

**Autore:** Carlo Carraro  
**Framework valutato:** Kobold2D (basato su framework Cocos2D)  
**Data:** 9 Settembre 2013

#### **Licenze e costi**

**Valutazione: 5**

Free/Open Source, Licenza MIT. Si può ridistribuire codice basato su questo framework sia in modalità open source sia senza rilasciare il codice sorgente. Esiste un prodotto commerciale basato su Kobold2D, chiamato KoboldTouch, che prevede una licenza a pagamento con abbonamento.

#### **API**

**Valutazione: 5**

Il framework è nativo, per cui le funzionalità del dispositivo si possono sfruttare al massimo. Cocos2D implementa un motore grafico per giochi in 2D ed è basato su OpenGL. Ci sono centinaia di API messe a disposizione che consentono di velocizzare le operazioni tipiche dei giochi in 2D. Qualche conoscenza di base su OpenGL sarebbe utile (ma non necessaria).

#### **Tutorial**

**Valutazione: 4**

Variati documenti disponibili in rete: Programming Guide, spiegazioni nella stessa documentazione API, FAQ online, Video Tutorial, sia per utenti alle prime armi che per utenti avanzati. Inoltre sono disponibili numerosi esempi distribuiti insieme al codice sorgente del framework; su App Store sono apparse numerose varianti di giochi classici basati su questi esempi e con grafica personalizzata.

Si possono inoltre acquistare numerosi libri scritti da esperti (anche gli stessi sviluppatori del framework) che aiutano ad imparare ad utilizzare il framework seguendo un percorso logico (utile per chi non ha idea di come iniziare e che non sa quali sono i principi fondamentali su cui si basa un qualsiasi framework per sviluppare giochi).

#### **Community**

**Valutazione: 5**

La community di Cocos2D è immensa. Nell'App Store sono stati distribuiti migliaia di giochi basati su questo framework. In rete si possono trovare moltissimi esempi di codice, risposte a domande, documentazione. Gli sviluppatori sono molto attivi: il mercato dei giochi è in continua espansione. Alcuni sviluppatori sono pagati da importanti aziende per mantenere il framework; queste aziende sviluppano giochi basati su di esso. Molti sviluppatori contribuiscono alla crescita del framework ritornando alla comunità pezzi di codice e migliorie.

#### **Sforzo realizzativo**

**Valutazione: 2**

Lo sforzo iniziale è notevole. Richiede di avere una certa confidenza sia con il linguaggio di programmazione C che con il linguaggio Objective-C. Le API di cui ci si può avvalere sono come già detto centinaia e per capire come sfruttarle a pieno ci vogliono sicuramente mesi e mesi di

esperienza fatta sul campo. Per realizzare giochi relativamente semplici ci si può comunque basare sugli esempi disponibili (talvolta anche giochi interi veri e propri) e sviluppare la propria idea partendo da quelli.

Dopo aver investito del tempo iniziale lo sviluppo dei progetti successivi è sicuramente molto più rapido; inoltre non si può pensare di sviluppare un gioco con un minimo di grafica e fisica senza l'utilizzo di un framework come questo.

## **Ambiente di sviluppo**

### **Valutazione: 4**

L'ambiente di sviluppo è Xcode. Per quanto riguarda la stesura di codice è sicuramente molto confortevole e ha tutto quello di cui uno sviluppatore ha bisogno. Tuttavia per lo sviluppo di giochi avanzati è comunque necessario avvalersi di altri programmi che consentono di lavorare con le sprite (eg. TexturePacker), programmi che consentono di produrre codice per gestire la fisica e le collisioni tra gli elementi del gioco (eg. PhysicsEditor) e programmi per la manipolazione delle immagini.

## **Supporto ai dispositivi mobili**

### **Valutazione: 3**

Kobold2D è stato realizzato a partire da Cocos2D for iPhone; Cocos2D for iPhone è scritto in Objective-C per sistemi iOS e Mac OS X, ed è il port dell'originale Cocos2D, scritto in Python. Esistono dei metodi per cross-compilare il codice Objective-C per altre piattaforme come Android (vedere Apportable).

## **Grafica nativa**

### **Valutazione: 5**

Tutto il codice è basato su OpenGL, per cui si può sfruttare al massimo il chip grafico del dispositivo senza caricare il processore.

## **Conoscenze richieste**

### **Valutazione: 2**

Le conoscenze per arrivare a produrre un gioco che abbia in sé un minimo di grafica e fisica sono molteplici; il linguaggio di programmazione è solo il mezzo con cui si arriva al risultato. Il motore grafico alla base di Cocos2D è stato originalmente scritto in Python e successivamente riscritto in altri linguaggi di programmazione quali C++, C#, Java, etc.. Detto ciò il framework in versione Cocos2D for iPhone richiede un minimo di conoscenza del linguaggio Objective-C. Tale linguaggio ha avuto una rapida evoluzione negli ultimi 5 anni grazie al successo della piattaforma iOS. Il linguaggio ha una sintassi all'inizio un po' ostica per chi viene da linguaggi quali C++ e Java, ma nel giro di qualche settimana di lavoro si riesce ad avere una padronanza sufficiente. È consigliato comunque seguire un tutorial o un libro che seguano un percorso logico per avere un minimo di base in modo da non intercorrere in errori grossolani o reinventare la ruota. Apple mette a disposizione svariati framework per sviluppare App su iOS; un minimo di conoscenza di questi framework e dei pattern che si trovano all'interno di essi sarebbe utile, sebbene non strettamente necessaria. I giochi sono totalmente diversi da un'app di utilità che prevede l'uso di componenti grafici tipici del dispositivo, quindi non è necessario conoscere framework come Cocoa Touch.

## **Note Finali**

Apple nell'estate 2013, con l'imminente release di iOS 7, ha annunciato il nuovo framework per la creazione di giochi in 2D: SpriteKit. Questo framework è basato su Cocos2D. Questo potrebbe significare che molti sviluppatori che hanno come target di sviluppo solo iOS potrebbero passare ad usare questo nuovo framework. Lo stesso autore di Kobold2D ha cominciato un nuovo progetto, chiamato KoboldKit, basato su SpriteKit. Saranno comunque disponibili metodi come quello citato in precedenza che permetteranno di compilare il codice anche per Android.