

# Game Design

## Type of Game

### Overview

The game will be a vertically scrolling shooter. The player will control a ship that will fly over a scrolling background fighting different enemies.

The background will be an ocean, static enemies will appear in the form of boats occasionally

### Gimmick

- Power distribution system to manage strength of shields, movement, and weapon power.
- If the player increases the strength of the shield, he will have to decrease something else
- This can be done on the fly
- Only a finite amount of power can be distributed to the systems

### Other Features

- Primary and secondary weapon with cool downs
- Shields
- Score system and high score
- Enemy AI
  - Pre planned routes/ formations
  - dynamic spawning of formations
- Different kinds of enemies
- Ground enemies (boats)
- Scrolling background

# Development Design

## Architecture

### Controller Module

- takes input from pc
- hands it off to the game state

### Game State Module

- Acts as the games model.
- takes information from other modules
- modifies the the game state and feeds information back ot modules that need it

### Collision Module

- Handles collisions of projectiles
- Tells the game state when things collide in an event queue

### Enemy Module

- Controls enemies with pathing routes (manually created routes for ships to fly)
- Chooses from a set number of enemy composition and route types

### Player Module

- Handles input from controller module and tells the Game State what to do with the player

### View Module

- Displays sprites to the screen based on information provided from the game state module
- Manages animations

### Sound Module

- Handles the playing of music and sound effects
- Takes input from the game state of when to play music and sounds.

## User Interface

I would like to have input from a game controller, with keyboard as a fallback control method.

### Controller control scheme (WIP)

- Left stick is movement of Player
- A button is fire primary weapon
- B button is fire secondary weapon
- D pad controls distribution of power
- start pauses the game

### Keyboard Control Scheme (WIP)

- Mouse controls player movement
- Left Mouse Button is fire primary weapon
- Right Mouse Button is fire secondary weapon
- WASD controls distribution of power
- ESC pauses the game

## Technical Challenges

1. Enemy AI
  - I think the enemy AI and routes could be difficult to get working in a consistent and dynamic way.
  - I will try to make this part as modular as possible to make building and spawning the enemy formations easier
2. Creating a proper game state
  - Creating enough classes, objects, and variables to manage the game state in way that is not cumbersome will be difficult.
  - I will try to make this part easier by planning out everything each module will need and what the game state needs to directly track.
3. Collisions
  - Given the amount of projectiles that could be on the screen at any given time, having a good collision system could become difficult to polish.
  - I will keep the collision system as functional as possible to make it easy to add hitboxes to different enemies and projectiles.

## Timeline

### Milestone 1 March 15

1. Sprites for player, enemies, projectiles
  - These sprites will serve as the backbone of the game graphics, more graphics will be created, but these are necessary first
2. Game State Module
  - The game state should be in a working state, with stub functions for adding future features
3. Control Module
  - The control module should take pc input and hand that to the Game state in an event queue
4. View Module (sprite displays, not animations)
  - The view module should be able to render a list of sprites to the screen
5. Player movement and shooting
  - The player should be able to move around on screen and shoot projectiles
  - This will most likely not be polished, but it should work

### Milestone 2 March 29

1. Collision Module
  - Create the module to handle collision checking
2. Scrolling background
  - Create a background to scroll through
3. Enemy AI
  - Create AI paths for the enemies to follow
  - Spawn the formations of enemies dynamically
4. Create first playable prototype
  - Have a proper game over screen, pause screen, and restart game
  - Game can be played as is

### Milestone 3 April 12

1. Power system

- Create the power system to modify specs of shields, weapons, and movement speed
- Always a tradeoff

