CATACOMBS

GAME DESIGN DOCUMENT

AUSTRALIAN STEM VIDEO GAME CHALLENGE

SARAH WARBURTON

LOBSTER AND FISH

Hey there! I'm Sarah Warburton, and for some reason or another need a team name, so you can refer to me as Lobster and Fish!
I picked up Godot in February and thought to myself, hey, why not make a 3D game? I've messed with (and wholly failed at) Unity before, and I have this inexplicable pull to create a massive 3D game which I never properly create.
So I compromised, and decided to scale down to a smaller game with a bit less sandbox-y intentions, and made my first Godot game.
It was terrible.
But anyway, here's my second game I've ever produced.
I proudly (or maybe not) present to you:

Catacombs

Planning

Organisation

As a solo developer, all responsibility is delegated to me. In order to deal with this, I will create myself a clear timeline and plan, as well as a location to organize my resources. As I am not a naturally organised person, this will help keep me on track for a clean finished product.
All resources will be stored in a single folder, which is then delineated further into a folder containing all game resources. The structure is as follows:

Key:

- Object = specific file

Object = folder

```
Project
    - GDD
    Screen recording files
        - Files and final recording [will be collated into a single file upon
          submission]
    Game Files
        Executables
            Mac [ If applicable]
            Windows
        Project Files
            Windows Godot files [for project]
                - Project
```

Submission guidelines

Below is my analysis and collation of key submission guidelines and competition rules. These have been evaluated at multiple points of progression during the coding and construction of the game. I've also included annotations I made during analysis of the guidelines that I used to guide me into completing the finished product.

KEY	🟥 Not done	🟨 Attempted	🟩 Complete

Guidelines	Check 1 - 25%	Check 2 - 75%	Check 3 - 95%	Check 4 - Submission	Questions to remember/address
Submission includes a GDD, Screen recording and game	🟨	🟨	🟨	🟩	What does the game format need to be? .exe? Can I put random stuff in the screen recording?
Must run on a windows operating system	🟩	🟩	🟩	🟩	
Must be built on free-to-use platforms (or free for eduction)	🟩	🟩	🟩	🟩	Godot
Judges must be able to see code	-	-	-	🟩	Do I export the file in some special way? Be careful here
Submitted games must use a keyboard or mouse-based control system				🟩	
And be playable for at least 5 mins	🟨	🟩	🟩	🟩	
Games must be playable in single player	🟩	🟩	🟩	🟩	
Main assets must be student's work	🟩	🟩	🟩	🟩	Totally my assets. Nothing (Except for font) taken from external sources
Screen recording - not a trailer - includes key high-level features,	🟥	🟥	🟨	🟩	

etc. 1-5 minutes long	🟥	🟥	🟨	🟩	
Ai usage must be clearly labelled	🟨	🟨	🟨	🟩	
Official Rules	-	-	-	-	(Not covered by guidelines)
Team does not exceed 4 members	🟩	🟩	🟩	🟩	
18+ Adult Mentor, Australian citizen, enrolled in a school (3-12) w/ Parent permission	🟩	🟩	🟩	🟩	
School provides judge	?	?	?	🟩	Mentor confirmed
Must be G rated	🟨	🟨	🟩	🟩	Check guidelines for G submissions
GDD completed	🟨	🟨	🟨	🟩	Doesn't have to follow outline
All intellectual property correctly utilised	🟩	🟩	🟩	🟩	

Furthermore, the judging rubric was analysed before completion (8/07/2025) to determine the quality of the submission. While this table was referenced earlier, the analysis was only done when the project was completed to my intended standard. This completed analysis can be seen below (appendix 2).

Workflow

When creating this game, various potential themes and functions were explored before deciding on the final concept.
These were ideated using the technique of divergent thinking – ideating a large number of potentially poor ideas, but with such a large number of concepts, one is likely to be good. This approach was taken to increase originality, which is difficult in a single-person team. These are then narrowed down to the final concept. Examples of preliminary concepts is demonstrated below. This brainstorming was separated into two stages – Broad, then specialized.

Broad Planning Excerpt (converted to table)

Concept/theme	Images/explanation	Rating
Spooky, dark game	- Evoking gothic style, moody structures	🟩 Easy to make without access to textures due to my blender being almost wholly nonfunctional
Desert themed game	- Empty, sort of haunted	🟧 *as above However, that was the first game I made and thus kind of boring to do again
Player is a drop of water	- And you can't actually control the player but you get to control the world around it to try and direct it to a location	🟧 Difficult to use water physics in Godot
Reincarnation cycle to complete an objective	- Like higher class johnny upgrade, but it's also about finding faster routes and ways to travel	🟥 Boring to repeat, and hard to make in my preferred style. Requires a lot of assets
Playing as a seed blowing in the wind and you have to bounce around to fertile land	- Some kind of cool wind mechanic	🟩 Should be easy enough, relatively interesting, but how to code bounces, etc?
A fantasy cute land where things get more uncanny the longer you play	- Stuff moves while you're not looking, gets reskinned, etc	🟧 Requires a lot of asset creation, and my blender cannot handle textures
You play as a bird	- Absolutely FIRE soaring through clouds and around mountains	🟧 Fun, but what kind of objective could we have here?
You play as a cat	- Not too unfamiliar, similar to other 3d movement mechanics	🟩 Easy to code, movement of cat can be studied, players are familiar with movement style
Some kind of dungeon	- Moody	🟩 Adds interest to game

Solving your own murder	- Just cool. You're a ghost.	While incredibly cool, it would require a complex internal system, which is a hard ask for my second game ever.

After this wide brainstorming, a few ideas were combined to create three major concept games.

Game concept	Interesting features	Pros	Cons
You start in a dry desert landscape as a seed from a lone flower, bouncing through valleys and canyons to mountains, trying to keep momentum. Kind of like catapult. You might get stuck or land in water and have to restart.	- Wind mechanics - Bounce mechanics - Beautiful landscaping	- Very original - Aesthetically pleasing - Displays coding skill	- I have very little capacity to model trees due to my blender restrictions - I have done work in deserts before, which makes it less challenging
Every full moon, you transform from a bird to your human form before you died. In the days between, you try to gather evidence of who killed you. The game	- Multiple player models - Complex npc and object interactions - Exciting and new	- Original - Large scale - Interactive - Multiple ways to progress	- Potentially not G rated - Very complex to make it all fit together - Large scale
You play as a cat, who Is totally lost and far from home, in a dark and moody environment. You explore and make new friends, trying to get back home.	- Npc interactions - Interesting shaders - Cutscene details	- Requires less asset texturing - Interesting character development - Achievable scale - Aesthetically pleasing	- Less innovative - less opportunity for interesting mechanics

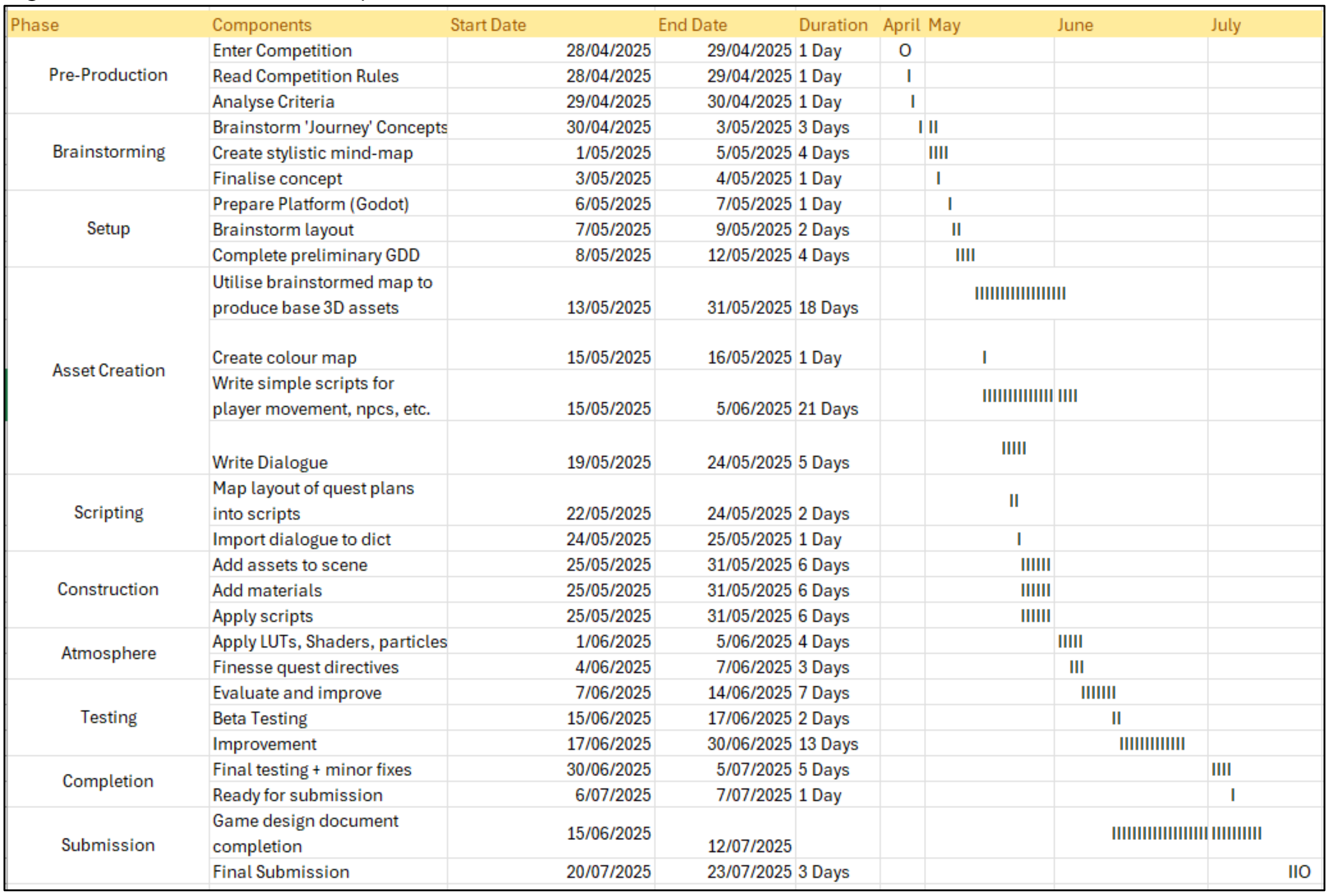
After evaluating between these options, given this was my first time in the competition, I have selected the final option, as the cons have less significant consequences to output if effected, and furthermore, the interesting features, while less notable than the other two option, still provide ample room for technical ability.

To develop my game, I will begin by deciding on a vague concept of the main elements I desire to include in my game. This will allow me to create a game that centers around a clear vision of the theme. After this, I will decide the medium that best will represent my game content (2D, 3D). After that, I will produce a rough sketch of the layout (2D) or scene (3D). Next, I will produce the major assets required for the game, and a storyboard. Subsequently, I will draft the base code of the game for the major components. Finally, I will fine-tune the game and add finishing touches and aesthetic requirements to create a polished game.

