

## CW1 Learning Journal Entries

16/02/21

Objects were being selected multiple times on a click, I worked out this is because I was using OnMouseClicked instead of OnMouseClickedUp.

17/02/21

Asets\SelectionComponent\Component\SelectionManager.cs(55,61): error CS1061: 'GetIfSelected' does not contain a definition for 'SetHoverStatus' and no accessible extension method 'SetHoverStatus' accepting a first argument of type 'GetIfSelected' could be found (are you missing a using directive or an assembly reference?)

Error saying I made a syntax error, miswrote the function that is actually called "SetHoverSelectionStatus".

17/02/21

Weird error where for some reason my inspector stopped showing objects I was selecting. Restarting unity fixed this.

17/02/21

Objects not deselecting when being clicked twice or when mouse click happens when the mouse is not over them. I fixed this by removing the checker for multi-selection. Ofcourse, doing this it caused a new issue where shift clicking objects did not work. I sorted this out by moving the shift click checker to the update loop which would deselect objects on mouse click when the mouse was not hovering over them. By doing this it made a whole function of my parent object irrelevant so that can be deleted.

17/02/21

I had trouble making the click and drag code for selecting so I watched this online tutorial to help me. [https://www.youtube.com/watch?v=cd7pgnw5OLA&ab\\_channel=Zenva](https://www.youtube.com/watch?v=cd7pgnw5OLA&ab_channel=Zenva)

The tutorial was very useful in showing me how to make the selection box. By using the print statement I can see that for some reason the objects are not being recognised by the click and drag. After some testing I found that sometimes the objects select if you select lower than them. This suggests the pivot I added has caused issues to the selection and I will instead follow the tutorial more strictly.

17/02/21

After editing the script by removing the pivot the code mostly works but it's somewhat inaccurate. I need to add calculate using the selected object width/height too to allow them to be selected when the mouse is hovering anywhere over them.

23/02/21

Complications with the code for standard clicking and drag clicking interfering with each other. There was no error message, but the code did not do what I wanted. Objects were not selecting properly and the click and drag caused objects to deselect when I did not want them to. Fixing this took over 2hrs. In the end I decided to separate the 2 actions so that when the player was clicking and dragging the standard selection code is temporarily paused and then reset. The video tutorial I mentioned above was helpful.

23/02/21

Selection box only selecting some objects. No error messages. After consulting my tutorial, it seemed I had made a mistake at the part of the code that causes the box to grow and shrink. Although visibly there is no issue, the actual size the code thinks the box has is much smaller than the actual box and is offset awkwardly. To fix this I just readjusted how the width and height were calculated.

23/02/21

I wanted to readjust the code so that it was no longer using parent and child objects to calculate the selection, as it would be silly and awkward to have all selectable objects as children of this object. Instead, I used a public list to store the objects. When adjusted the code I had a "Object reference not found error". When consulting the line (29) I found that the problem was that I had not removed a reference to transform.parent, which of course no longer exists.

23/02/21

While making sticky selection, I wanted to use a raycast to help me find if the mouse is overing hover any object. I used this example to help as I am unfamiliar with raycasts:

<https://answers.unity.com/questions/547513/how-do-i-detect-when-mouse-passes-over-an-object.html>

--Finished with first program

25/02/21

Syntax error with creating a new colour at variable initialization. I researched how to define a Color32 at initialization and found out that I needed to use the keyword "new" before writing the Color.

26/02/21

Error assigning the sizeDelta of the Bar object because Bar is null. I realised I made a mistake by not assigning this script to a sprite or canvas, it was assigned to an empty GameObject. To fix this issue I made a Canvas with an image in it, assigned the script to the image.

26/02/21

LifeBar was shrinking on both sides instead of just from the left. I wanted it to shrink towards the left side of the bar, not on both sides. I looked up a tutorial on how to make a health bar and saw they used the pivot to achieve this.

26/02/21

Had huge problems with getting the bar to change colour. I was using a weird maths method that worked but looked bad. In the end I did some research and found out that there is an inbuilt function for this called Lerp, so I used this instead.

26/02/21

Simple problem with the new Lerp function. The colours were going the wrong way, ie max hp = red and min hp = green. I simply had the 2 variables the wrong way in the code.

```
Color32.Lerp(MinHP_Colour, MaxHP_Colour, HP_Percent);
```

--Finished with second program

03/03/21

Objects didn't spawn loot when they hit the floor. Turns out they were being deleted by the floor before their code could run. I edited the script so that the floor does not delete the object, instead they delete themselves.

