

管理幾何圖形群組、 元件及外觀

從本章開始，你要開始管理你的幾何圖形物件。為了方便建立複雜的模型，你必須學習利用圖層，在設計進行中隱藏部份圖形，讓你容易專注在目前所工作的圖形上。在本章中你也會學習到群組化，以便重複利用一些圖形群組。

元件是很重要物件，元件可以賦予不同的屬性，而且可以將元件定義儲存在專用的檔案中，供其它模型共享。最後，你將在本章學習如何處理外觀，及組合材料的顏色及材質，讓你所繪製的物件看起來更真實。



5-1 圖層

圖層（Layers）用來顯示或隱藏作業中不相關的幾何圖形。將來你可能會將許多幾何圖形組合成你設計的模型，當繪圖視窗中的幾何圖形會愈來愈多，你可以將相關的幾何圖形放在同一圖層。有時候你會隱藏部份圖層的幾何圖形，以便於專心在特定圖層的幾何圖形。如果你做建築設計，可以分別為各樓層建立一個圖層，管理各樓層的平面及配置。如果你設計產品，可以將部件分別放置在不同的圖層，甚至將輔助線全部放置在同一圖層，需要使用時顯示圖層，不需要時將它隱藏。

在 SketchUp 中，你可以從 [視窗] 功能表中打開 [圖層] 視窗。在「圖層」視窗中，我們可以新增或刪除圖層、設定圖層顯示或隱藏，以及設定辨識的顏色。預設的圖層是 Layer0。你可以按一下圖層視窗中的「+」號，來新增圖層並為圖層命名。新增的圖層不用時可以刪除，但是 Layer0 圖層是無法刪除的。

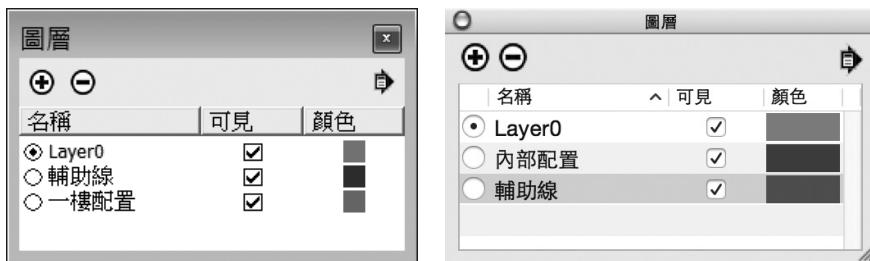


圖 5-1 圖層視窗（Windows 系統和 Mac OS 系統）

新增的圖層預設是可見的，「可見」欄中的核取方塊是勾選的。你可以取消「可見」欄中的核取方塊，將圖層隱藏起來。你可以同時讓幾個圖層顯示或隱藏，但只能有一個圖層是作用中圖層。左側的選項按鈕用來切換「作用中圖層」，你在繪圖視窗中所繪製的圖形，都會座落在作用中圖層中。所以，當你要在另一個圖層繪圖時，必須先切換圖層，將你要繪圖的圖層切換成為作用中圖層。

SketchUp 會預設給每一個圖層不同的顏色，你可以按一下右欄的顏色色塊，然後在顯示的「顏色」對話框中選取變更的顏色，將預設的顏色換掉。在圖層視窗右上

方的「詳細設定」功能表，你可以選擇「各圖層具有不同顏色」項目，讓圖層中所有的幾何圖形都暫時換上圖層的代表色，以便於我們辨識繪圖視窗中的幾何圖形各是屬於哪個圖層。

在程式中，我們可以透過方法來控制圖層的顯示或隱藏。SketchUp 放置所有圖層在 Layers 容器中，它是一個圖層陣列。當我們要處理圖層時，必須先呼叫 Model 類別的 Layers 方法。你可以再利用 Layer 提供的方法，取得個別 Layer 物件資訊。範例如下：

```
layer_array = Sketchup.active_model.layers
first_layer = layer_array[0]
first_layer.name
```

