

Cocos2d-JS游戏开发

---动作与动画

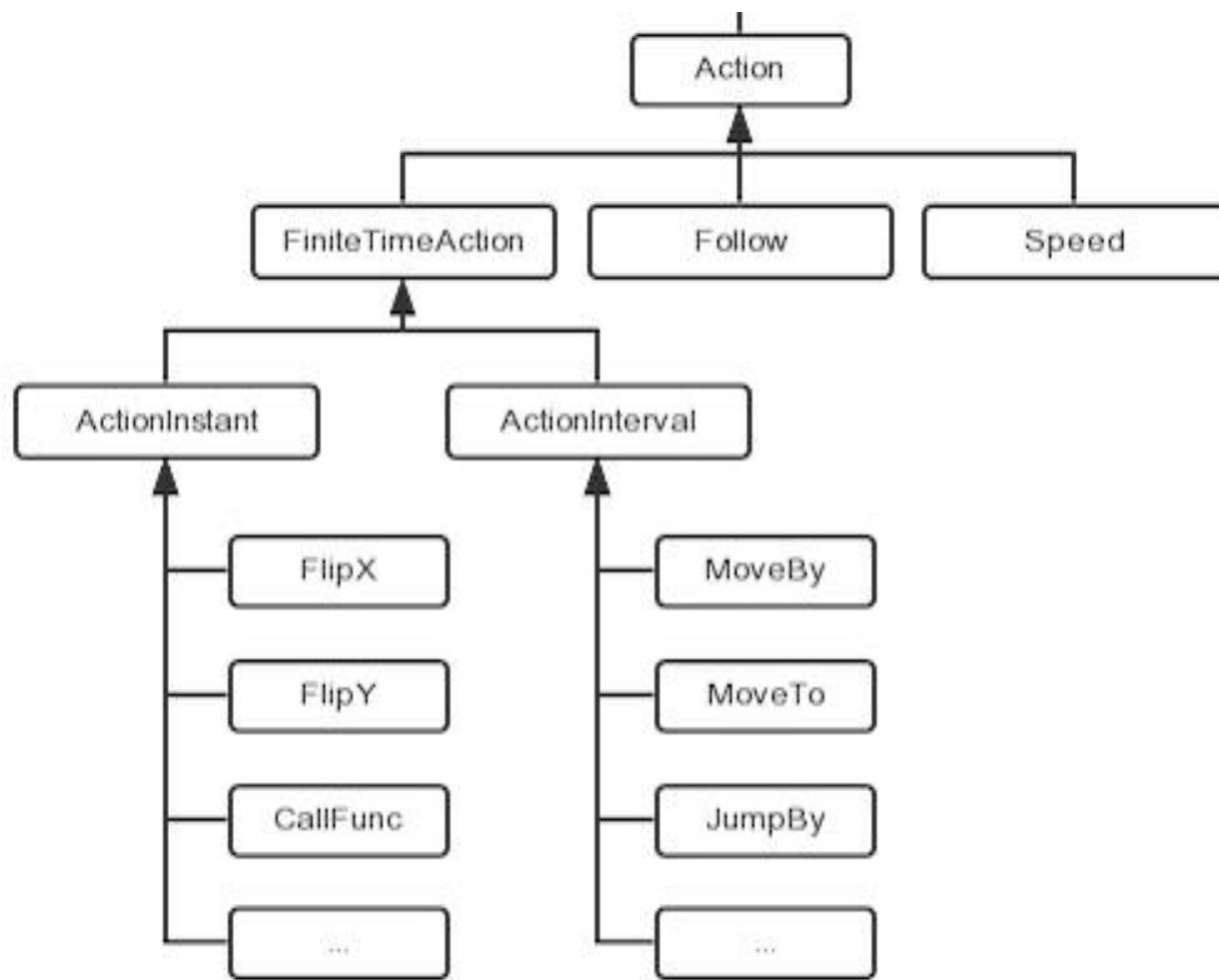
核心概念

- 坐标系、节点 (Node)
- 定时器 (Schedule)
- 标签及菜单 (Label、Menu、MenuItem)
- 场景与层 (Scene、Layer)
- 导演 (Director)
- 精灵 (Sprite)
- 动作与动画 (Action、Animation)

