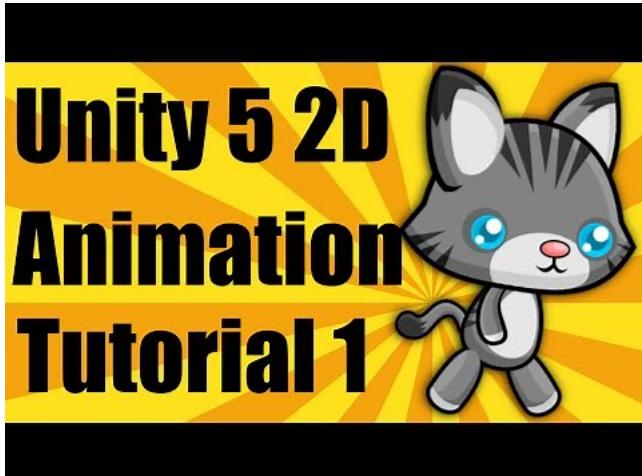


GAME PRODUCTION – Tutorial 3

Today's Sprite Animation Tutorial:

<https://www.youtube.com/watch?v=VAyWk0elqRc>



[Unity 5 2d Animation Tutorial - Part 1](#)

Unity 5 2d Animation Tutorial - Part 1: Welcome to my first Unity 5 2d Animation Tutorial where i show you how easy it is to create 2d sprite based animation using ...

www.youtube.com

2. Check out these 16 sites that contain free art assets you can use. You might find something that will help with your project.

<http://v-play.net/game-resources/16-sites-featuring-free-game-graphics>

3. This is a fairly extensive Unity tutorial that will help you revise and build a 2D game. It will take a while to work through but well worth it!



The screenshot shows a Unity Learn course page for "Beginning 2D Game Development". At the top, there are navigation icons and a search bar. Below the header, there's a preview image of a 2D game scene featuring a character standing next to a large, metallic robot. The scene is set in a lush, green environment with various plants and rocks. In the top left corner of the preview, there are five red hearts and a small number "1472". The title "Beginning 2D Game Development" is displayed prominently below the image. Below the title, it says "Course • Beginner • 17 Hours 45 Mins" and a brief description: "Begin your 2D journey in Unity, start with the 2D Game Kit and move on through to make your first 2D Game from scratch!". At the bottom left of the preview area, there's a "Unity Technologies" logo.

4. Tile Map Tutorial:

https://www.youtube.com/watch?v=rylSV_nH8qw



TILEMAP IN UNITY

[TILEMAPS in Unity](#)

Learn how to use the Tilemap tools in Unity to easily create cool 2D levels. Watch TILESET in Photoshop:
<https://youtu.be/aaEEujLtsr8>
Download Tileset: [http ...](http...)

www.youtube.com

5. Work through the code examples in Chapter 23 – Functions and Parameters.
This will describe how to define and use functions in C# and Unity.

I've copied my code below.

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class CodeExamples : MonoBehaviour
5 {
6
7     public int numTimesCalled = 0;
8
9     void Awake()
10    {
11        print(Add(1.0f, 2.5f)); //prints "3.5"
12
13        print(Add(new Vector3(1,0,0), new Vector3(0,1,0) )); //prints "(1.0, 1.0, 0.0)"
14
15        Color colorA = new Color(0.5f, 1, 0, 1);
16        Color colorB = new Color(0.25f, 0.33f, 0, 1);
17        print (Add(colorA,colorB)); // prints "RGBA(0.750, 1.000, 0.000, 1.000)"
18
19        print( Add(1));           // 1
20        print (Add(1,2));        // 3
21        print (Add(1,2,3));      // 6
22        print (Add(1,2,3,4));    // 10
23
24        print (Fac (-1));   // 0
25        print (Fac (0));    // 1
26        print (Fac (5));    // 120
27
28    } //Awake
29
```

```
--  
30     int Fac(int n)  
31     {  
32         int result = 0; //default for negative numbers  
33  
34         if (n < 0)  
35             result = 0;  
36         else if (n == 0)  
37             result = 1;  
38         else  
39             result = n * Fac(n-1); // recursive definition n * (n-1) * (n-2) * (n-3) * .... * 1  
40  
41         return (result);  
42     } //Fac  
43  
44     //overloading the "Add" function  
45     float Add (float f0, float f1) //Adds 2 floats  
46     {  
47         return (f0 + f1);  
48     } //float Add (float f0, float f1)  
49  
50     Vector3 Add (Vector3 v0, Vector3 v1) //Adds 2 3D vectors  
51     {  
52         return (v0 + v1);  
53     } //Add (Vector3f v0, Vector3f v1)  
54  
55     Color Add (Color c0, Color c1) //Adds 2 colors  
56     {  
57         float r, g, b, a;  
58         r = Mathf.Min(c0.r + c1.r, 1.0f); //maximum value of 1.0 for red  
59         g = Mathf.Min(c0.g + c1.g, 1.0f); //maximum value of 1.0 for green  
60         b = Mathf.Min(c0.b + c1.b, 1.0f); //maximum value of 1.0 for blue  
61         a = Mathf.Min(c0.a + c1.a, 1.0f); //maximum value of 1.0 for alpha  
62  
63         return (new Color(r,g,b,a));  
64     } //float Add (Color c0, Color c1)  
65  
66     //Add function with optional arguments - any number of ints  
67     int Add(params int[] ints)  
68     {  
69         int sum = 0;  
70         foreach (int i in ints)  
71         {  
72             sum = sum + i;  
73         }  
74         return(sum);  
75     } //Add(params int[] ints)  
76  
77  
78     // Update is called once per frame  
79     void Update ()  
80     {  
81         numTimesCalled++;  
82         CountUpdates ();  
83  
84     } //Update ()  
85  
86  
87     //function that returns nothing - has "void" type  
88     void CountUpdates()  
89     {  
90         string outputMessage = "Updates= " + numTimesCalled;  
91         //print (outputMessage);  
92     } //CountUpdates  
93  
94 } // class CodeExamples
```