

Empezando

En este capítulo instalaremos y configuraremos Cocos2D-X y crearemos un nuevo proyecto.

- ¿Qué es Cocos2D-X?
- Configurar Cocos2D-X.
- Creación del proyecto.

¿Qué es Cocos2D-X?

Cocos2D-X es un framework multiplataforma y de código libre para la programación de videojuegos. Podemos utilizar C++, Lua o Javascript como lenguaje de programación. Para un mismo desarrollo, podremos ejecutar nuestro juego en iOS, Android y Windows Phone, además de en Windows, Linux y Mac. Además, la versión en Javascript podrá ser ejecutada en un navegador web.

En este libro generaremos nuestro juego para iOS y Android, utilizando C++ como lenguaje de programación.

Configurar Cocos2D-X

El juego que vamos a programar en el transcurso de este libro será compatible con iOS y Android. Por tanto, vamos a configurar todo el entorno de desarrollo en un sistema MacOSX con XCode instalado.

Primero creamos una carpeta donde descargaremos e instalaremos todo lo necesario.

```
1 | mkdir /Users/jonseg/programacion
```

A continuación, descargamos los siguientes archivos y los guardamos en la carpeta "programacion" que acabamos de crear:

- **Cocos2D-X**: Lo podremos descargar desde esta url: <http://cocos2d-x.org/download> (en el momento de escribir este libro, la última versión estable era la 3.8.1).
- **Android Developer Tools (ADT)**: Lo necesitaremos para el desarrollo de juegos para Android. Lo podemos descargar de esta url: <http://developer.android.com/sdk/index.html#Other> (en el momento de escribir este libro, la última versión es la r24.4.1).

Inicialmente, el archivo que nos descargamos solo contiene las herramientas básicas. Una vez descomprimido, **debemos ejecutar el siguiente comando** para instalar las diferentes plataformas de Android:

```
1 | cd /Users/jonseg/programacion/android-sdk-macosx
2 | tools/android update sdk --no-ui
```

- **Android Studio.** <https://developer.android.com/sdk/index.html>
- **Native Development Kit (NDK):** También será necesario si queremos desarrollar el juego para Android. Lo descargaremos de aquí: <https://developer.android.com/ndk/downloads/index.html> En la misma página se explica el proceso de instalación.
- **Apache ANT:** También necesario para Android. Lo descargamos de esta url: <http://ant.apache.org/bindownload.cgi>

Vamos a proceder a su instalación.

1. Descomprimos todos los archivos que nos hemos descargado.
2. Abrimos el terminal y nos movemos a la carpeta donde hemos descomprimido Cocos2D-X.

```
1 | cd /Users/jonseg/programacion/cocos2d-x-3.8.1
```

3. Ejecutamos el siguiente comando, que nos ayudará a configurar Cocos2D-X:

```
1 | python ./setup.py
```

4. El comando nos preguntará donde se encuentra instalada cada una de las librerías que hemos descomprimido al principio.

En primer lugar nos preguntará por el `NDK_ROOT`, que es la ruta donde tenemos instalado el NDK. En nuestro ejemplo, será `/Users/jonseg/programacion/android-ndk-r10e`:

```

1 | Setting up cocos2d-x...
2 | ->Check environment variable COCOS_CONSOLE_ROOT
3 |   ->Search for environment variable COCOS_CONSOLE_ROOT...
4 |     ->COCOS_CONSOLE_ROOT is found : /Users/jonseg/programacion/cocos2d-x-3.
5 |
6 | ->Check environment variable COCOS_TEMPLATES_ROOT
7 |   ->Search for environment variable COCOS_TEMPLATES_ROOT...
8 |     ->COCOS_TEMPLATES_ROOT is found : /Users/jonseg/programacion/cocos2d-x-
9 |
10 | ->Configuration for Android platform only, you can also skip and manually e
11 |
12 | ->Check environment variable NDK_ROOT
13 |   ->Search for environment variable NDK_ROOT...
14 |     ->NDK_ROOT not found
15 |
16 |   -> Please enter the path of NDK_ROOT (or press Enter to skip):/Users/jons

```

A continuación nos preguntará por el ANDROID_SDK_ROOT, que en nuestro caso será /Users/jonseg/programacion/android-sdk-macosx

```

1 |
2 | ->Check environment variable ANDROID_SDK_ROOT
3 |   ->Search for environment variable ANDROID_SDK_ROOT...
4 |     ->ANDROID_SDK_ROOT not found
5 |
6 |   -> Please enter the path of ANDROID_SDK_ROOT (or press Enter to skip):/Us

