八方向摇杆

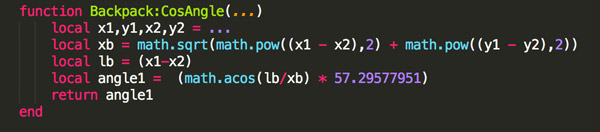
我是借鉴网上C++的版本，然后用lua实现了一下，下面是代码：

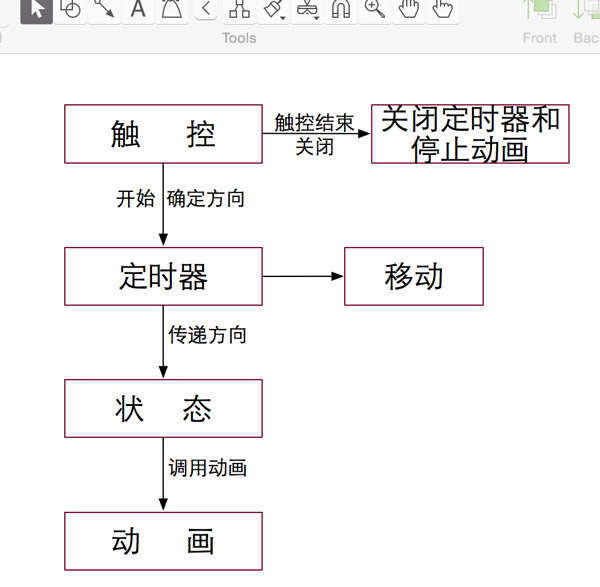
local HRocker = class("HRocker", function()  
return display.newLayer("HRocker")  
end)  
  
function HRocker:ctor()  
  
-- 用于标识摇杆与摇杆的背景  
self.tagForHRocker = { tag\_rocker = 0,   
tag\_rockerBG = 1}  
  
-- 用于标识摇杆方向 8 方向   
self.tagDirecton = {rocker\_stay = 0,  
rocker\_right = 1,  
rocker\_up = 2,  
rocker\_left = 3,  
rocker\_down = 4,  
  
rocker\_left\_up = 5,  
rocker\_left\_down = 6,  
rocker\_right\_up = 7,  
rocker\_right\_down = 8  
}  
-- 判断控制杆方向，用来判断精灵上、下、左、右运动  
self.rocketDirection = nil  
  
-- 当前人物行走方向,用来判断精灵的朝向，精灵脸朝右还是朝左  
self.rocketRun = false  
  
-- 是否可操作摇杆  
self.isCanMove = false  
  
-- 摇杆背景的坐标  
self.rockerBGPosition = nil  
  
-- const PI  
self.PI = 3.1415  
  
end  
  
-- 创建摇杆(摇杆的操作题图片资源名，摇杆背景图片资源名，起始坐标)  
function HRocker:createHRocker( rockerImageName, rockerBGImageName, position )  
local layer = HRocker:new()  
if layer then  
-- 1 按钮， 2 背景  
layer:rockerInit(rockerImageName, rockerBGImageName, position)  
return layer  
end  
return nil  
end  
  
-- privete   
-- 自定义初始化函数 ， 1 按钮，  2 背景图  
function HRocker:rockerInit( rockerImageName, rockerBGImageName, position )  
local spRockerBG = display.newSprite(rockerBGImageName)  
spRockerBG:setPosition( position )  
self:addChild(spRockerBG, 0, self.tagForHRocker.tag\_rockerBG)  
spRockerBG:setVisible(false)  
  
local spRocker = display.newSprite(rockerImageName)  
spRocker:setPosition( position )  
self:addChild(spRocker, 1, self.tagForHRocker.tag\_rocker)  
spRocker:setVisible(false)  
self.rockerBGPosition = position  
self.rockerBGR = spRockerBG:getContentSize().width \* 0.3  
  
-- 表示摇杆方向不变  
self.rocketDirection = self.tagDirecton.rocker\_stay  
end  
  
-- 启动摇杆(显示摇杆、监听摇杆触屏事件)  
function HRocker:startRocker( \_isStopOther )  
  local rocker = self:getChildByTag(self.tagForHRocker.tag\_rocker)  
  rocker:setVisible(true)  
  local rockerBG = self:getChildByTag(self.tagForHRocker.tag\_rockerBG)  
  rockerBG:setVisible(true)  
  
