27/10/2017

PG5200-1 17H Tools programmering

Post-Mortem 2

Character Editor in Unity

Project

As per the description of the assignment, the character editor lets a creator edit, save and load the properties of an RPG character inside Unity. The serialization format is to and from JSON.

Structure

The editor starts with four main editing function at the top: create, import, export, and generate (**WIP**). The rest of the editor shows only when a creature is being edited / loaded (i.e. the **_cachedCreature** is not null). The editable properties are common to most RPGs, and reflect directly the content of the Creature model object. Given the straightforward nature of the model, there was no need for a DTO nor a Service class to handle transformations to and from file.

As the structure has been heavily inspired from the one example shown in class, most of the issues have been with making serialisation work properly, as well as obtaining a semi-acceptable layout. Indeed, at first both Creature and Item models extended ScriptableObject, which was useless since I was not going to use the AssetDatabase or another form of serialisation. This alone took me hours to debug, because the serialisation would only happen on the first level - meaning that the current creature's attributes would save properly to and from JSON, but the inventory (a **List<Item>**) would always end up producing an array of **InstanceIDs**. On top of this, I was trying to display each item's property in a ReorderableList which required some internal serialization method I could not figure out and produced results that were not usable - since Items were **ScriptableObjects**, creating an Item required a call to **CreateInstance<Item>()** which I did not manage to make work with **ReorderableList**'s **addCallback**.
When I figured out the issue came from extending **ScriptableObject**, I was already finishing to implement manually a list to display all items in the inventory, so I kept going with the current solution (for lack of time).

Challenges - feedback request

- Handling the layout reminded me a lot of HTML, but since I could not find the proper attributes (and most examples on internet use MonoBehaviour objects and the Inspector) it made it very difficult to get things to look properly. I did not find a way yet to add some colours or make some elements selectable, which would have helped a lot with the inventory list. Packing elements together in sections (think in HTML) and show separators would be nice, but I have no clue how to do that.
- Regarding the inventory list, I still have an issue with the deletion of items which are not on the last index of the list (by clicking on the cross for each item display). Unity does not crash when the cross is clicked, but an exception is thrown.
- All in all I feel like my solution is very basic and perhaps not enough for what was required. As it is, the editor is pretty much a standalone program with little relation to Unity - except perhaps the “Position x,y” vector. I would like to make a “deeper” integration, by which I mean make the creatures available and tangible within a Scene. For instance, the character can be placeable or attached to an entity that is part of the scene (a prefab?).
- Any feedback would be appreciated greatly, regarding how the editor looks in general, and whether the choice of handle was sensible depending on the property to edit (the foldout arrow for the inventory, for instance), or anything else.