LXZHelper.lua 接口说明

[UpdateWindow](#UpdateWindow)

[AddWndUpdateFunc](#AddWndUpdateFunc)

[HelperCoroutine](#HelperCoroutine)

[HelperGetRoot](#HelperGetRoot)

[HelperSetWindowText](#HelperSetWindowText)

[HelperGetWindowText](#HelperGetWindowText)

[HelperSetWindowSymText](#HelperSetWindowSymText)

[HelperGetWindowPictureFile](#HelperGetWindowPictureFile)

[HelperSetWindowPictureFile](#HelperSetWindowPictureFile)

[HelperGetWindowPictureColor](#HelperGetWindowPictureColor)

[HelperSetWindowPictureColor](#HelperSetWindowPictureColor)

[HelperSetRectangleColor](#HelperSetRectangleColor)

[HelperGetRectangleColor](#HelperGetRectangleColor)

[HelperGetWindowMask](#HelperGetWindowMask)

[HelperSetWindowMask](#HelperSetWindowMask)

[HelperDBIntialize](#HelperDBIntialize)

[HelperDBCreateTable](#HelperDBCreateTable)

[HelperDBExecute](#HelperDBExecute)

[HelperDBHaveTable](#HelperDBHaveTable)

[HelperDBSelect](#HelperDBSelect)

[HelperDBUpdate](#HelperDBUpdate)

[HelperDBAlterTable](#HelperDBAlterTable)

[EffectEase](#EffectEase)

[EffectContainer](#EffectContainer)

[EffectTimer](#EffectTimer)

[HelperEncodeURI](#HelperEncodeURI)

[HelperDecodeURI](#HelperDecodeURI)

[HelperGetDayCount](#HelperGetDayCount)

[deepcopy](#deepcopy)

[HelperPostJSON](#HelperPostJSON)

**UpdateWindow()**

**--只要每帧都调用，因此常常放在OnUpdate事件处理代码调用。**

比如:

local function OnUpdate(window, msg, sender)

UpdateWindow();

end

--以及事件绑定

event\_callback ["OnUpdate"] = OnUpdate;

设置绑定LuaLogic组件的窗口属性Status:IsActive为true即可触发OnUpdate事件.

**AddWndUpdateFunc(w, EffectFunc, param,thread,id);**

**--给窗口添加效果接口，w窗口对象，EffectFunc效果接口, param效果参数,thread 协程对象,id给效果编号**

示例:

--效果参数

local param={type=tween.CIRC,fn=tween.easeOut,begin=0,offset=-0.5,change=0.5, duration=500,reset=true,attribute="CLXZWindow:Scale:fScaleX"};

--将动画加入w窗口, 效果接口EffectErase，效果参数param

AddWndUpdateFunc(w, EffectEase, param,nil, nil);

--该代码给窗口做个0.5-1.5倍之间渐变放大效果

--备注:该接口依赖UpdateWindow()调用,因此UpdateWindow()一般在OnUpdate事件调用被调用.

local function OnUpdate(window, msg, sender)

UpdateWindow();

end

--以及事件绑定

event\_callback ["OnUpdate"] = OnUpdate;

设置绑定LuaLogic组件的窗口属性Status:IsActive为true即可触发OnUpdate事件.

**HelperCoroutine(func);**

**--创建协程, 参数lua接口或者代码**

示例：

HelperCoroutine(function(thread)

--动画参数

local param={type=tween.CIRC,fn=tween.easeOut,begin=0,offset=-0.5,change=0.5, duration=500,reset=true,attribute="CLXZWindow:Scale:fScaleX"};

--将动画加入w窗口, 效果接口EffectErase，效果参数param

AddWndUpdateFunc(w, EffectEase, param,thread, 1);

--等待动画结束

coroutine.yield();

--上个动画结束后，开始新的动画

local param={type=tween.CIRC,fn=tween.easeOut,begin=0,offset=-0.5,change=0.5, duration=500,reset=true,attribute="CLXZWindow:Scale:fScaleY"};

--将动画加入w窗口, 效果接口EffectErase，效果参数param

AddWndUpdateFunc(w, EffectEase, param ,thread, 1);

--等待动画结束

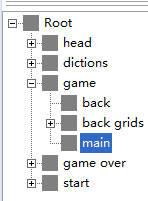
coroutine.yield();

end);

**HelperGetRoot()**

**--获得应用程序窗口根节点**

示例:



--获得根节点

local root=HelperGetRoot();

--通过根节点直接获得main节点窗口

local wnd = root:GetLXZWindow(“game:main”);

**HelperSetWindowText(wnd, text,rendername);**

**--设置窗口文字,wnd 窗口对象，text 设置的文字，rendername指定某个EditBox类型渲染对象名称,可为空**

**--窗口一定要有EditBox类型渲染组件，否则不起任何作用,相当于设置组件”EditBox:Text”的内容**

示例:

--给窗口文本组件内容设置成”Hello world!”.

