**Menu**

Single Player

New Game / Continue

Save Game (only while playing)

Load Game

Host Multiplayer

New Game (Select Character)

Load Game (Select Save)

Back

Join Multiplayer

Join Game (Select Character if new)

Back

Unlockables (As characters progress, global unlocks can be purchased)

Settings

Wiki (Site)

Quit

**Game Modes**

Story [S, M] (Contains cinematics, dialog, and mechanical unlockables)

Survival [S, M] (Endless waves, cosmetic unlockables)

Team Games [M] (Select type of team game and wait for participants, cosmetic unlockables)

Arena [M] (Free for all, cosmetic unlockables)

**Difficulty**

Easy (Ship can be hit 3x per life and enemies and hazards have less hp or are less capable)

Medium (Standard game, optional tutorial)

Hard (?No tutorial)

Insane (?No tutorial)

**Single Player Game Flow**

Create character.

Race

Magic Types

Difficulty

Very quick intro cinematic leading directly to combat.

Name of the game and panning over and zooming in to a battlezone.

Introduction to simple mechanics with easy combat (only on Easy and Medium).

Protect ship.

Dock.

Mission debriefing

Performance review and status.

Optional quick access to repair bay, research, and library.

Repair / Install / Remove systems, align ship to magic

Learn spells

Interact with crew members for additional dialog.

Other pilots (Story, secret unlockables)

Repair bay (repair, install, remove systems, align ship, unlockables)

Research division (Unlockables)

Library (learn spells)

Marines (Story, secret unlockables)

Other special crew members (Story, secret unlockables)

Launch prep.

Mission briefing, objective, and failure state

Ship and magic overview

Launch

Mission.

Complete objectives and survive

Collect optional collectibles

Optionally protect npcs

Return to Dock and repeat.

Cinematics and story events can be inserted anywhere during missions or outside of them.

**Outfitting Rules and Controls**

Up to four combat spells or systems can be selected; one for each face button.

Ship munitions are fired with Right Bumper and are limited but powerful.

Magic munitions are fired with Right Trigger and are unlimited but weaker.

Two utility spells or systems can be selected for the Left Bumper and Trigger.

Left thumbstick controls acceleration; some additional effects with certain alignments/spells.

Right thumbstick controls direction.

Directional pad is used for communications/commands/specials.

Select allows inventory and system management?

Start pauses game and displays menu.

**Enemies and Hazards**

Hazards

Asteroids and derelict ship fragments

Enemies

Other human pilots

Rorschach aliens

Old ones

AI

Zerg

**Ship Design**

Ships are all single-pilot vessels that have at least one fixed mount.

**Specs**

Base Maneuverability – How many and what kind of maneuvering jets that hook into the engine.

Mounts – How many and what kind of mounts are available.

Integral Systems – How many and what kind of mounted systems are built into this design.

Magical Alignment – The magical school this ship is aligned with.

**Mounts**

Fixed – Supports any ‘fixed’ capable weapons and equipment. Fires in one direction.

Turret – Supports any ‘turret’ capable weapons and equipment. Fires in a range of degrees.

Internal – Supports any ‘internal’ capable weapons and equipment. Cannot fire directly.

**Mount Specs**

Size – Size of weapon or equipment that can be installed.

Accuracy – The spread of weapon or equipment fire.

Base Ammo Capacity – The number of shots available to the weapon or equipment mounted here.

Enchantment Number – The number of enchantment slots available.

Enchantment Potency – The strength of an enchantment slot (per slot).

Magical Alignment – The magical school this mount is aligned with (compounds with ship alignment).

**Mount Size**

Heavy – Larger weapon or equipment.

Medium – Standard weapon or equipment.

Light – Small weapon or equipment.

**Mount Mod Types**

*Each mount supports 1 mod.*

Size+: Increase size (reduces accuracy and ammo capacity)

Size-: Decrease size (increases accuracy and ammo capacity)

Twin: Allow two weapons of same type (decreases size)

Dual-C: Allow two weapons of different types (decrease size and ammo capacity)

Dual-S: Allow two weapons of different types (decrease size and accuracy)

Accuracy+: Increase accuracy (decreases ammo capacity)

Accuracy-: Decrease accuracy (increases ammo capacity)

Ammo+S: Increase ammo capacity (decrease size)

Ammo+A: Increase ammo capacity (decrease accuracy)

AmmoSplit-A: Allow two types of ammo (decrease accuracy)

AmmoSplit-S: Allow two types of ammo (decrease size)

Enchantment Potency+

Enchantment Number+

Enchantment Number-

**Mount Enchantment Types**

Each mount supports up to 3 enchantments.

