

THE NVL MAKER

新手教程

(六)

地圖與養成面板的製
作

一、必備的基礎知識

因為這是 THE NVL Maker，而不是吉裡吉裡/KAG 的教程，所以之前的新手教程裡沒有提供多少吉裡吉裡/KAG 基礎知識。而且凡是講到基礎知識的部分，即使看不懂，也都可以無視……反正只要理解了什麼是標籤和跳轉，再加上會製作選擇按鈕，自製一個多路線的電子小說就完全沒有問題了。

不過如果想要深入研究的話，有些東西還是必須要說的。

很多圖形化遊戲製作軟件都喜歡用“不需要程序知識”之類的做宣傳。

如果說程序知識指的是對 C#或者 JAVA 諸如此類的代碼有瞭解，那當然是不需要了。

但是，對遊戲製作者來說，最重要的從來都不是對代碼知識的瞭解，而是“邏輯”。

你需要嚴密地考慮當遊戲進行到一個步驟，玩家一共有多少種選擇，這每種選擇又會導致怎樣的下一步。

這點來說，一定的程序知識還是需要的。如果能嘗試理解以下的概念的話……

（0）“層”和“頁”

KAG，或者說吉裡吉裡的畫面，是全部由所謂的「層」疊起來組成的（和 Photoshop 等軟件的層差不多）。

KAG 的每一層又可以分成兩部分，那就是被叫做表頁 fore、裡頁 back 的東西。所有表頁的內容組成了畫面上顯示的東西，而裡頁的內容，畫面上是看不到的。裡頁主要是在使用 [trans]（畫面切換效果）時，用於預載想要顯示的內容的。（轉自 KAG 官方文檔漢化版）

層分成“消息層”和“圖片層”兩種。前者用於顯示文字、文字連接[link]和按鈕，後者用於顯示背景圖片、角色圖片、界面底板圖片等等。

（1）腳本&指令

用一行一行的指令（TAG）來描述遊戲內容的文本文檔。遊戲開始時，會首先執行名叫 first.ks 的腳本，然後再根據 first.ks 裡的指令，“跳轉”或者“呼叫”其他的腳本，以讓遊戲繼續進行。

而指令則是以@開頭，或者是由括號[]包裹起來的內容。每一個指令都讓遊戲做一件事。例如，顯示一張圖片，播放一段音樂，顯示一個按鈕，等等。

（2）變量（變數）

記錄了數字、字符串等內容。可以對它進行操作、顯示它記錄的值，或者根據它記錄的值，來判斷接下來應該執行什麼指令（條件分歧）。

（3）跳轉&標籤

跳轉，中斷一個腳本段落的執行，跳到另外一個段落繼續。（不會返回原來的腳本）。

標籤：腳本中的一個“書簽”，可作為跳轉到的位置標記。

效果：A->B

（4）呼叫

在腳本 A 裡，假如執行了呼叫指令[call storage="腳本 B.ks"]，那麼就會中斷腳本 A，而繼續執行腳本 B，一直到腳本 B 執行到返回指令[return]。

