Team Number: 24

Team Members: Spencer Rodenberg, Evan Rogerson, Nevan Snider, Kyle Whitmer and Karsten Wolter

Project Name: Survival Space Game

Synopsis: Our project aims to develop a game where the player pilots a spaceship in search of materials to upgrade your vessel to aid in exploration.

Architecture Description:

Development Setup

Engine Selection

We determined that a game engine would be necessary for this project due to the physics-based movement system that serves as the core gameplay mechanic. We decided on the Godot engine for a few reasons.

1. Team Experience – One of our team members already had pretty extensive experience with the Godot engine and first suggested it.
2. Accessibility and Performance – Godot is a free and lightweight engine with minimal system requirements.
3. Physics Capabilities – Godot would provide the physics systems that are essential to our game for replicating the space-like environment with our momentum-based movement.

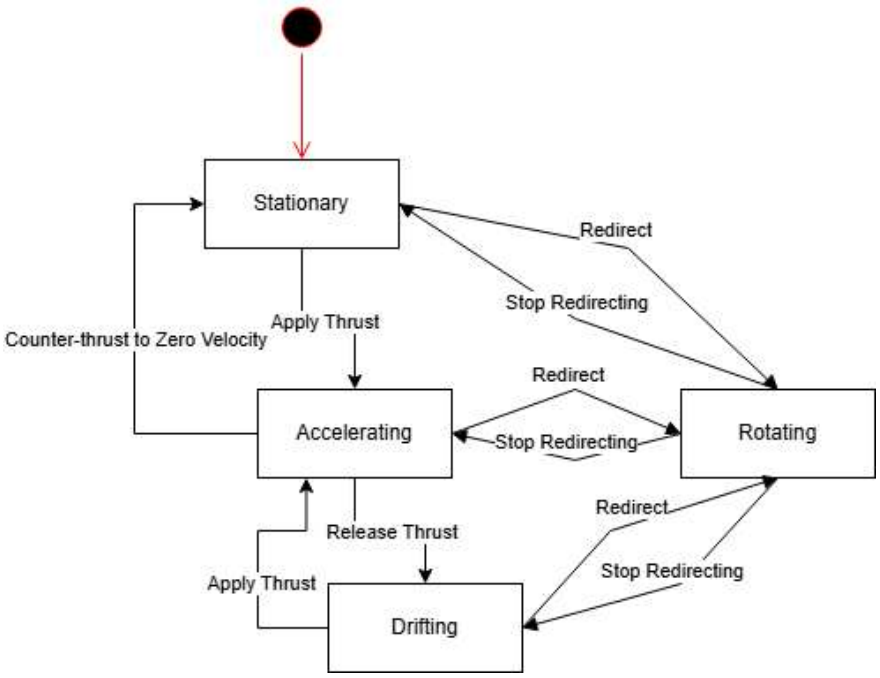
Beginning Game Architecture

Movement System

This game features a physics-based movement system:

- No Atmosphere Environment – No drag or friction allows your ship to continually drift with no interference.
- Momentum Conservation – Any velocity gained is maintained unless counter-thrusted by the spaceship's thruster.
- Thruster-Based Control – The player uses the thruster to accelerate themself forward and additionally has directional controls to redirect themselves.

Figure 1, diagram of the spaceship movement states



Early Gameplay

In the early development stage, the gameplay is a continuous loop and is relatively simple:

- Start – At the start of the game you spawn at your base.
- Exploration – You then begin your journey to find materials in the form of asteroids.
- Gameplay Loop – You repeatedly go on the search for asteroids and return them to your base all while dodging obstacles.
- End – The game ends when you hit an obstacle and your spaceship is destroyed.

