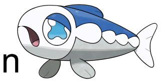
Team Milestone Report

Milestone 1

Team Name: Sang Rin 

Game Name: MANZO

Milestone: 1

Score: _____/67 pts

1) List all team members and their roles. (1 pt)

Gyuwon Na	Tech lead
Won Kim	Art lead
Seokhwa Hong	Test lead / sub art & audio
Seohyeon Min	Producer / sub art & audio
Jiyeop Kang	Level designer

2**) Simply explain the team's structure and organization. (2 pt)**

Our team has a flat and equal structure, where everyone is on the same level. This allows us to have open, comfortable communication, where all members can share ideas freely without hesitation.

3) Give a 3~5 sentence description about your game. (5 pts)

Our game is a rhythm-based exploration set underwater, where players move in sync with the rhythm by clicking in the desired direction. As the player dive deeper, they follow mysterious Morse code signals that lead them to strange fish. These strange fish seem to be hiding secrets about the missing grandmother. The shallow waters are bright and playful, but the deeper you go, the darker and more unsettling it becomes.

4) What are the gameplay instructions? (5 pts)

Click your mouse on the beat to move the player in that direction on the next beat. Avoid rocks using the same timing and try to catch fish for money. This money will allow you to upgrade your submarine, letting you explore deeper into the sea. We implemented moving, catching fish and earning money so far.

5) What are the primary methods of gameplay input? (Keyboard, mouse, VR, touch screen, Kinect, etc?) (3 pts)

We use the mouse and keyboard as an input. We still plan to implement a X-box controller later.

6) How many players can play this game at once? (1 pt)

It is a single player game.

7) List the major components and systems that were designed and created during this milestone for each of the following disciplines: (10 pts)

Debugging, Testing, Optimization
There is a imgui which is something like a console that only prints meaningful values for Debugging. I made it for easier variable tracking. Using imgui, all the printings are well-structure, so easy to understand the value and interact

Art
We are not making any more improvements on art from milestone0. We made all the concept arts that we needed for milestone 1 during milestone 0. We wanted to focus on game logic and engine design more than art. Contents will be added later.

- Gameplay, Game Design
- Music, SFX, Audio
- Physics
- Tool Development, Building Pipeline

8) List 5 quality attributes that are your project's area of focus during this milestone, or the qualities that your project excels in. How does your project demonstrate each quality attribute? (10 pts)

9) Rate your project in the following categories 1~10: (Your grade will not be changed based off of your ratings.) (5 pts, regardless of rating)

- Gameplay 7
- Visuals 3
- Audio 2
- Technical feats 10
- Overall presentation and cohesion 10

10) Look over this long list of features. For each one, mark if this feature has already been implemented, will be implemented in the future for the final product, or will not be implemented for your project: (5 pts)

Feature	Implemented	Will Implement	Won't Implement
Voice-over audio			X
In-game SFX		O	
Background music		O	
Spatial audio		O	

UI/menu SFX		O	
Advanced/dynamic audio filters		O	
Dynamic lighting/shading		O	
Vector graphics		O	
Sprite-based animation system		O	
Sprite scaling		O	
Sprite rotation		O	
Particle effects	O		
Image masks		O	
Partial-transparency image blending or alpha blending methods		O	
Parallax backgrounds		O	
Multiple layers of background graphics		O	
Kinematic/skeletal art		O	
Scripted motion using vector paths		O	
Animation tweens			X
Multiple levels/environments		O	
Asymmetrical gameplay			X
Advanced physics simulations		O	
Local multiplayer			X
Networked multiplayer			X
Downloadable content or online content fetching			X
Any other network features			X
Integration into web technologies (Such as web portals)			X
Component-based architecture	O		

