Unity Task - Carlos Iván Córdoba Quintana

System Explanation

The system implements player interaction with collectible items, movable objects, and switches, supported by an inventory and drag-and-drop UI. The ***Item*** class detects collisions with the player, adds the item to the global inventory, updates the UI via ***InventoryUI***, and uses ***DialogueManager*** to provide feedback. The ***Inventory*** singleton manages collected items and stores a reference to a ***Switchable*** when the player is in range. Using an item on a switch triggers ***Activate***, which changes object layers and colors to open progression paths. ***Movable*** objects adjust their colliders dynamically, allowing certain boxes to pass through specific barriers. The UI drag-and-drop system lets players rearrange inventory slots and swap items without losing them.

Thought Process During the Test

Due to a personal situation, I couldn't work on the project during the weekend after receiving the task. I communicated this, and the team kindly extended the deadline. This meant I had to focus on getting the most important systems operational within the remaining time. My approach was to first build the inventory as the core link between items, switches, and UI. I then implemented basic player interaction with switches and movable objects to establish puzzle mechanics. Finally, I added the drag-and-drop functionality to make inventory management interactive. The priority was ensuring that the main systems communicated correctly, even if the full gameplay loop wasn't complete.

Personal Assessment

I was able to implement several interconnected systems that demonstrate the intended mechanics, but the project is incomplete — there's no save system, no end condition, and the gameplay loop isn't fully closed. The code works but would benefit from significant refactoring using SOLID principles and an event-driven approach to decouple systems. This would make it cleaner, easier to extend, and better suited for long-term development. While not feature-complete, the work shows my ability to quickly set up functional systems under time constraints and identify where architectural improvements are needed.