APCAD 开发平台

入门指导与开发实践

阿依艾工程软件（大连）有限公司

入门篇

第一章　概述

第二章　开发

第一节　目标

对于软件开发而言，开始的第一个程序几乎都是相同的，即：

请打印输出以下内容：

Hello World！

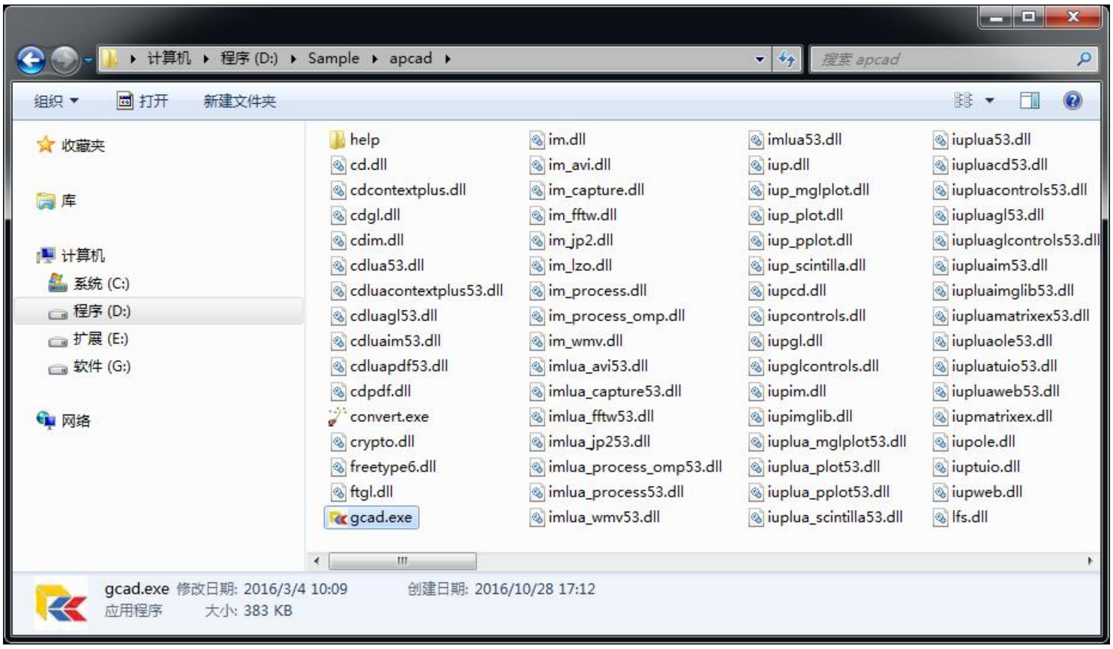
虽然这个程序很简单，但对于初学者来说，它仍然需要被认真对待，因为要实现这个目的，我们必须学会搭建基础的程序框架，编写程序文本，然后运行程序，把结果（Hello World！）输出到某个程序窗口。掌握了这些技术以后，就可以制作一个新的软件了。

这就开始吧！

第二节　准备

首先需要一台安装了64位的Windows操作系统的计算机，然后下载apcad开发平台（可以访问www.apcad.com 下载）到本地磁盘，安装（解压缩）到某个计算机文件夹（此文件夹必须为空文件夹，位置可由用户新建或指定）。

打开程序文件夹，能够看到有main.lua、gcad.exe以及多个扩展名为dll的文件，其中gcad.exe是可执行文件，执行该文件（一般可通过双击鼠标左键来执行）即可打开软件，并看到软件界面。

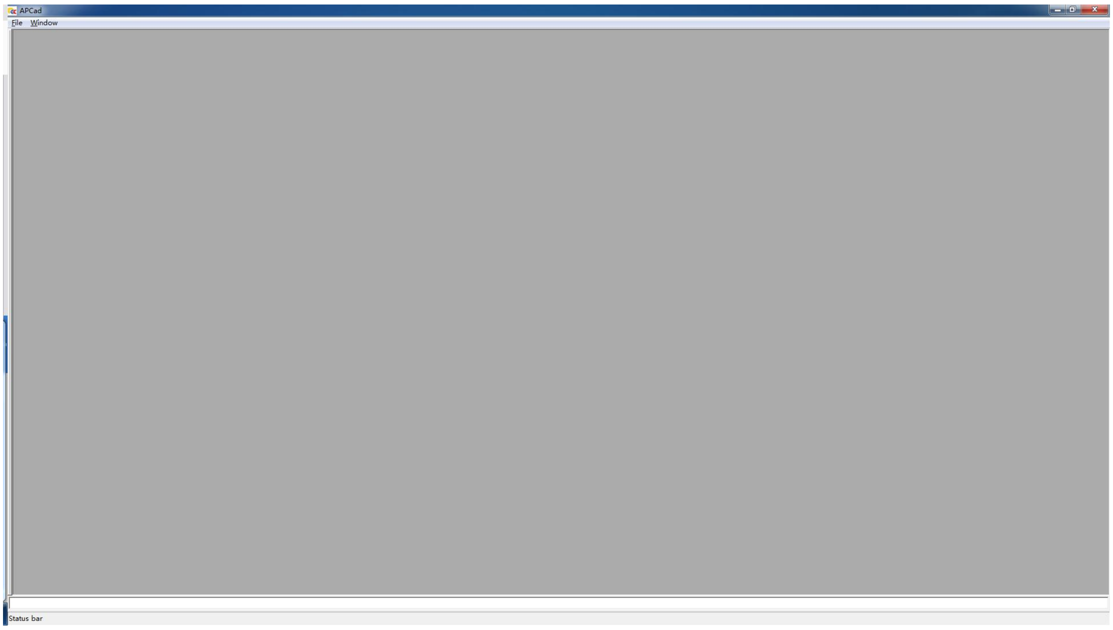


help
cd.dll
cdcontextplus.dll
cdgl.dll
cdim.dll
cdlua53.dll
cdluacontextplus53.dll
cdluagl53.dll
cdluaim53.dll
cdluapdf53.dll
cdpdf.dll
convert.exe
crypto.dll
freetype6.dll
ftgl.dll
gcad.exe
im.dll
im_avi.dll
im_capture.dll
im_fftw.dll
im_jp2.dll
im_lzo.dll
im_process.dll
im_process_omp.dll
im_wmv.dll
imlua_avi53.dll
imlua_capture53.dll
imlua_fftw53.dll
imlua_jp253.dll
imlua_process_omp53.dll
imlua_process53.dll
imlua_wmv53.dll
imlua53.dll
iup.dll
iup_mglplot.dll
iup_plot.dll
iup_pplot.dll
iup_scintilla.dll
iupcd.dll
iupcontrols.dll
iupgl.dll
iupglcontrols.dll
iupim.dll
iupimglib.dll
iuplua_mglplot53.dll
iuplua_plot53.dll
iuplua_pplot53.dll
iuplua_scintilla53.dll
iuplua53.dll
iupluacd53.dll
iupluacontrols53.dll
iupluagl53.dll
iupluaglcontrols53.dll
iupluaim53.dll
iupluaimglib53.dll
iupluamatrixex53.dll
iupluaole53.dll
iupluatuio53.dll
iupluaweb53.dll
iupmatrixex.dll
iupole.dll
iuptuio.dll
iupweb.dll
lfs.dll

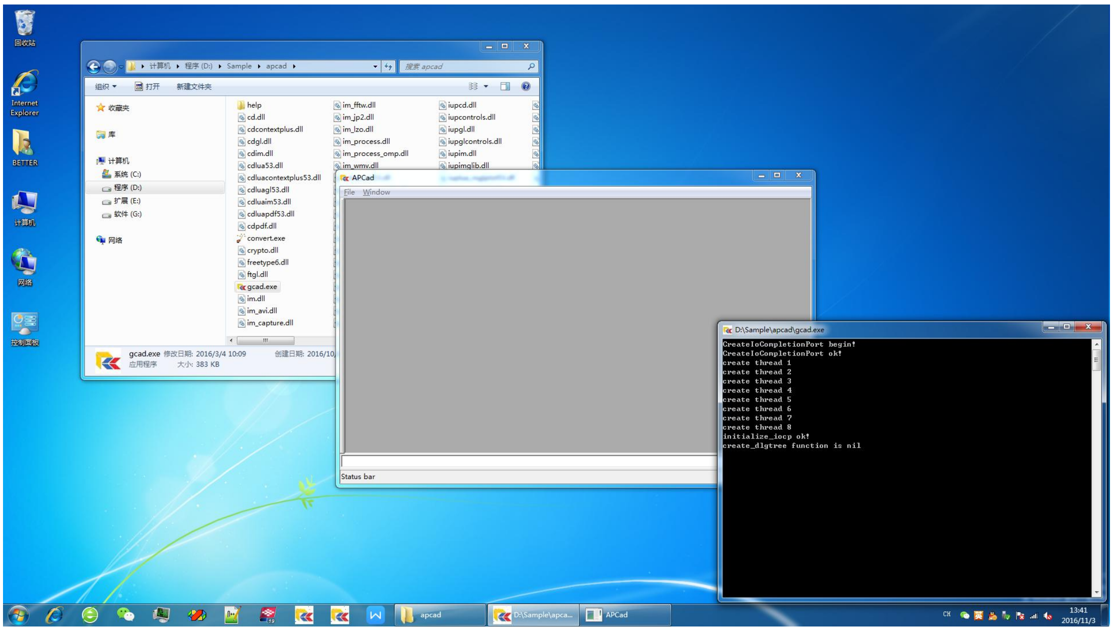
gcad.exe 修改日期: 2016/3/4 10:09 创建日期: 2016/10/28 17:12
应用程序 大小: 383 KB

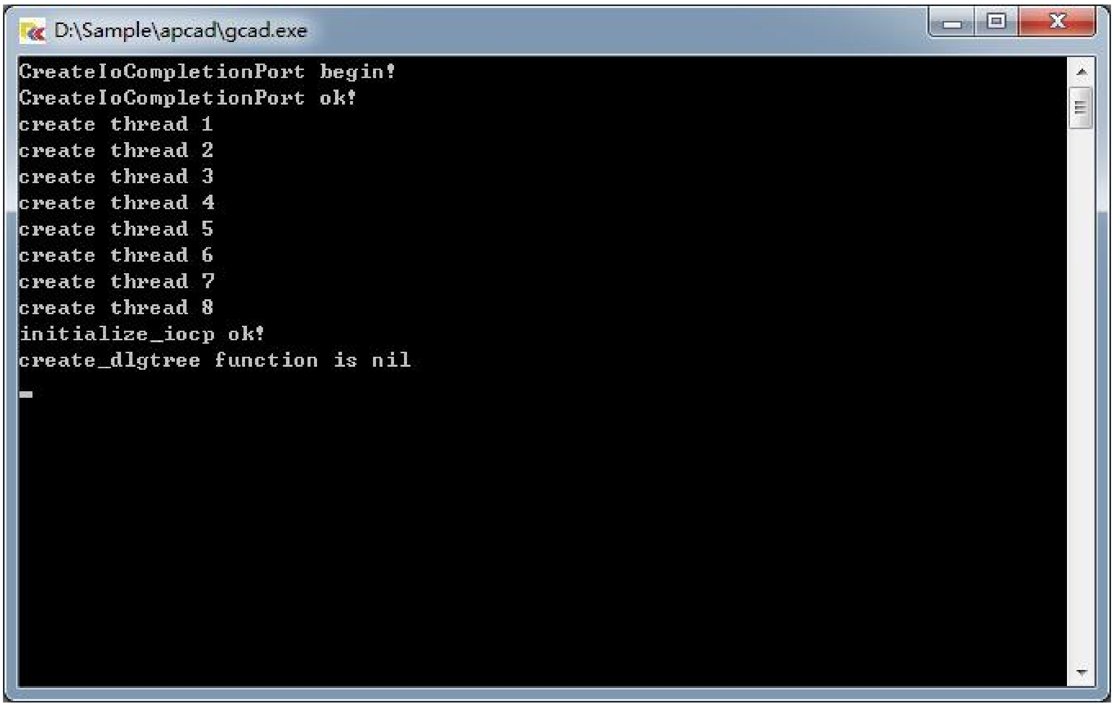
apcad 平台程序文件夹

运行主程序（gcad.exe），查看运行结果。



这里看到的是软件的前台主窗口，缩小窗口或者通过任务栏切换，能够看到软件的后台信息窗口。





```
CreateIoCompletionPort begin!
CreateIoCompletionPort ok!
create thread 1
create thread 2
create thread 3
create thread 4
create thread 5
create thread 6
create thread 7
create thread 8
initialize_iocp ok!
create_dlgtree function is nil
```

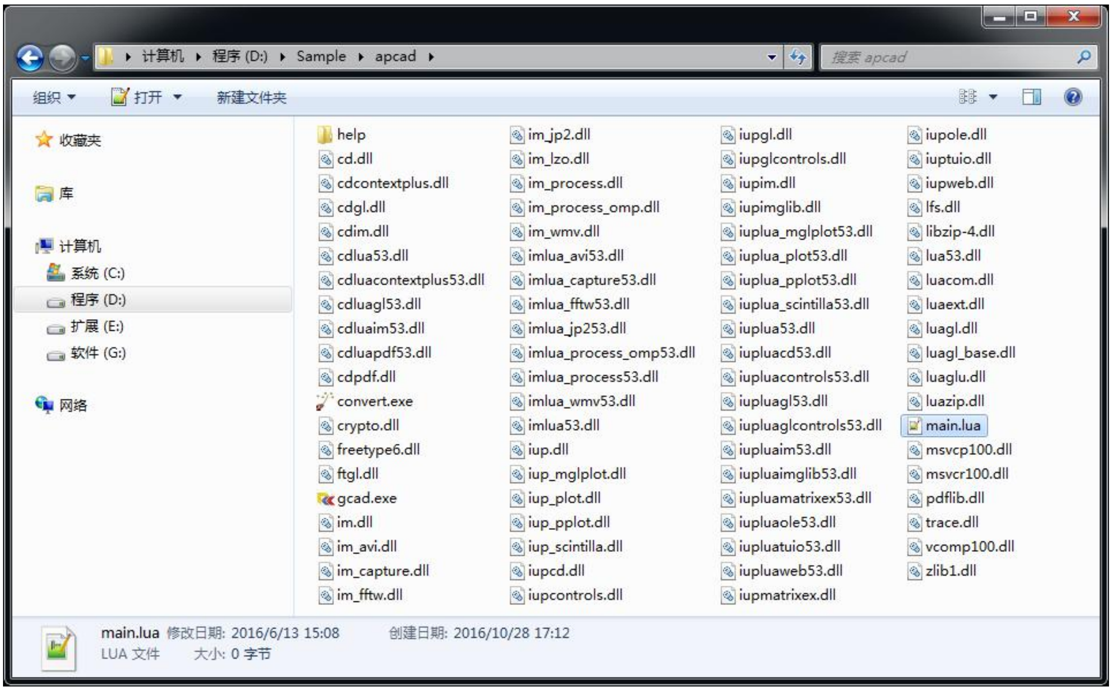
“Hello World”就将会在这个后台信息窗口中打印输出，不过，此时打开软件还不能看到”Hello World！”，因为我们还需要编写相应的程序代码。

*注：如果只能看到文件名（gcad）无法看到扩展名（exe），可以通过 Windows 文件夹设置显示已知文件的扩展名。

第三节　编码

在 apcad 平台中，我们可以用下面的方法打印输出“Hello World！”:

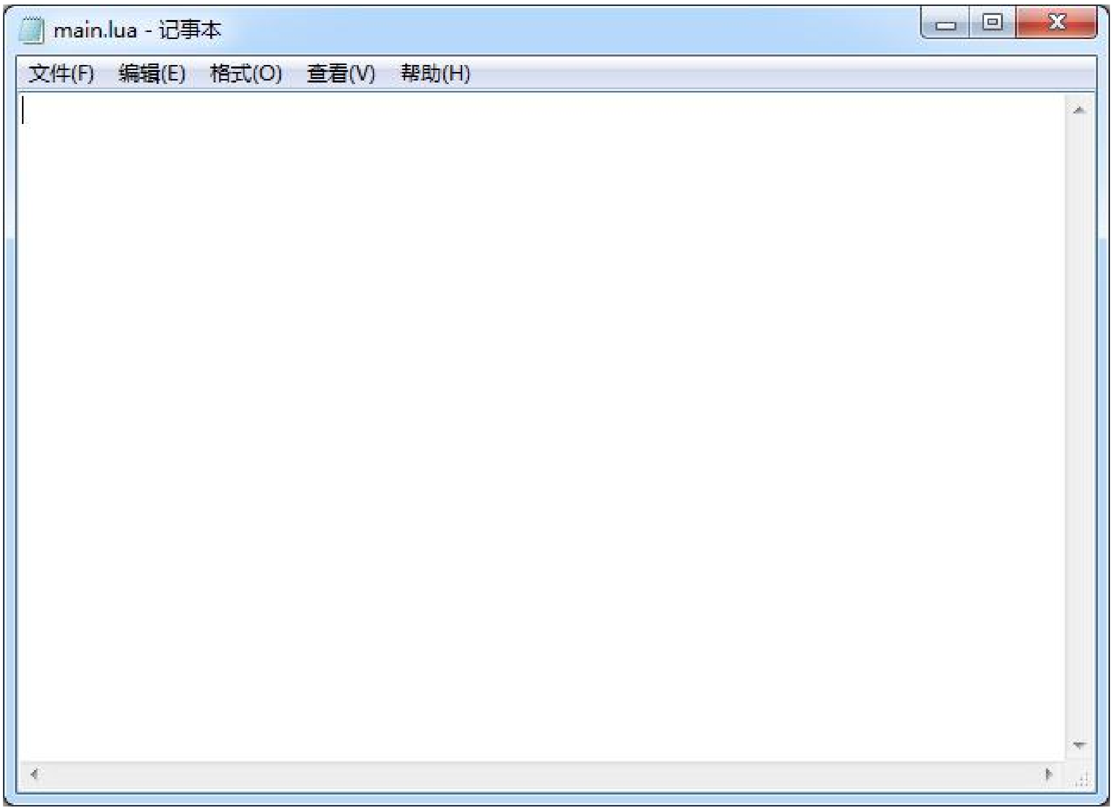
（一）打开计算机 apcad 程序文件夹。



help
cd.dll
cdcontextplus.dll
cdgl.dll
cdim.dll
cdlua53.dll
cdluacontextplus53.dll
cdluagl53.dll
cdluaim53.dll
cdluapdf53.dll
cdpdf.dll
convert.exe
crypto.dll
freetype6.dll
ftgl.dll
gcad.exe
im.dll
im_avi.dll
im_capture.dll
im_fftw.dll
im_jp2.dll
im_lzo.dll
im_process.dll
im_process_omp.dll
im_wmv.dll
imlua_avi53.dll
imlua_capture53.dll
imlua_fftw53.dll
imlua_jp253.dll
imlua_process_omp53.dll
imlua_process53.dll
imlua_wmv53.dll
imlua53.dll
iup.dll
iup_mglplot.dll
iup_plot.dll
iup_pplot.dll
iup_scintilla.dll
iupcd.dll
iupcontrols.dll
iupgl.dll
iupglcontrols.dll
iupim.dll
iupimglib.dll
iuplua_mglplot53.dll
iuplua_plot53.dll
iuplua_pplot53.dll
iuplua_scintilla53.dll
iuplua53.dll
iupluacd53.dll
iupluacontrols53.dll
iupluagl53.dll
iupluaglcontrols53.dll
iupluaim53.dll
iupluaimglib53.dll
iupluamatrixex53.dll
iupluaole53.dll
iupluatuio53.dll
iupluaweb53.dll
iupmatrixex.dll
iupole.dll
iuptuio.dll
iupweb.dll
lfs.dll
libzip-4.dll
lua53.dll
luacom.dll
luaext.dll
luagl.dll
luagl_base.dll
luaglu.dll
luazip.dll
main.lua
msvcp100.dll
msvcr100.dll
pdflib.dll
trace.dll
vcomp100.dll
zlib1.dll

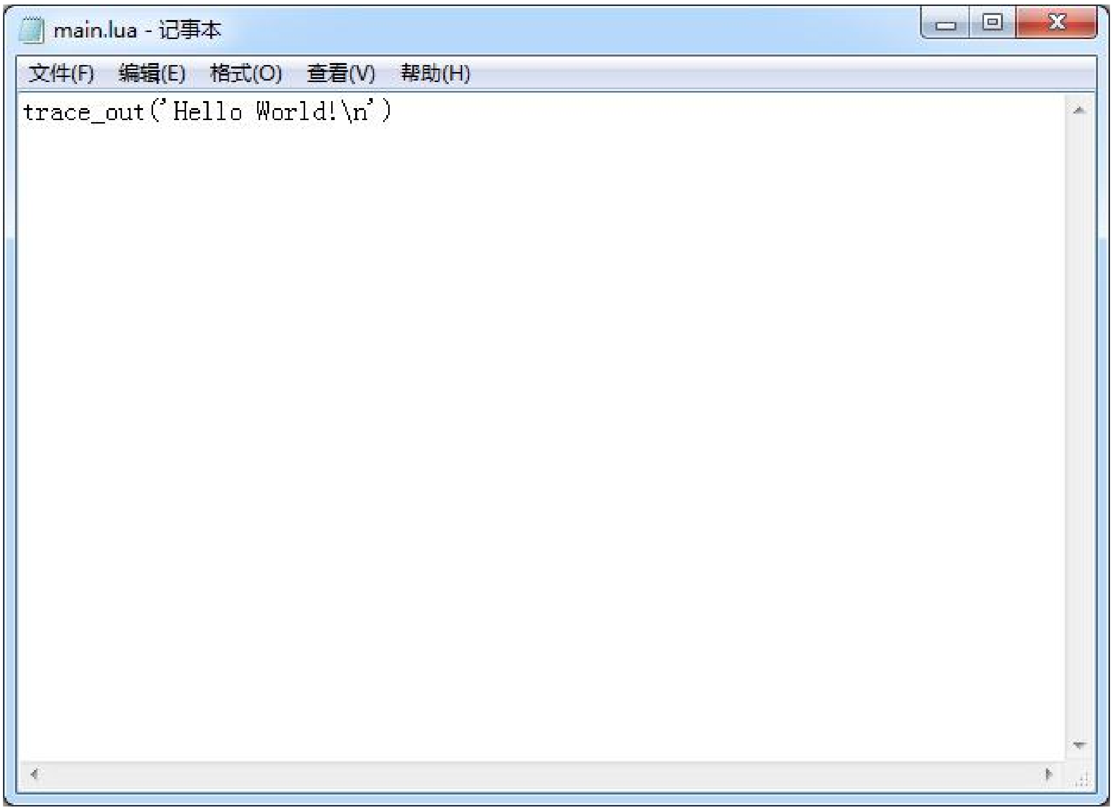
main.lua 修改日期: 2016/6/13 15:08 创建日期: 2016/10/28 17:12
LUA 文件 大小: 0 字节

（二）以纯文本方式（如 Windows 记事本）打开主文件（main.lua）。



（三）编写如下代码：

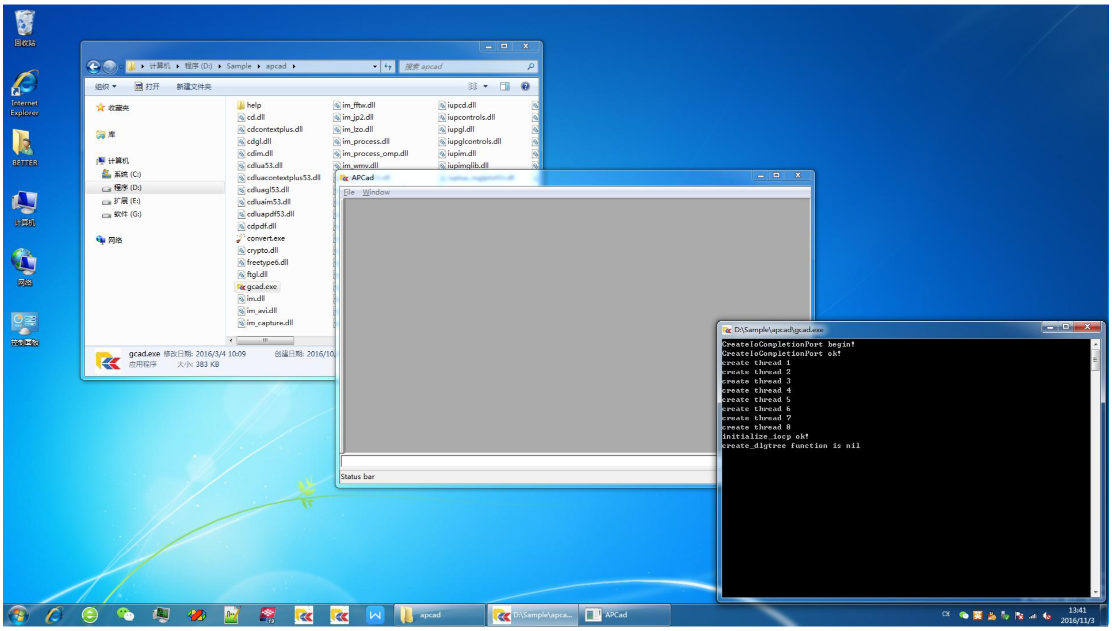
```
trace_out(“Hello World！\n”);
```



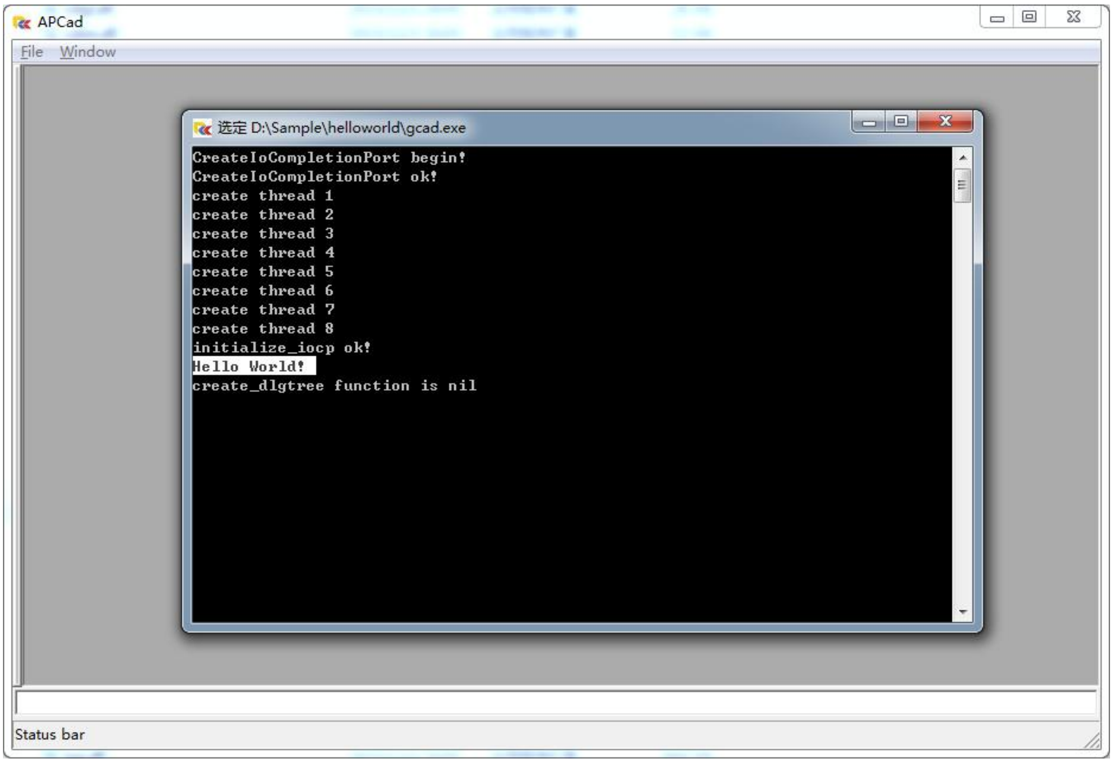
```
trace_out('Hello World!\n')
```

（四）保存代码文件（main.lua）。

运行主程序（gcad.exe），查看运行结果（如果主程序已经运行，需要关闭后重新运行，重新运行后，通过任务栏切换查看后台信息窗口）。



```
CreateIoCompletionPort begin!
CreateIoCompletionPort ok!
create thread 1
create thread 2
create thread 3
create thread 4
create thread 5
create thread 6
create thread 7
create thread 8
initialize_iocp ok!
create_dlgtree function is nil
```



```
CreateIoCompletionPort begin!
CreateIoCompletionPort ok!
create thread 1
create thread 2
create thread 3
create thread 4
create thread 5
create thread 6
create thread 7
create thread 8
initialize_iocp ok!
Hello World!
create_dlgtree function is nil
```

