

22.cocosCreator事件系统

1、访问别的节点

1.1 通过属性检查器设置节点来访问其他节点

将节点1挂到节点2之下，使节点1为节点2的组件

1.2访问子节点

```
//访问父节点    canvas 父节点为 scene ， scene父节点为null
this.node.parent
this.node.getParent()

//得到父节点便可以下列方式得到子节点
//通过节点名 筛选
this.node.parent.getChildByName(name:string)
//得到子节点数组
this.node.parent.children
//得到子节点的第一个元素
this.node.parent.children[0]

//通过路径去访问节点
//使用的是cc.find()方法
let logo_tex =cc.find("Canvas/background")
```

1.3 canvas与camera

canvas：画布，用来做屏幕的适配，建议所有ui节点以及可渲染节点都放canvas节点下

camera：相机

2、节点的常用属性

2.1

position , rotation , scale , anchor , size , color , opacity

2.2修改节点的常用属性

```
//this 当前类 当前脚本组件
//this.node 当前脚本所绑定的节点
this.node.position.x = 50
this.node.setPosition(50,50)
```

3、计时器

3.1 定时器

```
//开启定时器
//this.schedule(callback: Function, interval?: number, repeat?: number, delay?: number):void
//下面的代码表示设置定时器，每5秒打印一次11111
this.schedule(function(){
    console.log("11111");
},5);
```

3.2 只执行一次的计时器

```
//this.scheduleOnce(callback: Function, delay?: number):void
//下面的代码表示设置一次性定时器，5秒后打印一次11111
this.scheduleOnce(function(){
    console.log("2222");
},5);
```

3.3 定时器的关闭

```
//实际callback_fn为 关闭的函数名
this.unschedule(callback_fn: Function)
```

3.4 关闭当前组件的所有计时器

```
this.unscheduleAllCallbacks()
```

4、切换场景

cocos2dx: runWithScene replaceScene

cocosCreator: loadScene

```
//SceneName可以直接输入Scene的名字
cc.director.loadScene(sceneName: string, onLaunched?: Function)
```

5、事件系统

鼠标事件

触摸事件

键盘事件

重力感应事件

自定义事件

自定义事件

事件监听

```
//先定义一个事件监听器
let ec =new cc.EventTarget();

//事件的接收 参数一事件的名字 后续函数是回调函数的参数
//cc.EventTarget.on(type: string, callback: () => void, target?: any,
useCapture?: boolean)
ec.on("Dead",this.callFun3);

//事件取消监听
//要使得事件不能被监听, 需要在emit之前 降监听关闭
//cc.EventTarget.off(type: string, callback?: Function, target?: any): void
ec.off("Dead");

//事件的发送 key事件名之后, 可接参数用于回调函数
//cc.CallbacksInvoker.emit(key: string, arg1?: any, arg2?: any, arg3?: any,
arg4?: any, arg5?: any): void
ec.emit("Dead");

//*****//
//事件发送时, 带参数
callFun4(cs1,cs2,cs3){
    console.log(cs1);
    console.log(cs2);
    console.log(cs3);
}
let ec =new cc.EventTarget();
ec.on("Dead",this.callFun4);
ec.emit("Dead",1,2,3);
//*****//
```



