场景与层

通常需要子类化层而不是场景。HelloWorld类是模板产生的层。

1. 场景切换
2. 场景切换相关函数

场景切换是通过导演类Director实现的。

1. void runWithScene(Scene \*scene);//运行场景。只能在启动第一个场景时调用该函数。若已经有一个场景运行，则不能调用该函数
2. void replaceScene(Scene \*scene);//切换到下一个场景。用一个新的场景替换当前场景，当前场景被终端释放。
3. void pushScene(Scene \*scene);//切换到下一个场景。将当前场景挂起放入到堆栈中，然后切换到下一个场景
4. void popScene();//与(3)的pushScene配合使用，回到上一个场景。
5. void popToRootScene();//与pushScene配合使用，可以回到根场景
6. 使用菜单实现场景切换
7. 建立一个新的场景的.cpp和.h这两个文件将实现一个新的场景，如Setting
8. 在这两个文件中实现一个新的类，可以额外添加回调等等。
9. 在HelloWorld等第一个场景的某一个菜单（如Start）的回调函数中实现以下代码

auto sc=Setting::createScene();

Director::getInstance()->pushScene(sc);

在此之后若点击这一个菜单则从HelloWorld这一个场景切换到Setting场景，同时可在Setting中进行操作返回HelloWorld，如Director::getInstance()->popScene();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

新建的场景需要实现的代码

首先应继承与Scene class Setting:public Scene

1. virtual bool init();//init函数的具体内容与HelloWorld大致相同
2. static Scene\* createScene{return Trying::create()};

//以上两个函数是必须要有的，类似于精灵类的实现

1. 如有其它诸如回调函数这样的函数，也需要给出定义，实现自定义的代码
2. 在最后要使用宏CREATE\_FUNC（类名），确保自动删除；

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

3.场景过渡动画

场景过渡动画是有TransitionScene类和它的子类展示的

基本实现代码

auto sc=Setting::createScene();//创建一个Setting场景

auto reScene=TransitionJumpZoom::create(1.0f,sc);//具体的创建一个过场动画

Director::getInstance()->pushScene(reScene);//调用这样一种过场动画

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

所有转场特效

// TransitionJumpZoom

// 作用： 创建一个跳动的过渡动画

// 参数1：过渡动作的时间

// 参数2：切换到目标场景的对象

// reScene = CCTransitionJumpZoom ::create(t , s);

// CCTransitionProgressRadialCCW

// 作用： 创建一个扇形条形式的过渡动画， 逆时针方向

// 参数1：过渡动作的时间

// 参数2：切换到目标场景的对象

// reScene = CCTransitionProgressRadialCCW::create(t, s);

// CCTransitionProgressRadialCW

// 作用： 创建一个扇形条形式的过渡动画， 顺时针方向

// 参数1：过渡动作的时间

// 参数2：切换到目标场景的对象

// reScene = CCTransitionProgressRadialCW::create(t,s);

// CCTransitionProgressHorizontal

// 作用： 创建一个水平条形式的过渡动画，

// 参数1：过渡动作的时间

// 参数2：切换到目标场景的对象

// reScene = CCTransitionProgressHorizontal ::create(t,s);

// CCTransitionProgressVertical

// 作用： 创建一个垂直条形式的过渡动画，

// 参数1：过渡动作的时间

// 参数2：切换到目标场景的对象

// reScene = CCTransitionProgressVertical::create(t, s);

// CCTransitionProgressInOut

// 作用： 创建一个由里向外扩展的过渡动画，

// 参数1：过渡动作的时间

// 参数2：切换到目标场景的对象

// reScene = CCTransitionProgressInOut::create(t, s);

// CCTransitionProgressOutIn

// 作用： 创建一个由外向里扩展的过渡动画，

// 参数1：过渡动作的时间

// 参数2：切换到目标场景的对象

// reScene = CCTransitionProgressOutIn::create(t, s);

// CCTransitionCrossFade

// 作用：创建一个逐渐透明的过渡动画

// 参数1：过渡动作的时间

// 参数2：切换到目标场景的对象

// reScene = CCTransitionCrossFade::create(t, s);

// CCTransitionPageTurn

// 作用：创建一个翻页的过渡动画

// 参数1：过渡动作持续的时间

// 参数2：切换的目标场景的对象

// 参数3：是否逆向翻页

// reScene = CCTransitionPageTurn::create(t, s, false);

// CCTransitionFadeTR

// 作用：创建一个部落格过渡动画， 从左下到右上

// 参数1：过渡动作持续的时间

// 参数2：切换的目标场景的对象

// reScene =CCTransitionFadeTR::create(t, s);

// CCTransitionFadeBL

// 作用：创建一个部落格过渡动画， 从右上到左下

// 参数1：过渡动作持续的时间

// 参数2：切换的目标场景的对象

// reScene = CCTransitionFadeBL::create(t, s);

// CCTransitionFadeUp

// 作用：创建一个从下到上，条形折叠的过渡动画

// 参数1：过渡动作持续的时间

// 参数2：切换的目标场景的对象

// reScene= CCTransitionFadeUp::create(t, s);

