

TI 图形计算器 Lua 编程入门

TI-Nspire 图形计算器，能够轻松进行简单的算法编程，其语言为 TI-Basic，算法编程内容支持高中数学的学习，同时也能通过编写公用库程序的方式，增加产品的内置功能。在 TI-Nspire 3.0 版中，增加了 Lua 脚本功能。Lua 语言是成熟的、流行的和灵活的脚本语言，与 TI-Nspire 手持设备的整合，使 TI-Nspire 平台的编程能力得以增强，也能更好的拓展 TI 技术在数学与科学方面的应用，如图 1 的元素周期表，就是 Lua 编写的一个程序。下面笔者结合自己的摸索经历，简单的介绍一下 TI-Nspire 平台下 Lua 的使用。



图 1

一、Lua 语言简介

Lua 是一种小巧的脚本语言，其设计目的是为了嵌入应用程序中，从而为应用程序提供灵活的扩展和定制功能，Lua 语言的官网主页为 www.lua.org。Lua 本身简单，小巧；内容少但功能强大，这使得 Lua 易于学习，很容易实现一些小的应用。

Lua 由标准 C 编写而成，代码简洁优美，几乎在所有操作系统和平台上都可以编译与运行。一个完整的 Lua 解释器不过 200k，在目前所有脚本引擎中，Lua 的速度是最快的。这一切都决定了 Lua 是作为嵌入式脚本的最佳选择。

自 1995 年以来，Lua 已在各地拥有主流接受，并已在其他应用程序使用，特别是游戏中产品。它是 TI-Basic 程序员上升到一个发展的新水平，因为 Lua 通用的数据类型是高度兼容现有的 TI-Nspire 数据类型，能够以小内存占用。Lua 整合到 TI-Nspire 之上后，运行的 Lua 程序能够调用 TI-Nspire 原有的功能。

二、使用 Lua 的方法

在 TI-Nspire 平台下使用 Lua 程序，需要先用记事本等写程序代码的编辑器，写出一个 Lua 程序文件（特别注意的是 Lua 对大小写比较敏感），其扩展名命名为“Lua”，再利用工具软件“TI-Nspire™ Scripting Tools”将代码转化到 TI-Nspire 平台上。

1、简单的 Lua 程序—灯笼

先用编程软件写出一个简单的程序（代码如下），命名为“denglong.lua”。

```
function on.paint(gc)
  local x, y = screen_x/2, screen_y/2
  gc:drawArc(x,y,2,0,360)
  gc:drawString(string.uchar(28783,31548),10,25)
  for i=1,5 do
    x1 = x + 30 * i
    y1 = y
    gc:setColorRGB(50*i,255-50*i,100)
    for j=5,360,5 do
```

```

x2 = x + 30 * i * math.cos(math.rad(j))
y2 = y + 100 * math.sin(math.rad(j))
gc:drawLine(x1,y1,x2,y2)
x1, y1 = x2, y2
end
end
end
function on.resize(width,height)
screen_x = width
screen_y = height
end

```

2、工具软件的使用

第一步，到 TI 官网下载工具软件（TI-Nspire™ Scripting Tools）并解压，网址为：

<http://education.ti.com/calculators/downloads/US/Software/Detail?id=6840>

第二步，运行 TI-Nspire 学生版或教师版软件，新建一个 TI-Nspire 的文档。

第三步，运行 TI-Nspire Scripting Tools.exe，点击菜单 Tools 下的“Load Script App”（如图 2），再选择之前编写的文件，确定之后弹出如图 3 的提示框即为转换成功。

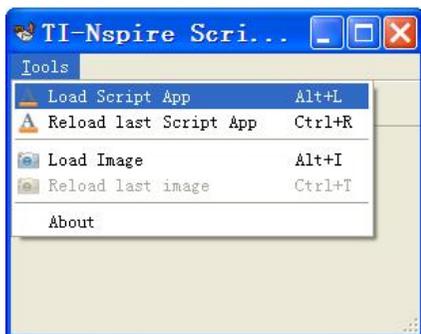


图 2



图 3

第四步，返回之前运行的 TI-Nspire 学生版或教师版软件中，添加一个页面，停留在菜单页（如图 4），选择软件“编辑”菜单中的“粘贴”（Lua 源代码就粘贴到文档当前页面里面了，同时会自动执行），那么 Lua 程序运行之后的效果，以一个新的页面展示在 TI-Nspire 文档中，前面“denglong.lua”的效果如图 5 所示。