```

Por último, nos preguntará por el ANT_ROOT. Aquí hay que tener en cuenta que tenemos que poner la ruta a la carpeta "bin", que en nuestro ejemplo será /Users/jonseg/programacion/apache-ant-1.9.6/bin

```

1 |
2 | ->Check environment variable ANT_ROOT
3 |   ->Search for environment variable ANT_ROOT...
4 |     ->ANT_ROOT not found
5 |
6 |   -> Please enter the path of ANT_ROOT (or press Enter to skip):/Users/jons

```

Para terminar, el propio comando nos indica que tenemos que actualizar las variables de entorno que acabamos de configurar. Para ello ejecutamos el comando que nos indica al final:

```

1 | source /Users/jonseg/.bash_profile

```

En este punto ya tenemos configurado todo el entorno de desarrollo para crear nuestro primer proyecto.

Creación del proyecto

Ha llegado el momento de crear nuestro primer proyecto en Cocos2D-X. Para ello, Cocos2D-X cuenta con un comando que nos ayudará en el proceso.

1. Abrimos el terminal.
2. Utilizaremos el comando "cocos" para crear nuestro primer proyecto. El comando cuenta con los siguientes parámetros:
 - new nombre del proyecto
 - -p nombre del paquete
 - -l lenguaje de programación
 - -d carpeta donde generar el proyecto

Para nuestro proyecto, ejecutaremos el comando con estos parámetros:

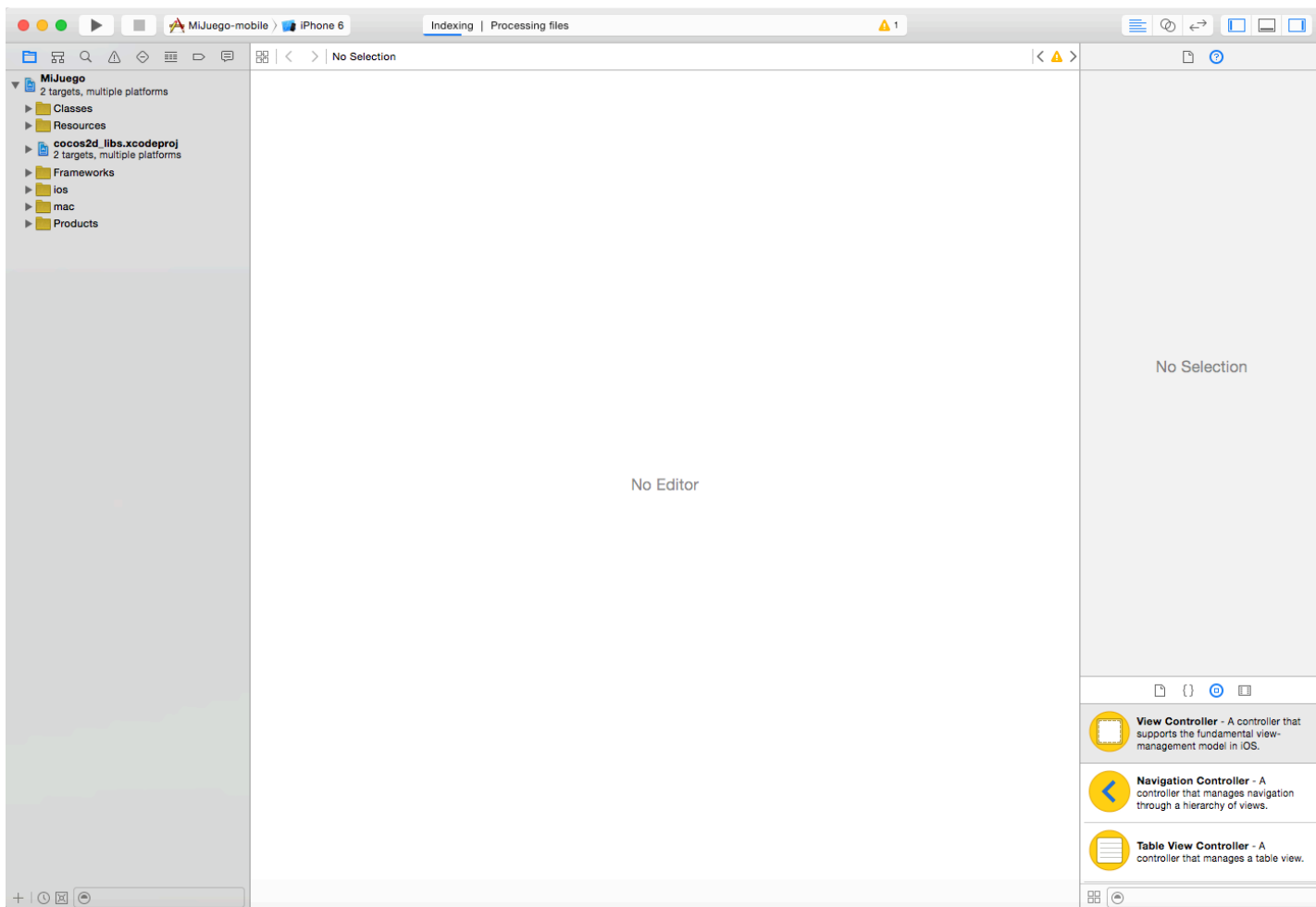
```
1 | cocos new MiJuego -p com.miempresa.mijuego -l cpp -d /Users/jonseg/programa
```

