Lab 01 Report - Gipson

# Introduction

Many projects fail due to not being organized, or having a structure that not everybody understands. A way to remedy this is to use a preset folder structure for your projects. In this lab I will write a script to create the folder structure for you.

# Methods

The code I will use will use Unity.Editor, Application.dataPath and the built in c# IO library. There will be a method for each level of folder aswell as a method for creating the labeling text files and a driver method, which will be a menu item. The methods will check if the directory needed exist, if it does not it will create it, and then move to the next directory. The driver method will call each method in succession and then call AssetDatabse.Refresh.

# 3 Conclusion

I learned that even though unity doesn’t always need it, it’s best to always use AssetDatabase.Refresh whenever affecting assets via script. I also learned that unity does not build scripts in the editor folder.

# Post-Lab

1. What is the purpose of the Editor folder?
2. For editor scripts. The folder is not included in builds and editor scripts only function if in this folder
3. Menu items can be added to the inspector using what statement?
4. [MenuItem()]
5. What must you watch out for when adding menu items?
6. Overriding existing menu items
7. How do you attach an action to a menu item?
8. By using the statement as an attribute to a method
9. What is the purpose of AssetDatabase.Refresh()?
10. It refreshes the database so that unity knows any changes made to the assets of a project
11. What is the difference between a static asset and a dynamic asset?
12. Assets that are added during runtime
13. Why is it important to keep static and dynamic assets in different folders?
14. Because static assets are used to create dynamic
15. Think about your workflow, what would be your optimal folder structure/organization?
16. \_Scripts, Prefabs, Prefabs/Models, Materials, Materials/Shaders, Materials/Textures, Scenes, Sounds, Editor

# Code

using UnityEngine;

using System.Collections;

using UnityEditor;

using System.IO;

/// <summary>

/// Author: Matt Gipson

/// Contact: Deadwynn@gmail.com

/// Domain: www.livingvalkyrie.com

///

/// Description: PreProcessor

/// </summary>

public class PreProcessor : MonoBehaviour {

#region Fields

static string assetsLocation = Application.dataPath;

#endregion

//generates the first level of folders after checking if they exist

public static void GenerateFolders() {

if (!Directory.Exists(assetsLocation + "/Dynamic")) {

AssetDatabase.CreateFolder("Assets", "Dynamic");

}

if (!Directory.Exists(assetsLocation + "/Editor")) {

AssetDatabase.CreateFolder("Assets", "Editor");

}

if (!Directory.Exists(assetsLocation + "/Extentions")) {

AssetDatabase.CreateFolder("Assets", "Extentions");

}

if (!Directory.Exists(assetsLocation + "/\_Scripts")) {

AssetDatabase.CreateFolder("Assets", "\_Scripts");

}

if (!Directory.Exists(assetsLocation + "/Plugins")) {

AssetDatabase.CreateFolder("Assets", "Plugins");

}

if (!Directory.Exists(assetsLocation + "/Gizmos")) {

AssetDatabase.CreateFolder("Assets", "Gizmos");

}

if (!Directory.Exists(assetsLocation + "/Shaders")) {

AssetDatabase.CreateFolder("Assets", "Shaders");

}

if (!Directory.Exists(assetsLocation + "/Static")) {

AssetDatabase.CreateFolder("Assets", "Static");

}

if (!Directory.Exists(assetsLocation + "/Testing")) {

AssetDatabase.CreateFolder("Assets", "Testing");

}

}

//generates the second level of folders after checking if they exist

public static void GenerateFoldersSecondLevel() {

if (!Directory.Exists(assetsLocation + "/\_Scripts/Common")) {

AssetDatabase.CreateFolder("Assets/\_Scripts", "Common");

}

//dynamic folder

if (!Directory.Exists(assetsLocation + "/Dynamic/Animations")) {

AssetDatabase.CreateFolder("Assets/Dynamic", "Animations");

}

if (!Directory.Exists(assetsLocation + "/Dynamic/AnimationControllers")) {

AssetDatabase.CreateFolder("Assets/Dynamic", "AnimationControllers");

}

if (!Directory.Exists(assetsLocation + "/Dynamic/Effects")) {

AssetDatabase.CreateFolder("Assets/Dynamic", "Effects");

}

if (!Directory.Exists(assetsLocation + "/Dynamic/Models")) {

AssetDatabase.CreateFolder("Assets/Dynamic", "Models");

}

if (!Directory.Exists(assetsLocation + "/Dynamic/Prefabs")) {

AssetDatabase.CreateFolder("Assets/Dynamic", "Prefabs");

}

if (!Directory.Exists(assetsLocation + "/Dynamic/Sounds")) {

AssetDatabase.CreateFolder("Assets/Dynamic", "Sounds");

}

if (!Directory.Exists(assetsLocation + "/Dynamic/Textures")) {

AssetDatabase.CreateFolder("Assets/Dynamic", "Textures");

}

//static folder

if (!Directory.Exists(assetsLocation + "/Static/Animations")) {

AssetDatabase.CreateFolder("Assets/Static", "Animations");

}

if (!Directory.Exists(assetsLocation + "/Static/AnimationControllers")) {

AssetDatabase.CreateFolder("Assets/Static", "AnimationControllers");

}

if (!Directory.Exists(assetsLocation + "/Static/Effects")) {

AssetDatabase.CreateFolder("Assets/Static", "Effects");

}

if (!Directory.Exists(assetsLocation + "/Static/Models")) {

AssetDatabase.CreateFolder("Assets/Static", "Models");