遇到[return]以後，將會停止 B 的執行，返回 A，從剛剛 A 被中斷的地方開始接著往下走。

效果：A->B->A

（5）條件分歧

用[if]和[endif]包裹起來的一段指令，只有當滿足某個條件（例如某個變量大於或者小於 10）的時候才會被執行，不滿足條件的時候會被忽略。

（6）等待玩家選擇

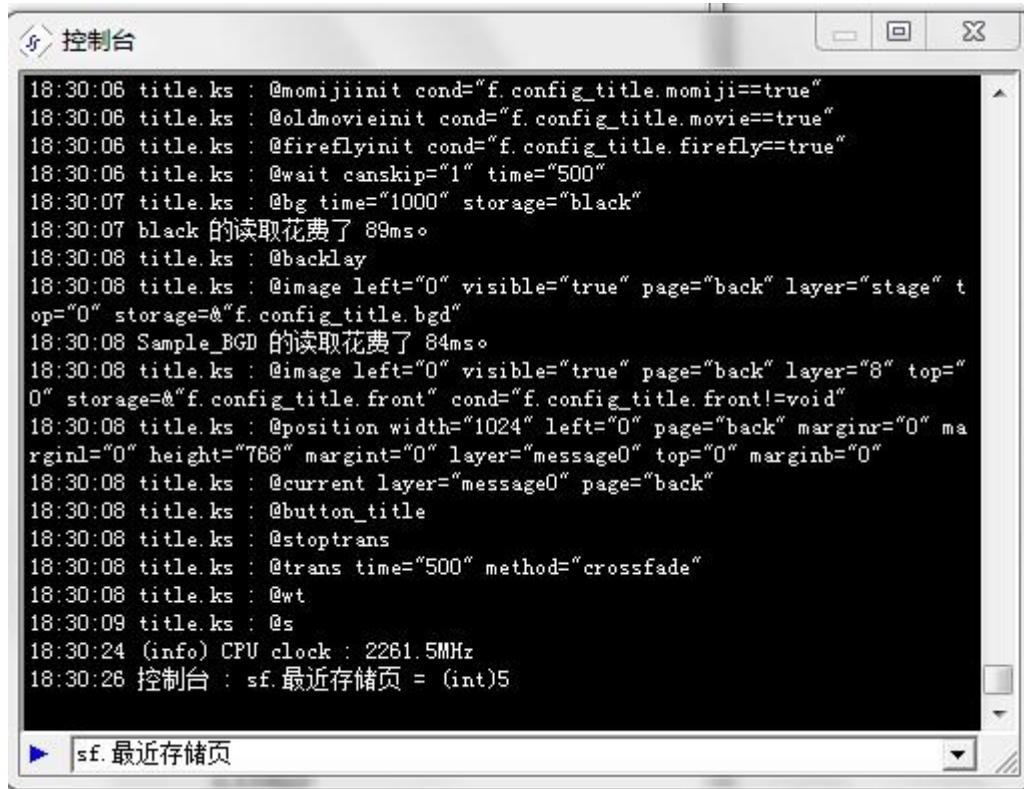
中斷腳本的執行，一直到玩家點下選項按鈕、按鈕、鼠標左鍵、右鍵等等。

中斷腳本時請一定保證畫面上有玩家可以操作的東西，否則的話遊戲就會卡住。

（7）調試窗口

點下遊戲窗口上的 調試->控制台，可以打開的黑色窗口，會顯示出遊戲當前的執行狀況。也就是“腳本走到了什麼地方”。

你可以試著在下面輸入變量的名字等，它會告訴你變量現在的值。（你也可以對這個變量進行操作……相當於使用了遊戲修改器調整遊戲數值 **www**）



```
18:30:06 title.ks : @momijiinit cond="f.config_title.momiji==true"
18:30:06 title.ks : @oldmovieinit cond="f.config_title.movie==true"
18:30:06 title.ks : @fireflyinit cond="f.config_title.firefly==true"
18:30:06 title.ks : @wait canskip="1" time="500"
18:30:07 title.ks : @bg time="1000" storage="black"
18:30:07 black 的读取花费了 89ms。
18:30:08 title.ks : @backlay
18:30:08 title.ks : @image left="0" visible="true" page="back" layer="stage" top="0" storage="@f.config_title.bgd"
18:30:08 Sample_BGD 的读取花费了 84ms。
18:30:08 title.ks : @image left="0" visible="true" page="back" layer="8" top="0" storage="@f.config_title.front" cond="f.config_title.front!=void"
18:30:08 title.ks : @position width="1024" left="0" page="back" marginr="0" marginl="0" height="768" margint="0" layer="message0" top="0" marginb="0"
18:30:08 title.ks : @current layer="message0" page="back"
18:30:08 title.ks : @button_title
18:30:08 title.ks : @stoptrans
18:30:08 title.ks : @trans time="500" method="crossfade"
18:30:08 title.ks : @wt
18:30:09 title.ks : @s
18:30:24 (info) CPU clock : 2261.5MHz
18:30:26 控制台 : sf.最近存储页 = (int)5
```