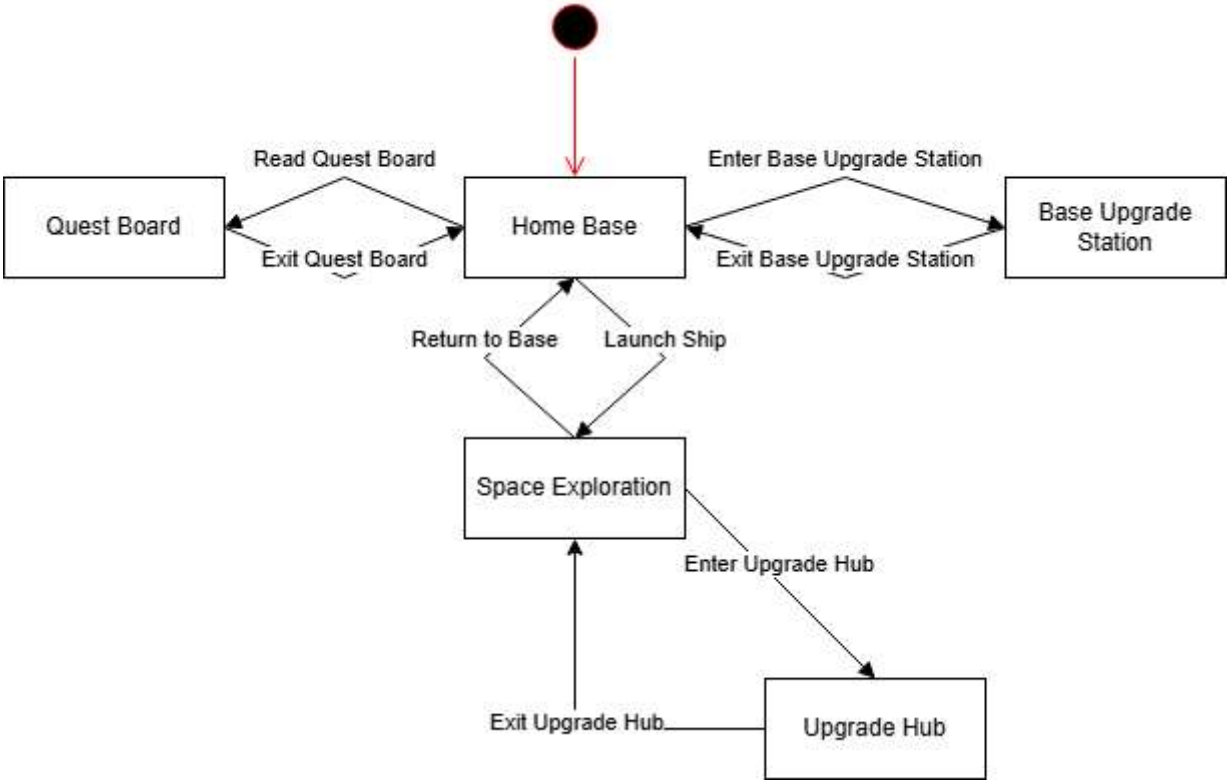
Late Game Architecture

Late Gameplay

Later, in the development of the game, when the fundamentals are established, there will be additional game features that add to the experience of playing the game:

- Mini Map – A small map that will allow you a better view of your surroundings.
- Fuel Depletion – A game mechanic that will limit the extent of your travels.
- Ship Upgrade – A reward system for the materials you're collecting, now they can be used to better your vessel.
- Oxygen Depletion – Another limiting game mechanic, but instead of limiting the range of your expedition, oxygen depletion will limit the duration of your asteroid extraction.
- Quest Log – Introducing new objectives and incentives.
- Save/Load Game – Ability to save game state and load saved state later.

