SPECTRA

Game Design Document

MASK-ON GAMES

Version	Date	Author	Changes
0	01/02/21	Joseph Skinner	Creation of Document
1.0	15/02/21	Neil Tubbs-Bates	First Draft Complete
1.1	16/02/2021	Joe Skinner	Antagonist and Protagonist names changed
1.2	01/03/2021	Joe Skinner	Enemies and locations updated
1.3	03/03/2021	Joe Skinner/Neil Tubbs Bates	UI details added
1.4	21/03/2021	Joe Skinner/Neil Tubbs-Bates	Locations updated and expanded
2.0	9/04/21	Joe Skinner / Neil Tubbs-Bates	Second draft complete
2.1	10/04/21	Joe Skinner	References Added
2.2	11/04/21	Samuel Wort / Samuel Robinson	Technical Specifications added
2.3	12/04/21	All	Minor revision and expansion
2.4	29/04	Joseph Skinner	Refining details, removing unneeded information

Game Features	Specification
Title:	Spectra
Game Engine and Type:	Unity 2D single player
Screens / Levels	Main Menu, Overworld, 6 main levels, 8 optional levels, 5 taverns, castle
Style	16 bit sprites / tiles
Sounds and Audio	Royalty free FX, custom soundtrack
Physics	Box2D, Cylinder2D, Tilemap 2D
AI	A* Pathfinding
Controls	Arrow keys movement, X attack / interact, V heal, LShift rotate Orb, C use ability, ESC pause
Screens / Levels	Specification
Title	Splash Screen, Backstory, Settings + Credits
Settings	Sound, Video, Controls
Play Levels	Gameplay + Story

TABLE OF CONTENTS

1. Introduction

This document encapsulates the brainstorming, design and implementation processes for our Game Studio Project Assignment, provisionally titled *Spectra*. It will detail ideas and be a record of developments discussed within meetings. Also included in the document are independant design proposals as of 01/02/2021.

1.1 Team

Name	Role and responsibility
Samuel Robinson	Team leader- *-Team management* *-Scheduling* Programmer- *-Save management* *-Pathfinding*
Sam Wort	Lead programmer- *-Enemy implementation* *-Boss implementation* *-Player implementation*
Joe Skinner	Level Designer- *-Level implementation* *-Dungeon Design* *-Overworld Design* Programmer- *-Mechanics implementation* UI designer- *-UI design* *-Menu design* *-UI implementation* *Audio Engineer -* *-Track Creator* *-SFX creator*
Neil Tubbs-Bates	Organisation manager- *-Meeting scheduler* *-Minute keeping* Programmer- *-Overworld implementation* *-General scripting* *-Menu Implementation* Story designer- *-Story implementation* *-Story designing*
Lewis Rogers	Lead designer- *-Dungeon design* *-Puzzle design* *-Boss design* Audio engineer- *-Track creator* *-Audio implementer*

Samuel Robinson, is our Team Leader due to his excellent initiative and ability to find resolve within conflicting ideas. He was chosen as he has exemplified that he will immediately act on any issues and resolve them quickly, i.e. contacting outside sources and setting up the project and communal data storage. On top of this, Samuel Robsinson will be programming the game's core mechanics.

Lewis Rogers is a Game Designer and talented musician, and therefore will be working on the musical accompaniments of the project. He has produced great results in his previous games where he took on a similar role. On top of this, Lewis will assist the team in the conception of the core gameplay loop and will develop levels.

Joseph Skinner is a Level Designer for this project, alongside programming segments of the game. His primary role is to create the levels and introduce the framework coding to be passed on to the most applicable programmer. Joseph has also contributed to the production of the game's soundtrack.

Neil Tubbs-Bates is Story Director and is in control of all narrative decisions. In addition, Neil will also serve as a programmer; he will develop the overall flow and implement the menus of the game, as well as assisting in the completion of other tasks.

Samuel Wort is our Lead Programmer and Animator due to his programming skill and capability. He will program the game as well as assist others with their scripting decisions. He has unique programming skills that will prove very useful when it comes to polishing the game into the final product.

1.2 Key principals

The key principles we will stick to are consistent communication and transparent scheduling. We will maintain communication though discord channels and through weekly meetings. Transparent scheduling is also important to prevent group members getting stuck. If someone is having issues or will not be able to work for a certain period of time, it is imperative to tell the group in advance, so we can avoid being blindsided by team members' personal lives. If issues arise with the set work, they can be talked over and a group solution could be found quickly.

1.3 Meetings and progress

In the first week of the project a GANTT chart was drawn up (see appendix) and a simple version of this document was created so certain aspects of the document could be filled out early on and during the development phase of the game. The Gannt chart was created with a section dedicated to each team member, more focusing on each individuals' strengths and where they would work most efficiently.

Game Gantt chart								
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8
Sam.R	Design earth dungeon	Design final dungeon	Implement save system	Work on extra dungeon assets	Work on extra enemy/player assets	Implement enemy animation	Implement boss and player animation	Building and submitting
Joe	Design tutorial area	Design UI and menus	Implement dungeons	Implement dungeons	Implement overworld	Implement inventory	Implement inventory	Play testing and bug fixing
Lewis	Design water dungeon	Design enemies	Work on dungeon music	Work other music	Work on sound effects	Implement music	Implement sound effects	Play testing and bug fixing
Sam.W	Design fire dungeon	Design bosses	Implement enemies	Implement enemies	Implement enemies	Implement bosses	Implement bosses	Play testing and bug fixing
Neil	Design storm dungeon	Design puzzle	Implement Menus	Implement UI	Implement Pausing	Add details to the GDD	Add details to the GDD	Play testing and bug fixing

Team meetings were held once a week on Mondays, and the group also met up on Tuesdays to confer with Michael Smith, our mentor, to ask for advice and keep him updated on our progress as well as talking more about the project if necessary. Each meeting was organised by Neil and led by Samuel and began with a progress check of each member's objectives given at the previous week's meeting. This was followed by an evaluation and discussion of each implemented element by the other team members, to ensure that everything implemented was understood and agreed upon by all members. Each meeting was ended by comparing progress against the GANTT chart and going over what will try to be achieved in the coming week.

The overall objective of each meeting was to ensure that things were running smoothly for each member and to collaboratively fix any problems or discuss any issues that had arisen during the previous week's work.

1.4 Documentation and communication

Most of the team's communication will occur through a discord server, not only will weekly meetings be held here but there are multiple channels set up for file sharing, quick communication, and meeting reminders. Having quick access to each other and to all files and assets communally will encourage highly responsive and active communication.

Unity's inbuilt version control (which will be touched on later) also allows for basic communication as when the project file will be updated, a note could be included saying what was added by whoever made said update.

Progress monitoring will occur on a near daily basis mainly through the discord channels where we will mention developments in our work, and on a more official basis during team meetings once a week. It is Neil's job to organise and run said meetings to ensure deadlines are being met each week.

1.5 Design Goals

Our goal with Spectra is to create a fun single-player RPG in which a hero traverses a range of different levels to collect the items needed to vanquish a great evil. Our aims are for players to utilise the tools provided to them to overcome challenges to progress through the game and explore the expansive environment.

1.6 Influences and Sources

We will draw heavy inspiration from Shining Wisdom (1996) and The Legend of Zelda (1986) in reference to the gameplay and aesthetic elements. Both use a top-down view of a character who fights their way across the land to save the day. We are using tile maps to create the scenery as this will help maintain aesthetic consistency. This ensures that there will not be any jarring changes in visuals when changing scenes; it is important not to detract from the immersive player experience in this way.

1.7 Target Audience

We are aiming our game at male players that are fans of old-school adventure-game fans.

2. Implementation plan

2.1 Initial planning and Game Justification

During the initial phase of the project, we drew up a GANTT chart and collectively decided that Unity would be our engine of choice, as it has in-built version control and we all have prior experience with it.

Upon release of the assignment, Joseph Skinner and Neil Tubbs-Bates formulated an idea that they proposed to the group. We agreed to continue with the concept, which allowed us to start development sooner.

