Ensure that when the engaged target is in range it is the only target which receives damage.

UC_30

UC_30_1: **Use Case**: Grid scene colour

UC_30_2: **Author**: CP

UC_30_3: **Date**: 6/12/21

UC_30_4: **Purpose**: To define the appearance of battle scenes for different areas of the map.

UC_30_5: **Overview**: The grid scene for battle nodes in different areas of the map should look different depending on what type of tile they're on e.g forest, water etc.

UC_30_6: **Cross References**: US_04_1, US_04_2, US_06_1, US_06_2, US_10_1, US_10_2

UC_30_7: **Actors**: Battle grid

UC_30_8: **Pre-conditions**:

UC_30_Pre-1: Player has entered a battle node on a distinctive tile.

UC_30_9: **Post Conditions**:

UC_30_Post_1: Battle grid is coloured representing the type of tile it is located at.

UC_30_10: **Main flow**:

The player enters a battle node on a distinctive tile, and transferred to the grid scene.

The grid scene appears tinted in the colour corresponding to the tile the battle node is located on

UC_30_11: **Alternative flow** of events:

The player enters a battle node on a distinctive tile, and transferred to the grid scene.

The grid scene does not appear tinted in the colour corresponding to the tile the battle node is located on

UC_30_12: **Testing**:

When testing ensure that the grid is given the correct coloured tint corresponding to the battle node it is located on.

UC_31

UC_31_1: **Use Case**: Unit health bar

UC_31_2: **Author**: CP

UC_31_3: **Date**: 6/12/21

UC_31_4: **Purpose**: To display the health points of each unit during the game

UC_31_5: **Overview**: During battle sequences, all units in the battle should have their health points displayed above the sprite in the form of a health bar.

UC_31_6: **Cross References**: US_22_1, US_22_2

UC_31_7: **Actors**: Unit, unit manager

UC_31_8: **Pre-conditions**:

UC_31_Pre-1: Battle sequence has been initiated

UC_31_9: **Post Conditions**:

UC_31_Post_1: each unit has a health bar displayed above their sprite.

UC_31_10: **Main flow**:

Player initiates fight sequence.

Each unit on the board is initialised with a health bar above their sprite.

Each time a unit takes damage, the respective amount of damage taken is deducted from their health bar.

UC_31_11: **Alternative flow** of events:

Player initiates fight sequences.

Each unit on the board fails to display a health bar above their sprite.

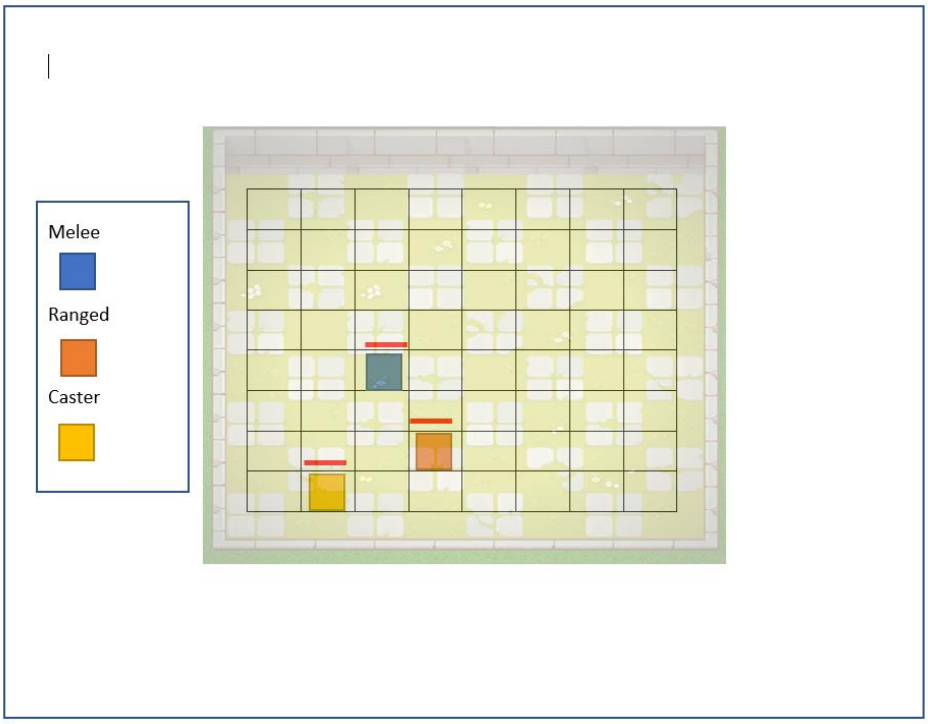
Each unit on the board successfully displays a health bar over their sprite; however, upon taking damage, the health bars fail to show the correct damage taken.

UC_31_12: **Testing**:

When testing ensure all units display a health bar. Ensure upon taking damage all units have the correct amount of health deducted from their health bars.

User interface design | 6.12.3

Auto-Battler UI



Elements

Unit Sprites - References **US_05_1, US_05_2, US_08_1, US_08_2, UC_03, Units, Renderer**

8x8 grid comprised of Tiles **US_08_01, US_08_02, US_11_01, US_11_02, US_14_01, US_14_01, US_15_01, US_15_02, UC_03, Tiles, Renderer, Grid Manager**

Unit selector on left side of window **US_5_1, US_5_2, US_8_1, US_8_2, US_13_1, US_13_2, US_15_1, US_15_2, UC_03, Renderer, Units**

Should indicate types available, and how many of each type remaining

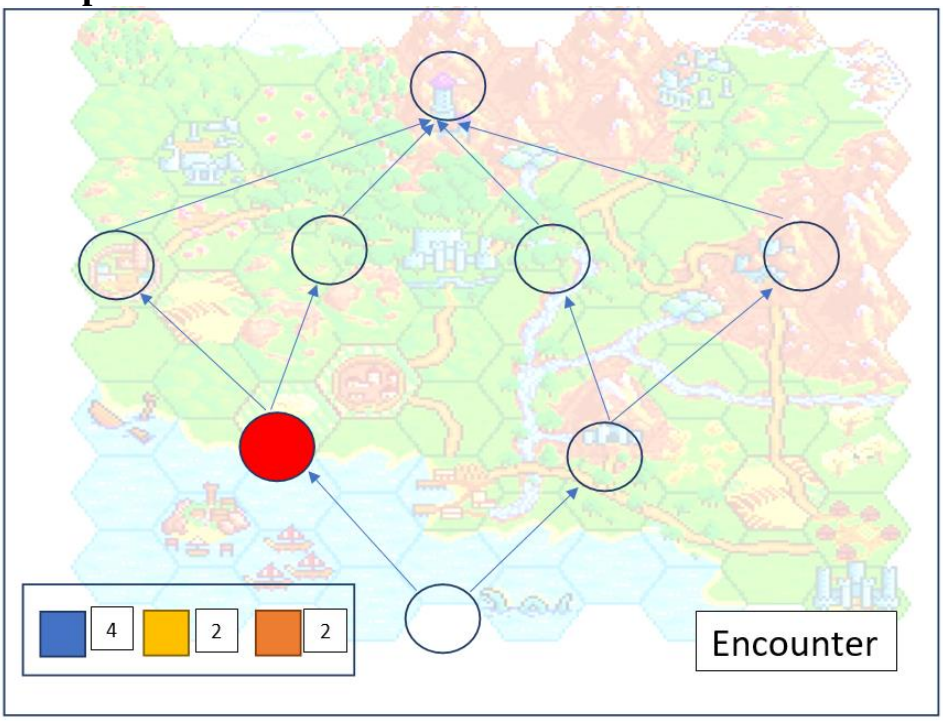
Appropriate methods implemented to enable clicking on unit type to select, and then clicking on Tile to place

Health bars above unit sprites **US_15_1, US_15_2**

Method should resize bar depending on current health

Background sprite should represent position of grid in game world, and should change depending on location.

Dungeon Map UI

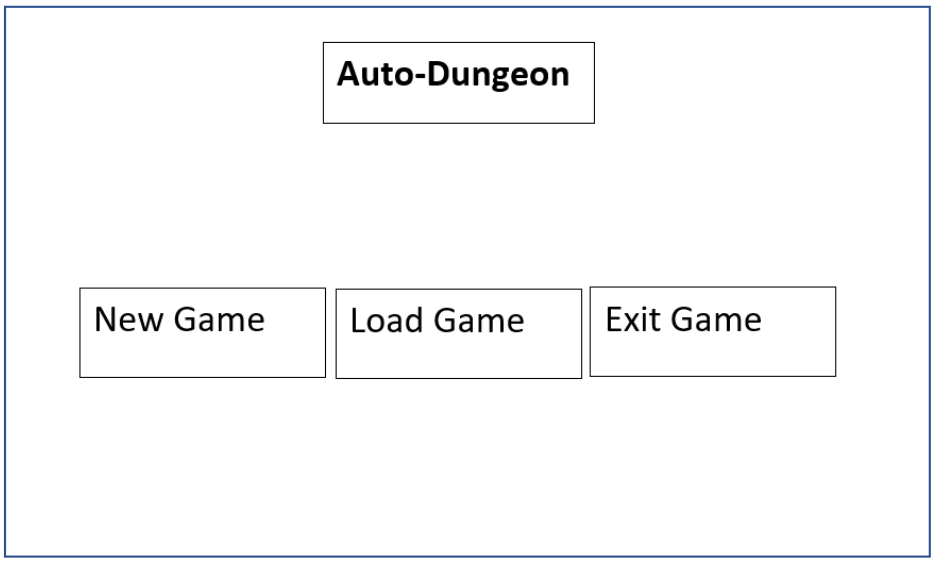


Elements

Tree Nodes **US_06_1, US_06_2, US_07_1, US_07_2, UC_04, Dungeon_Tree**
Should have method to change colour to indicate node currently occupied by party
Should have method to determine which encounter and then change scene
Unit's remaining indicator **US_07_1, US_07_2, US_09_1, US_09_2, US_11_1, US_11_2, US_14_1, US_14_2, US_15_1, US_15_2, Dungeon_Tree, Units**
Should visually show the number and composition of remaining units to aid decision making
Encounter button to enter encounter at current Node
Should have visual "overworld" map to help in telling underlying game story
Encounter difficulty should increase as game progresses

Main Menu UI

NOTE, MAIN MENU DEPRECATED, NOT IMPLEMENTED AT TIME OF DELIVERY DUE TO TIME CONSTRAINTS



Elements

New Game Button **US_03_1, US_03_2, UC_01**
Should have on-click method ready to start new game method
Load Game Button **US_02_1, US_02_2**
Should have on-click method ready to start load game method
Method should load available save game, display error message if no available save game
Exit Game Button **US_01_1, US_01_2, UC_02**
Should close game window and end all associated processes

Sprint 7: 10/12/21-17/12/21

The focus of sprint 7 was finishing touches to the game and uploading so that we could play test it as a real player would. The focus was also on collating our documentation.