To more efficiently produce the game, I can work on the rough layout sketch and the storyboard at the same time, as they are linked. Furthermore, I can code and produce assets at the same time, as upon inserting an asset into the game, I can include rudimentary code. Finally, I will finalise the visual aesthetic of the code and implement fine details to provide a smooth gaming experience. A rough timeline of features is included below.

Timeline

Figure 1: GANTT Chart of key milestones



Phase	Components	Start Date	End Date	Duration	April	May	June	July
Pre-Production	Enter Competition	28/04/2025	29/04/2025	1 Day	O			
	Read Competition Rules	28/04/2025	29/04/2025	1 Day	I			
	Analyse Criteria	29/04/2025	30/04/2025	1 Day	I			
Brainstorming	Brainstorm 'Journey' Concepts	30/04/2025	3/05/2025	3 Days	I	II		
	Create stylistic mind-map	1/05/2025	5/05/2025	4 Days		IIII		
	Finalise concept	3/05/2025	4/05/2025	1 Day		I		
Setup	Prepare Platform (Godot)	6/05/2025	7/05/2025	1 Day		I		
	Brainstorm layout	7/05/2025	9/05/2025	2 Days		II		
	Complete preliminary GDD	8/05/2025	12/05/2025	4 Days		IIII		
Asset Creation	Utilise brainstormed map to produce base 3D assets	13/05/2025	31/05/2025	18 Days		IIIIIIIIIIIIIIIIII		
	Create colour map	15/05/2025	16/05/2025	1 Day		I		
	Write simple scripts for player movement, npcs, etc.	15/05/2025	5/06/2025	21 Days		IIIIIIIIIIIIII	IIII	
	Write Dialogue	19/05/2025	24/05/2025	5 Days		IIIII		
Scripting	Map layout of quest plans into scripts	22/05/2025	24/05/2025	2 Days		II		
	Import dialogue to dict	24/05/2025	25/05/2025	1 Day		I		
Construction	Add assets to scene	25/05/2025	31/05/2025	6 Days		IIIIII		
	Add materials	25/05/2025	31/05/2025	6 Days		IIIIII		
	Apply scripts	25/05/2025	31/05/2025	6 Days		IIIIII		
Atmosphere	Apply LUTs, Shaders, particles	1/06/2025	5/06/2025	4 Days			IIIII	
	Finesse quest directives	4/06/2025	7/06/2025	3 Days			III	
Testing	Evaluate and improve	7/06/2025	14/06/2025	7 Days			IIIIIII	
	Beta Testing	15/06/2025	17/06/2025	2 Days			II	
	Improvement	17/06/2025	30/06/2025	13 Days			IIIIIIIIIIIII	
Completion	Final testing + minor fixes	30/06/2025	5/07/2025	5 Days				IIII
	Ready for submission	6/07/2025	7/07/2025	1 Day				I
Submission	Game design document completion	15/06/2025	12/07/2025				IIIIIIIIIIIIIIIIIII	IIIIIIIIII
	Final Submission	20/07/2025	23/07/2025	3 Days				IIO

**Note, all days presume from 8:00 AM that day*

My game will be intermittently tested during the process of production. However, a final draft will be tested by the middle of June – (15th), tested until the 17th, and a final copy by June 30th for minor bug fixes. (Figure 1).

This timeline was fully utilized during production, and thus no modifications were made between expected and actual production times. This effect was created by allowing ample time for each segment of the game.

Inspiration and points of originality

Works referred to below (Cited in references):

She-Ra and the Princesses of Power
Nausicaä of the Valley of the Wind
Curious Archive – The Most Powerful Type of Worldbuilding
Rain World
Sky: Children of the Light
Stray
Kiki's Delivery Service
Minecraft (C418)
Hide – CS01 Ver.

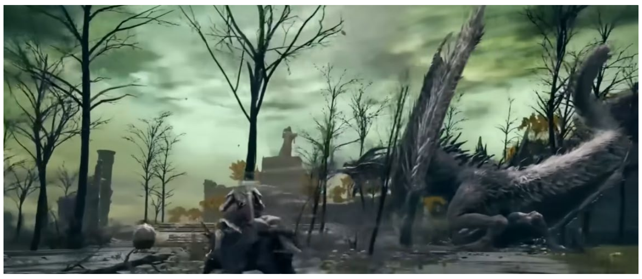
Physical Design

I was inspired by Studio Ghibli's ambience in their more sombre films. Jiji (Kiki's delivery service) inspires the cat that represents the player (Wikipedia Contributors, 2020). Furthermore, during the development of the initial game idea, I watched a multitude of curious archive videos (Curious Archive, 2024), which helped set the tone for the game, and provided a multitude of games to reference from. The modern She-Ra remake also inspired some of the landscape features, from the headquarters of the Horde to the First Ones ruins (Fandom Contributors, 2025).

I intend to use these landscapes and themes to build a world with a similar sense of uncanny emptiness, yet with an undertone of life.

Attached below are some images I used particularly when developing as inspiration. A full inspiration board can be found in appendix 1.

Beast Island, She-Ra (Fandom Contributors, 2025)


Elden ring (Curious archive, 2024)

The Jungle, Studio Ghibli (Jorstad, 2018)

Soundscape

Music for the game was inspired by two primary artists, the first being C418, the composer behind Minecraft's famous soundtrack (Rosenfeld, 2025), and Dorian Concept (primarily 'Hide – CS01 Version) (Play Instinct, 2023).

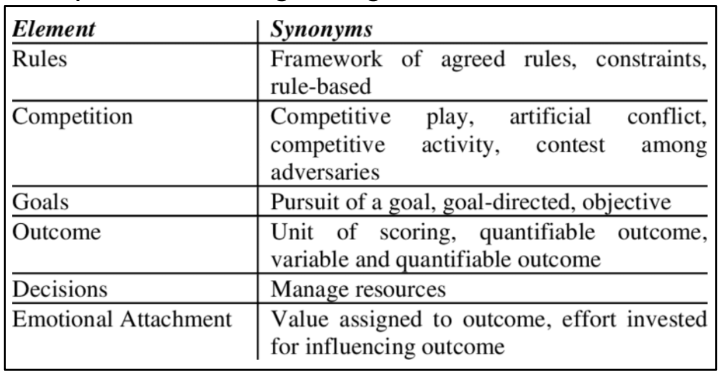
Gameplay

Rain world (2017) follows a similar storyline of an attempt to reconnect with family ending in something completely different entirely, as well as a discovery-themed gameplay style (Wikipedia Contributors, 2023). However, rain world is a 2D procedurally-generated animated ecosystem simulation game, making it vastly different. Additionally, sky: children of the light similarly follows this theme of discovery (Fandom Contributors, 2025), and mirrors my game's puzzle-adventure gameplay and notions of friendship. In contrast to my game, sky features an uplifting setting and humanoid main characters.
Finally, stray also utilizes a cat as the primary character of the game (Wikipedia Contributors, 2022), however, as with the previous game, the setting is completely different and with a clearly differentiated theme.
While all three games are critically acclaimed, and thus have a large cult following, my game is sufficiently different and interesting enough to be playable. Additionally, the simplicity of the game and moody playstyle may attract alternative demographics to the game. Furthermore, the value in making a game does not always lie in the fanbase, but sometimes in the skills acquired and personal value of the game.

Additionally, the soundtrack is totally original, while inspired by popular artists, providing another point of originality. An effort was made to ensure that the final product did not align too closely to the reference images for visual style, to create an interesting and new graphic experience.

In order to determine, objectively, whether the game is worth producing, the following criterion have been analysed. These components are widely agreed upon, and were first introduced to me in the tenth-grade curriculum. The particular table used here was sourced from a journal on the classification of pervasive games. (Hinske et al., 2007)

Components of a 'good' game:



Element	*Synonyms*
Rules	Framework of agreed rules, constraints, rule-based
Competition	Competitive play, artificial conflict, competitive activity, contest among adversaries
Goals	Pursuit of a goal, goal-directed, objective
Outcome	Unit of scoring, quantifiable outcome, variable and quantifiable outcome
Decisions	Manage resources
Emotional Attachment	Value assigned to outcome, effort invested for influencing outcome

Evaluation

Criterion	Evaluation
Rules	The player must follow certain constraints of the game. Movement follows gravity, the player is a certain form/being, cannot damage other objects, etc.
Competition	While not seemingly a competitive game, the game's customisable style can be used for speedrunning (My record is 7 minutes).
Goals	The game begins with a clear goal, to 'get home.' This is then separated into smaller quests, noted on the noticeboard.
Outcome	As above, there is a clear quantifiable outcome, which is to reach the title screen, thus 'completing' the game.
Decisions	The player is given multiple levels of agency, from which order to complete quests, to dialogue options, and the obvious, whether to return home.
Emotional Attachment	NPCs attempt to connect with the player, changing attitudes as they talk. Furthermore, the player becomes familiar with the environment and inhabitants, helping them and forming connections as they traverse the landscape.

As the game meets all these criteria, it can be definitively said that the game is worth producing, and as above, significantly different.

Technical requirements

Development Environment

The finished product will be created using the Godot engine (4.3), and thus run on both windows and mac operating systems (Godot, 2025). However, without a simulated mac device to export from and test on, the playability of this version is not guaranteed as it is for the windows executable. Godot provides an organized workspace that produces games playable in .exe and web formats, thus making them accessible to all players. I have arranged my submission into folders containing executables for different environments, as well as the source files and code. (As demonstrated above)

The only disadvantage of this workspace is lack of advanced systems in terms of post-processing or simulations in the way that Unity or similar may provide. However, as my intention is to create a simple but enjoyable game, these specs are not necessary. To play the game, the player requires a working mouse or touchpad input, and a keyboard. Headphones are advised to improve ambience. All of these inputs are readily available with standard computing systems and thus do not provide an obstacle to playability.
As the game was intended to not require complex calculations or visual draws, the game can be played on almost any system. A table below outlines system specifications:

	System age	Device type	RAM	Processor	System type / operating system	Other notes	Performance
Minimum testing capability	2021	Hp Envy	8.0 GB (6.96 GB usable)	Ryzen 5 2500U with Radeon Vega Mobilr Gfx 2.00Hz	Windows 10 home 64-bit operating system	Critically degraded battery Cracked screen	Working. No detectable errors.
Game production device capability	2025	Hp Elitebook	16.0 GB (15.5 usable)	Intel® core™ Ultra 5 125U 1.30 GHz	Windows 64-bit operating system (windows 10 education)	Commonly throws blue screen errors	Lagless, errorless play
Recommendation	Upwards of 2021	Hp	16.0GB RAM	Intel Core or AMD Ryzen 5	Windows 10	Any system is likely to be able to run this game.	-

Thus, testing has proved that even devices in terrible states of repair and with old specs can still run my game.

Optimisations:

- Scenes are not cached when exited, to keep space free. Particles are minimised. Instead of caching whole scenes, data is saved upon re-entry.
- All meshes are low-poly, and collision meshes are simplified for complex meshes (Etc. Fence)
- Some animations utilise fragment or vertex shaders to avoid script repeats (particularly that of the glowing stone.)
- Not all lights cast shadow.
- Anti-aliasing is disabled, as the visual improvement is not proportional to the performance cost.
- Particle systems were implemented for clouds and wind, to decrease unnecessary

scripting.