Figure 2, diagram demonstrating potential late game gameplay states



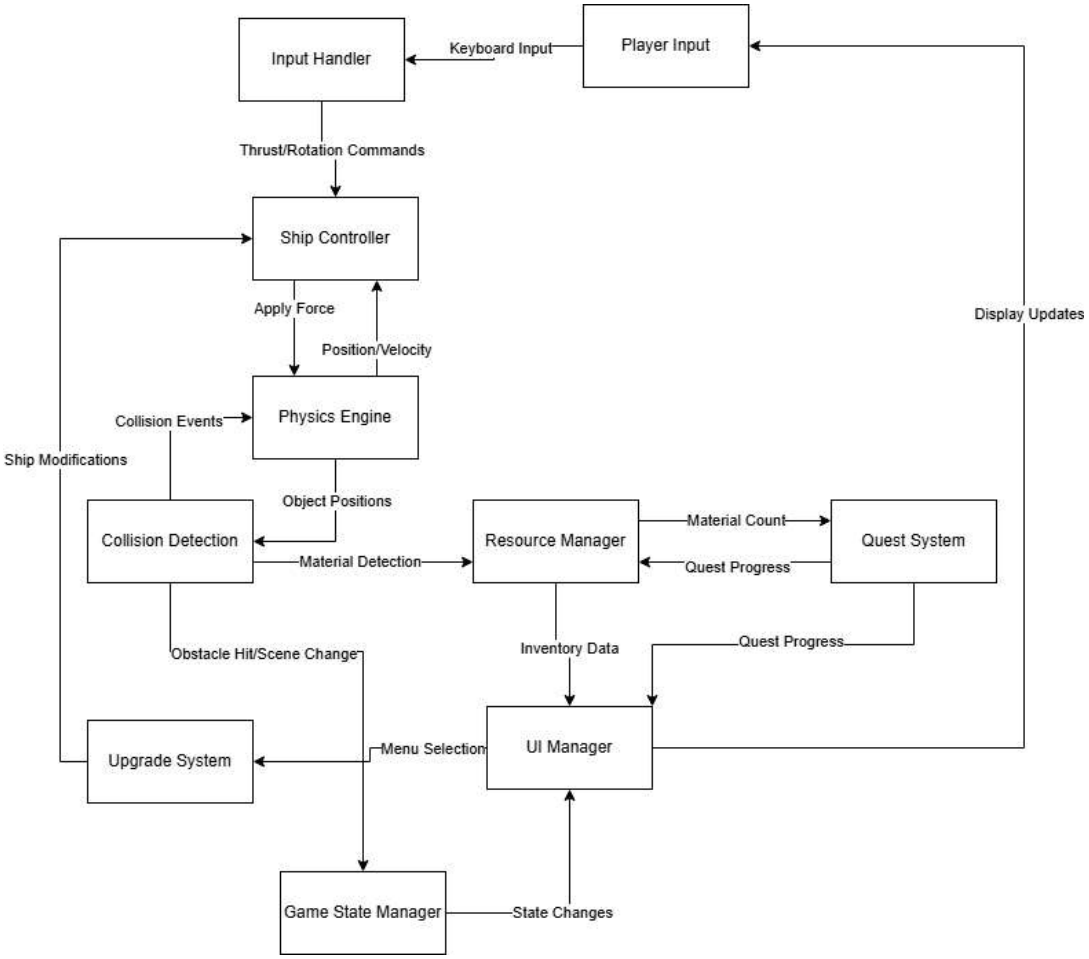
System Architecture

Core Components

The game is built using a modular component-based architecture:

- Input Handler– Handles input processing and ship control
- Ship Controller – Controls the ships movements
- Physics Engine – Manages momentum, collisions, and movement calculations
- Collision Detection – Handles object positions and collision events
- Resource Manager– Tracks collected materials
- Quest System – Manages objectives
- Upgrade System – Handles ship improvements
- UI Manager – Manages the user interface shown to the player
- Game State Manager – Handles the state of the game

Figure 3, shows simplified dataflow of system's architecture



Refining and Final Touches

Visuals

One of the last improvements made to our game will probably be to the visuals of the game, considering that functionality is vastly more important than aesthetics. But if time allowed, it could be worthwhile improving the visual appeal of the game for a more complete look.

Audio

A final change that would serve to give the player a better experience and allow for a more complete product would be the addition of sound design, including sound effects and maybe music.

Refining the Gameplay

Before we can call the game complete, we have a few last changes to make:

- Test and Debugging – Remove any issues in the code and make sure the game works as intended.
- Game Balance – We will probably want to think out what is fair for the player in terms of gameplay.
- Refine Ship Movement – We could possibly want to change how the ship behaves after testing.