动作 (Actions) 定义

- 动作类(cc.Action)是所有动作的基类，它实例化出来的对象代表一个动作（一般由子类来实例化具体动作）
- 动作作用于节点对象，每个动作都需要由节点对象来执行（如：精灵、按钮、层等）
- 在实际开发中，通常用到两类动作：**即时动作**和**间隔动作**，这两类动作都继承于有限时间动作类(FiniteTimeAction)

动作 (Actions) 相关类图



控制动作

- 创建动作

- var action = new cc.MoveBy(...);
- var action = cc.MoveBy.create(...);
- var action = cc.moveBy(...);

- 运行动作

- node.runAction(action);

- 停止动作

- node.stopAction(action);
- node.stopActinByTag(tag);
- node.stopAllAction();

控制动作

- 暂停/恢复动作

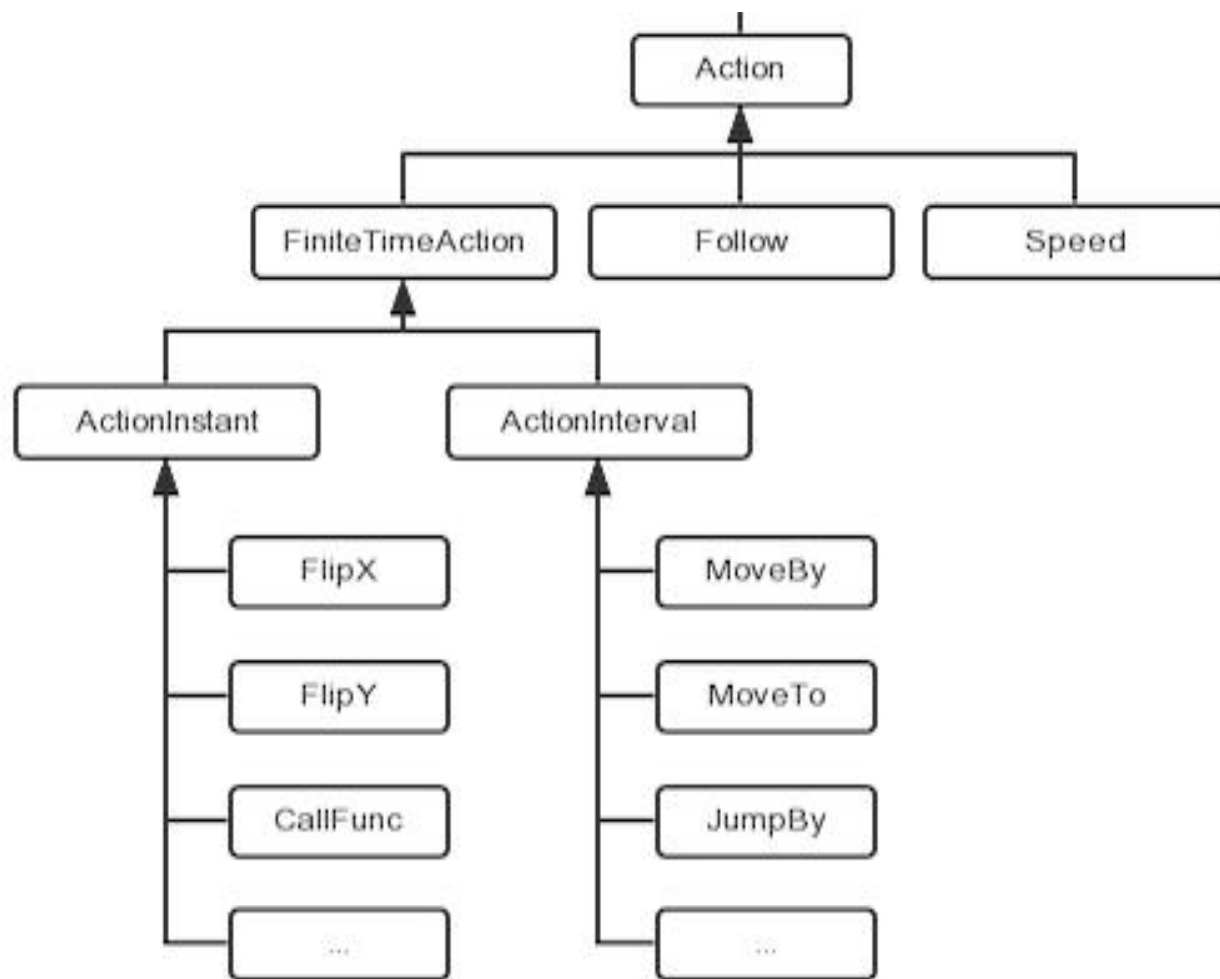
- `node.pause();`
- `node.resume();`

- 全局控制

- `cc.director.pause();`
- `cc.director.resume();`



动作 (Actions) 相关类图



即时动作 (ActionInstant)

- 位置调整

- cc.place(position)

- 水平/垂直反转

- cc.flipX(boolean)

- cc.flipY(boolean)

- 隐藏/显示

- cc.hide()

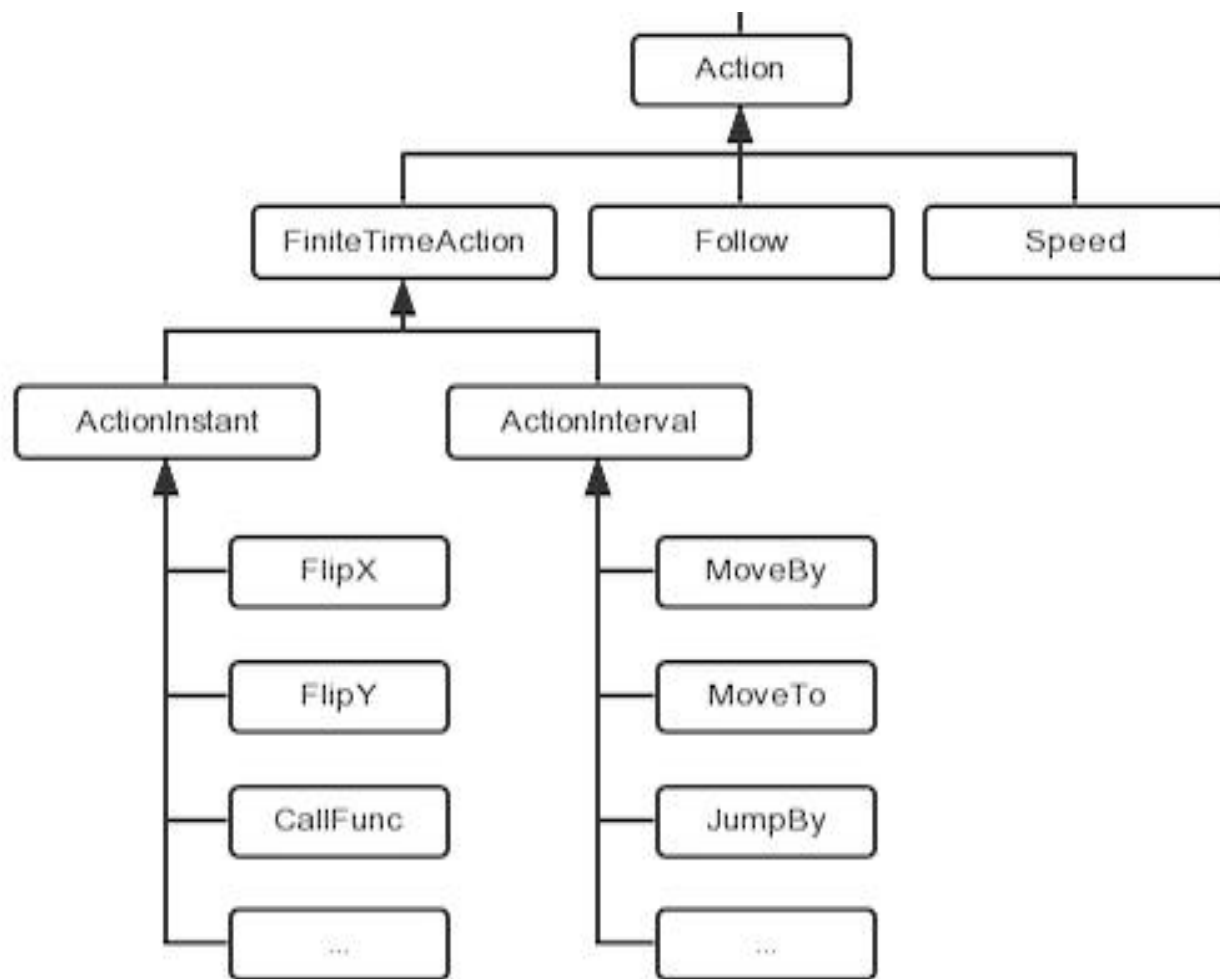
- cc.show()

