

# Tuesday 11 am Group 7 Design Report

## Contributors:

Adam Burgess (37653016)  
Archie Aiken-Wood (37666312)  
Indre Aruodziute (37695835)

## Table of Contents

1. Introduction.....	2
2. Project Ideas .....	2
2.1 Chosen idea .....	2
2.2 Other Initial Ideas .....	2
2.2.1 Physics Puzzled styled game .....	2
2.2.2 Birds Eye view 2D Driving game .....	3
2.2.3 Gravity Altering Game .....	3
3. Game Design .....	4
3.1 Game Features and Gameplay .....	4
3.1.1 Story .....	4
3.1.2 Dungeon Tiers .....	4
3.1.3 Dungeon doors and trapdoors .....	4
3.1.4 descending and ascending .....	4
3.1.5 Monsters .....	4
3.1.6 Items/Inventory .....	5
3.1.7 NPCs .....	5
3.1.8 Trading .....	5
3.1.9 Levels/Player progression .....	5
3.1.10 Challenges .....	5
3.2 Game Design Principles .....	5
3.2.1 Pursuing and Achieving Goals .....	5
3.2.2 Game Interactivity .....	6
3.2.3 Feedback .....	6
3.2.4 Gameplay Variety .....	6
3.2.5 Gameplay Consistency and Fairness .....	6
3.2.6 Avoiding Repetition .....	7
3.3 Game Rules .....	7
3.3.1 Constitutive .....	7
3.3.2 Operational .....	7
3.3.3 Implicit .....	8
4 Software Engineering .....	8
4.1 Software requirements .....	8
4.1.1 Functional requirements .....	8
4.1.2 Non-Functional requirements .....	9
4.2 Testing .....	10
5. Implementation Plans .....	15
5.1 Task List for Term 1 .....	15
5.2 Project Plan for Term 2 .....	16
5.3 Activity Network with Critical Path .....	19
5.4 Gantt Chart .....	20

# 1. Introduction

Design brief: **create a 2D arcade-style game, written in Java, which is suitable for an audience of 12 and up.**

We decided upon creating an RPG dungeon crawler hybrid with a thrilling storyline behind it. The game is set in a medieval fantasy world dominated by the elemental Gods; Earth, Water and Fire. The mix of RPG and exploration is a perfect amalgamation for our target market, it is simple yet engaging, with the imagination of the player playing a big part. The battling of monsters of various difficulty is perfect too, it allows for more challenging monsters to be beaten if a player has higher tactical intelligence, keeping all age ranges entertained.

## 2. Project Ideas

Upon finding our group, those present in the meeting started brainstorming ideas for our game. As a team we decided to develop everyone's individual suggestions first to flesh out experiences and interests. From here, we took a deeper look into the advantages and disadvantages of each game which allowed us to come up with an RPG dungeon crawler hybrid with an interesting story in which a player has to battle the elemental Gods of Askana to take back your kingdom.

### 2.1 Chosen idea

A RPG meets dungeon explorer video game is intertwined with the story of overcoming the elemental Gods where the player must play the game to improve in a range of skills by gaining EXP from completing tasks and slaying monsters. Exploration will include hidden rooms, which will allow monster fights as well which will allow for the game to be continued after the main story is added.

#### Key features:

- Medieval Fantasy themed dungeon crawler
- Tiers of the dungeon progress with difficulty
- Each tier contains unique theme of the game
- Birds-eye view dungeons
- Continuous game-play
- Challenges
- Story based
- Variety of fun themed enemies
- A range of mechanics

In the end, we decided to go forward with this idea due to the amount of choice the player will have within a game. Because of the wide range of mechanics, the player will always have something to do, followed by decisions and critical thinking in how and when to do it. The fantasy theme and elemental God story further adds to this, as the game will have a wide range of different themed dungeons. The scope of it will allow all team members to contribute to it significantly.

### 2.2 Other Initial Ideas

#### 2.2.1 Physics Puzzled styled game

Inspired by the very popular game during the 2010s on smartphone devices: "Cut the Rope", we thought of an idea which was a heavily physics-based puzzle game. The player continually learns new mechanics which are based on physics equations (such as a rope connected to a ball or running water affecting the speed of the ball). After learning each mechanic, the player is given a series of levels which test their cognitive ability and to make full use of the mechanics they have just learned as well as general problem-solving puzzles.

### **Key Features:**

- Physics Based
- Puzzle on Brain
- Colourful style
- Themes for each mechanic
- Ease of Use
- Progressively challenging levels
- Mechanics focused

The people present in our group decided to not go for this idea. This was due to the fact that despite the mechanics theoretically providing the player a challenge, a large amount of development time would be testing if the levels are difficult enough and making sure there is not an easier way to solve a level. This would take away from us developing further mechanics and therefore could hinder the user's overall experience. Also, we felt it would be difficult to implement any kind of story which could add replay value.

### **2.2.2 Birds Eye view 2D Driving game**

Once again taking an approach based on many popular smartphone games, a potential idea was to have a 2D driving game, in which the player would dodge other cars driving towards them with simple swiping mechanics. The game would feature a single level, in which the cars they collide with would be procedural generated. As they continued to survive and improved their score, the speed of their car would increase at a constant rate.

