Function Rendering Unity

Technical Manual

Last Updated: 21/5/2019

By Khiew Jian Bin

Contact: [khiewbin@gmail.com](mailto:khiewbin@gmail.com)

Table of Contents

[Reading Instructions 4](#_Toc9813902)

[1.Getting Started on Windows/Mac (Project Setup) 4](#_Toc9813903)

[1b.Project Settings 4](#_Toc9813904)

[2.Important Notes About using Unity 5](#_Toc9813905)

[3.Main Process to generate and display Mesh 7](#_Toc9813906)

[4.UI Implementation 8](#_Toc9813907)

[Header 9](#_Toc9813908)

[InputSection 9](#_Toc9813909)

[How to add new buttons in Header 10](#_Toc9813910)

[How to change color/graphic or display text of the static UI Panels and Buttons 10](#_Toc9813911)

[How to change color/graphic or display text of the inputfields 10](#_Toc9813912)

[How to change what buttons do, when clicked 10](#_Toc9813913)

[How to change Icon for Directory Item and File Item 11](#_Toc9813914)

[How to change/update the coordinate Axes Model 11](#_Toc9813915)

[Q: Why does clicking button do nothing 12](#_Toc9813916)

[Q: Why is the text font for the Code inputfield more sharper and different from the rest. 12](#_Toc9813917)

[Q: How to add more advance UI, aside from Unity’s basic dropdown,slider,buttons,toggle 12](#_Toc9813918)

[Main Script Implementation 13](#_Toc9813919)

[Main.cs 13](#_Toc9813920)

[Other Implementations 16](#_Toc9813921)

[Programable colors (Only in Unity Editor,Not available for other platforms) 16](#_Toc9813922)

[How to change material 16](#_Toc9813923)

[Developer Notes 17](#_Toc9813924)

[Things useful to know 17](#_Toc9813925)

[Working in teams on Unity Project 17](#_Toc9813926)

[Transferring the entire Unity Project 17](#_Toc9813927)

[Common Issues you might encounter 18](#_Toc9813928)

# Reading Instructions

Please first read Section 1,2,3. Next either read UI Implementation or Script Implementation

Unity Official Manual: <https://docs.unity3d.com/Manual/>

# 1.Getting Started on Windows/Mac (Project Setup)

1. **Download** and **Install** the **latest** or the project version **Unity 2018.2.18f1** Unity Personal from official Site: <https://store.unity.com/download>

Download previous versions here: <https://unity3d.com/get-unity/download/archive>

1. **Download** the project’s here:<https://github.com/frozonnorth/Function-Rendering>
2. **Locate** a file within the Asset Folder named: "(filename).unitypackage.txt"
3. Rename/Remove ‘.txt’ from the file name:

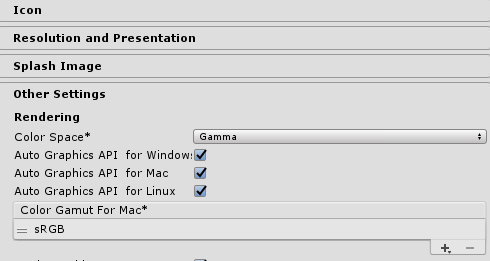
"(filename).unitypackage.txt" to "(filename).unitypackage"

1. Open the project folder in Unity
2. Within Unity's Project Explorer or Windows, double click and run the unitypackage to import into unity.
3. Delete the unitypackage if you want
4. Within Unity Project Window, Locate the file Assets/FunctionRendering/MainNewUI 2.unity (scene) file.
5. Double Click to load the main scene.
6. You should be able to Click Play without having any errors.

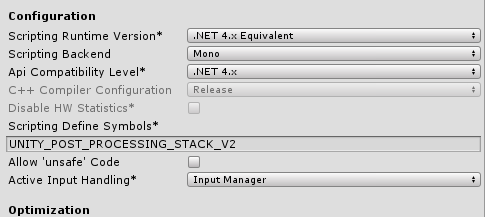
# 1b.Project Settings

If after importing the package and still unable to run, check and do the follow steps

1. In Unity, open the project settings by going [Edit]->[Project Settings]->[Player], this will show up in the inspector
2. Expand the Other Settings Tab



1. Scroll down to the Configuration
2. Ensure that the Scripting Runtime Version is .NET 4x Equivalent and Api Compatibility Level is .NET 4.x like on the image below



