

THE NVL MAKER

新手教程

(三)

選擇的製作

# 一、選項按鈕外觀設定

上一教程提到了腳本文件與標籤，這在電子小說裡用得最多的就是“選擇”了。THE NVL Maker 裡提供了比較簡單的選擇按鈕的製作方法。

首先我們需要對選擇按鈕的外觀等等進行設定。  
在工程設定（或界面設定）欄目下，可以看到“選項設定”。

選項按鈕

一般	bs_select_button_normal	<input type="checkbox"/>	
选中	bs_select_button_on	<input type="checkbox"/>	<input type="radio"/>
按下	bs_select_button_on	<input type="checkbox"/>	<input type="radio"/>

選項文字

字号	24		
一般	0xFFFFFFFF	<input type="checkbox"/>	
选中	0xCCFFFF	<input type="checkbox"/>	<input type="radio"/>
按下	0xCCFFFF	<input type="checkbox"/>	<input type="radio"/>
既读	0xBDD4E7	<input type="checkbox"/>	<input type="radio"/>

選項音效

选中		<input type="checkbox"/>	
按下		<input type="checkbox"/>	

在這裡用的默認設定，表現在遊戲裡是這樣的按鈕。

當然，你可以隨意修改圖片。



## 二、顯示選項按鈕

設定完之後我們返回“腳本編輯->打開腳本”，開始製作選擇~  
總之在之前的腳本的基礎上繼續吧。

### （1）準備選項

新建一行，在指令窗口裡選擇“準備選項”。彈出了下面這樣的窗口。



由於很可能是對話一陣之後，出現了可以分歧的路線，才讓玩家選擇的，因此對話框和系統按鈕可能都在畫面上。如果你喜歡在選擇的時候讓畫面空白一點，可以勾選這兩個選項，如果你希望讓玩家在選擇的時候也能做存檔等等操作，就這樣不改設定也沒問題。  
點下“確認”。

46

□ 准备选项

### （2）設定位置

現在可以開始顯示選項按鈕了。

不過，按鈕顯示之前，當然需要設定一下位置。

假設我們需要顯示兩個按鈕，在畫面上的位置分別是（0,200）和（0,300）。  
那麼就可以這樣做。

繼續新建一行，選擇“按鈕位置”，輸入第一個按鈕的位置，點下確認。



### （3）設定按鈕

再次新建一行，選擇“選項按鈕”。



选项按钮

按钮图形

选项文字

一般  ☐

选中  ☐ ☐

按下  ☐ ☐

选中SE  ☐

按下SE  ☐

执行操作

标签

文件  ☐

表达式

因為大部分的設定，例如“一般選中按下”時的按鈕樣式，我們都設定了默認值，所以這裡只需要填寫少量內容。

第一是【選項文字】，也就是要描繪在選項按鈕上的，玩家可以看到的文字。

第二是【標籤】和【文件】，用來指定當這個選擇按鈕點下後，會繼續執行的腳本段。

注意：對於選擇按鈕，除了選項文字以外，標籤也是必須填寫的，而文件名是可選的，在不填寫文件名的情況下，默認會在同一個文件內尋找對應的標籤名。

如果不填寫標籤的話，在測試時會默認跳至對應腳本的“\*start”這個標籤。假如這個標籤不存在，那自然就是出錯了（喂）。

總之先填成這樣吧，點擊“確認”。

46	□ 准备选项
47	□ 按钮&文字位置 x:0 y:200
48	□ 选项按钮 标签:*choose1 选项:选择一

現在，重複一的操作，建立一個位置在（0,300），選項文字為“選擇二”，標籤為\*choose2的按鈕。

46	□ 准备选项
47	□ 按钮&文字位置 x:0 y:200
48	□ 选项按钮 标签:*choose1 选项:选择一
49	□ 按钮&文字位置 x:0 y:300
50	□ 选项按钮 标签:*choose2 选项:选择二
51	(到达文件末端)