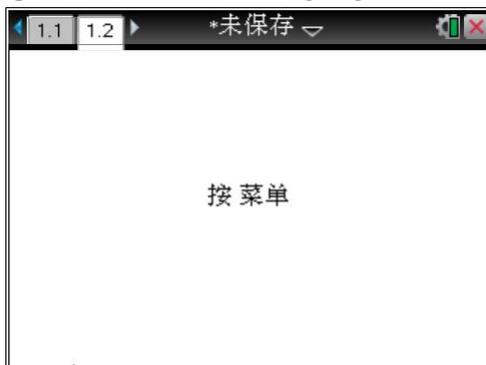


图 4

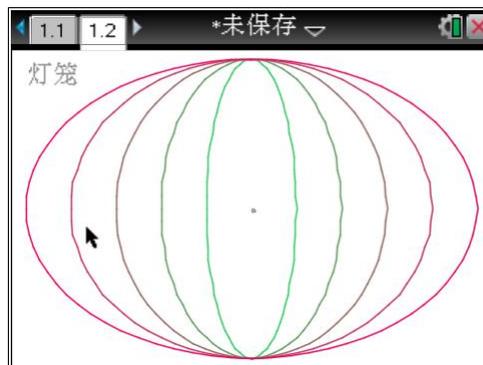


图 5

三、灯笼程序的代码释义

笔者采用的文本编辑器是“Notepad++ V5.9.3”，编辑器中的代码样式如下：

```
2  function on.paint(gc)
3      local x, y = screen_x/2, screen_y /2
4      gc:drawArc(x,y,2,2,0,360)
5      gc:drawString(string.uchar(28783,31548),10,25)
6      for i=1,5 do
7          x1 = x + 30 * i
8          y1 = y
9          gc:setColorRGB(50*i,255-50*i,100)
10         for j=5,360,5 do
11             x2 = x + 30 * i * math.cos(math.rad(j))
12             y2 = y + 100 * math.sin(math.rad(j))
13             gc:drawLine(x1,y1,x2,y2)
14             x1, y1 = x2, y2
15         end
16     end
17 end
18
19 function on.resize(width,height)
20     screen_x = width
21     screen_y = height
22 end
```

下面对以上代码的主要语句进行解释。

① function on.paint(gc)

类型：事件处理函数，TI-Lua 扩充语句。 **功能：**窗口绘制事件。

说明：对窗口进行绘制的应用程序脚本，使用 GC 图形方式绘制窗口。任何对屏幕的绘制操作都在这个事件里面进行绘制。如果屏幕需要重绘，可重新调用 paint 函数。每次执行程序，paint 都会被调用。当程序界面绘制时会引发此事件，任何对屏幕的显示等操作需要发生了此事件才会生效，另外当文档第一次载入和切换到其他页面再切换回来也会发生此事件。使用应注意不要省略参数列表中的 gc 对象。

② function on.resize(width, height)

类型：事件处理函数，TI-Lua 扩充语句。 **功能：**窗口大小改变事件。

说明：脚本的应用程序的窗口大小变化时引发这个事件，一般用于程序的初始化设计，或根据窗口大小的调整图形对象的位置与大小。参数列表 width 为窗口宽度，height 为窗口高度。

③ local

类型：标准 Lua 基本语句。 **功能：**定义局部变量。

说明：局部变量只在 local 所在的语句块内有效。在定义时可以同时对变量进行赋值。

④ For 变量=初值, 终值, 步长 do...(语句块) end

类型：标准 Lua 基本语句。 **功能：**依循环条件执行语句体。

说明：For 的循环变量总是只作用于 For 的局部变量；当省略步长时，For 循环会使用默认值 1 作为步长；在语句体中，可使用 Break 中止循环。

⑤ **gc:drawArc(x, y, width, height, startAngle, endAngle)**

类型: 图形库的方法, TI-Lua 扩充语句. **功能:** 画弧.

