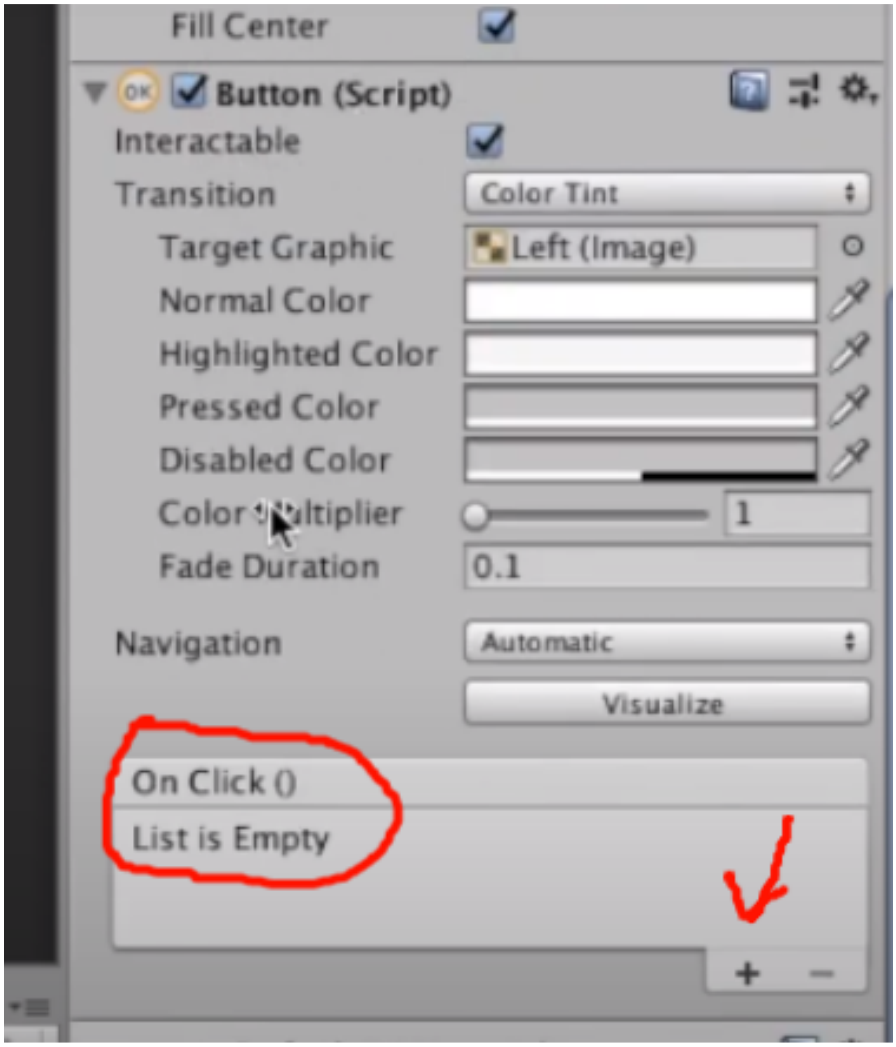
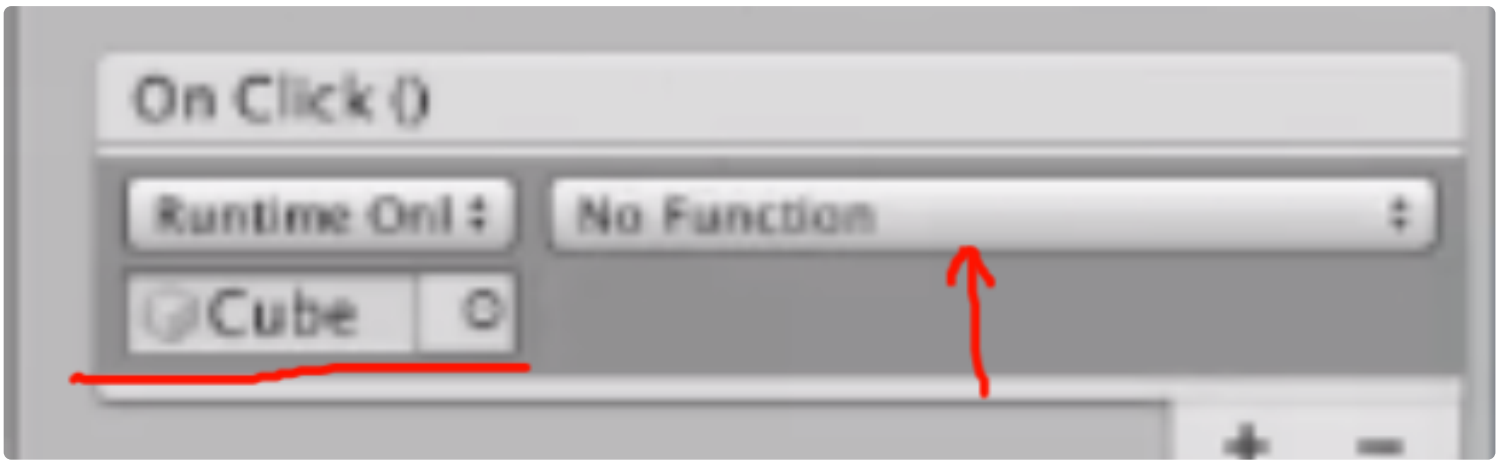
GameObject --> UI --> Select **Canvas**

