

Documentation

He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji,
Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

Contents

Week 1: 16th- 22nd October	5
Process	5
Weekly overview	5
Meeting minutes	6
backlog	8
Exception handling	9
product contents	10
Customer interview	10
User stories	11
User stories tests	12
Use cases	13
Use cases Tests	14
Software design Documentation	15
User interface design	16
Week 2: 23rd-29th October	17
Process	17
Weekly overview	17
Meeting minutes	19
Backlog	23
Exceptional handling	24
Product content	25
Customer meeting	25
User stories	26
User story testing	27
Use cases	28
Use cases testing	35
Software design documentation	36
User interface design	38

Week 3: 30th October - 5th November	39
Process	39
Weekly Overview	39
Meeting Minutes	40
Backlog	46
Exception Handling	48
product contents	49
Customer interview	49
User stories	50
User story testing	51
Use cases	52
Use cases testing	54
Software design Documentation	55
User interface design	56
Week 4: 6th - 12th November	57
Process	57
Weekly Overview	57
Meeting Minutes	58
backlog	70
Exception Handling	72
product contents	73
Customer interview	73
User stories	74
User story testing	75
Use cases	77
Use cases testing	80
Software design Documentation	81
User interface design	84
Week 5 documentation: 13th-19th November	85
Process	85
Weekly overview	85
Meeting minutes	86
Backlog	93
Exceptional handling	95
Product content	96
Customer meeting	96
User stories	97
User story testing	98
Use cases	100
Use case testing	104
Software design documentation	105

User interface design	106
Week 6 documentation: 20th-26th November	107
Process	107
Weekly Overview	107
Meeting Minutes	108
Backlog	114
Exception Handling	116
Product content	117
Customer meeting	117
User stories	117
User story testing	119
Use cases	121
Use cases testing	124
Software design Documentation	125
User interface design	126
Week 7 documentation: 27th November-3rd December	128
Process	128
Weekly Overview	128
Meeting Minutes	129
Backlog	131
Exception Handling	133
Product content	134
Customer meeting	134
User stories	135
User story testing	137
Use cases	139
Use cases testing	142
Software design Documentation	143
User interface design	144
Week 8 Documentation: 4th- 8th December	146
Process	146
Weekly overview	146
Meeting minutes	146
Backlog	148
Exception Handling:	150
Product content	151
customer meeting	151
User stories	152
User story testing	154
Use cases	156
Use case testing	157

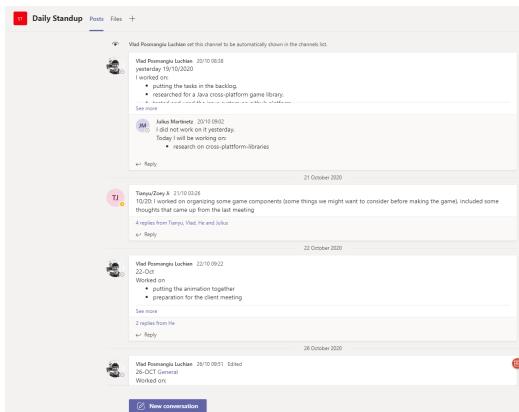
Software design documentation	158
User interface design	159

Week 1: 16th- 22nd October

Process

Weekly overview

This was the start of the project, meaning there was no coding. Instead, we planned: what game we wanted to create based on the initial customer specification; the programming language; control flow and communication (meetings), and the agile approach to the project. The week consisted of 2 meetings. The first meeting was to introduce ourselves and brainstorm ideas for the game, as well as plan the agile approach (Scrum) for the week. The second meeting was a sprint review followed by preparing the customer presentation. In between these two meetings we had daily stand-ups that were carried out on Microsoft teams.



Since the Scrum approach was chosen to be used in this project, we assigned a product lead and Scrum master. The product lead was responsible for setting out the sprint backlog from the product backlog created from the brainstorm meeting, while the Scrum master assured daily stand-up posts were being completed. All the sprint information was stored on GitHub for ease of access and updating the tasks.

From this initial sprint, we were able to analyse the use of teams for daily stand-up posts. This approach was great for dealing with multiple time zones as we had members in China, and clearly outlined what everyone had done, struggled with, and what they are working on. However, it did not notify everyone until the group was tagged in the post. This resulted in members missing the first couple of stand-ups. So, heading forward the Scrum master will tag the team in the initial post.

Meeting minutes

Brainstorming and sprint planning meeting

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

Minutes:

- Everyone in the group introduced themselves and discussed their programming background
- We decided what programming language we thought would be best for the project
- Each member presented their idea for the game based on the customer criteria given:
 - Top down view with the aim to escape the dungeon by doing tasks
 - Text type game where the user will pick answers that dedicate their journey through the dungeon
 - Side scroller game where the player can interact with the map as seen in Minecraft
- From the presentations we brainstormed a final game idea:
 - The player can generate tools from gathering resources from the map (blocks)
 - These tools will perish meaning the player will keep gathering resources throughout the game
 - The player can be assigned tasks to complete so they can escape the dungeon
 - There could be a bot which can either help you or you race against it to escape
 - The game will have a time constraint so the user can be aware how good they are doing
 - The user can battle bosses which can be in a top down view
- We established initial tools to use in this project. This was GitHub to contain the sprint documentation and share code between use. Then teams for communication within the project as well as daily stand-ups.
- The first sprint everyone was assigned the same tasks as it was all researched based. These include find a cross-platform Java library, research version control

and how GitHub works, game development research, Java research. The end of the sprint was the Friday.

sprint review and presentation creation

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

- From the assigned sprint tasks we were able to complete many of them for different individuals.
- As many were understanding concept tasks, not all could be considered done. E.G for GitHub, some team members were still not comfortable with it.
- We created a presentation for the customer meeting next week. This contained our original idea for the game for the customer to decide if it fills the starting criteria to develop from.

This is a link to the first presentation for the customer meeting 23rd October: [link](#)

backlog

Product backlog:

- Find cross platform library
- Understand GitHub for version control
- Java research
- Game development research
- Generation of block map (side on view)
- The blocks of the map can be interacted with by the player
- A player is spawned on the map
- The player can interact with the blocks (destroy and create)
- The player can collect these resources in an inventory
- The players resources can be used to craft items (pickaxe, sword, and light)
- The crafted tools will break over time
- The player will have a sidekick (bot) which will either help or race against the player
- There will be a timer shown to the player on the screen which runs down as they go through the game
- There will be tasks to complete in order for the player to escape
- The player can go through a door and the screen will change to top down view
- In the top down view, the player will fight bosses

sprint backlog

- Find cross platform library
- Understand GitHub for version control
- Java research
- Game development research

Exception handling

At this early stage there were no issues within the project.

product contents

Customer interview

In this week we had no customer meeting but instead the customer debrief:

- I want a dungeon game that I can play on my laptop, or perhaps on my phone.
- Perhaps it would be nice to be able to build a “bot” to play the game on its own or perhaps to compete with other bots?
- How can I know whether I or my bot are doing well?
- How can the challenge of the game be changed?

This was vague meaning we had plenty of choice of where we wanted to go with the game. One key part was the game being able to be played on multiple devices. This made Java a good choice for game development as it can be used for android gaming. The use of a bot in our mining dungeon game had multiple possibilities such as: mining for the player, hold items for the player, provide hints and tips, and fight for the player. In order to track progress, we could implement a progress bar, time limit, kill number, equipment gained (precious ores). The game can have multiple ways of how it can change the difficulty. We can include achievement system, creative or survival mode, altitude challenge, mining challenge and adjust time constraint.

User stories

ID	Version	Priority	Story
US_011	1		AS A player I WANT to be able to play a dungeon game on different platforms SO THAT I have choices for playing the game
US_021	4		AS A player I WANT to have a bot that will play for me SO THAT It can help me with playing through the game
US_031	5		AS A player I WANT to be able to compete with a bot SO THAT I can have a goal when playing
US_041	6		AS A player I WANT to be able to see my progress SO THAT So that I know if I am doing well at the game as I play
US_051	7		AS A player I WANT to have different choices between games modes SO THAT I have new scenarios to play
US_061	2		AS A player I WANT to move around and interact with the blocks on the map SO THAT I can gather resources, place resources, and make new paths
US_071	3		AS A player I WANT to use resources to craft items SO THAT I can use them throughout the game to make tasks easier

User stories tests

User story	ID	Test acceptance	comments	completed
	US_01	Run code on different systems and see if code compiles		NO
	US_02	A bot character is generated on the game screen. The bot aids the user with text aids	The bot will have to be able to follow the player around the map, meaning it will need similar movement animations to the main character	NO
	US_03	Monitor to see if the bot is attempting task within the game		NO
	US_04	There is a visual queue on the screen that will track specific progression.	There are multiple types of progress – elevation, time or collection	NO
	US_05	Within the menu screen the player sets the game mode and the characteristics of the game change	These characteristics could be unlimited health, unlimited resources and no enemies	NO
	US_06	The player can destroy a block and it disappears from the game screen. This block will then show up with in the inventory	From the inventory screen there are multiple options the player can take which require use cases and user stories	NO
	US_07	There is an option to craft items from the inventory screen. If the player has the required resources, they can craft the item and the resources are taken from their inventory	You can implement many ways to show what can be crafted with the resources provided.	NO

Use cases

At this stage of the process we had not considered to implement use cases in our development approach. This was due to us not having our game idea confirmed by the customer, as well as not being aware of the process since the lecture material was not released yet. Instead we had user stories as shown in the user story section, which is a similar to a less dressed version of the use cases.

Use cases Tests

As we had no use case there were no tests to conduct.

Software design Documentation

At this stage, no thought has been given into the software design and classes as the mining game has not been confirmed by the customer.

User interface design

This is a basic framework of how the potential game will look like to the user.

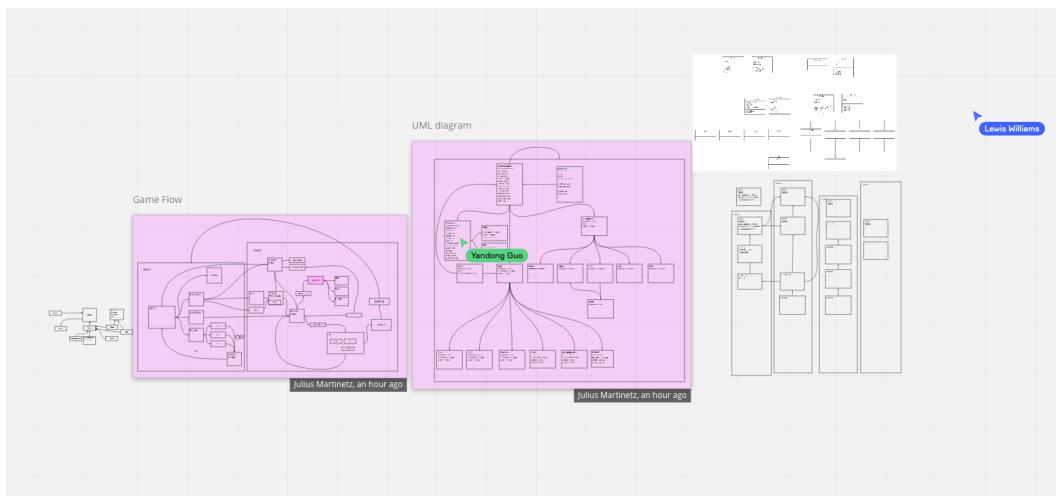
Design implement	Version
There will be a starting menu screen with the game setting and the start function	1
Once in the game there will be character that can move and interact with the whole map	1
When the player had successfully completed the game they will be sent back to the starting menu screen	1

Week 2: 23rd-29th October

Process

Weekly overview

With the customer happy with our initial game idea for a Minecraft inspired dungeon game, we were able to start planning out the game and wrote some basic code. This meant the product backlog continued to be filled out with ideas that we had, and the initial game architecture was looked at: menu system and game function. To achieve this planned architecture we used Miro board, which was also used to sort out the class UML. This was also based on our product backlog, but we did not use CRC cards which could have made the process simpler. The week had 3 meetings: customer debrief with sprint plan; game architecture; and sprint review with presentation creation. For the coding side everyone was assigned to generate a basic square that could move left, right, up, and down.



The Scrum technique was continued for this week with the Scrum master and product lead roles being switched around, so each member of the group can experience them. In between these first and last meeting we had daily stand-ups that were carried out on Microsoft teams. From the last sprint we made sure to tag the team within the post, so everyone will get a notification and reminder to post an update.

The screenshot shows a Jira Daily Standup board with four posts:

- Post 1 (26 October 2020):** Vlad Posmangiu Luchian at 26/10 09:51 Edited. 26-OCT General. Worked on:
 - nothing
 Had difficulties with: See more. 4 replies from you, Julius and Samuel.
- Post 2 (27 October 2020):** Vlad Posmangiu Luchian at 27/10 10:02. 27-OCT Daily Standup. Worked on:
 - dec environment on mac - working
 - getting the library installed and working - done
 See more. 5 replies from you, Julius, Vlad and 2 others.
- Post 3 (28 October 2020):** Tianyu/Zoey Ji at 28/10 02:44. 28-OCT. Worked on:
 - nothing
 See more. 3 replies from Vlad, Julius and Lewis.
- Post 4 (29 October 2020):** Vlad Posmangiu Luchian at 29/10 15:11. 29 OCT.

A "New conversation" button is visible at the bottom of the board.

This sprint was conducted in much the same way as the last one as we thought it was generating good results in the last sprint. The only problem was we still assigned everybody the coding task which is not efficient. However, there were 3 different systems (mac, windows Linux) used by different members of the team and the task would help in getting people comfortable with LibGDX. Heading into the next sprint we will make sure to spread the tasks more efficiently so that a greater progression of the game can be achieved.

We assigned the sprint tasks from looking at the product backlog and the existing user stories we had. Since US_01 was a great priority we assigned everybody to coding the character with movement as explained before.

Attached here is the presentation created for the customer meeting on the 30th November, which will include the GIF version of some of the pictures in this weeks report: [Presentation](#)

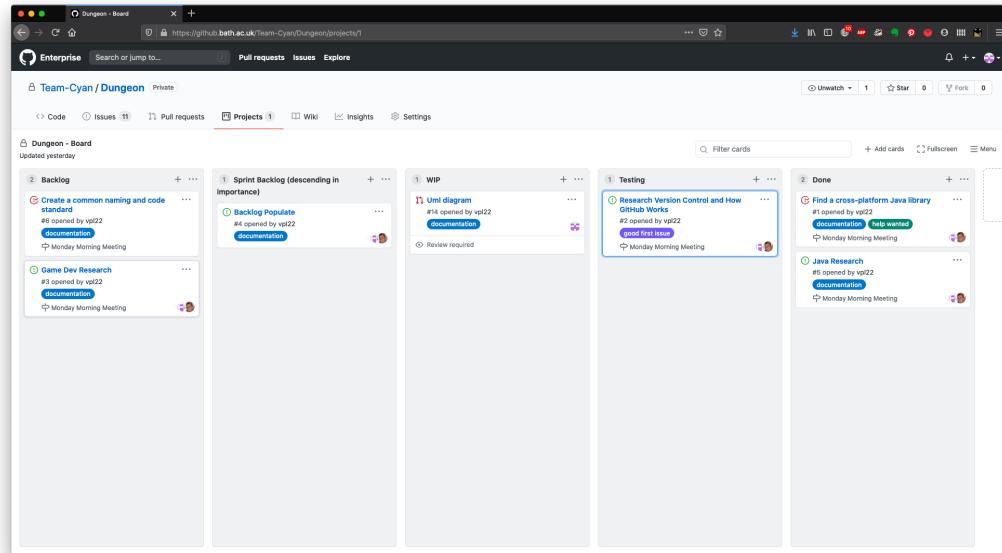
Meeting minutes

Customer meeting debrief and sprint planning:

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

We discussed how the customer meeting went and how it affected the game. In the meeting the customer said they were happy with the idea of doing an interactive game like Minecraft in a dungeon themed setting. We presented a broad overview of the game so there was plenty to look into as potential final ideas, meaning the customer still has plenty of choice in how the game will look and feel. One important part was clarification of cross platform relating to Windows, Linux and MacOS instead of referring to mobile. So, our focus will be ensuring the game runs on those operating systems with android as a potential bonus.

Now the initial game idea was confirmed, and we have established our coding language with framework (Java with Libgdx), we initiated coding with this sprint. As no one had used Libgdx before everyone was assigned to create a basic character (square) and give it basic movement (up, down, left and right). On top of this everyone was assigned finding a naming system, designing classes and UML diagram.

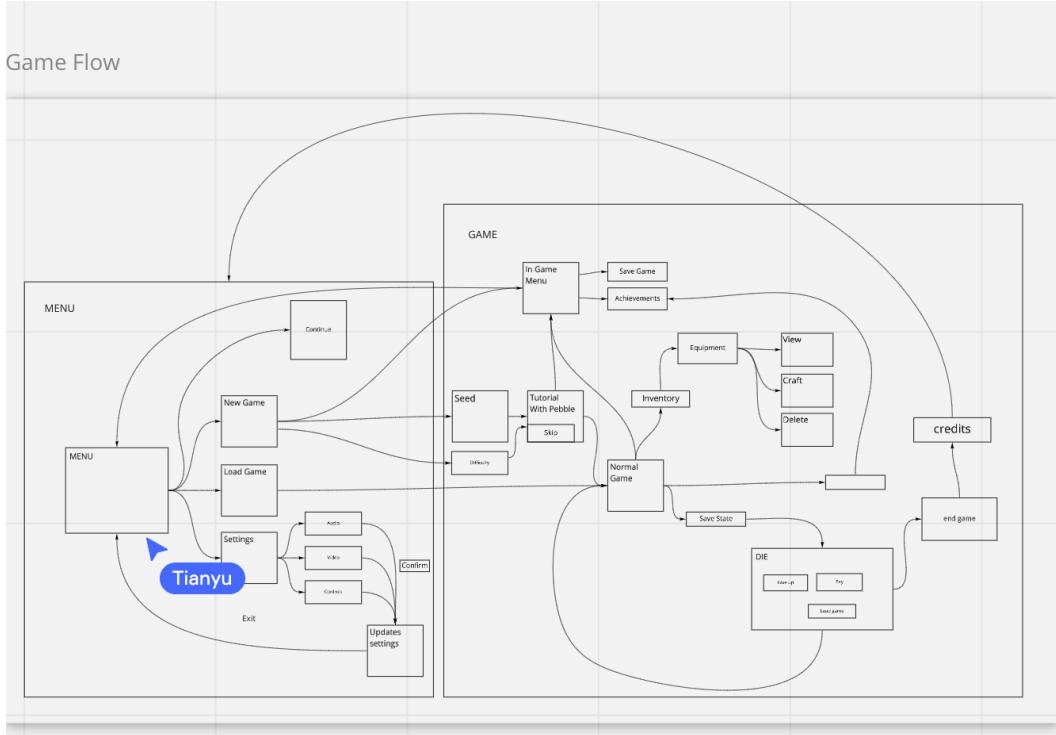


Miro board

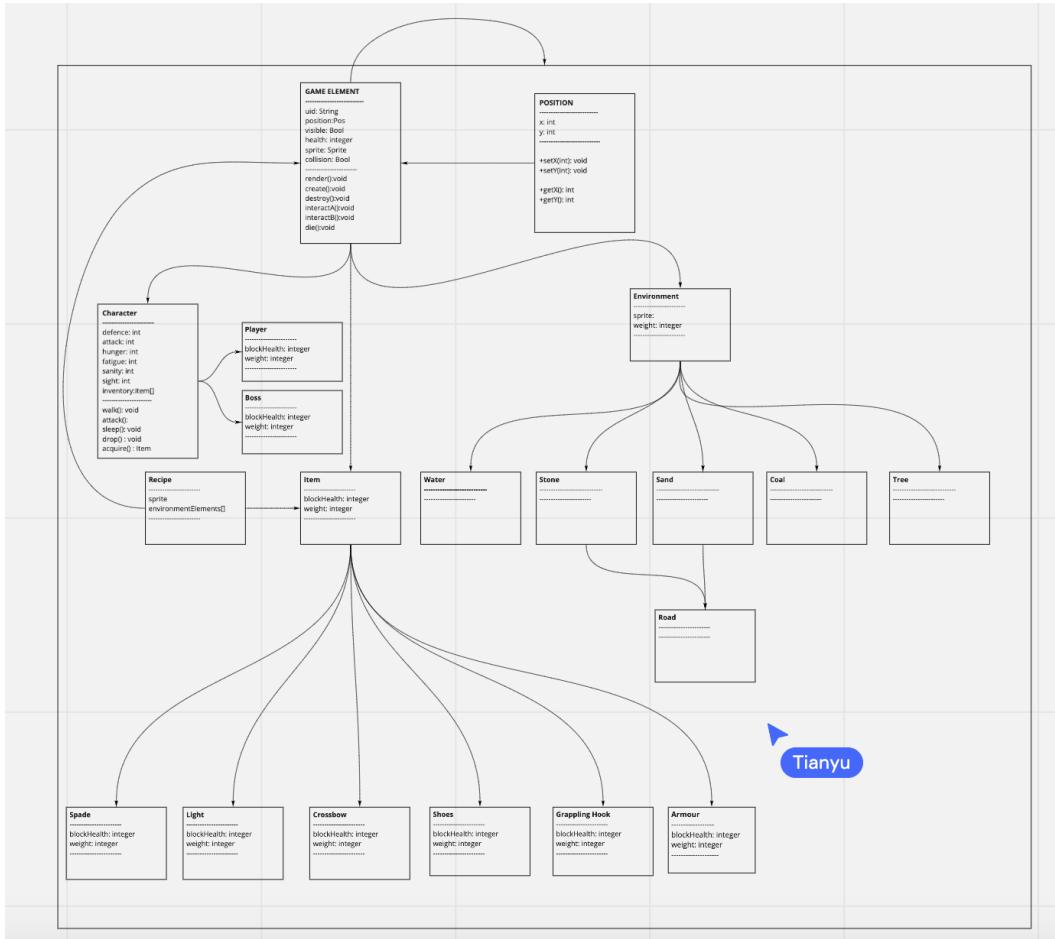
Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

In this meeting we used our thoughts on classes to generate the architecture of the

game as well as a UML diagram. First, we looked at the game flow which will be how the menu and game will function together, which can be seen below. The important part of this architecture is the ability to save the game and how this will function in different scenarios. So, it shows the flow with saving the game and then continuing the game from this save state. Upon dying the user will be able to start from the last save state. From the initiating a new game from the starting menu screen, a new seed will be generated. This will allow the user to have more unique experiences in future playthrough of our game.



