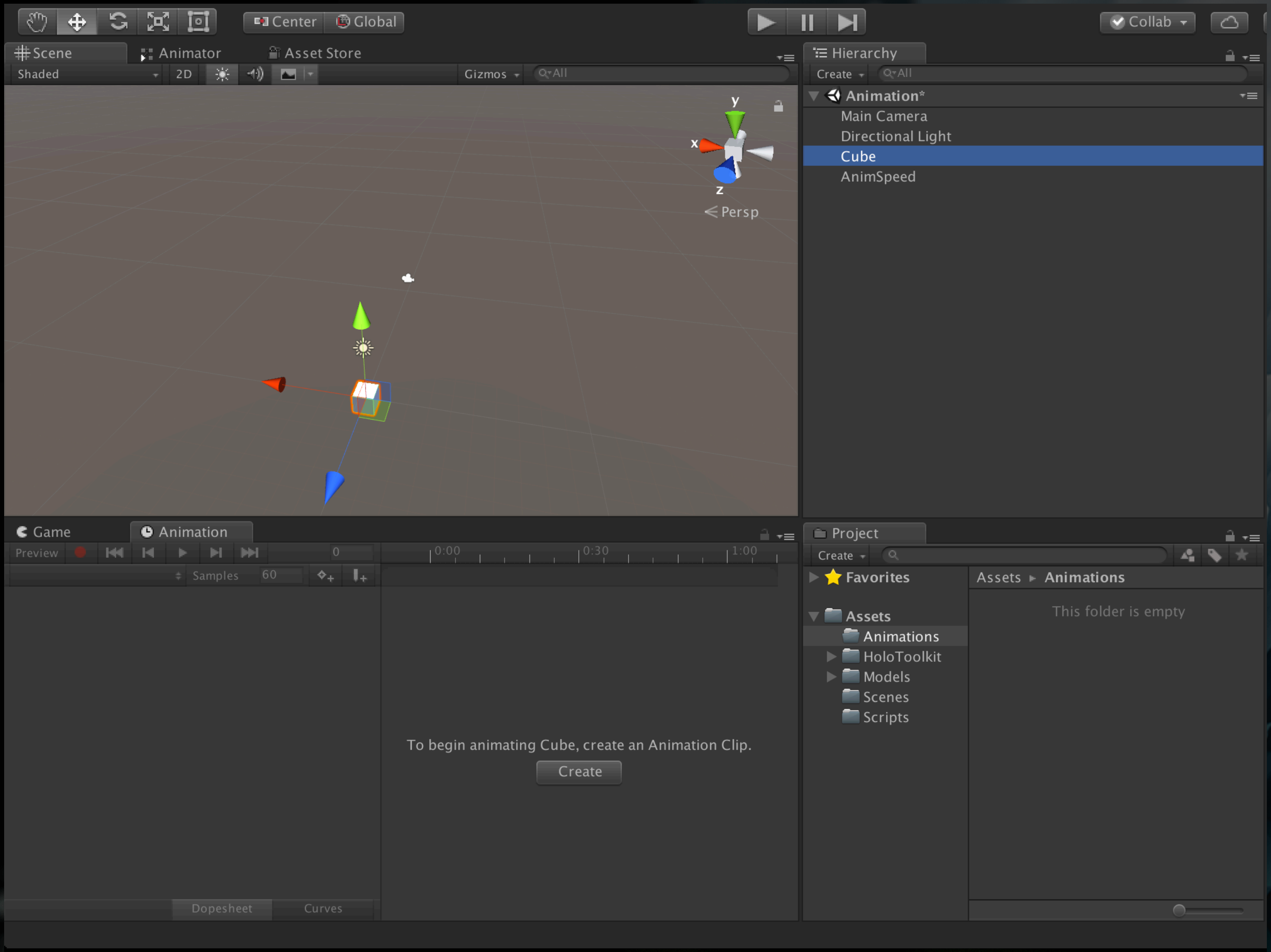
Dynamically Changing Speed of Animations

Some of you might need to change the speed of an animation while your apps are running. This is easy to do in a custom script and can open up some exciting creative possibilities.

Whenever you create a new Animation...