- 回调动作

- cc.callFunc(select,target,data)



动作 (Actions) 相关类图



间隔动作 (ActionInterval)

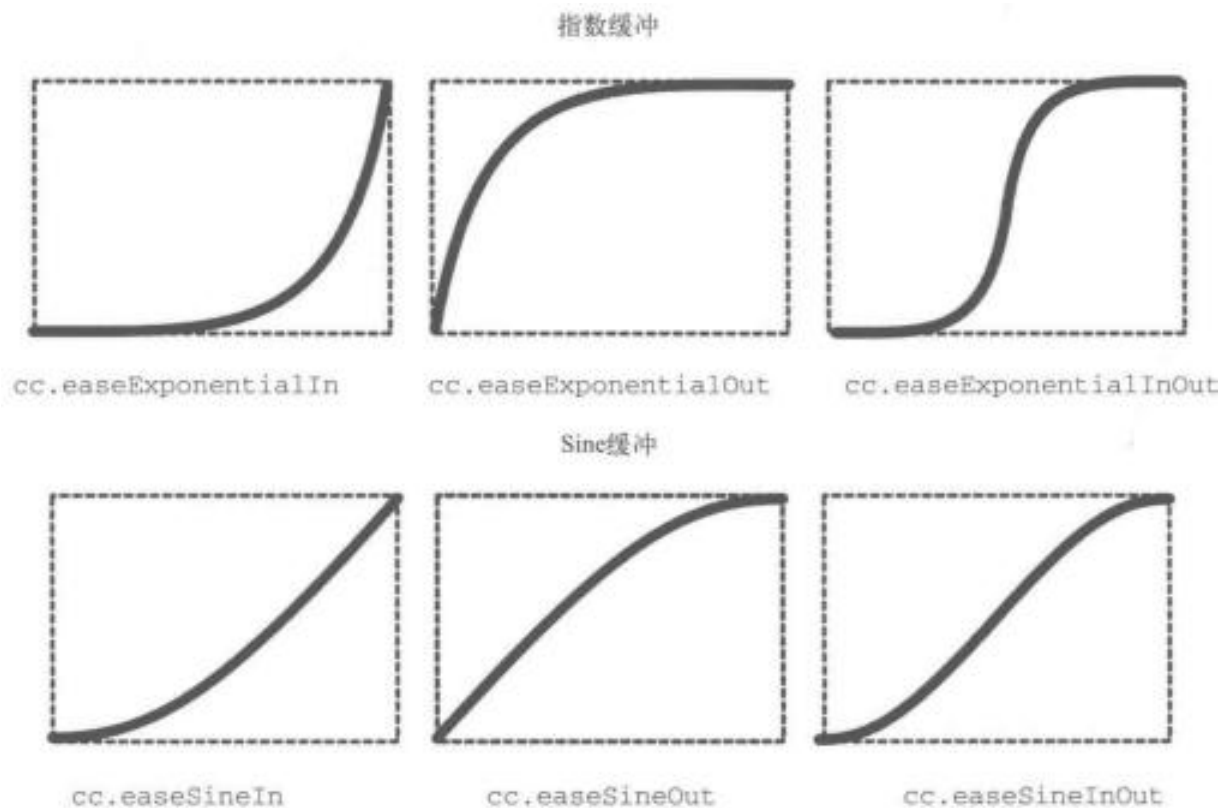
- 移动
 - `cc.moveBy(...); cc.moveTo(...);`
- 跳跃
 - `cc.jumpBy(...); cc.jumpTo(...);`
- 旋转
 - `cc.rotateBy(...); cc.rotateTo(...);`
- 缩放
 - `cc.scaleBy(...); cc.scaleTo(...);`
- 淡入 / 淡出
 - `cc.fadeIn(); cc.fadeOut(); cc.fadeTo();`

