

# Learning Journal

---

## Crafting System

---

### Problem:

Store information about items and what ones that you can craft at the given time.

### Solution:

Used a Scriptable object to store the items that can be crafted, using this i selected one out the list to craft, and check the relevant items are on the crafting table, if they are i then remove them from the list, and when the list is 0 then the item can be crafted, which then removes the items and spawns the crafted item.

### Problem:

Getting the book to generate buttons that set the onclick event. But got a error saying cannot convert 'void' to 'UnityEngine.Events.UnityAction' Looked up the issue and its because the AddListen function was then passing the SelectCraftingRecipe to the listener which was void.

### Solution:

In the end i didn't use events, i just made the book a child of the crafting table due to the fact it would be all one system anyway and using events was complicating things.

### Problem:

I also encountered a issue with the Physics.OverlapCube where it wasn't detecting some objects that were on the table, after doing some debugging, i used gizmos to draw where the collider was, and to my surprise it was where it said it was.

### Solution:

So out of ideas i tried using a Vector3 instead of a collider and its .size which worked.

### Problem:

Had to make it so it listed all the required items in a list, where it showed the amount you need and the item type.

**Solution:**

I used a dictionary to check if there was an entry for that required item, if not it will add to the dictionary, but if there was already one in there it would increase it by 1. Which then meant I had a dictionary full of items and the amount they need, which I used a foreach loop to then convert them to a string and add them to a text box.

## Item/Weapon System

---

**Problem:**

Had to get the custom inspector to display all the different types of items with their different properties.

**Solution:**

Had to set up a private variable to hold the item type, so when you click on the item, it checks to see what type it is, it then displays the correct fields for that item. For instance when you click item, it just the general fields, but if you click on a gun it shows other fields such as recoil amount, fire speed and range.

**Problem:**

The scriptable objects weren't saving when you close Unity, so all the changes you made to the scriptable objects, weren't getting saved when you close even if Unity thinks they are saved.

**Solution:**

The fix for this was to set the edited files as "dirty" so when Unity saves it also saves the scriptable objects as they were marked.

## Day & Night System

---

**Problem:**

The sun not rotating properly around, and snapping to certain position.

**Solution:**

Using an animation curve and lerp between two quaternions using the animation curve, it now smoothly rotates around and changes its angle depending on the season.

**Problem:**

The sun is the opposite direction then it should be at the corresponding time.

**Solution:**

Had to change the `Mathf.Lerp` from `Mathf.Lerp(0,360,t);` to `Mathf.Lerp(-180,180,t);` so when the time is set to 12 it would be 0 as its in the middle.

**Problem:**

The sun was casting shadows on objects when it was underneath the map.

**Solution:**

Made it so the intensity of the sun change depending on the time, as this stops shadows appearing from under the world when its night time.

## Terrain Generator

---

**Problem:**

The mesh generator wasn't generating the meshes, when the callback was being called in the current thread.

**Solution:**

It was because the thread wasn't being started.

**Problem:**

Getting the different colours in a shader graph to correctly fade between 2 colours depending on the fade distance and normals.

**Solution:**

I used a lerp which used a smoothstep to transition from the slope angle and the rock fade distance which were variables in the inspector.