In Canvas, **Panel** is also required as child of canvas for inserting UI elements.

<u>Interactable UI Buttons</u>





GameObject selected by drag and drop. Function will defined in movement script as created earlier.

```
0 references
void MoveLeft()
{
    transform.position -= new Vector3(xSpeed, 0, 0);
}
```

But it wont be shown on function tab because it is private. It has to be public by just adding public before void.

```
0 references
public void MoveLeft()
{
    transform.position -= new Vector3(xSpeed, 0, 0);
}
```

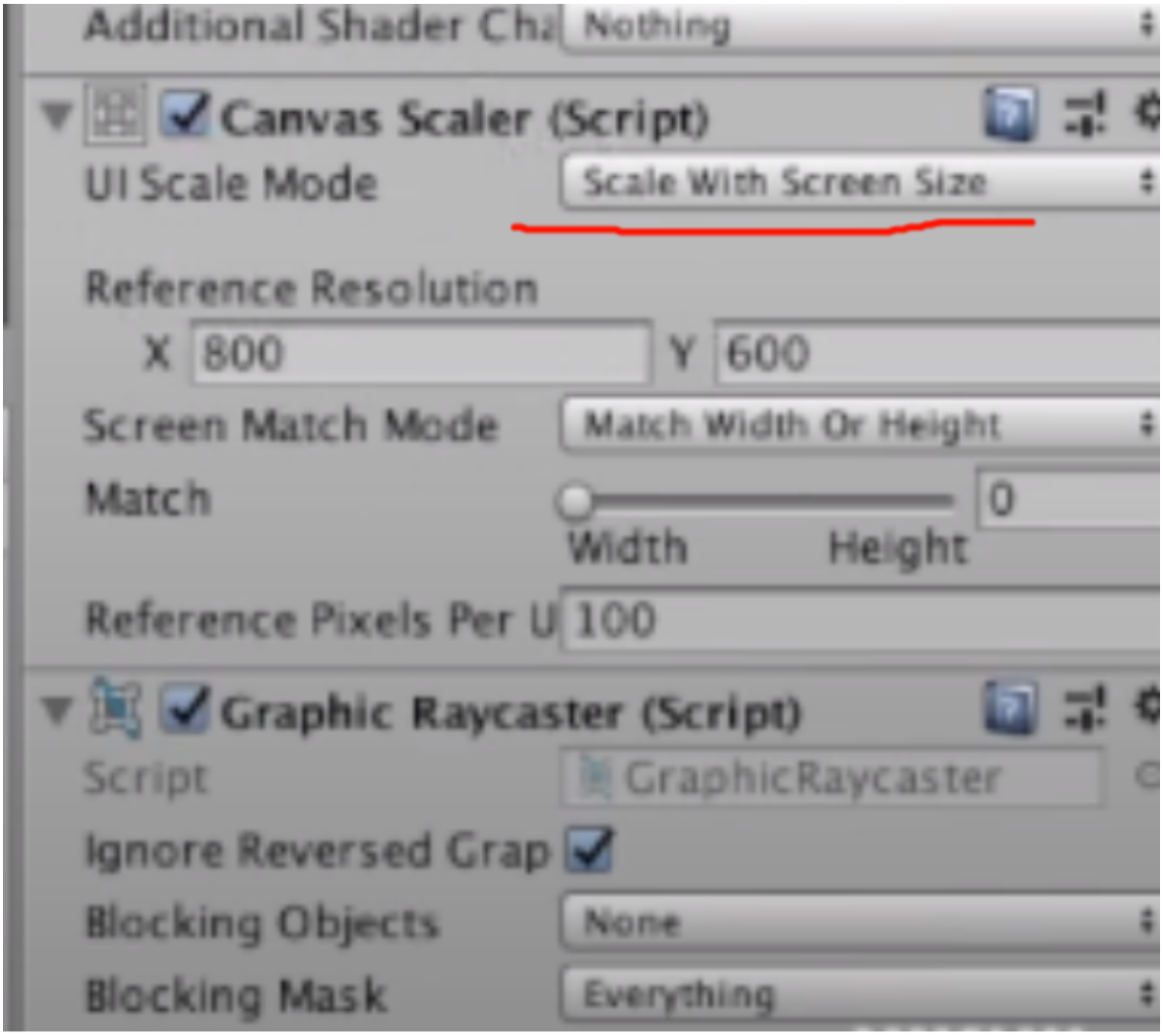
```
0 references
public void MoveLeft()
{
    transform.position -= new Vector3(xSpeed, 0, 0);
}

0 references
public void MoveRight()
{
    transform.position += new Vector3(xSpeed, 0, 0);
}

0 references
public void MoveUp()
{
    transform.position += new Vector3(0, ySpeed, 0);
}

0 references
public void MoveDown()
{
    transform.position -= new Vector3(0, ySpeed, 0);
}
}
```