HelperSetWindowText(wnd, “Hello world!”);

**HelperGetWindowText(wnd,rendername);**

**--获取窗口文本内容, wnd窗口对象，rendername指定EditBox类型渲染对象名称，可为空**

**--窗口必须存在EditBox类型，否则返回空。相当于获取组件”EditBox:Text”的内容**

示例:

--获取窗口wnd的文本内容

local text=HelperGetWindowText(wnd);

**HelperSetWindowSymText(wnd, text,rendername);**

**--设置窗口文字提示,wnd 窗口对象，text 设置的提示文字，rendername指定某个EditBox类型渲染对象名称,可为空**

**--窗口一定要有EditBox类型渲染组件，否则不起任何作用,相当于设置组件”** **EditBox:symContent”的内容，EditBox:Text提示内容为空时显示提示该内容。**

示例:

--给窗口文本组件提示内容设置成”Hello world!”,.

HelperSetWindowSymText (wnd, “Hello world!”);

**HelperGetWindowPictureFile(wnd,rendername)**

**--获取窗口图片文件名称,wnd窗口对象,rendername指定Picture类型渲染组件对象名称,可为空**

**--窗口必须存在Picture类型组件,否则返回值为nil。相当于获取组件”Picture:Sprite:file”属性**

示例:

--获取窗口wnd的图片文件

local pic\_file=HelperGetWindowPictureFile(wnd);

**HelperSetWindowPictureFile(wnd,filename,rendername)**

**--设置窗口图片文件名称,wnd窗口对象,filename图片文件名称,rendername指定Picture类型渲染组件对象名称,可为空**

**--窗口必须存在Picture类型组件,否则不起任何作用。相当于设置组件”Picture:Sprite:file”属性**

示例:

--设置窗口wnd的图片文件为”button.png”.

HelperSetWindowPictureFile(wnd, “button.png”);

**HelperGetWindowPictureColor(wnd, rendername);**

**--获取图片颜色,wnd窗口对象,rendername指定Picture类型渲染组件对象名称,可为空**

**--窗口必须存在Picture类型组件,否则返回nil。相当于获取组件”Picture:Sprite:color”属性中:red,green,blue,alpha的值**

示例:

--获取wnd窗口图片颜色

local r,g,b,a=HelperGetWindowPictureColor(wnd);

**HelperSetWindowPictureColor(wnd, rendername);**

**--设置图片颜色,wnd窗口对象, r 红色分量,g绿色分量,b蓝色分量,a透明度分量,rendername指定Picture类型渲染组件对象名称,可为空**

**--窗口必须存在Picture类型组件,否则不起任何作用。相当于设置组件”Picture:Sprite:color”属性中:red,green,blue,alpha的值**

示例:

--设置wnd窗口图片颜色成红色

HelperSetWindowPictureColor(wnd,255,0,0,255);

**HelperSetRectangleColor(wnd, r,g,b,a,rendername)**

**--设置方块填充颜色,r 红色分量,g绿色分量,b蓝色分量,a透明度分量,rendername指定Rectangle类型渲染组件对象名称,可为空**

**--窗口必须存在Rectangle类型组件,否则不起任何作用。相当于设置组件”** **Rect:FillColor”属性中red,green,blue,alpha的值**

**--窗口Rectangle组件中Rect:IsFill必须为true才起作用**

示例:

--设置该窗口成红色,

HelperSetRectangleColor(wnd, 255,0,0,255);

**HelperGetRectangleColor(wnd, rendername)**

**--获取方块填充颜色,rendername指定Rectangle类型渲染组件对象名称,可为空**

**--窗口必须存在Rectangle类型组件,否则返回nil。相当于获取组件”** **Rect:FillColor”属性中red,green,blue,alpha的值**

示例:

--获取该窗口颜色

local r,g,b,a=HelperGetRectangleColor(wnd);

**PlayEffect(sound\_file\_name)**

**--播放音效声音接口,sound\_file\_name 声音文件名，可以是mp3、wav、mini格式**

**--一般是比较短小的声音文件**

示例:

--播放”click.wav”

PlayEffect(“click.wav”);

**HelperGetWindowMask(wnd)**

**--获取窗口颜色,相当于获取CLXZWindow:Mask中的red,green,blue,alpha值**

示例:

local r,g,b,a=HelperGetWindowMask(wnd);

**HelperSetWindowMask(wnd,r,g,b,a)**

**--设置窗口颜色,相当于设置CLXZWindow:Mask中的red,green,blue,alpha值**

示例:

HelperGetWindowMask(wnd,r,g,b,a);