In the next part we designed an UML off the initial idea for the class system as shown below as well as our product backlog. Everything will stem from the Game element, as our map is fully interactive. This means each block will in theory die when the player destroys it. like and enemy and main character. The process of designing this UML diagram did not use CRC cards as we were not aware of that. Instead we all interacted with the Miro board and discussed our ideas until we had the classes mapped out. This was probably not as efficient as using CRC cards and is further discusses in the software design section.



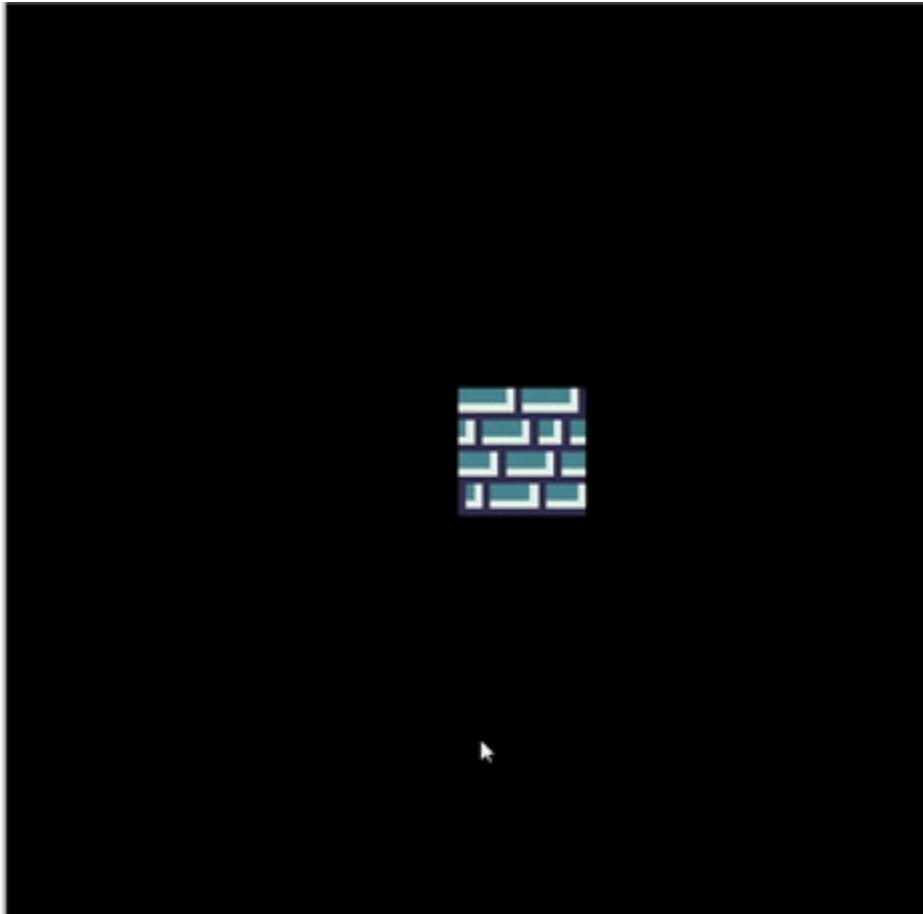
Sprint review and presentation creation

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

From this second sprint we were all able to generate a basic character and have basic movement using LibGdx and Java. The naming class we established and an initial UML board was generated. These can be seen below with the UML shown above. No one had any major problems, so the sprint was considered successful.

4 lines (4 sloc) 179 Bytes Raw Blame   

1. Sprite variable names: start with 'spr'
2. Character variable names: start with 'cha'
3. functions: lower CamelCase (i.e. 'twoWords')
4. Sound variable names: start with 'snd'



We then created the presentation for the customer at the end of the week. Instead of using PowerPoint we used Markdown. This was so we could show GIFs as well as keep all the documentation on GitHub with the code, meaning everyone could easily access it. The presentation is linked in the weekly overview section. To create it we met in teams, with one member sharing their screen as we discussed what to put in. We decided to use the same template every week following this one: last time, updates, and future.

Backlog

Product backlog:

- Create a character (square) and give it basic movement (up, down, left, right)
- Find a common naming system
- Design classes and create a UML diagram
- Generation of block map (side on view)
- The blocks of the map can be interacted with by the player
- A player is spawned on the map
- The player can interact with the blocks (destroy and create)
- The player can collect these resources in an inventory
- The players resources can be used to craft items (pickaxe, sword, and light)
- The crafted tools will break over time
- The player will have a sidekick (bot) which will either help or race against the player
- There will be a timer shown to the player on the screen which runs down as they go through the game
- There will be tasks to complete in order for the player to escape
- The player can go through a door and the screen will change to top down view
- In the top down view, the player will fight bosses

Sprint backlog:

- Create a simple character (square)
- give this character up, down, left, and right movement
- Find a common naming system
- Design classes and create a UML diagram

Completed:

- Find cross platform library
- Research on how GitHub version control works
- Java research
- Game development research

Exceptional handling

There were no problems with this sprint or week.

Product content

Customer meeting

At the start of the week we had our first customer meeting which was presenting our initial idea for the game. The customer was happy with the premise of a mining based game where the user is able to interact with the map, whether it be destroying or building. We did not present a rigid plot. Instead we simply suggested the idea of the player falling down a hole where they awake and see a small figure approach (name: Pebble). Pebble explains the world they are now in and that he can help the player escape back above the surface. We suggested that Pebble could be the imagination of the main player, which could lead to many story plots. The feature we suggested was sanity which could change throughout the game. As the player gets more insane Pebble will help the player out more. This game mechanic can be used to help the player when they are stuck. The customer liked the idea of this and how we can implement this, and suggested we looked into Senua's sacrifice. Finally, we showed some potential art styles for the game. The customer expressed interest in the high resolution with earth tones

We showed the progress of our game development. This was the UML diagram and game architecture. As the customer is very well versed in software engineering he suggested we look into Research Model view, and adapt out UML. We also showed the basic character moving around the screen to show visual game progress. All of these are shown in previous examples.

User stories

ID	Version	Priority	Story
US_01	2	1	AS A Player I WANT to be able to play a dungeon game on different platforms web platforms SO THAT I have choices for playing the game
US_02	2	4	AS A Player I WANT to have a bot that will play for me SO THAT It can help me with playing through the game by offering advice
US_03	Deleting	Deleting	AS A Player I WANT to be able to compete with a bot SO THAT I can have a goal when playing
US_04	1	5	AS A Player I WANT to be able to see my progress SO THAT So that I know if I am doing well at the game as I play
US_05	2	7	AS A Player I WANT to have different choices between games modes and different paths to the end in each game mode SO THAT I have new scenarios to play
US_06	1	2	AS A Player I WANT to interact with the blocks on the map SO THAT I can gather resources, place resources, and make new paths
US_07	1	3	AS A Player I WANT to use resources to craft items SO THAT I can use them throughout the game to make tasks easier
US_08	1	6	AS A player I WANT to be able to regenerate health SO THAT I can survive longer

User story testing

User story ID	Test	Comments	completed
US_01	Run code on different systems and see if code compiles	Web based systems on Windows, MacOS and Linux	YES
US_02	A bot character is generated on the game screen. The bot aids the user with text aids	The bot will have to be able to follow the player around the map, meaning it will need similar movement animations to the main character	NO
US_04	There is a visual queue on the screen that will track specific progression.	There are multiple types of progress – elevation, time or collection	NO
US_05	Within the menu screen the player sets the game mode and the characteristics of the game change	These characteristics could be unlimited health, unlimited resources and no enemies	NO
US_06	The player can destroy a block and it disappears from the game screen. This block will then show up with in the inventory	From the inventory screen there are multiple options the player can take which require use cases and user stories	NO
US_07	There is an option to craft items from the inventory screen. If the player has the required resources, they can craft the item and the resources are taken from their inventory	You can implement many ways to show what can be crafted with the resources provided.	NO
US_08	Have the player take damage, from then the health will decrease and then it regenerates once they stop taking damage		NO

Use cases

We started to think of use cases this week and setup some initial cases that will cover the basic mechanics of the game.

UC_01 Use case: Player starts game on different systems

Author: TW

Date: 27/10/2020

Modification date: 27/10/202

Purpose: the game can run and be played on different systems

Overview: First the player will boot up their system (Windows, Mac or Linux) and then open the game. If the code can compile and run on the system, the game starting screen will appear on a window else alternative: the code fails to run, and a system error will appear on their screen.

Cross reference: US_01

Actors: player

Precondition:

- The system must have all the game components downloaded and ready to run
- There must be a screen to show the game on
- Post condition: the system will access the code and run it
- The game is shown on the screen for the player to interact with

Normal flow of event:

Actor actions	System actions
1. The player will start up their system	2. the system will boot up and show the desktop
3. The player will click and run the game file	4. System will start to run the code
6. Player then interacts with the game	5. If the code can be run the game window will pop up on the players system

Alternative flow of events:

- The code cannot run, the system will show an error message

Exceptional flow of events:

- The players system will not boot up

- The game window will pop up but the player cannot interact with it and the player will then close the game window

UC_02 Use case: Player destroys a block on the map

Author: TW

Date: 27/10/2020

Modification date: 27/10/2020

Purpose: A major part of the game is allowing the user to interact with the map and change it. This means destroying the generated blocks and picking it up as a resource.

Overview: First the player will decide they want to destroy a certain block in the game. Then they will attack the block until it is destroyed. The block will not be shown on the map and the player will add that block to their inventory.

Cross reference: US_06, US_04

Actors: player

Precondition:

- There must be a block to destroy
- The player must have an object that can cause damage to the block (fist, pickaxe, sword)

Normal flow of events:

Actor actions	System actions
1. The player will begin to attack the block (mine) 3. The player will keep attacking the block 7. Player can then repeat cycle with new blocks	2. The map generated block will take damage 4. The map generated block takes damage until it reaches zero 5. The block will disappear 6. The block type will appear in the players' inventory

Alternative flow of events:

- The player destroys a block they are standing on. This will mean they will fall

when it is destroyed until they encounter another block. They still gather the resource.

Exceptional flow of events:

- The player will stop attacking the block halfway, meaning the block is left with half health and still present on the map.

UC_03 Use case: Player places a block on the map

Author: TW

Date: 27/10/2020

Modification date: 27/10/2020

Purpose: A major part of the game is allowing the user to interact with the map and change it. This means placing blocks onto the map and taking it out of the players' inventory

Overview: First the player will decide they want to place a certain block. Then the game will check if they have the resource to place. If they do, a block will appear on the game in the position they wanted to place

Cross reference: US_06

Actors: player

Precondition:

- There must be a map block to place the new one on
- The player must have the desired block to place within their inventory

Normal flow of events:

Actor actions	System actions
1. The player will enter the inventory	4. The map notes the request to place a block on an existing block
2. The player must have the desired block to place within their inventory	5. The map generates the new block on the existing block
3. The player attempts to place the block on the map	6. The block type will be taken from the players' inventory
7. Player can then repeat cycle with new blocks	

Alternative flow of events:

- The player does not check their inventory, so whatever block is assigned will be placed

Exceptional flow of events:

- The player does not have any of the block type in their inventory, meaning no block will be placed

UC_04 Use case: Player crafts an item from the inventory

Author: TW

Date: 27/10/2020

Modification date: 27/10/2020

Purpose: As the player gathers resources in the game, they will be able to create items such as a pickaxe. They will use these items to interact with enemies and the map.

Overview: First the player will decide they want to craft a certain item. Then the game will check they have the required resources for the item. If so, the item is crafted and the resources are taken from the players inventory. The item is also added to the inventory.

Cross reference: US_07

Actors: player

Precondition:

- There must be the correct amount of resources to craft the item

Normal flow of events:

Actor actions	System actions
1. The player will go to the inventory screen 2. The player will go to the craft menu 4. The player will choose an item which they have resources for 7. Player can then equip the item	3. The game will show what items can be crafted and what cannot 5. The game will generate the item for the player and add it to their inventory 6. The resources required for the item will be taken from the players overall resources

Alternative flow of events:

There are none

Exceptional flow of events:

- The player will attempt to craft an item where they do not have the resources.
A message will be displayed stating insufficient resources

UC_05

Use case: Player controls the movement of the character

Author: TJ

Date: 27/10/2020

Modification date: 27/10/2020

Purpose: the player can control the character to move up, down, left and right

Overview: The player will press arrow keys and the block (the character) will move accordingly.

Cross reference: US_01

Actors: player

Precondition:

- The system must be able to show the player's current position
- The player needs to have a way of giving input (e.g. key press)
- The input needs to be interpreted by the system
- The system needs to respond by showing the updated position

Actor actions	System actions
1. The player will press a key 4. The player will keep giving input based on visual feedback and goals	2. The system will interpret the input 3. The system will update and show the player's current position

Alternative flow of events:

- There are none

Exceptional flow of events:

- The system not able to interpret player's input
- The system does not update or show the player's position in time

UC_06

Use case: As the players sanity goes up his sidekick will offer advice

Author: TW

Date: 27/10/2020

Modification date: 27/10/2020

Purpose: the game can run and be played on different systems

Overview: During the game the player will go insane due to multiple conditions. This will often be due to parameters that are based on the player struggling to pass a certain stage of the game. Therefore, to help the player, the sidekick (Pebble) will offer advice to the certain stage of the game.

Cross reference: US_02

Actors: player, sidekick

Precondition: The player must be a certain level of sanity to retrieve a certain amount of helpful information. As the sanity goes up so does the amount of information given.

Post condition: The system will assess the sanity percentage. If it reaches the required level, information will pop up on screen coming from Pebble.

Actor actions	System actions
1. The player will start the game	2. The system will set sanity to zero percent
3. Players begin to play the game	5. System will constantly loop checking if the sanity level is at threshold for certain information
4. As they play the sanity percentage will increase	6. If sanity is high enough, system will display information coming from pebble (1min cycle)
7. Player sees information and keeps playing with advice	9. System will constantly loop checking if the sanity level is at threshold for certain information

Actor actions	System actions
8. Player sanity will decrease with progress	10. If sanity is below the threshold, then the information is no longer displayed.

Alternative flow of events:

- The sanity stays in a certain band, so level of information is always given
- Sanity jumps more than one band in one go, so the information given will skip one

Exceptional flow of events:

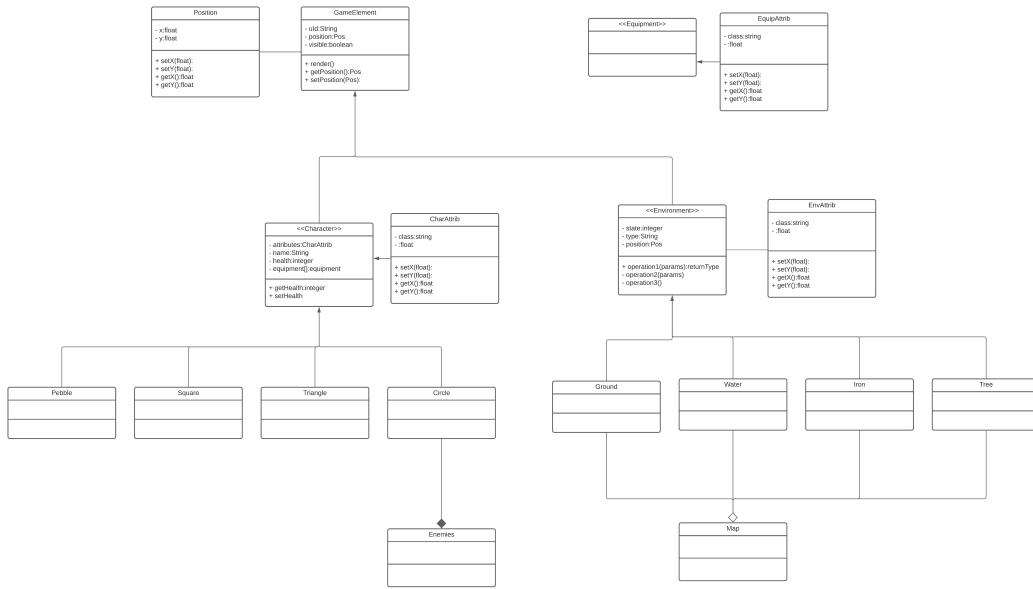
- The player sanity does not increase meaning that no information is given throughout the game.