We decided to produce a 2D Zelda-inspired action game for multiple different reasons. Firstly, we decided to make the game 2D as we were all more familiar with it so we wouldn't have to waste time learning new things at every step of the project. On top of this, we decided that to differentiate our game from similar games on the market we would allow the majority of our game to be completable in any order, having new upgrades and powers provide different solutions to problems instead of strictly getting more powerful with each dungeon completed. This sense of exploration and unique approach to difficulty is designed to differentiate our game from others like it and give it a unique selling point that our potential competitors wouldn't have.

2.2 Version Control / Management

We used Unity to build our game and for our version management we used it's inbuilt version control, "Unity teams". This is a simple addition to our engine that stored a master copy of our project in the cloud. Any changes to the projects were published to this master copy and re-synchronised with everybody else's files.

2.3 The Agile Method

We planned our project with the agile method in mind as this plan would enable us to create our projects in sprints. This type of workflow allowed us to work and implement our own parts of the project (enabling us to work to our individual strengths) and also allowed us to have a quick turnaround on features, and by breaking things into short sections in terms of planning it allowed us to adapt dynamically if things didn't work as intended. This process was also chosen as it is a lot easier for smaller teams to implement with smaller teams as we could all work on our own things within the sprints and our small group and constant communication made giving feedback much more quick and effective.

2.3.1 Roles in agile

Role	People
Scrum master	Neil Tubbs-Bates- *-Has final say over the final sprint plan* *-Ensures all team members are on track*
Team members	Everyone- *-Partakes in each sprint* *-Is responsible for sprints being completed*
Product owner	Everyone- *-Contributes to the overall vision of the project* *-Has an input on what each sprint contains*

2.3.2 Agile in meetings

As we had a GANTT chart that all the team members could refer back to, the main way that Agile affected our meetings was the discussion of sprints. We could discuss our objectives for the following week, but we mainly used our sprints to decide what these weekly objectives would be. Each sprint lasted around 3 weeks and if a

sprint had been completed that week we could take an extra long meeting to tweak and prepare the sprint for the next 3 weeks. The GANTT chart served as more of a guideline and the sprints themselves served as a detailed and bullet-pointed list that could be followed by each member.

2.3.3 Agile sprints

Sprints	**Objectives**
Sprint 1 **Duration- 2 weeks** **Begins- 8th March** **Ends- 21st of March**	-Brainstorming -Dungeon design -Puzzle design -Boss design -Enemy design -UI design
Sprint 2 **Duration- 3 weeks** **Begins- 22nd of March** **Ends- 11th of April**	-Asset creation -Music creation -Dungeon implementation -Enemy implementation -Menu implementation -Control implementation -Save system implementation
Sprint 3 **Duration- 3 weeks** **Begins- 12th of April** **Ends- 2nd of May**	-Boss implementation -Animation implementation -Music implementation -Sound effect implementation -Inventory implementation -Play testing -Bug testing -Building and submission

3. Specifications

3.1 Concept

You, the player, are tasked with defeating a dark threat looming over the fantasy kingdom of *Spectra*. Using your natural skills as a fighter and aptitude for magic, you make use of an arsenal of weapons to defeat enemies, clear dungeons, and reap the rewards. To ultimately Defeat the game's primary antagonist, the player must visit ancient ruins and dungeons in search of the power they each contain.

Spectra is a 2D Top-down RPG in which the player fights their way through various levels, solving puzzles and defeating powerful foes. It is our goal to make a game that brings new life back to the widely forgotten era of simple 2D RPGs, and to prove that fancy 3D graphics and lighting isn't the most important aspect of a game. Instead, we believe that thought and effort in the minor details are the key aspects to making a great game.

3.1.1 Genre

Being a Role-Playing Game, there is a range of generic conventions that integrate into said genre:

- Classically RPGs are set in fictional worlds wherein logic and physics are not applicable. An example of this is the inclusion of magic and magical creatures, or locations that defy our understanding of the world.
- They take place in a time period different to that of today. For example the dark ages or a dystopian future are common eras and locations.
- A heavy emphasis on character. You play the role of "the hero", who is the only person able to prevent calamity.
- A heavy emphasis on exploration. Other genres of game will transition from one level to the next until the end of the game; but Role-Playing Games take a different approach where the entire game often takes place in one sprawling landscape.

3.1.2 Concept Art

Fig. 1 - Earth Dungeon concept drawn in Excel

The dungeon levels are planned individually using an Excel spreadsheet or MS Paint to best emulate a tilemap grid. These designs provide the framework for the levels and allow for better puzzle planning. Excel also assists in formulating room sizes and the dungeon's overall shape and flow. Included within these designs are the level boundaries, puzzle mechanics and obstructions. Levels that did not include puzzles that required detailed planning were created with basic drawing applications, such as MS Paint.

These designs are then sent over to the appropriate team member who will begin working on implementing the level into the game [more excel level plans included in Section 8].

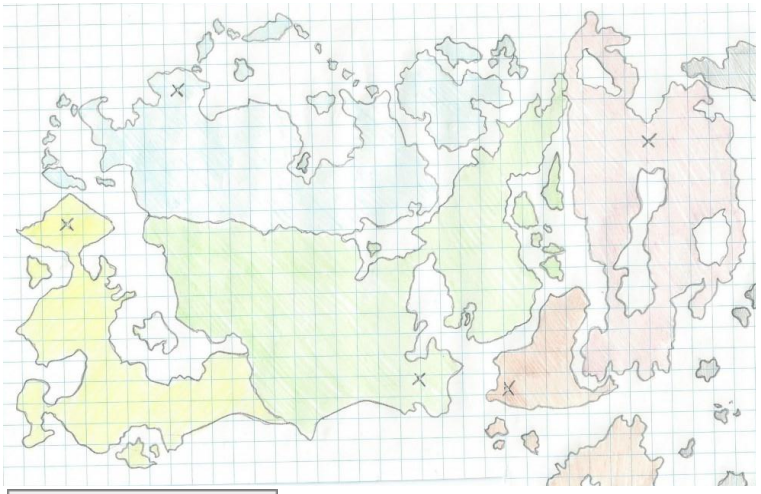


Fig. 2 - Initial Overworld Sketch

Fig. 3 - Mid-stage of Overworld implementation

As for the Overworld, the level was developed using an iterative process in which a hand-drawn sketch was used to draft the original design [fig. 2]. This design was recreated in Unity, and over the course of the game's development was refined into the final product. Changes to the Overworld include moving significant locations, increasing the size of open areas to better accommodate combat and enemies, and even implementing more islands to give the level the sense of scale it needed.

After this was completed, we worked on the camera and how large the game view should be. We found a balance that allows the player to not only see the most optimal view area around them, but for it to also not be too large to diminish the size of the player.

As an aide to the player, the Overworld scene will include a minimap at the top right of the screen to suggest travel routes ahead of time and to also highlight significant locations that can be entered (minimap elaborated in section 3).

3.2 Story and Narrative

Spectra revolves around a brave hero named Lux setting off on an adventure. To start with, the player will be equipped with a sword alongside a bow and arrow. Your arsenal can grow with the introduction of abilities and magic. The goal is to find the powerful elemental masks and powers that must be united in order to vanquish the great evil. These will be required for puzzle-solving skills to reach the end. On the way the player will encounter enemies with a range of abilities that they will need to defeat. Once all of the dungeons have been cleared you can access the final dungeon in which you face off against the primary antagonist of the game, Nox. Nox is the antithesis of Lux, possessing the same abilities but also using unique attack patterns.

3.2.1 Theme

The opposing forces of dark and light are the main motifs of the game, with the player encapsulating the idea of light with his heroics and Nox being the reflection of darkness and evil. The names of the protagonist and antagonist are interwoven with this theme; Lux being Latin for light and Nox as Latin for dark.

3.2.2 Tone

The Overworld is in a state of perpetual daylight. This will represent the fact that Nox's plan to End the World is yet to be fully realised, but the dungeons will have a gloomy, haunting atmosphere to invoke dread and the feeling of danger within the player. Some enemies have a different elemental type, and in turn will have a corresponding colour scheme.

3.2.3 Characters

Lux - The Player controls this character, who is destined to halt calamity.

Nox - The primary antagonist, hell-bent on shrouding the world in darkness. A dark palette swap of Lux.