### **Key Features:**

- Customizable Car
- Test the players reactions
- Endless
- Single continually level
- 16-bit classic arcade game style
- Compete against friends

Although this idea seems fun and could be something the player could play during a lunch break, we decided to not go with this game. This was because, despite there being a customizable car, we felt like even for the player putting in hours in the game, little progress is being achieved as well as because of the difficulty of making the game engaging for longer periods of time.

### **2.2.3 Gravity Altering Game**

It is a rhythm-gravity changing based game where the player takes over the geometric figure as a square or triangle and tries to jump over the different occurring obstacles and collect coins for purchasing things from the store (new skins, additional levels, upgrades). The game would include the ability for the player to rotate the walls to make their way up the puzzle.

### **Key features:**

- Different themed levels with variety of backgrounds
- Quick and exciting
- Mechanics based
- Fun, colorful style
- Customizable figures
- Challenge mode

Despite the idea of an interesting game we did not choose to implement it due to the difficulty of making it really exciting for the player to come back and play again. We felt like creating interesting levels would be quite challenging as well as making the game run smoothly on every device. Timing issues with jumping over obstacles might occur and since it is a speed driven game, it would not bring as much of an enjoyment to the players as they originally anticipated.

## 3. Game Design

Following the chosen game idea: RPG dungeon crawler, sections below describe the detailed overview of the game.

### 3.1 Game Features and Gameplay

#### 3.1.1 Story

A young child is sent to a neutral base camp called Askana. After the child's father dies in the massive battle occurring between the followers of each elemental God, and regular natural disasters continue to happen, the civilians of each nation blame the others for not worshipping the elementals. Askana is a base camp surrounded with myth and legend. It is rumoured that the creation of the land was started here, the meeting point between the Elemental Gods. Each tier of the dungeon represents an attempt by the Gods to create a balanced Planet, however, each attempt represents a more imbalance creation, resulting in a higher level of corruption, creating dark energy monsters. Throughout the dungeon, the player will meet previous adventurers who will trigger cutscenes, explaining their failings to continue, opening up the story to the player and more about the world around them. Towards the end of the game, the player will discover the truth that the Gods have fallen out resulting in them no longer wanting to work towards peace and balance, but individual supremacy. The dungeons were created in the past as the Gods worked together and the highest level is the perfectly balanced attempt and have been designed as a way of stopping people from finding them. Upon discovery of the fact, the Gods have fallen out, the player must work out a way of killing the Gods or uniting them to restore true balance.

#### 3.1.2 Dungeon Tiers

Due to the RPG nature of our game, all tiers of the dungeon will be playable at any time, however the increasing difficulty of monsters will make it hard for the player to walk around freely if they explore too deep. Doors and trapdoors can be used to find secret ways around into less dangerous rooms if the player wants to play a less combat intense approach. The dungeons will contain nodes, such as rocks, trees and monsters, that will regenerate or respawn for skill training or item gathering. The tiers will align with the story and therefore will be elemental themed and engaging. The first tier will be a “tutorial tier” where the core mechanics are introduced to the player.

#### 3.1.3 Dungeon doors and trapdoors

The dungeon will contain many doors and trapdoors that will be able to be unlocked by meeting certain requirements: having a key or being of high enough skill level (woodcutting or mining) to have unlocked (wood or metal) door breaking. This mechanism will allow the user to explore the dungeon to a further degree and can provide a different playstyle. A route to a chest with rare items could be found where combat level becomes less important to this player to progress deeper.

#### 3.1.4 descending and ascending

The player will be able to ascend and descend to different tiers of the dungeon by finding staircases, ladders or ropes.

#### 3.1.5 Monsters

Monsters will be found throughout the dungeon that provide the core challenge of the game. As the player gets lower in the dungeon, they will encounter more difficult monsters with higher attack damage and/or more health. In reward to the increased difficulty, the monster will potentially drop better rewards upon defeat for the player to pick up. The monsters will respawn after a cooldown allowing the game to continue forever.

### 3.1.6 Items/Inventory

A wide variety of items will be available in the form of tools and resources. The player will be able to hold them in their inventory or equip gear which will provide a boost to their base stats when worn.

### 3.1.7 NPCs

NPCs will be found in various locations and the player will be able to talk to them through predefined options. An NPC will have a specific function such as trading or storytelling.

### 3.1.8 Trading

A player will be able to exchange logs and ore for new gear at several shops throughout the dungeon. Rarer and higher volumes of resources will reward rarer gear with better stats and boost.

### 3.1.9 Levels/Player progression

Compilation of relevant activities will provide the player with experience points as the player levels up, increasing their base statistics, unlocking new abilities.

The combat skills can be increased by slaying monsters. Monsters which are a higher level and therefore more difficult, will reward the player with higher amounts of experience points and items dropped on the ground when defeated. As the player gains higher levels in the combat skills, their base statistics (weapons and armour) will increase. If a player is holding a shield and a weapon (melee or bow) the experience points gained from defeating a monster are evenly divided between defence and the other relevant skill. The combat skills are: Attack, Defence and Archery. On top of the combat skills, the player will have access to a range of non-combat related skills:

**Health** skill can be trained by finding gyms across the dungeons as well as taking 0 damage from monsters. For each health level gained, the player will receive 10 hit points. The player can restore lost hit points from a fight by either resting, or eating food which provides instant health and player boosts.