Layer0 是預設的圖層。我們呼叫 Model 的 layers 方法取得容器內所有圖層，同時將它們儲存在 layer_array，layer_array 陣列的第一個元素（指標為 0）便是 Layer0 物件。我們可以呼叫 name 方法來取得 Layer0 物件的名稱。如果不用變數替換，直接透過陣列表示法來呼叫方法也行。

```
layer_array[0].name
```

Layers 類別提供 add 方法用來建立圖層，Layer 物件建立之後便儲存在圖層容器。當我們要取出圖層容器陣列中的 Layer 物件，可以用陣列指標，也可以使用元素內容。如果要計算模型中有多少個圖層，可以使用相同類別提供的 count 方法。如果要刪除沒有使用的圖層，則呼叫 purge_unused 方法。如果要切換作用中圖層，得使用 Model 類別的 active_layer= 方法。在進行下列練習時，請打開圖層視窗。

```
mod = Sketchup.active_model
layer_array = mod.layers
new_layer = layer_array.add " 輔助線 "
layer_array.count
mod.active_layer=new_layer
```

```
Ruby Console
> mod = Sketchup.active_model
#<Sketchup::Model:0x007ffd036b57f8>
> layer_array = mod.layers
#<Sketchup::Layers:0x007ffd036420c8>
> new_layer = layer_array.add "輔助線"
#<Sketchup::Layer:0x007ffd0364a958>
> layer_array.count
2
> mod.active_layer= new_layer
#<Sketchup::Layer:0x007ffd0364a958>
```

圖 5-2 建立圖層及切換使用中圖層

一旦我們建立圖層之後，便可以呼叫 Layer 類別的 name 方法和 visible= 方法，分別來指定圖層名稱及設定可見或隱藏，如果要判斷圖層是否為可見狀態，呼叫 visible? 方法便會傳回 true/false。若傳回 true，表示圖層是可見的。

```
new_layer.visible = false
UI.messagebox(new_layer.visible?)
```

圖層可以用來管理較複雜的設計，但是僅能用來顯示或隱藏部份的幾何圖形，無法移動圖層內的所有物件，但是將幾何圖形群組化便能一起移動。圖層和群組是兩個不同的物件，如果你要進一步管理你建立的物件，就必須建立群組。

5-2 群組

我們可以將建立的幾何圖形，包括曲線、圓形、多邊形、矩形表面等群組起來，形成群組集合 (Groups)。群組是一個階層架構，一層又一層，我們可以把兩個群組集合起來變成一個上層群組，於是就有了父群組及子群組。當我們操作群組，處理結果將影響它所有的實體成員。

5-2-1 建立群組

Entities 類別的 `add_group` 方法專用來建立群組物件，並將幾何實體儲存於群組中。使用 `add_group` 方法必須以實體物件或實體陣列為參數，這些都是群組對象。在 SketchUp 中，你無法一次移動、複製或旋轉部份幾何圖形，因為有些幾何圖形並非實體，它可能是一個陣列，因此，必須將這些幾何圖形建立成 Group 物件。這些表面及邊線成為 Group 物件之後，便無法在繪圖視窗中群組的狀態下進行個別的編輯，但還是可以透過程式碼來處理單一個別實體。底下範例，我用一個立方體的例子來說明 Group 物件。

```
ent = Sketchup.active_model.entities
face = ent.add_face [0,0,0], [10,0,0], [10,10,0], [0,10,0]
face.reverse!
face.pushpull 10
test_group = ent.add_group face.all_connected
```

對 Face 物件轉形是較為容易的，由於立方體不是單一實體物件，如果你要對立方體轉形，最好先把它建立為 Group 物件之後才進行轉形。上述範例中 `test_group` 是代表立方體的 Group 物件，可以對它進行的轉形處理包括移動、旋轉及按比例縮放。

在第四章程式碼 4-11 中，我們曾建立切割線，然後建立 Transformation 物件移動切割線，將切割線拉起形成三角形屋頂。也曾在程式碼 4-13 中，直接對所有連接的物件進行轉形。現在，你可以試試看在 `test_group` 的表面上，也建立一條切割線然後移動它，是否可以拉起形成屋頂？結果是不可以的。直線單獨被提起。如果你連同立方體和直線一併做為群組物件，你將提起整個 Group 物件。

