Sweet Exodus - Design Statement

Applicant Name: Sweet Exodus
Role: Solo Developer (System Design, Programming, Gameplay)
Genre: 2D Precision Puzzle-Platformer

Project Summary

Sweet Exodus is a comprehensive solo remake of a previous Game Jam prototype, representing a personal exploration into constraint-based mechanics and the intersection of platforming precision with rhythmic flow. Moving away from traditional input mapping, the game features a unique **"Cyclic Action System,"** where distinct abilities (Jump, Dash, Float) are bound to a single execution key. Players cannot access these abilities simultaneously; instead, they must strategically toggle between states to navigate the environment. This project served as a technical and design playground to experiment with restrictive control schemes and modular system architecture.

Design Intentions

My primary goal was to challenge the player's muscle memory by replacing standard platformer reflexes with a system that demands planning and rhythm.

- **Constraint as Strategy:** By designing a non-traditional control scheme where players lack simultaneous access to movement abilities, I wanted to shift the focus from reaction speed to **sequence execution**. Jump, Dash, and Float are mutually exclusive states; success depends on managing the "cooldown" and flow of switching these states rather than just pressing a button at the right time.
- **Mechanic-Driven Level Design:** I constructed puzzle-platforming levels specifically tailored to this switching mechanic. The environment acts as a rhythmic sheet music where players must plan their input sequence (Rhythm & Flow), requiring a cognitive engagement that goes beyond simple traversal.

Personal Contribution

This is a solo project. I was responsible for all code architecture, gameplay mechanics, and AI behavior design.

- **The "Cyclic" Framework:** I engineered the core state-switching logic that governs the player's interaction. This involved creating a responsive input buffer system to ensure that switching between Jump, Dash, and Float felt fair and precise, even under the pressure of tight platforming sections.
- **Modular Level Management:** To handle complex level layouts without performance overhead, I implemented a **Prefab-based level streaming system**. This architecture ensures efficient scene management and seamless transitions

between different puzzle sectors, allowing for expansive level design without loading screens.

- **FSM AI Behavior:** I developed complex Boss AI using **Finite State Machines (FSM)**. This structure allows for dynamic phase transitions and predictable yet challenging attack patterns, ensuring that boss encounters test the player's mastery of the Cyclic mechanics rather than random chance.

- *Note on Visuals:* Some visual assets were generated using AI tools to allow me to focus entirely on system architecture and gameplay logic verification.

What I Learned

Through the development of *Sweet Exodus*, I gained a deeper understanding of how constraints can drive creativity in game design.

- **Balancing Restriction and Flow:** I learned that when you take away standard controls (like a dedicated jump button), the level design must be incredibly communicative. I learned to use visual cues and level geometry to "teach" the player the necessary rhythm for each section.
- **System Architecture:** Implementing the Prefab streaming system and FSMs taught me the importance of modularity. By keeping the level data and AI logic separate from the core player controller, I was able to iterate on level design rapidly without breaking the underlying game mechanics.

Access

- **Downloadable File Link:** Sweet Exodus by NekokoP

Installation & Interaction Instructions

1. **Platform:** Windows 10/11.
2. **Installation:** Download the zip file from the link provided, unzip, and run .exe file. No additional installation required.
3. **Controls:**
    - **A&D key:** Move left and right
    - **K key:** Use ability
    - **L key:** Switch ability