在 apcad 平台采用 Lua 语言编写程序代码，main.lua 是程序的主文件，用户编写程序的入口点，程序从这里开始执行第一行代码，直到文件结束。

*注：文件执行结束后，程序并没有真正结束，还可能会接收到来自平台的其他命令开始执行新的代码，这些内容会在后面的章节继续讨论。

trace_out 是平台提供的接口函数，它的功能是在平台后台窗口中输出一段文本，参数是需要输出的文本。

实战篇

第三章　三维建模软件

第一节　目标

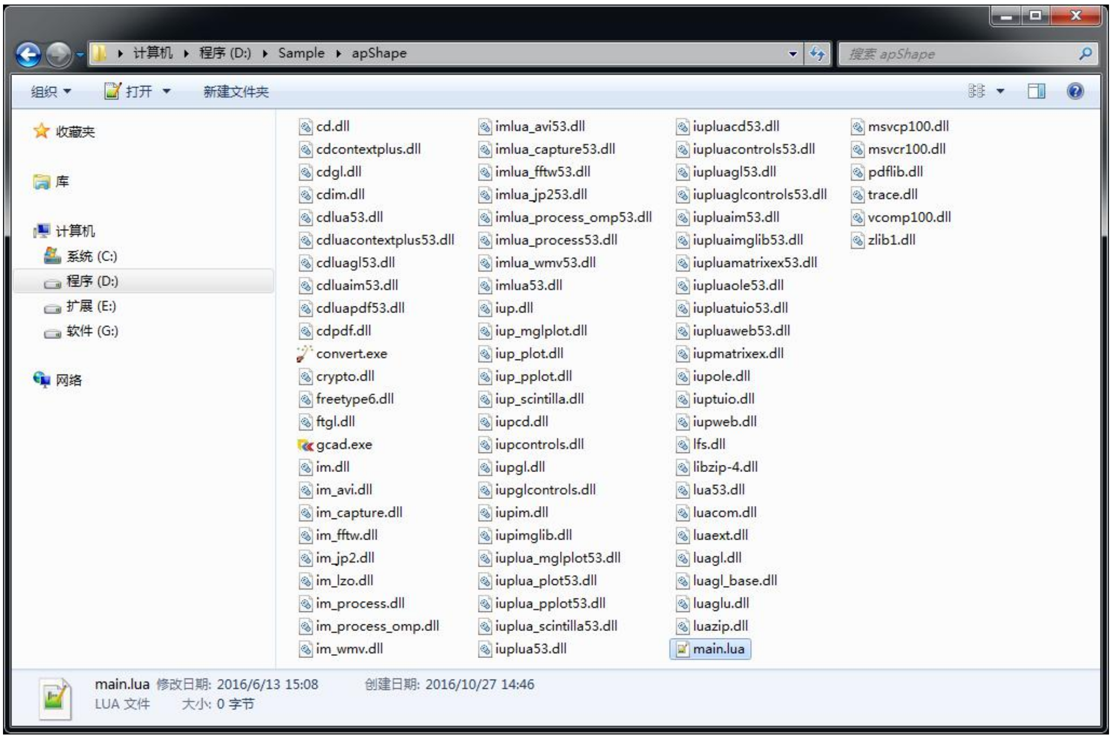
开发一个三维建模软件，实现绘制三维立方体，弹出属性对话框，修改三维立方体属性，以及相应鼠标消息、制作菜单工具条按钮等功能：

- 添加三维立方体
- 弹出属性对话框窗口
- 制作菜单和工具条按钮
- 使用鼠标可以选择某个立方体
- 使用鼠标绘制新的图块
- 捕捉
- 创建工作平面
- 制作停靠窗口
- 保存和打开

第二节　准备

和上面的”Hello World!”程序类似，现在也下载apcad开发平台（可以访问www.apcad.com 下载）并安装（解压缩）到某个计算机文件夹（此文件夹必须为空文件夹，位置可由用户新建或指定）。。

同样的，打开程序文件夹，能够看到有main.lua、gcad.exe以及多个扩展名为dll的文件，其中main.lua是主程序文件，以纯文本的方式打开并编辑该文件（可以使用Windows记事本或MS Notepad++，后面以Notepad++为例），输入所需代码（别急，后面的章节会讨论这些代码），制作这个新建的三维建模软件。



cd.dll
cdcontextplus.dll
cdgl.dll
cdim.dll
cdlua53.dll
cdluacontextplus53.dll
cdluagl53.dll
cdluaim53.dll
cdluapdf53.dll
cdpdf.dll
convert.exe
crypto.dll
freetype6.dll
ftgl.dll
gcad.exe
im.dll
im_avi.dll
im_capture.dll
im_fftw.dll
im_jp2.dll
im_lzo.dll
im_process.dll
im_process_omp.dll
im_wmv.dll
imlua_avi53.dll
imlua_capture53.dll
imlua_fftw53.dll
imlua_jp253.dll
imlua_process_omp53.dll
imlua_process53.dll
imlua_wmv53.dll
imlua53.dll
iup.dll
iup_mglplot.dll
iup_plot.dll
iup_pplot.dll
iup_scintilla.dll
iupcd.dll
iupcontrols.dll
iupgl.dll
iupglcontrols.dll
iupim.dll
iupimglib.dll
iuplua_mglplot53.dll
iuplua_plot53.dll
iuplua_pplot53.dll
iuplua_scintilla53.dll
iuplua53.dll
iupluacd53.dll
iupluacontrols53.dll
iupluagl53.dll
iupluaglcontrols53.dll
iupluaim53.dll
iupluaimglib53.dll
iupluamatrixex53.dll
iupluaole53.dll
iupluatuio53.dll
iupluaweb53.dll
iupmatrixex.dll
iupole.dll
iuptuio.dll
iupweb.dll
lfs.dll
libzip-4.dll
lua53.dll
luacom.dll
luaext.dll
luagl.dll
luagl_base.dll
luaglu.dll
luazip.dll
main.lua
msvcp100.dll
msvcr100.dll
pdflib.dll
trace.dll
vcomp100.dll
zlib1.dll

main.lua 修改日期: 2016/6/13 15:08 创建日期: 2016/10/27 14:46
LUA 文件 大小: 0 字节

第三节　新建子窗口

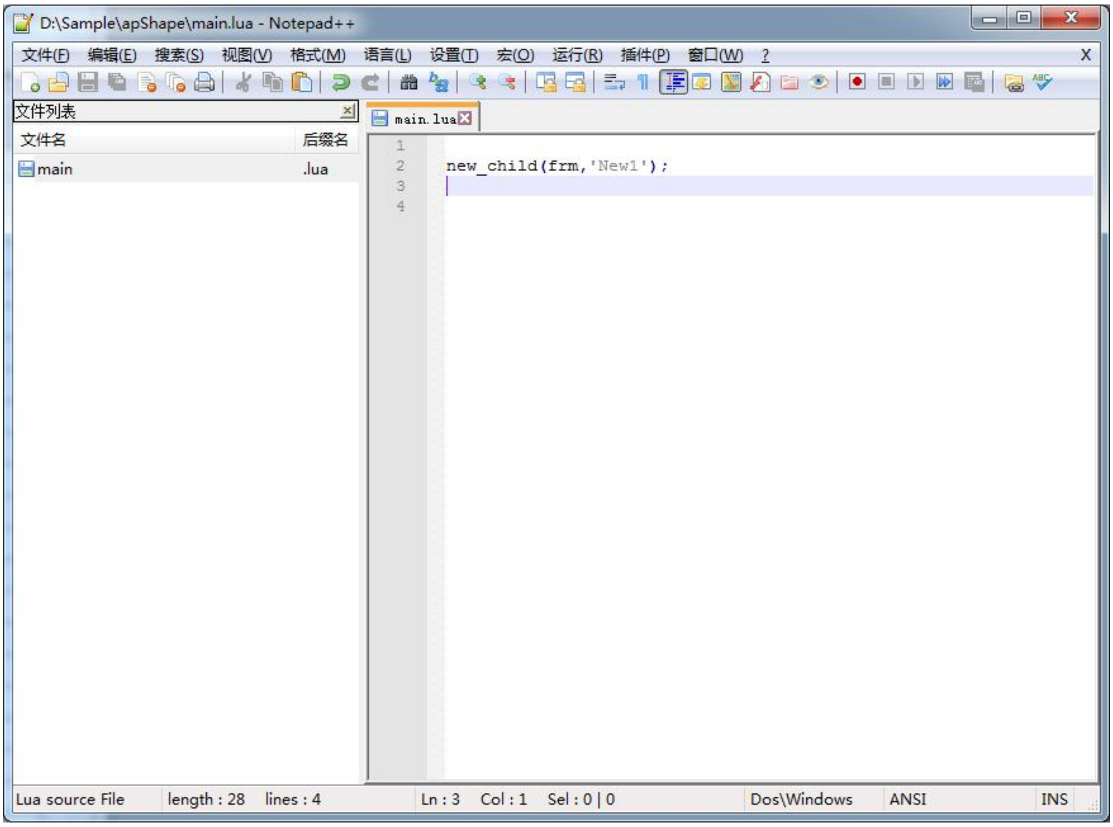
打开主程序文件（main.lua），编写下面代码（图 1）：

```
new_child(frm,"New1");
```

查看源代码（apcad/help/src/primer/3.3.txt ）

这段程序只有一行代码，调用平台 api 函数（new_child）新建一个子窗口，有两个参数，第一个参数是程序主窗口（frm 是平台提供全局变量，可以直接使用），第二个参数是新建的子窗口的标题名称文本（字符串）。

运行主程序（gcad.exe），查看运行结果（图 2）。


D:\Sample\apShape\main.lua - Notepad++
文件(F) 编辑(E) 搜索(S) 视图(V) 格式(M) 语言(L) 设置(T) 宏(O) 运行(R) 插件(P) 窗口(W) ?
文件列表
文件名
后缀名
main
.lua
main.lua
Lua source File
length : 28
lines : 4
Ln : 3
Col : 1
Sel : 0 | 0
Dos\Windows
ANSI
INS

```lua

new_child(frm,'New1');


```

图 1

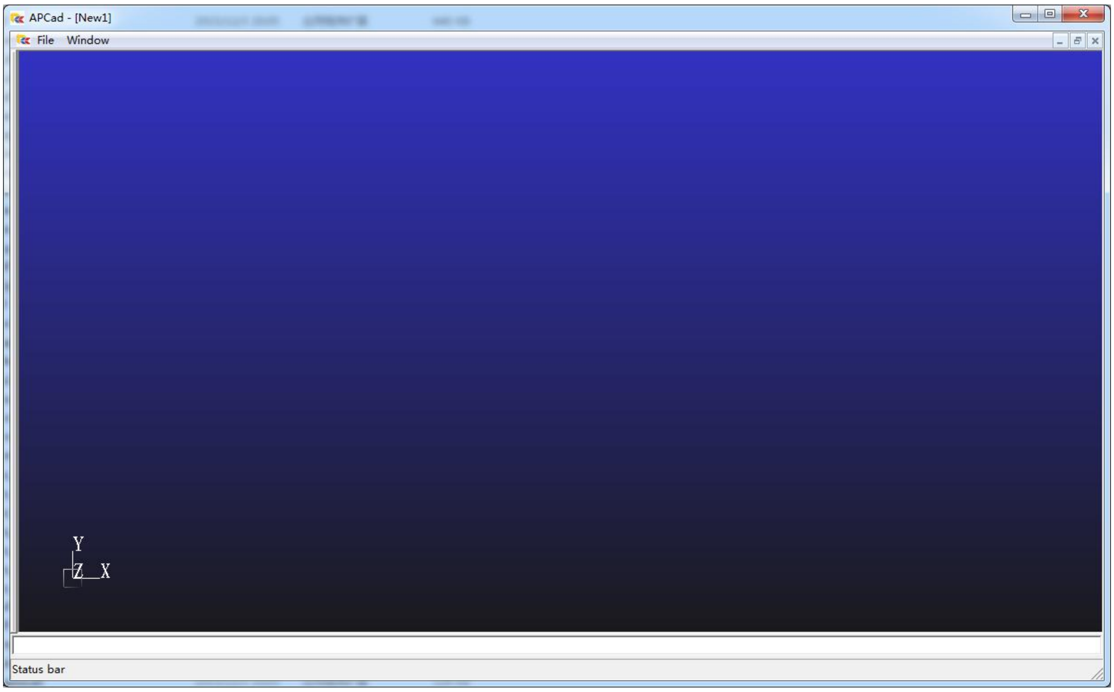


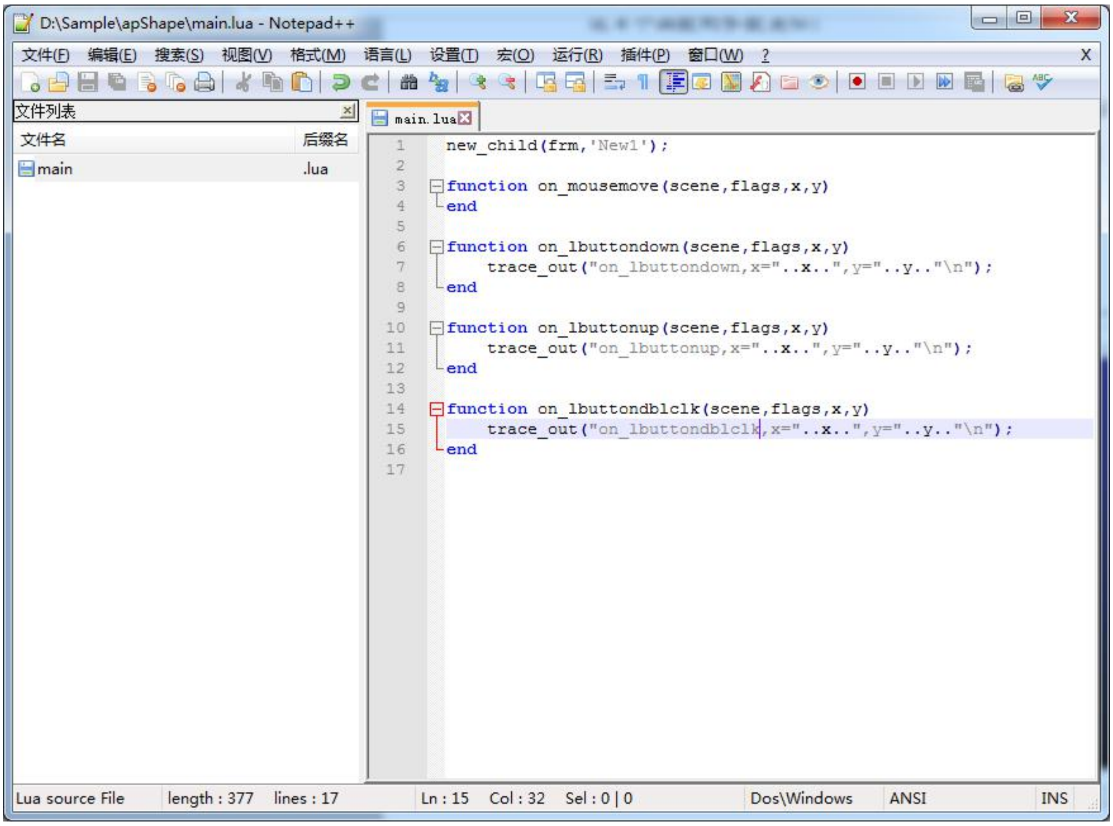
图 2

第四节　响应消息函数

打开主程序文件（main.lua），增加下面代码（图 1）：

```
function on_mousemove(scene,flags,x,y)
end
function on_lbuttondown(scene,flags,x,y)
      trace_out("on_lbuttondown,x="..x..",y="..y.."\n");
end
function on_lbuttonup(scene,flags,x,y)
      trace_out("on_lbuttonup,x="..x..",y="..y.."\n");
end
function on_lbuttondblclk(scene,flags,x,y)
      trace_out("on_lbuttondblclk,x="..x..",y="..y.."\n");
end
```

查看源代码（apcad/help/src/primer/3.4.txt ）


D:\Sample\apShape\main.lua - Notepad++
文件(F) 编辑(E) 搜索(S) 视图(V) 格式(M) 语言(L) 设置(T) 宏(O) 运行(R) 插件(P) 窗口(W) ?
文件列表
文件名 后缀名
main .lua
main.lua
Lua source File length : 377 lines : 17 Ln : 15 Col : 32 Sel : 0 | 0 Dos\Windows ANSI INS

```
new_child(frm,'New1');

function on_mousemove(scene,flags,x,y)
end

function on_lbuttondown(scene,flags,x,y)
    trace_out("on_lbuttondown,x="..x..",y="..y.."\n");
end

function on_lbuttonup(scene,flags,x,y)
    trace_out("on_lbuttonup,x="..x..",y="..y.."\n");
end

function on_lbuttondblclk(scene,flags,x,y)
    trace_out("on_lbuttondblclk,x="..x..",y="..y.."\n");
end
```

这段程序实现了 4 个全局消息函数：

on_mousemove：移动鼠标函数，当鼠标位置发生变化时会被平台调用。

on_lbuttondown：按下鼠标左键函数，当按下鼠标左键时会被平台调用。

on_lbuttonup：放开鼠标左键函数，当放开鼠标左键时会被平台调用。

on_lbuttondblclk：双击鼠标左键函数，当双击鼠标左键时会被平台调用。

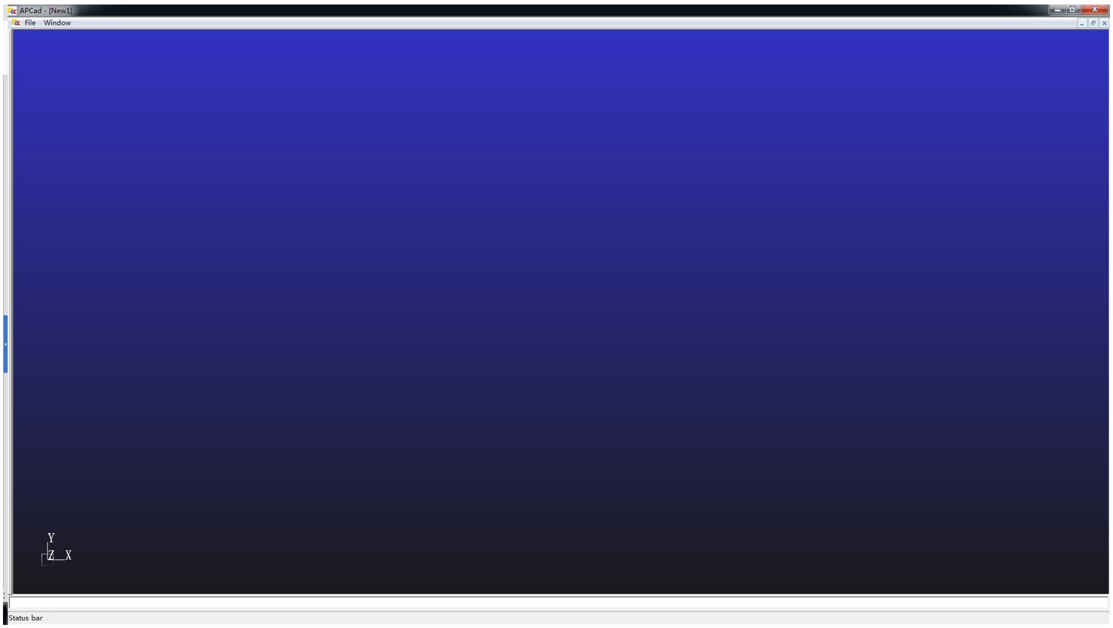
这 4 个函数的参数类似：

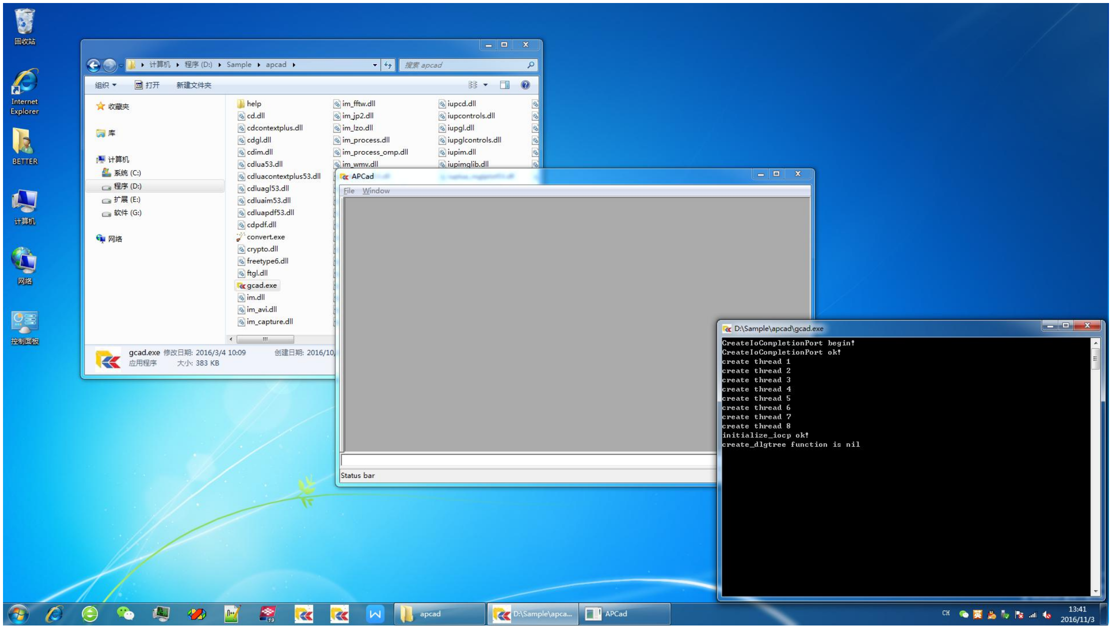
scene：当前子窗口

flags：当前鼠标动作的其他标志（详情请参见《apcad 开发手册》）

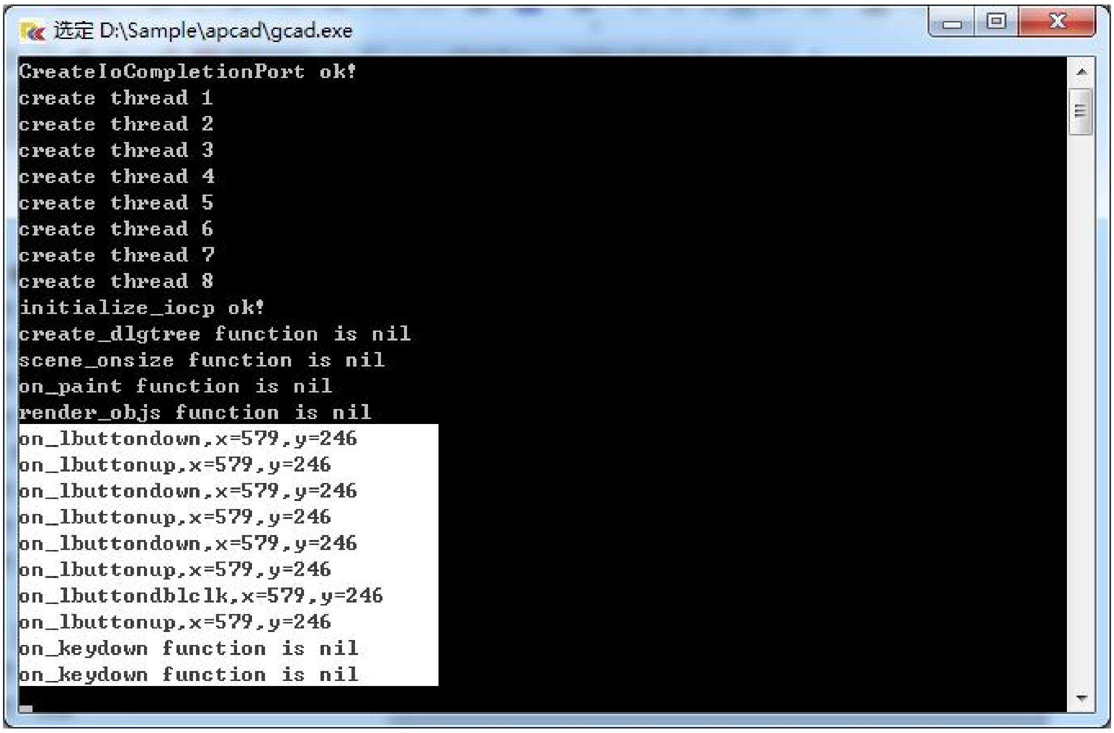
x，y：当前鼠标动作时，光标所在位置的屏幕坐标值

运行主程序（gcad.exe），查看运行结果（在前台主窗口点击几下鼠标左键，然后在后台信息窗口中查看输出信息）。





```
CreateIoCompletionPort begin!
CreateIoCompletionPort ok!
create thread 1
create thread 2
create thread 3
create thread 4
create thread 5
create thread 6
create thread 7
create thread 8
initialize_iocp ok!
create_dlgtree function is nil
```



```
CreateIoCompletionPort ok!
create thread 1
create thread 2
create thread 3
create thread 4
create thread 5
create thread 6
create thread 7
create thread 8
initialize_iocp ok!
create_dlgtree function is nil
scene_onsize function is nil
on_paint function is nil
render_objs function is nil
on_lbuttondown,x=579,y=246
on_lbuttonup,x=579,y=246
on_lbuttondown,x=579,y=246
on_lbuttonup,x=579,y=246
on_lbuttondown,x=579,y=246
on_lbuttonup,x=579,y=246
on_lbuttondblclk,x=579,y=246
on_lbuttonup,x=579,y=246
on_keydown function is nil
on_keydown function is nil
```

