



## Sample: Lua - Game Lua

```
display.setStatusBar( display.HiddenStatusBar )  
local storyboard = require "storyboard"  
storyboard.purgeOnSceneChange = true  
storyboard.gotoScene( "menu", "fade", 400 )
```



```
storyboard = require "storyboard"
storyboard.state = {}
storyboard.state.H_Score = 0
storyboard.state.C_Score = 0
storyboard.state.Is_Rewrite = false
local scene = storyboard.newScene()
local filePath = system.pathForFile( "data.txt" )
Start_Score = 0

--uncomment to control memory usage
--[[
local function garbagePrinting()
    collectgarbage("collect")

    local memUsage_str = string.format( "mem = %.3f KB", collectgarbage( "count" ) )
    local texMemUsage_str = system.getInfo( "textureMemoryUsed" )
    texMemUsage_str = texMemUsage_str/1000
    texMemUsage_str = string.format( "texMem = %.3f MB", texMemUsage_str )
    print( memUsage_str.." "..texMemUsage_str )
end

tmr_mem = timer.performWithDelay(1000, garbagePrinting, -1)
--]]

function start(event)
    if event.phase == "began" then
        collectgarbage("collect")
        storyboard.gotoScene("game", "fade", 200)
    end
end
end
```



```
function tutor(event)
    if event.phase == "began" then
        storyboard.gotoScene("tutorial", "fade", 200)
    end
end

function cred(event)
    if event.phase == "began" then
        storyboard.gotoScene("credit", "fade", 200)
    end
end

function scene:createScene( event )
    local ScreenGroup = self.view

    background = display.newImageRect( "Images/background.jpg", display.contentWidth,
display.contentHeight )

    background.anchorX = 0
    background.anchorY = 0
    background.x, background.y = 0, 0
    ScreenGroup:insert(background)

    play = display.newImage("Images/play.png")
    play:scale(0.3,0.3)
    play.x = 219
    play.y = 165
    play.scene = "game"
    ScreenGroup:insert(play)
```



```
tutorial = display.newImage("Images/tutorial.png")
tutorial:scale(0.3,0.3)
tutorial.x = 327
tutorial.y = 165
tutorial.scene = "try"
ScreenGroup:insert(tutorial)
```

```
credits = display.newImage("Images/credits.png")
credits:scale(0.3,0.3)
credits.x = 105
credits.y = 165
credits.scene = "credit"
ScreenGroup:insert(credits)
```

```
-- definition record of the games from file "data.txt"
local High_Score=0
local file = io.open( filePath, "r" )
if file then
    local contents = file:read( "*"a" )
    io.close( file )
    for line in io.lines(filePath) do
        High_Score = line
    end
else
    file = io.open( filePath, "w" )
    file:write( 0 )
    io.close( file )
    High_Score = 0
end
```



```
Start_Score = High_Score
storyboard.state.H_Score =High_Score
storyboard.state.C_Score =0
end

function scene:enterScene( event )
    --rewrite new record of the games to file "data.txt"
    if storyboard.state.Is_Rewrite then
        local filePath = system.pathForFile( "data.txt" )
        local file = io.open( filePath, "r" )
        file = io.open( filePath, "w" )
        file:write( storyboard.state.H_Score )
        io.close( file )
    end

    play:addEventListener("touch", start)
    tutorial:addEventListener("touch", tutor)
    credits:addEventListener("touch", cred)
    storyboard.removeAll()
end

function scene:exitScene( event )
    play:removeEventListener("touch", start)
    tutorial:removeEventListener("touch", tutor)
    credits:removeEventListener("touch", cred)
end

function scene:destroyScene( event )
```



end

scene:addEventListener( "createScene", scene )

scene:addEventListener( "enterScene", scene )

scene:addEventListener( "exitScene", scene )

scene:addEventListener( "destroyScene", scene )

