require "AudioEngine"

-- for CCLuaEngine traceback

function \_\_G\_\_TRACKBACK\_\_(msg)

print("----------------------------------------")

print("LUA ERROR: " .. tostring(msg) .. "\n")

print(debug.traceback())

print("----------------------------------------")

end

local function main()

-- avoid memory leak

collectgarbage("setpause", 100)--步长

collectgarbage("setstepmul", 5000)--步长增量

local cclog = function(...)

print(string.format(...))

end

require "hello2"

cclog("result is " .. myadd(3, 5))

---------------

local visibleSize = CCDirector:sharedDirector():getVisibleSize()

local origin = CCDirector:sharedDirector():getVisibleOrigin()

-- add the moving dog

local function creatDog()

local frameWidth = 105

local frameHeight = 95

-- create dog animate

local textureDog = CCTextureCache:sharedTextureCache():addImage("dog.png")

local rect = CCRectMake(0, 0, frameWidth, frameHeight)

local frame0 = CCSpriteFrame:createWithTexture(textureDog, rect)

rect = CCRectMake(frameWidth, 0, frameWidth, frameHeight)

local frame1 = CCSpriteFrame:createWithTexture(textureDog, rect)

local spriteDog = CCSprite:createWithSpriteFrame(frame0)

spriteDog.isPaused = false

spriteDog:setPosition(origin.x, origin.y + visibleSize.height / 4 \* 3)

local animFrames = CCArray:create()

animFrames:addObject(frame0)

animFrames:addObject(frame1)

local animation = CCAnimation:createWithSpriteFrames(animFrames, 0.5)

local animate = CCAnimate:create(animation);

spriteDog:runAction(CCRepeatForever:create(animate))

-- moving dog at every frame

local function tick()

if spriteDog.isPaused then return end

local x, y = spriteDog:getPosition()

if x > origin.x + visibleSize.width then

x = origin.x

else

x = x + 1

end

spriteDog:setPositionX(x)

end

CCDirector:sharedDirector():getScheduler():scheduleScriptFunc(tick, 0, false)

return spriteDog

end

-- create farm

local function createLayerFarm()

local layerFarm = CCLayer:create()

-- add in farm background

local bg = CCSprite:create("farm.jpg")

bg:setPosition(origin.x + visibleSize.width / 2 + 80, origin.y + visibleSize.height / 2)

layerFarm:addChild(bg)

-- add land sprite

for i = 0, 3 do

for j = 0, 1 do

local spriteLand = CCSprite:create("land.png")

spriteLand:setPosition(200 + j \* 180 - i % 2 \* 90, 10 + i \* 95 / 2)

layerFarm:addChild(spriteLand)

end

end

-- add crop

local frameCrop = CCSpriteFrame:create("crop.png", CCRectMake(0, 0, 105, 95))

for i = 0, 3 do

for j = 0, 1 do

local spriteCrop = CCSprite:createWithSpriteFrame(frameCrop);

spriteCrop:setPosition(10 + 200 + j \* 180 - i % 2 \* 90, 30 + 10 + i \* 95 / 2)

layerFarm:addChild(spriteCrop)

end

end

-- add moving dog

local spriteDog = creatDog()

layerFarm:addChild(spriteDog)

-- handing touch events

local touchBeginPoint = nil

local function onTouchBegan(x, y)

cclog("onTouchBegan: %0.2f, %0.2f", x, y)

touchBeginPoint = {x = x, y = y}

spriteDog.isPaused = true

-- CCTOUCHBEGAN event must return true

return true

end

local function onTouchMoved(x, y)

cclog("onTouchMoved: %0.2f, %0.2f", x, y)

if touchBeginPoint then

local cx, cy = layerFarm:getPosition()

layerFarm:setPosition(cx + x - touchBeginPoint.x,

cy + y - touchBeginPoint.y)

touchBeginPoint = {x = x, y = y}

end

end

local function onTouchEnded(x, y)

cclog("onTouchEnded: %0.2f, %0.2f", x, y)

touchBeginPoint = nil

spriteDog.isPaused = false

end

local function onTouch(eventType, x, y)

if eventType == "began" then

return onTouchBegan(x, y)

elseif eventType == "moved" then

return onTouchMoved(x, y)

else

return onTouchEnded(x, y)

end

end

layerFarm:registerScriptTouchHandler(onTouch)

layerFarm:setTouchEnabled(true)

return layerFarm

end

-- create menu

local function createLayerMenu()

local layerMenu = CCLayer:create()

local menuPopup, menuTools, effectID

local function menuCallbackClosePopup()

-- stop test sound effect

AudioEngine.stopEffect(effectID)

menuPopup:setVisible(false)

end

local function menuCallbackOpenPopup()

-- loop test sound effect

local effectPath = CCFileUtils:sharedFileUtils():fullPathForFilename("effect1.wav")

effectID = AudioEngine.playEffect(effectPath)

menuPopup:setVisible(true)

end

-- add a popup menu

local menuPopupItem = CCMenuItemImage:create("menu2.png", "menu2.png")

menuPopupItem:setPosition(0, 0)

menuPopupItem:registerScriptTapHandler(menuCallbackClosePopup)

menuPopup = CCMenu:createWithItem(menuPopupItem)

menuPopup:setPosition(origin.x + visibleSize.width / 2, origin.y + visibleSize.height / 2)

menuPopup:setVisible(false)

layerMenu:addChild(menuPopup)

-- add the left-bottom "tools" menu to invoke menuPopup

local menuToolsItem = CCMenuItemImage:create("menu1.png", "menu1.png")

menuToolsItem:setPosition(0, 0)

menuToolsItem:registerScriptTapHandler(menuCallbackOpenPopup)

menuTools = CCMenu:createWithItem(menuToolsItem)

local itemWidth = menuToolsItem:getContentSize().width

local itemHeight = menuToolsItem:getContentSize().height

menuTools:setPosition(origin.x + itemWidth/2, origin.y + itemHeight/2)

layerMenu:addChild(menuTools)

return layerMenu

end

-- play background music, preload effect

-- uncomment below for the BlackBerry version

-- local bgMusicPath = CCFileUtils:sharedFileUtils():fullPathForFilename("background.ogg")

local bgMusicPath = CCFileUtils:sharedFileUtils():fullPathForFilename("background.mp3")

AudioEngine.playMusic(bgMusicPath, true)

local effectPath = CCFileUtils:sharedFileUtils():fullPathForFilename("effect1.wav")

AudioEngine.preloadEffect(effectPath)

-- run

local sceneGame = CCScene:create()

sceneGame:addChild(createLayerFarm())

sceneGame:addChild(createLayerMenu())

CCDirector:sharedDirector():runWithScene(sceneGame)

end

xpcall(main, \_\_G\_\_TRACKBACK\_\_)