
Cocos2D Tutorial Documentation

Release 2019

Raphael Holzer

Apr 17, 2019

Contents:

1	Introduction	1
1.1	Hello world	1
1.2	Create a layer	1
1.3	Create a label	2
1.4	Create a class	2
2	Board games	5
3	Tile maps	7
4	Particle systems	9
5	Creating documentation	11
5.1	Doctest blocks	11
5.2	Footnotes	11
6	References	13
6.1	Reference	13
7	Indices and tables	17

Cocos2D is a platform for creating games.

1.1 Hello world

We begin this tutorial with the traditional *Hello World* program. This program

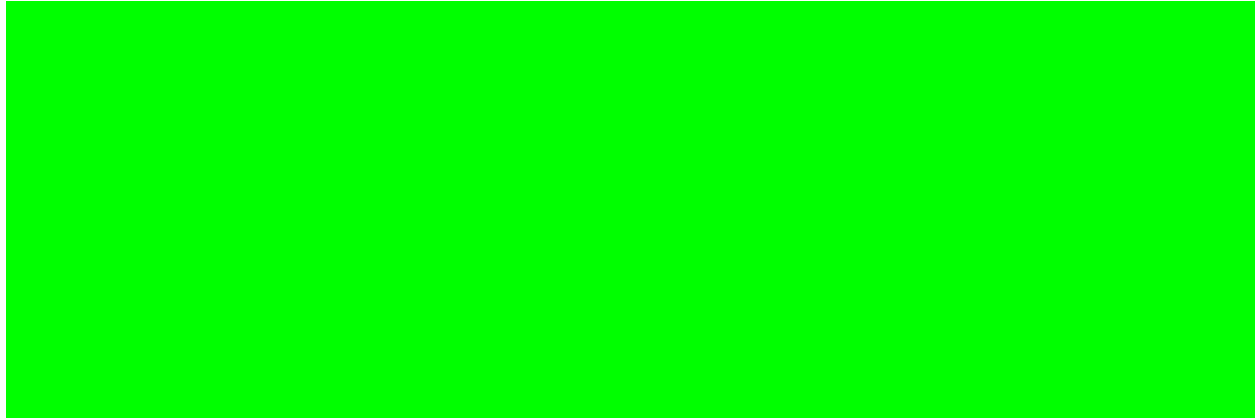
- imports the module Cocos2D
- initializes the module,
- creates a label,
- adds this lable to a layer,
- adds this layer to a scene, and finally
- runs the scene.

1.2 Create a layer

Type this code in a new file:

```
import cocos

cocos.director.director.init(600, 200)
layer = cocos.layer.ColorLayer(0, 255, 0, 255)
scene = cocos.scene.Scene(layer)
cocos.director.director.run(scene)
```



1.3 Create a label

Create a Label object and add it to the color layer:

```
import cocos

cocos.director.director.init(600, 200)

label = cocos.text.Label('Hello World', font_size=72)
layer = cocos.layer.ColorLayer(255, 0, 0, 255)
layer.add(label)
scene = cocos.scene.Scene(layer)
cocos.director.director.run(scene)
```

This is the result:



1.4 Create a class

In Cocos2D we normally subclass a Layer to create a new custom layer:

```
import cocos

# sub-class a layer
```

(continues on next page)

(continued from previous page)

```
class HelloWorld(cocos.layer.ColorLayer):
    """Create a new layer by sub-classing Layer."""

    # define the constructor function
    def __init__(self):
        # call the parent constructor
        super(HelloWorld, self).__init__(127, 127, 0, 255)
        # create a Label object
        label = cocos.text.Label('Hello World', font_size=72)
        # add the label to the layer
        self.add(label)

cocos.director.director.init(600, 200)
scene = cocos.scene.Scene(HelloWorld())
cocos.director.director.run(scene)
```

This is the result:



CHAPTER 2

Board games

In this section we create board games which are played on a rectangular grid.

CHAPTER 3

Tile maps

Cocos2D can work with rectangular and hexagonal tile maps

CHAPTER 4

Particle systems

Cocos2D uses particle systems to create special effects.

Creating documentation

This document is created with sphinx. In order to recreate it at your place, you have to clone

- clone the repository
- open a terminal
- change to the **docs** folder

Execute this commands:

```
cd docs
make html
```

This compiles the documentation and places the output into the `_build` folder.

5.1 Doctest blocks

```
>>> print('hello')
hello
```

```
>>> 1 + 2
3
```

5.2 Footnotes

This is a footnote reference¹.