-- 这里开启了点击事件  
self:setTouchEnabled(true)  
  self.touchId = self:addNodeEventListener(cc.NODE\_TOUCH\_EVENT, handler(self, self.addTouchEvent))  
  self:setTouchSwallowEnabled(true)  
end  
  
-- 停止摇杆（隐藏摇杆， 取消摇杆的触屏监听）  
function HRocker:stopRocker()  
local rocker = self:getChildByTag(self.tagForHRocker.tag\_rocker)  
rocker:setVisible(false)  
local rockerBG = self:getChildByTag(self.tagForHRocker.tag\_rockerBG)  
rockerBG:setVisible(false)  
  
-- 这里移除了点击事件  
self:removeNodeEventListener(self.touchId)  
end  
  
-- 获取当前摇杆与用户触屏点的角度  
function HRocker:getRad( pos1, pos2 )  
local px1 = pos1.x  
local py1 = pos1.y  
local px2 = pos2.x  
local py2 = pos2.y  
  
-- 得到两点x的距离  
local x = px2 - px1  
  
-- 得到两点y的距离   
local y = py1 - py2  
  
-- 算出斜边的长度  
local xie = math.sqrt(math.pow(x, 2) + math.pow(y, 2))  
-- 得到这个角度的余弦值（通过三角函数中的定里：角度余弦值 ＝ 斜边/斜边  
local cosAngle = x / xie  
  
-- 通过反余弦定理获取到期角度的弧度  
local rad = math.acos(cosAngle)  
  
-- 注意：当触屏的位置Y坐标<摇杆的Y坐标，我们要去反值-0~-180  
if py2 < py1 then  
rad = -rad  
end  
  
return rad  
end  
  
function HRocker:getAngelePosition( r, angle )  
-- ccp( r \*      cos( angle ), r \*      sin(angle));  
return cc.p( r \* math.cos( angle ), r \* math.sin(angle))  
end  
  
function HRocker:addTouchEvent( event )  
local x, y = event.x, event.y      
local location = cc.p(x, y)  
if event.name == "began" then  
local isBool = self:touchBegan(location)  
  
return isBool  
  
elseif event.name == "moved" then  
self:touchMoved(location)  
elseif event.name == "ended" then  
self:touchEnded(location)  
end  
end  
  
-- 抬起事件  
function HRocker:touchBegan( \_pTouch )  
local point = \_pTouch  
local rocker = self:getChildByTag(self.tagForHRocker.tag\_rocker)  
if cc.rectContainsPoint(rocker:getBoundingBox(), point) then  
self.isCanMove = true  
end  
return true  
end  
  
-- 移动事件  
function HRocker:touchMoved( \_pTouch )  
if not self.isCanMove then  
return  
end  
local point = \_pTouch  
local rocker = self:getChildByTag(self.tagForHRocker.tag\_rocker)  
-- 得到摇杆与触屏点所形成的角度  
local angle = self:getRad(self.rockerBGPosition, point)

-- 两个圆的圆心距  
local kf = math.sqrt(math.pow( self.rockerBGPosition.x - point.x, 2) + math.pow( self.rockerBGPosition.y - point.y, 2))  
  
-- 判断两个圆的圆心距是否大于摇杆背景的半径  
if kf >= self.rockerBGR then  
  
-- 保证内部小圆运动的长度限制  
rocker:setPosition(cc.pAdd(self:getAngelePosition(self.rockerBGR, angle), cc.p(self.rockerBGPosition.x, self.rockerBGPosition.y)))  
  
else  
-- 当没有超过，让摇杆跟随用户触屏点移动即可  
rocker:setPosition(point)  
end  
  
-- 判断八个方向   
local rockerBG = self:getChildByTag(self.tagForHRocker.tag\_rockerBG)  
local p\_dian = {x = rockerBG:getPositionX(), y = rockerBG:getPositionY()}  
  