随着鼠标的动作，在后台信息窗口中，能够看到打印输出了相应函数内输出的显示信息。

除了鼠标左键，还有鼠标右键、鼠标中键以及键盘、窗口大小尺寸变化、窗口显示、定时器等消息函数，详情请参见《apcad 开发手册》。

第五节　绘制线

在平台窗口上绘制三维物体（三维坐标体系下的线），需要以下几个步骤：

（一）定义一个三维物体（必须符合平台三维图形的数据格式），下面的代码定义了一条三维的线。

```
local shape = {
      surfaces = {
            {
                  points = {
                        {1,1,1,1,1,0,0,0};
                        {1,1,1,1,1,20000,0,0};
                  };
                  lines = {{1,2}};
            };
      };
};
```

查看源代码（apcad/help/src/primer/3.5.1.txt）

一个三维物体的图形，它包含了多个表面的集合（surfaces），每个表面又包含了多个三维点的集合（points）以及多条连接这些点的三维线的集合（lines），每个三维点包含 8 个参数：前三个参数是颜色（r，g，b），后面三个参数是坐标（x，y，z），中间两个参数是贴图位置（详情请参见《apcad 开发手册》），每条三维线包含两个参数是两个点的序号。

变量（shape）是一条三维线，第一个点的坐标的 x，y，z 值都是 0，颜色的 r，g，b 只分别是 1,0,1，第二个点的坐标的 x，y，z 值分别是 20000,0,0，颜色的 r，g，b 只分别是 0,1,1。

注：点颜色由三原色（红，绿，蓝）组成，三原色（r，g，b）数值的取值范围是 0~1 之间，最小是 0，最大是 1，三个值都是 0 是黑色，三个值都是 1 是白色，1,0,0 是红色，0,1,0 是绿色，0,0,1 是蓝色。线的颜色取决于它所连接的点的颜色，一条线连接的两个点，允许每个点的颜色不同，连接不同颜色点的线，颜色在两个点的颜色间渐变。

（二）创建新窗口。

```
local scene = new_child(frm,"New1");
```

查看源代码（apcad/help/src/primer/3.5.2.txt）

变量（scene）是新建的标题为“New1”的子窗口，因为后面的程序需要使用这个窗口，所以比起上一节的程序，这一次定义了变量（scene）。

（三）显示线。

```
local gl = require "luaext.gl"
local k,v = 1,makelist(scene,object);
function render_objs(scene)
    gl.glLoadName(k)
    gl.glCallList(v)
end
```

查看源代码（apcad/help/src/primer/3.5.3.txt）

require 可以引入外部模块，用于调用外部模块提供的接口（函数或者表、类等），为主程序提供了扩展的功能支持。

制作三维物体显示缓存对象（k，v），实现消息函数（render_objs），当子窗口显示图形时该函数会被平台调用，调用函数时显示三维物体的缓存对象。

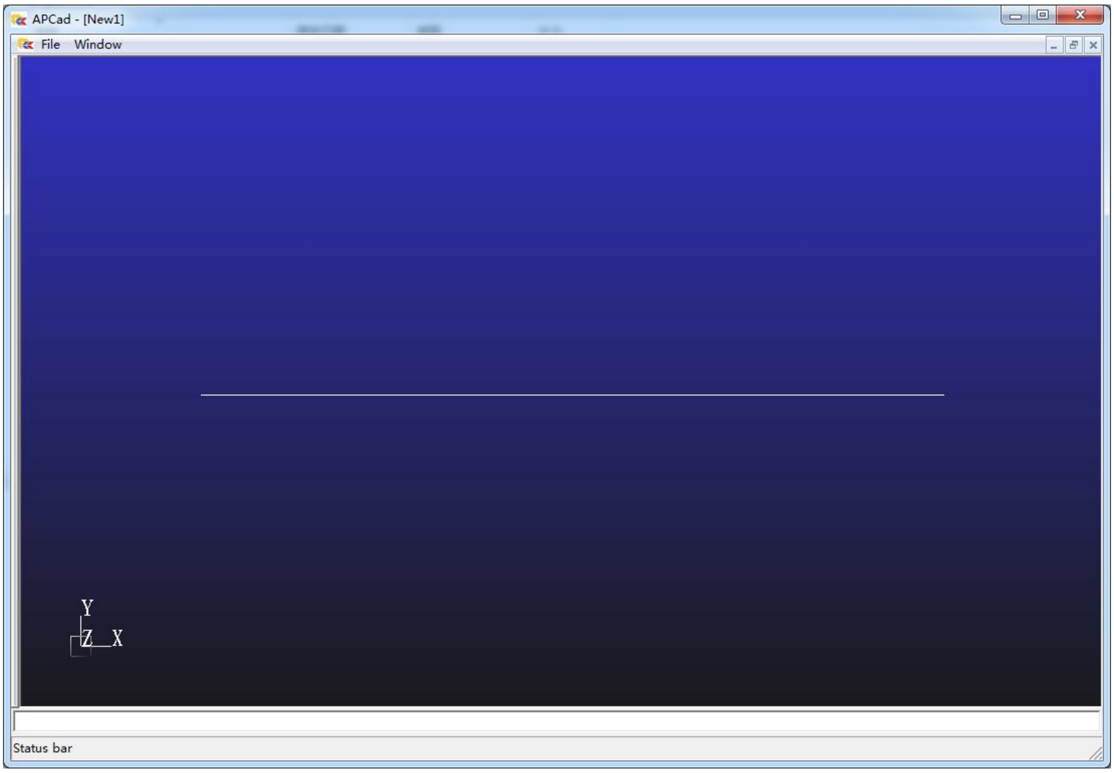
由于消息函数（render_objs）会在每次子窗口显示图形时均被平台调用，所以在该消息函数之外制作缓存对象，以提高子窗口显示图形的效率。

完整的绘制线的程序代码如下：

```
local object = {
    surfaces = {
        {
            points = {
                {1,1,1,1,1,0,0,0};
                {1,1,1,1,1,20000,0,0};
            };
            lines = {{1,2}};
        };
    };
};

local scene = new_child(frm,"New1");
local k,v = 1,makelist(scene,object);
local gl = require "luaext.gl"
function render_objs(scene)
    gl.glLoadName(k)
    gl.glCallList(v)
end
```

查看源代码（apcad/help/src/primer/3.5.4.txt ）

运行主程序（gcad.exe），查看运行结果。



第六节　绘制三维立方体

在平台窗口上绘制一个立方体，同样需要上节的几个步骤，只是在定义一个三维物体时，需要定义立方体而不是线，其它的步骤同上节一样。

```
local pts = {
      {0.5,1,0.5,1,1,0,0,0};
      {0.5,1,0.5,1,1,20000,0,0};
      {0.5,1,0.5,1,1,0,20000,0};
      {0.5,1,0.5,1,1,20000,20000,0};
      {0.5,1,0.5,1,1,0,0,20000};
      {0.5,1,0.5,1,1,20000,0,20000};
      {0.5,1,0.5,1,1,0,20000,20000};
      {0.5,1,0.5,1,1,20000,20000,20000};
};
local shape = {
      surfaces = {
            {
                  points = pts;
                  outer = {1,3,4,2};
            };
            {
                points = pts;
                outer = {5,6,8,7};
            };
            {
                points = pts;
                outer = {1,2,6,5};
            };
            {
                points = pts;
                outer = {1,5,7,3};
            };
            {
                points = pts;
                outer = {2,4,8,6};
            };
            {
                points = pts;
                outer = {3,7,8,4};
            };
        };
    };
```

查看源代码（apcad/help/src/primer/3.6.1.txt ）

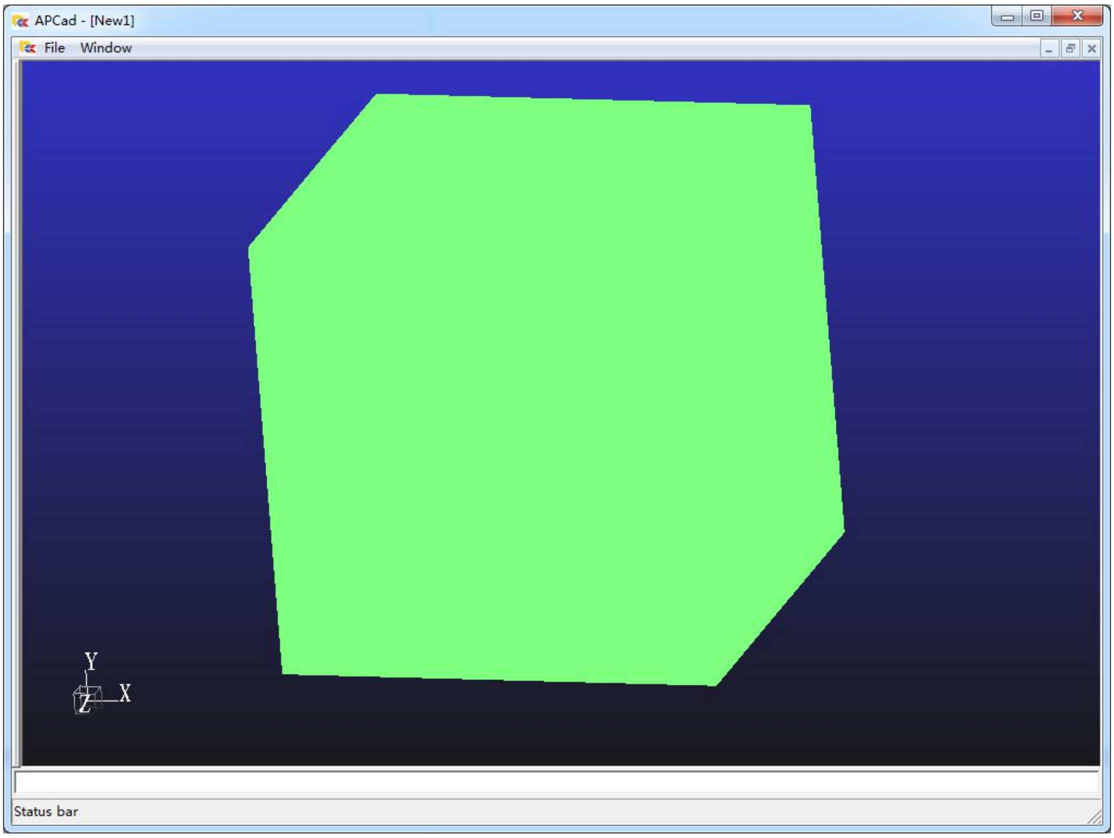
和一条线不同，绘制一个立方体，需要 8 个顶点和 6 个面，所以这段代码定义了一个 8 个顶点的集合（pts），以便于后面每个面（surface）中引用这 8 个点（points = pts），每个表包含一个外轮廓（outer），它顺次连接了需要的点（每个面需要连接 4 个点）。

完整的程序代码如下：

```
local pts = {
    {0.5,1,0.5,1,1,0,0,0};
    {0.5,1,0.5,1,1,20000,0,0};
    {0.5,1,0.5,1,1,0,20000,0};
    {0.5,1,0.5,1,1,20000,20000,0};
    {0.5,1,0.5,1,1,0,0,20000};
    {0.5,1,0.5,1,1,20000,0,20000};
    {0.5,1,0.5,1,1,0,20000,20000};
    {0.5,1,0.5,1,1,20000,20000,20000};
};
local shape = {
    surfaces = {
        {
            points = pts;
            outer = {1,3,4,2};
        };
        {
            points = pts;
            outer = {5,6,8,7};
        };
        {
            points = pts;
            outer = {1,2,6,5};
        };
        {
            points = pts;
            outer = {1,5,7,3};
        };
        {
            points = pts;
            outer = {2,4,8,6};
        };
        {
            points = pts;
            outer = {3,7,8,4};
        };
    };
};

local scene = new_child(frm,"New1");
local k,v = 1,makelist(scene,shape);
local gl = require "luaext.gl"
function render_objs(scene)
    gl.glLoadName(k)
    gl.glCallList(v)
end
```

查看源代码（apcad/help/src/primer/3.6.2.txt ）

运行主程序（gcad.exe），查看运行结果（转动鼠标滚轮可以放缩，按住 Ctrl+鼠标中间，移动鼠标可以旋转）。



等等，天啊，这似乎很难看得出来是一个立方体！

第七节　绘制三维立方体的轮廓线

稍稍改变一下，为这个立方体增加它的轮廓线，定义这个立方体的代码如下：

```
local outer_pts = {
    {0.5,1,0.5,1,1,0,0,0};
    {0.5,1,0.5,1,1,20000,0,0};
    {0.5,1,0.5,1,1,0,20000,0};
    {0.5,1,0.5,1,1,20000,20000,0};
    {0.5,1,0.5,1,1,0,0,20000};
    {0.5,1,0.5,1,1,20000,0,20000};
    {0.5,1,0.5,1,1,0,20000,20000};
    {0.5,1,0.5,1,1,20000,20000,20000};
};
local lines_pts = {
    {0,0,0,1,1,0,0,0};
    {0,0,0,1,1,20000,0,0};
    {0,0,0,1,1,0,20000,0};
    {0,0,0,1,1,20000,20000,0};
    {0,0,0,1,1,0,0,20000};
    {0,0,0,1,1,20000,0,20000};
    {0,0,0,1,1,0,20000,20000};
    {0,0,0,1,1,20000,20000,20000};
};
local shape = {
    surfaces = {
        {
            points = lines_pts;
            lines = {{1,2},{1,3},{2,4},{3,4};{5,6},{5,7},{6,8},{7,8};{1,5},{2,6},{3,7},{4,8}};
        };
        {
            points = outer_pts;
            outer = {1,3,4,2};
        };
         ...
};
```

查看源代码（apcad/help/src/primer/3.7.1.txt ）

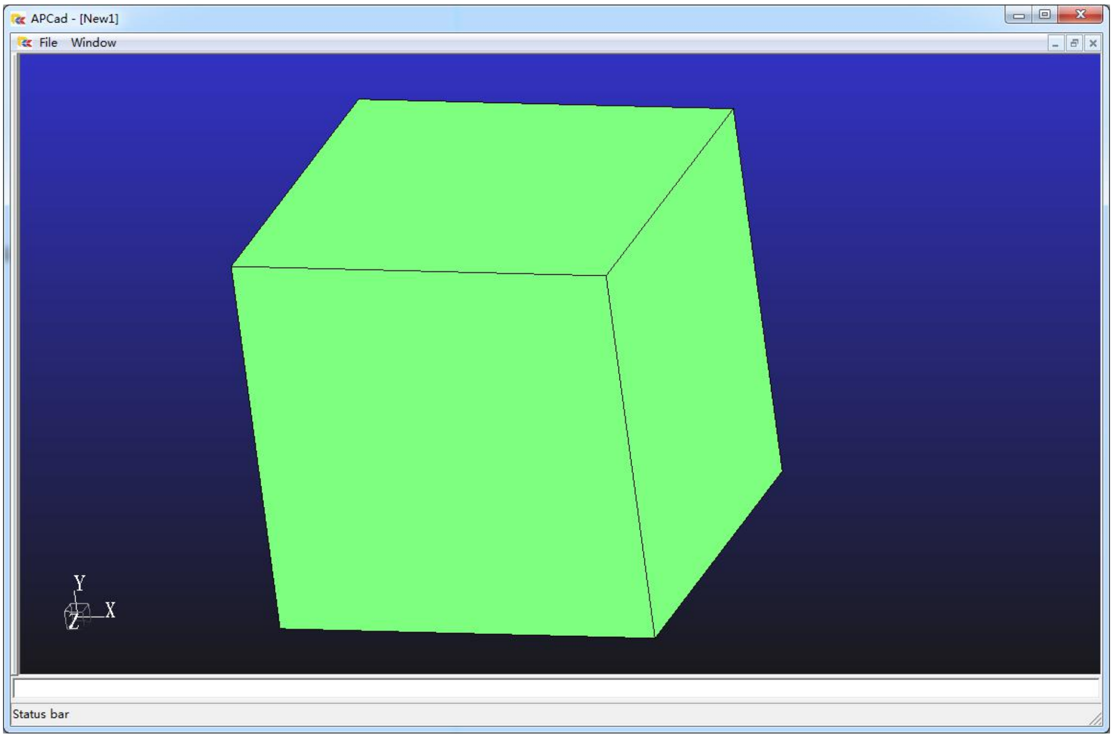
这次，我们为这个立方体增加了轮廓线（黑色），轮廓线的颜色与表面的颜色不同。

完整的程序代码如下：

```
local outer_pts = {
    {0.5,1,0.5,1,1,0,0,0};
    {0.5,1,0.5,1,1,20000,0,0};
    {0.5,1,0.5,1,1,0,20000,0};
    {0.5,1,0.5,1,1,20000,20000,0};
    {0.5,1,0.5,1,1,0,0,20000};
    {0.5,1,0.5,1,1,20000,0,20000};
    {0.5,1,0.5,1,1,0,20000,20000};
    {0.5,1,0.5,1,1,20000,20000,20000};
};
local lines_pts = {
    {0,0,0,1,1,0,0,0};
    {0,0,0,1,1,20000,0,0};
    {0,0,0,1,1,0,20000,0};
    {0,0,0,1,1,20000,20000,0};
    {0,0,0,1,1,0,0,20000};
    {0,0,0,1,1,20000,0,20000};
    {0,0,0,1,1,0,20000,20000};
    {0,0,0,1,1,20000,20000,20000};
};
local shape = {
    surfaces = {
        {
            points = lines_pts;
            lines = {{1,2},{1,3},{2,4},{3,4};{5,6},{5,7},{6,8},{7,8};{1,5},{2,6},{3,7},{4,8}};
        };
        {
            points = outer_pts;
            outer = {1,3,4,2};
        };
        {
            points = outer_pts;
            outer = {5,6,8,7};
        };
        {
            points = outer_pts;
            outer = {1,2,6,5};
        };
        {
            points = outer_pts;
            outer = {1,5,7,3};
        };
        {
            points = outer_pts;
            outer = {2,4,8,6};
        };
        {
            points = outer_pts;
            outer = {3,7,8,4};
        };
    };
};
```

查看源代码（apcad/help/src/primer/3.7.2.txt ）

运行主程序（gcad.exe），查看运行结果（转动鼠标滚轮可以放缩，按住 Ctrl+鼠标中间，移动鼠标可以旋转）。



第八节　弹出立方体属性对话框窗口

在平台窗口上绘制了立方体之后，鼠标左键双击窗口的任意位置，弹出属性对话框，可以显示并修改立方体的大小、位置和颜色等属性，这需要以下几个步骤（详情请参见《apcad 开发手册》）：

（一）定义参数变量控制立方体的大小、位置和颜色属性，对于一个横平竖直的标准六面体，斜对角的两个点的坐标即可控制其大小和位置，六个表面的颜色由变量（r,g,b）控制，轮廓线依然固定为黑色（0,0,0）。

```
local x1,y1,z1 = 0,0,0;
local x2,y2,z2 = 20000,20000,20000;
local r,g,b = 0.5,1,0.5;
local outer_pts = {
    {r,g,b,1,1,x1,y1,z1};
    {r,g,b,1,1,x2,y1,z1};
    {r,g,b,1,1,x1,y2,z1};
    {r,g,b,1,1,x2,y2,z1};
    {r,g,b,1,1,x1,y1,z2};
    {r,g,b,1,1,x2,y1,z2};
    {r,g,b,1,1,x1,y2,z2};
    {r,g,b,1,1,x2,y2,z2};
};
local lines_pts = {
    {0,0,0,1,1,x1,y1,z1};
    {0,0,0,1,1,x2,y1,z1};
    {0,0,0,1,1,x1,y2,z1};
    {0,0,0,1,1,x2,y2,z1};
    {0,0,0,1,1,x1,y1,z2};
    {0,0,0,1,1,x2,y1,z2};
    {0,0,0,1,1,x1,y2,z2};
    {0,0,0,1,1,x2,y2,z2};
};
```

查看源代码（apcad/help/src/primer/3.8.1.txt ）

（二）定义一个属性对话框。

```
package.cpath = "./?53.dll;./?.dll";
local iup = require"iuplua"
local pt1_lab = iup.label{title="Point1:"};
local pt1_x_lab = iup.label{title="X:"};
local pt1_x_txt = iup.text{expand="Yes"};
local pt1_y_lab = iup.label{title="Y:"};
local pt1_y_txt = iup.text{expand="Yes"};
local pt1_z_lab = iup.label{title="Z:"};
local pt1_z_txt = iup.text{expand="Yes"};
local pt2_lab = iup.label{title="Point2:"};
local pt2_x_lab = iup.label{title="X:"};
local pt2_x_txt = iup.text{expand="Yes"};
local pt2_y_lab = iup.label{title="Y:"};
local pt2_y_txt = iup.text{expand="Yes"};
local pt2_z_lab = iup.label{title="Z:"};
local pt2_z_txt = iup.text{expand="Yes"};
local color_lab = iup.label{title="Color:"};
local color_r_lab = iup.label{title="R:"};
local color_g_lab = iup.label{title="G:"};
local color_b_lab = iup.label{title="B:"};
local ok_btn = iup.button{title="OK",size="100X"};
local cancel_btn = iup.button{title="Cancel",size="100X"};
local dlg = iup.dialog{
    title = "Property";
    size = "500X100";
    margin = "5X5";
    iup.vbox{
        iup.hbox{pt1_lab,pt1_x_lab,pt1_x_txt,pt1_y_lab,pt1_y_txt,pt1_z_lab,pt1_z_txt};
        iup.hbox{pt2_lab,pt2_x_lab,pt2_x_txt,pt2_y_lab,pt2_y_txt,pt2_z_lab,pt2_z_txt};
        iup.hbox{color_lab,color_r_lab,color_r_txt,color_g_lab,color_g_txt,
                  color_b_lab,color_b_txt};
        iup.hbox{iup.fill{},ok_btn,cancel_btn};
    };
}
```

查看源代码（apcad/help/src/primer/3.8.2.txt）

同绘制线时类似，这次又一次引入（require）了一个外部模块（iuplua），它提供了用于制作对话框的扩展功能接口：

标题（title）：对话框窗口的标题显示文本（“Property”）。

尺寸（size）：对话框窗口的大小（“500X200”），即宽度 500，高度 200。

间距（margin）：对话框窗口内部各个控件之间的间距，即宽度间距 5，高度间距也是 5。

垂直排版（vbox）：指定了所包含的各个组成部分竖向垂直分布。

水平排版（hbox）：指定了所包含的各个组成部分横向水平分布。

这段代码定义了一个对话框变量（dlg），它的标题是“Property”，大小尺寸是宽度 500，高度 200，内部的各个控件间距横向竖向都是 5，第一行是第一点的信息（x,y,z），第二行是第二代的信息（x,y,z），第三行是颜色信息（r,g,b），最后一行放置了两个按钮（“OK”，“Cancel”），iup.fill{}用来占位，是的两个按钮在最后一行靠右侧对齐。

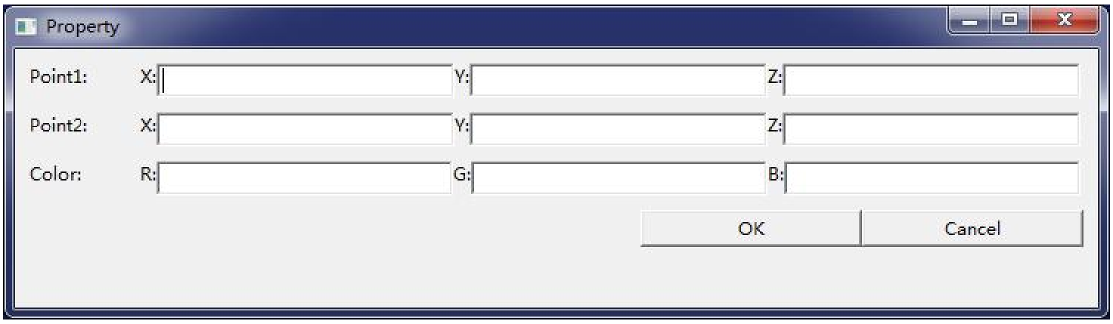
（三）双击鼠标左键弹出属性对话框。

定义对话框弹出接口函数，现实弹出对话框，定义鼠标左键双击消息函数，在函数中调用对话框弹出接口函数。

```
function show_dlg()
    dlg:popup();
end
function on_lbuttondblclk(scene,flags,x,y)
    show_dlg();
end
```

