

[APCS Capstone Project - Aerius Rift]

Authors: Alan Ma, Jerry Yang, Leon Wang

Last Revised: 5/23/22

Introduction:

We created this project as a tribute to Nova Drift, a 2D space shooter game developed by Chimera Entertainment. As for how accurate this attempt at a replica is, probably ***not at all***, considering how none of us have ever played the game in question.

Take the disparities as unique features, applications of the creative license, etc.

Gameplay:

Control a spaceship and shoot down waves of enemies that scale progressively over time. Experiment with different combinations of weapons, ship types, and shields. How far can you go?

Instructions:

Move the mouse around to aim your spaceship. Right-click to move, left-click to shoot. Press space to select weapons. You can only select weapons once every game.

That's it!

Features:

Baseline

☐ Fully Functional UI

This entails screens, buttons, images, etc. A list of such includes:

- The menu screen, containing a “play game” button, which would take you to the gameplay screen
- The gameplay screen, which would render enemies, bullets, and ships, and register user inputs to make the game playable
- The upgrade screen, containing a small selection types of shields, weapons, and hulls to change to
- The game over screen, which would contain buttons to go back to the main menu, and also display a message along the lines of “game over”

☐ Operational Gameplay

With things including:

- Bullet physics, with bullets dealing damage to enemy ships in its hitbox

- Ship physics, with ships that collide with each other, dealing ram damage
 - The player ship, moving and shooting based on user input
 - Enemies, which attack the player, each having health and damage scaling with waves
 - Enemy waves, spawning new enemies when all the ones on-screen are defeated
 - World boundaries warp the player around the edges, but despawn enemies and bullets.
- In other words, the edges of the screen

Goal

☐ Multiple Enemies

Each wave would spawn different types, providing variety to the gameplay. Some types of enemies may include:

- Sniping enemies, which kite and shoot the player
- Ramming enemies, which rely on ramming and knockback to damage the player
- Bombing enemies, which run across the screen dropping bombs

Above are just basic ideas for enemies, and are yet to be fully fleshed out.

☐ Multiple Weapons/Shields/Hulls

In other words, parts of the player that affect how the player receives damage, shoots, moves, and plays the game in general. A list includes:

- [Weapons](#), which the player can equip (one at a time) to change the shooting style and bullets the player creates. Examples include:
 - A railgun weapon that shoots large, damaging, piercing bullets
 - A laser weapon that creates an instant laser, piercing enemies
 - A shotgun weapon which shreds enemies next to it
- [Shields](#), which the player can equip (one at a time) to change what happens when the ship takes damage. Examples include:
 - A reflecting shield that reflects bullets around it whenever it gets hit
 - A shockwave shield which deals damage and clears bullets through releasing a shockwave when it's broken
 - A temporal shield that slows down time whenever you take damage
- [Hulls](#), which the player can equip (one at a time), contain different base stats, such as acceleration, reload, damage, and speed. Examples include:
 - A basic hull with no unique stats
 - A shotgun hull with faster acceleration/speed, more projectiles but also less projectile speed and reload speed
 - A tank hull with more health, more overall damage, but also slower speed
 - A sniping hull with greater overall damage and projectile speed, but also slower reload and less health
 - A ramming hull with more health, ram damage, and speed, but less bullet damage and reload speed

☐ Nice Assets/Graphics

Something that isn't just the squares we have right now in processing. Decent sprites and shaders would be nice, but those are stretch goals.

Stretch

☐ Bosses

These would appear on certain waves (25, 50, 100?) instead of the customarily spawned enemies, with unique attack patterns and weapons and a larger XP reward. Don't forget – they would also be larger than regular enemies.

☐ Multiplayer

Allow the player to play with others across the web. Yeah, I don't see this one happening.

☐ "Special" Modifiers

Special super-rare modifiers that completely overhaul some parts of the player. Think: turn the player into a chain of multiple ships, or make its bullets 10x bigger and suck in enemies.

☐ Music/SFX

We have never played with java and sound before.