间隔动作 (ActionInterval)

- 闪烁
 - `cc.blink(...);`
- 颜色
 - `cc.tintBy(...); cc.tintTo(...);`
- 进度条
 - `var timer = new cc.ProgressTimer(aSprite);`
 - 设置样式
 - `timer.runAction(cc.progressTo(...));`

变速动画 (cc.Speed、cc.ActionEase)

- 变速动作可以让指定的间隔动作发生速度上的变化
 - 线性动作变化 (cc.Speed)、非线性动作变化(cc.ActionEase)



组合动作（实现复杂的动画功能）

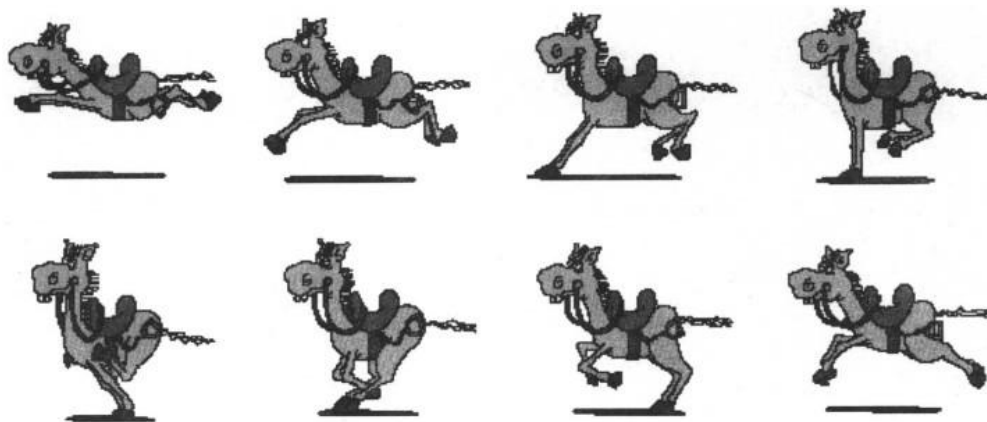
- 顺序执行
 - `cc.sequence(action1,action2,...);`
- 同步执行
 - `cc.spawn(action1,action2,...);`
- 重复执行
 - `cc.repeat(...); cc.repeatForever(...); action.repeat(...); action.repeatForever(...);`
- 延迟执行
 - `cc.delayTime(dt);`
- 反向执行
 - `cc.reverse(...);`

The background of the slide is decorated with various abstract shapes in shades of green and yellow. These shapes, which include circles, ovals, and irregular blobs, are scattered across the top and right sides of the slide, creating a modern, organic feel.

Have a Break!