(8) 宏 (macro)

由多個指令簡化而成的一行指令。每當使用時，則相當於執行了這多個指令的效果。
使用宏時需要先使用[macro][endmacro]定義並註冊。
註冊的方法是讓系統載入（通常是呼叫）包含有宏定義的腳本。

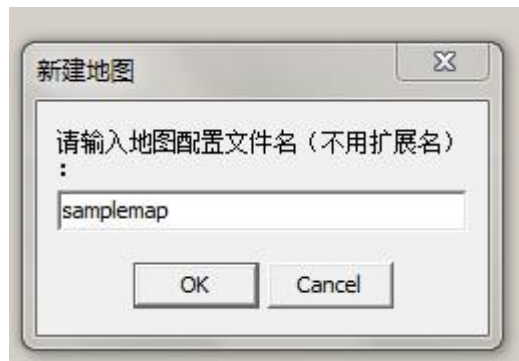
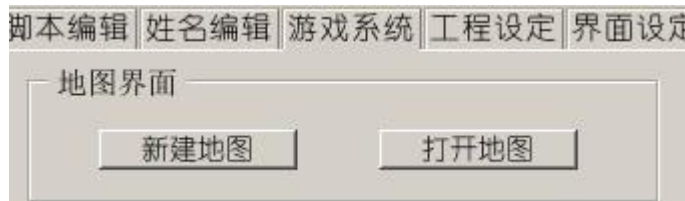
(9) 插件 (plug-in)

由 TJS 語言寫成的輔助腳本，通常也包含宏。用於擴展吉裡吉裡的功能。
使用對應插件的話，即使完全沒有 TJS 知識，也可以達成想要的效果。
例如遊戲裡的櫻花、紅葉、螢火蟲、舊電影等等畫面效果。
使用插件之前同樣需要讓系統載入插件腳本。