Use cases testing

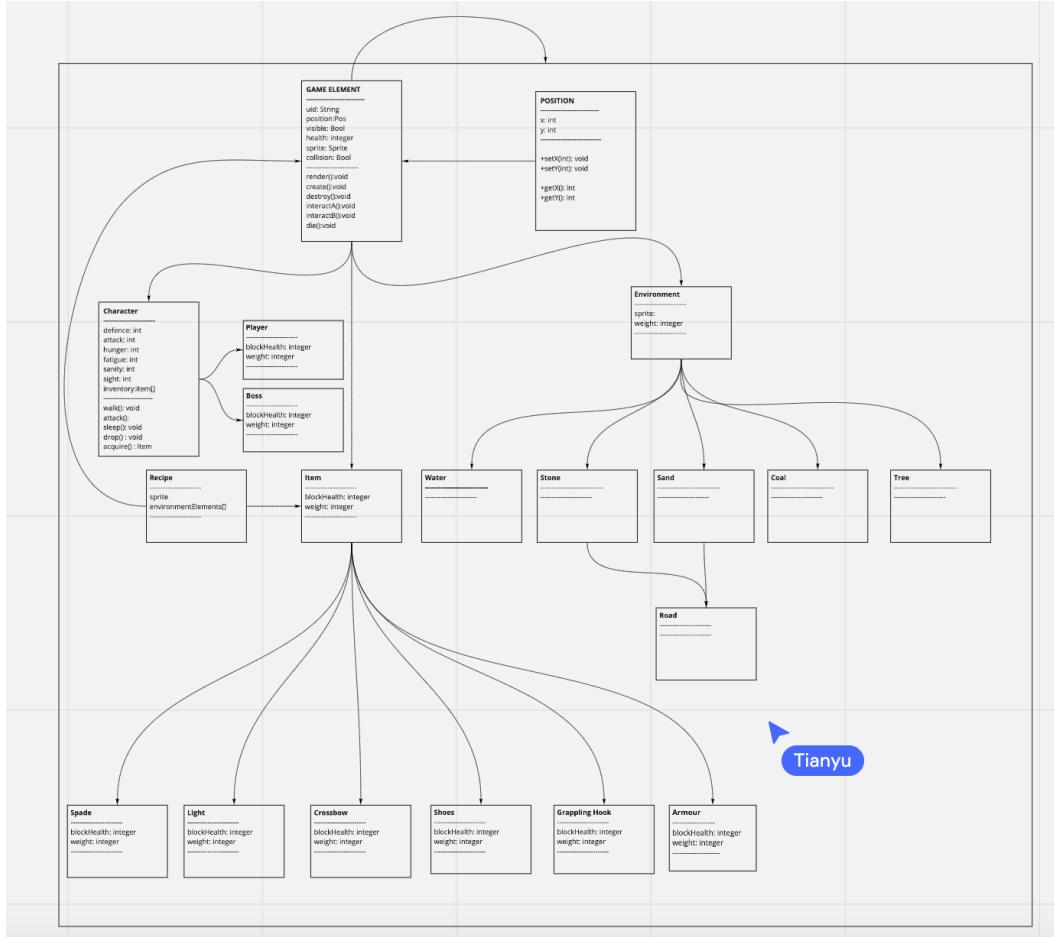
- UC_01**
- UC_02**
- UC_03**
- UC_04**
- UC_05:**
 - The player will press one of the directional keys which will move the character in that desired direction.
 - The character will move by the specified amount of pixels in the system
- UC_06:**
 - The sanity level will rise with the condition set
 - When the sanity level hits the threshold information function is triggered
 - When the information function is triggered, Pebble will display information
 - Information can be seen coming from pebble, not random on the screen
 - At higher sanity levels you don't have lower tier information given with the said tear

Software design documentation

During this week we had a meeting to discuss how the different parts of the game will interact with each other and create a UML diagram. Before the meeting a couple of members of the group had thought out potential class interactions which was used as a starting point for the discussion.



From this discussion the final initial UML was structured. As stated before this took about 2 hours and CRC cards were not used. The use of each member generating CRC cards would have likely reduced the time to create the UML diagram, as well as creating a better understanding of the diagram.



User interface design

Design implement	Version
There will be a starting menu screen with the game settings, load game, continue and new game	2
In the settings tab there will be choices for audio, visuals, and controls	1
Once in the game there will be character that can move and interact with the whole map	1
The player can go to an in game menu with save game and achievements	1
The player can access an inventory screen which has options for equipment	1
In the equipment tab there will be options for view, craft, and delete	1
When the player had successfully completed the game they will be sent to the credits and the starting screen	2

Week 3: 30th October - 5th November

Process

Weekly Overview

As UML class diagram was developed last week, game development followed the order and relationship between different classes, which made the process more structured. Moreover, team members experimented with the coding platform and created a character (a block) that can move. For this week, the team continued to use Miro board to create a MVC model based on the different classes generated in the UML diagram which further defines how the three main components of the game interact with each other: controller, map and elements. The sprint tasks were done according to the model: a map was created, players can see their input as they control the character. Camera view was implemented that always had the character in the centre of the screen as it moved around.

Lessons learned: The class diagram needs to be based on use cases. Internal user stories need to be created when developing game features so that user requirements will be fulfilled and the whole process should be documented for future reference. Moreover, it may be better to set a small rather than a hard-to-reach goal since each sprint is only one week long. It is also helpful to move the tasks which had been done on GitHub in time so that the sprint progress is up-to-date. The team may function better when taking course load into consideration while assigning tasks so that distribution of the development tasks can be more reasonable and effective.

The presentation created for the customer meeting on the 6th November is linked here, which will have GIFs for some of the artefacts in this weeks documentation: [Presentation](#)

Meeting Minutes

Customer meeting debrief and sprint planning:

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

A sprint planning meeting was held right after the customer meeting, in which the team added tasks to sprint backlog according to the user requirements and additional features that could be added to the game. These informal discussions of features were later turned into the use case as shown in this weeks documentation. The customer expressed a preference toward a high-resolution art style for the game so a task to find sprites that would fit the style was created and assigned. The customer also wanted to see a variety of backgrounds and scenes while playing the game. Therefore, map generation was added to this week's sprint tasks. In addition to the customer's suggestions, the team also decided to work on menu screens since they are a fundamental part of a game. Another component of a game is sounds so some member were assigned to find background music and short sound effects that fit with the game theme. In terms of character movement, since a map will be generated which will influence where the character can go, gravity and collision are needed to make the movements seem more real (e.g. the character jumps down - gravity; cannot pass the edge of the map - collision).

MVC model - Controller, Map and Elements:

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

MVC:

- User uses controller->Controller manipulates model->Model updates view->View sees user
 - Need a model class- the map of all the elements
 - View is the camera - main view takes input and send to controller
 - Controller is all of the characters and elements/methods of the elements Move:
- View reads position from model
- Player in the model gives player position to the main view->Render(): A key press in main view->sends to controller which interprets the key and then change player position in the model Map:
- Random map generator
- Menu->seed->generate map
- Every bit of the map has coordinate, get type of environment from coordinate
 - Map has get element at position function, returns game element
 - Destroy-return void
 - Getplayer(): return position

- private: horizontal & vertical size, seed; have setters & getters Elements
- Elements relative to the player position
- Elements in environment can become items that can be picked up (update map), interact with player
- Call items in the range of the player
- Every element is on the map, view just takes info from the map
- Action 1 hit-> takes element from the environment and reduces the health

Questions: - Main view has to be re-rendered when the player gets to the edge of the screen - controller or model? - Each class has a method called view? - How to implement main map controller?

Sprint review and presentation creation: Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

- A map could be generated that had a black background with edges, different obstacles and materials.

```
vim (Vim)          %1          ..ungeonCrafter (zsh)

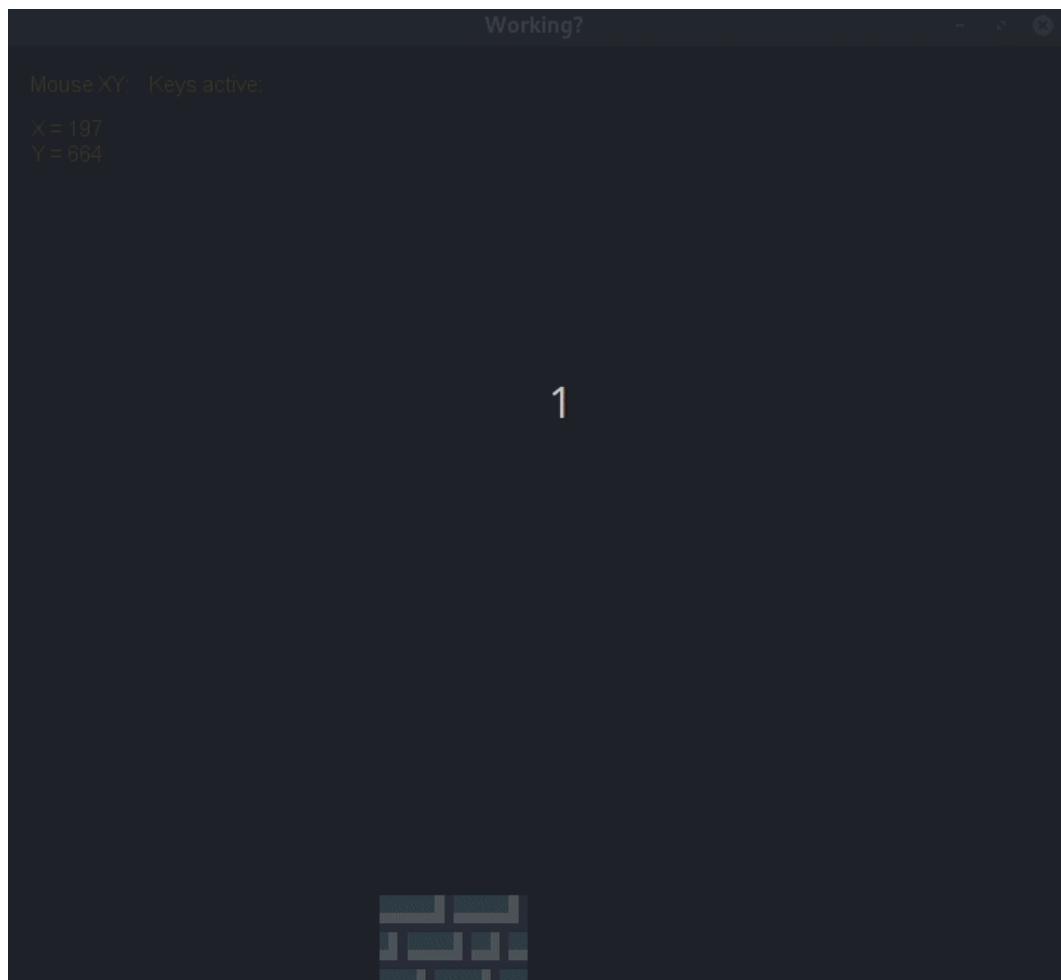
tile/water_6 (alias of tile/water_1)
tile/water_7 (alias of tile/water_1)
tile/water_8 (alias of tile/water_1)
tile/water_9 (alias of tile/water_1)
tile/water_10 (alias of tile/water_1)
tile/water_11 (alias of tile/water_1)
tile/water_12 (alias of tile/water_1)
tile/water_13 (alias of tile/water_1)
tile/water_t_3 (alias of tile/water_t_2)
tile/water_t_4 (alias of tile/water_t_1)
tile/water_t_6 (alias of tile/water_t_5)
Packing.....
Writing 512x128: ./textures.png

Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.7/userguide/command\_line\_interface.html#sec:command\_line\_warnings

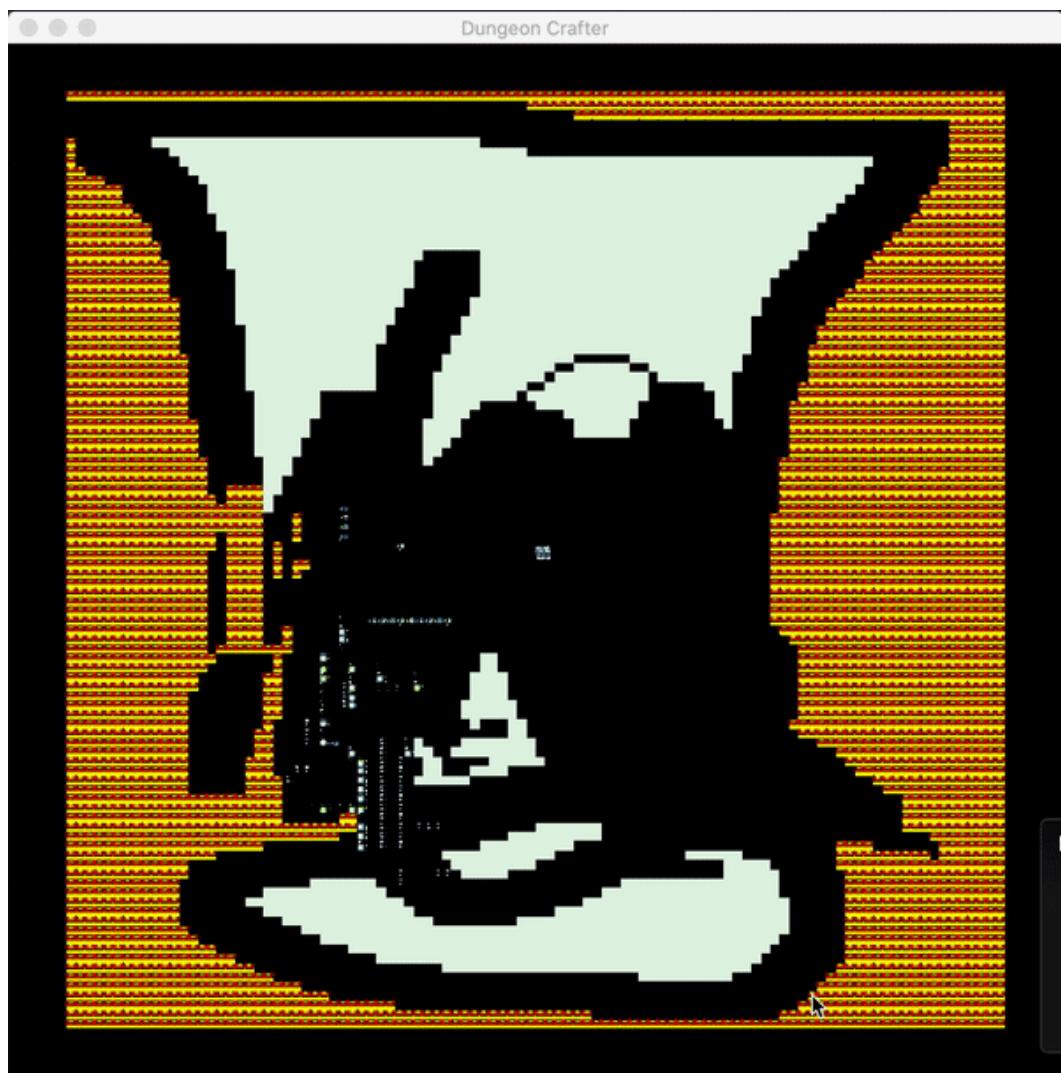
BUILD SUCCESSFUL in 27s
5 actionable tasks: 1 executed, 4 up-to-date
→ DungeonCrafter git:(map-using-glib) gradle desktop:run
```



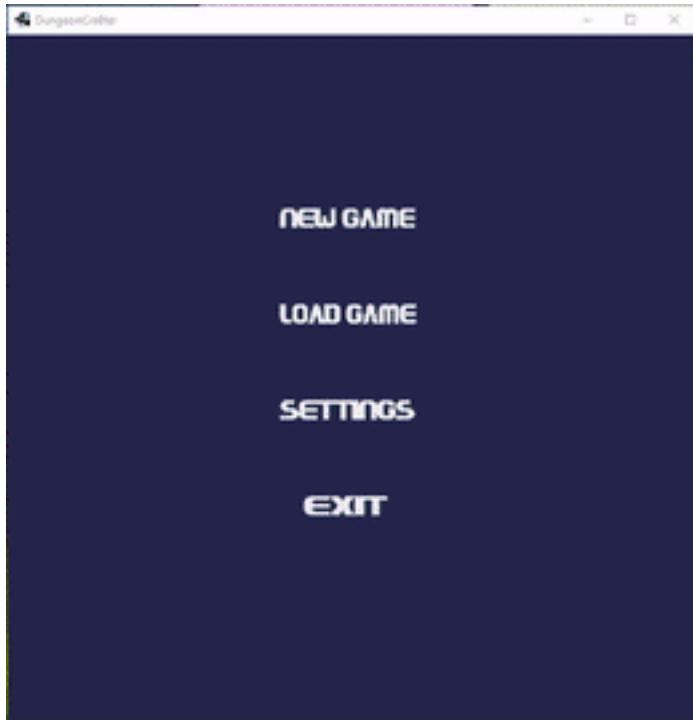
- User input can be shown as current position that constantly changes according to the user's control.



- Camera view was implemented to always have the character in the centre of the screen.



- A menu was created with basic functions such as new game, load game, settings and exit.



We were not able to achieve the gravity this week and the collision was still not properly implemented, so would both need work in the next sprint week. One problem which meant we did not achieve what we wanted was the coursework deadlines happening at the same time. This is further discussed in the exception handling section.

Backlog

Product backlog:

- Generation of block map (side on view)
- Sprite class and art style
- acquire sounds and implement
- The blocks of the map can be interacted with by the player
- A player is spawned on the map
- The player can interact with the blocks (destroy and create)
- The player can collect these resources in an inventory
- The players resources can be used to craft items (pickaxe, sword, and light)
- The crafted tools will break over time
- The player will have a sidekick (bot) which will either help or race against the player
- There will be a timer shown to the player on the screen which runs down as they go through the game
- There will be tasks to complete in order for the player to escape
- The player can go through a door and the screen will change to top down view
- In the top down view, the player will fight bosses

Sprint backlog:

- Map generation
- Sprite class and art style
- Menu screens
- Keymapping/controls
- Acquire sounds and implement
- Find character and enemy sprites
- Gravity/Collisions

Completed:

- Find cross platform library
- Research on how GitHub version control works
- Java research
- Game development research
- Create a simple character (square)
- give this character up, down, left, and right movement
- Find a common naming system
- Design classes and create a UML diagram

- General game structure (features & menu files)
- Pebble backstory

Exception Handling

Many members had course deadlines this week so the team had to balance between keeping up with the progress of game development and finishing individual coursework. Sprint planning meeting was delayed until after a deadline that several members had for their coursework. Heading forward to the final week with multiple deadlines, we will have to think what is achievable by each member.

product contents

Customer interview

The customer suggested having a way to interact with the game by controlling the character so that the main character needs to be able to move according to user's input. After the team presented a general idea of the background story and some possible features for the game, the customer expressed an interest in the sanity level feature again (same as the last customer meeting). After looking at several proposed art styles for the game, the customer had indicated that the one with higher resolution could show better layers of the ground and material types. Related to that, the customer also wanted to play in different backgrounds so a wider variety of scenes need to be constructed with relatively high resolution.

User stories

ID	Version	Priority	Story
US_01	2	1	AS A player I WANT to be able to play a dungeon game on different platforms web platforms SO THAT I have choices for playing the game
US_02	2	4	AS A player I WANT to have a bot that will play for me SO THAT It can help me with playing through the game by offering advice
US_04	1	5	AS A player I WANT to be able to see my progress SO THAT I know if I am doing well at the game as I play
US_05	2	7	AS A player I WANT to have different choices between games modes and different paths to the end in each game mode SO THAT I have new scenarios to play
US_06	1	2	AS A player I WANT to interact with the blocks on the map SO THAT I can gather resources, place resources, and make new paths
US_07	1	3	AS A player I WANT to use resources to craft items SO THAT I can use them throughout the game to make tasks easier
US_08	1	6	AS A player I WANT to be able to regenerate health SO THAT I can survive longer
US_09		9	AS A player I WANT to see a high-resolution game style SO THAT the background, items and characters seem clearer
US_10		8	AS A player I WANT to be able to explore a range of different backgrounds SO THAT the game is more playable

User story testing

User story	ID	Test	Comments	completed
	US_01	Run code on different systems and see if code compiles	Web based systems on Windows, MacOS and Linux	YES
	US_02	A bot character is generated on the game screen. The bot aids the user with text aids	The bot will have to be able to follow the player around the map, meaning it will need similar movement animations to the main character	NO
	US_04	There is a visual queue on the screen that will track specific progression.	There are multiple types of progress – elevation, time or collection	NO
	US_05	Within the menu screen the player sets the game mode and the characteristics of the game change	These characteristics could be unlimited health, unlimited resources and no enemies	NO
	US_06	The player can destroy a block and it disappears from the game screen. This block will then show up within the inventory	From the inventory screen there are multiple options the player can take which require use cases and user stories	NO
	US_07	There is an option to craft items from the inventory screen. If the player has the required resources, they can craft the item and the resources are taken from their inventory	You can implement many ways to show what can be crafted with the resources provided.	NO
	US_08	Have the player take damage, from then the health will decrease and then it regenerates once they stop taking damage		NO
	US_09	Add sprites into the game	The sprites need to be suitable in terms of the theme of the game and overall art style	NO
	US_10	Add sprites, generate different types of ground and environment texture	The player should be able to go to different places with various environment such as river, woods, stone road	NO

Use cases

Existing Use cases:

Use case ID	Use case description	Tested
UC_01	Player starts game on different systems	NO
UC_02	Player destroys a block on the map	NO
UC_03	Player places a block on the map	NO
UC_04	Player crafts an item from the inventory	NO
UC_05	Player controls the movement of the character	YES
UC_06	As the players sanity goes up his sidekick will offer advice	NO

New Use cases: UC_07

Use case: Have high-resolution items, characters and environment

Author: TJ

Date: 30/10/2020

Modification date: 30/10/2020

Purpose: make the game more visually appealing and make the elements clearer

Overview: The player will open the game and see the scenes, characters and items as high-resolution.

