

Documentation

Reused scripts

- Singleton Manager - A singleton
 - provided by my mentor in one of my classes, added a few things for this project
 - Register - register an object deriving from MonoBehaviour
 - Get - access a registered script
 - Remove - remove a registered script
 - RemoveAll - clear all registered scripts
 - Scripts in Singleton
 - PlayerController
 - CameraController
 - UI
 - DialogueManager

Major Scripts

- Inventory - holds and handles an array of item and equipment
 - Variables
 - Item[] Items - array of items
 - Equippable EquippedItem - equipped item
 - Action Evt_OnEquip - an event when player gets equipped
 - Methods
 - AddItem - adds item
 - Has an overloaded version that adds item in a specific index as a parameter
 - RemoveItem - removes item at index
 - Equip - equips an equippable Item
 - Unequip - unequips an equippable item
 - ToggleInventory - opens/Closes the UI for Inventory
- Shop - buy and sell
 - Variables
 - Action Evt_Open - an event when the shop is accessed
 - Action Evt_Close - an event when the shop is closed
 - List<Item> ItemsToSell - list of items that can be bought
 - Methods
 - void Buy - buy an item
 - void Sell - sell an Item from inventory
 - void OpenShop - opens UI and pauses Time
 - void CloseShop - closes UI and resumes Time
- PlayerController -
 - Variables
 - Player Control - the controlled player
 - ControlState State - control State

- Player
 - Shop - opened shop
 - Dialogue - in a dialogue
 - Inventory - opened inventory
- Methods
 - IEnumerator RaycastInteractables - check for interactable objects in Player State
- Input Controls
 - W, A, S, D - movement controls
 - Escape - access the Inventory, closes Shop
 - Mouse left click - continue dialogue
 - Mouse right click - buy, sell and equip
- Interactables
 - Variables
 - UnityEvent<Player> Evt_Interacted - fires an event when interacted to
 - Methods
 - void ShowInteractableUI - shows the object can be interacted by player
 - void Interact - interact with object
- InteractableDetection - used for detecting Interactables in range by using a trigger collider.
- DialogueManager - manage dialogues and UI for the dialogues.

UI

- InventoryUI - displays current and equipped items from the inventory
- Tooltip - displays an item's details
- ShopUI - displays shop and inventory items
- Slot - UI object that displays an assigned item in the inventory
 - Derived
 - SellSlot - displays assigned item with more detail
 - EquipmentSlot - allows equipping items
 - Interaction
 - Hover
 - Drag
 - Drop
 - Click

Thoughts

My thoughts in creating the project

- What to use?
 - Usage of Component Based Architecture and Inheritance based on complexity.
 - Model View Controller for UI

- Unreal's concept of using a PlayerController to encapsulate the controls
- Singleton
 - Despite this, I tried to make sure there were no dependencies of scripts by using events.
- UI Design?
 - Incorporation of Gestalt principles.
 - Stardew valley as reference.

Overall I think I did decently well. I'm glad I've been able to do the features in 4 days, while some of the features were new to me. I believe some implementations can be improved, especially for the UI.