¡Enhorabuena! Hemos creado nuestro primer proyecto. El siguiente paso es ejecutarlo en nuestros móviles iOS y Android.

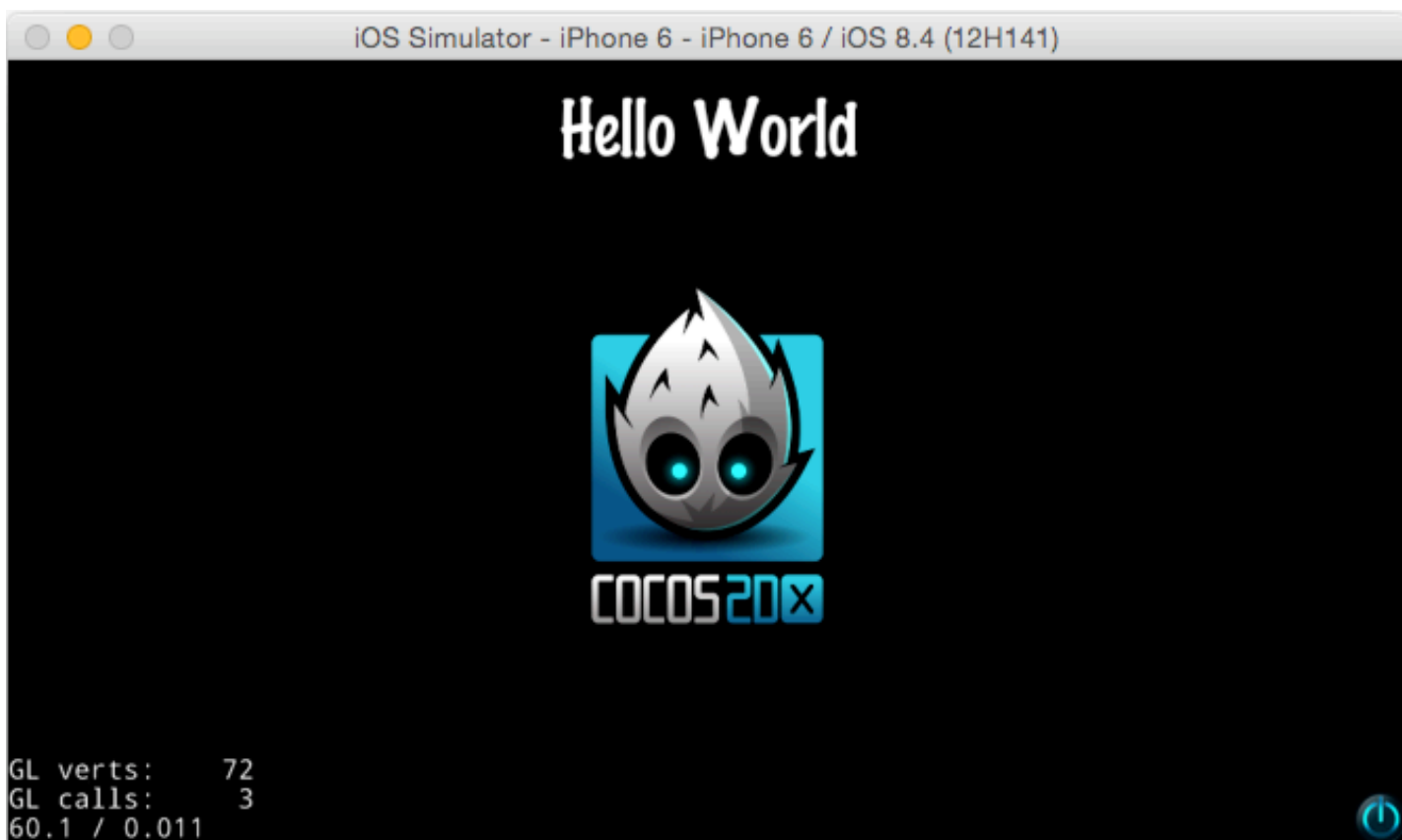
Si abrimos la carpeta del proyecto que acabamos de generar (se encuentra en `/Users/jonseg/programacion/MiJuego`), veremos que se ha creado una estructura de carpetas algo particular.