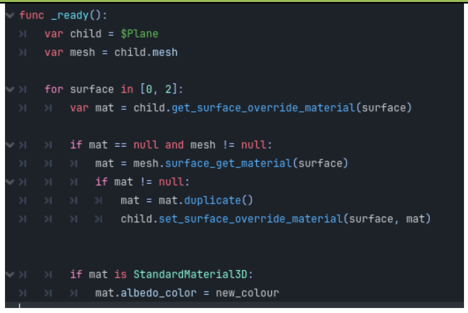
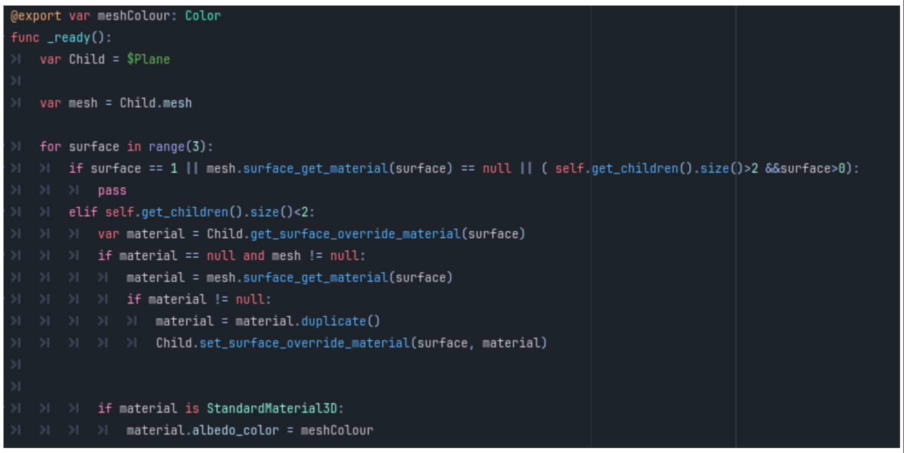
Resourcing/Capability

In order to fulfil the technical requirements of the game, I will need to utilize certain programs to create assets and plan designs. Most of the 2D planning will occur in either Canva or IbisPaintX, as both are useful artistic or logical planning applications. 3D assets will be produced in Blender, the simplest and most adept program for the task. While my device is not equipped to run higher-end blender functions (materials, textures, simulations etc), the base function will be enough to produce a working game. This issue cannot be rectified with another software, as blender is the most suited for the task as a free platform, and furthermore, all new applications to my device must be approved by my school's IT department, making installing new programs difficult and time-consuming. To attain new skills, particularly in the area of shader coding, I will watch various tutorials to gain a broader understanding of certain function of code. Utilised tutorials/resources will be linked below.

Resource	Topic	How was it used?
Godot 3D Spatial Shaders: Getting Started - YouTube	Shaders	Brief overview of key shader code
Hand drawn shader tutorial in godot (hand drawn/manga/spider verse) - YouTube	Shaders	Use of screen space shaders – particularly depth
How to make Wind Particles in Godot Engine 4 - YouTube	Shaders/Particles	Adapted a tutorial for 2D wind shaders into 3D space
Chatgpt (AI)	Shaders	As I had not used shaders before in Godot, Chatgpt's basic assistant helped me to debug certain niche issues, particularly in my water shader. These suggested fixes were implemented to my shader myself. However, chatgpt also suggested some code methods I could utilise to assist in this.
Chatgpt (AI)	Cleaning code	I struggle from very messy code, and I

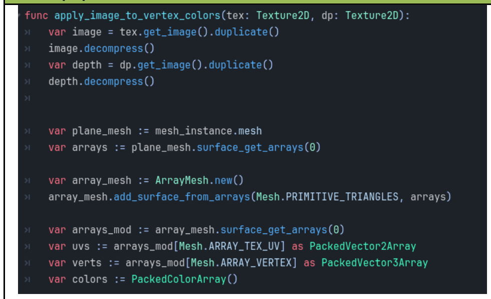
		often forget the purpose of certain variables. Some of the more complex and lesser-used scripts in my submission have been cleaned using AI. This has not affected the integral function of the script or optimisation, simply suggested notation or variable naming. See examples below.

Cleaned scripts and evaluation:

Snippet	Use	Evaluation
 res://MarketColor.gd	Ai was used to suggest variable naming and recommended conventions to loop through materials, cleaning my previous script.	While this did provide a neater script, it did not account for meshes with only one material/colour, which had been done in my original script. This was kept in the final script as a small runtime error to represent this oversight in the code.
Previous script		

```
func _ready():
	var child = $Plane
	var mesh = child.mesh

	for surface in [0, 2]:
		var mat = child.get_surface_override_material(surface)

		if mat == null and mesh != null:
			mat = mesh.surface_get_material(surface)
			if mat != null:
				mat = mat.duplicate()
				child.set_surface_override_material(surface, mat)


		if mat is StandardMaterial3D:
			mat.albedo_color = new_colour
```

```
@export var meshColour: Color
func _ready():
	var Child = $Plane
	
	var mesh = Child.mesh

	for surface in range(3):
		if surface == 1 || mesh.surface_get_material(surface) == null || ( self.get_children().size()>2 &&surface>0):
			pass
		elif self.get_children().size()<2:
			var material = Child.get_surface_override_material(surface)
			if material == null and mesh != null:
				material = mesh.surface_get_material(surface)
				if material != null:
					material = material.duplicate()
					Child.set_surface_override_material(surface, material)
	
	
			if material is StandardMaterial3D:
				material.albedo_color = meshColour
```

Snippet	Use	Evaluation
 Res://minimapmaker.gd	AI was used to rename certain variables and the function itself to a more sensible name. AI also introduced me to the concept of colon automatic type assignment, to keep variables clean, which I copied across many other scripts and code.	Use of AI for this region was highly successful. Variables were given logical names which were easy to read and interpret.
Previous variable/function names	Apply_image_to_vertex_colors(parameters) Prev. draw_map(parameters) mesh_instance Prev. mesh Arrays_mod Prev. editArray	Prompt >"Hey ChatGPT, please evaluate the variable naming in this script and suggest alternate titles?"

```
func apply_image_to_vertex_colors(tex: Texture2D, dp: Texture2D):
	var image = tex.get_image().duplicate()
	image.decompress()
	var depth = dp.get_image().duplicate()
	depth.decompress()
	

	var plane_mesh := mesh_instance.mesh
	var arrays := plane_mesh.surface_get_arrays(0)

	var array_mesh := ArrayMesh.new()
	array_mesh.add_surface_from_arrays(Mesh.PRIMITIVE_TRIANGLES, arrays)

	var arrays_mod := array_mesh.surface_get_arrays(0)
	var uvs := arrays_mod[Mesh.ARRAY_TEX_UV] as PackedVector2Array
	var verts := arrays_mod[Mesh.ARRAY_VERTEX] as PackedVector3Array
	var colors := PackedColorArray()
```

Designing

Game overview

Game title

To name the game, various options were suggested and evaluated by peers and myself.

Name	Significance	Peer rating	Memorability	Drawbacks
Catacombs	Pun on word 'cat,' the main character, and the mysterious and dark style of the game.	8	6	Not particularly thoughtful
Tailend	Connotes ending, an interesting phrase, and also a part of a cat. (tail)	0	5	Doesn't relate to the game
If I go	Impactful, short words. References the ending of the game	2	2	Too long, random

Kindred	Refers to the experience of making friends within the game as you complete quests.	4	7	Mildly irrelevant
Eideling	-ing suffix connotes fragility, while nonsense word/name 'eidel' is often taken to mean tender. Refers to the cat's position.	6	6	Hard to pronounce.
Kith	An old English word for kin, as the game's goal is to meet new NPCs.	0	2	Difficult to pronounce, forgettable
The Stray	The player is alone in an unfamiliar territory. (A stray)	6	7	Too close to 'Stray,' (2022)
Loam	Matches the moody aesthetic of the game, and connotates new soil	2	0	Irrelevant

Final Choice:

Thus, the most highly-rated name was 'Catacombs,' as it provides a relevant insight into the game's content, and is relatively memorable. It utilizes a pun on the name, 'cat,' in catacombs, and also by using this name alludes to the quest-like nature and moody theme of the game.

Game description

The game follows the journey of a cat left stranded in an unfamiliar city, not knowing how to return or speaking the language of most of the inhabitants of the desolate place. Players may explore this haunting landscape to determine the goals of the game, learning segments of information as they experience the vast landscape.

To complete the game, a series of interconnected quests must be completed, culminating in the player receiving a stone used to open a gate leading to a choice – to remain in the town as your new home, or to return to your old home, unknowing of what it may have become in your absence.

The intended audience for this game is primarily those between ages 12 and 17, with enough critical thinking to interpret the themes of the game while not being put-off by the mildly humorous take on mouldy worldbuilding. Given the dark mood of the game, it is likely to be less of a mainstream favourite and more of a niche experience.

The main reason why my game is interesting is the low-poly style of graphics and easy-to-follow, yet customizable gameplay. Without a clear system of progression, the player can find their way through the landscape, kept on track by quests.

These quests are given through NPCs, who give seemingly trivial tasks to the player in order to integrate them into the town, meeting new people and helping others. This creates a sense of community within the game, building the theme of an emotional journey.

The project was particularly inspired by the games mentioned above (*inspiration and points of originality*). However, some components not mentioned were:

Hypixel Skyblock:

- Inspired the non-linear way of interacting with NPCs (Hypixel Contributors, 2025). In Skyblock, quests aren't 'set out,' in a way, you can complete the game in any way you wish. This plays a major role in the addictiveness of the game.

There is No Game (Coolmath Games)

- Inspired the lack of clear instruction on how to play the game, just a strange new world. That's partially why the game is so popular, with a rating of 94% (Coolmath Games, 2017).

Environment

The game takes place in a gothic-style fantasy town, inspired by Czech and central European architecture (See below). The weather involves an eternal dusk and overcast sky, which affects the amount of available lighting. However, this was rectified through the use of a LUT to increase contrast and brighten the environment, along with careful placement of light sources.

Theme

The theme for this year's competition is 'Journey,' whether literal (from one place to another, through things, etc.) or metaphorical (an emotional or spiritual journey). Since the beginning of life itself on our planet, there has always been a journey, and even before that. Our earth is travelling around the sun, which in turn moves through the milky way. Each tiny cell in our body moves, and so does everything else. Growing up is a journey. Dying is a journey, a process. Journey, as described by the dictionary, is 'an act of travelling from one place to another.' But a journey can be so much more than that.