1. Do this for the different platforms that you want to target, Switch platform by clicking the Tabs



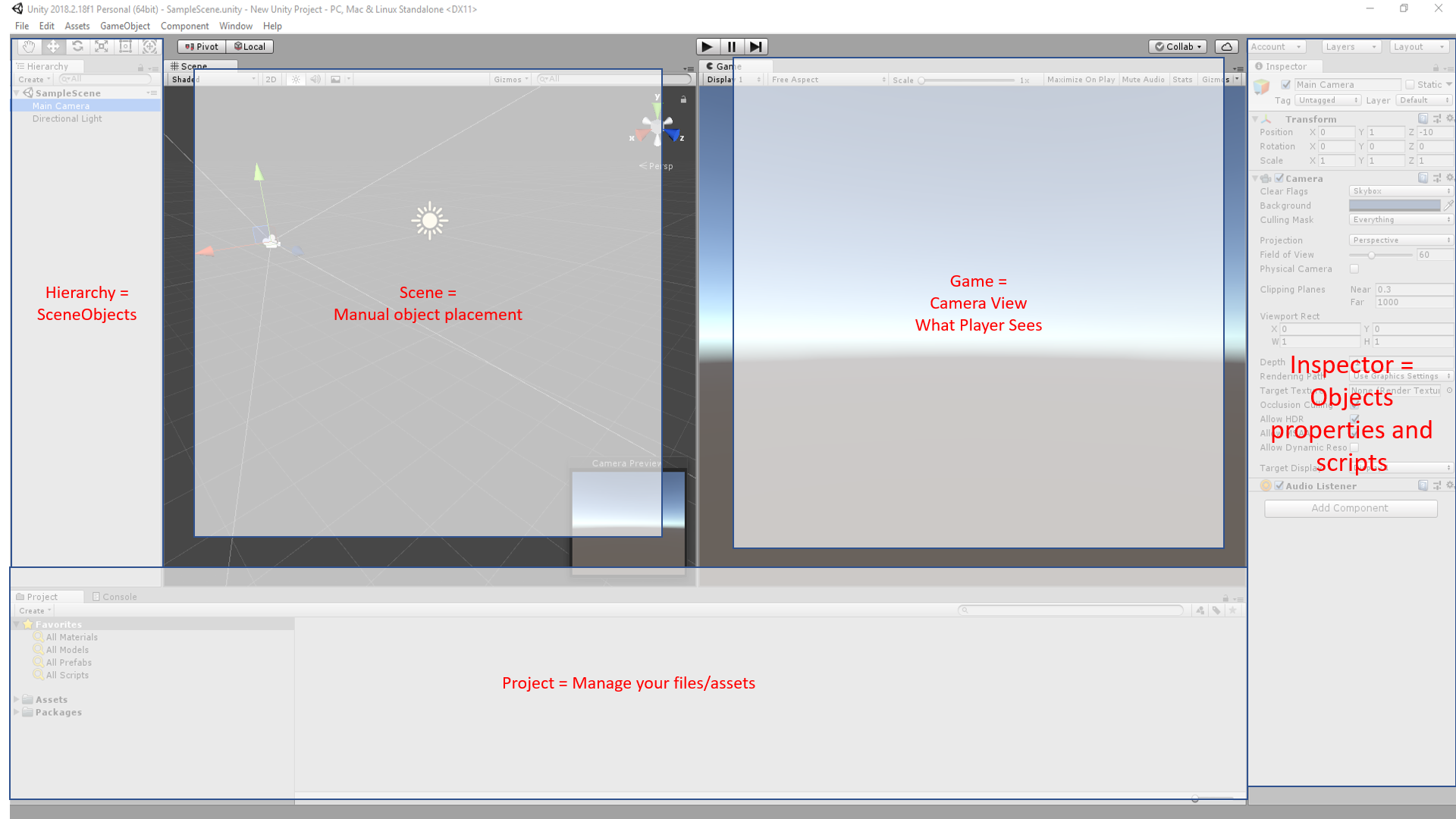
# 2.Important Notes About using Unity

You can find everything about unity here: <https://docs.unity3d.com/Manual/>

I will highlight certain important stuff required to understand how Unity works, Please Read in full before proceeding to next section of this manual