- **Classes:** En esta carpeta se guardará el código fuente de nuestro juego.
- **cocos2d:** Código fuente de Cocos2D-X.
- **proj.android:** Contiene los archivos para el proyecto Android.
- **proj.android-studio:** Contiene los archivos para el proyecto Android, compatible con Android Studio.
- **proj.ios_mac:** Contiene los archivos para el proyecto iOS y Mac.
- **proj.linux:** Contiene los archivos para el proyecto Linux.
- **proj.win8.1-universal:** Contiene los archivos para el proyecto Windows Phone 8.1.
- **proj.win10:** Contiene los archivos para el proyecto Windows 10.
- **proj.win32:** Contiene los archivos para el proyecto Windows.
- **Resources:** Aquí se guardarán las imágenes, sonidos, etc.

En primer lugar vamos a ejecutar el juego en iOS. Para ello, accedemos a la carpeta `proj.ios_mac` y hacemos doble click en `MiJuego.xcodeproj` para abrirlo con XCode.



Pulsamos el botón "run" y ejecutamos el juego en el simulador.



¡Listo! Nuestro proyecto funciona perfectamente en iOS. Lo que vemos es el proyecto de ejemplo que viene integrado en Cocos2D-X.

Ahora vamos a ejecutar nuestro juego en Android.

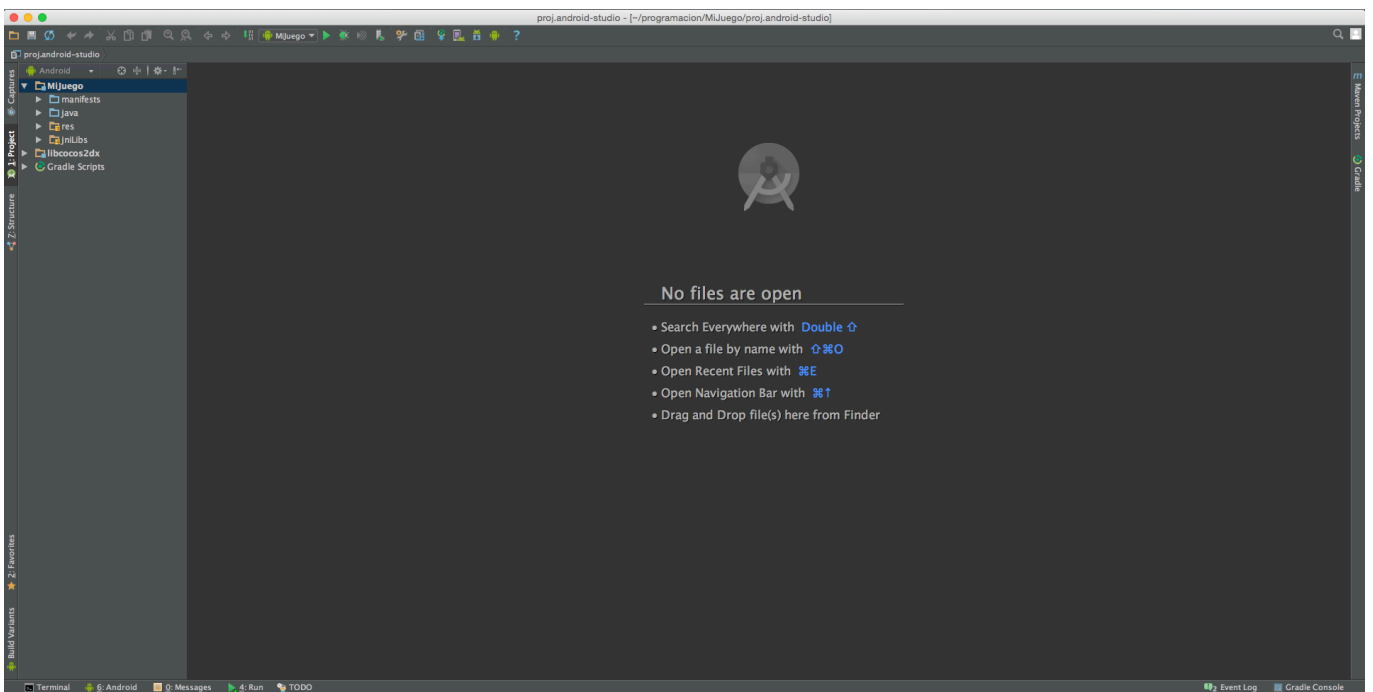
1. Abrimos el terminal.
2. Nos movemos a nuestro proyecto creado anteriormente.

```
1 | cd /Users/jonseg/programacion/MiJuego
```