This folder is empty

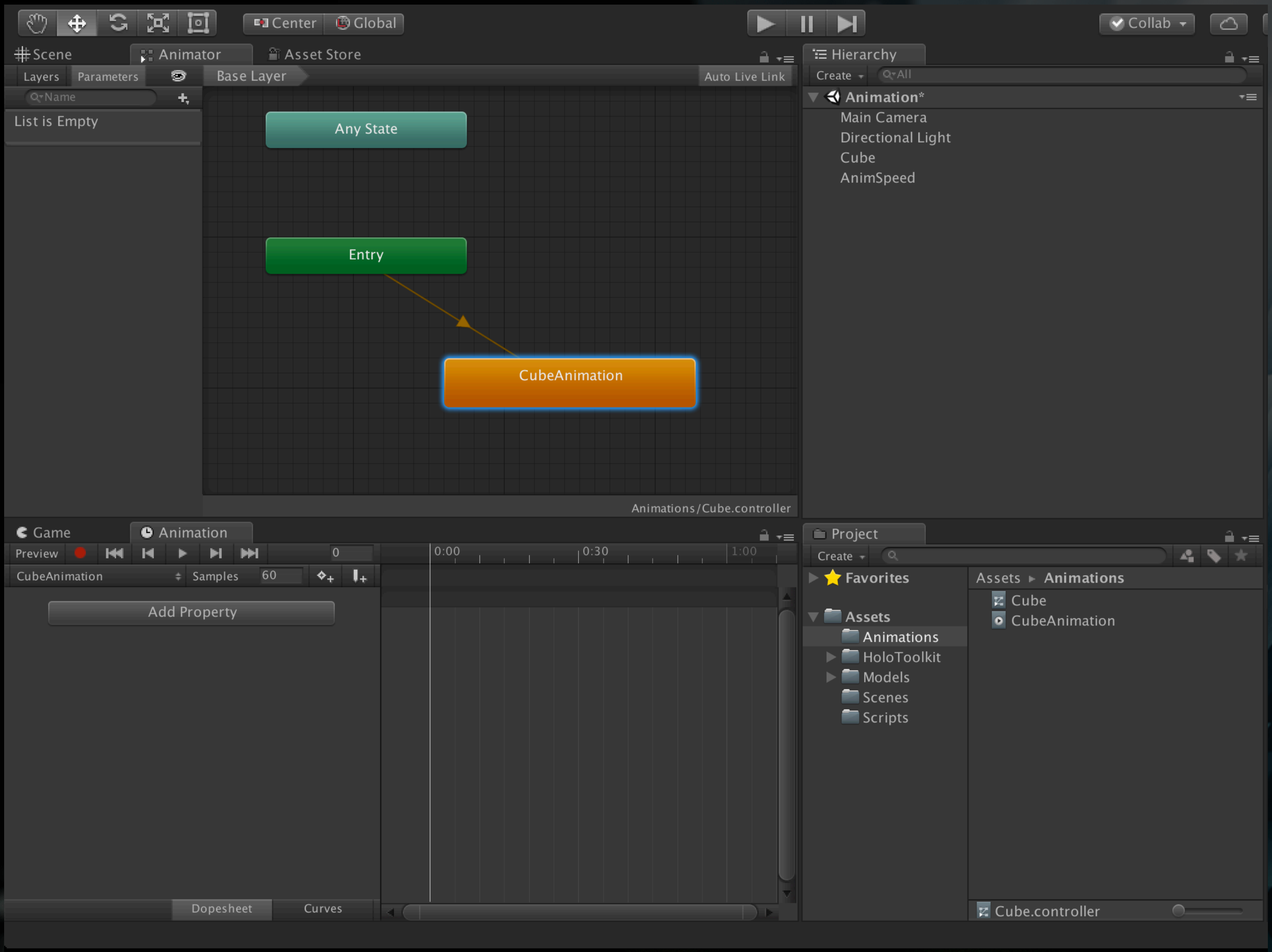
To begin animating Cube, create an Animation Clip.

Create

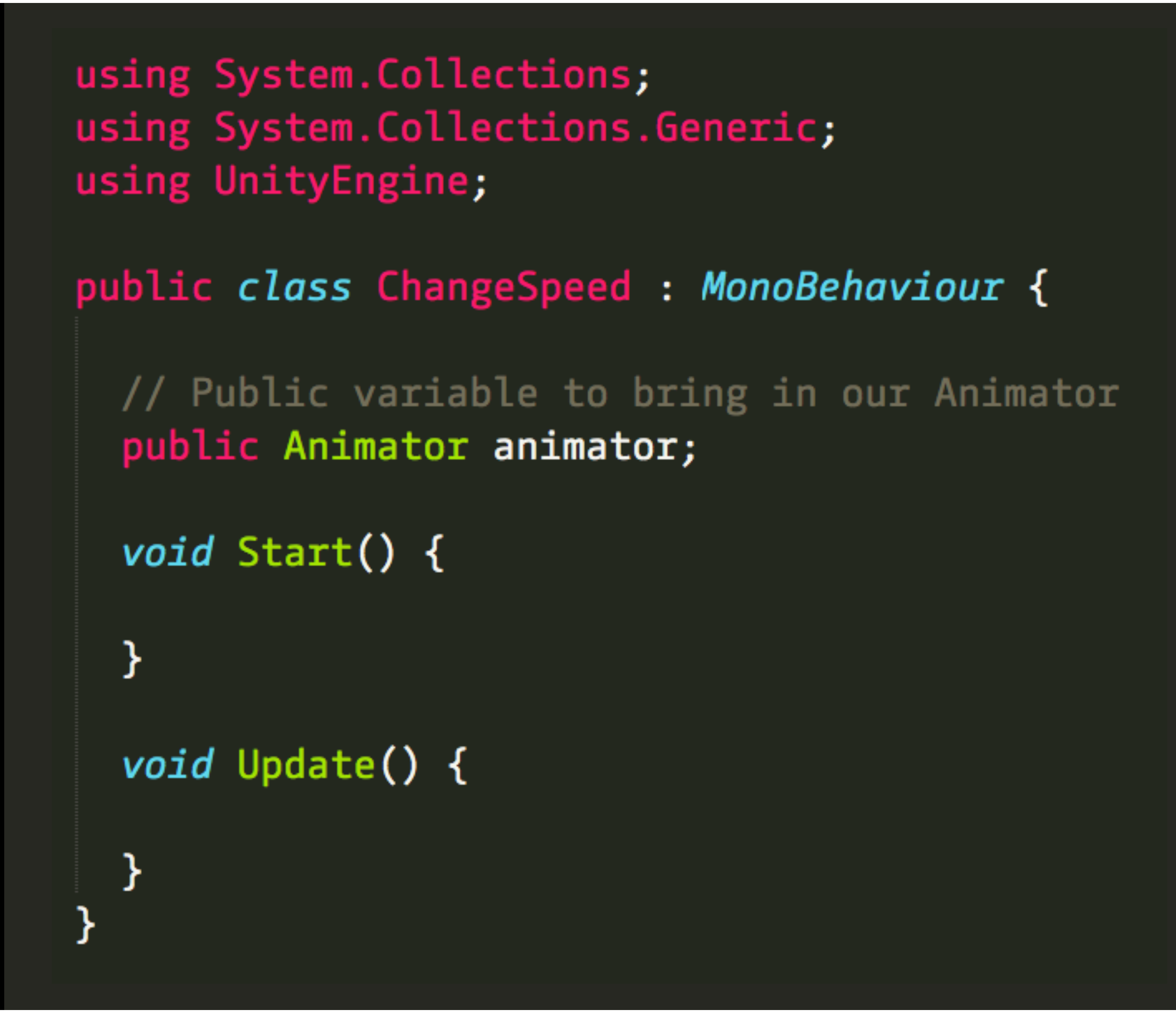
...Unity creates an *Animation* and and *Animator*.

