**Menu**

class CC\_DLL Menu : public Layer

菜单在init的时候就添加了单点触控监听器：

auto touchListener = EventListenerTouchOneByOne::create();

touchListener->setSwallowTouches(true);

touchListener->onTouchBegan = CC\_CALLBACK\_2(Menu::onTouchBegan, this);

touchListener->onTouchMoved = CC\_CALLBACK\_2(Menu::onTouchMoved, this);

touchListener->onTouchEnded = CC\_CALLBACK\_2(Menu::onTouchEnded, this);

touchListener->onTouchCancelled = CC\_CALLBACK\_2(Menu::onTouchCancelled, this); \_eventDispatcher->addEventListenerWithSceneGraphPriority(touchListener, this);

看看它的触控：

**bool Menu::onTouchBegan(Touch\* touch, Event\* event)**

{

auto camera = Camera::getVisitingCamera();

if (\_state != Menu::State::WAITING || ! \_visible || !\_enabled || !camera)

{

return false;

}

for (Node \*c = this->\_parent; c != nullptr; c = c->getParent())

{

if (c->isVisible() == false)

{

return false;

}

}

**\_selectedItem = this->getItemForTouch(touch, camera);**

if (\_selectedItem)

{

\_state = Menu::State::TRACKING\_TOUCH;

\_selectedWithCamera = camera;

\_selectedItem->selected();

return true;

}

return false;

}

void Menu::onTouchEnded(Touch\* touch, Event\* event)

{

CCASSERT(\_state == Menu::State::TRACKING\_TOUCH, "[Menu ccTouchEnded] -- invalid state");

this->retain();

if (\_selectedItem)

{

\_selectedItem->unselected();

\_selectedItem->activate();

}

\_state = Menu::State::WAITING;

\_selectedWithCamera = nullptr;

this->release();

}

void Menu::onTouchCancelled(Touch\* touch, Event\* event)

{

CCASSERT(\_state == Menu::State::TRACKING\_TOUCH, "[Menu ccTouchCancelled] -- invalid state");

this->retain();

if (\_selectedItem)

{

\_selectedItem->unselected();

}

\_state = Menu::State::WAITING;

this->release();

}

void Menu::onTouchMoved(Touch\* touch, Event\* event)

{

CCASSERT(\_state == Menu::State::TRACKING\_TOUCH, "[Menu ccTouchMoved] -- invalid state");

MenuItem \*currentItem = this->getItemForTouch(touch, \_selectedWithCamera);

if (currentItem != \_selectedItem)

{

if (\_selectedItem)

{

\_selectedItem->unselected();

}

\_selectedItem = currentItem;

if (\_selectedItem)

{

\_selectedItem->selected();

}

}

}

// 获取当前点击的菜单项

**MenuItem\* Menu::getItemForTouch(Touch \*touch, const Camera \*camera)**

{

Vec2 touchLocation = touch->getLocation();

if (!\_children.empty())

{

for (auto iter = \_children.crbegin(); iter != \_children.crend(); ++iter)

{

MenuItem\* child = dynamic\_cast<MenuItem\*>(\*iter);

if (nullptr == child || false == child->isVisible() || false == child->isEnabled())

{

continue;

}

Rect rect;

rect.size = child->getContentSize();

if (isScreenPointInRect(touchLocation, camera, child->getWorldToNodeTransform(), rect, nullptr))

{

return child;

}

}

}

return nullptr;

}

**MenuItem**

class CC\_DLL MenuItem : public Node

class CC\_DLL MenuItemLabel : public MenuItem

class CC\_DLL MenuItemAtlasFont : public MenuItemLabel

class CC\_DLL MenuItemFont : public MenuItemLabel

class CC\_DLL MenuItemSprite : public MenuItem

class CC\_DLL MenuItemImage : public MenuItemSprite

class CC\_DLL MenuItemToggle : public MenuItem

看看MenuItemLabel

void MenuItemLabel::activate()

{

if(\_enabled)

{

this->stopAllActions();

this->setScale( \_originalScale );

MenuItem::activate();

}

}

void MenuItemLabel::selected()

{

// subclass to change the default action

if(\_enabled)

{

MenuItem::selected();

Action \*action = getActionByTag(kZoomActionTag);

if (action)

{

this->stopAction(action);

}

else

{

\_originalScale = this->getScale();

}

Action \*zoomAction = ScaleTo::create(0.1f, \_originalScale \* 1.2f);

zoomAction->setTag(kZoomActionTag);

this->runAction(zoomAction);

}

}

void MenuItemLabel::unselected()

{

// subclass to change the default action

if(\_enabled)

{

MenuItem::unselected();

this->stopActionByTag(kZoomActionTag);

Action \*zoomAction = ScaleTo::create(0.1f, \_originalScale);

zoomAction->setTag(kZoomActionTag);

this->runAction(zoomAction);

}

}