#### CACollectionView

###### 类说明

同CATableView类似，主要用于数据的展示，实现了tableView的基本功能，同时对tableView拓展，更完美的进行展示数据。

###### 效果图

###### 基类

[CAScrollView](../CAScrollView.docx)

###### 属性

|  |  |  |
| --- | --- | --- |
| 访问修饰符 | 属性名 | 说明 |
| protected | [CollectionViewDataSource](#CollectionViewDataSource) | 数据代理 |
| protected | [CollectionViewDelegate](#CollectionViewDelegate) | 点击事件代理 |
| protected | [CollectionHeaderView](#CollectionHeaderView) | section的头部视图 |
| protected | [CollectionFooterView](#CollectionFooterView) | section的尾部视图 |
| protected | [CollectionHeaderHeight](#CollectionHeaderHeight) | 头部视图的高度 |
| protected | [CollectionFooterHeight](#CollectionFooterHeight) | 尾部视图的高度 |
| protected | [AllowsSelection](#AllowsSelection) | cell是否允许被选择 |
| protected | [AllowsMultipleSelection](#AllowsMultipleSelection) | cell是否允许被多选 |

###### 方法

|  |  |  |
| --- | --- | --- |
| 访问修饰符 | 方法名 | 说明 |
| public | [reloadData](#reloadData) | 刷新tableView |
| public | [setSelectRowAtIndexPath](#setSelectRowAtIndexPath) | 设置当前选中的item |

###### 属性介绍

**CollectionViewDataSource**

类型：CACollectionViewDataSource\*

解释：使用collectionView时必须要设置数据代理，代理类的详细接口请参考[CACollectionViewDataSource](../../../Delegate/CACollectionViewDataSource.docx)，public get/set{}。

示例：

CACollectionView\* p\_Conllection=CACollectionView::createWithCenter(CCRect(size.width\*0.5,size.height\*0.5,size.width,size.height));

p\_Conllection->setCollectionViewDataSource(this);

this->getView()->addSubview(p\_Conllection);

**CollectionViewDelegate**

类型：CACollectionViewDelegate\*

解释：点击事件的代理，代理类中包括响应选中cell和取消选中cell两个接口，使用时可根据实际情况选择是否需要设置此代理，代理类的详细接口请参考[CACollectionViewDelegate](../../../Delegate/CACollectionViewDelegate.docx)，public get/set{}。

示例：

CACollectionView\* p\_Conllection=CACollectionView::createWithCenter(CCRect(size.width\*0.5,size.height\*0.5,size.width,size.height));

p\_Conllection->setCollectionViewDelegate(this);

p\_Conllection->setCollectionViewDataSource(this);

this->getView()->addSubview(p\_Conllection);

**CollectionHeaderView**

类型：CAView\*

解释：collectionView的头部视图，public get/set{}。

**CollectionFooterView**

类型：CAView\*

解释：collectionView的尾部视图，public get/set{}。

**CollectionHeaderHeight**

类型：unsigned int

解释：collectionView的头部视图的高度，public get/set{}。

**CollectionFooterHeight**

类型：unsigned int

解释：collectionView的尾部视图的高度，public get/set{}。

**AllowsSelection**

类型：bool

解释：cell是否可选，public is/set{}。

示例：

CACollectionView\* p\_Conllection=CACollectionView::createWithCenter(CCRect(size.width\*0.5,size.height\*0.5,size.width,size.height));

p\_Conllection->setCollectionViewDelegate(this);

p\_Conllection->setCollectionViewDataSource(this);

p\_Conllection->setAllowsSelection(true);

this->getView()->addSubview(p\_Conllection);

**AllowsMultipleSelection**

类型：bool

解释：cell是否可多选，必须在cell设置为可选时才能设置多选，public is/set{}。

示例：

CACollectionView\* p\_Conllection=CACollectionView::createWithCenter(CCRect(size.width\*0.5,size.height\*0.5,size.width,size.height));

p\_Conllection->setCollectionViewDelegate(this);

p\_Conllection->setCollectionViewDataSource(this);

p\_Conllection->setAllowsSelection(true);

p\_Conllection->setAllowsMultipleSelection(true);

this->getView()->addSubview(p\_Conllection);

###### 方法介绍

**void** **reloadData()**

返回值：void

解释：刷新当前的collectionView，collectionView的任何数据变动，都需要调用此方法进行强制刷新。

**void** **setSelectRowAtIndexPath(unsigned int section, unsigned int row, unsigned int item)**

返回值：void

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| unsigned int | section | 点击的section |
| unsigned int | row | 选中的cell |
| unsigned int | item | 选中的item |

解释：通过调用此接口，可以设置当前collectionView选中的数据。

###### 用法参考

collectionView的具体用法请参考CrossApp官方Demo的CollectionViewTest示例，路径：CrossApp \samples\demo\Classes\CollectionViewTest。