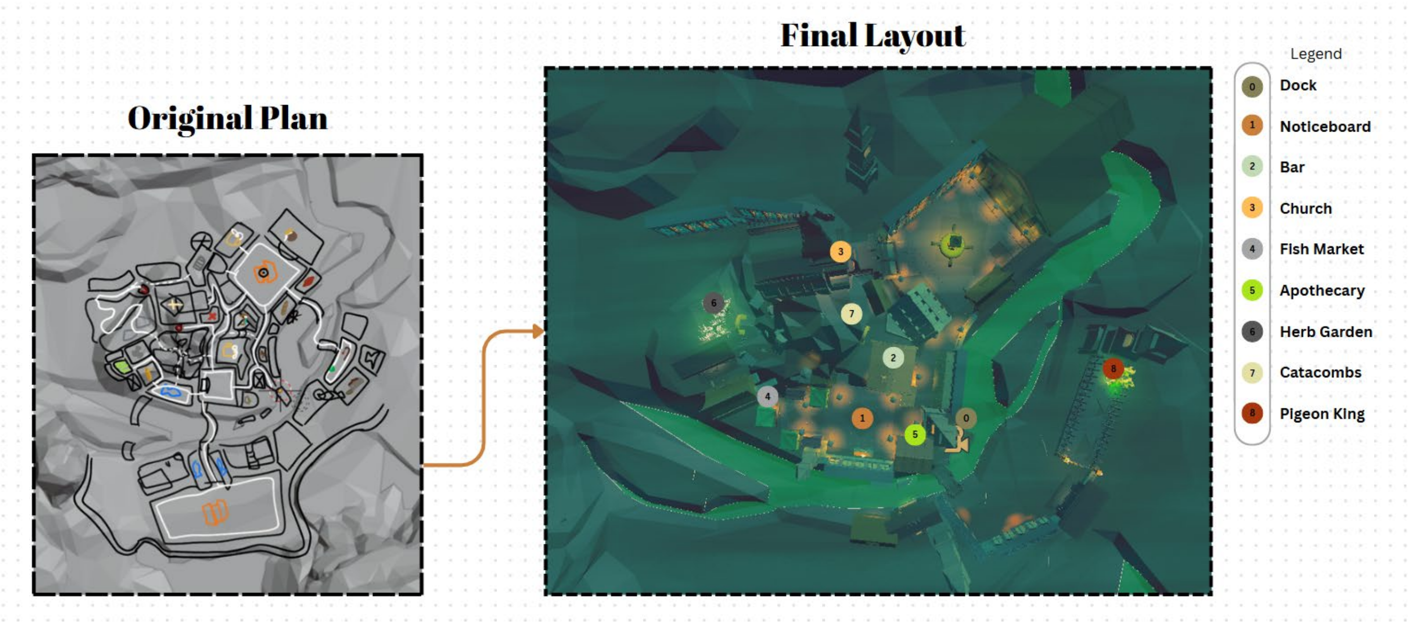
My game incorporates this idea in both a literal and metaphorical way. While the aim of the game is to find a way home (complete a journey), it also provides an interesting twist upon completion – the decision to stay or go. After connecting with various NPCs (particularly apothecary's bird, basilisk, rats) to complete quests in the game, it begins to feel like an emotional journey, too, from the mysterious outsider to part of the community. And begs the question, after all this, where is your home now?

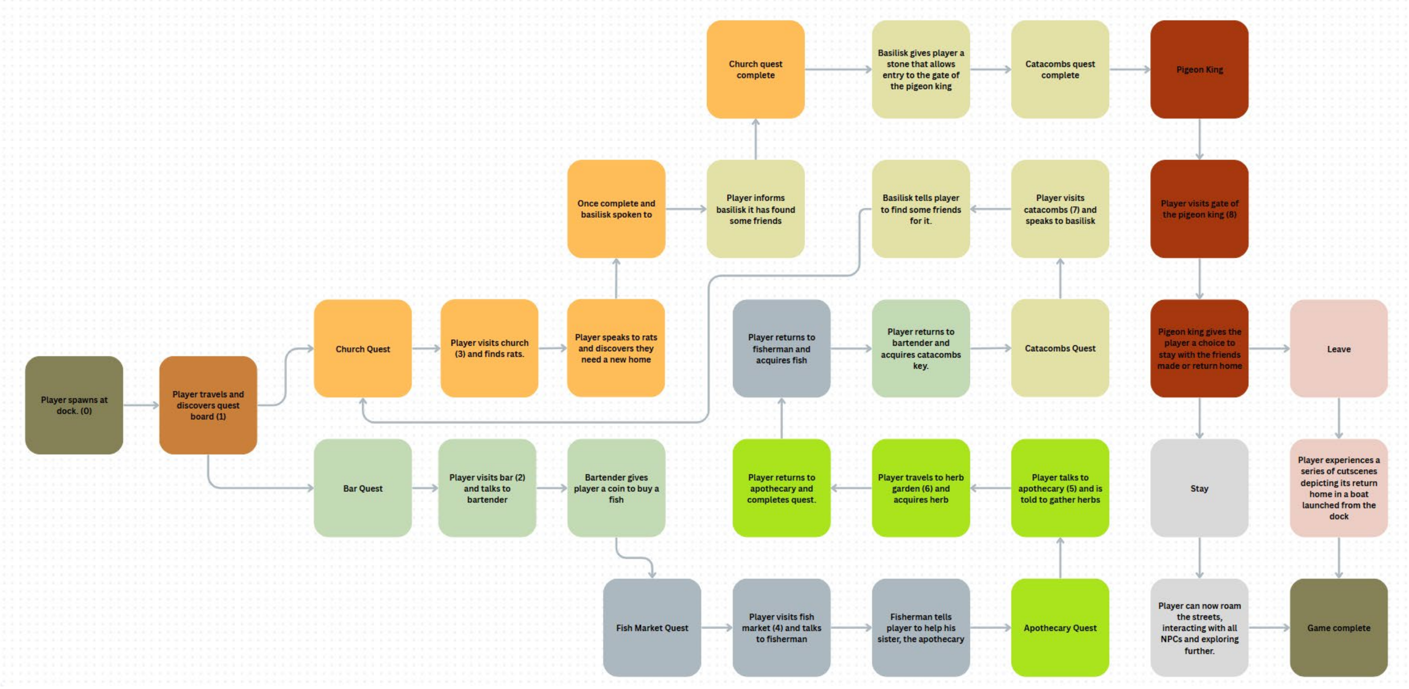
Gameplay/mechanics

Objectives/Goals

To complete the game, the player must finish all quests to reach the ultimate goal of reaching home. This, given the nature of the game, can be simply defined as the title screen, as the choice is given not to return to the expected home.

After being stranded on the dock of this unfamiliar town, the player only has one thing in mind – find its way home. To progress through, the player completes sequential quests and goals depicted on a noticeboard in the central square. These quests are interconnected and involve interactions between multiple NPCs. The progression of these quests is depicted below.




Player spawns at dock. (0)
Player travels and discovers quest board (1)
Church Quest
Player visits church (3) and finds rats.
Player speaks to rats and discovers they need a new home
Once complete and basilisk spoken to
Player informs basilisk it has found some friends
Church quest complete
Basilisk gives player a stone that allows entry to the gate of the pigeon king
Catacombs quest complete
Pigeon King
Player visits gate of the pigeon king (8)
Pigeon king gives the player a choice to stay with the friends made or return home
Leave
Player experiences a series of cutscenes depicting its return home in a boat launched from the dock
Stay
Player can now roam the streets, interacting with all NPCs and exploring further.
Game complete
Bar Quest
Player visits bar (2) and talks to bartender
Bartender gives player a coin to buy a fish
Fish Market Quest
Player visits fish market (4) and talks to fisherman
Fisherman tells player to help his sister, the apothecary
Apothecary Quest
Player talks to apothecary (5) and is told to gather herbs
Player travels to herb garden (6) and acquires herb
Player returns to apothecary and completes quest.
Player returns to fisherman and acquires fish
Player returns to bartender and acquires catacombs key.
Catacombs Quest
Player visits catacombs (7) and speaks to basilisk
Basilisk tells player to find some friends for it.

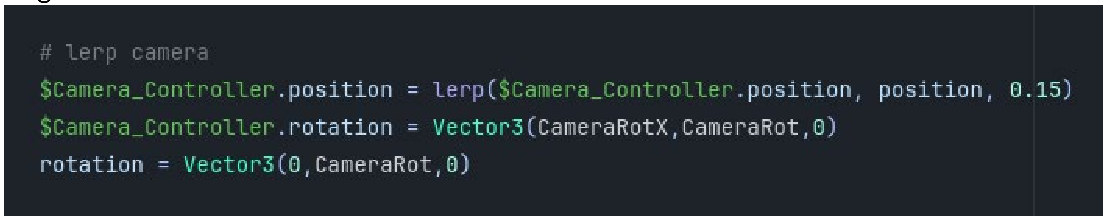
Below is a table used for planning of the game.

Quest	Steps
Church quest	• clear the rats out of the church. ~> open catacombs with key given by bar quest completion
Catacombs quest	• find the (vaguely menacing) serpent some friends. ~> lead the rats from the church to the catacombs with the promise of food to eat? / a place to live
Bar quest	– get the guy some fish. ~> go to fish market and pay with 1 coin from bartender and another from...?

Fish market quest:	• find a gold coin to pay for your fish. (Given one by bartender, find another somewhere else)
Apothecary quest:	• follow the rats' suggestions to visit the apothecary, then visit the hidden garden ~> visit the apothecary, then visit the garden. Bring back some herbs.

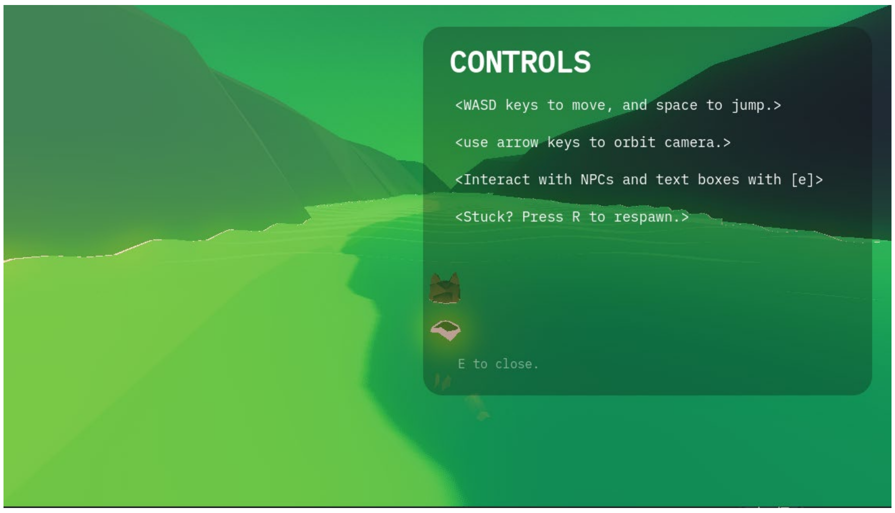
Perspective/Controls

The game is played in a three-dimensional landscape, using an orbiting camera around the player (a dark-coloured cat) from a third person perspective (Controlled arrow keys) and all-axis movement (Controlled WASD) with basic physics. This provides for an immersive feeling and a way to exploit all features of the game. To interact with NPCs, the player must move to within range and press E to interact. NPCs may offer choices, which can be selected with the mouse. The game's controls are intentionally simple, to make playing an enjoyable experience.
The player can be rotated independently of the camera (perspective), and movement direction accounts for both these transformations. This allows for an immersive experience with high levels of control.
Furthermore, the camera does not strictly follow the player, with a slight delay for a more organic feel.



```
# lerp camera
$Camera_Controller.position = lerp($Camera_Controller.position, position, 0.15)
$Camera_Controller.rotation = Vector3(CameraRotX,CameraRot,0)
rotation = Vector3(0,CameraRot,0)
```

To inform the player of these controls, a brief tutorial appears before entering the game. Some controls (zoom/minimap zoom with mouse wheel/touchpad) are not explained, for simplicity, as they are not integral to gameplay.