return scene



```
local _W = display.contentWidth
```

```
local _H = display.contentHeight
```

```
local Is_Error = false
```

```
local physics = require "physics"
```

```
physics.start()
```

```
my_score=0
```

```
local scene = storyboard.newScene()
```

```
local rnd = math.random
```

```
local balls = {"ball_green.png", "ball_yellow.png", "ball_blue.png", "ball_red.png"}
```

```
local jap_chars = {"c_1.png", "c_2.png", "c_3.png", "c_4.png", "c_5.png", "c_6.png", "c_7.png",  
"c_8.png", "c_9.png", "c_10.png",
```

```
                "c_11.png", "c_12.png", "c_13.png", "c_14.png", "c_15.png",  
"c_16.png", "c_17.png", "c_18.png", "c_19.png", "c_20.png",
```

```
                "c_21.png", "c_22.png", "c_23.png", "c_24.png", "c_25.png",  
"c_26.png", "c_27.png", "c_28.png", "c_29.png", "c_30.png",
```

```
                "c_31.png", "c_32.png", "c_33.png", "c_34.png", "c_35.png",  
"c_36.png", "c_37.png", "c_38.png", "c_39.png", "c_40.png",
```

```
                "c_41.png", "c_42.png", "c_43.png", "c_44.png", "c_45.png",  
"c_46.png"}
```

```
local Engl_pron = {"a", "i", "u", "e", "o", "ka", "ki", "ku", "ke", "ko",
```

```
                "sa", "shi", "su", "se", "so", "ta", "chi", "tsu", "te", "to",
```

```
                "na", "ni", "nu", "ne", "no", "ha", "hi", "fu", "he", "ho",
```

```
                "ma", "mi", "mu", "me", "mo", "ya", "yu", "yo", "ra", "ri",
```

```
                "ru", "re", "ro", "wa", "wo", "n"}
```



```
local press_button = function( event )
    timer.cancel(tmr_ball ); tmr_ball = nil;
    if (visi_balls.numChildren ~= 0) then
        for i = 1, visi_balls.numChildren do
            display.remove(visi_balls[i])
            visi_balls[i] = nil
        end
    end
    storyboard.gotoScene("menu", "fade", 200)
    return true
end

local function New_Char(First)
    curr_i = rnd(1,46)
    my_char=Engl_pron[curr_i]
    if not First then
        jap_char:removeSelf()
        jap_char = nil
    end
    jap_char = display.newImage("Images/Jappan_Char/".jap_chars[curr_i])
    jap_char.alpha = 0    -- visibility tip (1 - visible, 0 - invisible )
    jap_char.x = 440
    jap_char.y = 75
    screenGroup:insert(jap_char)
    return my_char
end

visi_balls = display.newGroup()
```





```
local function New_Ball(self, event)
    -- definition of the probability of occurrence of the ball (depending on points)
    k = rnd(1,my_score/5+2)
    if k == 1 then
        n_c = curr_i
    else
        n_c = rnd(1,46)
    end
    n_b = rnd(1,4)
    ball_char = display.newGroup()
    local ball      = display.newImage("Images/"..balls[n_b])
    ball_char:insert( ball, true )
    local j_char = display.newImage("Images/Jappan_Char/"..jap_chars[n_c])
    ball_char:insert( j_char, true )
    ball_char.anchorY=1
    ball_char.x = 460+rnd(-20, 20)
    ball_char.y = rnd (105,130)
    ball_char.Name="ball"
    ball_char.Eng_Char = EngI_pron[n_c]
    ball_char.speed=math.random(2,6)
    physics.addBody( ball_char, "static",{ density=0.1, friction=10, bounce=0.1, radius = 85} )
    ball_char.isFixedRotation = true
    ball_char.Index = visi_balls.numChildren+1
    screenGroup:insert(ball_char)
    visi_balls:insert(ball_char, true)
    screenGroup:insert(visi_balls)
end

function MoveScene(self, event)
```



```
if not Is_Error then
    local dx
    local del_b = -1
    Score.text="Score: "..my_score
    Target_Char.text=my_char
    background.x = background.x-3
    background1.x = background1.x-3
    floor.x = floor.x-3
    floor1.x = floor1.x-3
    if background.x < -240 then
        background.x = 717
        floor.x=717
    end
    if background1.x < -240 then
        background1.x = 717
        floor1.x=717
    end
    if (visi_balls.numChildren ~= 0) then
        for i = 1, visi_balls.numChildren do
            visi_balls[i].x = visi_balls[i].x - 2
            if(visi_balls[i].x < 0) then
                del_b=i
            end
        end
        if (del_b ~= -1) then
            display.remove(visi_balls[del_b])
            visi_balls[del_b] = nil
        end
    end
end
```



```
else
