Development Process & Performance Assessment

System Overview

The system consists of a third-person action-adventure game built in Unity with the following core features:

- Player Movement System:

Built upon Unity Starter Assets character controller with custom modifications for dynamic stat changes (speed, jump force, gravity). While using this external framework, I successfully debugged and resolved grounding detection issues and implemented custom jump behavior.

- Inventory System:

Grid-based with drag-and-drop functionality, item stacking, and consumable/equippable item support

- Dialogue System:

NPC interaction system with camera focusing using Cinemachine, featuring animated transitions and conversation management

- Item Effects:

Comprehensive system supporting health/stamina restoration, temporary stat boosts, scale changes, and gravity manipulation

- World Interaction:

Range-based detection for interactable objects and NPCs

- UI Systems:

Pause menu with time-scale management and inventory interface with tooltips

- Editor Tools:

Custom Item editor for efficient content creation and edition for designers.

The project incorporates free props and UI asset packs from the Unity Asset Store, with characters and animations sourced from Mixamo.

Thought Process

I approached the challenge systematically, prioritizing core functionality first: player movement, interaction detection, and inventory management. I used Unity Starter Assets for the character controller, a complex feature that could have taken weeks to develop from scratch, allowing me to focus on game-specific systems. Despite using this framework, I successfully debugged grounding issues and customized the controller to work with my dynamic stats system.

I focused on creating modular, extensible systems that could accommodate future features. The dialogue system was simplified to maintain functionality while ensuring stability.

Performance Assessment

I must acknowledge that I received this task without prior notice of the 48-hour deadline, and I discovered it later than expected. Despite my packed schedule, I'm quite proud of my performance under these constraints.

The system is mostly bug-free with functional core mechanics. While I missed some features due to time limitations, I successfully implemented: smooth player controls, working inventory with item effects, NPC dialogue with camera transitions, interaction systems, and custom editor tools for content creation.

The codebase demonstrates clean architecture with proper separation of concerns, extensive use of Unity's modern features (Input System, Cinemachine, URP), and attention to user experience details like visual feedback. My ability to integrate, debug, and extend the Unity character controller shows practical problem-solving skills.

Given more time, I would have refined the camera transitions further, added more visual polish, VFX, SFX, and implemented additional gameplay features like developing a more fleshed out sandbox map. Nevertheless, the foundation is solid and extensible for future development.

Credits: Developed by Santiago Barra