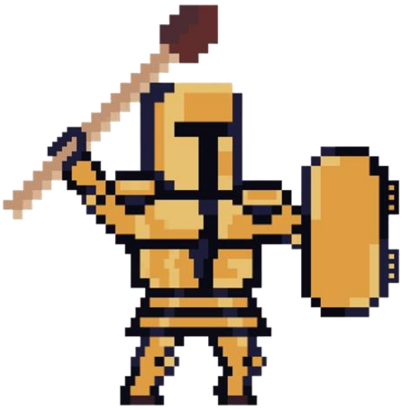


TinderBox Knight Documentation

Pink 2

Seth Whitehorn, Zening Zhu, Yijin Huang, Adil Moosa, Robert Williams, Yunni Xie, Peiwen Chi, Devendra Dhokare, Karan Desai

GitHub URL: https://github.com/Robzi11a/TinderBoxKnight/tree/delivery

Contents

Sprint 1 (03/11-10/11)

1.1 Overview

This first sprint and the following sprint form the ‘pre-game’ phase of the project. The aim of these two sprints is to plan and design the game, so that we can start coding efficiently in Sprint 3. As a result, no code will be produced. As a team, our aim for the sprint is to decide on a game idea and generate an initial list of requirements for it. The structure of this sprint will be slightly different to the rest of the project, as we were still forming our working processes and deciding what to do. By the end, we aim to have some concept art to show the customer to demonstrate what the game will look like, as well as an initial set of user stories that captured our original ideas for the game as well as the feedback we receive in the meeting. We also aim to have decided the process we will use to structure each sprint.

1.2 Process

1.2.1 Meeting Minutes

I. Meeting One
Date: 03/11/2021
Time: 13:00-15:00
Attendance: Yunni, Zening, Peiwen, Devendra, Adil, Rob, Karan, Yijin, Seth
Agenda:

- Meet other members of the team.
- Brainstorm ideas for the game.
- Plan first customer meeting.
- Decide on meeting schedule for the rest of term.
- Decide on project process used for development.

Outcome:

- Two ideas for the game were conceived (one side-on, vertically scrolling; one top-down), and were developed to a point where they could be shown to customer.
- A list of questions was created to ask the customer.
- Two meetings a sprint were agreed, a longer meeting on Wednesday before the customer meeting and a shorter one on Friday to update on progress.
- A project process based on Scrum was agreed, but with Scrum cycles limited to one sprint running Wednesday to Wednesday. The shortcomings of Scrum were also discussed and it was agreed to use supplementary elements from other methodologies like DSDM and XP.

Action items:

- Have first customer meeting.
- Complete formal skills audit before next meeting (person responsible: Seth).

II. Customer Meeting
Date: 03/11/2021
Time: 15:45-16:00
Agenda:

- Rob: Present game idea one to customer for feedback.
- Seth: Present game idea two to customer for feedback.
- Clarify customer statement by asking questions.

Outcome:

- The team will focus on developing idea one further.
- See products section for transcript of customer feedback.

III. Meeting Two
Date: 05/11/2021
Time: 11:00-11:15
Attendance (virtual): Yunni, Zening, Peiwen, Devendra, Adil, Rob, Karan, Seth, (Yijin: finishing coursework)
Agenda:

- Review skills audit.
- Discuss tools to use in development.
- Decide on tasks to be done before next meeting.

Outcome:

- Python is the most common language known by the team. See product section of Sprint 1 for results and analysis.
- Two broad tasks were identified: create concept art and initial asset design to show customer and start documenting requirements. Those who wanted to work on the design volunteered to do so. The rest of the team will work on gathering and specifying requirements.
- Trello and Jira were identified as potential project management tools. No member of the team had experience with either but based on previous discussion in one of the Q&A sessions it was decided to try Jira first. Git and Github were put forward as the only choices for version control and hosting the repository for the project. The team had already been using Whatsapp and Microsoft Teams to communicate so that will continue.

Action items:

- Create initial set of user stories (people responsible: Karan, Devendra, Zening, Seth).
- Create initial concept art and asset design (people responsible: Peiwen, Yunni, Rob, Yijin, Adil).
- Make Jira project (person responsible: Seth) and Github repository (person responsible: Rob).

1.2.2 Work Plan

Cycle	Sprint	Phase	Planned Milestone	Achieved
1	03/11-10/11	Pre-game		
2	10/11-17/11	Pre-game	Product backlog	
3	17/11 - 24/11	Game		
4	24/11 - 01/12	Game		
5	01/12 - 08/12	Game		
6	08/12 - 15/12	Game		
7	15/12 - 20/12	Post-game	Final hand in	

The work plan sets out the planned stages the team are in each week and the planned milestones. We agreed with the customer to have the product backlog and system design ready by the end of the second sprint, so that we can start development. These sprints together will form Scrum's pre-game phase. After this, the team will move into the game phase and start development. The products for this phase have been kept blank for now and will be agreed with the customer once we have a better idea of the system requirements. The final sprint is not a full sprint (Wednesday - Monday) so we should finish the game before then and use the extra time to ensure the quality of the documentation.

1.2.3 Backlog

Backlog item	Name	Cross references	Responsibility	Comments	Status
B-1	Week one user stories		Seth, Devendra, Karan, Zening		New

B-2	Asset design		Rob, Yunni, Yijin, Adil		New
B-3	Concept art		Rob		New
B-4	Develop story		Adil, Peiwen		New
B-5	Create skills audit		Seth		New
B-6	Set up Jira		Seth		New
B-7	Set up GitHub repository		Rob		New

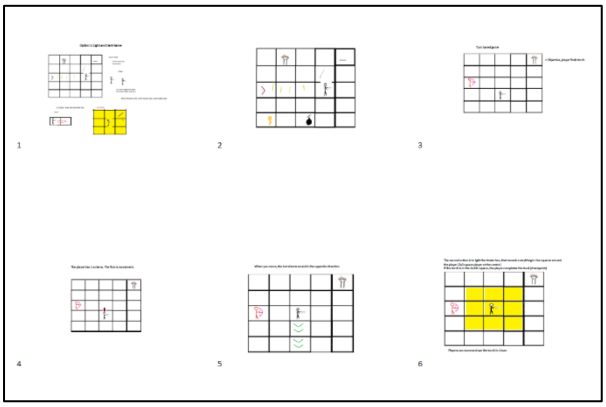
1.2.4 Exception Handling

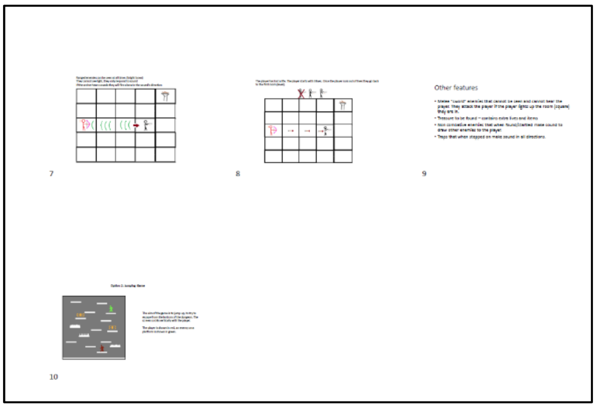
None.

1.3 Products

1.3.1 Customer Meeting

I. Presentation





Other features

- Melee "sound" enemies that cannot be seen and cannot hear the player. They attack the player if the player lights up the room (square) they are in.
- Treasure to be found – contains extra lives and items
- Non combative enemies that when found/startled make sound to draw other enemies to the player.
- Traps that when stepped on make sound in all directions.

The first presentation to the customer focused on the two game ideas which the team had identified in the initial meeting. The slides above mainly focus on potential game mechanics for the first game, showing the grid-based level and the lighting mechanic. Also included are some ideas about potential enemy interactions. The second game idea was simpler so only had one slide at the end.

II. Questions:

1. Can you define a dungeon?
    a. A space for exploration where you encounter unexpected circumstances.
2. What functionality would you most like to see in the game?
    a. That's a bit open-ended. The genre is limiting to a large extent, so the question comes to how to make the limitations contribute to the game rather than hinder it. And how to make sure that the game can be replayed - not necessarily with new content, but using the framework you've already made.
3. Should the bot compete with the player, or help it, and should competition come from direct competition or just by trying to complete the level?
    a. I think the latter. Trying to compete directly against the bot changes the game to be a speed or reflex test, which doesn't seem to fit with the nature of the game. I'm happy for the bot to help the player.
4. What is the target audience for the game?
    a. I'd rather let the game determine the audience, rather than the other way around.
5. In terms of platforms - would you prefer it to be mobile, or computer?

a. Having mobile is attractive but limits the game - input is more limited and it's harder to program, so unless you have the experience it might be best to concentrate on computer. If you did it browser-based, it could potentially work on both, though.

III. Analysis:

Based on the feedback of the customer, focusing our attention on the first game idea seems like the most promising avenue. The customer said it had some unusual features, can with more development could make for an engaging game, while the second idea is more derivative. Although he was open to us presenting the second idea again with more development, we will concentrate on the first.

Based on the questions, the customer was very open with their requirements, defining a dungeon quite broadly. We were unsure as to whether 'dungeon' was a thematic (setting) or content (type of game) requirement. The customer seemed to lean towards the latter but overall was flexible in the kind of game. The customer was similar open in terms of features of the game and the role of the bot. Although the initial brief indicated that the bot should be something that could play the game on its own, the customer saw that having a bot that would accompany the player would be more suited to the direction of the game.

IV. Key takeaway

- Focus on the first game idea

1.3.2 User stories

Name	Title	User Story	Acceptance Criteria
US-1	Movement	As a player, controlling a character I want to move the character, so that I can explore the dungeon.	User input results in the character sprite moving one tile.
US-2	Enemies	As a player, I want to have enemies in the dungeon, so that there are things to avoid.	Enemies are placed in levels.
US-3	Lives	As a player, I want to have limited lives, so that I must be careful about what I do.	The player starts with a finite number of lives.
US-4	Losing lives	As a player, I want to lose a life when an enemy attacks, so that I must avoid them.	The number of lives decreases when an enemy attacks.
US-5	Light	As a player, exploring the dungeon, I want to be able to light up areas of the dungeon, so that I can see the area.	On user input, areas of the grid switch from being dark to light.
US-6	Bot	As a player I want to have a bot to help me so that I have more options for how to deal with enemies.	Bot appears in the game and can perform an action.
US-7	Timer	As a player, I want to see how fast I can complete the dungeon so that I can replay the game and try to beat the score.	The time taken for a player to complete the level is displayed once the level is complete.
US-8	Captions	As a player, I want to see captions for certain activities so that I have more information about the game and story.	Captions appear for relevant game activities.
US-9	Items	As a player, I want the dungeon to have items so that there is an incentive to explore.	Items are placed in dungeon.
US-10	Using items	As a player, I want to be able to use items so that there is variety in game play.	On user input, an item in the user's inventory is used and has an effect in the game.

US-11	Placing enemies	As a player, I want to enemies to be in different places each time I play the level so that I can replay the level.	Enemies are placed in a non-predetermined place in the grid.
US-12	Enemy counter	As a player, exploring the dungeon, I want to see whether enemies are nearby, so that I can plan my next move.	Player can see whether nearby squares contain enemies.
US-13	Finishing level	As a player, I want to be able to light up a torch within the level so that I have a goal in each level.	When the torch is lit the level is complete.
US-14	Starting level	As a player, I want to choose a level from the menu, so that I can start the game.	On specified user input the level starts.
US-15	Randomising level	As a player, I want to be able to randomise the level, so that I can replay it.	There is a way for the user to generate a random level.

1.3.3 Use cases

None this sprint.

1.3.4 CRC cards

None this sprint.

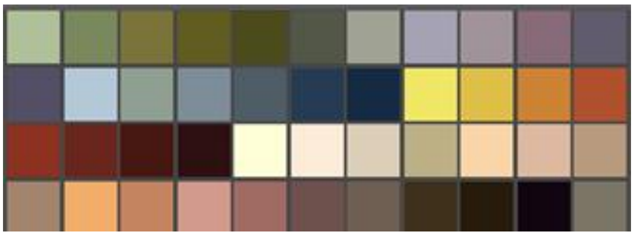
1.3.5 Tests

None this sprint.

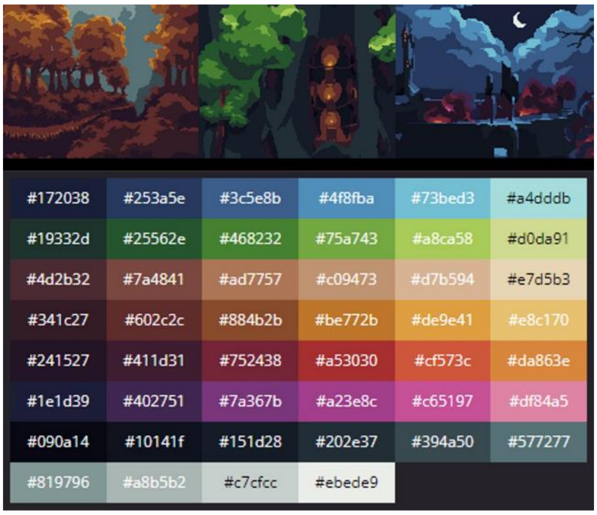
1.3.6 Design Elements

I. Colour Palette

Option one (custom):



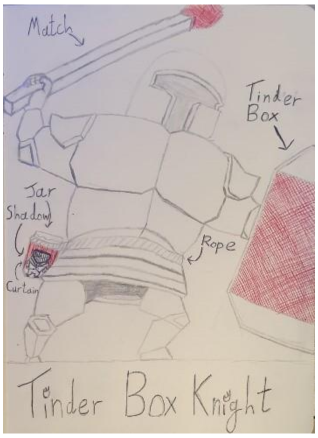
Option two: (from https://lospec.com/palette-list/apollo):

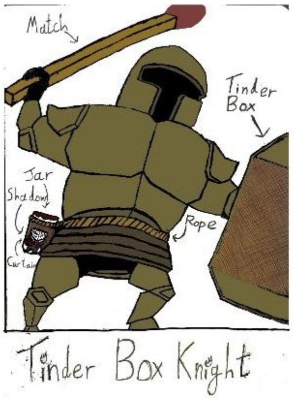


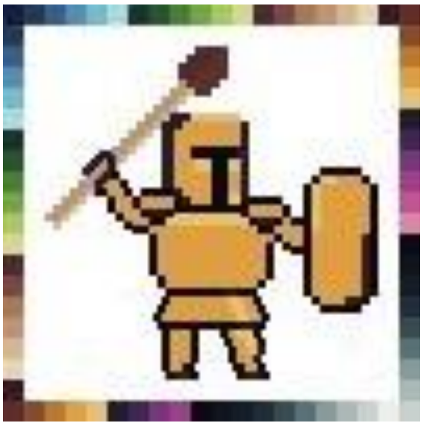
#172038	#253a5e	#3c5e8b	#4f8fba	#73bed3	#a4dddb
#19332d	#25562e	#468232	#75a743	#a8ca58	#d0da91
#4d2b32	#7a4841	#ad7757	#c09473	#d7b594	#e7d5b3
#341c27	#602c2c	#884b2b	#be772b	#de9e41	#e8c170
#241527	#411d31	#752438	#a53030	#cf573c	#da863e
#1e1d39	#402751	#7a367b	#a23e8c	#c65197	#df84a5
#090a14	#10141f	#151d28	#202e37	#394a50	#577277
#819796	#a8b5b2	#c7cfcc	#ebede9		

II. Main Character

Shown below are the three steps of drawing: drawing, concept art, pixel art

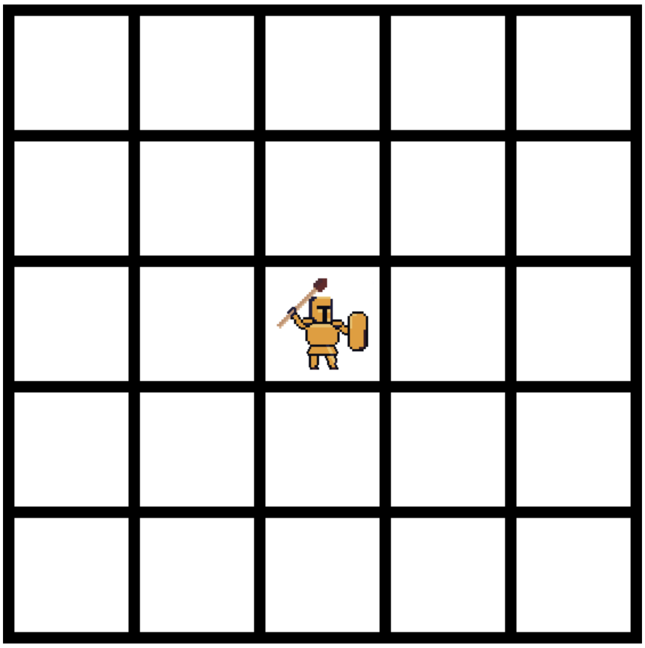






The pixel art is 100*100 pixels. To keep the detailing/shading consistent, only 2x2pixel, 3x3pixel, 4x4pixel and 5x5pixel brushes were used - this will be the case for all the pixel art produced. The surrounding colours are the palette, to keep drawing style consistent between team members.

III. Scale
Black line = 10 pixels
White box = 100 pixels



1.3.7 Other Products

I. Skills Audit

Name	Experience	Languages	Other skills	First degree
Seth	< 6 months	Python	Writing, organising	Politics
Rob	1-2 years	Python, C++	Pygame, visual design	Civil Engineering
Tsening	1-2 years	Java, JS	HTML, testing	Computer Science
Adil	< 6 months	Python	Writing /documenting, visual design	Aerospace Engineering
Yunni	1-2 years	Java, C, C++	Design	Internet Of Things
Peiwen	< 1 year	Python	Writing, design	Language and Literature
Karan	1-2 years	Python	HTML, design	Computer Science
Devendra	1-2 years	Python	HTML, design	Computer Science
Yijin	1-2 years	Java, C, C++	UI Design, writing	Computer Science

Python is the most common language known by the team, with six out of the nine members having some familiarity with it. The skills audit did not ask about the degree of familiarity, so ability may vary. The next most common languages are Java and C++, with three each. As over half the team are already familiar with Python, this will probably be the easiest for development. One member of the team (Rob) also has some experience with game development using Python and the Pygame module.

Programming experience within the team also varies, from under 6 months to between 1 to 2 years. However, no members of the team have zero previous programming experience so everyone should be able to contribute some code. Other skills vary, from design ability to writing ability.

1.4 Review

Our first meeting took place before the customer meeting. We had a significant amount to get through, so we took two hours to meet. Meetings in further weeks will likely be shorter, but it was worth having a long meeting to start as all the members of the team also had to introduce themselves. Prior to the meeting, we had set up a WhatsApp Chat and Microsoft Teams group. We were originally going to meet on Tuesday, but it was decided that it would be best if we spent Tuesday brainstorming game ideas individually, then met on Wednesday to present our ideas to the group. This meant session on Wednesday (see section 1.2.1) was not a brainstorm from scratch but involved combining ideas from different members. The team worked well together as people brought different ideas that combined synergistically - some members had focused on the story, while others had thought about the game mechanics. Combining these left us with two major ideas for the game. The first was a top-down game, tile-based game in which the player moved through an unlit dungeon and had the ability to place torches to turn grid tiles light, in order to see enemies. The story for this game was based on the idea that the player-character was hired by a local village to light up the dungeon. The second option was a side-on, vertically scrolling game, in which the player aimed to get as high as possible by jumping vertically from platform to platform. The story for this game was based on the idea that the player had been imprisoned at the bottom of a large dungeon in the shape of a tunnel running vertically. The player would have to climb to the top to escape.

After formulating these two ideas, we presented them to the customer (the slides are included in section 1.3.1). Although both ideas could have made viable games, going into the meeting, we felt that the first option had more scope for development. The customer echoed these feelings, so the decision was taken to focus on this game. We then asked the customer a list of questions to try to understand the requirements for game better. After the customer meeting the team had a quick debrief to discuss tasks for the upcoming week. The two main tasks identified were to start the art design and write up the initial requirements of the system. We also agreed a sprint structure, with one meeting on Wednesday before and after the customer meeting to serve as the review of the preceding week's sprint and the planning meeting for the coming sprint, and a second meeting on Friday as a mid-Scrum check-in.

In the Friday meeting we first discussed the skills audit (the audit and results of the discussion are attached in section 1.3.7). We then allocated tasks to do before Wednesday. It was agreed that two sub-teams would form, one focusing on art design and one focusing on requirements. Those that wanted to work on the art design team volunteered, and the remaining members of the team worked on the requirements. This worked well, and by the end of the sprint we had a good list of initial user stories as and a concept art to show the customer so they could visually see what the game will look like.

Sprint 2 (10/11 - 17/11)

2.1 Overview

This week is the second half the planning phase. By the end of the week are aiming to be in a position to start coding the initial features of the game. After the first week we had a good idea of what the game was going to be like in terms of design and mechanics, so during the customer meeting at the start of this cycle it was agreed that the team would deliver a working demo in two weeks' time. The activities this week therefore mainly focused on developing the ideas we had during the first cycle into concrete features that were ready to be implemented.

For the requirements, the aim for this week was to prioritise features so that we could start developing the most important features with use cases and CRC cards, as well as exploring how the classes would interact. We did not schedule any new user stories on the product backlog as we had developed a sufficient amount during the first week. On the design side, the aim was to create the tiles that would be used in the demo so that they were ready to be implemented.

2.2 Process

2.2.1 Meetings

I. Meeting One

Date: 10/11/2021
Time: 14:00-15:00 (including customer meeting).
Attendance: Yunni, Zening, Peiwen, Devendra, Adil, Karan, Yijin, Seth, (Rob: ill).
Agenda:

- Review the sprint's work.
- Prepare customer meeting.
- Plan tasks for coming sprint.

Outcome:

- The design work from the previous sprint and the concept art to show the customer was presented to the whole team. This has been included below in the product section.
- The current set of user stories were reviewed.
- The tasks to prioritise were decided by using the MoSCoW prioritisation system. The results of this are included in section 2.2.5.
- Outstanding and new tasks for the coming sprint were brainstormed. Again, they mainly fell into two groups: art design and system design. The initial set of user stories was planned to be completed by the next meeting (12/11), so that these can be used for system design elements like use cases and class diagrams.

To do:

- Get customer feedback on colour palate.
- Transfer brainstormed tasks onto Jira so that they can be added to project backlog (person responsible: Seth).

II. Customer Meeting

Date: 10/11/2021
Time: 14:30-14:45
Agenda:

- Present concept art and design to customer.
- Discuss the progress of the project and agree initial milestones.

Outcome:

- Customer was positive about art direction.
- It was agreed that next sprint the team would present the release plan and implementation schedule. In two sprints' time (25/11) the team will present the first working demo of the game to the customer.

III. Meeting Two

Date: 12/11/2021
Time: 11:00-11:20
Attendance: Yunni, Peiwen, Devendra, Adil, Yijin, Seth, Rob.
Agenda:

- Update team on documentation changes.
- Discuss basic game mechanics.
- Review everyone's work so far and planned work for rest for second half of cycle.

Outcome:

- Seth: Went through the changes to the documentation. Principally, these were moving the documentation from Overleaf to Word to allow for tables and images to be inserted more easily, and adding the work plan on to each sprint. Jira was also discussed to make sure that there were no problems in using it.
- The basic game mechanics were discussed, based on an earlier post in Teams which set out an idea for the foundational mechanic of the game. Adil suggested adding a score mechanic so that players can compare their performance based on moves taken, time taken, lives left, and other factors. This was added to the user stories for this week.

- Each member of the team briefly discussed what they were working on this sprint and when they aimed to achieve it by.
    - Rob: Pixel art of the knight and the torch, set up the GitHub repository and starter code.
    - Adil: Write the story, write CRC cards.
    - Peiwen: Write the story, add art design elements into the documentation.
    - Yunni: Work on enemy design and floor tiles (pressure plate).
    - Yijin: Work on enemy design and floor tiles (normal stone and cracked stone).
    - Devendra: UML diagrams and class diagrams.
    - Seth: Write use cases and CRC cards.