CONTROLS

<WASD keys to move, and space to jump.>

<use arrow keys to orbit camera.>

<Interact with NPCs and text boxes with [e]>

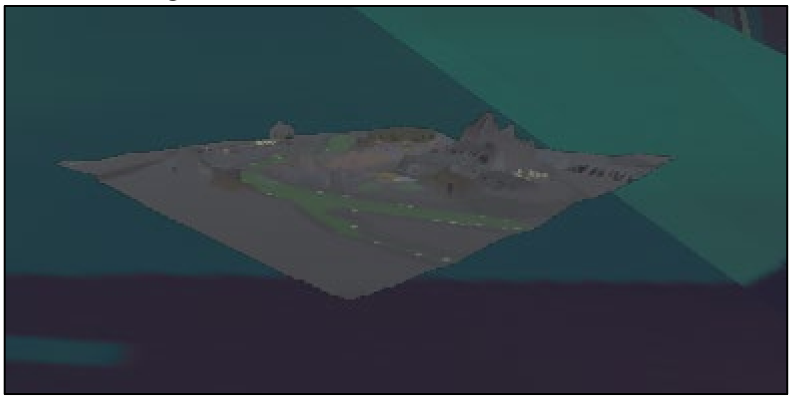
<Stuck? Press R to respawn.>

E to close.

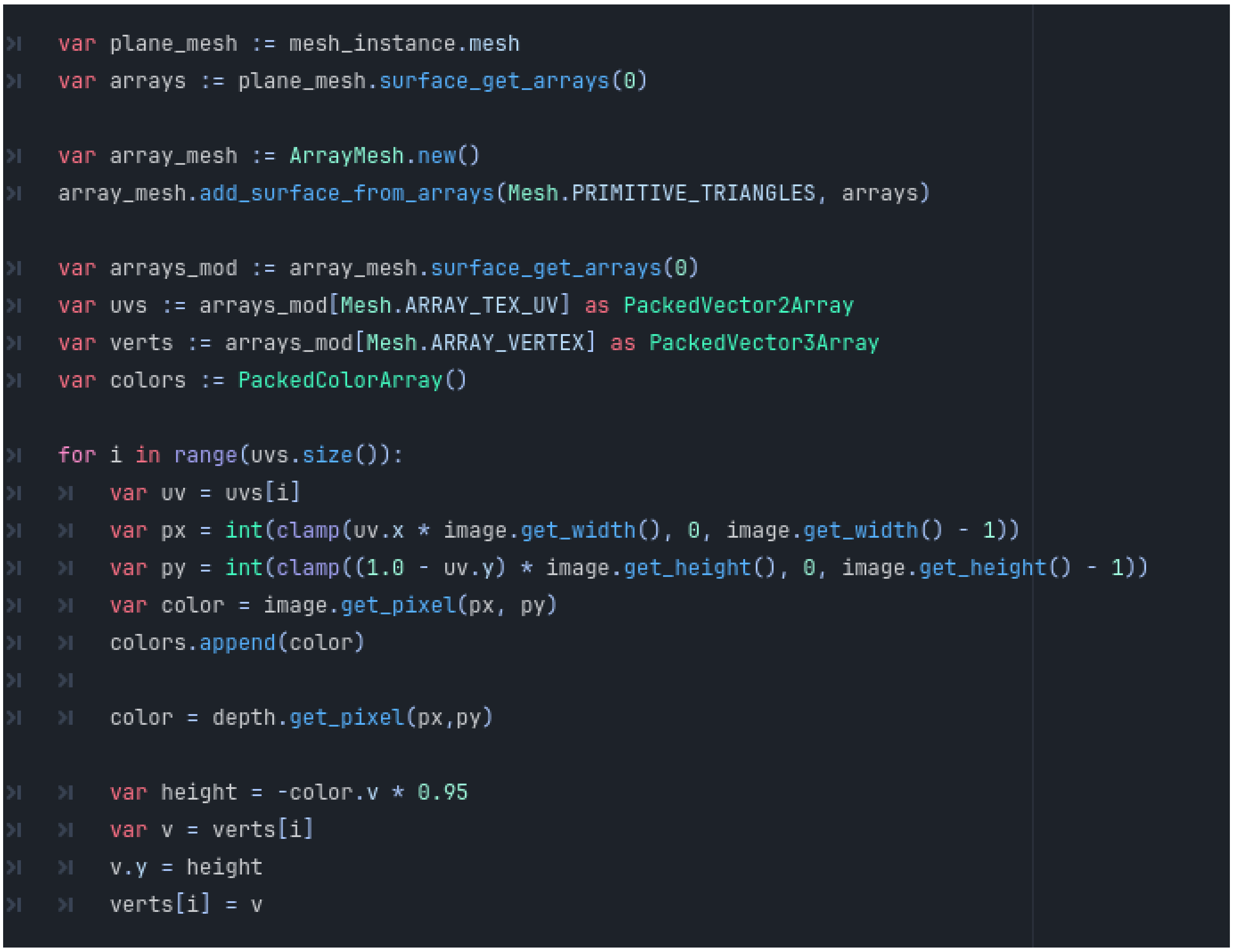
Instructions/Tutorials

In order for players to discover the controls of the game, a small how-to-play box is included during the first cutscene when the game is opened. This includes how to interact with NPCs, orbit camera and move the player. Furthermore, players are reminded of npc interaction controls when close to or interacting with an NPC through a small pop-up. I deliberately avoided using a tutorial, as this would make the game feel less organic and add unnecessary structure. The game is intended to be a learning experience, where the player discovers, rather than follows. This choice was made to enhance the 'journey' aspect of the game.

Interesting Features to Note



Minimap: A toggleable rotating 3D minimap. Created using a top-down viewport projection of the scene, and displacing vertexes on the mesh by a depth draw shader in a second render from this viewport. Then projected into a second viewport and rotated. This camera's feed is shown on the main game, in the top left-hand corner. The mesh is generated this way to optimise the number of vertexes required. A segment of the script to generate this mesh is shown below.

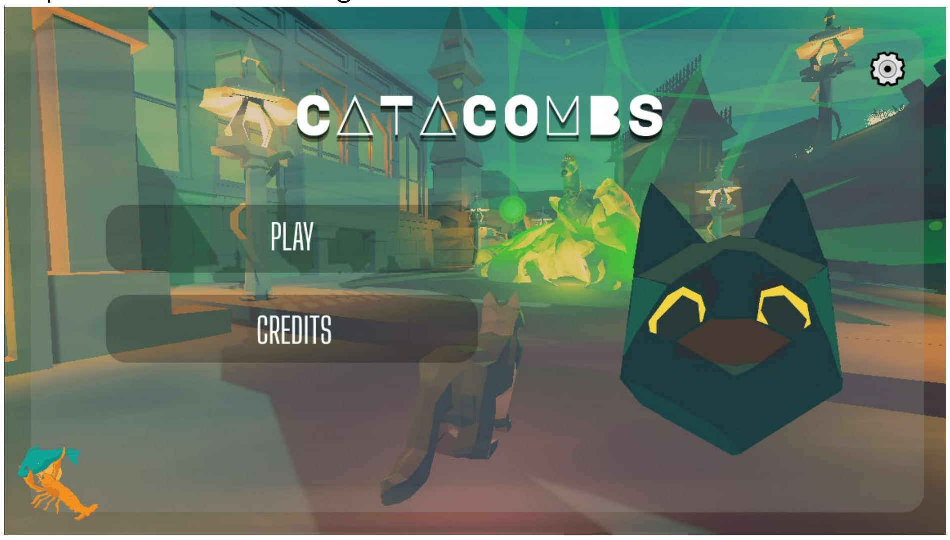


```
	var plane_mesh := mesh_instance.mesh
	var arrays := plane_mesh.surface_get_arrays(0)

	var array_mesh := ArrayMesh.new()
	array_mesh.add_surface_from_arrays(Mesh.PRIMITIVE_TRIANGLES, arrays)

	var arrays_mod := array_mesh.surface_get_arrays(0)
	var uvs := arrays_mod[Mesh.ARRAY_TEX_UV] as PackedVector2Array
	var verts := arrays_mod[Mesh.ARRAY_VERTEX] as PackedVector3Array
	var colors := PackedColorArray()

	for i in range(uvs.size()):
		var uv = uvs[i]
		var px = int(clamp(uv.x * image.get_width(), 0, image.get_width() - 1))
		var py = int(clamp((1.0 - uv.y) * image.get_height(), 0, image.get_height() - 1))
		var color = image.get_pixel(px, py)
		colors.append(color)
		
		color = depth.get_pixel(px,py)

		var height = -color.v * 0.95
		var v = verts[i]
		v.y = height
		verts[i] = v
```

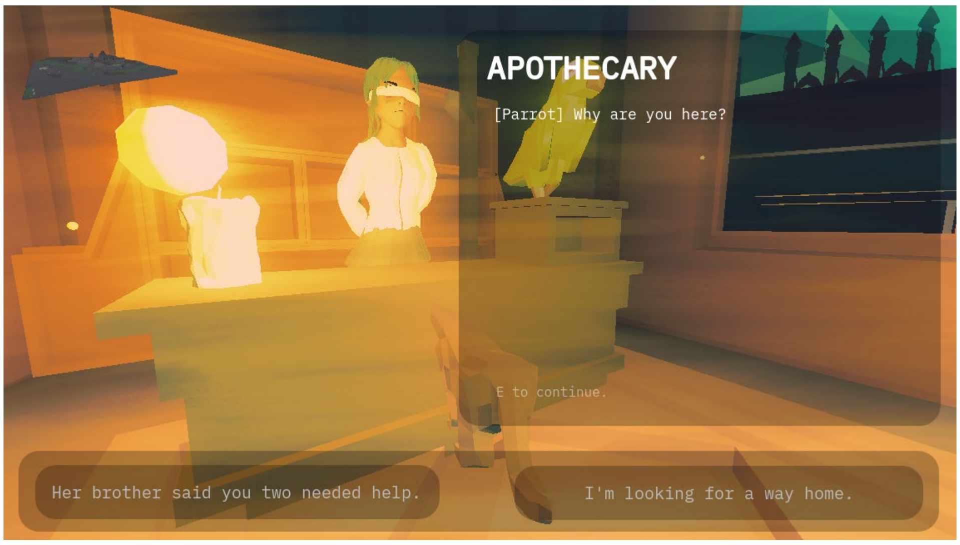
Visual and Audio Design

Style

The visual style of the game is primarily a low-poly, moody and dark environment with bloom lighting effects and vivid colours. Below is a picture of the starting screen, which is condensed into three simple icons. This streamlines the UX and leaves the player with a clear path to follow. Otherwise, no other factors contributed to the inspiration of this design.