local move\_x = p\_dian.x - point.x  
local move\_y = p\_dian.y - point.y  
  
-- printf("movex == %f, movey == %f", move\_x, move\_y)  
if move\_x >= 10 and move\_y <= -10 then  
--左上  
self.rocketDirection = self.tagDirecton.rocker\_left\_up  
-- print("左上 左上 左上 左上 左上 左上 左上 左上")  
self.rocketRun = true  
  
elseif move\_x >= 10 and move\_y >= 10 then  
-- 左下  
self.rocketDirection = self.tagDirecton.rocker\_left\_down  
-- print("左下 左下 左下 左下 左下 左下 左下 左下")  
self.rocketRun = true  
  
elseif move\_x <= -10 and move\_y <= -10 then  
-- 右上  
self.rocketDirection = self.tagDirecton.rocker\_right\_up  
-- print("右上 右上 右上 右上 右上 右上 右上 右上")  
self.rocketRun = false  
  
elseif move\_x <= -10 and move\_y >= 10 then  
-- 右下  
self.rocketDirection = self.tagDirecton.rocker\_right\_down  
-- print("右下 右下 右下 右下 右下 右下 右下 右下")  
self.rocketRun = false  
  
elseif move\_x > -10 and move\_x < 10 and move\_y > 0 then  
-- 下  
self.rocketDirection = self.tagDirecton.rocker\_down  
-- print("下 下  下  下  下  下  下  下  下  下  下 ")  
  
elseif move\_x > -10 and move\_x < 10 and move\_y < 0 then  
-- 上  
self.rocketDirection = self.tagDirecton.rocker\_up  
-- print("上 上 上 上  上 上  上 上  上 上  上 上 上")  
  
elseif move\_x > 0 and move\_y > -10 and move\_y < 10 then  
-- 左  
self.rocketDirection = self.tagDirecton.rocker\_left  
-- print("左 左 左 左 左 左 左 左 左 左 左 左 左 左 左")  
self.rocketRun = true  
  
elseif move\_x < 0 and move\_y > -10 and move\_y < 10 then  
-- 右  
self.rocketDirection = self.tagDirecton.rocker\_right  
-- print("右 右 右 右 右 右 右 右 右 右 右 右 右 右 右")  
self.rocketRun = false  
end  
  
-- 判断四个方向  
-- if angle >= -self.PI/4 and angle < self.PI/4 then  
  
--  self.rocketDirection = self.tagDirecton.rocker\_right  
--  self.rocketRun = false  
  
-- elseif angle >= self.PI/4 and angle < 3 \* self.PI/4 then  
--  self.rocketDirection = self.tagDirecton.rocker\_up  
-- elseif (angle >= (3 \* self.PI/4) and angle <= self.PI) or (angle >= -self.PI and angle < (-3 \* self.PI/4) ) then  
--  self.rocketDirection = self.tagDirecton.rocker\_left;  
 --        self.rocketRun = true  
-- elseif angle >= -3 \* self.PI/4 and angle < -self.PI/4 then  
--  self.rocketDirection = self.tagDirecton.rocker\_down;  
-- end  
end  
  
-- 离开事件  
function HRocker:touchEnded( \_pTouch )  
if not self.isCanMove then  
return  
end  
local rockerBG = self:getChildByTag(self.tagForHRocker.tag\_rockerBG)  
local rocker = self:getChildByTag(self.tagForHRocker.tag\_rocker)  
rocker:stopAllActions()  
transition.moveTo(rocker, { x = rockerBG:getPositionX(), y = rockerBG:getPositionY(), time = 0.08 })  
self.isCanMove = false  
self.rocketDirection = self.tagDirecton.rocker\_stay  
end  
return HRocker  
  
下面是摇杆的调用：

-- 添加摇杆 第一张图是按钮， 第二张图是背影

self.rocker = HRocker:createHRocker("Direction\_bt.png", "Direction\_bc.png", cc.p(100, 100))  
   self:addChild(self.rocker)  
   self.rocker:startRocker(true)