Group 物件儲存實體物件在群組實體容器中，我們用 `entities` 方法來取用群組容器，然後建立邊線、表面或其它實體物件，這些物件都會儲存到群組容器中。下列兩行程式碼就是這個例子。

```
ent_g = test_group.entities
circle = ent_g.add_circle [5,5,10], [0,0,1], 2.5
```

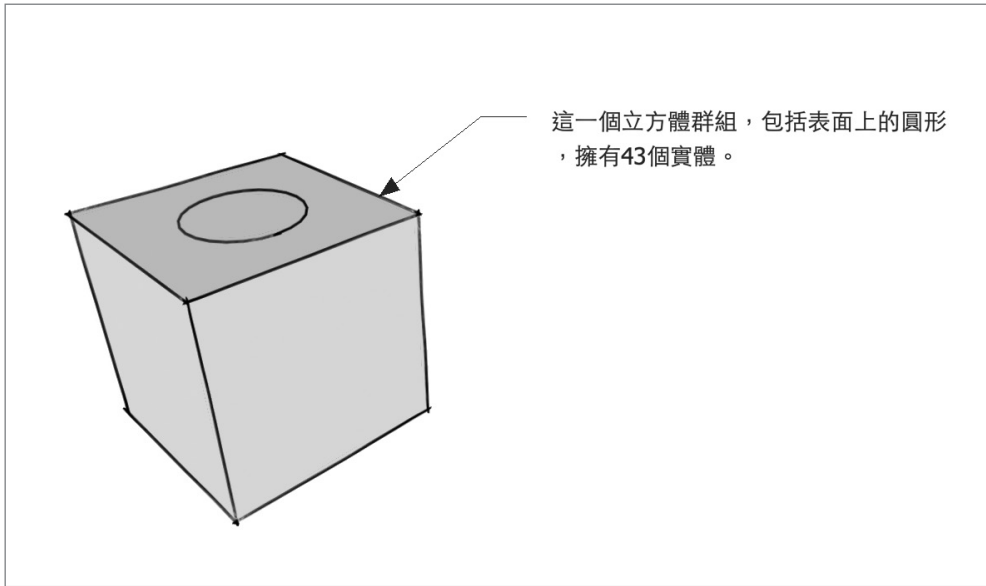
```
Ruby Console
> ent = Sketchup.active_model.entities
#<Sketchup::Entities:0x007fa2c59dda40>
> face = ent.add_face [0,0,0], [10,0,0], [10,10,0], [0,10,0]
#<Sketchup::Face:0x007fa2c59d6f88>
> face.reverse!
#<Sketchup::Face:0x007fa2c59d6f88>
> face.pushpull 10

> test_group = ent.add_group face.all_connected
#<Sketchup::Group:0x007fa2c59d6010>
> ent_g = test_group.entities
#<Sketchup::Entities:0x007fa2c59d4aa8>
> circle = ent_g.add_circle [5,5,10], [0,0,1], 2.5
[#<Sketchup::Edge:0x007fa2c59cfc38>, #<Sketchup::Edge:
0x007fa2c59cfbe8>, #<Sketchup::Edge:0x007fa2c59cfbc0>,
#<Sketchup::Edge:0x007fa2c59cfb98>, #<Sketchup::Edge:
0x007fa2c59cfb70>, #<Sketchup::Edge:0x007fa2c59cfb48>,
#<Sketchup::Edge:0x007fa2c59cfb20>, #<Sketchup::Edge:
0x007fa2c59cfaf8>, #<Sketchup::Edge:0x007fa2c59cfad0>,
#<Sketchup::Edge:0x007fa2c59cfaa8>, #<Sketchup::Edge:
0x007fa2c59cfa58>, #<Sketchup::Edge:0x007fa2c59cfa30>,
#<Sketchup::Edge:0x007fa2c59cfa08>, #<Sketchup::Edge:
0x007fa2c59cf9e0>, #<Sketchup::Edge:0x007fa2c59cf990>,
#<Sketchup::Edge:0x007fa2c59cf968>, #<Sketchup::Edge:
0x007fa2c59cf940>, #<Sketchup::Edge:0x007fa2c59cf8c8>,
#<Sketchup::Edge:0x007fa2c59cf8a0>, #<Sketchup::Edge:
0x007fa2c59cf878>, #<Sketchup::Edge:0x007fa2c59cf850>,
#<Sketchup::Edge:0x007fa2c59cf828>, #<Sketchup::Edge:
0x007fa2c59cf7d8>, #<Sketchup::Edge:0x007fa2c59cf7b0>]
```