3.3 Game Structure

3.3.1 Player

The game is single player.

3.3.2 Action

Weapons are for both ranges of combat but do not have a corresponding element. Magical attacks have greater range and can be used against enemies that have distance from the player.

3.3.3 Objectives

The game has the end goal of defeating Nox, but to do so involves the completion of different, smaller objectives (see 3.3.3.1 for examples).

3.3.3.1 Dungeons

Areas that require the player to either:

- Reach the end
- Solve the puzzle
- Kill the infestation of enemies
- Defeat the boss

Completing the dungeon will reward the player in some capacity and / or progress the story.

3.3.3.2 Villages and Taverns

Areas in the overworld that the player can visit to heal and recover, explore,and save. Some villages have been taken over by monsters and the player can help get rid of them.

3.3.4 Bosses

In each dungeon there is a large room that is designed to contain but due to time constraints, these bosses were not added to the final build. Each boss was designed to fit thematically within its respective dungeon and although they weren't implemented and no art was made for them, they each had an initial design with attacks and phases planned out and the table below was drawn up during the design period.

Dungeon	Name	Basic attack	Special attack	Design/Extras
Earth	Shaherra	Sending slider enemies towards the player	Sending large a wave of 'sliders' at a time	Shaherra operates in phases and is only vulnerable between said phases, each phases consists of dodging enemies spawned by the boss
Storm	Khanrai and Khathyr	Pushing the player/Summoning lighting from the sky	Releasing a large pattern of lightning/Pushing enemies off screen towards the player	Khanrai and Khathyr is designed to be two separate bosses, the wind at the lightning boss, each would have their own hit box and unique attacks but take turns attacking as to not overwhelm the player

Water	Czarime	Fires homing bolts of water and faster, straighter blocks of ice	Split the screen and reduce the movement area with jets of damaging water	Czarime is the most conventional but operates from range and will dart away when hit by the player, due to the boss being harder to hit they have much less health though
Fire	Rexash	Sweep of fire across one axis	Shoots a meteor that leaves fire on the ground	Rexash is designed to be more melee focused and the fire created by his sword slashed and the meteor fire adds challenge as it is harder to get closer to him
Final	Nox	See above	See above	As Nox is the culmination of all the elements he would use a combination of all the bosses attacks, although this seems tricky the player should be able to deal with all the previous attacks easier after obtaining all the powers

3.4 Graphics

The graphics will consist of 16 x 16 character sprites and tilemaps sourced from Pita from Itch.io [see references, section 8.2].

3.4.1 Landscape

Spectra will be viewed from a top-down perspective. The view size will be dependent on the level dimensions and layout. The camera will be mostly fixed in confined rooms but will move along an axis if said room is elongated in any direction. Dungeons will have a smaller view area while the overworld will be more zoomed out to show a significant amount of area around the player. Certain tiles, specifically flowing water and flickering torches will be animated to add life to the surroundings.

3.4.2 Objects

The player will be 16 x 16 sprites taken from a spritesheet..

Enemy sizes will vary depending on the species. Certain enemies will also have elemental variants that change the way they look and play.

3.4.3 Lighting

Spectra will make use of the Unity 2D Render Pipeline in order to implement lighting into the dungeon areas. This is to mimic torchlight and provide dark spaces to build the atmosphere.

3.5 Game Flow

The game begins in the first mandatory level, the “Tutorial Dungeon”. This dungeon is placed under a castle crypt, with the player being sent down there to clear out the monsters that have overstayed their welcome. Upon exiting, the player will be taken to the castle. From here, the player can explore the area or move on to the overworld. To do this the player simply follows the dirt track away from the castle entrance and houses. The overworld is a sprawling series of islands, the biggest of which is the central “hub” of the land. This hub island is where the player will begin once leaving the castle, allowing the freedom to choose where to go next instead of explicitly guiding the player onwards.

4. User Interface

4.1 Intro

The Main Menu will be the first thing that the player sees. It will have options to play, view setting or quit and also have our Mask On Games Logo in the corner

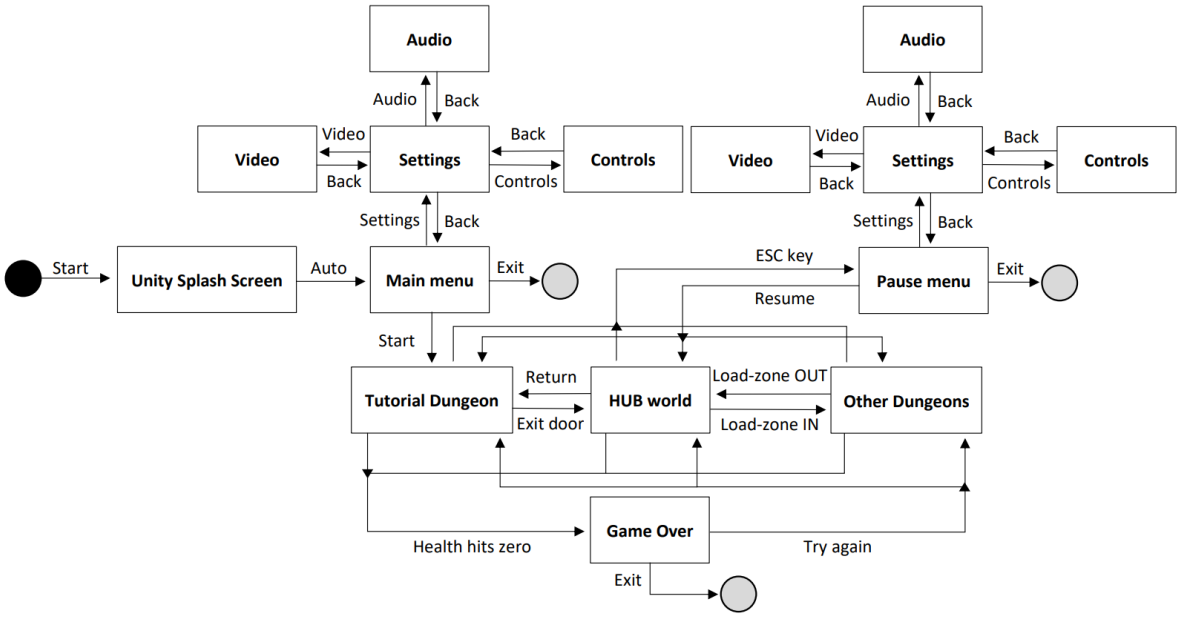
4.2 Screen Displays and UI

Fig. 4 - Overworld minimap

The game's UI is designed to be as minimal as possible. This displays the player's current health and magic as vertical bars in the bottom left of the screen, as shown by red and blue bars. Potion sprites appear next to the respective UI elements to show the player can restore a partial amount of a resource when that specific potion has been collected. A gold counter displays the players' current amount of gold. The Overworld scene will include a minimap to help guide the player and to reveal incoming enemies. The player is represented by a blue dot and enemies will be red.

4.3 Screen Flow and Menus

Our menus are designed to be themed to fit the game and display all necessary information to the player regarding their health and magicka, vial count, and which objectives they have completed.



4.3.1 Main Menu

The Main Menu of the game will include the *Spectra* Logo and the list of options in a medieval stylized font. It will include these options:

- Play
- Settings
- Exit

4.3.2 Pause Menu

Spectra's pause menu like the main menu features the options to play the game, access the settings and to quit to close the game entirely. The menu also features a drawing of the main character Lux again for visual appeal. Unlike the main menu the pause menu can also be used to track progress as the players' collected powers will be displayed on this menu.

4.4 HUD

The HUD for spectra features many elements designed to visually indicate information in the game to the player. In the bottom corner of the screen are two vials one red and one blue, the red bar is the player's health which is affected by damage and obtaining health pickup. The blue bar indicates the player's power recharge, this depletes on use of powers and recharges when said powers are not in use. Next to this is the player's money that can be used in taverns to purchase health. There can also be found the power icon that indicates to the player what power is currently selected. The HUD also features a mini map which shows the player which locations are enterable and a quest log to guide the player through the game.

