Lautaro Ferreira – Unity Programmer Task

For this technical test, my main goal was to design and implement a small but solid gameplay slice that reflects production-ready patterns rather than a quick prototype. I focused on clarity, scalability, and separation of responsibilities, prioritizing systems that could naturally grow in a larger project.

The core of the implementation is a slot-based inventory system that supports stacking rules, item usage, equipment handling, drag/reorder logic, and persistence. Items are represented through data objects, while the inventory logic is centralized in a single controller that exposes clear events for UI and save systems. This allowed me to keep the UI reactive and decoupled from gameplay logic. Interactions such as picking up items, equipping weapons, or consuming consumables are handled through explicit player actions, avoiding hidden side effects or UI-driven logic.

During development, I deliberately chose an event-driven approach for saving. Instead of saving only on UI events, the game automatically persists data whenever the inventory state changes, ensuring reliability both in editor and standalone builds. This mirrors how I would approach autosave systems in a real production environment.

From a gameplay and presentation standpoint, the character uses directional animations driven by movement vectors, with support for switching animation sets when a weapon is equipped. While the current implementation swaps animator controllers directly, it was designed with extensibility in mind so that additional weapons or animation profiles could be added with minimal refactoring.

There are several areas I would expand with more time. I could have integrated Cinemachine for more polished camera behavior, implemented full localization support for all dialogue and UI text, and added more visual feedback such as UI animations, particles or sound cues. Additionally, there are a few minor bugs and edge cases that could be easily addressed with additional iteration time, mainly related to UI interactions and polish rather than core system design.

Overall, I am satisfied with my performance. The final result demonstrates how I think about systems, trade-offs, and maintainability, and reflects how I would approach similar tasks in a professional indie or AA production setting.