// CCTransitionFadeDown

// 作用：创建一个从上到下，条形折叠的过渡动画

// 参数1：过渡动作持续的时间

// 参数2：切换的目标场景的对象

// reScene = CCTransitionFadeDown::create(t, s);

// CCTransitionTurnOffTiles

// 作用：创建一个随机方格消失的过渡动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// reScene= CCTransitionTurnOffTiles::create(t, s);

// CCTransitionSplitRows

// 作用：创建一个分行划分切换的过渡动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// reScene = CCTransitionSplitRows::create(t, s);

// CCTransitionSplitCols

// 作用：创建一个分列划分切换的过渡动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// reScene = CCTransitionSplitCols::create(t, s);

// CCTransitionFade

// 作用：创建一个逐渐过渡到目标颜色的切换动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// 参数3：目标颜色

// reScene= CCTransitionFade::create(t, s, ccc3(255, 0, 0));

// CCTransitionFlipX

// 作用：创建一个x轴反转的切换动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// 参数3：反转类型的枚举变量 左右上下

// reScene = CCTransitionFlipX::create(t, s, kOrientationRightOver);

// CCTransitionFlipY

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// 参数3：反转类型的枚举变量 左右上下

// reScene = CCTransitionFlipY::create(t, s

// , kOrientationDownOver);

// CCTransitionFlipAngular

// 作用：创建一个带有反转角切换动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// 参数3：反转类型的枚举变量 左右上下

// reScene = CCTransitionFlipAngular::create(t, s, kOrientationLeftOver);

// CCTransitionZoomFlipX

// 作用：创建一个带有缩放的x轴反转切换的动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// 参数3：反转类型的枚举变量 左右上下

// reScene=CCTransitionZoomFlipX::create(t, s, kOrientationLeftOver);

// CCTransitionZoomFlipY

// 作用：创建一个带有缩放的Y轴反转切换的动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// 参数3：反转类型的枚举变量 左右上下

// reScene=CCTransitionZoomFlipY::create(t, s, kOrientationDownOver);

// CCTransitionZoomFlipAngular

// 作用：创建一个带有缩放 ，反转角切换的动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// 参数3：反转类型的枚举变量 左右上下

// reScene=CCTransitionZoomFlipAngular::create(t, s, kOrientationRightOver);

// CCTransitionShrinkGrow

// 创建一个放缩交替的过渡动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// reScene = CCTransitionShrinkGrow::create(t, s);

// CCTransitionRotoZoom

// 创建一个旋转放缩交替的过渡动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// reScene = CCTransitionRotoZoom::create(t, s);

// CCTransitionMoveInL

// 作用：创建一个从左边推入覆盖的过渡动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// reScene = CCTransitionMoveInL::create(t, s);

// CCTransitionMoveInR

// 作用：创建一个从右边推入覆盖的过渡动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// reScene = CCTransitionMoveInR::create(t, s);

// CCTransitionMoveInB

// 作用：创建一个从下边推入覆盖的过渡动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// reScene = CCTransitionMoveInB::create(t, s);

// CCTransitionMoveInT

// 作用：创建一个从上边推入覆盖的过渡动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// reScene = CCTransitionMoveInT::create(t, s);

// CCTransitionSlideInL

// 作用：创建一个从左侧推入并顶出旧场景的过渡动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// reScene =CCTransitionSlideInL::create(t, s);

// CCTransitionSlideInR

// 作用：创建一个从右侧推入并顶出旧场景的过渡动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// reScene =CCTransitionSlideInR::create(t, s);

// CCTransitionSlideInT

// 作用：创建一个从顶部推入并顶出旧场景的过渡动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

// reScene =CCTransitionSlideInT::create(t, s);

// CCTransitionSlideInB

// 作用：创建一个从下部推入并顶出旧场景的过渡动画

// 参数1：过渡动作的持续时间

// 参数2：切换的目标场景的对象

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. 场景的生命周期
2. 生命周期函数
3. bool init();//初始化层调用
4. void onEnter();//进入层时调用
5. void onEnterTransitionDidFinish();//进入层而且过渡动画结束时调用
6. void onExit();//退出层时调用
7. void onExitTransitionDidStart();//退出层而且开始过渡动画时调用
8. void cleanup();//层对象被清除时调用

注：事实上这些函数均继承与Node,所以场景、精灵、层都有这些函数，并且都可以重写这些函数

从一个场景切换（A）到另一个场景（B）的函数调用顺序：

A::onExitTransitionDidStart()->B::onEnter()->A::onExit()->B::onEnterTransitionDidFinish()(->A::cleanup(),只有使用replace转场时才会调用)

从一个场景（B）回到初始的场景（A）

B::onExitTransitionDidStart()->B::onExiti()->B::cleanup()->A::onEnter()->A::onEnter()->A::onEnterTransitionDidFinish()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

特殊声明：

在派生类重写声明周期函数时，第一行应该调用父类的此函数

如：class Setting:public Layer;

Void Steeing::onEnter()

{

Layer::onExit();//一定要调用父类的函数

log(“Setting”);//添加自己想要实现的函数

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/