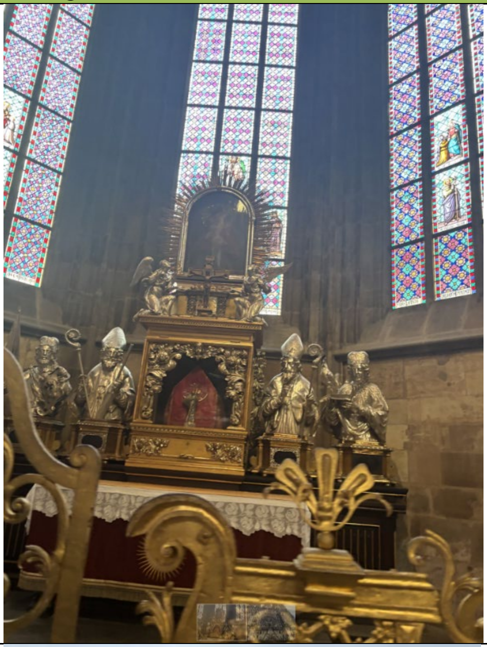
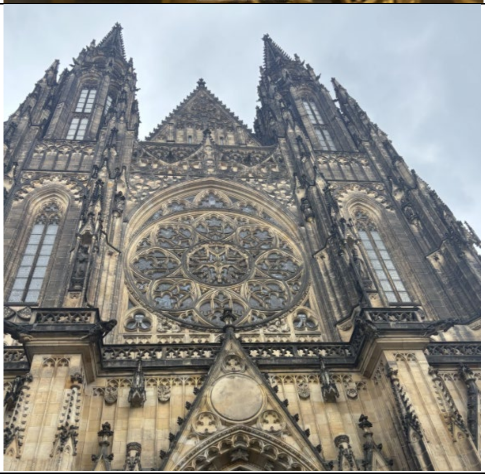
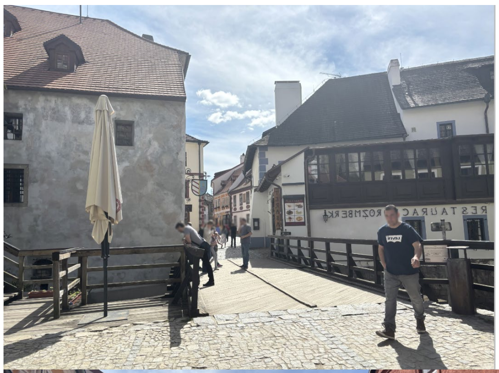
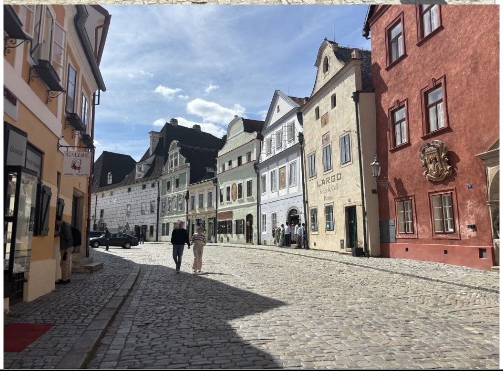


The dialogue component of the game is a particular example of how opacity is used in the game to differentiate key information. Backgrounds of text boxes are slightly translucent, to keep the player immersed in the gameplay. However, key information has a more opaque background to draw the player's attention to these locations, whereas reminders ('E to continue') are less visible.



APOTHECARY
[Parrot] Why are you here?
E to continue.
Her brother said you two needed help.
I'm looking for a way home.

The game was not particularly inspired by any particular concepts for the low-poly style, however, the dark ambience and mouldy worldbuilding referenced Elden Ring (as above) and Studio Ghibli, as well as various other indie games (Appendix 1). It particularly mirrors key aspects of gothic literature. The architecture of the game was, as mentioned earlier, inspired by Czech and central European heritage style.

Location	Image	Inspiration use
Chapel of St. Adalbert, St Vitus' Cathedral		Inspired the central church ornament
St Vitus' Cathedral, Prague		Referenced to create central steeple of church
Český Krumlov	 	Inspired the inn/bar, tenements and apothecary buildings.

Prague markets		The shape and atmosphere of the Prague markets created a framework for the fish market.
Vienna		Vienna's rich statuework also inspired the church altar.

*All images sourced from friends or family of participant

To link this style to the theme of the game, I decided to give the game a deliberate desolate and hostile vibe. However, by the end of the game, small quirks like the behaviour of the NPCs make the landscape seem less hollow and empty, furthering the theme of a journey.
The music also compliments this visual style, designed with a thin texture and varied instrumentation, along with repetitive parts to create a liminal style. A space modifier was also added to the mixdown to improve the eerie quality. Very few external and no platform assets were used, a deliberate choice to make the game more unique. The explanation for the few assets that were used was the difficulty of creating new fonts and files.

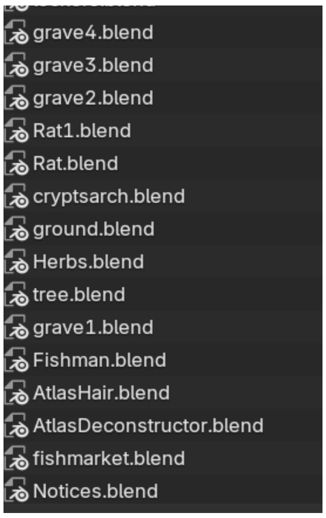
External assets:

Name	Type	Source
IBM Plex Mono	Font	Google Fonts IBM Plex Mono - Google Fonts
Major Mono Display	Font	Canva
Big Shoulders Display	Font	Canva

Process

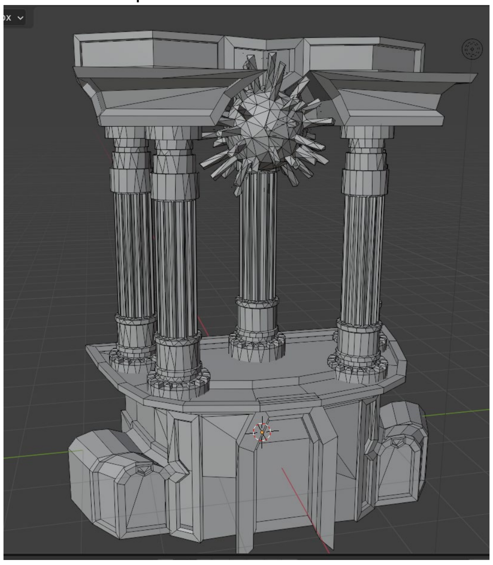
To achieve my desired visual style, I created all assets with no colour or texture in blender. This was a necessary step, as adding these would crash the application. Then, I added colour in Godot.

These files were named and stored in a separate folder. In future, I would improve my naming conventions on these files to make them more readable.



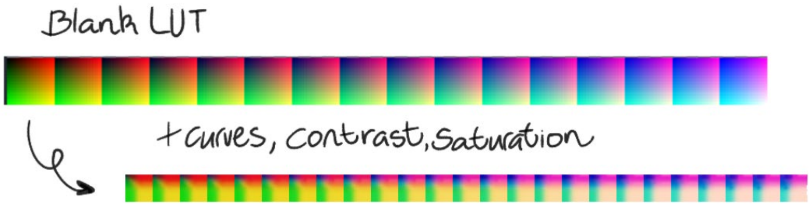
```
grave4.blend
grave3.blend
grave2.blend
Rat1.blend
Rat.blend
cryptsarch.blend
ground.blend
Herbs.blend
tree.blend
grave1.blend
Fishman.blend
AtlasHair.blend
AtlasDeconstructor.blend
fishmarket.blend
Notices.blend
```

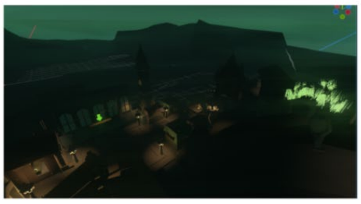
Topology was kept simple to lower vertex count and improve performance. However, some models still required a larger number of vertexes for curved surfaces. Both of these are seen in the church altar model. I particularly made an effort to keep any religious icons to a minimum, creating a place of worship in my game that was objective, and wouldn't detract from the overall experience.

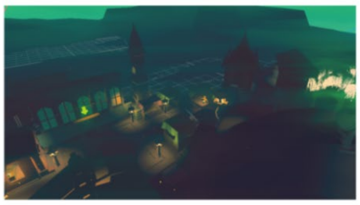


Lighting played a large role in creating the desired style, which was created through a series of omnilights. Light was used to draw attention to major features, with greenish lighting highlighting the pigeon king's abode, the noticeboard, and the herb garden. Green, as a colour, is used throughout the game as a cohesive visual element in all of the more arcane interactions, including the basilisk's stone, the pigeon king, and the herbs.

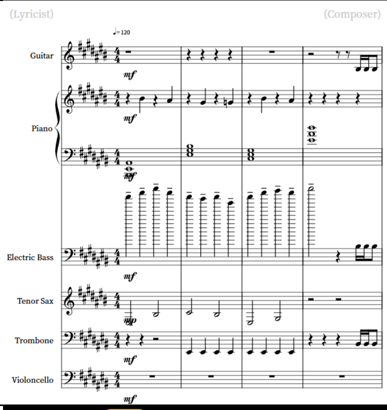
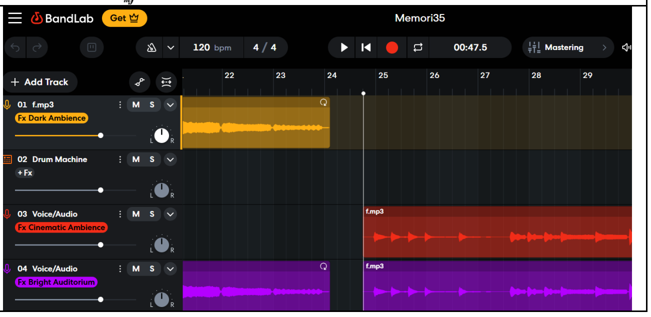
The scene, however, was slightly dark after these techniques, and thus, a colour LUT (look up table) was implemented to create more vivid colours and contrast. This filters colours to change the appearance of the scene, using a 3D texture.