    Sun.IsVisible = false
    if jap_char.alpha >= 0.9 then
        jap_char.alpha = 0
    else
        jap_char.alpha = jap_char.alpha +0.05
    end
end
end

tmr_ball = timer.performWithDelay(1500, New_Ball, -1)

local function onLocalCollision( self, event )
    if ( event.phase == "began" ) then
        if event.other.Name == "ball" then
            if event.other.Eng_Char == my_char then
                event.other:removeSelf()
                my_score=my_score+1
                my_char=New_Char(False)
            else
                samurai:setSequence( "shock" )
                samurai:play()
                Is_Error = "True"
                timer.pause(tmr_ball)
                physics.pause()
                Sun.alpha = 0
                Score.alpha=1
                jap_char.alpha = 1
                storyboard.state.C_Score =my_score
            end
        end
    end
end
```



```
        if my_score > tonumber(storyboard.state.H_Score) then
            H_Score_Im.alpha = 1
            storyboard.state.H_Score =tonumber(my_score)
            storyboard.state.Is_Rewrite = true
        end
        button_back:addEventListener("touch", press_button)
    end
else
    samurai:setSequence( "run" )
    samurai:play()
end
end
end
```

```
function touchScreen(event)
    if event.phase == "began" then
        if samurai.y > 270 then
            samurai:applyForce(0,-0.85, samurai.x, samurai.y)
            samurai:setSequence( "jump" )
            samurai:play()
        end
    end
end
```

```
function scene:createScene(event)
    screenGroup = self.view

    background = display.newImageRect("Images/default.png", _W, _H)
```



```
background.x = 240
background.y = 150
screenGroup:insert(background)

background1 = display.newImageRect("Images/default.png", _W, _H)
background1.x = 720
background1.y = 150
screenGroup:insert(background1)

floor = display.newImageRect( "Images/floor.png", _W, 32 )
floor.Name="land"
floor.x = 240
floor.y = 300
physics.addBody( floor, "static",{density=0, friction=10, bounce=0} )
screenGroup:insert(floor)

floor1 = display.newImageRect( "Images/floor.png", 480, 32 )
floor1.Name="land"
floor1.x = 719
floor1.y = 300
physics.addBody( floor1, "static",{density=0, friction=10, bounce=0} )
screenGroup:insert(floor1)

local options_run = {frames = require("samurai_run").frames,}
local options_jump = {frames = require("samurai_jump").frames,}
local options_shock = {frames = require("samurai_shock").frames,}
local sheet_run = graphics.newImageSheet( "Images/samurai_run.png", options_run )
local sheet_jump = graphics.newImageSheet( "Images/samurai_jump.png", options_jump )
local sheet_shock = graphics.newImageSheet( "Images/samurai_shock.png", options_shock )
```



```
local sequenceData = { { name="run", sheet=sheet_run, start=1, count=8, time=1000,
loopCount=0 },
                        { name="jump", sheet=sheet_jump, frames={ 1,
2, 8, 9, 10, 11, 13, 14, 14, 16, 16, 16, 16, 16, 16, 16, 17 }, time=1800, loopCount=1 },
                        { name="shock", sheet=sheet_shock, start=1,
count=1, time=1000, loopCount=1 }}

samurai = display.newSprite( sheet_run, sequenceData)
samurai.Name="samurai"
physics.addBody( samurai, "dinamic",{ density=1, friction=0.1, bounce=0.1, radius = 1} )
samurai.isFixedRotation = true
samurai.anchorY=0.8
samurai.x = 50
samurai.y = 300
samurai:play()
screenGroup:insert(samurai)

Score = display.newText( "SCORE!", 350, 260, native.systemFont, 40 )
Score:setFillColor( 1, 0, 0 )
Score.alpha = 1
screenGroup:insert(Score)

my_char=New_Char("True")

Target_Char = display.newText( "", 440, 42, native.systemFont, 40 )
Target_Char:setFillColor( 0, 0, 1 )
screenGroup:insert(Target_Char)

button_back = display.newImage("Images/reload.png")
button_back:scale(0.3,0.3)
button_back.x = 70
```



```
button_back.y = 60
button_back.scene = "credit"
screenGroup:insert(button_back)

Sun = display.newImageRect( "Images/Sun.png", 95, 80 )
Sun.Name="sun"
Sun.x = 69
Sun.y = 60
screenGroup:insert(Sun)

H_Score_Im = display.newImage( "Images/H_Score.png")
H_Score_Im:scale(0.5,0.5)