Cross reference: US_09

Actors: player

Precondition: - Sprites need to be found for different ground and material types

Normal flow of event:

Actor actions	System actions
1. Open the game	2. Different elements are shown on the map

Alternative flow of events: - Some elements load slower than the others so some items are missing after the player starts the game. The system will have a loading icon and the character cannot move until all the elements are fully loaded

Exceptional flow of events: - Some elements cannot be seen on the map - The elements appear blurry

UC_08

Use case: Player can explore in different environments

Author: TJ

Date: 30/10/2020

Modification date: 30/10/2020

Purpose: Provide the player with various scenes to play in

Overview: The player controls character to walk around the map and different sections of the map shows different environments

Cross reference: US_10

Actors: player

Precondition: - Sprites need to be found for elements in the environment such as water and land

Normal flow of event:

Actor actions	System actions
1. Control the character to walk around 3. Character interacts with the environment	2. The system will show the environment continuously

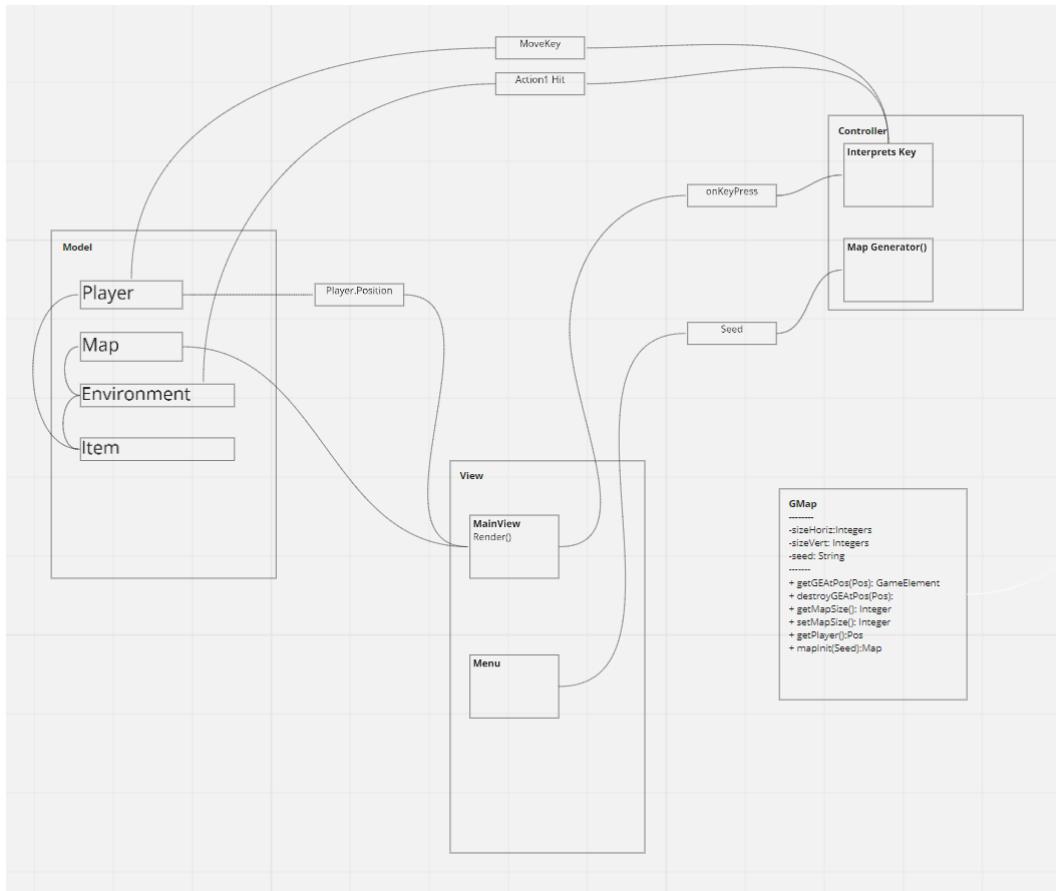
Alternative flow of events: - There are none **Exceptional flow of events:** - The player not able to enter certain parts of the map - The environment is shown in a blurry or confusing manner

Use cases testing

- UC_01**
- UC_02**
- UC_03**
- UC_04**
- UC_05**
- UC_06**
- UC_07**
 - The player will open up the game
 - The player will see the main character
 - The player will see blocks that can be destroyed/mined and can tell what they are and differentiate between types of blocks/materials
- UC_08**
 - The player will enter different sections of the map which contain various environment such as sand, water, stone, woods

Software design Documentation

MVC model was added based on the UML diagram to get a more specific relationship between controller, map and elements. No changes had been made to UML since last week.



User interface design

As the game becomes more complicated we decided it would be better to display the interface design in sections than one chronological table.

Character:

- Once in the game there will be character that can move and interact with the whole map [1]

Menu:

- There will be a starting menu screen with the game settings, load game, continue and new game [2]
- In the settings tab there will be choices for audio, visuals, and controls [1]
- The player can go to an in game menu with save game and achievements [1]

Sounds:

- There should be background music played at the menu screen and during the game. Background music can change along with the character's state such as sanity levels and become more intense during events like a boss fight [1]
- Sounds should be triggered when certain actions are performed (e.g. menu click, fight, mining) [1]

Inventory: - The player can access an inventory screen which has options for equipment [1] - In the equipment tab there will be options for view, craft, and delete [1]

Game flow:

- When the player had successfully completed the game they will be sent to the credits and the starting screen [2]

Week 4: 6th - 12th November

Process

Weekly Overview

More progress was made on making the character's movements seem more real by adding gravity and collision. The sounds are starting to be implemented as well for the menu screen. Character and enemy sprites as well as walking animation that fit with the style of the game were found. Moreover, game name and general storyline were established so that further game development can have a template or structure to refer to when it comes to making scenes, environments and characters.

What we learned: On top of the daily stand-ups we began to show visual progress within Microsoft Teams or direct them to pull an update from a branch to check if its functional and show progress. This on top of daily-stand-ups, helped give everyone a better idea what was going on throughout the week. The sprint was conducted in much the same format before, with us setting up the week following the customer meeting. Here we informally talked about the features that need implementing to progress the game, and after the meeting had finished they were written up into user stories and use cases.

The presentation created for the customer meeting on the 13th November can be found here, which will have GIFs for some of the artefacts shown here: [Presentation](#)

Meeting Minutes

Customer meeting debrief and sprint planning:

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

As with previous weeks, the team had a meeting to organize customer requirement mentioned in the customer meeting and plan the sprint accordingly. After seeing the progress of the game development which was having basic elements such as generating a map, character movement, user input and a main menu, the customer was satisfied of the initial state of the game. Nevertheless, the customer mentioned having a more detailed storyline. In the last sprints, we only had a general idea that the main character is in a world that requires crafting items and fighting monsters to survive along with a Bot called Pebble which is a small stone. As time passes by, the character's sanity changes which influences the process of the game. The customer was interested in this idea and wanted to see the game divided into episodes with a more detailed storyline and the character can gain various abilities or items as the storyline progresses. Therefore, the team needs to work on adding more content to the story such as character motivation and expand the background story. The following game development can be based on this storyline to create certain features. In addition to customer's suggestion, the team decided to have tasks that addressed the combination of previously developed features such as gravity, map and user input. Moreover, the sprint included finding character sprites and related animation.

Sprint re-planning and progress check

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

- Documentation: can combine the different markdown files into one pdf file in the end
- Sprites for character had been found that conform to user's preferred high resolution, considered actions that the main character need such as attacking, walking and mining Rock enemy sprite was found. The sprites can be changed in size as the game needs, such as reducing or increasing the rock enemy sprite size



- Found background music for different environments and scenes (e.g. boss fight)
- Implement sounds in reaction to input/an event, have background music playing when at main menu and when playing the game
- Add more details to storyline, and find a name for the game

- Found some potential main menu background design images
- A testing map can be created with different types of objects (e.g. trees, rocks...)
- Think about crafting in the game, can find more sprites
- Create settings menu
- Implement walking animation
- Implement destroying a block

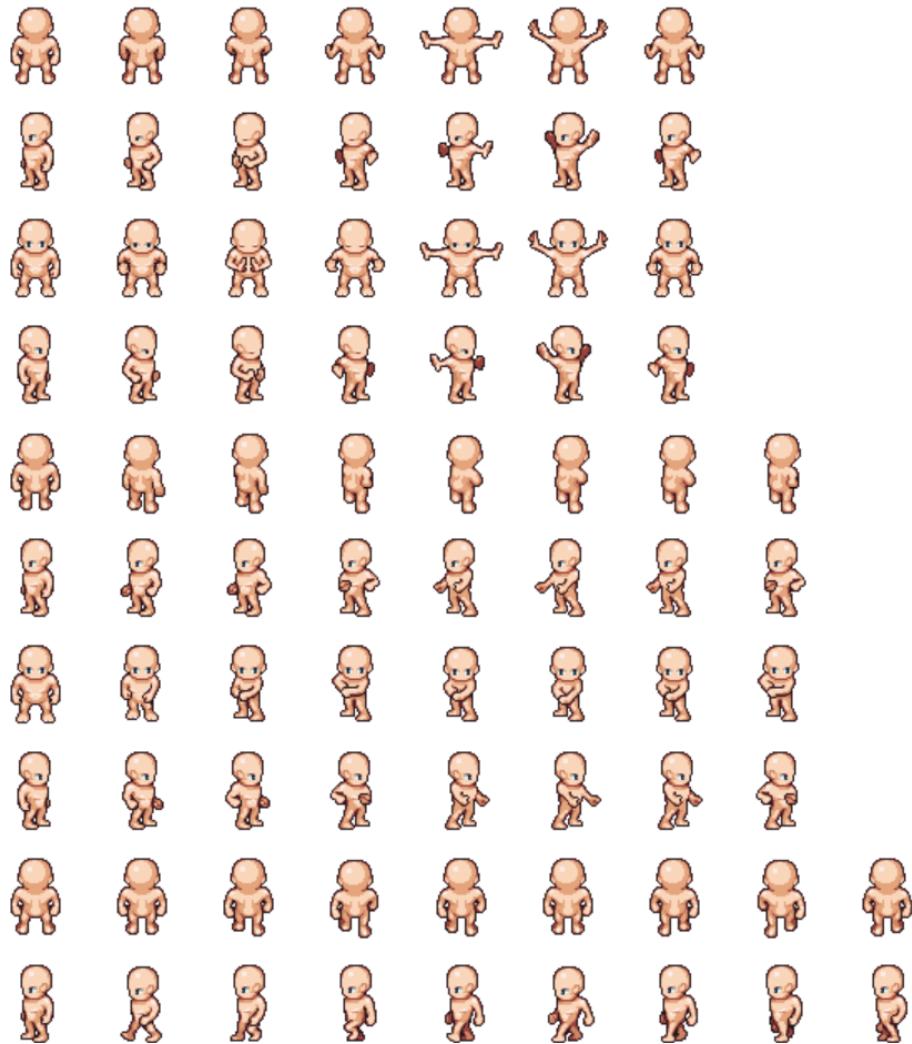
Sprint review and presentation creation

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

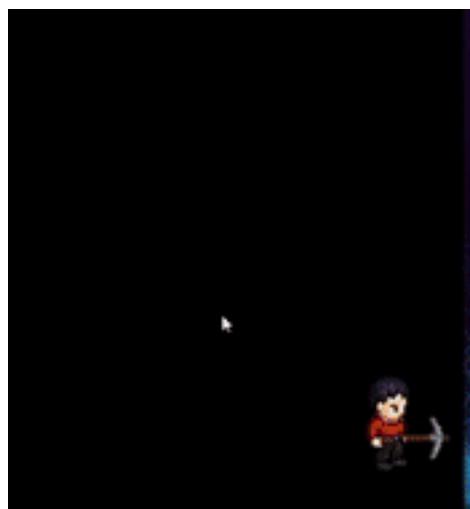
Collision and gravity had been implemented with the character (a block) and user input shown above.



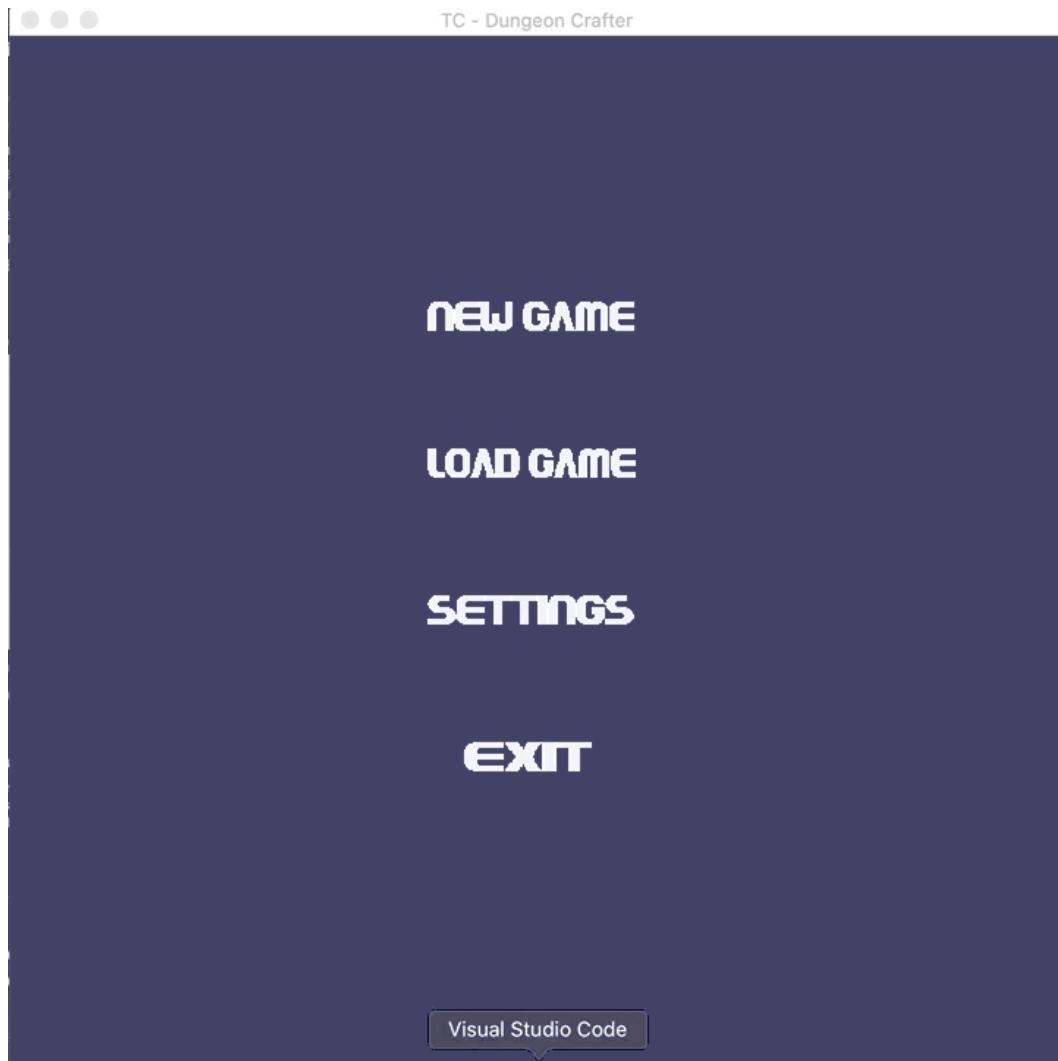
A new main character sprite has been chosen instead of the previous one because there are more potential actions and the sprite can be modified more flexibly.



The main character's walking animation had been created with a tool (pickaxe) in hand.



The map had been linked with the game, providing interaction from selecting an option (new game) and generating a map. A boss map was created.

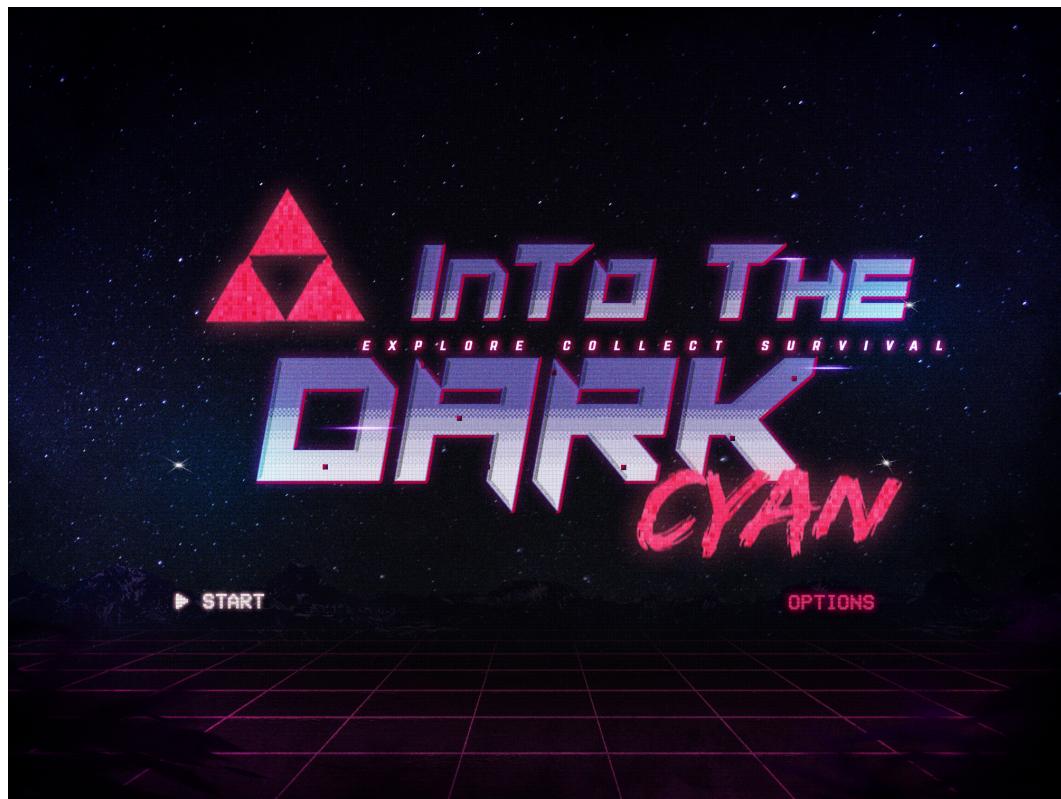


The team also decided on a game name: Into the Dark, which fits with the sanity feature of the game. Several menu background images were found that correspond to the theme of the game.









As the customer required, a more detailed storyline was generated with episodes that gradually unwrap plots that give the main character certain items or abilities with Pebble as the guide to tell the story and give instructions.

