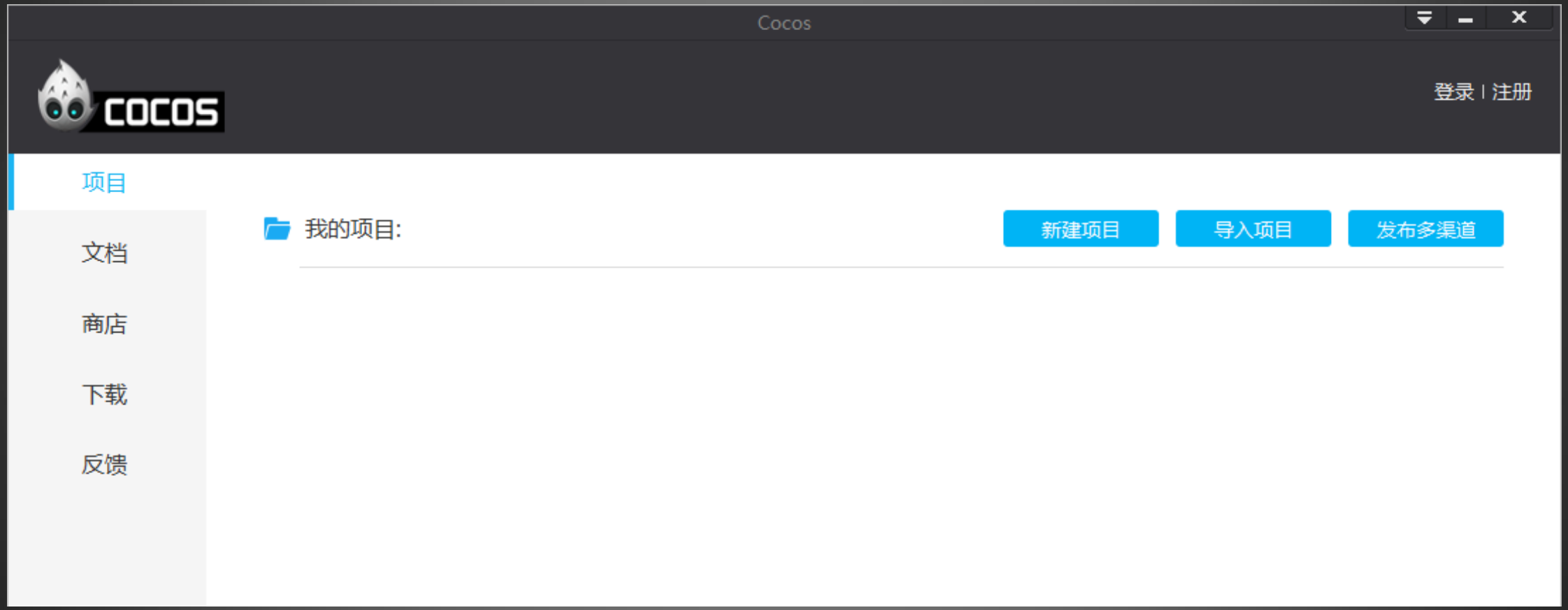
Cocos2d-x基础概念

陈漪凡

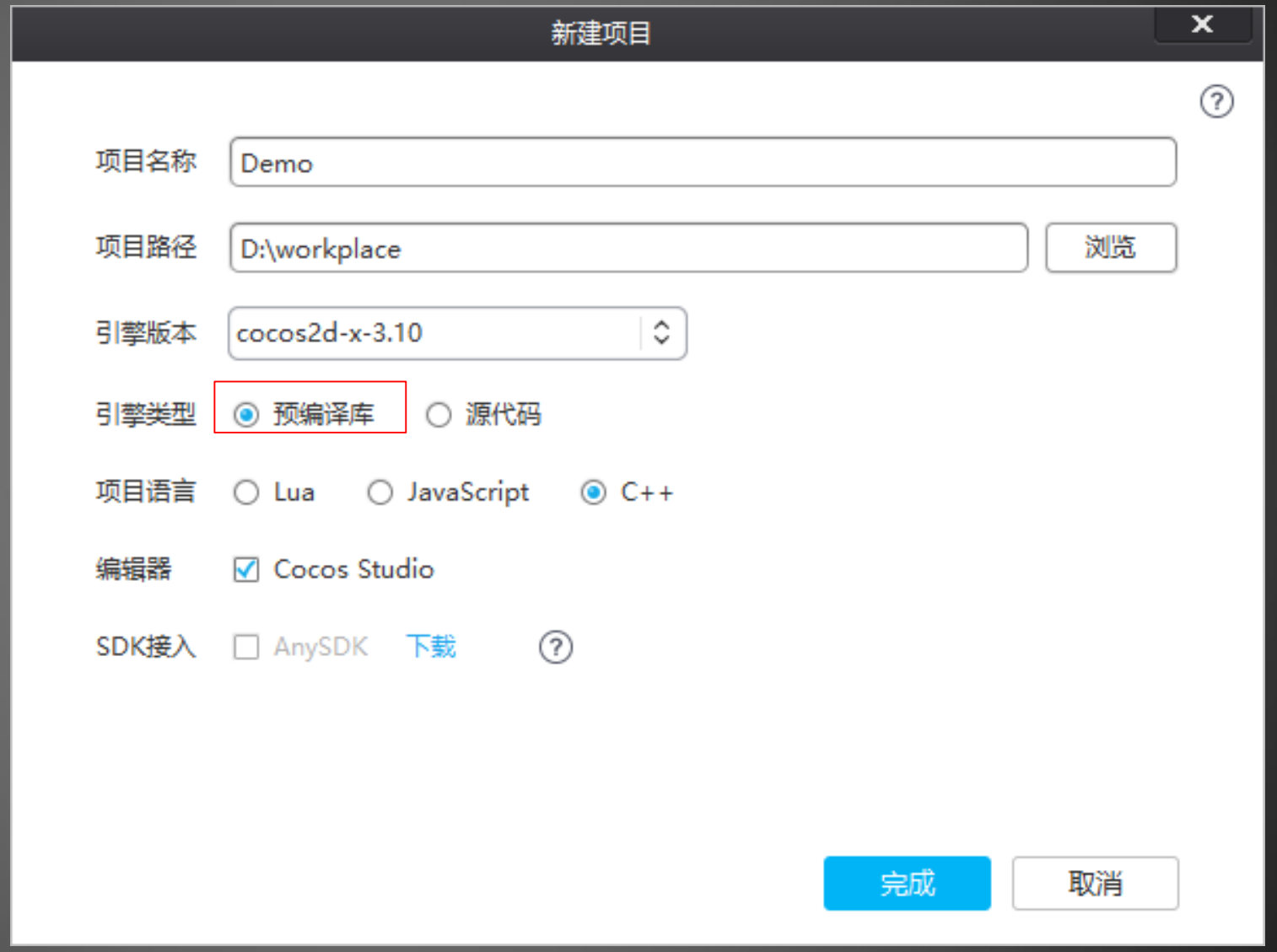
目录 / contents

Cocos引擎



Cocos引擎

新建项目



Cocos引擎

VS代码编辑　Android Studio　Cocos Studio编辑UI和动画

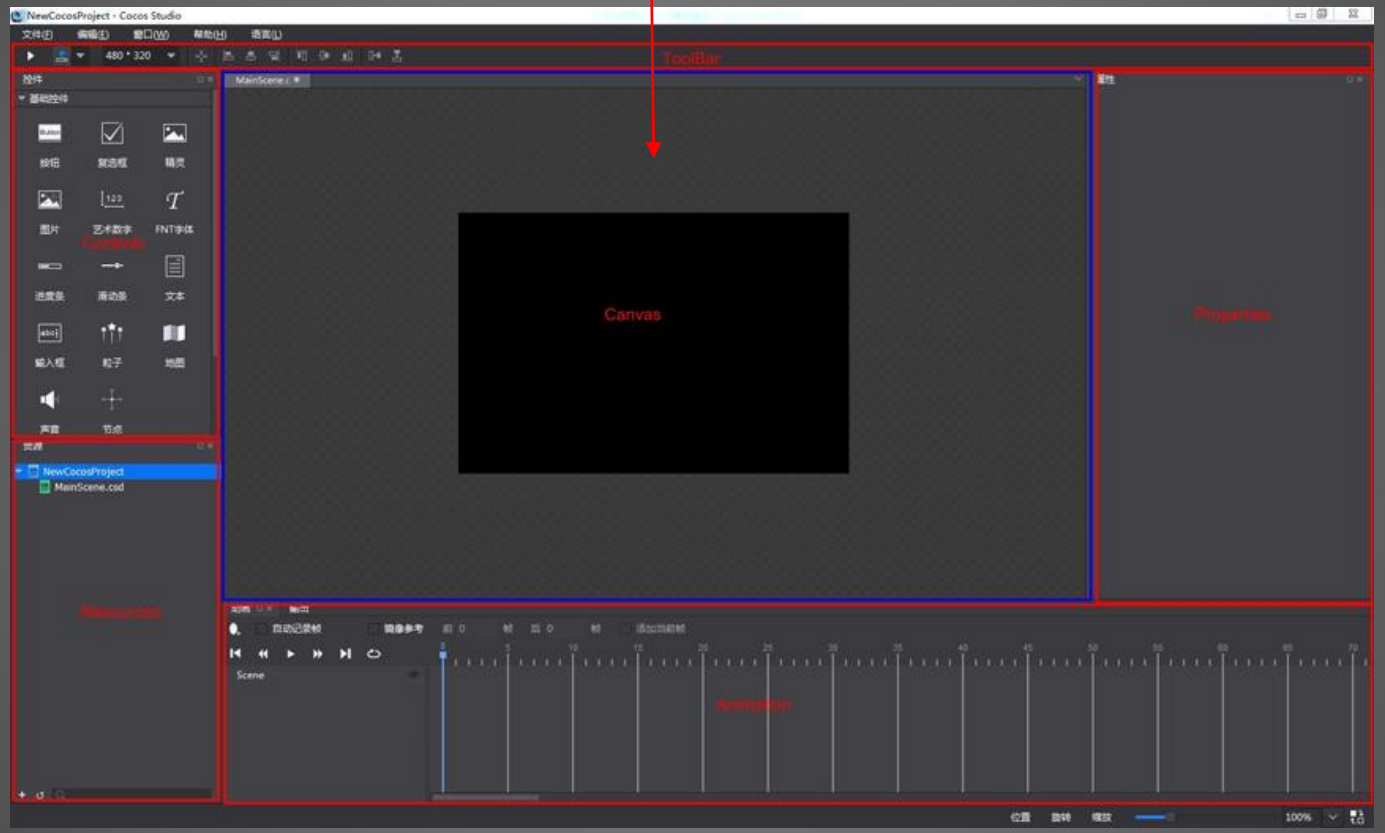
Cocos Studio基本界面

画布面板

工具栏

控制面板

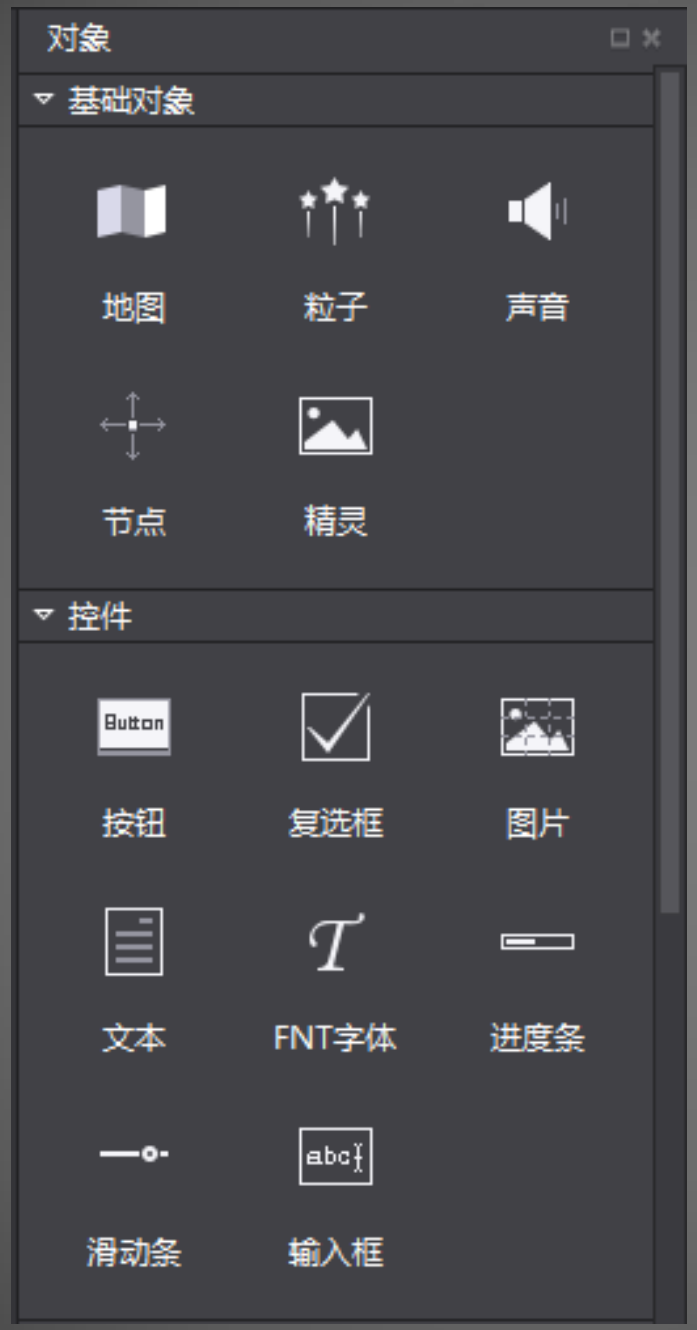
资源面板



属性面板