4.5 Help system

4.5.1 Main Menu

In the beginning of our game the player will be placed in a tutorial dungeon. Here the player will encounter weak enemies so that they can get used to the combat and controls of the game. Originally in the tutorial dungeon we were going to implement a button prompt feature that would tell the player the controls as they would progress through the tutorial dungeons unfortunately due to time constraints we were unable to implement such a feature.

4.5.2 Quest system

Spectra will feature a quest system that guides the player through the game. This will display the players' current objective and in some cases their progress with said objective.

4.5.3 Options

Will allow the player to adjust Audio with the video being handled by a built in unity menu. Allowing the player to customise their gameplay experience.

4.5.4 Overworld

The Overworld will act as a means to getting from one dungeon to the other inbetween interspersing enemies to keep the player engaged between dungeons. Scattered through the overworld will be a handful of bonus areas for the player to explore.

4.5.5 Map

Whilst in the overworld the player will be guided by the Mini-map that can be found in the top corner of the players screen.

4.5.6 Areas

Gameplay will be split between three areas; dungeons, the over world and bonus areas.

4.5.7 Pause

Will allow players to temporarily halt gameplay and take a break along with acting as a gateway to the setting menu and the option to quit the game.

4.5.8 Game Over

Once the player has had their health depleted the game over screen will appear allowing the player to try again respawning them at the beginning of the area they died in or letting them quit and close the game.

5. Gameplay

5.1 Game Controls

The controls in Spectra were chosen as they are commonly used in other video games so the controls will seem familiar to the player and will not require much assistance in regards to telling the player what the controls are for movement.

Controls	Keyboard and mouse	Alternatives keys/Notes
Forward	Up arrow	W key
Backwards	Down arrow	S key
Left	Left Arrow	A key
Right	Right Arrow	D key
Pause	ESC key	-
Switch Orb	Left Shift	(Cannot be used if you have no orbs)
Use Orb	C key	(Effect differs depending on equipped orb)
Sword	X key	-
Bow	Z key	(Cannot be spammed)
Use vial	V key	(Cannot be used unless you have vials)
Super speed on- DEBUG	T key	(DEBUG controls Not intended for regular play)
Super speed off- DEBUG	Y key	-
Unlimited health- DEBUG	H key	(Added health does not increase health on UI)
Teleport to overworld- DEBUG	O key	-
Get all orbs- DEBUG	G key	(Does not increment Orbs on quest list)

5.2 World

The overworld is the hub of the game and will be very large. It will be on a 2D plane in which the player will have the ability to scroll up, down, left and right. It consists of many islands that each have their own elemental dungeon. This overworld is where the player moves from level to level and will include clear landmarks such as villages, lakes, mountains and more. There will be a minimap included on the UI for the overworld to guide the players to significant locations. The player can reference this if they are lost. The landscape was designed so the paths divert and lead players in a multitude of possible routes so the player is given choice of direction and freedom of exploration.

Fig. 5 - Overworld Final Design

5.3 Landscape

The overworld landscape will consist of:

- Roads (footpaths)
- Grassland and deserts
- Beaches
- Buildings and Caves
- Roadside obstacles (rocks/ trees)
- Deep Water
- Shallow Water

This landscape is fixed, but the player is able to freely explore the areas within the boundaries. The village landscapes will include a number of levels such as the floors of a house, and interior vs exterior viewpoints. The player will have a fixed starting point in the game, and after clearing that initial area will have the freedom to decide where they go next.

5.3.1 Roads

Roads will be constructed with tile maps using fixed pieces: straights, corners and ends; and will be one block wide. It is not necessary to follow these paths, as they are simply pointers to significant locations. Doing so however will lead you to a settlement or other POI. Their natural flow and random twists and turns are added to maximise the player's immersion as they emulate real world roads on maps.

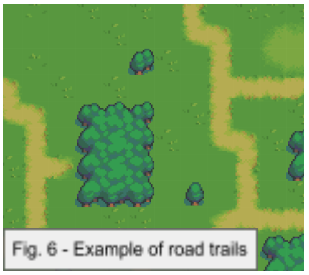


Fig. 6 - Example of road trails

5.3.2 Grassland and Desert

Grassland will cover the most land and will be used as the open spaces for general exploration and overworld battles. Deserts cover smaller areas and can be found in certain locations near the North West and South East coasts. Both of these are constructed using tilemaps and can vary in size due to the nature of the overworld. The player can walk on these areas to get from point A to B if they desire. These open plains are also home to secrets such as hidden dungeons and pathways.

Fig. 7 - Crossover of two area types

5.3.3 Buildings and Caves

Fig. 8 - Overworld Castle

These landscape components vary in size and shapes, and can be found strewn across the overworld. Approaching these areas will take you to a different scene where their overworld counterpart is elaborated on a lot more. Leaving those villages will take the player back to the overworld scene view. Cave openings will lead to a dungeon of some description; some of which are pathways to other areas and others are dead ends.

5.3.4 Roadside Obstacles

Trees and rock formations dot the landscape as a means of adding dimension to the play area. Some of these obstacles are large enough to the point that the player can enter them, leading to an extra level. Others are simply used as detail for the world. Each obstacle will have a collision system in place to stop the player from clipping out of bounds.

Fig. 9 - Enterable Forest

5.3.5 Water

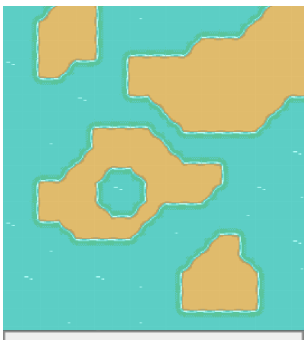


Fig. 10 - Sandbars over shallow water

Deep blue water surrounds the islands that make up the overworld and are in place to form the shape of the playable area. The player cannot walk onto this water as it acts as a barrier to contain them within.

Shallower, lighter blue waters however can be walked across. These are found at beaches and connect each island to the central island and can be identified by it's lighter colour and connection to sand.

Enemy types found on these sandbars will have the water element to best suit their environment and act as a challenge to those who wish to cross.

5.4 Dungeons

Within the overworld are a variety of dungeons. Each of these dungeons will have a similar visual design, yet they will include unique puzzles for the player to solve. There are 5 dungeons that are necessary to progress the story; the middle four can be completed in any order. They will require the player to travel in the overworld to find them however.

5.4.1 Earth Dungeon

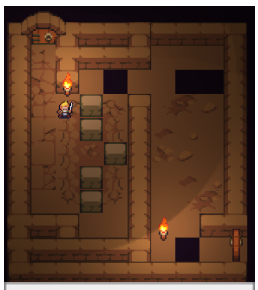


Fig. 11 - Earth Dungeon Puzzle Element

The Earth Dungeon involves the player defeating various foes in the first half, and problem solving skills in the second. Specific boulders around the level can be shifted with use of your abilities, and they will need to be moved in a way that they fall into a hole and provide the player with a path to continue. The player can get these boulder placements wrong in some instances, and will require them to leave and re-enter the room to reset their states.

These puzzles will also show up on occasion in the final dungeon.

5.4.2 Fire Dungeon

Fig. 12 - Fire Dungeon Puzzle Element

The Fire Dungeon is very combat and movement heavy. Not only will it require the player to carefully progress the level with use of all of their weapons, the Fire Dungeon will also test the players ability to dodge and time counter attacks as well. The puzzle for the level revolves around the lighting of torches in a specific order to unlock the way out. To do this will require a small amount of problem solving skills based on algebra.

5.4.3 Storm Dungeon

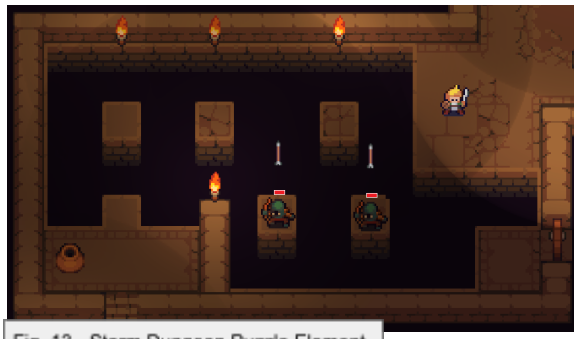


Fig. 13 - Storm Dungeon Puzzle Element

The Storm Dungeon introduces a new movement mechanic, the *Storm Jump*. This ability allows the player to quickly zip across large pitfalls and avoid enemies. This ability requires good timing however, as the player may under/over charge the jump and end up falling into the abyss, sending them back to the start. Alongside this are projectile shooting enemies whose aim is to knock you off mid-jump, so being quick yet accurate is the key to success.