****

**1，触控模块代码:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83 | box\_tm:setTouchEnabled(true)  box\_tm:setTouchSwallowEnabled(true)  box\_tm:addTouchEventListener(function(event, x, y, prevX, prevY)       local x,y = x,y             if event == 'began' then               elseif event == "moved" then                cc,yy = box:getParent():convertToNodeSpace(ccp(x,y)) -- 变为世界坐标系              --box\_tm:convertToWorldSpace(ccp(x,y))              box\_tm:setPosition(cc,yy)                local up\_x , up\_y = box\_tm:getPosition()              local loangle =  self:CosAngle(up\_x, up\_y, box\_x ,box\_y)                if  handle2 == nil then                   handle2 =   scheduler.scheduleGlobal(handler(self,self.TouchScheduler),0.1) -- 启动定时器              end                if up\_y > box\_tm\_y then -- 判断在上边                --todo                 if loangle >= 0 and loangle <= 22.5  then                  --todo                    dir\_bar = "右"                   elseif loangle > 22.5 and loangle <= 62.5  then                  --todo                   dir\_bar = "右上"                   elseif loangle >67.5 and loangle <= 112.5 then                   dir\_bar = "上"                   elseif  loangle >= 112.5 and loangle < 157.5 then                   dir\_bar = "左上"                   else                   dir\_bar = "左"                   end              else                      -- 判断在下面                --todo                 if loangle >= 0 and loangle <= 22.5  then                  --todo                     dir\_bar = "右"                 elseif loangle > 22.5 and loangle <= 62.5  then                  --todo                     dir\_bar = "右下"                 elseif loangle >67.5 and loangle <= 112.5 then                     dir\_bar = "下"                 elseif  loangle >= 112.5 and loangle < 157.5 then                     dir\_bar = "左下"                 else                  --todo                     dir\_bar = "左"                 end              end               elseif event == "ended" then                box\_tm:setPosition(box\_tm\_x,box\_tm\_y)                self.critSprite:stopAllActions()                        self.up\_bar = 0                      self.leftUp\_bar =0                      self.left\_bar =0                      self.rightUp\_bar =0                      self.right\_bar =0                      self.down\_bar = 0                      self.leftDown\_bar = 0                      self.rightDown\_bar = 0               scheduler.unscheduleGlobal(handle2)             handle2 = nil             end           return true          end) |

**2，定时器中代码：**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | function Backpack:TouchScheduler()       run\_x,run\_y = self.critSprite:getPosition()       local speed = 3       if dir\_bar == "上" then          if self.up\_bar == 0 then           self:State("上")             self.up\_bar = 1           self.rightUp\_bar = 0           self.right\_bar = 0           self.leftUp\_bar = 0           self.left\_bar = 0           self.down\_bar = 0           self.leftDown\_bar = 0           self.rightDown\_bar = 0          end        self.critSprite:setPosition(run\_x, run\_y + speed)       elseif dir\_bar == "下" then          if self.down\_bar == 0 then           self:State("下")             self.down\_bar = 1           self.rightUp\_bar = 0           self.right\_bar = 0           self.leftUp\_bar = 0           self.left\_bar = 0           self.up\_bar = 0           self.leftDown\_bar = 0           self.rightDown\_bar = 0        end        self.critSprite:setPosition(run\_x, run\_y - speed) |

就这样的不多写了 最好把self:State("下")  中的 汉字改为英文（编码问题要小心）。我这就不改了，大家注意就是。

**3，状态：**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | function Backpack:State(...)        local dir = ...       if dir == "上" then          self:Action(0,8)  -- 这里要注意参数是从第几个编号开始，往后多少张图         elseif  dir == "下" then          self:Action(32,8)         elseif  dir == "左" then          self:Action(48,8)         elseif  dir == "右" then          self:Action(16,8)         elseif  dir == "左上" then          self:Action(56,8)         elseif  dir == "右上" then          self:Action(8,8)         elseif  dir == "左下" then          self:Action(40,8)         elseif  dir == "右下" then          self:Action(24,8)         end    end |

**4，动画：**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | function Backpack:Action(st,la)       self.critSprite:removeFromParentAndCleanup(true)     self.frames = display.newFrames("000%d.tga", st, la)       self.critSprite = display.newSprite(self.frames[1])                          :addTo(display:getRunningScene(), SECOND\_MENU\_Z\_ORDER)     self.critSprite:setScale(0.6)       animation = display.newAnimation(self.frames, 1 / 9)     transition.execute(self.critSprite,CCAnimate:create(animation),{                 onComplete = function()                    end})       self.critSprite:playAnimationForever(animation)  end |