

Aether of Enclaves

Explore the sky.

Samuel Eubanks
McKenzie Weller
v0.0

February 8, 2018

Contents

1	Project Requirements	3
1.1	Requirements	3
1.2	Questions	3
2	Overview	5
2.1	Main Concept	5
2.2	World/Environment	5
2.3	Objects in the Game	5
2.4	Art Style	5
2.5	Controls	5
3	Mechanics	6
3.1	Base	6
3.2	Crew	6
3.3	Crafting	6
3.4	Creatures	6
3.4.1	Main character	6
3.4.2	NPCs	6
3.4.3	Monsters	6
3.4.4	Robots	7
3.5	Combat	7
3.5.1	Ship Combat	7
3.5.2	Player Combat	7
3.6	World	7
3.6.1	Islands	7
3.6.2	Caves	7
3.6.3	Dungeon	7
3.6.4	Settlement	7
3.7	Agriculture	7
3.8	Items	7
4	Technology	8
4.1	Target Systems	8
4.2	Hardware	8
4.3	Development Systems/Tools	8

1 Project Requirements

1.1 Requirements

1. Version Control: GitHub, Travis CI
2. Testing Framework: Stainless
3. Design Patterns:
 - a) Singleton (Logger)
 - b) Flyweight (Graphics classes)
 - c) Command (Input handler)
 - d) Prototype (Enemy spawning - TBD)
 - e) Observer (Achievements - TBD)
4. User Interface Method: GUI

1.2 Questions

1. What is your project?

Aether of Enclaves will be a 32-bit exploration game, in which the user controls a main character and an airship and travels through the sky - picking up crew members, discovering new islands, interacting with NPCs, and exploring.

2. What technologies will you use?

We intend to write the game in [Rust](#) and use a corresponding game engine framework called [Piston](#). The testing framework we will use will be [Stainless](#), which has been developed for Rust. Most, if not all, of the graphics will be produced using [Aesprite](#), a pixel art editor.

As of current, these are the technologies we anticipate using. We are unfamiliar with Rust as a whole, including its frameworks, and will be learning as we go. We both have a mild exposure to using Aesprite, but this program is rather straightforward.

3. What are the essential parts of the project?

Clearly, since we are creating a game, there must be gameplay for the project to be successful. The essentials for the type of game we are trying to produce specifically are as follows: a main character, saved game data, a variety of NPCs and NPC data, a generated environment, and graphics. As a baseline, we will need to implement each of these even in their simplest forms (e.g., in a console with ASCII art). Of course, we would plan to maintain a large library of NPCs and procedurally generate the world.

4. What outside resources will you require?

There is a possibility of the need for outside artwork (if we are unable to complete enough original art) and sound (background music, etc.). As of right now, this is all TBD.

5. Make a proposal for your architecture.

Our back-end will handle the AI, generation of the game world, data storage (this will be saved locally), game mechanics, user input, logging, and interaction within the game world (e.g. with items or NPCs). This will not be written using a specific framework.

Our front-end will be the user interface and game graphics respectively. This will be created using the Piston game engine, which allows incorporation / production of graphics.

The front-end and back-end are in many ways incorporated in the application (i.e. we do not have something like a server / database that would require an RPC). They will not be asynchronous as a result. The back-end will load saved data locally (as mentioned previously) which is savestate functionality included with the game engine.

The design patterns we will have are listed in 1.1.1. Objects will use composition over inheritance. Other objects include: Player, Creature (for NPCs / Enemies), Airship, World, and Game. (I would predict that we'll develop a few more as we move forward). There will additionally be structs for Items and smaller objects.

6. Detailed Plan

2/28 - I would like to have our classes written, but maybe not tested yet (TBD on the Game class, since this will be the last to implement). There is going to be a lot of learning in this timeframe in order to understand Rust.

3/21 - I would like to have testing for each class completed. Additionally, if we could have the beginnings (or more) of the GUI completed...

4/4 - This is right after break. There's only 2 weeks between this due date and the previous due date, for most of which I won't be around. Not sure what should be accomplished here, except cleaning up / completion of the Game functionality. Inclusion of the savestate right away would be beneficial. This would be a great time for you to work on world generation.

4/18 - Let's try to be done with graphics (or have them imported from elsewhere). Having a full GUI by now would be awesome, so we can maybe user test / make further changes for the rest of the semester.

5/2 - Er, done?

2 Overview

2.1 Main Concept

This is an exploration game where you are the captain of a ship and fly from island to island (hopefully procedurally generated) gathering resources and items, explore the world, and manage your resources.

2.2 World/Environment

Steampunk with magical creatures

2.3 Objects in the Game

The goal of the game is to explore, grow your crew, and upgrade your ship.

2.4 Art Style

Pixel art

2.5 Controls

insert controller diagram here

3 Mechanics

The following is not planned to be added in the scope of this class

Player must adapt to environment to be efficient

3.1 Base

Ship that is upgradeable over time.

Ship is "leveled up" rather than main character

conveyor belts / mine tracks can be put around base for a cost to move things around

Made of modular parts that can be replaced / upgraded.

Different parts together create synergies.

Ship has a heart that needs to be defended or ship dies

3.2 Crew

People around the world can be recruited to your team after a certain requirement is met.

Playstyle is determined more by crew than character

3.3 Crafting

Use gathered resources to craft into potions and other items.

3.4 Creatures

There are no words for communication in this game, only icons will be used to express how / what a creature is feeling / thinking.

3.4.1 Main character

Does not have many skills and not good at fighting but is able to recruit NPCs with skills to help

Can command team?

3.4.2 NPCs

Can be recruited to work on ship

3.4.3 Monsters

Creatures are found around the world (both passive and aggressive).

Can capture monsters (enslave / capture bribe / hire breed farm), can also be trained to be stronger / better at what they do?

different types of monsters can do different things (produce crafting ingredient, work, fight)

3.4.4 Robots

Steampunk robots can be created to automate resource gathering, crafting, transportation, and fighting?

3.5 Combat

3.5.1 Ship Combat

Shoot other ships by putting items into cannon

3.5.2 Player Combat

Topdown real time combat

Go into battle screen like ultima but only for boss monsters?

3.6 World

Randomly generated islands with creature settlements generated.

3.6.1 Islands

Some islands are good better for certain things

Some islands are more exploration focused, some are farming focused, some are combat focused

3.6.2 Caves

Use cellular automata to generate

3.6.3 Dungeon

3.6.4 Settlement

3.7 Agriculture

soil quality means that different plants thrive in different conditions

seeds can only be obtained by eating food or converting plants into seeds

3.8 Items

Craft / find items and level them up

4 Technology

4.1 Target Systems

Windows / Mac / Linux

4.2 Hardware

Monitor, Keyboard / Controller

4.3 Development Systems/Tools

Language: Rust

Libraries: Piston

Graphics tools: Asprite