**Magic Schools**

Void (reality distorting, teleporting, etc…)

Elemental (Maybe break up?)

Life / Death (Maybe break up?)

Arcane / Energy

Time

**Ship Alignment Effects**

Color and some details change

Spell stats favor the aligned school

More powerful spells of the aligned school can be used

One unique affect for the ship per school can be selected.

Examples:

Void – when charged, ship will phase out of reality for 2 seconds upon taking damage.

Elemental – ship fires elemental projectiles every 0.3 seconds in a random trajectory.

Life / Death – biological deaths increase charge rate of all spells.

Arcane / Energy – a torrent of arcane energies constantly flow outward from the ship.

Time – ship can slow time to 0.5x speed for 2 seconds.

Each school will have numerous choices.

Weapon Types

Ballistic

Energy

Missile

Player

The player is called a Trancer.

Suffering destruction of the relay results in “trance-lash”, causing the trancer to lose all his progress for the run.

A player has three attributes: grace, hunger, and wit. These affect stats and game effects.

**Terminology Glossary**

Augury: The ability to read and shape reality through incantation and intense will. Once only used for prediction.

Trancer: The player character. A master of augury that casts his astral presence into a relay.

Relay: The player ship. A vessel through which a trancer may act and cast their magic at a distance.

Trancelash: The effect of losing a relay while tethered to it. Results in severe temporary injury in body and mind.

Grace: A trancer’s natural talent for augury. Governs mana pool and regeneration.

Hunger: A trancer’s drive and force of will. Governs spell strength and casts.

Wit: A trancer’s cleverness and flexibility. Governs number and effect of spells.

The Paths: The augural currents that weave through the void, reminiscent of a tangled mass of conduit. Even the graceless often discover hidden talent at the entrance to these winding corridors, but while many voyage within every year, none have returned from beyond the first juncture. These currents appear to flow in one direction and are entombed by an endless cocoon of mundane and peculiar stone and wreckage.

Juncture: Though The Paths begin as a single corridor, they quickly and frequently split, sometimes creating loops, sometimes never to be rejoined. These splits are referred to as junctures.

Mission Types

Survival

Defense

Exploration

Experimentation

Recovery

Rest

Mission Primary Modifiers

Waves

Waypoints

Boss

Magical Chaos

Mission Modifiers

Magical Saturation

Magical Turbulence

Magical Draught

**Where to pick back up**

Hazard collision should be toggleable.

Collisions should be bouncy.

Level generation is rough right now.

GameData stores the playerPrefab.

MissionGenerationData is used to store the generation parameters for a mission.

The LevelManager serves as the source of truth for the finalized level, after generation.

The MissionManager serves as a persistent mission tracker that allows selecting a mission from a list, and it currently keeps track of the mission, doing things like spawning the player in and activating the spawnzones.

The LevelContentPicker uses the MissionGenerationData that it receives from the MissionManager to stay within the parameters and pick mission generation settings. Selected parameters are stored in the LevelManager.

The LevelGenerator uses some of the MissionGenerationData that it receives from the MissionManager and the picked settings that are stored in the LevelManager to actually spawn in the features of the level.

After everything in the level is picked and then generated, the player is spawned in and the spawn zones are enabled.

This whole thing works, but should be refactored so that the functionality is more cleanly encapsulated. A big part of the issue is the naming scheme. Maybe something like this:

MissionGenerationData – This is just a packet of data that guides what parameters must, can, or cannot be selected for this mission.

MissionContentSelector (LevelContentPicker) – This class uses the MissionGenerationData to configure the MissionData.

MissionData (LevelManager) – This is the packet of data that has been selected to guide the generation of this mission.

MissionGenerator (LevelGenerator) – This class uses the MissionData to generate the contents of the mission.

MissionManager – The big persistent class that tracks everything mission related throughout the game, but only does top level stuff.

Mission – The class in charge of a specific mission’s progress, events, and success/failure.

ActorManager – A mission-specific class in charge of tracking all actors and providing convenient access to them.

ProjectileManager – A mission-specific class in charge of tracking all projectiles and providing convenient access to them.

After refactoring the generation functionality and creating a flowchart, you should allow for weights to be used when specifying MissionGenerationData and MissionData. Don’t use dictionaries.

Obstacle wrapping needs to have a toggle to allow for mirrored wrapping, where the two obstacles are symmetrical and the warp in location can be calculated correctly so that it appears that the actors are wrapping seamlessly around the screen.

After you get all of this done, add projectiles so that actors can be destroyed.