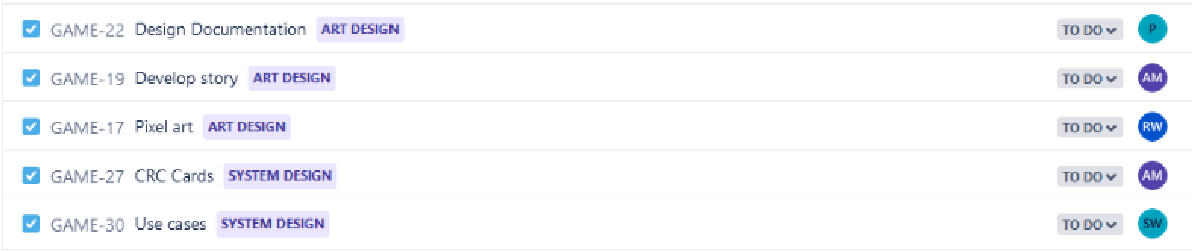
2.2.2 Work Plan

Cycle	Sprint	Phase	Planned Milestones	Achieved?
1	03/11-10/11	Pre-game		N/A
2	10/11-17/11	Pre-game	Plan for first prototype	Yes
3	17/11 - 24/11	Game	Prototype one - demo	
4	24/11 - 01/12	Game		
5	01/12 - 08/12	Game	Prototype two	
6	08/12 - 15/12	Game		
7	15/12 - 20/12	Post-game		

Dates for full prototypes were agreed with the customer this sprint. The work schedule has been updated to show this. The first prototype is due after only one game cycle, so will necessarily be quite basic. The second prototype is due two sprints later.

2.2.3 Backlog

I. Jira backlog



☑ GAME-22 Design Documentation ART DESIGN TO DO P
☑ GAME-19 Develop story ART DESIGN TO DO AM
☑ GAME-17 Pixel art ART DESIGN TO DO RW
☑ GAME-27 CRC Cards SYSTEM DESIGN TO DO AM
☑ GAME-30 Use cases SYSTEM DESIGN TO DO SW

II. Detailed backlog

Backlog item	Name	Cross references	Responsibility	Comments	Status at start of cycle
B-1	Week one user stories		Seth, Devendra, Karan, Zening	See Sprint 1 products	Completed
B-2	Initial asset design		Rob, Yunni, Yijin, Adil	Renamed pixel art on Jira	Ongoing
B-3	Concept art		Rob		Completed

B-4	Develop story		Adil, Peiwen		Ongoing
B-5	Create skills audit		Seth		Completed
B-6	Set up Jira		Seth		Completed
B-7	Set up GitHub repository		Rob		Completed
B-8	Design documentation		Peiwen		New
B-9	Use cases		Seth, Devendra, Karan		New
B-10	CRC cards		Seth, Zening		New
B-11	Start sprint 2 documentation		Seth		New

2.2.4 Exception Handling

No significant problems were encountered.

2.2.5 Prioritisation table

The MoSCoW prioritisation system separates items into must have, should have, could have, and won't have. The must have items are those items which must be in the game to meet the minimum requirements. The should have elements are those which would substantially improve the game, but without which the game could still be delivered. The could have items are those which are desirable but will not have a major impact if not included. The won't have items are those which the team has chosen not to implement. When we start coding, we will work through the must have items, should have, and could have in turn.

Must have	**Should have**	**Could have**	**Won't have**
Movement	Clue tiles	Usable items	Sound-based mechanics
Level grid	Ranged enemy	Randomised tiles	
Stationary enemy	Main menu	Sound effects	
Placeable light	Lives	Score mechanics	
Enemy scan	Environmental tiles		
End of level torch			

2.3 Products

2.3.1 Customer Meeting

I. Presentation

Tinder Box Knight

10 Nov 2021 | 2nd Customer Meeting | Design

1

Concept Drawing

Tinder Box Knight

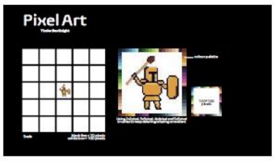
2

Concept Art

Tinder Box Knight

Tinder Box Knight

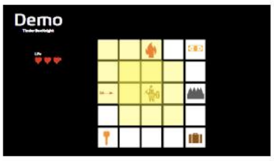
3



Pixel Art

Tinder Box Knight

4



Demo

Tinder Box Knight

5

Colour Palette

Tinder Box Knight

Goal: Keep consistency

Choice 1
Combined Different PALLETTE

Choice 2
APOLLO PALLETTE

6

II. Questions:

1. Do you have a preference for the size of the level and how it looks?
    a. It depends on what you want. The room could vary in size: getting narrow or large depending on the room. It could be a cave or a passage. If the room is non-regular it could lead it in different direction - that could be interesting for the player.
2. Which colour palette do you prefer? Our recommendation would be the second, to allow for both bright and dark colours within the game.
    a. Both look like they would fit with the game, so I will trust your recommendation. As the contrast between dark and light seems central to the game mechanics.

III. Analysis:

The customer meeting this week focused on presenting a more detailed version of the game from the first week. This had two strands. We first presented the intended art direction of the game, as the first customer meeting had little indication of what the game would look like. The customer gave positive feedback about the character design, saying that it was distinctive and captured the feel and story of the game well. We also presented the results of the prioritisation table, so the customer could see which features we were planning on developing first. They were happy with the placements of features so we will use this as the basis for future work.

2.3.2 User stories

Name	Title	User Story	Acceptance Criteria
US-16	Clue	As a player, I want to have tiles with clues to the end on so that I know where I am meant to be heading.	A clue tile exists in the level and can be seen by the player.
US-17	Ranged enemy	As a player, I want to have to avoid the view of a ranged enemy so that I am forced to be careful about where I step.	Ranged enemy exists in the level and can attack the player if they walk in front of it.

US-18	Score	As a player, I want to have a score of how well I have done on a level so that I can go back and try to beat it.	When the level ends the player's score is displayed.
US-19	Sound	As a player, I want to have a sound effect when player interacts with the environment around.	A sound effect is played when the player finds an enemy.

2.3.3 Use cases

Each use case has two variants. Requirements use cases are first, and are designated with a -r. These are then developed into design use cases, with a -d. Use cases with the same number refer to the same feature, regardless of the variant, so for cross referencing purposes only the number will be included. There are three levels for the use cases at this stage. 'System framework' refers to use cases taking place outside of the level. 'Level interaction' are use cases which interact with the player but are not deliberately triggered by the player. 'User action' are those use cases which are deliberately triggered by the player.

Use cases selected to develop this week are the features which were identified as 'must haves' in the prioritisation exercise conducted in meeting one and summarised in 2.3.7.

I. Requirements Use Cases

UC-1-r
Title: Displaying a level
Creation date: 11/11/2021
Level: Level framework
Cross references:
Context:
- When the player starts the game, the level is displayed on the screen.

Frequency: Whenever the player starts a level.

UC-2-r
Title: Moving the character
Creation date: 11/11/2021
Cross references: US-1
Level: User action
Context:
- The player can move the knight one tile to the north, east, south and west, as long as the target tile is empty and in the bounds of the level.
- If the tile contains an enemy it will light up and the knight will reset.

Frequency: Whenever the player presses an arrow key in the level.

UC-3-r
Title: Lighting up an area
Creation date: 11/11/2021
Cross references: US-5
Level: User action
Context:
- Players can press 'f' to light up a three-by-three grid centred on their location.
- Tiles within this grid become visible.

Frequency: Whenever the player presses the 'f' key within a level.

UC-4-r
Title: Losing a life

Creation date: 11/11/2021
Cross-references: US-3, US-4, US-2
Level: Level interaction
Context:

- The player will lose a life when they find an enemy.
- If the number of lives equals zero, they will be taken back to the start menu.

Frequency: Whenever the player finds an enemy.

UC-5-r
Title: Detecting enemies
Creation date: 11/11/2021
Cross-references: US-12
Level: User action
Context:

- When the player presses ‘s’, the number of enemies in a 5x5 grid centred on their location will be displayed.

Frequency: Whenever the user presses s.

UC-6-r
Title: Finishing the level
Creation date: 11/11/2021
Cross-references: US-13
Level: User action
Context:

- The player can use the light function to reveal a torch within the level.
- Activating the torch will end the level.

Frequency: Once per level.

II. Design Use Cases

UC-1-d
Title: Displaying the level
Author: Seth
Creation date: 12/11/2021
Purpose: Display the level when the player starts the game.
Overview: When the player starts a level, it should be loaded on the screen.
Cross references:
Actors: Player
Dependencies: Start menu (when implemented)
Pre-condition:

1. A level text file has been made.
2. The player is at the start menu of the game.

Post-condition:

1. The first level will be displayed on the screen.

Normal flow of events:

1. The player selects the ‘start’ option.
2. The text file containing the level is read in.
3. The images are displayed on the screen.

Alternative flow of events:
Exception flow of events:

2. The file contains a code that the game cannot associate with a tile. The game closes, and a message is displayed indicating the problem code.

Frequency: Whenever the player starts the game.

UC-2-d

Title: Moving the character
Author: Seth
Creation date: 12/11/2021
Purpose: Allow the player to move the character.
Overview: When the player presses a specified movement key, the character will move to a new tile on the screen.
Cross references: US-1
Actors: Player
Dependencies: UC-1
Pre-condition:

1. The player must be currently in a level.
2. The player-character must be in an idle state, not in the middle of another activity.
3. The player has at least one life.

Post-condition:

1. The player-character will be in a grid tile above the starting tile.

Normal flow of events:

1. The player presses the up arrow on the keyboard.
2. Check that the tile above the player is a valid movement spot.
3. Check that the new tile does not contain a monster.
4. The player-character sprite is moved from the centre of the initial tile to the centre of the tile above.

Alternative flow of events:

1. The player presses the right arrow. The tile checked is the one to the east, and the character will move to the to the east.
1. The player presses the down arrow. The tile checked is the one to the south, and the character will move to the to the south.
1. The player presses the left arrow. The tile checked is the one to the east, and the character will move to the to the west.
2. The intended tile is blocked by another object. The player-character stays on the tile where they currently are.
3. The tile contains a monster. See UC-4-r, alternative flow.

UC-3-d
Title: Lighting up an area
Author: Seth
Creation Date: 12/11/2021
Purpose: Allow the player to light up an area.
Overview: When the user presses the specified key, a torch is placed on the square and the 3x3 grid centred on the player is lit up.
Cross references: US-5
Actors: Player
Dependencies: UC-1
Pre-condition:

1. The player is currently in a level.
2. The player-character is not currently in the middle of another action.
3. The player has at least one life.

Post-condition:

1. The 3x3 grid of tiles centred on the player is lit up and objects in the tiles are revealed.

Normal flow of events:

1. The player presses the space bar on the keyboard.
2. A check is run that there is not already a torch on the square that the player-character is standing on.
3. A check is run on the 3x3 grid centred on the player and the torch to see which squares are dark and which are light.
4. Any dark squares are changed to light, with a corresponding visual change.

5. Any objects which those tiles contain are made visible.

Alternative flow of events:

3. A torch has already been used on the square and surrounding squares are lit. There is no change to the square or surrounding grid.

UC-4-d
Title: Lives
Author: Seth
Creation Date: 12/11/2021
Purpose: Allow for enemies to attack the player.
Overview: When the player finds an enemy, they lose a life.
Cross references: US-4, US-3, US-2
Dependencies: UC-1, UC-2, UC-3
Actors: Player, enemy
Pre-condition:

1. The player is currently in a level.
2. Enemies have been placed in the level.
3. The player has at least one life.

Post-condition:

1. The number of lives is decreased by one.
2. The player respawns at the beginning of the level.
3. The enemy that attacked the player disappears.

Normal flow of events:

1. The player moves into a dark square.
2. The player lights up the dark square. In one of the tiles in the lit up 3x3 grid, there is an enemy sprite.
3. Text displays telling the player they have found an enemy.
4. The number of lives the player-character has decreases by one. A check is run that the number of lives is above zero.
5. The player-character sprite and camera are moved to the start of the level. The enemy that attacked the player disappears from the level, but otherwise the level is the same.

Alternative flow of events:

2. The square the player moves into contains an enemy. Jump to 3 and continue from there.
4. The number of lives equal zero. A message tells the player that they have failed the level and gives them the option to retry it. The player-character is moved to the start of the level. The rest of the level is reset - all enemies are replaced, and all changes the player made (placing torches, etc) are removed.

UC-5-d
Title: Detecting enemies
Author: Seth
Creation Date: 12/11/2021
Purpose: Allow the player to detect whether there are enemies nearby.
Overview: On user input, a count of nearby enemies in a specified location will be displayed to the player.
Cross references: US-12
Actors: Player, enemies
Pre-condition:

1. The player is currently in a level.
2. The player-character has at least one life.
3. The player is not currently in the middle of another action.

Post-condition:

1. A count of the number of enemies in the 3x3 grid in front of the player is displayed on the screen.

Normal flow of events:

1. The player presses the control button on the keyboard.
2. A check is run that the player is standing in a lit square.
3. A check is run on the number of enemies in the 3x3 grid to the north, south, east, and west of the player is standing on (this is done irrespective of whether the squares are dark or light).
4. The number of enemies existing in each grid is displayed on the screen to the player, on the grid square adjacent to the player in each direction.
5. The number remains for one second then fades away.

Alternative flow of events:

2. The player is standing in a dark square. Nothing happens.
3. The full 3x3 grid in a specified direction does not exist due to the level boundaries. The check is only run on the squares that exist. If no squares exist because the player is standing next to the boundary, then a count of zero is displayed for that direction. Normal flow resumes.

UC-6-d
Title: Finishing the level
Author: Seth
Creation date: 12/11/2021
Purpose: To allow the player to finish a level.
Overview: The square on which the torch is placed is lit. A message tells the player that the level is complete, and the level ends.
Cross references: US-13
Dependencies: UC-1, UC-2, UC-3
Actors: Player
Pre-condition:

1. The player is currently in a level.
2. The player-character has at least one life.

Post-condition:

1. The level ends by fading to black, and the next level starts.

Normal flow of events:

1. The player moves into a new square (UC-1).
2. The player lights a torch on the new square, lighting up surrounding tiles (UC-2).
3. One of the lit squares contains the torch. The player presses space while standing next to the torch.
4. An animation plays of the torch lighting up, and all squares in the map switch from being dark to light.
5. A message tells the player that they have completed the level.
6. Transition to the start of the next level plays.

Alternative flow of events:

3. The lit squares do not contain the torch. Use case ends.

2.3.4 CRC Cards

The initial set of CRC cards presented here were identified using a process of noun identification on the use cases above. The interactions are also based on the flows specified in the design use cases.

CRC-1: Knight	
Responsibility	**Collaboration**
• Keep track of player location • Keep track of player lives • Allow for player to see the number of nearby enemies. • Interaction with enemies	• Dungeon tile • Main

CRC-2: Light	
Responsibility	**Collaboration**
• Switch tiles from dark to light in 3x3 grid around player.	• Knight • Main

CRC-3: Tile	
Responsibility	**Collaboration**
• Map value in level array to image.	• Main

CRC-4: Spider	
Responsibility	**Collaboration**
• Attack character when they walk into the enemy, causing them to lose a life. • Attack character when the tile the enemy is on is lit up, causing them to lose a life.	• Dungeon tile • Main

CRC-5: End-of-level Torch	
Responsibility	**Collaboration**
• Ends the level when lit.	• Tile • Knight • Main

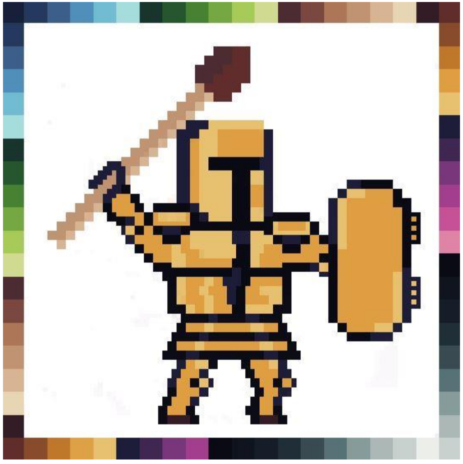
CRC-6: Main	
Responsibility	**Collaboration**
• Read in the level from the text file • Create objects • Handle user input • Draw the level	• Knight • Tile • Spider

2.3.5 Tests

None this sprint.

2.3.6 Design Elements

I. Knight
Updated to have basic shading and more detail on breast plate.



II. Spider

Concept art (left) and pixel art (right).

III. Ghost

Concept art for second enemy type.

IV. Torch

Lit and unlit pixel art.

2.4 Review

The first task of the week was working out which user stories we would focus on to have as part of the demo. For this we used the MoSCoW prioritisation system, featured in the Dynamic System Development Method. The results of this are included in 2.2.5. The elements in the must-have category included movement, the level grid, a stationary enemy, a way to light up tiles, and a way to end the level through lighting a torch. These features are the elements that are most critical to the game, as together they form the foundation of every level. We therefore focused on developing these this week. Similar to the previous week, this was divided into tasks creating the art assets for these features and tasks to develop the requirements and specify the behaviour of these features.

For the behaviour and requirements task, the relevant user stories were first identified, and high-level use cases were produced for the features in the must have category (see section 2.3.3). The relevant user story(s) has been included with each use case. The high-level use cases were then developed into more detailed design use cases which specified the behaviour more exactly. Noun identification was then done on the user stories and use cases to get an initial list of classes/objects, which were developed into the CRC cards (section 2.3.4). There was no specific task to generate user stories, but some were produced naturally during the week. These were added to the prioritisation table in the relevant slot

On the design side, the tiles that were produced are included in section 2.3.6. These include an updated version of the knight, the basic tiles for the level, the first enemy type (a spider), and the end of level torch that the player must find. An idea for a second enemy type of enemy was also sketched out, although no pixel art was produced as it will not feature in the first version of the game.

As we have not started development yet, there were no exceptions this week. All work proceeded smoothly and no major errors were encountered.

Sprint 3 (17/11 - 24/11)

3.1 Overview

In this week, the team will move from the 'pre-game' phase where we were primarily focused on gathering requirements and designing the art assets to be used, to the 'game' phase where we will begin active implementation of the features in code. As agreed with the customer, we are aiming to be able to deliver a first demo version of the game in the customer meeting at the start of sprint 4. This will have all the basic features present. These will be based on the 'must have' items from the prioritisation table in section 2.3.7: a level, a moveable knight, a stationary enemy type, a torch, a way to end the level, and a way for the player to scan the number of enemies in nearby tiles.

3.2 Process Documents

3.2.1 Meeting Minutes

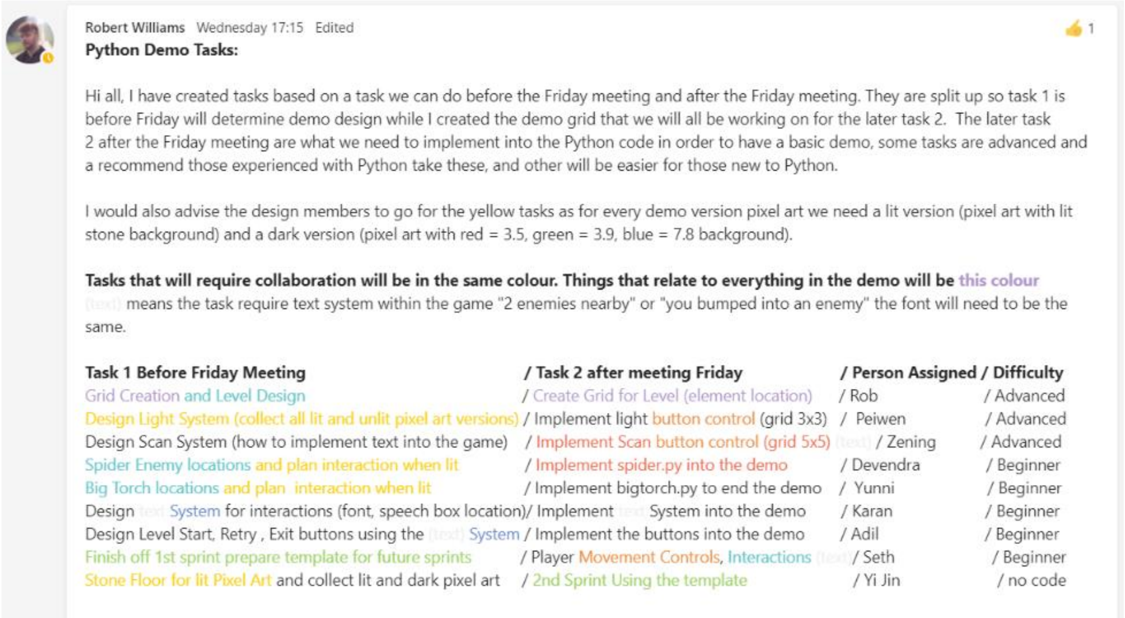
I. Meeting One
Date: 17/11/2021
Time: 14:25-15:15 (including customer meeting)
Attendance: Yunni, Zening, Peiwen, Devendra, Adil, Rob, Karan, Yijin, Seth
Agenda:

- Review previous week's work.
- Discuss prioritisation of tasks to establish what will be in the demo level.
- Review material to present to the customer.
- Plan tasks for coming week.

Outcome:

- Completed material from the previous week was presented, principally the design work and the concept for the story.
- The features that will be included in the demo level were agreed. They will be the level, the knight (with movement), the placeable torch and the end of level torch, and a stationary enemy type.
- The team had a brief discussion of coding standards. Using comments to explain what functions are doing and how they are doing it (if non-obvious), and using clear variable names were both agreed.
- How to write tests for the coding tasks was discussed. No one in the team had experience with automating tests, so it was decided not to do this. Instead, all tests will be in the form of test cases, which will provide detailed instructions and acceptance criteria for the tests to be run manually. Initially, testing features will be the responsibility of the team member developing that feature. In later weeks members of the team will be specifically assigned to run all the tests and verify them.

- Other tasks identified included finishing the documentation from the previous two weeks and continuing the design work. These were written up after the meeting (see image below). As a team we also estimated the difficulty of each task - this was based on the size and the complexity of the task. Task assignment was again primarily voluntary; team members opted to do tasks that suited their skills.



Robert Williams Wednesday 17:15 Edited

Python Demo Tasks:

Hi all, I have created tasks based on a task we can do before the Friday meeting and after the Friday meeting. They are split up so task 1 is before Friday will determine demo design while I created the demo grid that we will all be working on for the later task 2. The later task 2 after the Friday meeting are what we need to implement into the Python code in order to have a basic demo, some tasks are advanced and a recommend those experienced with Python take these, and other will be easier for those new to Python.

I would also advise the design members to go for the yellow tasks as for every demo version pixel art we need a lit version (pixel art with lit stone background) and a dark version (pixel art with red = 3.5, green = 3.9, blue = 7.8 background).

Tasks that will require collaboration will be in the same colour. Things that relate to everything in the demo will be this colour
(text) means the task require text system within the game "2 enemies nearby" or "you bumped into an enemy" the font will need to be the same.

Task 1 Before Friday Meeting	/ Task 2 after meeting Friday	/ Person Assigned	/ Difficulty
Grid Creation and Level Design	/ Create Grid for Level (element location)	/ Rob	/ Advanced
Design Light System (collect all lit and unlit pixel art versions)	/ Implement light button control (grid 3x3)	/ Peiwen	/ Advanced
Design Scan System (how to implement text into the game)	/ Implement Scan button control (grid 5x5) (text)	/ Zening	/ Advanced
Spider Enemy locations and plan interaction when lit	/ Implement spider.py into the demo	/ Devendra	/ Beginner
Big Torch locations and plan interaction when lit	/ Implement bigtorch.py to end the demo	/ Yunni	/ Beginner
Design (text) System for interactions (font, speech box location)	/ Implement (text) System into the demo	/ Karan	/ Beginner
Design Level Start, Retry , Exit buttons using the (text) System	/ Implement the buttons into the demo	/ Adil	/ Beginner
Finish off 1st sprint prepare template for future sprints	/ Player Movement Controls, Interactions (text)	/ Seth	/ Beginner
Stone Floor for lit Pixel Art and collect lit and dark pixel art	/ 2nd Sprint Using the template	/ Yi Jin	/ no code

II. Customer Meeting

Date: 17/11/2021
Time: 14:45-15:00
Agenda:

- Present this week's prepared presentation to the customer. The slides are included under section 3.3.1 I. They focus on the latest pixel art assets, the plan for the demo version, and the story for the game.
- Discuss and clarify the aim for the demo with the customer.