03/03/21

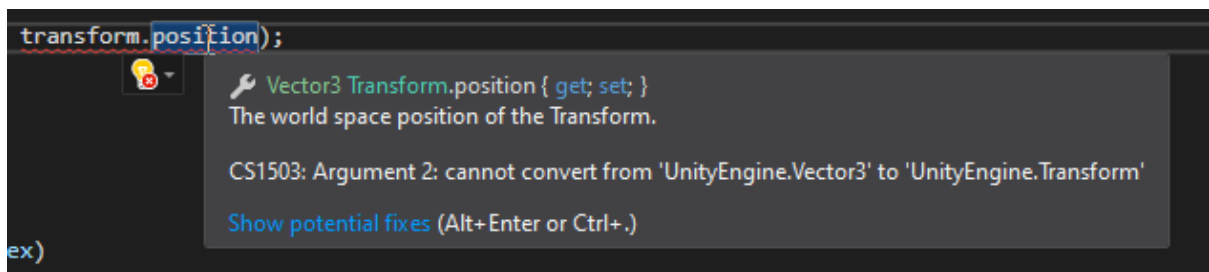
Objects still weren't spawning loot as much as I wanted them to. After a good look at my code I found out the problem was:

```
float TempRandom = Random.Range(0, 100);  
for (int i = 0; i < AllLoot.Length; i++)  
{  
    if(LootChance[i] <= TempRandom)
```

I had the symbol the wrong way round, it should be >=.

16/04/21

Quickly revisited code to make it so loot drops where the object is destroyed. Added to the Instantiate and got this error:



The code thought transform.position was trying to act as a parent, which was incorrect. In the end I finished the whole syntax instead of just doing the game object and position:

```
Instantiate(AllLoot[ItemIndexToDrop], transform.position, new Quaternion(), null);
```

16/04/21

Not really an error but I did not like how the objects were all flying around instead of just falling. I made them trigger objects. This broke the code because now the floor did not know they were hitting it so I edited the code for the collider to react to OnTrigger instead of Collision. As well as that I spread them about so that the loot did not fly outwards.

--Finished with third program

14/03/21

No real issues straight away as the EBPs/SBPs code is simple data container. Creating the production template was fine also with no issues. The code for creating the buttons was much more problematic. I had an issue in updating the number of buttons. I was using an array to hold the number of buttons, and the rest of the production info. This was problematic as the number of items to be produced was unknown straight away. I had a problem with when iterating to add a new

button I had to constantly create a new array while holding onto the data in a temp array like a stack. This created a lot of null reference issues as well as annoying complications and so I decided to scrap this idea and use a list instead which was much more fitting for this job!

16/03/21

Binding the button worked, however when using multiple production templates, the buttons would all be binded to the final production item incorrectly. This stumped me as I have never used the AddListener function before this moment. I worked out the only problem must be that I am using the local variable "i" from the for loop for the listener.

```
Temp_NewButton.onClick.AddListener(delegate { BuildEBP(i); } );
```

Turns out as the value of the local variable does change, the listener actually records this and stays with the value of the variable, even though it is not global! This really surprised me but taught me I need to create another CONSTANT variable for each button. To do this I simply gave each button a function that module that only acts to hold a single constant int. I could also achieve this with making another list of ints, however I decided it may be more useful to store into the buttons, though it is probably somewhat irrelevant.

```
ProductionButtonInfo PBI = Temp_NewButton.GetComponent<ProductionButtonInfo>();  
PBI.EntityToProduce = i;  
  
Temp_NewButton.onClick.AddListener(delegate { BuildEBP(PBI.EntityToProduce); } );
```

16/03/21

```
[19:49:35] IndexOutOfRangeException: Index was outside the bounds of the array.  
ProduceAnObject.Start () (at Assets/ProductionComponent/Component/ProduceAnObject.cs:28)
```

```
i <= AllProducedEntities.Length
```

<- Error

Forgot that for Arrays I should use < not <=.

--Finished with fourth program

Overall, I did not have too many problems with these components, although there were still a lot of errors to overcome. I did not list every error I had because that would have taken so much time to do, I would get barely any programming done, instead I listed a few for each component. The selection component was by far the hardest to make and I am still having some issues with it in a game that I am using it in.