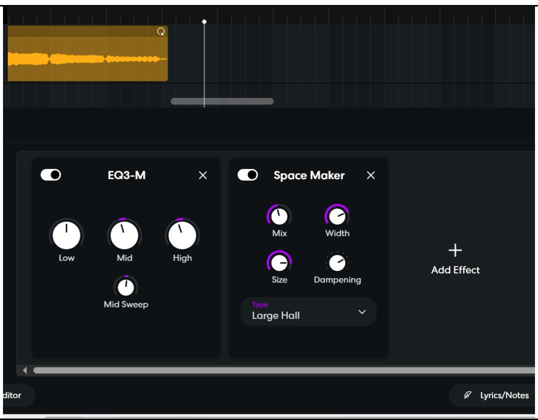



Before


After

Music was created for the game using a combination of Noteflight (opening theme and game music riffs) and Bandlab to create drum patterns (game music), as well as apply FX. The process for producing this music is as follows:

Stage	Process	Image
Stage one: Riffs	Major themes and riffs are produced note-by-note in Noteflight.	 (Lyricist) (Composer) Guitar Piano Electric Bass Tenor Sax Trombone Violoncello
Stage two: Collation	Riffs or tracks are collated and placed into a cohesive tune, and edited where necessary.	 BandLab Memori35 120 bpm 4 / 4 00:47.5 Mastering Add Track 01 f.mp3 Fx Dark Ambience 02 Drum Machine + Fx 03 Voice/Audio Fx Cinematic Ambience 04 Voice/Audio Fx Bright Auditorium

Stage three: Effects	Sound is tuned to create the hollow ambience required for the game.	 EQ3-M Low Mid High Mid Sweep Space Maker Mix Width Size Dampening Add Effect Lyrics/Notes Type Large Hall

The main game audio ('Seven Town'), was composed centrally around electric guitar sequences, returning throughout the theme. The texture was manipulated throughout the song to produce an effect similar to that of an auditory journey. The choice of naming convention for this piece was a simple process, in which the setting, 'town' was combined with a concept, in this case, the number seven. Seven was chosen due to its encompassing nature. The world we live in has seven continents, seven wonders, seven days of the week, seven colours of the rainbow, and even seven chakras. Many religions hold the number seven dear, which is why I have selected it as my song title.

After some evaluation, it was decided that no other sound effects would be required, as after surveying 10 regular players of games, 70% agreed that they either played without sound on, or preferred only music to effects and music. This was also rationalised in the fact that cats are generally thought of as quiet creatures.
To convert concepts to finished products, I often begin by blocking out major visual themes. For example, creating lighting maps, using ibispaintX to visualize LUT effects and using placeholder particles to simulate my desired style greatly assisted in visualizing concepts. Furthermore, mapping out reference images in a similar way to appendix 1 and the reference map above also helped create buildings.

Reflecting

Testing, fixing and project execution

Testing

The game was tested by five people, including two family members, my teacher, and two randoms. When testing, I instructed them to try and figure out how to play the game without any explicit instructions, with the only objective being 'find your way home.' After testing, I asked for feedback on visual appeal, playability and adherence to design. Major issues were collated into the table below.

Fixing

The issues identified were evaluated and solved in a way that was deemed sufficient and constructive. This is demonstrated in the table below.

Problem	Addressed?	Explanation	Solution
NPCs named incorrectly	🟩	One NPC had in improper internal name, which resulted in incorrect dialogue.	This was easily rectified by correcting the internal name.
Quest milestones loaded incorrectly	🟩	Quests did not show the correct stage of completion on the noticeboard (added below)	As above, fixed with code tweaks.
Movement animation was too slow	🟩	Movement speed animation was set to 0.85x, which seemed unnatural to some players.	This was increased to 1x.
Quests completed in wrong order	🟩	Errors occurred where the church rats quest did not need to be completed in order to complete the basilisk catacombs quest.	This was simply fixed with some code changes.
Hard to see what needs to be done	🟩	Players found it difficult to keep track of quest order and stage.	A noticeboard was added to keep track of this and remind players what to do next
Was hard to determine what were NPCs	🟩	NPCs such as the rats were difficult to see.	To solve this, a small message was added when approaching an NPC reminding of interaction controls.
Player clipped and was stuck in mesh	🟩	Meshes were too complex and had odd normal alignments, causing collision errors	Meshes were simplified in blender, and then backface collision enabled.
Needs a restart button	🟧	This was a recommendation to fix mesh issues and getting stuck. However, this would reset all progress.	Thus, an unstuck button was added to compromise.
Should normalize NPC names to have location on top and person speaking in brackets	🟥	This was not addressed, as the layout was not intended to display a	-

	🟥	location, and the parrot NPC used as an example was an exception, contributing to the lore that the apothecary does not speak the cat’s language.	

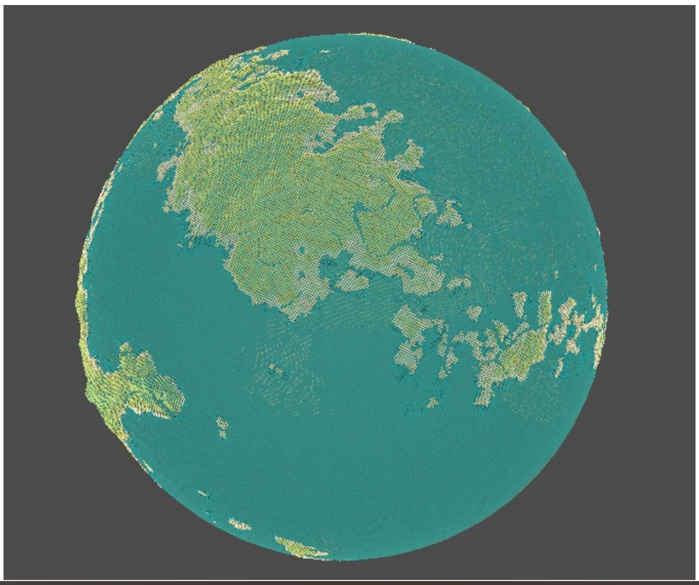
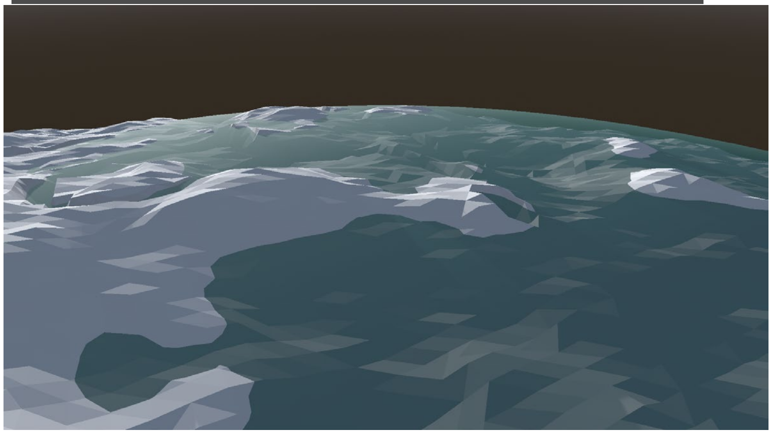
Project execution

After the following modifications were made, the game was complete, as of 30/05/2025. [10:01 pm]
The submitted game included the majority of planned features, however, some features, particularly physical decorations, were scrapped to improve performance, given the lack of functionality.

Scrapped features:

Feature	Reasoning
Candle particles	Greatly affected performance, and wasn’t worth the visual improvement
Customisable cat	Made the game less cohesive, if players could choose the colour of their cat, the distinctive logo and thematics would be lost.
Randomly generated pigeon realm	Took too long to load and made the swap between the game and dream scene too obvious.
Volumetric clouds	Didn’t match the visual style of the game
Currency: Players had hunger, and could buy things like speed potions by selling fish they caught in the river	Seemed to take the player off course from the game’s main flow.

In summation, I, as the producer of said game, am extremely satisfied with the results.
I intend to create more games and enter this competition in the future. For next time, I would like to work on how to create complex simulations similar to those done by Sebastian Lague (Etc. Marching cubes) in Godot. I’ve done them before in P5.js (Etc. procedural animation), but the node system of Godot is still unfamiliar to these concepts. Below is an image from a script I produced when testing some other game concepts. This script procedurally generates a planet, and then meshes it using marching cubes to create chunked terraformable terrain. This idea was scrapped for obvious performance issues and lack of experience reasons.



I'd also like to be more advanced in texturing (potentially finding a new device to run blender, as this laptop does not have enough capacity to texture) and in shader creation.
I believe my approach of starting early with 3D assets worked well, as it allowed me to visualise what I was trying to create, providing motivation and satisfaction. However, I would spend more time in the ideation phase, as after the event, I would have liked to fiddle with a few more ideas before diving into the comfortable one. Finally, my skill in musical composition was very much lacking, as I am certainly not an expert, while I did attempt to upskill, I would have liked to study more musical theory to create my pieces. I did like the way I dealt with my software issues, which resulted in a clean product with a low-poly style that I was quite satisfied with.
I believe I managed my time well, completing everything and having my project ready for submission by 11/07/2025. I liked my approach to working alone, by learning new skills before trying to attempt something alone, and leaning heavily on overview and tutorial videos. (as mentioned above)

Summary

And that's all, folks! I hope you had a great time reading a little about my game. I figured I might as well give it a shot.
-Sarah Warburton, St Aidans (Lobster and Fish)

References

Curious Archive. (2024, November 15). The Most Powerful Type of Worldbuilding. YouTube. https://www.youtube.com/watch?v=iHSEFMYjbnE

Fandom Contributors. (2025a). Beast Island. She-Ra and the Princesses of Power Wiki; Fandom, Inc. https://she-raandtheprincessesofpower.fandom.com/wiki/Beast_Island

Fandom Contributors. (2025b). Sky: Children of the Light Wiki. Sky-Children-of-The-Light.fandom.com. https://sky-children-of-the-light.fandom.com/wiki/Sky:_Children_of_the_Light_Wiki

Games, C. (2017, February 9). There Is No Game. Coolmathgames.com. https://www.coolmathgames.com/0-there-is-no-game

Godot. (2025). Exporting projects. Godot Engine Documentation. https://docs.godotengine.org/en/stable/tutorials/export/exporting_projects.html

Hinske, S., Lampe, M., Magerkurth, C., & Carsten Röcker. (2007). Classifying pervasive games: On pervasive computing and mixed reality. *Classifying Pervasive Games: On Pervasive Computing and Mixed Reality*, *1*(1). https://www.researchgate.net/publication/228626515_Classifying_pervasive_games_On_pervasive_computing_and_mixed_reality?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6Il9kaXJlY3QiLCJwYWdlIjoiX2RpcmVjdCJ9fQ

Hypixel Contributors. (2025). Hypixel SkyBlock Wiki. Wiki.hypixel.net. https://wiki.hypixel.net/Main_Page

Jorstad, L. (2018, August 3). Nausicaä of the Valley of the Wind (1984): The Movie Structure Archives - The Novel Smithy. The Novel Smithy. https://thenovelsmithy.com/nausicaa-1984-msa/

Play Instinct. (2023, December 8). Hide. Dorian Concept. https://dorianconcept.bandcamp.com/album/hide-2

Rosenfeld, D. (2025). Minecraft Volume Alpha. C418. https://c418.org/albums/minecraft-volume-alpha/

