#### CASegmentedControl

###### 类说明

分段按钮，segmentedControl由一个或多个segment组成，点击不同段得到不同的信息，每次只能选择一段。

###### 基类

[CAControl](../CAControl.docx)

###### 属性

|  |  |  |
| --- | --- | --- |
| 访问修饰符 | 属性名 | 说明 |
| public | [Target](#Target) | 点击事件 |
| public | [Title](#Title) | 标题 |
| public | [BackGroundImage](#BackGroundImage) | 背景图片 |
| public | [Image](#Image) | 前景图片 |
| public | [TitleColor](#TitleColor) | 标题颜色 |
| protected | [selectedIndex](#selectedIndex) | segment的索引值 |
| protected | [ItemSize](#ItemSize) | segment的size |

###### 方法

|  |  |  |
| --- | --- | --- |
| 访问修饰符 | 方法名 | 说明 |
| public | [createWithFrame](#createWithFrame) | 创建segmentedControl |
| public | [createWithCenter](#createWithCenter) | 创建segmentedControl |
| public | [addTarget](#addTarget) | 添加点击事件 |
| public | [removeTarget](#removeTarget) | 移除点击事件 |
| public | [insertSegmentWithTitle](#insertSegmentWithTitle) | 插入新段 |
| public | [insertSegmentWithBackgroundImage](#insertSegmentWithBackgroundImage) | 插入新段 |
| public | [insertSegmentWithImage](#insertSegmentWithImage) | 插入新段 |
| public | [removeSegmentAtIndex](#removeSegmentAtIndex) | 移除segment的某一段 |
| public | [removeAllSegments](#removeAllSegments) | 移除所有段 |
| public | [setTitleAtIndex](#setTitleAtIndex) | 设置某一段的标题 |
| public | [setTitleColorAtIndex](#setTitleColorAtIndex) | 设置某一段的标题颜色 |
| public | [setBackgroundImageAtIndex](#setBackgroundImageAtIndex) | 设置某一段的背景图片 |
| public | [setImageAtIndex](#setImageAtIndex) | 设置某一段的前景图片 |
| public | [setSelectedAtIndex](#setSelectedAtIndex) | 设置被选中的segment |
| public | [getItemCount](#getItemCount) | 获取segment的数量 |

###### 属性介绍

**Target**

类型：SEL\_CAControl

解释：函数回调器，响应segment的点击事件

**Title**

类型：const char\*

解释：segment的标题

**BackGroundImage**

类型：CAImage\*

解释：segment的背景图片

**Image**

类型：CAImage\*

解释：segment的前景图片

**TitleColor**

类型：CAColor4B

解释：segment标题颜色

**selectedIndex**

类型：int

解释：只读属性，segment对应的索引值。

**ItemSize**

类型：CCSize

解释：只读属性，segment的size，每个segment的size都一样。

###### 方法介绍

**static CASegmentedControl\*** **createWithFrame(const CCRect& rect, unsigned int**

**itemsCount)**

返回值：CASegmentedControl\*

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| const CCRect | rect | segmentedControl的frame |
| unsigned int | itemsCount | segment的数量 |

解释：创建一个segmentedControl，并指定rect和segment的数量。

**static CASegmentedControl\*** **createWithCenter(const CCRect& rect, unsigned int**

**itemsCount)**

返回值：CASegmentedControl\*

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| const CCRect | rect | segmentedControl的center |
| unsigned int | itemsCount | segment的数量 |

解释：创建一个segmentedControl，并指定center和segment的数量。

**void** **addTarget(CAObject\* target, SEL\_CAControl selector)**

返回值：void

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| CAObject\* | target | 当前对象 |
| SEL\_CAControl | selector | 函数回调器 |

解释：segment的点击事件

**void** **removeTarget(CAObject\* target, SEL\_CAControl selector)**

返回值：void

解释：移除掉函数回调器

**virtual bool** **insertSegmentWithTitle(const char\* title, int index, CAControlState**

**controlState)**

返回值：bool

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| const char\* | title | segment显示的标题 |
| int | index | 插入segment的位置 |
| CAControlState | controlState | segment的状态 |

解释：在当前的segmentedControl中插入新的segment，插入的位置为传递的index参数(segment的index值默认从0开始)。注意：如果传递的index值和已有的segment的index值相等，则原有的segment的index值及后续所有index值都加1，这条规则适用于segment的所有insert方法。

示例：

CCRect segmentedControl\_rect = CCRect(size.width/2, off\_Y \* 3 + 60, size.width \* 0.8f, height);

segmentedControl = CASegmentedControl::createWithCenter(segmentedControl\_rect, 2);

segmentedControl->insertSegmentWithTitle("2", 1, CAControlStateAll);

segmentedControl->insertSegmentWithTitle("3", 3, CAControlStateAll);

