Game Programming Assignment Part 1 Unity Tutorial Document of Detecting Mouse Clicks!

Detecting mouse clicks can be used on a Collider or GUI element when using Unity to select, convert or drag the shape objects around to position and create them into a different shape in any way you want.

However, when using the Alt key plus scroll wheel on Unity, I can move the Unity screen up, down, right and left to have more places to build and create more shapes in Unity. Also, by using the Alt key and the left side of the mouse, I can freely move the screen around Unity horizontally up and down and vertically right and left. The mouse can utilise rotation, scale, react and movement tools to convert shapes into characters, buildings, weapons, etc. Lastly, just by using the scroll wheel, I can zoom in and out on Unity, and if I were to select a shape object, I could use the keyboard F to zoom in to it much closer but not to zoom out from the selected shape.

When using the detective tool in Unity, it's crucial also to utilise the on-mouse-down tool. This tool allows the Mono Behaviour, which has the exact implementation as the on-mouse-down tool, to attach to game objects with a collider. This ensures that the on-mouse-down will respond instantly when you select the left-down button, enabling it to work correctly.

Here is an example of clicking on the game object when the left mouse button has been pressed.

using UnityEngine;

public class Clicker: MonoBehaviour

{

void OnMouseDown ()

{

// Code here is called when the GameObjects is clicked on.

}

}

Here is a unity OnMouseDown example of a programming language about a Door Object with a rigid body attached to a Box Collider

Using UnityEngine;

public class Mouseclick : MonoBehaviour

{

void OnMouseDown()

{

Debug.Log (“Clicked on the Door!”);

}

}

Another Unity example of OnMouseDown programming language is adding force to the door when clicked.

using UnityEngine;

public class MouseClick : MonoBehaviour

{

private RigidBody rb;

private void Awake ()

{

rb = GetComponent<RigidBody>();

}

}

void OnMouseDown()

{

rb. AddForce (- Transform.Forward \* 500f);

rb . useGravity= true;

}