**HelperDBIntialize(dbname);**

**--创建或者加载sqlite3的db文件,dbname 文件名,返回sqlite3对象**

示例：

--创建game2048.db数据库文件,如果是lae工具运行，则创建的数据库名称是Editgame2048.db.

local db=HelperDBIntialize(“game2048.db”);

**HelperDBCreateTable(db,table\_name,remove)**

**--在数据库中创建表单, db sqlite3对象，table\_name 表名,remove 是否删除已存在的表单**

示例:

--如果已经存在表单”user\_info”，则先删除. 创建一个”user\_info”表单。

HelperDBCreateTable(db, “user\_info”, true);

**HelperDBExecute(db, sql, msgbox);**

**--执行SQL语句,db sqlite3对象，sql字符串，是否显示弹窗显示sql语句**

示例:

--从返回”users”表所有数据

HelperDBExecute(db, “select \* from users”, false);

**HelperDBHaveTable(db, tablename, field);**

**--是否存在表单,db sqlite3对象，tablename表单名，field字段名 可为空**

示例:

--是否存在表单”users”

local ret=HelperDBHaveTable(db, “users”);

--“users”是否存在字段name

local isexist= HelperDBHaveTable(db,”users”,”name”);

**HelperDBSelect(db,sql,t);**

**--返回select的结果，db sqlite3对象，sql(select语句),t 是否返回table类型**

**--t 为true时，返回值为table类型；为false,返回rows迭代器(iterator)**

示例1:

--返回users表所有内容，并填充到table中。

local users= HelperDBSelect(db, “select \* from users”, true);

--遍历打印所有用户名字

for i=1,#users,1 do

print(users[i].name);

end

示例2:

--返回users表所有内容。

local rows= HelperDBSelect(db, “select \* from users”, false);

if rows == nil then

return;

end

--遍历打印所有用户名字

for row in rows do

print(row.name);

end

**HelperDBUpdate(db,tablename,kvtbl,select,msgbox);**

**--更新记录,db sqlite3对象，tablename表名,更新内容{k=v},select条件，msgbox是否最终显示弹窗**

示例:

--更新DeviceList表，设置name=txt 当devid=lightid时.

HelperDBUpdateL(db, "DeviceList", {name=txt}, "devid="..lightid);

**HelperDBAlterTable(db, tblname, addfield, columntype,default)**

**--修改表结构,db sqlite3对象,addfield添加字段,字段类型，默认值**

示例:

--创建Light表

HelperDBCreateTable(database,"Light", false);

--给Light表添加字段lightid,’INTEGER’类型,默认值10000

HelperDBAlterTable(database,"Light", "lightid", 'INTEGER',10000);

--给Light表添加字段groupid,’INTEGER’类型,默认值1000

HelperDBAlterTable(database,"Light", "groupid", 'INTEGER',1000);

**EffectEase(wnd, t)**

**--EffectEase效果接口，通过时间算法接口更新窗口属性，达到动画效果目的,wnd窗口对象，t是table类型参数。**

**--t包含attribute属性名，render（如果是空指窗口本真，不为空，则是渲染组件)，type 算法函数, time,begin开始时间,change幅度,duration执行时间,offset起始时间，reset执行结束是否恢复原值,hide执行结束是否隐藏窗口,del执行结束是否删除窗口。**

示例:

--2048游戏中，刷出新的数字时，从0.5开始，幅度是0.5用tween.CIRC算法，调用tween.easeOut，执行时间500ms,结束是恢复窗口缩放系数.

AddWndUpdateFunc(w, EffectEase,{type=tween.CIRC, fn=tween.easeOut, begin=0, offset=-0.5, change=0.5, duration=500,reset=true,attribute="CLXZWindow:Scale:fScaleX"},nil, 1);

AddWndUpdateFunc(w, EffectEase,{type=tween.CIRC, fn=tween.easeOut, begin=0, offset=-0.5, change=0.5, duration=500,reset=true,attribute="CLXZWindow:Scale:fScaleY"},nil, 2);

--从0.5开始，幅度是3.14159用tween.CIRC算法，调用tween.easeOut，执行时间500ms,对图片组件作旋转角度控制,结束是恢复图片的原有的角度.