## （5）等待選項

現在所有的準備工作都做好了，只需要再加上一條“等待選項”，就可以看到成果了。所謂“等待選項”，意思就是顯示出已經設定好的選項按鈕，然後等待玩家選擇。在玩家做出選擇之前，遊戲會一直靜止在那裡。



可以設定各種各樣的顯示效果，不過這裡就全部用默認值好了。點下“確認”。

- 46    □ 准备选项
- 47    □ 按钮&文字位置 x:0 y:200
- 48    □ 选项按钮 标签:\*choose1 选项:选择一
- 49    □ 按钮&文字位置 x:0 y:300
- 50    □ 选项按钮 标签:\*choose2 选项:选择二
- 51    □ 等待选项
- 52    (到达文件末端)



## 三、設定選擇後的處理

### (1) 添加標籤和內容

現在可以開始測……呃等一下，雖然選項設定了標籤“\*choose1”和“\*choose2”，可這兩個標籤好像還沒有加？

先加上好了。

```
46  □ 准备选项
47  □ 按钮&文字位置 x:0 y:200
48  □ 选项按钮 标签:*choose1 选项:选择一
49  □ 按钮&文字位置 x:0 y:300
50  □ 选项按钮 标签:*choose2 选项:选择二
51  □ 等待选项
52  ★标签 标签名:*choose1
53  选择了一。[w]
54  ★标签 标签名:*choose2
55  选择了二。[w]
56  (到达文件末端)
```

不知道怎麼加這些內容的，回去看前面一講哦~

現在測試一下：



好像很簡單，隨便選一個吧……



## （2）好像有哪裡不對

現在問題來了，不管點了哪個按鈕，理論上說都會出現“選擇了一。”“選擇了二。”之類的對話啊。

對話哪裡去了？為什麼遊戲還是靜止的？

不過如果你點開遊戲菜單欄上面的“調試->控制台”，打開黑色的窗口，就可以看到這樣的警告：

```
10:38 prelogue.ks : jumped to : *choosel  
10:38 开始处理  
10:38 prelogue.ks : 选择了一。[w]  
10:38 警告 : 在没有显示出来的里文字层1等待换页点击。
```

因為 THE NVL Maker 裡選擇按鈕顯示時候用到的是“裡文字層 1”，而平時用的對話顯示是在“表文字層 0”，這個設置現在還沒改過來呢。

有些時候不知道出了什麼問題，可以打開這個窗口看看。

文字和圖片，設定在“表頁”的時候，可以立即顯示。

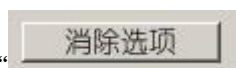
而設定在“裡頁”的時候，可以通過“執行切換”和“等待切換”來做出各種各樣的效果。

（比如剛剛的選擇按鈕，就是漸入顯示的。）

頁和層的概念，後面也會不時涉及到。請先大概留個印象。

## （3）清理一下畫面，讓一切恢復正常

不過現在不用管那麼多基本原理，只要知道如何改掉這個設置就行了。



用到的指令“消除选项”，它會幫你重置文字層，並且清理掉選項按鈕。

注意，跳轉到的每個標籤後，都要加上這個指令。

```
46   □ 准备选项
47   □ 按钮&文字位置 x:0 y:200
48   □ 选项按钮 标签:*choose1 选项:选择一
49   □ 按钮&文字位置 x:0 y:300
50   □ 选项按钮 标签:*choose2 选项:选择二
51   □ 等待选项
52   ★标签 标签名:*choose1
53   □ 消除选项
54   选择了一。[w]
55   ★标签 标签名:*choose2
56   □ 消除选项
57   选择了二。[w]
58   (到达文件末端)
```