查看源代码（apcad/help/src/primer/3.8.3.txt）

*注：package.cpath = "./?53.dll;./?.dll"，定义了引入外部扩展模块时的默认配置，详情请参见《apcad 开发手册》。



（四）初始化属性对话框。

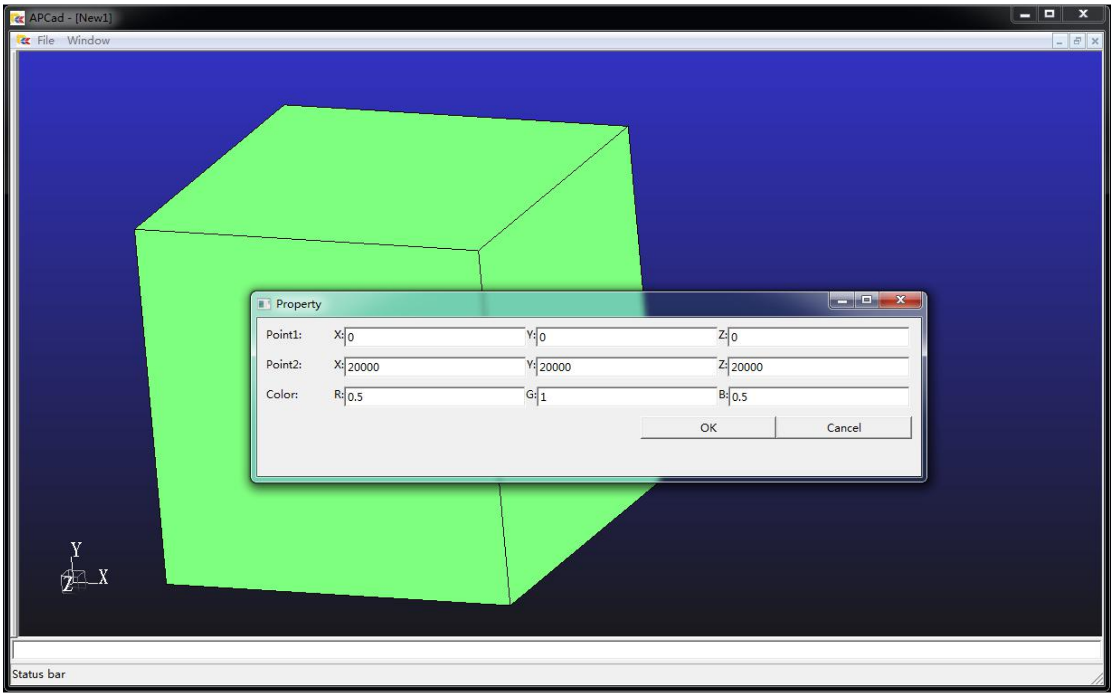
在属性对话框上面的文本栏里显示控制立方体大小、位置和颜色的控制变量的数值，可以用于查看当前立方体的属性。

```
function init_dlg()
    pt1_x_txt.value = x1;
    pt1_y_txt.value = y1;
    pt1_z_txt.value = z1;
    pt2_x_txt.value = x2;
    pt2_y_txt.value = y2;
    pt2_z_txt.value = z2;
    color_r_txt.value = r;
    color_g_txt.value = g;
    color_b_txt.value = b;
end
function show_dlg()
    init_dlg();
    dlg:popup();
end
```

查看源代码（apcad/help/src/primer/3.8.4.txt）

定义初始化对话框的函数，在弹出对话框的同时，调用初始化对话框函数，用相关的控制变量填充文本栏的数值。

运行主程序（gcad.exe），查看运行结果（双击鼠标左键弹出对话框）。



Point1: X:0 Y:0 Z:0

Point2: X:20000 Y:20000 Z:20000

Color: R:0.5 G:1 B:0.5

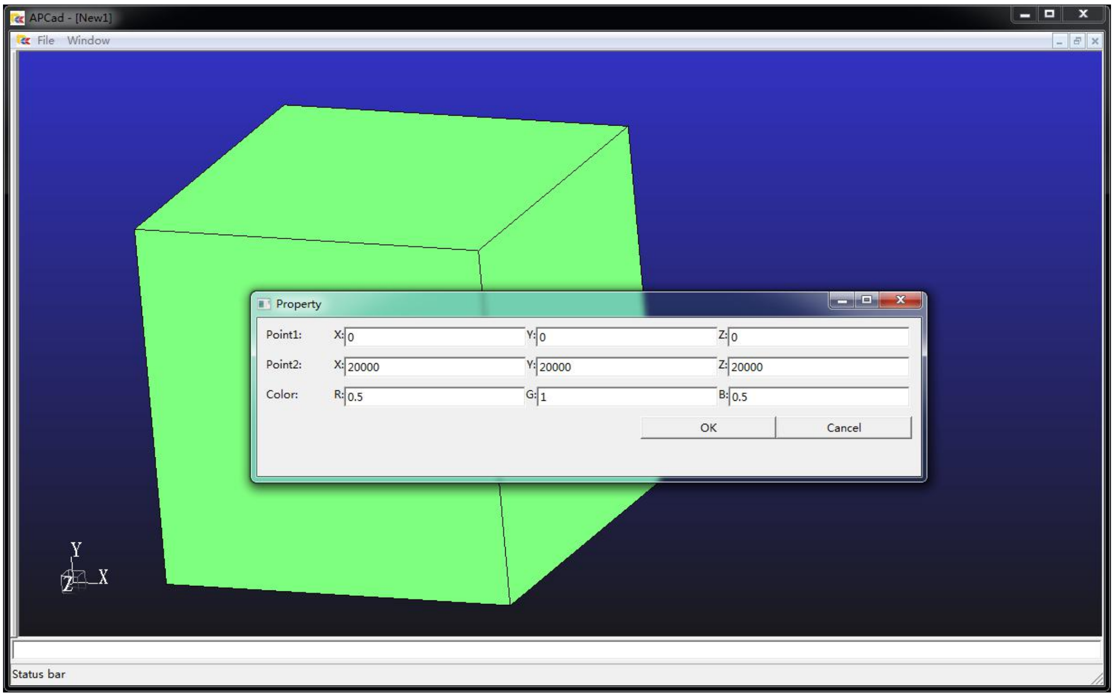
（五）修改属性。

```
function ok_btn:action()
    x1 = pt1_x_txt.value;
    y1 = pt1_y_txt.value;
    z1 = pt1_z_txt.value;
    x2 = pt2_x_txt.value;
    y2 = pt2_y_txt.value;
    z2 = pt2_z_txt.value;
    r = color_r_txt.value;
    g = color_g_txt.value;
    b = color_b_txt.value;
    dlg:hide();
end
function cancel_btn:action()
    dlg:hide();
end
```

查看源代码（apcad/help/src/primer/3.8.5.txt ）

定义两个按钮（“OK”，“Cancel”）的消息函数，在 OK 按钮函数中取出文本栏里的数值（用户可能会修改了这个数值），同时关闭对话框；在 Cancel 按钮的消息函数中，直接关闭对话框。

运行主程序（gcad.exe），查看运行结果（双击鼠标左键弹出对话框，试着修改其中的某些数值，点击 OK 按钮后，再次双击弹出对话框查看属性）。



Point1: X: 0 Y: 0 Z: 0

Point2: X: 20000 Y: 20000 Z: 20000

Color: R: 0.5 G: 1 B: 0.5

等等，属性变了，为什么立方体没有随着属性变化而变化呢？

（六）重新绘制立方体。

```
local x1,y1,z1 = 0,0,0;
local x2,y2,z2 = 20000,20000,20000;
local r,g,b = 0.5,1,0.5;
function get_shape()
    local outer_pts = {
        {r,g,b,1,1,x1,y1,z1};
        {r,g,b,1,1,x2,y1,z1};
        {r,g,b,1,1,x1,y2,z1};
        {r,g,b,1,1,x2,y2,z1};
        {r,g,b,1,1,x1,y1,z2};
        {r,g,b,1,1,x2,y1,z2};
        {r,g,b,1,1,x1,y2,z2};
        {r,g,b,1,1,x2,y2,z2};
    };
    local lines_pts = {
        {0,0,0,1,1,x1,y1,z1};
        {0,0,0,1,1,x2,y1,z1};
        {0,0,0,1,1,x1,y2,z1};
        {0,0,0,1,1,x2,y2,z1};
        {0,0,0,1,1,x1,y1,z2};
        {0,0,0,1,1,x2,y1,z2};
        {0,0,0,1,1,x1,y2,z2};
        {0,0,0,1,1,x2,y2,z2};
    };
```

```
	local shape = {
		surfaces = {
			{
				points = lines_pts;
				lines = {{1,2},{1,3},{2,4},{3,4};{5,6},{5,7},{6,8},{7,8};{1,5},{2,6},{3,7},{4,8}};
			};
			{
				points = outer_pts;
				outer = {1,3,4,2};
			};
			{
				points = outer_pts;
				outer = {5,6,8,7};
			};
			{
				points = outer_pts;
				outer = {1,2,6,5};
			};
			{
				points = outer_pts;
				outer = {1,5,7,3};
			};
			{
				points = outer_pts;
				outer = {2,4,8,6};
			};
			{
				points = outer_pts;
				outer = {3,7,8,4};
			};
		};
	};
	return shape;
end
```

查看源代码（apcad/help/src/primer/3.8.6.txt ）

把计算立方体的代码写成函数，可以在需要的时候重复调用，以使立方体的形状、颜色随属性变化。

```
local scene = new_child(frm,"New1");
local k,v = 1,makelist(scene,get_shape(scene));
local gl = require "luaext.gl"
function render_objs()
	gl.glLoadName(k)
```

```
    gl.glCallList(v)
end
```

查看源代码（apcad/help/src/primer/3.8.7.txt）

使用立方体制作缓存并显示。

```
function ok_btn:action()
    x1 = pt1_x_txt.value;
    y1 = pt1_y_txt.value;
    z1 = pt1_z_txt.value;
    x2 = pt2_x_txt.value;
    y2 = pt2_y_txt.value;
    z2 = pt2_z_txt.value;
    r = color_r_txt.value;
    g = color_g_txt.value;
    b = color_b_txt.value;
    k,v = 1,makelist(scene,get_shape(scene));
    scene_onpaint(scene);
    dlg:hide();
end
```

查看源代码（apcad/help/src/primer/3.8.8.txt）

在 OK 按钮的消息函数中重新制作缓存。

完整的代码如下：

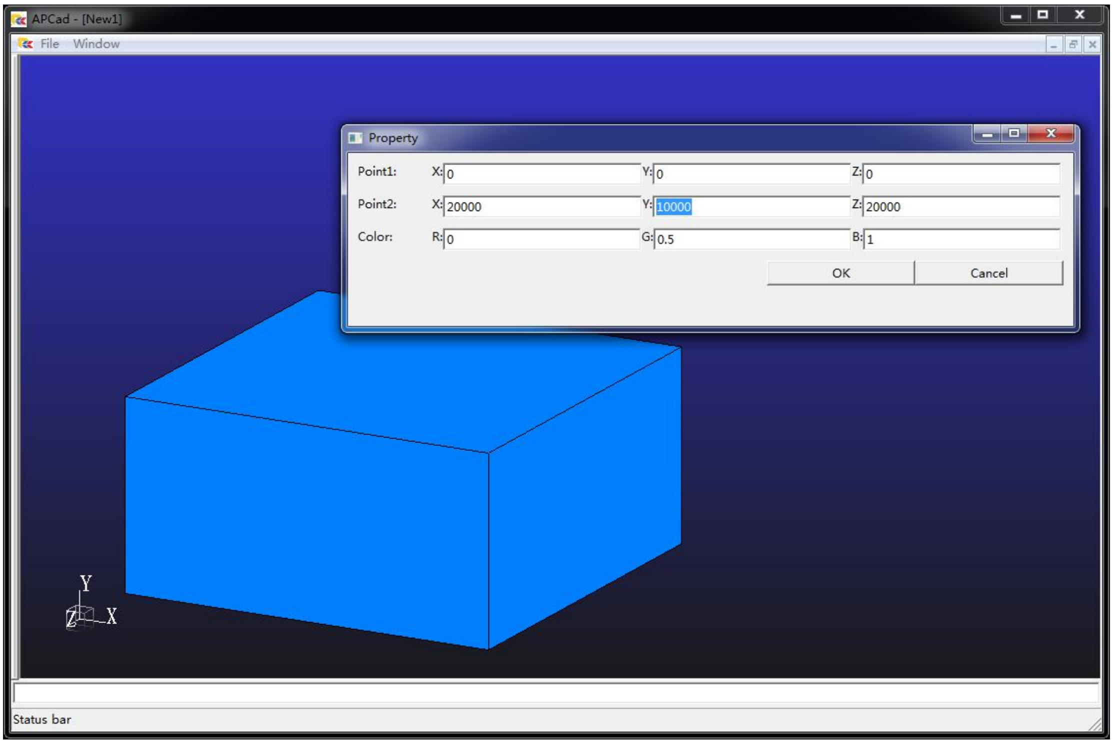
```
package.cpath = "./?53.dll;./?.dll";

local x1,y1,z1 = 0,0,0;
local x2,y2,z2 = 20000,20000,20000;
local r,g,b = 0.5,1,0.5;
function get_object()
    local outer_pts = {
        {r,g,b,1,1,x1,y1,z1};
        {r,g,b,1,1,x2,y1,z1};
        {r,g,b,1,1,x1,y2,z1};
        {r,g,b,1,1,x2,y2,z1};
        {r,g,b,1,1,x1,y1,z2};
        {r,g,b,1,1,x2,y1,z2};
        {r,g,b,1,1,x1,y2,z2};
        {r,g,b,1,1,x2,y2,z2};
    };
    local lines_pts = {
        {0,0,0,1,1,x1,y1,z1};
        {0,0,0,1,1,x2,y1,z1};
        {0,0,0,1,1,x1,y2,z1};
        {0,0,0,1,1,x2,y2,z1};
        {0,0,0,1,1,x1,y1,z2};
        {0,0,0,1,1,x2,y1,z2};
        {0,0,0,1,1,x1,y2,z2};
        {0,0,0,1,1,x2,y2,z2};
    };
    local object = {
        surfaces = {
            {
                points = lines_pts;
                lines = {{1,2},{1,3},{2,4},{3,4};{5,6},{5,7},{6,8},{7,8};{1,5},{2,6},{3,7},{4,8}};
            };
            {
                points = outer_pts;
                outer = {1,3,4,2};
            };
            {
                points = outer_pts;
                outer = {5,6,8,7};
            };
            {
                points = outer_pts;
                outer = {1,2,6,5};
            };
            {
                points = outer_pts;
                outer = {1,5,7,3};
            };
            {
                points = outer_pts;
                outer = {2,4,8,6};
            };
            {
                points = outer_pts;
                outer = {3,7,8,4};
            };
        };
    };
    return object;
end
local scene = new_child(frm,"New1");
local k,v = 1,makelist(scene,get_object(scene));
local gl = require "luaext.gl"
function render_objs()
```

```
        gl.glLoadName(k)
        gl.glCallList(v)
end

local iup = require"iuplua"
local pt1_lab = iup.label{title="Point1:",size="50x"};
local pt1_x_lab = iup.label{title="X:"};
local pt1_x_txt = iup.text{expand="Horizontal"};
local pt1_y_lab = iup.label{title="Y:"};
local pt1_y_txt = iup.text{expand="Horizontal"};
local pt1_z_lab = iup.label{title="Z:"};
local pt1_z_txt = iup.text{expand="Horizontal"};
local pt2_lab = iup.label{title="Point2:",size="50X"};
local pt2_x_lab = iup.label{title="X:"};
local pt2_x_txt = iup.text{expand="Horizontal"};
local pt2_y_lab = iup.label{title="Y:"};
local pt2_y_txt = iup.text{expand="Horizontal"};
local pt2_z_lab = iup.label{title="Z:"};
local pt2_z_txt = iup.text{expand="Horizontal"};
local color_lab = iup.label{title="Color:",size="50X"};
local color_r_lab = iup.label{title="R:"};
local color_r_txt = iup.text{expand="Horizontal"};
local color_g_lab = iup.label{title="G:"};
local color_g_txt = iup.text{expand="Horizontal"};
local color_b_lab = iup.label{title="B:"};
local color_b_txt = iup.text{expand="Horizontal"};
local ok_btn = iup.button{title="OK",size="100X"};
local cancel_btn = iup.button{title="Cancel",size="100X"};
local dlg = iup.dialog{
        title = "Property";
        size = "500X100";
        margin = "5X5";
        iup.vbox{
                iup.hbox{pt1_lab,pt1_x_lab,pt1_x_txt,pt1_y_lab,pt1_y_txt,pt1_z_lab,pt1_z_txt};
                iup.hbox{pt2_lab,pt2_x_lab,pt2_x_txt,pt2_y_lab,pt2_y_txt,pt2_z_lab,pt2_z_txt};

        iup.hbox{color_lab,color_r_lab,color_r_txt,color_g_lab,color_g_txt,color_b_lab,color_b
_txt};
                iup.hbox{iup.fill{},ok_btn,cancel_btn};
        };
}
function init_dlg()
        pt1_x_txt.value = x1;
        pt1_y_txt.value = y1;
```

```
    pt1_z_txt.value = z1;
    pt2_x_txt.value = x2;
    pt2_y_txt.value = y2;
    pt2_z_txt.value = z2;
    color_r_txt.value = r;
    color_g_txt.value = g;
    color_b_txt.value = b;
end
function ok_btn:action()
    x1 = pt1_x_txt.value;
    y1 = pt1_y_txt.value;
    z1 = pt1_z_txt.value;
    x2 = pt2_x_txt.value;
    y2 = pt2_y_txt.value;
    z2 = pt2_z_txt.value;
    r = color_r_txt.value;
    g = color_g_txt.value;
    b = color_b_txt.value;
    k,v = 1,makelist(scene,get_object(scene));
    scene_onpaint(scene);
    dlg:hide();
end
function cancel_btn:action()
    dlg:hide();
end
function show_dlg()
    init_dlg();
    dlg:popup();
end
function on_lbuttondblclk(scene,flags,x,y)
    show_dlg();
end
```

查看源代码（apcad/help/src/primer/3.8.9.txt ）

运行主程序（gcad.exe），查看运行结果（双击鼠标左键弹出对话框，试着修改其中的某些数值，点击 OK 按钮后，再次双击弹出对话框查看属性）。



Point1: X: 0 Y: 0 Z: 0

Point2: X: 20000 Y: 10000 Z: 20000

Color: R: 0 G: 0.5 B: 1

第九节　菜单栏

双击鼠标左键可以弹出属性对话框，同时，也可以制作菜单项，通过点击菜单弹出属性对话框。

制作菜单，并响应菜单消息执行菜单命令，需要以下几个步骤：

（一）定义菜单项 ID。

```
local ID_PROPERTY = ID+1;
```

查看源代码（apcad/help/src/primer/3.9.1.txt）

（二）添加菜单项到菜单栏。

```
add_menu(
    frm,
    {
        name = "Cube",
        nposition = 2,
        items =
        {
            {id=ID_PROPERTY,name="Property"},
        },
    }
);
```

查看源代码（apcad/help/src/primer/3.9.2.txt）

这段代码制作了一个主菜单项（Cube），并在该菜单项下添加了一个字菜单项（Property），制作子菜单项时，如果同名主菜单项已经存在，则子菜单项直接添加到原有的同名主菜单项下。

（三）相应菜单消息并执行命令。

```
function on_command(id,scene)
    if id==ID_PROPERTY then
        show_dlg();
    end
end
```

查看源代码（apcad/help/src/primer/3.9.3.txt）

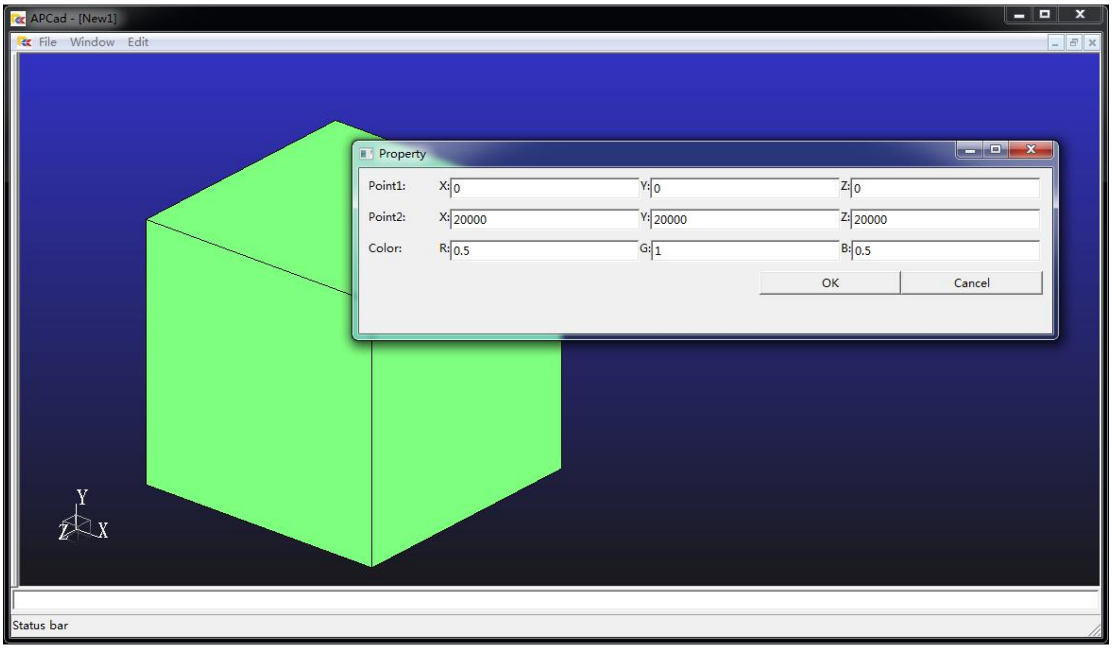
定义了平台消息函数（on_command），会在用户点击菜单项后被平台调用，两个参数分别是用户点击的菜单ID和当前窗口（scene），判断id确定用户点击的菜单，调用弹出属性对话框函数（show_dlg）。

完整的菜单程序代码如下：

```
local ID_PROPERTY = ID+1;
add_menu(
    frm,
    {
        name = "Cube",
        nposition = 2,
        items =
        {
            {id=ID_PROPERTY,name="Property"},
        },
    }
);
function on_command(id,scene)
    if id==ID_PROPERTY then
        show_dlg();
    end
end
```

查看源代码（apcad/help/src/primer/3.9.4.txt）

运行主程序（gcad.exe），查看运行结果（点击菜单Cube-Property弹出对话框查看属性）。



Point1: X: 0 Y: 0 Z: 0

Point2: X: 20000 Y: 20000 Z: 20000

Color: R: 0.5 G: 1 B: 0.5

OK Cancel

第十节　工具条按钮

制作工具条按钮，并响应按钮消息执行菜单命令，需要以下几个步骤：

（一）定义工具条按钮 ID。

local ID_PROPERTY = ID+1;

查看源代码（apcad/help/src/primer/3.9.1.txt）

同菜单一样，需要定义 ID，如果这个按钮执行的是某个已经存在的菜单命令，可以直接使用上面菜单定义过的 ID，而不必重新定义。

（二）准备工具条图片。

准备工具条图片（BMP），高度是 16 像素，宽度是 16 的整数倍像素，每个 16X16 部分是一个按钮显示的图标。

（三）制作工具条。

```
crt_toolbar(frm,
    {
        bmpname = "toolbar1.bmp",
        nbmps = 3,
        dxButton = 0,
        dyButton = 0,
        dxBitmap = 16,
        dyBitmap = 16,
        buttons = {
                {iBitmap=5,idCommand=ID_PROPERTY,iString="Property",
                    fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
            },
        }
);
```

查看源代码（apcad/help/src/primer/3.10.1.txt）

iBitmap=4，指定了该按钮图标从图片的16X4的位置开始选取，idCommand指定了按钮的ID（这里使用了上节菜单定义过的ID），iString制定了鼠标悬停在该按钮图标上时的提示信息。

（四）相应菜单消息并执行命令。

```
function on_command(id,scene)
    if id==ID_PROPERTY then
        show_dlg();
    end
end
```

查看源代码（apcad/help/src/primer/3.9.3.txt）

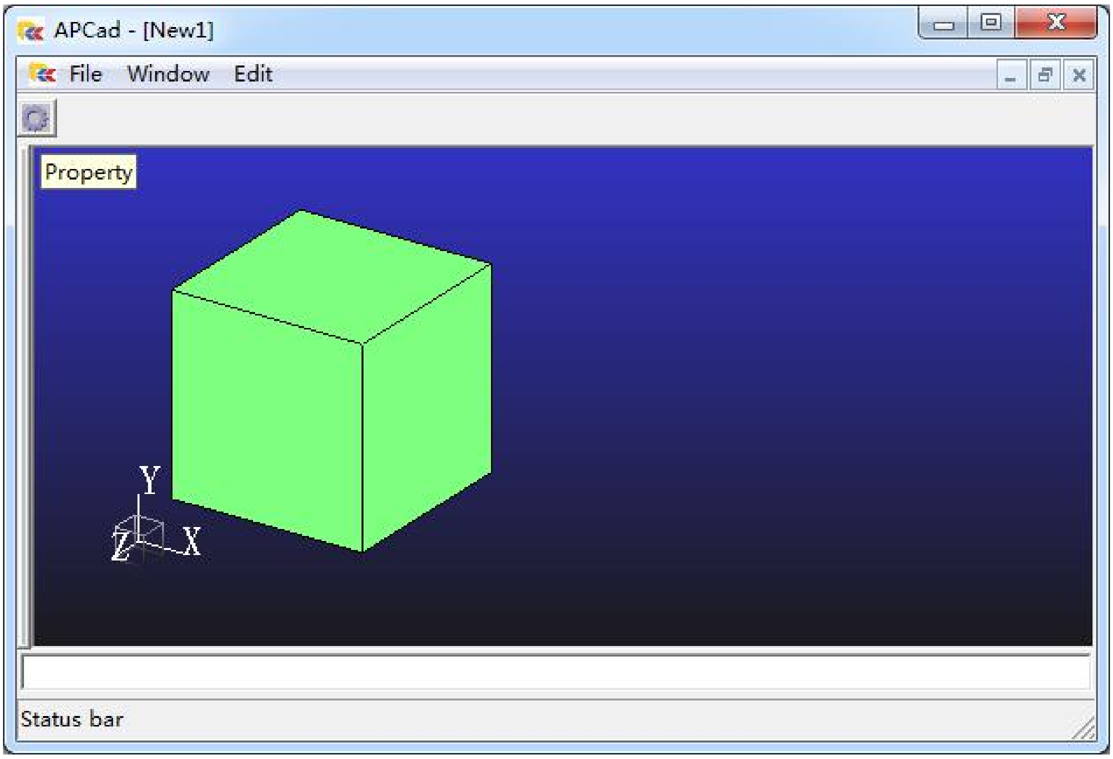
同菜单一样，需要定义ID，如果这个按钮执行的是某个已经存在的菜单命令，可以省略该步骤，直接执行菜单。

