

Course Work 2 - Endless Delve - Process Documentation

Team Pink: Diparati Sen, Shuaiming Zhu, Oguz Tecirlioglu, Jack Mcgee,
Philip Read, Tom Poultney, Sam Higgs, Ruoxi Li

November 2021

Contents

1	Abstract	1
2	Sprint 1: 28/10/2021 - 06/11/2021	2
2.1	Group Meeting 1	2
2.2	Customer Meeting 1	2
2.3	First Brainstorm	3
2.4	Product Backlog 1	4
2.4.1	Map Generation	4
2.4.2	Map Design	5
2.4.3	Destructible Terrain.	5
2.4.4	Items	5
2.4.5	Audio	6
2.4.6	Loading Screen	6
2.4.7	User Interface	7
2.4.8	Player	7
2.4.9	Camera Setup	7
2.4.10	Basic Inputs	7
2.4.11	Advanced Inputs - Movement	7
2.4.12	Animations	8
2.4.13	Collisions	8
2.4.14	Items	8
2.4.15	Health Scoring	9
2.4.16	Insanity Scoring	9
2.4.17	Design	9
2.4.18	Audio	9
2.4.19	NPC's	9
2.4.20	Boss Design	9
2.4.21	Mob Design	9
2.5	Sprint Backlog 1	9
2.6	Sprint Review 1	10
2.7	Exception handling 1	10
2.8	Weekly Sprint Retrospective 1	11
3	Sprint 2: 07/11/2021 - 13/11/2021	11
3.1	Roles	11
3.2	Group Meeting 2	11
3.3	Customer Meeting 2	12
3.4	Product Backlog 2	13

3.5	User Stories 2	14
3.6	Use Cases Sprint 2	15
3.7	Sprint Backlog 2	17
3.8	CRC Cards Sprint 2	18
3.9	Sprint Review 2	19
3.10	Exception handling 2	20
3.11	Weekly Sprint Retrospective 2	20
3.12	Exception Handling 2	20
4	Sprint 3: 15/11/2021 - 20/11/2021	21
4.1	Roles	21
4.2	Group Meeting 3	21
4.3	Customer Meeting 3	21
4.4	Product Backlog 3	22
4.5	User Stories 3	26
4.6	Use Cases Sprint 3	27
4.7	CRC Cards Sprint 3	29
4.8	Sprint Backlog 3	32
4.9	Sprint Review 3	32
4.10	Exception handling 3	35
4.11	Weekly Sprint Retrospective 3	35
5	Sprint 4: 20/11/2021 - 28/11/2021	35
5.1	Roles	35
5.2	Group Meeting 4	36
5.3	Customer Meeting 4	36
5.4	Product Backlog 4	37
5.5	User Stories 4	40
5.6	Use Cases Sprint 4	40
5.7	CRC Cards Sprint 4	41
5.8	Sprint Backlog 4	44
5.9	Exception handling 4	44
5.10	Sprint Review 4	45
5.11	Weekly Sprint Retrospective 4	46
5.12	Doxygen	46
6	Sprint 5: 29/11/2021 - 05/12/2021	48
6.1	Roles	48
6.2	Group Meeting 5	48
6.3	Customer Meeting 5	49
6.4	Product Backlog 5	49
6.5	User Stories 5	51
6.6	Use Cases Sprint 5	51
6.7	CRC Cards Sprint 5	52
6.8	Formal Game Testing For Gameplay Testers	52
6.9	Sprint Backlog 5	54
6.10	Sprint Review 5	54
6.11	Exception handling 5	54
6.12	Weekly Sprint Retrospective 5	55

7	Sprint 6: 06/12/2021 - 12/12/2021	55
7.1	Roles	55
7.2	Group Meeting 6	55
7.3	Customer Meeting 6	56
7.4	Product Backlog 6	56
7.5	User Stories 6	59
7.6	Use Cases Sprint 6	59
7.7	CRC Cards Sprint 6	60
7.8	Sprint Backlog 6	61
7.9	Sprint Review 6	62
7.10	User Interface	62
7.11	Weekly Sprint Retrospective 6	68
8	Sprint 7: 13/12/2021 - 20/12/2021	68
8.1	Roles	68
8.2	Group Meetings 7	68
8.3	Customer Meeting 7	69
8.4	Product Backlog 7	69
8.5	User Stories	72
8.6	Use Cases Sprint 7	72
8.7	Sprint Backlog 7	73
8.8	Sprint Review 7	74
8.9	Exception handling 7	75
8.10	Weekly Sprint Retrospective 7	76
9	User Questionnaire Form	76

List of Figures

1	Doxygen Class Listings	47
2	Example Doxygen Class View of the Player_Controller Class	48

1 Abstract

This report documents and details the Agile development process for our game "Endless Delve". The game is a 2-D dungeon platformer, develop in Unity. The player is tasked with surviving and exploring for as long as possible, slaying mobs and collecting precious emeralds. When the player inevitably dies, their score is uploaded to an online database, and they can see if they have made it to the top of the leaderboard or not! This documentation is organised on a sprint by sprint basis, and each sprint is one week long, containing the group meeting for that week, customer meeting, product backlog, sprint backlog, use cases and user stories, CRC cards, user tests, and exception handling documentations. In total, there were seven sprints. Various digital tools were also used such as Trello, Microsoft Teams, Discord, Doxygen and Git to facilitate the implementation of agile practices. This documentation was developed throughout the seven week timeline, and thoroughly covers what the team achieved in each week. The link to our GitHub repository is available at: <https://github.com/CM50109PinkTeam/WorkingVersion>. The "Delivery" branch also contains all the documentation, as specified. The Installation Manual, also provided by the team, can be used to install, test, and play Endless Delve!

2 Sprint 1: 28/10/2021 - 06/11/2021

2.1 Group Meeting 1

- Date: 28/10/2021
- Location: CB 4.6
- Present: Jack McGee, Diparati Sen, Oguz Tecirlioglu, Phillip Read, Ruoxi Li, Sam Higgs
- Absent: Shuaiming Zhu, Tom Poultney

Agenda Covered – Meeting most of the group and establishing the strengths and weaknesses of each member. Roles within the development process were assigned based on these discussions. Discussion of the methodology we would follow, creating a risk assessment and planning for barriers to success. General Brainstorming of ideas for the game, agile techniques to utilise and software to use.

Outcomes:

- Assigned members to development and documentation groups
- Identification of scrum methodology with some techniques from XP such as pair programming
- Identification of two possible game engines (Unity and Unreal)
- Identification of barriers to success and planned around these accordingly
- Base idea for the game (dungeon game, platforming, time limited, treasure hunting)
- Identification of Trello for product backlog management
- Identification of Overleaf for documentation
- Identification of Discord for communication
- Identification of GitHub for iteration management

Additions to the Agenda:

- Research the differences between Unity and Unreal
- Set up Discord, Overleaf, Trello, GitHub
- Familiarisation of software

Meeting Duration – 60mins

2.2 Customer Meeting 1

- Date – 03/11/2021
- Location – CB 5.13
- Present - Jack McGee, Diparati Sen, Phillip Read, Sam Higgs, Tom Poultney
- Absent - Oguz Tecirlioglu, Ruoxi Li, Shuaiming Zhu

Agenda Covered – Presented the initial game ideas to the customer and received feedback. Discussed ways to retain players by randomising maps and implementing hidden rooms and perks. Explained the process and techniques we would be using to ensure an agile methodology is followed. Discussed whether to produce a single platform game or whether to produce a game compatible with mobiles and computers

Outcomes:

- Implement more ways to retain the customer
- Develop a game which runs on computer
- Continue with planned methodology

Additions to the Agenda:

- Plan perks, 'random' map generation and hidden rooms
- Ensure we go into meetings knowing who is absent and why

Meeting Duration – 15mins**2.3 First Brainstorm****Dungeon game ideas/inspiration:**

Side scrolling 'Mario' style game We discussed implementing a similar game aesthetic and play style but adapting it to suit a dungeon escape narrative.

- Cartoon art style
- Points based on time to complete each level
- Pre-defined number of levels (more would have to be released to keep the user playing)
- Minor emphasis on mobs

Exploration 'Minecraft' style game We talked through some ideas surrounding a semi open-world exploration game in which the player explores dungeons and fight mobs to find items which carry over from game to game, making the player stronger and be able to explore more of the dungeon.

- Initially one large map which the user would require multiple returns to complete it
- Each run through the level, the player would try to explore for as long as they can whilst finding items
- The further away from the initial spawn point, the better the items found, the stronger the mobs and the more dangerous the terrain

2D platforming 'Spelunky' style game We brainstormed ideas for a game in which the environment became a player, requiring the player to contend with the environment as much as contending with exploration and mobs.

- 'Insanity' meter which would increase the strength and number of mobs as well as decreasing the visibility of the player
- Player is battling against mobs, dwindling light and environmental hazards to find the exit of the level
- Each exit of a level loads another level in which the mobs and terrain are exponentially tougher
- Points accumulated through finding treasure

Game engine / coding language Those of us who had industry experience in coding were mostly familiar with C#. We discussed the use of either Unity or Unreal as our game engine. Those with experience in this area recommended Unity over Unreal. We decided to all go away and research the differences between the two.

Methodology To ensure that as a group we follow an agile methodology, we discussed which aspect of Scrum and XP we would utilise in this project. We decided on using pair programming and test-driven-development from XP and use the basic methodology of Scrum with some adaptations to fit around other commitments:

- Weekly meetings and constant availability on social media replaced daily sprint meetings
- Meetings were scheduled for Microsoft Teams instead of being in-person
- Github, Trello and Overleaf were decided on to ensure visibility of the project

Risk Assessment The group identified some risks and barriers to success. These risks were then discussed and incorporated into our plan of action. These included:

- Implications of COVID-19 risks and the possibility of restrictions and group members having to self-isolate
- Time restraints and deadlines for other university coursework
- The shift pattern of group members jobs
- Lack of coding knowledge hindering some people's ability to help with development tasks

Strengths and weaknesses / team splits As our team was clearly split by coding experience (4 were competent in programming whereas the other 4 were not) we thought it would be best to split into two subgroups, the development team and the documentation team. To adhere to agile methodology principles we discussed there being fluidity within these groups and all work would be visible to each team. Members of each team would be welcome to help with tasks on both sides but the majority of development tasks would be undertaken by those who were stronger in programming and vice versa.

2.4 Product Backlog 1

All requirements related to the environment.

2.4.1 Map Generation

1. All requirements are loaded in at the start of the game.
2. Rooms are join together to form the games map.
3. Room connections.
4. Random map generation algorithm.
5. Random mob spawners.
6. Random treasure spawners.
7. Map audio.

2.4.2 Map Design

All requirements for design of the environment.

1. Room Designs - top connections.
2. Room Designs - Bottom connections.
3. Room Designs - Left connections.
4. Room Designs - Right connections.
5. Room Designs - Start rooms.
6. Room Designs - End room.
7. Room Designs - Boss room.
8. Map lighting - dark.
9. Block Textures - Blocks are used to build each room.
10. Traps - Trap doors.
11. Traps - Spikes.
12. Traps - Arrow traps.
13. Traps - Rotating blade.
14. Map Elements - lava.
15. Map Elements - Water.
16. Map Elements - Fire.

2.4.3 Destructible Terrain.

1. All Blocks that can be destroyed using dynamite.
2. All Blocks that can't be destroyed using dynamite.

2.4.4 Items

Interactive items that are generated with the map.

1. Map Items - Wall Torches.
2. Map Items - Ladders.
3. Map Items - Ropes/ vines.
4. Map Items - Chests.
5. Map Items - Treasure.
6. Map Items - Power Ups.

2.4.5 Audio

If the sound isn't constant only when using/ interacting item.

1. Map Audio - Music.
2. Sounds effects - Fire crackly.
3. Sounds effects - Running water.
4. Sounds effects - Walking on different surfaces.
5. Sounds effects - Fire torch crackling.
6. Sounds effects - Trap door opening.
7. Sounds effects - Spikes.
8. Sounds effects - Arrow trap.
9. Sounds effects - Rotating blade.
10. Sounds effects - Rope/ vine creaking.
11. Sounds effects - Ladder creaking.
12. Sounds effects - turning on/off torch.
13. Sounds effects - Collecting treasure.
14. Sounds effects - Using items.
15. Sounds effects - Dynamite timer/ bang.
16. Sounds effects - Opening Chest.

2.4.6 Loading Screen

All requirements related to menu screen.

- Start Screen/ Menu.
- In-game pause Screen/ Menu.
- Settings Screen/ Menu.
- Leader board Screen/ Menu.
- Achievements board Screen/ Menu.
- Victory/ Defeat Screen/ Menu.
- Credits.

2.4.7 User Interface

All requirements related to the users interface in game.

1. Health Bar.
2. Insanity Bar.
3. Torch Battery Bar.
4. Game Timer.
5. Item's button/ available.
6. Treasure Collected Bar.
7. Chest Item Screen?

2.4.8 Player

All requirements related to the player.

2.4.9 Camera Setup

2.4.10 Basic Inputs

1. Movement keys - Left - key: D
2. Movement keys - Right - key: A
3. Movement keys - Up - key: W
4. Movement keys - Down - key: S
5. Movement keys - Jump - key: Space
6. Movement keys - Crouch - key: C
7. Items keys - Using Items
8. Items keys - Using torch

2.4.11 Advanced Inputs - Movement

1. Movement keys - Double jump - key: Space x2
2. Movement keys - Wall jump - key; space + wall
3. Movement keys - Rowling - key: C x2
4. Movement keys - Combos

2.4.12 Animations

1. Movement - walking.
2. Movement - Jumping.
3. Movement - crouching.
4. Movement - rolling.
5. Using items.
6. Victory animation.
7. Death animation.

2.4.13 Collisions

All collision with environment elements.

1. Lava - death.
2. Fire - damage.
3. Under water - delay damage.

All collision with environment traps.

1. Spikes - damage.
2. Arrow traps - damage.
3. Falling - certain heights and above = damage.
4. Trap Door - opening.

All collision with environment items.

1. Ladder - moving up/ down.
2. Ropes/ vines - moving up/ down.
3. NPC's - damage.

All collision with Items.

1. Explosion - damage/ death dependant of distance from explosion.

2.4.14 Items

1. Torch - animation/ lighting.
2. Dynamite - animation.

2.4.15 Health Scoring

2.4.16 Insanity Scoring

2.4.17 Design

2.4.18 Audio

1. Sounds effects - Foot steps.
2. Sounds effects - Breathing when insanity low.
3. Voice lines.

2.4.19 NPC's

All requirements related to NPC's.

2.4.20 Boss Design

2.4.21 Mob Design

2.5 Sprint Backlog 1

Development:

- Get acquainted with Unity and C#
- Set up GitHub
- Creating a project
- Risk assess merging branches – DON'T commit straight to the master branch
- Agree on a coding standard
- Look into Pair Programming and set up Live share.

Documentation:

- Weekly sprint backlog email
- Weekly sprint retrospective email
- Set up Trello
- Present basic usage
- Set up overleaf for documentation
- Set up group email for overleaf
- Set up weekly Teams/ Discord for meetings and channels
- Write up product backlog
- Write up weekly scrum backlog
- Write up first customer meeting

2.6 Sprint Review 1

For the next week, the team decided that everyone, but specifically the developers get familiar with Unity, what can be built using it, and C#. Tom set up the GitHub repository, and created the project. In the documentation side, Trello was configured and automated to provide weekly summaries of the backlog,

Development:

- Get acquainted with Unity and C# (all)
- Set up GitHub (thomasapoultney)
- Creating a project (thomasapoultney)
- Look into Pair Programming e.g. set up Live share. (thomasapoultney)

Documentation:

- Weekly sprint backlog email (samhiggs)
- Weekly sprint retrospective email(samhiggs)
- Set up Trello (samhiggs)
- Present basic usage (samhiggs)
- Set up overleaf for documentation (samhiggs)
- Set up group email for overleaf (samhiggs)
- Set up weekly Teams/ Discord for meetings and channels (thomasapoultney)
- Write up product backlog (samhiggs)
- Write up weekly scrum backlog (samhiggs)
- Write up first customer meeting (jackmcgee)

2.7 Exception handling 1

Unexpected Situation	Status	Actions to resolve
The team was unable to decide between using unity or unreal engine for the project. six group members favoured using unity and the remaining two members favoured unreal	Resolved	We felt it was important to make a start on the project immediately so we temporarily began using unity until unreal engine could be properly researched. We set a hard deadline for the 8/11/2021 where the team members favouring unreal could present evidence for its benefits. After this pitch we will have an anonymous vote and whichever game engine has the majority of votes will be the one that we use.

2.8 Weekly Sprint Retrospective 1

This week was used to mimic a Scrum's "Pre-Game" phase. Specifically the planning sub-Phase looking at creating a product backlog for the system and and Architecture and Design sub-phase. Our first group meeting was used to brainstorm our ideas for the game and questions that we had for the first customer meeting, such as the platform of the game. On reviewing the first sprint some of the outlined items were deemed to be unnecessary such as a WhatsApp group as Microsoft Teams and Discord seemed satisfactory.

