在cocos2d中进度条有loadingBar

|  |
| --- |
|  |

实例1，进度条的简单使用使用LoadingBarDemo类

|  |  |
| --- | --- |
| //LoadingBarDemo.h  #pragma once  #include"cocos2d.h"  #include"ui/CocosGUI.h"  USING\_NS\_CC;  using namespace ui;  #include"spine/spine.h"  #include"spine/spine-cocos2dx.h"  using namespace spine;  class LoadingBarDemo:public Scene  {  public:  static cocos2d::Scene\* createScene();  virtual bool init();  //menu callback  void menuCloseCallback(cocos2d::Ref\* sender);  CREATE\_FUNC(LoadingBarDemo);  private:  }; | //LoadingBarDemo.cpp  #include "LoadingBarDemo.h"  cocos2d::Scene \* LoadingBarDemo::createScene()  {  return LoadingBarDemo::create();  }  // Print useful error message instead of segfaulting when files are not there.  static void problemLoading(const char\* filename)  {  printf("Error while loading: %s\n", filename);  printf("Depending on how you compiled you might have to add 'Resources/' in front of filenames in HelloWorldScene.cpp\n");  }  bool LoadingBarDemo::init()  {  if (!Scene::init())  return false;  auto visibleSize = Director::getInstance()->getVisibleSize();  //创建进度条  LoadingBar\* pbar = LoadingBar::create();  pbar->setAnchorPoint(Vec2::ANCHOR\_MIDDLE);  pbar->setPosition(Vec2(visibleSize / 2));  this->addChild(pbar);  //加载进度条图片  pbar->loadTexture("blood.jpg");  //设置进度条百分比  pbar->setPercent(25);  //设置进度条朝向  pbar->setDirection(LoadingBar::Direction::LEFT);  return true;  }  void LoadingBarDemo::menuCloseCallback(cocos2d::Ref \* sender)  {  } |

实例2.任务血量进度条

|  |
| --- |
|  |

|  |
| --- |
| 需要资源1.    需要资源2 |

实例架构

|  |
| --- |
|  |

主要代码是LoadingBar类

|  |  |
| --- | --- |
| //LoadingBarDemo.h  #pragma once  #include"cocos2d.h"  #include"ui/CocosGUI.h"  USING\_NS\_CC;  using namespace ui;  #include"spine/spine.h"  #include"spine/spine-cocos2dx.h"  using namespace spine;  class LoadingBarDemo:public Scene  {  public:  static cocos2d::Scene\* createScene();  virtual bool init();  //menu callback  void menuCloseCallback(cocos2d::Ref\* sender);  CREATE\_FUNC(LoadingBarDemo);  private:  //骨骼动画节点  SkeletonAnimation\* p\_skeletonNode;  LoadingBar\* p\_bar;  int full;  int reduction;  //创建游戏角色骨骼动画  void createRoleAnimation();  }; | // LoadingBarDemo.cpp  #include "LoadingBarDemo.h"  cocos2d::Scene \* LoadingBarDemo::createScene()  {  return LoadingBarDemo::create();  }  // Print useful error message instead of segfaulting when files are not there.  static void problemLoading(const char\* filename)  {  printf("Error while loading: %s\n", filename);  printf("Depending on how you compiled you might have to add 'Resources/' in front of filenames in HelloWorldScene.cpp\n");  }  bool LoadingBarDemo::init()  {  if (!Scene::init())  return false;  this->createRoleAnimation();  auto visibleSize = Director::getInstance()->getVisibleSize();  auto origin = Director::getInstance()->getVisibleOrigin();  //创建进度条  p\_bar = LoadingBar::create();  p\_bar->setAnchorPoint(Vec2::ANCHOR\_MIDDLE);  p\_bar->setPosition(Vec2(0,430));  //->addChild(pbar);  p\_skeletonNode->addChild(p\_bar);  //加载进度条图片  p\_bar->loadTexture("blood.png");  //设置进度条百分比  full = 100;  float percent = 0.1;  reduction = int(full \*percent);  p\_bar->setPercent(100);  //设置进度条朝向  p\_bar->setDirection(LoadingBar::Direction::LEFT);  //添加一个减血的按钮  auto button = Button::create("attack1.png", "attack2.png", "attack3.png");  button->setAnchorPoint(Vec2::ANCHOR\_BOTTOM\_RIGHT);  button->setPosition(Vec2((origin.x + visibleSize.width - button->getContentSize().width/4), 10));  this->addChild(button);  button->addTouchEventListener([&](Ref\* sender,Widget::TouchEventType type) {  Button\* btn = (Button\*)sender;  //写法1  /\*switch (type)  {  case Widget::TouchEventType::BEGAN:  break;  case Widget::TouchEventType::ENDED:  full -= reduction;  p\_bar->setPercent(full);  if (full == 0) {  CCLOG("no blood left");  btn->setEnabled(false);  }    break;    default:  break;  }\*/  //注意,在lambda表达式中是不能调用外面的局部变量的，如果需要使用，必须将其设置为类的成员变量  //写法2  /\*if (type == Widget::TouchEventType::ENDED) {  full -= reduction;  p\_bar->setPercent(full);  if (full == 0) {  CCLOG("no blood left");  btn->setEnabled(false);  p\_skeletonNode->setAnimation(0, "death", false);  }  }\*/  //改进代码  SkeletonAnimation\* skAnim= p\_skeletonNode;  if (type == Widget::TouchEventType::ENDED) {  if (p\_bar->getPercent() <= 0)  {  return;  }  p\_skeletonNode->setAnimation(0, "idle", false);  full -= reduction;  p\_bar->setPercent(full);  btn->setEnabled(false);  if (p\_bar->getPercent() > 0)  {  btn->runAction(Sequence::create(  DelayTime::create(1.f),  CallFunc::create([btn, skAnim]() {  btn->setEnabled(true);  skAnim->setAnimation(0, "walk", true);  }),NULL));    }  }  });  return true;  }  void LoadingBarDemo::menuCloseCallback(cocos2d::Ref \* sender)  {  }  void LoadingBarDemo::createRoleAnimation()  {  auto visibleSize = Director::getInstance()->getVisibleSize();  //加载骨骼资源  auto atlas = spAtlas\_createFromFile("spineboy/spineboy.atlas", 0);    auto attachmentLoader = (spAttachmentLoader\*)Cocos2dAttachmentLoader\_create(atlas);  //加载骨骼动画数据  spSkeletonJson\* json = spSkeletonJson\_createWithLoader(attachmentLoader);  //缩小比例  json->scale = 0.6f;  auto skeletonData = spSkeletonJson\_readSkeletonDataFile(json, "spineboy/spineboy.json");    spSkeletonJson\_dispose(json);  auto stateData = spAnimationStateData\_create(skeletonData);    //创建骨骼动画节点并且设置状态数据  p\_skeletonNode = SkeletonAnimation::createWithData(skeletonData, false);  p\_skeletonNode->setAnimationStateData(stateData);  //设置坐标  p\_skeletonNode->setPosition(Vec2(visibleSize.width/2,50));  //进行缩放  p\_skeletonNode->setScale(0.5f);  //将骨骼动画添加到场景  this->addChild(p\_skeletonNode);  //播放动画  p\_skeletonNode->setAnimation(0, "walk", true);  } |

效果：

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