}

if (!Directory.Exists(assetsLocation + "/Static/Prefabs")) {

AssetDatabase.CreateFolder("Assets/Static", "Prefabs");

}

if (!Directory.Exists(assetsLocation + "/Static/Sounds")) {

AssetDatabase.CreateFolder("Assets/Static", "Sounds");

}

if (!Directory.Exists(assetsLocation + "/Static/Textures")) {

AssetDatabase.CreateFolder("Assets/Static", "Textures");

}

}

//generates the third level of folders after checking if they exist

public static void GenerateFoldersThirdLevel() {

if (!Directory.Exists(assetsLocation + "/Dynamic/Animations/Sources")) {

AssetDatabase.CreateFolder("Assets/Dynamic/Animations", "Sources");

}

if (!Directory.Exists(assetsLocation + "/Static/Animations/Sources")) {

AssetDatabase.CreateFolder("Assets/Static/Animations", "Sources");

}

if (!Directory.Exists(assetsLocation + "/Dynamic/Models/Character")) {

AssetDatabase.CreateFolder("Assets/Dynamic/Models", "Character");

}

if (!Directory.Exists(assetsLocation + "/Static/Models/Character")) {

AssetDatabase.CreateFolder("Assets/Static/Models", "Character");

}

if (!Directory.Exists(assetsLocation + "/Dynamic/Models/Environment")) {

AssetDatabase.CreateFolder("Assets/Dynamic/Models", "Environment");

}

if (!Directory.Exists(assetsLocation + "/Static/Models/Environment")) {

AssetDatabase.CreateFolder("Assets/Static/Models", "Environment");

}

if (!Directory.Exists(assetsLocation + "/Dynamic/Prefabs/Common")) {

AssetDatabase.CreateFolder("Assets/Dynamic/Prefabs", "Common");

}

if (!Directory.Exists(assetsLocation + "/Static/Prefabs/Common")) {

AssetDatabase.CreateFolder("Assets/Static/Prefabs", "Common");

}

if (!Directory.Exists(assetsLocation + "/Dynamic/Sounds/Music")) {

AssetDatabase.CreateFolder("Assets/Dynamic/Sounds", "Music");

}

if (!Directory.Exists(assetsLocation + "/Static/Sounds/Music")) {

AssetDatabase.CreateFolder("Assets/Static/Sounds", "Music");

}

if (!Directory.Exists(assetsLocation + "/Dynamic/Sounds/SFX")) {

AssetDatabase.CreateFolder("Assets/Dynamic/Sounds", "SFX");

}

if (!Directory.Exists(assetsLocation + "/Static/Sounds/SFX")) {

AssetDatabase.CreateFolder("Assets/Static/Sounds", "SFX");

}

if (!Directory.Exists(assetsLocation + "/Dynamic/Textures/Common")) {

AssetDatabase.CreateFolder("Assets/Dynamic/Textures", "Common");

}

if (!Directory.Exists(assetsLocation + "/Static/Textures/Common")) {

AssetDatabase.CreateFolder("Assets/Static/Textures", "Common");

}

}

//generates the forth level of folders after checking if they exist

public static void GenerateFoldersForthLevel() {

if (!Directory.Exists(assetsLocation + "/Dynamic/Sounds/Music/Common")) {

AssetDatabase.CreateFolder("Assets/Dynamic/Sounds/Music", "Common");

}

if (!Directory.Exists(assetsLocation + "/Dynamic/Sounds/SFX/Common")) {

AssetDatabase.CreateFolder("Assets/Dynamic/Sounds/SFX", "Common");

}

if (!Directory.Exists(assetsLocation + "/Static/Sounds/Music/Common")) {

AssetDatabase.CreateFolder("Assets/Static/Sounds/Music", "Common");

}

if (!Directory.Exists(assetsLocation + "/Static/Sounds/SFX/Common")) {

AssetDatabase.CreateFolder("Assets/Static/Sounds/SFX", "Common");

}

}

//generates text files that say what the containing folder is used for

static void GenerateFiles() {

Debug.Log("generating files");

File.WriteAllText(assetsLocation + "/FolderStructure.txt", "This is the root of the folder structure");

File.WriteAllText(assetsLocation + "/Dynamic/FolderStructure.txt", "This Folder is for dynamic assets");

File.WriteAllText(assetsLocation + "/\_Scripts/FolderStructure.txt", "This Folder is for Scripts");

File.WriteAllText(assetsLocation + "/Shaders/FolderStructure.txt", "This Folder is for Shaders");

File.WriteAllText(assetsLocation + "/Plugins/FolderStructure.txt", "This Folder is for plugins");

File.WriteAllText(assetsLocation + "/Gizmos/FolderStructure.txt", "This Folder is for gizmo scripts");

File.WriteAllText(assetsLocation + "/Extentions/FolderStructure.txt", "This Folder is for extentions");

File.WriteAllText(assetsLocation + "/Editor/FolderStructure.txt", "This Folder is for Editor scripts");

File.WriteAllText(assetsLocation + "/Testing/FolderStructure.txt", "This Folder is for testing/placeholder assets");

File.WriteAllText(assetsLocation + "/Static/FolderStructure.txt", "This Folder is for dynamic assets");

}

//calls all directory methods and file method then refreshes the database

[MenuItem("PreProcessor/Generate Folder Structure")]

static void GenerateStructure() {

GenerateFolders();

GenerateFoldersSecondLevel();

GenerateFoldersThirdLevel();

GenerateFoldersForthLevel();

GenerateFiles();

AssetDatabase.Refresh();

}

}