说明: 在距左上角 (x, y) 的位置, 像素宽为 **width**, 高为 **height** 的矩形内绘制一段圆弧, 圆弧绘制时从 **startAngle** 度开始, 在 **endAngle** 度结束. 零度点在右边, 度数顺时针旋转. 若要画一个圆, 像素宽和高必须相等, 开始和结束点的角度必须为 0 和 360, 若宽与高不同, 将绘制一个椭圆.

⑥ **gc:drawString("text", x, y [,verticalignment])**

类型: 图形库的方法, TI-Lua 扩充语句. **功能:** 绘制文本.

说明: 在窗口左上角像素位置 (x, y) 开始绘制文本 **text**, 垂直对齐方式为可选项, 分别是 "baseline" (基线), "bottom" (底), "middle" (中间), 或 "top" (顶). "**text**" 可以是具体的文本, 也可以用字符串变量或表达式.

⑦ **string.uchar(chnum, chnum, ...)**

类型: 字符串库函数, TI-Lua 扩充语句. **功能:** 转换字符.

说明: 参数列表为 Unicode 字符在 UTF-8 编码的数字代码, 它能转换一个或多个 UTF-8 字符串的 Unicode 字符, 参数最多 20 个, 函数返回值是一个转换后的字符串. (可以利用这个来得到非 ASCII 字符串)

⑧ **gc:drawLine(x1, y1, x2, y2)**

类型: 图形库的方法, TI-Lua 扩充语句. **功能:** 画线.

说明: 绘制一条线, 从 (x1, y1) 到 (x2, y2).

⑨ **gc:setColorRGB(red, green, blue)**

类型: 图形库的方法, TI-Lua 扩充语句. **功能:** 设置颜色.

说明: 设置后续绘制和填充的颜色, 红、绿、蓝的范围为 0 到 255, 黑色是 0,0,0, 白色是 255,255,255. (采用 RGB888 标准, 即 24 位真彩色)

编写程序的目的是为了使解决一个问题具有简捷性、移植性, 下面将灯笼程序代码稍稍变化, 得到一个指纹的图像.

灯笼程序变化代码 (指纹) 如下, 其效果如图 6 所示.

```
2 function on.paint(gc)
3     local x, y = screen_x/2, screen_y /2
4     gc:drawString(string.uchar(25351,32441),10,25)
5     local x1, y1 = x, y
6     radio = 1
7     for i=1,5 do
8         gc:setColorRGB(50*i,255-50*i,100)
9         for j=5,360,5 do
10            x2 = x + radio * math.cos(math.rad(j))
11            y2 = y + radio * math.sin(math.rad(j))
12            gc:drawLine(x1,y1,x2,y2)
13            x1, y1 = x2, y2
14            radio = radio + .3
15        end
16    end
17 end
18
19 function on.resize(width,height)
20     screen_x = width
21     screen_y = height
22 end
```