Overview

What's supposed to happen this sprint

No.	Expected Start	Expected Finish	Objective
1	10/12/21	12/12/21	added a health scaling to enemy units to raise the difficulty
2	10/12/21	12/12/21	Complete the whole game map
3	10/12/21	12/12/21	Add colour tinting for the battles
4	10/12/21	13/12/21	Merge all branches and get all existing bugs fixed
5	10/12/21	14/12/21	Publish the game on the website
6	10/12/21	17/12/21	Complete the documentation, user guide, installation guide
7	10/12/21	17/12/21	Merge all documentation and get a final report

Reviews

What actually happened

No.	Started	Finished	Task	Doc
1	13/12/21	13/12/21	Daily scrum	7.1.1
	15/12/21	15/12/21	Customer meeting with answers analysed	7.2.1
2	15/12/21	15/12/21	Daily scrum	7.3.1
3	18/12/21	18/12/21	Daily scrum, sprint review, sprint retrospective, sprint planning	7.4.1
5	Throughout		Backlog updated	7.5.1
6	Throughout		Exception handling	7.6.1
9	Throughout		Requirements use cases	7.7.5
10	Throughout		CRC cards	7.8.5
11	Throughout		Design use cases	7.9.5
12	Throughout		User interface design	7.10.4

Daily scrum | 7.1.1

13/12/21, 13:00-13:15, Microsoft Teams

Attendance

Everyone present but NO
Minutes taken by QZ
Meeting led by DD

Minutes

1. Within the last 48 hours:

    a. DD changed the health and damage of all the units, add update function to the unitpool, added a health scaling to enemy units.
    b. TW finished the whole map and pushed it to Github, added colour tinting for the battles in each zone and pushed that on top of Drew's changes in the TestMerge branch.
    c. NO almost done with the maintenance doc and have completed the spring 5 doc
    d. QZ fixed the pathfinding issue and pushed the newest branch to Github.
    e. LW Resolved merge conflict using UnityYAMLMerge tool - should've used the tool right from the start

2. Issues that arose within the last 48 hours:

    a. Figuring out how to make it work post integration
    b. Managing all the different workloads

3. Within the next 48 hours:

    a. NO will be working on the survey documentation.
    b. All team member will keep on doing their documentation.

4. Foreseeable issues within the next 48 hours are:

    a. LW might not be able to come in for customer meeting.

Customer meeting | 7.2.1

15/11/21, 10:30-10:45, CB 5.13

Attendance

All present apart from TW and most members joined virtually
Recording taken by DD
Minutes taken by DD

Minutes

1. During meeting with customer, presented current state of game.

2. Positive feedback from this, customer pleased with the novelty of the product.

3. Asked directly whether product in its current state meets minimum requirements, and confirmed that current product is more than meeting requirements.

4. When presenting product, also demonstrated capability of product to run in browser, and be hosted remotely, which again customer received very positively.

5. Noted that browser scaling works poorly on smaller phones, but perfectly on tablets or other larger devices, but customer very pleased regardless with this capability.

6. Customer did provide some notes on current state of game, and in particular felt that more information provided to the player before a battle (number of enemy units, formation etc) would help make the game feel more “fair” and help skill expression in more difficult sections.

7. Customer did however appreciate that implementing a new feature at this stage of the development cycle would not be productive, and agreed that it would be more than sufficient for product to be delivered in current state.

Analysis

1. Meeting on Wednesday with the TA’s and Julian. Julian pleased with the novelty of the product and capability of product to run in browser. Julian provide some notes on current state of game, and in particular felt that more information provided to the player before a battle (number of enemy units, formation etc) would help make the game feel more “fair” and help skill expression in more difficult sections.

Daily scrum | 7.3.1

15/12/21, 14:15-14:30, Microsoft Teams

Attendance

Everyone was present
Minutes taken by QZ
Meeting led by DD

Minutes

1. Within the last 48 hours:

    a. LW completed the CRC cards.
    b. LG finished sprint 4 documentation and user stories.
    c. NO finished sprint 5 documentation and was finishing maintenance doc.
    d. DD and CP successfully deployed game to webGL Unity hosting service.
    e. CP have been doing the design and requirement use cases.
    f. TW finished Sprint 2 and 3 documentation and was finishing touches to the map
    g. IL finished sprint 6 documentation and done user guide

2. Issues that arose within the last 48 hours:

    a. Too many courseworks from other unist were due this week, so the team have agreed to move the sprint meeting to 18/12/21

3. Within the next 48 hours:

    a. All team members will contintue completing their documentation.

End of sprint meeting | 7.4.1

18/12/21, 13:15-14:30, NH 2.17a

Attendance

Everyone present except for LG
Minutes taken by TW
Meeting led by TW

Daily scrum

1. The documentation is still in progress

Sprint review

1. User evaluation (ethics form, briefing doc/participant info sheet, or evaluation survey) in our documentation was not assigned to a person, Drew will be putting together some documentation for it.

2. We need to record a short video to demonstrate our game.

Sprint retrospective

1. We have been able to meet almost all of our use cases and user stories, and have produced a playable game. Most importantly, our customer pleased with this game, and felt that we had more than met his required functionalities. The focus of the final sprint will be to amalgamate the documentation completed thus far.

2. We reviewed the documented user stories, use cases etcetera, and made careful note of which were achieved and which were not. On reflection at times our use cases and user stories did diverge at times from those required by our customer, prior to achieving our minimum viable product. More focus on this may have enabled us to further expand on our core game loop and have a more polished product.

3. We also noted that we had placed less emphasis on our “dungeon” concept required by the customer, and would benefit from more storytelling within the game, both mechanically and explicitly. A compromise at this stage was to add story to the manual as an old school way to add substance to game without needing to code.

Sprint planning

1. DD to create updated UI document.

2. Lawrence to sort out github

3. Tim to check maintenace guide and feedback to Naomi

4. Callum to continue use cases for 4, 5, and 6

5. Tim will do some documentation stuff for sprint 8 (time permitting)

6. Tim will merge docs into final word doc in ascending order of sprint.

7. Tim to check user guide and feedback to Ieuan. When user guide done and uploaded, Tim to send link to Drew who will add to description of game on unity

8. Ieuan and Qinyuan to continue sprint 6 and 7 respectively if not already complete

Backlog | 7.5.1

Complete backlog tasks

1. Merge game into a usable web application
2. Fixing bugs within map and combat screen
3. Fixing path finding
4. added a health scaling to enemy units to raise the difficulty
5. Complete the whole game map
6. Complete the documentation, user guide, installation guide

Current backlog

1. Main menu
2. Map randomiser
3. Change the tilesets based on difficulty
4. Main menu music
5. Recording feature
6. Sound effects

Exception handling | 7.6.1

Integration issues

1. The integration problems were unexpected as we had expected that using Github would reduce these issues instead. The problems were resolved within 48 hours with manual integration being performed by Andrew Dundee.

Bugs

1. Minor bugs with the unit's pathfinding method. Units sometimes get stuck and do nothing until they die. Resolved within 48 hours by Qinyuan Zhuang. Resolved by setting the grid node to empty when destroying the unit.

Requirements use cases | 7.7.5

We were unable to complete UC_01, UC_02, UC_03, UC_04 and UC_32 as they proved to be challenging and as a group ran out of time.

UC_01:
Use Case: Load Game
Scope: Main menu
Level: User Goal
Context: Users can load a game progression by selecting the “load game” option from the main menu
Users are unable to select a “load game” option from the main menu
Frequency of occurrence: Any time the player enters the programme or wishes to load a game from the map scene
Open issues:

How to ensure a failsafe option to save game prompt in case player wishes to return to original game progression?

UC_02:
Use Case: Exit game
Scope: Main menu
Level: User Goal
Context: Users can exit the game by selecting the “exit game” option from the main menu
Users are unable to select “exit game” option from the main menu
Frequency of occurrence: Any time the player wishes to start exit the game.
Open issues:

How to ensure a failsafe option to save game prompt in case player forgets to save their progress before quitting the programme.

UC_03:
Use Case: Save Game
Scope: N/A
Level: User Goal
Context: Users can save a game by selecting the “save game” option from the main menu
Users are unable to select the “save game” option from the main menu
Frequency of occurrence: Any time the player wishes to save a game.
Open issues:

How do we store saves?
How many saves can be stored by the game?
How many characters can be used in naming the save?
Can we implement a failsafe in case two saves are given the same name?

UC_04:
Use Case: Main menu
Scope: N/A
Level: Subfunction
Context: Users can decide to save load or exit the game by entering the main menu
Users are unable to save load or exit the game as they cannot access the main menu
Frequency of occurrence: Any time the player enters the game or wishes to save, load or exit the game.
Open issues:

Do we display this as a popup or a scene?

UC_05: (COMPLETE)
Use Case: Popup displayed
Scope: N/A

Level: Subfunction
Context: Users are informed of a consequence when they make a game action.
Users fail to be informed of the consequence of a game action taken
Frequency of occurrence: Any time the player makes a game action.
Open issues:
How do we ensure consistent alignment of popups any time they are instantiated? (RESOLVED)
How do we set the informational text of each popup to be dependent on the action taken? (RESOLVED)
How do we ensure the background is still visible when a popup is triggered? (RESOLVED)

UC_06: (COMPLETE)
Use Case: Enter treasure node
Scope: N/A
Level: Subfunction
Context: Upon entering a treasure node, users are rewarded with extra units.
Upon entering a treasure node, users fail to be rewarded with extra units.
Frequency of occurrence: Any time the player enters a treasure node.
Open issues:
How is the number of units rewarded determined? (RESOLVED)
How do we link this to the update unit pool functionality? (RESOLVED)

UC_07: (COMPLETE)
Use Case: Welcome node
Scope: N/A
Level: Subfunction
Context: Users can select the welcome node to initiate progression through the map, and are informed of their objective.
Users can select the welcome node to initiate progression through the map but aren't informed of their objective
Frequency of occurrence: Any time the player enters the map scene of a new game.
Open issues:

UC_08: (COMPLETE)
Use Case: New Arena node
Scope: N/A
Level: Subfunction
Context: Users can select a subsequent new arena node and are informed they are progressing to a new arena.
Users can select a subsequent new arena node but aren't informed they're progressing to a new arena.
Frequency of occurrence: Any time the player can progress to a new arena.
Open issues:

UC_09: (COMPLETE)
Use Case: Game completion node
Scope: N/A
Level: Subfunction
Context: Users can select the game completion node to complete the game and are informed they have won.
Users can select the game completion node to complete the game but aren't informed they have won.
Frequency of occurrence: When the player has completed the final battle encounter of the game.
Open issues:

UC_10: (COMPLETE)
Use Case: Update unit pool
Scope: N/A
Level: User goal
Context: The number of units of the unit count and unit pool can be updated.
The number of units of the unit count and unit pool cannot be updated.
Frequency of occurrence: Every time the player is awarded units.
Open issues:
How do we ensure there is no conflict between the unit count number and the unit pool contents? (RESOLVED)

UC_11: (COMPLETE)
Use Case: Enter battle node
Scope: N/A
Level: Subfunction
Context: Users can select a battle node which initiates the grid scene.
Users can select a battle node but the grid scene fails to be initialised.
Frequency of occurrence: When the player has a battle node adjacent to their current node.
Open issues:
How do we trigger grid scene initialisation. (RESOLVED)
How do we ensure the grid scene initialised is the correct encounter for the current node (ie enemy difficulty etc) ? (RESOLVED)