**Mining/Woodcutting** allows a user to extract ore/logs from rocks/trees, which can be used for trading armour and weapons. A higher level allows a user to hold better pickaxes/axes to extract rarer materials and therefore trade better gear. Experience points in this skill are gained by depleting a node. At certain levels the ability to break some doors or trap doors will be unlocked

### 3.1.10 Challenges

Our game includes challenges, at random intervals a player will be given the option of a challenge, where they must complete the given goal in 3 minutes. If successfully done, a reward will be given to the player in the form of XP or a random item.

## 3.2 Game Design Principles

### 3.2.1 Pursuing and Achieving Goals

Progression is an important part of our game with lots of paths to achieve such being present due to the multiple levels and range of gear available:

- The player can explore undiscovered areas of the dungeon to find new monsters and nodes
- Challenges will provide the player with a short-term goal (3 minutes).
- Fighting an enemy will provide a short-term goal (seconds to minutes).
- Training skills for better gear and better base stats is a constant short-term goal.

- Wide range of monsters allows the user to always be able to push themselves.
- There is the long-term goal of finishing the story.
- Having the best stats available is a long-term goal.

### 3.2.2 Game Interactivity

- Due to the challenges feature, a player will need to pay attention at random durations of the game.
- Monsters of higher difficulty will require the player to focus and play tactically with boost and food keeping them engaged.
- Monsters will automatically attack a player if the player is not of high enough level, the player will be unable to stand idle in many parts of the dungeon.
- The non-combat skills require focus, as the relevant items will need to be pressed (trees, rocks and food). The nodes will disappear and replenish.

### 3.2.3 Feedback

The game will provide feedback to the player at all times:

- Know where they are by looking at a static map with the dungeon tier that the player is in as well as already visited and completed tiers.
- Know their progress towards the 3-minute challenges.
- Know what items are in their inventory and currently worn on top of their boost.
- Know their current level and how close they are to levelling up.
- Know what resources are needed and the stats of what they'll receive for a trade.
- Know what requirements are needed to open a door/trap door if they don't meet such.
- Know what to do by going through an introductory tutorial in the beginning of the game and following the storyline.
- Use your own knowledge at certain situations and make choices by combining it with details, storyline, statistics displayed on the screen, damage, health points.
- Know how much XP a player is earning per task completion.
- Know what the players base stats are post-boost stats.
- Know their progress towards killing a monster.

The game will not focus on:

- The way to get the information from the interface. Everything that the player needs will be displayed on the main game panel or shown in a tutorial.
- How to defeat enemies using only mechanics. For the full enjoyment player needs to follow the storyline and remember collected equipment.

### 3.2.4 Gameplay Variety

There is a wide range of skills to train which provides the player with a wide range of activities to do:

- The mix of exploration, story and skilling provides 3 styles of play the player can choose.
- Various strategies can be played to defeat monsters, such as using a bow across a river or bringing an inventory of food.
- A large collection of resources, gears, items and monsters with individual benefits such as higher experience points or more damage will provide the player the opportunity to make decisions about gear to use throughout the game

### 3.2.5 Gameplay Consistency and Fairness

Fair gameplay will be created by:

- The ability to explore every tier of the dungeon at all times ensures that consistency exists as it allows the player to do everything depending on their skill level.
- Experience gain will be consistent for completion of the same activity.
- All enemies will be killable, and never impossible. The ease in which a player kills enemies will be dictated by their skill level or tactical approach.
- Skills will be naturally increased through gameplay; they will feel rewarded because the system will remain consistent.
- Depending on which themed dungeon they are in, the enemy types will be consistent with the style of the dungeon.

### 3.2.6 Avoiding Repetition

- Players will be able to skip cutscenes/chats with NPCs
- Dungeon tiers which are revisited will have areas the player previously couldn't access due to not meeting the requirements to do so.
- Player progression will make skills more exciting the more they use them, and therefore although certain weapon types from feeling repetitive.
- Players can replay the game with a completely different playstyle, avoiding repetition if they wish to do so.
- Players will be able to save their progress to avoid having to play the same parts every time.

## 3.3 Game Rules

### 3.3.1 Constitutive

- Enemies health is set to 100% as they spawn and will regenerate once out of combat.
- Enemies are defeated and disappear once the health reaches 0%.
- Enemies follow a player in a specified zone of a dungeon if the combat level of the player is less than that of the monster's combat level +5 and the monster is hostile.
- A certain number of enemies with corresponding strengths appear randomly in each dungeon.
- A player will gain experience points in a relevant skill if they complete the relevant action.
- If a player is holding a shield and a melee weapon or bow, experience points for defeating a monster will be evenly divided between both skills.
- Enemies inflict damage to the player depending on the defence level of the player and the boost the player currently has from armour, weapons and food boost.
- Players inflict damage to the monsters depending on the relevant attack level of the player and the boost the player currently has from armour, weapons and food boost.
- Player completes the story when the final Elemental God is defeated.

### 3.3.2 Operational

- A player can attempt all the automatically unlocked levels depending on the player's previous performance.
- A player can access each tier of the dungeon at any time and revisit them if need be.
- Equipment can be upgraded by completing challenges, trading in ore or defeating enemies.
- Experience points can be progressed by completing tasks or challenges.
- Combat skills can be progressed by defeating monsters.
- Skills can be progressed by completing related activities.
- Health can be instantly gained by eating the prepared food.
- Health will slowly regenerate when the player is not in combat.
- Defeated enemies drop a set of things that might be used later in the game.
- A player can collect things dropped by enemies by the right mouse click.