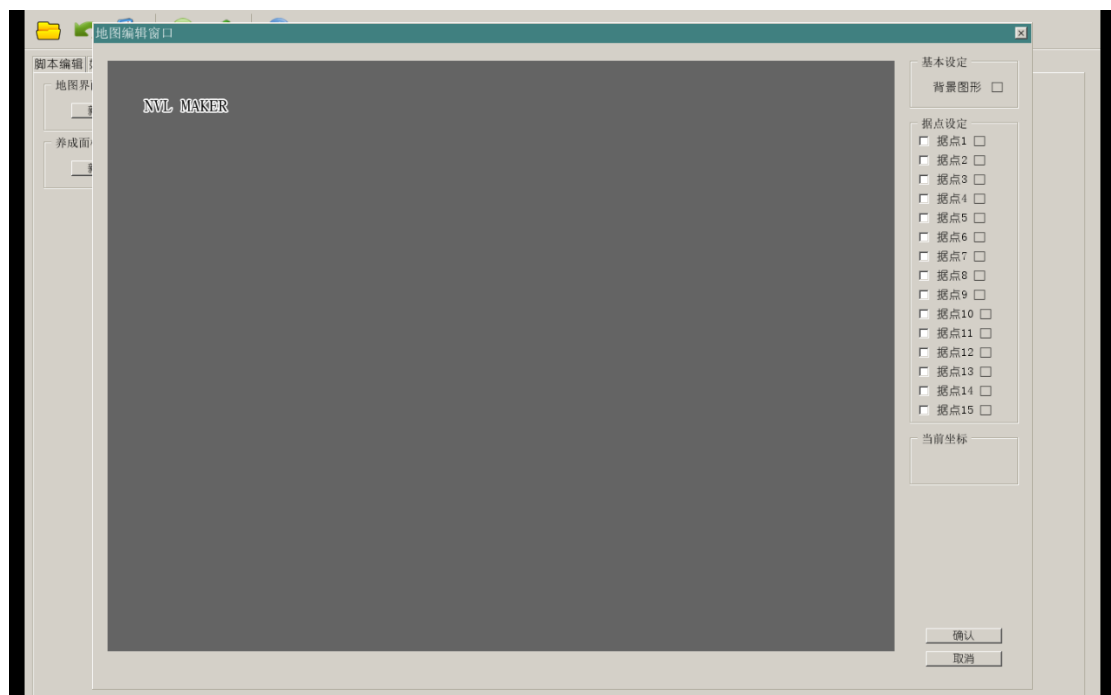
二、選擇的擴展——地圖

無論是選擇還是地圖，根本都是一個東西，那就是“按鈕”。
在一個界面上顯示很多個按鈕，等待玩家選擇，並且在選擇之後跳轉到不同的腳本繼續劇情。
地圖編輯器的操作和其他界面編輯器是類似的。不管怎樣，先來試著製作一下吧？

（1）新建地圖



（2）編輯地圖



打開新建的地圖，右邊是據點設定。所謂“據點”就是地圖上一個可以去的地方。目前最多可以設定 15 個。（通常來說，據點的數量太多，會導致地圖畫面很混亂，所以不推薦設置太多。）