图 6

四、一个较复杂的 Lua 程序

程序代码如下：

```
2  a = image.new("\168\000\000\000\236\000\000\000\000\000\000\080\001\000\000\016\0
3
4  jd = 0
5  sd = true
6
7  function on.paint(gc)
8      gc.drawImage(a,135,0)
9      wjx = Star(65, 45, 20)
10     wjx:paint(gc)
11     wjx = Star(65, 125, 20)
12     sd = not sd
13     if sd then
14         wjx.r = 10
15     end
16     wjx:paint(gc)
17     local x0, y0 =65, 125
18     r0 = var.recall("dx")
19     s0 = var.recall("gs")
20     d0 = var.recall("sd") * 5
21     for i=1,s0 do
22         wjx = Star(x0+ 40* math.cos(math.rad((i-1)*360/s0 -90+jd)),
23                 y0 - 40 * math.sin(math.rad((i-1)*360/s0 -90+jd)), r0)
24         wjx:paint(gc)
25     end
26     gc.drawString(string.uchar(25105,29233,27611,20027,24109),20,25)
27 end
28
29 function on.create()
30     timer.start(1/3)
31 end
32
33 function on.timer()
34     jd = jd +d0
35     if jd == 360 then
36         jd = 0
37     end
38     platform.window:invalidate()
39 end
40
41 Color = {
42     red = {0xFF, 0x00, 0x00}, green = {0x00, 0xFF, 0x00},
43 }
44
45 Star = class()
46 function Star:init(x, y, r)
47     self.x = x
48     self.y = y
49     self.r = r or 20
50     self.color = Color.red
51 end
52
53 function Star:paint(gc)
54     gc.setColorRGB(unpack(self.color))
55     local x0, y0, r = self.x, self.y, self.r
56     local a, b = {}, {}
57     for i=1,5 do
58         a[(i-1)*2+1] = x0 + r * math.cos(math.rad((i-1)*72+18))
59         b[(i-1)*2+1] = y0 - r * math.sin(math.rad((i-1)*72+18))
60         a[2*i] = x0 + r * (1-0.618) * math.cos(math.rad((i-1)*72+54))
61         b[2*i] = y0 - r * (1-0.618) * math.sin(math.rad((i-1)*72+54))
62     end
63     gc:fillPolygon({a[1],b[1], a[2],b[2], a[3],b[3], a[4],b[4], a[5],b[5],
64                   a[6],b[6], a[7],b[7], a[8],b[8], a[9],b[9],a[10],b[10]})
65 end
```

先制作好如图 7 所示的界面，再将以上程序用工具转换后进行粘贴，效果如图 8。



图 7



图 8

图 7 中，下部分为几何页，隐藏单位显示，插入三个游标，设置最小化显示，**dx** 关联环绕的五角星的大小，值的范围为 5~25，步长为 5；**gs** 关联环绕的五角星的个数，值的范围为 3~8，步长为 1；**sd** 关联五角星转动速度，值的范围为 1~6，步长为 1。

本例程序涉及到 Lua 的许多关键用法，例如访问 Nspire 中的数据、定时器的运用、图像库的运用，特别是类的使用。下面对本例 Lua 程序的主要语句进行解释。

① **image.new(str)**

类型：图片库函数，TI-Lua 扩充语句。 **功能：**新建图像。（返回 image 类对象）

说明：返回一个字符串对应的图像对象，该字符串由网友编写的工具软件进行转换。图像通常是小的图形用户界面对象，如按钮、箭头。一般图像字符串比较大，可能有几十到几百 kb 不等，如果这样用记事本就比较吃力了。

② **gc:drawImage(image, x, y)**

类型：图形库方法，TI-Lua 扩充语句。 **功能：**绘制图像。

说明：在 (x, y) 处绘制图像，之前必须调用 image.new 新建图像对象。

③ **var.recall(name)**

类型：符号表库函数，TI-Lua 扩充语句。 **功能：**返回 Nspire 变量值。

说明：返回 Nspire 中给定名称的数学变量的值。符号表是用来访问 Nspire 数学引擎的计算和存储值。

④ **timer.start(period)**

类型：定时器库方法，TI-Lua 扩充语句。 **功能：**定时器启动触发。

说明：在一定周期内启动定时器。每个脚本的应用程序有一个定时器，定时器的速率取决于平台，手持设备最少 0.01 秒触发。脚本应用程序需执行“on.timer ()”函数来响应计时器事件。

⑤ **on.timer()**

类型：事件处理方法，TI-Lua 扩充语句。 **功能：**定时器触发事件。

说明：定时器每次时间间隔触发后引发此事件。

⑥ **class([parent_class])**

类型：类定义方法，TI-Lua 扩充语句。 **功能：**类定义。

说明：返回一个新的类，如果指定一个父类，新类继承父类的方法。类可以继承，是一种对象，能定义并设置方法、属性等。

五、补充说明

在 TI-Nspire 平台下, Lua 是面向对象的编程语言, 所以使用时应当对面向对象设计程序有一定了解, 认真阅读 Lua 的语法手册, 了解运算符、数据类型、基本函数、对象、属性、方法、类、标准库等编程必备知识, 推荐阅读《lua 程序设计》。