Similarly other functions created for OnClick actions.



UI scale with screen size automatically.

Continuous movement on clicking buttons

Continous movement on pressing comes using **Event System**.

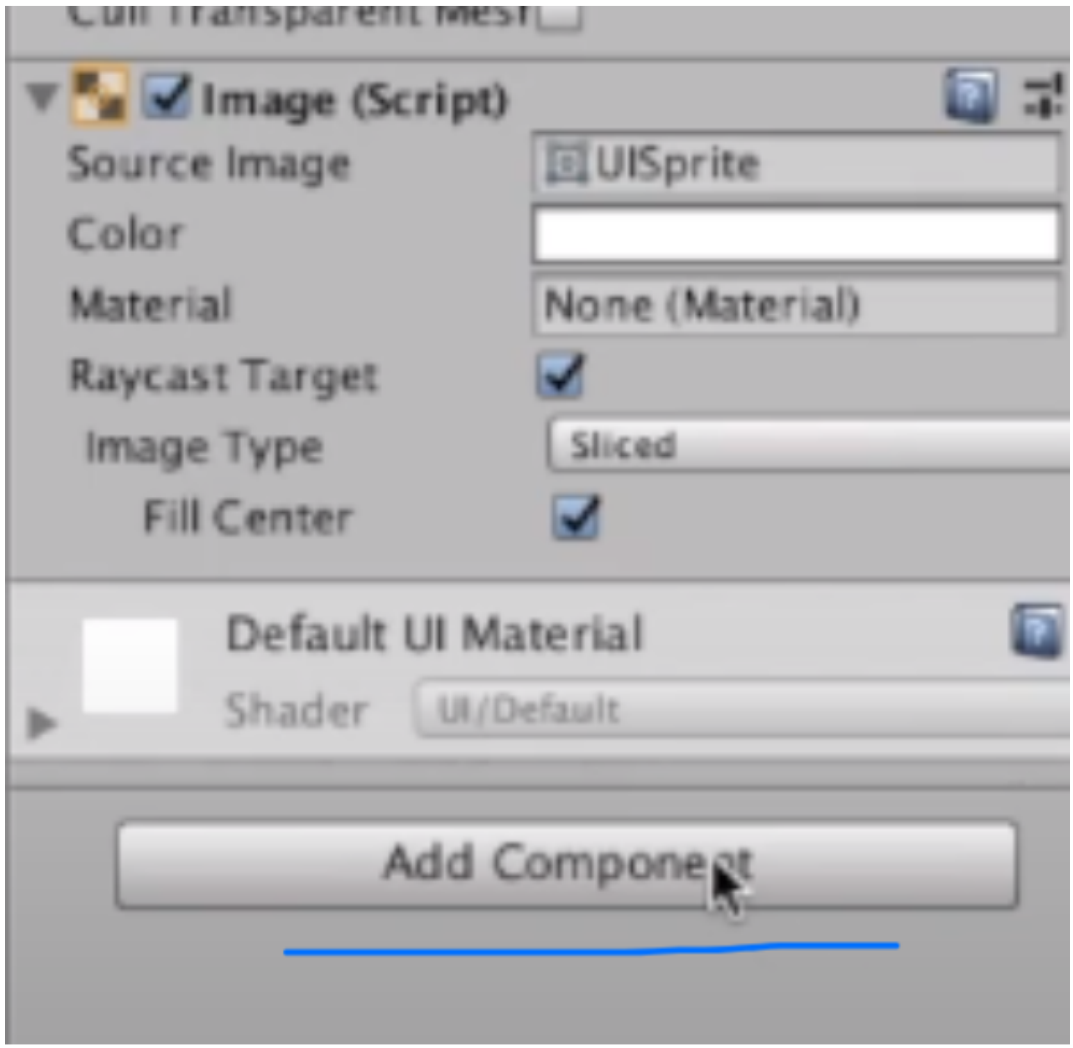
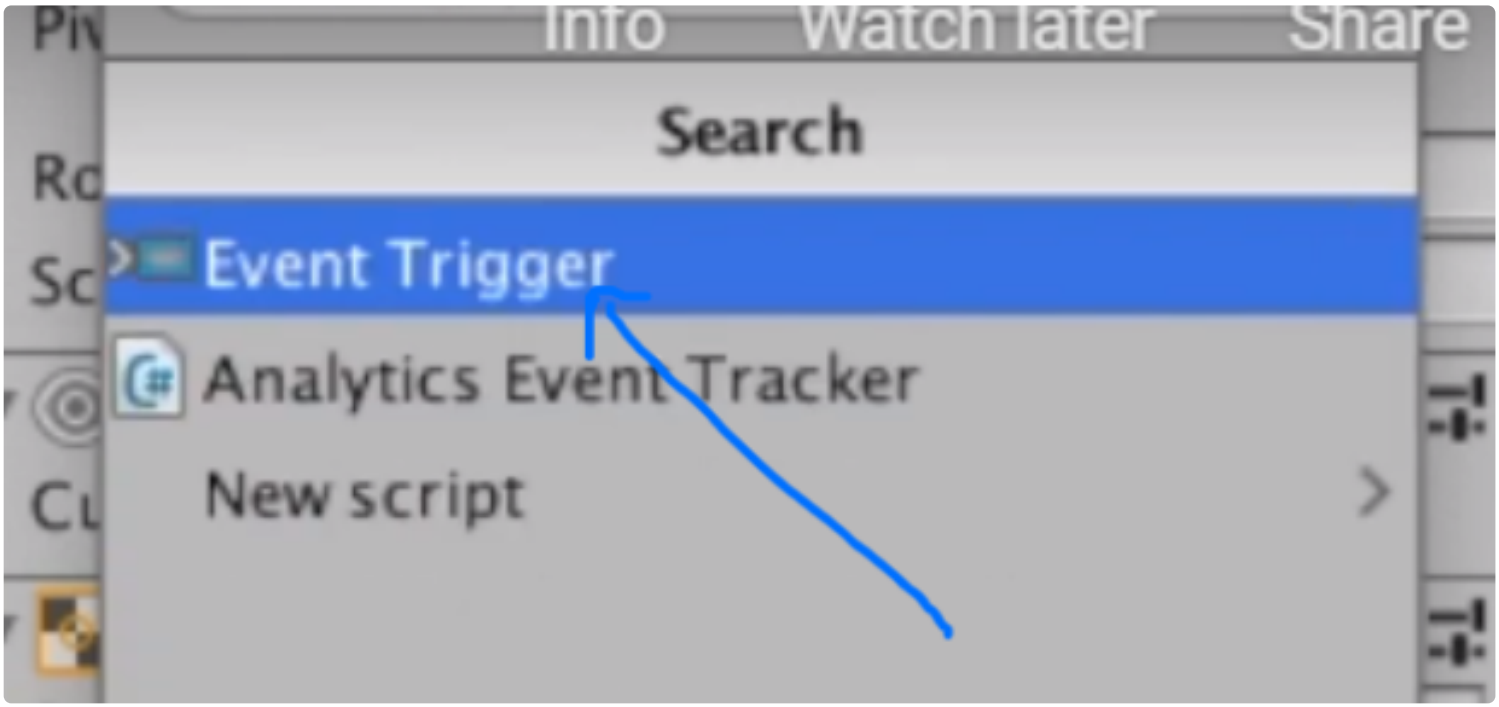
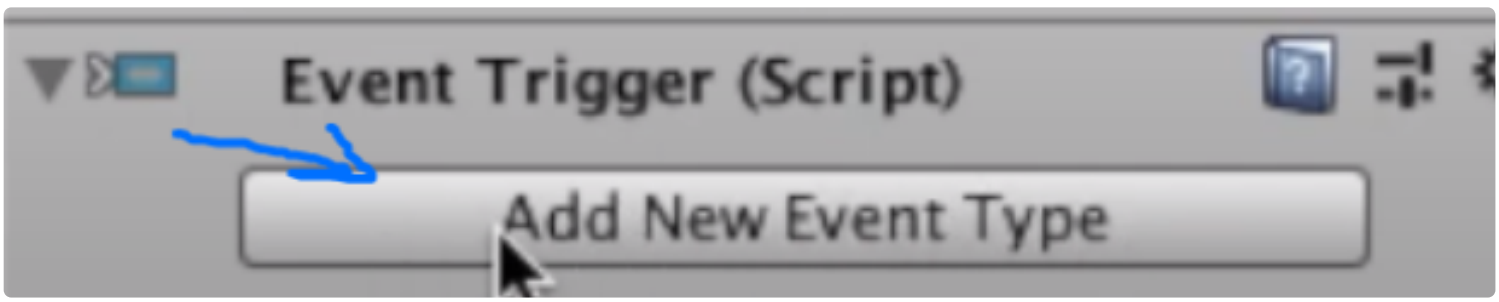