During our brainstorm for the game we discussed each of the team members skills and roles they would like to take on. Due to the variation in programming experience some members didn't wish to develop any code without the help of a more experience programmer. Thomas, Oguz, Diparati and Shuaiming all had previous experience with either C#, unity or other programming projects. Sam, Jack Ruoxi and Phillip and had less programming experience and would mainly handle the documentation side of the project. Sam would be the Product Owner and handle/ write up the Product backlog. Sam and Tom would alternate as Scrum masters each, deciding what would go into the Sprint backlogs and updating Trello with the correct artefacts.

Following our process meeting with the TA's we decided that our developer task assignment process was not suitable for future sprints. The process in this week was that each developer would choose the features that they personally felt most comfortable developing in a first come, first served nature. During this week we found that we significantly underestimated the length of time it would take to complete a given task. Furthermore, we were accepting tasks without fully understanding the inherent risks associated with them. To overcome this issue we will discuss each item in the product backlog in-depth in the next sprint-planning meeting. Each item of the backlog will be assigned a story points value. The story point value will be an estimated measure of the overall effort that it will take to complete a given item. It will account for the amount of work required, The complexity of the work and most importantly any associated uncertainty. This is particularly important since most team members have limited experience with Unity and therefore cannot accurately provide an exact estimate of how long a task will take to complete.

3 Sprint 2: 07/11/2021 - 13/11/2021

3.1 Roles

- Product Owner - Sam Higgs
- Scrum Master - Thomas Poultney

3.2 Group Meeting 2

- Date – 09/11/2021
- Location – Teams
- Present - Jack McGee, Diparati Sen, Sam Higgs, Tom Poultney, Oguz Tecirlioglu, Ruoxi Li, Shuaiming Zhu
- Absent – Phillip Read

Agenda Covered – Investigated methods of map generation using the Unity game engine. Discussed further elements of the game such as hidden rooms, art design and UI. Assigned areas of further research in preparation for the next customer meeting

Outcomes:

- Jack McGee was assigned to collate art designs to present to the customer
- Diparati Sen and Oguz Tecirlioglu were assigned to research the differences between Unity and Unreal game engines
- Diparati Sen was also assigned to investigate creating a UX to present to the customer
- Tom Poultney and Shuaiming Zhu were assigned to begin looking at map generation and create a short presentation for
- Sam Higgs and Phillip Read were assigned to begin writing up the product backlog

Added to the Agenda:

- Present research to the customer to gain a better understanding of the aesthetic and design of the game
- Test map generation
- Code for game physics

Meeting Duration – 60mins**3.3 Customer Meeting 2**

- Date – 10/11/2021
- Location – CB 5.13
- Present - Jack McGee, Diparati Sen, Sam Higgs, Tom Poultney, Phillip Read
- Absent - Oguz Tecirlioglu, Ruoxi Li, Shuaiming Zhu

Agenda Covered – Presented five different art styles to the customer along with our idea for how the map generation will work. Discussed the addition of hidden rooms/vaults for an added exploration element.

Outcomes:

- Customer decided on a more realistic looking art style (similar to Tomb Raider: Underground)
- Customer was happy with the map generation and the addition of vaults
- Customer suggested a customisation element of the character and the environment

Added to the Agenda:

- Create a character and map in the desired art design
- Look in to customisation options
- Provide a working prototype for the next meeting

Meeting Duration – 15mins

3.4 Product Backlog 2

Table 1: Product Backlog

ID	Description	Story Points	Priority	Date Completed
001	Map: Room Spawning Algorithm	300	1	12/11/2021
002	Map: Map loaded at the start of the game	20	1	12/11/2021
003	Map: Room design	15 (per room)	1	12/11/2021
004	Map: Rooms joined together	100	1	12/11/2021
005	Map: Room connections	50	1	12/11/2021
006	Player: Movement left and right	20	1	
007	Player: Movement Jumping	20	1	
008	Player: Movement up and down	20	1	
009	Player: Movement crouching	30	1	
010	Player: Physics (gravity)	10	1	
011	Player: Input left = D	10	2	
012	Player: Input right = A	10	2	
013	Player: Input Up = W	10	2	
014	Player: Input Down = S	10	2	
015	Player: Input Jump = SPACE	10	2	
016	Player: Input crouch = C	10	2	
017	Player: Climb ropes	100	2	
018	Player: Movement variable jumping height	75	2	
019	Player: Animations for all movements	400	2	
020	Player: Animations for all attacks	400	2	
021	Player: Animations for all collisions	300	2	
022	Map: Random treasure spawners	150	2	
023	Map: Lighting	400	2	
024	Map: Start and end to map	150	2	
025	Map: Map background music	25	2	
026	Player: Attack kicking	75	3	
027	Player: Attack punching	75	3	
028	Player: Movement sliding	75	3	
029	Player: Movement crouching	75	3	
030	Player: Attack combos on ground	75	3	
031	Player: Attack combos in air	75	3	
032	NPC's: Ghost npc's Design	150	3	
033	NPC's: Skull npc's Design	150	3	
034	NPC's: Archer npc's Design	250	3	
035	NPC's: Fire skull npc's Design	150	3	
036	NPC's: Slime npc's Design	150	3	
037	NPC's: Ghost npc's scripts	150	3	
038	NPC's: Skull npc's scripts	200	3	
039	NPC's: Archer npc's scripts	300	3	
040	NPC's: Fire skull npc's scripts	200	3	
041	NPC's: Slime npc's scripts	200	3	
042	NPC's: Ghost npc's animations	100	3	
043	NPC's: Skull npc's animations	200	3	
044	NPC's: Archer npc's animations	250	3	
045	NPC's: Fire skull npc's animations	100	3	

046	NPC's: Slime npc's animations	200	3	
047	NPC's: Mob patrolling script	150	3	
048	Map: Random mob spawners	150	3	
049	Player: Swinging from ropes	300	4	
050	Player: Spawn/ remove ropes	100	4	
051	Map: Ropes	100	4	
052	Map: Vines	100	4	
053	Map: ladders	100	4	
054	Map: Rope and Vine physics	300	4	
055	Map: Rope max and min angles	100	4	
056	Map: Wall torches	100	4	
057	NPC's: Mob collision	75 (per mob)	4	
058	Player: Wall Jumping	100	5	
059	Player: Blood animation on collision	100	5	
060	Player: Taking damage on collision	250	5	
061	Player: Spawn torch = F	75	5	
062	Player: Bounce animation when landing	50	5	
063	NPC's: Blacksmith Design	250	5	
064	NPC's: Blacksmith Animations	250	5	
065	User Interface: Game behind main menu	150	6	
066	User Interface: Health bar	150	6	
067	User Interface: Insanity bar	150	6	
068	User Interface: Treasure counter	150	6	
069	Loading Screen: Start menu	150	6	
070	Loading Screen: leader board menu	150	6	
071	Loading Screen: Winner screen	150	6	
072	Loading Screen: Death screen	150	6	
073	Loading Screen: Settings menu	150	6	

3.5 User Stories 2

User Story 001: As a player,

The game map to be made of rooms randomly connected together,

So that,

Each time I play a game the map will potentially be different.

Acceptance Criteria;

The game loads and runs with a different map made of preset connected rooms.

User Story 002: As a player,

I want the map to load in when starting the game,

So that,

Each time I play a game the map will appear when starting the game.

Acceptance Criteria;

The game map is loaded upon starting the game.

User Story 003: As a player,

I want multiple unique rooms that make up an overall map,

So that,

Each time I play a game, each room of the map should feel unique.
Acceptance Criteria;
No two room are the same/similar in design.

User Story 004: As a player,
I want each room of the map to join together,
So that,
Each time I play a game the map will entrance(s) and exit(s) to each room.
Acceptance Criteria;
The game loads rooms correctly joined together.

User Story 005: As a player,
I want each room to have unique entrance and exit points
So that,
Each time I play a game i will be able to enter and exit each room of the map.
Acceptance Criteria;
The game loads with rooms that have endurance(s) and exit(s).

3.6 Use Cases Sprint 2

For the rest of the documentation, note that Use Cases may be referred to as **UC**, and User Stories may be referred to as **US**.

UC 000: Player Movement Fundamentals

Scope: N/A

Level: User goal

Primary Actors: Player

Description: The User must be able to interact with the player in the game, and control the player by providing basic movement inputs through the "A", "W", "S", and "D" keys on their keyboard. The movements must be animated accordingly.

Dependencies: UC 014: Player Starts the game.

UC 003: Room design.

UC 004: Rooms join together.

UC 005: Room connections.

Assumptions: None.

Preconditions: The maps is generated, and the game has started.

Main Flow: The player is spawned into the map, and the user would like to start playing the game. The player moves with these basic keyboard inputs, and navigates the 2D map. The movements are all animated.

Subflows: None.

Alternative Flows: None.

Post Conditions: When the player dies or falls into lava, input controls are disabled.

Frequency of Occurrence: Every time the game Starts.

Open Issues: None

Originating User Story: US 007, US 008

UC 001: Room Spawning Algorithm

Scope: N/A

Level: User goal

Primary Actors: Player

Description: Spawning rooms
Dependencies: UC 014: Player Starts the game.
UC 003: Room design.
UC 004: Rooms join together.
UC 005: Room connections.
Assumptions: None.
Preconditions: Player has selected start the game.
Main Flow: Right after the player starts game, the map should be created and be made up of rooms.
Subflows: None.
Alternative Flows: None.
Post Conditions: The the map is then loaded into the game.
Frequency of Occurrence: Every time the game Starts.
Open Issues: How many rooms make up the map?
Originating User Story: US 001

UC 002: Map loaded at the start of the game

Scope: N/A
Level: User goal
Primary Actors: Player
Description: loading the entire map at the start of the game.
Dependencies: UC 014: Player Starts the game from Main Menu.
UC 001: Room Spawning Algorithm.
Assumptions: rooms are correctly joined together to form the map.
Preconditions: Player has selected start the game.
The map has be generated correctly.
Main Flow: Right after the player starts game, the map should be created and loaded into the game.
Subflows: None.
Alternative Flows: None.
Post Conditions: The player is then loaded into the map.
Frequency of Occurrence: Every time the game Starts.
Open Issues: None.
Originating User Story: US 002

UC 003: Room Design

Scope: N/A
Level: User goal
Primary Actors: Player
Description: Uniquely designed rooms that make up map.
Dependencies: None.
Assumptions: None.
Preconditions: None.
Main Flow: Each room prefab has unique map design elements in it. These elements should have a chance of spawning in rather than always spawning to facilitate unique room design.
Subflows: None.
Alternative Flows: None.
Post Conditions: the rooms are then loaded into the room spawning algorithm.
Frequency of Occurrence: Every time the game Starts.
Open Issues: Enough unique room designs to make unique feeling maps?
Originating User Story: US 003

UC 004: Rooms join together

Scope: N/A

Level: User goal

Primary Actors: Player

Description: All rooms join together correctly to form the map.

Dependencies: UC 003: Room Design.

UC 005: Room connections.

Assumptions: Each room has entrance(s) and exit(s) in preset positions to allow for movement between each room.

Preconditions: None.

Main Flow: The map spawner script begins running in the scene and iteratively spawns suitable rooms together, joining them together at their borders.

Subflows: None.

Alternative Flows: None.

Post Conditions: the rooms are then loaded into the room spawning algorithm.

Frequency of Occurrence: Every time the game starts.

Open Issues: A correct path through the map from start to finish.

Originating User Story: US 004

UC 005: Room connections

Scope: N/A

Level: User goal

Primary Actors: Player

Description: All rooms have preset joins.

Dependencies: UC 003: Room Design.

Assumptions: Each room is designed to have entrance(s) and exit(s) in preset positions.

Preconditions: None.

Main Flow: None.

Subflows: None.

Alternative Flows: None.

Post Conditions: the rooms are then loaded into the room spawning algorithm.

Frequency of Occurrence: Every time the game starts.

Open Issues: Room opening(s) and exit(s) all occur in the middle of each room's side in the same positions.

Originating User Story: US 005

3.7 Sprint Backlog 2

ID	Description	Developer
001	Map: Room Spawning Algorithm	Thomas
002	Map: Map loaded at the start of the game	Thomas
003	Map: Room design	Thomas
004	Map: Rooms joined together	Thomas
005	Map: Room connections	Thomas

3.8 CRC Cards Sprint 2

Class Name: Map Spawner	Version: 1
Description: Spawns all the rooms within a map	Associated Use Cases: UC 001, UC 002
Responsibilities	Collaborators
Spawn World Borders Spawn Critical Path Rooms Spawn World Borders	Room Connections Room

Class Name: Room Spawner	Version: 1
Description: Spawn a single room and its contents	Associated Use Cases: UC 002, UC 003
Responsibilities	Collaborators
Spawn block Spawn spike	Block Spike

Class Name: Room Connection	Version: 1
Description: Checks rooms fit together	Associated Use Cases: UC004, UC005
Responsibilities	Collaborators
Check room type connects with previous room Delete room	Map Spawner Room

Class Name: Spike	Version: 1
Description: Spawns a single spike	Associated Use Cases: N/A
Responsibilities	Collaborators
Damage Player Delete room	Player Room

Class Name: Block	Version: 1
Description: Spawns a single block	Associated Use Cases: N/A
Responsibilities	Collaborators
	Player

Class Name: Player	Version: 1
Description: Controls the player behaviour	Associated Use Cases: N/A
Responsibilities	Collaborators
Move Jump Attack tracks health	

3.9 Sprint Review 2

This week the map boundaries, map generation, and room prefabs were developed. Additionally, test environments (special rooms where specific functionality can be tested) were created so that it would be easier to test features. For next week, the room generation is envisaged to be finalised, and will be tweaked according to the customers desire or how the team sees ideal for the customers needs.

In Development: None.

Bugs: None.

Blocked: None.

Done:

- Creating solid map boundaries (thomasapoultney)
- Generating critical path rooms (thomasapoultney)
 - Rooms are joined in the middle of the room so no room entrances are blocked.
- Generating padding rooms (thomasapoultney)
- Creating basic block prefab (thomasapoultney)
- Creating spike prefab/spawning script (thomasapoultney)
- Creating prefab creation scene (thomasapoultney)
- Creating player controller test environment (thomasapoultney)
- Creating map demonstration (thomasapoultney)
- Set up the initial Layer Masks (Room, BasicBuildingBlock, Spike, Rope) (thomasapoultney)
- Tags (only Destructible so far) (thomasapoultney)

Documentation:

- Updating Product Backlog (samhiggs)
- Updating Sprint Backlog (samhiggs)
- Adding Product Backlog to Trello (samhiggs), (phillipread), (rouxili)
- Updated Weekly Activity Document (samhiggs)
- Created concept artwork presentation (jackmcgee)
- Written up meeting minutes (jackmcgee)

Unexpected Situation	Status	Actions to resolve
An unforeseen situation for one of the group members that caused some disruptions in the completion of tasks occurred	Risk mitigated, but not resolved	We communicated effectively through Microsoft Teams, and the sprint and product backlogs were adjusted to account for the lost development time.
Git-hub merging issues	Fixed	For this issue, we restructure the project to match structure of backlog and it broke the git-hub when merging. We also unified the version of Unity that we were using.

3.10 Exception handling 2

3.11 Weekly Sprint Retrospective 2

The Development for this week was to product a functioning map and testing area to allow for the developments of items in preceding sprints i.e. player movement etc. this included implementing a complex algorithm that was join rooms together to form a amp that has a distinct path from start to finish. The development of this went well with some online aids being used to implement the correct algorithm, however due to it's complexity not all the developers could aid in this task and as such a development 'bottleneck' occurred. this meant that some of the developers decided to aid in documenting this week; Oguz and Dipariti worked on complete various User Stories and Cases.

This week the documenting team updated the Product Backlog to be numerated and cross referenced so artifacts can be easily referenced in the Sprint Backlogs, increasing transparency for the developers. Each of the items were also given a story point estimation to estimate the relative difficult of implementing a specific artifact. However, As each developer has different skill levels, an overall estimation was deemed 'too broad' and unsatisfactory for modeling the perceived difficulty of implementation. For example, a developer with experience using the universal render pipeline in unity would not estimate Map lighting as being four times as difficult as creating animations. As of the next Sprint the Product Backlog will have estimations for each individual Developer. This will allow us to more efficiently assign tasks to the most suitable developer. Furthermore, The Product Backlog is now ranked and includes a priority index for the relative importance of each item. This will help document which artifacts need to be implemented first and provide clarity as to what will be added to the next sprint backlog. Each item in the Backlogs are then cross referenced with each weeks User Stories and User Cases.