- A player can switch between weapons and shields by pressing the S key on the keyboard and selecting equipment to use.
- Arrow keys can be used to move around the dungeon.
- Right mouse click can be used to select things.
- M key can be used to open a static map.

### 3.3.3 Implicit

- More expensive weapons and shields are more powerful.
- Different weapons will have different attack styles.
- Enemies will continue hitting the player till either player's or enemy's health reaches 0%, or player moves from the enemies-controlled field.
- More powerful enemies are more likely to drop better things for the player to collect.
- Number of enemies with their corresponding strengths are dependent upon the difficulty of a level and expected player's equipment.
- Enemies inflicted damage is based upon the strength of the enemy.
- Player's damage is dependent upon the used equipment and skills.
- The percentage of health will be clearly displayed above the enemy and a player.
- A player can eat food to gain health.
- A player cannot gain more than 100% health.
- Throughout mining, a player can gain ore and use it to buy equipment.
- Equipment can be bought which costs less or equal to the value of ores the player has.
- A player moves around the dungeon to the direction of the pressed arrow key.
- Game state can be saved, paused.
- Game can be started from the beginning.
- Playing the game will not affect the computer in any permanent way.

## 4 Software Engineering

### 4.1 Software requirements

#### 4.1.1 Functional requirements

- F1: Shall display relevant information to the player through GUI, containing:
  - F1.1: Current place on a map.
  - F1.2: Current items and tools in inventory.
  - F1.3: Current items and tools equipped.
  - F1.4: Current player's health.
  - F1.5: Current enemy's health whilst in a fight.
  - F1.6: Current progress towards levelling up in each skill.
  - F1.7: Current progress towards a challenge.
  - F1.8: Message on why a player does not reach requirements to do a task.
  - F1.9: Current message from and optional replies when talking to NPCs.
  - F1.10: A save option to save progress.
  - F1.11: Option to open door/trap door.
- F2: Shall display animations including:
  - F2.1: Monster moving.
  - F2.2: Player moving.
  - F2.3: Monster fighting.
  - F2.4: Player fighting.
  - F2.5: Player gaining health through eating.
  - F2.6: Player gaining a resource when depleting a resource node
  - F2.7: Enemies dying.



- F2.8: Player dying.
- F3: Should play a variety of different sounds:
  - F3.1: Atmospheric background music throughout the dungeon
  - F3.2: Should play corresponding sounds on events as:
    - F3.2.1: When the enemy is defeated.
    - F3.2.2: The completion of a game.
    - F3.2.3: A resource node is depleted.
    - F3.2.4: Eating food.
    - F3.2.5: Woodcutting.
    - F3.2.6: Equipping gear.
    - F3.2.7: Item collection from the ground.
    - F3.2.8: Mining.
    - F3.2.9: Ascending into a higher tier.
    - F3.2.10: Descending into a lower tier.
    - F3.2.11: Player takes damage.
    - F3.2.12: Monster takes damage.
    - F3.2.13: A trade is made.
    - F3.2.14: The map is opened.
    - F3.2.15: A door or trap door is broken.
    - F3.2.16: A door or trap door is unlocked.
    - F3.2.17: When a save is made.
    - F3.2.18: When a challenge is completed.
    - F3.2.19: When the game is finished.
- F4: The player should be able to respawn at the base camp without resources.
- F5: Shall have weapons, shields and armour of different damage/protection.
- F6: Shall allow the player to exchange ore to new weapons/shields and upgrade the equipment.
- F7: Shall be able to record and display health of contributors:
  - F7.1: Shall allow the player to gain health points by eating.
  - F7.2: Shall allow the player to lose health points if hit by the enemy.
  - F7.3: Shall allow the enemy to lose health points if hit by the player.
- F8: Shall make enemies disappear once defeated.
- F9: Game shall be able to be saved.
- F10: Players shall be able to go up and down at ladders, stairs and ropes.
- F11: Players shall be able to unlock or break doors and trapdoors if they have the requirements to do so.
- F12: The game shall have various fightable monsters.
- F13: The game shall have various equitable tools.
- F14: The game shall have various resources (logs, ores and foods).
- F15: The player shall be able to move with the WASD keys.
- F16: The game shall have NPCs which can be talked to.
- F17: The game shall consist of multiple themed tiers, each with resource nodes and monsters.
- F18: The game shall have multiple skippable cutscenes.
- F19: The game shall include challenges for the player to complete

#### 4.1.2 Non-Functional requirements

- NF1: The game shall run on Windows, OSX and Linux operating systems.
- NF2: The project shall be programmed with Java version 11.
- NF3: Shall run on Java version 11 or newer.
- NF4: Shall use the Java Swing and JSFML libraries for the handling of GUI and Graphics.
- NF5: Shall load the game within 20 seconds on average hardware.
- NF6: Shall run consistently at least 30 frames per second.