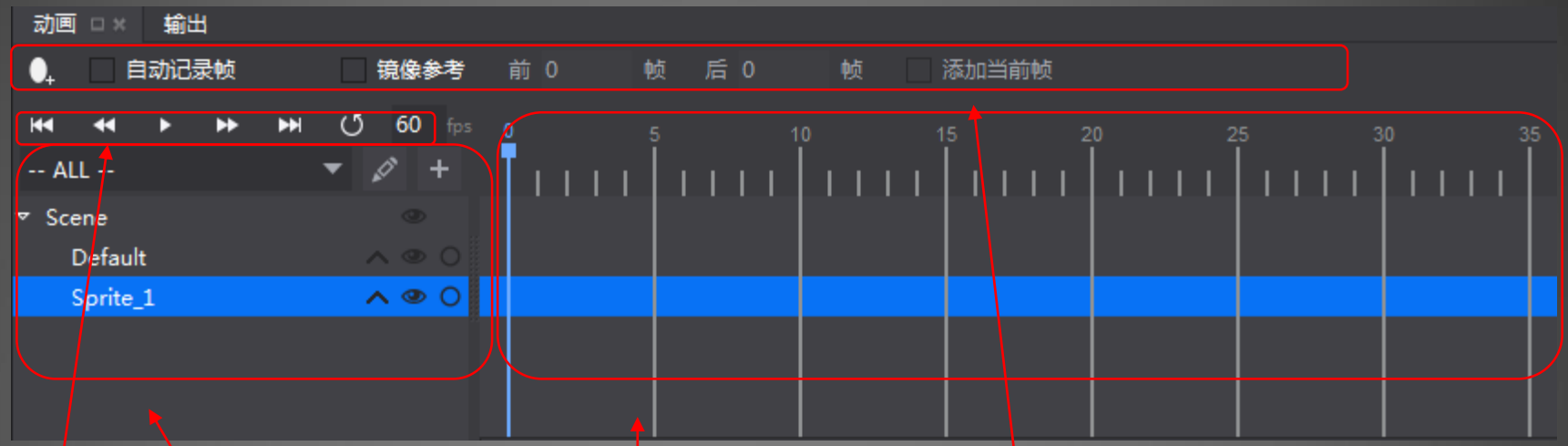
动画面板

控件列表


对象
基础对象
地图
粒子
声音
节点
精灵
控件
Button
按钮
复选框
图片
文本
FNT字体
进度条
滑动条
abc
输入框


容器
基础容器
列表容器
翻页容器
滚动容器
自定义控件
Armature

动画面板



动画控制工具

对象结构树

时间轴

动画编辑工具

游戏案例：捕鱼达人


1000


波克捕鱼-时空秘境-50
595114
V3.0.5
66/100
80/100
99
2
5
13
36
25
25
20
Baidu 百科

点击控制大炮的方向，发射子弹
——坐标转换

点击属于触屏响应事件，监听器返回触摸点的坐标，Touch坐标

