链滴

开源跨平台移动项目 Ngui【Action 动作系统】

作者：louistru

原文链接：https://ld246.com/article/1511526655820

来源网站：链滴

Ngui简介

这是一个GUI的排版显示引擎和跨平台的GUI应用程序开发框架，基于NodeJS/OpenGL，这也是第个在移动端Android/iOS融合NodeJS的前端GUI项目，至此JavaScript成为了真正意义上前后端通吃语言。

Ngui的目标：在此基础上开发GUI应用程序可拥有开发WEB应用般简单与速度同时兼顾Native应用序的性能与体验。

- 开源跨平台移动项目Ngui【简介】
- 开源跨平台移动项目Ngui【入门】
- 开源跨平台移动项目Ngui【视图与布局系统】
- 开源跨平台移动项目Ngui【Action动作系统】
- 开源跨平台移动项目Ngui【CSS样式表规则及用法】
- Ngui API 文档

什么是Action动作

什么是动作呢？顾名思义它是管理运行环境中所有动作的中枢，通俗点讲就是动画。它也是总个框架心组件之一，它提供动作的创建、删除、插入，以及提供对关键帧与过渡的诸多操作。关键帧的过渡以使用三次贝塞尔曲线，或内置的曲线 linear/ease/ease_in/ease_out/ease_in_out，这也和大多数流框架以及游戏引擎类似。

动作是什么原理

动作怎么驱动视图进行流畅运动的呢？其实原理很简单，我们可以把动作系统看做一个独立的系统与图或渲染完全不相关。它们之间的关系是动作自身的变化最终会映射调视图，这个过程是通过调用视暴露的公有方法或属性来完成的。这个过程完全是单向的，且视图不会向动作发出任何指令。

比如说现在创建了一个新的关键帧动作，给它设置两个关键帧，且x的值经过1秒钟从0变化到100。个过程是动作自身发生的变化并且带动与之相关的视图一同发生改变，请记住这个过程视图是被动的而动作才是主动的发生改变。

```
import { ngui, Div } from 'ngui';
import KeyframeAction from 'ngui/action';
var div = new Div();
var act = new KeyframeAction();
act.add({ x: 0, time: 0 });
act.add({ x: 100, time: 1e3/*毫秒*/ });
div.width = 50;
div.height = 50;
div.backgroundColor = '#f00';
div.action = act;
div.appendTo(ngui.root);
act.play();
```

动作类别

以下是框架提供的几个类型与继承联系

注：带*号的为抽象类型或协议没有构造函数

- Action*
  - GroupAction*
    - SpawnAction
    - SequenceAction
  - KeyframeAction

Action*

这是所有动作的基础类型，也是抽象类型不可以直接被实例。

提供了一些基本的api操作，播放，停止，跳转 等，具体可查看API手册。

GroupAction*

这是个集合的动作类型，提供子动作的添加删除插入。有了子动作就可以帮你实现更加复杂的动画场。

它也有两个具体的子类型 SpawnAction 、SequenceAction。

SpawnAction

并行动作顾名思义即就是它的子动作都是并行运行的。并且以最长子动作的时长做为自身的时长来执动作，较短时长的子动作则在结尾等待动作的结束或一个循环的的终止。

SequenceAction

串行动作这个比较好理解，子动作一个接着一个执行，全部执行完成后结束或开始新的循环。

KeyframeAction与Frame

关键帧动作这是动作系统的核心。所有动作的实现均在这里完成它是动作系统基本单元，前面的GrouAction只有包含KeyframeAction类型的动作才有真正的义意。

而关键帧动作又包含理更加基本的元素关键帧Frame,关键帧包含的属性与CSS属性是同名的且与所有图的属性都是对应关键。通俗的说比如View上会有x属性而Frame上也会有x属性，如果关键帧上有视上并不存在的属性，那么这个属性对视图是无效的。比如View上就不存在width属性那么这个属性的变不会影响到View,但如果绑定的视图是Div那么width的改变一定会影响到它，这与CSS样式表类似。

看下面的例子:

```
// 这是有效的动作
var act1 = new KeyframeAction();
var div = new Div();
div.backgroundColor = '#f00';
act1.add({ width: 10, height: 10 });
act1.add({ width: 100, height: 100, time: 1e3 });
div.action = act1;
```

```
act1.paly();
// 这是无效的
var act2 = new KeyframeAction();
var view = new View();
act2.add({ width: 10, height: 10 });
act2.add({ width: 100, height: 100, time: 1e3 });
view.action = act2;
act2.paly();
```

View.action属性

View.action做为View的一个属性可接收多种类型的参数，之前给大家展示的例子中创建动作是很繁的，但active提供多种类型的参数类型的支持，包括json数据与Action对像实例本身。前面的例子中介绍过Action方式，下面着重说json数据方法。大家也可研读ngui.js与action.js中的源代码，其它View.action属性只是做简单的调用转发，功能的实现其实是在action.js文件中的create()方法里实现的。

看例子:

```
// 这是创建KeyframeAction
var div = new Div();
div.action = {
    keyframe: [
        { x: 0 },
        { x: 100, time: 500 },
        { x: 0, time: 1000 },
    ],
    loop: -1,
};
var div2 = new Div();
div.action = [
    { x: 0 },
    { x: 100, time: 500 },
];

// 这是创建SequenceAction
var div3 = new Div();
div3.action = {
    seq: [
        [ // 这是个子KeyframeAction
            { x: 0 },
            { x: 100, time: 1e3 },
        ],
        { // 这是个子SpawnAction
            spawn: [
                [ // 这是个子KeyframeAction
                    { y : 100 }, { y: 200, time: 2e3 }
                ],
                [ // 这是个子KeyframeAction
                    { width : 200 }, { width: 100, time: 1e3 }
                ],
            ]
        }
    ],
};
```

```
// 这是创建SpawnAction
var div4 = new Div();
div4.action = {
    spawn: [ // 这里只包含一个子KeyframeAction
        {x: 0}, {x: 200, time: 2e3}
    ]
};
```

大家可以看到上面的例子中有4种典型的创建方法。主要看你给的json数据是否存在这三个属性seq、spawn、keyframe，对应SpawnAction、SequenceAction、KeyframeAction，外加一个json数据类检查，数据类型为数组就创建KeyframeAction。并且这可以嵌套使用。

View.transition()方法

这是一个简单创建简单过渡动画的方法，实现原型为action.js的transition()方法，与View.action一样iew.transition()只做简单的转发。

典型应用：

```
view.transition({
    time: 1000,
    y: 100,
    x: 100,
})
```

具体可查阅手册。

View.onActionKeyframe与View.onActionLoop

这两个事件是由动作产生并发送的。

- View.onActionKeyframe为动作执行到达关键帧时触发。因为画面渲染是固定的帧率，触发总是发在帧的渲染时，所以可能会与理想中的时间值有所误差提前或延后，这个延时值会保存在事件数据的dlay上。提前为负数，延时为正数。
- View.onActionLoop动作循环开始时触发，第一次执行动作并不会触发。同样它也会有延时，也同记录在delay。

ActionGroupActionSpawnActionSequenceAction
eyframeActionFrameViewDivVie
.actionView.transition()