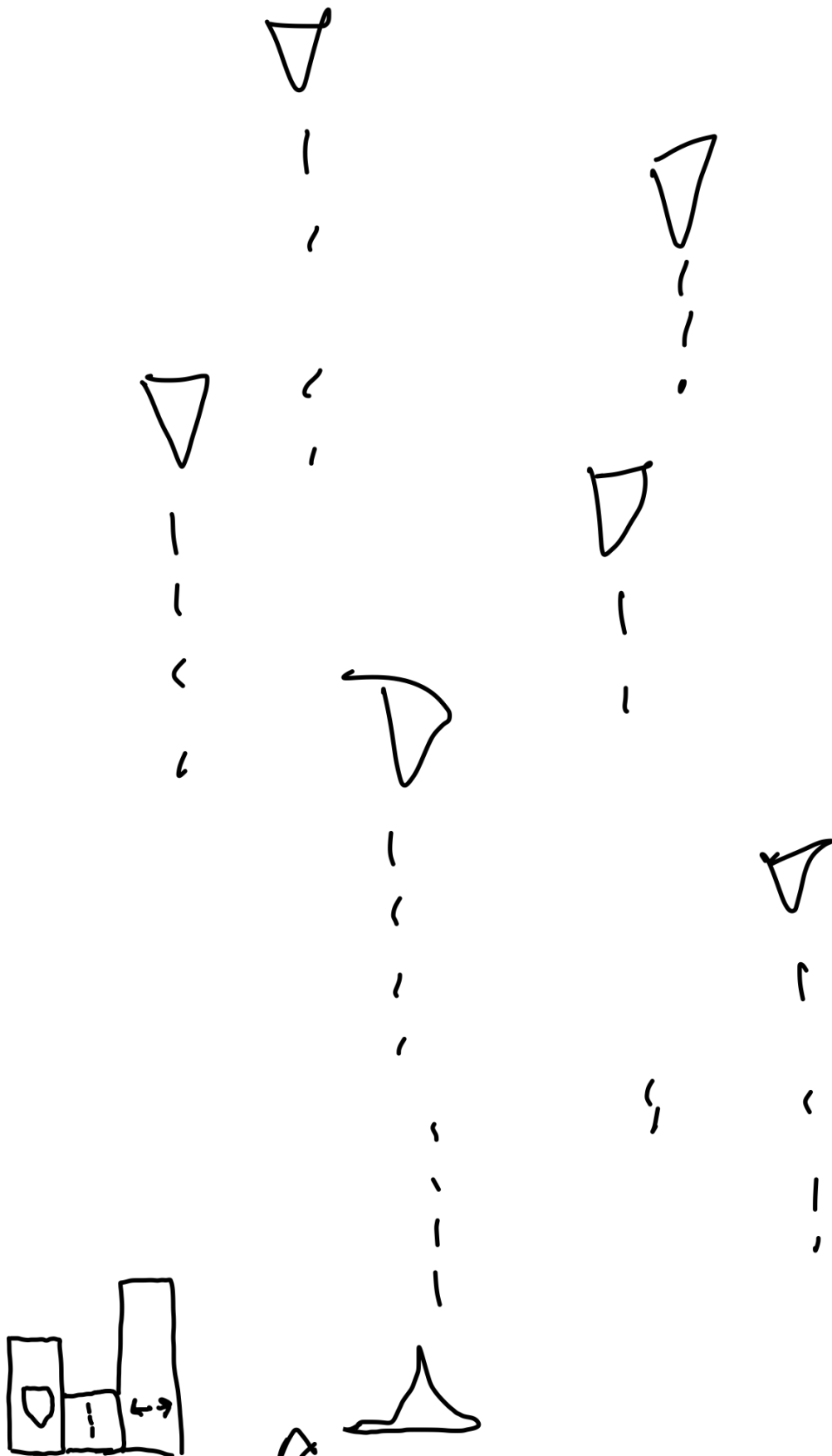
2. Music/ Sound Effects

- Create (or find public domain) music and sound effects and create the requisite module to add them to the game

## **Final Submission April 26**

1. Work on polishing mechanics and fixing bugs
2. Built-in time in case something takes longer

SCORE: 3300



player character