- Episode 1 - The Cave: Main character(X) is having a dream in which he/she fell down a hole, woke up and found himself in a cave. It's dark. X's heart rate increases and until he finds a torch and lights it up [sanity]. Then, X sees a stone and picks it up. X is scared so he imagines it to be a companion that talks to and helps him. X: "I'll call you Pebble." Pebble comes to life and follows X around.
- Episode 2 - Pebble & the Darkness: Pebble tells X that it is a small stone which was bullied by other bigger ones. They are in a world ruled by stones which absorb light. Since light sources are limited and slow to regenerate, the world will become dark sometimes. X has to find light in order to survive.
- Episode 3 - Stone Monsters & Weapons: No one is allowed to leave this world. There are stone monsters guarding paths/doors. However, Pebble had had enough and was trying to find a way out when it met X but it was too weak to fight against others. X and Pebble plan to escape the underworld together. X starts to collect materials for weapons (e.g. wood, iron, brass...) to defeat the stones.
- Episode 4 - Sanity: As X walks around, he/she finds that the environment is strange and can change when he/she gets to certain places (e.g. desert, grassland, river..., layers of dream?). The weather is unpredictable, sunny for now and yet it rains hard later. The days and nights can rotate at a random speed. This is different from the world X is used to so he constantly feels unsettled. During the night, it gets dark and X starts to panic. Pebble talks more to X (all imagination to comfort X) and being less/more helpful. The visibility is limited to a small circle around X so he/she needs to find torch or light again to calm himself/herself or else the sanity will continue to drop [heart rate becomes faster, the environment gets blurry, X moves slower?]. The sanity may also drop due to more enemies and X's mood or progress [time constraint].
- Episode 5 - The Final Battle: X sees a light spot far away, believes it's the exit and walks towards it. As X and Pebble approach the exit, more and more stones appear and start to accumulate in front of them and turn into a massive monster that completely blocks the exit. After a fight, the monster breaks into pieces and disappears, a large amount of light goes in via the exit, the world is lightened up.
- Episode 6 - The End: In the end, X escapes the underworld, wakes up and sees "Pebble" on his bed (lifeless). The darkness ends and the sun shines through the window. "What an experience".

Sound effects and game background music had been further selected for implementation.

backlog

Product backlog:

- Generation of block map (side on view)
- The blocks of the map can be interacted with by the player
- A player is spawned on the map
- The player can interact with the blocks (destroy and create)
- The player can collect these resources in an inventory
- The players resources can be used to craft items (pickaxe, sword, and light)
- The crafted tools will break over time
- The player will have a sidekick (bot) which will either help or race against the player
- There will be a timer shown to the player on the screen which runs down as they go through the game
- There will be tasks to complete in order for the player to escape
- The player can go through a door and the screen will change to top down view
- In the top down view, the player will fight bosses

Sprint backlog:

- Collision
- Implementing sounds
- Implement animation
- Find character and enemy sprites
- Documentation
- Game name
- A more detailed storyline

Completed:

- Find cross platform library
- Research on how GitHub version control works
- Java research
- Game development research
- Create a simple character (square)
- give this character up, down, left, and right movement
- Find a common naming system
- Design classes and create a UML diagram
- General game structure (features & menu files)
- Pebble backstory
- Map generation
- Main menu screen
- Gravity
- Camera view

- User input

Exception Handling

Some members have other coursework to finish so not much progress was made after the customer meeting. In the second meeting, the team further planned the sprint and started to get more work done after that.

product contents

Customer interview

The customer mentioned that more details of the game background story should be added. Episodes can be developed that advance the game progress with more main character capabilities introduced so that there are layers to the storyline. The rest of the comments were positive about the features that have been implemented .

User stories

ID	Version	Priority	Story
US_012	1		AS A player I WANT to be able to play a dungeon game on different platforms web platforms SO THAT I have choices for playing the game
US_022	4		AS A player I WANT to have a bot that will play for me SO THAT It can help me with playing through the game by offering advice
US_041	5		AS A player I WANT to be able to see my progress SO THAT I know if I am doing well at the game as I play
US_052	7		AS A player I WANT to have different choices between games modes and different paths to the end in each game mode SO THAT I have new scenarios to play
US_061	2		AS A player I WANT to control the character's movement and interact with the blocks on the map SO THAT I can gather resources, place resources, and make new paths
US_071	3		AS A player I WANT to use resources to craft items SO THAT I can use them throughout the game to make tasks easier
US_081	6		AS A player I WANT to be able to regenerate health SO THAT I can survive longer
US_091	8		AS A player I WANT to see a high-resolution game style SO THAT the background, items and characters seem clearer
US_101	9		AS A player I WANT to be able to explore a range of different backgrounds SO THAT the game is more playable
US_11	10		AS A player I WANT to see a more detailed storyline with episodes and gain skills along the way SO THAT I can be more motivated to play and see how an episode will end and how the character develops

User story testing

User story ID	Test	Comments	completed
US_01	Run code on different systems and see if code compiles	Web based systems on Windows, Mac and Linux	YES
US_02	A bot character is generated on the game screen. The bot aids the user with text aids	The bot will have to be able to follow the player around the map, meaning it will need similar movement animations to the main character	NO
US_04	There is a visual queue on the screen that will track specific progression.	There are multiple types of progress – elevation, time or collection	NO
US_05	Within the menu screen the player sets the game mode and the characteristics of the game change	These characteristics could be unlimited health, unlimited resources and no enemies	NO
US_06	The player can destroy a block and it disappears from the game screen. This block will then show up within the inventory	From the inventory screen there are multiple options the player can take which require use cases and user stories	NO
US_07	There is an option to craft items from the inventory screen. If the player has the required resources, they can craft the item and the resources are taken from their inventory	You can implement many ways to show what can be crafted with the resources provided.	NO
US_08	Have the player take damage, from then the health will decrease and then it regenerates once they stop taking damage		NO
US_09	Add sprites into the game	The sprites need to be suitable in terms of the theme of the game and overall art style	NO
US_10	Add sprites, generate different types of ground and environment texture	The player should be able to go to different places with various environment such as river, woods, stone road	NO

User story	ID	Test	Comments	completed
	US_11	Have a storyline and give character various abilities as the plots continue	The player should be able to find torches, mine, craft weapons as the story progresses	NO

Use cases

Existing Use cases:

Use case ID	Use case description	Tested
UC_01	Player starts game on different systems	YES
UC_02	Player destroys a block on the map	NO
UC_03	Player places a block on the map	NO
UC_04	Player crafts an item from the inventory	NO
UC_05	Player controls the movement of the character	YES
UC_06	As the players sanity goes up his sidekick will offer advice	NO
UC_07	have high-resolution items, characters and environment	NO
UC_08	Play in different environments	NO

New Use cases:

UC_09

Use case: The character's down movements

Author: TJ

Date: 06/11/2020

Modification date: 06/11/2020

Purpose: The player needs to be able to control the character to drop from a certain height in order to proceed in the game

Overview: The player will control the character to walk towards the edge of a platform and go down

Cross reference: US_06

Actors: player

Precondition: - Have a working map with different heights of platforms - Character can move

Normal flow of event:

Actor actions	System actions
1. Control the character to walk to the edge of a platform and further 3. Continue playing	2. The system will let the character drop to the next firm ground/platform

Alternative flow of events: - There are none

Exceptional flow of events: - The character not able to go down, stays on the same height - The character drops when getting close to the edge but the player is not ready to go down yet

UC_10

Use case: Collision

Author: TJ

Date: 06/11/2020

Modification date: 06/11/2020

Purpose: Define edges in the game map which the player cannot pass through, like a wall

Overview: The player will control the character to walk until reaching a wall and will not be able to go further

Cross reference: US_06

Actors: player

Precondition: - Have a working map with edges that has certain shapes

Normal flow of event:

Actor actions	System actions
1. Control the character to walk towards a wall	2. The system will stop the character to walk any further
3. Change walking direction	

Alternative flow of events: - There are none

Exceptional flow of events: - The system fails to detect character's position and the character goes out the edge of the map or gets stuck

UC_11

Use case: Main character gain skills as the story progresses

Author: TJ

Date: 11/19/2020

Modification date: 11/19/2020

Purpose: make the game layered so that the player will be introduced to different functions the game has and build up the concept of the storyline and what the main character can do

Overview: The player will listen to Pebble telling the storyline so that reasons for a particular ability are given and related tutorials are shown.

Cross reference: US_11

Actors: player

Precondition: - Storyline needs to be in place with main episodes with a beginning and an end

Normal flow of event:

Actor actions	System actions
1. Play until a certain point in the story	2. Pebble talks (pop-up dialog box), gives instructions of mining and crafting items
3. Follow the tutorial	4. Update the character status and the environment

Alternative flow of events: - The player may not want to see the tutorial due to familiarity with the game (e.g. second time playing). The dialog box can have a skip button so that the player does not need to see the tutorial again.

Exceptional flow of events: - The system does not provide tutorial when the player gets to a certain point. - The player sees the tutorial but does not gain the new capabilities or make progress.

Use cases testing

- UC_01**
- UC_02**
- UC_03**
- UC_04**
- UC_05**
- UC_06**
- UC_07**
- UC_08**
- UC_09**
 - The player will control the character to move downwards
 - The character will drop to the ground/platform on the next level
- UC_10**
 - The player will walk until reaching the edge of a map or a wall
 - The character stops
- UC_11**
 - The player gets to the start of a new episode
 - Pebble tells new parts of the storyline, gives tutorial and the character gains additional abilities/materials

Software design Documentation

Class: Position (see UC_01)

Responsibilities:

- Get an element (character/item/environment)'s position
- Set an element's position

Collaborators:

- Game Element

Class Game element (see UC_01)

Responsibilities:

- Get an element's position
- Provide interactions such as being attacked, destroyed or created

Collaborators:

- Recipe
- Item
- Character
- Environment

Class: Character (see UC_01)

Responsibilities:

- Make the character walk, attack, sleep and drop
- Acquire items

Collaborators:

- Game Element
- Player
- Boss

Class: Recipe (see UC_03)

Responsibilities:

- Show different elements of the environment which can be crafted

Collaborators:

- Game element
- Item

Class: Environment (see UC_04)

Responsibilities:

- Present different environments such as water, stone, coal and tree

Collaborators:

- Game Element

Class: Gravity (see UC_09)

Responsibilities:

- Make the character drop when there's no ground below him

Collaborators:

- Position
- Character
- Environment

Class: Collision (see UC_10)

Responsibilities:

- Make the character stop when reaching obstacle or the edge of the map

Collaborators:

- Position
- Character

- Environment

User interface design

Character:

- Once in the game there will be character that can move and interact with the whole map [1]
- The character needs to be able to drop to lower heights [1]

Menu:

- There will be a starting menu screen with the game settings, load game, continue and new game [2]
- In the settings tab there will be choices for audio, visuals, and controls [1]
- The player can go to an in game menu with save game and achievements [1]

Sounds:

- There should be background music played at the menu screen and during the game. Background music can change along with the character's state such as sanity levels and become more intense during events like a boss fight [1]
- Sounds should be triggered when certain actions are performed (e.g. menu click, fight, mining) [1]

Inventory:

- The player can access an inventory screen which has options for equipment [1]
- In the equipment tab there will be options for view, craft, and delete [1]

Game flow:

- When the player had successfully completed the game they will be sent to the credits and the starting screen [2]
- The map should have edges and certain sections that the user cannot pass through such as a wall [1]

Pebble:

- Dialog box at the bottom of the screen at the beginning of each episode (instructions and story-telling from Pebble), which can be skipped [1]

Week 5 documentation: 13th-19th November

Process

Weekly overview

From week 4 we had a greater sense of the direction of the story and as a result, was able to develop the game with great progress. From both sprints we were able to achieve: documentation for week 2-3, code refactoring, animation added to the main character, enemy implementation, pebble walk animation and pebble follows player, digging and block placement, countdown timer, arrow projectile, menu screen. Many of these were achieved in early implementation so would be continued in next week's sprint. Within the customer meeting, there was great interest in gravity implemented from the week prior, and how it could be changed throughout the map. Therefore, we planned how gravity can be implemented in the game under different circumstances.

From last week we attempted to have shorter sprints so that we could review what we achieved and plan a second sprint before the next customer meeting. This resulted in the first sprint from 13th- 17th and the second sprint from 17th- 19th. The results from both sprints did result in plenty of progress from the team and served a good update where other members of the team were on top of the daily stand-ups. However, the team agreed many of the tasks set ended up taking longer than the first sprint so it was not beneficial to have two sprints. Heading forward we will go back to one long sprint with an extra sprint meeting between the planning and sprint review if required.

As a group, we became concerned about what we set out to achieve and the time to do it. Therefore, the user stories and use cases were adjusted to their basic level. An example being the inventory being just full armour and weapon upgrade instead of having a crafting table to create different types of armour. These revised changes would give the basics of the game we wanted to promise to our customer. It was also apparent we were not putting in enough thought to prioritise our backlog and what needed achieving. Future weeks will focus on analysing what needs doing first and then focus on the added touches to features. To achieve this we would assess the previous use cases and user stories, instead of previous weeks where we informally talked about what has been done and needs completing next in the sprint planning meeting.

The presentation created for the customer meeting on the 13th November can be found here, which will have GIFs for some of the artefacts shown here: [Presentation](#)

Meeting minutes

Customer meeting debrief and sprint planning:

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

Following the costumer meeting we agreed that it would be much appreciated by the customer if we definitely included varying gravity in the game. So , we talked about implementing reverse gravity in the game in order to elevate the payer to new heights.

For the sprint we decided to focus on getting some core components of the game down: character sprite implemented, dig and placing blocks, pebble movement, sound, enemy projectiles, and menu screen. At the same time, the documentation was continued to be worked on. The sprint goal was to finish week 4 and 5.

First sprint review and second sprint plan:

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

At the start of the meeting we went over what has been achieved by each member of the group. This ended up being a mix of finishing what we aimed to do and not. After this, we discussed our concerns on what could be achieved by the end of this game development process. It was clear we had not thought enough previously on what is achievable by us as a group and that we can split task more effectively. The new revised changes were:

- Having an example map that we make so that there is a playable map for the player. Map will consist of 3 different types of staged areas and a side scroller boss area for the player to fight against
- The inventory will not have a crafting element, but just options to upgrade full armour set. This is based on if the payer has enough of a certain block (e.g. gold)
- One enemy type that will have a projectile. The enemies can have different armour and damage ratings

With all this we made our second sprint goals. These were to continue projectiles, digging and placing blocks, documentation. Then new targets were refactoring code and countdown timer.

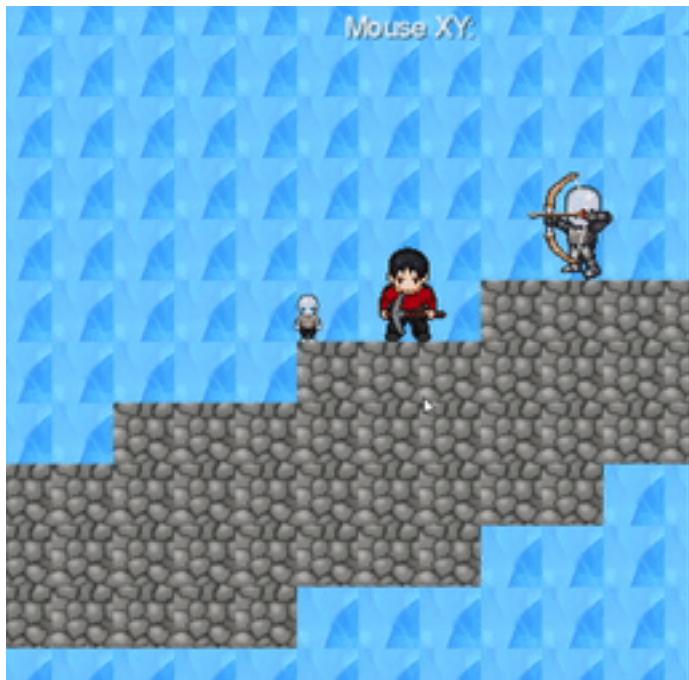
Second sprint review and presentation creation

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

As this was a big week and many of the previous sprint tasks were not completed, so we were excited to see the progress be put all together. All the previous sprint goals were achieved. One major change was the refactoring of the code, so now it is clearer, and any future developments should not cause as many bugs when being merged.

Below are all the features achieved, but make sure to check the presentation link in the weekly overview to see the GIF version