## 4.2 Testing

Nr.	Description	Input	Expected Output
<b>SYSTEM</b>			
<b>1.1</b>	NF1: The game shall ran on Windows, OSX and Linux systems	Launch game on each OS	
		Load base camp on each on Each OS	Game botos and runs
<b>1.2</b>	NF2: The project shall be programmed with Java version 11	Compile Game on Java 11	Game complies and produces classes
<b>1.3</b>	NF3: Shall run on Java version 11 or newer	Run game on Java 11 or newer	Game boots and runs.
<b>1.4</b>	F10: : Game shall be able to be saved	Save game	Game is in the same state as it was when the game was saved prior to closing
		Close game	
		Reboot game	
<b>1.5</b>	NF5: Shall load the game within 20 seconds on average hardware	Boot game	Game boots and is playable within 20 seconds
<b>1.6</b>	NF6: Shall run consistently at least 30 frames per second	Boot game and play tier which is intensive	Game is playable and runs at a consistent 30 fps
<b>GAME'S GUI</b>			
<b>2</b>	F1 Shall display relevant information to the player through GUI, containing:		
<b>2.1</b>	F1.1 Current place on map	Open game	
		Check map	Player location corresponds with location on map
<b>2.2</b>	F1.2 Current items and tools in the inventory	Open a save various tools and equipment	
		Open Inventory	All tools and equipment in present and visible in GUI
<b>2.3</b>	F1.3 Current items and tools equipped	Open a save	Items which are currently equipped are presented in GUI
<b>2.4</b>	F1.4 Current player's health	Open game	Player's health is displayed on GUI
		Player has full health	Player's health is displayed as full on GUI
		Player has no health after taking too much damage	Players Health GUI is empty, death shown.
<b>2.5</b>	F1.5 Current enemies health whilst in a fight	Encounter an enemy	Enemies health is displayed on GUI

2.6	F1.6 Current progress levelling up a skill	Open a save.	Progress of skills is presented in GUI
2.7	F.1.7 Current progress towards a challenge	Open a save which is mid-way through a challenge	Progress of challenge is displayed in GUI
2.8	F.1.8 Message on why a player does not reach requirements to do a task	Encounter current inaccessible task	GUI displays that the player can't access tasks visually or via text.
2.9	F1.9 Current message from and optional replies when talking to NPCs	Encounter an NPC	
		Trigger talking option	Current messages and replies to NPC are present.
2.10	F1.10 A option to save progress	Open a game save	GUI presents a button which will allow the player to save.
2.11	F1.11 Option to open/trap door	Player walks past a door which they can open	GUI presents visually the player can use this door
		Player walks past a trap door which they can go through	GUI presents visually the player can use this trap door.
ANIMATIONS			
3	F2 Shall display animations including		
3.1	F2.1 Monster moving	Player encounters a monster	
		Player goes close enough to engage	Monster moves along with animation
3.2	F2.2 Player moving	User presses arrows keys to move	Player moves along with the animation.
3.3	F2.3 Enemies Fighting	Player triggers encounter with enemy	
		Enemy attacks	Enemy attack animation is triggered in sync with it's attack time.
3.4	F2.4 Player Fighting	Player triggers encounter with enemy	
		User triggers attack key (Sword)	Player attack animation is shown with a sword
		User triggers attack key (Bow)	Player attack animation is shown with bow
3.5	F2.5 Player gaining health through eating	User presses eating key	Player animation is show they have gained health is shown
3.6	F2.6 Player gaining a resource when depleting a resource node	User encounters a resource node	

		User engages with resource node	Animation for the resource node depleting is shown
3.8	F2.7 Enemies dying	Player encounters an enemy	
		Player causes enough damage on enemy to die	Enemy death animation is shown
3.9	F2.8 Player dying	Player encounters an enemy	
		Enemy causes enough damage on player to die	Health bar is 0 on player. Player death animation is shown
AUDIO TESTS			
4	F3 Should play a variety of different sounds		
4.1	F3.1 F3.1: Atmospheric background music throughout the dungeon on location.	Player enters certain dungeon	Music is played due to entering that dungeon, music continues until player leaves dungeon
4.3	F3.2 Should play corresponding sounds on events when:		
4.4	F3.2.1 When enemy is defeated	Player encounters and kills enemy	Corresponding sound when enemy is killed is played
4.5	F3.2.2 The completion of game	Set game to completed status	Corresponding sound when game is completed is played
4.6	F3.2.3 A resource note is depleted	Player finds a resource node	Corresponding sound is played when the player depletes resource node.
4.8	F3.2.4 Eating food	Player eats food	Corresponding sound is played when the player eats.
4.9	F3.2.5 Wood cutting	Player carries out wood cutting activity	Corresponding sound is played when the player wood cuts.
4.11	F3.2.6	Player equips an item node	Corresponding sound is played when item is equipped
4.10	F3.2.7 Item collection from the ground.	Player encounters enemy	Corresponding sound is played when the player picks up item
		Player kills enemy, loot is dropped	
		Player collects item	
4.11	F3.2.8 Mining	Player triggers mining event	Corresponding sound event is played when the player mines
4.12	F3.2.9 Ascending into a higher tier	Player ascends into a higher tier	Sound event is played when the player ascends into a higher tier
4.13	F3.2.10 Descending into a lower tier	Player ascends into a higher tier	Sound event is played when the player ascends into a lower tier
4.14	F3.2.11 Player takes damage	Player encounters enemy	