The *Animator* gets attached to your GameObject and manages the *Animations* you create.

You can use the *Animator* to control things like speed.



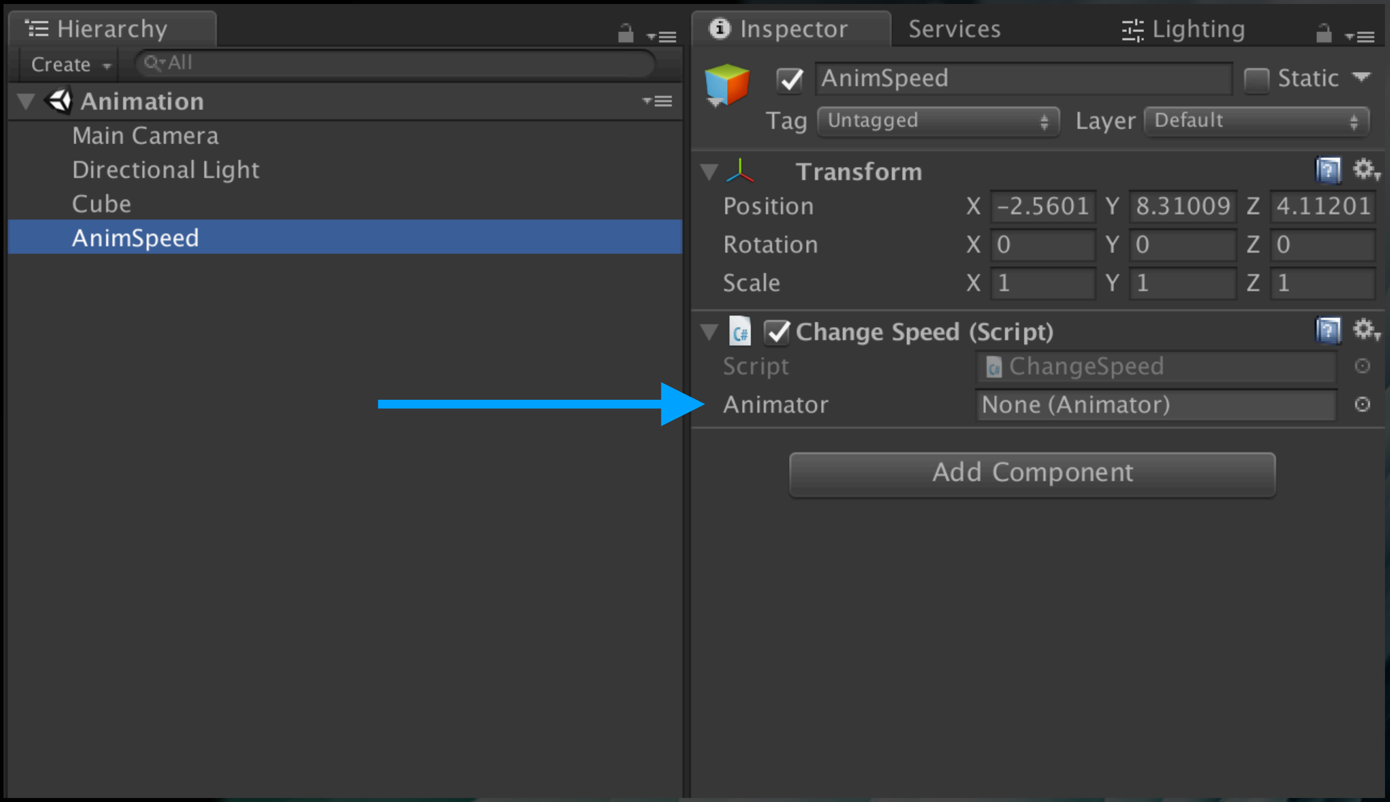
Create a script called ChangeSpeed.cs and add a public Animator variable.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ChangeSpeed : MonoBehaviour {

