**Showing and hiding objects from view**

The first thing needed in the scene is a script that sets the variable of the player’s view. To do this, create an empty game object in the scene and attach a script to it called SwitchView. Open up the script, and assign a bool called viewchange. Set the bool to false.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class SwitchView : MonoBehaviour {

public bool viewchange = false;

// Use this for initialization

void Start () {

}

// Update is called once per frame

void Update () {

}

}

In void Update, create an if statement. This will allow the game to change the newly created Boolean at the press of a key. In this case, the M key will be used. Inside the brackets, type “Input.GetKeyDown(KeyCode.M)”. Then, within the if statement, create another if statement. In this if statement, make it so that if viewchange is false, set it to true. After this if statement create an else function and set viewchange to false.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class SwitchView : MonoBehaviour {

public bool viewchange = false;

// Use this for initialization

void Start () {

}

// Update is called once per frame

void Update () {

if (Input.GetKeyDown(KeyCode.M))

{

if (viewchange == false)

{

viewchange = true;

}

else

{

viewchange = false;

}

}

}

}

Now add a cube into the scene. Create a script called ViewObject1 and attach it to the cube. To allow the object to disappear from view, you will need to call the script that we created earlier. To do this, use the script name as a public variable (“public SwitchView switchview”). To actually hide the object, you’ll need to call the object’s renderer. Set the renderer to private to call the renderer attached to the object associated with the script.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class ViewObject1 : MonoBehaviour {

public SwitchView switchview;

Renderer rend;

// Use this for initialization

void Start () {

}

// Update is called once per frame

void Update () {

}

}

Now we need to set the renderer to the object’s renderer. In void Start, use GetComponent and type “Renderer” in the triangle brackets. Remember to place a regular set of brackets after this, otherwise the function will not work. Then, to disable the object in normal view, you set the “enabled” value of the renderer to false.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class ViewObject1 : MonoBehaviour {

public SwitchView switchview;

Renderer rend;

// Use this for initialization

void Start () {

rend = GetComponent<Renderer>();

rend.enabled = false;

}

// Update is called once per frame

void Update () {

}

}

Now, in the update void, make an if function. Now we need to call the viewchange bool from SwitchView. To do this, type switchview.viewchange == true in the if brackets. Now set rend.enabled to true. Make an else function and set rend.enabled to false.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class ViewObject1 : MonoBehaviour {

public SwitchView switchview;

Renderer rend;

// Use this for initialization

void Start () {

rend = GetComponent<Renderer>();

rend.enabled = false;

}

// Update is called once per frame

void Update () {

if (switchview.viewchange == true)

{

rend.enabled == true;

}

else

{

rend.enabled == false;

}

}

}

Now that the scripts are completed, select your cube in the hierarchy. Look at the script in component view, then drag the empty object into the SwitchView slot.