Outcome:

- The presentation was delivered to the customer. Rob presented the sections on the pixel art and the demo features; Peiwen presented the story updates.
- The customer gave feedback on elements of the presentation. Extended comments and analysis are included in section 3.3.1.
- The meeting concluded with a discussion of the state of the documentation, including the existing state and what had been focused on.

III. Meeting Two

Date: 19/11/21
Time: 18:30 - 19:00
Attendance: Yunni, Zening, Peiwen, Devendra, Adil, Rob, Karan, Yijin, Seth
Agenda:

- Review progress on the task for the first half of the sprint and share any problems.
- Introduce structure of the current code so members were clear on how to integrate their work.

Outcome:

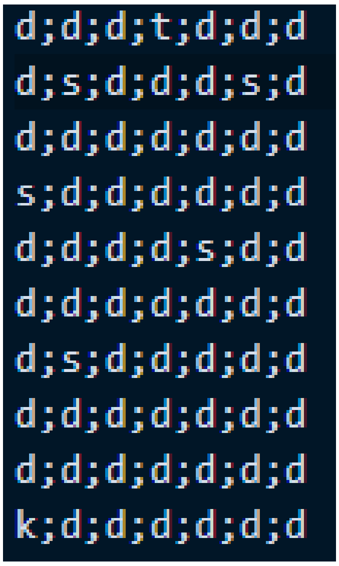
- No one had encountered any problems in doing their first task. Due to time constraints, we were unable to go round and individually share our progress, but all members of the team indicated that they were happy.
- Rob: Walked through the existing code on the GitHub repository. Levels are created using text files, with each letter representing a different tile. This text file is read in and used to

generate the tiles for the level (see items shared element one). The main file of the game is TinderBoxKnight.py, which handles the basic generation of the level. Other functionality will be written as classes in separate files, which will then be imported into the main file. This is to keep the design as modular as possible.

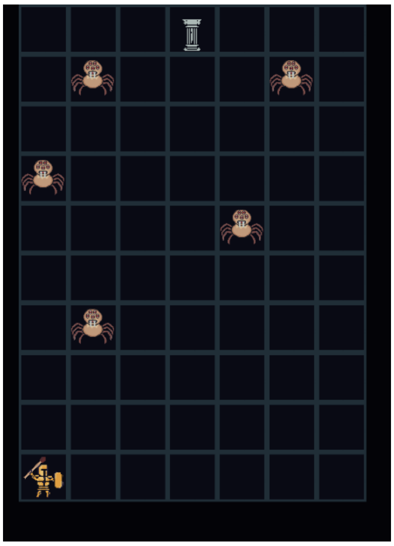
- The planned tasks for the second half of the week were discussed so that everyone knew what they were meant to be doing. The tasks were mapped out visually (see items shared element two).

Items shared during meeting:

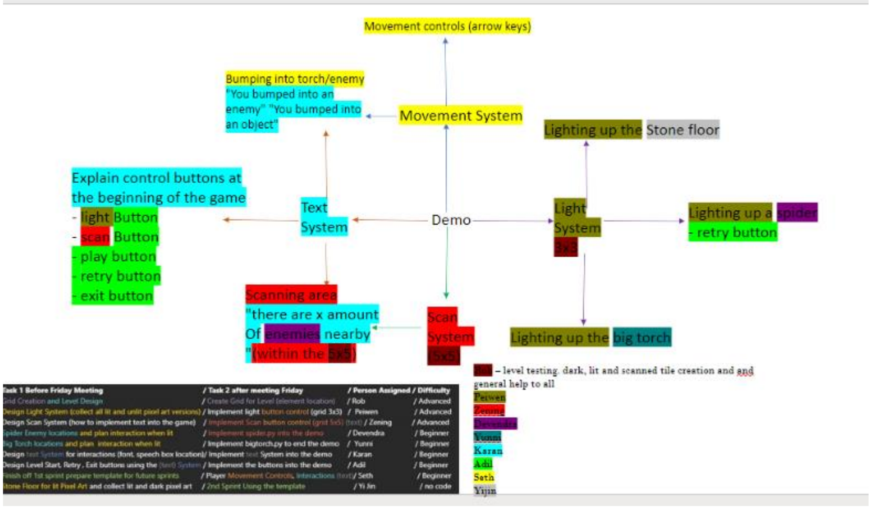
1. Text file and the level generated by it:



```
d;d;d;t;d;d;d
d;s;d;d;d;s;d
d;d;d;d;d;d;d
s;d;d;d;d;d;d
d;d;d;d;s;d;d
d;d;d;d;d;d;d
d;s;d;d;d;d;d
d;d;d;d;d;d;d
d;d;d;d;d;d;d
k;d;d;d;d;d;d
```



2. Work plan for week showing dependencies and how tasks relate:


Movement controls (arrow keys)
Bumping into torch/enemy
"You bumped into an enemy" "You bumped into an object"
Movement System
Lighting up the Stone floor
Explain control buttons at the beginning of the game
- light Button
- scan Button
- play button
- retry button
- exit button
Text System
Demo
Light System (3x3)
Lighting up a spider
- retry button
Scanning area
"there are x amount Of enemies nearby "(within the 5x5)
Scan System (5x5)
Lighting up the big torch

```
Task 1 Before Friday Meeting                                   / Task 2 after meeting Friday                          / Person Assigned / Difficulty
Grid Creation and Level Design                                 / Create Grid for Level (element location)             / Rob             / Advanced
Design Light System (collect all lit and unlit pixel art versions) / Implement light button control (grid 3x3)        / Peiwen          / Advanced
Design Scan System (how to implement text into the game)       / Implement Scan button control (grid 5x5) (text) / Zening             / Advanced
Spider Enemy locations and plan interaction when lit           / Implement spider.py into the demo                    / Devendra        / Beginner
Big Torch locations and plan  interaction when lit             / Implement bigtorch.py to end the demo                / Yunni           / Beginner
Design text System for interactions (font, speech box location)/ Implement text System into the demo                 / Karan           / Beginner
Design Level Start, Retry , Exit buttons using the (text) System / Implement the buttons into the demo                / Adil            / Beginner
Finish off 1st sprint prepare template for future sprints      / Player Movement Controls, Interactions (text)/ Seth                   / Beginner
Stone Floor for lit Pixel Art and collect lit and dark pixel art / 2nd Sprint Using the template                       / Yi Jin          / no code
```

Rob – level testing. dark, lit and scanned tile creation and and general help to all
Peiwen
Zening
Devendra
Yunni
Karan
Adil
Seth
Yijin

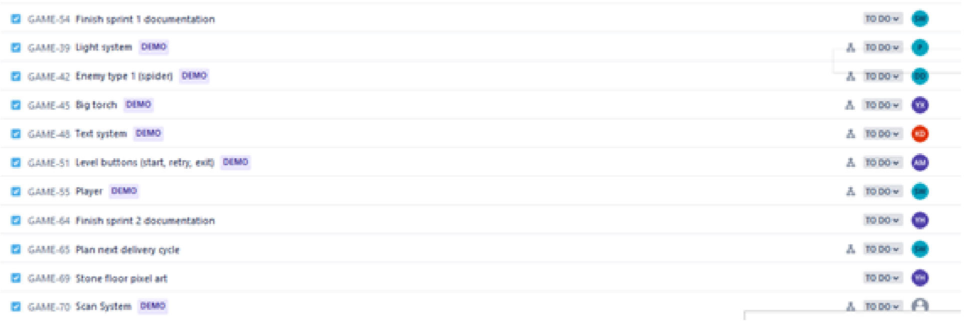
3.2.2 Work Plan

Cycle	Sprint	Phase	Planned Milestone	Achieved
1	03/11-10/11	Pre-game		N/A
2	10/11-17/11	Pre-game	Plan for first prototype	Yes
3	17/11 - 24/11	Game	Delivery one - demo	
4	24/11 - 01/12	Game		

5	01/12 - 08/12	Game	Delivery two	
6	08/12 - 15/12	Game		
7	15/12 - 20/12	Post-game	Final hand in	

3.2.3 Backlog

I. Jira backlog



GAME-54 Finish sprint 1 documentation TO DO
GAME-39 Light system DEMO TO DO
GAME-42 Enemy type 1 (spider) DEMO TO DO
GAME-45 Big torch DEMO TO DO
GAME-48 Text system DEMO TO DO
GAME-51 Level buttons (start, retry, exit) DEMO TO DO
GAME-55 Player DEMO TO DO
GAME-64 Finish sprint 2 documentation TO DO
GAME-65 Plan next delivery cycle TO DO
GAME-69 Stone floor pixel art TO DO
GAME-70 Scan System DEMO TO DO

II. Detailed Backlog

Backlog Item	Name	Cross references	Responsibility	Comments	Status at start of cycle
B-2	Initial asset design		Rob, Yunni, Yijin, Adil	All tiles needed for the demo are done.	Completed
B-4	Develop story		Adil, Peiwen	Presented in customer meeting.	Completed
B-7	Set up GitHub Repository		Rob		Completed
B-8	Design documentation		Peiwen		Completed
B-10	CRC cards		Seth, Zening	See 2.3.4	Completed
B-9	Use cases		Seth	See 2.3.3	Completed
B-11	Sprint 2 documentation		Seth, Yijin, Peiwen		Ongoing
B-12	Sprint 2 review				New
B-13	Review and finish sprint 1 documentation		Seth	For feedback next seminar.	New
B-14	Light System	US-5, UC-3, CRC-2	Peiwen, Zening		New
B-15	Scan system	US-12, UC-4, CRC-7	Zening		New
B-16	Spider	US-2, UC-4, CRC-4	Devendra		New
B-17	Big torch	US-13, UC-6, CRC-5	Yunni		New
B-18	Text system	US-8	Karan		New
B-19	Buttons	UC-1	Adil		New

B-20	Character movement	US-1, UC-2, CRC-1	Seth		New
B-21	Update floor pixel art		Yijin		New
B-22	Sprint 3 documentation		Seth		New
B-23	Plan next set of features		Seth	Including writing use cases and CRC cards.	New

Feature tasks are listed as one task for space reasons on the backlog, but within Jira have been separated into two subtasks. The first task, covering the first half of the scrum cycle (from the Wednesday meeting to the Friday), is to design the feature and write the tests. The task for the second half of the cycle is to implement it. This will be the structure for all coding tasks.

3.2.4 Exception Handling

EX-1
Problem: The way the level was being generated made it difficult to dynamically interact with the tiles.
Details: The initial code read the text file and used this to directly create a list of tiles, which were then used to draw the visuals on the screen. However, this list was one-dimensional, so did not represent the row and columns of the grid and did not obviously distinguish between different tiles (as all tiles are objects of the same class). This made features that required interacting with the tile system (principally B-14, B-15, and B-20) hard to implement.
Handling: A new step was added to the code so that the text file is first read into a 2-dimensional list, where each sub-list is a full row of tiles. This means that any tile can be easily accessed by first indexing to the correct row, and then to the correct column. This list is then passed to the draw function to create and draw the new list of tiles.
Status: Closed

EX-2
Problem: When the knight moves it overwrites whatever tile was there previously.
Details: Movement was implemented in a way that means the knight’s tile does not dynamically change to whatever light was there previously. In addition, when the knight moves off a tile it is set automatically to a blank dark square as there is no way of telling what tile was there before.
Handling: Movement has been left as implemented for the first demo. It will be explained to the customer that this is just the first version and that the issue will be fixed by next week. The use case (UC-1) has been updated to reflect the new understanding of the knight’s movement and the task has been added to the project backlog for Sprint 4.
Status: Open

EX-3
Problem: When using the default tile size, the level does not fit on some screens.
Details: The default tile size in the game is set as constant variable equally 100, meaning 100x100 pixels. The demo level is then formed of 10 tiles of this size vertically, and 7 horizontally. However, although the game window scales to the size of the player’s screen, the size of the tiles does not. This means that on smaller screens the level does not fit on the screen. Several members of the development team ran into this issue.
Handling: Tile size has been left as a constant for this week, with each member editing the TILESIZE variable on their local repository to fit on their screen before they started editing or running the code. Potential solutions to this which did not involve editing the code were discussed, with the consensus being setting the variable to a set fraction of the total screen dimensions. This has been added to the backlog for Sprint 4.

Status: Open

3.2.5 Prioritisation table

The prioritisation table will be included in each week and updated as we progress through it. Blank items are unimplemented, yellow items are in progress, and green items are fully implemented. Progress will be marked at the start of the week.

Must have	Should have	Could have	Won't have
Movement	Clue tiles	Usable items	Sound-based mechanics
Level grid	Ranged enemy	Randomised tiles	
Stationary enemy	Main menu	Sound effects	
Light ability	Lives	Score mechanics	
Enemy scan	Environmental tiles		
End-of-level torch			

3.3 Product Documents

3.3.1 Customer Meeting

I. Presentation

1

2

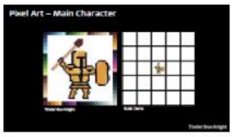
Tinder Box Knight
17 Nov 2021 | 3rd Customer Meeting | Design
3

NEW UPDATES

ART | Main Character
Monster *1
Torch
Pressure Plat *3
Stone Floor

STORY | Opening Story
Caption

Tinder Box Knight
4


Pixel Art – Main Character
Tinder Box Knight
5

Concert Art – Monster1

Tinder Box Knight
6

Pixel Art – Monster1

Tinder Box Knight
7

Pixel Art – Torch
Tinder Box Knight
8

Concert Art - Pressure Plate

Tinder Box Knight
9

Pixel Art - Pressure Plate

Tinder Box Knight
10

Concert Art - Floor

Tinder Box Knight
11

Tinder Box Knight Demo
Demo Structure
Demo Player Control Features
Tinder Box Knight
12

Demo Enemy Types

13

Story

14

Caption

15

The presentation for this week had three sections. Slides 5-11 show the pixel art assets that had been developed during the previous week. These were the second version of the knight, the spider, the end-of-level torch, and some ideas for the floor tiles. Slides 12 and 13 show what we will be aiming to deliver next week as part of the first working demo. Slides 14 and 15 show the latest version of the story was presented.

II. Questions

1. Are there any features you would specifically like to see in the demo?
    a. I think I have made my bottom line clear - you should have something with a beginning, middle, and end. Anything you can have on top of that is great.

III. Analysis

The main takeaway from the meeting is that the demo should be a level in miniature, with a defined start and end point. The customer said they wanted a level where you can start the game, play part of the level, and then come to an end, so there is a beginning, middle, and end to the gameplay. The customer indicated that this would be their preference over a demo which simply showcases multiple features but does not have a defined structure. When making the demo during the coming sprint, therefore, we will ensure to have a way for the player to complete the level (even if the interactions of the level is significantly different to the final game). The customer also advised us not to have too long of a story before the game starts. It should be enough to tell what the player what is going on and engage them, but not so long that it bores a repeat player.

IV. Key takeaway

- The demo version of the game should have the character moving within the game environment and able to perform some limited interactions.

3.3.2 User Stories

US-20	Wall	As a player, I want there to be walls in the level that block the character so that the route to the torch is not obvious.	Tiles exist in the level that the player cannot walk on.
US-21	Gate	As a player, I want to be able to open gates in the wall by finding switches in the dungeon, so that I can progress to the next section.	The player can activate a switch and can then walk on the tile where the gate was.

Updates: US-9 has been dropped and replaced with US-20 and US-21. These give the same incentive to the player to fully explore the dungeon in order to open up new paths to the final torch.

3.3.3 Use cases

I. Requirements Use Cases

UC-7-r
Title: Ranged enemy interaction
Creation date: 18/11/2021
Cross-references: US-17
Level: Level interaction
Context:

- If the player stands on a lit tile (through movement or lighting a torch) on the same row as a shadow enemy, they will lose a life.

Frequency: Whenever the player enters the monster's field of view.

UC-8-r
Title: Walls and gates
Creation date: 18/11/2021
Cross-references: US-20, US-21
Level: User interaction
Context:

- Allow the player to press 'o' when standing on a specific tile to replace a gate in a wall with a tile the player can walk on.

Frequency: Whenever the player presses 'o'.

II. Design Use Cases

UC-7-d
Title: Ranged enemy interaction
Author: Seth
Creation date: 20/11/2021
Purpose: Player interaction with ranged enemy.
Overview: When the player walks in front of the ranged enemy while standing on a lit square, the ranged enemy attacks.
Cross references: US-16, UC-1, UC-3, CRC-8
Actors: Player, Ranged enemy
Pre-condition:

1. The player must be currently in a level.
2. The player-character must not in the middle of another activity.
3. The player has at least one life.
4. The level has a ranged enemy within it.

Post-condition:

1. The number of lives is decreased by one.
2. The player respawns at the beginning of the level.
3. The enemy that attacked the player disappears.

Normal flow of events:

1. The player moves into a lit square (UC-1).
2. A check is run whether the player's square within view of the ranged enemy (ranged enemies can face north, south, east, and west). The check returns true if the enemy is facing the player and the player is on the same row or column and standing on a lit square.
3. Text displays telling the player that they have been spotted by the enemy.

4. The number of lives the player-character has decreases by one. A check is run that the number of lives is above zero.
5. The player-character sprite and camera are moved to the start of the level. The enemy that attacked the player disappears from the level, but otherwise the level is the same.

Alternative flow of events:

1. The square the player moves into is dark. No check is run and use case ends.
4. The number of lives equals zero. A message tells the player that they have failed the level and gives them the option to retry it. The player-character is moved to the start of the level. The rest of the level is reset - all enemies are replaced and all changes the player made (placing torches, etc) are removed (identical to UC-3).

UC-8-d
Title: Walls and gates
Author: Seth
Creation Date: 20/11/21
Purpose: To allow the player to open a gate in a wall in the level.
Overview: When the player walks on a specific tile in the level, a gate in a wall in the level opens, allowing the player to get to the next section.
Cross-reference: US-20, US-21
Actors: Player
Dependencies: UC-2, UC-3
Pre-condition:

1. The player is in a level.
2. The level has a wall in it.
3. The wall has a gate tile in it.

Post-condition:

1. The gate in the wall is replaced by a tile the player can walk on.

Normal flow of events:

1. The player moves on a tile that is visually distinct from a normal tile - the tile will only be visually distinct when lit.
2. The player presses the ‘o’ button to activate the tile.
3. Text displays above the player saying that something in the level has changed.
4. The gate tile within the wall is replaced by a different tile that the player can walk on.

Alternative flow of events:

2. The player is not on a pressure plate tile. Nothing happens and use case ends.

UC-2-d (updated)
Title: Moving the character
Author: Seth
Creation date: 24/11/2021
Purpose: Allow the player to move the character.
Overview: When the player presses a specified movement key, the character will move to a new tile on the screen.
Cross references: US-1
Actors: Player
Dependencies: UC-1
Pre-condition:

1. The player must be currently in a level.
2. The player-character must be in an idle state, not in the middle of another activity.
3. The player has at least one life.

Post-condition:

1. The player-character will be in a grid tile above the starting tile.

Normal flow of events:

1. The player presses the up arrow on the keyboard.
2. Check that the tile above the player is a valid movement spot.
3. Check that the new tile does not contain a monster.
4. Check that the light on the tile is the same as the current tile (i.e. if the knight is standing on a dark square, the new square is also dark).
5. The player-character sprite is moved from the centre of the initial tile to the centre of the tile above.

Alternative flow of events:

1. The player presses the right arrow. The tile checked is the one to the east, and the character will move to the to the east.
1. The player presses the down arrow. The tile checked is the one to the south, and the character will move to the to the south.
1. The player presses the left arrow. The tile checked is the one to the east, and the character will move to the to the west.
2. The intended tile is blocked by another object. The player-character stays on the tile where they currently are.
3. The tile contains a monster. See UC-4-r, alternative flow.
4. The new tile is different to the current tile (i.e. lit if the player is currently on a dark square). The image for the knight tile is updated to match the lighting value of the new time. Continue to 5.

3.3.4 CRC Cards

A separate class has been made for the scan function, to avoid overloading any other class.

CRC-7: Scan	
Responsibility	**Collaboration**
• Display the number of monster tiles in the 5x5 grid of tiles centred on the player.	• Knight • Main

3.3.5 Tests

Tests have been written in the form of test cases, which will provide the information for a human tester to carry them out. Tests will focus on functionality testing, to ensure that the intended result is displayed on the screen. Unit tests at the level of checking return values of methods have not been included. Because the system is very user-oriented - i.e., there are few parts of the system which do not result in an obvious visual change from the point of view of the user if they work correctly - it was felt that it was possible to test everything with full coverage just by verifying the changes visually. This also has the advantage that we are not having to build extra parts into the code that are purely for testing purposes, so the code base should be kept neater. The tests will also not be automated. At this stage, it is the responsibility of the developer assigned to the feature to run the relevant tests. In later weeks when there are more components to the system, running through all existing tests will be a specific item on the backlog. It was felt that due to the size of the project and the limited time for development, automating the tests would not produce enough return on the initial time invested.

One further item to note is that where tests are very similar (for example, the tests checking for movement in the four directions), they have been combined into four separate variants of the same test (e.g. T-1a, T-1b…). If it is said that the test T-1 is passing with no qualifier, it can be assumed that all sub-tests are passing.

T-1a
Name: Normal movement test.
Description: Test that when the player presses the up arrow, the knight moves one square north.
Cross-references: US-1, UC-2, CRC-1, B-20

Pre-conditions:

1. The player is in a level.
2. The player is not in the middle of another activity.
3. The knight is not standing on a tile that borders the north barrier of the level.
4. The tile to the north of the knight is a normal tile that can be walked on,

Test steps:

1. Press the up arrow.

Expected result:

1. The knight should appear in the square to the north of where they were before.

T-1b

Name: Normal movement test.

Description: Test that when the player presses the right arrow, the knight moves one square east.

Cross-references: US-1, UC-2, CRC-1, B-20

Pre-conditions:

1. The player is in a level.
2. The player is not in the middle of another activity.
3. The knight is not standing on a tile that borders the east barrier of the level.
4. The tile to the east of the knight is a normal tile that can be walked on,

Test steps:

1. Press the right arrow.

Expected result:

2. The knight should appear in the square to the east of where they were before.

T-1c

Name: Normal movement test.

Description: Test that when the player presses the down arrow, the knight moves one square south.

Cross-references: US-1, UC-2, CRC-1, B-20

Pre-conditions:

1. The player is in a level.
2. The player is not in the middle of another activity.
3. The knight is not standing on a tile that borders the south barrier of the level.
4. The tile to the south of the knight is a normal tile that can be walked on,

Test steps:

1. Press the down arrow.

Expected result:

1. The knight should appear in the square to the south of where they were before.

T-1d

Name: Normal movement test.

Description: Test that when the player presses the left arrow, the knight moves one square west.

Cross-references: US-1, UC-2, CRC-1, B-20

Pre-conditions:

1. The player is in a level.
2. The player is not in the middle of another activity.
3. The knight is not standing on a tile that borders the west barrier of the level.
4. The tile to the west of the knight is a normal tile that can be walked on,

Test steps:

2. Press the down arrow.

Expected result:

2. The knight should appear in the square to the south of where they were before.

T-2

Name: Moving into a border

Description: Test that the player cannot move outside of the level.
Cross references: US-1, UC-2, CRC-1, B-20
Pre-conditions:

1. The player is in a level.
2. The player is not in the middle of another activity.
3. The knight is standing on a tile that borders the north barrier of the level.

Test steps:

1. Press the up arrow.

Expected result:

1. The knight does not move.

T-3

Name: Lighting the final torch
Description: Test that when the torch is lit, the level ends.
Cross references: US-13, UC-6, B-17
Pre-conditions:

1. The player is in a level.
2. The player is not in the middle of another activity.
3. The player is in the square next to the torch.
4. The torch is in an unlit state.
5. The player is not in the final level.