* Unity’s “GameObjects” are just containers that holds “Components”
* The most basic component is the Transform component, similar to the Transform Nodes in FVRML
* C# scripts is for to have a custom(script) Component added to a GameObject to control other Components within the same GameObject
  + **Read:** <https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>
  + **Read:**[https://docs.unity3d.com/Manual/ControllingGameObjectsComponents.htm](https://docs.unity3d.com/Manual/ControllingGameObjectsComponents.html)
* Understand how script components work with the “Inspector”
  + **Read:** <https://docs.unity3d.com/Manual/VariablesAndTheInspector.html>
* Unity works on Hierarchical system, Components are executed according to the hierarchical order, of which objects is parent,child and comes first.
  + Normally you wont have to worry about this, but it’s important to take note

Unity Editor Overview



* This is the suggested window layouts for working on the project
* To customize: <https://docs.unity3d.com/Manual/CustomizingYourWorkspace.html>
* Hierarchy: Shows all gameObjects in the current Scene File
* Scene: Move/Rotate/Scale Objects
* Game: Shows the view of the Main Camera in the Scene
* Inspector: Shows the components of the GameObject selected in the hierarchy
* Project: Your files – scripts, textures, 3dmodels Scene files

# 3.Main Process to generate and display Mesh

Files involved: Main.cs, FunctionGeneratedMesh.cs, MarchingCube.cs

When user clicks Run button, it executes main.OnRunClick() which executes

main.createMeshFromTabData()

1. Looks at all the inputfields to do inputvalidation.
2. Set the object diffuse,specilar,shininsss
3. Remove Previous Mesh
4. Add functiongeneratedMesh component
5. Copy object info over to functiongeneratedMesh component

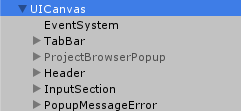
functiongeneratedMesh.StartDrawing() gets executed

1. Determine if its parametric or implicit
2. Process the object info to prepare for generation
   1. Check to see if time is used
   2. Use Regex to process inputs code
   3. Inject processed code into the script template
   4. Copy generation info over. E.g parametric domain/resolution
3. Compile a new c# script (CompliedMesh) and add as component.
   1. Parametric meshes algorithm are directly written in.
   2. Implict meshes algorithm are called and done in MarchingCubes.cs
4. The compiledMesh component will do two main things
   1. Start():Generate the Mesh and assign the mesh to the gameobject
   2. Update():if time is used, increment var ‘t’ and generate and reassign mesh every frame

# 4.UI Implementation

Please first read: <https://docs.unity3d.com/Manual/UICanvas.html> and <https://docs.unity3d.com/Manual/UIBasicLayout.html> and <https://docs.unity3d.com/Manual/script-Button.html>

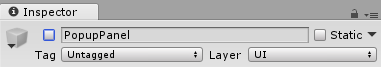
This section will explain the main parts of the implementation which all lies under UICanvas.



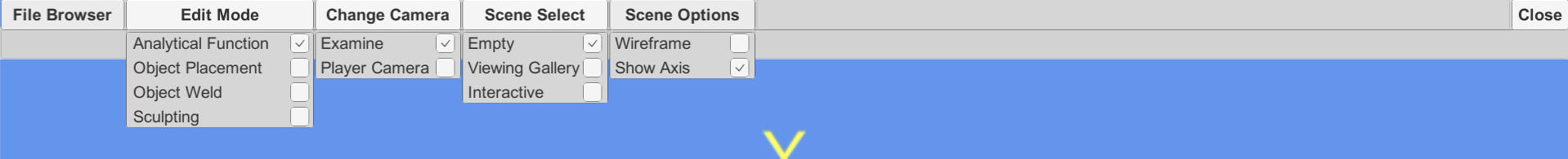
* **UI Canvas** overlays UI across the entire screen.
* **EventSystem** handles UI user interaction such as mouse clicks and touch for desktops and phones platform. (Do not Touch)
* **TabBar** is implementation to have multiple tabs (like internet browsertabs) which just stores the objects details
* **ProjectBrowserPopup** is implementation to have an internal filebrowser for easy loading of files
* **Header** is a container of multiple buttons at the **top**
  + Buttons will bug and mispositions when you “play” then exit “play” mode.
  + HeaderMisalignFix.cs fixes this.
* **InputSection** is a container of multiple inputfields and buttons at the bottom
* **PopupMessageError** is where the error text appears.