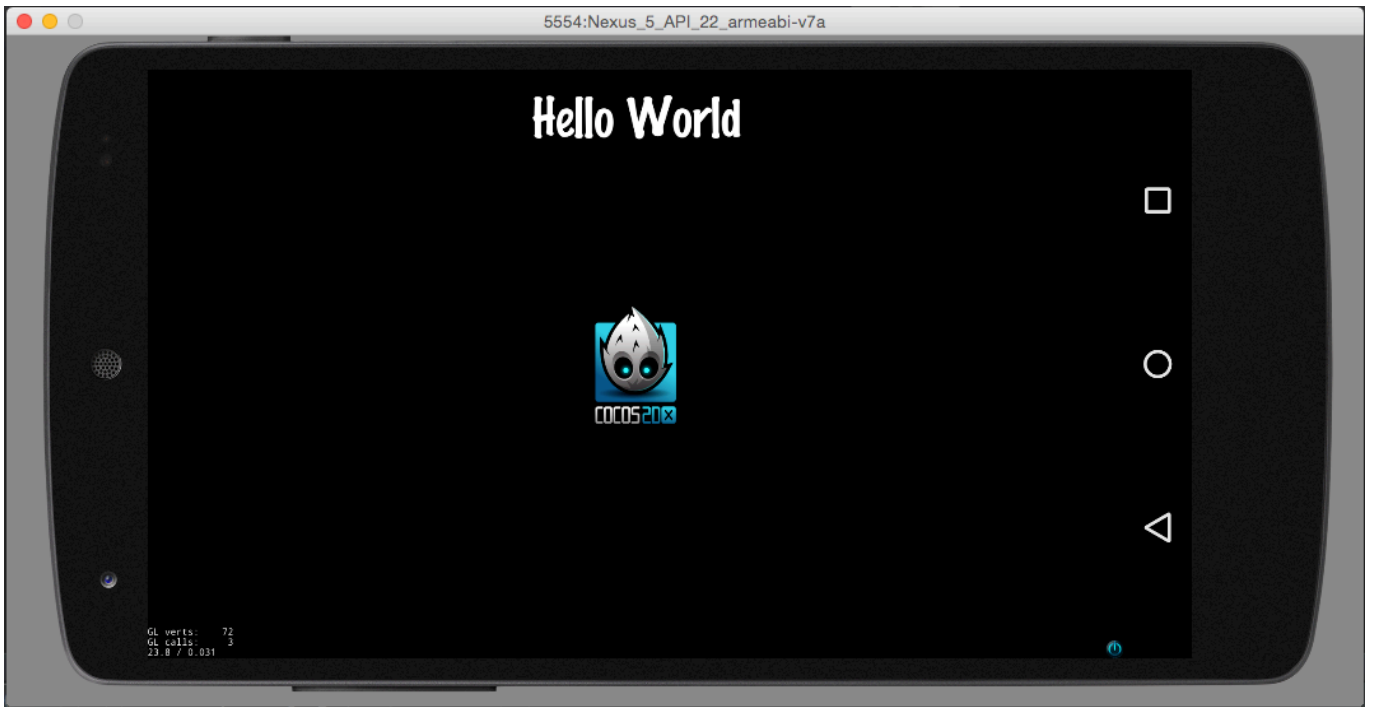
3. Compilamos el proyecto para poder ejecutarlo posteriormente en Android Studio.

```
1 | cocos compile -p android --android-studio
```

4. Abrimos Android Studio y pulsamos en "Open an existing Android Studio project". Seleccionamos la carpeta "proj.android-studio" de nuestro proyecto.



5. Pulsamos el botón "run" y ejecutamos el juego en el simulador (si tienes que crear un nuevo simulador, asegúrate de que la arquitectura sea armeabi o armeabi-v7a).



¡Perfecto! Ya tenemos el juego de ejemplo ejecutándose tanto en iOS como en Android. Ha llegado el momento de empezar a desarrollar nuestro juego.