UC_12: (COMPLETE)
Use Case: Enemy unit health scaling
Scope: N/A
Level: User goal
Context: Users can select the game completion node to complete the game and are informed they have won.
Users can select the game completion node to complete the game but aren't informed they have won.
Frequency of occurrence: When the player has completed the final battle encounter of the game.
Open issues:
How is scaling factor per node determined? (RESOLVED)

UC_13:
Use Case: Highlight tile (COMPLETE)
Scope: N/A
Level: Sub function
Context: When a user moves their cursor over a tile in the grid scene the tile is highlighted
When a user moves their cursor over a tile in the grid scene the tile remains unchanged.
Frequency of occurrence: Any time the user is in the grid scene and moves their cursor over a tile.
Open issues:
How do we identify which tile the cursor is over? (RESOLVED)
What colour should the tile be highlighted? (RESOLVED)

UC_14:
Use Case: Select highlighted tile (COMPLETE)
Scope: N/A
Level: User goal
Context: User can select a highlighted tile and the programme recognises the action
Users cannot select the highlighted tile
Frequency of occurrence: Whenever the cursor is above a tile and it is highlighted.
Open issues:

How do we ensure selection of the tile is linked to a distinct action? (RESOLVED)

UC_15: (COMPLETE)
Use Case: Place units on the grid
Scope: N/A
Level: User goal
Context: User can select a unit and place it on a grid tile
User can select a unit but cannot place it on a grid tile
Frequency of occurrence: Whenever the user is in the unit placement sequence.
Open issues:
How do we select a unit? (RESOLVED)
How do we ensure a unit is placed when the tile is selected? (RESOLVED)
How do we ensure the placed unit is deducted from the unit pool? (RESOLVED)

UC_16: (COMPLETE)
Use Case: Unit constant damage output
Scope: N/A
Level: Sub function
Context: During battle sequence, units will output damage consistently which can then be accessed by opposing units.
During battle sequence, units will output damage consistently but output can't be accessed by opposing units.
Frequency of occurrence: Every time a fight sequence is initiated, for every unit.
Open issues:
How do we determine damage output and frequency? (RESOLVED)
When should damage output be initiated? (RESOLVED)
How is each unit's damage output accessed by opposing units? (RESOLVED)

UC_17: (COMPLETE)
Use Case: Unit take damage
Level: Sub function
Context: During battle sequence, units will take damage if they are within range of an opposing unit
During battle sequence, units are unable to take damage when in range of an opposing unit.
Frequency of occurrence: Whenever a unit is within range of an enemy unit.
Open issues:
How do we determine when a unit is within range of another? (RESOLVED)
How does a unit access an opposing unit's damage output? (RESOLVED)
How is health deducted from the unit's health points variable? (RESOLVED)

UC_18: (COMPLETE)
Use Case: Unit death
Level: Sub function
Context: During battle sequence, when a unit's health subceeds zero it will be eliminated from the battle
During battle sequence, when a unit's health subceeds zero it fails to be eliminated from the battle
Frequency of occurrence: Whenever a unit's health subceeds zero.
Open issues:
How to we eliminate an object from the battle? (RESOLVED)
How do we give a unit a "dead" attribute? (RESOLVED)

UC_19: (COMPLETE)
Use Case: Play game in browser
Level: Summary
Context: Users can access the game through a web browser

Users cannot access the game through a web browser
Frequency of occurrence: Dependent on User
Open issues:
What hosting service do we upload the game to? (RESOLVED)
How do we make the game as accessible as possible? (RESOLVED)
What devices will be able to run the game online? (RESOLVED)

UC_20: (COMPLETE)
Use Case: Move on map
Level: User goal
Context: Users can select an adjacent node and progress through the map
Users are unable to select adjacent nodes.
Frequency of occurrence: Any time the user is in the map scene.
Open issues:
How do we show the current location of the user on the map? (RESOLVED)

UC_21: (COMPLETE)
Use Case: Return to map from grid
Level: User goal
Context: The user can select a button to return to the map scene from the grid scene.
The user is unable to return to the map scene from the grid scene.
Frequency of occurrence: Whenever the user wishes to return to the map from the grid scene.
Open issues:
How to we ensure the player is returned to the node they exited? (RESOLVED)

UC_22: (COMPLETE)
Use Case: Dismiss popup
Level: User goal
Context: Users are able to select a dismissal button which closes the popup
Users are able to select a dismissal button but the popup fails to close
Frequency of occurrence: Whenever a unit's health subceeds zero.
Open issues:
How to we destroy the popup object? (RESOLVED)
How do we change the dismissal button text depending on the trigger scenario? (RESOLVED)
Can we add a function to the dismissal button when required (e.g. a "try again" button for a battle lost popup)? (RESOLVED)

UC_23: (COMPLETE)
Use Case: Fight sequence initiation
Level: Sub function
Context: Users are able to initiate a fight sequence, spawning the enemy units, upon clicking a fight button
Users are unable to initiate a fight sequence upon clicking a fight button
Frequency of occurrence: When the user is satisfied with their unit placement for the battle encounter.
Open issues:
How do we initialise the fight sequence? (RESOLVED)
How to we initialise the enemy units? (RESOLVED)
How do we give the enemy units a random formation each time? (RESOLVED)

UC_24: (COMPLETE)
Use Case: Path-finder/ unit-engagement
Level: Sub function

Context: Units are able to locate the closest opposing unit, lock onto them and plot a course to meet them

Units are able to locate the closest opposing unit but unable to plot a course to meet them.

Frequency of occurrence: Constant during the fight sequence

Open issues:

How do we locate the closest opposing unit? (RESOLVED)

How is the path determined, specifically when there are many units on the board? (RESOLVED)

UC_25: (COMPLETE)

Use Case: 1v1 fight sequence

Level: Sub function

Context: When engaged units are in range they will deal damage to each other until one of them is eliminated.

When engaged units are in range they fail to deal damage to each other.

Frequency of occurrence: Any time two engaged units are within range.

Open issues:

How do we ensure after a unit has defeated another it searches for a new target? (RESOLVED)

UC_26: (COMPLETE)

Use Case: Battle won

Level: User goal

Context: Once all enemy units are eliminated from the board, the battle is terminated and the user is informed they have won.

Once all enemy units are eliminated from the board the battle fails to terminate.

Frequency of occurrence: Any time all enemy units have been eliminated during a fight sequence.

Open issues:

How do we identify when all enemy units have been destroyed? (RESOLVED)

How does the user return to the map scene once the battle is won? (RESOLVED)

UC_27: (COMPLETE)

Use Case: Battle lost

Level: Sub function

Context: When all a player's units are eliminated from the board, the battle is terminated and the user is informed they have lost the battle.

When all a player's units are eliminated from the board, the battle fails to be terminated.

Frequency of occurrence: Any time all player units have been eliminated during a fight sequence.

Open issues:

How do we identify when all player units have been destroyed? (RESOLVED)

How do we return to the unit placement sequence in order try the battle again? (RESOLVED)

UC_28: (COMPLETE)

Use Case: Deploy ranged units

Level: Sub function

Context: Users can deploy ranged units in support of their melee units

Users are unable to deploy ranged units.

Frequency of occurrence: Every time the user is in the unit placement sequence.

Open issues:

How will the pathfinding work with ranged units? (RESOLVED)

What is the optimal amount of damage they should be dealing? (RESOLVED)

How do we ensure that melee units are able to lock onto ranged units so that they are also under threat along with the melee troops? (RESOLVED)

What is the optimum ratio of ranged to melee units for a competitive battle? (RESOLVED)

Do we need to improve the enemy pathfinding system to ensure that they are engaged by enemy units? (RESOLVED)

UC_29: (COMPELTE)
Use Case: Deal damage to non-engaged targets in range.
Level: Sub function
Context: When a unit is engaged with an opposing unit, but the opposing unit is out of range, the unit will deal damage to another non-engaged units which are in range
When a unit is engaged with an opposing unit, but the opposing unit is out of range, the unit cannot deal damage to other non-engaged units which are in range
Frequency of occurrence: When the opposing unit of an engaged unit is out of range but there are non-engaged opposing units in range, constant during fight sequence.
Open issues:
How to we keep track of which units are in range? (RESOLVED)
How do we determine which enemy unit receives the damage if multiple are in range? (RESOLVED)

UC_30: (COMPLETE)
Use Case: Grid scene colour.
Level: Sub function
Context: The colour of the grid scene changes depending on which map tile the battle node is located on.
The colour of the grid scene does not change depending on which map tile the battle node is located on.
Frequency of occurrence: Every time a battle node is entered.
Open issues:
How do we determine what colour tile the node is on? (RESOLVED)
How do we change the colour of the grid scene depending on the tile colour? (RESOLVED)

UC_31: (COMPLETE)
Use Case: Unit health bar.
Level: Sub function
Context: During the fight sequence all units have a health bar displayed above their sprite which changes according to their current health points.
During the fight sequence all units have a health bar displayed above their sprite but it fails to change according to their health points.
Frequency of occurrence: Constant during every fight sequence, for every alive unit.
Open issues:
How do we link the health bar to the unit's health points and ensure it changes accordingly? (RESOLVED)
How do we display the health bar above the unit sprite? (RESOLVED)
How do we ensure the health bar moves with the unit sprite? (RESOLVED)

UC_32:
Use Case: Randomised map creation.
Level: User goal
Context: Each time a new game is initiated a random map is generated
Each time a new game is initiated the same map is generated.
Frequency of occurrence: Every time a new game is initiated.
Open issues:
How do we randomise node pathways such that a satisfactory map is always generated?