5.4.4 Water Dungeon

Fig. 14 - Water Dungeon Puzzle Element

The water dungeon is unique in the sense that it takes place both outside on a small archipelago of islands and also inside as well. There are two puzzle elements to the level. The first is a looping maze in which the player must find the correct path by crossing long bridges between the islands. Make the wrong move and the player will be sent back to where they were, and will have to fight the monsters again. Beyond the bridge puzzle is a new task which requires the freezing of water in order to cross from one part to another. The catch of this however is that only 4 water blocks may be frozen at a time, so the player must think about their movement before they attempt to cross.

5.4.5 Final Dungeon

Fig. 15 - Final Dungeon Puzzle Element

The final Dungeon is an amalgamation of the previous elemental dungeons, involving puzzles from each and combining some together. The player must use their experience to handle the situations as they come whilst also fighting off the hordes of monsters. The final boss Nox will use a range of abilities to try and defeat you, requiring the player to carefully dodge and attack at the most opportune time.

5.4.6 Extra Dungeons

Fig. 16 - Example of optional area

Spectra's core levels are accompanied by many smaller scale "dungeons". These can be in the form of Deep Forest areas, small hidden caves or abandoned villages overrun by monsters. These dungeons will not include puzzles but will deploy unique enemies.

5.5 Taverns

Fig. 17 - Tavern

Taverns are passive locations that are dotted all around the Overworld and act as a place to replenish your health and save your game. They consist of a main room and a bedroom. Paying 65 gold allows the player to stay the night and heal.

Each tavern in the game has a different interior to evoke a sense of individuality.

5.6 Object Types

The types of objects that can appear include:

- Hero
- Enemies
- Treasure Chests
- Doors
- Breakable Items
- Interactable Objects

5.6.1 Player

Fig. 18 - Lux

The hero is player-controlled and will have statistics such as health and magicka. Health will be shown by the health bar on the UI screen with a red bar which depletes the closer the player is to dying. Magicka will be represented by a blue bar. Lux can traverse levels, attack and evade enemies and use abilities to solve puzzles and / or clear obstacles. These abilities will be learned as the player finds new dungeons.

5.6.2 Enemies

AI controlled enemies have the ability to attack the player using AStar pathfinding. Similarly to the player, enemies will have health and damage. There are a range of different enemies to encounter:

Fig. 19 - Goblin Archer

- Regular Enemies
- Elemental Enemies
    - Close RangeLong Range
    -
- Nox (Final Boss)

5.6.2.1 Regular Enemies

Fig. 20 - Troll

Regular Enemies have no corresponding element and have simple attack patterns, all close range. They are commonly taken down very quickly. These are the most common enemies and are encountered all the way through the game. They can range in difficulty, from a simple rat to the larger trolls and orcs.

5.6.2.2 Elemental Enemies

Fig. 21 - Example of fire monsters

Elemental Enemies are first found in dungeons and in the overworld. They are separated into different classes: Long or Close Range. Enemies like these typically have more health than regular enemies, and sometimes deal more damage.

5.6.3 Treasure Chests

Fig. 22 - Treasure Chest

The player can find treasure chests across the land and in dungeons that have loot in them. Examples of loot are health vials, magicka vials, and gold. Chests are single-use per level and their contents are randomised. To open, the player must hit them with their sword.

5.6.4 Doors

Fig. 23 - Doors

Doors are interactable objects that allow passage from one area to another. In the overworld these act as a passage between scenes, whilst in dungeons they will unblock new areas. Some doors however will not be interacted with, and will simply be used as the landscaping of inaccessible buildings in the larger cities.

5.6.5 Breakable Items

Fig. 24 - Breakable Items

Some objects such as pieces of furniture, vases and crates can be broken by the player and have the chance to drop gold for the player. These are placed in dungeons as extra objects to find if the player wishes.

5.6.6 Interactable Objects

Fig. 25 - Interactable Items

There are a few objects in dungeons that the player can interact with to cause a change in the gameplay. Examples of this would be levers that open locked doors, turn off traps, etc. These are placed in rooms full of monsters / rooms that contain a puzzle and can open doors you have already passed or doors in the same room.

5.7 Powers

Each dungeon contains a gem that will grant the player a new power once picked-up and equipped, the powers and the details of them can be seen below.

Ability	Dungeon	Effect	Range	Limitations
Dashing	Storm Dungeon	Allows the player to dash over gaps and to avoid enemy attacks	Medium	Does not grant invulnerability and the movement of the player cannot change once a dash has begun
Fireball	Fire Dungeon	Shoots a fireball that deals more damage than an arrow and activate torches that will open some doors	Long	Has a considerable mana cost so it cannot be used to replace the arrow and must be used carefully
Freezing	Water Dungeon	Freezes water tiles on contact allowing you and enemies to walk over them	Contact	Only three water tiles can be frozen at a time and freezing some tiles with allow more enemies to reach you and cannot be used on the overworld
Pushing	Earth Dungeon	Pushes boulders when the player runs into them allowing the player to create cover and to fill in holes when boulders are pushed into them	Contact/Long	Pushing boulders cannot damage enemies and only select boulders can be moved

5.8 Sound

Spectra will include a range of sounds in the game. These will be included to:

- Add ambience
- Provide a sense of danger
- Give item use / pickup feedback
- Grant completion feedback
- Add menu and selection feedback

5.7.1 Sound Effects

Spectra will include feedback effects to let the user know if they are successful, if they have picked up an item, hit the target / been hit themself.

5.7.1.1 Player

- Attacks (Sword, magic, bow and arrow, ability)
- Damage taken
- Using items

5.7.1.2 Enemies

- Attacks
- Damage taken

5.7.1.3 Environment

- Torches flickering
- Waves
- Items breaking
- Objects being used

5.7.2 Soundtrack

The game will have a complete soundtrack comprising all new music produced by the team specifically for the game. Each level in the game will have a corresponding track. The Soundtrack will be created on the software Mixcraft and will incorporate loops and MIDIs.

6. Technical Specifications

6.1 Game Mechanics

The game will begin with the Main Menu displaying the Game's Logo in the centre. Music will accompany.

The most significant mechanics of *Spectra* are the mechanics regarding basic combat, specifically damage and knockback dealt by enemy and player attacks alike.

The player can use their Earth Ability to **push boulders** at a constant velocity, until they hit a wall or fall into a hole.

The player can shoot a **fireball** to deal damage to enemies or light torches that are involved in puzzles.

The player can **zip a distance** to cross gaps and avoid enemies.

The player can **freeze water** to provide a means of crossing dangerous waters.

6.2 Code Objects

The game objects used in our project can be categorized in three separate ways. The first category are prefabs such as the player, various enemies, empty objects for loading zones and interactable objects like chests/levers. The second category are objects such as the tile map that renders the level, level collisions and lighting. The final category are objects such as the Image UI object that represents the players health, the Image UI object that represents the players magic reservoirs, the sprite that represents the players current amount of gold and finally the amount of "magic/health vials" the player has in their inventory.

6.3 Game Object Data

Fig. 26 - Example of player constituents

The most significant part of the coding concerns the combat between our player object and the various enemy prefabs. Combat in *Spectra* works by creating a BoxCollider2D component which has no size named "LocalCollider", this LocalCollider is manipulated on each frame that has a hitbox to be the size and offset of the specified hitbox. Player and enemy prefabs have an inactive child object that stores the sprites of each attack that have a hitbox, each of those sprites have a BoxCollider2D that is shaped as we wish and then mapped to LocalCollider, they also have a script attached that stores the damage and knockback variables of each hitbox. Each attack animation for the player/enemy prefabs have events on every frame that sets the LocalCollider to the information of the hitboxes stored in the empty object, there's also a special clear event that resets the hitbox to size zero.