AddWndUpdateFunc(w, EffectEase,{render=”Picture”,type=tween.CIRC, fn=tween.easeOut, begin=0, offset=0, change=3.14159, duration=500,reset=true,attribute=" Picture:Sprite:fRadia"},nil, 2);

**EffectContainer (wnd, tbl)**

**--对多个效果进行打包, 比如{playcnt=1,effect={{f=接口，p={参数}},{f=接口，p={参数}}, {f=接口，p={参数}}}},动画效果按顺序播放。**

**--AddWndUpdateFunc (wnd, func, param, thread, id, updateself)，作为func参数使用**

示例:

--下面代码是w窗口对象先做个x轴方向缩放效果，结束后换成Y轴方向再做个缩放效果动画,总共做1次。

local e1 = {f= EffectEase, p=,{type=tween.CIRC, fn=tween.easeOut, begin=0, offset=-0.5, change=0.5, duration=500,reset=true,attribute="CLXZWindow:Scale:fScaleX"}};

local e2 = {f= EffectEase, p=,{type=tween.CIRC, fn=tween.easeOut, begin=0, offset=-0.5, change=0.5, duration=500,reset=true,attribute="CLXZWindow:Scale:fScaleY"}};

AddWndUpdateFunc(w, EffectContainer, {playcnt=1,effect={deepcopy(e1),deepcopy(e2)}, playcnt=1}, nil, 1);

**EffectTimer（wnd,tbl）**

**--应用于异步执行或者等待时间**

**--{relativetime=100,** **func=异步调用接口,playcnt=1}**

示例1:

--每间隔100毫秒，执行一次func，总共10次

AddWndUpdateFunc(w, EffectTimer, { relativetime=100, playcnt=10,func=function(wnd, tbl) end}, nil, 1);

示例2:

--两个动画之间等待10秒钟

HelperCoroutine(function(thread)

--动画参数

local param={type=tween.CIRC,fn=tween.easeOut,begin=0,offset=-0.5,change=0.5, duration=500,reset=true,attribute="CLXZWindow:Scale:fScaleX"};

--将动画加入w窗口, 效果接口EffectErase，效果参数param

AddWndUpdateFunc(w, EffectEase, param,thread, 1);

--等待动画结束

coroutine.yield();

--等待10秒钟

AddWndUpdateFunc(w, EffectTimer, { relativetime=10000 }, thread, 1);

coroutine.yield();

--上个动画结束后，开始新的动画

local param={type=tween.CIRC,fn=tween.easeOut,begin=0,offset=-0.5,change=0.5, duration=500,reset=true,attribute="CLXZWindow:Scale:fScaleY"};

--将动画加入w窗口, 效果接口EffectErase，效果参数param

AddWndUpdateFunc(w, EffectEase, param ,thread, 1);

--等待动画结束

coroutine.yield();

end);

**HelperEncodeURI(url)**

**--对编码处理将需要转码的字符（非字母和数字）转为16进制,比如”[: .= 空格]”**

示例:

--对“http://www.github.com?userid=1000”进行编码,返回结果是”http%3A%2F%2Fwww.github.com%3Fuserid%3D1000”

local encoded=HelperEncodeURI(“[www.github.com](http://www.github.com)”);

**HelperDecodeURI(s)**

**--对由HelperEncodeURI 编码的16进制字符串解码,返回url.**

示例:

--对“http%3A%2F%2Fwww.github.com%3Fuserid%3D1000”进行解码,返回结果是” http://www.github.com?userid=1000”

local encoded=HelperDecodeURI (“http%3A%2F%2Fwww.github.com%3Fuserid%3D1000”);

**HelperGetDayCount(year,month)**

**--获取某年某月的天数**

示例:

--获取2016年2月份的天数为29天

local days=HelperGetDayCount(2016,2);

**deepcopy(obj)**

**--拷贝table对象**

示例:

local t1={a=1,b=2,c=3,d={k=1}}; --构建table t1

local t2=t1; --让t2指向t1，而不是拷贝, 此时t1、t2等同

local t3=deepcopy(t1); --t3是完全独立的内存,内容和t1,t2完全一样.

t1.a=555; --修改t1.a值，此时t2.a也是555,但t3.a仍然是1

**HelperPostJSON(url,tbl,fn)**

**--HTTP请求, url请求地址,kv table{a=v1,b=v2},fn 响应接口**

示例1:

--提交分数

local Bonus =10000;

HelperPostJSON("http://115.28.78.174:80/rank", {level=1, bonus=Bonus},nil);

示例2：

***可参考博客:http://my.oschina.net/u/1030910/blog/725493***

--从服务器获取数据

local service\_host = "http://120.55.84.103:8080/";--服务器地址

local function OnServer(window, msg, sender)

local root=HelperGetRoot();--获得根节点

local wnd=root:GetLXZWindow("id");--获得id窗口

--创建协程，在协程里可以顺序执行异步代码

HelperCoroutine(function(thread)

HelperPostGetJSON(service\_host.."getid", {}, function(urlresponse)--异步调用接口

LXZMessageBox("urlresponse:getResponseData:"..urlresponse:getResponseData())

local res = json.decode(urlresponse:getResponseData());

if res then

HelperSetWindowText(wnd, res.id);

else

HelperSetWindowText(wnd, "从服务器获取失败");

end

coroutine.resume(thread);

end);

coroutine.yield();

end);

end