

Game Programming CW1 Journal

Inventory system

I wasn't aware of the grid layout group component in the engine, i originally tried to use Rect transforms and float to arrange all of the, but the grid layout handles this automatically.

Creating new items: The "CreateAssetMenu" keyword is a new thing to me but it is very useful, it only works for scriptable object but this means there is no need to make multiple scripts or game objects for different item.

Using an item: i first thought that if i recreated the scriptable object into a monobehaviour object in the scene, i could methods to the game object and run functions with the UI button, but i found out that methods can also be put into a scriptable object, so i can just call the method to use the item from the scriptable object.

Dialogue System

First i attempted to use an array to handle the list of sentence that needed to be sorted through, but i found out about the queue variable type that allows me to handle this automatically without use of and index and a for loop to iterate through all of the items in the array

Save System

Trying to create a basic save system for saving player data into system storage so that it can be reloaded later for any other reason. For the example player controller, i also need to create some basic data types that will be used to test the save system.

When i was trying to figure out how to unity saves data and i found out that unity has many methods for saving data, databases like MySql would be good for saving large data quantities, but this is isn't useful because i want to simply save player data like a health variable, name, level and even a constructor like a vector 3 for the position, so the PlayerPrefs is the best option as it can easily store these types of data in a secure way that makes it difficult for users to tamper with.

Ability Package

Trying to figure out how the abilities will be activated when a specific input is pressed, i fixed this by using a list of enum containing ability number and assigned them to a Keycode that activates the ability that is attached to the class this makes it easier that having to create a bunch of inputs in the input manager and connect them by code, this allows the code to be more modular as a user simply need to create a ability input scriptable object and assign the keycode they desire.

Using documentation to understand input system and how to receive specific inputs

The ability cooldown is supposed to be handled using floats and a boolean, but as i try to test the ability, i noticed that the resetting of ability doesn't work properly, so i decided to use an enumerator to handle the different states of the ability, this means that the cooldown switches the state to a cooldown state meaning that the ability won't be active anymore, and once the cooldown timer is done the ability state will be set to ready meaning that the next time the player presses the corresponding input for the ability, then the ability will activate again.