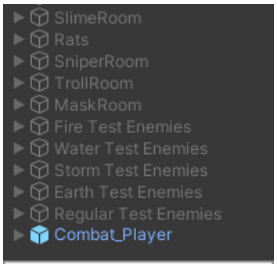


Fig. 27 - Tutorial Dungeon Enemy Rooms

Perhaps the most challenging part of the coding has been manipulating the astar pathfinding components to ignore its singleton pattern and allow for multiple objects that use astar pathfinding. By instantiating new versions of the enemies of a room each time you enter that room, and by derendering/removing the collision of an enemy that uses astar pathfinding instead of deleting them once their health is depleted, it is possible to have multiple enemy objects use pathfinding at any one time.

Scene transitions that allow for putting the player at specific coordinates in the loaded scene work by storing x/y coordinates and scale variables in the object that handle the scene loading such that once the next scene is loaded it moves the player to the next scene with those preset positions and at that scale. We do this instead of simply instantiating a new player object at the specified position so the health and magic values aren't reset. It's possible there is a more efficient way to achieve this but we've yet to experience any issues with this method.

```
}

// Update is called once per frame
void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.gameObject.tag == "Player")
    {
        collision.gameObject.GetComponent<Combat_PlayerController>().overworldX = x;
        collision.gameObject.GetComponent<Combat_PlayerController>().overworldY = y;
        collision.gameObject.transform.localScale = new Vector2(scaleX, scaleY);

Debug.Log(collision.gameObject.GetComponent<Combat_PlayerController>().overworldX + " "
+ collision.gameObject.GetComponent<Combat_PlayerController>().overworldY);
        SceneManager.LoadScene(nextLevel);
    }
}
```

There are interactable objects such as chests or pots that instantiate a random obtainable item after colliding with a hitbox. Items such as coins, instant health potions and retainable health potions (in that they can be used at any time after obtaining them) are such examples. This is achieved with a simple random number generator and IF statements that decide which object to instantiate based on the outcome.

6.4 A.I

Most of the enemies use the A* pathfinding algorithm to pathfind and the script that controls this allows certain tagged objects to be considered ‘obstacles’ that the algorithm will work around. Additionally a set area can be set for each object with the script, both of these allow each enemy to easily pathfind within their respective rooms.

7. Testing and Evaluation

7.1 Alpha Testing

Every significant step made in the game was followed up with testing. When a level was finished we individually played it through and took notes on subjects such as it's flow and difficulty. Any significant issues were sorted out immediately followed by a retest. This iterative loop was our means of polishing the game at every possible step, and assuring that it is as good as it could possibly be.

7.2 Beta Testing

Once near completion, the game was given to friends and family of the developers in order to gauge an audience reception. We gained valuable data such as the difficulty and enjoyability, alongside any bugs that we may have missed during the Alpha Testing stage.

We compiled all of the data and highlighted the most common answers in order to prioritise fixes and further developments.

7.3 Known Bugs

Although we took our time with bug testing, we did not manage to fix every bug that we encountered. Bugs that broke the game or prevented progress were prioritized so the list of known bugs below are smaller and although not ideal, still allow the game to be played.

- Some UI changes on different screen size
- Dashing near holes makes you slow down
- Some enemy pathfinding doesn't work
- Money and vials don't transition between scenes
- Some enemies only deal damage once
- Vials don't work in some scenes
- Some of the controls do not appear on the controls screen

7.4 Evaluation

The game has a lot of content and we are happy with the progress made on tha game. We set our standards high and put in the effort. Some parts of the game are unfinished however, a few features are less polished / missing from our original vision when planning the game. Things like the bosses were not finished due to more immediately vital aspects needing more manpower than we planned for in order to get completed on time and could have definitely spent more time fixing bugs and less time finding them. Despite a few small organisational changes that should have gone differently, things followed the initial plan resulting in a product that we are all happy with.

8. Appendix

This section includes supplementary material that provides extra information to figures previously shown within the document.

