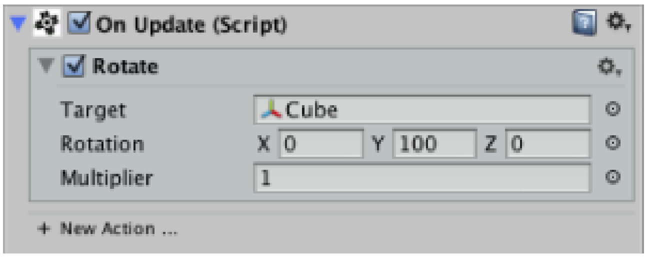
Tutorial 2: Rotate and Set Color

Now we'll do something a little more interesting: we'll put a cube on the screen, will rotate it and will change its color when you click with the mouse on it.

1. The first step is to create a cube with the menu *GameObject*> *3D Object*> *Cube.*
2. Adjust the camera scene (Main Camera) to approach the cube you just created, you add a light *(Component*> *Rendering*> *Light).* Here will be useful also use the *GameObject* > *Align With View* option for the camera fits what we have on the *Scene* window.
3. Again select the cube and in the Inspector window add an *On Update* component, which is a program that runs on every frame of the game.
4. Add a *Rotate* action to this program. Adjust the Target property to point to the created cube (this can be done by dragging the item from the *Hierarchy* window to the field or by selecting the object by clicking the small circle that appears field to the right). And also change the value of the *Rotation* property worth 100.



On Update (Script)

Rotate

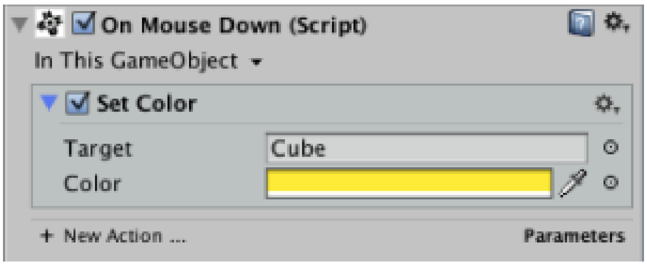
Target Cube

Rotation X 0 Y 100 Z 0

Multiplier 1

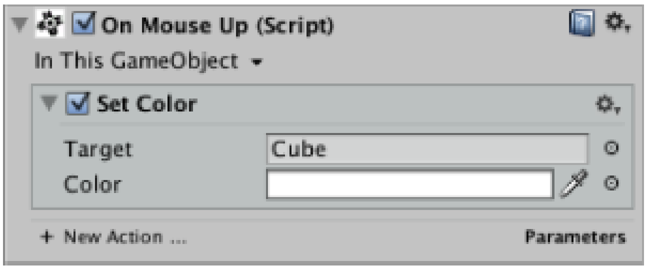
5. Pressing *Play* you should see our cube rotating.
6. Now create a second program *On Mouse Down* to the cube, which will tell you what actions you want to perform when you click on the hub.
7. Add a Set Color action to the cube and again adjust the Target of action to be the cube itself. Also modify the value of

the *Color* property to change the color cube that we like.



☑ On Mouse Down (Script)
In This GameObject
☑ Set Color
Target: Cube
Color: 🟨
+ New Action ...
Parameters

8. If you press *Play* again we should see our cube rotating but now also it will change its color if you click on it.
9. We would like to return the cube to its original color when you release the mouse button, so finally add another program *On Mouse Up*.
10. Add again a *Set Color* action and adjust the *Target* of action but this time without changing the color (which should be the default white).



☑ On Mouse Up (Script)
In This GameObject
☑ Set Color
Target: Cube
Color: ⬜
+ New Action ...
Parameters

And that's it. Pressing *Play* our full schedule should have already done something for which necessarily would have needed to write some code. GameFlow made this possible without it being necessary to exit the editor of Unity.

As we have seen is really easy to create small programs that perform actions on objects in the scene, but this is just a small sample of what GameFlow can do. As we'll explain in next tutorials, by combining event programs, actions, conditions and other components is possible to build complex logical game and even sophisticated prefab ready for others to use them as black boxes, without having to understand the internal programs.