Touch坐标是触摸点OpenGL坐标系中的点坐标，也就是在Cocos2d-x坐标系位置

点击控制大炮的方向，发射子弹
——坐标转换

cocos2d中的元素是有父子关系的层级结构，我们通过Node的setPosition设定元素的位置使用的是相对与其父节点的本地坐标，最后在绘制屏幕的时候cocos2d会把这些元素的本地坐标映射成世界坐标系坐标。

点击控制大炮的方向，发射子弹
——坐标转换

1. 触摸点的世界坐标转换为精灵所在层的本地坐标

2. 计算精灵坐标与触摸点坐标的关系，做出相应的反应

其他技术

3d鱼模型 -> 3D的精灵类
鱼身上的波光效果 -> 纹理动画
鱼并不是静止不动的 -> 骨骼动画
判断是否击中鱼 -> 碰撞检测
攻击时的闪电效果 -> LightLineRender类
随机出现和波动 -> Noise类

……

点击控制大炮的方向，发射子弹
——坐标转换

本地坐标与世界坐标的相互转换：

```
CCPoint CCNode::convertToNodeSpace(const CCPoint& worldPoint);
CCPoint CCNode::convertToWorldSpace(const CCPoint& nodePoint);

CCPoint CCNode::convertToNodeSpaceAR(const CCPoint& worldPoint);
CCPoint CCNode::convertToWorldSpaceAR(const CCPoint& nodePoint);
```