☐ LeGeNdArY GrApHiCs

See *"Nice Assets/Graphics" in the Goals section.*

Class List

Content:

- Enemies: Stores enemy types. Implements ContentList
- Hulls: Stores types of hulls. Implements ContentList
- Shields: Stores types of shields. Implements ContentList
- StatusEffects: Stores types of status effects. Implements ContentList
- Weapons: Stores types of weapons. Implements ContentList

Core:

- Canvas: Extends PApplet. This is the drawing surface
- Content: Loads everything in the content folder
 - ContentList: Represents a list of content
 - ContentType: Represents a type of content
- Events: Processes and stores events, which, when called, update a part of the game
- Input: Handles all inputs by the user.
- Rules: Contains overall multipliers for each team, after the player modifiers are processed.
- Sounds: Stores all the sounds in the game.
- UI: Stores all the UI components
- World: Stores entity lists, the player, and initializes the game

Game:

- Entities: Stores a list of entities, along with a quadtree, if specified.
- Experience: Represents an orb of experience.
- Player: Represents the player ship
- Team: Represents a team in the game. All enemies, bullets, and players have a team.
- Waves: Stores and simulates all waves in the game.

Graphics:

- Effects: Contains everything required to create, process, and draw effects in the game
- Pal: A palette, containing different colors
- Sprite: Represents a sprite

UI:

- Table: Represents a piece of UI. Has an x, y, width, height, and color
 - List: Represents a list of items.
 - ProgressBar: Represents a progress bar.
 - SegmentedBar: Represents a segmented bar.
 - SmoothBar: Represents a smooth bar.
 - Button: Represents a button.
 - Slider: Represents a slider
 - Tooltip: Represents a tooltip
 - Text: Represents a line of text
 - Hexagon: Represents a hexagon. Used for formatting.
 - Drawable: Represents any piece of UI that supports a custom renderer.
- Screen: Represents a screen. Only one can be up at a time, and is rendered and updated in Canvas
 - GameScreen: Updates and draws the game screen. Extends screen.
 - MenuScreen: Updates and draws the menu screen. Extends screen.
 - UpgradeScreen: Updates and draws the upgrade screen. Extends pause screen.
 - LoseScreen: Updates and draws the game over screen. Extends screen.
 - PauseScreen: Updates and draws a pause screen. Extends screen.

World:

- Bullet: Extends Type. Contains data for a bullet type
 - LanceBullet: Extends Bullet. Contains stats for a blowtorch lance bullet
 - FlakBullet: Extends Bullet. Contains stats for a flak bullet
 - MissileBullet: Extends Bullet. Represents a missile bullet
 - RailgunBullet: Extends Bullet. Contains stats for a railgun bullet
 - VolleyBullet: Extends Bullet. Represents a bullet in a volley
 - GrenadeBullet: Represents a grenade bullet
- Enemy: Extends Type. Contains data for an enemy type
 - SprayerEnemy: An enemy which rotates and spams bullets around itself.
 - BomberEnemy: Stores stats for a carpet bombing enemy.
 - MultiEnemy: Stores stats for an enemy with multiple parts.
 - DroneOrbitEnemy: Stores stats for an enemy with drones orbiting around it.
 - RammingEnemy: Stores stats for a ramming enemy.
 - EnemyPart: Stores stats for a part of an enemy.
 - SniperEnemy: Stores stats for a sniping enemy.
- EnemyVersions: Stores data for versions of an enemy
- Modifier: Extends Type. Contains data for a modifier
- Ship: Extends Entity. Simulates a basic ship
 - Hull: Extends Type. Contains data for a hull type
 - MultiBarrelHull: Extends Hull. Represents a ship with multiple barrels.
 - Weapon: Extends Type. Contains data for a weapon type
 - LanceWeapon: Stores stats for a blowtorch lance weapon
 - RailgunWeapon: Stores stats for a railgun weapon
 - SalvoWeapon: Stores stats for a missile salvo weapon

- ShotgunWeapon: Stores stats for a weapon that shoots multiple bullets
- VolleyWeapon: Stores stats for a volley weapon
- Weapon: Stores stats for a weapon
- Shield: Extends Type. Contains data for a shield type
 - CooldownShield: A shield with a cooldown
 - ReflectShield: A shield that reflects bullets
 - ShockwaveShield: A shield that creates shockwaves
 - TemporalShield: A shield that slows down time
 - WarpShield: A shield that teleports the player
- Thruster: Represents a thruster on a ship. For visuals only.
- Entity: Extends Instance. Has a position and velocity.
- Instance: Contains a type, which changes how the entity interacts with the world
- Type: Represents a type of entity. Contains stats and behavior

Main:

- Main: Contains the main method. Starts the program
- Tmp: Contains temporary variables as fields
- Vars: Stores static constants as fields that control gameplay

Credits

- Alan Ma: Content, Bullets, Enemies, Sprites
- Jerry Yang: Buttons, Bars, Screens, UML, Sound
- Leon Wang: Physics, Entities, Core