动画简介

- 帧动画



- 骨骼动画

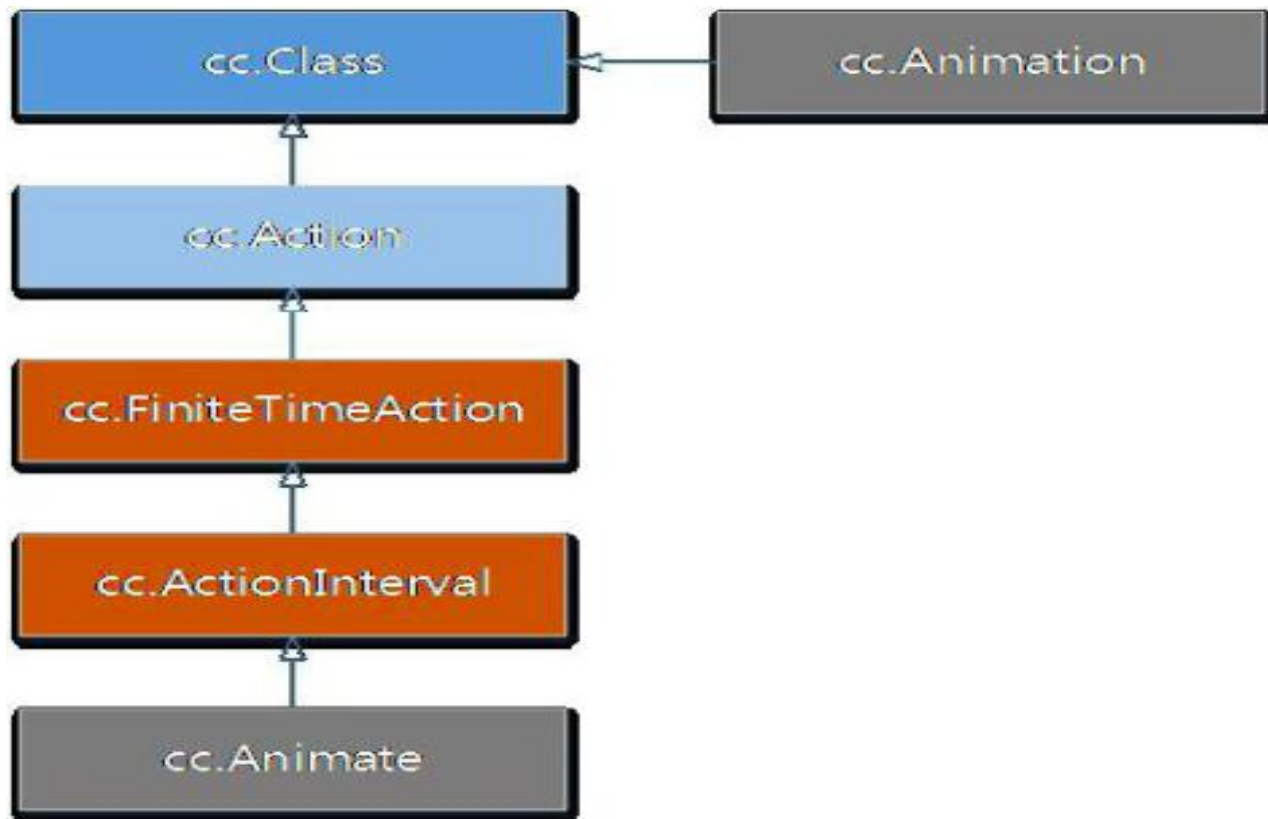


帧动画

- 帧是动画或影像的基本单位。每一帧代表一画面，连续多帧画面组合在一起播放就形成了动画影像，就像电影胶卷连续播放。而帧频就是一秒内帧的数量,通常用FPS (Frames Per Second)表示, 帧频越高, 画面就越流畅
- 一般电影为每秒24帧, 而游戏一般以60fps作为最高帧频, 符合人眼正常识别的最高频率
- project.json 可以设置, 默认60

动画 (Animation)

- 涉及两个类：cc.Animate / cc.Animation



动画 (Animation)

- 帧动画是按一定时间间隔、一定顺序、逐帧地显示图片，与动作密不可分
- cc.Animate类是动作类，属于间隔动作类，它的作用是将cc.Animation定义的动画转换成动作来执行
- cc.Animation类，负责装载帧动画里所有的帧
- 区分Animate与Animation不同的职责

帧动画实现方法 (Animation、Animate)

- 使用方法一:

```
var animation = new cc.Animation();
for(var i = 1; i<=6; i++){
    var frameName = res["Pao_" + i + "_png"];
    animation.addSpriteFrameWithFile(frameName);
}
animation.setDelayPerUnit(1.0/35);
animation.setRestoreOriginalFrame(true);
var action = cc.animate(animation);
this._player.runAction(action.repeatForever());
```



帧动画实现方法 (Animation、Animate)

- 使用方法二:

```
cc.spriteFrameCache.addSpriteFrames(res.Grossini_plist);
var spriteFrames = [];
for(var i = 1; i<=14; i++){
    var frame = cc.spriteFrameCache.getSpriteFrame(
        "grossini_dance_generic_" + i + ".png");
    spriteFrames.push(frame);
}
var animation2 = new cc.Animation(spriteFrames, 0.2, 2);
var animate = cc.animate(animation2);
animation2.setRestoreOriginalFrame(true);
this._grossini.runAction(animate.repeatForever());
```



The background of the slide is decorated with various abstract shapes in shades of green and yellow. These shapes, which include circles, ovals, and teardrops, are scattered across the top and right sides of the slide, creating a modern and organic feel.

Thank You!



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