

Cocos2dx 游戏开发教程

by 邓立



目录 / contents

01

事件分发与响应

02

音乐与音效



section1

事件分发与响应



COCOS2D X

现在大部分移动游戏应用中，都是通过玩家触摸屏幕的方法进行人机交互，桌面游戏中，则以鼠键输入为主，这一节我们会涉及到cocos2dx自带的各种事件的分发与响应的知识，以及如何注册一个自定义事件，以及自定义事件的分发。



什么是事件分发机制

基本概念

- 事件监听器封装了事件处理的代码；
- 事件调度器通知用户事件的监听器；
- 事件对象包含了关于事件的信息。



使用方法及步骤

//一： 获取一个事件分发器

```
auto dispatcher = Director::getInstance()->getEventDispatcher();
```

//二： 创建一个监听器 （以单点触摸为例）

```
auto touchListener = EventListenerTouchOneByOne::create();
```

//三： 响应事件绑定

```
touchListener->onTouchBegan = CC_CALLBACK_2(HelloWorld::onTouchBegan, this);
```

//四： 在事件分发器中，添加触摸监听器，事件响应委托给 this 处理

```
dispatcher->addEventListenerWithSceneGraphPriority(touchListener,  
this);
```



事件监听器的5种类型

- EventListenerTouch - 响应触摸事件
- EventListenerKeyboard - 响应键盘事件
- EventListenerAcceleration - 响应加速度计的事件
- EventListenMouse - 响应鼠标事件
- EventListenerCustom - 响应自定义事件



EventListenerTouch - 响应触摸事件

//创建单点触摸监听器

```
auto touchListener = EventListenerTouchOneByOne::create();
```

- onTouchBegan - 响应触摸初始事件
- onTouchMoved - 响应触摸移动事件
- onTouchEnded - 响应触摸结束事件
- onTouchCancelled - 响应触摸取消事件



EventListenerTouch - 响应触摸事件

//创建多点触摸监听器

```
auto touchListener = EventListenerTouchAllAtOnce::create();
```

- onTouchesBegan - 响应触摸初始事件
- onTouchesMoved - 响应触摸移动事件
- onTouchesEnded - 响应触摸结束事件
- onTouchesCancelled - 响应触摸取消事件



编写事件响应函数

//方法一：使用CC_CALLBACK_X

touchListener->onTouchBegan =

CC_CALLBACK_2(HelloWorld::onTouchBegan, this);

```
bool HelloWorld::onTouchBegan(Touch *touch, Event *unused_event) {  
    Vec2 postion = touch->getLocation();  
    //do somgthing  
}
```



编写事件响应函数

//方法二：使用lambda函数表达式

```
touchListener->onTouchBegan = [] (Touch* touch, Event* event) {  
    //do something  
};
```

查阅参数：

```
listener->onTouchBegan = [] (Touch* touch, Event* event) {  
    cocos2d::EventListenerTouchOneByOne::ccTouchBeganCallback cocos2d::EventListenerTouchOneByOne::onTouchBegan  
    typedef std::function<bool(Touch*, Event*)> ccTouchBeganCallback  
};
```



EventListenerKeyboard - 响应键盘事件

//创建单点触摸监听器

```
auto keyboardListener = EventListenerKeyboard::create();
```

- onKeyPressed - 响应键盘按下事件
- onKeyReleased - 响应键盘放开事件
- 通过参数中的KeyCode判断按下哪个按钮



COCOS2D X

EventListenerMouse - 响应鼠标事件

//创建单点触摸监听器

```
auto mouseListener = EventListenerMouse::create();
```

- onMouseDown - 响应鼠标点击按下事件
- onTouchesUp - 响应鼠标点击松开事件
- onTouchesMove - 响应鼠标移动事件
- onTouchesScroll - 响应鼠标滚轮事件