Test steps:

1. Press the space bar on keyboard.

Expected result:

1. The torch tile picture changes into a lit torch picture.
2. The next level is displayed on the screen.

T-4

Name: Player resets on spider interaction
Description: Test that text appears and the player resets when the player walks into an enemy.
Cross references: US-2, US-8, UC-4, B-16
Pre-conditions:

1. The player is in a level.
2. The player is not in the middle of another activity.
3. The player is in a square immediately south of a spider.
4. The player has at least two lives (once implemented)

Test steps:

1. Press the up-arrow key.

Expected result:

1. Text is flashed on the screen saying, “You hit a monster!”.
2. The knight tile returns to the starting square.

T-5

Name: Walking into a spider.
Description: Test that the discovered spider by knight/player becomes hidden again.
Cross-references: US-2, UC-4, B-16
Pre-conditions:

1. The player is in a level.
2. The player is in a square immediately south of a spider.
3. The player has at least two lives (once implemented).

Test steps:

1. Press the up-arrow key to move player into the spider tile.

Expected result:

1. The discovered spider tile becomes dark/hidden again when the player respawns.

T-6
Name: Lighting up surrounding squares
Description: Test that on user input the player can light up squares surrounding their position.
Cross-references: US-5, UC-3, CRC-2, B-14
Pre-conditions:

1. The player is in a level.
2. The player is not on a border tile.

Test steps:

1. Press the f key.

Expected result:

1. The 3x3 around the player switches from dark tile variants to light tiles.

T-7
Name: Checking light on board
Description: Test that only those tiles on the level board change when the light is used.
Cross-references: US-5, UC-3, CRC-2, B-14
Pre-conditions:

1. The player is in a level.
2. The player is bordering the edge of the level.

Test steps:

1. Press the f key.

Expected result:

1. Only the tiles within a 3x3 grid and that are within the level change to light variants.

T-8
Name: Scanning for enemies
Description: Test that on user input the player see the number of surrounding enemies.
Cross-references: US-12, UC-4, CRC-7, B-15
Pre-conditions:

1. The player is in a level.
2. There are two spider tiles in the 5x5 grid centred on the character.

Test steps:

2. Press the s key.

Expected result:

2. The number ‘2’ will appear on the board.

T-9
Name: Quitting the game.
Description: Test that on user input the player can quit the game.
Cross-references:
Pre-conditions:

1. The player is in a level in the game.

Test steps:

1. Press the q key.

Expected result:

1. The game will quit.

3.3.6 Design Elements

None this week.

3.4 Review

Almost all tasks this week were focused on code implementation. As we were on quite a tight timescale for getting the first demo ready, all members of the team but one took on a coding task (the spare person worked on further art design work). These tasks were based on the prioritisation table in 2.3.7, and the specific tasks were discussed in meeting one. The difficulty of the code for each task was estimated by team consensus, so that the difficulty assigned to each task was not just one person's view. Members of the team who wanted to do specific tasks then volunteered to do them (or the opposite, and indicated they did not want to do them). The remainder of the tasks were then assigned, taking into account the skills audit in 1.3.7 so that members with more programming experience took on harder tasks. This system worked well, so will be used for allocating tasks for future weeks. Although features were allocated individually and formal pair programming will not be used, the elements of the system interact to a degree that most coding will be done collaboratively. The plan for how the system would interact was discussed in Meeting Two (the produced visual is attached in 3.2.1). This meant that team members understood how their feature related to others and who they would need to work most closely with.

By the meeting on Friday, most members had thought out how to implement their features. To allow for new levels to be created easily, we are using a system where levels are created in a text file formatted like a csv file (an example can be seen in 3.2.1 Meeting Two). Each entry is a tile in the game, with all tiles being represented by a unique two (occasionally three) letter code. Each line on the text file corresponds to a row in the game level. The text file is then read in and stored as a two-dimensional list within the game's memory. This has the advantage of making the level creation process easy. Within this framework it would even be possible to allow users to create their own level in the game, although we will not be able to implement this in the time we have. Once this had been completed, coding the other features could begin. Most of these features involve interacting with the list by changing individual values to changing tiles in the level.

The team faced three problems during this sprint, documented in 3.2.4. All three were coding bugs. Two (EX-1 and EX-3) were introduced by how the level was initially displayed, although only EX-1 had a direct impact on progress this week. This was solved by adding the step of reading the text file to an array, discussed above. EX-3 has a workaround, so due to the volume of work this week it will be postponed until a later sprint. EX-2 was introduced during the coding of features and will be fixed in the following Sprint 4. A more general problem this week was the amount of work scheduled - almost everybody was working on significantly-sized new coding tasks. This meant there was very little reserve capacity to address bugs, hence why fixing two of the exceptions have been postponed. In later weeks we will aim to have fewer people working on smaller coding tasks so that we have the capacity to address issues as they arise. However, this was not possible this week due to having to deliver the demo to the customer on time.

Sprint 4 (24/11 - 01/12)

4.1 Overview

The aim this week is to finish the features of the demonstration level so that in the next customer meeting we can present a level featuring almost all the elements that will be present in the final game. Tasks therefore mainly fall into two strands. One group of tasks focuses on fixing the bugs and polishing the features from the level we presented in the customer meeting at the start of this week - these included B-14, B-15, B-17, and B-20. The second group of tasks aim at designing and implementing several new features (B-24, B-25, B-26). Alongside this are the normal non-coding tasks.

4.2 Process Documents

4.2.1 Meeting Minutes

Meeting One

Date: 24/11/2021
Time: 13:45-15:00 (including customer meeting).
Attendance: Yunni, Zening, Yijin, Devendra, Adil, Karan, Seth, Rob, (Peiwen: away for the day).
Agenda:

- Review work produced in Sprint 3.
- Ensure that the demo level is working to show customer.
- Plan tasks for coming week.

Outcome:

- Work on the demo that had been done the previous week was reviewed. Tasks had gone well but due to the short development time several were not fully polished and there were a few minor bugs. Fixing these bugs were added to the product backlog for this sprint (4.2.3).
- There was a brief reminder on coding standards, principally the use of separate class files to implement functionality wherever possible. This is to keep the design modular and prevent the main file from becoming too large.
- The demo level was tested on a laptop before showing the customer.
- Tasks for next week were agreed. These fell into two sets: fixing the bugs and refactoring the code from the previous sprint, and implementing new features. The new features will be a ranged enemy and pressure plate that opens a wall in the level.
- The decision was taken to stop using Jira, as the team is small enough that allocating tasks on Teams and using the documentation is sufficient. Members working on code have preferred to put a message on Teams rather than update Jira, as it will be seen sooner by everyone else. This will therefore be the process from now on.

Action items:

- Add agreed tasks to backlog.

Customer Meeting
Date: 24/11/2021
Time: 14:15-14:30 (including customer meeting).
Agenda:

- Present demo level to customer.
- Discuss plan for the coming weeks.

Outcome:

- The demo level was presented to the customer. Features that were presented included level generation, movement, scanning nearby enemies, being attacked by an enemy, and lighting the torch to finish the level.

- We discussed the exceptions which had occurred during Sprint 3 that the customer should be aware of. These were EX-1, as they will affect future development, as well as EX-3 as that will affect how the end user will see the game.
- A plan for the next two weeks was agreed. In the next customer meeting we will present a fully realised demo level, containing most of the features of the final game without the bugs present in the current demo level. In two weeks, we will present the next major version of the game; this will include the currently missing 'game infrastructure' like a menu to start the game and change levels from.

Action items:

- Update work plan.

Meeting Two

Date: 27/11/2021
Time: 12:00-12:10 (online)
Attendance: Yunni, Yijin, Devendra, Adil, Karan, Seth, Rob, Peiwen
Agenda:

- Review work done so far.
- Ensure everyone is happy with their tasks.
- Discuss ideas for next customer meeting.

Outcome:

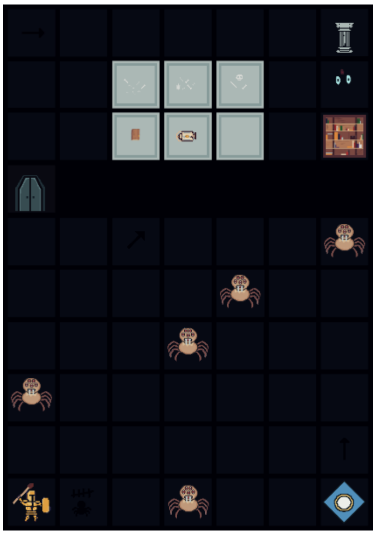
- The present state of the level was presented (see attached items).
- Each member of the team individually discussed their task this week, how they were getting on with it, and any problems that had run into. Below are each member's tasks and their progress:
  - Seth: Update movement system to fix EX-3 and work with the new tiles (completed), start documentation for sprint 4, spend time refactoring code to make future work easier.
  - Rob: Implement new tiles (completed, although with bug: see item below), implement the ranged enemy.
  - Karan: Update spider to work with the light feature, complete the end of level torch functionality.
  - Peiwen: Fix bugs in the light function.
  - Yunni: Implement new tile behaviour (pressure plate and wall gate).
  - Devendra: Update spider to work with the light feature.
  - Adil: Documentation - finish sprint 3, work on sprint 4.
  - Yijin: Design new tiles (completed), sprint 4 documentation.
  - Zening (absent from meeting but messaged the team later): Fix bugs in light function, design the start menu to be implemented next week.
- Rob discussed an error message he had run into when implementing the new tiles and asked the rest of the team to have a look and see if they could see what was causing it.
- Seth presented some ideas about the next customer meeting. It was suggested that we discuss the 'bot' mentioned in the initial requirements and US-6 with the customer and suggest to the customer that this requirement be dropped as the game has evolved in a different direction. Instead, to build replay value into the game we can have an option that generates a random level at a specific difficulty setting. These suggestions were agreed by the rest of the team.

Action items:

- Add agreed tasks to backlog.
- Add tile error to exception handling.

Attached items:

1. Level with the new tiles implemented. Items that are on non-lit tiles would normally be invisible to the player but are visible here for demonstration purposes.



4.2.2 Work Plan

Cycle	Sprint	Phase	Planned Milestone	Achieved
1	03/11-10/11	Pre-game		N/A
2	10/11-17/11	Pre-game	Plan for first prototype	Yes
3	17/11 - 24/11	Game	Delivery one - demo	Yes, although with bugs.
4	24/11 - 01/12	Game	Fully realised first level	
5	01/12 - 08/12	Game	Delivery two - multiple levels	
6	08/12 - 15/12	Game		
7	15/12 - 20/12	Post-game	Final hand in	

4.2.3 Backlog

Backlog Item	Name	Cross references	Responsibility	Comments	Status at start of cycle
B-11	Sprint 2 Documentation		Seth, Yijin, Peiwen		Completed
B-12	Sprint 2 Review		Seth		Completed
B-13	Review and finish sprint 1 documentation		Seth		Completed
B-14	Light System	US-3, UC-5	Peiwen, Zening	Bug fixes needed. Zening added to task.	Rolled over
B-15	Scan system	US-12, UC-4	Zening		Completed

B-16	Spider	US-2, UC-4	Devendra	Interaction when player walks into square is done; need to add interaction when the square is lit up.	Rolled over
B-17	Big torch	US-13, UC-6	Karan	Minor polishing needed. Task changed to Karan.	Rolled over
B-18	Text system	US-8	Karan		Completed
B-19	Buttons	UC-1	Adil	Task proved harder than expected - requirement for the buttons to be mouse-clickable have been dropped	Dropped
B-20	Character movement	US-1, UC-2, EX-3	Seth	Fix bug documented in 3.2.4 EX-3.	Rolled over
B-21	Update floor pixel art		Yijin		Completed
B-21	Sprint 3 review write-up		Adil		New
B-22	Sprint 4 documentation		Seth, Yijin, Adil		New
B-23	Plan next set of features		Seth	Continuing for Sprint 5’s features.	Rolled over
B-24	Ranged enemy	US-16, UC-8	Rob	Including tile design and code.	New
B-25	Obstacle tiles	US-19, US-20, UC-9	Rob, Yijin, Yunni (code)	Wall that the player cannot move through and a pressure plate that opens a gate in the wall. Including tile design and code.	New
B-26	Environment tiles		Rob, Yijin, Yunni	Pixel art for new tiles to add variation to the level.	New
B-27	Fix level size bug	EX-4	N/A	Postponed until next week as there is a work-around for now.	New
B-28	Design start menu	US-17, UC-7	Zening		New

4.2.4 Exception Handling

EX-2
Status: Open → closed.

EX-4
Problem: Running the game with the new tiles results in a TypeError.
Details: The new tiles were added to the game in the same manner as previous tiles. Images for the tiles are put into a folder. Within the game each tile has a two or three letter identifying code which is mapped to a more descriptive name, which is then used to fetch the image matching the name. However running the game with the new tiles resulted in 'TypeError: join() argument must be str, bytes, or os.PathLike object, not 'tuple'.
Handling: The error was brought to the attention of the team during meeting 2. Shortly after the meeting Zening found that one of the new names for the tiles had been accidentally implemented as a tuple. Fixing this resolved the error.
Status: Closed.

EX-5
Problem: Using the light ability on the right-most column of tiles causes an out of bounds error.
Details: During the implementation of the check for whether the player has lit up a spider, changes were made that meant that if the player pressed f (to light up the surrounding squares) the game would crash, citing an out-of-bounds error.
Handling: Members responsible for the spider implementation tried to fix it but were not able to before the end of the sprint. The bug has been added to the agenda for the review meeting at the start of the next sprint to discuss it with the rest of the team.
Status: Open

4.2.5 Prioritisation table

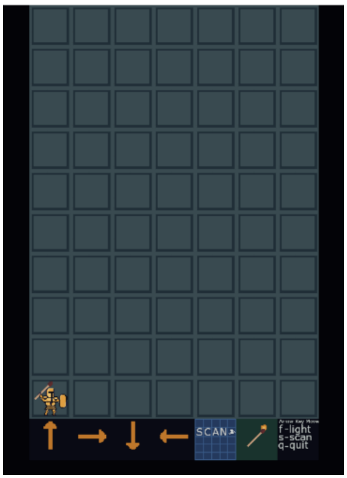
Must have	Should have	Could have	Won't have
Movement	Clue tiles	Usable items	Sound-based mechanics
Level grid	Ranged enemy	Randomised tiles	
Stationary enemy	Main menu	Sound effects	
Placeable light	Lives	Score mechanics	
Enemy scan	Environmental tiles (including obstacles)		
End of level torch			

4.3 Product Documents

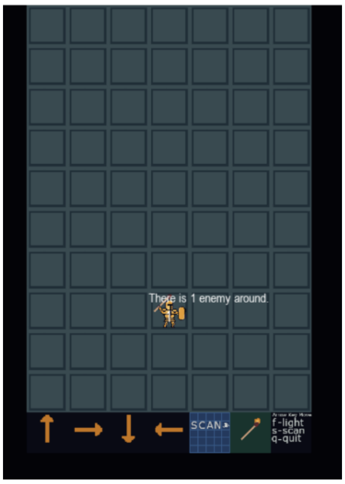
4.3.1 Customer Meeting

I. Presentation

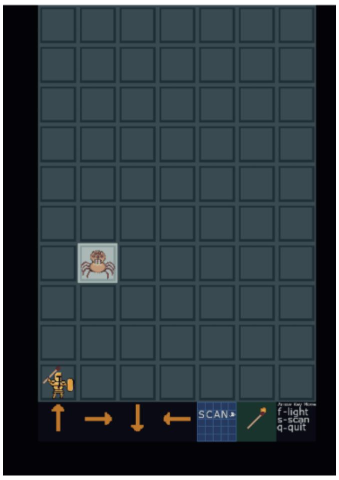
There was no presentation this week; instead, we presented the demo level to the customer. Below are screenshots of the level as it was presented.



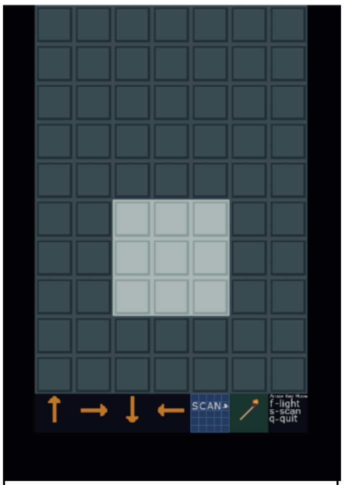
Screenshot 1: The opening of the game, featuring the knight in the starting square.


There is 1 enemy around.

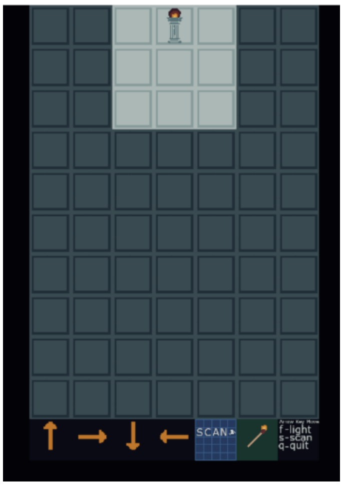
Screenshot 2: The knight has moved and used the scan feature.



Screenshot 3: The knight has run into a spider; they have been reset back to the starting square the spider has been lit up.



Screenshot 4: The result of using the scan function. The knight was at the central tile of the lit 3x3.



Screenshot 5: Using the light within range of the torch reveals it.

II. Questions

1. Are there any features you particularly want to see in the next iteration of the game?
   a. I need to know what the options are… [discussion of options for the coming week]. Ok, so it's a matter of filling out the current level before going on to the next one. That sounds good to me.

III. Analysis

The customer reacted well to the demo, with positive comments about the scan feature, saying it was a good way to give feedback. The level we presented seemed to be in line with what the customer was expecting, while also helping to clarify with the customer exactly what the end game will be like. Unfortunately, due to the time spent developing this week we had not had a chance to produce a presentation like normal, and for this reason the meeting was less structured than normal. However, we still received some good feedback on the level. The customer also made a comment about levels being different shapes, which has been written as a user story for this sprint to look into in a later sprint.

IV. Key takeaway

- Look into varying the level shape.

4.3.2 User Stories

US-22	Level shape	As a player, I want the levels to be different shapes, so that each level feels different.	At least two different level shapes exist.

4.3.3 Use cases

New use cases have been added for the start menu feature. The lives use case has been updated to include a visual representation of the player’s lives.

I. Requirements Use Cases

UC-9-r
Title: Start menu
Creation date: 26/11/2021
Cross-references: US-14
Level: System framework
Context:

- Display a start menu to the player.
- Allow the player to pick an option from the menu and start the game.

Frequency: Whenever the game starts or the user quits a level.

II. Design Use Cases

UC-9-d
Title: Start menu
Author: Seth
Creation date: 30/11/2021
Purpose: Allow the player to start the game.
Overview: When the player opens the game a menu appears which allows them to start the game.
Cross references: US-14, US-14
Actors: Player
Pre-condition:

1. The player has opened the game.
2. The start menu is currently on the screen.
3. The cursor is at the default position (the top option).

Post-condition:

1. The screen changes to display the first level.

Normal flow of events:

1. The start menu displays on the screen. There are two options: ‘start game’ and ‘end game’.
2. The player presses enter to select start game.
3. The start menu disappears and is replaced by the first level.

Alternative flow of events:

2. The player presses the down arrow, to move the cursor to ‘quit game’, and presses enter. The game quits.

UC-4-d (update)
Title: Lives
Author: Seth
Update Date: 30/11/2021
Purpose: Allow for enemies to attack the player.
Overview: When the player finds an enemy, they lose a life. A tile is changed on the HUD to show the lives left.
Cross references: US-4, UC-1, UC-2
Actors: Player, enemy
Pre-condition:

1. The player is currently in a level.
2. Enemies have been placed in the level.
3. The player has at least one life.

Post-condition:

1. The number of lives is decreased by one.
2. The player respawns at the beginning of the level.
3. The enemy that attacked the player disappears.
4. The lives tile changes to a tile showing one fewer lives.

Normal flow of events:

1. The player moves into a dark square (UC-1).
2. The player lights up the dark square (UC-2). In one of the tiles in the lit up 3x3 grid, there is an enemy sprite.
3. Text displays telling the player they have found an enemy.
4. The number of lives the player-character has decreases by one. A check is run that the number of lives is above zero.
5. The player-character is moved to the start of the level. The enemy that attacked the player disappears from the level, but otherwise the level is the same.
6. A tile at the bottom of the screen changes from one with the previous number of lives on to one with one fewer lives on.

Alternative flow of events:

2. The square the player moves into contains an enemy. Jump to 3 and continue from there.
4. The number of lives equals zero. A message tells the player that they have failed the level and gives them the option to retry it. The player-character is moved to the start of the level. The rest of the level is reset - all enemies are replaced and all changes the player made (placing torches, etc) are removed.

4.3.4 CRC Cards

CRC-8: Ranged Enemy	
Responsibility	**Collaboration**
• Attack character when they walk into the enemy, causing them to lose a life. • Attack the character when they walk on the same row or column as the ghost on a light square.	• Main game

CRC-9: Pressure Plate	
Responsibility	**Collaboration**
• Store location of pressure plate and gates in the level.	• Main game

• Test whether player is standing on the plate when 'o' is pressed. • Open the corresponding date	

4.3.5 Tests

T-10
Name: Updated movement test.
Description: Test that when the knight moves onto a square then off it, the square returns to what it originally was.
Cross-references: US-1, UC-2, CRC-2, B-20
Author: Seth
Pre-conditions:

5. The player is in a level.
6. The player is not in the middle of another activity.
7. The tile to the north of the knight is a lit clue tile.

Test steps:

3. Press the up arrow.
4. Press the down arrow.

Expected result:

3. The knight should appear in the square to the north of where they were before after the first button press, then should move back to the original square on the second.
4. After the second button press the lit clue tile should be in its original position.

T-11
Name: Light and dark movement test
Description: Test that when the knight moves from a dark square to a light square and back, the knight's tile changes accordingly.
Cross-references: US-1, UC-2, CRC-2, B-20
Pre-conditions:

1. The player is in a level.
2. The player is not in the middle of another activity.
3. The knight currently has a dark background.
4. The tile to the north of the knight is a blank lit tile.

Test steps:

1. Press the up arrow.
2. Press the down arrow.

Expected result:

1. After the first press the tile to the north of the original position of the knight should contain a knight with a lit background.
2. After the second button press the knight should move back to original tile, and have a dark background.

T-12
Description: Test that the ranged enemy attacks player when in the light.
Cross-references: US-16, UC-6, CRC-8, B-23
Pre-conditions:

1. The player is in a level.
2. The knight currently has a dark background.
3. There is a hidden ranged enemy on the same column as the player.
4. There are no walls between the enemy and the player.

Test steps:

1. Press the f key

Expected result:

1. The 3x3 grid centred on the player lights up.
2. The knight tile and the ranged enemy tile are highlighted in purple.
3. The knight resets to the starting position and the ranged enemy returns to normal.

T-13
Description: Test that the ranged enemy does not attack the player when there is a wall in the way.
Cross-references: US-16, UC-6, CRC-8, B-23
Pre-conditions:

1. The player is in a level.
2. The knight currently has a dark background.
3. There is a hidden ranged enemy on the same column as the player.
4. There is a wall tile between the enemy and the player.

Test steps:

1. Press the f key

Expected result:

1. The 3x3 grid centred on the player lights up.
2. There is no change to the ranged enemy.

T-14
Description: Test that the player can open the gate in the wall.
Cross-references: US-19, US-20, UC-7, B-25
Pre-conditions:

1. The player is in a level.
2. There is a lit pressure plate tile immediately west of the player.
3. A gate tile is visible in the level.

Test steps:

1. Press the right key to move the player on to the pressure plate.
2. Press the o key.

Expected result:

1. The gate should be replaced with an open gate tile.

T-15
Description: Test that the ranged enemy does not attack the player when it has already been lit up.
Cross-references: US-16, UC-6, CRC-8, B-23
Pre-conditions:

1. The player is in a level.
2. The knight currently has a dark background.
3. There is a visible ranged enemy on the same column as the player.
4. There is no wall tile between the enemy and the player.

Test steps:

1. Press the f key

Expected result:

1. The 3x3 grid centred on the player lights up.
2. There is no change to the ranged enemy and the player does not lose a life.