3.12 Exception Handling 2

Date: 04/11/2021 - 20/11/2021. Group Members Affected: 1 An unforeseen situation for one of the group members that caused some disruptions in the completion of tasks occurred between the 4th of November and 20th of November. A developer went to Manchester to visit a friend, but upon discovery of house pests (carpet beetle infestation) at the location, had to extend their stay to ensure cross-contamination did not occur between the houses. The group member had limited ability to carry out work in such an environment, and could not return to their home for an additional week than originally planned. This was a totally unexpected event for the whole group, The allocation of some of the items on the backlog had to be shifted between remaining developers. Specifically, player movement scripts were picked up by Thomas Poultney. This was an extremely low probability, high impact event, but strong communication enabled it to be overcome. The situation was communicated effectively through Microsoft Teams, and the sprint and product backlogs were reduced in scope accordingly.

4 Sprint 3: 15/11/2021 - 20/11/2021

4.1 Roles

- Product Owner - Sam Higgs, Phillip Read and Ruoxi Li
- Scrum Master - Thomas Poultney and Sam Higgs

4.2 Group Meeting 3

- Date – 15/11/2021
- Location – Teams
- Present - Diparati Sen, Sam Higgs, Tom Poultney, Phillip Read, Oguz Tecirlioglu, Shuaiming Zhu
- Absent – Jack McGee, Ruoxi Li

Agenda Covered – The development team presented their code and the way which the map generation will work. Discussed how to optimise the code and the size of the individual rooms and total map.

Outcomes:

- Working map generation
- Development team to work on character generation and movement
- Test environment is needed
- Map available on GitHub

Added to the Agenda:

- Present the working map generation to the customer
- Begin coding for a working character with physics
- Implement code to ensure that the customer always has a clear path from start to finish in the game

Meeting duration – 67mins

4.3 Customer Meeting 3

- Date – 17/11/2021
- Location – CB 5.13
- Present – Diparati Sen, Phillip Read, Tom Poultney, Sam Higgs, Ruoxi Li, Shuaiming Zhu
- Absent – Jack McGee

Agenda Covered – Presented the map generation in practice, customer was happy with this method of introducing replay ability. Introduced the idea of having boss fights following escape from each room. Increasing difficulty with each successful escape. Presented the product backlog.

Outcomes:

- Focus on creating a working game before fleshing out the game
- Develop the language within the product backlog

Added to the Agenda:

- Create a fully functioning game environment and character movement
- Update the product backlog and develop the language
- Enumerate the backlog so its easier to keep track of everything

Meeting Duration – 15mins

4.4 Product Backlog 3

ID	Description	Tom	Gary	Diparati	Oguz	Priority	Date Completed
006	Player: Movement left and right	5 mins	5mins	15 mins	5 mins	1	18/11/2021
007	Player: Movement Jumping	5 mins	5mins	15 mins	5 mins	1	18/11/2021
008	Player: Movement up and down	5 mins	5mins	15 mins	5 mins	1	18/11/2021
009	Player: Movement crouching	5 mins	5mins	15 mins	5 mins	1	18/11/2021
010	Player: Physics (gravity)	5 mins	5mins	15 mins	0 mins	1	18/11/2021
011	Player: Input left = D	5 mins	5mins	15 mins	5 mins	2	18/11/2021
012	Player: Input right = A	5 mins	5mins	15 mins	5 mins	2	18/11/2021
013	Player: Input Up = W	5 mins	5mins	15 mins	5 mins	2	18/11/2021
014	Player: Input Down = S	5 mins	5mins	15 mins	5 mins	2	18/11/2021
015	Player: Input Jump = SPACE	5 mins	5mins	15 mins	5 mins	2	18/11/2021
016	Player: Input crouch = C	5 mins	5mins	15 mins	5 mins	2	18/11/2021
017	Player: Climb ropes	2 hrs	5hrs	15 hrs	10 hrs	2	15/11/2021
018	Player: Movement variable jumping height	30 mins	1hrs	3 hrs	2 hrs	2	
019	Player: Animations for all movements	8 hrs	9hrs	12 hrs	10 hrs	2	17/11/2021
020	Player: Animations for all attacks	4 hrs	5hrs	12 hrs	10 hrs	2	17/11/2021
021	Player: Animations for all collisions	2hrs	4hrs	6 hrs	5 hrs	2	
022	Map: Random treasure spawners	1 hrs	2hrs	3 hrs	2 hrs	2	17/11/2021
023	Map: Lighting	2 hrs	5hrs	10 hrs	10 hrs	2	2021/11/15
024	Map: Start and end to map	2 hrs	4hrs	10 hrs	10 hrs	2	2021/11/15
025	Map: Map background music	15 mins	30mins	30 mins	30 mins	2	2021/11/15
026	Player: Attack kicking	1 hr	2hrs	3 hrs	2 hrs	3	
027	Player: Attack punching	1 hr	3hrs	3 hrs	2 hrs	3	
028	Player: Movement sliding	1 hr	3hrs	3 hrs	2 hrs	3	
029	Player: Movement crouching	1 hr	3hrs	3 hrs	2 hrs	3	
030	Player: Attack combos on ground	1hr	4hrs	5 hrs	4 hrs	3	
031	Player: Attack combos in air	1h	3hr	5 hrs	4 hrs	3	
032	NPC's: Ghost npc's Design	4 hrs	5hrs	10 hrs	10 hrs	3	
033	NPC's: Skull npc's Design	4 hrs	5hrs	10 hrs	10 hrs	3	
034	NPC's: Archer npc's Design	4 hrs	4hrs	10 hrs	10 hrs	3	
035	NPC's: Fire skull npc's Design	4 hrs	6hrs	10 hrs	10 hrs	3	

036	NPC's: Slime npc's Design	4 hrs	4hrs	10 hrs	10 hrs	3	
037	NPC's: Ghost npc's scripts	4 hrs	5hrs	10 hrs	10 hrs	3	
038	NPC's: Skull npc's scripts	4 hrs	9hrs	10 hrs	10 hrs	3	
039	NPC's: Archer npc's scripts	4 hrs	4hrs	10 hrs	10 hrs	3	
040	NPC's: Fire skull npc's scripts	4 hrs	6hrs	10 hrs	10 hrs	3	
041	NPC's: Slime npc's scripts	2 hrs	5hrs	10 hrs	10 hrs	3	
042	NPC's: Ghost npc's animations	2 hrs	3hrs	10 hrs	10 hrs	3	
043	NPC's: Skull npc's animations	2 hrs	4hrs	10 hrs	10 hrs	3	
044	NPC's: Archer npc's animations	2 hrs	2hrs	10 hrs	10 hrs	3	
045	NPC's: Fire skull npc's animations	2 hrs	5hrs	10 hrs	10 hrs	3	
046	NPC's: Slime npc's animations	2 hrs	3hrs	10 hrs	10 hrs	3	15/11/2021
047	NPC's: Mob patrolling script	4 hrs	5hrs	10 hrs	10 hrs	3	
048	Map: Random mob spawners	2 hrs	5hrs	10 hrs	10 hrs	3	15/11/2021
049	Player: Swinging from ropes	4 hrs	6hrs	10 hrs	10 hrs	4	15/11/2021
050	Player: Spawn/ remove ropes	1 hrs	2hrs	4 hrs	2 hrs	4	15/11/2021
051	Map: Ropes	4 hrs	6hrs	10 hrs	10 hrs	4	15/11/2021
052	Map: Vines	2 hrs	4hrs	10 hrs	10 hrs	4	15/11/2021
053	Map: ladders	2 hrs	2hrs	6 hrs	5 hrs	4	
054	Map: Rope and Vine physics	2 hrs	3hrs	6 hours	5 hrs	4	15/11/2021
055	Map: Rope max and min angles	3 hrs	6hrs	5 hrs	5 hrs	4	15/11/2021
056	Map: Wall torches	1 hr	3hrs	5 hrs	3 hrs	4	
057	NPC's: Mob collision	2 hrs	5hrs	6 hrs	4 hrs	4	
058	Player: Wall Jumping	2 hrs	3hrs	5 hrs	4 hrs	5	
059	Player: Blood animation on collision	1 hr	1hrs	2 hrs	1 hrs	5	
060	Player: Taking damage on collision	1 hr	1hrs	2 hrs	1 hrs	5	
061	Player: Spawn torch = F	30 mins	30mins	2 hrs	1 hrs	5	
062	Player: Bounce animation when landing	10 mins	25mins	1 hr	20 mins	5	
063	NPC's: Blacksmith Design	3 hrs	4hrs	10 hrs	10 hrs	5	
064	NPC's: Blacksmith Animations	2 hrs	3hrs	10 hrs	10 hrs	5	
065	User Interface: Game behind main menu	8 hrs	4hrs	2 hrs	10 hrs	6	
066	User Interface: Health bar	1 hrs	2hrs	2 hrs	4 hrs	6	

067	User Interface: Insanity bar	1 hrs	3hrs	2 hrs	4 hrs	6	
068	User Interface: Treasure counter	1 hrs	3hrs	2 hrs	4 hrs	6	
069	Loading Screen: Start menu	5 hrs	4hrs	3 hrs	10 hrs	6	
070	Loading Screen: leader board menu	20 hrs	4hrs	3 hrs	10 hrs	6	
071	Loading Screen: Winner screen	5 hrs	6hrs	5 hrs	10 hrs	6	
072	Loading Screen: Death screen	5 hrs	6hrs	6 hrs	5 hrs	6	
073	Loading Screen: Settings menu	3 hrs	4hrs	3 hrs	10 hrs	6	
074	Map: Rope physics	6 hrs	4hrs	3 hrs	10 hrs	1	15/11/2021
B001	Map: Rope max and min angles adjusted	1 hrs	2hrs	3 hrs	10 hrs	4	16/11/2021
B002	Map: Mobs spawning behind tile	3 hrs	4hrs	3 hrs	10 hrs	3	16/11/2021
B003	Map Generation: Borders Box Collider Not Resizing	30 mins	1hrs	3 hrs	10 hrs	1	16/11/2021
B004	Map Generation: Player getting stuck on tiles	2 hrs	3hrs	3 hrs	10 hrs	1	16/11/2021
B005	Map Generation: Ropes spawn on top of each other.	1 hr	3hrs	3 hrs	10 hrs	4	16/11/2021

4.5 User Stories 3

User Story 006: As a player,
I want map to appear mobs randomly.
So that,
Each time I play a game the map will potentially be different.
Acceptance Criteria; The game loads and runs with a different place of mobs in room.

User Story 007: As a player,
I want I have movement keys: A, W, S, D
So that,
I can move with movement keys in game to reach where I want to go.
Acceptance Criteria;
Player moves left with A, climbs up ropws with W, slides down ropes or walls with S, and moves right with D.

User Story 008: As a player,
I want all movements have animations
So that,
every time I move I can clearly see my movement changes.
Acceptance Criteria;
Players have animations when they move.

User Story 009: As a player,
I want my all attacks have animations
So that,
every time I attack I can see my posture change in game.
Acceptance Criteria;
Players have animations when they have attacks.

User Story 010: As a player,
I want map to appear treasure spawners randomly.
So that,
Each time I play a game the map will potentially be different and I can collect the treasures to gain points.
Acceptance Criteria; The game loads and runs with a different place of treasure spawners in room and the player can collect the treasures to gain points.

User Story 011: As a player,
I want map has sound effect.
So that,
Each time I play a game, I can sound audio.
Acceptance Criteria; The game loads and runs with audio and audio can be controlled by player.

User Story 012: As a player,
I want each rooms to have ropes to swing, climb, and jump from it
So that,
the user has a unique interaction with the game.
Acceptance Criteria;
Players can climb with ropes in game.

User Story 013: As a player,

I want the map to have no natural light, and my player to be the main source of illumination. Through time, my insanity will increase, decreasing my illumination.

So that,

each time I play a game, I can see no natural light, except the one my player radiates.

Acceptance Criteria; The game map is only illuminated by the player, Lava, and torches.

User Story 014: As a player,

I want each map to have lava.

So that,

I can try to get through more different obstacles.

Acceptance Criteria; The game runs with map has lava.

4.6 Use Cases Sprint 3

UC 006: Random mob spawners

Scope: N/A

Level: User goal

Primary Actors: Player

Description: Random mob spawners

Dependencies: UC 014: Player Starts the game.

UC 003: Room design.

Assumptions: None.

Preconditions: Player is in the game. Main Flow: Right after the player starts game, the map should be created with random mob spawner.

Subflows: None.

Alternative Flows: None.

Post Conditions: The rooms are then loaded into the room with mob spawners.

Frequency of Occurrence: Every time the game Starts.

Open Issues: How many mob spawners in the map?

Originating User Story: US 006

UC 007: Random treasure spawners

Scope: N/A

Level: User goal

Primary Actors: Player

Description: Random treasure spawners

Dependencies: UC 014: Player Starts the game.

UC 003: Room design.

Assumptions: None.

Preconditions: Player is in the game. Main Flow: Right after the player starts game, the map should be created with random treasure spawner.

Subflows: None.

Alternative Flows: The game is loaded without treasure spawners.

Post Conditions: The rooms are then loaded into the room with treasure spawners, the player can collect the treasures to gain points.

Frequency of Occurrence: Every time the game Starts.

Open Issues: How many treasure spawners in the map?

Originating User Story: US 010

UC 008: Audio

Scope: N/A

Level: User goal

Primary Actors: Player

Description: Game starts with audio.

Dependencies: None.

Assumptions: None.

Preconditions: Player is in the game.

Main Flow: None.

Subflows: None.

Alternative Flows: None.

Post Conditions: The game is played with audio.

Frequency of Occurrence: Every time the game Starts.

Open Issues: Is there any music when the game end?

Originating User Story: US 011

UC 009: Rope

Scope: N/A

Level: User goal

Primary Actors: Player

Description: Ropes are implemented into the map design so that the player can navigate the map in more than one way. By being able to swing, climb, and jump from ropes, the user has a unique interaction with the game.

Dependencies: None.

Assumptions: None.

Preconditions: The prefabs for each room is completed, so that the rope spawn tiles can be added to them.

Main Flow: Player spawns in to the map. Map Generation script adds the ropes, with their respective lengths, from the rope spawn tile. The player starts navigating the map, and depending on the situation, can jump to grab a rope. By using the W or S keys, the player can traverse up and down the rope, and A and D keys to swing. The player can detach or jump off from the rope by pressing "Space".

Subflows: None.

Alternative Flows: None.

Post Conditions: Rope keeps swinging in a realistic manner after the player jumps off, conserving momentum.

Frequency of Occurrence: Every tile that a rope spawn tile exists.

Open Issues: None.

Originating User Story: US 012

UC 010: Lighting

Scope: N/A

Level: User goal

Primary Actors: Player and Lava

Description: The game needs lighting to be played, as the dungeon is naturally dark. The player will naturally illuminate their environment, but the light that the player radiates decreases over time, as Insanity increases. The player can place down a torch using the "F" key, which will illuminate a small radius.

Additionally, the Lava must act as a source of light, illuminating a small distance around itself.

Dependencies: UC 013, insanity.

Assumptions: None.

Preconditions: The rest of the rooms and map generation has been completed, so that these illuminations have an environment to be implemented in.

Main Flow: Player spawns in the map, constantly illuminating around themselves, but at a decreasing radius (due to insanity). If the player finds the exit to the current maze, upon entering the new maze insanity will be reset. The Lava also illuminates its surrounding. Whenever the player wants, they can place down a torch using "F".

Subflows: None.

Alternative Flows: User stays in the same map for so long that insanity causes the illumination around the player to go to 0, effectively making the player blind.

Post Conditions: The game is played with audio.

Frequency of Occurrence: Every time the game Starts.

Open Issues: Is there any music when the game end?

Originating User Story: US 013

4.7 CRC Cards Sprint 3

Overview of CRC Card Changes: The rope class will be introduced this week. As a result The player object gained too many responsibilities so was broken down into two separate classes: 'The player Rope Controller' and the 'Player Movement Controller'. This will be reflected in our code base. The map must also spawn in mobs, treasure and lava now so new versions of the map spawner and room spawner CRC cards have been produced. CRC Cards have been created for the audio and lighting use cases that were added this week. They have been broken down into separate audio objects for each type of object we want to control audio for. This will ensure modularity and cohesion in the system. We created two separate classes for mob controller and spawner for the same reasons.

