

## System Explanation

This project is a 2D top-down RPG demo inspired by games I currently play. The core idea is simple: the player must explore the map, collect items, unlock new areas, and eventually fight a final boss. Progression depends on exploring the environment, picking up useful items like potions. Potions give the player temporary advantages like speed boosts or color changes, adding a small twist to the gameplay loop.

From the beginning, I followed a clear architecture: separating data, UI, and logic to ensure maintainability and modularity. The inventory system is the most complete feature, built around an event-driven architecture using ScriptableObjects (InventorySO, ItemSO, DialogueSO) for clean data handling. The inventory supports stackable and non-stackable items, drag-and-drop functionality via InventoryItemSlot, item usage through an ItemActionPanel, and a dynamic description UI with InventoryDescription.

The PlayerMovement uses Unity's new Input System for handling movement input and combines Rigidbody2D with Blend Trees for responsive and smooth animation control. For interaction, the PlayerInteractor component uses Physics2D.OverlapCircle to detect nearby interactables and communicates with them through a custom IInteractable interface. This allows the player to interact with everything from doors to NPCs using the same unified logic.

The DialogSystem is responsible for displaying scripted dialogues using a typewriter effect and blocking input while active. It integrates with interactables like NPCInteraction or DoorInteraction, both of which implement the IInteractable interface.

The InventorySaveSystem keeps track of the current inventory during gameplay using PlayerPrefs. However, it does not persist across sessions, a known limitation I'd resolve with a proper file-based save system. Another thing worth mentioning is the responsiveness of the UI. While functional, it could be improved to better adapt across different screen resolutions and aspect ratios.

Given the time constraints, I'm proud of what I was able to achieve. I designed and implemented several systems, interaction, inventory, dialog, input handling, that all work together cohesively. I believe this is the best I could do within the available time and scope. It reflects a solid grasp of Unity development, event-driven architecture, and best practices around separation of concerns.

**Disclaimer:**

All art and visual elements used in this project come from free Unity Asset Store packages. No scripts were reused - all code was written from scratch by me.

- Game Font : [Free Pixel Font - Thaleah](#)
- Environment tiles: [Pixel Art Top Down - Basic](#)
- UI visuals: [Free Inventory System for 2D Games](#) (visuals only such asCharacter)
- Interactable/Inventory icons: [Pixel Art Icon Pack - RPG](#)