T-16
Description: Test that the game opens with a start menu.
Cross references: US-17, UC-7, B-28
Pre-conditions:

1. The user has an open command terminal at the folder containing the program.

Test steps:

1. Run ‘python main.py’

Expected result:

1. A menu displaying the option to start the game and exit appears on the screen.

T-17
Description: Test that the start menu allows the player to start the levels.
Cross references: US-17, UC-7, B-28
Pre-conditions:

1. The user has the start menu open on the screen.
2. The cursor is showing as selecting ‘start game’

Test steps:

1. Press the enter key.

Expected result:

1. The first level of the game appears.

T-18
Description: Test that the start menu allows the player to exit the game.
Cross references: US-17, UC-7, B-28
Pre-conditions:

1. The user has the start menu open on the screen.
2. The cursor is showing as selecting ‘exit’

Test steps:

1. Press the enter key.

Expected result:

1. The game exits and the user returns to the normal computer screen.

T-9 (updated)
Name: Quitting a level.
Description: Test that on user input the player can quit a level and return to the start menu.
Cross-references: US-17, UC-7, B-28

Pre-conditions:

1. The player is in a level in the game.

Test steps:

1. Press the q key.

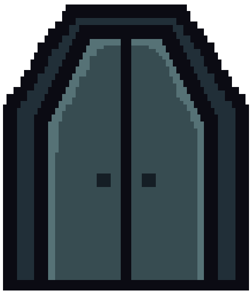
Expected result:

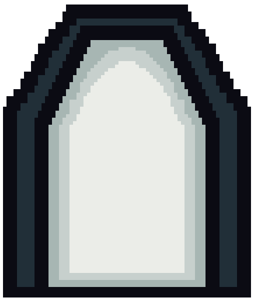
1. The level will be replaced by the start menu.

4.3.6 Design Elements

Design elements were produced as part of B-24, B-25 and B-26.

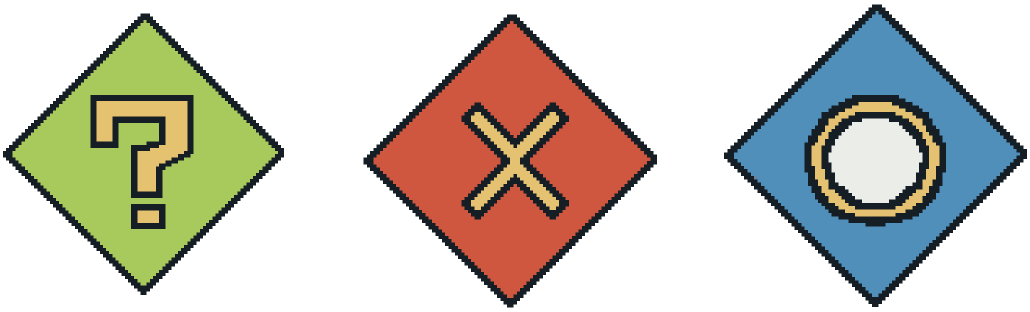
I. Gate (closed and open)





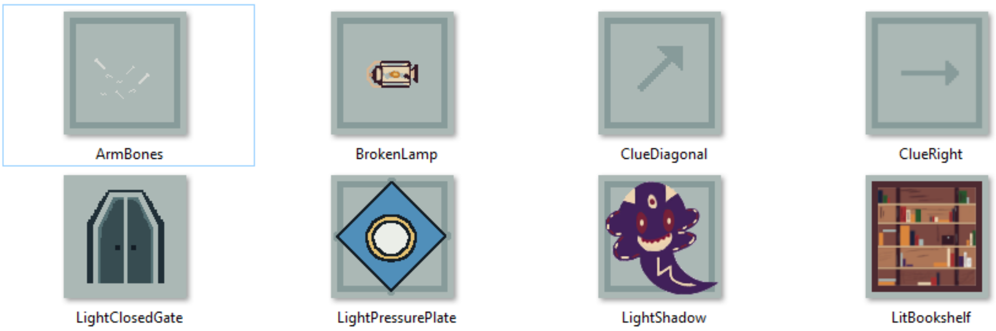
II. Pressure plates (three variants)

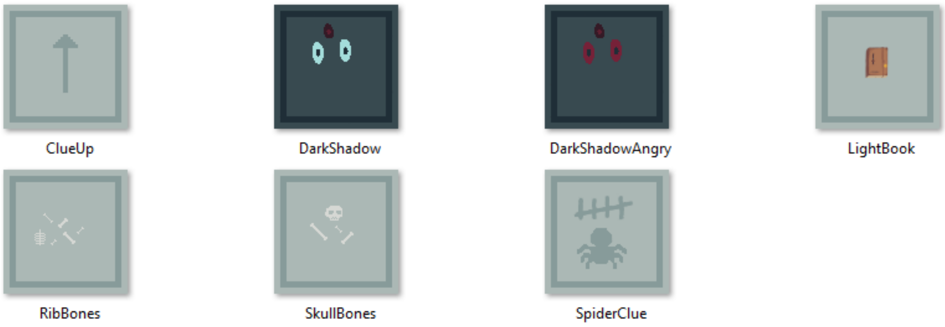
Three variants of the pressure plate were produced and submitted to the team for review. It was chosen to use the plate, as it fit the colour scheme of the game the best. The other plates will be kept as spares for future features.



III. New tiles (including tile versions of the blue pressure plate and closed gate)

The gate, shadow (ranged) enemy, and the pressure plate had tile versions made for inclusion in the game. Other tiles were made to provide visual variety to the level and give the player clues about the level.





4.4 Review

Compared to last week, tasks this week were more balanced between coding and non-coding tasks. Although the progress made in Sprint 3 was good, the very tight timeframe to produce the initial protype of the game inevitably meant that several areas were left unfinished, and bugs were present. The first set of coding tasks this week therefore aimed to polish and finish the features already present in the level - these were principally for the light feature, the spider, and the movement. Members who worked on the relevant task in Sprint 3 and were happy to continue to finish it in Sprint 4 were re-allocated the task. In choosing to do this, we were aware of the risk that knowledge within the team may become segmented and members may only have a full understanding of the features they have worked on while remaining unfamiliar with other parts of the system. This would cause problems by introducing multiple single points of failure in the case that a team member was unable to work, and also slow down the testing and debugging process later. However, for this sprint it was felt that the advantage gained by having the people most familiar with the section work on completing the feature was worth it, as it would avoid the time delay of a new person having to understand the code already written before they could fix it. In future weeks we will ensure to rotate tasks to avoid making this risk worse. For several tasks members did not want to continue; in this case another team member self-assigned themselves to the task.

The second group of coding tasks related to adding new features to the demo level to enrich and enhance it. There were two significantly sized tasks to complete here - implementing the ranged enemy (US-16) and adding tiles that would create barriers within the level as well as the functionality for the player to open the barriers (US-17). Additionally, there was a task to add non-interactive tiles to add visual variety to the level. Each of these tasks had a lead member responsible for coding the feature, as well as extra members to work on the design of the new tiles. Finally, work began on designing a start menu for the game to be integrated with the code next week.

Progress this week was generally smooth. By the second meeting, the movement had been fixed. This was important as it allowed the other features to be tested more easily. Designs had also been made for the new set of tiles and implementing them within the game had begun. As we discussed in the meeting with the customer at the start of the Sprint, much of the work last week was invested in making the tile system more straightforward to interact with. The results of this were shown this sprint, as coding progresses faster and with fewer bugs.

Although there were fewer errors this week, members working on the code still encountered a couple. These have been documented in 4.2.4 Exception Handling. The first, EX-4, was discussed at the second meeting and a solution was quickly found by another member of the team after the meeting. The second, EX-5, was found later in the week, so we were unable to fix it before the end of the sprint. Fixing the bug will be a high priority next sprint as the bug does not just result in non-functionality but causes the game to crash.

Sprint 5 (01/12 - 08/12)

5.1 Overview

As agreed in the customer meeting at the start of Sprint 4, work this week is mainly aimed at building the level infrastructure to allow for multiple levels to be played sequentially. This is the penultimate full sprint of the project, so by the end the game should be in a relatively complete state with only minor additions to be made in the final week. By the end of this sprint, we are aiming to have a game which opens with a start menu, from which the player can start the first level and then play through multiple levels, or simply select a level that they wish to play. New tasks added to achieve this include B-29, B-30, and B-31. We will also implement some small requirements which are not presently in the game, like US-3 (Lives).

5.2 Process Documents

5.2.1 Meeting Minutes

I. Meeting One
Date: 01/12/2021
Time: 1800-1815 (online)
Attendance: Yunni, Zening, Peiwen, Devendra, Rob, Karan, Seth, Yijin
Agenda:

- Review previous week’s work.
- Plan tasks for coming week.

Outcome:

- Progress made on tasks from the previous week was reviewed. Karan and Devendra outlined the bug that they had been working on so the rest of the team was aware (see 4.2.4 EX-6).
- The importance of writing and uploading tests was discussed again.
- Tasks for the week were allocated. Prior to the meeting, a list of tasks was put on Teams by Seth. These were discussed, and any tasks left off the list were added. The tasks for this week fell into three areas: new coding tasks, documentation, and quality assurance tasks. Members who wanted to do a particular area indicated this (equally, those who did not want to do a particular task also indicated this). Several members said they were happy to do anything. The list of tasks was left on teams for members to put their names down for individual items, with anything else being allocated the following morning.

Action items:

- Allocate spare tasks and distribute task list.

II. Meeting Two
Date: 03/12/2021
Time: 1100-1115 (online)
Attendance: Yunni, Zening, Peiwen, Devendra, Rob, Karan, Seth, Yijin, Adil
Agenda:

- Review work done so far.
- Ensure there are no problems with the tasks so far.

Outcome:

- Rob presented the latest framework for the levels. Each level will be based on a 15x15 grid, where all outer tiles are all wall tiles. Any tiles not used will also be filled in with wall tiles. This allows the creation of differently shaped levels within the 15x15 grid, while keeping the total size of the level consistent.
- Karan and Devendra shared their progress on the bug that they discussed in the previous customer meeting. As it was still an issue, Seth was added to the task to provide extra help.

- Seth reminded the team about writing tests and the importance of doing work throughout the week, rather than Tuesday evening, so that any problems could be shared with the team in time and dealt with.

5.2.2 Work Plan

Cycle	Sprint	Phase	Planned Milestone	Achieved
1	03/11-10/11	Pre-game		
2	10/11-17/11	Pre-game	Plan for first prototype	Yes
3	17/11 - 24/11	Game	Delivery one - demo	Yes, although with bugs.
4	24/11 - 01/12	Game	Fully realised first level	Yes, although unable to show customer.
5	01/12 - 08/12	Game	Delivery two - multiple levels	
6	08/12 - 15/12	Game		
7	15/12 - 20/12	Post-game	Final hand in	

5.2.3 Backlog

Backlog Item	Name	Cross references	Responsibility	Comments	Status at start of cycle
B-14	Light System	US-13, UC-5	Peiwen, Zening		Completed
B-16	Spider	US-2, UC-4	Devendra		Completed with bug
B-17	Big torch	US-13, UC-6	Karan		Completed
B-20	Character movement	US-1, UC-2, EX-3	Seth		Completed
B-21	Sprint 3 review write-up		Seth	Task changed to Seth	Rolled over
B-22	Sprint 4 documentation		Seth, Yijin, Adil		Completed
B-23	Plan next set of features		Seth		Completed
B-24	Ranged enemy	US-16, UC-8	Rob		Completed
B-25	Obstacle tiles	US-17, UC-9	Yunni		Minor fixes needed
B-26	Environment tiles		Rob, Yijin, Yunni		Completed
B-27	Fix level size bug	EX-4	Rob		Started

B-28	Start menu	US-14, UC-9	Zening	Version with the start menu currently runs from a different file, so does not have the most recent level functionality. Integrating this has been added to the sprint backlog for next week.	Completed
B-29	Design new levels		Rob		New
B-30	User manual first draft		Adil		New
B-31	Lives	US-3, UC-4	Yunni		New
B-32	Playtesting		Adil		New
B-33	Testing and bug fixing		Karan, Devendra, Seth	Seth added for second half of Sprint.	New
B-34	Random level feature	US-15	Zening		New
B-35	Torch	US-13, UC-6	Peiwen	Change so that it ends level when lit	New
B-36	Sprint 5 Documentation		Seth, Yijin		New

5.2.4 Exception Handling

EX-6
Status: Open → closed.

EX-7
Problem: Changes were made to the wrong file on GitHub and were then lost.
Details: During the sprint a duplicate of the main TinderBoxKnight.py file was pushed to the GitHub repository. After around day, the duplicate file was removed. However, several commits were made to the duplicate file which were then not in the original file.
Handling: Using the version control system, a copy of the duplicate file was retrieved. This was then read through manually with all changes being copied over to the original file.
Status: Closed.

5.2.5 Prioritisation table

Must have	**Should have**	**Could have**	**Won't have**
Movement	Clue tiles	Usable items	Sound-based mechanics
Level grid	Ranged enemy	Randomised tiles	
Stationary enemy	Main menu	Sound effects	
Placeable light	Lives	Score mechanics	

Enemy scan	Environmental tiles (including obstacles)		
End of level torch			

5.3 Product Documents

5.3.1 Customer Meeting

No customer meeting this week

5.3.2 User Stories

No new user stories.

5.3.3 Use cases

New use cases relate to the new random level feature. UC-9-d has been updated to allow the player to select this option.

I. Requirements Use Cases

UC-10-r
Title: Random level
Creation Date: 02/12/2021
Cross-references: US-15, US-11
Level: System framework
Context:

- Allow the user to generate a level where enemies are placed randomly.

Frequency of occurrence: Whenever the user selects 'random level'.

II. Design Use Cases

UC-9-d (update)
Title: Start menu
Author: Seth
Update date: 03/11/2021
Purpose: Allow the player to start the game.
Overview: When the player opens the game a menu appears which allows them to start the game.
Cross references: US-14, US-14
Actors: Player
Pre-condition:

1. The player has opened the game.
2. The start menu is currently on the screen.

Post-condition:

1. The screen changes to display the first level.

Normal flow of events:

1. The start menu displays on the screen. There are three options: 'start game', 'random level', and 'end game'.
2. The player presses the enter key on the keyboard to select 'start game'.
3. The start menu disappears and is replaced by the first level.

Alternative flow of events:

2. The player presses the down arrow and enter to select 'random level'. See UC-10.

2. The player presses the down arrow, to move the cursor to ‘quit game’, and presses enter. The game quits.

UC-10-d
Title: Random level
Author: Seth
Update date: 03/11/2021
Purpose: Allow the player to play a random level.
Overview: When the player selects ‘random level’ at the main menu, they can select a difficulty.
Cross references: US-14, US-14
Actors: Player
Pre-condition:

1. The player has opened the game.
2. The start menu is currently on the screen.

Post-condition:

1. The screen changes to display a ‘random’ level with the given difficulty.

Normal flow of events:

1. The start menu displays on the screen. There are three options: ‘start game’, ‘random level’, and ‘end game’.
2. The player presses the down arrow and enter key on the keyboard to select ‘random level’.
3. A second menu comes up allowing the player to choose between easy, medium, and hard difficultly.
4. The player presses enter to select ‘easy’.
5. One level template out of the five is randomly selected. 5 enemies are placed in pseudorandom positions in the level.
6. The level is displayed to the user.

Alternative flow of events:

4. The user selects medium. 6 enemies are placed in the level.
4. The user selects hard. 9 enemies are placed in the level.

5.3.4 CRC Cards

In preparation for integrating the start menu and new level architecture next week, the CRC cards have been updated. CRC-9 (Level) has taken over the majority of the functions previously handled by CRC-6 (Main). Main has become a simple class that opens the game window and transitions between different game states. Finally, there is a main menu class to provide the main menu. All references to CRC-6 Main in previous CRC cards are now be CRC-9 Level, but for reasons of space they are not all listed here. CRC-9 Level has a long list of collaborative classes. This would normally not be ideal, but is unavoidable here as each class implements a function in the level so must be called by the Level class. The CRC-11 has been added to implement the random level feature.

CRC-6: Main (updated)	
Responsibility	**Collaboration**
• Set-up pygame window • Set-up list of game states • Facilitate transition between menu and level	• Level • Main menu

CRC-10: Level	
Responsibility	**Collaboration**
• Read in the level from the text file	• Main • Knight

• Create 2-dimensional list of level elements. • Handle user input during levels. • Draw the level	• Light • Tile • Spider • Torch • Scan • Ranged enemy • Pressure plate.

CRC-11: Main menu	
Responsibility	**Collaboration**
• Display start menu to player. • Allow user to choose a level.	• Main game • Level • Random difficulty

CRC-12: Random Level	
Responsibility	**Collaboration**
• Populate a level array with enemies placed at pseudo-random positions. • Pass the generated array to the level class so it can be played.	• Level • Random difficulty

5.3.5 Tests

I. New Tests

T-19
Name: Lives decrease.
Description: Test that the player’s lives decrease when they are attacked by an enemy.
Cross-references: US-4, US-5, UC-3
Pre-conditions:

1. The player is in a level.
2. The player is not in the middle of another activity.
3. The tile to the north of the knight is spider.
4. The player has three lives.

Test steps:

1. Press the up arrow.

Expected result:

1. The lives tile change from having three hearts to two,

T-20
Name: Running out of lives.
Description: Test that when the player runs out of lives the level ends.
Cross-references: US-4, US-5, UC-3
Pre-conditions:

1. The player is in a level.
2. The player is not in the middle of another activity.
3. The tile to the north of the knight is spider.
4. The player has one life.

Test steps:

1. Press the up arrow.

Expected result:

1. The player sees text saying that they have run out of lives.
2. The level resets and the player returns to the start menu.

T-21a
Name: Random level (easy)
Description: Test that the player can produce two different levels when selecting the easy mode.
Cross-references: US-14, US-15, UC-7
Pre-conditions:

1. The player is at the start menu.

Test steps:

1. Press the down arrow to move the cursor to ‘random level’ and press enter.
2. Press enter to select ‘easy’.
3. Press q tot return to the start menu.
4. Repeat steps 1 and 2.

Expected result:

1. A level is generated when the player selects ‘easy’.
2. The second level is visibly different to the first.

T-21b
Name: Random level (medium)
Description: Test that the player can produce two different levels when selecting the medium mode.
Cross-references: US-14, US-15, UC-7
Pre-conditions:

1. The player is at the start menu.

Test steps:

1. Press the down arrow to move the cursor to ‘random level’ and press enter.
2. Press the down arrow and enter to select ‘medium.
3. Press q to return to the start menu.
4. Repeat steps 1 and 2.

Expected result:

3. A level is generated when the player selects ‘medium’.
4. The second level is visibly different to the first.

T-21c
Name: Random level (hard)
Description: Test that the player can produce two different levels when selecting the hard mode.
Cross-references: US-14, US-15, UC-7
Pre-conditions:

1. The player is at the start menu.

Test steps:

1. Press the down arrow to move the cursor to ‘random level’ and press enter.
2. Press the down arrow and enter to select ‘hard’.
3. Press q to return to the start menu.
4. Repeat steps 1 and 2.

Expected result:

1. A level is generated when the player selects ‘hard’.
2. The second level is visibly different to the first.

T-22a
Name: Random level
Description: Test that there are more enemies in the medium mode than the easy more.
Cross-references: US-14, US-15, UC-7
Pre-conditions:

1. The player is at the start menu.

Test steps:

1. Press the down arrow to move the cursor to ‘random level’ and press enter.
2. Select ‘easy’ mode.
3. Press the ‘m’ key.

4. Press q to return to the start menu.
5. Repeat steps 1, 2, 3, and 4 but select ‘medium’ mode.
6. Select exit to quit the game.

Expected result:

1. On the console will be two numbers preceded by ‘number of enemies’. The second number should be larger than the first.

Note: The feature to get the number of enemies is just for testing and will be removed in the final system.

T-22b

Name: Random level

Description: Test that there are more enemies in the hard mode than the medium mode.

Cross-references: US-14, US-15, UC-7

Pre-conditions:

1. The player is at the start menu.

Test steps:

1. Press the down arrow to move the cursor to ‘random level’ and press enter.
2. Select ‘medium mode.
3. Press the ‘m’ key.
4. Press q to return to the start menu.
5. Repeat steps 1, 2, 3, and 4 but select ‘hard mode.
6. Select exit to quit the game.

Expected result:

2. On the console will be two numbers preceded by ‘number of enemies’. The second number should be larger than the first.

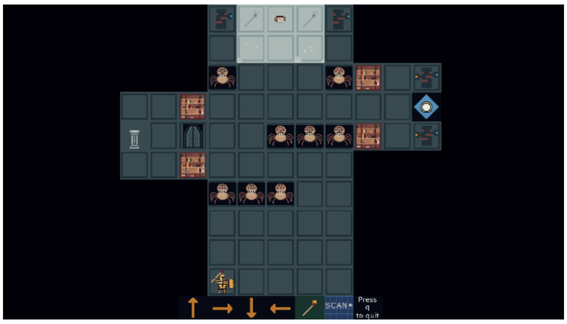
Note: The feature to get the number of enemies is just for testing and will be removed in the final system.

Test	**Feature**	**Date run**	**Pass?**	**Comments**
T-1	Movement	03/12	Yes	
T-2	Movement	03/12	Yes	
T-3	Torch	03/12	No	No change in the level. Added to bug list.
T-4	Spider	03/12	Yes	
T-5	Spider	03/12	Yes	
T-6	Light	03/21	Yes	
T-7	Light	03/21	Yes	
T-8	Start menu	N/A	N/A	Not tested as tests are focusing on demo level with start menu
T-9	Scan	03/21	Yes	
T-10	Movement	04/12	Yes	
T-11	Movement	04/12	Yes	
T-12	Ranged enemy	04/12	No	Values were hardcoded so if the location of the enemy moves then the attack does not work. Added to bug list.
T-13	Ranged enemy	04/12	Yes	
T-14	Gate	04/12	Yes	
T-15	Ranged enemy	06/12	No	Ranged enemy attacks regardless of lit value. Added to bug list.

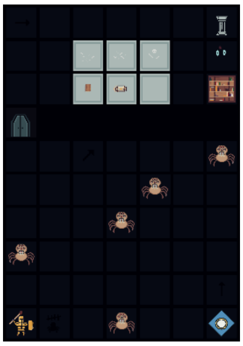
T-16	Start menu	N/A	N/A	Not tested.
T-17	Start menu	N/A	N/A	Not tested.
T-18	Start menu	N/A	N/A	Not tested.

5.3.6 Design Elements

I. New levels

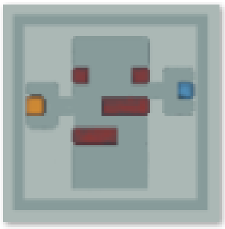


Level three



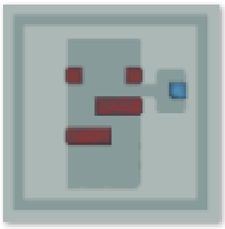
Level four

II. New tiles

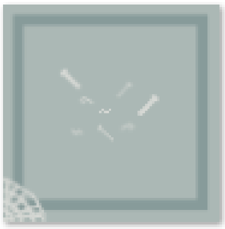


VisibleLeftRoomClue

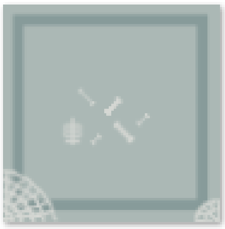
VisibleMatchClue



VisibleRightRoomClue



VisibleWebArmBones



VisibleWebRibBones

5.3.7 Bug List

Feature	Bug description	Date bug found	Class (if known)	Solution
Spider interaction	Lighting a spider while in the right-hand column creates an out-of-bounds error.	29/11/2021	TinderBoxKnight	Added check that value was not out of bounds before accessing it. Also levels will now have wall tiles surrounding them so the player will never be able to access the outer tiles.
Torch	If the space bar is pressed by the torch, nothing happens.	03/12	TinderBoxKnight and Torch	Several issues: for loops were not iterating over full length of list to check for lit torch; wrong tile code was being checked for; and wrong location was being used for knight. Now fixed.

