## 1.常见的标签效果

|  |
| --- |
|  |

## 2、实现打字机显示效果

tips.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <dict>  <key>bgkstory</key>  <string>人神魔共存的时代，本来庇护人类的善神和恶魔联手攻击人类，人类由一位持有轩辕剑的侠士领导了一支抗魔军，最后轩辕剑和善恶神同归于尽。</string>  <key>start</key>  <string>开始</string>  <key>regret</key>  <string>悔棋</string>  <key>actlost</key>  <string>认输</string>  <key>exit</key>  <string>退出</string>  <key>readdoc</key>  <string>是否读取存档</string>  <key>yes</key>  <string>是</string>  <key>no</key>  <string>否</string>  <key>vscom</key>  <string>与电脑对战?</string>  <key>blorwh</key>  <string>选择黑棋还是白棋</string>  <key>blkchess</key>  <string>黑棋</string>  <key>whchess</key>  <string>白棋</string>  <key>uwon</key>  <string>你赢了!</string>  <key>ulost</key>  <string>你输了!</string>  <key>blkwin</key>  <string>黑棋胜利!</string>  <key>whwin</key>  <string>白棋胜利</string>  </dict> |

|  |  |
| --- | --- |
| //ChineseDemo.h  #pragma once  #include "cocos2d.h"  USING\_NS\_CC;  #include<map>  #include<string>  using namespace std;  class ChineseDemo:public Scene  {  public:    static Scene\* createScene();  virtual bool init();  CREATE\_FUNC(ChineseDemo);  private:  //记录中文字集合key:中文编号value:中文字  map<int, Value> prt\_cnWords;  //记录当前文字内容  string prt\_content;  //记录并且UI空间  Label\* prt\_label;  }; | //ChineseDemo.cpp  #include "ChineseDemo.h"  #include<string>  using namespace std;  //#pragma execution\_character\_set("utf-8") //设置vs2017支持中文  Scene \* ChineseDemo::createScene()  {  return ChineseDemo::create();//create()方法其实是会自动调用init方法，也就是说初始化的功能其实是在init中实现的，只不过暴露给外面使用的是create方法  }  bool ChineseDemo::init()  {  //先调用父类的init方法  if (!Scene::init()) {  //如果父类初始化失败就返回false  return false;  }  auto visibleSize = Director::getInstance()->getVisibleSize();  Vec2 origin = Director::getInstance()->getVisibleOrigin();    /\*auto dict = Dictionary::createWithContentsOfFile("zhcn.xml");  auto str = ((String\*)dict->objectForKey("hello"))->getCString();  log(str);\*/  //auto keys = dict->allKeys();    //遍历输出  /\*for (int i = 0; i < keys->count();i++) {  auto obj = keys->getObjectAtIndex(i);  auto k=((String\*)obj)->getCString();  log(((String\*)dict->objectForKey(k))->getCString());  }\*/    //我们只需要有自己项目需要的就行  auto dict = Dictionary::createWithContentsOfFile("tips.xml");  prt\_content = ((String\*)dict->objectForKey("bgkstory"))->getCString();  CCLOG("%s", prt\_content);  /\*auto label = Label::createWithSystemFont(str,"Arial",36);  label->setAnchorPoint(Vec2::ANCHOR\_MIDDLE);  label->setPosition(Vec2(visibleSize.width/2,visibleSize.height /2));  this->addChild(label);\*/  //创建背景图片  auto sprite = Sprite::create("bg.jpg");  sprite->setAnchorPoint(Vec2::ANCHOR\_MIDDLE);  sprite->setPosition(Vec2(visibleSize.width / 2, visibleSize.height / 2));  this->addChild(sprite);  //显示文字类容，    prt\_label = Label::createWithSystemFont("", "Arial", 24);  prt\_label->setAnchorPoint(Vec2::ANCHOR\_MIDDLE);  //设置显示位置  prt\_label->setPosition(Vec2(visibleSize / 2));  //设置颜色  prt\_label->setColor(Color3B(255,0,200));  //设置字体加粗  prt\_label->enableBold();  prt\_label->setDimensions(330,0); //设置在330处换行  this->addChild(prt\_label);  //实现打字机显示效果,需要用到调度器,这个调度器会一直执行，直到你把它销毁  static int i = 0;  this->schedule([&](float dlt) {  char ch = prt\_content[i];  //判断是否是中文  if (ch > -127 && ch < 0)  {  i += 3;//UTF-8编码中文占3个字节  }  else  {  i++;  }  //截取字符串  string str = prt\_content.substr(0, i);  prt\_label->setString(str);  //注意当i的值超过字符串的长度的时候,注销调度器,使用unschedule方法  if (i > prt\_content.length())  {  this->unschedule("schedule\_callback");  }  },0.1f,"schedule\_callback");  return true;  } |

效果：

|  |  |
| --- | --- |
|  |  |
|  |  |