#### CANavigationController

###### 类说明

导航控制器CANavigationController同[CATabBarController](CATabBarController.docx)一样，也是负责协调多个CAViewController之间的管理，但tabBarController所管理的viewController之间是出于平级的，而navigationController管理的viewController之间是堆栈的关系，所谓的堆栈关系即每添加一个view则进行一次入栈操作，每返回一次则进行一次出栈操作。

###### 基类

[CAViewController](../CAViewController.docx)，[CANavigationBarDelegate](../../代理类/CANavigationBarDelegate.docx)

###### 属性

|  |  |  |
| --- | --- | --- |
| 访问修饰符 | 属性名 | 说明 |
| protected | [NavigationBarHidden](#NavigationBarHidden) | navigationBar是否隐藏 |
| protected | [NavigationBar](#NavigationBar) | 当前的navigationBar |
| protected | [NavigationBarVerticalAlignment](#NavigationBarVerticalAlignment) | navigationBar的位置 |

###### 方法

|  |  |  |
| --- | --- | --- |
| 访问修饰符 | 方法名 | 说明 |
| public | [initWithRootViewController](#initWithRootViewController) | 使用一个viewController来构建CANavigationController |
| public | [pushViewController](#pushViewController) | 将新的viewController压入栈顶 |
| public | [popViewControllerAnimated](#popViewControllerAnimated) | 移除栈顶的viewController |
| public | [setNavigationBarHidden](#setNavigationBarHidden) | 设置是否隐藏navigationBar |
| public | [updateItem](#updateItem) | 更新navigationBarItem |

###### 属性介绍

**NavigationBarHidden**

类型：bool

解释：只读属性，返回一个布尔值判断navigationBar是否隐藏。

**NavigationBar**

类型：CANavigationBar\*

解释：只读属性，用于标识当前CANavigationController中的navigationBar，通过getNavigationBar方法获取。

**NavigationBarVerticalAlignment**

类型：CABarVerticalAlignment

解释：只读属性，枚举类型，获取navigationController的navigationBar的位置，默认在屏幕的顶部。

typedef enum

{

CABarVerticalAlignmentTop = 0,

CABarVerticalAlignmentBottom

}CABarVerticalAlignment;

###### 方法介绍

**bool** **initWithRootViewController(CAViewController\* viewController，**

**CABarVerticalAlignment var = CABarVerticalAlignmentTop)**

返回值：bool

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| CAViewController\* | viewController | 用于构建navigationController的viewController |

解释：如果没有传递var参数，则默认在屏幕顶部。

示例：

CAWindow\* \_window = CAWindow::create();

NewViewController\* viewController1 = new NewViewController();

viewController1->init();

viewController1->setNavigationBarItem(CANavigationBarItem::create("viewController"));

viewController1->setTitle("view1");

CANavigationController\* navigationController = new CANavigationController();

navigationController->initWithRootViewController(viewController1);

navigationController->setTabBarItem(CATabBarItem::create("First", CAImage::create("h.png"), CAImage::create("h.png")));

SecondViewController\* viewController2 = new SecondViewController();

viewController2->init();

viewController2->setTabBarItem(CATabBarItem::create("Second", CAImage::create("e.png"), CAImage::create("e.png")));

viewController2->setTitle("view2");

ThirdViewController\* viewController3 = new ThirdViewController();

viewController3->init();

viewController3->setTabBarItem(CATabBarItem::create("Third", CAImage::create("a.png"), CAImage::create("a.png")));

viewController3->setTitle("view3");

FourthViewController\* viewController4 = new FourthViewController();

viewController4->init();

viewController4->setTabBarItem(CATabBarItem::create("Fourth", CAImage::create("o.png"), CAImage::create("o.png")));

viewController4->setTitle("view4");

FifthViewController\* viewController5 = new FifthViewController();

viewController5->init();

viewController5->setTabBarItem(CATabBarItem::create("Fifth", CAImage::create("s.png"), CAImage::create("s.png")));

viewController5->setTitle("view5");

std::vector<CAViewController\*> views;

views.push\_back(navigationController);

views.push\_back(viewController2);

views.push\_back(viewController3);

views.push\_back(viewController4);

views.push\_back(viewController5);