CRC cards | 7.8.5

GridManager **References – UC_03**	
RESPONSIBILITIES: **Responsible for creating and containing a data structure containing all Tiles of grid and their positions** **Responsible for making this data available publicly to other functions** **Responsible for making units move around on the grid using a pathfinding algorithm**	COLLABORATORS: GridGraph Tiles BaseUnit Renderer

GridGraph **References – UC_03**	
RESPONSIBILITIES: **Responsible for defining the logic of the graph structure underlying the board** **Responsible for assigning a node to each tile** **Responsible for implementing the pathfinding algorithm**	COLLABORATORS: GridManager Tiles

Renderer **References – UC_01, UC_02, UC_03, UC_04**	
RESPONSIBILITIES: **Responsible for drawing all active sprites onto the game window**	COLLABORATORS: Tiles BaseUnit Dungeon_Tree

Dungeon_Tree **References - UC_04**	
RESPONSIBILITIES: **Responsible for containing recursive tree structure of other trees and relationships between trees** **Responsible for containing sprite representation of Nodes as well as connections between Nodes** **Responsible for listening for mouse click then altering sprite of node clicked to indicate it is selected** **Responsible for containing the current node position of the party** **Responsible for ensuring directly related distal nodes are the only nodes which are able to be selected on click** **Responsible for providing method to change current position of party to newly selected node**	COLLABORATORS: Renderer

BaseUnit **References – US_05_1, US_05_2, US_08_1, US_08_2, UC_03**	
RESPONSIBILITIES: **Responsible for containing and updating unit position** **Responsible for having method available for other functions/classes to change Unit position** **Responsible for containing and updating health, attack speed, and damage** **Responsible for having method available for other functions/classes to "attack" surrounding units** **Responsible for removing the unit and freeing the node when a unit is dead** **Responsible for finding the nearest enemy unit and move towards it**	COLLABORATORS: UnitManager Renderer

UnitManager **References – US_05_1, US_05_2, US_08_1, US_08_2, UC_03**	
RESPONSIBILITIES: **Create new instances of the player's units** **Check whether the win or lose condition is met** **Place enemy units in random formations**	COLLABORATORS: BaseUnit GridManager Renderer

UnitSelector **References – UC_03**	
RESPONSIBILITIES: **Handle drag-and-drop action for all units** **Pass the drag-and-drop position to other classes/functions** **Update the units count when drag-and-drop actions are performed**	COLLABORATORS: GameManager GridManager UnitManager

UIManager **Reference – UC_01, UC_04**	
RESPONSIBILITIES: Responsible for providing pop ups on certain dungeon levels Responsible for providing item counter Responsible for providing asset counter	COLLABORATORS: MapManager

MapManager **Reference – UC_04**	
RESPONSIBILITIES: Responsible to generating the map Responsible for saving the current state of the map Responsible for saving the player's progress	COLLABORATORS: UIManager

HoverHighlight **References – UC_03**	
RESPONSIBILITIES: **Highlight the tile which the cursor is hovering over**	COLLABORATORS: GridManager GridGraph

HealthBar **References – UC_03, US_05_1**	
RESPONSIBILITIES: **Responsible for displaying a health bar for each unit on the board** **Responsible for taking in the max health and current health of each unit**	COLLABORATORS: UnitManager

Popup **References – US_14_1**	
RESPONSIBILITIES: **Show different messages and rewards for when the battle is won or lost**	COLLABORATORS: UnitManager GridManager

ObjectClick **References – US_14_1**	
RESPONSIBILITIES: **Create different lightings for active and inactive nodes on the map** **Use different lightings for different kinds of nodes on the map**	COLLABORATORS: UnitManager MapManager

PlayerController **References – UC_04, US_07_1, US_15_1**	
RESPONSIBILITIES: **Enable the player to pan their camera around the map** **Restrict the camera to be inside the map**	COLLABORATORS: MapManager CameraPosition

CameraPosition **References - UC_04, US_07_1, US_15_1**	
RESPONSIBILITIES: **Keep track of the player's progress through the map** **Return the camera to the furthest node after a battle is finished**	COLLABORATORS: MapManager PlayerController

GridLight **References – UC_04, US_07_1, US_15_1**	
RESPONSIBILITIES: **Apply different color tints in grid base on how far the player is in the map**	COLLABORATORS: MapManager GridManager

Design use cases | 7.9.5

UC_01

UC_01_1: **Use Case**: Load game
UC_01_2: **Author**: DD
UC_01_3: **Date**: 15/11/21
UC_01_4: **Purpose**: Initiate new game from within main menu
UC_01_5: **Overview**: Starts when User presses load game button on main menu screen. On button press the player should be prompted to select a game save with one of the options being a new game. The selected game instance should begin, with scene either in dungeon map or in hub depending on program progress.
UC_01_6: **Cross References**: US_03_1, US_03_2
UC_01_7: **Actors**: User
UC_01_8: **Pre-conditions**:
UC_01_Pre-1: Program should be in main menu scene
UC_01_Pre-2: load game button must be "listening" for click event with prepared new game and load game functions actioned on-click
UC_01_9: **Post Conditions**:
UC_01_Post_1: Scene should be changed to a dungeon map with the respective progression selected
UC_01_10: **Main flow of events**:
User Starts the game and enters the main menu scene.
User correctly selects the load game button which prompts the user to select a save.
User selects a save and is transferred to the dungeon scene with map progression corresponding to the respective save selected.
UC_01_11: **Alternative flow of events**:
If user clicks outside of area of button actively listening for input, no action taken.
User selects the load game button and selects a save but is transferred to a new dungeon map scene with the incorrect progression loaded.
In the case of a failure of this process with user unable to begin new game instance, product not viable, if not rectified would necessitate removing main meu scene.
UC_01_12: **Testing**:
In testing this function, ensure that on mouse click, the new game button calls the new game function, and that the scene then changes appropriately to the hub/dungeon depending on the stage of development.

UC_02

UC_02_1: **Use Case**: Exit Game
UC_02_2: **Author**: DD
UC_02_3: **Date**: 15/11/21
UC_02_4: **Purpose**: End game process from within game window
UC_02_5: **Overview**: Starts when User presses quit game button on main menu screen. Button should use built in Unity application.quit() method to end game process.
UC_02_6: **Cross References**: US_01_1, US_01_2
UC_02_7: **Actors**: User
UC_02_8: **Pre-conditions**:
UC_02_Pre-1: Program should be in main menu scene
UC_02_Pre-2: Exit button must be "listening" for click event with prepared quit game function actioned on-click
UC_02_9: **Post Conditions**:
UC_02_Post_1: The program and all related processes must end.

UC_02_10: **Main flow of events**:

User enters the main menu scene and correctly selects the exit game button.

The programme successfully calls the application.quit() method and is terminated.

UC_02_11: **Alternative flow** of events:

If user clicks outside of area of button actively listening for input, no action taken.

In the case of a failure of this process the User can instead use system services to close the game, through the keyboard alt-F4 shortcut or through the windows task manager.

UC_02_ 12: **Testing**: When testing, ensure that on mouse click, the exit game button calls the exit game function.

UC_03

UC_03_1: **Use Case**: Save Game
UC_03_2: **Author**: CP
UC_03_3: **Date**: 15/11/21
UC_03_4: **Purpose**: Save game progress from within game window
UC_03_5: **Overview**: Starts when User presses save game button on main menu screen. This should save the player's progress through the map where the user has the option to save the game under a name.
UC_03_6: **Cross References**: US_02_1, US_02_2
UC_03_7: **Actors**: User
UC_03_8: **Pre-conditions**:
UC_03_Pre-1: Program should be in main menu scene
UC_03_Pre-2: Save button must be "listening" for click event with prepared save game function actioned on-click
UC_03_9: **Post Conditions**:
UC_03_Post_1: The program saves and stores current game progress under the player's name.
UC_03_10: **Main flow**:

When the player is in the main menu and decides they want to save their progress they select the save game button and the player is prompted to enter their name.

The player enters their name and selects okay.

The player sees their name listed under saved games.

UC_02_11: **Alternative flow** of events:

The player selects the save game button but is not prompted to enter their name.

After the player enters their name, they cannot see it listed under saved games.

UC_02_12: **Testing**:
When testing ensure the game is saved with the correct name entered and when the game is re-opened the user is at the same map position with the same number of troops as when they left the game.

UC_04

UC_04_1: **Use Case**: Main Menu
UC_04_2: **Author**: CP
UC_04_3: **Date**: 15/11/21
UC_04_4: **Purpose**: Interface for saving and exiting the game
UC_04_5: **Overview**: The user should be able to select a main menu option from the map scene from which they can choose to load a game, save or exit the game. User should also be met by the main menu upon entry to the game.
UC_04_6: **Cross References**: US_02_1, US_02_2
UC_04_7: **Actors**: User
UC_04_8: **Pre-conditions**:
UC_04_Pre-1: Program should be in map scene or have just been run
UC_04_9: **Post Conditions**:

UC_04_Post_1: The program displays the main menu with all its options.
UC_04_10: **Main flow**:

When the player is in the map scene, they select the main menu option, triggering the main menu screen.
The main menu is displayed with options to load game, save game or exit game
When the player enters the game they are met by the main menu scene displaying options to load game, save game or exit game

UC_04_11: **Alternative flow** of events:

The player selects the main menu button but the main menu scene is not triggered.
After the player enters the game they are not met by the main menu screen and instead transferred directly to the dungeon map scene.

UC_04_12: **Testing**:
When testing ensure the main menu screen is triggered upon selecting the main menu button. Ensure that upon entry to the game the player is met by the main menu screen.

UC_05
UC_05_1: **Use Case**: Popup displayed
UC_05_2: **Author**: CP
UC_05_3: **Date**: 15/11/21
UC_05_4: **Purpose**: Inform player of a game consequence.
UC_05_5: **Overview**: When the user takes an action and a consequence is reached, they should be informed of the resulting consequence.
UC_05_6: **Cross References**: US_07_01, US_07_02, US_15_01, US_15_02.
UC_05_7: **Actors**: User interface.
UC_05_8: **Pre-conditions**:

UC_05_Pre-1: Player has made an action in the game.
UC_05_Pre-2: The action taken has a consequence.

UC_05_9: **Post Conditions**:

UC_04_Post_1: The player understands the consequence of the action taken.

UC_05_10: **Main flow**:

The player makes an action in the game (e.g. enters a treasure node)
The popup is displayed clearly on the scene, with the background still visible
The player understands the consequence of the action taken (e.g. has been awarded 4 units).

UC_05_11: **Alternative flow** of events:

The player makes an action in the game (e.g. enters a treasure node)
The popup is not displayed clearly on the scene.
The popup is displayed clearly on the scene; however, the background is not visible.
The popup is displayed clearly on the scene but the player does not understand the consequence of the action taken.

UC_05_12: **Testing**:
When testing ensure that the popup is correctly aligned in the centre of the screen and the background scene is still visible to the user. Ensure that a situationally required message can be displayed dependent on the action triggering the popup.

UC_06
UC_06_1: **Use Case**: Enter treasure node.
UC_06_2: **Author**: CP
UC_06_3: **Date**: 15/11/21
UC_06_4: **Purpose**: Informs user of their treasure reward units
UC_06_5: **Overview**: When a user selects a treasure node they should be greeted by a popup informing them of how many troops they have gained as a reward.

UC_06_6: **Cross References**: US_07_1, US_07_2, US_09_1, US_09_2, US_10_1, US_10_2,
UC_06_7: **Actors**: User, User interface
UC_06_8: **Pre-conditions**:
UC_06_Pre-1: Programme should be on map scene
UC_06_Pre-2: Treasure node is adjacent to current node.
UC_06_9: **Post Conditions**:
UC_06_Post_1: New occupied node should now be the treasure node
UC_06_Post_2: A dismissible popup should now displayed on the screen informing the user of troop reward
UC_06_Post_3: Unit pool should be updated to reflect additional troops gained.
UC_06_10: **Main flow**:
User sees a treasure node adjacent to current node and selects it.
After the treasure node is selected, a dismissible popup is displayed informing the user of their reward.
After the popup is dismissed the unit pool is updated according to the number of troops awarded.
UC_06_11: **Alternative flow** of events:
User sees a treasure node adjacent to current node, but cannot select it.
After selecting the treasure node, the programme fails to display a popup message.
After selecting a treasure node, a popup is displayed showing the chest reward, but the user is unable to dismiss it.
After the popup is dismissed, the unit pool is not updated to show the extra units awarded.
UC_06_12: **Testing**:
When testing, ensure selection of the treasure node is recognised by the programme and this can be used to trigger a popup. Ensure the popup can successfully be dismissed and that the popup management system updates the unit manager to reflect the additional units gained so that this is displayed in the unit pool.