完整的代码如下：

```
local ID_PROPERTY = ID+1;
add_menu(
    frm,
    {
        name = "Cube",
        nposition = 2,
        items =
        {
            {id=ID_PROPERTY,name="Property"},
        },
    }
);
crt_toolbar(frm,
    {
        bmpname = "toolbar1.bmp",
        nbmps = 3,
        dxButton = 0,
        dyButton = 0,
        dxBitmap = 16,
        dyBitmap = 16,
            buttons = {
                {iBitmap=4,idCommand=ID_PROPERTY,iString="Property",
                    fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
            },
        }
);
function on_command(id,scene)
    if id==ID_PROPERTY then
        show_dlg();
    end
end
```

查看源代码（apcad/help/src/primer/3.10.2.txt ）

运行主程序（gcad.exe），查看运行结果（点击工具条按钮）。



第十一节　添加多个三维立方体

上面的程序，只能绘制一个立方体，需要绘制多个立方体，需要以下几个步骤：

（一）定义一个集合（shapes）记录多个立方体的属性，并提供函数（add_shape）增加新的立方体图块。

```
local objects = {};
```

```
function add_object()
    local n = #objects;
    local pt1 = {x=5000*n,y=5000*n,z=5000*n};
    local pt2 = {x=5000*n+3000,y=5000*n+3000,z=5000*n+3000};
    local color = {r=0,g=0.5,b=1};
    local shape = get_object(pt1,pt2,color);
    local glname,gllist = n+1,makelist(scene,shape);
    objects[n+1] = {pt1=pt1,pt2=pt2,color=color,glname=glname,gllist=gllist};
    scene_onpaint(scene);
end
```

查看源代码（apcad/help/src/primer/3.11.1.txt ）

（二）同时，计算图形数据的方法（get_object）也需要根据实际属性（pt1,pt2,color）的值计算所对应的图形。

```
function get_shape(pt1,pt2,color)
    local outer_pts = {
        {color.r,color.g,color.b,1,1,pt1.x,pt1.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt1.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt2.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt2.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt1.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt1.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt2.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt2.y,pt2.z};
    };
    local lines_pts = {
        {0,0,0,1,1,pt1.x,pt1.y,pt1.z};
        {0,0,0,1,1,pt2.x,pt1.y,pt1.z};
        {0,0,0,1,1,pt1.x,pt2.y,pt1.z};
        {0,0,0,1,1,pt2.x,pt2.y,pt1.z};
        {0,0,0,1,1,pt1.x,pt1.y,pt2.z};
        {0,0,0,1,1,pt2.x,pt1.y,pt2.z};
        {0,0,0,1,1,pt1.x,pt2.y,pt2.z};
        {0,0,0,1,1,pt2.x,pt2.y,pt2.z};
    };
    local shape = {
        surfaces = {
            {
                points = lines_pts;
                lines = {{1,2},{1,3},{2,4},{3,4};{5,6},{5,7},{6,8},{7,8};{1,5},{2,6},{3,7},{4,8}};
            };
            {
                points = outer_pts;
                outer = {1,3,4,2};
            };
            {
                points = outer_pts;
                outer = {5,6,8,7};
            };
            {
                points = outer_pts;
                outer = {1,2,6,5};
            };
            {
                points = outer_pts;
                outer = {1,5,7,3};
            };
            {
                points = outer_pts;
                outer = {2,4,8,6};
            };
            {
                points = outer_pts;
                outer = {3,7,8,4};
            };
        };
    };
    return shape;
end
```

查看源代码（apcad/help/src/primer/3.11.2.txt）

（三）重新定义消息函数（render_objs）。

```
function render_objs()
    for i,v in ipairs(objects) do
        gl.glLoadName(v.glname);
        gl.glCallList(v.gllist);
    end
end
```

查看源代码（apcad/help/src/primer/3.11.3.txt）

（四）增加新的菜单项和工具条按钮（Add）。

```
local ID_ADD = ID+2;
add_menu(
    frm,
    {
        name = "Cube",
        nposition = 2,
        items =
        {
            {id=ID_ADD,name="Add"},
            {id=ID_PROPERTY,name="Property"},
        },
    }
);
crt_toolbar(frm,
    {
        bmpname = "toolbar1.bmp",
        nbmps = 3,
        dxButton = 0,
        dyButton = 0,
        dxBitmap = 16,
        dyBitmap = 16,
        buttons = {
            {iBitmap=2,idCommand=ID_ADD,iString="Add",
                fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
            {iBitmap=5,idCommand=ID_PROPERTY,iString="Property",
                fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
        },
    }
);
```

查看源代码（apcad/help/src/primer/3.11.4.txt ）

（五）响应新菜单项的命令。

```
local commands = {};
commands[ID_PROPERTY] = function(scene)
    show_dlg();
end
commands[ID_ADD] = function(scene)
    add_object();
end
function on_command(id,scene)
    if type(commands[id])=="function" then
        commands[id](scene);
    end
end
```

查看源代码（apcad/help/src/primer/3.11.5.txt ）

需要在 on_command 消息函数中处理多个不同 ID 对应的不同命令，一种方法是上面用过的方法，用 if 语句判断 id 的值，调用不同的命令函数，现在使用

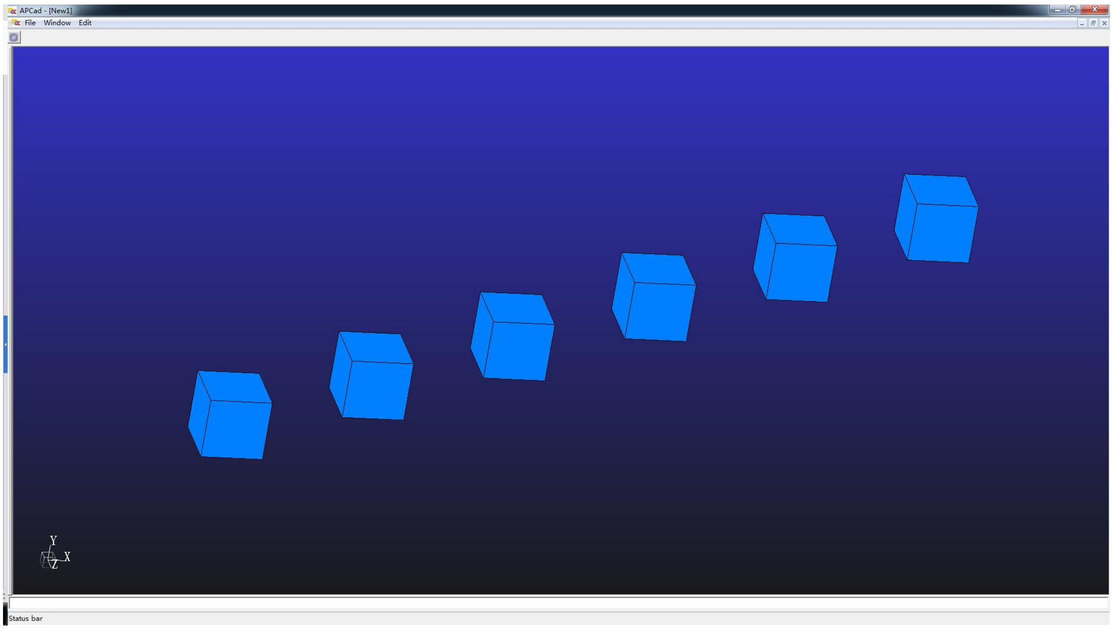
的是另一种方法，定义了一个命令函数表，用 id 作为表索引，调用相对应的命令函数。

完整的源代码如下：

```
package.cpath = "./?53.dll;./?.dll";

function frmclose()
    os.exit();
end
function on_mousemove()
end
function on_paint()
end

function get_shape(pt1,pt2,color)
    local outer_pts = {
        {color.r,color.g,color.b,1,1,pt1.x,pt1.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt1.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt2.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt2.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt1.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt1.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt2.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt2.y,pt2.z};
    };
    local lines_pts = {
        {0,0,0,1,1,pt1.x,pt1.y,pt1.z};
        {0,0,0,1,1,pt2.x,pt1.y,pt1.z};
        {0,0,0,1,1,pt1.x,pt2.y,pt1.z};
        {0,0,0,1,1,pt2.x,pt2.y,pt1.z};
        {0,0,0,1,1,pt1.x,pt1.y,pt2.z};
        {0,0,0,1,1,pt2.x,pt1.y,pt2.z};
        {0,0,0,1,1,pt1.x,pt2.y,pt2.z};
        {0,0,0,1,1,pt2.x,pt2.y,pt2.z};
    };
    local shape = {
        surfaces = {
            {
                points = lines_pts;
                lines = {{1,2},{1,3},{2,4},{3,4};{5,6},{5,7},{6,8},{7,8};{1,5},{2,6},{3,7},{4,8}};
            };
            {
                points = outer_pts;
                outer = {1,3,4,2};
                };
                {
                        points = outer_pts;
                        outer = {5,6,8,7};
                };
                {
                        points = outer_pts;
                        outer = {1,2,6,5};
                };
                {
                        points = outer_pts;
                        outer = {1,5,7,3};
                };
                {
                        points = outer_pts;
                        outer = {2,4,8,6};
                };
                {
                        points = outer_pts;
                        outer = {3,7,8,4};
                };
        };
    };
    return shape;
end

local scene = new_child(frm,"New1");
local objects = {};
function add_object()
    local n = #objects;
    local pt1 = {x=5000*n,y=5000*n,z=5000*n};
    local pt2 = {x=5000*n+3000,y=5000*n+3000,z=5000*n+3000};
    local color = {r=0,g=0.5,b=1};
    local shape = get_shape(pt1,pt2,color);
    local glname,gllist = n+1,makelist(scene,shape);
    objects[n+1] = {pt1=pt1,pt2=pt2,color=color,glname=glname,gllist=gllist};
    scene_onpaint(scene);
end

local gl = require "luaext.gl"
function render_objs()
    for i,v in ipairs(objects) do
        gl.glLoadName(v.glname);
        gl.glCallList(v.gllist);
        end
end

local iup = require"iuplua"
local pt1_lab = iup.label{title="Point1:",size="50x"};
local pt1_x_lab = iup.label{title="X:"};
local pt1_x_txt = iup.text{expand="Horizontal"};
local pt1_y_lab = iup.label{title="Y:"};
local pt1_y_txt = iup.text{expand="Horizontal"};
local pt1_z_lab = iup.label{title="Z:"};
local pt1_z_txt = iup.text{expand="Horizontal"};
local pt2_lab = iup.label{title="Point2:",size="50X"};
local pt2_x_lab = iup.label{title="X:"};
local pt2_x_txt = iup.text{expand="Horizontal"};
local pt2_y_lab = iup.label{title="Y:"};
local pt2_y_txt = iup.text{expand="Horizontal"};
local pt2_z_lab = iup.label{title="Z:"};
local pt2_z_txt = iup.text{expand="Horizontal"};
local color_lab = iup.label{title="Color:",size="50X"};
local color_r_lab = iup.label{title="R:"};
local color_r_txt = iup.text{expand="Horizontal"};
local color_g_lab = iup.label{title="G:"};
local color_g_txt = iup.text{expand="Horizontal"};
local color_b_lab = iup.label{title="B:"};
local color_b_txt = iup.text{expand="Horizontal"};
local ok_btn = iup.button{title="OK",size="100X"};
local cancel_btn = iup.button{title="Cancel",size="100X"};
local dlg = iup.dialog{
        title = "Property";
        size = "500X100";
        margin = "5X5";
        iup.vbox{
                iup.hbox{pt1_lab,pt1_x_lab,pt1_x_txt,pt1_y_lab,pt1_y_txt,pt1_z_lab,pt1_z_txt};
                iup.hbox{pt2_lab,pt2_x_lab,pt2_x_txt,pt2_y_lab,pt2_y_txt,pt2_z_lab,pt2_z_txt};
                iup.hbox{color_lab,color_r_lab,color_r_txt,color_g_lab,color_g_txt,
                                        color_b_lab,color_b_txt};
                iup.hbox{iup.fill{},ok_btn,cancel_btn};
        };
}
function init_dlg()
        pt1_x_txt.value = x1;
        pt1_y_txt.value = y1;
        pt1_z_txt.value = z1;
        pt2_x_txt.value = x2;
        pt2_y_txt.value = y2;
        pt2_z_txt.value = z2;
        color_r_txt.value = r;
        color_g_txt.value = g;
        color_b_txt.value = b;
end
function ok_btn:action()
        x1 = pt1_x_txt.value;
        y1 = pt1_y_txt.value;
        z1 = pt1_z_txt.value;
        x2 = pt2_x_txt.value;
        y2 = pt2_y_txt.value;
        z2 = pt2_z_txt.value;
        r = color_r_txt.value;
        g = color_g_txt.value;
        b = color_b_txt.value;
        k,v = 1,makelist(scene,get_shape());
        scene_onpaint(scene);
        dlg:hide();
end
function cancel_btn:action()
        dlg:hide();
end
function show_dlg()
        init_dlg();
        dlg:popup();
end
function on_lbuttondblclk(scene,flags,x,y)
        show_dlg();
end

local ID_PROPERTY = ID+1;
local ID_ADD = ID+2;
add_menu(
        frm,
        {
                name = "Cube",
                nposition = 2,
                items =
                {
                        {id=ID_ADD,name="Add"},
                        {id=ID_PROPERTY,name="Property"},
                },
        }
);
crt_toolbar(frm,
    {
        bmpname = "toolbar1.bmp",
        nbmps = 3,
        dxButton = 0,
        dyButton = 0,
        dxBitmap = 16,
        dyBitmap = 16,
        buttons = {
            {iBitmap=2,idCommand=ID_ADD,iString="Add",
                fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
            {iBitmap=5,idCommand=ID_PROPERTY,iString="Property",
                fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
        },
    }
);
local commands = {};
commands[ID_PROPERTY] = function(scene)
    show_dlg();
end
commands[ID_ADD] = function(scene)
    add_object();
end
function on_command(id,scene)
    if type(commands[id])=="function" then
        commands[id](scene);
    end
end
```

查看源代码（apcad/help/src/primer/3.11.6.txt ）

运行主程序（gcad.exe），查看运行结果（点击菜单 Cube-Add 增加新的立方体）。



等等（怎么又等等），刚刚制作的对话框好像不好用了呢！

第十二节　选择一个立方体

原本制作的对话框，直接修改立方体的属性，现在有了多个立方体，修改的是哪一个立方体的属性呢？这时，可能就需要用户在多个立方体中做出选择了。

（一）定义平台消息函数（on_lbuttondown），在鼠标左键按下时，根据鼠标点的屏幕坐标（x，y），调用平台提供的接口函数（scene_select），选择鼠标左键点击的立方体。

```
function on_lbuttondown(scene,flags,x,y)
    scene_select(scene,x,y,1,1,1);
end
```

查看源代码（apcad/help/src/primer/3.12.1.txt ）

（二）定义平台消息函数（select_main），当有立方体被选择时会被平台调用，参数是该立方体的序号，调用选择函数修（select_shape）改这个立方体的颜色。

```
function select_main(i)
    select_object(i)
end
```

查看源代码（apcad/help/src/primer/3.12.2.txt ）

（三）定义选择立方的函数

```
function select_object(i)
	objects[i].color.r = (objects[i].color.r-0.5)%1.5;
	objects[i].color.g = (objects[i].color.g-0.5)%1.5;
	objects[i].color.b = (objects[i].color.b-0.5)%1.5;
	local shape = get_shape(objects[i].pt1,objects[i].pt2,objects[i].color);
	local glname,gllist = i,makelist(scene,shape);
	objects[i].gllist = gllist;
	scene_onpaint(scene);
end
```

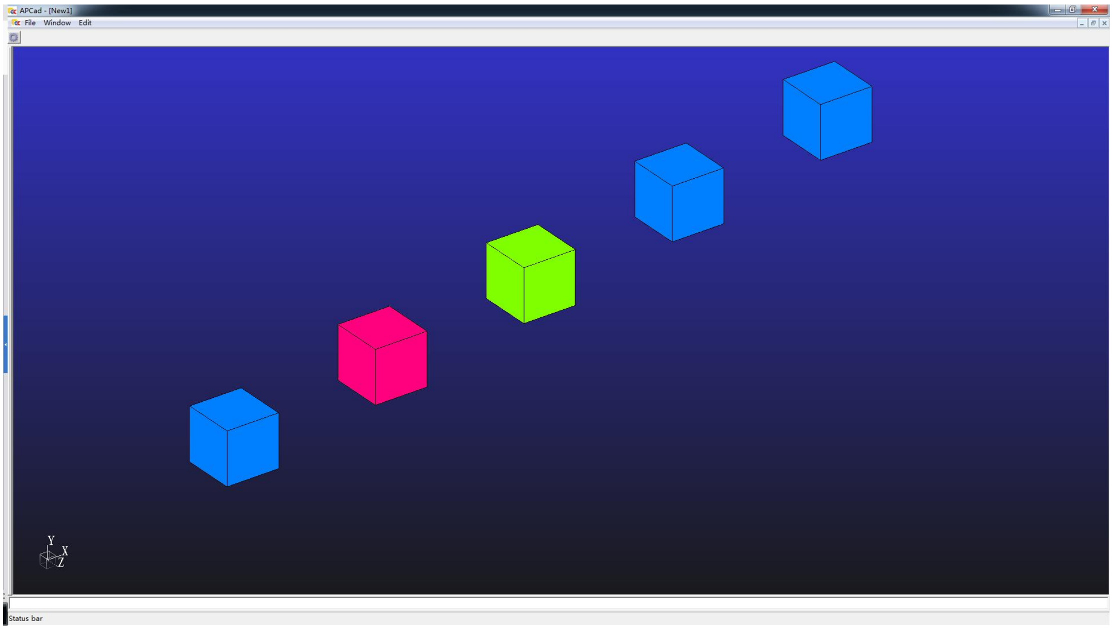
查看源代码（apcad/help/src/primer/3.12.3.txt）

完整的代码如下：

```
function select_object(i)
	objects[i].color.r = (objects[i].color.r-0.5)%1.5;
	objects[i].color.g = (objects[i].color.g-0.5)%1.5;
	objects[i].color.b = (objects[i].color.b-0.5)%1.5;
	local shape = get_shape(objects[i].pt1,objects[i].pt2,objects[i].color);
	local glname,gllist = i,makelist(scene,shape);
	objects[i].gllist = gllist;
	scene_onpaint(scene);
end
function select_main(i)
	select_object(i)
end
function on_lbuttondown(scene,flags,x,y)
	scene_select(scene,x,y,1,1,1);
end
```

查看源代码（apcad/help/src/primer/3.12.4.txt）

运行主程序（gcad.exe），查看运行结果（新建几个立方体，点击其中的一个）。



可是，对话框的问题还没有解决哦！

第十三节　继续属性对话框

（一）定义一个变量，记录最后选择的立方体。

```
local selected = nil;
function select_object(i)
    objects[i].color.r = (objects[i].color.r-0.5)%1.5;
    objects[i].color.g = (objects[i].color.g-0.5)%1.5;
    objects[i].color.b = (objects[i].color.b-0.5)%1.5;
    local shape = get_shape(objects[i].pt1,objects[i].pt2,objects[i].color);
    local glname,gllist = i,makelist(scene,shape);
    objects[i].gllist = gllist;
    scene_onpaint(scene);
    selected = i;
end
```

查看源代码（apcad/help/src/primer/3.13.1.txt ）

（二）设置对话框上颜色文本栏为只读（readonly="Yes"），并设置颜色为灰色（192 192 192）。

```
local color_lab = iup.label{title="Color:",size="50X"};
local color_r_lab = iup.label{title="R:"};
local color_r_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local color_g_lab = iup.label{title="G:"};
local color_g_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local color_b_lab = iup.label{title="B:"};
local color_b_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
```

查看源代码（apcad/help/src/primer/3.13.2.txt ）

（三）用最后选择的立方体的属性（包括颜色）初始化对话框。

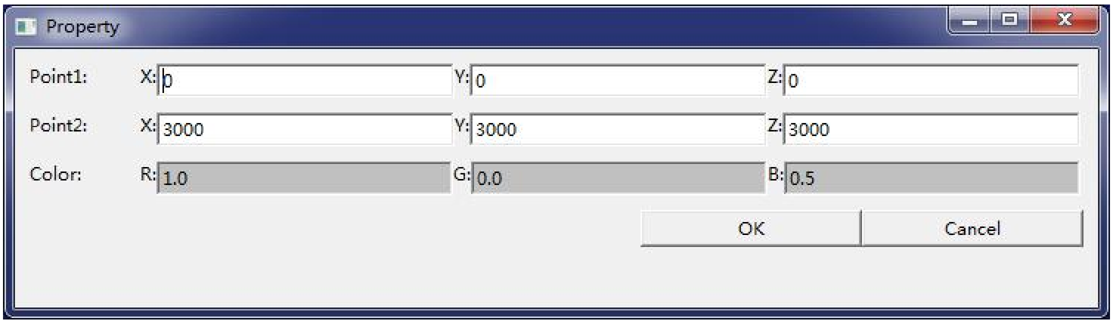
```
function init_dlg()
    if type(selected)~="number" or selected<=0 or selected>#objects then return end
    pt1_x_txt.value = objects[selected].pt1.x;
    pt1_y_txt.value = objects[selected].pt1.y;
    pt1_z_txt.value = objects[selected].pt1.z;
    pt2_x_txt.value = objects[selected].pt2.x;
    pt2_y_txt.value = objects[selected].pt2.y;
    pt2_z_txt.value = objects[selected].pt2.z;
    color_r_txt.value = objects[selected].color.r;
    color_g_txt.value = objects[selected].color.g;
    color_b_txt.value = objects[selected].color.b;
end
```

查看源代码（apcad/help/src/primer/3.13.3.txt ）

（四）根据对话框上面的属性数值修改最后选择的立方体的属性（忽略颜色）。

```
function ok_btn:action()
    if type(selected)~="number" or selected<=0 or selected>#objects then return end
    objects[selected].pt1.x = pt1_x_txt.value;
    objects[selected].pt1.y = pt1_y_txt.value;
    objects[selected].pt1.z = pt1_z_txt.value;
    objects[selected].pt2.x = pt2_x_txt.value;
    objects[selected].pt2.y = pt2_y_txt.value;
    objects[selected].pt2.z = pt2_z_txt.value;
    select_object(selected)
    scene_onpaint(scene);
    dlg:hide();
end
```

查看源代码（apcad/help/src/primer/3.13.4.txt ）



Point1: X: 0 Y: 0 Z: 0

Point2: X: 3000 Y: 3000 Z: 3000

Color: R: 1.0 G: 0.0 B: 0.5

完整的代码如下：

```
package.cpath = "./?53.dll;./?.dll";

function frmclose()
    os.exit();
end
function on_mousemove()
end
function on_paint()
end

function get_shape(pt1,pt2,color)
    local outer_pts = {
        {color.r,color.g,color.b,1,1,pt1.x,pt1.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt1.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt2.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt2.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt1.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt1.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt2.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt2.y,pt2.z};
    };
    local lines_pts = {
        {0,0,0,1,1,pt1.x,pt1.y,pt1.z};
        {0,0,0,1,1,pt2.x,pt1.y,pt1.z};
        {0,0,0,1,1,pt1.x,pt2.y,pt1.z};
        {0,0,0,1,1,pt2.x,pt2.y,pt1.z};
        {0,0,0,1,1,pt1.x,pt1.y,pt2.z};
        {0,0,0,1,1,pt2.x,pt1.y,pt2.z};
        {0,0,0,1,1,pt1.x,pt2.y,pt2.z};
        {0,0,0,1,1,pt2.x,pt2.y,pt2.z};
    };
    local shape = {
        surfaces = {
            {
                points = lines_pts;
                lines = {{1,2},{1,3},{2,4},{3,4};{5,6},{5,7},{6,8},{7,8};{1,5},{2,6},{3,7},{4,8}};
            };
            {
                points = outer_pts;
                outer = {1,3,4,2};
            };
            {
                points = outer_pts;
                outer = {5,6,8,7};
            };
            {
                points = outer_pts;
                outer = {1,2,6,5};
            };
            {
                points = outer_pts;
                outer = {1,5,7,3};
            };
            {
                points = outer_pts;
                outer = {2,4,8,6};
            };
            {
                points = outer_pts;
                outer = {3,7,8,4};
            };
        };
    };
    return shape;
end

local scene = new_child(frm,"New1");
local objects = {};
function add_object()
    local n = #objects;
    local pt1 = {x=5000*n,y=5000*n,z=5000*n};
    local pt2 = {x=5000*n+3000,y=5000*n+3000,z=5000*n+3000};
    local color = {r=0,g=0.5,b=1};
    local shape = get_shape(pt1,pt2,color);
    local glname,gllist = n+1,makelist(scene,shape);
    objects[n+1] = {pt1=pt1,pt2=pt2,color=color,glname=glname,gllist=gllist};
    scene_onpaint(scene);
end

local selected = nil;
function select_object(i)
    objects[i].color.r = (objects[i].color.r-0.5)%1.5;
    objects[i].color.g = (objects[i].color.g-0.5)%1.5;
    objects[i].color.b = (objects[i].color.b-0.5)%1.5;
    local shape = get_shape(objects[i].pt1,objects[i].pt2,objects[i].color);
    local glname,gllist = i,makelist(scene,shape);
    objects[i].gllist = gllist;
    scene_onpaint(scene);
```

