

APCS Final Project - Nova Subrift

Authors: Alan Ma, Jerry Yang, Leon Wang

Last Revised: *Check last commit date*

Introduction:

This is a project created as a sub-replica of the game 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. Upgrade your ship with modifiers that allow you to scale with the enemies.

How far can you go?

Instructions

Move the mouse around to aim your spaceship. Right click to move, left click to shoot.

That's it!

Features

Baseline

- Fully Functional UI
- This entails screens, buttons, images, and etc., each which function accordingly. 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 of modifiers to choose from/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 it’s 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 all the ones on screen are defeated
 - Ship modifiers, with some that increase attack speed, bullet damage, ram damage, etc. when chosen and applied in the upgrade screen
 - A leveling system, in which players get xp from defeating enemies, allowing the player to level up and get upgrades from the upgrade screen
 - World boundaries, which 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
 - Teleporting enemies, which dodge bullets and damage the player with bullets
 - Above are just basic ideas for enemies, and are yet to be fully fleshed out.
- Multiple Weapons/Shields/Hulls/Modifiers (Maybe stretch goals?)
- 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 thruster weapon that propels the player forward (ram damage) and also drops mines that collide with enemies, dealing damage
 - A minigun weapon that spams low damage bullets at enemies
 - A railgun weapon that shoots large, damaging, piercing bullets
 - A laser weapon that creates an instant laser, piercing enemies
 - A drone weapon which creates mini-ships to damage enemies
 - Shields, which the player can equip (one at a time) to change what happens when the ship takes damage. Examples include:
 - A warping shield that teleports the player whenever it gets hit
 - A radial shield that burns enemies around it
 - A reflecting shield which reflects bullets around it whenever it gets hit
 - A orbiting shield which creates orbiting orbs around it, which block bullets and deal damage
 - A cloaking shield which hides the player when it is not moving
 - Hulls, which the player can equip (one at a time), each containing different base stats, such as speed, reload, and damage. Examples include:
 - A basic hull with no special 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 higher 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
 - Modifiers, which the player can collect, each modifying ship stats in a different way. Some can be taken more than once, and some are rarer than others. Examples include:
 - Overdrive, a common modifier that gives +10% reload speed but -5% projectile damage
 - Piercing Hull, an uncommon modifier that gives +50% ram damage, but -10% projectile damage
 - Compressed Munitions, an uncommon modifier that gives +20% splash damage radius and +5% splash damage, but -15% projectile speed
 - Tailgate Unloading, a rare modifier that makes the ship shoot backwards, but gives +40% overall damage and +10% projectile speed
- Bosses
- These would appear on certain waves (25, 50, 100?) instead of the normally spawned enemies, with special attack patterns and weapons, and a larger xp reward as well. Don't forget – they would also be of a bigger size than normal enemies.
- 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 definitely stretch goals. Enough said.
- Background Events
- Since Nova Drift is a game that takes place in space, you should see spacelike objects that interact with the player. Give them something to think about other than enemies, that's all.
 - Asteroids, which float around the screen and deal damage in collisions. Destroyable, whether by ramming, shooting, or etc.
 - Shooting stars, which would quickly move across the screen, damaging anything in their path. Maybe a warning before they appear, so the player can dodge.
 - Black holes, which suck everything in around them. Instakills anything inside.
 - Planets. Dunno about this one. Is it just a bigger asteroid with gravity?

Stretch

- Multiplayer (*Jerry???*)
- 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
- I have never played with java and sound before.
- LeGeNdArY GrApHiCs
- See “Nice Assets/Graphics” in the Goals section

Class List

- Content:
 - Bullets: Stores bullet types
 - Enemies: Stores enemy types
 - Gear: Stores weapon/shield/hull types
 - Modifiers: Stores modifier types
- Core:
 - Canvas: Extends PApplet. This is the drawing surface
 - Content: Loads everything in the content folder
 - Entities: A list of entities. Provides helper methods to managing them
 - Events: Processes and stores events, which, when called, update a part of the game
 - Player: Extends Ship. Updates, draws, and stores player data. Also processes inputs
 - Rules: Contains overall multipliers for each team, after the player modifiers are processed.
 - Team: Stores the different types of teams
 - UI: Stores all the UI components
 - Waves: Stores and simulates waves
 - World: Stores entity lists, the player, and initializes the game
- Graphics:
 - Effects: Contains everything required to create, process, and draw effects in the game
 - Pal: A palette, containing different colors
- UI:
 - Table: Represents a piece of UI. Has an x, y, width, height, and color
 - Bar: Extends TRepresents a progress bar
 - Button: Represents a button
 - Text: Represents a line of text
 - Icon: Represents a icon
- Screens:
 - GameScreen: Updates and draws the game screen
 - MenuScreen: Updates and draws the menu screen
 - UpgradeScreen: Updates and draws the upgrade screen
 - LoseScreen: Updates and draws the game over screen
- World:
 - Bullet: Extends Type. Contains data for a bullet type
 - Enemy: Extends Type. Contains data for an enemy type
 - Modifier: Extends Type. Contains data for a modifier
 - Ship: Extends Entity. Simulates a basic ship
 - Hull: Extends Type. Contains data for a hull type
 - Weapon: Extends Type. Contains data for a weapon type
 - Shield: Extends Type. Contains data for a shield type
 - Entity: Contains a Type, which changes how the entity interacts with the world
 - Type: Represents a type of entity. Contains stats and behavior

Credits

Alan Ma: Skeleton Classes

Jerry Yang: UI, Javadocs

Leon Wang: Javadocs, UML