UC_07
UC_07_1: **Use Case**: Welcome node
UC_07_2: **Author**: CP
UC_07_3: **Date**: 15/11/21
UC_07_4: **Purpose**: Introduce the player to the game.
UC_07_5: **Overview**: Upon entering the game the player should be able to select the starting node where a welcome message is displayed, welcoming the player to the game and informing them of the point they must reach to complete it.
UC_07_6: **Cross References**: US_03_1, US_03_2
UC_07_7: **Actors**: User, User Interface
UC_07_8: **Pre-conditions**:
UC_07_Pre-1: The game has been started and the map scene initiated.
UC_07_Pre-2: No nodes have been clicked and activated yet.
UC_07_9: **Post Conditions**:
UC_07_Post_1: A dismissible popup should be displayed welcoming the player and informing them of the game endpoint (star)
UC_07_10: **Main flow**:
The player enters the game and clicks on the first node on the map.
A dismissible popup is displayed welcoming the player to the game and informing them they must reach the star node at the end of the map.
Popup is dismissed and welcome node is now darkened indicating it is the user's current position on the map.
UC_07_11: **Alternative flow** of events:

The player enters the game and clicks on the first node.
A dismissible popup fails to be displayed to welcome the player or inform them of the objective.
A dismissible popup is created but it is not visible on the screen and therefore cannot be dismissed.
Popup is dismissed but the welcome node is still selectable indicating the user hasn't taken position at the welcome node yet.

UC_07_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct welcome message. Ensure node is darkened and no longer selectable after dismissal.

UC_08

UC_08_1: **Use Case**: New arena node
UC_08_2: **Author**: CP
UC_08_3: **Date**: 15/11/21
UC_08_4: **Purpose**: Transfers the player to the first node of a new arena and informs them they've completed the previous arena.
UC_08_5: **Overview**: When a player have taken a path through an arena and made it to the end, they should be able to select a new arena node which informs them they've completed the previous arena. The player can then choose between different paths through the new arena.
UC_08_6: **Cross References**: US_06_1, US_06_2
UC_08_7: **Actors**: User
UC_08_8: **Pre-conditions**:
UC_08_Pre-1: Player has reached and completed one of the final nodes of their current arena.
UC_08_Pre-2:
UC_08_9: **Post Conditions**:
UC_08_Post_1: Player has the option to select from different starting nodes through the new arena with different paths and encounters.
UC_08_10: **Main flow**:
The player selects the new arena node.
A popup is displayed informing the player that they can now enter a new arena.
The player is transferred to the new arena node and can now select a path through the new arena.
UC_08_11: **Alternative flow** of events:
The player is unable to select the new arena node
The player selects the new arena node but a popup informing the player they can now enter a new arena fails to be displayed.
The player fails to be transferred to the new arena node and therefore cannot select a path through the new arena.
UC_08_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct enter new arena message displayed. Ensure node is darkened and no longer selectable after dismissal.

UC_09

UC_09_1: **Use Case**: Game completion node
UC_09_2: **Author**: CP
UC_09_3: **Date**: 15/11/21
UC_09_4: **Purpose**: Initiates and informs user of completion of the game.
UC_09_5: **Overview**: There should be a final node on the map, whose reaching is the objective of the game. Upon entry, the player should be informed they have completed the game and congratulated.

UC_09_6: **Cross References**: US_01_1, US_01_2
UC_09_7: **Actors**: User
UC_09_8: **Pre-conditions**:
UC_09_Pre-1: The player has successfully completed the last battle node of the game.
UC_09_Pre-2: The player has returned to the map scene.
UC_09_9: **Post Conditions**:
UC_09_Post_1: The game is complete and a message is shown informing the user they have won.
UC_09_10: **Main flow**:
After completing the final battle encounter the player has returned to the map scene.
The player selects the final node on the map.
A dismissible popup is initiated, with a message congratulating the user for winning the game.
UC_09_11: **Alternative flow** of events:
After completing the final battle encounter the player has returned to the map scene.
The player selects the final node on the map.
A dismissible popup fails to be initiated and the player is not informed they have won the game.
UC_09_12: **Testing:**
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct congratulatory message displayed. Ensure node is darkened and no longer selectable after dismissal.

UC_10
UC_10_1: **Use Case**: Update unit pool
UC_10_2: **Author**: CP
UC_10_3: **Date**: 15/11/21
UC_10_4: **Purpose**: Reflects additional troops awarded to the player
UC_10_5: **Overview**: When a player is awarded units from a treasure node, the unit pool should be updated to show the new total number of units.
UC_10_6: **Cross References**: US_06_1, US_06_2
UC_10_7: **Actors**: User
UC_10_8: **Pre-conditions**:
UC_10_Pre-1: Player has selected a treasure node.
UC_10_Pre-2: Unit reward has been determined.
UC_10_9: **Post Conditions**:
UC_10_Post_1: Unit pool should reflect the units awarded on top of the units in possession preceding node selection
UC_10_10: **Main flow**:
User selects a treasure node and is informed of unit reward.
Unit pool is updated to show the sum of the player's previous number of units and the number of units awarded.
UC_10_11: **Alternative flow** of events:
User selects a treasure node and is informed of unit reward.
Unit pool is not updated and remains the same as before node entry.
Unit pool is updated; however the new value is not the sum of the player's previous number of units and the number of units awarded.
UC_10_12: **Testing**:
When testing ensure popup displayed is visible, dismissible and aligned correctly on the screen, with the correct unit award message displayed. Ensure that the unit count and unit pool are updated to reflect the extra troops rewarded.

UC_11

UC_11_1: **Use Case**: Enter battle node
UC_11_2: **Author**: CP
UC_11_3: **Date**: 15/11/21
UC_11_4: **Purpose**: Allow user to enter a battle node to progress through the map.
UC_11_5: **Overview**: When the user is in the map scene and the adjacent node is a battle node, they should be able to select the node and be transferred to the respective level's grid scene.
UC_11_6: **Cross References**: US_04_1, US_04_2, US_08_1, US_08_2, US_09_1, US_09_2, US_10_1, US_10_2
UC_11_7: **Actors**: User
UC_11_8: **Pre-conditions**:
UC_11_Pre-1: Programme is on map scene.
UC_11_Pre-2: These is a battle node adjacent to the current node.
UC_11_9: **Post Conditions**:
UC_11_Post_1: The player is transferred to grid scene
UC_11_10: **Main flow**:
The player is currently at a node with a battle node adjacent.
The player selects the battle node.
Upon selecting the battle node, the player is transferred to the grid scene.
UC_11_11: **Alternative flow** of events:
The player selects the battle node; however, they are unable to select the battle node.
The player selects the battle node; however, they are not transferred to the grid scene.
UC_11_12: Testing:
When testing ensure that the grid scene is loaded successfully with the unit pool of the unit placement scene reflecting the number of units in the unit count of the map scene.

UC_12

UC_12_1: **Use Case**: Enemy unit health scaling
UC_12_2: **Author**: CP
UC_12_3: **Date**: 15/11/21
UC_12_4: **Purpose**: To add difficulty to battle encounters as the player progresses through the game
UC_12_5: **Overview**: As the player progresses through the game, the health of enemy units should increase at each battle encounter.
UC_12_6: **Cross References**: US_09_1, US_09_2, US_12_1, US_12_2
UC_12_7: **Actors**: User
UC_12_8: **Pre-conditions**:
UC_12_Pre-1: Player has completed at least one battle encounter.
UC_12_9: **Post Conditions**:
UC_12_Post_1: The enemy units encountered at the subsequent battle node have more health points than those of the previous node.
UC_12_10: **Main flow**:
The player completes a battle encounter.
The player enters a subsequent battle node and encounters enemy units with one more health point than those of the previous node.
UC_12_11: **Alternative flow** of events:
The player completes a battle encounter.
The player enters a subsequent battle node; however, the enemy units encountered have the same amount of or less health points than those of the previous node.
UC_12_12: **Testing**:
When testing ensure that the enemy units in each subsequent battle node have one extra health point.

UC_13

UC_13_1: **Use Case**: Highlight tile
UC_13_2: **Author**: CP
UC_13_3: **Date**: 15/11/21
UC_13_4: **Purpose**: Highlights the tile which the user's cursor is interacting with at its current position.
UC_13_5: **Overview**: During the grid scene, the player should be able to see which tile their cursor is interacting with according to its position, where the current tile is highlighted
UC_13_6: **Cross References**: US_05_1, US_05_2, US_13_1, US_13_2
UC_13_7: **Actors**: User
UC_13_8: **Pre-conditions**:
UC_13_Pre-1: The programme should be in the grid scene, and the board displayed
UC_13_9: **Post Conditions**:
UC_13_Post_1: When the cursor moves over a tile, the tile is highlighted.
UC_13_10: **Main flow**:
During the grid scene the player moves the cursor over a specific tile.
The tile the cursor is positioned over is highlighted.
UC_13_11: **Alternative flow** of events:
During the grid scene the player moves the cursor over a specific tile.
The tile the cursor is positioned over is highlighted along with one or more other tiles which the cursor is not positioned over.
The tile the cursor is positioned over is not highlighted; however one or more other tiles which the cursor is not positioned over are highlighted.
The tile the cursor is positioned over is not highlighted nor is any other tile on the board.
UC_13_12: **Testing**:
When testing ensure that only the tile the cursor is hovering over is highlighted and this is reflected as the cursor is moved around the grid.

UC_14

UC_14_1: **Use Case**: Select highlighted tile
UC_14_2: **Author**: CP
UC_14_3: **Date**: 15/11/21
UC_14_4: **Purpose**: To enable action to be performed upon selection of a tile.
UC_14_5: **Overview**: During the grid scene, the player should be able to select a highlighted tile which the cursor is positioned over and the programme should recognise this selection enabling an action to be performed.
UC_14_6: **Cross References**: US_08_01, US_08_02, US_05_1, US_05_2, US_13_1, US_13_2
UC_14_7: **Actors**: User
UC_14_8: **Pre-conditions**:
UC_14_Pre-1: The programme should be in the grid scene
UC_14_Pre-2: The user should have the cursor positioned over the highlighted tile which they wish to select.
UC_14_9: **Post Conditions**:
UC_14_Post_1: The tile is selected, turning grey and the action is recognised by the programme.
UC_14_10: **Main flow**:
During the grid scene the player moves the cursor over a specific tile, which is highlighted.
The cursor changes to a hand symbol letting the player know its clickable,
The player selects the tile, which turns grey, meaning the programme recognises the action.
UC_14_11: **Alternative flow** of events:
During the grid scene the player moves the cursor over a specific tile, which is highlighted

The cursor remains unchanged, so they do not know its clickable.
The cursor changes to a hand symbol, letting the player know its clickable, the player selects the tile; However, the tile does not change colour, meaning the programme does not recognise the action.

UC_14_12: **Testing**:
When testing ensure that it is only the highlighted tile which is selected and the programme recognises the action.