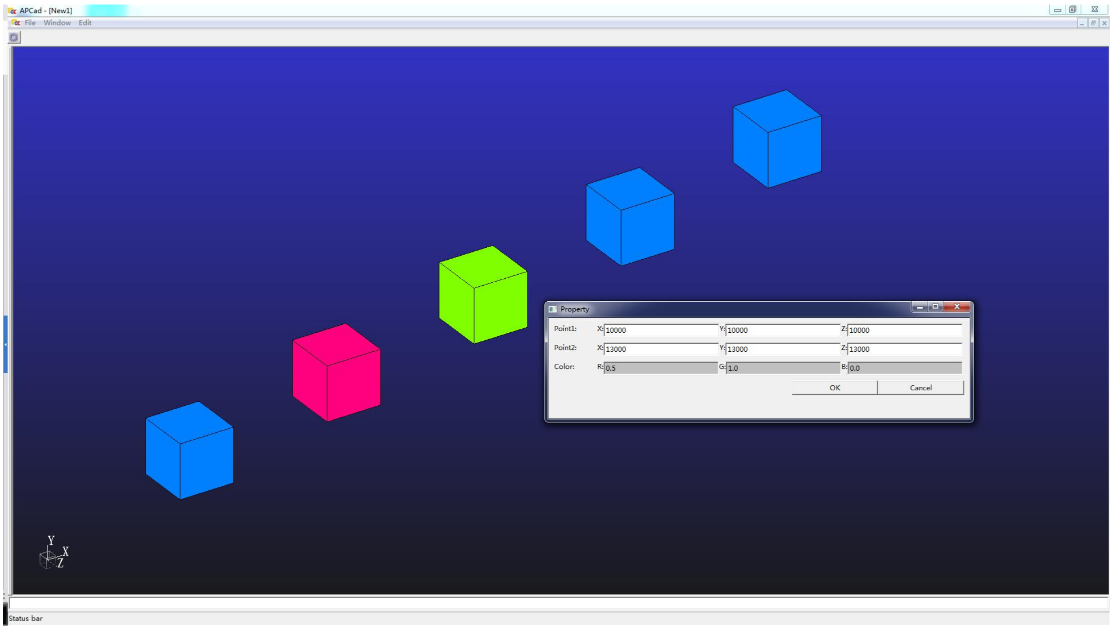
```
	selected = i;
end
function select_main(i)
	select_object(i)
end
function on_lbuttondown(scene,flags,x,y)
	scene_select(scene,x,y,1,1,1);
end

local gl = require "luaext.gl"
function render_objs()
	for i,v in ipairs(objects) do
		gl.glLoadName(v.glname);
		gl.glCallList(v.gllist);
	end
end

local iup = require"iuplua"
local pt1_lab = iup.label{title="Point1:",size="50x"};
local pt1_x_lab = iup.label{title="X:"};
local pt1_x_txt = iup.text{expand="Horizontal"};
local pt1_y_lab = iup.label{title="Y:"};
local pt1_y_txt = iup.text{expand="Horizontal"};
local pt1_z_lab = iup.label{title="Z:"};
local pt1_z_txt = iup.text{expand="Horizontal"};
local pt2_lab = iup.label{title="Point2:",size="50X"};
local pt2_x_lab = iup.label{title="X:"};
local pt2_x_txt = iup.text{expand="Horizontal"};
local pt2_y_lab = iup.label{title="Y:"};
local pt2_y_txt = iup.text{expand="Horizontal"};
local pt2_z_lab = iup.label{title="Z:"};
local pt2_z_txt = iup.text{expand="Horizontal"};
local color_lab = iup.label{title="Color:",size="50X"};
local color_r_lab = iup.label{title="R:"};
local color_r_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local color_g_lab = iup.label{title="G:"};
local color_g_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local color_b_lab = iup.label{title="B:"};
local color_b_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local ok_btn = iup.button{title="OK",size="100X"};
local cancel_btn = iup.button{title="Cancel",size="100X"};
local dlg = iup.dialog{
	title = "Property";
	size = "500X100";
```

```
        margin = "5X5";
        iup.vbox{
            iup.hbox{pt1_lab,pt1_x_lab,pt1_x_txt,pt1_y_lab,pt1_y_txt,pt1_z_lab,pt1_z_txt};
            iup.hbox{pt2_lab,pt2_x_lab,pt2_x_txt,pt2_y_lab,pt2_y_txt,pt2_z_lab,pt2_z_txt};
            iup.hbox{color_lab,color_r_lab,color_r_txt,color_g_lab,color_g_txt,
                                color_b_lab,color_b_txt};
            iup.hbox{iup.fill{},ok_btn,cancel_btn};
        };
    }
    function init_dlg()
        if type(selected)~="number" or selected<=0 or selected>#objects then return end
        pt1_x_txt.value = objects[selected].pt1.x;
        pt1_y_txt.value = objects[selected].pt1.y;
        pt1_z_txt.value = objects[selected].pt1.z;
        pt2_x_txt.value = objects[selected].pt2.x;
        pt2_y_txt.value = objects[selected].pt2.y;
        pt2_z_txt.value = objects[selected].pt2.z;
        color_r_txt.value = objects[selected].color.r;
        color_g_txt.value = objects[selected].color.g;
        color_b_txt.value = objects[selected].color.b;
    end
    function ok_btn:action()
        if type(selected)~="number" or selected<=0 or selected>#objects then return end
        objects[selected].pt1.x = pt1_x_txt.value;
        objects[selected].pt1.y = pt1_y_txt.value;
        objects[selected].pt1.z = pt1_z_txt.value;
        objects[selected].pt2.x = pt2_x_txt.value;
        objects[selected].pt2.y = pt2_y_txt.value;
        objects[selected].pt2.z = pt2_z_txt.value;
        select_object(selected)
        scene_onpaint(scene);
        dlg:hide();
    end
    function cancel_btn:action()
        dlg:hide();
    end
    function show_dlg()
        init_dlg();
        dlg:popup();
    end
    function on_lbuttondblclk(scene,flags,x,y)
        show_dlg();
    end
```

```
local ID_PROPERTY = ID+1;
local ID_ADD = ID+2;
add_menu(
    frm,
    {
        name = "Cube",
        nposition = 2,
        items =
        {
            {id=ID_ADD,name="Add"},
            {id=ID_PROPERTY,name="Property"},
        },
    }
);
crt_toolbar(frm,
    {
        bmpname = "toolbar1.bmp",
        nbmps = 3,
        dxButton = 0,
        dyButton = 0,
        dxBitmap = 16,
        dyBitmap = 16,
        buttons = {
            {iBitmap=2,idCommand=ID_ADD,iString="Add",
                    fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
            {iBitmap=5,idCommand=ID_PROPERTY,iString="Property",
                    fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
        },
    }
);
local commands = {};
commands[ID_PROPERTY] = function(scene)
    show_dlg();
end
commands[ID_ADD] = function(scene)
    add_object();
end
function on_command(id,scene)
    if type(commands[id])=="function" then
        commands[id](scene);
    end
end
```

查看源代码（apcad/help/src/primer/3.13.5.txt ）

运行主程序（gcad.exe），查看运行结果（新建几个立方体，双击查看属性对话框）。



Point1: X: 10000 Y: 10000 Z: 10000

Point2: X: 13000 Y: 13000 Z: 13000

Color: R: 0.5 G: 1.0 B: 0.0

好吧，属性对话框终于又正常了。

第十四节　状态栏显示

（一）设置状态栏，函数（statusbar_set_parts）的第二个参数是一个数组，按数组的数量设置状态栏分栏，数组的每个数值是这个分栏的宽度。

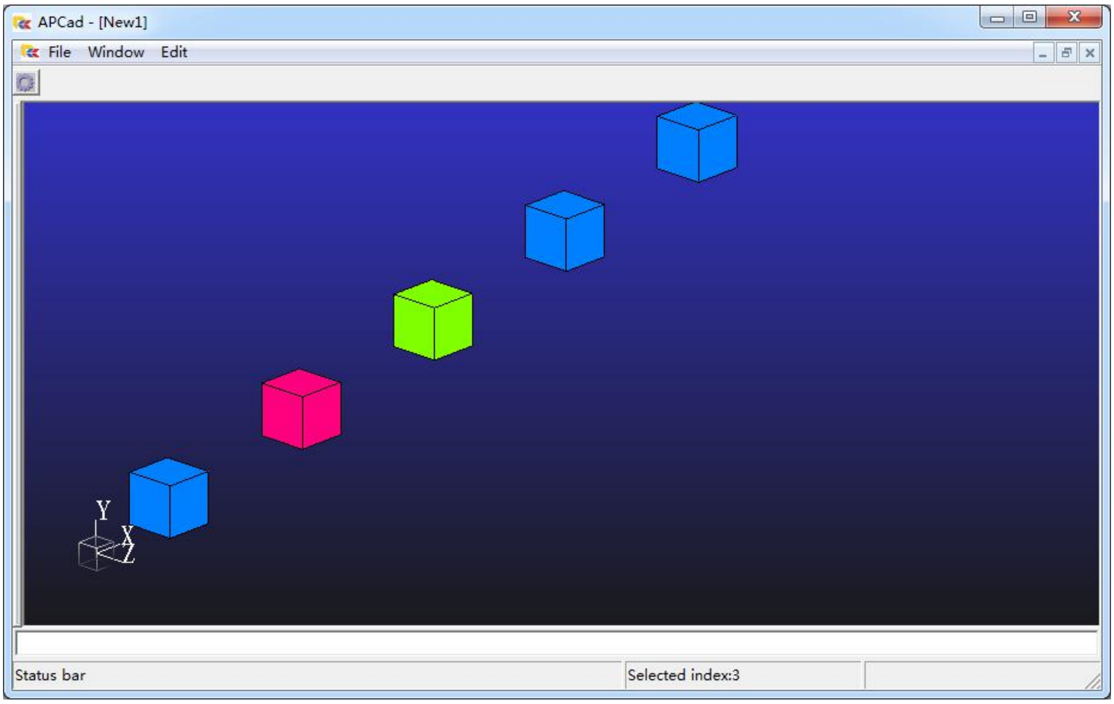
```
statusbar_set_parts(frm,{200,200})
```

（二）选择立方体时，设置状态栏的显示文本。

```
function select_main(i)
    select_shape(i)
    statusbar_set_text(frm,1,"Selected index:"..i);
end
```

查看源代码（apcad/help/src/primer/3.14.txt）

运行主程序（gcad.exe），查看运行结果（选择某个立方体）。



第十五节　鼠标绘制立方体

（一）制作菜单和工具条。

```
add_menu(
    frm,
    {
        name = "Cube",
        nposition = 2,
        items =
        {
            {id=ID_ADD,name="Add"},
            {id=ID_DRAW,name="Draw"},
            {id=ID_PROPERTY,name="Property"},
        },
    }
);
crt_toolbar(frm,
    {
        bmpname = "toolbar1.bmp",
        nbmps = 3,
        dxButton = 0,
        dyButton = 0,
        dxBitmap = 16,
        dyBitmap = 16,
        buttons = {
```

```
                    {iBitmap=2,idCommand=ID_ADD,iString="Add",
                                fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
                    {iBitmap=3,idCommand=ID_DRAW,iString="Draw",
                                fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
                    {iBitmap=5,idCommand=ID_PROPERTY,iString="Property",
                                fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
              },
        }
);
```

查看源代码（apcad/help/src/primer/3.15.1.txt）

（二）响应消息，设置进入绘制状态。

```
commands[ID_DRAW] = function(scene)
      start_pt = true;
end
```

查看源代码（apcad/help/src/primer/3.15.2.txt）

（三）在左键按下时，判断绘制状态，并记录按下的点坐标，函数（client_2_world）把屏幕坐标换算成三维坐标

```
local start_pt = nil;
function on_lbuttondown(scene,flags,x,y)
      if start_pt then
            start_pt = {x,y};
            local x,y,z = client_2_world(scene,x,y);
            start_pt = {x=x,y=y,z=z};
      else
            scene_select(scene,x,y,1,1,1);
      end
end
```

查看源代码（apcad/help/src/primer/3.15.3.txt）

（四）在左键抬起时，记录抬起点的坐标，换算成三维坐标后，用前后两个点的坐标添加新的立方体（add_object）。

```
function on_lbuttonup(scene,flags,x,y)
      if type(start_pt)=="table" then
            local x,y,z = client_2_world(scene,x,y);
            local pt = {x=x,y=y,z=z};
            add_object(start_pt,pt);
            start_pt = nil;
      end
end
```

查看源代码（apcad/help/src/primer/3.15.4.txt ）

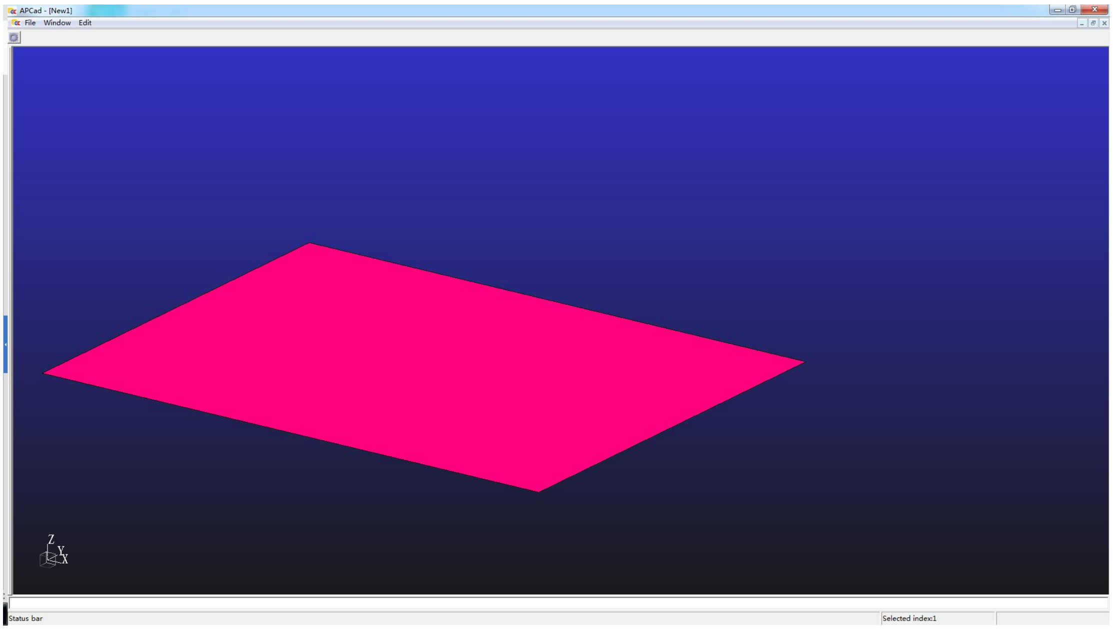
完整的代码如下：

```
package.cpath = "./?53.dll;./?.dll";

function frmclose()
    os.exit();
end
function on_mousemove()
end
function on_paint()
end

function get_shape(pt1,pt2,color)
    local outer_pts = {
        {color.r,color.g,color.b,1,1,pt1.x,pt1.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt1.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt2.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt2.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt1.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt1.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt2.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt2.y,pt2.z};
    };
    local lines_pts = {
        {0,0,0,1,1,pt1.x,pt1.y,pt1.z};
        {0,0,0,1,1,pt2.x,pt1.y,pt1.z};
        {0,0,0,1,1,pt1.x,pt2.y,pt1.z};
        {0,0,0,1,1,pt2.x,pt2.y,pt1.z};
        {0,0,0,1,1,pt1.x,pt1.y,pt2.z};
        {0,0,0,1,1,pt2.x,pt1.y,pt2.z};
        {0,0,0,1,1,pt1.x,pt2.y,pt2.z};
        {0,0,0,1,1,pt2.x,pt2.y,pt2.z};
    };
    local shape = {
        surfaces = {
            {
                points = lines_pts;
                lines = {{1,2},{1,3},{2,4},{3,4};{5,6},{5,7},{6,8},{7,8};{1,5},{2,6},{3,7},{4,8}};
            };
            {
                points = outer_pts;
                outer = {1,3,4,2};
            };
            {
                points = outer_pts;
                outer = {5,6,8,7};
            };
            {
                points = outer_pts;
                outer = {1,2,6,5};
            };
            {
                points = outer_pts;
                outer = {1,5,7,3};
            };
            {
                points = outer_pts;
                outer = {2,4,8,6};
            };
            {
                points = outer_pts;
                outer = {3,7,8,4};
            };
        };
    };
    return shape;
end

statusbar_set_parts(frm,{200,200})
local scene = new_child(frm,"New1");
local objects = {};
function add_object(pt1,pt2)
    local n = #objects;
    pt1 = pt1 or {x=5000*n,y=5000*n,z=5000*n};
    pt2 = pt2 or {x=5000*n+3000,y=5000*n+3000,z=5000*n+3000};
    local color = {r=0,g=0.5,b=1};
    local shape = get_shape(pt1,pt2,color);
    local glname,gllist = n+1,makelist(scene,shape);
    objects[n+1] = {pt1=pt1,pt2=pt2,color=color,glname=glname,gllist=gllist};
    scene_onpaint(scene);
end

local selected = nil;
function select_object(i)
    objects[i].color.r = (objects[i].color.r-0.5)%1.5;
    objects[i].color.g = (objects[i].color.g-0.5)%1.5;
    objects[i].color.b = (objects[i].color.b-0.5)%1.5;
    local object = get_shape(objects[i].pt1,objects[i].pt2,objects[i].color);
    local glname,gllist = i,makelist(scene,object);
    objects[i].gllist = gllist;
    scene_onpaint(scene);
    selected = i;
end
function select_main(i)
    select_object(i)
    statusbar_set_text(frm,1,"Selected index:"..i);
end

local start_pt = nil;
function on_lbuttondown(scene,flags,x,y)
    if start_pt then
        start_pt = {x,y};
        local x,y,z = client_2_world(scene,x,y);
        start_pt = {x=x,y=y,z=z};
    else
        scene_select(scene,x,y,1,1,1);
    end
end

function on_lbuttonup(scene,flags,x,y)
    if type(start_pt)=="table" then
        local x,y,z = client_2_world(scene,x,y);
        local pt = {x=x,y=y,z=z};
        add_object(start_pt,pt);
        start_pt = nil;
    end
end

local gl = require "luaext.gl"
function render_objs()
    for i,v in ipairs(objects) do
        gl.glLoadName(v.glname);
        gl.glCallList(v.gllist);
    end
end

local iup = require"iuplua"
local pt1_lab = iup.label{title="Point1:",size="50x"};
local pt1_x_lab = iup.label{title="X:"};
local pt1_x_txt = iup.text{expand="Horizontal"};
local pt1_y_lab = iup.label{title="Y:"};
local pt1_y_txt = iup.text{expand="Horizontal"};
local pt1_z_lab = iup.label{title="Z:"};
local pt1_z_txt = iup.text{expand="Horizontal"};
local pt2_lab = iup.label{title="Point2:",size="50X"};
local pt2_x_lab = iup.label{title="X:"};
local pt2_x_txt = iup.text{expand="Horizontal"};
local pt2_y_lab = iup.label{title="Y:"};
local pt2_y_txt = iup.text{expand="Horizontal"};
local pt2_z_lab = iup.label{title="Z:"};
local pt2_z_txt = iup.text{expand="Horizontal"};
local color_lab = iup.label{title="Color:",size="50X"};
local color_r_lab = iup.label{title="R:"};
local color_r_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local color_g_lab = iup.label{title="G:"};
local color_g_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local color_b_lab = iup.label{title="B:"};
local color_b_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local ok_btn = iup.button{title="OK",size="100X"};
local cancel_btn = iup.button{title="Cancel",size="100X"};
local dlg = iup.dialog{
    title = "Property";
    size = "500X100";
    margin = "5X5";
    iup.vbox{
        iup.hbox{pt1_lab,pt1_x_lab,pt1_x_txt,pt1_y_lab,pt1_y_txt,pt1_z_lab,pt1_z_txt};
        iup.hbox{pt2_lab,pt2_x_lab,pt2_x_txt,pt2_y_lab,pt2_y_txt,pt2_z_lab,pt2_z_txt};
        iup.hbox{color_lab,color_r_lab,color_r_txt,color_g_lab,color_g_txt,
                    color_b_lab,color_b_txt};
        iup.hbox{iup.fill{},ok_btn,cancel_btn};
    };
}
function init_dlg()
    if type(selected)~="number" or selected<=0 or selected>#objects then return end
    pt1_x_txt.value = objects[selected].pt1.x;
    pt1_y_txt.value = objects[selected].pt1.y;
    pt1_z_txt.value = objects[selected].pt1.z;
    pt2_x_txt.value = objects[selected].pt2.x;
    pt2_y_txt.value = objects[selected].pt2.y;
    pt2_z_txt.value = objects[selected].pt2.z;
    color_r_txt.value = objects[selected].color.r;
    color_g_txt.value = objects[selected].color.g;
    color_b_txt.value = objects[selected].color.b;
end
function ok_btn:action()
```

```
    if type(selected)~="number" or selected<=0 or selected>#objects then return end
    objects[selected].pt1.x = pt1_x_txt.value;
    objects[selected].pt1.y = pt1_y_txt.value;
    objects[selected].pt1.z = pt1_z_txt.value;
    objects[selected].pt2.x = pt2_x_txt.value;
    objects[selected].pt2.y = pt2_y_txt.value;
    objects[selected].pt2.z = pt2_z_txt.value;
    select_object(selected)
    scene_onpaint(scene);
    dlg:hide();
end
function cancel_btn:action()
    dlg:hide();
end
function show_dlg()
    init_dlg();
    dlg:popup();
end
function on_lbuttondblclk(scene,flags,x,y)
    show_dlg();
end

local ID_PROPERTY = ID+1;
local ID_ADD = ID+2;
local ID_DRAW = ID+3;
add_menu(
    frm,
    {
        name = "Cube",
        nposition = 2,
        items =
        {
            {id=ID_ADD,name="Add"},
            {id=ID_DRAW,name="Draw"},
            {id=ID_PROPERTY,name="Property"},
        },
    }
);
crt_toolbar(frm,
    {
        bmpname = "toolbar1.bmp",
        nbmps = 3,
        dxButton = 0,
        dyButton = 0,
                dxBitmap = 16,
                dyBitmap = 16,
                buttons = {
                        {iBitmap=2,idCommand=ID_ADD,iString="Add",
                                        fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
                        {iBitmap=3,idCommand=ID_DRAW,iString="Draw",
                                        fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
                        {iBitmap=5,idCommand=ID_PROPERTY,iString="Property",
                                        fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
                },
        }
);
local commands = {};
commands[ID_PROPERTY] = function(scene)
        show_dlg();
end
commands[ID_ADD] = function(scene)
        add_object();
end
commands[ID_DRAW] = function(scene)
        start_pt = true;
end
function on_command(id,scene)
        if type(commands[id])=="function" then
                commands[id](scene);
        end
end
```

查看源代码（apcad/help/src/primer/3.15.5.txt）

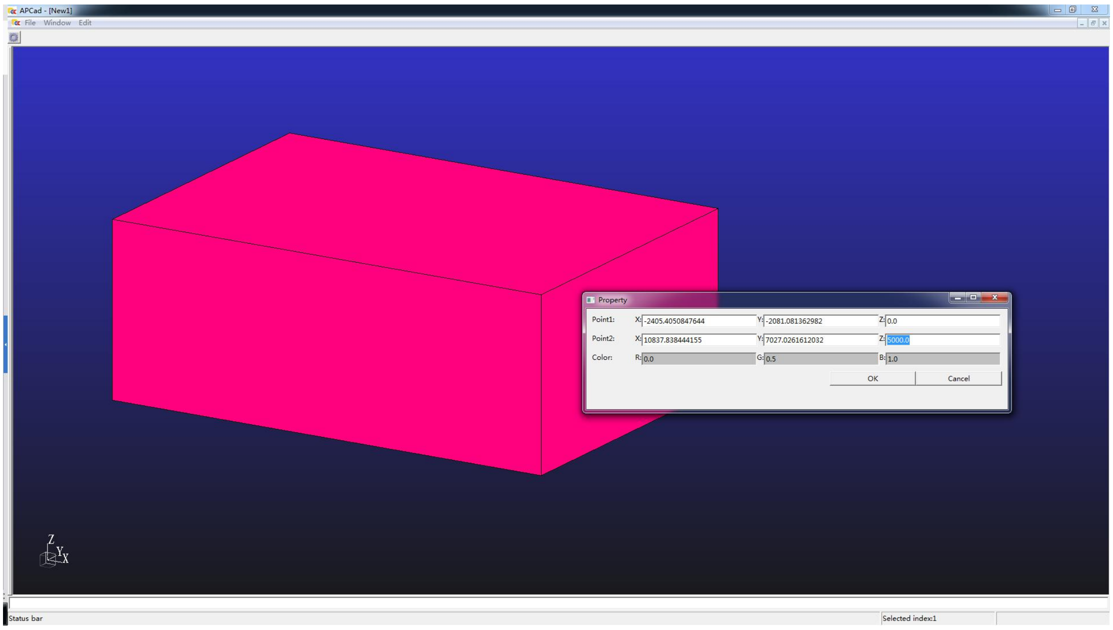
运行主程序（gcad.exe），查看运行结果（点击 Cube-Draw 菜单进入绘制状态，按下左键按住移动后抬起）。



等等（又等等），说好的立方体呢，这只是一个矩形！

第十六节　捕捉

关于上节的矩形，没有绘制成为立方体是因为，鼠标选取的两个点都是屏幕坐标，换算到三维坐标之后两个点在一个平面上，可以通过弹出属性对话框修改两个点的坐标值的方式得到一个立方体，如下：



Point1:	X: -2405.4050847644	Y: -2081.081362982	Z: 0.0
Point2:	X: 10837.838444155	Y: 7027.0261612032	Z: 5000.0
Color:	R: 0.0	G: 0.5	B: 1.0

或者，可以在选取点的时候捕捉到原有的不在一个平面上的两个，以实现绘制三维立方体的目的。

（一）定义捕捉函数（snap），计算所有立方体的每个顶点在屏幕坐标体系

中与鼠标选取点之间的距离。

```
function snap(scene,x,y)
    for i,v in pairs(objects) do
        local shape = get_shape(v.pt1,v.pt2,v.color);
        for i,v in pairs(shape.surfaces[1].points) do
            local pt = {world_2_client(scene,v[6],v[7],v[8])};
            if math.abs(pt[1]-x)<=15 and math.abs(pt[2]-y)<=15 then
                return {x=v[6],y=v[7],z=v[8]};
            end
        end
    end
    local x,y,z = client_2_world(scene,x,y);
    return {x=x,y=y,z=z};
end
```