CATabBarController\* tabBarController = new CATabBarController();

tabBarController->initWithViewControllers(views);

tabBarController->getTabBar()->showSelectedIndicator();

viewController1->release();

navigationController->release();

viewController2->release();

viewController3->release();

viewController4->release();

viewController5->release();

\_window->setRootViewController(tabBarController);

tabBarController->release();

return \_window;

上面的代码和我们在[CATabBarController](CATabBarController.docx)中的用法一样，针对第一个子项做了适当的修改，增加了一个CANavigationController，如下图运行所示的顶部栏，并在view中添加了一个按钮[CAButton](../CAView/CAControl/CAButton.docx)。



**void** **pushViewController(CAViewController\* viewController, bool animated)**

返回值：void

示例：

bool NewViewController::buttonCallback(CAButton\* btn, CCPoint point)

{

NewViewController\* viewController = new NewViewController();

viewController->init();

viewController->setNavigationBarItem(CANavigationBarItem::create("The next view"));

viewController->setTitle("New View");

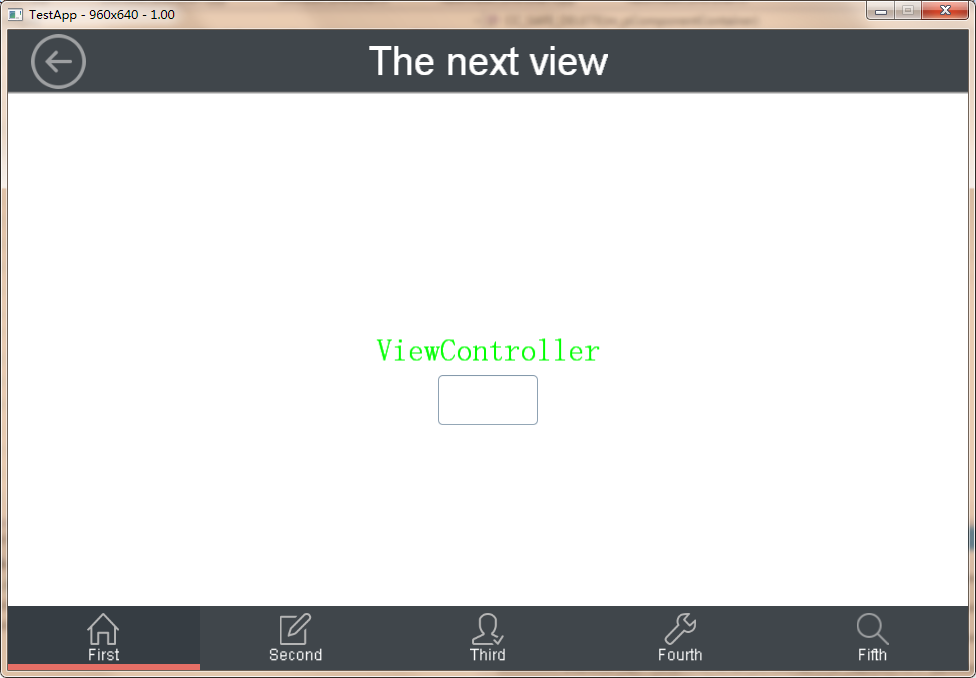
this->getNavigationController()->pushViewController(viewController,true);

viewController->autorelease();

return false;

}

在按钮的点击事件中，我们将新的viewController压入栈顶显示出来，当前栈内元素为2，所以在导航栏中默认为我们创建了一个返回按钮，用于返回上一层，点击返回按钮时将调用popViewControllerAnimated方法。



当然，我们可以不断的点击按钮，产生新的viewController，并自动进行入栈操作。

**CAViewController\*** **popViewControllerAnimated(bool animated)**

返回值：CAViewController\*

解释：移除栈顶的CAViewController并显示上一个，animated为true是播放切换动画，并获取到被移除的CAViewController

**void** **setNavigationBarHidden(bool hidden, bool animated)**

返回值：void

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| bool | hidden | 是否隐藏 |
| bool | animated | 是否播放隐藏动画 |

**void** **updateItem(CAViewController\* viewController)**

返回值：void

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| CAViewController\* | viewController | 需要更新的视图管理器 |