UC_15

UC_15_1: **Use Case**: Place units on grid
UC_15_2: **Author**: DD
UC_15_3: **Date**: 15/11/21
UC_15_4: **Purpose**: Allow user to place units on grid before battle
UC_15_5: **Overview**: Starts during placement phase of auto-battler encounter. Player should be able to select from persistent pool of units, visible on the lower edge of the screen in a relevant UI object, and these can then be placed in empty hexes on the board on the lower half of the game board.
UC_15_6: **Cross References**: US_08_01, US_08_02, US_11_01, US_11_02, US_14_01, US_14_01, US_15_01, US_15_02
UC_15_7: **Actors**: User
UC_15_8: **Pre-conditions**:
UC_15_Pre-1: Program should be within encounter scene
UC_15_Pre-2: Current state of encounter should be planning
UC_15_Pre-3: User must have units available to place
UC_15_Pre-4: At least one square on board should not be full
UC_15_Pre-5: Current units placed must be less than maximum placeable units
UC_15_9: **Post Conditions**:
UC_15_Post_1: Selected unit should be placed on the game board
UC_15_Post_2: The Square on the board the unit now occupies should no longer be empty
UC_15_Post_3: One copy of the selected unit should be removed from the unit pool temporarily
UC_15_10: **Main flow of events**:
Player Enters the battle node initiating the placement phase of the battle encounter.
Player select a unit from their unit pool.
Unit is placed on one of the empty hexes on the board.
UC_15_11: **Alternative flow** of events:
User tries to place unit on non-empty square, explanatory error message is shown
User tries to place unit from empty pool of units, explanatory error message is shown
UC_15_12: Testing:
When testing, ensure unit clicked becomes selected. Also ensure when subsequently clicking on tile that a new unit of the selected type is instantiated, with a position equal to the selected tile.

UC_16

UC_16_1: **Use Case**: Unit constant damage output
UC_16_2: **Author**: CP
UC_16_3: **Date**: 15/11/21
UC_16_4: **Purpose**: Allows units to deal damage to opposing units.
UC_16_5: **Overview**: Units consistently output damage with amount and rate determined by the unit type.
UC_16_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2
UC_16_7: **Actors**: User
UC_16_8: **Pre-conditions**:

UC_16_Pre-1: Units are placed on the board
UC_16_Pre-2: Fight sequence is initiated

UC_16_9: **Post Conditions**:
UC_16_Post_1: Damage is consistently output while the unit is not dead.

UC_16_10: **Main flow**:
The fight sequence is initiated and all units on the board begin outputting damage.
Damage is consistently output until the unit is dead.

UC_16_11: **Alternative flow** of events:
The fight sequence is initiated; however, some or all the units do not begin outputting damage.
Damage is consistently output by each unit, even after the unit is dead.

UC_16_12: **Testing**:
When testing ensure the unit damage output is accessible to other units so that they can use it to take damage.

UC_17

UC_17_1: **Use Case**: Unit take damage
UC_17_2: **Author**: CP
UC_17_3: **Date**: 15/11/21
UC_17_4: **Purpose**: Allows units to take damage from other units
UC_17_5: **Overview**: When a unit is in range of an opposing unit, it should be dealt damage corresponding to the opposing unit's damage output.
UC_17_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2
UC_17_7: **Actors**: User
UC_17_8: **Pre-conditions**:
UC_17_Pre-1: The unit has been engaged by an opposing unit.
UC_17_Pre-2: The unit is in range of the opposing unit.

UC_17_9: **Post Conditions**:
UC_17_Post_1: The unit health is consistently reduced according to the opposing unit's damage output.

UC_17_10: **Main flow**:
The unit comes into range of an opposing unit which has engaged it.
The opposing unit begins consistently dealing damage to the unit while it is in range.
The health variable of the unit is consistently reduced according to the opposing unit's damage output.

UC_17_11: **Alternative flow** of events:
The unit comes into range of the opposing unit which has engaged it.
The opposing unit fails to deal damage to the unit
The opposing unit
The health attribute of a unit engaged in a 1v1 battle subceeds zero.
The dead Boolean of the unit object fails to be set to true.

UC_17_12: **Testing**:
When testing ensure the damage output of the unit dealing the damage is accessible to the one taking it. Ensure a corresponding amount of health points are deducted from the unit's health variable.

UC_18

UC_18_1: **Use Case**: Unit death
UC_18_2: **Author**: CP
UC_18_3: **Date**: 15/11/21
UC_18_4: **Purpose**: Criteria for the elimination of a unit from a battle.

UC_18_5: **Overview**: When the health of a unit has reached or subceeded zero, the unit is given the dead attribute, which can then be used to disengage the opposing unit and identify unit objects which need to be destroyed.
UC_18_6: **Cross References**: US_05_1, US_05_2, US_08_1, US_08_2
UC_18_7: **Actors**: User
UC_18_8: **Pre-conditions**:
UC_18_Pre-1: The unit in question has been engaged in a 1v1 battle.
UC_18_Pre-2: The health of the unit is less than or equal to zero.
UC_18_9: **Post Conditions**:
UC_17_Post_1: The unit is given the dead attribute which is a Boolean set to true.
UC_18_10: **Main flow**:
The health attribute of a unit engaged in a 1v1 battle subceeds zero.
The "dead" Boolean of the unit object is set to true.
The "dead" unit object is destroyed, eliminating it from the battle
UC_18_11: **Alternative flow** of events:
The health attribute of a unit engaged in a 1v1 battle subceeds zero.
The dead Boolean of the unit object fails to be set to true.
The dead Boolean of the object is set to true but it is not destroyed, failing to eliminate it from the game.
UC_18_12: **Testing**:
When testing ensure that the unit object is successfully destroyed upon its dead Boolean being set to true

UC_19
UC_19_1: **Use Case**: Play game in browser
UC_19_2: **Author**: CP
UC_19_3: **Date**: 15/11/21
UC_19_4: **Purpose**: Allow players to play game in a browser across multiple devices.
UC_19_5: **Overview**: The game should be playable in a browser.
UC_19_6: **Cross References**: US_16_1, US_16_2
UC_19_7: **Actors**: User
UC_19_8: **Pre-conditions**:
UC_19_Pre-1: The game is fully functional.
UC_19_9: **Post Conditions**:
UC_19_Post_1: The game is playable on a web browser.
UC_19_10: **Main flow**:
The game is uploaded to unity free hosting service and embedded in a web page.
The game is playable online
UC_19_11: **Alternative flow** of events:
The game fails to be uploaded to unity free hosting service
UC_19_12: **Testing**:
When testing ensure that the game is playable across a range of devices on a range of web browsers.

UC_20
UC_20_1: **Use Case**: Move on Map
UC_20_2: **Author**: DD
UC_20_3: **Date**: 22/11/21
UC_20_4: **Purpose**: Allow user to change current location to adjacent node to progress through map
UC_20_5: **Overview**: When user is in the map scene, they should be able to select any of the adjacent nodes on the map, and their current position should then be changed to this node. This should also be visually indicated on the scene

UC_20_6: **Cross References**: US_05_1, US_05_2, US_06_1, US_06_2, US_07_1, US_07_2, US_13_1, US_13_2, US_19_1, US_19_2
UC_20_7: **Actors**: User
UC_20_8: **Pre-conditions**:
UC_20_Pre-1: Program should be in map scene
UC_20_Pre-2: Current party position should not be at end node
UC_20_9: **Post Conditions**:
UC_20_Post_1: Current party position should change to selected node
UC_20_Post_2: The visual indicator for party position should change to selected node
UC_20_Post_3: Event for new node position should be triggered
UC_20_10: **Main flow**:
UC_20_11: **Alternative flow** of events:
If a user selects a node proximal in the tree, the party position should remain at the current node
If a user selects the currently occupied node, the party position should remain at the current node
UC_20_12: **Testing**: When testing, ensure only nodes distal and adjacent to active nodes are selectable. Also ensure that when selectable nodes are clicked, the party position then changes to this current node.

UC_21
UC_21_1: **Use Case**: Dismiss Popup
UC_21_2: **Author**: CP
UC_21_3: **Date**: 22/11/21
UC_21_4: **Purpose**: Allow user to Dismiss a popup from the screen.
UC_21_5: **Overview**: When a popup is displayed, the user should be able to easily select a dismiss button to close it.
UC_21_6: **Cross References**: US_15_1, US_15_2, US_20_1, US_20_2
UC_21_7: **Actors**: User
UC_21_8: **Pre-conditions**:
UC_04_Pre-1: A dismissible popup is currently in display on the current scene.
UC_21_9: **Post Conditions**:
UC_04_Post_1: Popup is dismissed from the scene.
UC_20_10: **Main flow**:
The user successfully selects the dismiss button.
The popup is closed.
UC_21_11: **Alternative flow** of events:
The user fails to select the dismiss button.
The user successfully selects the dismiss button; however, the popup isn't closed
UC_21_12: **Testing**:
When testing ensure that selecting the dismiss option of the popup successfully destroys the popup object.

UC_22
UC_22_1: **Use Case**: Fight sequence initiation
UC_22_2: **Author**: CP
UC_22_3: **Date**: 29/11/21
UC_22_4: **Purpose**: Positions enemy units in a random formation, giving variety to battle gameplay, and initiates fight sequence.
UC_22_5: **Overview**: When a user selects the fight button, the respective enemy units for that level should be positioned in a random formation and the fight sequence should be initiated.

UC_22_6: **Cross References**: US_08_1, US_08_2, US_21_1: , US_21_1:2, US_22_1, US_22_2
UC_22_7: **Actors**: User, units
UC_22_8: **Pre-conditions**:
UC_22_Pre-1: Programme is in grid scene with all player units placed
UC_22_Pre-2: fight button has been selected.
UC_22_9: **Post Conditions**:
UC_22_Post_1: All enemy units are placed on the board in a random formation.
UC_22_Post_3: Fight sequence is initiated.
UC_22_10: **Main flow**:
After the player is satisfied with their unit placement, they select the fight button on the grid scene.
The enemy units for the respective level are placed in a random formation on the board.
The fight sequence is initiated and the battle begins.
Battle continues until one side's units have all been eliminated.
UC_22_11: **Alternative flow** of events:
After the player is satisfied with their unit placement, they select the fight button on the grid scene.
The enemy units for the respective level are placed in a non-random formation on the board, which is the same every time.
The enemy units are placed on the board in a random formation; however, the fight sequence fails to be initiated and the battle does not begin.
The fight sequence is successfully initiated however one or more units fail to engage an opponent.
All units engage an opponent however one or more units fail to lock onto a new opponent after their opponent has been destroyed. This could lead to units stopping fight before one side has been eliminated.
UC_05_12: **Testing**:
When testing ensure all units are activated and engage an opponent. Ensure that units keep fighting unit all the units of one team are eliminated. Ensure units only deal damage to opposing units. Ensure after destroying an opponent, units are successfully able to lock onto a new opponent.

UC_23
UC_23_1: **Use Case**: Path-finder/ unit engagement
UC_23_2: **Author**: CP
UC_23_3: **Date**: 29/11/21
UC_23_4: **Purpose**: Engages troops in combat
UC_23_5: **Overview**: When a battle is initiated, both user and enemy units will search for the closest opposing unit and lock onto them. The units will then move towards each other until they are within attacking range at which point they will stop and begin fighting. If a unit successfully wins, it then locks onto a new opponent and the process is repeated.
UC_23_6: **Cross References**: US_08_1, US_08_2, US_21_1: , US_21_1:2, US_22_1, US_22_2
UC_23_7: **Actors**: Units
UC_23_8: **Pre-conditions**:
UC_23_Pre-1: User has placed all units on the board and initiated the fight sequence.
UC_23_Pre-2: Enemy units have successfully been deployed on the board upon initiation of fight sequence.
UC_23_9: **Post Conditions**:
UC_23_Post_1: All units on the board have successfully locked onto an opposing unit
UC_23_Post_2: All units have managed to reach and engage their opposing unit in battle.
UC_23_10: **Main flow**:

Fight sequence is initiated, all units on the board successfully lock onto an opposing unit and begin moving towards them.