8.1 Additional Concept Art

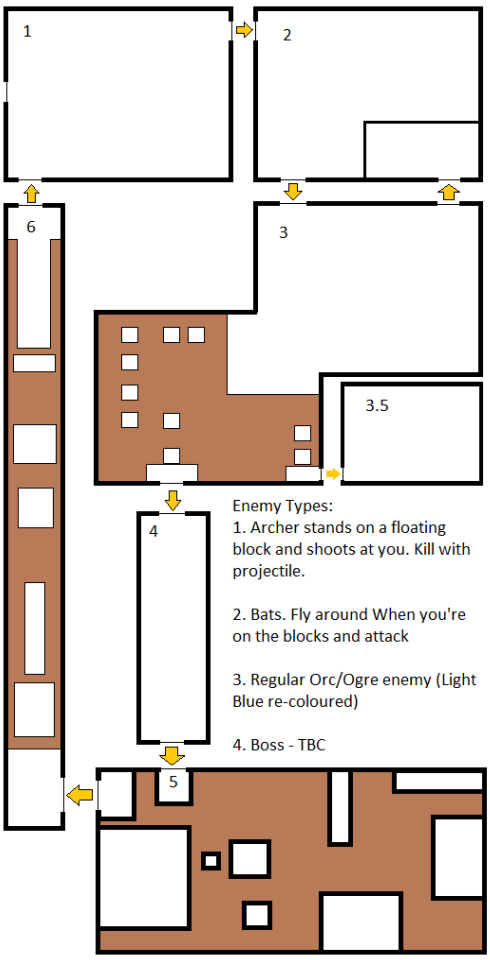


Fig. 31 - Storm Dungeon Design

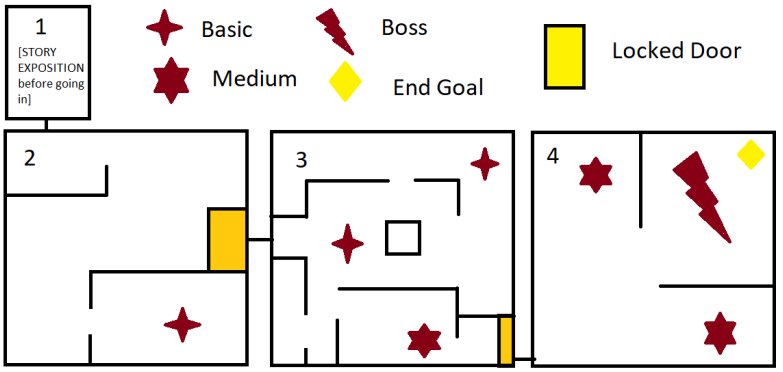


Fig. 30 - Original Tutorial Dungeon Design

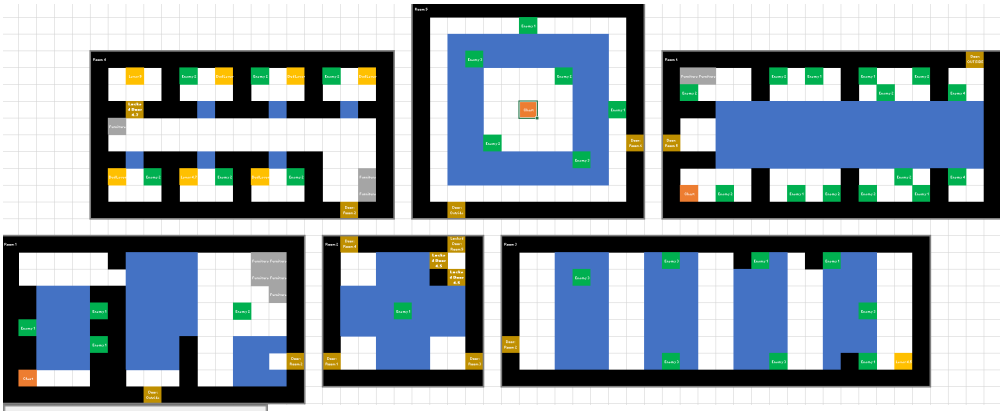


Fig. 32 - Water Dungeon Design

8.2 Completed Levels

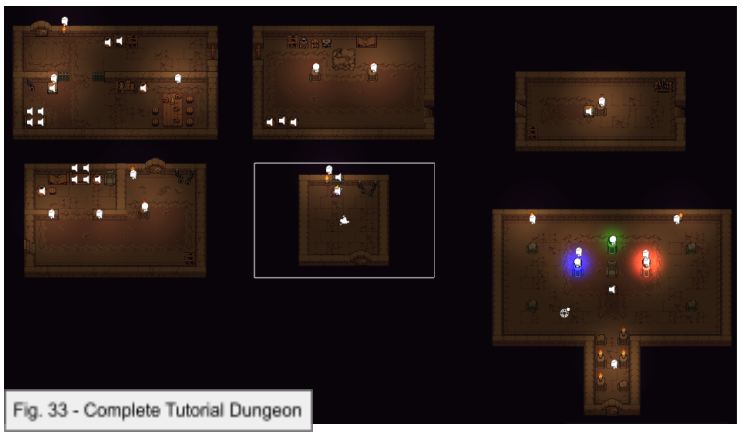


Fig. 33 - Complete Tutorial Dungeon

Fig. 34 - Complete Earth Dungeon

Fig. 34 - Complete Fire Dungeon

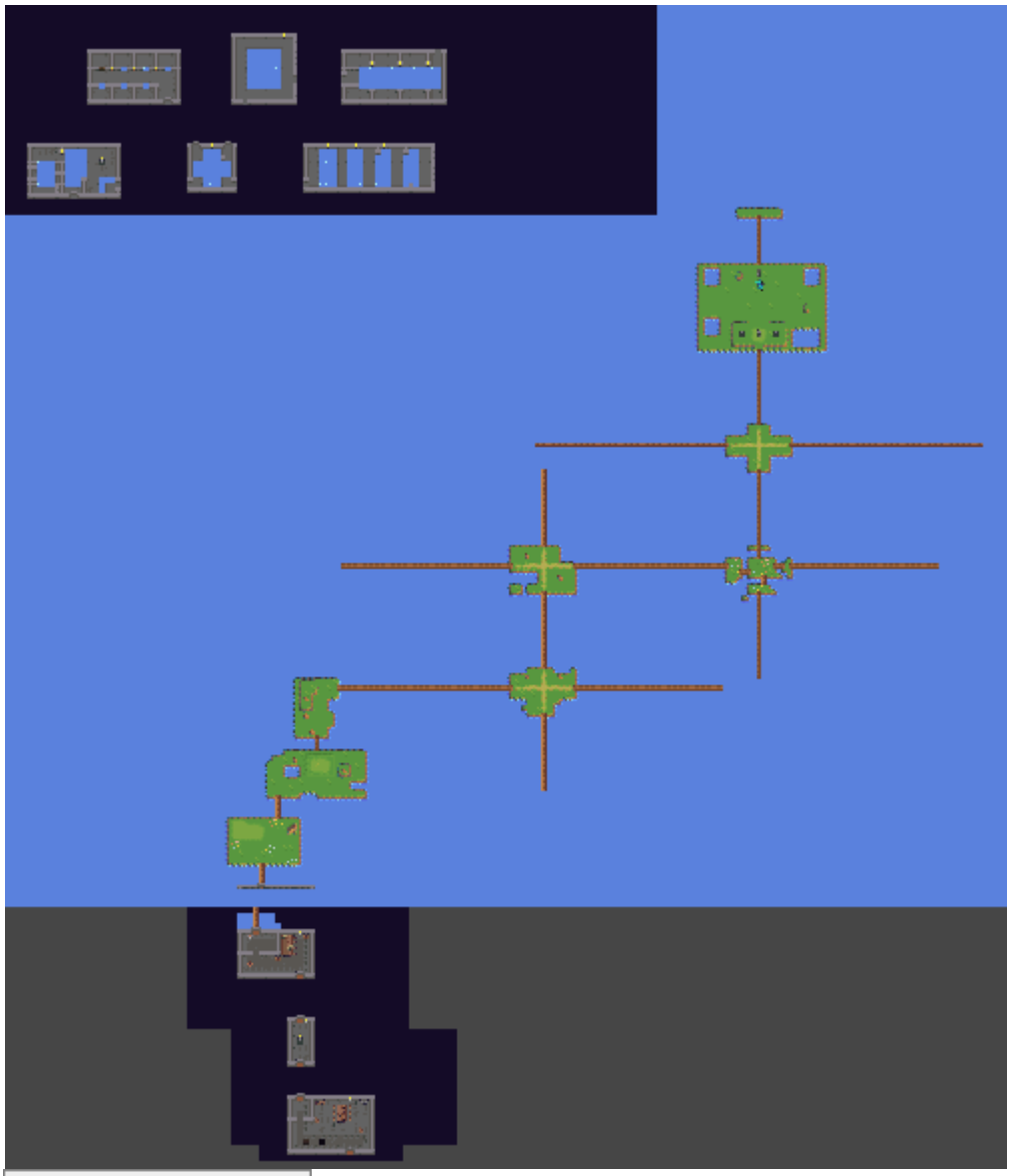


Fig. 35 - Complete Water Dungeon

Fig. 36 - Complete Storm Dungeon

Fig. 37 - [Optional] Ambush Forest Level

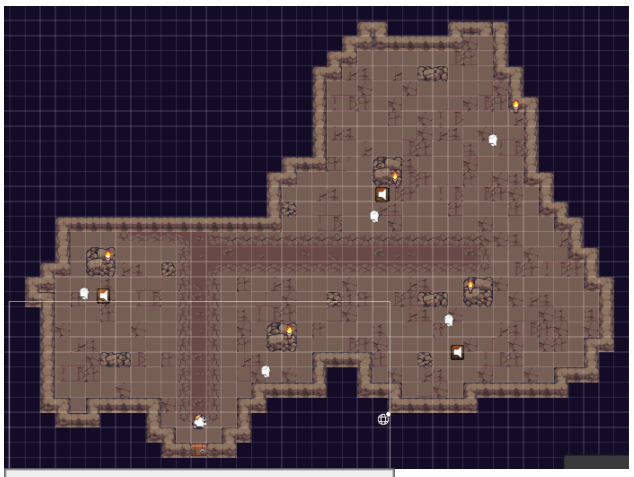


Fig. 38 - [Optional] Cave Dungeon Level

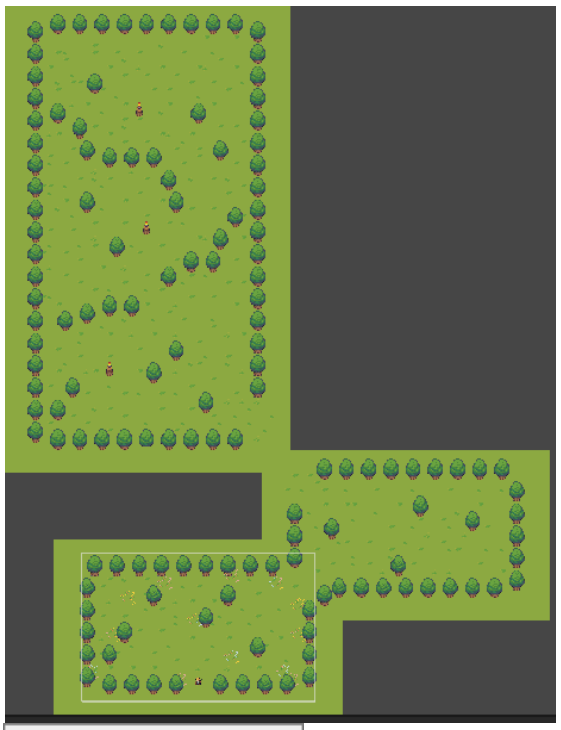


Fig. 39 - [Optional] Forest Level

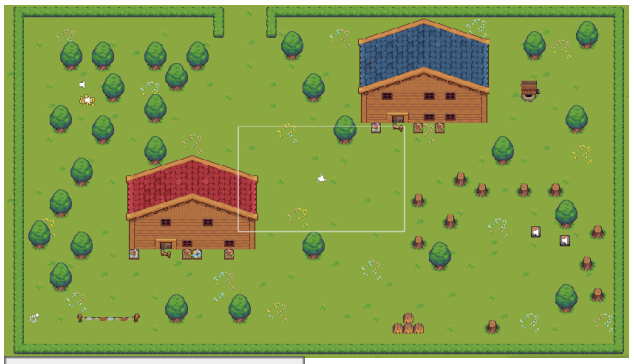


Fig. 40 - [Optional] Village Level

Fig. 41 - [Optional] Village Level 2

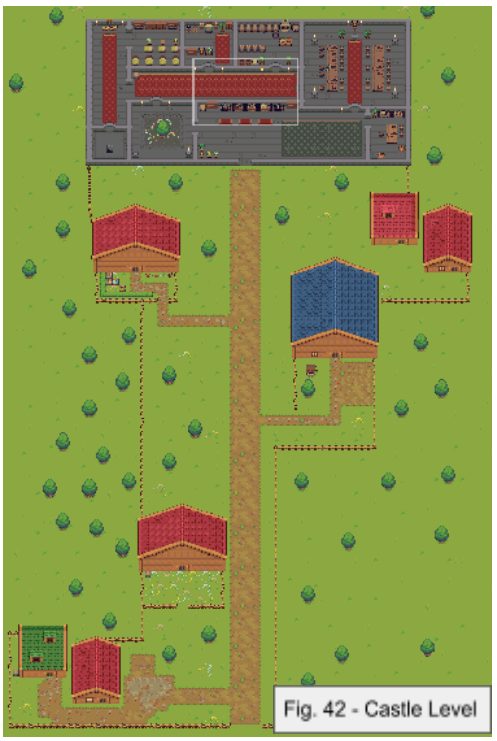


Fig. 42 - Castle Level

Fig. 43 - [Optional] Village Level 3

8.2 Beta Testing Sheet

SPECTRA USER REVIEW

What did you enjoy about the game?

The Game was well crafted and had a professional look to it and I also loved the art style of the game. On top of that I love how exciting main areas will lead to this overworld for the player to explore.

What didn't you enjoy about the game?

The combat was a bit clunky and the enemies appeared to have more mobility than the player themselves making it hard to attack them. I also found myself being locked into dungeons that I could not exit as I did not have the powers to progress through the dungeon.

Overall enjoyment of the game and final thoughts?

I enjoyed exploring the game's world and the game was overall crafted well and had a great aesthetic. The game was buggy in areas that did hinder the gameplay experience but I think if these can be fixed then the game can be a very fun experience.

Bug	Description	Severity High, Medium, Low
Bug 1	Torches in tutorial dungeon move at different speeds	Low
Bug 2	Health pickups do not disappear on pick up	Medium
Bug 3	Arrow limit does not deplete	High
Bug 4	HUD oddly placed	Medium
Bug5		

SPECTRA USER REVIEW

What did you enjoy about the game?

What didn't you enjoy about the game?

Overall enjoyment of the game and final thoughts?

Bug	Description	Severity High, Medium, Low
Bug 1		
Bug 2		
Bug 3		
Bug 4		
Bug5		

SPECTRA USER REVIEW

What did you enjoy about the game?

I loved the dungeons and the enemies that are within them. The transitions between them in the overworld is cool too. I also loved the music of the game which managed to capture the spirit of the levels they were in.

What didn't you enjoy about the game?

I found a few bugs that could lead to the player being able to break out of the playable area. Some bugs also stopped progression in some areas such as a forest area that crashes the game.

Overall enjoyment of the game and final thoughts?

The game was fun overall but a lot of the kinks need to be ironed out before it can be really fun to play. Like the ones that hinder progression and crash the game.

Bug	Description	Severity High, Medium, Low
Bug 1	A few boundary breaks in the castle	High
Bug 2	Health pickups do not disappear on pick up	High
Bug 3	You can stand on trees in the castle courtyard	Low
Bug 4	Overworld map placed in a weird spot	Medium
Bug5	Weird clipping on houses in the castle	Medium

Fig. 44 - Audience Feedback Sheet examples

SPECTRA USER REVIEW

What did you enjoy about the game?

I enjoyed the puzzles found in the dungeons and the powers you can pick up in the dungeons and how they can be used later on to complete other challenges. I also loved the bonus areas that allowed you to get extra pickups or money.