EventListenerAcceleration - 响应加速度计的事件

//打开加速度计

```
Device::setAccelerometerEnabled(true);
```

//创建加速度计监听器

```
auto listener = EventListenerAcceleration::create([] (Acceleration*  
acceleration, Event* event) {  
    log("X: %f; Y: %f; Z:%f; ", acceleration->x,  
acceleration->y, acceleration->z);  
});
```



加速计信息类Acceleration

```
class CC_DLL Acceleration
{
public:
    double x;
    double y;
    double z;

    double timestamp;

    Acceleration(): x(0), y(0), z(0), timestamp(0) {}
};
```



EventListenerCustom - 响应自定义事件

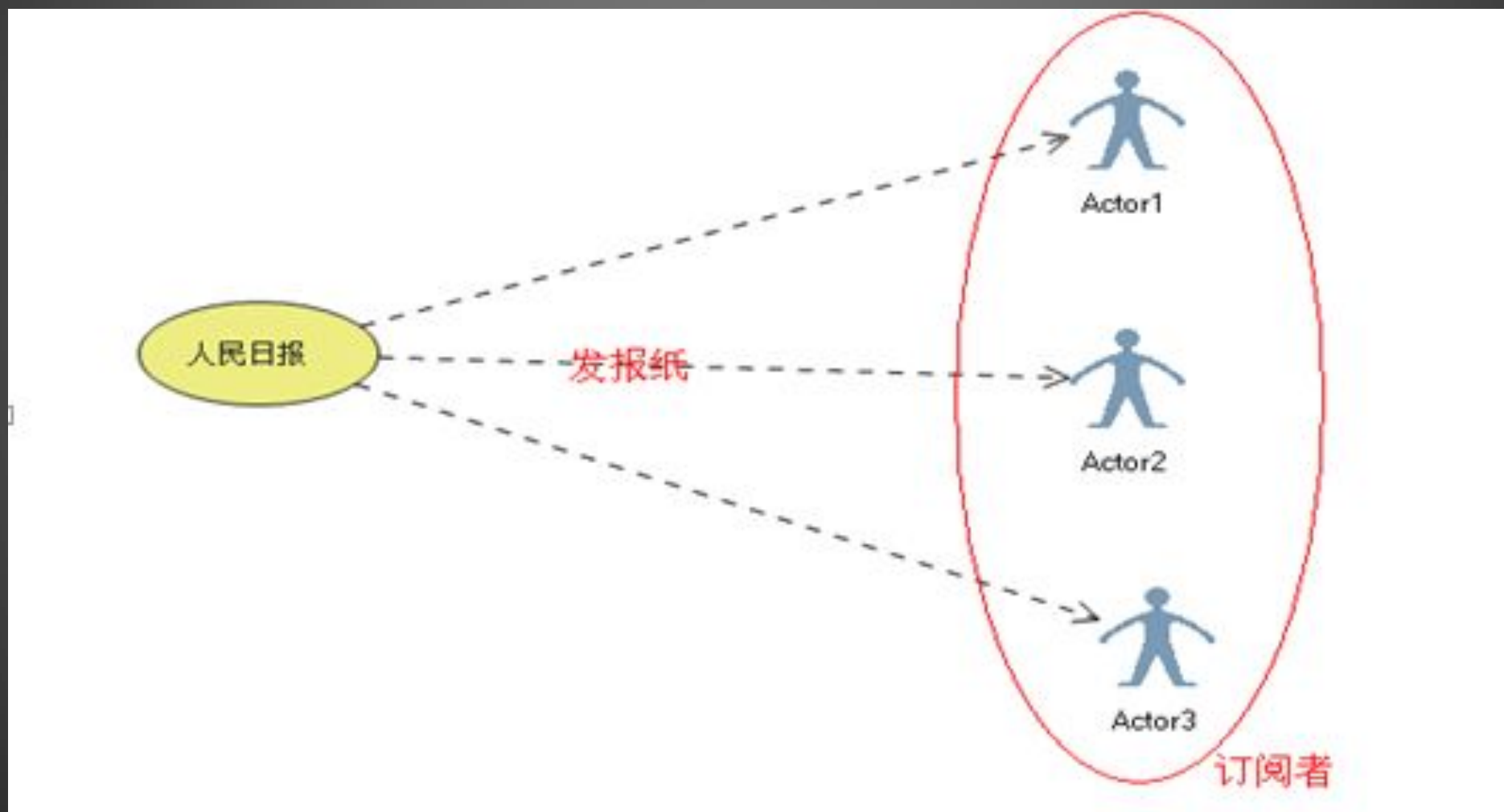
- 以上是系统自带的事件类型，事件由系统内部自动触发，如 触摸屏幕，键盘响应等。
- EventListenerCustom 自定义事件，它不是由系统自动触发，而是人为的干涉。