Torch	No cutscene plays	03/12	TinderBoxKnight and Torch	Discovered after previous bug had been fixed. Left for now to focus on other issues.
Ranged enemy	If the ranged enemy is moved to a different location, the attack does not work.	04/12	RangedEnemy	Changed ranged enemy location from being hard-coded to dynamically generated by searching the array when the level is loaded.
Spider interaction	If two spiders are lit, only one is hidden.	06/12	Spider	Changes made in loop in spider.py
Pressure Plate	Gate was not opening when o pressed on plate.	06/12	TinderBoxKnight	Added s to pressure plate as function was not being called due to typing error.
Ranged enemy	If there is a non-basic tile between the enemy and the player, the game will crash when trying to convert it to a poisoned variant.	06/12	RangedEnemy	Altered so that only the ranged enemy tile and player tile are changed.
Ranged enemy	If the enemy is lit, the attack still functions as normal.	07/12	RangedEnemy	Added Boolean value to ranged enemy class so attack code only evaluates if the enemy is not lit.
Scan	Scan number is appearing in wrong place	07/12	Scan	Fixed by adjusting values used to position text. Error is due to tile size changing to get a good fit on the screen - until the tile size is fixed permanently this may happen again.
Scan	Scan numbers remain after level has been reset.	08/12	TinderBoxKnight	Added a line to code that reset the level to clear the list of scanned tiles.

5.4 Review

Tasks were more focused on continuing to finalise the code and ensure the game would be ready by the end of the sprint, with thorough playtesting allowing negative feedback and bugs to be minimised. If tasks were completed early, then testing of the game occurred by each member and issues were noted down. Non-coding tasks were prioritised, as a user manual draft with a scenario-driven explanation of how the game works needed to be produced. The importance of writing tests and documenting it was also prioritised and emphasised.

Progress was relatively smooth this week as the game was in a playable state with a functioning start menu and completable levels. New clue and bone tiles were introduced which aimed to help the player.

Branches were merged on GitHub to a final master and allowed playtesting from all members to give constructive feedback and highlight any bugs so that they can be fixed earlier on as the deadline was nearing. The customer was happy with the demo, and no major additional features were needed to be added. By the meeting on Friday, the second level was designed, and the torch function was successfully implemented.

Fewer problems occurred this week as the first level was fully playable and the major features worked as intended. This allowed an increased amount of confidence in reaching the final product and it was a case of ironing out bugs and finalising documentation. The game breaking bug EX-6 was resolved, and focus was made on adding minor features such as lives and a soundtrack. An issue with a duplicate of the main TinderBoxKnight.py occurred where several important commits were made before being removed. A copy was luckily able to be retrieved and the changes were manually merged with the original.

Sprint 6 (09/12 - 10/12)

6.1 Summary

6.1.1 Overview

The aim of this week is to finish the game, defined as having the game at a point where we are happy to deliver it to the customer. The primary tasks this week will consist of integrating the features from the file we have been using to demonstrate the level functionality into the new system architecture starting the game from main.py (B-38). As part of this, all tests need to rerun on the new code and the levels must be played through and any bugs documented and fixed (if possible). Two new features will be added. The main feature is the option for the player to generate a random level. Work on this feature started in Sprint 5 but it was unable to be completed by the end. Although this is a large feature to be integrated in the final week, it should be separate to the main game and not introduce any significant errors. The second feature will be adding sound effects. Additionally, new levels will be created and tested.

6.2 Process Documents

6.2.1 Meeting Minutes

I. Meeting One
Date: 08/12/2021
Time: 1400-1500 (including customer meeting)
Attendance: Yunni, Zening, Peiwen, Devendra, Rob, Karan, Seth, Adil, (Yijin: ill)
Agenda:

- Ensure the game can be demonstrated to the customer.
- Review previous week's work.
- Plan tasks for coming week.

Outcome:

- Peiwen's laptop was used for the code demonstration, so all functions were tested to make sure they worked correctly. A small issue was found where the scan number was slightly out of place: this was fixed by adjusting the numbers used to position the scan tile.
- While making sure all the functions worked, the features integrated in the previous week were reviewed and demonstrated to the rest of the team.
- Tasks for the coming week were discussed and planned. The only new coding task will be to add the sound effects; the other coding tasks will be rolled over from the previous sprint. The developer manual will also be written this week.

Action items:

- Post task list to Teams

II. Customer Meeting
Date: 08/12/2021
Time: 1440-1445 (including customer meeting)
Attendance: Yunni, Zening, Peiwen, Devendra, Rob, Karan, Seth, Adil
Agenda:

- Present the latest version of the game to the customer.
- Update the customer on the state of the documentation.
- Clarify where we are intending the game to be by the time of the final delivery.

Outcome:

- The presentation of the game was done by Rob. The features of the level were presented using the demo version (running the game from TinderBoxKnight.py). We presented the demo level, which contained all the elements of the final game in a condensed space. The main menu of the game was then demonstrated by running the game from main.py.

- We discussed our ideas for the final set of levels. The customer was walked through how the levels are created and stored, especially the change from a 7x10 grid to a 15x15 grid.
- The structure of the documentation was shown to the customer by Seth. We overviewed the user stories, use cases, and how we had written the tests. We also presented the first version of the user manual for approval.
- A discussion was had of what the final game would look like. Details of questions are contained in section 6.3.1.

III. Meeting Two
Date: 10/12/2021
Time: 1800-1810
Attendance: Yunni, Zening, Peiwen, Devendra, Rob, Karan, Seth, Adil, Yijin
Agenda:

- Clarify what we want to have achieved by the end of the Sprint.
- Review work done and any problems encountered.

Outcome:

- The delivery dates for the next (calendar) week were discussed. We agreed to try to have the game ready for delivery by Wednesday, and the documentation ready to submit by Friday. There was a reminder for everyone to complete their tasks on time to achieve this.
- The only problem encountered was EX-8, detailed in 6.2.4. This was talked through so that all team members were aware of how to avoid it occurring again.
- The new levels that had been created (level 1 and level 2) were presented.
- Team members were reminded to spend some time playing through the game and checking for bugs.

6.2.2 Work Plan

Cycle	Sprint	Phase	Planned Milestone	Achieved
1	03/11-10/11	Pre-game		N/A
2	10/11-17/11	Pre-game	Plan for first prototype	Yes
3	17/11 - 24/11	Game	Delivery one - demo	Yes, although with bugs.
4	24/11 - 01/12	Game	Fully realised first level	Yes, although unable to show customer.
5	01/12 - 08/12	Game	Delivery two - multiple levels	No - all features are present but start menu and level switching has not been fully integrated.
6	08/12 - 15/12	Game	Full game	
7	15/12 - 20/12	Post-game	Final hand in	

6.2.3 Backlog

Backlog Item	Name	Cross references	Responsibility	Comments	Status at start of cycle
B-21	Sprint 3 review write-up		Seth	Task changed to Seth	Completed
B-25	Obstacle tiles	US-17, UC-7	Yunni		Completed
B-29	Design new levels		Rob, Peiwen	Peiwen added	Ongoing

B-30	User manual first draft		Adil		Completed
B-31	Lives	US-4, US-5, UC-3	Yunni		Completed
B-32	Playtest available levels		Adil		Completed
B-33	Testing and bug fixing		Karan, Devendra, Seth	Seth added for second half of Sprint.	Completed
B-34	Random level feature	US-15	Zening	Started but not yet finished.	Ongoing
B-35	Torch	US-13, UC-5	Peiwen	Change so that it ends level when lit	Completed
B-36	Sprint 5 Documentation		Seth, Yijin		Completed
B-37	Run tests on game with main menu		Seth		New
B-38	Integrate features from demo into main game		Zening, Seth	For the last few weeks, features have added to the game via TinderBoxKnight.py. Simultaneously, a version of the game with a main menu that has been developed as part of B-28, which starts from a different file. The level functionality is handled by a different file (level.py) when starting the game from the main menu, so the latest features from TinderBoxKnight.py need to be added to level.py.	New
B-39	Add sound effects to game		Karan	Background music and sound effects for actions	New
B-40	Review previous documentation		Devendra	Ensure all sections have been completed	New
B-41	Write sprint 5 review		Adil		New
B-42	Sprint 6 documentation		Adil, Yijin, Seth		New
B-43	Review and update user manual		Yijin	Ensure that it matches current game	New

B-44	Developer manual first draft		Seth		New
B-45	Refactor and tidy code		Seth	In preparation for B-44	New
B-46	Playtest levels and document bugs		All members (if initial task completed).		New
B-47	Tidy GitHub structure.		Rob		New
B-48	Compose music for menu		Rob		New

6.2.4 Exception Handling

EX-8
Problem: When pushing changes, the GitHub repository was accidentally reverted to a version from several weeks ago and all commits since then were gone.
Details: Initial coding for the game in Sprint 3 was done on the master branch of the GitHub repository. At some point during Sprint 4, a new branch was created for the Demo level. This then became the standard working branch for the previous few Sprints. During Meeting One this week, it was decided to merge the Demo branch back into the master branch to simplify the structure. However, on pushing a change the following morning, an error occurred, and the branch reverted to the original master branch before Demo was merged in, thereby losing the work of Sprint 4 and 5.
Handling: A complete copy of the most up to date files was still stored locally, so the files were uploaded to a new GitHub repository as a backup. These files were then manually uploaded back to the original repository, replacing the old files. The problem causing the error was dealt with my removing and readding the remote branch of the local git repository. The online repository is now restored to the most recent version of the code, but the record of commits has been lost.
Status: Closed.

6.2.5 Prioritisation table

Must have	**Should have**	**Could have**	**Won't have**
Movement	Clue tiles	Usable items	Sound-based mechanics
Level grid	Ranged enemy	Randomised tiles	
Stationary enemy	Main menu	Sound effects	
Placeable light	Lives	Score mechanics	
Enemy scan	Environmental tiles (including obstacles)		
End of level torch			

6.3 Product Documents

6.3.1 Customer Meeting

I. Presentation

The presentation to the customer this week consistent of a live demonstration of the game. We first presented the features of the level by running the game from TinderBoxKnight.py, as this version of the game had been tested in Sprint 4 and contained the most recent features. We then showed the game running from main.py, as this contained the start menu. We did not show the features running from this as they had not been tested.

II. Questions

1. The initial brief for the game mentioned a ‘bot’. We feel that the game has evolved to the point where including this feature would be incongruous. We want to make a puzzle game that is challenging and enjoyable for human players and including a bot would put a limit on how interesting and varied the levels could be. Before the final week, we want to confirm that this is acceptable to you.
   a. Yes, I agree. The initial brief was just a rough guideline, and it’s clear that that element no longer fits with the direction of the game. I’m more than happy for this to be dropped in order to have more interesting levels.
2. The features that we have just presented to you are the total set of features that we plan to include in levels the final game. Is this satisfactory to you?
   a. Yes, absolutely. I can see that it has everything we discussed in the earlier meetings, so I am happy for this to be the set of features.

III. Analysis

As we did not have a customer meeting last week, there was a significant amount of material to cover with the customer this week. The customer liked the demo and was positive about how the options available to the player worked together with the level to provide a compelling experience. The customer was particularly interested in the changes we had made to the level, as this gives us the option to easily vary the size and layout of levels. The customer’s answers to the questions we asked also puts us in a good position for the final sprint, as we will not have to add any major new functionality like a bot to satisfy their requirements. This allows us to focus on integrating the game’s features and making sure everything is properly tested.

6.3.2 User Stories

No new user stories.

6.3.3 Use cases

No new use cases as no new features are being added this week - random level will be integrated, but use cases are contained in 5.3.3. Sound effects do not have a use case as they do not change any system or player behaviour.

6.3.4 CRC Cards

The menu to allow the player to select a difficulty has been placed in a separate class, so a CRC card has been made.

CRC-12: Random difficulty	
Responsibility	**Collaboration**
• Display a menu to the player allowing them to select a difficulty option.	• Random level • Main menu

6.3.5 Tests

I. Tests run

All tests are now run on the version of the game starting from main.py (prior to B-38 beginning).

Test	Feature	Date run	Pass?	Comments
T-1	Movement	09/12	Yes	
T-2	Movement	09/12	Yes	
T-3	Torch	09/12	Yes	
T-4	Spider	09/12	Yes	
T-5	Spider	09/12	Yes	
T-6	Light	09/12	Yes	
T-7	Light	09/12	Yes	
T-8	Scan	09/21	No	Number appears in the tile to the north west of the player.
T-9	Quit	09/12	Yes	
T-10	Movement	09/12	Yes	
T-11	Movement	09/12	Yes	
T-12	Ranged enemy	09/12	No	Tile flashes purple then returns to normal before the player is reset.
T-13	Ranged enemy	09/12	Yes	
T-14	Gate	09/12	No	Game crashes with 'IndexError: list index out of range'
T-15	Ranged enemy	09/12	Yes	
T-16	Start menu	09/12	Yes	
T-17	Start menu	09/12	Yes	
T-18	Start menu	09/12	Yes	
T-19	Lives	09/12	Yes	
T-20	Lives	09/12	No	The player resets but the message that they have run out of lives remains on the screen.
T-21	Random Level	09/12	No	Feature not yet in game.
T-22	Random Level	09/12	No	Feature not yet in game.
T-23	Ranged Enemy	11/12	No	Ranged enemy turns into a spider when it is walked into.
T-24	Sound	11/12	No	No code for sound in level.py yet.

II. New Tests

T-23
Description: Test that the ranged enemy becomes visible when the player walks into it.
Cross-references: US-16, UC-6, CRC-8
Pre-conditions:

1. The player is in a level.
2. The knight currently has a dark background.
3. There is a hidden ranged enemy on the tile south of the player.

Test steps:

1. Press the down arrow

Expected result:

1. The ranged enemy tile becomes visible
2. The text is displaying saying 'be careful of the shadows'.

T-24a
Description: Test for sound effect when the player uses the light feature.
Cross-references: US-18
Pre-conditions:

1. The player is in a level.

Test steps:

1. Press the f key.

Expected result:

1. A sound effect of a fire lighting plays.

T-24b
Description: Test for sound effect when the player opens a gate.
Cross-references: US-18
Pre-conditions:

1. The player is in a level.
2. The player is standing on a pressure plate

Test steps:

1. Press the o key.

Expected result:

1. A sound effect of a gate opening plays.

T-24c
Description: Test for sound effect when the walks into a monster.
Cross-references: US-18
Pre-conditions:

1. The player is in a level.
2. The tile to the north of the player is a spider tile

Test steps:

2. Press the up arrow.

Expected result:

2. A sound effect of a spider plays.

T-24d
Description: Test for sound effect when the player is spotted by the ranged enemy.
Cross-references: US-18
Pre-conditions:

1. The player is in a level.
2. The tile to the north of the player is lit
3. There is a ranged enemy on the same row as the lit tile.

Test steps:

3. Press the up arrow.

Expected result:

3. A sound effect for the player being spotted plays.

T-24e
Description: Test for sound effect when the player has one life.
Cross-references: US-18
Pre-conditions:

1. The player is in a level.
2. The tile to the north of the player is a spider.
3. The player has two lives.

Test steps:

1. Press the up arrow.

Expected result:

1. A sound effect for the player having one life plays.

T-24f
Description: Test for sound effect when the player loses all three lives.
Cross-references: US-18
Pre-conditions:

1. The player is in a level.
2. The tile to the north of the player is a spider.
3. The player has one life.

Test steps:

1. Press the up arrow.

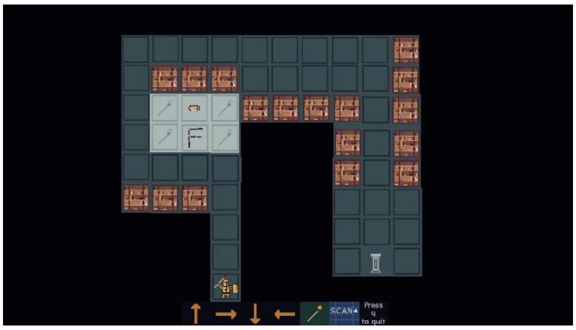
Expected result:

1. A sound effect for the player losing all lives plays.

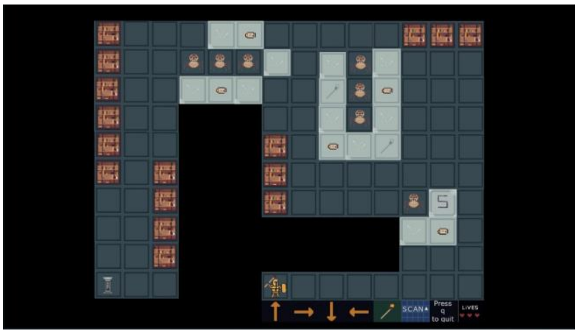
6.3.6 Design Elements

The design elements this week are the new levels that have been created for the game. Each level aims at introducing a new feature: level one introduces movement, level two introduces enemies and scanning, and level five will be the last level, which will contain all the features from previous levels.

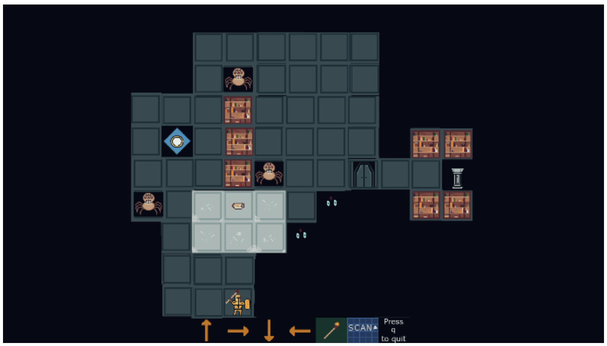
I. New levels



Level one



Level two



Level five

6.3.7 Other Products

I. Bug list

Feature	Bug description	Date bug found	Class (if known)	Cause of problem	Solution
Lives	When the level resets after the player runs out of lives, the message informing the player that they have no lives remains on the screen.	09/12	Level	The Boolean value of whether to display the end of game message or not was not reset to false when the level reset.	Setting value to false when the level is read in fixed the issue. Position of text was also adjusted to centre it.
Scan	The number for the scan appears in the tile to the north and west of the player's location.	09/12	Scan	Error was caused by having a constant addition to the player's location. The tile size varied depending on the player's screen size, but the constant added on to position the text did not, meaning the position of the text was not consistent between screens.	The value used for centring the text has been changed from constant to a multiple of the size of the tile.
Ranged Enemy	When the player is spotted by the ranged enemy the tile only changes to purple very briefly instead of staying until the player is reset.	09/12	Level	The level was reset to have the non-poisoned squares immediately after changing them to poison, with no delay.	The call to draw the level was moved later in the flow to give more time with the purple squares.
Pressure Plate	When the player tries to open a gate the game crashes after trying to access past the bounds of a list.	09/12	Pressure Plate (error caused by line 63)	The line initialising the pressure plates had been commented out as it required an attribute to be passed in that had been removed from level.py compared to TinderBoxKnight.py. This meant the list of stored pressure plate locations was empty.	The passed in parameter was used for a separate draw method in pressureplate.py. This method has been removed and integrated with the main draw method in level.py.
Ranged Enemy	When the player lights up a spider and also becomes visible to the ranged enemy, the ranged enemy tile does not reset.	11/12			

Ranged Enemy	When the player walks into a dark ranged enemy the tile becomes a spider. Bug not covered by tests, added and run.	11/12	Knight		
Torch	Pressing when not near the torch changes the top left square to a lit torch	12/12	Level	A method to check the eligibility of the player to light the torch was called but then not checked.	Put an if statement around check_eligibility.
Pressure plate	Pressure plates only work in the first level they appear, and not in future levels (not dependent on what the initial level is)	14/12	Pressureplate	The lists storing the location of the gates and pressure plates were not being reset between levels.	Changed the lists from global to local (within class).
Ranged Enemy	Ranged enemy does not attack when player steps in light unless they are next to it.	14/12	Ranged enemy	The ranged enemy checked for a wall by checking if any tiles in between contain a ‘w’. Since then extra tiles have been which also contain a w.	The check was changed to check specifically for walls (hw and vw).
Sound	Main menu sound does not stop after the user exits the main menu. In addition, the music does not loop and level music takes over.	14/12	Mainmenu and Level	The sound did not stop as there was no code unloading the menu music before the level was loaded. The second issue was caused by the background music for the level being placed in the __init__ method, so starting automatically when the level object was created.	Added a line to unload sound when a menu option is selected. Moved background music from __init__ to the method called when the level starts.
Random level	The random level generator can sometimes create impossible levels.	14/12	Random Level	If a level map had a single tile width passage, there was a chance a spider could be placed there, thereby blocking the passage.	Added a check so that enemies will not be placed in single passages.

6.4 Review

At the start of the week were slightly behind were we wanted to be, as we did not originally intend to still be implementing new features in the final full week. However, this was not a problem as the week had deliberately been kept open to allow for tasks from earlier weeks overrunning, which inevitably happened. Allocation of tasks this week followed the procedure used in the previous sprints. After the start meeting in which tasks were brainstormed, a list was put on Teams of all tasks for people to sign up to. All members assigned themselves to tasks this week so there was no need for any central allocation. Ongoing tasks (B-25 and B-34) kept the same member as the previous week, as both Rob and Zening were happy to continue (Peiwen was also added to B-25 to speed up the process of level creation). In addition to the individual tasks this week, a collective task was also added. Once members had completed their initial task, their second task was B-46 (playtesting and bug finding). The size of initial tasks varied, so members spent different amounts of time on this. It was not assigned to one person due to the need to test the game on multiple devices.

Overall progress this week was good, and the game is in a state where we are able to deliver it to the customer. Integrating the features from the demo level to the main game took less time than expected. All the tests were first run on the game starting from main.py to establish where the new system differed from the demo level (see section 6.3.5). Due to the work carried out at the end of the previous sprint, most tests were already passing, leaving only a few features to be ported across. This was completed by the time of the second meeting. Additionally, the sound had also been done by the second meeting. In the second half of the week the focus switched to trying to find and fix bugs, the results of which can be seen in section 6.3.7. The ability to generate a random level was integrated quite late in the sprint, but as predicted this did not impact the normal levels in any way. Only one bug was found when testing the random level feature (see 6.3.7), and this was quickly fixed. On the documentation side, the developer manual was written in the second half of the sprint.

The main problem this week was encountered on the GitHub repository. As a result of the structure changing and branches being renamed, a push at the start of the week overwrote the branch history and reset it to a version from several weeks ago (see EX-8 for details). Luckily this was able to be fixed by re-uploading the files from local copies, but it did mean that the commit history was lost. This is unfortunate as it makes it hard to compare the game against earlier copies, especially if we find that a test is now failing that was passing before. Two factors ameliorate this issue: this is the last week of coding so the new features are unlikely to affect the tests, and the code uploaded passed all of the tests before we demonstrated it at the customer meeting. As this was a problem could affect all members, a message was posted on teams at the time detailing what had happened and how to avoid it (i.e. making sure that the changes in the remote git are updated locally).

Sprint 7 (15/12 - 17/12)

7.1 Overview

The aim for this sprint is to submit the game and documentation. We have set a deadline for this of Friday 17/12, so this sprint will end there. As all development on the game was finished by the end of Sprint 6, the structure of this Sprint's documentation will be slightly different. There is no new product documentation. Instead, section 7.3.1 contains a list of the acceptance criteria of all user stories within the documentation so far and whether they have been met. 7.3.2 contains a final check of all tests. Finally, due to the short length of this sprint the review in 7.4 covers the entire game development process rather than solely this week.

7.2 Process Documents

7.2.1 Meeting Minutes

Week 7 Meeting One

Date: 15/12/2021
Time: 2000-2015
Attendance: Yunni, Zening, Peiwen, Devendra, Rob, Karan, Seth, Adil, Yijin
Agenda:

- Review work done in Sprint 6.
- Update the team on the results of the TA meeting earlier.
- Discuss what needs to be done for the delivery of the game and documentation.

Outcome:

- The new features implemented in sprint 6 (random levels and sound) were discussed so the team understood how they work.
- The game was played through to ensure everything worked. A minor bug was found with lives - this will be added to the bug list.
- Seth had met the TAs earlier in the day so relayed their feedback. This was mostly relating to minor changes within the documentation and how the game should be delivered.
- The plan for the rest of the week was reviewed. The tests will all be run a final time to ensure they all pass, then the game will be frozen as an executable. The tests will then be run again on the executable version of the game.