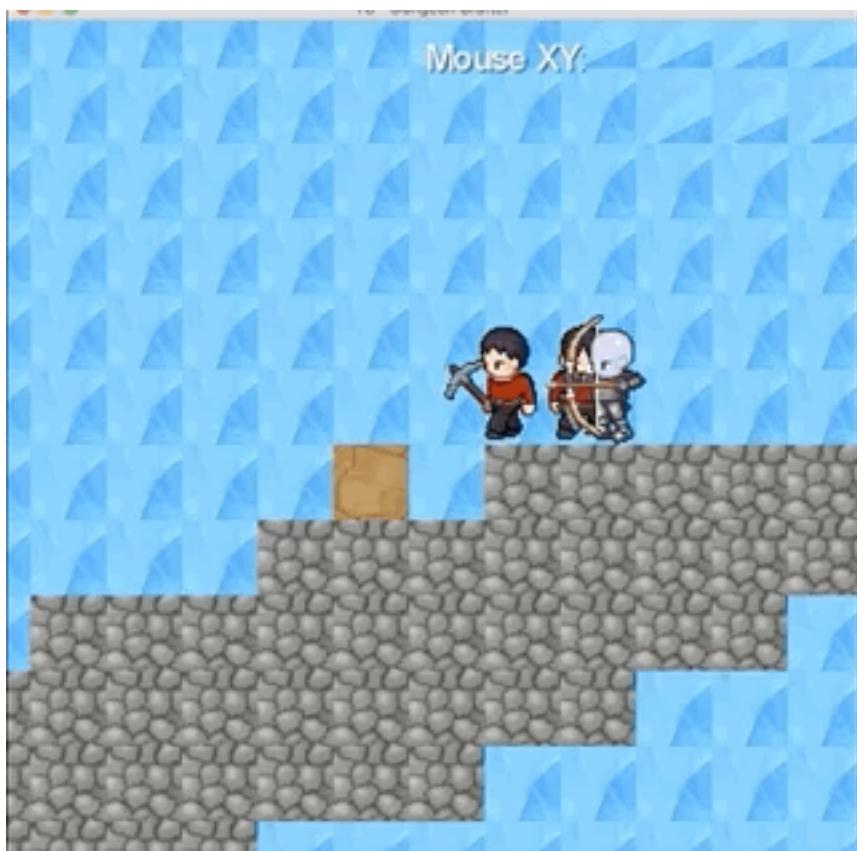
Game animations



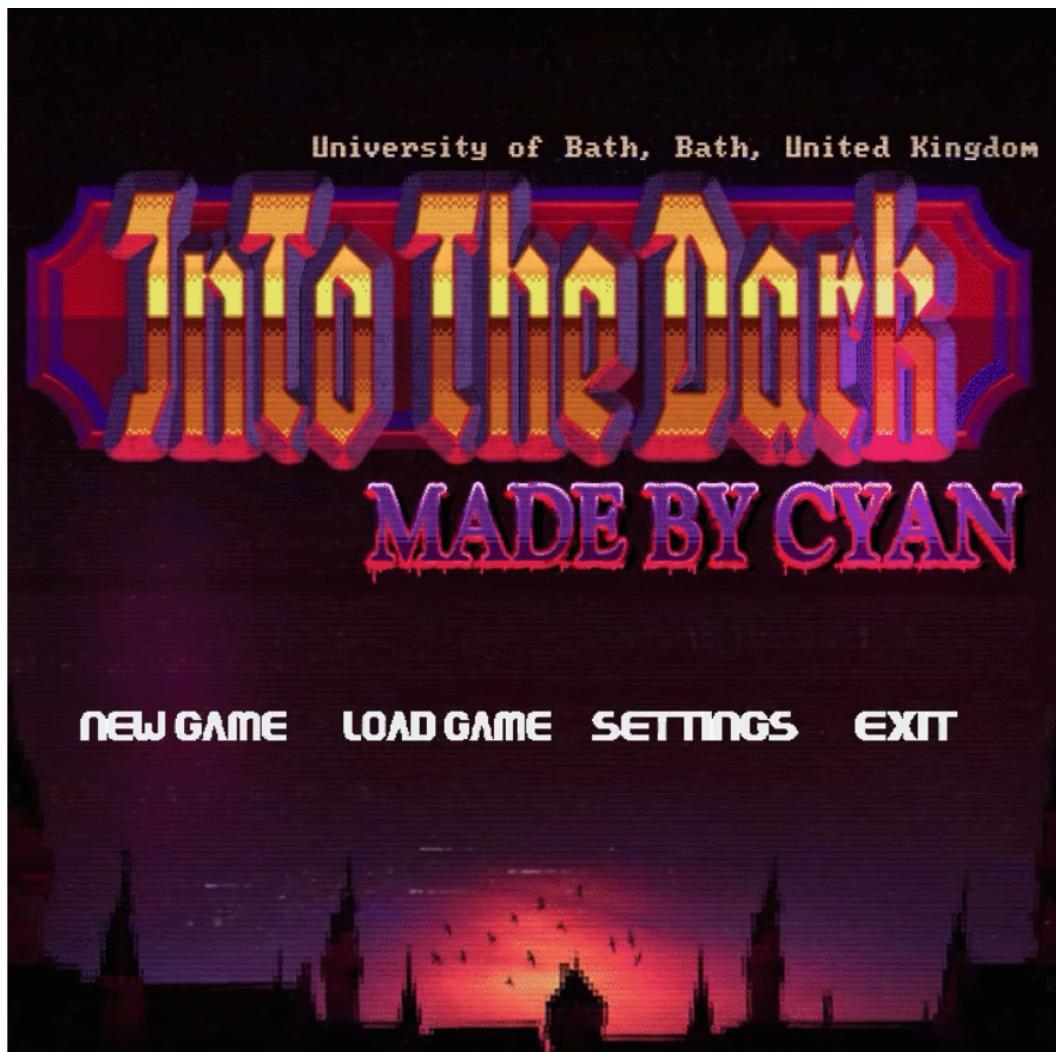
Enemy projectile



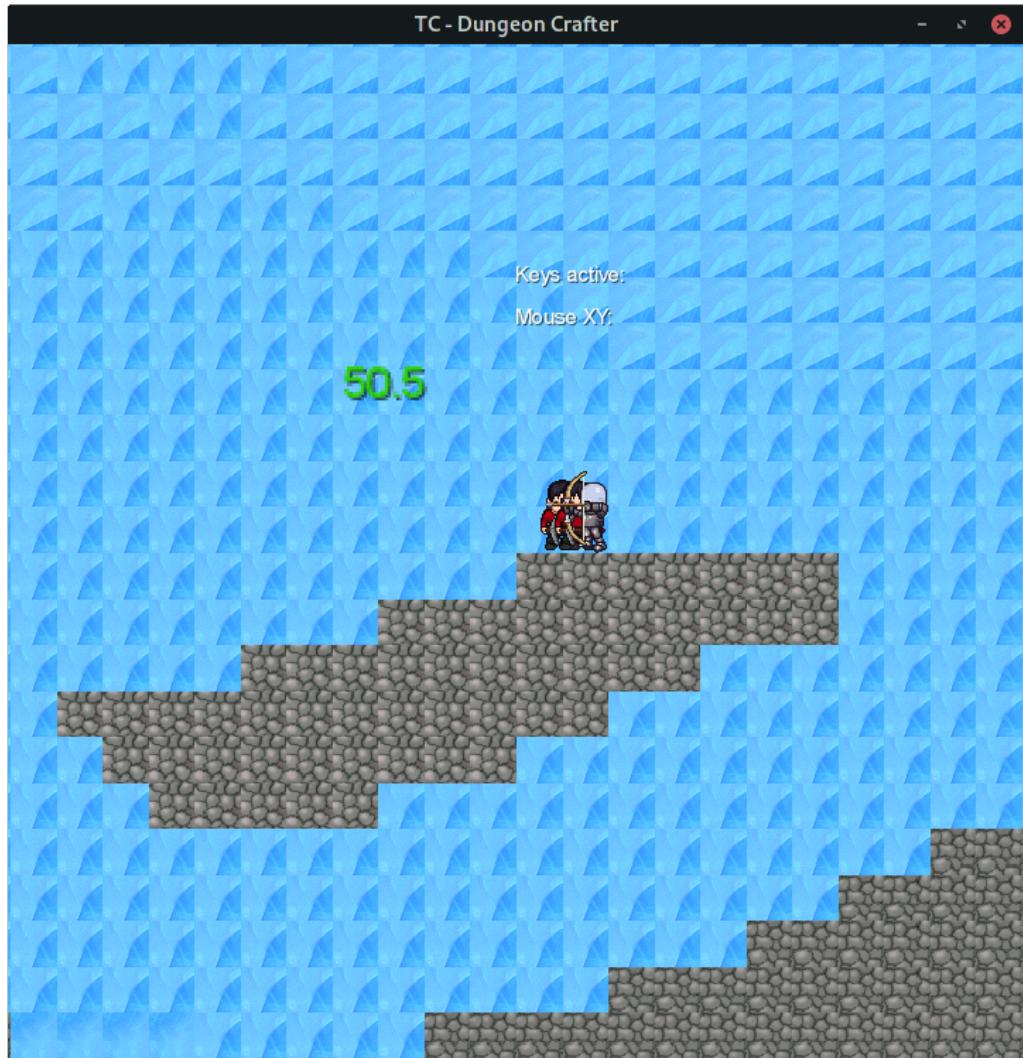
Mining and placing



Final menu screen choice



Timer



Code refactoring

Following our sprint review, we created the presentation for the customer. This followed the same template with reminding the customer of last week's presentation, then showing updated and finally telling them our future work. This was done on Markdown so that we could show GIFs, meaning the customer could visually see the changes.

Backlog

Product backlog:

This product backlog was revised this week as we discussed what is achievable in the time frame given. The items in bold are potential implementations if there is time.

- **The crafted tools will break over time**
- **There will be tasks to complete in order for the player to escape**
- The player will fight a boss in the final game scene to escape the dungeon
- There is an antigravity block allowing the player to rise until he is out of the vertical axis or collides with block
- **Random generation of block map (side on view)**

Sprint backlog:

- Implementing sounds
- Implement animations
- Example map that the game will load (horizontal view)
- Documentation
- A player is spawned on the map
- The player can interact with the blocks of the map (destroy and create)
- The player can collect blocks as resources which is stored in an inventory
- The players resources can be used to upgrade armour and weapon sets
- The player will have a sidekick (bot) which will help the player by offering advice
- Code refactoring
- Enemy projectiles
- There will be a timer shown to the player on the screen which runs down as they go through the game

Completed:

- Find cross platform library
- Research on how GitHub version control works
- Java research
- Game development research
- Create a simple character (square)
- give this character up, down, left, and right movement
- Find a common naming system

- Design classes and create a UML diagram
- General game structure (features & menu files)
- Pebble backstory
- Map generation
- Main menu screen
- Gravity
- Camera view
- User input
- A player is spawned on the map
- Collision
- Find character and enemy sprites
- Game name
- A more detailed storyline

Exceptional handling

There were no problems with this sprint or week.

Product content

Customer meeting

This followed the previous weeks. The customer was happy with the progress again and was curious about how we could develop some first feature implementations. The greatest curiosity was in the use of gravity throughout the game. Therefore, we discussed how best this could be implemented into the game. The game had still not sorted out the player climbing and ascending in the game, so we thought it would be good to have a block that could reverse gravity. This would mean the player could gain height in empty space, for that vertical game.

User stories

ID	Version	Priority	Story
US_01	2	1	AS A player I WANT to be able to play a dungeon game on different platforms web platforms SO THAT I have choices for playing the game
US_02	2	4	AS A player I WANT to have a bot that will play for me SO THAT It can help me with playing through the game by offering advice
US_04	2	5	AS A player I WANT to be able to see a timer go down SO THAT I know if I am doing well at the game as I play
US_05	2	12	AS A player I WANT to have different choices between games modes and different paths to the end in each game mode SO THAT I have new scenarios to play
US_06	1	2	AS A player I WANT to control the character's movement and interact with the blocks on the map SO THAT I can gather resources, place resources, and make new paths
US_07	2	3	AS A player I WANT to use resources to upgrade armour and weapons SO THAT I can use them throughout the game to make fighting enemies easier
US_08	1	9	AS A player I WANT to be able to regenerate health SO THAT I can survive longer
US_09	1	7	AS A player I WANT to see a high-resolution game style SO THAT the background, items and characters seem clearer
US_10	1	11	AS A player I WANT to be able to explore a range of different backgrounds SO THAT the game is more playable
US_11	1	10	AS A player I WANT to see a more detailed storyline with episodes and gain skills along the way SO THAT I can be more motivated to play and see how an episode will end and how the character develops
US_12	1	7	AS A player I WANT to be able to gain height in the map quickly SO THAT I can save time in reaching parts of the map
US_13	1	6	AS A player I WANT to battle enemies SO THAT the game has a challenge to it

User story testing

User story ID	Test	Comments	completed
US_01	Run code on different systems and see if code compiles	Web based systems on Windows, Mac and Linux	YES
US_02	A bot character is generated on the game screen. The bot aids the user with text aids	The bot will have to be able to follow the player around the map, meaning it will need similar movement animations to the main character	NO
US_04	There is a visual queue on the screen that will track specific progression.	There are multiple types of progress – elevation, time or collection	NO
US_05	Within the menu screen the player sets the game mode and the characteristics of the game change	These characteristics could be unlimited health, unlimited resources and no enemies	NO
US_06	The player can destroy a block and it disappears from the game screen. This block will then show up within the inventory	From the inventory screen there are multiple options the player can take which require use cases and user stories	NO
US_07	There is an option to craft items from the inventory screen. If the player has the required resources, they can craft the item and the resources are taken from their inventory	You can implement many ways to show what can be crafted with the resources provided.	NO
US_08	Have the player take damage, from then the health will decrease and then it regenerates once they stop taking damage		NO
US_09	Add sprites into the game	The sprites need to be suitable in terms of the theme of the game and overall art style	NO
US_10	Add sprites, generate different types of ground and environment texture	The player should be able to go to different places with various environment such as river, woods, stone road	NO

User story	ID	Test	Comments	completed
	US_11	Have a storyline and give character various abilities as the plots continue	The player should be able to find torches, mine, craft weapons as the story progresses	NO
	US_12	The player starts at a certain y position. Then in a time shorter than climbing blocks, the player is elevated to a higher y coordinate	There are different ways to achieve this, such as placing a block down which will change the gravity in the vertical plane	NO
	US_13	The player will come into contact with the enemy, where the enemy will begin to attack the player	The enemy could use projectiles or melee to attack the player. They could also be stationary or mobile.	NO

Use cases

Existing use cases:

Use case ID	Use case description	Tested
UC_01	Player starts game on different systems	YES
UC_02	Player destroys a block on the map	NO
UC_03	Player places a block on the map	NO
UC_04	Player crafts an item from the inventory	NO
UC_05	Player controls the movement of the character	YES
UC_06	As the players sanity goes up his sidekick will offer advice	NO
UC_07	have high-resolution items, characters and environment	YES
UC_08	Play in different environments	NO
UC_09	The characters down movements	NO
UC_10	Collison system	NO
UC_11	Main character gain skills as the story progresses	NO

New use cases:

UC_12

Use case: The enemy will fire a projectile at the enemy

Author: TW

Date: 13/11/2020

Modification date: 13/11/2020

Purpose: The player needs a challenge in the game. One part is battling enemies so they need to be able to identify and attack the player.

Overview: The player will come in range of the enemy where it will begin to fire projectiles targeted at the player. Either the player will die or the move out of the firing range. Then the enemy will stop shooting.

Cross reference: US_13

Actors: Player, enemy

Precondition:

- Character can move into range of the enemy
- The enemy is spawned within the map

Normal flow of events:

Actors actions	System actions (enemy)
1. Player will walk until they are in range of enemy	2. The enemy will pick up the player in the attack range
4. The player will take damage from the projectile	3. The enemy will begin to fire projectiles targeted at the player (projectile animation)
5. The player ends up dying and goes through death animation	6. The enemy no longer sees player in range so will stop firing projectiles

Alternative flow of events:

- The player will attack the enemy and kills it. The enemy will have death animation and no longer fire projectiles at the player when in range
- The player runs away from the enemy so is no longer in range. The enemy will therefore stop firing projectiles

Exceptional flow of events:

- The player builds a wall between them and the enemy. The enemy will stop firing projectiles even if it was the player was in range before as the enemy can no longer see the player

UC_13

Use case: The boss attacks the player

Author: TW

Date: 13/11/2020

Modification date: 13/11/2020

Purpose: The player needs a challenge in the game. One part is battling enemies so they need to be able to identify and attack the player. In this case this is a more powerful enemy.

Overview: The player will come in range of the enemy where it will begin to follow and attack the player through melee. Either the player will die or the move out of attack range. Then the boss will stop following.

Cross reference: US_13

Actors: Player, boss

Precondition:

- Character can move into range of the enemy
- The enemy is spawned within the map

Normal flow of events:

Actors actions	System actions (boss)
1. Player will walk until they are in range of boss	2. The boss will pick up the player in the attack range
4. The player will take damage from the attack	3. The enemy will begin to follow and attack the player (melee animation)
5. The player ends up dying and goes through death animation	6. The enemy no longer sees player in range so will stop following and attacking

Alternative flow of events:

- The player will attack the enemy and kills it. The boss will have death animation and no follows or attacks the player
- The player runs away from the boss so is no longer in range. The boss will therefore stop following

Exceptional flow of events:

- The player builds a wall between them and the boss. The boss will stop following the player even if the player was in range before as the boss can no longer see the player

UC_14

Use case: Player walks on block and reverses gravity

Author: TW

Date: 13/11/2020

Modification date: 13/11/2020

Purpose: The player needs to move up the dungeon to progress. This antigravity method will allow quick upward movement in the game.

Overview: This block will allow the player to quickly elevate in the map, adding more depth to the game and feel.

Cross reference: US_12

Actors: Player

Precondition:

- Antigravity block is placed within the map
- There is no block directly on top of the antigravity block

Normal flow of events:

Actors actions	System actions (boss)
1. Player will walk until they are on top of the antigravity block	2. The system will acknowledge player is in y plane of the block
4. The player will appear to move up on the map	3. The system will push the player upwards
5. The player will collide with a block above them within the map	6. The system will acknowledge the collision but will keep attempting to push the player upwards
7. The player will no longer move up	9. System will acknowledge the player is not in y plane of the antigravity block, so reverts gravity back to normal for the player
8. The player will move out of the y plane	
10. The player will fall down until they collide with a block	

Alternative flow of events:

- The player will place the block from there inventory. Then it follows the same flow of events
- The player leaves the y plane before they collide with a block above them. This will cause the plater gravity to revert back to normal

Exceptional flow of events:

- The player will place block above them as they go upwards which will cause them to stop in that direction

Use case testing

UC_01

UC_02

UC_03

UC_04

UC_05

UC_06

UC_07

UC_08

UC_09

UC_10

UC_11

UC_12

- The enemy is able to identify player when in range
- The enemy can fire projectiles when they pick up the player
- The projectile will disappear when it hits the enemy
- The projectile causes damage to the player

UC_13

- The enemy is able to identify player when in range
- The enemy will follow and attack the player
- The melee causes damage to the player

UC_14

- The system can pick up when the player is in the blocks y plane
- Gravity is reversed to a certain speed set
- The player does not travel through any blocks and does not get caught on block horizontally to them

Software design documentation

Class: antigravity (see UC_14)

Responsibilities:

- reverse the gravity on that block
- allows the character to fell this affect

Collaborators:

- Position
- Character
- Environment

Class: Boss (see UC_13)

Responsibilities:

- Player damage

Collaborators:

- Player health

Class: Enemy (see UC_12)

Responsibilities:

- Player damage

Collaborators:

- Player health
- gravity

User interface design

Character:

- Once in the game there will be character that can move and interact with the whole map [1]
- The character needs to be able to drop to lower heights [1]

Menu:

- There will be a starting menu screen with the game settings, load game, continue and new game [2]
- In the settings tab there will be choices for audio, visuals, and controls [1]
- The player can go to an in game menu with save game and achievements [1]

Sounds:

- There should be background music played at the menu screen and during the game. Background music can change along with the character's state such as sanity levels and become more intense during events like a boss fight [1]
- Sounds should be triggered when certain actions are performed (e.g. menu click, fight, mining) [1]

Inventory:

- The player can access an inventory screen which has options for equipment [1]
- On this screen the player can see amount of each block, craft option for armour and which block is being placed[2]

Game flow:

- When the player had successfully completed the game they will be sent to the credits and the starting screen [2]
- The map should have edges and certain sections that the user cannot pass through such as a wall [1]

Pebble:

- Dialog box at the bottom of the screen at the beginning of each episode (instructions and story-telling from Pebble), which can be skipped [1]

Week 6 documentation: 20th-26th November

Process

Weekly Overview

In Week 5, the team reconsidered what could be achieved in a limited amount of time left for this project and adjusted user stories and use cases to make sure that the most basic customer requirements get fulfilled first before spending time on other features. Therefore, this week the team focused on improving the existing features of the game such as enemy attack, destroy and place blocks, anti-gravity and map. Moreover, sprint tasks that connect different components of the game were accomplished such as Pebble speech and inventory screen, which allow players to see tutorial, storyline and collect materials for crafting after destroying blocks. Use cases were generated based on the previous user stories and use cases to have a more specific implementation.

Judging from last week, shorter sprints were not suited to the current situation since some tasks took longer to finish than planned. Thus, the team went back to having one long sprint with three meetings for this week. As a result, this week's sprint tasks could be achieved on time and there was no rush to finish them in a short period.

The presentation created for the customer meeting on the 13th November can be found here, which will have GIFs for some of the artefacts shown here: [Presentation](#)

Meeting Minutes

Customer meeting debrief and sprint planning

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

In the debrief, the team decided to have more features from the storyline implemented such as Pebble speech based on what the customer suggested. The grappling hook idea was interesting but the team did not think there is enough time left to do that and it can be hard to be implemented so the idea was recorded but not put into the product backlog. It would be attempted if other features have been developed to a satisfying level.

After discussing the customer requirements, the team continued talking about how to improve some existing features of the game. The current background of the game is blue, which may go against the narrative of the character being in a cave. Therefore, a darker background is needed. Moreover, since there is attack animation which shows an arrow being shot at the main character, the character should have a health bar that decreases with each attack received from the enemies. Now the arrow goes through the character so the animation needs to be shown in a way that the arrow hits the character and disappears.

Sprint planning

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

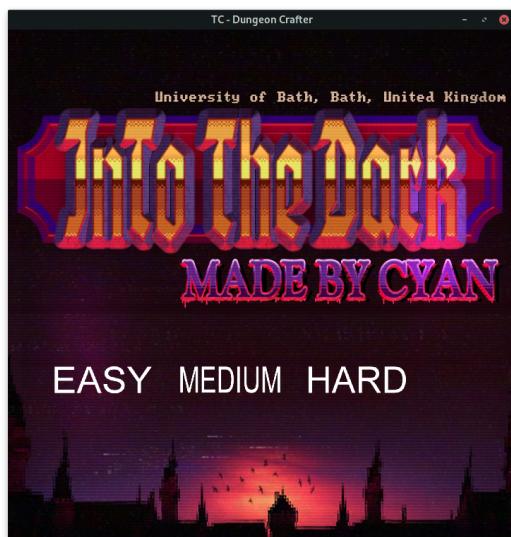
- Markdown can be converted to a pdf file, just need to format style sheet
- Has an example horizontal map, needs to get the block into correct sizes
- Block health was added
- Change background colour
- Enemy attack should have a refined animation: arrow projectiles that aim for the main character, and shows the character's health decreases
- Anti-gravity: can have sections of the map which allow the character to get to a higher position quickly
- Pebble dialog box that gives players instructions
- Inventory screen is needed so that the players can see how many resources have been gathered
- In-game buttons are needed to give access to the main menu and inventory

Sprint review and presentation creation

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

Documentation can be converted from Markdown to PDF so a PDF file will be created after all the sprint documentations are uploaded and revised to a satisfying level without worrying about formatting the documentation according to the requirement in the end.

Difficulty level is added to the main menu screen so that the players can choose from three levels of difficulties. Moreover, an in-game menu was added which displays the inventory screen: it has three types of materials along with how many the player currently has.



