MS Move Objects

Created by: Marcos Ismael Schultz
Site: www.schultzgames.com
Email: marcos11-24@hotmail.com

MS Move Objects is a simple and effective system that allows you to move objects around the scene, preferably using the mouse.

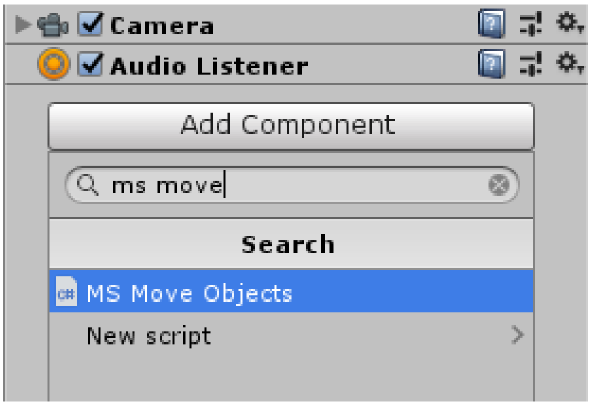
The system has the following features:

- Move objects;
- Throw objects;
- Rotate objects
- It is possible to choose between two different modes to detect objects in the scene, allowing the system to be used in mobile cameras or fixed cameras.
- It is possible to choose between three modes of movement for objects, making the system compatible with most projects without any problems related to physics.
- It is possible to configure how the movement, rotation or throwing of objects will happen, regulating force, speed, among other properties.
- It is possible to choose the controls that execute each command separately, making the system customized and compatible with most projects.
- It is possible to identify the objects that can be moved in the scene through their tag, name, or even through their PhysicMaterial.
- The system has indicative textures that work automatically together with the system.
- The code has debugging functions, which are called when an object is grabbed or dropped.
- All public variables in the code have their functionality explained through 'tooltips', in order to facilitate the understanding and use of the system.

To make understanding the system simple, it is explained using 'tooltips'. To view the explanation, just rest your mouse over the desired variable, and the text will appear.

How to use

To use the system, simply add the code 'MS Move Objects' on your player camera. To do this, simply select your player camera, search for the component by its name, and add it, as shown in the image below.

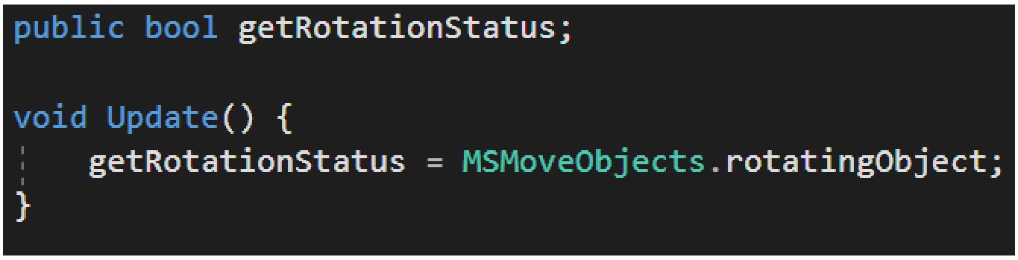


After adding the component, just configure it, following the tips that are shown in each variable, when the mouse is resting on them.

The objects that can be moved must have a **Rigidbody** component, a **Collider**, and must have the tag mentioned in the **'MS Move Objects'** script, or the name mentioned in the same code, or the physic material associated with the code.

The script also has a texture system, where a texture appears when the object can be moved and another texture appears while the player is moving the object, thus making it easier to use the system when the mouse is not visible. Also included in the asset is an example scene, already configured, where the system is demonstrated in a simple way, to facilitate the understanding of its operation and use.

It may be necessary to lock the rotation of your player's camera while he is rotating objects. To facilitate the integration of the system with other player systems, the code has a static variable, which indicates when an object is being rotated. The name of this variable is '**rotatingObject**', which is a variable of type bool, which can be consulted easily according to the command shown in the following figure.



```
public bool getRotationStatus;

void Update() {
    getRotationStatus = MSMoveObjects.rotatingObject;
}
```

In this way, the rotation of the player's camera can be deactivated when an object is being rotated, to make the system easier to deal with.

Recommendations

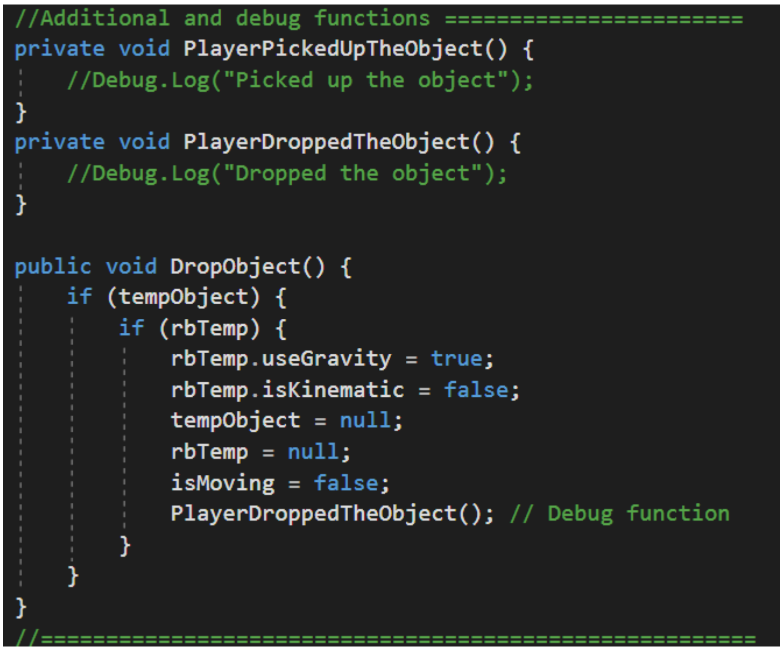
It is advised to leave the player and all their child objects in the '**IGNORE RAYCAST**' layer so that the code works correctly.

It is advisable to leave movement types in "**AddForce**" or "**Velocity**" because they behave correctly with external components.

Other functions

The code has a public function called **DropObject**, which can be called using other codes. If this function is called, the player will immediately release the object he is moving.

The code also has two functions, which are called when the player picks up an object, or when he drop an object, allowing debugging the system. The functions are shown in the following image.



```
//Additional and debug functions ========================
private void PlayerPickedUpTheObject() {
    //Debug.Log("Picked up the object");
}
private void PlayerDroppedTheObject() {
    //Debug.Log("Dropped the object");
}

public void DropObject() {
    if (tempObject) {
        if (rbTemp) {
            rbTemp.useGravity = true;
            rbTemp.isKinematic = false;
            tempObject = null;
            rbTemp = null;
            isMoving = false;
            PlayerDroppedTheObject(); // Debug function
        }
    }
}
//=========================================================
```

<u>If you have any questions, please contact us:</u> marcos11-24@hotmail.com