作业

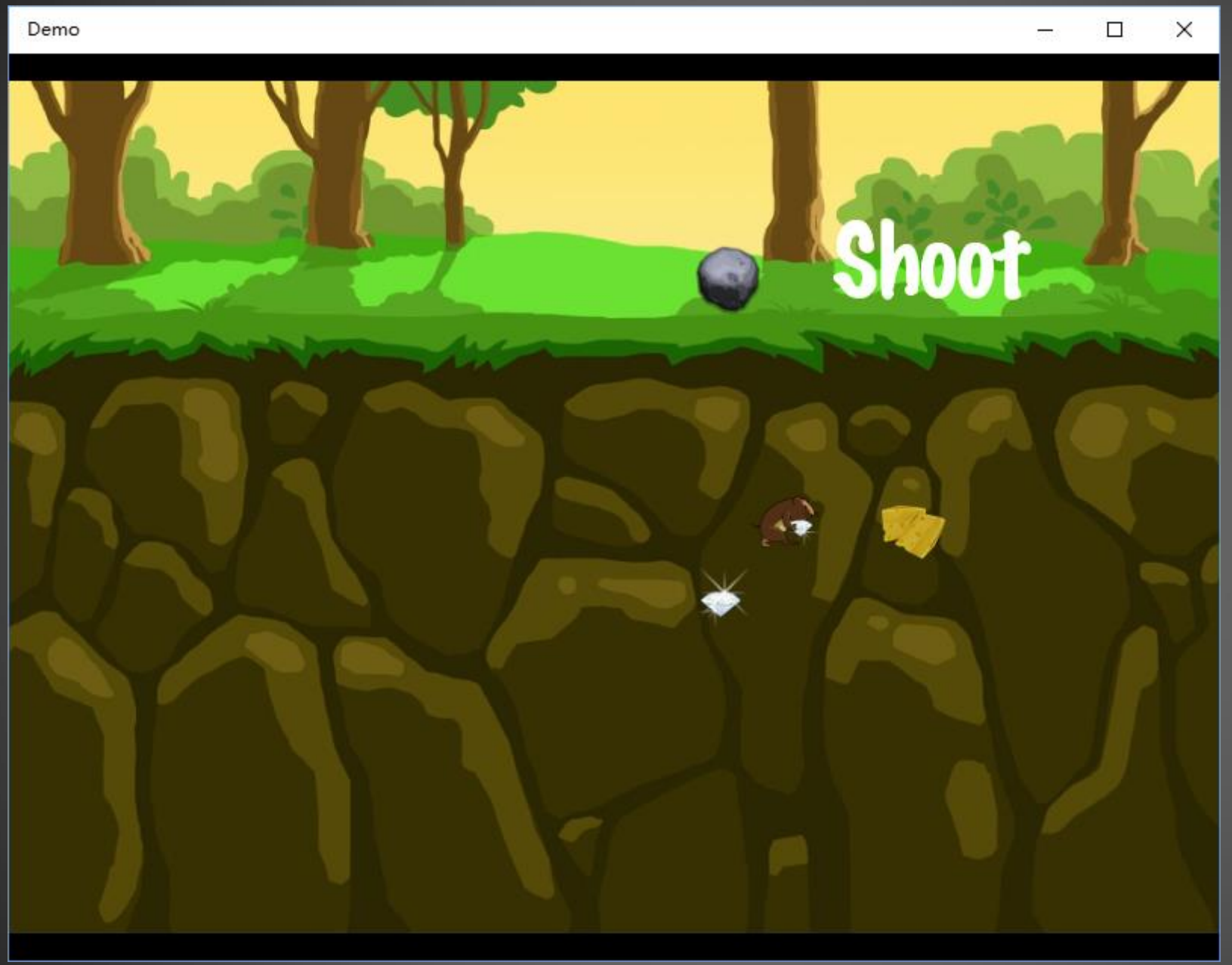
新版黄金矿工游戏，共有两个界面：主界面与游戏界面

主界面：在demo代码基础上完善场景，添加开始按钮（MenuItem），
点击进入游戏界面。
游戏界面：两个Layer，StoneLayer锚点位于左下角，坐标设为(0,0)，
其上有一石头精灵，初始坐标为（560，480）；
MouseLayer锚点位于左下角，坐标设为(0,屏幕高度的一半),
其上有一老鼠精灵，初始坐标为（屏幕宽度的一半，0）。
有一个Label，作为shoot按钮

游戏要求：游戏开始后，点击屏幕任意位置，在该位置添加一块奶酪，老鼠跑到该位置吃掉奶酪；点击shoot按钮，石头发射到老鼠所在的位置，老鼠跑开，留下钻石。
加分项：尝试添加一两个动画