    // Public variable to bring in our Animator
    public Animator animator;

    void Start() {

    }

    void Update() {

    }
}
```

Create an empty game object to hold your script (in this example, I've named it *AnimSpeed*) and add your script to it.

You should see a slot for your Animator (because you made this variable public)



☑ AnimSpeed

Tag Untagged Layer Default

Transform

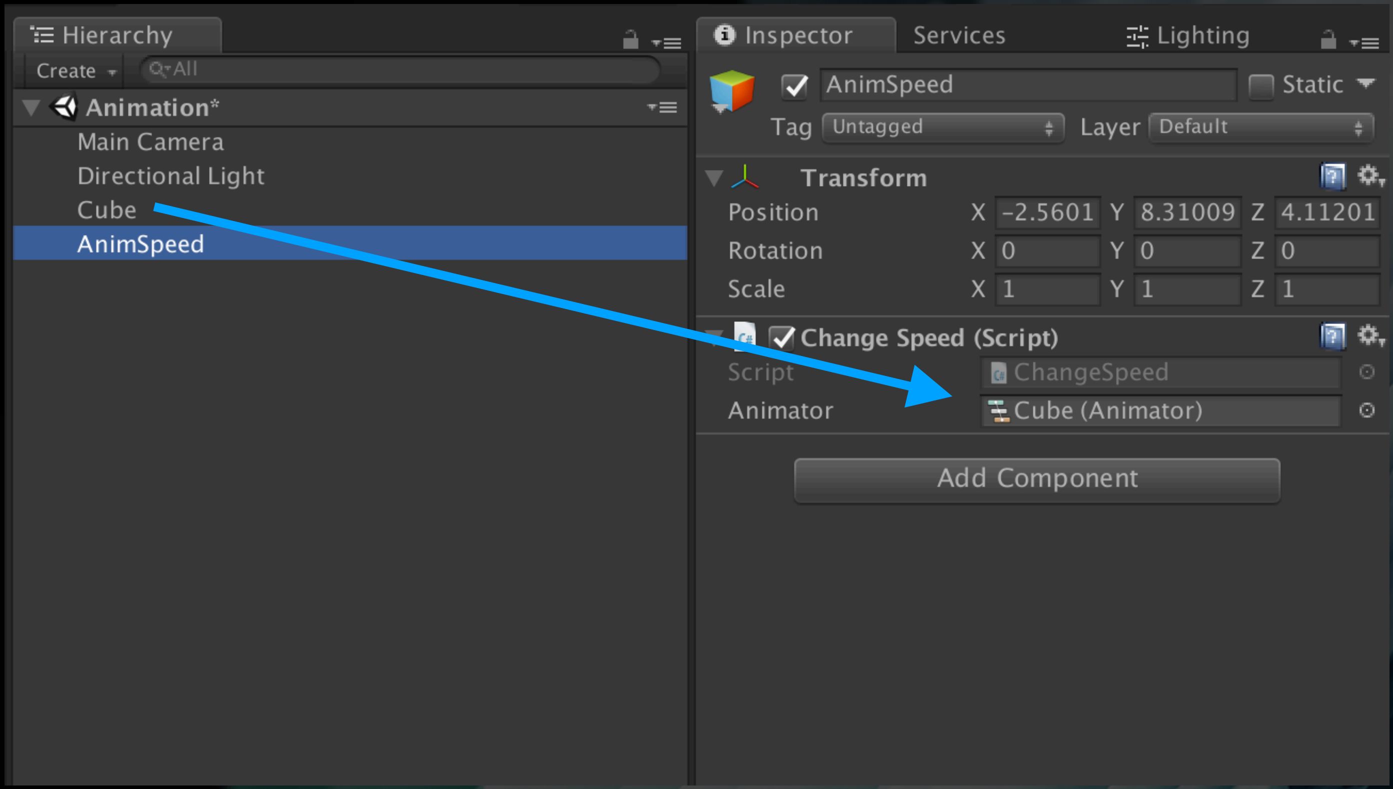
Position	X	-2.5601	Y	8.31009	Z	4.11201
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

☑ Change Speed (Script)

Script ChangeSpeed