订阅者模式

- 设计模式的一种，可用于对象间信息传递与事件发布，降低对象间的耦合性，也叫观察者模式
- 由订阅者与发布者组成，游戏开发中，订阅者向事件发布者（发布者）注册监听器以监听某个事件的发生，当此事件发生时，事件发布者根据事件ID向监听器分发事件，然后监听器进行事件响应

订阅者模式--示意图



订阅

//方法一:

```
auto listener = EventListenerCustom::create("1", [] (EventCustom* event) {  
    //do something  
});  
dispatcher->addEventListenerWithFixedPriority(listener, 1);
```

//方法二:

```
dispatcher->addCustomEventListener("1", [] (EventCustom* event) {  
    //do something  
});
```

发布

//创建自定义事件

```
EventCustom e("1");
```

//传递信息

```
MyMessage* m = new MyMessage();
```

```
e->setUserData(m);
```

//发布事件

```
Director::getInstance()->getEventDispatcher()->dispatchEvent(&e);
```

section2

音乐与音效



好的游戏会俘虏玩家的感官，出色的背景音乐会让玩家更容易沉浸在你的虚拟世界中，应景的音效会让人玩起来更带感。

为此，Cocos2d-x为开发者提供了对声音和音效的支持，能够十分方便地实现音乐与音效的播放、暂停和循环功能。



SimpleAudioEngine

- Cocos2d-x自带的CocosDenshion库的一个实现
- 提供音乐与音效的播放暂停
- 方便易用



简易使用方法及例子

//SimpleAudioEngine是一个单例类，使用时需先获取单例

//预加载音乐与音效

```
SimpleAudioEngine::getInstance()->preloadBackgroundMusic("music/bgm.mp3");
```

```
SimpleAudioEngine::getInstance()->preloadEffect("music/shoot.mp3");
```

//播放BGM或音效

```
playBackgroundMusic("music/bgm.mp3", true);
```



简易使用方法及例子

//暂停、恢复与停止音乐

```
pauseBackgroundMusic();
```

```
resumeBackgroundMusic();
```

```
stopBackgroundMusic();
```

//设置音量

暂未实现...



COCOS2D X

支持平台与格式

CocosDesion支持的音乐格式如下：

平台	支持的常见文件格式	备注
Android	mp3, mid, oggg, wav	可以播放android.media.MediaPlayer所支持的所有格式
iOS	aac, caf, mp3, m4a, wav	可以播放AVAudioPlayer所支持的所有格式
Windows	mid, mp3, wav	无



支持平台与格式

CocosDesion支持的音效格式如下：

平台	支持的常见文件格式	备注
Android	oggg, wav	对wav的支持不完美
iOS	caf, m4a	可以播放Cocos2d-iPhone CocosDesion所支持的所有格式
Windows	mid, wav	无



hw: 割绳子

要求：

- 1：利用单点触摸事件实现割绳子的效果
- 2：游戏过程中有背景音乐，割绳子时有音效，割绳子后box执行下落动作
- 3：box下落到达目标后触发一个自定义事件去关闭背景音乐。



COCOS2D X

THE END

THANKS FOR WATCHING