勾選“據點 1”，畫面上顯示了一個按鈕。接著點“據點 1”右邊的□，開始設定內容。

地图据点设定

显示设定

地名 据点1

条件 ☐

按钮图形

x 50

y 100

一般 sample_off ☐

选中 sample_on ☐ ☐

按下 sample_on ☐ ☐

执行操作

剧本 ☐

标签 ☐

表达式 ☐

确认

取消

這裡要注意的是地名，修改這裡並不會影響遊戲的內容，但是會被顯示在地圖編輯器側邊欄上，可以方便管理。

至於“條件”，則是這個地圖什麼時候可以去的設置。和選擇按鈕一樣，你可以設定只有在特定條件下，這個地圖據點才開放。

所以，在遊戲裡，要顯示一個據點，有兩個條件，第一是在側邊欄上勾選了這個據點，第二是滿足了這個觸發條件。


當然，條件留空的話，就是任何情況下都會顯示。

按鈕圖形和其他界面按鈕是一樣的，可以設定位置，一般選中按下的圖像等。

劇本、標籤等等就不用說了，和選擇按鈕是完全一樣的。

（3）調用地圖

編輯保存地圖後，就可以在遊戲裡使用到了。

腳本編輯器裡，調用地圖的指令是這個：。



打開文件選擇，選擇剛剛製作的那張地圖。然後點確認。

另外，和選項一樣，當玩家選擇完畢，跳轉到某個腳本段落繼續的時候，請記得先清理畫面。

和清除選項一樣，這裡使用的是 。

同樣是每個跳轉到的地方，都要加上這句喲。

养成按钮设定

x 50

y 100

一般 sample_off ☐

选中 sample_on ☐ ☐

按下 sample_on ☐ ☐

名称 按钮1

剧本 ☐

标签

条件 ☐

执行

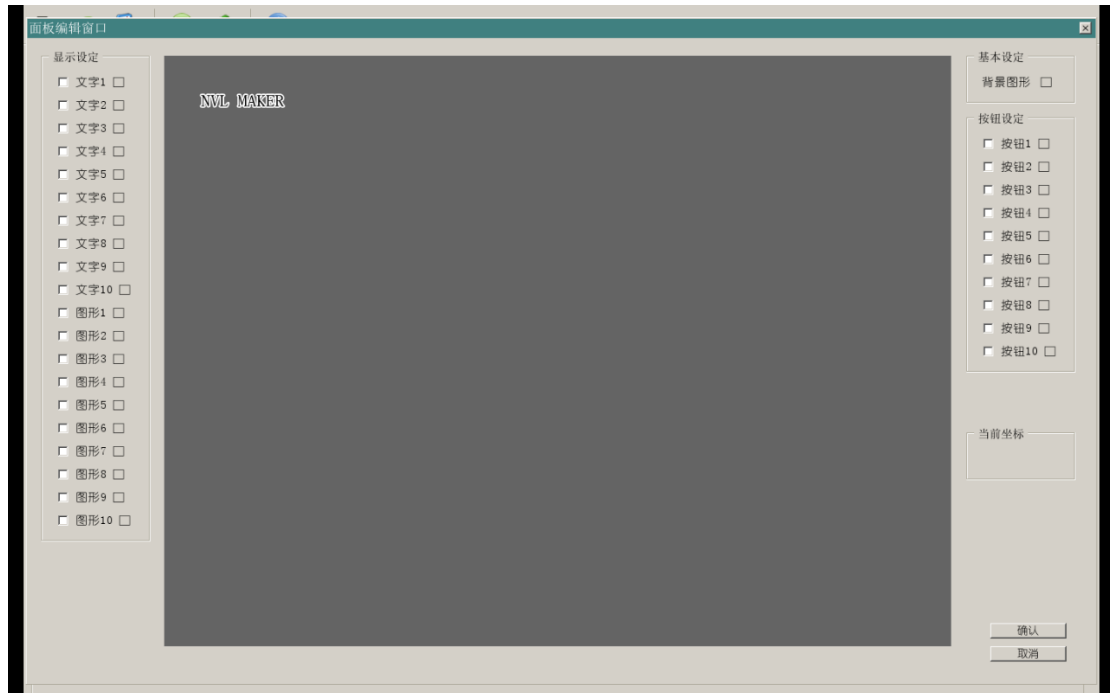
确认

取消

（最下方的“执行”就相当於選擇按钮裡的“执行-TJS”欄，可以填入變數操作。）

三、養成面板

（1）文字、圖形、按鈕



和新建地圖一樣，新建一個養成面板，然後打開。可以看到界面複雜了一些。

左邊是顯示“文字”和“圖形”的地方，右邊則是“按鈕”。

右邊的按鈕也可以當做地圖的據點來使用。

而左邊的文字、圖形可以用來在面板上顯示數據。

假如你需要做一個帶有“幾月幾日”日期框的地圖，那麼就可以用養成面板來實現。

（2）顯示變數的內容



“文字”部分的使用和其他部件一樣，勾選就能顯示在面板上。□開始設定。



“文字一” 同樣只是名稱，修改之後顯示在編輯器上，方便管理。

這裡新鮮的應該就只有“變數”一項了。

沒有錯，在這裡直接填入變數的名字，當調用養成面板的時候，這裡就會顯示出變數的內容。

例如我們可以把之前操作的那個變數顯示在這裡……

那麼當調用養成面板的時候就可以看到這樣的東西了：



如果變量的內容是字符串，那麼就會顯示出字符串來。



（3）顯示數字圖片

之前在存取系統裡調用的那個顯示數字圖片的宏，其實就是在這裡用的。

相比單純的顯示數字，有的時候更喜歡稍微華麗一點的效果吧。也就是用 0-9 的個位數字圖片，來組成多位的數字。這時候就要用到左邊欄的“圖形”了。



养成面板图形设定

显示设定

名称

条件 ☐

变数

坐标设定

x

y

变数值作为图形数字

前缀

间距

变数值作为图片名

演示图形 ☐

确认

取消

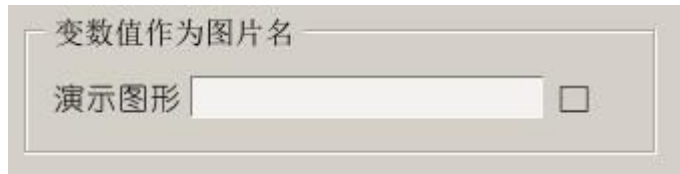
填寫變數，則變數的值會被當做數字顯示。

而數字圖片的外觀，是由數字圖片前綴決定的，代表 0-9 數字圖片前面的部分。