Animator None (Animator)

Find the GameObject that has the animation you want to control. Drag it from your scene onto this slot and it will find the Animator component automatically.



☑ AnimSpeed □ Static

Tag Untagged Layer Default

Transform

Position X -2.5601 Y 8.31009 Z 4.11201

Rotation X 0 Y 0 Z 0

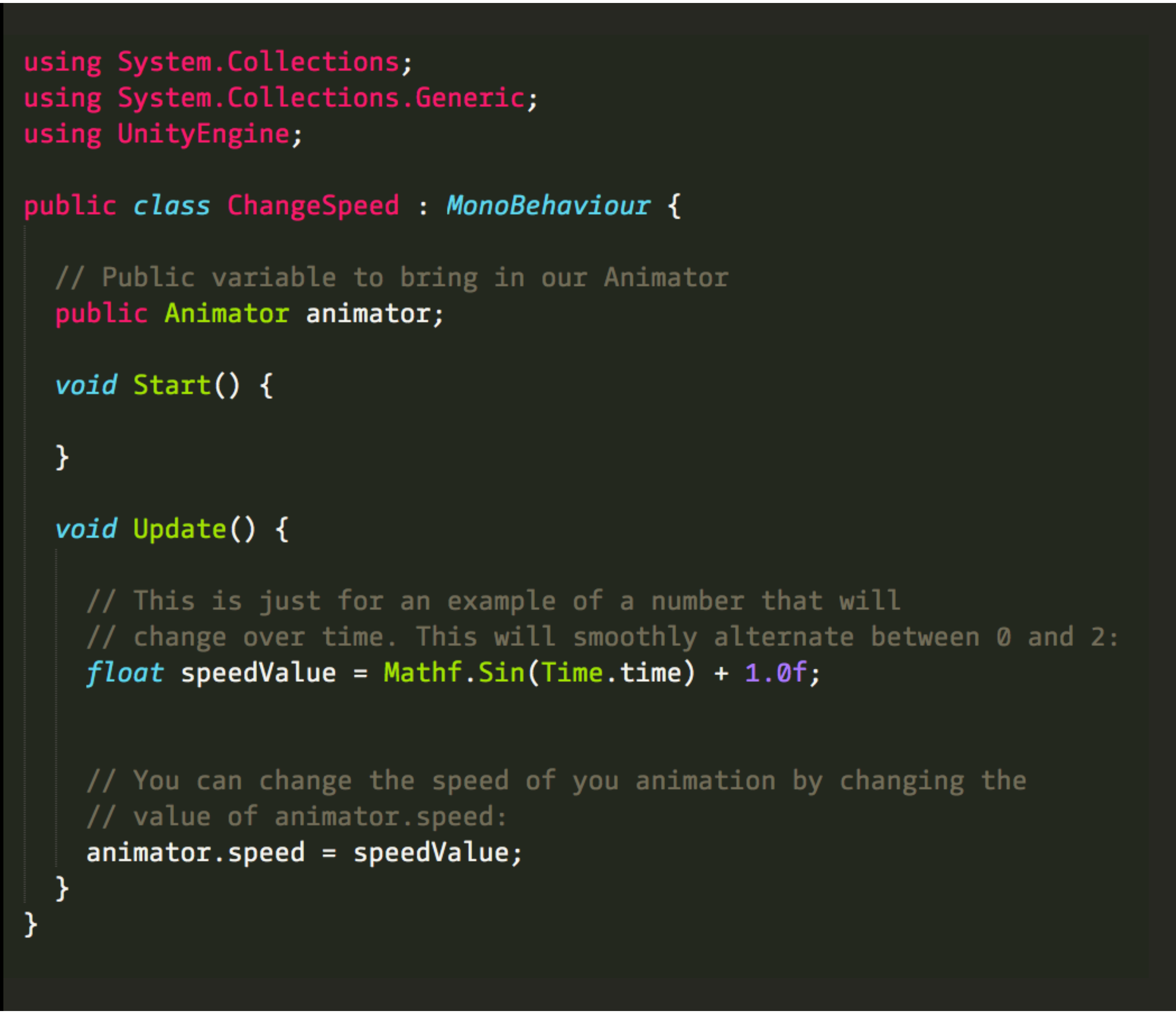
Scale X 1 Y 1 Z 1

☑ Change Speed (Script)

Script ChangeSpeed

Animator Cube (Animator)

You can change the speed of your animation by changing the ‘speed’ property of your Animator variable.

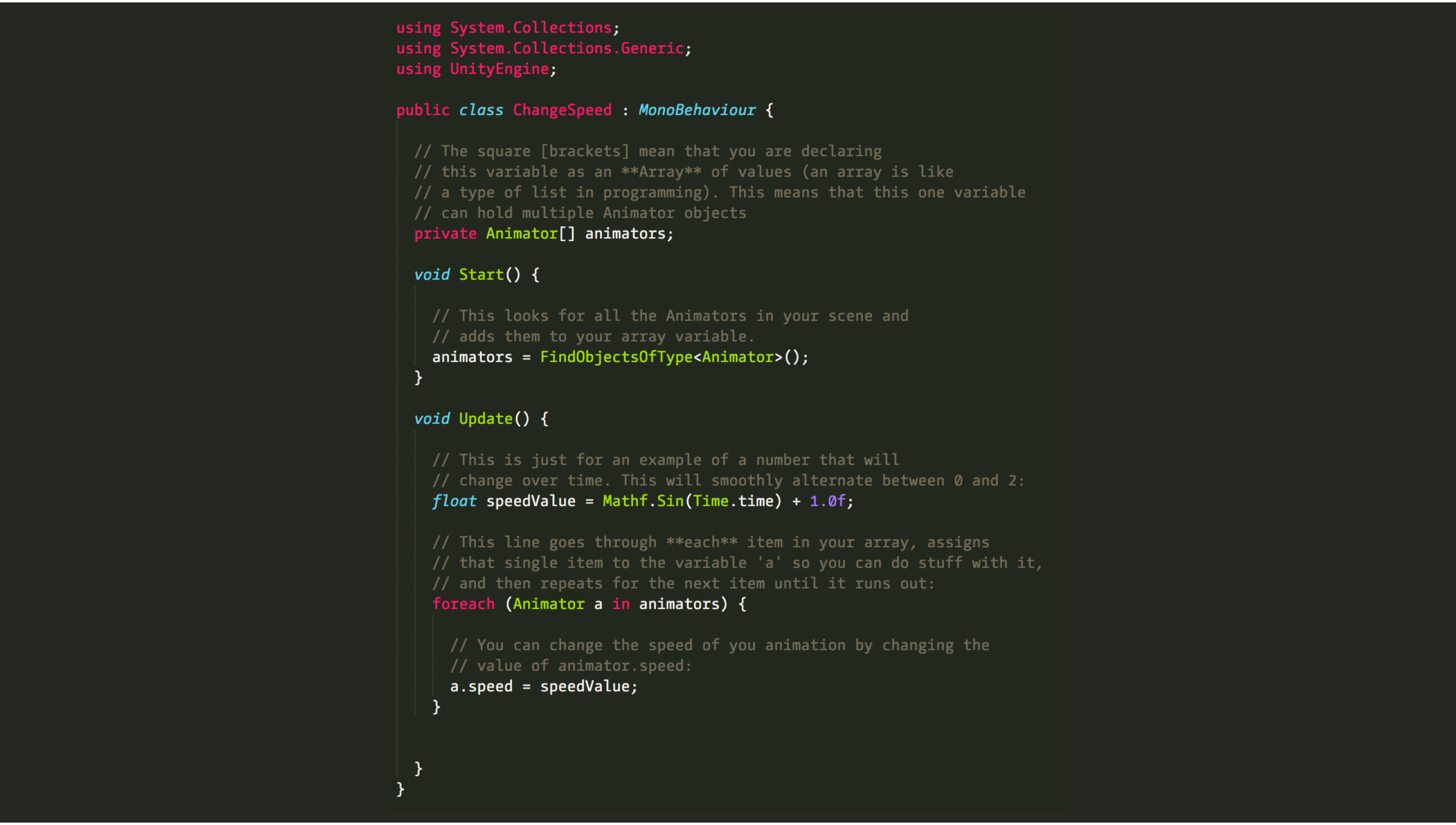


