Zero-G Delivery

A Latitude Studios Production

Prepared By Drew Kowal

Designed for Windows

Version: 1.0.0.0

Table of contents

1) Introduction

1.1 Key Points:

Genre: Physics Simulation/ Delivery Challenge
Platform: PC
Engine: Unreal Engine 5
Playtime: Open Session ~ 5 minutes total
Perspective: Third-person (over-should/ dynamic top-down)

1.2 Game Overview:

Zero-G Delivery is a short physics based game where players pilot a delivery drone through a low gravity industrial environment. The player locates and transports shipping containers to designated delivery pads using thrust based motion and a gravity manipulator. The goal is to complete deliveries efficiently while avoiding collisions and maintaining cargo integrity. Scores are based on time, precision, and cargo condition.

2) Core Gameplay Loop:

1. Locate a container anywhere on the open map
2. Use thrusters to navigate while avoiding debris and obstacles
3. Use gravity manipulator to lift the container into a locked position
4. Fly to the containers matching delivery pad
5. Lower the container to the ground within the specified location
6. Score based on delivery time and pristine condition, reduced score for damage
7. Repeat for remaining containers
8. End game when no containers remain, this displays final score and previous round score.

3. Game Mechanics:

3.1 Drone (Player)

- Third- person camera angle with a dynamic change to top down when over cargo locations
- Movement applies physics forces (forward, backward, strafe)
- Mouse controls Yaw
- Camera is locked behind the drone and will rotate with it
- Controls:
    - Mouse to rotate on Yaw

- WASD to move forward/ backward/ strafe
- Left click to pick up cargo
- Right click to lower cargo
- Q to pause

3.2 Cargo Containers

- Type: Physics actors with unique Ids linked to delivery locations
- Carried Via gravity manipulator, attaches to drone and remains in place unless collision occurs or the player specifically lowers it
- Health and base score start at 100, Damage reduces base score by 1 per point of damage.

3.3 Game Rules

- Open world allowing players to deliver cargo in any order
- Collisions cause damage to the cargo and can knock containers free from the drone
- Completed deliveries earn a base score with bonuses from pristine and prompt delivery times
- Score penalties are applied based on damage -1 pt score/ pt of damage to containers.
- Top score persists between play through

4. UI Design:

4.1 Main Menu (UMG Widget: W_MainMenu)

- Title Text: “Zero-G Delivery”
- Play button: Loads game play level
- Quit button: Exits the game

4.2 In-Game HUD (UMG Widget: W_HUD)

- Timer: Top center, counts down from 60 (Will be adjusted based on game play)
- Score: Top left, shows current score
- Cargo integrity meter: displays at the screen bottom or on the containers
- Target pad indicator: basic flat pointer arrow that rotates around the drone showing where to deliver the cargo

4.3 Pause Menu (UMG Widget: W_PauseMenu)

- Resume Button: Exits the pause menu
- Quit to menu Button: Quits the current play through to main menu
- Quit Game Button: Quits the Game to desktop

4.4 Game Over screen (UMG Widget: W_GameOver)

- Current score text: Displays “Targets hit: X”
- High Score text: Displays “Best Score: Y”
- Play Again button: reloads gameplay level
- Quit button: Exits the game

5. Complex System: Score & Time Tracking:

5.1 C++ Components:

- ScoreManager: Handles delivery point, bonuses, and persistent top score
- TimerManager: Tracks run time per delivery
- HUDManager: Updates score/time/cargo integrity UI/delivery indicator
- ContainerHealth: Manages the cargo health and damage handling (Based on impact velocity?)

5.2 Scoring:

- Base delivery: +100
- Container condition: +50 for pristine, -1/ point of damage up to a max of 100 (At 100 the container will be destroyed)
- Time bonus: time bonus will start at 100 but after a period of time it will begin being reduced based on additional time required for delivery.

6. Week Development road map:

6.1 Week 1: Planning & Setup

- Install and configure UE5
- Set up project structure adn version control
- Import FPS template and verify working controls

6.2 Week 2: Basic Gameplay Prototype

- Create and test with a ground plane and basic lighting
- Implement basic Player Controller
- Create simple BP_Target actor and enable destruction via ray casting

6.3 Week 3: Timer + Score system

- Add 30s countdown timer via GameMode
- Add score counter that tracks hits
- Create HUD display for timer and score

6.4 Week 4: UI Phase 1 – Main Menu

- Design and implement W_MainMenu
- Add functionality to start game and quit game buttons
- Route UI navigation correctly to gameplay level

6.5 Week 5: UI Phase 2 – Game Over Screen

- Create W_GameOver widget
- Show score vs best score
- Add functionality to restart or quit

6.6 Week 6: SaveGame System

- Implement BP_SaveData
- Save and load best score using SaveGame object
- Integrate SaveGame with GameMode and GameOver UI

6.7 Week 7: Polish and Feedback

- Add simple target particle FX and reticle flash upon destroy
- Add end of game effects. Maybe screen fade...

6.8 Week 8: Testing and Optimization

- Playtest for bugs, timing issues, score tracking
- Optimize target spawning logic
- Ensure smooth transitions between game states

6.9 Week 9: Final QA and Submission Prep

- Final Polish pass on visuals and UI
- Package and test build for PC
- Submit playable demo and documentation