Class Name: Map Spawner	Version: 2
Description: Spawns all the rooms within a map, Storing the position of each object and populating the world	Associated Use Cases: UC001, UC002, UC006, UC007
Responsibilities	Collaborators
Spawn World Borders Spawn Critical Path Rooms Spawn Mobs Spawn treasure Track Positions of objects	Room Connections Room Mob Treasure

Class Name: Player Controller	Version: 2
Description: Handles player basic movement	Associated Use Cases: N/A
Responsibilities	Collaborators
Move Jump Attack Mob Controller tracks health	

Class Name: Player Rope Controller	Version: 1
Description: Handles player interactions with ropes	Associated Use Cases: UC009
Responsibilities	Collaborators
Climb Rope Slide on Rope Swing on rope Attach to rope	Rope Player Controller

Class Name: Rope	Version: 1
Description: The position and physics constraints of the rope	Associated Use Cases: UC009
Responsibilities	Collaborators
calculate swinging physics Attach player to rope section	Player Controller

Class Name: Room Spawner	Version: 2
Description: Spawn a single room and its contents	Associated Use Cases: UC002, UC003
Responsibilities	Collaborators
Spawn block Spawn spike Spawn Rope Spawn Lava Spawn World Borders	Block Spike Rope Lava

Class Name: Treasure Spawner	Version: 1
Description: Responsible for spawning a single piece of treasure and tracking its value	Associated Use Cases: UC007
Responsibilities	Collaborators
Spawn a piece of treasure Return the treasure's Value	Player

Class Name: Mob Spawner	Version: 1
Description: Responsible for spawning a single mob	Associated Use Cases: UC006
Responsibilities	Collaborators
Spawn a random mob	Map Spawner

Class Name: Mob controller	Version: 1
Description: Responsible for controlling a mob's behaviour	Associated Use Cases: UC006
Responsibilities	Collaborators
Set mob to patrol Attack Player Die Take damage	Map Spawner Player

Class Name: Player Audio controller	Version: 1
Description: Responsible for playing audio when a player event occurs	Associated Use Cases: UC008
Responsibilities	Collaborators
Play death sound Play attack sound Play sword attack sound Play jump sound	Player

Class Name: Map Audio controller	Version: 1
Description: Responsible for playing audio when a map event occurs	Associated Use Cases: UC008
Responsibilities	Collaborators
Play open door sound Play lava effect sounds	Map Spawner Lava

Class Name: Mob Audio controller	Version: 1
Description: Responsible for playing audio when a mob event occurs	Associated Use Cases: UC008
Responsibilities	Collaborators
Play death sound Play attack sound Play damage sound Play explosion sound	Mob

Class Name: Lighting	Version: 1
Description: Responsible for generating the lighting for a class	Associated Use Cases: UC010
Responsibilities	Collaborators
Produce torch flickering effect Reduce lighting around player based on insanity Add glow to lava	Spawn Torch Player Lava

Class Name: Lava	Version: 1
Description: Spawn a single tile of lava	Associated Use Cases: N/A
Responsibilities	Collaborators
Kill Player Spawn Lava	Player

4.8 Sprint Backlog 3

ID	Description	Developer
006	Player: Movement left and right	Oguz, Shuaiming
007	Player: Movement Jumping	Oguz, Shuaiming
008	Player: Movement up and down	Oguz, Shuaiming
009	Player: Movement crouching	Oguz, Shuaiming
010	Player: Physics (gravity)	Oguz, Shuaiming
011	Player: Input left = D	Thomas, Oguz, Shuaiming
012	Player: Input right = A	Thomas, Oguz, Shuaiming
013	Player: Input Up = W	Thomas, Oguz, Shuaiming
014	Player: Input Down = S	Thomas, Oguz, Shuaiming
015	Player: Input Jump = SPACE	Thomas, Oguz, Shuaiming
016	Player: Input crouch = C	Thomas, Oguz, Shuaiming
017	Player: Climb ropes	Thomas
018	Player: Movement variable jumping height	Thomas
049	Player: Swing from ropes	Thomas
050	Player: Spawning ropes	Thomas
019	Player: Animations for all movements	Thomas
020	Player: Animations for all attacks	Thomas
026	Player: Attack kicking	Thomas
027	Player: Attack punching	Thomas
028	Player: Movement sliding	Thomas
029	Player: Movement crouching	Thomas
030	Player: Attack combos on ground	Thomas
031	Player: Attack combos in air	Thomas
B001	Map: Rope max and min angles adjusted	Thomas
074	Map: Rope physics	Thomas
022	Map: Random treasure spawners	Thomas
048	Map: Random mob spawners	Thomas
B002	Map: Mobs spawning behind tiles	Thomas
023	Map: Lighting	Thomas
051	Map: Ropes	Thomas
052	Map: Vines	Thomas
024	Map: Start and end to map	Thomas
025	Map: Map background music	Thomas
054	Map: Rope and Vine physics	Thomas
055	Map: Rope max and min angles	Thomas
B003	Map Generation: Borders Box Collider Not Resizing	Thomas
B004	Map Generation: Player getting stuck on tiles	Thomas
B005	Map Generation: Ropes spawn on top of each other.	Thomas

4.9 Sprint Review 3

This sprint, the development team completed the coding of player movement and map generation. This allowed the whole group to enter a testing environment made up of 1 cell of the map and the player-controlled character. We were then able to test the player physics to ensure they were smooth and balanced. This was important in fine-tuning the player movement code. The product backlog was

updated so that the terminology was clearer and could be understood by the whole group. The product backlog was also enumerated to allow for better cross referencing. Tom stated that his plan for next week would be to further improve the advanced player inputs, such as variable jump heights.

In Development:

- Player Scripts - Advanced Inputs (thomasapoultney)
 - Added Variable jump height, Longer you hold space the higher you go
 - Added unarmed melee attack combos, Both punching and kicking
 - Added Walking
 - Added sliding and crouch walking
 - Added attack combos for both air and ground

Bugs: None.

Blocked: None.

Done:

- Map Generation - Borders Box Collider Not Resizing (thomasapoultney)
 - Fixed it by setting Box Collider size to be equal to sprite renderer size when they are spawned
- Player Scripts - Swing From Ropes (thomasapoultney)
 - Needs merging with main player controller scripts
 - Player can attach to ropes in map and use them to swing
- Map Design - Ropes Max and Min Angles need Adjusting (thomasapoultney)
 - The rope segments Hinge joints needs their max and min angle limits and the ropes mass to be adjusted until they feel natural
 - fixed by setting an initial value for rope limits and then incrementing it each rope segment spawned, making ropes flex more the longer they are
 - Potential Fix is increasing the limit the further down to rope you go.
- Player - Animations (thomasapoultney)
 - Full list of animations implemented as of 23/11/2021.
 - Added the most Important animations such as jumping, falling, running, idle to the player controller Assets/Player/Scripts/Player_Controller
 - Added all animations to the player, there is a Method in the player controller class called ChangeAnimationState(string newState) which will change the animation playing to the given string.
 - Animations names are given in the form:
 - * Player_Run
 - * Player_Attack_0
 - * Player_Attack_1
 - * Player_Attack_2

- Setting all animations sample rate to 12. Too many animations to control neatly in animator window, better to control in code so we can bug test easier
- Map Design - Mobs Spawning Behind Some Tiles (thomasapoultney)
- Map Generation - Treasure Spawner (thomasapoultney)
 - Added in treasure spawningMethod need to collect more assets for treasure spawning
- Player Scripts - Advanced Movement (thomasapoultney)
 - Added Wall jumping
 - Added Sprinting and another run animation
 - Added Wall sliding
 - Added sliding, crouch walking, crouching, walking and falling animations to player
- Map Generation - Mob Spawner (thomasapoultney)
 - Added Control Scripts to Ghost and Skull
 - Added Archer, Ghost and FireSkull mobs. All animated but need control scripts
 - Added Mob spawning to the map, Added Mob Patrolling script, Added Mob Animations, Currently Only two mobs - Skeleton And Slime
- Lighting - Universal Render Pipeline/ Basic Structure (thomasapoultney)
 - Upgraded project to universal rendering pipeline, upgraded all assets to interact with light
- Map Design - Ropes/ vines (thomasapoultney)
- Map Design - Ladders (thomasapoultney)
- Player Scripts - Spawn Ropes (thomasapoultney)
 - Added players ability to spawn ropes above them if there is nothing blocking the path, added a delay between rope spawns.
 - Removes ropes if there are too many spawned by the player
- Map Generation - Player gets stuck on tiles sometimes (thomasapoultney)
 - Fix - Set all surface tiles box colliders collision detection to continuous, Can be done the same way as detecting surface tiles that mobs can spawn on using raycasts
- Map Design - Rope Physics (thomasapoultney)
 - Added Hinges to rope links to allow physics
- Map Generation - Rope Spawn in on Top of Each other Sometimes (thomasapoultney)
 - Also doesn't work if the block the rope is placed on is not at the centre of the unity grid tile. Adjusted room layout to fix issue e.g. (0.5,0.5)
 - Fixed by disabling the box collider on the rope fragments until they have finished spawning
- Player Scripts - Climb Ropes (thomasapoultney)
 - Added Lerp to climbing ropes to make animation smoother
- Map Generation - Start and End (thomasapoultney)

- Added spawnDoor method to the worldSpawner script which spawns a door randomly in a room where there is space.
- Moving between levels is blocked until player movement is completed
- Player Scripts - Basic Inputs (shuaimingzhu, oguztecirlioglu, thomasapoultney)
- Player Scripts - Basic Movement (shuaimingzhu, oguztecirlioglu)
- Lighting - Insanity Demo (thomasapoultney)
- Sound Design - Background Music (thomasapoultney)

4.10 Exception handling 3

Unexpected Situation	Status	Actions to resolve
Github Merging issues: In this week we restructured the entirety of the project to closer reassembly the structure of the product backlog. When we merged this project we still had active branches. When other developers tried to merge the active branches this caused an unresolvable number of merge conflicts.	Resolved: 18/11/2021	We entered the git reset –soft HEAD 1 to remove the last commit we did. The developers then merged their active branches. Once there was no active branches we restructured the project.

4.11 Weekly Sprint Retrospective 3

This sprint the development team worked through their targets quickly and efficiently. Motivation and work ethic was high due to the ease of tasks and the clear progress made on completion of each task. Each developer fully understood their tasks and how completion of these tasks would benefit the development of the game. These factors allowed the development team to complete all tasks ahead of schedule, allowing us to spend the rest of the sprint performing tests on the testing environment developed through our map generation and player physics models. During the early stages of development communication between the development team and documentation team was below the required level. This meant that although development was proceeding at a good pace, this was at time unbeknown by the documentation team. This was corrected during team meetings, but during the sprint the development team should take more time to explain their progress to the documentation team in terms which they understand.

The documentation team managed to act on advice from the customer and successfully develop the product backlog to allow for easier cross referencing. In theory this was well executed, tasks were enumerated and ordered by importance. However, the terminology and format of the backlog still left much to be desired. This reduced visibility for the group regarding which tasks were to be completed next. The group would often just discuss which tasks to complete next instead of refereeing to the product backlog, making the development of the product backlog pointless. In future the whole group should take advantage of the product backlog, this would allow us to streamline both the development and documentation processes and ensure that work is not repeated by different group members.

5 Sprint 4: 20/11/2021 - 28/11/2021

5.1 Roles

- Product Owner - Sam Higgs
- Scrum Master - Thomas Poultney

5.2 Group Meeting 4

- Date - 20/11/2021
- Location – Teams
- Present – Sam Higgs, Oguz Tecirlioglu, Diparati Sen, Tom Poultney
- Absent – Jack McGee, Phillip Read, Shuaiming Zhu, Ruoxi Li

Agenda Covered – Working map design with mobs and player movement. A base art style was implemented to the test environment. Product backlog was addressed and updated. Mechanics for the timing effect of the light were discussed. Thomas Poultney demonstrated the new project structure and explained the use of the Doxygen tool.

Outcomes:

- Working game and test environment
- Sprites for mobs
- Implementation of a base art style

Added to the Agenda:

- Bugs to fix
- Keep developing player movement and game physics
- Present to the customer

Meeting Duration – 40mins

5.3 Customer Meeting 4

- Date - 24/11/2021
- Location – CB 5.13
- Present – Tom Poultney, Diparati Sen, Jack McGee, Sam Higgs, Ruoxi Li
- Absent – Oguz Tecirlioglu, Shuaiming Zhu

Agenda Covered – Customer was presented with the working game and areas which still need implementing were discussed. Issues surrounding the interaction between lava flow and destructible blocks were discussed. The main menu was discussed with the customer and the customer was asked which options they would like to have regarding changing the game (difficulty, theme etc). Possibilities of a tutorial were discussed. Methods of increasing difficulty over time were presented. Testing process was discussed and explained by the customer.

Outcomes:

- Focus on finishing the main functionality of the game
- Ensure game works efficiently – remove lava flow
- Customer would like a tutorial and the option to change the starting difficulty
- Testing must be approved by ethics board
- Testing process starts in the next sprint

Added to the Agenda:

- Develop a tutorial level
- Develop difficulty changes in the options menu
- Increase difficulty per level passed
- Write up or develop a formal testing process
- Get testing process approved

Duration – 15mins

5.4 Product Backlog 4

ID	Description	Tom	Gary	Diparati	Oguz	Priority	Date Completed
018	Player: Movement variable jumping height	30 mins	30mins	3 hrs	2 hrs	2	21/11/2021
021	Player: Animations for all collisions	2hrs	5hrs	6 hrs	5 hrs	2	
026	Player: Attack kicking	1 hr	2hrs	3 hrs	2 hrs	3	21/11/2021
027	Player: Attack punching	1 hr	2hrs	3 hrs	2 hrs	3	21/11/2021
028	Player: Movement sliding	1 hr	3hrs	3 hrs	2 hrs	3	21/11/2021
029	Player: Movement crouching	1 hr	3hrs	3 hrs	2 hrs	3	21/11/2021
030	Player: Attack combos on ground	1hr	2hrs	5 hrs	4 hrs	3	21/11/2021
031	Player: Attack combos in air	1h	3hrs	5 hrs	4 hrs	3	21/11/2021
032	NPC's: Ghost npc's Design	4 hrs	5hrs	10 hrs	10 hrs	3	25/11/2021
033	NPC's: Skull npc's Design	4 hrs	3hrs	10 hrs	10 hrs	3	25/11/2021
034	NPC's: Archer npc's Design	4 hrs	4hrs	10 hrs	10 hrs	3	
035	NPC's: Fire skull npc's Design	4 hrs	5hrs	10 hrs	10 hrs	3	25/11/2021
036	NPC's: Slime npc's Design	4 hrs	4hrs	10 hrs	10 hrs	3	
037	NPC's: Ghost npc's scripts	4 hrs	5hrs	10 hrs	10 hrs	3	25/11/2021
038	NPC's: Skull npc's scripts	4 hrs	6hrs	10 hrs	10 hrs	3	25/11/2021
039	NPC's: Archer npc's scripts	4 hrs	4hrs	10 hrs	10 hrs	3	
040	NPC's: Fire skull npc's scripts	4 hrs	5hrs	10 hrs	10 hrs	3	25/11/2021
041	NPC's: Slime npc's scripts	2 hrs	2hrs	10 hrs	10 hrs	3	
042	NPC's: Ghost npc's animations	2 hrs	3hrs	10 hrs	10 hrs	3	25/11/2021
043	NPC's: Skull npc's animations	2 hrs	5hrs	10 hrs	10 hrs	3	25/11/2021
044	NPC's: Archer npc's animations	2 hrs	5hrs	10 hrs	10 hrs	3	
045	NPC's: Fire skull npc's animations	2 hrs	5hrs	10 hrs	10 hrs	3	25/11/2021
046	NPC's: Slime npc's animations	2 hrs	3hrs	10 hrs	10 hrs	3	
047	NPC's: Mob patrolling script	4 hrs	2hrs	10 hrs	10 hrs	3	
053	Map: ladders	2 hrs	3hrs	6 hrs	5 hrs	4	
056	Map: Wall torches	1 hr	2hrs	5 hrs	3 hrs	4	21/11/2021
057	NPC's: Mob collision	2 hrs	4hrs	6 hrs	4 hrs	4	
058	Player: Wall Jumping	2 hrs	3hrs	5 hrs	4 hrs	5	21/11/2021
059	Player: Blood animation on collision	1 hr	2hrs	2 hrs	1 hrs	5	21/11/2021
060	Player: Taking damage on collision	1 hr	2hrs	2 hrs	1 hrs	5	

061	Player: Spawn torch = F	30 mins	1hrs	2 hrs	1 hrs	5	21/11/2021
062	Player: Bounce animation when landing	10 mins	1hrs	1 hr	20 mins	5	
063	NPC's: Blacksmith Design	3 hrs	6hrs	10 hrs	10 hrs	5	
064	NPC's: Blacksmith Animations	2 hrs	5hrs	10 hrs	10 hrs	5	
065	User Interface: Game behind main menu	8 hrs	4hrs	2 hrs	10 hrs	6	21/11/2021
066	User Interface: Health bar	1 hrs	2hrs	2 hrs	4 hrs	6	
067	User Interface: Insanity bar	1 hrs	2hrs	2 hrs	4 hrs	6	
068	User Interface: Treasure counter	1 hrs	3hrs	2 hrs	4 hrs	6	
069	Loading Screen: Start menu	5 hrs	4hrs	3 hrs	10 hrs	6	
070	Loading Screen: leader board menu	20 hrs	5hrs	3 hrs	10 hrs	6	
071	Loading Screen: Winner screen	5 hrs	6hrs	5 hrs	10 hrs	6	
072	Loading Screen: Death screen	5 hrs	6hrs	6 hrs	5 hrs	6	
073	Loading Screen: Settings menu	3 hrs	5hrs	3 hrs	10 hrs	6	
076	Player: Wall Grabbing	3 hrs	6hrs	3 hrs	10 hrs	6	21/11/2021
077	Player: Bouncing when landing	3 hrs	4hrs	3 hrs	10 hrs	6	21/11/2021
078	Player: collision stun/knock back	3 hrs	6hrs	3 hrs	10 hrs	6	21/11/2021
079	Player: Add iFrames to character	3 hrs	5hrs	3 hrs	10 hrs	6	21/11/2021
080	Player: player moonwalks when attack and moving	3 hrs	4hrs	3 hrs	10 hrs	6	21/11/2021
081	Map: arrow traps	3 hrs	7hrs	3 hrs	10 hrs	6	
082	NPC's: Mob collisions	3 hrs	10hrs	3 hrs	10 hrs	6	21/11/2021
083	NPC's: Skeleton npc's Design	3 hrs	6hrs	3 hrs	10 hrs	6	21/11/2021
084	NPC's: Skeleton npc's scripts	3 hrs	8hrs	3 hrs	10 hrs	6	21/11/2021
085	NPC's: Skeleton npc's animation	3 hrs	8hrs	3 hrs	10 hrs	6	21/11/2021
086	Doc: Doxygen documentation	3 hrs	6hrs	3 hrs	10 hrs	6	21/11/2021
B006	Map: Mobs spawning on player	3 hrs	5hrs	3 hrs	10 hrs	6	21/11/2021
B007	NPC's: Ghost Chasing Player breaks	3 hrs	10hrs	3 hrs	10 hrs	6	21/11/2021

5.5 User Stories 4

User Story 015: As a player,
I want map has ladders
So that,
I can climb and slide from them to complete my task.
Acceptance Criteria;
Players can climb and slide from ladders.