```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ChangeSpeed : MonoBehaviour {

  // Public variable to bring in our Animator
  public Animator animator;

  void Start() {

  }

  void Update() {

    // This is just for an example of a number that will
    // change over time. This will smoothly alternate between 0 and 2:
    float speedValue = Mathf.Sin(Time.time) + 1.0f;


    // You can change the speed of you animation by changing the
    // value of animator.speed:
    animator.speed = speedValue;
  }
}
```

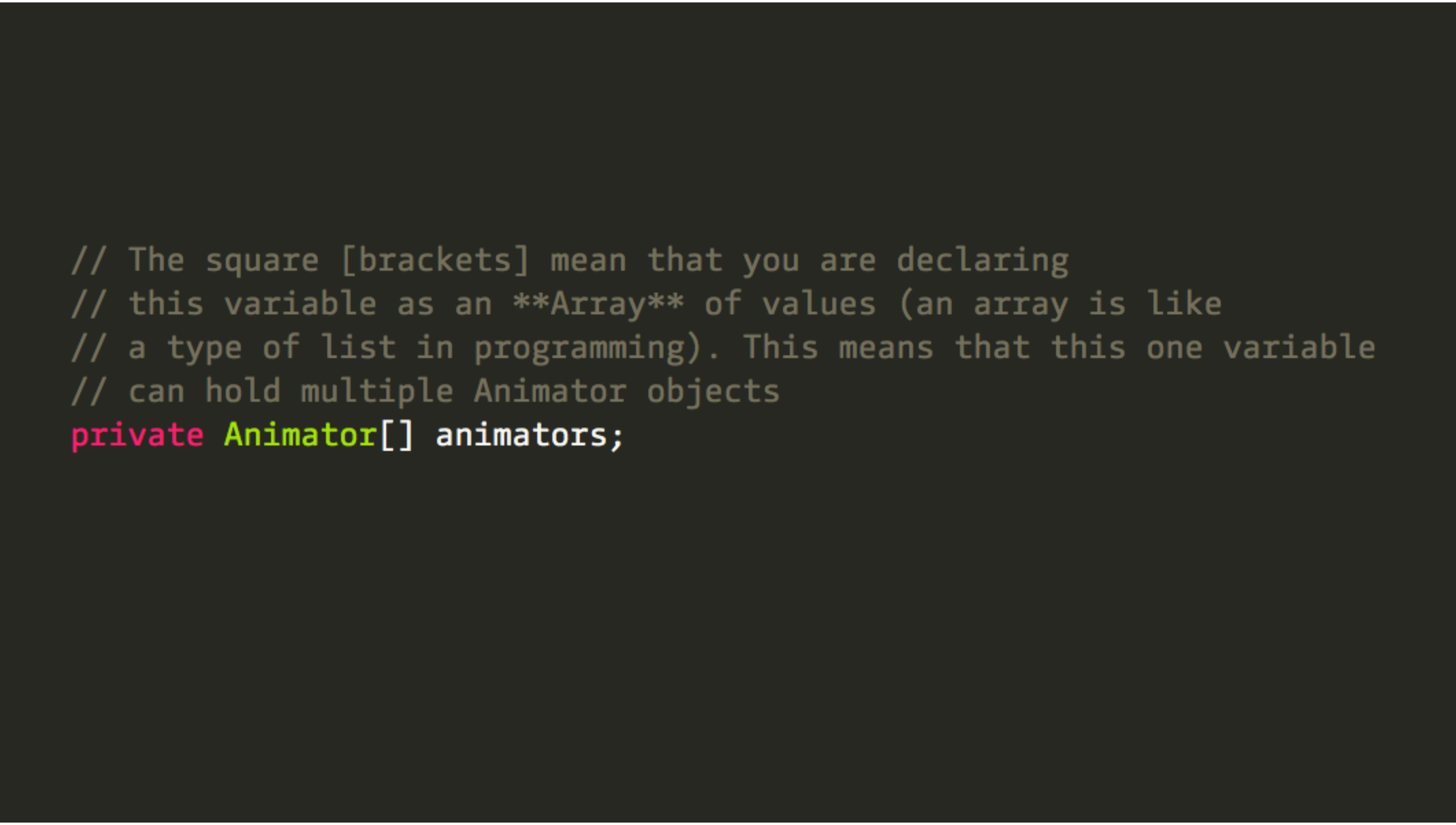
Changing Multiple Animations Simultaneously

If you have more than a few animations in your scene, it may become tedious to do this to each animation individually, especially if you want to change them by the same amount.

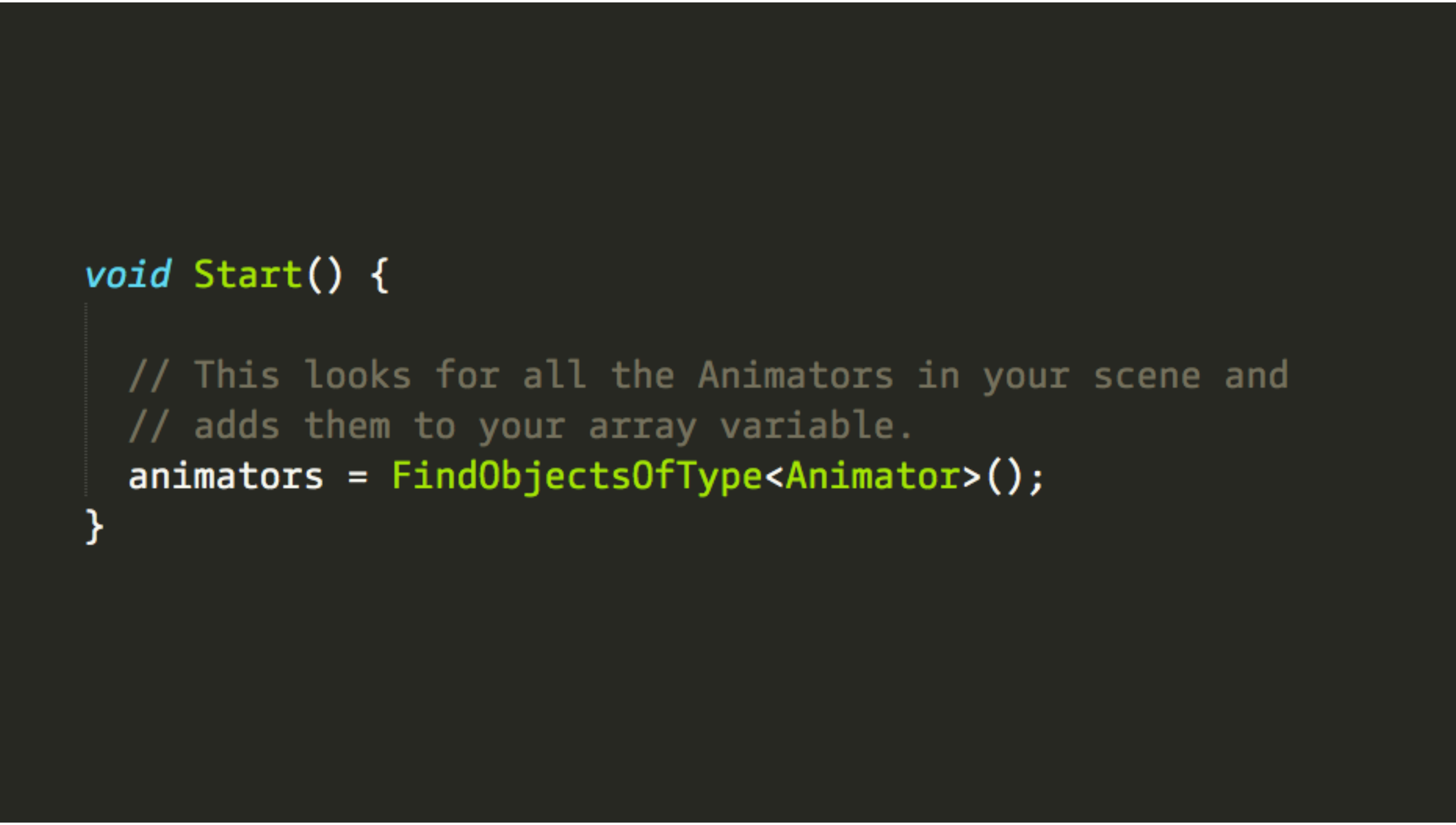
Instead of using a public variable and dragging in your GameObject, you can use code to find all the Animators in your scene.



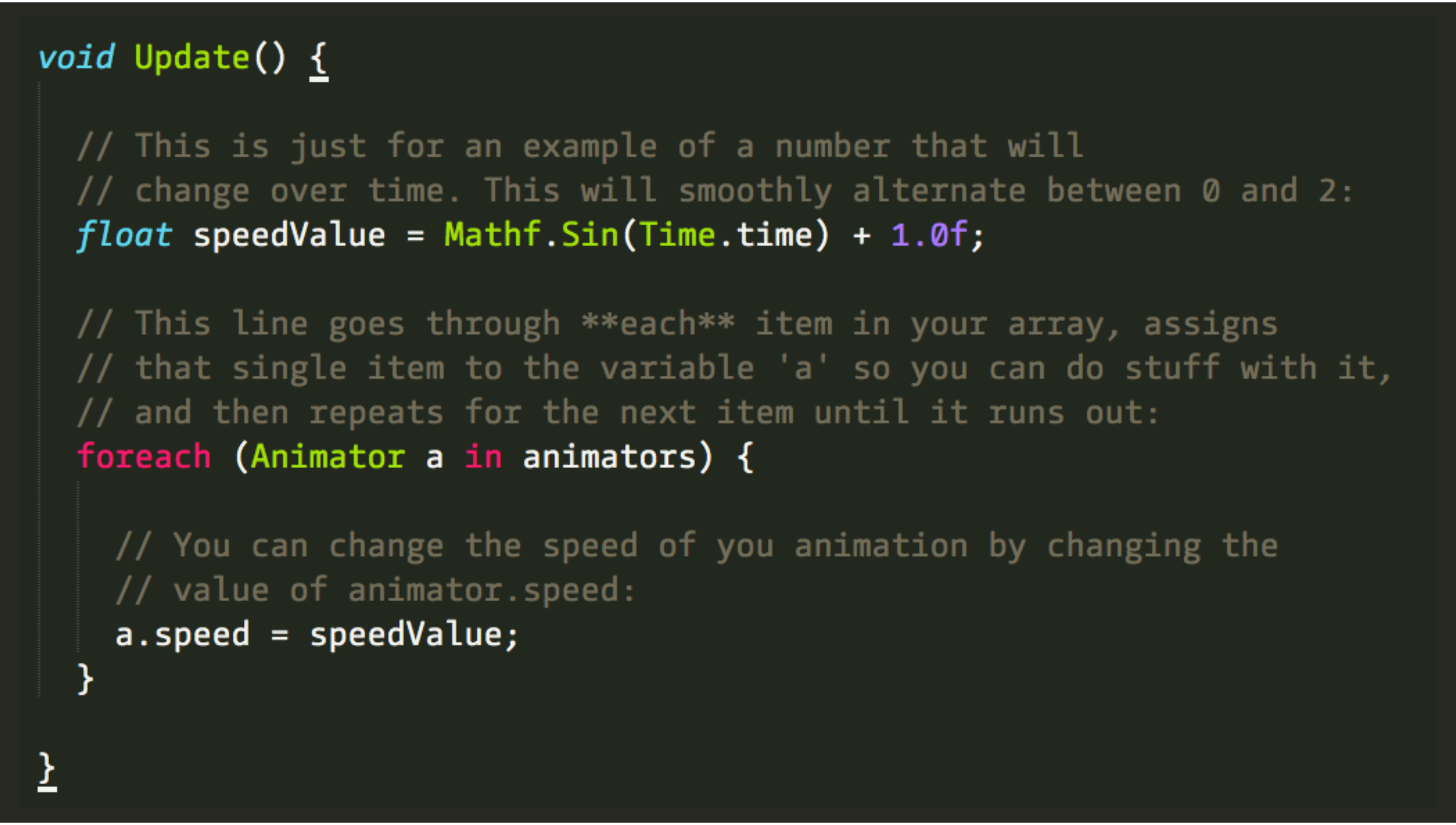