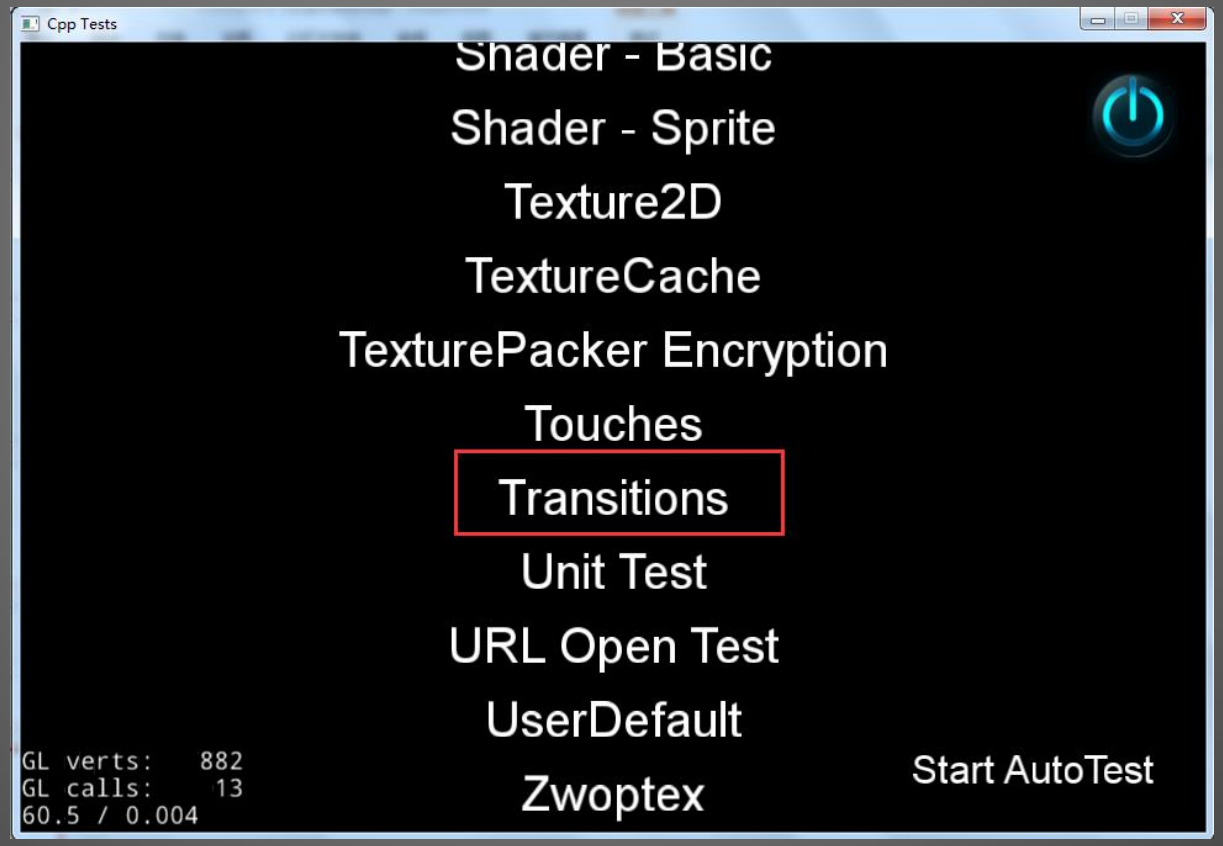
作业提交：提交实验报告（文档），Classes（文件夹），Resources（文件夹）。实验报告要求有截图。