User Story 016: As a player,
I want to have to encounter multiple mobs,
So that,
Each time I play the game, I have a varied experience.
Acceptance Criteria;
The game needs to present the player with multiple mobs to battle.

User Story 017: As a player,
I want to interact with elements like wall torches,
So that,
the torches act as the source of light in the game.
Acceptance Criteria;
The game needs to have elements like wall torches which act as the source of light in the player area.

User Story 018: As a player,
I want to be able to perform advanced movements like wall jump, sliding and crouching
So that,
linear movement does not become monotonous.
Acceptance Criteria;
The game needs to have advanced movement features like wall jump, sliding and crouching.

5.6 Use Cases Sprint 4

UC 011: Multiple mobs
Scope: N/A
Level: User goal
Primary Actors: Player
Description: Player encounters multiple mobs.
Dependencies: C 014: Player Starts the game.
UC 003: Room design.
UC 004: Rooms join together.
UC 005: Room connections.
Assumptions: None.
Preconditions: Player is in the game.
Main Flow: Right after the player starts game, the map should be created with multiple mobs for the player to encounter.
Subflows: None.
Alternative Flows: None.
Post Conditions: The player then battle the multiple mobs as encountered.
Frequency of Occurrence: Every time the game Starts.

Originating User Story: US 016

UC 012: Wall torches

Scope: N/A

Level: User goal

Primary Actors: Player

Description: Player interacts with wall torches.

Dependencies: C 014: Player Starts the game.

UC 003: Room design.

UC 004: Rooms join together.

UC 005: Room connections.

Assumptions: None.

Preconditions: Player is in the game.

Main Flow: Right after the player starts game, the map should be created with wall torches in place.

Subflows: None.

Alternative Flows: None.

Post Conditions: The player continues playing the game in the presence of wall torches.

Frequency of Occurrence: Every time the game Starts.

Originating User Story: US 017

UC 013: Advanced movements

Scope: N/A

Level: User goal

Primary Actors: Player

Description: Player can perform advanced movements like wall jump, sliding and crouching.

Dependencies: C 014: Player Starts the game.

UC 003: Room design.

UC 004: Rooms join together.

UC 005: Room connections.

Assumptions: None.

Preconditions: Player is in the game.

Main Flow: Right after the player starts game, the map should be created and be made up of rooms and the player should be able to move around performing the advanced movements.

Subflows: None.

Alternative Flows: None.

Post Conditions: The player plays the game utilizing the advanced moves.

Frequency of Occurrence: Every time the game Starts.

Originating User Story: US 018

5.7 CRC Cards Sprint 4

Overview of CRC Card Changes: Advanced player movement user case has been added this week. We therefore need to further split down the player controller class down to account for the added responsibilities. A ladder controller class will be added to control players interaction with ladders. We also intend to add several new mob types to the game so we will split the mob controller class into two categories: Flying Mob controller, Ground patrol Mob controller. Finally, The player can place a torch in the world so we will reflect this behaviour in a new CRC card called spawn torch. We will then add this as a collaboration from the player controller to the spawn torch card.

Class Name: Map Spawner	Version: 3
Description: Spawns all the rooms within a map, Storing the position of each object and populating the world	Associated Use Cases: UC001, UC002, UC006, UC007, UC011
Responsibilities	Collaborators
Spawn World Borders Spawn Critical Path Rooms Spawn Random Mob Spawn treasure Track Positions of objects	Room Connections Room Ground Mob, Flying Mob Treasure

Class Name: Room Spawner	Version: 2
Description: Spawn a single room and its contents	Associated Use Cases: UC002, UC003
Responsibilities	Collaborators
Spawn block Spawn spike Spawn Rope Spawn Lava Spawn Ladder	Block Spike Rope Lava Ladder

Class Name: Mob Spawner	Version: 2
Description: Responsible for spawning a single ground or flying mob	Associated Use Cases: UC006, UC011
Responsibilities	Collaborators
Spawn a random ground mob	Map Spawner

Class Name: Player Ladder controller	Version: 1
Description: Responsible for controlling players interaction with ladders	Associated Use Cases: UC013 ,
Responsibilities	Collaborators
Climb Ladder Slide Down Ladder Die Take damage	

Class Name: Player Controller	Version: 3
Description: Handles player basic movement	Associated Use Cases: UC013
Responsibilities	Collaborators
Move Jump Attack tracks health Place Torch	Mob Controller Spawn Torch

Class Name: Spawn Torch	Version: 1
Description: Spawns a single torch	Associated Use Cases: UC013
Responsibilities	Collaborators
Spawn a torch Apply Flickering effect	Lighting controller

Class Name: Air Mob controller	Version: 1
Description: Responsible for controlling a flying mobs behaviour	Associated Use Cases: UC006 , 011
Responsibilities	Collaborators
Mob hovers idle in air Mob hunts player when close Die Take damage	Player Controller

5.8 Sprint Backlog 4

ID	Description	Developer
058	Player: Wall Jumping	Thomas
076	Player: Wall Grabbing	Thomas
059	Player: Blood animation on collision	Thomas
061	Player: Spawn torch = F	Thomas
018	Player: Movement variable jumping height	Thomas
026	Player: Attack kicking	Thomas
027	Player: Attack punching	Thomas
028	Player: Movement sliding	Thomas
029	Player: Movement crouching	Thomas
030	Player: Attack combos on ground	Thomas
031	Player: Attack combos in air	Thomas
077	Player: Bouncing when landing	Thomas
078	Player: collision stun/knock back	Thomas
079	Player: Add iFrames to character	Thomas
080	Player: player moonwalks when attack and moving	Thomas
065	User Interface: Game behind main menu	Thomas, Diparati
056	Map: Wall torches	Thomas
081	Map: arrow traps	Oguz
B006	Map: Mobs spawning on player	Oguz
082	NPC's: Mob collisions	Thomas
063	NPC's: Blacksmith Design	Thomas
064	NPC's: Blacksmith Animations	Thomas
032	NPC's: Ghost npc's Design	Thomas
033	NPC's: Skull npc's Design	Thomas
036	NPC's: Slime npc's Design	Thomas
083	NPC's: Skeleton npc's Design	Thomas
037	NPC's: Ghost npc's scripts	Thomas
038	NPC's: Skull npc's scripts	Thomas
041	NPC's: Slime npc's scripts	Thomas
084	NPC's: Skeleton npc's scripts	Thomas
042	NPC's: Ghost npc's animation	Thomas
043	NPC's: Skull npc's animation	Thomas
046	NPC's: Slime npc's animation	Thomas
085	NPC's: Skeleton npc's animation	Thomas
B007	NPC's: Ghost Chasing Player breaks	Thomas
086	Doc: Doxygen documentation	Thomas

5.9 Exception handling 4

Unexpected Situation	Status	Actions to resolve
An unforeseen situation for one of the group members that caused some disruptions in the completion of tasks occurred	Reduced the impact	We communicated effectively through Microsoft Teams, and the sprint and product backlogs were updated accordingly

5.10 Sprint Review 4

This week, the development team were able to implement an art style which was similar to the preferences expressed by the customer. A compromise had to be reached by the high realism art style requested and what we were able to implement. More advanced movement features were added to the player movement such as wall jumping. Mob designs were made and implemented into the map generation. The user interface was developed and implemented. The documentation team began drafting some tests to perform to track the game's progress and find unforeseen bugs.

In Development:

- Map Design - Arrow Traps (traps) (oguztecirlioglu)
 - Arrow traps are done, but we need to add the player damaging effect to them.
 - Also the arrow trap spawner tile prefab should be added to each rooms design.

Bugs: None.

Blocked: None.

Done:

- NPC's Scripts - Mob Collisions - Mobs attack player after they are dead (thomasapoultney)
- Player Scripts - Wall Jump (thomasapoultney)
 - Added Wall Grabbing, Can press vertical input down to slide down wall faster
 - Values need tweaking, but functionality is there.
- User Interface - Game Spawning Behind Main Menu (thomasapoultney, diparatisen)
- NPC's - Blacksmith Animations (thomasapoultney)
- Map Design - Wall Torches (thomasapoultney)
- User Interface/Player Scripts - Public Static Class to track variables (thomasapoultney)
- Player Scripts - Collisions (thomasapoultney)
 - Added Blood particle effect when taking damage
 - added a method that causes players to take damage and allows them to be stun locked or knocked back.
 - Added a script in assets/Player/Scripts/Player_Collisions to manage all player collisions
- Map - Design - Mobs Spawning On Player (oguztecirlioglu)
- Player Scripts - Advanced Inputs (thomasapoultney)
 - Player can spawn torches by pressing 'F'
 - Added Variable jump height, Longer you hold space the higher you go
 - Added unarmed melee attack combos, Both punching and kicking
 - Added Walking
 - Added sliding and crouch walking
 - Added attack combos for both air and ground

- NPC's Scripts - Mob Collisions (thomasapoultney)
 - Added Ghost Collisions
 - Added in exploding skull collisions
 - Added Slime Collisions
 - Added Mob collisions to Skeleton
- Player Scripts - Player Bouncing when Landing (thomasapoultney)
- Mob Design - Ghost Chasing Player breaks if you move through it (thomasapoultney)
- Player Scripts - Collisions - Damage from attack and stun/knock back effect applying at start of animation (thomasapoultney)
- Player Scripts - Add iFrames to charecter (thomasapoultney)
- Player Scripts - Player moonwalks when attacking while moving. (thomasapoultney)
 - Set player velocity to zero if we are grounded and attacking.
- Doxygen - Generate the Doxygen documentation we have produce thus far. (thomasapoultney)

5.11 Weekly Sprint Retrospective 4

During this sprint the development team, again, worked quickly and efficiently through their tasks. This meant that advanced player physics and mobs could be introduced to the game. One of the positives of this sprint is that communication between the developers and documentation team was clear, concise, and consistent. With every task that the development team completed, they provided an update to the documentation team, allowing them to write down the project progress and update the product backlog accordingly. When this sprint's tasks were completed early, the development team consulted with the documentation team to identify areas to develop next instead of just developing random mobs or traps. This ensured that visibility of the project was upheld throughout the sprint. The development team were also quick to identify bugs in the game environment and managed to log, find solutions and present these to the whole group. This allowed the group and the customer to reach a consensus regarding the solution to these bugs. Overall, the development team performed exceptionally this sprint.

The progress of the documentation team was slow this week. Other commitments prevented the team from keeping up with various group meetings and product backlog updates. However, during the planning of this project we anticipated this may happen so ensured that all meetings were recorded with transcriptions so that minutes and progress can be tracked retrospectively. The team all agreed that the documentation team will commit more time to catching up on tasks whilst also completing next weeks tasks on time.

5.12 Doxygen

In this week we generated all the Doxygen documentation we have produced so far. The details of the process we followed to achieve this and why it is important is included in-depth in the Doxygen section of the maintenance documentation. Our Doxygen includes documenting all public and private class variables and methods. It is produced as an .html file.

EndlessDelve 1.0
A Dungeon Game Created by Pink Team

Main Page	Packages	Classes	Files
<ul style="list-style-type: none"> EndlessDelve <ul style="list-style-type: none"> Packages Classes <ul style="list-style-type: none"> Class List Class Index Class Hierarchy Class Members Files 		<ul style="list-style-type: none"> GhostController HealthBarScript HighScores InsanityBar InsanityDemonstration MapVariables Player_Collisions Player_Controller Player_Variables PlayerRopeController PlayerScore RandomAnimationDelay RoomType RopeSegment SkeletonMobController SkullController SlimeController spawnArrowtrap SpawnLadder SpawnLava SpawnMob spawnRope SpawnSpike spawnSwingingTrap SpawnTile startgame SwingingTrapSegment TemplateChooser TreasureValue UpdateText WorldSpawner 	<ul style="list-style-type: none"> Class responsible for handling the behaviour of the ghost mob A script responsible for updating the health bar HUD element a class responsible for downloading the leader board values from our database A class responsible for controlling the insanity bar HUD element Reduces the light produced by the player based upon their current insanity A static Class that can be used to communicate map variables with other objects and classes Class Responsible for handling all player collisions, including interactions with world object such as doors or ladders This class controls the movement, animations and combat for the player charecter A static class containing all variables associated with the player class responsible for handling players interactions with ropes and calculating rope physics A struct to store the leaderboard variables, It includes the name and score of each player starts the animation of the attached object at a random time between 0 and 1s A script which is attached to a room to specify its type A class which stores information about the attached rope segment Class responsible for handling the behaviour of the skeleton mob Class responsible for controlling the flaming skull mob Class responsible for handling the behaviour of the slime mob spawns an arrow trap or a basic building block according to a random chance Responsible for spawning a set amount of a random ladder prefab Spawns a random lava object and sets its layer, and adds a particle system Spawns a random mob from a given list of mob prefabs Spawns a given number of random rope prefabs and sets up their physics constraints, Spawns a random spike prefabs and sets up it layers A script responsible for spawning a length of spike ball traps and setting its physics constraints Spawns a random block or ore prefab a class responsible for controlling the UI of the main menu and loading the scene of the main game A class which spawns a set length of a Swinging spike trap A class that is responsible for selecting a random room to spawn from a list A simple script which assigns the amount of score the attached object is worth A simple script that updates the text of the attached text element to a given value A Script that is responsible for generating the world and spawning its contents based upon the current difficulty

Figure 1: An example of our Full Doxygen Class listings, Used to improve transparency between developers.

Player_Controller Class Reference

This class controls the movement, animations and combat for the player character. [More...](#)

Inheritance diagram for Player_Controller:

```

graph BT
    MonoBehaviour --|> Player_Controller
  
```

Public Member Functions

- `void FixedUpdate ()`
Monobehaviour that is called at a fixed timestamp. In this behaviour we control which animation the player should be executing based on inputs and player states. [More...](#)
- `void setVelocity ()`
updates the velocity of the player based on user inputs and the current state of the player. [More...](#)
- `void Jump ()`
sets the player to jump or wall jump if the player is on the wall and changes animation state to jumping. [More...](#)

Public Attributes

- `string currentAnimationState`
The Current Animation playing. [More...](#)
- `Animator animator`
The animator that is attached to the player object. [More...](#)
- `bool facingLeft`
Whether the player is facing left. [More...](#)
- `bool isWallSliding`
Whether the player is wall sliding. [More...](#)
- `bool isWallGrabbing`
Whether the player is grabbing onto a wall. [More...](#)
- `bool canWallGrab = true`
Whether the player meets the criteria to grab onto a wall. [More...](#)
- `float timeSinceCannotWallGrab = 0`
Reset timer for grabbing a wall. [More...](#)
- `float wallGrabResetTimer = 0.1f`

Figure 2: A more detailed view of the Doxygen produced for our *Player_controller* class.

6 Sprint 5: 29/11/2021 - 05/12/2021

6.1 Roles

- Product Owner - Sam Higgs
- Scrum Master - Thomas Poultney and Sam Higgs

6.2 Group Meeting 5

- Date – 29/11/2021
- Location – Teams
- Present – Diparati Sen, Oguz Tecirlioglu, Phillip Reed, Jack McGee, Tom Poultney, Ruoxi Li, Sam Higgs
- Absent - Shuaiming Zhu

Agenda Covered – Discussed the competition element of the game and how the ‘score’ accumulates. Presented the working start screen to the group. Ideas for the leaderboard were discussed. Difficulty progression was discussed. Testing questionnaire was discussed and presented. Additionally, a table that formally tracks the manual tests that game testers carry out will be developed this week. This is available further down the documentation. Finally, the timeline from now until submission was discussed.