Note: Unity’s UI system can be buggy at times when using Vertical/Horizontal Layout Group

Note2: Greyed Items = Not Active and invisible. You can Activate/Deactivate in Inspector checkbox left of the name



## Header



* Header
  + has a MouseOverUI script to trigger when a mouse is over the UI
  + Used to deactivate/stop camera from rotating
* FileBrowser
  + OnClick() show/hides the ProjectBrowserPopup
  + ProjectBrowserPopup has its own buttons which interacts with **main.cs** under

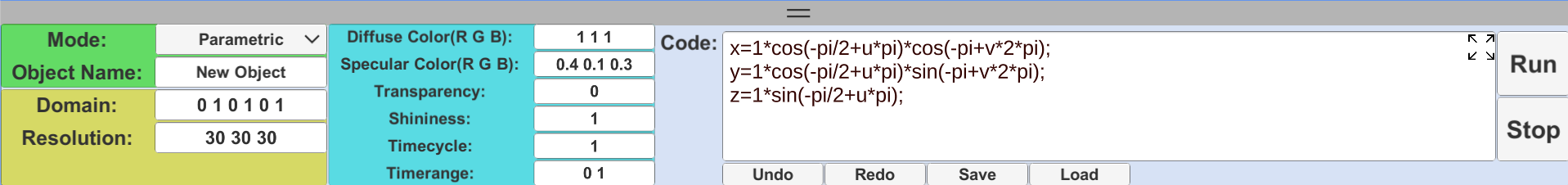
#region Directory and Files Click Events

* EditMode (PlaceHolder)
* Change Camera (PlaceHolder)
* Scene Select (PlaceHolder)
* Scene Options
  + Wireframe OnClick() interacts with maincamera’s **EditorCamera** component under

region To Allow Wireframe Mode or not

* + Show Axis OnClick() interacts with CoordinateAxes that has a script component that allows to show or hide
* Close
  + OnClick() stops or close application

## InputSection



* Inputsection
  + has a MouseOverUI script to trigger when a mouse is over the UI
  + Used to deactivate/stop camera from rotating
  + Has a ExpandButton to move the UI off the canvas when onClick()
* Inputfields
  + All input fields objects are assign to main.cs by inspector used when user clicks Run button
* Buttons
  + All buttons OnClick() calls functions in main.cs under #region Run/Stop/Save/Load/ Button Clicks
* Code Enlarge button
  + OnClick() interacts with a non-active ExpandedCodeMenu that has a PopupButton component to activate/deactivate (show/hide)

## How to add new buttons in Header

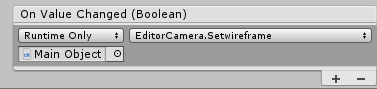
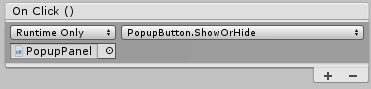
1. Select **Header Object** in hierarchy
2. Expand and Choose one existing buttonpanel in hierarchy

(e.g.**SceneOptionPanel** for a dropdown menu, or **CloseItemPanel** for a normal button)

1. Right-Click **Duplicate** the SceneOptionPanel
2. **Reorder/Shuffle** by carefully drag drop the duplicated panel in hierarchy as **child** of Header.
3. The Header will grow to exceed out of the UICanvas.
4. Expand FillerItemPanel, Select **FillerContent** and use Rect Tool(Hotkey T) and manually resize so that UI does not exit the UICanvas.

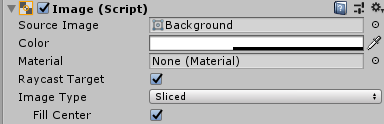


1. Right-Click Rename the new panel and its child objects
2. Modify the Contents in the child PopupPanel in the inspector(such as display text)
3. Ensure that the OnClick or OnValueChanged is also modified

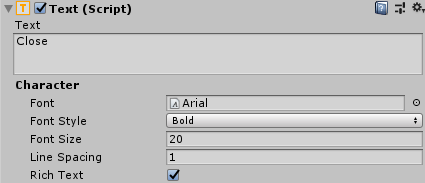


## How to change color/graphic or display text of the static UI Panels and Buttons

Color: Modify the **Image** component using the inspector.



Text: Modify the **Text** component using the inspector. (Usually on child object)



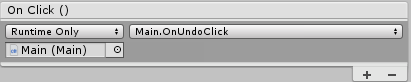
## How to change color/graphic or display text of the inputfields

Color: Modify the **Image** component using the inspector.