All units successfully reach their respective opposing units and are able to engage them in battle.

The unit which wins the battle then locks onto a new opponent and the process is repeated.

UC_23_11: **Alternative flow** of events:

Fight sequence is initiated; however, some or all units are unable to successfully lock onto an opposing unit.

All units successfully lock onto an opposing unit; however, some or all units are unable to successfully reach their respective opposing unit.

All units successfully reach their respective opposing unit; however, some or all units fail to engage their respective opposing unit in battle.

All units successfully engage their opposing unit in battle; however, if they win their battle, they fail to engage a new unit and become inactive on the board.

UC_23_12: **Testing**:

When testing ensure each unit locks onto its closest opposing unit. Ensure after

UC_24

UC_24_1: **Use Case**: 1v1 fighting sequence

UC_24_2: **Author**: CP

UC_24_3: **Date**: 29/11/21

UC_24_4: **Purpose**: Process for engaged units to fight one another.

UC_24_5: **Overview**: When locked on units reach each other and are in range, they begin to deal damage to each other, according to their attributes, until one defeats the other.

UC_24_6: **Cross References**: US_08_1, US_08_2, US_21_1: , US_21_1:2, US_22_1, US_22_2

UC_24_7: **Actors**: User

UC_24_8: **Pre-conditions**:

UC_24_Pre-1: All units have been placed and fight sequence has been initiated.

UC_24_Pre-2: Pathfinding has successfully brought two opposing units together so that they're within attack range of each other.

UC_24_9: **Post Conditions**:

UC_24_Post_1: Units have successfully engaged in battle and one has been destroyed.

UC_24_10: **Main flow**:

Two engaged units are within range at which point they begin dealing damage to each other.

Each unit deals a certain amount of damage to the other at a constant rate. These variables are determined by the unit type.

Once the health of one of the units is less than or equal to zero, the unit object is destroyed and the opposing unit has won the battle.

UC_24_11: **Alternative flow** of events:

Two engaged units are within range of each other; however, on or both the units fail to deal damage to the other.

Once the health of one of the units is less than or equal to zero, the unit object fails to be destroyed. The units stay engaged in battle and keep dealing damage to each other.

UC_24_12: **Testing**:

When testing ensure that both units, when in range, deal damage to each other. Ensure that as soon as one unit's health points subceeds zero, it is destroyed. Ensure the correct damage output and frequencies of units are received by the opposing unit.

UC_25

UC_07_1: **Use Case**: Return to map button.

UC_25_2: **Author**: CP

UC_25_3: **Date**: 29/11/21
UC_25_4: **Purpose**: Allows the player to exit the battle sequence and go back to the grid scene.
UC_25_5: **Overview**: When the player is in the grid scene, they should be able to return to the map by selecting a return button at any point the grid scene sequences.
UC_25_6: **Cross References**: US_04_1, US_04_2, US_10_1, US_10_2
UC_25_7: **Actors**: User, grid manager
UC_25_8: **Pre-conditions**:
UC_04_Pre-1: Player is currently in the grid scene
UC_25_9: **Post Conditions**:
UC_04_Post_1: Player is returned to the map scene positioned at the node they exited.
UC_25_10: **Main flow**:
Player is in the grid scene and selects the return to map button.
Player is returned to the map and they are positioned at the node they exited.
UC_25_11: **Alternative flow** of events:
Player is in grid scene and fails to select the return to map button.
Player selects the return to map button; however is not successfully transferred to the map scene.
The player returns to the map without completing the battle encounter; however, they are no longer positioned at the node they exited.
UC_25_12: **Testing**:
When testing ensure that upon selecting the return to map button the user is transferred to the map scene and positioned at the node they exited if they did not complete the battle and the node they entered if they did.

UC_26
UC_26_1: **Use Case**: Battle won
UC_26_2: **Author**: CP
UC_26_3: **Date**: 6/12/21
UC_26_4: **Purpose**: Allow user to progress forward in the game once they've won a battle encounter.
UC_26_5: **Overview**: When the user has initiated a fight sequence in a battle encounter and successfully eliminated all enemy units, they should be able to choose to return to the map scene and progress through the game.
UC_26_6: **Cross References**: US_23_1, US_23_2
UC_26_7: **Actors**: User
UC_26_8: **Pre-conditions**:
UC_26_Pre-1: Player has initiated a fight sequence.
UC_26_Pre-2: Player has successfully eliminated all enemy units.
UC_26_9: **Post Conditions**:
UC_26_Post_1: Player is returned to the map positioned at the node they entered.
UC_26_10: **Main flow**:
Player has successfully eliminated all enemy units.
Player is displayed a popup informing them the battle is won with a button to return to the map.
Player selects the button and is returned to the map positioned at the node they entered.
UC_25_11: **Alternative flow** of events:
Player initiates a fight sequence and successfully eliminates all enemy units.
Player is not displayed a popup informing them the battle is won with a button to return to the map.
Player selects the button; however they are not returned to the map scene.
Player selects the button and is returned to the map scene, however they are not returned to the node they entered.

UC_26_12: **Testing**:
When testing ensure all enemy units are destroyed before battle won popup is triggered. Ensure return to map button successfully returns the player to the map positioned at the battle node they entered.

UC_27

UC_27_1: **Use Case**: Battle lost
UC_27_2: **Author**: CP
UC_27_3: **Date**: 6/12/21
UC_27_4: **Purpose**: Allow user to replay a battle if they loose.
UC_27_5: **Overview**: When the user has initiated a fight sequence in a battle encounter and all their troops have been eliminated by the enemy units, they should be able to restart the battle and try again.
UC_27_6: **Cross References**: US_23_1, US_23_2
UC_27_7: **Actors**: User
UC_27_8: **Pre-conditions**:
UC_27_Pre-1: Player has initiated a fight sequence.
UC_27_Pre-2: Player has had all their units eliminated by the enemy units.
UC_27_9: **Post Conditions**:
UC_27_Post_1: The player returns to the troop placement sequence in the grid scene.
UC_27_10: **Main flow**:
The player initiates a fight sequence and all their units are eliminated by the enemy units.
The player is displayed a popup message informing they have lost the battle with an option to select and try again.
The player selects try again and is transferred back to the start of the unit placement sequence, with the same number of units they had preceding the battle.
The player can either replace their units and initiate a new fight sequence or return to the map.
UC_27_11: **Alternative flow** of events:
The player initiates a fight sequence and all their units are eliminated by the enemy units.
The player is not displayed a popup message informing them they have lost the battle with the option to select try again.
The player selects try again but is not transferred back to the start of the unit placement sequence.
The player is transferred back to the start of the unit placement sequence but no longer has the same number of units as they had preceding the battle.
UC_27_12: **Testing**:
When testing ensure that all player units have been destroyed before battle lost popup is triggered. Ensure the player is returned to the start of the unit placement sequence when the try again button is selected.

UC_28

UC_28_1: **Use Case**: Deploy ranged units
UC_28_2: **Author**: CP
UC_28_3: **Date**: 6/12/21
UC_28_4: **Purpose**: Gives the player gameplay and tactical variety by allowing them to deploy ranged units on the board to support their melee units.
UC_28_5: **Overview**: When the user is placing their units, they should be able to strategically place ranged units as tactical support for their melee units
UC_28_6: **Cross References**:
UC_28_7: **Actors**: User
UC_28_8: **Pre-conditions**:
UC_28_Pre-1: Player is in the grid scene in the unit placement phase.
UC_28_Pre-2: Player has one or more ranged units in their unit pool

UC_28_9: **Post Conditions**:

UC_28_Post_1: The player has strategically placed ranged units on the board to support their melee units.

UC_28_10: **Main flow**:

Once the troop placement scene is initiated, the player will see ranged units in their arsenal.

The player selects ranged units from their unit arsenal and places them where they see fit.

The player hits “fight” and the ranged units support the melee units by dealing ranged damage to enemy units.

The enemy units are able to detect the placed ranged units and will try to attack them along with the melee units.

UC_28_11: **Alternative flow** of events:

Once the troop placement scene is initiated, the player does not see ranged units in their arsenal.

After the player selects ranged units from their arsenal they are unable to place them on the board.

After the player hits “fight” the ranged units fail to deal damage to the enemy units from range.

The enemy units are unable to detect the placed ranged units and do not try to attack them.

UC_28_12: **Testing**:

When testing ensure pathfinding is adapted and functional for ranged units so that ranged units are able to engage opposing units and vice versa. Ensure that ranged units function correctly with the unit and grid managers. Ensure that Ranged units are able to deal and take damage successfully.

UC_29

UC_29_1: **Use Case**: Deal damage to non-engaged targets in range.

UC_29_2: **Author**: CP

UC_29_3: **Date**: 6/12/21

UC_29_4: **Purpose**: To enable melee units to damage targets other than the one they have engaged

UC_29_5: **Overview**: During the battle sequence, units should be able to damage opposing units which are in range but not necessarily engaged.

UC_29_6: **Cross References**:

UC_29_7: **Actors**: User

UC_29_8: **Pre-conditions**:

UC_29_Pre-1: Fight sequence has been initiated.

UC_29_Pre-2: One or more non-engaged opposing units are in range.

UC_29_9: **Post Conditions**:

UC_29_Post_1: Damage is dealt to a non-engaged unit.

UC_29_10: **Main flow**:

The player initiates the fight sequence.

One or more non-engaged opposing units come into range of a unit, while the engaged unit is out of range.

Damage is consistently dealt to a random non-engaged unit which is in range while the engaged unit is not in range.

UC_29_11: **Alternative flow** of events:

The player initiates the fight sequence.

No non-engaged enemy units come into range of a unit.

One or more enemy units come into range of a unit; however, the engaged unit is also in range so damage is consistently dealt to the engaged unit.

One or more enemy units come into range of a unit, the engaged unit is also in range; however the unit fails to deal damage to any of the opposing units in range.

One or more non-engaged opposing units come into range of a unit, while the engaged unit is out of range; however, the unit fails to deal damage to any of the opposing units in range.

UC_29_12: **Testing**:

When testing ensure that damage is randomly dealt to a target within range and this is only triggered when the engaged target is out of range. Ensure that when the engaged target is in range it is the only target which receives damage.

UC_30

UC_30_1: **Use Case**: Grid scene colour

UC_30_2: **Author**: CP

UC_30_3: **Date**: 6/12/21

UC_30_4: **Purpose**: To define the appearance of battle scenes for different areas of the map.

UC_30_5: **Overview**: The grid scene for battle nodes in different areas of the map should look different depending on what type of tile they're on e.g forest, water etc.