Outcomes:

- Use a leader board for competition

- Score will be calculated using time, levels completed, and treasure collected
- Difficulty increases through more difficult rooms, increased traps, and increased mobs.
- Exponential difficulty increase. Selected starting difficulty starts the player at different points on this scale.

Added to the Agenda:

- Format the questionnaire
- Adjust the questionnaire to be more specific to our game
- Write up final timeline from now until submission
- Discuss the logistics of distributing the questionnaires

Duration – 45mins

6.3 Customer Meeting 5

No customer meeting due to University and College Union (UCU) strike.

6.4 Product Backlog 5

ID	Description	Tom	Gary	Diparati	Oguz	Priority	Date Completed
021	Player: Animations for all collisions	2hrs	4hrs	6 hrs	5 hrs	2	
034	NPC's: Archer npc's Design	4 hrs	4hrs	10 hrs	10 hrs	3	
036	NPC's: Slime npc's Design	4 hrs	40hrs	10 hrs	10 hrs	3	
039	NPC's: Archer npc's scripts	4 hrs	4hrs	10 hrs	10 hrs	3	
041	NPC's: Slime npc's scripts	2 hrs	2hrs	10 hrs	10 hrs	3	
044	NPC's: Archer npc's animations	2 hrs	7hrs	10 hrs	10 hrs	3	
046	NPC's: Slime npc's animations	2 hrs	5hrs	10 hrs	10 hrs	3	
047	NPC's: Mob patrolling script	4 hrs	5hrs	10 hrs	10 hrs	3	
053	Map: ladders	2 hrs	3hrs	6 hrs	5 hrs	4	
057	NPC's: Mob collision	2 hrs	3hrs	6 hrs	4 hrs	4	
060	Player: Taking damage on collision	1 hr	2hrs	2 hrs	1 hrs	5	
062	Player: Bounce animation when landing	10 mins	30mins	1 hr	20 mins	5	
063	NPC's: Blacksmith Design	3 hrs	10hrs	10 hrs	10 hrs	5	
064	NPC's: Blacksmith Animations	2 hrs	3hrs	10 hrs	10 hrs	5	
066	User Interface: Health bar	1 hrs	2hrs	2 hrs	4 hrs	6	
067	User Interface: Insanity bar	1 hrs	2hrs	2 hrs	4 hrs	6	
068	User Interface: Treasure counter	1 hrs	2hrs	2 hrs	4 hrs	6	
069	Loading Screen: Start menu	5 hrs	4hrs	3 hrs	10 hrs	6	
070	Loading Screen: leader board menu	20 hrs	4hrs	3 hrs	10 hrs	6	
071	Loading Screen: Winner screen	5 hrs	10hrs	5 hrs	10 hrs	6	
072	Loading Screen: Death screen	5 hrs	7hrs	6 hrs	5 hrs	6	
073	Loading Screen: Settings menu	3 hrs	8hrs	3 hrs	10 hrs	6	
081	Map: arrow traps	6 hrs	7hrs	3 hrs	5 hrs	4	01/12/2021
087	Player: Arrow Trap damage	2 hrs	3hrs	3 hrs	2 hrs	4	01/12/2021
B008	Map: Arrow Trap spawning bug	2 hrs	4hrs	3 hrs	2 hrs	4	01/12/2021

6.5 User Stories 5

User Story 019: As a player,
I want to see a start screen to enter the game when I click “Start” button,
So that,
I can enter the game when I am ready.
Acceptance Criteria;
Player enters the game by click “Start” button.

User Story 020: As a player,
I want to have to see a leaderboard,
So that,
Each time I end the game, I can verify if my score comes up on the highest scorers’ list.
Acceptance Criteria;
The game needs to feature a leaderboard containing the names and scores of the top players.

6.6 Use Cases Sprint 5

UC 014: Player starts the game
Scope: N/A
Level: user goal
Primary Actors: Player
Description: Players can click start button from main menu to enter the game and see the map and all requirements after entering the game.
Assumptions: The start button is accessible, and the main menu exists.
Preconditions: Player is in the main menu.
Dependencies: N/A
Frequency of occurrence: Every time players click start button. Originating User Story: US 019

UC 015: Leaderboard
Scope: N/A
Level: User goal
Primary Actors: Player
Description: Player’s score comes up in a leaderboard.
Dependencies: C 014: Player Starts the game.
UC 003: Room design.
UC 004: Rooms join together.
UC 005: Room connections.
Assumptions: The online leaderboard is accessible.
Preconditions: Player has ended the game.
Main Flow: Right after the player ends the game, the player gets an option to view the leaderboard.
Subflows: None.
Alternative Flows: The player does not view the leaderboard. This can occurs if the users connection to the internet is weak/non-existent.
Post Conditions: The player sees their name and score in the leaderboard, if it is one of the top scores.
Frequency of Occurrence: Every time the game ends.
Originating User Story: US 020

6.7 CRC Cards Sprint 5

Overview of CRC Card Changes: In this section we add the CRC Card for the leaderboard. We will also add a player variables class to reduce the responsibilities of the player controller class.

Class Name: Leaderboard	Version: 3
Description: Retrieves and uploads players high-scores from the website	Associated Use Cases: UC015
Responsibilities	Collaborators
Upload current score to database Retrieve top 10 highscores from database	Player Variables Database Website

Class Name: Player Controller	Version: 4
Description: Handles player basic movement	Associated Use Cases: UC013
Responsibilities	Collaborators
Move Jump Attack Place Torch	Mob Controller Spawn Torch

Class Name: Player Variables	Version: 1
Description: Handles player basic movement	Associated Use Cases: UC013
Responsibilities	Collaborators
stores players current health Stores the amount of treasure collected	player Controller

6.8 Formal Game Testing For Gameplay Testers

The table below was given to all manual gameplay testers, to guide them throughout the testing process. Each functionality seen will be tested from Sprint 5 onwards, at least weekly. The advantage of manual tests is that complex gameplay interactions can be quickly tested, and for some of these interactions writing unit tests can be very lengthy and certainly not worth the time commitment.

Test items	Originating User Story	Originating Use Case	Scenario Description	Result
Main Menu	026	UC 019	The user inputs their nickname.	Pass
	027	UC 020	The user checks player controls by selecting the player control button	Pass
	027	UC 020	User returns back to the main menu by clicking the “BACK” button	Bug*
	019	UC 014	Player enters the game by click “Start” button.	Pass
Game control	007	UC 000	The Player could move the character by pressing W/S/A/D. Where W moves up, S moves down, A moves left, and D moves right.	Pass
	018	UC 013	Player jumps by click ”Space”	Pass
	018	UC 013	Player attacks by pressing the left mouse button down.	Pass
	018	UC 013	Player interacts with doors by pressing E	Pass
	018	UC 013	Player walks by pressing ctrl + A/D (ctrl and A or D)	Pass
	018	UC 013	Player sprints by pressing shift + A/D (shift and A or D)	Pass
	012	UC 009	Player can swing on ropes by pressing A or D	Pass
	013	UC 010	Player could place torch by pressing F	Pass
Mob	016	UC 011	Slime mob could attack the player if the player is near it	Pass
	016	UC 011	The player takes damage from all mob attacks	Pass
	016	UC 011	The Ghost mob tracks and follows the player through walls if in vicinity.	Pass
	016	UC 011	The Ghost mob lights up if it finds the player.	Pass
	016	UC 011	Red skull mob self-destructs and damages the player after it touches the player.	Pass
	016	UC 011	The Skeleton mob can attack the player, knocking the player back on impact.	Pass
	016	UC 011	All foot mobs (Slime and Skeleton) patrol the map.	Pass
	016	UC 011	The player’s attacks could hit mobs and even kill mobs	Pass
Map element	007	UC 000	The player can run on normal, flat ground.	Pass
	018	UC 013	The player can grab walls, and jump from walls as well.	Pass
	010	UC 007	The player can pick up Emeralds, adding to their total score on the top right.	Pass
	023	UC 016	The Spike Trap and Swinging Spike Traps damage the player	Pass
	012	UC 009	Ropes could be climbed up and down.	Pass
	014	UC 010	The Lava is an instant death if the player falls into it.	Pass
	012	UC 009	Player can swing from rope to rope, like Tarzan, and catch the next one.	Pass
	017	UC 012	Torces placed down provide light.	Pass
Leaderboard	020	UC 015	The player could check the leaderboard after the game finishes, and everything loads correctly.	Pass
* If the user enters their name, and views the controls, the name disappears.				

6.9 Sprint Backlog 5

6.10 Sprint Review 5

This sprint, the Arrow Traps were further improved and bugs related to it were fixed. Importantly, since not a lot of development was carried out this week, it was a good opportunity to discuss the tests needed to be done with the team, and formalise the testing procedure by creating a table. This table will be used by the dedicated testers, and serves as a minimum benchmark for the testing that has to be carried weekly after updates to the game are carried. The game has matured significantly since the start, and the team is beginning to search for ways to further diversify the games assets by looking into new traps, and figuring out how the end game mechanics will work (what happens when the player finds the current mazes door, dies, the leaderboard, etc.)

In Development:

- Player Scripts – Add Arrow Trap damage, which references other scripts developed by Tom to add damage and knockback effects on player. (Oguz Tecirlioglu)

Bugs:

- Arrow traps spawn in all rooms at the same time, meaning they either spawn or don't spawn in all rooms. Fixed by fiddling with the tile spawner code. [Fixed] (Oguz Tecirlioglu)

Blocked: None.

Done:

- Map Design – Arrow Traps added to the game, shooting arrows from left to right. (Oguz Tecirlioglu)
- Map Design – Arrow Trap tiles added to room prefabs, giving them a 50% chance to randomly spawn in selected rooms. (Oguz Tecirlioglu)

Documentation:

6.11 Exception handling 5

Unexpected Situation	Status	Actions to resolve
Bug in the archer mob that was unable to be resolved. During the design our game crashed whenever the archer mob attacked and cannot find where the bug cause it.	Fixed	Extensive Pair programming between the developer who originally created the mob behaviour and the developer experiencing the issue is required to resolve this issue. If that fails we will redesign mob behaviour from the ground up.

6.12 Weekly Sprint Retrospective 5

The development team experienced similar issues to those experienced with the documentation team last week. A culmination of other deadlines prevented much progress from being made in terms of game development this week. Fortunately, this coincided with union action which prevented customer meetings from taking place. Because of this, the group had planned for this week to be set aside of other coursework. Therefore, through effective planning processes, we were able to inadvertently account for the union action. During group meetings however, some group members were not as transparent with their ability to complete tasks in the product backlog along with completing other coursework. This left the rest of the group waiting on tasks to be completed which ultimately couldn't be completed. Remaining transparent and realistic about what can be completed during the sprint would prevent other group members from having to wait for work which is never completed.

The documentation team were able to effectively utilise this week to catch up on previous missed tasks. Tasks were assigned during the group meeting and were finished to a high quality within a couple of days. This meant that the documentation team could go back through previous work and make improvements where necessary. Overall, the documentation team worked well to compensate for previous sprints – the lack of development progress was planned for and allowed the documentation team to ensure all tasks on their side were up to date. .

7 Sprint 6: 06/12/2021 - 12/12/2021

7.1 Roles

- Product Owner - Sam Higgs
- Scrum Master - Thomas Poultney and Sam Higgs

7.2 Group Meeting 6

- Date – 06/12/2021
- Location – Teams
- Present – Diparati Sen, Tom Poultney, Oguz Tecirlioglu, Jack McGee, Sam Higgs, Ruoxi Li, Shuaiming Zhu
- Absent – None

Agenda Covered – The progress of creating the leader board was discussed. Timeline was updated to account for lecturer strikes and technology issues with one of our developers. Global health interaction with new traps was discussed. Jobs were discussed for each person over the following week.

Outcomes:

- Additional traps to be added
- Existing arrow trap needs to be improved

Added to the Agenda:

- Code for different types of traps that interact with global health in different ways

Duration – 15mins

7.3 Customer Meeting 6

- Date – 08/12/2021
- Location - CB 5.13
- Present – Sam Higgs, Phillip Reed, Diparati Sen, Shuaiming Zhu
- Absent – Jack McGee, Tom Poultney, Ruoxi Li, Oguz Tecirlioglu

Agenda Covered – Demonstrated the new UI and the new mechanics within the game (ie new traps, movement options, insanity mechanic and game menu). Discussed introducing the leaderboard. Presented the newly updated documentation system. Clarified requirements for the remaining documentation tasks. Use Cases and User Stories may need to be reformatted.

Outcomes:

- Change in format for the product backlog
- Clarity for the user manual and installation guide

Added to the Agenda:

- Reorganise the documentation to better display the product backlog

Meeting Duration: 15mins

7.4 Product Backlog 6

ID	Description	Tom	Gary	Diparati	Oguz	Priority	Date Completed
021	Player: Animations for all collisions	2hrs	5hrs	6 hrs	5 hrs	2	
034	NPC's: Archer npc's Design	4 hrs	4hrs	10 hrs	10 hrs	3	
036	NPC's: Slime npc's Design	4 hrs	4hrs	10 hrs	10 hrs	3	
039	NPC's: Archer npc's scripts	4 hrs	4hrs	10 hrs	10 hrs	3	
041	NPC's: Slime npc's scripts	2 hrs	2hrs	10 hrs	10 hrs	3	
044	NPC's: Archer npc's animations	2 hrs	4hrs	10 hrs	10 hrs	3	
046	NPC's: Slime npc's animations	2 hrs	5hrs	10 hrs	10 hrs	3	
047	NPC's: Mob patrolling script	4 hrs	6hrs	10 hrs	10 hrs	3	
053	Map: ladders	2 hrs	3hrs	6 hrs	5 hrs	4	
057	NPC's: Mob collision	2 hrs	4hrs	6 hrs	4 hrs	4	
060	Player: Taking damage on collision	1 hr	2hrs	2 hrs	1 hrs	5	
062	Player: Bounce animation when landing	10 mins	30mins	1 hr	20 mins	5	
063	NPC's: Blacksmith Design	3 hrs	4hrs	10 hrs	10 hrs	5	
064	NPC's: Blacksmith Animations	2 hrs	4hrs	10 hrs	10 hrs	5	
066	User Interface: Health bar	1 hrs	2hrs	2 hrs	4 hrs	6	10/12/2021
067	User Interface: Insanity bar	1 hrs	2hrs	2 hrs	4 hrs	6	10/12/2021
068	User Interface: Treasure counter	1 hrs	2hrs	2 hrs	4 hrs	6	
069	Loading Screen: Start menu	5 hrs	7hrs	3 hrs	10 hrs	6	11/12/2021
070	Loading Screen: leader board menu	20 hrs	4hrs	3 hrs	10 hrs	6	11/12/2021
071	Loading Screen: Winner screen	5 hrs	6hrs	5 hrs	10 hrs	6	
072	Loading Screen: Death screen	5 hrs	6hrs	6 hrs	5 hrs	6	11/12/2021
073	Loading Screen: Settings menu	3 hrs	4hrs	3 hrs	10 hrs	6	
088	Game Install: Build on Windows	3 hrs	4hrs	6 hrs	5 hrs	6	08/12/2021
089	Game Install: Build on MacOS	10 hrs	6hrs	6 hrs	5 hrs	6	08/12/2021
090	Game Install: Build on Linux	10 hrs	6hrs	6 hrs	5 hrs	6	08/12/2021
091	Player Script: Camera Setup	2 hrs	3hrs	6 hrs	5 hrs	6	10/12/2021
092	Player Script: Player damage and insanity	2 hrs	4hrs	6 hrs	5 hrs	6 08/12/2021	
093	Map: Swinging ball trap	4 hrs	6hrs	6 hrs	5 hrs	6	08/12/2021
B009	Game Install: check builds run.	3 hrs	6hrs	6 hrs	5 hrs	6	
B010	Map: Bug on swinging ball trap	2 hrs	3hrs	6 hrs	5 hrs	6	10/12/2021

B011	User Interface: leader board scaling issue	2 hrs	4hrs	6 hrs	5 hrs	6	08/12/2021
------	--	-------	------	-------	-------	---	------------

7.5 User Stories 6

User Story 021: As a player,
I want to have multiple traps in the game,
So that,
Each time I play the game, it does not get boring.
Acceptance Criteria;

The game needs not only one (the spike trap) trap, but other traps with different mechanics, These new trap should be a swinging spike ball trap, that is like a rope but damages the player on collision, and an arrow trap, which shoots arrows regularly for the player to dodge.

User Story 022: As a player,
I want to have the swinging spike trap be vary in length,
So that,
Each time I play the game, it is a unique experience, with randomised changes.
Acceptance Criteria;

The swinging spike traps should vary in length between 3 units and 5 units long, hence if the room is the same the spike traps length will be different, allowing unique experiences across game plays.