7.2.2 Work Plan

Cycle	Sprint	Phase	Planned Milestone	Achieved
1	03/11-10/11	Pre-game		N/A
2	10/11-17/11	Pre-game	Plan for first prototype	Yes
3	17/11 - 24/11	Game	Delivery one - demo	Yes, although with bugs.
4	24/11 - 01/12	Game	Fully realised first level	Yes, although unable to show customer.
5	01/12 - 08/12	Game	Delivery two - multiple levels	No - all features are present but start menu and level switching has not been fully integrated.
6	08/12 - 15/12	Game	Full game	Yes
7	15/12 - 20/12	Post-game	Final hand in	

7.2.3 Backlog

Backlog Item	Name	Cross references	Responsibility	Comments	Status at start of cycle
B-29	Design new levels		Rob, Peiwen		Completed
B-34	Random level feature	US-15	Zening		Completed
B-37	Run tests on game with main menu		Seth		Completed
B-38	Integrate features from demo into main game		Zening, Seth		Completed
B-39	Add sound to game		Rob, Karan		Completed
B-40	Review previous documentation		Devendra		Completed
B-41	Write sprint 5 review		Adil		Completed
B-42	Sprint 6 documentation		Adil, Yijin, Seth		Completed
B-43	Review and update user manual		Yijin		Completed
B-44	Developer manual first draft		Seth		Completed
B-45	Refactor and tidy code		Seth	In preparation for B-44	Completed
B-46	Playtest levels and document bugs		All members (if initial task completed).		Completed
B-47	Tidy GitHub structure.		Rob		Completed
B-48	Compose music for menu		Rob		Completed
B-49	Create mac executable file		Adil		New
B-50	Create windows executable file		Yunni		New
B-51	Run all tests on code		Seth		New
B-52	Run all tests on Windows app		Seth		New
B-53	Run all tests on mac app		Adil		

B-54	Sprint 7 documentation		Seth		New
B-55	Final documentation check		Seth		New
B-56	Submit documentation		Seth		New

7.2.4 Exception Handling

EX-9
Problem: When running the windows executable file, an error message is generated.
Details: A error message box appears whenever the executable file is run, saying that the file libmpg123-0.dll failed to load. Further research indicates this file is used to play the mp3s of the sound effects and music.
Handling: A copy of the file was put into the folder containing the application and the assets. This solved the error.
Status: Closed.

7.2.5 Prioritisation table

Must have	**Should have**	**Could have**	**Won't have**
Movement	Clue tiles	Usable items	Sound-based mechanics
Level grid	Ranged enemy	Randomised tiles	
Stationary enemy	Main menu	Sound effects	
Placeable light	Lives	Score	
Enemy scan	Environmental tiles (including obstacles)		
End of level torch			

7.3 Product Documents

7.3.1 User Stories

Name	Title	Acceptance Criteria	Acceptance Status
US-1	Movement	User input results in the character sprite moving one tile.	Yes
US-2	Enemies	Enemies are placed in levels.	Yes
US-3	Lives	The player starts with a finite number of lives.	Yes
US-4	Losing lives	The number of lives decreases when an enemy attacks.	Yes
US-5	Light	On user input, areas of the grid switch from being dark to light.	Yes

US-6	Bot	Bot appears in the game and can perform an action.	No
US-7	Timer	The time taken for a player to complete the level is displayed once the level is complete.	No
US-8	Captions	Captions appear for relevant game activities.	Yes
US-9	Items	Items are placed in dungeon.	Replaced
US-10	Using items	On user input, an item in the user's inventory is used and has an effect in the game.	No
US-11	Placing enemies	Enemies are placed in a non-predetermined place in the grid.	Yes (although not in normal levels)
US-12	Enemy counter	Player can see whether nearby squares contain enemies.	Yes
US-13	Finishing level	When the torch is lit the level is complete.	Yes
US-14	Starting level	On specified user input the level starts.	Yes
US-15	Randomising level	There is a way for the user to generate a random level.	Yes
US-16	Clue	A clue tile exists in the level and can be seen by the player.	Yes
US-17	Ranged enemy	Ranged enemy exists in the level and can attack the player if they walk in front of it.	Yes
US-18	Score	When the level ends the player's score is displayed.	No
US-19	Sound	A sound effect is played when the player finds an enemy.	Yes
US-20	Wall	Tiles exist in the level that the player cannot walk on.	Yes
US-21	Gate	The player can activate a switch and can then walk on the tile where the gate was.	Yes
US-22	Level shape	At least two different level shapes exist.	Yes

7.3.2 Tests

I. New tests

T-25a
Description: Test that the executable file opens the game on Windows machines.
Cross-references:
Pre-conditions:

1. The user is on a computer running Windows.
2. The user has downloaded and extracted the game files to a folder.
3. The user has the folder open.

Test steps:

1. Double click on the file called TinderBoxKnight_Start

Expected result:

1. The game will open.

T-25b
Description: Test that the executable file opens the game on macOS.
Cross-references:
Pre-conditions:

1. The user is on a machine with macOS.
2. The user has downloaded and extracted the game files to a folder.
3. The user has the folder open.

Test steps:

1. Double click on the file called TinderBoxKnight_Start

Expected result:

1. The game will open.

II. Tests running on .py files

Test	Feature	Date run	Pass?	Comments
T-1	Movement	16/12	Yes	
T-2	Movement	16/12	Yes	
T-3	Torch	16/12	Yes	
T-4	Spider	16/12	Yes	
T-5	Spider	16/12	Yes	
T-6	Light	16/12	Yes	
T-7	Light	16/12	Yes	
T-8	Scan	16/21	Yes	
T-9	Quit	16/12	Yes	
T-10	Movement	16/12	Yes	
T-11	Movement	16/12	Yes	
T-12	Ranged enemy	16/12	Yes	
T-13	Ranged enemy	16/12	Yes	
T-14	Gate	16/12	Yes	
T-15	Ranged enemy	16/12	Yes	
T-16	Start menu	16/12	Yes	
T-17	Start menu	16/12	Yes	
T-18	Start menu	16/12	Yes	
T-19	Lives	16/12	Yes	
T-20	Lives	16/12	Yes	
T-21	Random Level	16/12	Yes	
T-22	Random Level	16/12	Yes	
T-23	Ranged Enemy	16/12	Yes	
T-24	Sound	16/12	Yes	
T-25	Executable file	16/12	N/A	

III. Tests running on Windows executable

Test	Feature	Date run	Pass?	Comments
T-1	Movement	16/12	Yes	
T-2	Movement	16/12	Yes	
T-3	Torch	16/12	Yes	
T-4	Spider	16/12	Yes	
T-5	Spider	16/12	Yes	
T-6	Light	16/12	Yes	
T-7	Light	16/12	Yes	
T-8	Scan	16/21	Yes	
T-9	Quit	16/12	Yes	
T-10	Movement	16/12	Yes	
T-11	Movement	16/12	Yes	
T-12	Ranged enemy	16/12	Yes	

T-13	Ranged enemy	16/12	Yes	
T-14	Gate	16/12	Yes	
T-15	Ranged enemy	16/12	Yes	
T-16	Start menu	16/12	Yes	
T-17	Start menu	16/12	Yes	
T-18	Start menu	16/12	Yes	
T-19	Lives	16/12	Yes	
T-20	Lives	16/12	Yes	
T-21	Random Level	16/12	Yes	
T-22	Random Level	16/12	Yes	
T-23	Ranged Enemy	16/12	Yes	
T-24	Sound	16/12	Yes	
T-25	Executable	16/12	Yes	T-25a run

IV. Tests running on Mac executable

Test	Feature	Date run	Pass?	Comments
T-1	Movement	17/12	Yes	
T-2	Movement	17/12	Yes	
T-3	Torch	17/12	Yes	
T-4	Spider	17/12	Yes	
T-5	Spider	17/12	Yes	
T-6	Light	17/12	Yes	
T-7	Light	17/12	Yes	
T-8	Scan	17/21	Yes	
T-9	Quit	17/12	Yes	
T-10	Movement	17/12	Yes	
T-11	Movement	17/12	Yes	
T-12	Ranged enemy	17/12	Yes	
T-13	Ranged enemy	17/12	Yes	
T-14	Gate	17/12	Yes	
T-15	Ranged enemy	17/12	Yes	
T-16	Start menu	17/12	Yes	
T-17	Start menu	17/12	Yes	
T-18	Start menu	17/12	Yes	
T-19	Lives	17/12	Yes	
T-20	Lives	17/12	Yes	
T-21	Random Level	17/12	Yes	
T-22	Random Level	17/12	Yes	
T-23	Ranged Enemy	17/12	Yes	
T-24	Sound	17/12	Yes	
T-25	Executable	17/12	Yes	T-25b run

7.4 Review

As a team, we are happy that the game we have built and delivered meets the user's criteria. When the user opens the game, they see a start menu with the option to start the game, generate a random level, and quit the game. If they select start game, they can see and play through the levels we have made. These feature the ability to move the player, to light up the squares around the player and to scan the number of nearby enemies. The levels feature two types of enemies, an interactive gate and switch to open it, and a way to end each level. The game therefore satisfies criteria for the minimum viable product.

Moreover, we think the game is not only functionality but enjoyable. The game has been built around the central dynamic of light and dark. In order to complete a level, the player is forced to light up squares in order to find clues and level elements needed to progress like gate switches and the torch to end the level. At the same time, lighting up squares carries the risk that one of the squares will contain a hidden enemy. This tension is at the heart of the game. Replayability of the game is ensured through the ability to generate completely new levels. Five pre-made levels are included the game. Level creation is an easy process from a technical point of view so given time more would have been created.

Although we completed most of the features in our prioritisation grid, there a few that we did not implement, along with several user stories that were not developed further. The exercise of initial prioritisation meant that the absence of these features does not affect the core gameplay. We originally had the idea of giving the player a score based on the number of moves that they used to complete the level, to provide an incentive to replay the level. This could have been combined with US-7, to provide a time taken metric as well. Unfortunately, it was clear that we would not have time for this feature, so development on it was not started. Our estimate for this task was that it would be quite minor, so it would have been a good candidate for inclusion next. The other element unimplemented, usable items, would have been more significant and would have given the players more options in the level. This would have been our aim for a third delivery cycle. However, the game is completely functional without it, so the absence is not significant. In addition, no bot was implemented in the game. Although we had this as a user story (US-6), over the following sprints it became apparent that it would not fit the game we had built. This was confirmed with the customer in Sprint 6, and both the team and customer agreed that implementing it would harm rather than help the game. We were also planning on including a cutscene for when the player finished the game, and a first version of this was made. However, we could not get the video to play in front of the pygame window. As this was not a central feature we opted to not spend time trying to get it to work, and instead focus on delivering more game features. However, it would be a nice addition if we were able to get it to work in future.

In addition, if we were to do the project again we would have kept a closer watch over the structure of the code. For the demonstration to the customer, the game ran off a file which started the game and also contained all level functionality. When the start menu framework was added, this level functionality was put in a separate file, which did not contain the most recent updates to the level functionality. We left it quite late in the process before reconciling these two versions and integrating the most recent features over to the new file. This created some duplication of work in that all tests had to be re-run and the failing tests had to be addressed, and the system had to be rechecked for bugs. Although no major bugs were introduced, there was the possibility that this could happen. If we were to redo the project, greater attention would be paid at the start of the development process to build a system architecture that would last the length of the project and not have to be changed. As it was, the system architecture was developed on a week-by-week basis which limited designing for the long term.

The process used to develop the game remained reasonably consistent the entire way. Each sprint started with a large meeting which combined the review meeting for the previous sprint, the customer meeting, and then a planning meeting to discuss the customer's feedback and plan the upcoming sprint (for clarity, the minutes for the team meeting and for the customer meeting have been separated in the documentation). A second meeting was then had on Friday, to review any problems and update the team on any developments. Having the meeting on Friday meant the first section of the sprint was relatively short. However, as a team we wanted to ensure that we kept weekends free from meetings, so that we were following the XP principle of developing sustainably. This meant the alternative was Monday, and it was felt that it would be better to have the meeting early within the week rather than later to correct any problems as soon as possible. In the first meeting, we would brainstorm tasks based on the previous week and what we had left to implement. Team members then self-assigned themselves to tasks over the afternoon, so that by Thursday morning all tasks were taken. Although when using a self-assignment system there is the danger that members become parochial if they take the same feature each week, this was not encountered to a significant degree and coding tasks were generally rotated frequently.

An element that did change within the process was the system of assigning tasks. Initially, we used Jira to create an online backlog and then a scrum board during the week. This continued for the first couple of weeks of development, but the online board always felt quite external the team and never became fully integrated into our workflow. In comparison, the group on MS Teams was central to our workflow and there were frequent posts by members to update on progress. As a result, the decision was taken to abandon the use of Jira and switch fully to Teams. The Teams message board therefore became our information radiator for the project. While if we did the project again Teams would be used alone from the start, trying out Jira was a worthwhile experiment this time and little time was lost due to it.

Other tools used remained in use the entire time. Git was used for version control, with the repository hosted on GitHub. Apart from EX-8, no major problems were encountered with this. Not all team members had used GitHub before, so some instruction was required, and a few members contributed their work indirectly by sending the relevant section of code to another member who was able to push it to the GitHub repository. Overall, the process was smooth - as Git is such a common tool it was also good to be able to practice using it as part of a collaborative project. For writing the documentation, the decision was taken early on to used Word. OverLeaf was considered, but it was felt that it would be easier to insert tables and images in Word than OverLeaf. Throughout the process we had a shared master documentation file, which everyone was able to view and edit and contained all previous sprints. Alongside this was a working file (also open to everyone), which contained the current Sprint’s documentation for easy access. At the end of each sprint the file was added onto the end of the master file and a fresh file was created for the upcoming sprint. The links to these were posted at the start of each sprint, so all members had easy access to edit or view the files.

Appendix

8.1 Final User Stories

Name	Title	User Story	Acceptance Criteria
US-1	Movement	As a player, controlling a character I want to move the character, so that I can explore the dungeon.	User input results in the character sprite moving one tile.
US-2	Enemies	As a player, I want to have enemies in the dungeon, so that there are things to avoid.	Enemies are placed in levels.
US-3	Lives	As a player, I want to have limited lives, so that I must be careful about what I do.	The player starts with a finite number of lives.
US-4	Losing lives	As a player, I want to lose a life when an enemy attacks, so that I must avoid them.	The number of lives decreases when an enemy attacks.
US-5	Light	As a player, exploring the dungeon, I want to be able to light up areas of the dungeon, so that I can see the area.	On user input, areas of the grid switch from being dark to light.
US-6	Bot	As a player I want to have a bot to help me so that I have more options for how to deal with enemies.	Bot appears in the game and can perform an action.
US-7	Timer	As a player, I want to see how fast I can complete the dungeon so that I can replay the game and try to beat the score.	The time taken for a player to complete the level is displayed once the level is complete.
US-8	Captions	As a player, I want to see captions for certain activities so that I have more information about the game and story.	Captions appear for relevant game activities.
US-9	Items	As a player, I want the dungeon to have items so that there is an incentive to explore.	Items are placed in dungeon.
US-10	Using items	As a player, I want to be able to use items so that there is variety in game play.	On user input, an item in the user's inventory is used and has an effect in the game.
US-11	Placing enemies	As a player, I want to enemies to be in different places each time I play the level so that I can replay the level.	Enemies are placed in a non-predetermined place in the grid.
US-12	Enemy counter	As a player, exploring the dungeon, I want to see whether enemies are nearby, so that I can plan my next move.	Player can see whether nearby squares contain enemies.
US-13	Finishing level	As a player, I want to be able to light up a torch within the level so that I have a goal in each level.	When the torch is lit the level is complete.
US-14	Starting level	As a player, I want to choose a level from the menu, so that I can start the game.	On specified user input the level starts.
US-15	Randomising level	As a player, I want to be able to randomise the level, so that I can replay it.	There is a way for the user to generate a random level.
US-16	Clue	As a player, I want to have tiles with clues to the end on so that I know where I am meant to be heading.	A clue tile exists in the level and can be seen by the player.
US-17	Ranged enemy	As a player, I want to have to avoid the view of a ranged enemy so that I am forced to be careful about where I step.	Ranged enemy exists in the level and can attack the

			player if they walk in front of it.
US-18	Score	As a player, I want to have a score of how well I have done on a level so that I can go back and try to beat it.	When the level ends the player's score is displayed.
US-19	Sound	As a player, I want to have a sound effect when player interacts with the environment around.	A sound effect is played when the player finds an enemy.
US-20	Wall	As a player, I want there to be walls in the level that block the character so that the route to the torch is not obvious.	Tiles exist in the level that the player cannot walk on.
US-21	Gate	As a player, I want to be able to open gates in the wall by finding switches in the dungeon, so that I can progress to the next section.	The player can activate a switch and can then walk on the tile where the gate was.
US-22	Level shape	As a player, I want the levels to be different shapes, so that each level feels different.	At least two different level shapes exist.

8.2 Final Use Cases

I. Requirements Use Cases

UC-1-r
Title: Displaying a level
Creation date: 11/11/2021
Level: Level framework
Cross references:
Context:

- When the player starts the game, the level is displayed on the screen.

Frequency: Whenever the player starts a level.

UC-2-r
Title: Moving the character
Creation date: 11/11/2021
Cross references: US-1
Level: User action
Context:

- The player can move the knight one tile to the north, east, south and west, as long as the target tile is empty and in the bounds of the level.
- If the tile contains an enemy it will light up and the knight will reset.

Frequency: Whenever the player presses an arrow key in the level.

UC-3-r
Title: Lighting up an area
Creation date: 11/11/2021
Cross references: US-5
Level: User action
Context:

- Players can press 'f' to light up a three-by-three grid centred on their location.
- Tiles within this grid become visible.

Frequency: Whenever the player presses the 'f' key within a level.

UC-4-r
Title: Losing a life

Creation date: 11/11/2021
Cross-references: US-3, US-4, US-2
Level: Level interaction
Context:

- The player will lose a life when they find an enemy.
- If the number of lives equals zero, they will be taken back to the start menu.

Frequency: Whenever the player finds an enemy.

UC-5-r
Title: Detecting enemies
Creation date: 11/11/2021
Cross-references: US-12
Level: User action
Context:

- When the player presses ‘s’, the number of enemies in a 5x5 grid centred on their location will be displayed.

Frequency: Whenever the user presses s.

UC-6-r
Title: Finishing the level
Creation date: 11/11/2021
Cross-references: US-13
Level: User action
Context:

- The player can use the light function to reveal a torch within the level.
- Activating the torch will end the level.

Frequency: Once per level.

UC-7-r
Title: Ranged enemy interaction
Creation date: 18/11/2021
Cross-references: US-17
Level: Level interaction
Context:

- If the player stands on a lit tile (through movement or lighting a torch) on the same row as a shadow enemy, they will lose a life.

Frequency: Whenever the player enters the monster’s field of view.

UC-8-r
Title: Walls and gates
Creation date: 18/11/2021
Cross-references: US-20, US-21
Level: User interaction
Context:

- Allow the player to press ‘o’ when standing on a specific tile to replace a gate in a wall with a tile the player can walk on.

Frequency: Whenever the player presses ‘o’.

UC-9-r
Title: Start menu
Creation date: 26/11/2021
Cross-references: US-14
Level: System framework
Context:

- Display a start menu to the player.
- Allow the player to pick an option from the menu and start the game.

Frequency: Whenever the game starts or the user quits a level.

UC-10-r
Title: Random level
Creation Date: 02/12/2021
Cross-references: US-15, US-11
Level: System framework
Context:

- Allow the user to generate a level where enemies are placed randomly.

Frequency of occurrence: Whenever the user selects ‘random level’.

II. Design Use Cases

UC-1-d
Title: Displaying the level
Author: Seth
Creation date: 12/11/2021
Purpose: Display the level when the player starts the game.
Overview: When the player starts a level, it should be loaded on the screen.
Cross references:
Actors: Player
Dependencies: Start menu (when implemented)
Pre-condition:

1. A level text file has been made.
2. The player is at the start menu of the game.

Post-condition:

1. The first level will be displayed on the screen.

Normal flow of events:

1. The player selects the ‘start’ option.
2. The text file containing the level is read in.
3. The images are displayed on the screen.

Exception flow of events:

2. The file contains a code that the game cannot associate with a tile. The game closes, and a message is displayed indicating the problem code.

UC-2-d (updated)
Title: Moving the character
Author: Seth
Creation date: 24/11/2021
Purpose: Allow the player to move the character.
Overview: When the player presses a specified movement key, the character will move to a new tile on the screen.
Cross references: US-1
Actors: Player
Dependencies: UC-1
Pre-condition:

1. The player must be currently in a level.
2. The player-character must be in an idle state, not in the middle of another activity.
3. The player has at least one life.

Post-condition:

1. The player-character will be in a grid tile above the starting tile.

Normal flow of events:

1. The player presses the up arrow on the keyboard.
2. Check that the tile above the player is a valid movement spot.
3. Check that the new tile does not contain a monster.
4. Check that the light on the tile is the same as the current tile (i.e. if the knight is standing on a dark square, the new square is also dark).
5. The player-character sprite is moved from the centre of the initial tile to the centre of the tile above.

Alternative flow of events:

1. The player presses the right arrow. The tile checked is the one to the east, and the character will move to the to the east.
1. The player presses the down arrow. The tile checked is the one to the south, and the character will move to the to the south.
1. The player presses the left arrow. The tile checked is the one to the east, and the character will move to the to the west.
2. The intended tile is blocked by another object. The player-character stays on the tile where they currently are.
3. The tile contains a monster. See UC-4-r, alternative flow.
4. The new tile is different to the current tile (i.e. lit if the player is currently on a dark square). The image for the knight tile is updated to match the lighting value of the new time. Continue to 5.

UC-3-d
Title: Lighting up an area
Author: Seth
Creation Date: 12/11/2021
Purpose: Allow the player to light up an area.
Overview: When the user presses the specified key, a torch is placed on the square and the 3x3 grid centred on the player is lit up.
Cross references: US-5
Actors: Player
Dependencies: UC-1
Pre-condition:

1. The player is currently in a level.
2. The player-character is not currently in the middle of another action.
3. The player has at least one life.

Post-condition:

1. The 3x3 grid of tiles centred on the player is lit up and objects in the tiles are revealed.

Normal flow of events:

1. The player presses the space bar on the keyboard.
2. A check is run that there is not already a torch on the square that the player-character is standing on.
3. A check is run on the 3x3 grid centred on the player and the torch to see which squares are dark and which are light.
4. Any dark squares are changed to light, with a corresponding visual change.
5. Any objects which those tiles contain are made visible.

Alternative flow of events:

3. A torch has already been used on the square and surrounding squares are lit. There is no change to the square or surrounding grid.

UC-4-d (update)
Title: Lives
Author: Seth
Update Date: 30/11/2021
Purpose: Allow for enemies to attack the player.
Overview: When the player finds an enemy, they lose a life. A tile is changed on the HUD to show the lives left.

Cross references: US-4, UC-1, UC-2
Actors: Player, enemy
Pre-condition:

1. The player is currently in a level.
2. Enemies have been placed in the level.
3. The player has at least one life.

Post-condition:

1. The number of lives is decreased by one.
2. The player respawns at the beginning of the level.
3. The enemy that attacked the player disappears.
4. The lives tile changes to a tile showing one fewer lives.

Normal flow of events:

1. The player moves into a dark square (UC-1).
2. The player lights up the dark square (UC-2). In one of the tiles in the lit up 3x3 grid, there is an enemy sprite.
3. Text displays telling the player they have found an enemy.
4. The number of lives the player-character has decreases by one. A check is run that the number of lives is above zero.
5. The player-character is moved to the start of the level. The enemy that attacked the player disappears from the level, but otherwise the level is the same.
6. A tile at the bottom of the screen changes from one with the previous number of lives on to one with one fewer lives on.

Alternative flow of events:

2. The square the player moves into contains an enemy. Jump to 3 and continue from there.
4. The number of lives equals zero. A message tells the player that they have failed the level and gives them the option to retry it. The player-character is moved to the start of the level. The rest of the level is reset - all enemies are replaced and all changes the player made (placing torches, etc) are removed.