Game objects use C++ interfaces	O		
Game objects are data-driven from factories			X
Menu systems		O	
HUD		O	
File parsing for gameplay content, such as levels, scenarios, enemies, etc.	O		
In-game level editor			X
Scripting language integration		O	
Metaclasses, reflection, data-binding		O	
Live object property inspection			X
Art pipeline tools		O	
Audio pipeline tools		O	
Engine/game build (compilation) tools		O	
External gameplay editor			X
In-game gameplay editor			X
Testing tools and advanced debugging features			X
Ability to jump into the game in a given scenario for testing		O	
Fast-forward and rewinding gameplay			X
Extensive/complex debug drawing		O	
In-game performance visualization		O	

11) What other advanced features do you plan on implementing that are not listed, above? What other advanced features have you implemented, already, that's not listed, above? (0 pts)

Procedural Animation:

We plan to use procedural animation to create more natural movements for marine creatures and structures. This approach will be applied in various contexts, such as boss battles and auxiliary drones beside the main marine drone.

Lighting Manager:

Since clear distinction between light and dark areas is essential, we will use a lighting manager to handle this effectively.

Color Manager:

To easily manage the color tones of the ocean and the grayscale shades of the deep sea, we are developing a color manager.

12) Assess your project's scope based off of questions #10 and #11… How has the project's scope changed since last milestone? Did it grow/shrink or stay the same? Why? Justify the change in scope. Do you think your team is on track to complete every feature that is planned? Why or why not? Please go into detail with these questions. (10 pts)

After milestone 0, our team began to focus on two major tasks: detailed concept art and the removal of raylib from the engine along with adding new features.

Our team agreed to postpone the concept art and concentrate more on the engine. While we can dedicate time to completing assets and artistic aspects later, the game development cannot progress if the engine is not finished.

Therefore, we have focused on completely removing raylib from the engine and putting significant effort into adding new functions and features. This work has been quite productive, and we believe our progress has been substantial.

13) Generally speaking, what's the next step for your project? What's the direction that this project is moving and how will the future features' work be split between your members? (5 pts)

As more features are added to the engine, we can now focus on the core systems of the game. By milestone 2 (playtest), we aim to implement the following key features to bring the game to a "playable" state:

- **Map**:

Since we can parse and load SVG files, we can design levels with greater detail, allowing us to implement maps that can be played.

- **Boss Battles**:

We plan to implement at least one boss encounter, where players will face a boss in a confined or specific area, engaging in battle for a set period of time.

- **UI**:

We will display essential player information on-screen, moving away from console outputs and integrating it into the game's interface.

- **Backgrounds**:

We will create multi-parallax backgrounds with several layers, providing visual satisfaction and enhancing the player's immersion in the game.

- **Merchant**

A merchant will be added to enrich the game experience.

14) Please list any issues your team has dealt with during this milestone: (This can be technical issues, personal hardships, team disagreements, etc.) What was the plan to overcome these issues? (5 pts)

- Our game, Manzo, had a lot of major features to be implemented like Rhythm System, SVG file parsing, Collision and game engine, teammates worked individually. In addition, while the engine was incomplete, there were works performed without considering integrating to the game engine. Because of these, We took a long time to deal with fixing points and errors. All teammates decided to do the work based on the engine.

- We've used Eigen Library to write the mathematical expression for Collision Detection. It was fine until we merged them into the game engine. After merging, We got lots of errors. It took so much time to fix it so we decided to remove the Eigen library. After this, we've learned that we should include external libraries carefully.

15) Extra notes for the instructor! (Completely optional!) As a team, you can use this area to tell me anything else you find important! Does your team think you'll have the best game in the class? Why? Does your team deserve extra credit for something? Tell me! Seriously, anything else you want me to know, put it, here! (0 pts)

Between milestone 0 and milestone 1, we replaced the entire raylib engine and additionally implemented convenient functions for SVG parsing, SAT collision, rendering, shaders, and more. Through efficient time management and communication, we monitored the team's progress weekly, or even every few days when necessary, and flexibly adjusted schedules to achieve these results.

Our strength in this approach allowed us to produce noticeably more results compared to other teams, and it suggests that we have the potential to further improve in the upcoming milestones and team projects.