#### CAViewController

###### 类说明

CAViewController是抽象类，作为[CAView](CAView.docx)的管理器，其最基本的功能就是控制视图的切换。视图控制器在MVC设计模式中扮演控制层(C)的角色, CAViewController的作用就是管理与之关联的view，同时与其他CAViewController相互通信和协调。一个视图控制器管理一个视图（它可以有子视图），其view属性指向它所管理的视图。

###### 基类

[CCObject](../CCObject.docx), [CAViewDelegate](../CAViewDelegate.docx), [CCKeypadDelegate](../CCKeypadDelegate.docx)

###### 属性

|  |  |  |
| --- | --- | --- |
| 访问修饰符 | 属性名 | 说明 |
| protected | [Title](#Title) | 标题 |
| protected | [NavigationController](#NavigationController) | viewController当前管理的navigationController |
| protected | [tabBarController](#tabBarController) | viewController当前管理的tabBarController |

###### 方法

|  |  |  |
| --- | --- | --- |
| 访问修饰符 | 方法名 | 说明 |
| protected | [viewDidLoad](#viewDidLoad) | 视图管理器中的根视图加载完毕后自动触发 |
| protected | [viewDidUnload](#viewDidUnload) | 释放视图管理器的view以及view相关的对象 |
| protected | [reshapeViewRectDidFinish](http://www.9miao.com/index.php?m=content&c=content&a=edit&catid=16&id=1149&pc_hash=tD2pwW#reshapeViewRectDidFinish) | 定义接口，重新设置view |
| protected | [getView](#getView) | 返回当前视图管理器的根view |
| public | [getNibName](#getNibName) | 获取当前视图管理器的类型 |
| public | [setNavigationBarItem](#setNavigationBarItem) | 设置navigationBarItem |
| public | [setTabBarItem](#setTabBarItem) | 设置tabBarItem |
| public | [isKeypadEnabled](#isKeypadEnabled) | 是否响应键盘事件 |
| public | [setKeypadEnabled](#setKeypadEnabled) | 设置键盘响应事件 |
| public | [keyBackCliched](#keyBackClicked) | 响应返回键 |
| public | [keyMenuClicked](#keyMenuClicked) | 响应Menu键 |
| public | [presentModalViewController](#presentModalViewController) | 在当前的viewcontroller中弹出一个新的viewController |
| public | [dismissModalViewController](#dismissModalViewController) | 隐藏新弹出的viewController |

###### 属性介绍

**Title**

类型：string

解释：viewController的标题

**NavigationController**

类型：CANavigationController\*

解释：只读属性

**tabBarController**

类型：CATabBarController\*

解释：只读属性

###### 方法介绍

**virtual void viewDidLoad()=0**

返回值:void

解释：纯虚函数，不需要手动调用此函数，视图加载完成后会自动调用，主要用于进一步初始化视图，提供给开发者编码的接口。

示例：

void FirstViewController::viewDidLoad()

{

CAImageView\* imageView =

CAImageView::createWithImage(CAImage::create("9m.jpg"));

imageView->setFrame(CCRect(100, 100, 0, 0));

this->getView()->addSubview(imageView);

CCLabelTTF\* ttf = CCLabelTTF::create("Hello World!", "Arial", 50);

ttf->setColor(ccc4(0,0,0,255));

ttf->setFrame(CCRect(200, 450, 0, 0));

this->getView()->addSubview(ttf);

}

**virtual void viewDidUnload()=0**

返回值：void

解释：存虚函数，当内存过高时，自动调用此方法进行内存的释放。

**virtual void****reshapeViewRectDidFinish()**

返回值：void

解释：进行某些操作需要改变view的位置和大小时，可重写此方法，并自己实现相应处理逻辑。

**CAView\* getView()**

返回值：CAView\*

解释：返回根视图的对象，viewController会默认创建一个view作为根视图。

**virtual bool init()**

返回值：bool

解释：虚函数，调用方法直接返回true

**const char\* getNibName()**

返回值：const char\*

解释：获取当前的视图管理器是属于哪种视图管理器

示例：

const char\* CAViewController::getNibName()

{

return typeid(\*this).name();

}

**void** **setNavigationBarItem(CANavigationBarItem\* item)**

返回值：void

解释：设置当前naviagtionController所管理的navigationBar的navigationBarItem

**void** **setTabBarItem(CATabBarItem\* item)**

返回值：void

解释：设置当前tabBarController所管理的tabBar的tabBarItem

**virtual bool isKeypadEnabled**()

返回值：bool

解释：虚函数，返回一个布尔值，判断是否响应键盘事件

**virtual void setKeypadEnabled**(bool value)

返回值：void

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| bool | value | 响应键盘事件标志位 |

解释：虚函数，传递一个布尔值，确定是否响应键盘事件。

示例：

void CAViewController::setKeypadEnabled(bool enabled)

{

if (enabled != m\_bKeypadEnabled)

{

m\_bKeypadEnabled = enabled;

CCDirector\* pDirector = CCDirector::sharedDirector();

if (enabled)

{

pDirector->getKeypadDispatcher()->addDelegate(this);

}

else

{

pDirector->getKeypadDispatcher()->removeDelegate(this);

}

}

}

**virtual void** **keyBackClicked()**

返回值：void

解释：响应设备返回键操作的接口

**virtual void** **keyMenuClicked()**

返回值：void

解释：响应设备菜单键操作的接口

**void** **presentModalViewController(CAViewController\* controller, bool animated)**

返回值：void

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| CAViewController | controller | 新的viewController |
| bool | animated | 是否播放切换动画 |

解释：默认从屏幕的底部向上弹出一个新的viewController，将原先的viewController覆盖。

**void** **dismissModalViewController(bool animated)**

返回值：void

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| bool | animated | 是否播放切换动画 |

解释：隐藏新弹出的viewController。