## CW2 Game Project

25/03/21

No specific error, but a problem arose using arrays. I was using arrays to store the spawned items to allow them to be interacted with in the same script. Using arrays caused problems with adding new items and removing existing items, as I would have huge arrays full of null values. There were errors with the program getting confused and finding null instead of the desired GameObject. To fix this I replaced the arrays with lists. The List.RemoveAt function totally fixed the issues I was having!

29/03/21

Objects not spawning with a parent, thus making the inspector very messy. I added the self.transform to the instantiate function to achieve this.

```
GameObject NewItem = Instantiate(ItemToSpawn, SpawnPos, new Quaternion(), transform);  
NewItem.transform.localScale = new Vector3(SpawnSize, SpawnSize, 0.001f);
```

29/03/21

Before being deleted, objects turned inside out and started growing. Checked inspector and found out that the scale was going into the negative numbers. To remedy this, I added code to check if the object was bigger than the scale change, and if not, it reset the scale to 0.

```
if (Object.localScale[0] > Ammount)  
{  
    Object.localScale -= new Vector3(Ammount, Ammount, Ammount);  
}  
else  
{  
    Object.localScale = new Vector3();  
}
```

29/03/21

Objects that were not meant to be destroyed were being destroyed in place of other objects. My initial solution used invoke that would run at instantiation of the object using the objects lifetime to run the delete function. This was not going to work as the invoke function does not take parameters and so I could not say which object was ready to be deleted. Looking back now I could have created extra code to sort the lists, but instead I did a different solution. I created a second set of lists which would separate objects that are ready to be destroyed from other objects that are shrinking. As well as this I changed the shrink functions so that they check each object at every shrink tick to see which ones need to be destroyed.

```

2 references
private void PreDestroyAnItem(int position)
{
    ScoreManager.AddObjectDecayTime(CurrentItems_TimeTakenToDie[position]);

    DelayedDeathList.Add(CurrentItems[position]);
    DelayedDeathList_Time.Add(DelayedDeathTime);

    CurrentItems.RemoveAt(position);
    CurrentItems_Lifetime.RemoveAt(position);
    CurrentItems_ShrinkPerTick.RemoveAt(position);
    CurrentItems_TimeTakenToDie.RemoveAt(position);
}

1 reference
private void FinalDestroyAnItem(int position)
{
    Destroy(DelayedDeathList[position].gameObject);
    DelayedDeathList.RemoveAt(position);
    DelayedDeathList_Time.RemoveAt(position);
}

```

02/04/21

Shooting reset the score to 0. To check why this is happening I added print statements around my shooting code. I realised that on a mouse click the shooting function was being run many many times. This was due to the shooting being linked to GetMouseButton instead of GetMouseDown (first frame of being clicked only). I changed this, but also added a minute reload time to further prevent this issue for people who click very fast, or problematic mice.

```

Active = false;
Invoke("Reload", ReloadTime);
...

```

09/04/21

UI of the score not changing despite functions being called. I searched up why this might be happening, but nothing was fixing it. I even saw in the inspector that the value was changing. After I realised that the code was actually working correctly, I discovered that simply the UI element was not large enough to contain the text and the score, and so to fix it I just made the UI object bigger...

12/04/21

Accuracy for some reason always being 100% or 0%. This was a problem with the maths, rather than the code. I thought to find accuracy you would have to do a complex calculation including 3 variables, number of shots fired, number of shots hit, and number of shots missed. I found this post on the unity forum which helped me work out how to actually do it:

<https://forum.unity.com/threads/calculating-shooting-accuracy.403323/#:~:text=NinjaRubberBand,-Joined%3A%20Feb%2022&text=HemiMG%20said%3A%20%E2%86%91->

,If%20I%20understand%20the%20question%20right%2C%20the%20you%20would%20divide,%2C%200times%20100%20is%20100%25.

HemiMG said: ↑

*If I understand the question right, the you would divide 6 by 56 and then multiply by 100. It would be a little over 10% accurate. For example 6 divided by 12 is 0.5, multiply that by 100 and you get 50%, 6 divided by 6 is 1, times 100 is 100%. Don't multiply by 100 if you need the accuracy for other calculations rather than for user display.*

It turns out that the maths is rather simpler. You just need to do  $\text{Accuracy} = (\text{Hits} / \text{Shots}) * 100$ .

28/04/21

Restart button not working because the scene was not added to the build index. Simple fix by adding it to the build list.