What didn't you enjoy about the game?

There were some bugs that hindered the gameplay of spectra such as some areas leading to me getting stuck in them and having to restart the game, which was made more frustrating by there being no save system in the game.

Overall enjoyment of the game and final thoughts?

The game was fun when it worked but the glitches and bugs that I experienced hindered the game from playing fluidly and I was not able to complete the game because of these glitches. The absence of a save system really made the repeating of gameplay frustrating

Bug	Description	Severity High, Medium, Low
Bug 1	Certain areas will not allow you to go back to the overworld	High
Bug 2	Health pickups do not disappear when picked up	Medium
Bug 3	Arrow limit does not go down	Medium
Bug 4	HUD oddly placed	Low
Bug5	Quit Game option does not work on pause menu	Medium

9. Acknowledgements and References

9.1 Coding Research Sources

Granberg, A., 2019. *A* Pathfinding Project*. [online] Arongranberg.com. Available at: <https://arongranberg.com/astar/> [Accessed 6 March 2021].

Unity.com. 2015. *2D Renderer in Universal Render Pipeline*. [online] Available at: <https://forum.unity.com/threads/2d-renderer-in-universal-render-pipeline-in-2019-3.829851/> [Accessed 14 March 2021].

9.2 Visual Sources

2017. *Dungeon Tileset*. [image] Available at: <https://pita.itch.io/rpg-dungeon-tileset> [Accessed 9 February 2021].

2017. *Monster Sprite Pack*. [image] Available at: <https://pita.itch.io/rpg-monster-pack> [Accessed 9 February 2021].

2018. *Overworld Tileset*. [image] Available at: <https://pita.itch.io/rpg-overworld-tileset> [Accessed 9 February 2021].

2018. *Village Tileset*. [image] Available at: <https://pita.itch.io/rpg-village-tileset> [Accessed 9 February 2021].

9.3 Audio Sources

2017. *FX_Orc.wav*. Available at: <https://freesound.org/people/ForTheHorde68/sounds/407356/> [Accessed 4 March 2021].

2020. *Tomato squish;wet*. Available at: <https://freesound.org/people/HonorHunter/sounds/271666/> [Accessed 8 March 2021].

2020. *Rat.ogg*. Available at: <https://freesound.org/people/egomassive/sounds/536753/> [Accessed 8 March 2021].

2013. *Knife cling*. Available at: <https://freesound.org/people/sgarrison2127/sounds/178079/> [Accessed 8 March 2021].

2014. *Ambient Spacecraft Hum*. Available at: <https://freesound.org/people/AlaskaRobotics/sounds/221570/> [Accessed 8 March 2021].

2018. *Door, Front, Opening*. Available at: <https://freesound.org/people/InspectorJ/sounds/431117/> [Accessed 8 March 2021].

2014. *Glug.wav*. Available at: <https://freesound.org/people/LloydEvans09/sounds/221488/> [Accessed 8 March 2021].

2021. *knife thrust into wall*. Available at: <https://freesound.org/people/NeoSpica/sounds/504618/> [Accessed 8 March 2021].

2021. *Clay Jar Break*. Available at: <https://freesound.org/people/Clearwavsound/sounds/540337/> [Accessed 8 March 2021].

2015. *Regular Arrow Shot*. Available at: <https://freesound.org/people/brendan89/sounds/321552/> [Accessed 12 March 2021].

2019. Another Fire. Available at: <https://freesound.org/people/AngriestBrad/sounds/495894/> [Accessed 12 March 2021].

2020. *Splat/Squish 3*. Available at: <https://freesound.org/people/mrickey13/sounds/515620/> [Accessed 10 April 2021].

2020. Available at: <https://freesound.org/people/mrickey13/sounds/515625/> [Accessed 12 March 2021].

2012. Available at: <https://freesound.org/people/DWOBoyle/sounds/140382/> [Accessed 12 March 2021].

Freesound. 2021. *Electric zap.wav by michael_grinnell*. [online] Available at: <https://freesound.org/people/michael_grinnell/sounds/512471/> [Accessed 12 April 2021].