H_Score_Im.x = 250
H_Score_Im.y = 55
H_Score_Im.alpha = 0
screenGroup:insert(H_Score_Im)

end

function scene:enterScene(event)
    Runtime:addEventListener("touch",touchScreen)
    Runtime:addEventListener("enterFrame", MoveScene)
    samurai.collision = onLocalCollision
    samurai:addEventListener( "collision", samurai )
end

function scene:exitScene(event)
```



```
end
```

```
function scene:destroyScene(event)
```

```
    Runtime:removeEventListener("touch",touchScreen)
```

```
    Runtime:removeEventListener("enterFrame", MoveScene)
```

```
    button_back:removeEventListener("touch", press_button)
```

```
    samurai:removeEventListener( "collision", samurai )
```

```
    display.remove(samurai)
```

```
    samurai = nil
```

```
    display.remove(H_Score_Im)
```

```
    H_Score_Im = nil
```

```
    display.remove(screenGroup)
```

```
    screenGroup = nil
```

```
    visi_balls:remove()
```

```
end
```

```
scene:addEventListener("createScene", scene)
```

```
scene:addEventListener("enterScene", scene)
```

```
scene:addEventListener("exitScene", scene)
```

```
scene:addEventListener("destroyScene", scene)
```

```
return scene
```





```
local scene = storyboard.newScene()
```

```
local press_button = function(event)
```

```
    storyboard.gotoScene("menu", "fade", 200)
```

```
    return true
```

```
end
```

```
function scene:createScene(event)
```

```
    local screenGroup = self.view
```

```
    local background = display.newImage( "Images/background.jpg", centerX, centerY )
```

```
    background.x = 300
```

```
    background.y = 150
```

```
    screenGroup:insert(background)
```

```
    local t1 = display.newText( "High score - "..storyboard.state.H_Score, 200, 160, nil, 60 );
```

```
    t1:setFillColor( 1, 1, 136/255 );
```

```
    screenGroup:insert(t1)
```

```
    button_back = display.newImage("Images/previous.png")
```

```
    button_back:scale(0.5,0.5)
```

```
    button_back.x = 430
```

```
    button_back.y = 260
```

```
    button_back.scene = "credit"
```

```
    screenGroup:insert(button_back)
```

```
end
```

```
function scene:enterScene(event)
```

```
    button_back:addEventListener("touch", press_button)
```

```
end
```

```
function scene:exitScene(event)
```



```
end
```

```
function scene:destroyScene(event)
```

```
    button_back:removeEventListener("touch", press_button)
```

```
    display.remove(screenGroup)
```

```
    screenGroup = nil
```

```
end
```

```
scene:addEventListener("createScene", scene)
```

```
scene:addEventListener("enterScene", scene)
```

```
scene:addEventListener("exitScene", scene)
```

```
scene:addEventListener("destroyScene", scene)
```

```
return scene
```



```
application = {
    content = {
        width = 320,
        height = 480,
        scale = "zoomEven",
        fps = 30,

        --[[
imageSuffix = {
            ["@2x"] = 2,
        }
        --]]
    },

    --[[
-- Push notifications

notification =
{
    iphone =
    {
        types =
        {
            "badge", "sound", "alert", "newsstand"
        }
    }
}
--]]
}
```



```
local sheetData = {}
```

```
sheetData.frames = {
```

```
  {
```

```
    x = 5,
```

```
    y = 0,
```

```
    width = 59,
```

```
    height = 94,
```

```
  },
```

```
  {
```

```
    x = 64,
```

```
    y = 0,
```

```
    width = 66,
```

```
    height = 94,
```

```
  },
```

```
  {
```

```
    x = 130,
```

```
    y = 0,
```

```
    width = 68,
```

```
    height = 94,
```

```
  },
```

```
  {
```

```
    x = 198,
```

```
    y = 0,
```

```
    width = 68,
```



```
        height = 94,  
    },  
    {  
        x = 266,  
        y = 0,  
        width = 70,  
        height = 94,  
    },  
    {  
        x = 336,  
        y = 0,  
        width = 68,  
        height = 94,  
    },  
    {  
        x = 404,  
        y = 0,  
        width = 68,  
        height = 94,  
    },  
    {  
        x = 472,  
        y = 0,  
        width = 60,  
        height = 94,  
    },  
    {  
        x = 532,  
        y = 0,  
    }  
    ...
```