UC-5-d
Title: Detecting enemies
Author: Seth
Creation Date: 12/11/2021
Purpose: Allow the player to detect whether there are enemies nearby.
Overview: On user input, a count of nearby enemies in a specified location will be displayed to the player.
Cross references: US-12
Actors: Player, enemies
Pre-condition:

1. The player is currently in a level.
2. The player-character has at least one life.
3. The player is not currently in the middle of another action.

Post-condition:

1. A count of the number of enemies in the 3x3 grid in front of the player is displayed on the screen.

Normal flow of events:

1. The player presses the control button on the keyboard.
2. A check is run that the player is standing in a lit square.
3. A check is run on the number of enemies in the 3x3 grid to the north, south, east, and west of the player is standing on (this is done irrespective of whether the squares are dark or light).
4. The number of enemies existing in each grid is displayed on the screen to the player, on the grid square adjacent to the player in each direction.
5. The number remains for one second then fades away.

Alternative flow of events:

2. The player is standing in a dark square. Nothing happens.

2. The full 3x3 grid in a specified direction does not exist due to the level boundaries. The check is only run on the squares that exist. If no squares exist because the player is standing next to the boundary, then a count of zero is displayed for that direction. Normal flow resumes.

UC-6-d
Title: Finishing the level
Author: Seth
Creation date: 12/11/2021
Purpose: To allow the player to finish a level.
Overview: The square on which the torch is placed is lit. A message tells the player that the level is complete, and the level ends.
Cross references: US-13
Dependencies: UC-1, UC-2, UC-3
Actors: Player
Pre-condition:

1. The player is currently in a level.
2. The player-character has at least one life.

Post-condition:

1. The level ends by fading to black, and the next level starts.

Normal flow of events:

1. The player moves into a new square (UC-1).
2. The player lights a torch on the new square, lighting up surrounding tiles (UC-2).
3. One of the lit squares contains the torch. The player presses space while standing next to the torch.
4. An animation plays of the torch lighting up, and all squares in the map switch from being dark to light.
5. A message tells the player that they have completed the level.
6. Transition to the start of the next level plays.

Alternative flow of events:

3. The lit squares do not contain the torch. Use case ends.

UC-7-d
Title: Ranged enemy interaction
Author: Seth
Creation date: 20/11/2021
Purpose: Player interaction with ranged enemy.
Overview: When the player walks in front of the ranged enemy while standing on a lit square, the ranged enemy attacks.
Cross references: US-16, UC-1, UC-3, CRC-8
Actors: Player, Ranged enemy
Pre-condition:

1. The player must be currently in a level.
2. The player-character must not in the middle of another activity.
3. The player has at least one life.
4. The level has a ranged enemy within it.

Post-condition:

4. The number of lives is decreased by one.
5. The player respawns at the beginning of the level.
6. The enemy that attacked the player disappears.

Normal flow of events:

6. The player moves into a lit square (UC-1).

7. A check is run whether the player's square within view of the ranged enemy (ranged enemies can face north, south, east, and west). The check returns true if the enemy is facing the player and the player is on the same row or column and standing on a lit square.
8. Text displays telling the player that they have been spotted by the enemy.
9. The number of lives the player-character has decreases by one. A check is run that the number of lives is above zero.
10. The player-character sprite and camera are moved to the start of the level. The enemy that attacked the player disappears from the level, but otherwise the level is the same.

Alternative flow of events:

2. The square the player moves into is dark. No check is run and use case ends.
5. The number of lives equals zero. A message tells the player that they have failed the level and gives them the option to retry it. The player-character is moved to the start of the level. The rest of the level is reset - all enemies are replaced and all changes the player made (placing torches, etc) are removed (identical to UC-3).

UC-8-d
Title: Walls and gates
Author: Seth
Creation Date: 20/11/21
Purpose: To allow the player to open a gate in a wall in the level.
Overview: When the player walks on a specific tile in the level, a gate in a wall in the level opens, allowing the player to get to the next section.
Cross-reference: US-20, US-21
Actors: Player
Dependencies: UC-2, UC-3
Pre-condition:

1. The player is in a level.
2. The level has a wall in it.
3. The wall has a gate tile in it.

Post-condition:

1. The gate in the wall is replaced by a tile the player can walk on.

Normal flow of events:

1. The player moves on a tile that is visually distinct from a normal tile - the tile will only be visually distinct when lit.
2. The player presses the 'o' button to activate the tile.
3. Text displays above the player saying that something in the level has changed.
4. The gate tile within the wall is replaced by a different tile that the player can walk on.

Alternative flow of events:

2. The player is not on a pressure plate tile. Nothing happens and use case ends.

UC-9-d
Title: Start menu
Author: Seth
Update date: 03/11/2021
Purpose: Allow the player to start the game.
Overview: When the player opens the game a menu appears which allows them to start the game.
Cross references: US-14, US-14
Actors: Player
Pre-condition:

1. The player has opened the game.
2. The start menu is currently on the screen.

Post-condition:

1. The screen changes to display the first level.

Normal flow of events:

1. The start menu displays on the screen. There are three options: ‘start game’, ‘random level’, and ‘end game’.
2. The player presses the enter key on the keyboard to select ‘start game’.
3. The start menu disappears and is replaced by the first level.

Alternative flow of events:

2. The player presses the down arrow and enter to select ‘random level’. See UC-10.
2. The player presses the down arrow, to move the cursor to ‘quit game’, and presses enter. The game quits.

UC-10-d
Title: Random level
Author: Seth
Update date: 03/11/2021
Purpose: Allow the player to play a random level.
Overview: When the player selects ‘random level’ at the main menu, they can select a diffic.
Cross references: US-14, US-14
Actors: Player
Pre-condition:

1. The player has opened the game.
2. The start menu is currently on the screen.

Post-condition:

1. The screen changes to display a ‘random’ level with the given difficulty.

Normal flow of events:

1. The start menu displays on the screen. There are three options: ‘start game’, ‘random level’, and ‘end game’.
2. The player presses the down arrow and enter key on the keyboard to select ‘random level’.
3. A second menu comes up allowing the player to choose between easy, medium, and hard difficultly.
4. The player presses enter to select ‘easy’.
5. One level template out of the five is randomly selected. 5 enemies are placed in pseudorandom positions in the level.
6. The level is displayed to the user.

Alternative flow of events:

4. The user selects medium. 6 enemies are placed in the level.
4. The user selects hard. 9 enemies are placed in the level.

8.3 Final CRC Cards

CRC-1: Knight	
Responsibility	**Collaboration**
• Keep track of player location • Keep track of player lives • Allow for player to see the number of nearby enemies. • Interaction with enemies	• Dungeon tile • Main

CRC-2: Light	
Responsibility	**Collaboration**
• Switch tiles from dark to light in 3x3 grid around player.	• Knight • Main

CRC-3: Tile	
Responsibility	**Collaboration**

• Map value in level array to image.	• Main

CRC-4: Spider	
Responsibility	**Collaboration**
• Attack character when they walk into the enemy, causing them to lose a life. • Attack character when the tile the enemy is on is lit up, causing them to lose a life.	• Dungeon tile • Main

CRC-5: End-of-level Torch	
Responsibility	**Collaboration**
• Ends the level when lit.	• Tile • Knight • Main

CRC-6: Main	
Responsibility	**Collaboration**
• Read in the level from the text file • Create objects • Handle user input • Draw the level	• Knight • Tile • Spider

CRC-7: Scan	
Responsibility	**Collaboration**
• Display the number of monster tiles in the 5x5 grid of tiles centred on the player.	• Knight • Main

CRC-8: Ranged Enemy	
Responsibility	**Collaboration**
• Attack character when they walk into the enemy, causing them to lose a life. • Attack the character when they walk on the same row or column as the ghost on a light square.	• Main game

CRC-9: Pressure Plate	
Responsibility	**Collaboration**
• Store location of pressure plate and gates in the level. • Test whether player is standing on the plate when ‘o’ is pressed. • Open the corresponding date	• Main game

CRC-10: Level	
Responsibility	**Collaboration**
• Read in the level from the text file • Create 2-dimensional list of level elements. • Handle user input during levels. • Draw the level	• Main • Knight • Light • Tile • Spider

	• Torch • Scan • Ranged enemy • Pressure plate.

CRC-11: Main menu	
Responsibility	**Collaboration**
• Display start menu to player. • Allow user to choose a level.	• Main game • Level • Random difficulty

CRC-12: Random Level	
Responsibility	**Collaboration**
• Populate a level array with enemies placed at pseudo-random positions. • Pass the generated array to the level class so it can be played.	• Level • Random difficulty

CRC-13: Random difficulty	
Responsibility	**Collaboration**
• Display a menu to the player allowing them to select a difficulty option.	• Random level • Main menu

8.4 Final Tests

T-1a
Name: Normal movement test.
Description: Test that when the player presses the up arrow, the knight moves one square north.
Cross-references: US-1, UC-2, CRC-1, B-20
Pre-conditions:

5. The player is in a level.
6. The player is not in the middle of another activity.
7. The knight is not standing on a tile that borders the north barrier of the level.
8. The tile to the north of the knight is a normal tile that can be walked on,

Test steps:

2. Press the up arrow.

Expected result:

3. The knight should appear in the square to the north of where they were before.

T-1b
Name: Normal movement test.
Description: Test that when the player presses the right arrow, the knight moves one square east.
Cross-references: US-1, UC-2, CRC-1, B-20
Pre-conditions:

5. The player is in a level.
6. The player is not in the middle of another activity.
7. The knight is not standing on a tile that borders the east barrier of the level.
8. The tile to the east of the knight is a normal tile that can be walked on,

Test steps:

2. Press the right arrow.

Expected result:

4. The knight should appear in the square to the east of where they were before.

T-1c
Name: Normal movement test.
Description: Test that when the player presses the down arrow, the knight moves one square south.
Cross-references: US-1, UC-2, CRC-1, B-20
Pre-conditions:

5. The player is in a level.
6. The player is not in the middle of another activity.
7. The knight is not standing on a tile that borders the south barrier of the level.
8. The tile to the south of the knight is a normal tile that can be walked on,

Test steps:

5. Press the down arrow.

Expected result:

5. The knight should appear in the square to the south of where they were before.

T-1d
Name: Normal movement test.
Description: Test that when the player presses the left arrow, the knight moves one square west.
Cross-references: US-1, UC-2, CRC-1, B-20
Pre-conditions:

8. The player is in a level.
9. The player is not in the middle of another activity.
10. The knight is not standing on a tile that borders the west barrier of the level.
11. The tile to the west of the knight is a normal tile that can be walked on,

Test steps:

6. Press the down arrow.

Expected result:

6. The knight should appear in the square to the south of where they were before.

T-2
Name: Moving into a border
Description: Test that the player cannot move outside of the level.
Cross references: US-1, UC-2, CRC-1, B-20
Pre-conditions:

4. The player is in a level.
5. The player is not in the middle of another activity.
6. The knight is standing on a tile that borders the north barrier of the level.

Test steps:

2. Press the up arrow.

Expected result:

2. The knight does not move.

T-3
Name: Lighting the final torch
Description: Test that when the torch is lit, the level ends.
Cross references: US-13, UC-6, B-17
Pre-conditions:

6. The player is in a level.
7. The player is not in the middle of another activity.
8. The player is in the square next to the torch.
9. The torch is in an unlit state.
10. The player is not in the final level.

Test steps:

2. Press the space bar on keyboard.

Expected result:

3. The torch tile picture changes into a lit torch picture.
4. The next level is displayed on the screen.

T-4

Name: Player resets on spider interaction

Description: Test that text appears and the player resets when the player walks into an enemy.

Cross references: US-2, US-8, UC-4, B-16

Pre-conditions:

5. The player is in a level.
6. The player is not in the middle of another activity.
7. The player is in a square immediately south of a spider.
8. The player has at least two lives (once implemented)

Test steps:

2. Press the up-arrow key.

Expected result:

3. Text is flashed on the screen saying, “You hit a monster!”.
4. The knight tile returns to the starting square.

T-5

Name: Walking into a spider.

Description: Test that the discovered spider by knight/player becomes hidden again.

Cross-references: US-2, UC-4, B-16

Pre-conditions:

4. The player is in a level.
5. The player is in a square immediately south of a spider.
6. The player has at least two lives (once implemented).

Test steps:

2. Press the up-arrow key to move player into the spider tile.

Expected result:

2. The discovered spider tile becomes dark/hidden again when the player respawns.

T-6

Name: Lighting up surrounding squares

Description: Test that on user input the player can light up squares surrounding their position.

Cross-references: US-5, UC-3, CRC-2, B-14

Pre-conditions:

3. The player is in a level.
4. The player is not on a border tile.

Test steps:

2. Press the f key.

Expected result:

2. The 3x3 around the player switches from dark tile variants to light tiles.

T-7

Name: Checking light on board

Description: Test that only those tiles on the level board change when the light is used.

Cross-references: US-5, UC-3, CRC-2, B-14

Pre-conditions:

3. The player is in a level.
4. The player is bordering the edge of the level.

Test steps:

3. Press the f key.

Expected result:

3. Only the tiles within a 3x3 grid and that are within the level change to light variants.

T-8
Name: Scanning for enemies
Description: Test that on user input the player see the number of surrounding enemies.
Cross-references: US-12, UC-4, CRC-7, B-15
Pre-conditions:

3. The player is in a level.
4. There are two spider tiles in the 5x5 grid centred on the character.

Test steps:

4. Press the s key.

Expected result:

4. The number ‘2’ will appear on the board.

T-9 (updated)
Name: Quitting a level.
Description: Test that on user input the player can quit a level and return to the start menu.
Cross-references: US-17, UC-7, B-28

Pre-conditions:

2. The player is in a level in the game.

Test steps:

2. Press the q key.

Expected result:

2. The level will be replaced by the start menu.

T-10
Name: Updated movement test.
Description: Test that when the knight moves onto a square then off it, the square returns to what it originally was.
Cross-references: US-1, UC-2, CRC-2, B-20
Author: Seth
Pre-conditions:

12. The player is in a level.
13. The player is not in the middle of another activity.
14. The tile to the north of the knight is a lit clue tile.

Test steps:

7. Press the up arrow.
8. Press the down arrow.

Expected result:

7. The knight should appear in the square to the north of where they were before after the first button press, then should move back to the original square on the second.
8. After the second button press the lit clue tile should be in its original position.

T-11
Name: Light and dark movement test
Description: Test that when the knight moves from a dark square to a light square and back, the knight’s tile changes accordingly.
Cross-references: US-1, UC-2, CRC-2, B-20
Pre-conditions:

5. The player is in a level.
6. The player is not in the middle of another activity.
7. The knight currently has a dark background.
8. The tile to the north of the knight is a blank lit tile.

Test steps:

3. Press the up arrow.
4. Press the down arrow.

Expected result:

3. After the first press the tile to the north of the original position of the knight should contain a knight with a lit background.
4. After the second button press the knight should move back to original tile, and have a dark background.

T-12
Description: Test that the ranged enemy attacks player when in the light.
Cross-references: US-16, UC-6, CRC-8, B-23
Pre-conditions:

5. The player is in a level.
6. The knight currently has a dark background.
7. There is a hidden ranged enemy on the same column as the player.
8. There are no walls between the enemy and the player.

Test steps:

2. Press the f key

Expected result:

4. The 3x3 grid centred on the player lights up.
5. The knight tile and the ranged enemy tile are highlighted in purple.
6. The knight resets to the starting position and the ranged enemy returns to normal.

T-13
Description: Test that the ranged enemy does not attack the player when there is a wall in the way.
Cross-references: US-16, UC-6, CRC-8, B-23
Pre-conditions:

5. The player is in a level.
6. The knight currently has a dark background.
7. There is a hidden ranged enemy on the same column as the player.
8. There is a wall tile between the enemy and the player.

Test steps:

2. Press the f key

Expected result:

3. The 3x3 grid centred on the player lights up.
4. There is no change to the ranged enemy.

T-14
Description: Test that the player can open the gate in the wall.
Cross-references: US-19, US-20, UC-7, B-25
Pre-conditions:

4. The player is in a level.
5. There is a lit pressure plate tile immediately west of the player.
6. A gate tile is visible in the level.

Test steps:

3. Press the right key to move the player on to the pressure plate.
4. Press the o key.

Expected result:

2. The gate should be replaced with an open gate tile.

T-15
Description: Test that the ranged enemy does not attack the player when it has already been lit up.
Cross-references: US-16, UC-6, CRC-8, B-23
Pre-conditions:

5. The player is in a level.
6. The knight currently has a dark background.
7. There is a visible ranged enemy on the same column as the player.

8. There is no wall tile between the enemy and the player.

Test steps:

2. Press the f key

Expected result:

3. The 3x3 grid centred on the player lights up.
4. There is no change to the ranged enemy and the player does not lose a life.

T-16

Description: Test that the game opens with a start menu.
Cross references: US-17, UC-7, B-28
Pre-conditions:

2. The user has an open command terminal at the folder containing the program.

Test steps:

2. Run ‘python main.py’

Expected result:

2. A menu displaying the option to start the game and exit appears on the screen.

T-17

Description: Test that the start menu allows the player to start the levels.
Cross references: US-17, UC-7, B-28
Pre-conditions:

3. The user has the start menu open on the screen.
4. The cursor is showing as selecting ‘start game’

Test steps:

2. Press the enter key.

Expected result:

2. The first level of the game appears.

T-18

Description: Test that the start menu allows the player to exit the game.
Cross references: US-17, UC-7, B-28
Pre-conditions:

3. The user has the start menu open on the screen.
4. The cursor is showing as selecting ‘exit’

Test steps:

2. Press the enter key.

Expected result:

2. The game exits and the user returns to the normal computer screen.

T-19

Name: Lives decrease.
Description: Test that the player’s lives decrease when they are attacked by an enemy.
Cross-references: US-4, US-5, UC-3
Pre-conditions:

5. The player is in a level.
6. The player is not in the middle of another activity.
7. The tile to the north of the knight is spider.
8. The player has three lives.

Test steps:

2. Press the up arrow.

Expected result:

2. The lives tile change from having three hearts to two,

T-20

Name: Running out of lives.

Description: Test that when the player runs out of lives the level ends.
Cross-references: US-4, US-5, UC-3
Pre-conditions:

5. The player is in a level.
6. The player is not in the middle of another activity.
7. The tile to the north of the knight is spider.
8. The player has one life.

Test steps:

2. Press the up arrow.

Expected result:

3. The player sees text saying that they have run out of lives.
4. The level resets and the player returns to the start menu.

T-21a
Name: Random level (easy)
Description: Test that the player can produce two different levels when selecting the easy mode.
Cross-references: US-14, US-15, UC-7
Pre-conditions:

2. The player is at the start menu.

Test steps:

5. Press the down arrow to move the cursor to ‘random level’ and press enter.
6. Press enter to select ‘easy’.
7. Press q tot return to the start menu.
8. Repeat steps 1 and 2.

Expected result:

5. A level is generated when the player selects ‘easy’.
6. The second level is visibly different to the first.

T-21b
Name: Random level (medium)
Description: Test that the player can produce two different levels when selecting the medium mode.
Cross-references: US-14, US-15, UC-7
Pre-conditions:

2. The player is at the start menu.

Test steps:

5. Press the down arrow to move the cursor to ‘random level’ and press enter.
6. Press the down arrow and enter to select ‘medium.
7. Press q to return to the start menu.
8. Repeat steps 1 and 2.

Expected result:

7. A level is generated when the player selects ‘medium’.
8. The second level is visibly different to the first.

T-21c
Name: Random level (hard)
Description: Test that the player can produce two different levels when selecting the hard mode.
Cross-references: US-14, US-15, UC-7
Pre-conditions:

2. The player is at the start menu.

Test steps:

5. Press the down arrow to move the cursor to ‘random level’ and press enter.
6. Press the down arrow and enter to select ‘hard’.
7. Press q to return to the start menu.
8. Repeat steps 1 and 2.

Expected result:

3. A level is generated when the player selects ‘hard’.
4. The second level is visibly different to the first.

T-22a
Name: Random level
Description: Test that there are more enemies in the medium mode than the easy more.
Cross-references: US-14, US-15, UC-7
Pre-conditions:

2. The player is at the start menu.

Test steps:

7. Press the down arrow to move the cursor to ‘random level’ and press enter.
8. Select ‘easy’ mode.
9. Press the ‘m’ key.
10. Press q to return to the start menu.
11. Repeat steps 1, 2, 3, and 4 but select ‘medium’ mode.
12. Select exit to quit the game.

Expected result:

3. On the console will be two numbers preceded by ‘number of enemies’. The second number should be larger than the first.

Note: The feature to get the number of enemies is just for testing and will be removed in the final system.

T-22b
Name: Random level
Description: Test that there are more enemies in the hard mode than the medium mode.
Cross-references: US-14, US-15, UC-7
Pre-conditions:

2. The player is at the start menu.

Test steps:

7. Press the down arrow to move the cursor to ‘random level’ and press enter.
8. Select ‘medium mode.
9. Press the ‘m’ key.
10. Press q to return to the start menu.
11. Repeat steps 1, 2, 3, and 4 but select ‘hard mode.
12. Select exit to quit the game.

Expected result:

4. On the console will be two numbers preceded by ‘number of enemies’. The second number should be larger than the first.

Note: The feature to get the number of enemies is just for testing and will be removed in the final system.

T-23
Description: Test that the ranged enemy becomes visible when the player walks into it.
Cross-references: US-16, UC-6, CRC-8
Pre-conditions:

4. The player is in a level.
5. The knight currently has a dark background.
6. There is a hidden ranged enemy on the tile south of the player.

Test steps:

2. Press the down arrow

Expected result:

3. The ranged enemy tile becomes visible
4. The text is displaying saying ‘be careful of the shadows’.

T-24a

Description: Test for sound effect when the player uses the light feature.
Cross-references: US-18
Pre-conditions:

2. The player is in a level.

Test steps:

2. Press the f key.

Expected result:

2. A sound effect of a fire lighting plays.

T-24b
Description: Test for sound effect when the player opens a gate.
Cross-references: US-18
Pre-conditions:

3. The player is in a level.
4. The player is standing on a pressure plate

Test steps:

4. Press the o key.

Expected result:

4. A sound effect of a gate opening plays.

T-24c
Description: Test for sound effect when the walks into a monster.
Cross-references: US-18
Pre-conditions:

3. The player is in a level.
4. The tile to the north of the player is a spider tile

Test steps:

5. Press the up arrow.

Expected result:

5. A sound effect of a spider plays.

T-24d
Description: Test for sound effect when the player is spotted by the ranged enemy.
Cross-references: US-18
Pre-conditions:

4. The player is in a level.
5. The tile to the north of the player is lit
6. There is a ranged enemy on the same row as the lit tile.

Test steps:

6. Press the up arrow.

Expected result:

6. A sound effect for the player being spotted plays.

T-24e
Description: Test for sound effect when the player has one life.
Cross-references: US-18
Pre-conditions:

4. The player is in a level.
5. The tile to the north of the player is a spider.
6. The player has two lives.

Test steps:

2. Press the up arrow.

Expected result:

2. A sound effect for the player having one life plays.

T-24f
Description: Test for sound effect when the player loses all three lives.
Cross-references: US-18
Pre-conditions:

4. The player is in a level.
5. The tile to the north of the player is a spider.
6. The player has one life.

Test steps:

2. Press the up arrow.

Expected result:

2. A sound effect for the player losing all lives plays.

T-25a
Description: Test that the executable file opens the game on Windows machines.
Cross-references:
Pre-conditions:

4. The user is on a computer running Windows.
5. The user has downloaded and extracted the game files to a folder.
6. The user has the folder open.

Test steps:

2. Double click on the file called TinderBoxKnight_Start

Expected result:

2. The game will open.

T-25b
Description: Test that the executable file opens the game on macOS.
Cross-references:
Pre-conditions:

4. The user is on a machine with macOS.
5. The user has downloaded and extracted the game files to a folder.
6. The user has the folder open.

Test steps:

2. Double click on the file called TinderBoxKnight_Start

Expected result:

2. The game will open.