查看源代码（apcad/help/src/primer/3.16.1.txt）

（二）在鼠标左键按下/抬起消息函数中，调用捕捉函数以获得相对应的三维坐标点。

```
function on_lbuttondown(scene,flags,x,y)
    if start_pt then
        start_pt = snap(scene,x,y);
    else
        scene_select(scene,x,y,1,1,1);
    end
end
function on_lbuttonup(scene,flags,x,y)
    if type(start_pt)=="table" then
        local pt = snap(scene,x,y);
        add_object(start_pt,pt);
        start_pt = nil;
        scene_cursor(scene,IDC_ARROW)
    end
end
```

查看源代码（apcad/help/src/primer/3.16.2.txt）

完整的程序代码如下：

```
package.cpath = "./?53.dll;./?.dll";

function frmclose()
    os.exit();
end
function on_mousemove()
```

```
end
function on_paint()
end

function get_shape(pt1,pt2,color)
    local outer_pts = {
        {color.r,color.g,color.b,1,1,pt1.x,pt1.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt1.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt2.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt2.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt1.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt1.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt2.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt2.y,pt2.z};
    };
    local lines_pts = {
        {0,0,0,1,1,pt1.x,pt1.y,pt1.z};
        {0,0,0,1,1,pt2.x,pt1.y,pt1.z};
        {0,0,0,1,1,pt1.x,pt2.y,pt1.z};
        {0,0,0,1,1,pt2.x,pt2.y,pt1.z};
        {0,0,0,1,1,pt1.x,pt1.y,pt2.z};
        {0,0,0,1,1,pt2.x,pt1.y,pt2.z};
        {0,0,0,1,1,pt1.x,pt2.y,pt2.z};
        {0,0,0,1,1,pt2.x,pt2.y,pt2.z};
    };
    local shape = {
        surfaces = {
            {
                points = lines_pts;
                lines = {{1,2},{1,3},{2,4},{3,4};{5,6},{5,7},{6,8},{7,8};{1,5},{2,6},{3,7},{4,8}};
            };
            {
                points = outer_pts;
                outer = {1,3,4,2};
            };
            {
                points = outer_pts;
                outer = {5,6,8,7};
            };
            {
                points = outer_pts;
                outer = {1,2,6,5};
            };
            {
                points = outer_pts;
                outer = {1,5,7,3};
            };
            {
                points = outer_pts;
                outer = {2,4,8,6};
            };
            {
                points = outer_pts;
                outer = {3,7,8,4};
            };
        };
    };
    return shape;
end

statusbar_set_parts(frm,{200,200})
local scene = new_child(frm,"New1");
local objects = {};
function add_object(pt1,pt2)
    local n = #objects;
    pt1 = pt1 or {x=5000*n,y=5000*n,z=5000*n};
    pt2 = pt2 or {x=5000*n+3000,y=5000*n+3000,z=5000*n+3000};
    local color = {r=0,g=0.5,b=1};
    local shape = get_shape(pt1,pt2,color);
    local glname,gllist = n+1,makelist(scene,shape);
    objects[n+1] = {pt1=pt1,pt2=pt2,color=color,glname=glname,gllist=gllist};
    scene_onpaint(scene);
end

local selected = nil;
function select_object(i)
    objects[i].color.r = (objects[i].color.r-0.5)%1.5;
    objects[i].color.g = (objects[i].color.g-0.5)%1.5;
    objects[i].color.b = (objects[i].color.b-0.5)%1.5;
    local object = get_shape(objects[i].pt1,objects[i].pt2,objects[i].color);
    local glname,gllist = i,makelist(scene,object);
    objects[i].gllist = gllist;
    scene_onpaint(scene);
    selected = i;
end
function select_main(i)
    select_object(i)
    statusbar_set_text(frm,1,"Selected index:"..i);
```

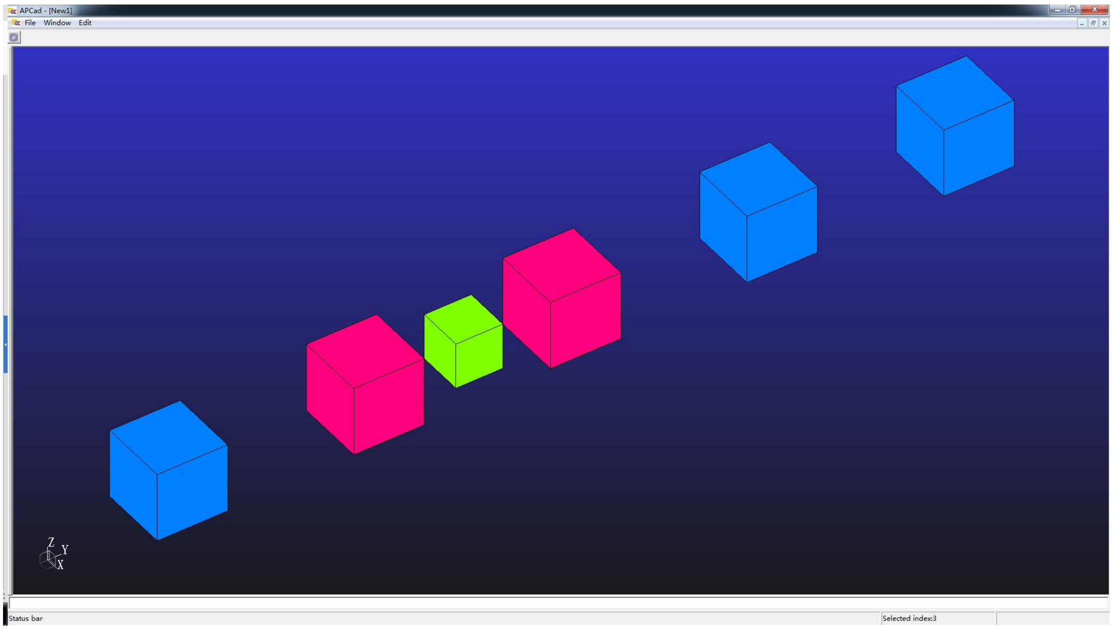
```
end

function snap(scene,x,y)
    for i,v in pairs(objects) do
        local shape = get_shape(v.pt1,v.pt2,v.color);
        for i,v in pairs(shape.surfaces[1].points) do
            local pt = {world_2_client(scene,v[6],v[7],v[8])};
            if math.abs(pt[1]-x)<=15 and math.abs(pt[2]-y)<=15 then
                return {x=v[6],y=v[7],z=v[8]};
            end
        end
    end
    local x,y,z = client_2_world(scene,x,y);
    return {x=x,y=y,z=z};
end

local start_pt = nil;
function on_lbuttondown(scene,flags,x,y)
    if start_pt then
        start_pt = snap(scene,x,y);
    else
        scene_select(scene,x,y,1,1,1);
    end
end
function on_lbuttonup(scene,flags,x,y)
    if type(start_pt)=="table" then
        local pt = snap(scene,x,y);
        add_object(start_pt,pt);
        start_pt = nil;
        scene_cursor(scene,IDC_ARROW)
    end
end

local gl = require "luaext.gl"
function render_objs()
    for i,v in ipairs(objects) do
        gl.glLoadName(v.glname);
        gl.glCallList(v.gllist);
    end
end

local iup = require"iuplua"
local pt1_lab = iup.label{title="Point1:",size="50x"};
local pt1_x_lab = iup.label{title="X:"};
local pt1_x_txt = iup.text{expand="Horizontal"};
local pt1_y_lab = iup.label{title="Y:"};
local pt1_y_txt = iup.text{expand="Horizontal"};
local pt1_z_lab = iup.label{title="Z:"};
local pt1_z_txt = iup.text{expand="Horizontal"};
local pt2_lab = iup.label{title="Point2:",size="50X"};
local pt2_x_lab = iup.label{title="X:"};
local pt2_x_txt = iup.text{expand="Horizontal"};
local pt2_y_lab = iup.label{title="Y:"};
local pt2_y_txt = iup.text{expand="Horizontal"};
local pt2_z_lab = iup.label{title="Z:"};
local pt2_z_txt = iup.text{expand="Horizontal"};
local color_lab = iup.label{title="Color:",size="50X"};
local color_r_lab = iup.label{title="R:"};
local color_r_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local color_g_lab = iup.label{title="G:"};
local color_g_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local color_b_lab = iup.label{title="B:"};
local color_b_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local ok_btn = iup.button{title="OK",size="100X"};
local cancel_btn = iup.button{title="Cancel",size="100X"};
local dlg = iup.dialog{
    title = "Property";
    size = "500X100";
    margin = "5X5";
    iup.vbox{
        iup.hbox{pt1_lab,pt1_x_lab,pt1_x_txt,pt1_y_lab,pt1_y_txt,pt1_z_lab,pt1_z_txt};
        iup.hbox{pt2_lab,pt2_x_lab,pt2_x_txt,pt2_y_lab,pt2_y_txt,pt2_z_lab,pt2_z_txt};
        iup.hbox{color_lab,color_r_lab,color_r_txt,color_g_lab,color_g_txt,
            color_b_lab,color_b_txt};
        iup.hbox{iup.fill{},ok_btn,cancel_btn};
    };
}
function init_dlg()
    if type(selected)~="number" or selected<=0 or selected>#objects then return end
    pt1_x_txt.value = objects[selected].pt1.x;
    pt1_y_txt.value = objects[selected].pt1.y;
    pt1_z_txt.value = objects[selected].pt1.z;
    pt2_x_txt.value = objects[selected].pt2.x;
    pt2_y_txt.value = objects[selected].pt2.y;
    pt2_z_txt.value = objects[selected].pt2.z;
    color_r_txt.value = objects[selected].color.r;
    color_g_txt.value = objects[selected].color.g;
    color_b_txt.value = objects[selected].color.b;
```

```
end
function ok_btn:action()
    if type(selected)~="number" or selected<=0 or selected>#objects then return end
    objects[selected].pt1.x = pt1_x_txt.value;
    objects[selected].pt1.y = pt1_y_txt.value;
    objects[selected].pt1.z = pt1_z_txt.value;
    objects[selected].pt2.x = pt2_x_txt.value;
    objects[selected].pt2.y = pt2_y_txt.value;
    objects[selected].pt2.z = pt2_z_txt.value;
    select_object(selected)
    scene_onpaint(scene);
    dlg:hide();
end
function cancel_btn:action()
    dlg:hide();
end
function show_dlg()
    init_dlg();
    dlg:popup();
end
function on_lbuttondblclk(scene,flags,x,y)
    show_dlg();
end

local ID_PROPERTY = ID+1;
local ID_ADD = ID+2;
local ID_DRAW = ID+3;
add_menu(
    frm,
    {
        name = "Cube",
        nposition = 2,
        items =
        {
            {id=ID_ADD,name="Add"},
            {id=ID_DRAW,name="Draw"},
            {id=ID_PROPERTY,name="Property"},
        },
    }
);
crt_toolbar(frm,
    {
        bmpname = "toolbar1.bmp",
        nbmps = 3,
            dxButton = 0,
            dyButton = 0,
            dxBitmap = 16,
            dyBitmap = 16,
            buttons = {
                {iBitmap=2,idCommand=ID_ADD,iString="Add",
                        fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
                {iBitmap=3,idCommand=ID_DRAW,iString="Draw",
                        fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
                {iBitmap=5,idCommand=ID_PROPERTY,iString="Property",
                        fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
            },
        }
    );
    local commands = {};
    commands[ID_PROPERTY] = function(scene)
        show_dlg();
    end
    commands[ID_ADD] = function(scene)
        add_object();
    end
    commands[ID_DRAW] = function(scene)
        start_pt = true;
        scene_cursor(scene,IDC_CROSS)
    end
    function on_command(id,scene)
        if type(commands[id])=="function" then
            commands[id](scene);
        end
    end
```

查看源代码（apcad/help/src/primer/3.16.3.txt ）

运行主程序（gcad.exe），查看运行结果（先点击 Add 菜单创建几个三维立方体，然后点击 Draw 菜单捕捉立方体的顶点绘制新的立方体）。

数。



第十七节　拖拽橡皮线

上面的按住左键拖拽绘制立方体的过程，现在增加显示鼠标移动线的过程，也就是拖拽橡皮线。

（一）定义计算拖拽线图形的函数。

```
function get_drag_shape(pt1,pt2)
    local shape = {
        surfaces = {
            {
                points = {
                    {1,1,1,1,1,pt1.x,pt1.y,pt1.z};
                    {1,1,1,1,1,pt2.x,pt2.y,pt2.z};
                };
                lines = {{1,2}};
            };
        };
    };
    return shape;
end
```

查看源代码（apcad/help/src/primer/3.17.1.txt ）

（二）定义拖拽线变量（drag_line）以及设置（set）/删除（del）拖拽线函数。

```
local gl = require "luaext.gl"
local drag_line = nil;
function del_drag_line()
```

```
	if drag_line then
		gl.glDeleteLists(drag_line);
		drag_line = nil;
	end
end
function set_drag_line(pt1,pt2)
	del_drag_line();
	drag_line = makelist(scene,get_drag_shape(pt1,pt2));
end
```

查看源代码（apcad/help/src/primer/3.17.2.txt ）

（三）定义平台消息函数（render_drags），判断如果拖拽线存在就绘制它。

```
function render_drags()
	if drag_line then
		gl.glCallList(drag_line);
	end
end
```

查看源代码（apcad/help/src/primer/3.17.3.txt ）

（四）定义平台消息函数（on_paint），在窗口刷新时删除拖拽线。

```
function on_paint()
	del_drag_line();
end
```

查看源代码（apcad/help/src/primer/3.17.4.txt ）

（五）定义平台消息函数（on_mousemove），在按住左键拖动鼠标移动时，绘制调用平台接口函数（draw_drag）绘制拖拽线。

```
function on_mousemove(scene,flags,x,y)
	if type(start_pt)=="table" then
		local pt = snap(scene,x,y);
		draw_drag(scene);
		set_drag_line(start_pt,pt);
		draw_drag(scene);
	end
end
```

查看源代码（apcad/help/src/primer/3.17.5.txt ）

绘制拖拽线时，需先擦除前一条拖拽线，然后绘制新的拖拽线，因此绘制拖拽线函数需要调用两次。

完整的程序代码如下：

```
package.cpath = "./?53.dll;./?.dll";

function frmclose()
```

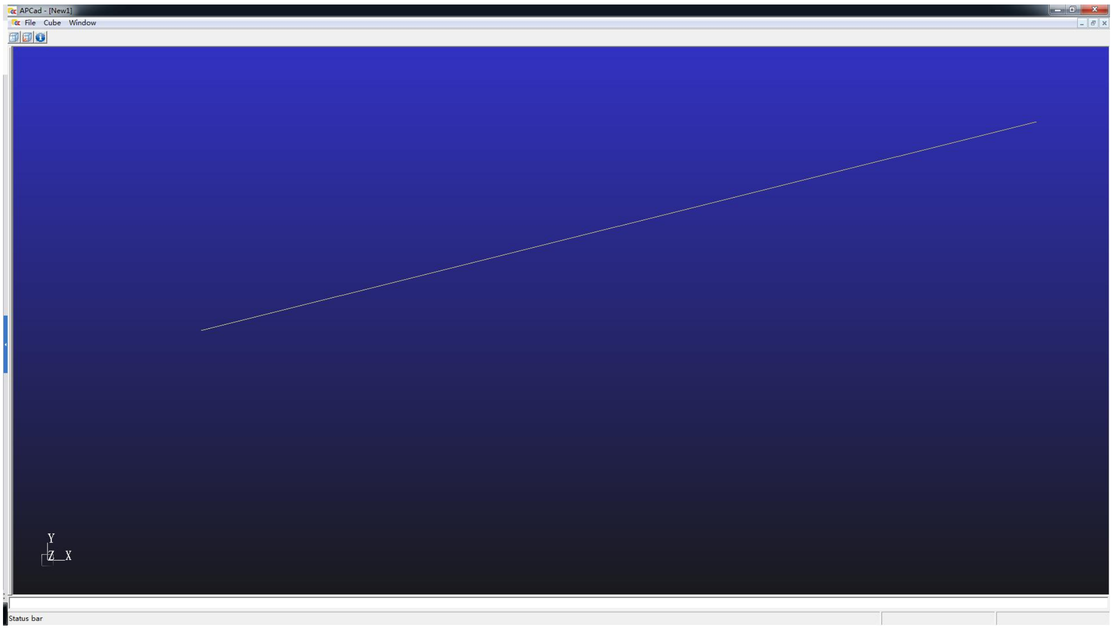
```
    os.exit();
end

function get_shape(pt1,pt2,color)
    local outer_pts = {
        {color.r,color.g,color.b,1,1,pt1.x,pt1.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt1.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt2.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt2.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt1.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt1.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt2.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt2.y,pt2.z};
    };
    local lines_pts = {
        {0,0,0,1,1,pt1.x,pt1.y,pt1.z};
        {0,0,0,1,1,pt2.x,pt1.y,pt1.z};
        {0,0,0,1,1,pt1.x,pt2.y,pt1.z};
        {0,0,0,1,1,pt2.x,pt2.y,pt1.z};
        {0,0,0,1,1,pt1.x,pt1.y,pt2.z};
        {0,0,0,1,1,pt2.x,pt1.y,pt2.z};
        {0,0,0,1,1,pt1.x,pt2.y,pt2.z};
        {0,0,0,1,1,pt2.x,pt2.y,pt2.z};
    };
    local shape = {
        surfaces = {
            {
                points = lines_pts;
                lines = {{1,2},{1,3},{2,4},{3,4};{5,6},{5,7},{6,8},{7,8};{1,5},{2,6},{3,7},{4,8}};
            };
            {
                points = outer_pts;
                outer = {1,3,4,2};
            };
            {
                points = outer_pts;
                outer = {5,6,8,7};
            };
            {
                points = outer_pts;
                outer = {1,2,6,5};
            };
            {
                points = outer_pts;
```

```
                        outer = {1,5,7,3};
                };
                {
                        points = outer_pts;
                        outer = {2,4,8,6};
                };
                {
                        points = outer_pts;
                        outer = {3,7,8,4};
                };
            };
    };
    return shape;
end

statusbar_set_parts(frm,{200,200})
local scene = new_child(frm,"New1");
local objects = {};
function add_object(pt1,pt2)
    local n = #objects;
    pt1 = pt1 or {x=5000*n,y=5000*n,z=5000*n};
    pt2 = pt2 or {x=5000*n+3000,y=5000*n+3000,z=5000*n+3000};
    local color = {r=0,g=0.5,b=1};
    local shape = get_shape(pt1,pt2,color);
    local glname,gllist = n+1,makelist(scene,shape);
    objects[n+1] = {pt1=pt1,pt2=pt2,color=color,glname=glname,gllist=gllist};
    scene_onpaint(scene);
end

local selected = nil;
function select_object(i)
    objects[i].color.r = (objects[i].color.r-0.5)%1.5;
    objects[i].color.g = (objects[i].color.g-0.5)%1.5;
    objects[i].color.b = (objects[i].color.b-0.5)%1.5;
    local object = get_shape(objects[i].pt1,objects[i].pt2,objects[i].color);
    local glname,gllist = i,makelist(scene,object);
    objects[i].gllist = gllist;
    scene_onpaint(scene);
    selected = i;
end
function select_main(i)
    select_object(i)
    statusbar_set_text(frm,1,"Selected index:"..i);
end
```

```
function snap(scene,x,y)
    for i,v in pairs(objects) do
        local shape = get_shape(v.pt1,v.pt2,v.color);
        for i,v in pairs(shape.surfaces[1].points) do
            local pt = {world_2_client(scene,v[6],v[7],v[8])};
            if math.abs(pt[1]-x)<=15 and math.abs(pt[2]-y)<=15 then
                return {x=v[6],y=v[7],z=v[8]};
            end
        end
    end
    local x,y,z = client_2_world(scene,x,y);
    return {x=x,y=y,z=z};
end

function get_drag_shape(pt1,pt2)
    local shape = {
        surfaces = {
            {
                points = {
                    {1,1,1,1,1,pt1.x,pt1.y,pt1.z};
                    {1,1,1,1,1,pt2.x,pt2.y,pt2.z};
                };
                lines = {{1,2}};
            };
        };
    };
    return shape;
end
local gl = require "luaext.gl"
local drag_line = nil;
function del_drag_line()
    if drag_line then
         gl.glDeleteLists(drag_line);
         drag_line = nil;
    end
end
function set_drag_line(pt1,pt2)
    del_drag_line();
    drag_line = makelist(scene,get_drag_shape(pt1,pt2));
end
function render_drags()
    if drag_line then
        gl.glCallList(drag_line);
    end
end
function on_paint()
    del_drag_line();
end

local start_pt = nil;
function on_lbuttondown(scene,flags,x,y)
    if start_pt then
        start_pt = snap(scene,x,y);
    else
        scene_select(scene,x,y,1,1,1);
    end
end
function on_mousemove(scene,flags,x,y)
    if type(start_pt)=="table" then
        local pt = snap(scene,x,y);
        draw_drag(scene);
        set_drag_line(start_pt,pt);
        draw_drag(scene);
    end
end
function on_lbuttonup(scene,flags,x,y)
    if type(start_pt)=="table" then
        local pt = snap(scene,x,y);
        add_object(start_pt,pt);
        scene_cursor(scene,IDC_ARROW)
        start_pt = nil;
    end
end

function render_objs()
    for i,v in ipairs(objects) do
        gl.glLoadName(v.glname);
        gl.glCallList(v.gllist);
    end
end

local iup = require"iuplua"
local pt1_lab = iup.label{title="Point1:",size="50x"};
local pt1_x_lab = iup.label{title="X:"};
local pt1_x_txt = iup.text{expand="Horizontal"};
local pt1_y_lab = iup.label{title="Y:"};
local pt1_y_txt = iup.text{expand="Horizontal"};
local pt1_z_lab = iup.label{title="Z:"};
local pt1_z_txt = iup.text{expand="Horizontal"};
local pt2_lab = iup.label{title="Point2:",size="50X"};
local pt2_x_lab = iup.label{title="X:"};
local pt2_x_txt = iup.text{expand="Horizontal"};
local pt2_y_lab = iup.label{title="Y:"};
local pt2_y_txt = iup.text{expand="Horizontal"};
local pt2_z_lab = iup.label{title="Z:"};
local pt2_z_txt = iup.text{expand="Horizontal"};
local color_lab = iup.label{title="Color:",size="50X"};
local color_r_lab = iup.label{title="R:"};
local color_r_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local color_g_lab = iup.label{title="G:"};
local color_g_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local color_b_lab = iup.label{title="B:"};
local color_b_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local ok_btn = iup.button{title="OK",size="100X"};
local cancel_btn = iup.button{title="Cancel",size="100X"};
local dlg = iup.dialog{
    title = "Property";
    size = "500X100";
    margin = "5X5";
    iup.vbox{
        iup.hbox{pt1_lab,pt1_x_lab,pt1_x_txt,pt1_y_lab,pt1_y_txt,pt1_z_lab,pt1_z_txt};
        iup.hbox{pt2_lab,pt2_x_lab,pt2_x_txt,pt2_y_lab,pt2_y_txt,pt2_z_lab,pt2_z_txt};
        iup.hbox{color_lab,color_r_lab,color_r_txt,color_g_lab,color_g_txt,
                                                    color_b_lab,color_b_txt};
        iup.hbox{iup.fill{},ok_btn,cancel_btn};
    };
}
function init_dlg()
    if type(selected)~="number" or selected<=0 or selected>#objects then return end
    pt1_x_txt.value = objects[selected].pt1.x;
    pt1_y_txt.value = objects[selected].pt1.y;
    pt1_z_txt.value = objects[selected].pt1.z;
    pt2_x_txt.value = objects[selected].pt2.x;
    pt2_y_txt.value = objects[selected].pt2.y;
    pt2_z_txt.value = objects[selected].pt2.z;
    color_r_txt.value = objects[selected].color.r;
    color_g_txt.value = objects[selected].color.g;
    color_b_txt.value = objects[selected].color.b;
end
function ok_btn:action()
    if type(selected)~="number" or selected<=0 or selected>#objects then return end
    objects[selected].pt1.x = pt1_x_txt.value;
    objects[selected].pt1.y = pt1_y_txt.value;
    objects[selected].pt1.z = pt1_z_txt.value;
    objects[selected].pt2.x = pt2_x_txt.value;
    objects[selected].pt2.y = pt2_y_txt.value;
    objects[selected].pt2.z = pt2_z_txt.value;
    select_object(selected)
    scene_onpaint(scene);
    dlg:hide();
end
function cancel_btn:action()
    dlg:hide();
end
function show_dlg()
    init_dlg();
    dlg:popup();
end
function on_lbuttondblclk(scene,flags,x,y)
    show_dlg();
end

local ID_PROPERTY = ID+1;
local ID_ADD = ID+2;
local ID_DRAW = ID+3;
add_menu(
    frm,
    {
        name = "Cube",
        nposition = 2,
        items =
        {
            {id=ID_ADD,name="Add"},
            {id=ID_DRAW,name="Draw"},
            {id=ID_PROPERTY,name="Property"},
        },
    }
);
crt_toolbar(frm,
    {
        bmpname = "toolbar1.bmp",
        nbmps = 3,
        dxButton = 0,
        dyButton = 0,
        dxBitmap = 16,
            dyBitmap = 16,
            buttons = {
                {iBitmap=2,idCommand=ID_ADD,iString="Add",fsState=TBSTATE_ENABLED,
                        fsStyle=BTNS_BUTTON,},
                {iBitmap=3,idCommand=ID_DRAW,iString="Draw",
                        fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
                {iBitmap=5,idCommand=ID_PROPERTY,iString="Property",
                        fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
            },
        }
);
local commands = {};
commands[ID_PROPERTY] = function(scene)
    show_dlg();
end
commands[ID_ADD] = function(scene)
    add_object();
end
commands[ID_DRAW] = function(scene)
    start_pt = true;
    scene_cursor(scene,IDC_CROSS)
end
function on_command(id,scene)
    if type(commands[id])=="function" then
        commands[id](scene);
    end
end
```