Text: Modify the **InputField** component using the inspector. (Not Text Componentt)

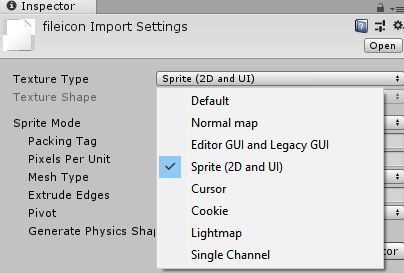
## How to change what buttons do, when clicked

OnClick: Modify the Button Component OnClick() field. Drag drop from the hierarchy and choose the components.methodname in the right dropdown

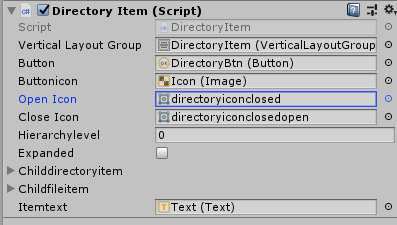
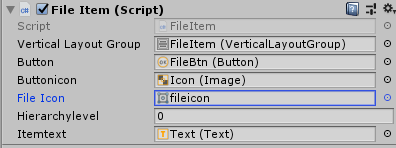


## How to change Icon for Directory Item and File Item

1. Copy your new icon into Assets/UIButtonsInputFields/DirectoryExplorer/Icon Folder
2. Locate your new icon and view it in the inspector. The inspector will show import settings of the icon.
3. Change icon **Texture Type** to **Sprite** and click **Apply** at the bottom right corner



1. Locate directoryItem.prefab or fileitem.prefab in DirectoryExplorer folder and click on it to view it in the inspector.
2. In the inspector, look under Directory Item script component.
3. Assign your new Icons by either dragging in the icon or clicking the circle reticle and choosing the icon.

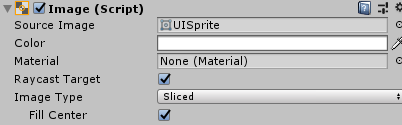
## How to change/update the coordinate Axes Model

1. Locate CoordinateAxes object in the hierarchy
2. Take note of the ‘CoordinateAxes’ script component.
3. Update/Delete the object
4. You can change the color using the inspector
5. Note that the “X,Y,Z” are textMesh components with billboarding script
6. Locate the new 3dModel axes in the project window, drag drop onto the Scene
7. Add the ‘CoordinateAxes’ script components.
8. Locate the SystemControlLoop-Main component, assign your new axes in inspector



## Q: Why does clicking button do nothing

A: Check the button OnClick() in inspector, and make sure the button has a Image Component and Raycast Target is checked.



## Q: Why is the text font for the Code inputfield more sharper and different from the rest.

A: Code inputfield uses TextMeshPro-inputfield, instead of the regular inputfield to allow scrolling of the inputfield. Scrolling can be achieved using regular inputfield but is very tricky and buggy to do.

## Q: How to add more advance UI, aside from Unity’s basic dropdown,slider,buttons,toggle

A: You have to experiment with the ui system, additional UI and interactions are usually done using customscripts. Or you can use other UI systems (created using scripts) in the Unity AssetStore made by other people.

For Example, for inputfields to have input validation, you need to do this using scripts

## Q: How was the filebrowser implemented to popup OS specific dialog box

A: Using **https://github.com/gkngkc/UnityStandaloneFileBrowser**

# Main Script Implementation

Note:most scripts are commented and segmented by #regions, #regions only works for Microsoft visual studio to collapse code.

Note: includes platfrom specific code, see <https://docs.unity3d.com/Manual/PlatformDependentCompilation.html>

This section will point out where certain things are being done in the scripts.

## Main.cs

* Has variable References to and controls all the UI/Inputfields
* Has variable References to and Controls Camera
* References the mesh material used for object generation

### Main.Start()

* Application settings (fullscreen)
* Creating an empty gameobject to hold the mesh (mesh filter + mesh renderer)
* Setting up Camera
* Setting up Default values of inputfields
* Setting up code undo support
* Obtain projectpath from Playerprefs (app program data cache/settings)
  + See <https://docs.unity3d.com/ScriptReference/PlayerPrefs.html>

### Main button events

* OnButtonClick() events can be found at buttom of the main.cs file
  + The UI buttons component OnClick() references their respective main.OnButtonClick() function
  + This is how the UI interacts with the script component to do stuff