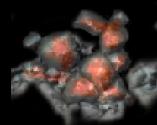
Back



Dirt: 6



Stone: 0



Iron: 0

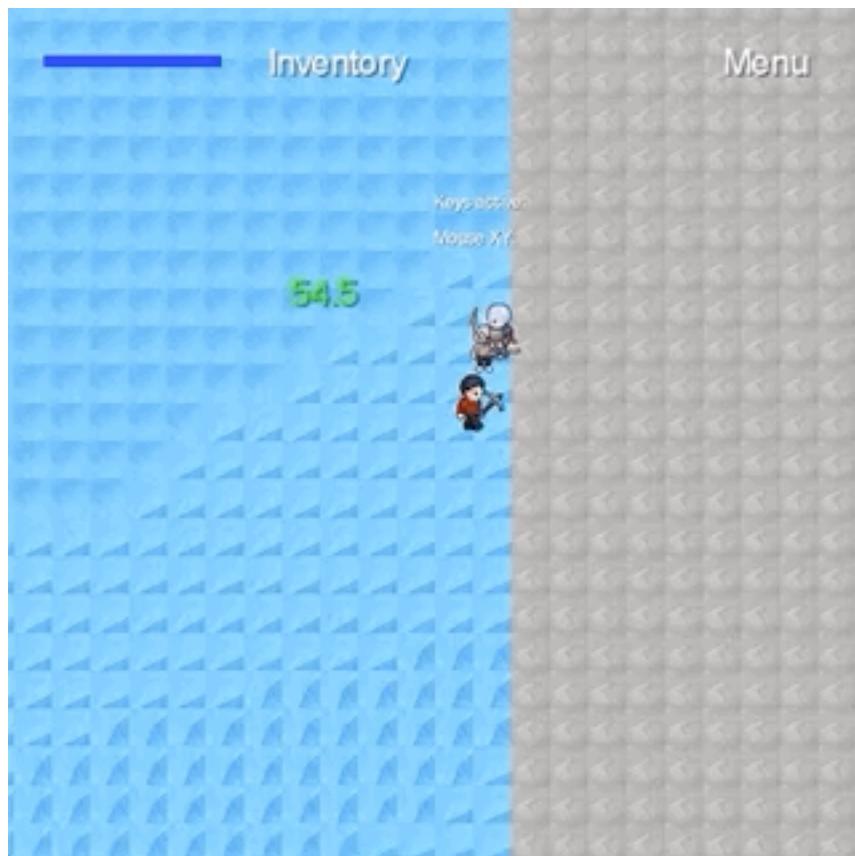
Choose a block to craft with:



Pebble speech is shown by a speech bubble above its head.

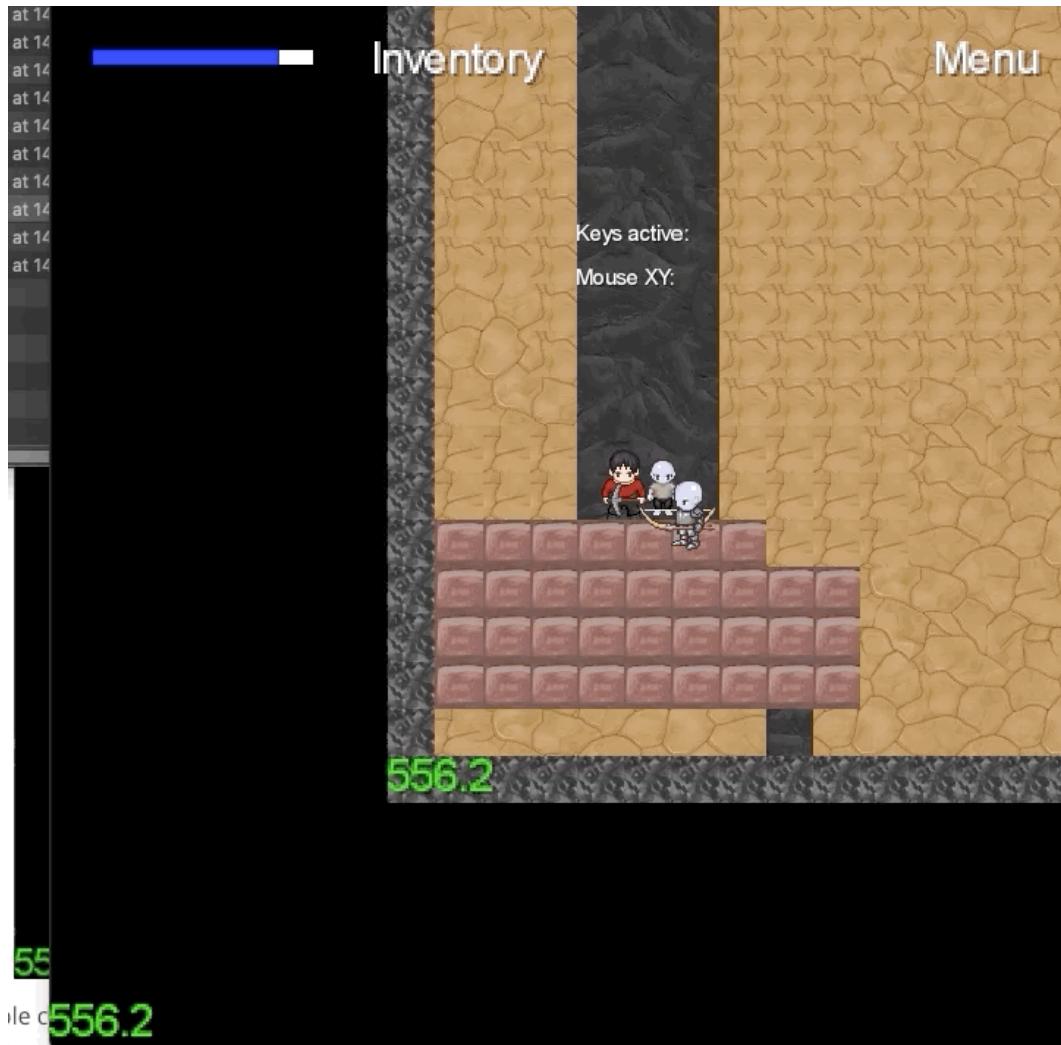


Anti-gravity upward movement has been implemented for certain sections of the map.



A bigger map was generated with an underground theme (black background which shows a dark environment). HP and properties of blocks inside the map have been implemented.

Enemy attack projectiles make the arrow fly towards the character and character's health decreases are shown by a health bar



Backlog

Product backlog:

- The crafted tools will break over time
- There will be tasks to complete in order for the player to escape
- The player will fight a boss in the final game scene to escape the dungeon
- Random generation of block map (side on view)
- Have a map that includes blocks made out of different materials such as iron and gold

Sprint backlog:

- Enemy projectiles
- Arrow collision with player when attacked by enemies
- Inventory screen
- Add buttons on game screen (inventory & menu)
- Add blocks with properties to the map
- Character can destroy and place blocks right and down in addition to left
- Pebble speech
- There is an antigravity block allowing the player to rise until he is out of the vertical axis or collides with block
- Video for the end of the game (credits)
- Documentation

Completed:

- Find cross platform library
- Research on how GitHub version control works
- Java research
- Game development research
- Create a simple character (square)
- give this character up, down, left, and right movement
- Find a common naming system
- Design classes and create a UML diagram
- General game structure (features & menu files)
- Pebble backstory
- Map generation

- Main menu screen
- Gravity
- Camera view
- User input
- A player is spawned on the map
- Collision
- Find character and enemy sprites
- Game name
- A more detailed storyline
- Implement sounds
- Animations added to main character, Pebble and enemies
- The player can interact with the blocks of the map (destroy and create)
- Example map that the game will load (horizontal view)
- Documentation formatting and sprint documentation
- Code refactoring which restructured codes into classes
- There will be a timer shown to the player on the screen which runs down as they go through the game

Exception Handling

There were no exceptions to be handled for this week.

Product content

Customer meeting

The customer indicated a liking for the sanity implementation which zooms in the screen as the countdown timer decreases. According to the customer, this feature creates a claustrophobic feeling which has a stronger psychological effect. However, the customer mentioned that the relationship between where the player is, what the player can do and the storyline seems disconnected. An additional feature that the customer proposed was having a grappling iron that can be thrown to the next platform to aid in movement.

User stories

ID	Version	Priority	Story
US_01	2	1	AS A player I WANT to be able to play a dungeon game on different platforms web platforms SO THAT I have choices for playing the game
US_02	2	5	AS A player I WANT to have a bot that will play for me SO THAT It can help me with playing through the game by offering advice
US_04	2	6	AS A player I WANT to be able to see a timer go down SO THAT I know if I am doing well at the game as I play
US_05	2	13	AS A player I WANT to have different choices between games modes and different paths to the end in each game mode SO THAT I have new scenarios to play
US_06	1	2	AS A player I WANT to control the character's movement and interact with the blocks on the map SO THAT I can gather resources, place resources, and make new paths
US_07	2	3	AS A player I WANT to use resources to upgrade armour and weapons SO THAT I can use them throughout the game to make fighting enemies easier
US_08	1	10	AS A player I WANT to be able to regenerate health SO THAT I can survive longer
US_09	1	8	AS A player I WANT to see a high-resolution game style SO THAT the background, items and characters seem clearer
US_10	1	12	AS A player I WANT to be able to explore a range of different backgrounds SO THAT the game is more playable
US_11	1	11	AS A player I WANT to see a more detailed storyline with episodes and gain skills along the way SO THAT I can be more motivated to play and see how an episode will end and how the character develops

ID	Version	Priority	Story
US_121	8		AS A player I WANT to be able to gain height in the map quickly SO THAT I can save time in reaching parts of the map
US_131	7		AS A player I WANT to battle enemies SO THAT the game has a challenge to it
US_141	4		AS A player I WANT to see how many resources have been collected SO THAT I know the current number of each materials already collected and how many more I need in order to upgrade armour or craft items

User story testing

User story ID	Test	Comments	completed
US_01	Run code on different systems and see if code compiles	Web based systems on Windows, Mac and Linux	YES
US_02	A bot character is generated on the game screen. The bot aids the user with text aids	The bot will have to be able to follow the player around the map, meaning it will need similar movement animations to the main character	NO
US_04	There is a visual queue on the screen that will track specific progression.	There are multiple types of progress – elevation, time or collection	NO
US_05	Within the menu screen the player sets the game mode and the characteristics of the game change	These characteristics could be unlimited health, unlimited resources and no enemies	NO
US_06	The player can destroy a block and it disappears from the game screen. This block will then show up within the inventory	From the inventory screen there are multiple options the player can take which require use cases and user stories	NO
US_07	There is an option to craft items from the inventory screen. If the player has the required resources, they can craft the item and the resources are taken from their inventory	You can implement many ways to show what can be crafted with the resources provided.	NO
US_08	Have the player take damage, from then the health will decrease and then it regenerates once they stop taking damage		NO
US_09	Add sprites into the game	The sprites need to be suitable in terms of the theme of the game and overall art style	NO
US_10	Add sprites, generate different types of ground and environment texture	The player should be able to go to different places with various environment such as river, woods, stone road	NO

User story	ID	Test	Comments	completed
	US_11	Have a storyline and give character various abilities as the plots continue	The player should be able to find torches, mine, craft weapons as the story progresses	NO
	US_12	The player starts at a certain y position. Then in a time shorter than climbing blocks, the player is elevated to a higher y coordinate	There are different ways to achieve this, such as placing a block down which will change the gravity in the vertical plane	YES
	US_13	The player will come into contact with the enemy, where the enemy will begin to attack the player	The enemy could use projectiles or melee to attack the player. They could also be stationary or mobile.	NO
	US_14	The player will see an inventory screen with different materials shown and craft items	Each type of block will be shown using both a small picture and name such as iron, with a number behind.	NO

Use cases

Existing Use cases:**

Use case ID	Use case description	Tested
UC_01	Player starts game on different systems	YES
UC_02	Player destroys a block on the map	YES
UC_03	Player places a block on the map	YES
UC_04	Player crafts an item from the inventory	NO
UC_05	Player controls the movement of the character	YES
UC_06	As the players sanity goes up his sidekick will offer advice	NO
UC_07	have high-resolution items, characters and environment	YES
UC_08	Play in different environments	NO
UC_09	The characters' down movements	YES
UC_10	Collision system	YES
UC_11	Main character gain skills as the story progresses	NO
UC_12	The enemy will fire a projectile at the main character	YES
UC_13	The boss attacks the player	NO
UC_14	Player walks on block and reverses gravity	YES

New Use cases:

UC_15 Use case: Check inventory

Author: TJ

Date: 20/11/2020

Modification date: 20/11/2020

Purpose: show the number of each material collected

Overview: The player will gather resources and then check how many they have by accessing the inventory screen

Cross reference: US_06, US_14

Actors: player

Precondition:

- Have different types of blocks that are placed on the map which can be collected

Normal flow of event:

Actor actions	System actions
1. Collect different materials from blocks	2. Update the number of each material in inventory
3. Open inventory menu	4. Shows the updated inventory

Alternative flow of events: - There are none

Exceptional flow of events: - The number does not update according to the users' collection

UC_16 (related to UC_12)

Use case: Player's health decreases when taking damage

Author: TJ

Date: 20/11/2020

Modification date: 20/11/2020

Purpose: Give player a sense of the character's health status during a fight

Overview: The character will take damage from an enemy and the health bar reduces in length

Cross reference: US_08

Actors: player

Precondition: - A health bar is shown to the player

Normal flow of event:

Actor actions	System actions
1. Character shot by the enemies' arrows	2. Decrease the character's health 3. Show the changes through health bar

Alternative flow of events: - There are none

Exceptional flow of events: - The attack fails to reduce the character's health
- The changes in health bar react slowly so the character has taken more damage than shown and will be dead which appears sudden to the player

UC_17

Use case: Difficulty selection

Author: TJ

Date: 20/11/2020

Modification date: 20/11/2020

Purpose: Let players choose what level of gaming difficulty they want

Overview: The player will choose difficulty level and the countdown timer will change accordingly

Cross reference: US_05

Actors: player

Precondition: - Countdown timer is implemented - Menu screen has options for different levels of difficulties

Normal flow of event:

Actor actions	System actions
1. Select a desired difficulty level	2. Interprets the option and picks a corresponding total time in the countdown timer 3. Show the countdown timer

Alternative flow of events: - The player chooses “hard” and finds it too challenging. He/She can click on the menu button in the game, go back to the main menu and change the difficulty level

Exceptional flow of events: - The timer does not correspond with the difficulty level; for example, more time for a harder level

Use cases testing

- UC_01**
- UC_02**
- UC_03**
- UC_04**
- UC_05**
- UC_06**
- UC_07**
- UC_08**
- UC_09**
- UC_10**
- UC_11**
- UC_12**
- UC_13**
- UC_14**
- UC_15**
 - The player collects materials for crafting
 - The system updates the number of materials in the inventory screen
 - The player checks how many resources have been gathered and make decisions during game play such as how many more materials to collect
- UC_16**
 - The enemy attacks the player
 - The health bar reduces in length
- UC_17**
 - The player chooses a difficulty level
 - Time limitation indicated by a countdown timer is shown to the player differently in all three levels.

Software design Documentation

Class: Inventory (see UC_04, UC_15) - Responsibilities - shows the number of each type of materials collected - crafts items
- Collaborators - Environment - Element - Character

User interface design

Character:

- Once in the game there will be character that can move and interact with the whole map [1]
- The character needs to be able to drop to lower heights [1]

Menu:

- There will be a starting menu screen with the game settings, load game, continue and new game [2]
- In the settings tab there will be choices for audio, visuals, and controls [1]
- The player can go to an in game menu with save game and achievements [1]
- **When player is in the game, two buttons are shown on the top of the screen (inventory and menu) [1]**

Sounds:

- There should be background music played at the menu screen and during the game. Background music can change along with the character's state such as sanity levels and become more intense during events like a boss fight [1]
- Sounds should be triggered when certain actions are performed (e.g. menu click, fight, mining) [1]

Inventory:

- The player can access an inventory screen which has options for equipment [1]
- On this screen the player can see amount of each block, craft option for armour and which block is being placed[2]
- **Different types of materials gathered from a variety of blocks are shown, each with a number that indicates how many have been collected [1]**

Game flow:

- When the player had successfully completed the game they will be sent to the credits and the starting screen [2]
- **A video will be shown that includes credits at the end of the game [1]**

Map:

- **The map should have edges and certain sections that the user cannot pass through such as a wall [1]**
- **The map should have different types of blocks [1]**

Pebble:

- A speech bubble will appear above Pebble that gives players instructions [1]
- Dialog box at the bottom of the screen at the beginning of each episode (instructions and story-telling from Pebble), which can be skipped [1]

Week 7 documentation: 27th November-3rd December

Process

Weekly Overview

During last week, the team decided to stick to minimum requirements that could be achieved in the last few weeks of this project considering the need to work on product documentation during the final week. Basic functions that are core to the game were further developed such as enemy attack, block interaction, map and anti-gravity movement. As a result, the game appeared as a whole rather than components that can be shown to the customer.

This week the team had a clearer goal on what needs to be achieved by the end of the term and continued working on essential features such as creating a final map with different types of blocks and enemies in addition to fixing bugs. Enemy health and death animation for all characters were added. The inventory was made to count number of blocks collected by the player. More sounds were implemented for different scenarios. Documentation was generated up to last week for the preparation of final organization into pdf file which was tested last week. From these aspects we can see that the sprint went successfully and many tasks were finished on time. The game starts to grow into a more finalized product.

The presentation created for the customer meeting on the 4th December can be found here, which will have GIFs for some of the artefacts shown here: [Presentation](#)

Meeting Minutes

Customer meeting debrief and sprint planning

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

After the customer meeting, the team held a meeting to discuss game features that need to be worked on:

- A general map larger than the previously developed ones, which will be the final map for this game with blocks that can be mined as well as anti-gravity blocks
- Enemies need to follow the player around within a certain distance
- Enemies need to have health bar as well
- Death animation will need to be implemented for the characters
- Armor can be crafted within the inventory menu after the player collects a certain number of blocks and animation will need to be added with the character wearing the armour. After the health drops below a certain level, the armour will break
- Need to have more Pebble speech added

Sprint planning

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

- Change menu screen options to make them look better
- Non-linear zoom for sanity feature
- Different types of armour can be made that add to the character's health
- Now has a general map
- add blocks that can't be destroyed for the edge of the map
- add some more blocks for mining
- add more enemies
- can have secret passages to be discovered
- Have death animation
- Add gold to inventory
- Pebble will appear beside the main character again if the distance between them reaches a certain amount
- Decide style for the final pdf document
- page break for each title
- have hyperlinks
- format for user stories
- Keep working on sprint documents
- Start to work on a template for user manual and installation guide

Sprint review and presentation creation

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

- Final map has been created
- Mining blocks has been improved
- Added enemy health bar
- Inventory can count number of blocks collected for each type
- Sound has been implemented
- Pebble's speech bubble has improved to be larger and more clear
- Menu screen was improved
- Player can attack
- Death animation added to every character

Backlog

Product backlog:

- There will be tasks to complete in order for the player to escape
- Random generation of block map (side on view)
- Interaction between characters (Pebble, enemies)

Sprint backlog:

- Non-linear zoom
- Armor crafting in inventory, with it breaking over time(health dependent)
- Final map (add blocks (gold, iron), enemies, possibly secret passages)
- Death animation
- Inventory (add gold)
- Pebble will appear beside the main character again if the distance between them reaches a certain amount
- Refine final pdf document style
- Sprint documents
- Template for user manual and installation guide
- The player will fight a boss in the final game scene to escape the dungeon

Completed:

- Find cross platform library
- Research on how GitHub version control works
- Java research
- Game development research
- Create a simple character (square)
- give this character up, down, left, and right movement
- Find a common naming system
- Design classes and create a UML diagram
- General game structure (features & menu files)
- Pebble backstory
- Map generation (early implementation)
- Main menu screen
- Gravity
- Camera view

- User input
- A player is spawned on the map
- Collision
- Find character and enemy sprites
- Game name
- A more detailed storyline
- Implement sounds
- Animations added to main character, Pebble and enemies
- The player can interact with the blocks of the map (destroy and create)
- Example map that the game will load (horizontal view)
- Documentation formatting and sprint documentation
- Code refactoring which restructured codes into classes
- There will be a timer shown to the player on the screen which runs down as they go through the game
- Enemy projectiles
- Arrow collision with player when attacked by enemies
- Inventory screen
- Add buttons on game screen (inventory & menu)
- Add blocks with properties to the map
- Character can destroy and place blocks right and down in addition to left
- Pebble speech
- Anti-gravity
- Video for credits

Exception Handling

There were no exceptions to be handled for this week.

