Lab Report – 03

Andrew Seba  
Game221 – 001

This lab report takes on the programming practice of creating custom editor scripts for objects and goes even further and goes about how to create an even more custom editor called a custom property drawer. Sometimes the default inspector doesn’t have enough control or visual feedback to help a designer. The main goal is to allow a designer to have full control of an object that makes sense and follows UCD (User Center Design).

The simplest from this lab was custom editors the example was you’ll find out that the default inspector just doesn’t have everything you need to select the right options. You’ll need a new script with ‘using UnityEditor’ and putting in [CustomEditor(typeof(“Class You want to have the custom editor for”))] before your class definition and inherits Editor will allow you to override the default inspector with your own layout. You’ll be assigning variables from the class with EditorGUILayout. > and however you want your data to be selected. Some functions allow you to indent and have foldouts to help organize. Lets say you need to make it even more custom than that, you’ll need to have a script that will be serialized with the original data structure then add a script that holds the original class as a variable this ClassController will be the script that will be on the object in the scene. Using custom property drawer tag and inheriting ‘editor’ in a new script will override the inspector. Doing this wont be able to access the data directly so we need to setup variables to find the serializedObject’s propertys. Now similar to custom editors we can set how to access the data with EditorGUI functions. The important part about custom property drawer is that you can put any editorGui function anywhere in this drawer, and customize the functions themselves.

For this lab I’ve learned about custom editors and to further that knowledge the ability to create custom property drawers. Creating serialized classes to edit objects will be really easy to implement now , and I’ll be able to apply that to almost any game project in the future. This will help designers take complete control over the class’s that make up any game.