### Main.CreateMeshFromTabdata

* Large function to generate the mesh and assign to the empty gameobject
* Add a new FunctionGeneratedMesh component
* Do input validation on parametric/implicit inputfields and give to new FunctionGeneratedMesh component
* Do input validation on the rest of the inputfields (specular,diffuse etc)
* Display error message if any from input validation and stop exeucution
* Set material and set the diffuse/specular/transparency values
* Tells FunctionGeneratedMesh component to start drawing

### FunctionGeneratedMesh.cs

* StartDrawing will check settings to compile for parametic or implict code
* CompileParametricCodeAndAttachToGO or CompileImplicitCodeAndAttachToGO
* Each will process the inputcode to allow for
  + Math functions
  + Carret symbol ‘^’
  + Custom variables
  + Using DoStringFormat(inputcode)
* Inject the process input code into script template (string) with parametric/implicit algorithm
* From the string, compile the script, attach the script as a component and Invoke the CompliedMesh.Setup() function.
* The component will execute CompileMesh.Start()/Update() to generate mesh and update the mesh.

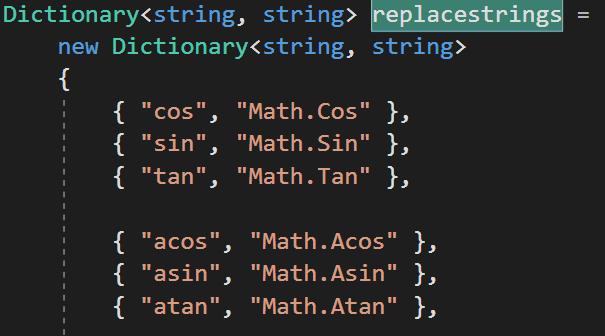
## Original Implicit Marching Cubes used

<https://github.com/ttammear/unitymcubes>

MarchingCube.cs was taken and modified to be the current version

Note: I choose this online sample, because its all-in-one file, other examples have a lot of additional .cs files

## How to add/modify math function namings to support

* Look at FunctionGeneratedMesh.DoStringFormat() replacestrings
* This will format the inputcode to find the entry on the left and replace with entry on the right where the right side is a proper c# Code to execute the math function
* Example. cos(###) will be replace with Math.Cos(###)
* You can expand to include all of the c# .net math function
* You can also create your custom math function.

## How to add/modify custom symbols support for code (like carret ‘^’)

* You will have to process the custom symbols using .NET Regex

## Q: Is there a neater way to support math and custom variable and carrets than using complex REGEX

A: Yes, you can use a proper mathparser, that can evauate math expressions, but due to the algorithm used to generate parametric, this makes generation time for parametric veryslow.

## How to save the generated mesh object, that have the same behavior (like animating) into unity (file)

1. Normally you can select the object in the hierarchy and drag into the project window to save the object as prefab. See <https://docs.unity3d.com/Manual/Prefabs.html>
2. But because the object is code generated, you will need to save entire object and it’s components into prefab and unity asset files using code.
3. You will need to write your own custom script to do this.

## How to change “starting/default” inputfield values displayed on application start

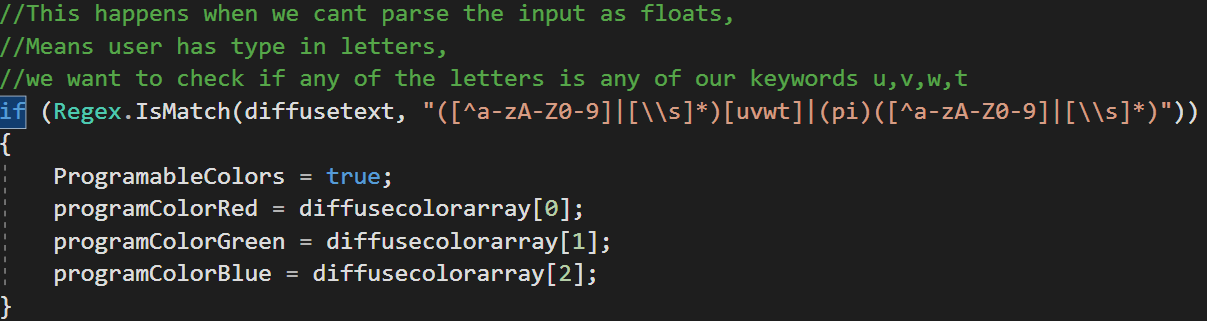
Look at SetDefaultValuesToInputField() in main.cs

