时钟逻辑实现

头文件

```cpp
#pragma once

#include "cocos2d.h"
#include "SceneManager.h"
USING_NS_CC;
class ClockLayer :public Layer
{
public:
    CREATE_FUNC(ClockLayer);
    virtual bool init();

    //给他一个返回按钮 回调函数 作用：返回到开始场景
    void menuCallBack(Ref*pSender);
    //对一个时钟而言，它需要每秒都去判断，秒针去转动，所以需要使用计时器
    void timeUpdate(float dt);//这个计时器就不是之前的一次性的，这个需要每帧执行

    SceneManager*tsm;

    //时钟时针
    Sprite * _hour;
    //时钟分针
    Sprite * _minute;
    //时钟秒针
    Sprite * _second;
    //时钟背景
    Sprite * _backgroud;
};

```

实现cpp

```cpp
#include "ClockLayer.h"

bool ClockLayer::init()
{
    Size size = Director::getInstance()->getWinSize();

    //返回Label
    MenuItemLabel *menuItem = MenuItemLabel::create(Label::createWithTTF("Back", "fonts/Marker Felt.ttf", 32),
        CC_CALLBACK_1(ClockLayer::menuCallBack, this));
    menuItem->setPosition(Vec2(size.width*0.9,size.height*0.9));
    menuItem->setColor(Color3B(100, 255, 100));
    auto menu = Menu::create(menuItem, nullptr);
    menu->setPosition(Vec2::ZERO);
    this->addChild(menu);

    //时针
    _hour = Sprite::create("shi.png");
    _hour->setPosition(Vec2(size.width / 2, size.height / 2));
    _hour->setScale(0.5);
    this->addChild(_hour,1);
    //分针
    _minute = Sprite::create("fen.png");
    _minute->setPosition(Vec2(size.width / 2, size.height / 2));
    _minute->setScale(0.5);
    this->addChild(_minute,2);
    //秒针
    _second = Sprite::create("miao.png");
    _second->setPosition(Vec2(size.width / 2, size.height / 2));
    _second->setScale(0.5);
    this->addChild(_second,3);

    //背景Sprite
    _backgroud = Sprite::create("background.jpg");
    _backgroud->setPosition(Vec2(size.width / 2, size.height / 2));
    _backgroud->setScale(0.5);
    this->addChild(_backgroud);

    return true;
}

void ClockLayer::menuCallBack(Ref * pSender)
{
    //返回开始场景
    tsm->goOpenScene();
}

void ClockLayer::timeUpdate(float dt)
{
}
```

添加了时针、分针、秒针、背景sprite以及返回的Label

效果展示：