User Story 023: As a player,
I want to have the swinging spike trap damage me just as any other mob or trap would,
So that,
The mechanism of players taking damage is consistent across the game.
Acceptance Criteria;
The player will splatter blood and take 1 unit of damage upon colliding with the swinging spike trap.

User Story 024: As a player,
I want to be knocked back when an arrow hits me,
So that,
The arrow traps are realistically implemented in the game.
Acceptance Criteria;
The a collision with an arrow will knock the player back one unit, in the opposite direction it came from, similar to how the skeleton attacks work.

7.6 Use Cases Sprint 6

UC 016: Swinging Spike Trap

Scope: N/A

Level: User goal

Primary Actors: Player

Description: Add a new type of trap to the game. The Swinging Spike Trap can vary in its length, ranging from 3 to 5 blocks, inclusive.

Dependencies: UC XXX: Player Starts the game. Traps spawn in.

UC 003: Room design.

UC 004: Rooms join together.

UC 005: Room connections.

Assumptions: None.

Preconditions: Player has selected start the game.

Main Flow: Right after the player starts game, the map should be created and be made up of rooms. Some rooms have a chance to spawn the swinging spike trap in specific locations, and these rooms will have the traps. The Swinging Spike Traps must be spawned with a random length, between 3 and 5 units inclusive.

Subflows: None.

Alternative Flows: None.

Post Conditions: The map is loaded into the game, and the appropriate rooms potentially have a swinging spike trap.

Frequency of Occurrence: 50% in each room prefab, that has a spawn swinging spike trap tile.

Open Issues: Should we add spawn tiles for the swinging spike trap to more or less rooms? The current amount seems good. Testing may be needed to justify this.

Originating User Story: US 021, US 022, US 023

UC 017: Arrow Trap

Scope: N/A

Level: User goal

Primary Actors: Player

Description: The Arrow Trap is a new type of trap in the game. It damages and knockbacks the player in the opposite direction of the incoming arrow if a collision between the arrow and the player is detected.

Dependencies: Player Starts the game. Traps spawn in. Trap fires arrows, and the player gets hit by one of these arrows.

UC 003: Room design.

Assumptions: None.

Preconditions: Arrow has hit the player.

Main Flow: This Use Case extends UC 003, by improving map design. It assumes that an arrow has hit the player. When this happens, blood should splatter as in any other case where the player takes damage.

Additionally, the player should be knocked back in the opposite direction the arrow came from.

Subflows: None.

Alternative Flows: Player is already knocked back by a skeleton swing or a bomb explosion, so if an arrow hits the player in this state, it will not register another knockback.

Post Conditions: Player gets up after the knockback timer runs out.

Frequency of Occurrence: Knockback and blood splatter should occur each time the player gets hit by an arrow.

Open Issues: None - arrow trap functionality fully implemented.

Originating User Story: US 021, US 024

7.7 CRC Cards Sprint 6

Overview of CRC Card Changes: In this section we add the CRC Card for swinging spike traps and arrow traps and then update the room spawner CRC card to account for this.

Class Name: Room Spawner	Version: 3
Description: Spawn a single room and its contents	Associated Use Cases: UC002, UC003, UC016, UC017
Responsibilities	Collaborators
Spawn block Spawn spike Spawn Rope Spawn Lava Spawn Traps	Block Spike Rope Lava Map Spawner Trap Spawner

Class Name: Spawn Trap	Version: 1
Description: Chooses the most suitable trap to spawn at a given location and passes it back to room spawner to spawn	Associated Use Cases: UC016, UC017
Responsibilities	Collaborators
choose most suitable trap to spawn at a given location	Room Spawner

Class Name: Spawn Swinging Spike Ball	Version: 1
Description: Spawns a length of spike ball and handles its physics	Associated Use Cases: UC016
Responsibilities	Collaborators
Generate physics constraints for spike trap Damage Player Spawn multiple hanging traps based on most suitable length for trap	Player Variables Room Spawner

Class Name: Spawn Arrow Trap	Version: 1
Description: Spawns a trap which periodically fires arrows in a direction.	Associated Use Cases: UC017
Responsibilities	Collaborators
Choose most suitable direction to fire arrow Spawn arrow trap block periodically spawn Arrow	Map Spawner Room Spawner Arrow Projectile

Class Name: Arrow Projectile	Version: 1
Description: Controls arrows velocity and life time	Associated Use Cases: UC017
Responsibilities	Collaborators
Delete arrow if it hits something Damage Player Delete arrow once its life time has passed Set arrow Physics	Map Spawner Player Variables

7.8 Sprint Backlog 6

ID	Description	Developer
088	Game Install: Build on Windows	Oguz
089	Game Install: Build on MacOS	Oguz
090	Game Install: Build on Linux	Oguz
B009	Game Install: check builds run.	Oguz
091	Player Script: Camera Setup	Thomas
092	Player Script: Player damage and insanity	Thomas, Oguz
093	Map: Swinging ball trap	Thomas
B010	Map: Bug on swinging ball trap	Thomas
B011	User Interface: leader board scaling issue	Thomas
066	User Interface: Health bar	Thomas
067	User Interface: Insanity bar	Thomas
069	Loading Screen: Start menu	Diparati
070	Loading Screen: leader board menu	Diparati
072	Loading Screen: Death screen	Thomas

7.9 Sprint Review 6

This week, the spike ball trap was fully implemented, and its bugs were fixed. The process of re-searching how users would be able to install the game on different platforms began. Scaling issues across different screen sizes and pixel densities were also fixed. The game loop is finally completed, with the leaderboard appearing when the player dies, and a new, more challenging maze is generated when the player goes exits the current maze. The Health and Insanity Bars were visually added (they existed in effect, but were not visible in the UI). It was decided that next week, a comprehensive testing of the entire game would be carried out, and the installers or build files would be finalised and uploaded to the appropriate websites.

In Development:

- Game Installation - Build game and check that it runs on Windows, Linux, MacOS. (Oguz Tecirlioglu)
- Game Installation - Create a windows Installer for the game. (Oguz Tecirlioglu)

Bugs:

- Map Design - Bug related to the player attaching to the swinging spike ball traps, climbing it like a rope, was fixed. (Thomas Poultney, Oguz Tecirlioglu)
- User Interface - Scaling issues on leader board and treasure score across screen size and resolutions were fixed. (Thomas Poultney).

Blocked: None.

Done:

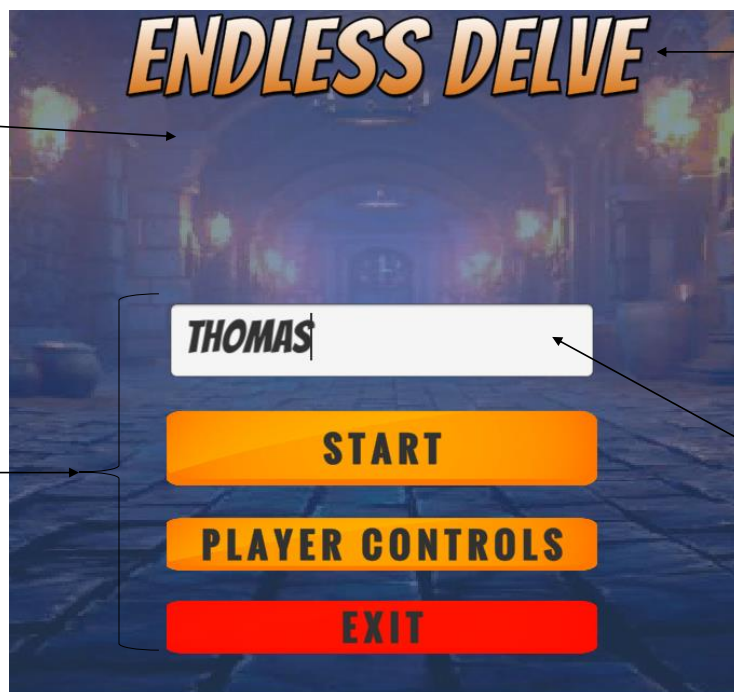
- Map Design - Swinging spike ball traps were added to the game. (Oguz Tecirlioglu)
- User Interface - The main menu and loading screen were added to the game. (Diparati Sen)
- Loading Screens - The Leader board was added to the game. (Diparati Sen)
- Scripts (In game) – Health Bar / Insanity Bar were added. (Thomas Poultney)
- User Interface – End of Game (Death) Menu. (Thomas Poultney)
- User Interface – Start Loading Screen added. (Diparati Sen)
- User Interface – Death Menu linked up with Scoreboard, so players can see their scores after the game is over. (Thomas Poultney)
- Player Scripts – Camera setup (Thomas Poultney).
- Player Scripts – Add damage, which references other scripts developed by Tom to add damage and knock-back effects on player. (Oguz Tecirlioglu)

7.10 User Interface

The next page contains the final summary of the user interface design, which is a reflection of the cumulative developments made in the past weeks regarding the user interface.

We used a scene from our dungeon as the background. This is a regular feature of games in the fantasy genre, including Runescape, Dungeon Siege, etc. The flickering torches on the stone walls call to mind the vibe we want to instil in the player.

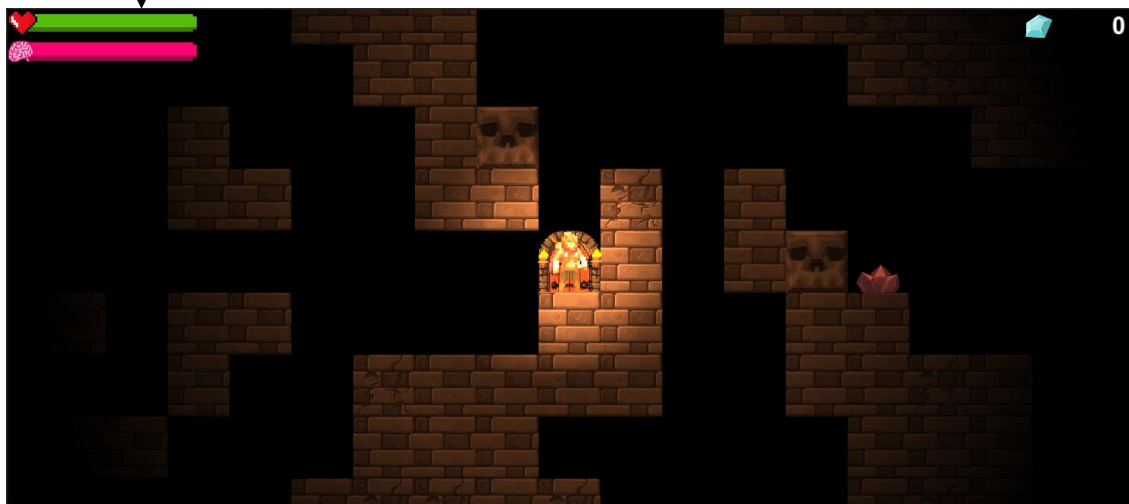
The simplistic start menu is standard for retro games, and the stacking of icons remains a staple of modern day menus. The name and start are the most important options, so we decided to make their fields slightly larger. Orange and blue are complementary colours, so they appear more vivid against the background. The colour red is used for exit, as humans psychologically associate red with stop.



We chose this font as it was reminiscent of games we played in the 90s, and was appropriate for the level of sophistication we planned to implement in our game. The yellow/orange gradient colour scheme was fairly typical for start icons of that era. As the title of our game and the heading for the menu, it is large and centred at the top of the screen.

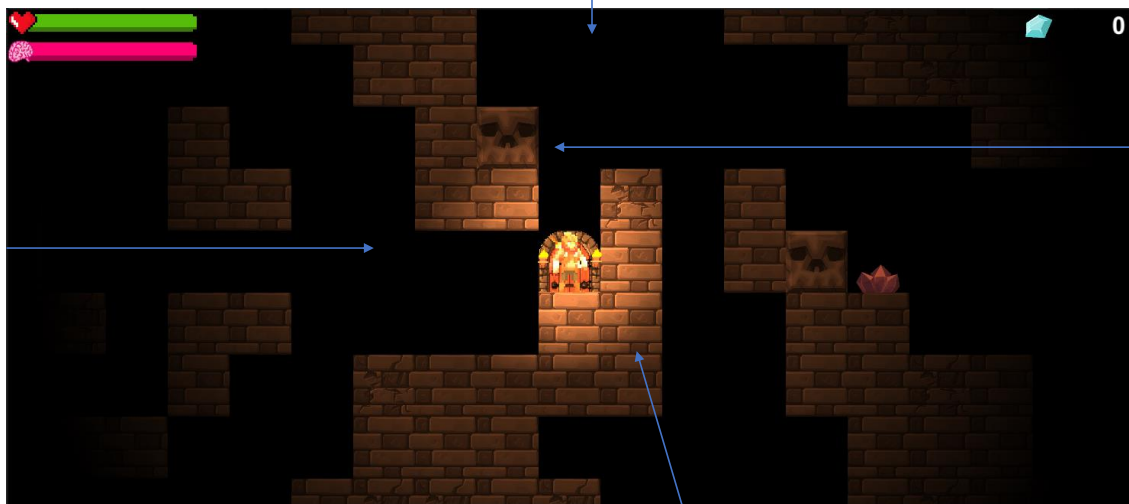
We needed to include a "name" field that enables the start option to become available, as it is pulled to the leader board. This is similar to arcade games.

We used bars to represent health and sanity. We debated tokens, however felt watching the bars steadily decrease gave a sense of time limit that was a major aspect of our game. We chose a heart with a green bar to represent health as this is traditional. We used a brain icon and pink bar to represent sanity as this too felt like a natural choice, and has the benefit of off setting the dark background.



A gem icon is used to portray score, as it reconciles with the objects you need to collect during the game to increase it.

A black background gives a sense of a lack of light reaching this underground environment, which we felt would immerse the player more in the idea that they are delving deep in to this cavernous dungeon.

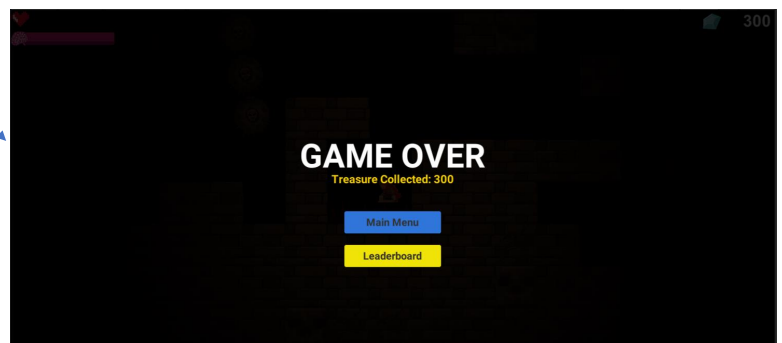
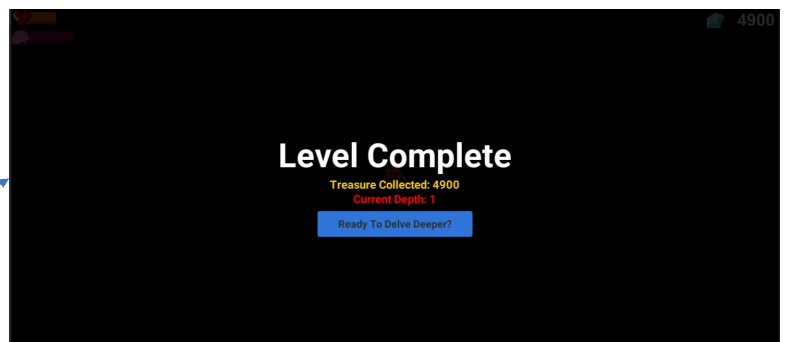


The platformer style we were implementing dictated the point of view of the player being from the side.

The brown platforms were designed to feel as though it was a crumbling, ancient dungeon. The skull textures reinforce the idea of danger.

The immersion is reinforced by the fact the main light source emanates from the player, and fades towards the edge of the field of view. We hoped this would yield a “fear of the unknown” as players cannot see enemies until they are close to them, and influences the difficulty.

These interfaces display at the end of each level, or when you die. The colour scheme of white on a black background is used as it does not pull the user out of the darker feel they're currently immersed in within the game. It keeps separation between the options menus, and the game. This is also consistent with the colour scheme of our user manual.



These options are reached at the beginning of the game from the main menu, and at the end of the game once your score is finalised respectively. As such, it made sense to keep the background, headings, and button textures consistent with the main menu.



7.11 Weekly Sprint Retrospective 6

This week the development team produced increments in a streamlined and efficient manner, and exceeded expectations. What was good about this sprint was the strong visibility between developers throughout the entire sprint. Tasks were allocated clearly and correctly, and each developer had a very confident understanding of the problem they had to solve, and managed their expectations efficiently (allocated an adequate amount of time, to something they knew they could do). This was something we had struggled with before, but now managed to overcome. Additionally, although developers worked on different areas of the game, pair programming was implemented from time to time followed by short brainstorming to explore other ways features could be implemented. This proved to be massively useful, and each developer completed their task. Cross-referencing artifacts is important to the process, and could sometimes get confusing. By talking with other developers, and ensuring that the increment being worked on is headed to the right direction, and compatible with the rest of the codebase that someone else may have worked on, increments were completed quickly.

