

# Cocos2d-JS游戏开发

---用户事件

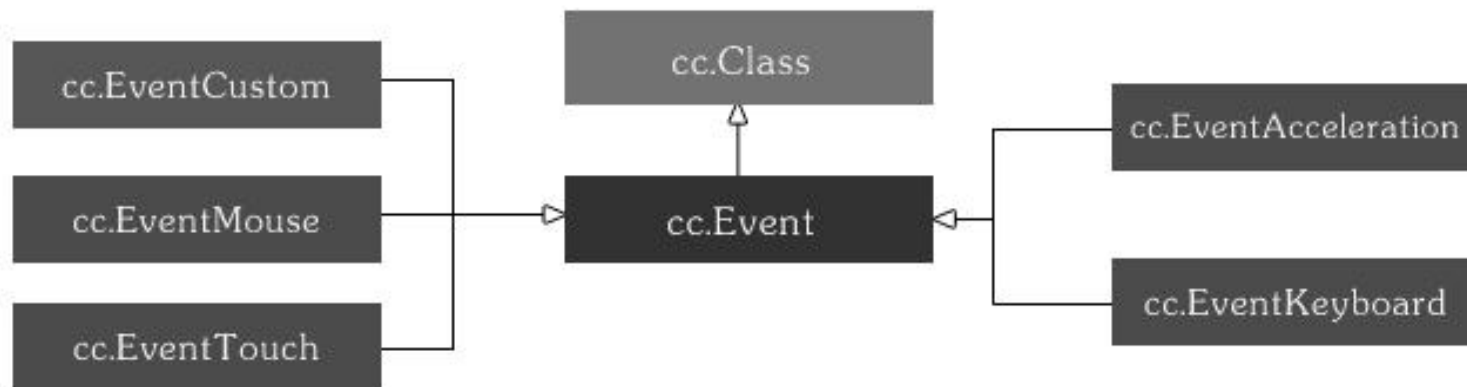


河北师范大学软件学院  
Software College of Hebei Normal University

# Cocos2d-JS事件及事件源

## • 事件 (cc.Event)

- 触摸事件
- 鼠标事件
- 键盘事件
- 加速度事件
- 自定义事件



- 事件对象是包含事件相关信息对象，如点击的坐标，滑动的距离等

# Cocos2d-JS事件监听器及管理器

- 事件监听器 (cc.EventListener)

- cc.EventListener.TOUCH\_ONE\_BY\_ONE (单点)
- cc.EventListener.TOUCH\_ALL\_AT\_ONCE (多点)
- cc.EventListener.MOUSE、cc.EventListener.KEYBOARD
- cc.EventListener.ACCELERATION
- cc.EventListener.CUSTOM

封装用户的事件处理逻辑

- 事件管理器 (cc.eventManager)

- cc.EventManager //负责添加、删除事件监听器
- cc.eventManager.addListener(listener,nodeOrPriority);
- cc.eventManager.removeListener(listener);
- cc.eventManager.removeListeners(listenerType,recursive)//类型、递归

# Cocos2d-JS事件处理流程

---

- 首先需要创建一个事件监听器
- 在监听器中实现各种事件的处理逻辑
- 将监听器加入到事件管理器中
- 当事件触发时，事件管理器会根据事件类型分发给相应的事件监听器
- 下面以一个简单的示例来演示使用的方法

# 单点触摸（完整写法）

---

```
var listener = cc.EventListener.create({  
    event: cc.EventListener.TOUCH_ONE_BY_ONE,  
    swallowTouches: true,  
    onTouchBegan: function (touch, event) {  
        return true; //返回布尔类型，若返回为false则，后边回调都不再执行  
    },  
    onTouchMoved: function (touch, event) { },  
    onTouchEnded: function (touch, event) { },  
    onTouchCancelled: function (touch, event) { }  
});  
cc.eventManager.addListener(listener, this);
```

# 单点触摸（简化写法）

```
cc.eventManager.addListener({  
    event: cc.EventListener.TOUCH_ONE_BY_ONE,  
    swallowTouches: true,  
    onTouchBegan: function (touch, event) {  
        return true; //返回布尔类型，若返回为false则，后边回调都不再  
        执行  
    },  
    onTouchMoved: function (touch, event) { },  
    onTouchEnded: function (touch, event) { },  
    onTouchCancelled: function (touch, event) { }  
}, this);
```

注：完整写法先定义监听器a，然后  
cc.eventManager.addListener(a,b);

# 自定义精灵的单点触摸

---

- 自定义精灵，添加事件监听器，两种方式
- cc.Touch
  - touch.getLocation( ); //点击坐标
  - touch.getDelta(); // 移动偏移量
- cc.Event
  - event.getCurrentTarget(); //事件源
- 使用getBoundingBox或坐标系转换判断点位置（后者更为准确，参见实例）

# 多点触摸

```
cc.eventManager.addListener({  
    event: cc.EventListener.TOUCH_ALL_AT_ONCE,  
    onTouchesBegan: function (touches, event) { },  
    onTouchesMoved: function (touches, event) { },  
    onTouchesEnded: function (touches, event) { },  
    onTouchesCancelled: function (touches, event) { }  
}, this);
```

## 多点触摸实验

注: if ('touches' in cc.sys.capabilities){}



# 键盘事件

---

```
if ('keyboard' in cc.sys.capabilities) {  
    cc.eventManager.addListener({  
        event: cc.EventListener.KEYBOARD,  
        onKeyPressed: function (key, event) {  
            var strTemp = "Key down:" + key;  
        },  
        onKeyReleased: function (key, event) {  
            var strTemp = "Key up:" + key;  
        }  
    }, this);  
}
```

# 鼠标事件

```
if( 'mouse' in cc.sys.capabilities ) {  
    cc.eventManager.addListener({  
        event: cc.EventListener.MOUSE,  
        onMouseDown: function(event){  
            },  
        onMouseMove: function(event){  
            },  
        onMouseUp: function(event){  
            }  
    }, this);  
}
```



注: `if ( 'mouse' in cc.sys.capabilities ){}`

# 加速计事件

---

```
if ('accelerometer' in cc.sys.capabilities) {  
    cc.eventManager.addListener({  
        event: cc.EventListener.ACCELERATION,  
        callback: function (accelEvent, event) {  
            //accelEvent.x  
            //accelEvent.timestamp  
        }  
    }, this);  
}
```

注： if ( ' accelerometer' in cc.sys.capabilities){}

# 自定义事件

---

- 事件创建与分发

- var event = new cc.EventCustom( "event\_name" );
- event.setUserData( "Data" );
- cc.eventManager.dispatchEvent(event);

- 事件响应

```
var listener = cc.EventListener.create({  
    event: cc.EventListener.CUSTOM,  
    eventName: " event_name",  
    callback: function(event){ cc.log("Custom event received, " + event.getUserData( ) ); }  
});  
cc.eventManager.addListener(listener , 1);
```



Thank You!