

# Ampliaciones Cocos 1

## Arkanoid

### 1 Bloques aleatorios:

Añadimos los recursos gráficos necesarios y cacheamos las animaciones:

```
cc.spriteFrameCache.addSpriteFrames(res.animacioncocodrilo_plist);  
cc.spriteFrameCache.addSpriteFrames(res.animacionpanda_plist);  
cc.spriteFrameCache.addSpriteFrames(res.animaciontigre_plist);
```

Creamos una variable aleatoria que será la que decida que tipo de bloque crear:

```
while (insertados < numBloques) {  
  
    var animal = "";  
    var framesBloque = [];  
  
    var num = Math.floor(Math.random() * 3) + 1;  
  
    var accionAnimacionBloque;  
  
    switch (num) {  
        case 1:  
            animal = "cocodrilo";  
            break;  
  
        case 2:  
            animal = "panda";  
            break;  
  
        case 3:  
            animal = "tigre";  
            break;  
    }  
  
    for (var i = 1; i <= 8; i++) {  
        var str = animal + i + ".png";  
        var frame = cc.spriteFrameCache.getSpriteFrame(str);  
        framesBloque.push(frame);  
    }  
}
```

Por último creamos el sprite con la animación aleatoria:

```
var accionAnimacionBloque = new cc.RepeatForever(new cc.Animation({
var spriteBloqueActual = new cc.Sprite("#"+animal+"1.png");
```

## 2 Niveles

Creamos una variable numBloques que representará el número de bloques por nivel:

```
var GameLayer = cc.Layer.extend({
    spritePelota: null,
    velocidadX: null,
    velocidadY: null,
    spriteBarra: null,
    spriteBloque: null,
    arrayBloques: [],
    numBloques: 10,
```

Esta variable será utilizada en el método inicializarBloques para saber el número de bloques que debe crear:

```
inicializarBloques: function (numBloques) {
    var insertados = 0;
    var fila = 0;
    var columna = 0;

    while (insertados < this.numBloques) {

        var animal = "";
```

Por último, en el update comprobamos si se han destruido todos los bloques, y en caso de ser cierto aumentamos en 10 el número de bloques y vuelve a empezar el juego:

```
update: function (dt) {

    // Si se acaban los bloques cambiamos de nivel aumentando en 10 el número de bloques
    if (this.arrayBloques.length == 0) {
        this.numBloques +=10;
        console.log("NUMBLOQUEEEEEES"+this.numBloques);
        this.ctor();
    }
}
```

## TiraLaTorre

### 5 Intentos:

Definimos una variable intentos con valor 3.

En el método actualizar comprobamos si los bloques que quedan son  $> 0$ . En caso de que lo sean, comprobamos que están todos quietos y una vez pasado un tiempo, si el número de intentos es mayor de 1, creamos una nueva bola para que el jugador vuelva a intentarlo. Si el número de intentos es  $\leq 1$ . El juego vuelve a empezar.

```
if( this.arrayBloques.length > 0){
    var quietos = true;
    for(var i = 0; i < this.arrayBloques.length; i++) {
        var vel = this.arrayBloques[i].body.getVel();
        if( vel.x < -0.09 && vel.x > 0.09 ){
            quietos = false;
        }
    }
}
if ( this.tiempoTiro > 300 &&
    quietos && this.intentos>1){
    this.tiempoTiro=0;
    this.mundoActivo=false;
    this.intentos--;
    //Devolver pelota a posición inicial
    var size = cc.winSize;
    var body = new cp.Body(1,
        cp.momentForCircle(1, 0, this.spritePelota.width/2, cp.vzero));
    body.p = cc.p(size.width*0.1 , size.height*0.5);
    this.removeChild(this.spritePelota);
    this.spritePelota.setBody(body);
    this.space.addBody(body);
    var shape =
        new cp.CircleShape(body, this.spritePelota.width/2, cp.vzero);

    this.space.addShape(shape);

    this.addChild(this.spritePelota);
}
else if ( this.tiempoTiro > 300 &&
    quietos && this.intentos<=1){
    cc.director.pause();
    cc.audioEngine.stopMusic();
    this.getParent().addChild(new GameOverLayer());
}

} else {
    cc.director.pause();
    cc.audioEngine.stopMusic();
    this.getParent().addChild(new GameOverLayer());
}
```