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ChangeSpeed : MonoBehaviour {

  // The square [brackets] mean that you are declaring
  // this variable as an **Array** of values (an array is like
  // a type of list in programming). This means that this one variable
  // can hold multiple Animator objects
  private Animator[] animators;

  void Start() {

    // This looks for all the Animators in your scene and
    // adds them to your array variable.
    animators = FindObjectsOfType<Animator>();
  }

  void Update() {

    // This is just for an example of a number that will
    // change over time. This will smoothly alternate between 0 and 2:
    float speedValue = Mathf.Sin(Time.time) + 1.0f;

    // This line goes through **each** item in your array, assigns
    // that single item to the variable 'a' so you can do stuff with it,
    // and then repeats for the next item until it runs out:
    foreach (Animator a in animators) {

      // You can change the speed of you animation by changing the
      // value of animator.speed:
      a.speed = speedValue;
    }


  }
}
```



```
// The square [brackets] mean that you are declaring
// this variable as an **Array** of values (an array is like
// a type of list in programming). This means that this one variable
// can hold multiple Animator objects
private Animator[] animators;
```



```
void Start() {

  // This looks for all the Animators in your scene and
  // adds them to your array variable.
  animators = FindObjectsOfType<Animator>();
}
```



```
void Update() {

  // This is just for an example of a number that will
  // change over time. This will smoothly alternate between 0 and 2:
  float speedValue = Mathf.Sin(Time.time) + 1.0f;

  // This line goes through **each** item in your array, assigns
  // that single item to the variable 'a' so you can do stuff with it,
  // and then repeats for the next item until it runs out:
  foreach (Animator a in animators) {

    // You can change the speed of you animation by changing the
    // value of animator.speed:
    a.speed = speedValue;
  }

}
```