UC_30_6: **Cross References**: US_04_1, US_04_2, US_06_1, US_06_2, US_10_1, US_10_2

UC_30_7: **Actors**: Battle grid

UC_30_8: **Pre-conditions**:

UC_30_Pre-1: Player has entered a battle node on a distinctive tile.

UC_30_9: **Post Conditions**:

UC_30_Post_1: Battle grid is coloured representing the type of tile it is located at.

UC_30_10: **Main flow**:

The player enters a battle node on a distinctive tile, and transferred to the grid scene.

The grid scene appears tinted in the colour corresponding to the tile the battle node is located on

UC_30_11: **Alternative flow** of events:

The player enters a battle node on a distinctive tile, and transferred to the grid scene.

The grid scene does not appear tinted in the colour corresponding to the tile the battle node is located on

UC_30_12: **Testing**:

When testing ensure that the grid is given the correct coloured tint corresponding to the battle node it is located on.

UC_31

UC_31_1: **Use Case**: Unit health bar

UC_31_2: **Author**: CP

UC_31_3: **Date**: 6/12/21

UC_31_4: **Purpose**: To display the health points of each unit during the game

UC_31_5: **Overview**: During battle sequences, all units in the battle should have their health points displayed above the sprite in the form of a health bar.

UC_31_6: **Cross References**: US_22_1, US_22_2

UC_31_7: **Actors**: Unit, unit manager

UC_31_8: **Pre-conditions**:

UC_31_Pre-1: Battle sequence has been initiated

UC_31_9: **Post Conditions**:

UC_31_Post_1: each unit has a health bar displayed above their sprite.

UC_31_10: **Main flow**:

Player initiates fight sequence.

Each unit on the board is initialised with a health bar above their sprite.

Each time a unit takes damage, the respective amount of damage taken is deducted from their health bar.

UC_31_11: **Alternative flow** of events:

Player initiates fight sequences.

Each unit on the board fails to display a health bar above their sprite.
Each unit on the board successfully displays a health bar over their sprite; however, upon taking damage, the health bars fail to show the correct damage taken.

UC_31_12: **Testing**:

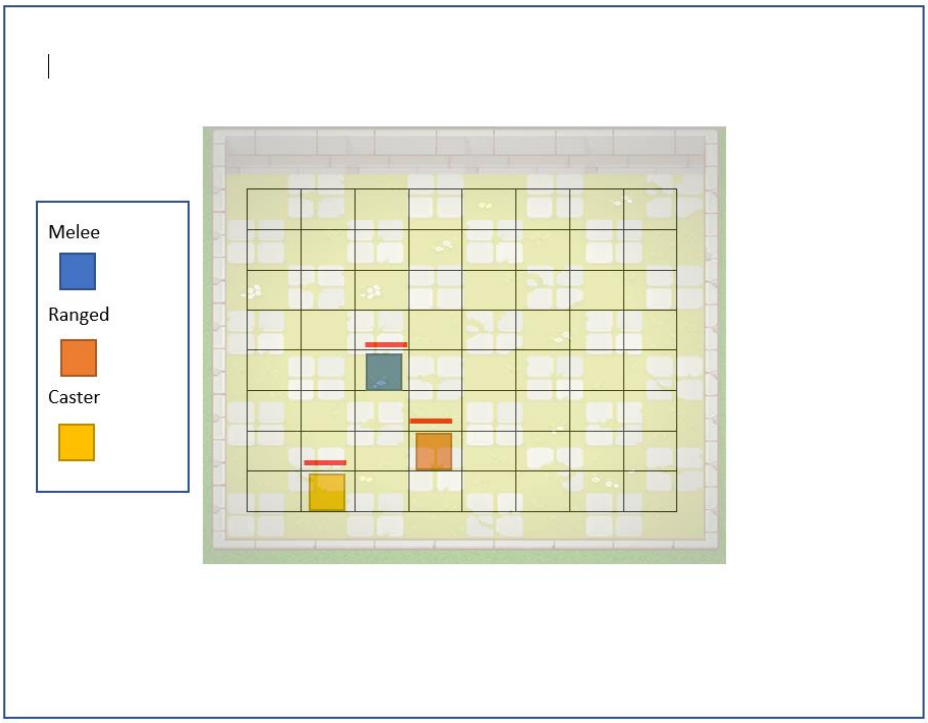
When testing ensure all units display a health bar. Ensure upon taking damage all units have the correct amount of health deducted from their health bars.

UC_32

UC_32_1: **Use Case**: Randomised map creation
UC_32_2: **Author**: CP
UC_32_3: **Date**: 13/12/21
UC_32_4: **Purpose**: To create a game map with randomised nodes adding variety in player experience game to game
UC_32_5: **Overview**: Upon starting the game, a map is created with randomised treasure battle and arena nodes.
UC_32_6: **Cross References**: US_04_1, US_04_2, US_06_1, US_06_2, US_07_1, US_07_2, US_08_1, US_08_2, US_10_1, US_10_2
UC_32_7: **Actors**: Map
UC_32_8: **Pre-conditions**:
UC_32_Pre-1: Game has been started
UC_32_9: **Post Conditions**:
UC_32_Post_1: A map with randomised battle and treasure nodes is created
UC_32_10: **Main flow**:
The player starts the game and a randomised map is generated.
UC_32_11: **Alternative flow** of events:
Each time the player starts the game, the same map is generated.
Each time the player starts the game, the map fails to be generated.
UC_32_12: **Testing**:

User interface design | 7.10.4

Auto-Battler UI



Elements

Unit Sprites - References **US_05_1, US_05_2, US_08_1, US_08_2, UC_03, Units, Renderer**

8x8 grid comprised of Tiles **US_08_01, US_08_02, US_11_01, US_11_02, US_14_01, US_14_01, US_15_01, US_15_02, UC_03, Tiles, Renderer, Grid Manager**

Unit selector on left side of window **US_5_1, US_5_2, US_8_1, US_8_2, US_13_1, US_13_2, US_15_1, US_15_2, UC_03, Renderer, Units**

Should indicate types available, and how many of each type remaining

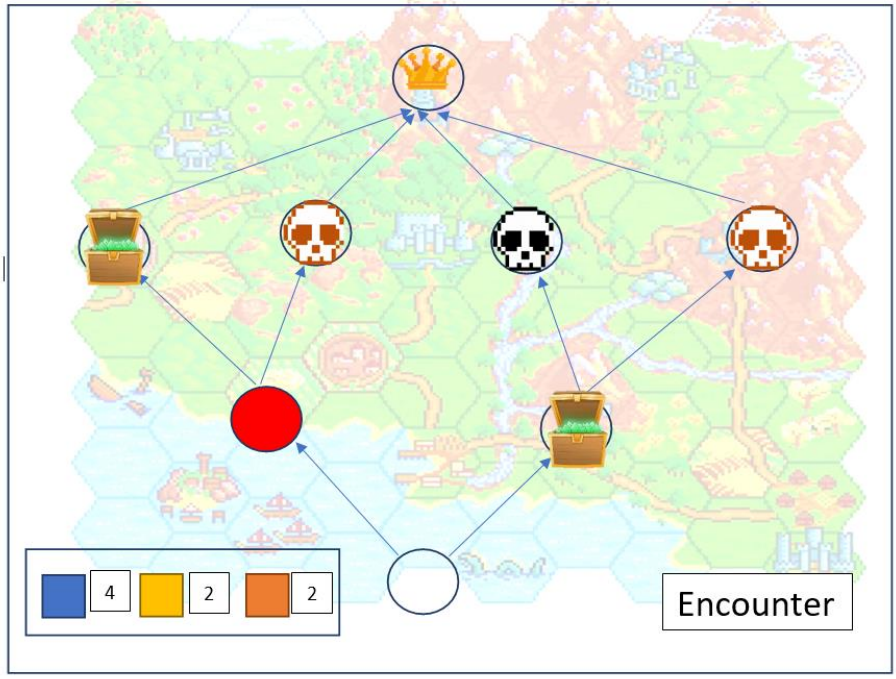
Appropriate methods implemented to enable clicking on unit type to select, and then clicking on Tile to place

Health bars above unit sprites **US_15_1, US_15_2**

Method should resize bar depending on current health

Background sprite should represent position of grid in game world, and should change depending on location.

Dungeon Map UI

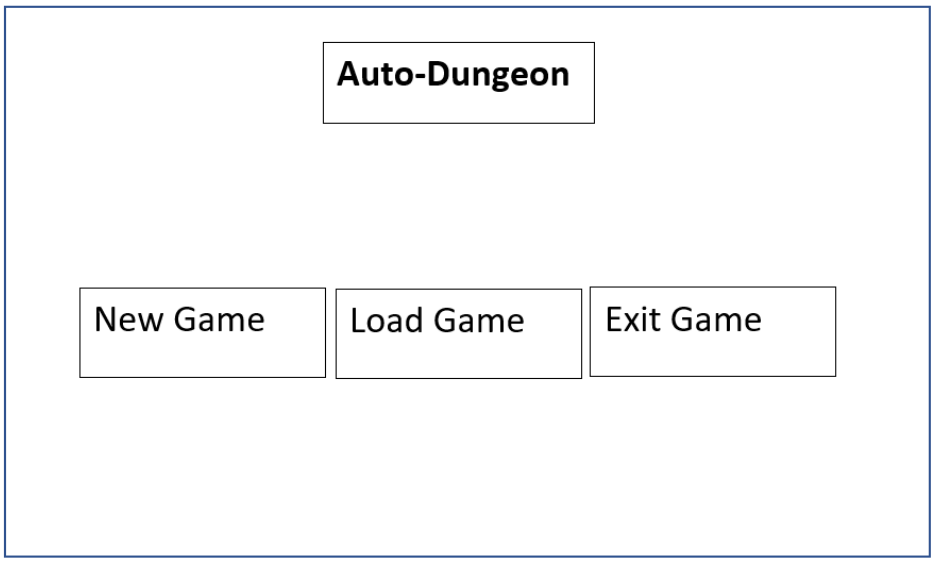


Elements

Tree Nodes **US_06_1, US_06_2, US_07_1, US_07_2, UC_04, Dungeon_Tree**
Should have method to change colour to indicate node currently occupied by party
Should have method to determine which encounter and then change scene
Unit's remaining indicator **US_07_1, US_07_2, US_09_1, US_09_2, US_11_1, US_11_2, US_14_1, US_14_2, US_15_1, US_15_2, Dungeon_Tree, Units**
Should visually show the number and composition of remaining units to aid decision making
Encounter button to enter encounter at current Node
Should have visual "overworld" map to help in telling underlying game story
Encounter difficulty should increase as game progresses
Each node should have a representative sprite which indicates both the type and difficulty of the encounter, with a unique sprite to indicate the end node.

Main Menu UI

NOTE, MAIN MENU DEPRECATED, NOT IMPLEMENTED AT TIME OF DELIVERY DUE TO TIME CONSTRAINTS.



Elements

New Game Button **US_03_1, US_03_2, UC_01**
Should have on-click method ready to start new game method
Load Game Button **US_02_1, US_02_2**
Should have on-click method ready to start load game method
Method should load available save game, display error message if no available save game
Exit Game Button **US_01_1, US_01_2, UC_02**
Should close game window and end all associated processes