查看源代码（apcad/help/src/primer/3.17.6.txt ）

运行主程序（gcad.exe），查看运行结果（点击 Cube-Draw 菜单进入绘制状态，按下左键按住移动后抬起，查看鼠标移动的拖拽线）。



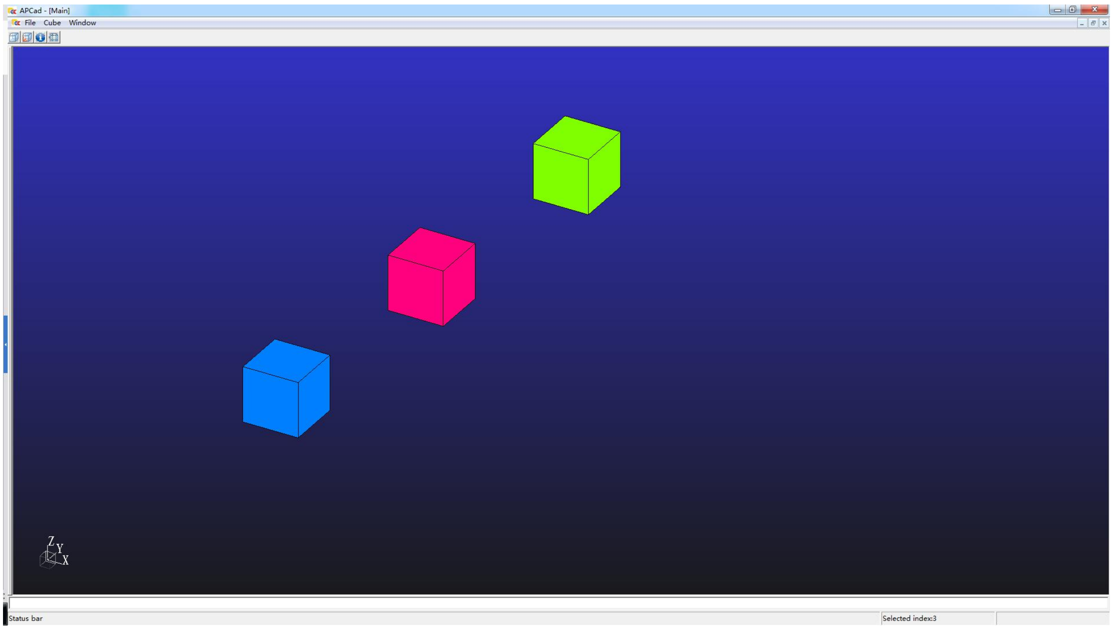
第十八节　设置旋转窗口

运行主程序时，窗口默认显示 XOY 平面，如果需要看到三维效果，需要旋转窗口，现在设置程序默认显示三维窗口。

```
local active_scene = new_child(frm,"Main");
local active_plan = get_scene_t(active_scene);
active_plan.rotate.x = -60;
active_plan.rotate.y = 0;
active_plan.rotate.z = -30;
active_plan.matrix = nil;
set_scene_t(active_scene,active_plan);
```

查看源代码（apcad/help/src/primer/3.18.txt）

运行主程序（gcad.exe），查看运行结果。



第十九节　创建工作平面

默认了三维窗口，还可以创建需要的工作平面（不只是 XOY 平面）。

（一）制作菜单和相应的工具条按钮。

```
add_menu(
    frm,
    {
        name = "Cube",
        nposition = 2,
        items =
        {
            {id=ID_PROPERTY,name="Property"},
            {id=ID_ADD,name="Add"},
            {id=ID_DRAW,name="Draw"},
            {id=ID_PLAN,name="Plan"},
        },
    }
);
crt_toolbar(frm,
    {
        bmpname = "toolbar1.bmp",
        nbmps = 3,
        dxButton = 0,
        dyButton = 0,
        dxBitmap = 16,
        dyBitmap = 16,
        buttons = {
```

```
                    {iBitmap=2,idCommand=ID_ADD,iString="Add",
                            fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
                    {iBitmap=3,idCommand=ID_DRAW,iString="Draw",
                            fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
                    {iBitmap=5,idCommand=ID_PROPERTY,iString="Property",
                            fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
                    {iBitmap=6,idCommand=ID_PLAN,iString="Plan",
                            fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
                },
            }
        );
```

查看源代码（apcad/help/src/primer/3.19.1.txt）

（二）定义创建工作平面函数，以及定义平台消息函数(frm_on_command)，该函数在软件没有打开任何窗口的情况下，用户点击菜单或者工具条按钮时会被平台调用。

```
commands[ID_PLAN] = function(scene)
    clip.show_dlg();
end
function on_command(id,scene)
    if type(commands[id])=="function" then
        commands[id](scene);
    end
end
local frmcommands = {};
frmcommands[ID_PLAN] = function(scene)
    clip.show_dlg();
end
function frm_on_command(id,scene)
    if type(frmcommands[id])=="function" then
        frmcommands[id](scene);
    end
end
```

查看源代码（apcad/help/src/primer/3.19.2.txt）

（三）制作工作平面对话框，并定义初始化对话框、显示对话框以及用户点击 OK、Cancel 按钮时调用的函数。

```
local clip = {};
clip.name_lab = iup.label{title="Name:",size="50x"};
clip.name_txt = iup.text{expand="Horizontal"};
clip.o_lab = iup.label{title="Origin:",size="50x"};
clip.o_x_lab = iup.label{title="X:"};
clip.o_x_txt = iup.text{expand="Horizontal"};
clip.o_y_lab = iup.label{title="Y:"};
clip.o_y_txt = iup.text{expand="Horizontal"};
clip.o_z_lab = iup.label{title="Z:"};
clip.o_z_txt = iup.text{expand="Horizontal"};
clip.x_lab = iup.label{title="X-Axis:",size="50X"};
clip.x_x_lab = iup.label{title="X:"};
clip.x_x_txt = iup.text{expand="Horizontal"};
clip.x_y_lab = iup.label{title="Y:"};
clip.x_y_txt = iup.text{expand="Horizontal"};
clip.x_z_lab = iup.label{title="Z:"};
clip.x_z_txt = iup.text{expand="Horizontal"};
clip.z_lab = iup.label{title="Z-Axis:",size="50x"};
clip.z_x_lab = iup.label{title="X:"};
clip.z_x_txt = iup.text{expand="Horizontal"};
clip.z_y_lab = iup.label{title="Y:"};
clip.z_y_txt = iup.text{expand="Horizontal"};
clip.z_z_lab = iup.label{title="Z:"};
clip.z_z_txt = iup.text{expand="Horizontal"};
clip.ok_btn = iup.button{title="OK",size="100X"};
clip.cancel_btn = iup.button{title="Cancel",size="100X"};
clip.dlg = iup.dialog{
    title = "Work Plan";
    size = "500X100";
    margin = "5X5";
    iup.vbox{
        iup.hbox{clip.name_lab,clip.name_txt};
        iup.hbox{clip.o_lab,clip.o_x_lab,clip.o_x_txt,clip.o_y_lab,clip.o_y_txt,
                    clip.o_z_lab,clip.o_z_txt};
        iup.hbox{clip.x_lab,clip.x_x_lab,clip.x_x_txt,clip.x_y_lab,clip.x_y_txt,
                    clip.x_z_lab,clip.x_z_txt};
        iup.hbox{clip.z_lab,clip.z_x_lab,clip.z_x_txt,clip.z_y_lab,clip.z_y_txt,
                    clip.z_z_lab,clip.z_z_txt};
        iup.hbox{iup.fill{},clip.ok_btn,clip.cancel_btn};
    };
}
function clip.init_dlg()
    local plan = active_scene and get_scene_t(active_scene);
    clip.o_x_txt.value = plan and plan.clip.pt.x or 0;
    clip.o_y_txt.value = plan and plan.clip.pt.y or 0;
    clip.o_z_txt.value = plan and plan.clip.pt.z or 0;
    clip.x_x_txt.value = plan and plan.clip.x.x or 1;
    clip.x_y_txt.value = plan and plan.clip.x.y or 0;
    clip.x_z_txt.value = plan and plan.clip.x.z or 0;
    clip.z_x_txt.value = plan and plan.clip.z.x or 0;
    clip.z_y_txt.value = plan and plan.clip.z.y or 0;
    clip.z_z_txt.value = plan and plan.clip.z.z or 1;
end
function clip.ok_btn:action()
    active_scene = new_child(frm,clip.name_txt.value);
    local plan = get_scene_t(active_scene);
    plan.clip.pt.x = clip.o_x_txt.value;
    plan.clip.pt.y = clip.o_y_txt.value;
    plan.clip.pt.z = clip.o_z_txt.value;
    plan.clip.x.x = clip.x_x_txt.value;
    plan.clip.x.y = clip.x_y_txt.value;
    plan.clip.x.z = clip.x_z_txt.value;
    plan.clip.z.x = clip.z_x_txt.value;
    plan.clip.z.y = clip.z_y_txt.value;
    plan.clip.z.z = clip.z_z_txt.value;
    set_scene_t(active_scene,plan);
    scene_onpaint(active_scene);
    clip.dlg:hide();
end
function clip.cancel_btn:action()
    clip.dlg:hide();
end
function clip.show_dlg()
    clip.init_dlg();
    clip.dlg:popup();
end
```

查看源代码（apcad/help/src/primer/3.19.3.txt ）

使用平台函数（get_scene_t/set_scene_t）设置新窗口的工作平面。

完整的代码如下：

```
package.cpath = "./?53.dll;./?.dll";

function frmclose()
    os.exit();
end
statusbar_set_parts(frm,{200,200})

local active_scene = new_child(frm,"Main");
local active_plan = get_scene_t(active_scene);
active_plan.rotate.x = -60;
active_plan.rotate.y = 0;
active_plan.rotate.z = -30;
active_plan.matrix = nil;
set_scene_t(active_scene,active_plan);

function get_shape(pt1,pt2,color)
    local outer_pts = {
        {color.r,color.g,color.b,1,1,pt1.x,pt1.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt1.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt2.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt2.y,pt1.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt1.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt1.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt1.x,pt2.y,pt2.z};
        {color.r,color.g,color.b,1,1,pt2.x,pt2.y,pt2.z};
    };
    local lines_pts = {
        {0,0,0,1,1,pt1.x,pt1.y,pt1.z};
        {0,0,0,1,1,pt2.x,pt1.y,pt1.z};
        {0,0,0,1,1,pt1.x,pt2.y,pt1.z};
        {0,0,0,1,1,pt2.x,pt2.y,pt1.z};
        {0,0,0,1,1,pt1.x,pt1.y,pt2.z};
        {0,0,0,1,1,pt2.x,pt1.y,pt2.z};
        {0,0,0,1,1,pt1.x,pt2.y,pt2.z};
        {0,0,0,1,1,pt2.x,pt2.y,pt2.z};
    };
    local shape = {
        surfaces = {
            {
                points = lines_pts;
                lines = {{1,2},{1,3},{2,4},{3,4};{5,6},{5,7},{6,8},{7,8};{1,5},{2,6},{3,7},{4,8}};
            };
            {
                points = outer_pts;
                outer = {1,3,4,2};
            };
            {
                points = outer_pts;
                outer = {5,6,8,7};
            };
            {
                points = outer_pts;
                outer = {1,2,6,5};
            };
            {
                points = outer_pts;
                outer = {1,5,7,3};
            };
            {
                points = outer_pts;
                outer = {2,4,8,6};
            };
            {
                points = outer_pts;
                outer = {3,7,8,4};
            };
        };
    };
    return shape;
end

local objects = {};
function add_object(pt1,pt2)
    local n = #objects;
    pt1 = pt1 or {x=5000*n,y=5000*n,z=5000*n};
    pt2 = pt2 or {x=5000*n+3000,y=5000*n+3000,z=5000*n+3000};
    local color = {r=0,g=0.5,b=1};
    local shape = get_shape(pt1,pt2,color);
    local glname,gllist = n+1,makelist(active_scene,shape);
    objects[n+1] = {pt1=pt1,pt2=pt2,color=color,glname=glname,gllist=gllist};
    scene_onpaint(active_scene);
end

local selected = nil;
function select_object(i)
    objects[i].color.r = (objects[i].color.r-0.5)%1.5;
    objects[i].color.g = (objects[i].color.g-0.5)%1.5;
    objects[i].color.b = (objects[i].color.b-0.5)%1.5;
    local object = get_shape(objects[i].pt1,objects[i].pt2,objects[i].color);
    local glname,gllist = i,makelist(active_scene,object);
    objects[i].gllist = gllist;
    scene_onpaint(active_scene);
    selected = i;
end
function select_main(i)
    select_object(i)
    statusbar_set_text(frm,1,"Selected index:"..i);
end

function snap(scene,x,y)
```

```
    for i,v in pairs(objects) do
        local shape = get_shape(v.pt1,v.pt2,v.color);
        for i,v in pairs(shape.surfaces[1].points) do
            local pt = {world_2_client(scene,v[6],v[7],v[8])};
            if math.abs(pt[1]-x)<=15 and math.abs(pt[2]-y)<=15 then
                return {x=v[6],y=v[7],z=v[8]};
            end
        end
    end
    local x,y,z = client_2_world(scene,x,y);
    return {x=x,y=y,z=z};
end

function get_drag_shape(pt1,pt2)
    local shape = {
        surfaces = {
            {
                points = {
                    {1,1,1,1,1,pt1.x,pt1.y,pt1.z};
                    {1,1,1,1,1,pt2.x,pt2.y,pt2.z};
                };
                lines = {{1,2}};
            };
        };
    };
    return shape;
end
local gl = require "luaext.gl"
local drag_line = nil;
function del_drag_line()
    if drag_line then
          gl.glDeleteLists(drag_line);
          drag_line = nil;
    end
end
function set_drag_line(pt1,pt2)
    del_drag_line();
    drag_line = makelist(active_scene,get_drag_shape(pt1,pt2));
end
function render_drags()
    if drag_line then
        gl.glCallList(drag_line);
    end
end
```

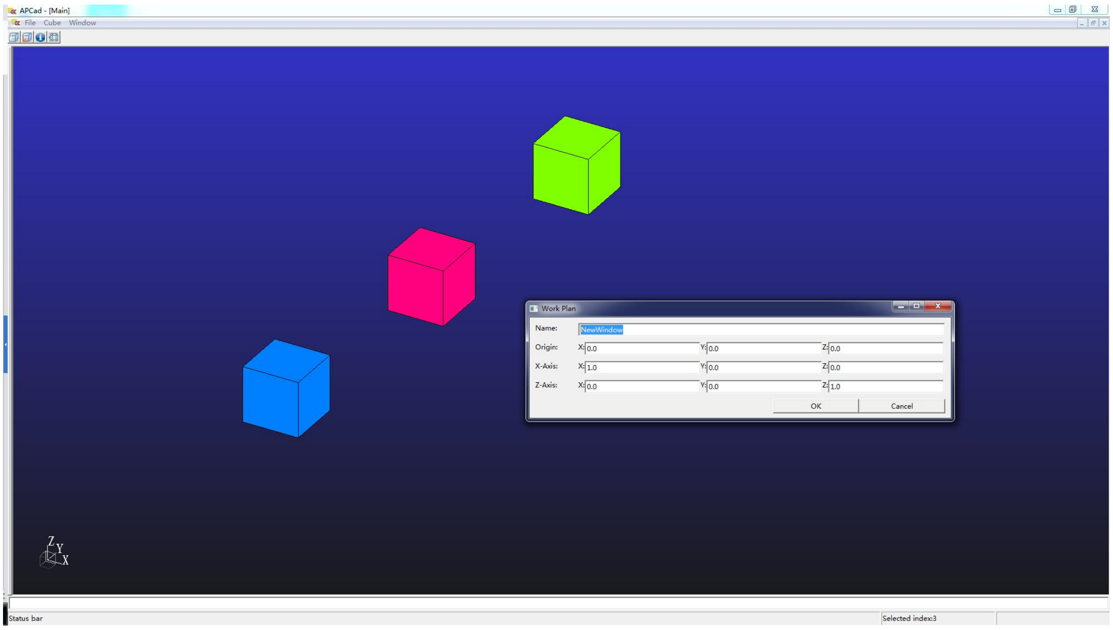
```
function on_paint(scene)
	del_drag_line();
	active_scene = scene;
end

local start_pt = nil;
function on_lbuttondown(scene,flags,x,y)
	if start_pt then
		start_pt = snap(scene,x,y);
	else
		scene_select(scene,x,y,1,1,1);
	end
end
function on_mousemove(scene,flags,x,y)
	if type(start_pt)=="table" then
		local pt = snap(scene,x,y);
		draw_drag(scene);
		set_drag_line(start_pt,pt);
		draw_drag(scene);
	end
end
function on_lbuttonup(scene,flags,x,y)
	if type(start_pt)=="table" then
		local pt = snap(scene,x,y);
		add_object(start_pt,pt);
		scene_cursor(scene,IDC_ARROW)
		start_pt = nil;
	end
end

function render_objs()
	for i,v in ipairs(objects) do
		gl.glLoadName(v.glname);
		gl.glCallList(v.gllist);
	end
end

local iup = require"iuplua"
local pt1_lab = iup.label{title="Point1:",size="50x"};
local pt1_x_lab = iup.label{title="X:"};
local pt1_x_txt = iup.text{expand="Horizontal"};
local pt1_y_lab = iup.label{title="Y:"};
local pt1_y_txt = iup.text{expand="Horizontal"};
local pt1_z_lab = iup.label{title="Z:"};
local pt1_z_txt = iup.text{expand="Horizontal"};
local pt2_lab = iup.label{title="Point2:",size="50X"};
local pt2_x_lab = iup.label{title="X:"};
local pt2_x_txt = iup.text{expand="Horizontal"};
local pt2_y_lab = iup.label{title="Y:"};
local pt2_y_txt = iup.text{expand="Horizontal"};
local pt2_z_lab = iup.label{title="Z:"};
local pt2_z_txt = iup.text{expand="Horizontal"};
local color_lab = iup.label{title="Color:",size="50X"};
local color_r_lab = iup.label{title="R:"};
local color_r_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local color_g_lab = iup.label{title="G:"};
local color_g_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local color_b_lab = iup.label{title="B:"};
local color_b_txt = iup.text{expand="Horizontal",readonly="Yes",bgcolor="192 192 192"};
local ok_btn = iup.button{title="OK",size="100X"};
local cancel_btn = iup.button{title="Cancel",size="100X"};
local dlg = iup.dialog{
    title = "Property";
    size = "500X100";
    margin = "5X5";
    iup.vbox{
        iup.hbox{pt1_lab,pt1_x_lab,pt1_x_txt,pt1_y_lab,pt1_y_txt,pt1_z_lab,pt1_z_txt};
        iup.hbox{pt2_lab,pt2_x_lab,pt2_x_txt,pt2_y_lab,pt2_y_txt,pt2_z_lab,pt2_z_txt};

    iup.hbox{color_lab,color_r_lab,color_r_txt,color_g_lab,color_g_txt,color_b_lab,color_b
_txt};
        iup.hbox{iup.fill{},ok_btn,cancel_btn};
    };
}
function init_dlg()
    if type(selected)~="number" or selected<=0 or selected>#objects then return end
    pt1_x_txt.value = objects[selected].pt1.x;
    pt1_y_txt.value = objects[selected].pt1.y;
    pt1_z_txt.value = objects[selected].pt1.z;
    pt2_x_txt.value = objects[selected].pt2.x;
    pt2_y_txt.value = objects[selected].pt2.y;
    pt2_z_txt.value = objects[selected].pt2.z;
    color_r_txt.value = objects[selected].color.r;
    color_g_txt.value = objects[selected].color.g;
    color_b_txt.value = objects[selected].color.b;
end
function ok_btn:action()
    if type(selected)~="number" or selected<=0 or selected>#objects then return end
    objects[selected].pt1.x = pt1_x_txt.value;
    objects[selected].pt1.y = pt1_y_txt.value;
    objects[selected].pt1.z = pt1_z_txt.value;
    objects[selected].pt2.x = pt2_x_txt.value;
    objects[selected].pt2.y = pt2_y_txt.value;
    objects[selected].pt2.z = pt2_z_txt.value;
    select_object(selected)
    scene_onpaint(active_scene);
    dlg:hide();
end
function cancel_btn:action()
    dlg:hide();
end
function show_dlg()
    init_dlg();
    dlg:popup();
end
function on_lbuttondblclk(scene,flags,x,y)
    show_dlg();
end

local clip = {};
clip.name_lab = iup.label{title="Name:",size="50x"};
clip.name_txt = iup.text{expand="Horizontal"};
clip.o_lab = iup.label{title="Origin:",size="50x"};
clip.o_x_lab = iup.label{title="X:"};
clip.o_x_txt = iup.text{expand="Horizontal"};
clip.o_y_lab = iup.label{title="Y:"};
clip.o_y_txt = iup.text{expand="Horizontal"};
clip.o_z_lab = iup.label{title="Z:"};
clip.o_z_txt = iup.text{expand="Horizontal"};
clip.x_lab = iup.label{title="X-Axis:",size="50X"};
clip.x_x_lab = iup.label{title="X:"};
clip.x_x_txt = iup.text{expand="Horizontal"};
clip.x_y_lab = iup.label{title="Y:"};
clip.x_y_txt = iup.text{expand="Horizontal"};
clip.x_z_lab = iup.label{title="Z:"};
clip.x_z_txt = iup.text{expand="Horizontal"};
clip.z_lab = iup.label{title="Z-Axis:",size="50x"};
clip.z_x_lab = iup.label{title="X:"};
clip.z_x_txt = iup.text{expand="Horizontal"};
clip.z_y_lab = iup.label{title="Y:"};
clip.z_y_txt = iup.text{expand="Horizontal"};
clip.z_z_lab = iup.label{title="Z:"};
clip.z_z_txt = iup.text{expand="Horizontal"};
clip.ok_btn = iup.button{title="OK",size="100X"};
clip.cancel_btn = iup.button{title="Cancel",size="100X"};
clip.dlg = iup.dialog{
    title = "Work Plan";
    size = "500X100";
    margin = "5X5";
    iup.vbox{
        iup.hbox{clip.name_lab,clip.name_txt};

    iup.hbox{clip.o_lab,clip.o_x_lab,clip.o_x_txt,clip.o_y_lab,clip.o_y_txt,clip.o_z_lab,clip.o
_z_txt};

    iup.hbox{clip.x_lab,clip.x_x_lab,clip.x_x_txt,clip.x_y_lab,clip.x_y_txt,clip.x_z_lab,clip.x_
z_txt};

    iup.hbox{clip.z_lab,clip.z_x_lab,clip.z_x_txt,clip.z_y_lab,clip.z_y_txt,clip.z_z_lab,clip.z_z
_txt};
        iup.hbox{iup.fill{},clip.ok_btn,clip.cancel_btn};
    };
}
function clip.init_dlg()
    local plan = active_scene and get_scene_t(active_scene);
    clip.o_x_txt.value = plan and plan.clip.pt.x or 0;
    clip.o_y_txt.value = plan and plan.clip.pt.y or 0;
    clip.o_z_txt.value = plan and plan.clip.pt.z or 0;
    clip.x_x_txt.value = plan and plan.clip.x.x or 1;
    clip.x_y_txt.value = plan and plan.clip.x.y or 0;
    clip.x_z_txt.value = plan and plan.clip.x.z or 0;
    clip.z_x_txt.value = plan and plan.clip.z.x or 0;
    clip.z_y_txt.value = plan and plan.clip.z.y or 0;
    clip.z_z_txt.value = plan and plan.clip.z.z or 1;
end
function clip.ok_btn:action()
    active_scene = new_child(frm,clip.name_txt.value);
    local plan = get_scene_t(active_scene);
    plan.clip.pt.x = clip.o_x_txt.value;
    plan.clip.pt.y = clip.o_y_txt.value;
    plan.clip.pt.z = clip.o_z_txt.value;
    plan.clip.x.x = clip.x_x_txt.value;
    plan.clip.x.y = clip.x_y_txt.value;
    plan.clip.x.z = clip.x_z_txt.value;
    plan.clip.z.x = clip.z_x_txt.value;
    plan.clip.z.y = clip.z_y_txt.value;
```

```
        plan.clip.z.z = clip.z_z_txt.value;
        set_scene_t(active_scene,plan);
        scene_onpaint(active_scene);
        clip.dlg:hide();
end
function clip.cancel_btn:action()
        clip.dlg:hide();
end
function clip.show_dlg()
        clip.init_dlg();
        clip.dlg:popup();
end



local ID_PROPERTY = ID+1;
local ID_ADD = ID+2;
local ID_DRAW = ID+3;
local ID_PLAN = ID+4;
add_menu(
        frm,
        {
                name = "Cube",
                nposition = 2,
                items =
                {
                        {id=ID_PROPERTY,name="Property"},
                        {id=ID_ADD,name="Add"},
                        {id=ID_DRAW,name="Draw"},
                        {id=ID_PLAN,name="Plan"},
                },
        }
);
crt_toolbar(frm,
        {
                bmpname = "toolbar1.bmp",
                nbmps = 3,
                dxButton = 0,
                dyButton = 0,
                dxBitmap = 16,
                dyBitmap = 16,
                buttons = {

        {iBitmap=2,idCommand=ID_ADD,iString="Add",fsState=TBSTATE_ENABLED,fsStyle=BTN
S_BUTTON,},
	{iBitmap=3,idCommand=ID_DRAW,iString="Draw",fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},

	{iBitmap=5,idCommand=ID_PROPERTY,iString="Property",fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},

	{iBitmap=6,idCommand=ID_PLAN,iString="Plan",fsState=TBSTATE_ENABLED,fsStyle=BTNS_BUTTON,},
		},
	}
);
local commands = {};
commands[ID_PROPERTY] = function(scene)
	show_dlg();
end
commands[ID_ADD] = function(scene)
	add_object();
end
commands[ID_DRAW] = function(scene)
	start_pt = true;
	scene_cursor(scene,IDC_CROSS)
end
commands[ID_PLAN] = function(scene)
	clip.show_dlg();
end
function on_command(id,scene)
	if type(commands[id])=="function" then
		commands[id](scene);
	end
end
local frmcommands = {};
frmcommands[ID_PLAN] = function(scene)
	clip.show_dlg();
end
function frm_on_command(id,scene)
	if type(frmcommands[id])=="function" then
		frmcommands[id](scene);
	end
end
```

查看源代码（apcad/help/src/primer/3.19.4.txt ）

运行主程序（gcad.exe），查看运行结果（点击菜单 Cube-Plan 或其对应的工

具条按钮）。



Work Plan

Name:	NewWindow		
Origin:	X: 0.0	Y: 0.0	Z: 0.0
X-Axis:	X: 1.0	Y: 0.0	Z: 0.0
Z-Axis:	X: 0.0	Y: 0.0	Z: 1.0