UNITY PROGRAMMER TASK - MICHAEL GARCIA

Personal assessment

I began by brainstorming key features before writing any code, focusing on simple and clear game mechanics. The result is a 3D game prototype with an inventory and interaction system. Players can move left or right, interact with items (coins and a pistol), and talk to an NPC. Throughout development, I prioritized creating modular, extensible architecture while maintaining simple, readable code within the time constraints.

I implemented core gameplay using Unity's new Input System for player movement and interaction. The interaction system detects nearby interactive objects using an overlap sphere. When players press the interact button, the system triggers an animation and calls the appropriate method on detected objects that implement the interactable interface.

The inventory system features drag-and-drop functionality using the event system. A pickup script adds items to available inventory slots when players interact with them, enabling dynamic object interaction with visual representation in the UI.

Following design patterns and programming the best practices ensured maintainable code that enables quick iteration and future flexibility. The scalable architecture allows for additional features like new item types or NPC interactions with minimal changes.

To speed up prototyping, I used some free assets from the asset store, which allowed me to focus on core gameplay mechanics rather than asset creation.

The final result demonstrates my programming capabilities in input handling, UI integration, physics-based detection, and component-based design. The game's core mechanics are solid and provide a strong foundation for future additions.