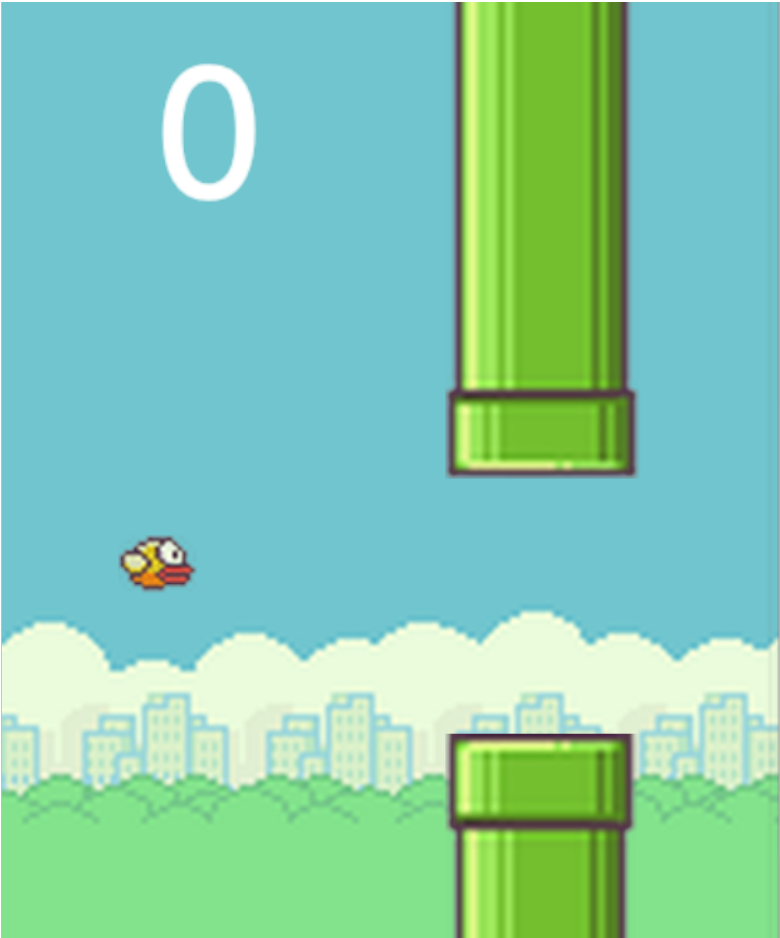
Autonumbered footnotes are possible, like using² and³.

¹ A numerical footnote.

² This is the first one.

³ This is the second one.

External hyperlinks, like Cocos2D.



6.1 Reference

This section presents the different demo examples. Each scene is implemented as a layer.

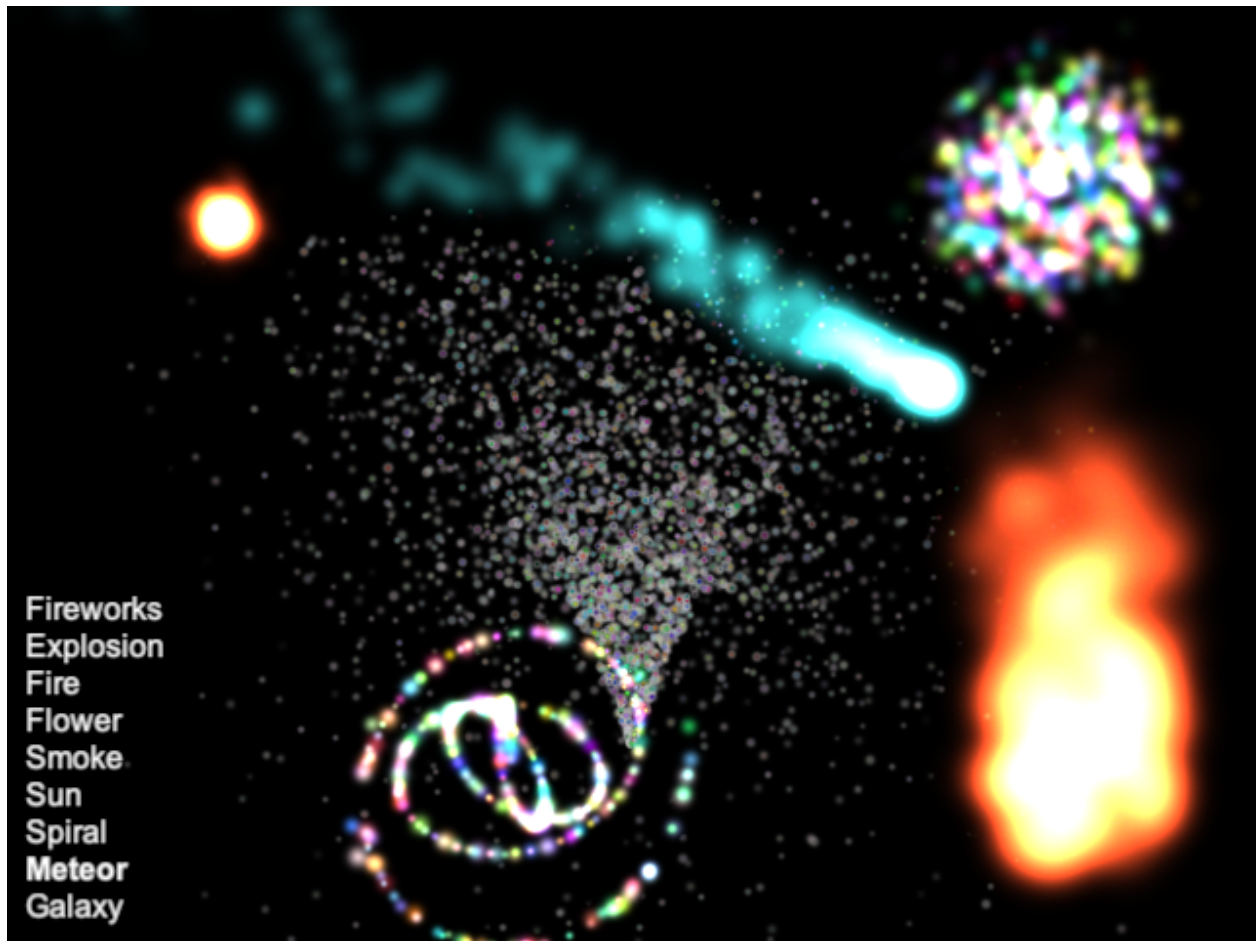
6.1.1 ListMenu

The ListMenu class allows to display a menu of list items.



6.1.2 Particle systems

Particle systems are used to create special effects such as fire or smoke.



6.1.3 Inspector

6.1.4 mylib

6.1.5 demo

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`