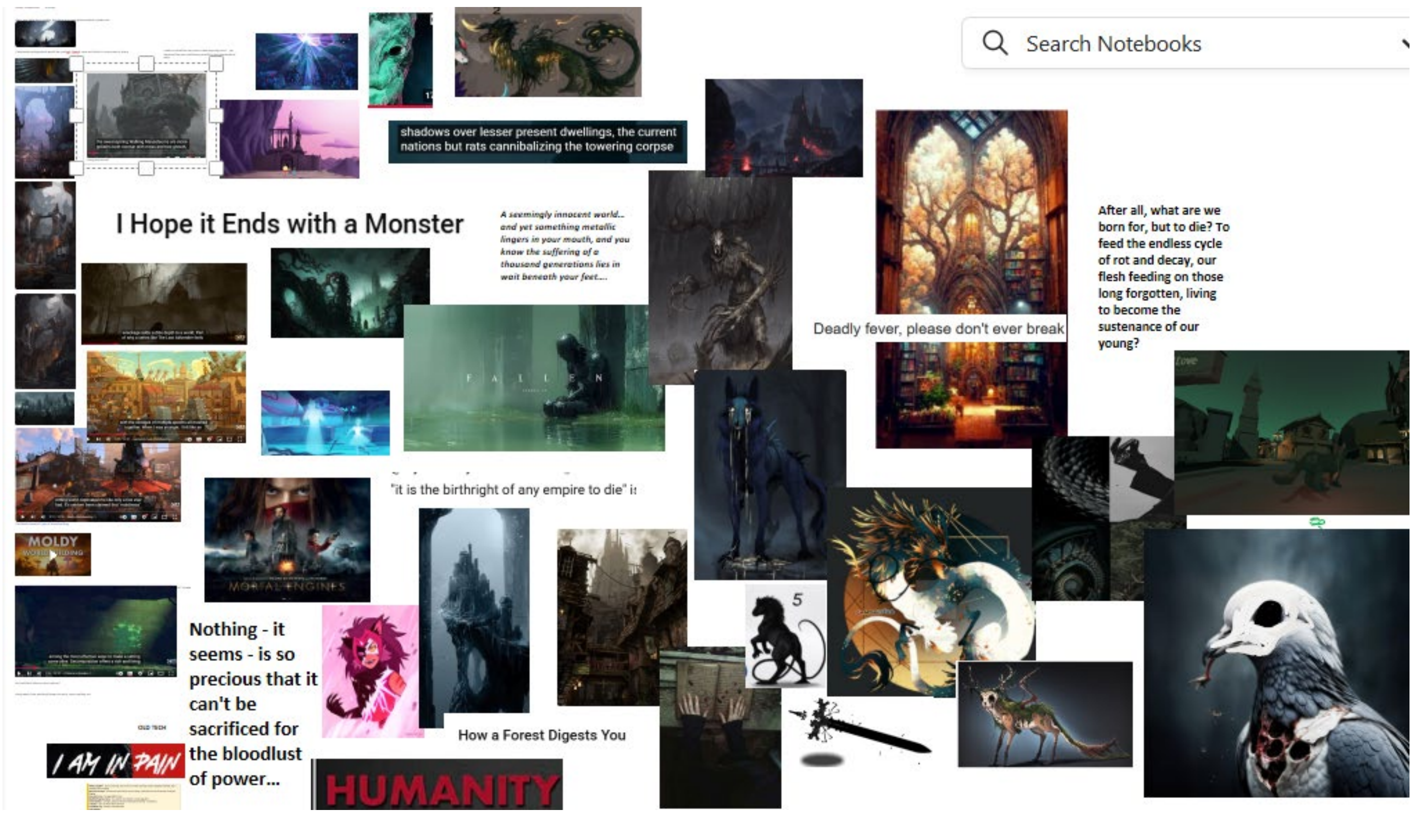
Wikipedia Contributors. (2020, June 8). Kiki's Delivery Service. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Kiki%27s_Delivery_Service

Wikipedia Contributors. (2022, June 29). Stray (video game). Wikipedia. https://en.wikipedia.org/wiki/Stray_(video_game)

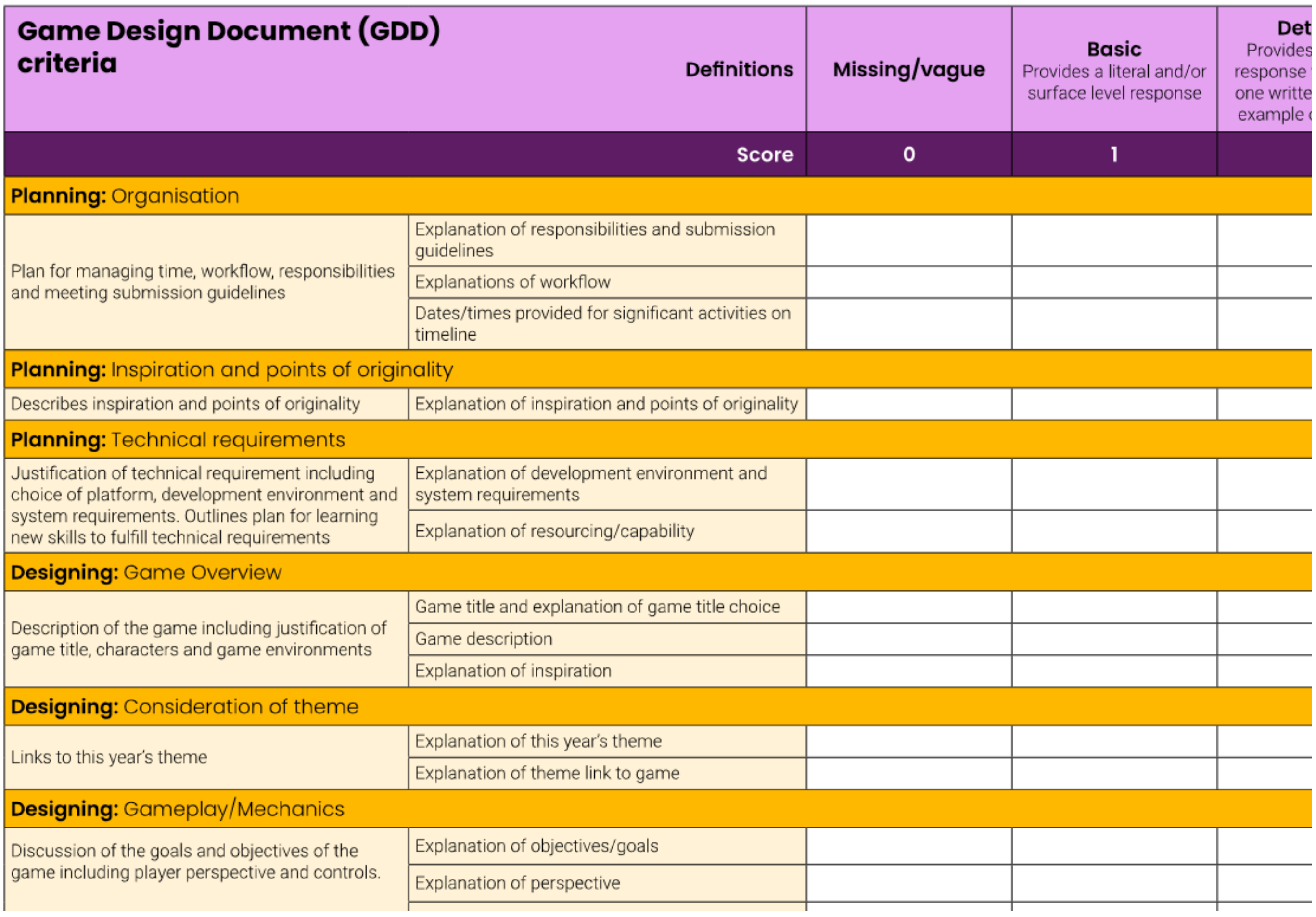
Wikipedia Contributors. (2023, September 12). Rain World. Wikipedia. https://en.wikipedia.org/wiki/Rain_World

Appendix

Appendix 1: Full reference image inspiration board (Images not sourced)


Search Notebooks
shadows over lesser present dwellings, the current nations but rats cannibalizing the towering corpse
I Hope it Ends with a Monster
A seemingly innocent world... and yet something metallic lingers in your mouth, and you know the suffering of a thousand generations lies in wait beneath your feet....
After all, what are we born for, but to die? To feed the endless cycle of rot and decay, our flesh feeding on those long forgotten, living to become the sustenance of our young?
Deadly fever, please don't ever break
F A L L E N
"it is the birthright of any empire to die" is
MOLDY
MORTAL ENGINES
Nothing - it seems - is so precious that it can't be sacrificed for the bloodlust of power...
How a Forest Digests You
I AM IN PAIN
HUMANITY
5

Appendix 2: Annotated judging rubric



Game Design Document (GDD) criteria	**Definitions**	**Missing/vague**	**Basic** Provides a literal and/or surface level response	**Det** Provides response one writte example (
	Score	**0**	**1**	
Planning: Organisation				
Plan for managing time, workflow, responsibilities and meeting submission guidelines	Explanation of responsibilities and submission guidelines			
	Explanations of workflow			
	Dates/times provided for significant activities on timeline			
Planning: Inspiration and points of originality				
Describes inspiration and points of originality	Explanation of inspiration and points of originality			
Planning: Technical requirements				
Justification of technical requirement including choice of platform, development environment and system requirements. Outlines plan for learning new skills to fulfill technical requirements	Explanation of development environment and system requirements			
	Explanation of resourcing/capability			
Designing: Game Overview				
Description of the game including justification of game title, characters and game environments	Game title and explanation of game title choice			
	Game description			
	Explanation of inspiration			
Designing: Consideration of theme				
Links to this year's theme	Explanation of this year's theme			
	Explanation of theme link to game			
Designing: Gameplay/Mechanics				
Discussion of the goals and objectives of the game including player perspective and controls.	Explanation of objectives/goals			
	Explanation of perspective			

Game judging criteria **Star rating**	★	★★	★★★	★★★★
Score	4	6	8	10
Functionality				
Completion, playability and testing	The game has significant glitches that impact playability.	The game is difficult to experience without intrusive bugs.	The game is largely stable and bugs do not impact playability.	The game has rare minor gli and bugs. It may have room further improvement.
Visual and audio design				
How do the visual and audio elements impact user experience	Includes only platform assets in its visual design. Audio elements are missing or from platform assets only.	The game has the beginnings of a consistent visual design, but it is incomplete. May have some original work. May or may not have original or platform audio elements.	The game uses some original visual design throughout which is appealing and related to the theme. Audio elements are included which may or may not be original.	The game uses a consisten visual design throughout wh appealing. Main characters, backgrounds and character are original (not from assets Audio elements are original.
Gameplay				
Ludology: Events, actions, rules and choices within the game	The game's goals are difficult to understand, some may not be achievable, and/or the game's overall objective is somewhat unclear.	The game's goals are reasonably clear, but there is some confusion around the purpose and method of play that the game uses.	The game's goals are mostly clear, but there are some elements of play that are confusing, unfinished, or do not easily correlate with overall game objectives.	The game's goals are mostl and players can work out ho the game is meant to be pla Game mechanics are thoug designed but may have inconsistencies across level cutscenes don't use movement mec However, interaction remains the sa scenes.
Technical competency				
Programming, coding and design	The game's programming is problematic, affecting the design, functionality, or gameplay.	The game demonstrates a basic understanding of programming with examples of successful simple coding.	The game demonstrates examples of successful programming and evidence of some more advanced coding. There may be flaws in coding and design.	The game's programming is thoughtful and mostly well executed. There may be a nu of minor technical issues pr in the finished version. All programming works, howeve It's tame. I know I can do better. cube noise planets, for pete's sa something interesting. A depth n or like a minimap that loads by r
Engagement				
Is it fun to play? Is there	The game is hard to engage with	Parts of this game are engaging	The game has engaging and fun	The game is engaging and f