## How to change default fullscreen yes/no

Look at Start() in main.cs

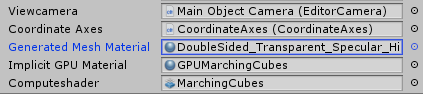
# Other Implementations

## Programable colors (Only in Unity Editor,Not available for other platforms)

1. When user click run, it will go though input validation of the inputfields and, check the diffuse color inputfield for keywords u,v,w,t using Regex
2. After setting the flag, it then proceed to set the material, created using **ProgrammableColor.CreateMaterial()**
3. The function creates a new shader file using a template, and saving onto disk,injecting red,blue,green diffuse color values
4. It forces unity to update and compile the shader code, to be used Only in Unity Editor

## How to change material

1. Obtain your own material, either by first creating an empty material, and assigning a shader, or download some free unity shaders online, or write your own.
2. Select the main object component and assign the material in the inspector



## Q: I changed my material, my new material has a new apperenace property like EmissionColor, how do I access and modify the settings in code

A: depending on the property, use material.setColor,material.setFloat or material.setVector. See <https://docs.unity3d.com/ScriptReference/Material.html> and <https://docs.unity3d.com/ScriptReference/Material-shader.html>

## How to use GPUImplicitRendering (Note that this is a test implementation)

1. Look at FunctionGeneratedMesh.StartDrawing() and uncomment CompileGPUImplicitCodeAndAttachToGO(inputCode); and comment the normal one

# Developer Notes

## Things useful to know

Anything done within unity editor during playmode, is reverted when playmode ends.

## Working in teams on Unity Project

Three ways to do this:

* Share selected work files by “**Export Package**” within Unity Editor (recommended)
  + <https://docs.unity3d.com/Manual/HOWTO-exportpackage.html>
  + Preserves **.meta** settings
  + Packages must be carefully imported if modifications was made as it could cause non-reversable change to scenes, objects and prefabs.
  + Packaging the entire assets folder, is a way to transfer the entire project.
  + Best when each member only works on their specialized task – Programming, Art Assets, UI Design.
* Use SVN/Version control on the entire Unity Project to share files
  + Including **.meta** files and other auto generated files but this will drastically increase upload and download time.
  + Note: **.meta** files have import settings and some of them contains linking information to scenes and prefabs. Excluding **.meta** file might break this linkage and scenes might not work properly.
* Use Unity Teams
  + <https://unity3d.com/how-to/happier-faster-teamwork>
  + See it as an integrated SVN in Unity

## Transferring the entire Unity Project

The best way to transfer your unity project to another person to run is:

Sender:

1. “Export Package” the whole Assets folder
   * <https://docs.unity3d.com/Manual/HOWTO-exportpackage.html>
2. Transfer the Exported Package together with the ProjectSettings Folder
3. Note: Unity ProjectSettings folder contains configuration settings which is not included in the package.

Receiver:

1. Create an empty folder (main project folder)
2. Inside the new folder, paste in the ProjectSettings folder and create a folder name “Assets”. (This is required for unity to detect it’s a unity project)
3. Launch Unity and open the folder/project
4. Execute/DoubleClick the exported package and select to import everything

## Common Issues you might encounter

1. After click Play to test the game, Unity Editor hangs
   1. There is an endless loop in your code or
2. After click Play to test the game, Unity Editor does not hang, but the game freezes,
   1. Unity encountered an error and “Pause” the game.
3. After making changes to your c# files and clicking Play to test your changes, you find that the game does not update the changes.
   1. Normally after saving, unity will detect and compile the c# files but sometimes it does not compile.
   2. Force unity to compile by making another edit to the c# files and saving again.
   3. You should also take note of platform specific code #if Windows
4. After renaming variables and functions, the game stops working/ certain features are broken.
   1. The variable or function is referencing, or is referenced in someway though the inspector.
   2. After renaming, components are links/references are not auto updated, make sure to check and reassign gameobjects components in the inspector
5. When importing a modified script/scene using Unity Packages, the game/scene no longer works
   1. Becareful when importing modified stuff, importing a script with renamed variables or renamed filenames will cause problems.
   2. Importing will overwrite the existing, by the time you found out that the imported script caused problems with the existing scene. You will not be able to obtain the old script back and revert.