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

创建一个segmentedControl，包含两个segment，index和title值都默认从0开始编号，插入新的segment位置发生变化。



**virtual bool** **insertSegmentWithBackgroundImage(CAImage \*image, int index,**

**CAControlState controlState)**

返回值：bool

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| CAImage\* | image | segment背景图片 |
| int | index | segment的插入位置 |
| CAControlState | controlState | segment的状态 |

解释：根据传递的参数，插入segment并设置背景图片，标题默认为空。

**virtual bool** **insertSegmentWithImage(CAImage \*image, int index, CAControlState**

**controlState)**

返回值：bool

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| CAImage\* | image | segment前景图片 |
| int | index | segment的插入位置 |
| CAControlState | controlState | segment的状态 |

解释：根据传递的参数，插入segment并设置前景图片，标题默认为空。

**virtual void** **removeSegmentAtIndex(int index)**

返回值：void

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| int | index | segment的序号 |

解释：根据索引值index，移除指定segment。

**virtual void** **removeAllSegments()**

返回值：void

解释：移除所有的segment

**virtual bool** **setTitleAtIndex(const char\* title, int index, CAControlState controlState)**

返回值：bool

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| const char\* | title | segment标题 |
| int | index | segment的位置 |
| CAControlState | controlState | segment的状态 |

解释：根据传递的参数，设置segment的标题。

**virtual bool** **setTitleColorAtIndex(CAColor4B color, int index, CAControlState controlState)**

返回值：bool

参数：

|  |  |  |
| --- | --- | --- |
| 类型 | 参数名 | 说明 |
| CAColor4B\* | color | 标题的颜色 |
| int | index | segment的位置 |
| CAControlState | controlState | segment的状态 |

解释：根据指定参数，设置segment标题的颜色。

**virtual bool** **setBackgroundImageAtIndex(CAImage \*image, int index, CAControlState**

**controlState)**

返回值：bool

解释：根据传递参数，设置指定segment的背景图片。

**virtual bool** **setImageAtIndex(CAImage \*image, int index, CAControlState controlState)**

返回值：bool

解释：根据传递参数，设置指定segment的前景图片。

**void** **setSelectedAtIndex(int index)**

返回值：void

解释：根据当前index值，设置被选中的segment。

**int** **getItemCount() const**

返回值：int

解释：获取segment的数量

示例：

CCRect segmentedControl\_rect = CCRect(size.width/2, off\_Y \* 3 + 60, size.width \* 0.8f, height);

segmentedControl = CASegmentedControl::createWithCenter(segmentedControl\_rect, 2);

segmentedControl->insertSegmentWithTitle("2", 1, CAControlStateAll);

segmentedControl->insertSegmentWithTitle("3", 3, CAControlStateAll);

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

segmentedControl->addTarget(this,CAControl\_selector(FirstViewController::segmentCallback));

void FirstViewController::segmentCallback(CAControl\* btn, CCPoint point)

{

char desc[50];

if (content)

{

this->getView()->removeSubview(content);

}

content = CALabel::createWithCenter(CCRect(size.width\*0.5, 450, 150, 80));

content->setFontSize(30);

content->setTextAlignment(CATextAlignmentCenter);

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

for (int i = 0; i < segmentedControl->getItemCount(); i++)

{

if (i == segmentedControl->getselectedIndex())

{

sprintf(desc,"当前选择为%d",i);

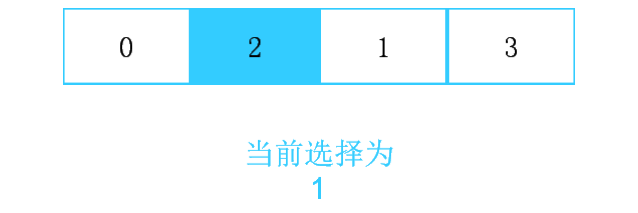
content->setText(desc);

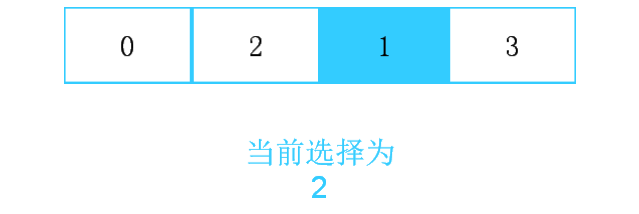
content->setColor(ccc4(51,204,255,255));

}

}

}

****

****