例如你可以把一套 0-9 的數字圖片命名為 abc0~abc9，那麼這裡就應該填寫 abc。

（4）根據變數的值（字符串）顯示圖片

除此之外，還有另外一個功能，



你可以指定在這裡顯示固定的一張圖片作為演示圖片。

在指定了演示圖片以後，數字圖片功能將會失效，改為第二個功能。

根據變數的值顯示圖片。

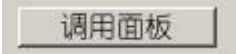
例如你的變數裡保存的不是“1~7”，而是“星期一~星期日”的字符串。

那麼準備名叫“星期一~星期日”的7張圖片，

並且在遊戲裡根據日期操作變數的值，那麼調用養成面板的時候，就會讀取這個字符串，然後把字符串當做圖片名，顯示出來。（找不到叫這個名字的圖片的話，當然會報錯。）

（5）調用養成面板

在腳本裡調用養成面板和調用地圖的方式幾乎完全一樣。

使用：來進行。



注意要做地圖的時候，這裡的“等待玩家選擇”需要勾上。



這樣的話遊戲才會停留在這裡，等待玩家點選按鈕。

否則的話，遊戲就會繼續前進了。

（6）不等待的話？

假如不等待玩家輸入，遊戲就會繼續進行了。

為什麼要為養成面板設定這樣的功能呢？

其實也很簡單。

養成遊戲裡經常用到的設定是一周只需要輸入一次日程。

這樣只有週一是需要“等待玩家安排”的，其他時候都只是刷一下畫面，顯示一

下數值的增減。

這時候，養成面板就成為一個“記分牌”一樣的存在了。

當然要注意，既然不需要等待，這個面板就沒有必要設定按鈕。

你可以製作“輸入日程專用”和“顯示專用”的兩張面板，根據條件輪番使用。

四、養成系統製作淺談

（1）沒有那麼簡單啦！

如果到製作養成系統還能夠說“完全不研究吉裡吉裡/KAG”就可以搞定。那是……不可能的！死心吧！

好在這只是新手教程，也沒有要負責教你怎麼打造一個完整的養成系統。不過，可以大概講一下訣竅。

（2）不變的日常——遊戲主循環

對一般的電子小說來說，是沒有循環這種概念的，因為它的每一步都是確定發生的，從 A→B→C，並不會再返回到 A。

但是養成系統則不同，它通常是以一周為單位，隨著日期的前進，根據今天是周幾，調用不同的面板。

很可能整個遊戲都是這樣的，週一，安排日程，週二到週五，養成，週六日，外出。

那麼這整個遊戲的主循環很可能就是一個簡單的“一天”腳本。每天要做的事情就是根據條件分歧調用面板，以及操作時間變數，再跳回開始，繼續下一天。

（3）地圖事件

因此養成的地圖，也和其他地圖 AVG 的形式不太一樣，通常一個地圖據點下面會有很多“可能觸發的事件”隨機發生，而不是點下固定據點就一定會跳到那個事件去。

如果你連續 10 天每天都去保健室，每天都發現同一個粉紅色頭髮的少女微笑著對你打招呼並自我介紹，要嘛你是鬼打牆了，要嘛你是黑長直（喂）。

所以當製作地圖等面板時，跳躍到的應該也不是固定的事件，而是一個新的判斷腳本。在這個腳本裡，根據各種各樣的變數值，來判斷現在哪一個事件可以被觸發。

這種時候當然是要用到“條件分歧”了。

例如，在某個地圖據點下，有 10 個可能發生的事件。1-10。

那麼這個判斷腳本可能是這樣的：

```
[if exp=條件]
呼叫事件 1
[elseif exp=條件]
呼叫事件 2
[elseif exp=條件]
呼叫事件 3
.....
[else]
呼叫事件 10
[endif]
```



```
[jump storage=主循環]
```