Demo
GOLD MINER
START

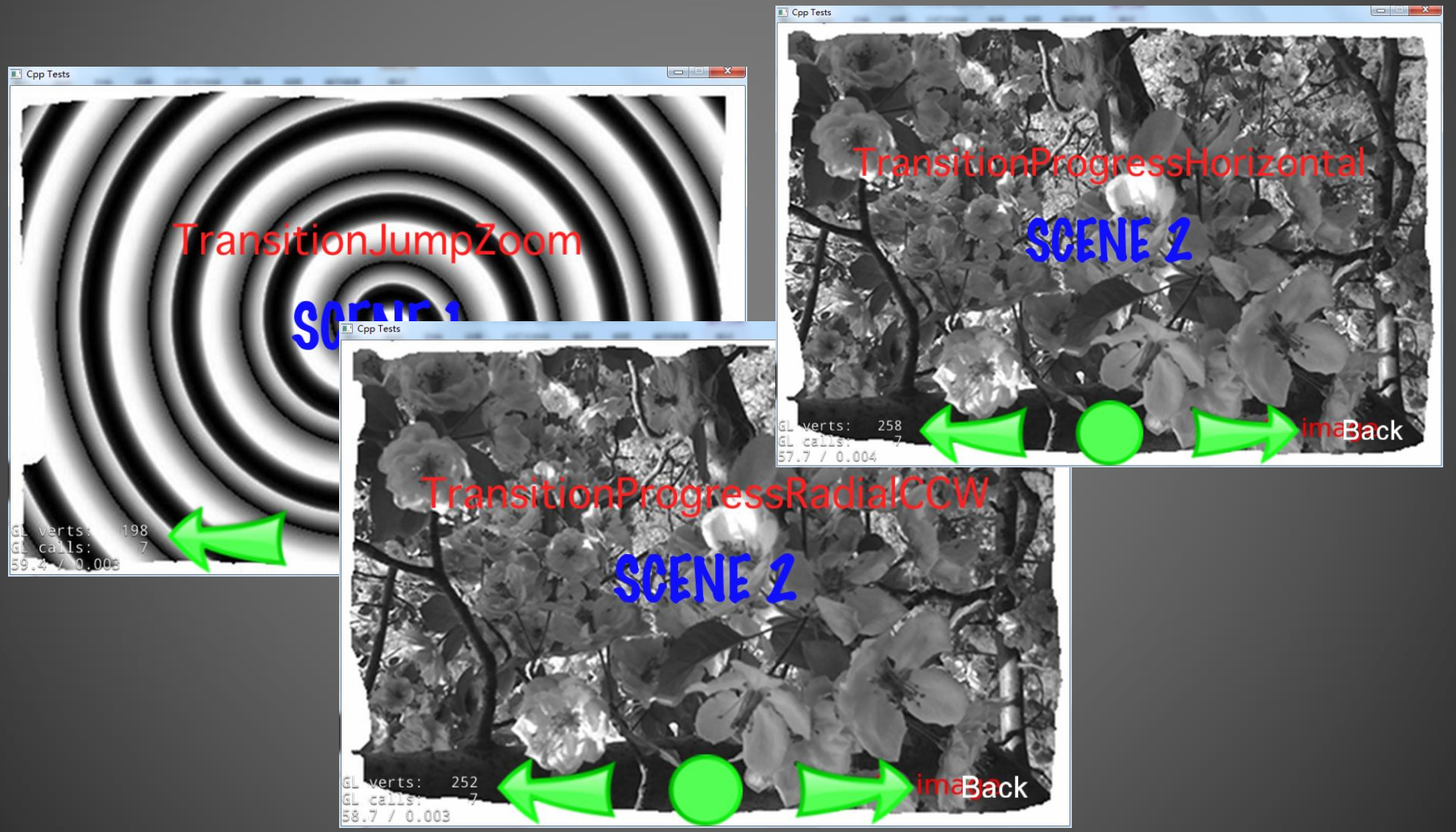

Demo
Shoot

场景切换

- Cocos2d-x test示例——Transitions



场景切换


Cpp Tests
TransitionJumpZoom
GL verts: 198
GL calls: 7
59.4 / 0.003
Cpp Tests
TransitionProgressHorizontal
SCENE 2
GL verts: 258
GL calls: 7
57.7 / 0.004
Back
Cpp Tests
TransitionProgressRadialCCW
SCENE 2
GL verts: 252
GL calls: 7
58.7 / 0.003
Back

触屏响应

添加触摸事件监听器

```
//add touch listener
EventListenerTouchOneByOne* listener = EventListenerTouchOneByOne::create();
listener->setSwallowTouches(true);
listener->onTouchBegan = CC_CALLBACK_2(GameSence::onTouchBegan, this);
Director::getInstance()->getEventDispatcher()->addEventListenerWithSceneGraphPriority(listener, this);
```

有三个函数onTouchBegan,onTouchMoved和onTouchEnded可供重写，本次作业重写onTouchBegan方法即可

```
bool GameSence::onTouchBegan(Touch *touch, Event *unused_event) {

    return true;
}
```

序列帧动画

AppDelegate.cpp中预先加载动画资源

```
// load game resource
SpriteFrameCache::getInstance()->addSpriteFramesWithFile("general-sheet.plist");
char totalFrames = 3;
char frameName[20];
Animation* legAnimation = Animation::create();

for (int i = 0; i < totalFrames; i++)
{
    sprintf(frameName, "miner-leg-%d.png", i);
    legAnimation->addSpriteFrame(SpriteFrameCache::getInstance()->getSpriteFrameByName(frameName));
}
legAnimation->setDelayPerUnit(0.1);
AnimationCache::getInstance()->addAnimation(legAnimation, "legAnimation");
```

使用动画资源资源

```
auto leg = Sprite::createWithSpriteFrameName("miner-leg-0.png");
Animate* legAnimate = Animate::create(AnimationCache::getInstance()->getAnimation("legAnimation"));
leg->runAction(RepeatForever::create(legAnimate));
leg->setPosition(110 + origin.x, origin.y + 102);
this->addChild(leg, 1);
```

THE END

THANKS FOR WATCHING