

Ampliaciones Infinite Runner

1 Varias vidas

Añadimos un contador de vidas la clase ControlesLayer:

```
etiquetaVidas:null,
ctor:function () {
    this._super();
    var size = cc.winSize;

    // BotonSaltar
    this.spriteBotonSaltar = cc.Sprite.create(res.boton_saltar_png);
    this.spriteBotonSaltar.setPosition(
        cc.p(size.width*0.8, size.height*0.5));

    this.addChild(this.spriteBotonSaltar);

    //Vidas
    this.etiquetaVidas=new cc.LabelTTF("Vidas: 3","Helvetica",20);
    this.etiquetaVidas.setPosition(cc.p(size.width*0.1, size.height*0.9));
    this.etiquetaVidas.fillStyle.color="#000000";
    this.addChild(this.etiquetaVidas);
}
```

Además, creamos una función para establecer el número de vidas:

```
,setVidas:function (vidas) {
    this.etiquetaVidas.setString("Vidas: "+ vidas);
}
```

Por último, llamamos a la función en la GameScene cuando colisionemos con un enemigo.

```
collisionJugadorConEnemigo:function (arbiter, space) {
    this._emitter.setEmissionRate(1);
    this.tiempoEfecto = 1;
    // Marcar el enemigo para eliminarla
    var shapes = arbiter.getShapes();
    this.formasEliminar.push(shapes[1]);
    var capaControles = this.getParent().getChildByTag(idCapaControles);
    this.jugador.vidas--;
    capaControles.setVidas(this.jugador.vidas);

    if(capaControles.vidas<=0){
        this.jugador.body.p = cc.p(50,150);
        this.jugador.vidas=3;
        capaControles.setVidas(this.jugador.vidas);
    }
}
```

2 Scroll en el eje Y

En la GameScene, únicamente añadimos y modificamos una línea:

```

var posicionXJugador = this.jugador.body.p.x - 200;
var posicionYJugador = this.jugador.body.p.y-100;
this.setPosition(cc.p( -posicionXJugador,-posicionYJugador));

```

6 Pinchos

He reaprovechado la animación del tigre del juego Arkanoid para la representación de pinchos.

Creamos la clase Pinchos:

```

var Pinchos = cc.Class.extend({
    gameLayer:null,
    sprite:null,
    shape:null,
    ctor:function (gameLayer, posicion) {
        this.gameLayer = gameLayer;

        this.sprite = new cc.PhysicsSprite("#tigrel.png");
        var body = new cp.StaticBody();
        body.setPos(posicion);
        this.sprite.setBody(body);
        var radio = this.sprite.getContentSize().width / 2;

        this.shape = new cp.CircleShape(body, radio , cp.vzero);

        this.shape.setSensor(true);
        this.shape.setCollisionType(tipoPincho);

        gameLayer.space.addStaticShape(this.shape);

        gameLayer.addChild(this.sprite,10);

        var framesAnimacion = [];
        for (var i = 1; i <= 8; i++) {
            var str = "tigre" + i + ".png";
            var frame = cc.spriteFrameCache.getSpriteFrame(str);
            framesAnimacion.push(frame);
        }
        var animacion = new cc.Animation(framesAnimacion, 0.2);
        var actionAnimacionBucle =
            new cc.RepeatForever(new cc.Animate(animacion));

    }, update:function (dt, jugadorX) {

    }, eliminar: function () {
        this.gameLayer.space.removeShape(this.shape);
        this.gameLayer.removeChild(this.sprite);
    }
});

```

Por último, dibujamos los pinchos en la clase GameScene y detectamos su colisión con el jugador, al igual que en la ampliación 1.