而被呼叫的每個事件腳本，都用[return]結尾。這樣當事件執行完畢之後，就會返回判斷腳本，再從判斷腳本跳到主循環。不必在意這個事件具體是什麼時候觸發的，或者觸發完畢以後接著幹什麼。

讓每個據點都對應一個判斷腳本，管理事件會方便一些。

（4）主線事件

主線事件，養成遊戲裡常見的“滿足一定條件之後”必定觸發的事件。

同樣也可以用和管理地圖事件一樣的方法來管理。只不過這部分的内容應該插入到主循環內。

例如這樣的一天：

*主循環

【判斷早上可能發生什麼事件】->

根據今天是周幾調用養成面板或者地圖->

數值變化->

【判斷下午可能發生什麼事件】->

一天結束->

```
[jump target=*主循環]
```

而【判斷可能發生什麼事件】自然就是跳到一個主線事件判斷腳本了。

（5）避免事件重複發生

假如一個事件已經發生過了，不希望它反復發生，那麼怎麼從事件判斷腳本裡把這個事件去掉呢？答案當然是“用變量”。

每個事件都應該有一個屬 自己的變量。當這個事件執行的時候，在這個事件的腳本裡，把變量設為 1。

在事件判斷的時候，要觸發這個事件，必須滿足的條件要包括“對應變量==0”。這樣的話，就可以製作出不重複發生的事件了。

除了用一般的變量以外，為了方便管理，“數組”和“字典”也是不錯的選擇。至於具體的就請去看吉裡吉裡和 KAG 的說明文檔了……

（6）事件的優先級

剛剛展示的事件判斷腳本（偽）是這樣的結構。假如事件 1 觸發了的話，那麼以後的事件（即使滿足條件）也都不會觸發了，一直到下次調用這個事件判斷腳本。所以希望先發生的事件，應該寫在比較前面的位置。

而在最後[else]以後的，則是優先級最低的事件，通常是在沒有事件的時候，用來湊數的。例如“去 XX 地方逛了逛，什麼也沒發生，又回來了。”

假如每次只觸發一個事件的話，就可以用這樣的事件判斷腳本。

但假如你希望的是一旦進行判斷，就讓所有滿足條件的事件都觸發的話。

就把事件判斷腳本寫成這樣的形式吧。

```
[if exp=條件]

呼叫事件 1

[endif]

[if exp=條件]

呼叫事件 2

[endif]

[if exp=條件]

呼叫事件 3

[endif]

.....

[if exp=條件]

呼叫事件 10

[endif]

[jump storage=主循環]
```

（7）日期系統

本質上就是幾個變數。f. 年，f. 月，f. 日，f. 星期幾……等等。每天（一次主循環）給日加上 1，當滿足條件的時候給月加上 1……以此類推。

不過如果覺得這段寫起來麻煩的話，也有插件可以用，具體看這裡：

<http://kcddp.keyfc.net/bbs/viewthread.php?tid=195>

五、後記

嗯，這個漫長的新手教程終於結束了。

在學吉裡吉裡的時候，總看到網上很多人喊著沒有教程沒有範例，

不過已經存在的那麼多範例和教程，認真去看的人又有幾個呢？

<http://kcddp.keyfc.net/bbs/viewthread.php?tid=1188&extra=page%3D1>

做遊戲，其實並不是那麼難的事。

但是也沒有簡單到不用動腦，不用動手的程度。

不管怎樣，希望你看完教程以後還願意堅持。

打開 **THE NVL Maker**，開始遊戲製作之旅吧……

期待著作品哟。^_^