在新的 TI-Nspire OS3.2 推出之后, 学生版与教师版软件都内置了代码编辑器(图 9), 解决了以上程序中汉字输入与注释、图片插入的问题。新的软件的代码编辑器, 相当于在 TI 技术平台上增添了 Lua 这一流行语言的编写平台, 可算得上是又一次锦上添花。

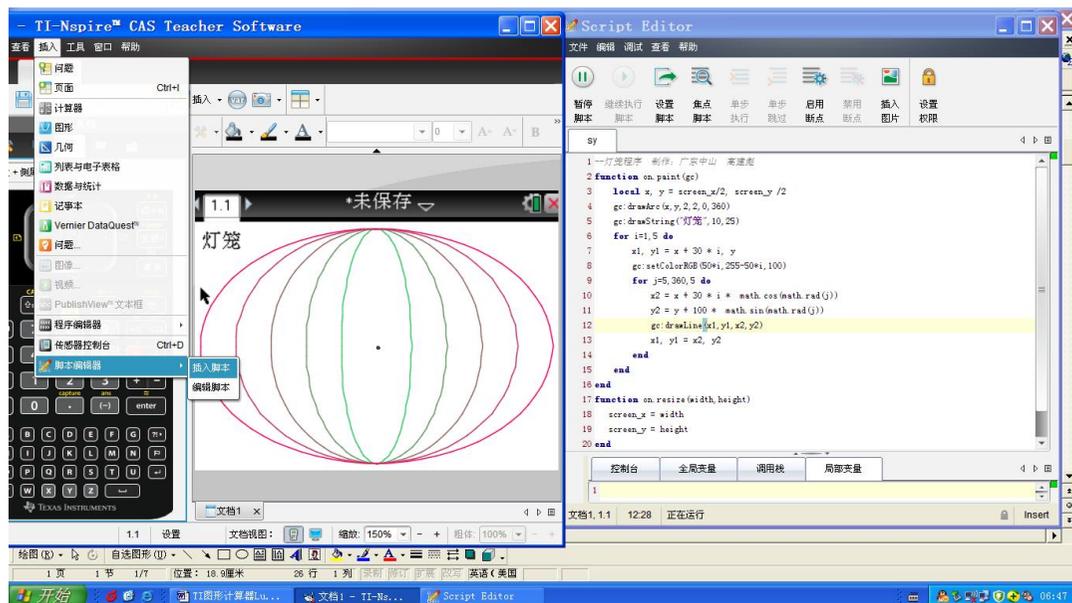


图 9

在撰写此教程的过程中, 由于笔者英文基础极差, 只能借助 google 翻译研究 TI 官方工具包中的文件“TI-Nspire Scripting Interface.doc”及国外关于 TI Nspire 与 Lua 的教程, 所以理解与表述中有许多地方不太准确, 敬请大家指正, 也特别感谢两位中学生网友“wtof1996”、“imath”对我学习 Lua 的帮助, 希望 TI 公司能尽早将“TI-Nspire Scripting Interface.doc”进行翻译。

本教程中涉及到的程序源代码和 TNS 文件, 下载地址为<http://sx.zsedu.net/ti/luajc.rar> 附上几个资源网站如下:

- ① 开源脚本语言 Lua 学习专题: <http://tech.it168.com/zt/lua/>
- ② Lua 中国开发者: <http://www.luaer.cn/>
- ③ 学习用书--《lua 程序设计》:
<http://wenku.baidu.com/view/5460898471fe910ef12df80a.html>
- ④ 国外关于 TI Nspire 与 Lua 的教程:
http://compasstech.com.au/TNSINTRO/TI-NspireCD/Start_Here.html
- ⑤ 网友开发的工具: Lua Booster V1.31
<http://fxesms.5d6d.net/viewthread.php?tid=7465>

(作者: 高建彪 邮箱:dsgjb@163.com, QQ:76456245 2012年5月20日完稿于中山市东升高中)
特别说明: 本资料的整理完成, 感谢两位中学生网友“wtof1996”、“imath”的指导与支持。