Image (Script)
Source Image
UISprite
Color
Material
None (Material)
Raycast Target
Image Type
Sliced
Fill Center
Default UI Material
Shader
UI/Default
Add Component


Info
Watch later
Share
Search
Event Trigger
Analytics Event Tracker
New script


Event Trigger (Script)
Add New Event Type

Click on Add New Event Type and Select **PointerDown** and **PointerUp**

Here, PointerDown means press and hold and PointerUp means when left pressing the button.

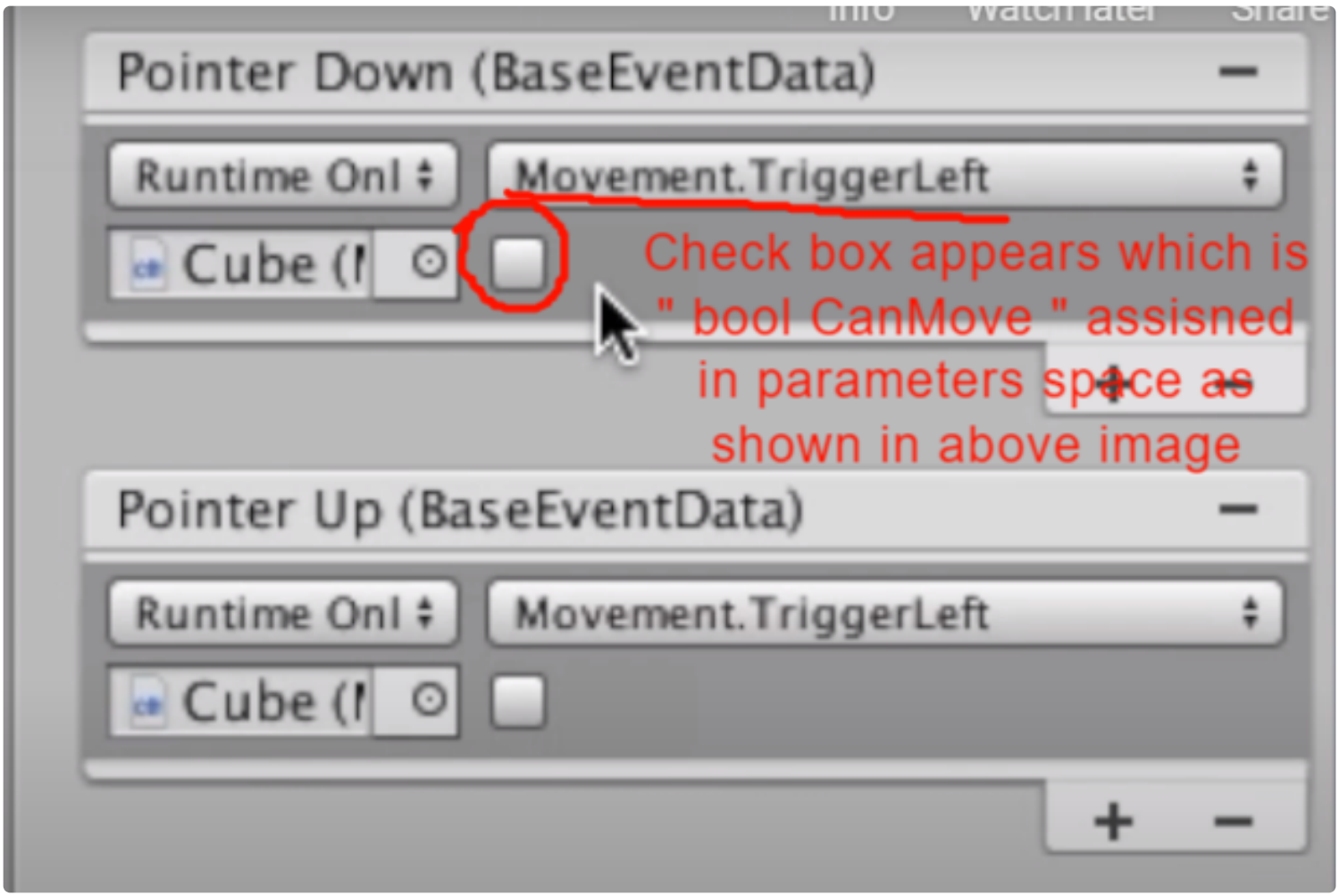
```
public class Movement : MonoBehaviour
{
    2 references
    public float xSpeed = 0.1f;
    2 references
    public float ySpeed = 0.05f;

    0 references
    public bool isMovingLeft = false;
```

```
    //Move the object in left
    if(isMovingLeft)
    {
        MoveLeft();
    }

    //Move the object up
    if(Input.GetKey(KeyCode.W))
    {
        MoveUp();
    }
```

```
0 references
public void TriggerLeft(bool canMove)
{
    isMovingLeft = canMove;
}
```



+

Similarly, write code for other buttons.

```
2 references
public bool isMovingLeft = false;
0 references
public bool isMovingRight = false;
0 references
public bool isMovingUp = false;
2 references
public bool isMovingDown = false;
```

Logic behind continous movement :

```
public class Movement : MonoBehaviour
{   public float xSpeed = 0.01f;
    public float ySpeed = 0.01f;
    public bool isMovingUp = false;
    public bool isMovingDown = false;
    public bool isMovingLeft = false;
    public bool isMovingRight = false;

    void Start()
    {

    }
    void Update()
    {
        movement();
    }
```

Variable of boolean data type declared.

```
if(isMovingUp)
{
    transform.position += new Vector3(0, ySpeed, 0);
}
```

```
}
public void TriggerUp(bool canMove)
{
    isMovingUp = canMove;
}
public void TriggerDown(bool canMove)
{
    isMovingDown = canMove;
}
public void TriggerLeft(bool canMove)
{
    isMovingLeft = canMove;
}
public void TriggerRight(bool canMove)
{
    isMovingRight = canMove;
}
```

TriggerUp function created with parameter "bool canMove"