		Player takes damage	Sound event is triggered when the enemy damages the player.
4.15	F3.2.12 Monster takes damage	Player encounters enemy	
		Enemy takes damage	Sound event is triggered when the player damages the enemy.
4.16	F3.2.13 A trade is made	Player triggers a trade event	Corresponding sound is made when the player trades
4.17	F3.2.14 The map is opened	Player triggers opening map event	Corresponding sound is made when the player opens the map.
4.18	F3.2.15 A door or trap door is broken	Player triggers event in which a door breaks	Corresponding sound is made when the door first breaks.
		Player triggers event in which trap door breaks	Corresponding sound is made when the trap door initially first breaks.
4.19	F3.2.16 A door or trapped door is unlocked	Player unlocks a door	Corresponding sound is made when the door is first unlocked.
		Player unlocks trap door	Corresponding sound is made when the trap door is first unlocked.
4.20	F3.2.17 When a save is made	Player triggers save event	Sound event is played when the sound save has been completed.
4.21	F3.2.18 When a challenge is completed	Player completes a challenge	Sound event is played when the player completes a challenge.
4.22	F3.2.19 When the game has finished	Player completes game	Sound event is triggered when the player corresponding to when the player completes the entirety of the game
MISC TESTS			
5	F4 The player should be able to respawn at base camp without resources	Player opens save	
		Player dies	Player respawns at base camp without resources
6	F5 Shall have weapons, shields and armour of different damage/protection.	Save is opened containing all types of weapons and armour within inventory	All weapons and inventory are available
7	F6 Shall allow the player to exchange ore to new weapons/shields and upgrade the equipment.	Player encounters a trade event	
		Player triggers a transaction with the use of ore	Weapon is upgraded OR Shield is upgraded
HEALTH TESTS			

8.1	F7.1 Shall allow the player gain health points by eating	Player finds food source	
		Player consumes food	Players' health is increased on GUI and on records.
8.2	F7.2 Shall allow the player to lose health points if hit by an enemy.	Player encounters enemy	
		Player is hit by enemy	Players' health is increased on GUI and on records.
8.3	F7.3 Shall allow the enemy is lose health points if hit by the player	Player encounters enemy	
		Player hits enemy	Enemy's health is increased on GUI and on records.
9	F8 Shall make enemies to disappear once defeated	Player encounters enemy	
		Player kills enemy	Enemy disappears after period of timer
10	F9: Game shall be able to be saved.	Player triggers save event	
		Player closes game	
		Player reopens game	Game is in the same state as it was prior to save
11	F10: Players shall be able to go up and down at ladders, stairs and ropes.	Player encounters a stair mechanism (Ladder, stair, rope)	
		Player clicks icon	Player moves location.
12	F11: Players shall be able to unlock or break doors and trapdoors if they have the requirements to do so.	Player encounters a door within their requirements	
		Player triggers breaking door event	Door is unlocked and usable.
14	F12: The game shall have various fightable monsters.	View enemy assets	Various enemies are shown
15	F13: The game shall have various equitable tools.	View tools assets	Various tools are shown
			Player equips tools
16	F14: The game shall have various resources (logs, ores and foods) .	View resource assets	Log assets are shown
			Ore assets are shown
			Food assets are shown
17	F15: The player shall be able to move with the WASD keys.	Player presses "W" key	Player moves up.
		Player presses "A" key	Player moves left.
		Player presses "D" key	Player moves right.
		Player presses "S" key	Player moves down.
118	F16: The game shall have NPCs which can be talked to.	Player spawns	
		Player encounters NPC	

		Player triggers talking event	Player can talk to the user.
19	F17: The game shall consist of multiple themed tiers, each with resource nodes and monsters.	Load each individual tier	Visually, dungeons have a clear differentiation
20	F18: The game shall have multiple skippable cutscenes.	Player encounters cutscene	Player has the ability to skip cutscenes.
21	F19: The game shall include challenges for the player to complete	Player is playing game	Player encounters a challenge they can interact with

## 5. Implementation Plans

### 5.1 Task List for Term 1

**Week 3:** During the first workshop we started the team building process by introducing ourselves, discussing interests, strengths as well as the workflow and expectations for this project. Brainstorming game ideas and taking inspirations from Google Play/Steam formed a list of games we would want to implement. After the workshop each member of a group individually thought of a few more games and shared it in a group discussion. By combining ideas, we settled on creating a dungeon crawler.

**Week 4:** Our idea on a dungeon crawler got approved, thus we started developing this game more in depth. For the game to be engaging, it needs a combination of a thrilling storyline and interesting mechanics. Also, developing the game for a single player sounded reasonable with the amount of team members and time we have. After the workshop, we continued on brainstorming functionalities and the storyline of the game. We agreed upon creating multiple dungeon tiers and hopefully at least one quest for each of them with a varying number of enemies. We thought of 3 things a player could do within a game that would help him to get through the levels (cooking, mining and woodcutting). With all these ideas we started writing the game design report up to section 3 in the document.

**Week 5:** After having a portion of our design document written, we all read through parts that the other team members worked on, merged our ideas even more and got a clear understanding how the game will work in more detail. We removed a few things and settled on creating 10 dungeon tiers. The storyline and enemies got developed and we started working on software requirements/acceptance tests.

**Week 6:** During the workshop we started working on implementation plans. We were able to divide our game into tasks and structure it for easier development. We settled on deadlines for each module and thought of specific sounds/graphics that the game should have. Meeting after the workshop was used to get some more feedback from other team members on the tasks we did, thus we peer read each other's work and edited it to our understanding. We started writing the project plan for term 2 and structured our ideas into java classes.

