Game Programming Assignment Part 4 Unity Tutorial Document of Scripts as Behaviour Components

Scripts in Unity are about the behaviour of components that apply in Unity scripts and how to Create and attach them to objects.

Although scripts can be considered behaviour components in Unity, other components in Unity can be applied to objects and are visible in the inspector. With this specific example, the cube has a Rigidbody component, which applies to a physics mass, and as you press play, the cube will fall to the ground using gravity. A script has also been added as an example of this behaviour. The script has code that can convert the colour into a different cube colour by affecting the color value of the default material joined up to that object.

However, when pressing the keyboard's R key, the colour converts into red. When pressing G, the colour converts to green. When pressing B, it will change to blue when attaching the script to the object. When referring to Game Objects mentioned in this item, we drill down to the value we prefer and affect it. Here, we’re directing the game object this script attached to and then addressing the renderer, which is the component from using mesh render. We then addressed the material joined to that render, moreover, the colour of that material, and we are positioning it to a shortcut called red, which is part of the colour class.

Therefore, if you press play, then use the keyboard's R, G or B keys to change the colours, and you notice that the material is being affected, the material can be applied to the renderer. The default diffuse is that you can see the list there. They then affect the primary colour value and set it to a specific value. Something identical as it would be if I were doing it by hand in the editor. Scripts can be created in the project panel by selecting Create. For instance, they can then be attached to objects either by dragging and dropping or by choosing the Add Component button at the bottom of the component menu and then choosing from the list of scripts in the current project. Scripts can also be created utilising the add component button by choosing new scripts from the bottom and naming them from the drop-down. This can be created and added in one step.

Finally, adding a script to your object means selecting the object in the hierarchy, choosing component > Scripts, and then choosing from the list of scripts for the current project. You can even apply a script to do all manner of other behaviours of objects by thinking of scripts as a component that you can use to create a type of behaviour for different Game Objects in your game. It can be about the character, environment or scripts that manage the game's functionality.

using UnityEngine; using System.Collections;

public class ExampleBehaviourScript : MonoBehaviuor {

void Update() {

if (Input.GetKeyDown(KeyCode.R)) {

GetComponent<Renderer> ().material.colour = Color.red;

} if (Input.GetKeyDown(KeyCode.G)) {

GetComponent<Renderer>().material.color = Color.green;

} if (Input.GetKeyDown(KeyCode.B)) {

Get Component<Renderer>().material.color = Color.blue;

}

}

}