圖 5-3 建立群組容器增添幾何物件

ent_g 是 Entities 容器，當你使用 entities 方法取用時，它有 18 個實體。建立圓形邊線到容器之後，透過 size 方法取得容器大小傳回 43 個實體。

```
ent_g.size
```



我們可以對 Group 物件命名、增加描述、鎖定、分解及判斷它是否被刪除。方法如下：

- ◆ name= - 用來為 Group 物件命名，大綱視窗中會顯示這個命名結果。
- ◆ description= - 新增 Group 物件的說明。
- ◆ locked= - 設定 Group 物件是否鎖定。鎖定後便不能在繪圖視窗中分解及改變群組的大小、形狀和位置等。
- ◆ explode - 將 Group 物件分解。
- ◆ deleted? - 判斷 Group 物件是否被刪除。

接續圖 5-3，請在 Ruby 控制台一行一行輸入下列程式：

```
test_group.name = "立方體"  
test_group.description = "這是我的第一個群組"  
test_group.locked= true  
test_group.explode  
test_group.deleted?
```

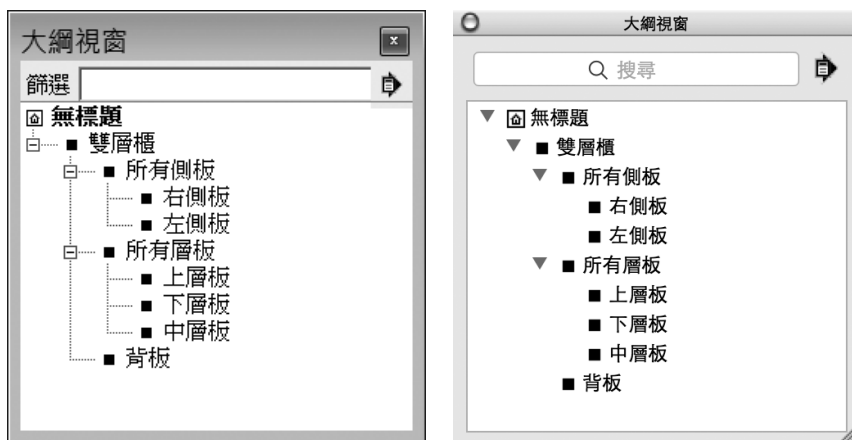


圖 5-4 從大綱視窗查看群組的上下層關係 (Windows 系統和 Mac OS 系統)

打開 [視窗] 功能表裡的「大綱視窗」。我們可以在這裡看到檔案中的所有群組，從這裡可以看出 Group 物件的上下層關係。大綱視窗中的大綱可以展開或折疊 Group 項目，按一下 Group 項目則表示選取這個 Group 物件。最上層的「無標題」用來顯示檔案名稱，如果你將檔案儲存，便會顯示檔案名稱。

5-2-2 群組的複製與轉形

Group 物件可以複製，使用 copy 方法可以複製一個和原始 Group 物件一模一樣的群組，然後再賦予一個新的名稱。在模型中，重覆利用群組是很常見的。將模型中的組件做成 Group 物件，便可以移動、複製或改變位置、角度，提昇你的工作效率。Group 類別提供下述方法，處理 Group 物件的複製與轉形。

- ◆ copy — 依照調用此方法的 Group 物件，建立新的 Group 物件。
- ◆ move! — 套用 Transformation 物件的轉形功能，依指定的座標轉形。
- ◆ transform! — 套用 Transformation 物件的轉形功能，依指定的向量轉形。

兩個和轉形有關的方法非常相似，容易混淆。以移動 Group 物件為例，若要將 Group 移動到指定的座標，應使用 move! 方法。若要將 Group 物件向某一方向移動某一距離，則應使用 transform! 方法，此時 Transformation 物件所指定的參數就