**Week 7:** The goal for this week was to create Gantt and the Activity Network with Critical Path charts as well as to do final changes to the document and get it ready for submission. Reading through the document allowed us to unify our writing styles and add additional information. Final amendments were done and the document was formatted. We made sure that no mistakes are left within the report and that everything is coherent. We removed the unnecessary information to fit into 20 pages. During the manufacture of the Gantt chart, we realised that we would not be able to complete the game with the features we initially wanted due to our reduced group size. Because of this we filtered through our work and found the not needed game elements, and removed them from the plan.



## 5.2 Project Plan for Term 2

Project plan for term 2 consists of developing a game. Table below describes milestones that we intend to reach each week. These goals are meant to be completed by 3 members of a team because we did not get any engagement from others.

PROGRAMMING				
ID	Milestone	Dependencies	Deliverables	Deadline: End of Week
P1	Program an unstyled core GUI with placeholders	None	Java class files providing functionality for the core GUI	1
P2	Program the core functionality for a player	None	Java class files providing the functionality for the player	2
P3	Program the core functionality for a dungeon tier	None	Java class files providing the core functionality for a dungeon tier	3
P4	Program the core functionality for an enemy	None	Java class files providing the core functionality for an enemy	3
P5	Program the core functionality for a challenge	None	Java class files providing the core functionality for a challenge	2
P6	Stylise the core GUI	P1, G1	Amended Java class files for the core GUI that improve on the appearance	2
P7	Program each dungeon tier	P3, C1	Java class files that add each dungeon tier of the game	5
P8	Stylise each dungeon tier	P7, G1	Amended Java class files for the core GUI that improve on the appearance	6
P9	Program each dungeon challenge	P5, C2	Java class files that add each dungeon challenge to the game	3
P10	Stylise each dungeon challenge	P9, G2	Amended Java class files for the core GUI that improve on the appearance	4
P11	Program each enemy	P4, C3	Java class files that add each enemy to the game	4
P12	Stylise each enemy	P11, G3	Amended Java class files for the core GUI that improve on the appearance	5
P13	Program the flow of the game	P7, P9, P11	Java class files that create instances of other Java classes to create a working game	6
P14	Program the increasing difficulty of a game	P3, P4, P5	New and amended Java classes with the increasing difficulty of a game	7

<b>P15</b>	Add background music	S1,S8	New and amended Java class files that add sound effects	6
<b>P16</b>	Program the tutorial level	P7, P9, P11, P13, P15, P20	New Java class files that add tutorial level to the game	6
<b>P17</b>	Program and implement the storyline	P8	New and amended Java class files that add the storyline to the game	7
<b>P18</b>	Implement eating	C5, G6, S4	Java class files that add eating to the game	7
<b>P19</b>	Implement mining	C6, G7, S3	Java class files that add mining to the game	7
<b>P20</b>	Implement woodcutting	C7, G8, S5	Java class files that add woodcutting to the game	5
<b>P21</b>	Implement item collection from the ground	C8, G9, S6	Java class files that add item collection from the ground to the game	5
<b>P22</b>	Implement saving	P1	Java can be saved	2
<b>P23</b>	Create core functionality for NPCs	None	Java class files that allow the ability to create NPCs	2
<b>P24</b>	Implement trade feature	P23	Java Class files that allow the trade feature to function.	4
<b>P25</b>	Implement fighting	P2, P4, C9, G10, S7, S2	Java Class files are added which allow for encounters to occur.	4