On the documentation side, there was not much to reflect on. The process is straightforward, and the system we have installed, by automating Trello, works well. Trello provides weekly logs of what was completed in the backlog, what tasks were blocked, what tasks were bug fixes etc. By using this automation, an accurate and correct timeline of the process was produced weekly, which made document the sprints easier. Unfortunately, as mentioned in the agenda covered in the customer meeting this week (Customer Meeting 6), the User Stories and Use Cases need to be reformatted, and specifically the Use Cases need to present more detail regarding which other Use Cases they might be linked to, dependencies, assumptions, preconditions, etc. This meant that all Use Cases and Stories were reformatted, which was not ideal but necessary for strong documentation of the process.

8 Sprint 7: 13/12/2021 - 20/12/2021

8.1 Roles

8.2 Group Meetings 7

- Date – 13/12/2021
- Location – Teams
- Present – Phillip Reed, Diparati Sen, Tom Poultney, Oguz Tecirlioglu, Jack McGee, Sam Higgs, Ruoxi Li, Shuaiming Zhu
- Absent – None

Agenda Covered – Fully working game was presented to the group with intricacies explained, including the leaderboard and issues with gameplay mechanics. The process of unit testing and what areas to perform unit tests on was discussed. Allocation of remaining product documentation tasks. Allocation of remaining process documentation tasks.

Outcomes:

- Bugs were identified in the game i.e. spike traps were healing the player
- Remaining documentation tasks were identified and assigned to individuals
- Areas to unit test were identified
- The need to create an installer for the Windows and MacOS versions of the game, and a build folder for Linux was identified.

Added to the Agenda

- Write up and carry out unit tests
- Fix final few bugs ready to distribute the game
- Completion of remaining documentation tasks in the next week
- Create or finish process documentation files (User Manual, Maintenance Guide, etc.)
- Create installer for the game on Windows, a DMG file for MacOS, and build game files for Linux players.

Meeting Duration – 35mins**8.3 Customer Meeting 7**

- Date - 15/12/2021
- Location – CB5.13 / Teams
- Present – Phillip Reed, Sam Higgs, Oguz Tecirlioglu, Tom Poultney, Diparati Sen, Jack McGee
- Absent – Ruoxi Li, Shuaiming Zhu

Agenda Covered: Customer played the working game and provided their feedback. The areas we have decided to unit test were discussed with the customer. We explained why we were not able to unit test certain other features.

Outcomes:

- Customer was satisfied with the game and found it difficult and enjoyable
- Customer was happy with our choices of unit tests and our reasons for the specific areas

Added to the Agenda:

- Complete unit testing of specified features

Meeting Duration – 15mins**8.4 Product Backlog 7**

ID	Description	Tom	Gary	Diparati	Oguz	Priority	Date Completed
021	Player: Animations for all collisions	2hrs	2hrs	6 hrs	5 hrs	2	16/12/2021
034	NPC's: Archer npc's Design	4 hrs	4hrs	10 hrs	10 hrs	3	
036	NPC's: Slime npc's Design	4 hrs	4hrs	10 hrs	10 hrs	3	
039	NPC's: Archer npc's scripts	4 hrs	4hrs	10 hrs	10 hrs	3	
041	NPC's: Slime npc's scripts	2 hrs	2hrs	10 hrs	10 hrs	3	
044	NPC's: Archer npc's animations	2 hrs	3hrs	10 hrs	10 hrs	3	
046	NPC's: Slime npc's animations	2 hrs	3hrs	10 hrs	10 hrs	3	
047	NPC's: Mob patrolling script	4 hrs	6hrs	10 hrs	10 hrs	3	
053	Map: ladders	2 hrs	5hrs	6 hrs	5 hrs	4	
057	NPC's: Mob collision	2 hrs	4hrs	6 hrs	4 hrs	4	
060	Player: Taking damage on collision	1 hr	3hrs	2 hrs	1 hrs	5	
062	Player: Bounce animation when landing	10 mins	30mins	1 hr	20 mins	5	
063	NPC's: Blacksmith Design	3 hrs	4hrs	10 hrs	10 hrs	5	
064	NPC's: Blacksmith Animations	2 hrs	4hrs	10 hrs	10 hrs	5	
068	User Interface: Treasure counter	1 hrs	2hrs	2 hrs	4 hrs	6	
071	Loading Screen: Winner screen	5 hrs	6hrs	5 hrs	10 hrs	6	
073	Loading Screen: Settings menu	3 hrs	4hrs	3 hrs	10 hrs	6	
094	Loading Screen: Exit button to main menu	1 hrs	2hrs	6 hrs	5 hrs	1	16/12/2021
095	Loading Screen: Player Controls menu linked to main menu	2 hrs	3hrs	6 hrs	5 hrs	1	16/12/2021
096	Loading Screen: links leader board to death screen	2 hrs	3hrs	6 hrs	5 hrs	1	16/12/2021
097	Loading Screen: Death screen fixed	3 hrs	4hrs	6 hrs	5 hrs	1	16/12/2021
098	Loading Screen: Next map summary	3 hrs	4hrs	6 hrs	5 hrs	1	18/12/2021
099	Player Script: Interaction with end door	2 hrs	6hrs	6 hrs	5 hrs	1	18/12/2021
100	Map: End door animation	1 hr	4hrs	6 hrs	5 hrs	1	18/12/2021
101	Map: Lava animation/ particle effect	3 hrs	4hrs	6 hrs	5 hrs	1	16/12/2021
102	Map: Fire traps	3 hrs	6hrs	6 hrs	5 hrs	1	18/12/2021
103	Map Generation: Scale map difficulty	2 hrs	4hrs	6 hrs	5 hrs	1	16/12/2021
104	Map Generation: added more rooms	2 hrs	5hrs	6 hrs	5 hrs	1	18/12/2021
105	Map Generation: Spawning new map when end door reached	5 hrs	6hrs	6 hrs	5 hrs	1	18/12/2021
B009	Game Install: check builds run.	5 hrs	6hrs	6 hrs	5 hrs	1	12/16/2021

B012	User Interface: Health bar fix	1 hrs	3hrs	6 hrs	5 hrs	1	14/12/2021
B013	User Interface: Insanity bar fix	1 hrs	3hrs	6 hrs	5 hrs	1	14/12/2021
B014	Player Script: fix health and insanity levels	1 hrs	3hrs	6 hrs	5 hrs	1	14/12/2021
B015	Player Script: fixed Camera Setup	1 hrs	2hrs	6 hrs	5 hrs	1	14/12/2021
B016	Map: Arrow traps damage fixed	2 hrs	4hrs	6 hrs	5 hrs	1	16/12/2021
B017	Map: traps tiles added to prefab	3 hrs	4hrs	6 hrs	5 hrs	1	16/12/2021

8.5 User Stories

User Story 025: As a user,

I want to install the game on my operating system easily

So that,

It is easy for me to install and play the game on these platforms.

Acceptance Criteria;

Only one download file should be required to setup and play the game on Windows and MacOS. On Linux, the user will have to download the entire build file and run the executable inside with. The process may be lengthier for Linux, as Linux players are not target audience.

User Story 026: As a user,

I want to input my Player's nickname from the main menu

So that,

I can see where I rank on the leaderboard when I finish the game.

Acceptance Criteria;

The user inputs their nickname, and it gets saved. When the game is over, if the correct nickname is on the leaderboard with the correct score, the user is satisfied.

User Story 027: As a user,

I want to navigate the main menu, and view player controls

So that,

I can see what inputs affect my in game player, and how I can play the game.

Acceptance Criteria;

The user selects "Controls" from the main menu and sees a screen showing the in-game controls. By pressing the "Back" button, the user returns back to the main menu.

8.6 Use Cases Sprint 7

UC 018: Game Installation & Setup

Scope: N/A

Level: User goal

Primary Actors: User (Not player yet)

Description: The User must download and install the game from somewhere before they can play it. For Windows and MacOS, a single, installer is enough for this. For Linux, the whole build folder must be downloaded.

Dependencies: None.

Assumptions: User is capable of downloading files and running an installer, or downloading applications from the command line in Linux.

Preconditions: Player has downloaded the installer.

Main Flow: Upon downloading the respective installer file, the user double clicks it and runs the installer. For windows, the user selects a directory, creates a desktop shortcut, and launches the game from the desktop shortcut. For MacOS, the user double clicks the DMG file, and runs the Application File inside of it. For Linux, the user must open the build folder, set the file with the extension `.x86_64` as an executable, and launch the game.

Subflows: None.

Alternative Flows: The users anti-virus could give an warning. This problem is mentioned and solved in the User Installation Guide.

Post Conditions: The map is loaded into the game, and the appropriate rooms potentially have a swinging spike trap.

Frequency of Occurrence: 50% in each room that has a spawn swinging spike trap.

Open Issues: Should we add spawn tiles for the swinging spike trap to more or less rooms? The current

amount seems good. Testing may be needed to justify this.

Originating User Story: US 025

UC 019: User Nickname Input

Scope: N/A

Level: User goal

Primary Actors: User (Not player yet)

Description: The User navigates inputs the players nickname on the main menu.

Dependencies: UC 014: Player Starts the game from Main Menu.

Assumptions: Main menu is implemented.

Preconditions: None.

Main Flow: The user launches the game, and is greeted by the Main Menu. The user enters their nickname in the white, text input box. The user plays the game, and finally sees their nickname on the leaderboard if a high enough score is achieved.

Subflows: None.

Alternative Flows: The user does not enter a nickname, and cannot launch the game.

Frequency of Occurrence: At the start of each game.

Open Issues: If the user enters their name, and views the controls, the name disappears. Should the nickname be saved in state?

Originating User Story: US 026

UC 020: Controls Window Navigation

Scope: N/A

Level: User goal

Primary Actors: User (Not player yet)

Description: The User enters the main menu, and selects the "Controls" tab. From here, a screen showing in-game controls appears. After learning the game controls, the user can return back to the main menu by selecting "Back".

Dependencies: UC 014: Player Starts the game from Main Menu.

Assumptions: Main menu is implemented.

Preconditions: None.

Main Flow: The user launches the game, and is greeted by the Main Menu. User selects "Controls, sees and learns the controls, and selects "Back" to return back to the main menu.

Subflows: None.

Alternative Flows: None.

Frequency of Occurrence: Optional, as desired by the user.

Open Issues: If the user enters their name, and views the controls, the name disappears. Should the nickname be saved in state?

Originating User Story: US 027

8.7 Sprint Backlog 7

ID	Description	Developer
B012	User Interface: Health bar fix	Thomas
B013	User Interface: Insanity bar fix	Thomas
068	User Interface: Treasure counter	Thomas
094	Loading Screen: Exit button to main menu	Thomas

095	Loading Screen: Player Controls menu linked to main menu	Thomas, Diparati
096	Loading Screen: links leader board to death screen	Thomas, Diparati
097	Loading Screen: Death screen fixed	Thomas
098	Loading Screen: Next map summary	Thomas
B014	Player Script: fix health and insanity levels	Thomas
B015	Player Script: fixed Camera Setup	Thomas
099	Player Script: Interaction with end door	Thomas
100	Map: End door animation	Thomas
101	Map: Lava animation/ particle effect	Thomas
B016	Map: Arrow traps damage fixed	Oguz
102	Map: Fire traps	Oguz
B017	Map: traps tiles added to prefab	Oguz
104	Map Generation: Scale map difficulty	Thomas
105	Map Generation: added more rooms	Thomas
106	Map Generation: Spawning new map when end door reached	Thomas

8.8 Sprint Review 7

This week was the final week. The swinging spike trap was fully implemented. Initially unit tests were intended to be implemented but they proved to be too problematic. The code base had too many inter-dependencies, and the effort to implement them at this stage was not justified. The user interface elements were all implemented, with health/insanity bars, score counters, leaderboards, death menu added. Game continuity was achieved by making the player enter a new room / maze, when the room exit is found and "E" is pressed. The game was finished, and Build files and installers for various operating systems were created. The team finished the documentation.

In Development: None.

Bugs:

- Map Design - Player can attach to spike chain traps (thomasapoultney, oguztecirlioglu)
 - Fixed by adding a new layer mask whatIsWall to the player controller, This is used when checking if the player is able to attach to wall or ledge. It was previously checking for whatIsGround which included the spike layer
- User Interface - Main Menu Rescaling issues [Fixed] (thomasapoultney)
- Player Scripts - Insanity doesnt restart when spawning new world [Fixed] (thomasapoultney)
- User Interface - Rescaling issues on leaderboard and treasure score display [Fixed] (thomasapoultney)
- User Interface Leaderboard - Leaderboard shows error too many requests if player dies too fast [Fixed] (thomasapoultney)

Blocked:

- Unit Testing - Unit tests were almost impossible to implement at this stage, as the code base had too many inter-dependencies. This made it nearly impossible to isolate functions, and a lot of work would have to be carried out for this. The opportunity cost was not worth it, and manual tests were carried out instead.

Done:

- User Interface Scripts (In game) - Health Bar/ Insanity Bar (thomasapoultney)
- User Interface Scripts (In game) - Treasure collected (thomasapoultney)
- User Interface - Add Exit Button to Main Menu (thomasapoultney)
- User Interface - Link up Death Menu with scoreboard (thomasapoultney)
- User Interface - Death Menu (thomasapoultney) (thomasapoultney)
- User Interface - Load next level summary (thomasapoultney)
- Player Scripts - Health and insanity levels (thomasapoultney)
- Player Scripts - Camera setup (thomasapoultney)
- Player Scripts - Interact with door (thomasapoultney)
- Player Scripts - Camera setup (thomasapoultney)
- Animations - Door Opening (thomasapoultney)
- Loading Screens - Start (diparatisen)
- Loading Screens - Leaderboard (diparatisen)
- Lava animation and particle effect (thomasapoultney)
- Map Design - Arrow Traps (traps) (oguztecirlioglu)
 - Arrow damage added. Spawner tile added to multiple rooms.
 - Arrow traps are done, but need to add the player damaging effect to them. Also the arrow trap spawner tile prefab should be added to each rooms design.
- Map Design - Fire Traps or other (traps) (oguztecirlioglu)
- Map Design - Arrow Trap Tile (Bug) (oguztecirlioglu)
- Map Design - Add Trap Tiles to Room Prefabs (oguztecirlioglu)
- Map Design - Lava animation and particle effect (thomasapoultney)
- Map Generation - Scale Map difficulty depending on dungeon depth (thomasapoultney)
- Map Generation - More rooms (thomasapoultney)
- Map Generation - Spawn new world when player reaches the end Door (thomasapoultney)

8.9 Exception handling 7

Unexpected Situation	Status	Actions to resolve
Leaderboard website went down in this week. This stopped us from fetching the high scores from the leaderboard and caused our menu to infinitely displayed fetching	unresolved since it is the final week and we have to focus on documentation.	We do not have enough time to implement the fix this week. We have set this as a future expansion of the project with a high priority. The process of fixing this issue is discussed in detail in the maintenance document future expansions section.

8.10 Weekly Sprint Retrospective 7

This week was the final sprint. The developers finalised development. An emergency meeting was called on Saturday the 18th of December, to finalise the documentation and ensure that everything was cross-referenced, documented, and presented properly. There was concern that the developers may have implemented changes that the team was not fully aware of, or the team may have written the same increment multiple times. The team felt confident that they had mastered working together as a software engineering group, but piecing together the documentation required an additional degree of cooperation that no one had anticipated. This caused some stress, regarding whether the submission would encapsulate everything we had done and document the process perfectly.

The emergency meeting helped immensely. A new Trello board specifically focused on organising the documentation process was created, so that the team could allocate the remaining documentation work efficiently and everyone would be aware of the procedure. The team remained in strong contact through Discord for the rest of the sprint, and the documentation was completed successfully.

9 User Questionnaire Form

The template we have elected to use in gathering user evaluations is called the Game User Experience Satisfaction Scale (GUESS) – 18. This is a truncated version of the more comprehensive GUESS, and was created by Keebler et al. As the name suggests, GUESS-18 is made up of 18 statements across 9 constructs. It uses a 7-point Likert scale to assess a user's feeling towards each statement.

There is a host of different evaluation surveys of a similar nature, each of which have a slightly different design depending on its specific goal. When deciding on a template, we outlined what we wanted our user evaluations to achieve. We came to the conclusion that we wanted to cast a relatively wide net and gather feedback across a range of aspects to the game, as opposed to focusing in on any particular area. We believed this was suitable given our experience level, and also felt it would allow us to prioritise developing the weakest areas of our game. This led us initially to GUESS, which was created through analysing existing game scales to find common patterns. It is a widely used survey, and we felt its 9 constructs were a good representation of what we wanted to gain from conducting the evaluations. However, at 55 statements long it was cumbersome, and overkill for a relatively simple game. Instead, we opted for GUESS-18. This is a more streamlined version that still maintains the core principles of GUESS.