Product content

Customer meeting

The customer seemed to be satisfied with the progress shown. No requirements were mentioned during the meeting. The main point raised was having all the features we showed joined onto one to form the game. We agreed with this, which is why the major end result of the sprint week was to have the features running in one version of the game. This would then be shown to the customer in the next meeting with a little playthrough

User stories

ID	Version	Priority	Story
US_01	2	1	AS A player I WANT to be able to play a dungeon game on different platforms web platforms SO THAT I have choices for playing the game
US_02	2	5	AS A player I WANT to have a bot that will play for me SO THAT It can help me with playing through the game by offering advice
US_04	2	6	AS A player I WANT to be able to see a timer go down SO THAT I know if I am doing well at the game as I play
US_05	2	14	AS A player I WANT to have different choices between games modes and different paths to the end in each game mode SO THAT I have new scenarios to play
US_06	1	2	AS A player I WANT to control the character's movement and interact with the blocks on the map SO THAT I can gather resources, place resources, and make new paths
US_07	2	3	AS A player I WANT to use resources to upgrade armour and weapons SO THAT I can use them throughout the game to make fighting enemies easier
US_08	1	11	AS A player I WANT to be able to regenerate health SO THAT I can survive longer
US_09	1	9	AS A player I WANT to see a high-resolution game style SO THAT the background, items and characters seem clearer
US_10	1	13	AS A player I WANT to be able to explore a range of different backgrounds SO THAT the game is more playable
US_11	1	12	AS A player I WANT to see a more detailed storyline with episodes and gain skills along the way SO THAT I can be more motivated to play and see how an episode will end and how the character develops
US_12	1	10	AS A player I WANT to be able to gain height in the map quickly SO THAT I can save time in reaching parts of the map
US_13	1	7	AS A player I WANT to battle enemies SO THAT the game has a challenge to it
US_14	1	4	AS A player I WANT to see how many resources have been collected SO THAT I know the current number of each materials already collected and how many more I need in order to upgrade armour or craft items

ID	Version	Priority	Story
US_15	8		AS A player I WANT to see how much health the enemies I'm fighting against have left SO THAT I have an idea of how much time it takes to defeat them

User story testing

User story ID	Test	Comments	completed
US_01	Run code on different systems and see if code compiles	Web based systems on Windows, Mac and Linux	YES
US_02	A bot character is generated on the game screen. The bot aids the user with text aids	The bot will have to be able to follow the player around the map, meaning it will need similar movement animations to the main character	NO
US_04	There is a visual queue on the screen that will track specific progression.	There are multiple types of progress – elevation, time or collection	NO
US_05	Within the menu screen the player sets the game mode and the characteristics of the game change	These characteristics could be unlimited health, unlimited resources and no enemies	NO
US_06	The player can destroy a block and it disappears from the game screen. This block will then show up within the inventory	From the inventory screen there are multiple options the player can take which require use cases and user stories	NO
US_07	There is an option to craft items from the inventory screen. If the player has the required resources, they can craft the item and the resources are taken from their inventory	You can implement many ways to show what can be crafted with the resources provided.	NO
US_08	Have the player take damage, from then the player can craft armour which will reset the health		NO
US_09	Add sprites into the game	The sprites need to be suitable in terms of the theme of the game and overall art style	YES
US_10	Add sprites, generate different types of ground and environment texture	The player should be able to go to different places with various environment such as river, woods, stone road	NO

User story	ID	Test	Comments	completed
	US_11	Have a storyline and give character various abilities as the plots continue	The player should be able to find torches, mine, craft weapons as the story progresses	NO
	US_12	The player starts at a certain y position. Then in a time shorter than climbing blocks, the player is elevated to a higher y coordinate	There are different ways to achieve this, such as placing a block down which will change the gravity in the vertical plane	YES
	US_13	The player will come into contact with the enemy, where the enemy will begin to attack the player	The enemy could use projectiles or melee to attack the player. They could also be stationary or mobile.	YES
	US_14	The player will see an inventory screen with different materials shown and craft items	Each type of block will be shown using both a small picture and name such as iron, with a number behind.	YES
	US_15	The player will see a health bar above each enemy's head	The health bar reduces when the player attacks the enemy.	YES

Use cases

Existing Use cases:**

Use case ID	Use case description	Tested
UC_01	Player starts game on different systems	YES
UC_02	Player destroys a block on the map	YES
UC_03	Player places a block on the map	YES
UC_04	Player crafts an item from the inventory	NO
UC_05	Player controls the movement of the character	YES
UC_06	As the players sanity goes up his sidekick will offer advice	NO
UC_07	have high-resolution items, characters and environment	YES
UC_08	Play in different environments	NO
UC_09	The characters' down movements	YES
UC_10	Collision system	YES
UC_11	Main character gain skills as the story progresses	NO
UC_12	The enemy will fire a projectile at the main character	YES
UC_13	The boss attacks the player	NO
UC_14	Player walks on block and reverses gravity	YES
UC_15	Check inventory	YES
UC_16	Player's health decreases when taking damage	YES
UC_17	Player walks on block and reverses gravity	YES

New and updated use cases: UC_04

Use case: Player crafts armour from the inventory

Author: TJ

Date: 27/11/2020

Modification date: 27/11/2020

Purpose: Allow player to regenerate health and make fighting enemies easier

Overview: The player will collect blocks until a certain number and open the inventory menu to craft armour

Cross reference: US_07

Actors: player

Precondition: - There are enough blocks to be crafted

Normal flow of event:

Actor actions	System actions
1. Gather different types of blocks	2. Keep a record of number of blocks collected
4. Open the inventory menu and see how many blocks have been collected for each type	3. Update the inventory screen
6. Craft armour	5. Show the inventory screen
7. Close inventory screen	8. Armor animation is shown on the character

Alternative flow of events: - Player wants to craft armour without enough resources. Error message shows: “Insufficient resources” - The health added by the armour is used up. The armour animation disappears, which indicates that it has broke.

Exceptional flow of events: - Player has gathered many resources but the system fails to update the number on inventory screen

UC_18

Use case: Enemy has a health bar that will go down when attacked

Author: TJ

Date: 27/11/2020

Modification date: 27/11/2020

Purpose: Allow player to see how much health has left for enemies

Overview: The player will attack an enemy and the health bar reduces

Cross reference: US_15

Actors: player

Precondition: - The enemy is being attacked

Normal flow of event:

Actor actions	System actions
1. Attack an enemy	2. Show health bar for the enemy
3. Keep attacking	4. Health bar reduces

Alternative flow of events: - Player attacks the enemy and runs away. Health bar disappears after a certain range of distance

Exceptional flow of events: - Health bar is not shown and the player does not know whether the enemy can be defeated

Use cases testing

- UC_01**
- UC_02**
- UC_03**
- UC_04**
- UC_05**
- UC_06**
- UC_07**
- UC_08**
- UC_09**
- UC_10**
- UC_11**
- UC_12**
- UC_13**
- UC_14**
- UC_15**
- UC_16**
- UC_17**
- UC_18**
 - Player gathers blocks
 - Player opens inventory
 - Inventory shows how many blocks collected for each type
 - Player crafts armour
- UC_19**
 - Player attacks enemy
 - Enemy's health bar reduces in length

Software design Documentation

Class: Armor (see UC_18) - Responsibilities - shows the number of each type of materials collected - crafts items
- Collaborators - Character - Inventory

User interface design

Character:

- Once in the game there will be character that can move and interact with the whole map [1]
- The character needs to be able to drop to lower heights [1]

Menu:

- There will be a starting menu screen with the game settings, load game, continue and new game [2]
- In the settings tab there will be choices for audio, visuals, and controls [1]
- The player can go to an in game menu with save game and achievements [1]
- When player is in the game, two buttons are shown on the top of the screen (inventory and menu) [1]

Sounds:

- There should be background music played at the menu screen and during the game. Background music can change along with the character's state such as sanity levels and become more intense during events like a boss fight [1]
- Sounds should be triggered when certain actions are performed (e.g. menu click, fight, mining) [1]

Inventory:

- The player can access an inventory screen which has options for equipment [1]
- In the equipment tab there will be options for view, craft, and delete [1]
- Different types of materials gathered from a variety of blocks are shown, each with a number that indicates how many have been collected [1]
- **Player can craft armour from gathered blocks [1]**

Game flow:

- When the player had successfully completed the game they will be sent to the credits and the starting screen [2]
- A video will be shown that includes credits at the end of the game [1]

Map:

- **The map should have blocks that can't be destroyed along the edge [2]**
- The map should have different types of blocks [1]

Pebble:

- A speech bubble will appear above Pebble that gives players instructions [1]

- Pebble will appear near the main character again if the distance between them gets to a certain amount [1]

Week 8 Documentation: 4th- 8th December

Process

Weekly overview

This was the final week of the Software Engineering unit, meaning we decided to focus most the week on documentation. This is why the dates are from 4th-7th December, as we wanted to have our final version of the game finished on Monday. The game was able to fulfil most of our user stories and use cases to its basic level, which we adjusted back in week 5.

We had one meeting on the Friday after the customer meeting to discuss what needed implementing and what bugs needed to be looked at in the last few days. We only had one think to implement and the rest was just fixing bugs and tweaking already implemented features. Then on Monday we had our final meeting to check the game and agree that was our final version. While this was going on we did break up the team to start tackling the different parts of the documentation.

We carried on the same style of sprint as last week and although we were not implanting a lot this week, we still looked over the documentation to assess if we were able to fulfil user stories and use cases. This helped decide what bugs to fix in this sprint. This was a small sprint so not too much was learnt from the process. Instead we appreciated the time to develop features within a game, and the importance of having up to date use cases and user stories to optimize your time.

pagebreak

Meeting minutes

Customer meeting debrief and sprint planning

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

After the customer meeting, the team held a meeting to discuss game features bugs that need to be worked on:

- Have the end credits pop up onto the screen when the boss is killed
- After the end credits you are taken to the starting screen
- Fix some bugs within the game:
 - The archers will face the way they are shooting
 - The enemies will only follow the player when the player is in a certain radius of the player
 -

Sprint review and final game development meeting

Attendance: He Jiang, Julius Martinez, Lewis Williams, Samuel Love, Tianyu Ji, Tom Wells, Vlad Postmangiu Luchian, Yandong Guo

have not had this yet

Backlog

The items left in the product backlog were ideas that if we have time could be implemented. Unfortunately we were not able to implement them into the game, but we were able to get all core mechanics in.

Product backlog:

- There will be tasks to complete in order for the player to escape
- Random generation of block map (side on view)
- Interaction between characters (Pebble, enemies)

Sprint backlog:

- Add credit scenes which are triggered when the boss dies or the timer runs out
- From the credit scene the user is taken back to the starting game screen
- fix bugs
- Documentation part 1

completed:

- Find cross platform library
- Research on how GitHub version control works
- Java research
- Game development research
- Create a simple character (square)
- give this character up, down, left, and right movement
- Find a common naming system
- Design classes and create a UML diagram
- General game structure (features & menu files)
- Pebble backstory
- Map generation (early implementation)
- Main menu screen
- Gravity
- Camera view
- User input
- A player is spawned on the map
- Collision
- Find character and enemy sprites
- Game name
- A more detailed storyline
- Implement sounds
- Animations added to main character, Pebble and enemies
- The player can interact with the blocks of the map (destroy and create)
- Example map that the game will load (horizontal view)

- Documentation formatting and sprint documentation
- Code refactoring which restructured codes into classes
- There will be a timer shown to the player on the screen which runs down as they go through the game
- Enemy projectiles
- Arrow collision with player when attacked by enemies
- Inventory screen
- Add buttons on game screen (inventory & menu)
- Add blocks with properties to the map
- Character can destroy and place blocks right and down in addition to left
- Pebble speech
- Anti-gravity
- Video for credits
- Non-linear zoom
- Armor crafting in inventory, with it breaking over time(health dependent)
- Final map (add blocks (gold, iron), enemies, possibly secret passages)
- Death animation
- Inventory (add gold)
- Pebble will appear beside the main character again if the distance between them reaches a certain amount
- Refine final pdf document style
- Template for user manual and installation guide
- The player will fight a boss in the final game scene to escape the dungeon

Exception Handling:

Due to a major part of the unit being the documentation, we had to accept what our final game is capable of being.

Product content

customer meeting

The customer seemed to be satisfied with the progress shown. No requirements were mentioned during the meeting. The game clearly showed bugs with certain features so the customer was curious to know what we were going to do about them in the time remaining. We would assess the bugs, ranking them in importance, which would help decide what was going to be worked on.

User stories

ID	Version	Priority	Story
US_01	2	1	AS A player I WANT to be able to play a dungeon game on different platforms web platforms SO THAT I have choices for playing the game
US_02	2	5	AS A player I WANT to have a bot that will play for me SO THAT It can help me with playing through the game by offering advice
US_04	2	6	AS A player I WANT to be able to see a timer go down SO THAT I know if I am doing well at the game as I play
US_05	2	14	AS A player I WANT to have different choices between games modes and different paths to the end in each game mode SO THAT I have new scenarios to play
US_06	1	2	AS A player I WANT to control the character's movement and interact with the blocks on the map SO THAT I can gather resources, place resources, and make new paths
US_07	2	3	AS A player I WANT to use resources to upgrade armour and weapons SO THAT I can use them throughout the game to make fighting enemies easier
US_08	1	11	AS A player I WANT to be able to regenerate health SO THAT I can survive longer
US_09	1	9	AS A player I WANT to see a high-resolution game style SO THAT the background, items and characters seem clearer
US_10	1	13	AS A player I WANT to be able to explore a range of different backgrounds SO THAT the game is more playable
US_11	1	12	AS A player I WANT to see a more detailed storyline with episodes and gain skills along the way SO THAT I can be more motivated to play and see how an episode will end and how the character develops
US_12	1	10	AS A player I WANT to be able to gain height in the map quickly SO THAT I can save time in reaching parts of the map
US_13	1	7	AS A player I WANT to battle enemies SO THAT the game has a challenge to it

ID	Version	Priority	Story
US_14	1	4	AS A player I WANT to see how many resources have been collected SO THAT I know the current number of each materials already collected and how many more I need in order to upgrade armour or craft items
US_15	1	8	AS A player I WANT to see how much health the enemies I'm fighting against have left SO THAT I have an idea of how much time it takes to defeat them
US_16	1	15	AS A player I WANT to see the end credits when I kill the boss at the end of the game, run out of time or die SO THAT I know I have completed the game

User story testing

User story ID	Test	Comments	completed
US_01	Run code on different systems and see if code compiles	Web based systems on Windows, Mac and Linux	YES
US_02	A bot character is generated on the game screen. The bot aids the user with text aids	The bot will have to be able to follow the player around the map, meaning it will need similar movement animations to the main character	YES
US_04	There is a visual queue on the screen that will track specific progression.	There are multiple types of progress – elevation, time or collection	YES
US_05	Within the menu screen the player sets the game mode and the characteristics of the game change	These characteristics could be unlimited health, unlimited resources and no enemies	YES
US_06	The player can destroy a block and it disappears from the game screen. This block will then show up within the inventory	From the inventory screen there are multiple options the player can take which require use cases and user stories	YES
US_07	There is an option to craft items from the inventory screen. If the player has the required resources, they can craft the item and the resources are taken from their inventory	You can implement many ways to show what can be crafted with the resources provided.	YES
US_08	Have the player take damage, from then the player can craft armour which will reset the health		YES
US_09	Add sprites into the game	The sprites need to be suitable in terms of the theme of the game and overall art style	YES
US_10	Add sprites, generate different types of ground and environment texture	The player should be able to go to different places with various environment such as river, woods, stone road	YES

User story	ID	Test	Comments	completed
	US_11	Have a storyline and give character various abilities as the plots continue	The player should be able to find torches, mine, craft weapons as the story progresses	NO
	US_12	The player starts at a certain y position. Then in a time shorter than climbing blocks, the player is elevated to a higher y coordinate	There are different ways to achieve this, such as placing a block down which will change the gravity in the vertical plane	YES
	US_13	The player will come into contact with the enemy, where the enemy will begin to attack the player	The enemy could use projectiles or melee to attack the player. They could also be stationary or mobile.	YES
	US_14	The player will see an inventory screen with different materials shown and craft items	Each type of block will be shown using both a small picture and name such as iron, with a number behind.	YES
	US_15	The player will see a health bar above each enemy's head	The health bar reduces when the player attacks the enemy.	YES
	US_16	The player will either kill the boss, run out of time or die. As soon as this happens the credits will take up the screen.	Following the credits the user should be taken to the starting screen	YES

Use cases

Use case ID	Use case description	Tested
UC_01	Player starts game on different systems	YES
UC_02	Player destroys a block on the map	YES
UC_03	Player places a block on the map	YES
UC_04	Player crafts armour from the inventory	YES
UC_05	Player controls the movement of the character	YES
UC_06	As the players sanity goes up his sidekick will offer advice	NO
UC_07	have high-resolution items, characters and environment	YES
UC_08	Play in different environments	YES
UC_09	The characters' down movements	YES
UC_10	Collision system	YES
UC_11	Main character gain skills as the story progresses	NO
UC_12	The enemy will fire a projectile at the main character	YES
UC_13	The boss attacks the player	YES
UC_14	Player walks on block and reverses gravity	YES
UC_15	Check inventory	YES
UC_16	Player's health decreases when taking damage	YES
UC_17	Player walks on block and reverses gravity	YES
UC_18	Enemy has a health bar that will go down when attacked	YES

Use case testing

- UC_01
- UC_02
- UC_03
- UC_04
- UC_05
- UC_06
- UC_07
- UC_08
- UC_09
- UC_10
- UC_11
- UC_12
- UC_13
- UC_14
- UC_15
- UC_16
- UC_17
- UC_18

Software design documentation

Class: credits (see US_16)

Responsibilities:

- Show the user the authors of the game
- be able to close main game (return to starting screen)

Collaborators:

- Boss enemy
- Character
- Timer

User interface design

Character:

- Once in the game there will be character that can move and interact with the whole map [1]
- The character needs to be able to drop to lower heights [1]

Menu:

- There will be a starting menu screen with the game settings, load game, continue and new game [2]
- In the settings tab there will be choices for audio, visuals, and controls [1]
- The player can go to an in game menu with save game and achievements [1]
- When player is in the game, two buttons are shown on the top of the screen (inventory and menu) [1]

Sounds:

- There should be background music played at the menu screen and during the game. Background music can change along with the character's state such as sanity levels and become more intense during events like a boss fight [1]
- Sounds should be triggered when certain actions are performed (e.g. menu click, fight, mining) [1]

Inventory:

- The player can access an inventory screen which has options for equipment [1]
- In the equipment tab there will be options for view, craft, and delete [1]
- Different types of materials gathered from a variety of blocks are shown, each with a number that indicates how many have been collected [1]
- Player can craft Armor from gathered blocks [1]

Game flow:

- When the player had successfully completed the game (kill boss) they will be sent to the credits and the starting screen [3]
- When the player has run out of time or has been killed they will be sent to the credits and the starting screen [1]
- A video will be shown that includes credits at the end of the game [1]
- Following the end credits the plater will be taken to the starting screen [1]

Map:

- The map should have blocks that can't be destroyed along the edge [2]
- The map should have different types of blocks [1]

Pebble:

- A speech bubble will appear above Pebble that gives players instructions [1]
- Pebble will appear near the main character again if the distance between them gets to a certain amount [1]