### CONTENT DESIGN

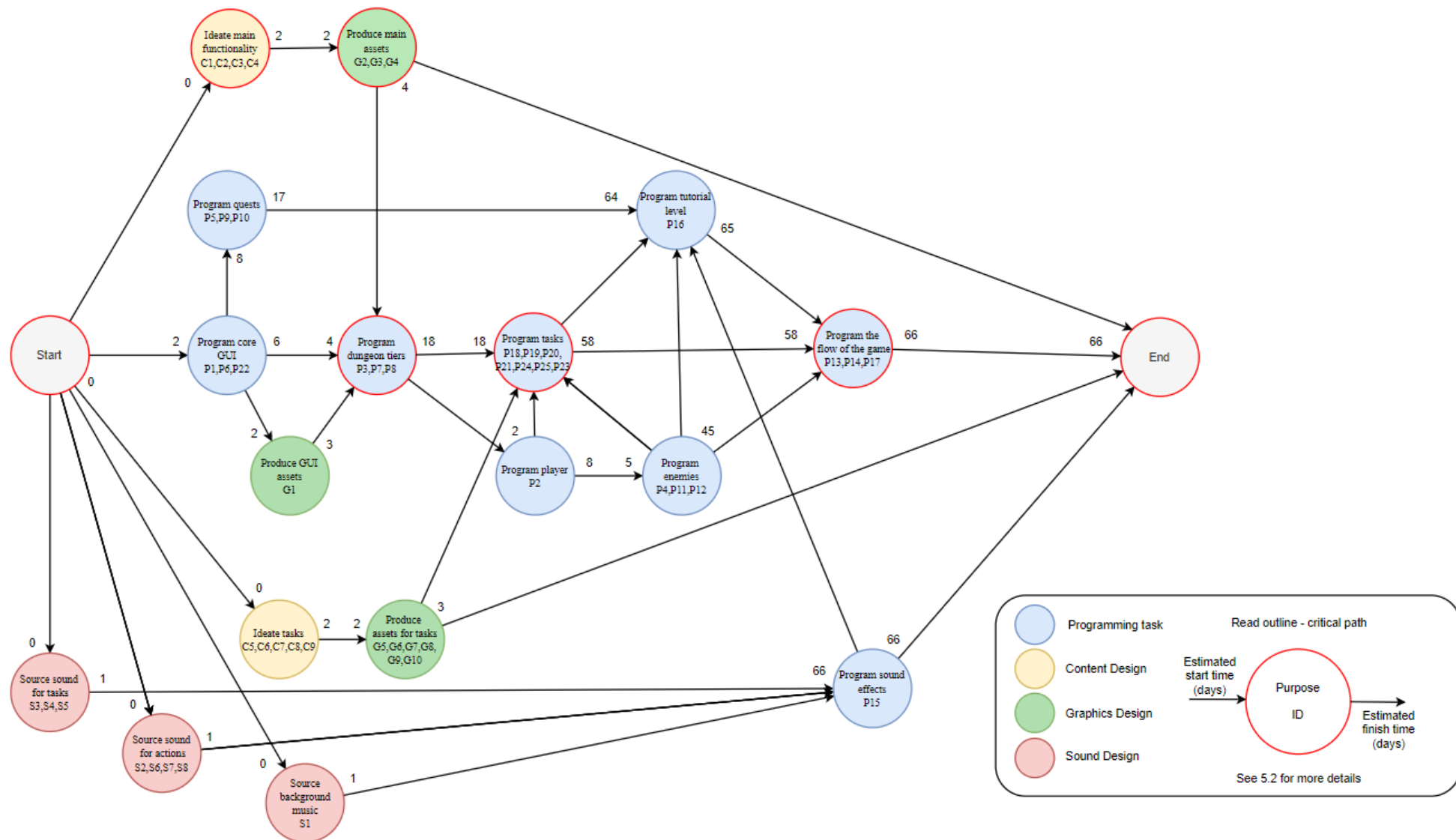
<b>ID</b>	<b>Task</b>	<b>Dependencies</b>	<b>Deliverables</b>	<b>Deadline: End of Week</b>
<b>C1</b>	Ideate 10 dungeon tiers	None	Document describing dungeon tiers	1
<b>C2</b>	Ideate 1 challenge per tier	None	Document describing challenges	1
<b>C3</b>	Ideate 20 enemies	None	Document describing enemies	1
<b>C4</b>	Write a script for the storyline	None	Script for the storyline including the script for the tutorial as well	2
<b>C5</b>	Ideate eating	None	Document describing eating	1
<b>C6</b>	Ideate mining	None	Document describing mining	1
<b>C7</b>	Ideate woodcutting	None	Document describing woodcutting	1
<b>C8</b>	Ideate 3 things that can be dropped on the floor by the enemy	None	Document describing 3 things that can be dropped on the floor by the enemy	1
<b>C9</b>	Ideate fighting	None	Document describing fighting	1

### GRAPHICS DESIGN

ID	Task	Dependencies	Deliverables	Deadline: End of Week
G1	Produce assets for the core GUI	None	.PNG files for each required asset to create the core GUI.	1
G2	Produce assets for dungeon tiers	C1	.PNG files for each required asset to render dungeon tiers.	2
G3	Produce assets for challenges	C2	.PNG files for each required asset to render challenges.	2
G4	Produce assets for enemies	C3	.PNG files for each required asset to render enemies	2
G5	Produce assets for food	C5	.PNG files for each required asset to render food	2
G6	Produce assets for eating	C5	.PNG files for each required asset to render eating	2
G7	Produce assets for mining	C6	.PNG files for each required asset to render mining	2
G8	Produce assets for woodcutting	C7	.PNG files for each required asset to render woodcutting	2
G9	Produce assets for things dropped on the floor	C8	.PNG files for each required asset to render things dropped on the floor	2
G10	Produce assets for fighting	C9	.PNG files for each required asset to render fighting	2
<b>SOUND DESIGN</b>				
ID	Task	Dependencies	Deliverables	Deadline: End of Week
S1	Source a background music for each themed level	None	.WAV files for each required soundtrack	2
S2	Source sound effects for weapon/shield selection	None	.WAV files for each required soundtrack	2
S3	Source sound effects for mining	None	.WAV files for each required soundtrack	2
S4	Source sound effects for eating	None	.WAV files for each required soundtrack	2
S5	Source sound effects for woodcutting	None	.WAV files for each required soundtrack	2
S6	Source sound effects for enemies dropping things on the ground	None	.WAV files for each required soundtrack	2
S7	Source sound effects for fighting	None	.WAV files for each required soundtrack	2
S8	Source music for the completion of a game	None	.WAV files for each required soundtrack	2

## 5.3 Activity Network with Critical Path

Diagram below shows the project network activities with a critical path including the longest duration of planned milestones.



# 5.4 Gantt Chart

The Gantt chart has been designed to create a schedule of tasks, making sure dependencies are reached before hand, so that workload can be managed and progression is ensured. We made an estimate of how long each task will take and added it to the Gantt chart, and included a slack period in which allows us to catch up if a deadline is missed due to various complications.