現在再測試一下。沒有問題了吧？

接下來記得在選擇 1 和選擇 2 對話完畢的時候，加上“事件跳轉”，讓兩個選擇最後能夠合併到一條線上來繼續劇情。（可以參考 Template 裡的範例寫法。）

如果不這麼做的話，在顯示“選擇了一”之後，還會繼續顯示“選擇了二”的對話。

## 四、好感度？

你一定會遇到這樣的劇情設計，就是不管選擇了哪一項，臺詞都是一樣的，但是根據選擇不同，遊戲裡可攻略人物的好感度會上升或下降。

這是劇本偷懶的經典方式（喂）。

可是總不能因為劇本偷懶，腳本就寫兩段完全一模一樣的內容，然後再合併吧？

這個時候就要用到“讓兩個選擇都跳到同個段落，但是數值根據選擇變化”的功能了。

在吉裡吉裡中，數值變化的操作是用一些簡單的代碼來執行的，被稱為“TJS 表達式”。

……喂，也不要聽到“代碼”兩個字就點叉啊，我保證這個簡單到你會加減乘除就一定會寫啊。

## （1）重新設定按鈕

```
46   □ 准备选项
47   □ 按钮&文字位置 x:0 y:200
48   □ 选项按钮 标签:*继续 选项:选择一
49   □ 按钮&文字位置 x:0 y:300
50   □ 选项按钮 标签:*继续 选项:选择二
51   □ 等待选项
52   ★标签 标签名:*继续
53   □ 消除选项
54   不管选择哪一项，剧情都会这样继续的。[w]
55   (到达文件末端)
```

稍微修改一點點設置，現在腳本變成了這樣。

## （2）表達式（TJS 式）

然後假設“選擇一的時候好感度+5”，“選擇二的時候好感度-5”。

設定第一個按鈕是這樣的。

**选项按钮**

按钮图形

选项文字

一般  ☐

选中  ☐ ☐

按下  ☐ ☐

选中SE  ☐

按下SE  ☐

执行操作

标签

文件  ☐

表达式

表达式一欄，填入【f.好感度+=5】。

那麼第二個按鈕的表達式，當然是填上【f.好感度-=5】了。

```

46  □ 准备选项
47  □ 按钮&文字位置 x:0 y:200
48  □ 选项按钮 TJS:f.好感度+=5 标签:*继续 选项:选择一
49  □ 按钮&文字位置 x:0 y:300
50  □ 选项按钮 TJS:f.好感度-=5 标签:*继续 选项:选择二
51  □ 等待选项
52  ★ 标签 标签名:*继续
53  □ 消除选项
54  不管选择哪一项，剧情都会这样继续的。[w]

```

現在變成了這樣。

保存測試一下，一切正常。

### (3) 變數

每次寫教程都要講這個還真是麻煩啊，把 Nscripter 全攻略的變數介紹拿來一下

吧（喂）。

可以加加減減的數值，在遊戲裡就叫做“變量”或者“變數”——因為是可以變化的嘛。

變量本身需要有一個名字，例如“某人的好感度”，“某人的 BT 度” “HP”，“昨天的颱風等級”等，而變量的內容，就是一個數字。（有些時候不只數字，不過這裡先知道可以存儲數字就行了。）

KAG 系統中有三種變數。

以“f.”開頭的，被稱為一般變數，遊戲裡最常使用到的也是這種。會保存在遊戲存檔裡。

以“sf.”開頭的，被稱為全域變數（系統變數），會保存在獨立的記錄裡。

以“tf.”開頭的，叫做臨時變數，遊戲裡隨時可以用，但是不會被保存，一旦關閉遊戲下次再打開，就和沒存在過一樣。

剛剛說到的“f. 好感度”就是一個變數的名字。

為變數命名，在吉裡吉裡中還是比較簡單的，只要是“f.”開頭，支持英語，中文，克林貢語，多斯拉克語，或者隨便什麼語，以及以上語種的混排（只要開頭不是數字就行）。根據你的習慣起名吧。

當你寫下一個新的變數名的時候，系統會自動送給它一個空的值，可以被視為 0。

而接下來你通過 TJS 式對這個變數的操作，就會被一步步記錄下來。

在玩家存檔的時候，他們也會被保存。

如果覺得以上太複雜，我們先回到“f. 好感度”這個變數上。

在遊戲剛開始它是不存在的，只有在玩家點下任何一個按鈕以後，它才被創建出來。

所以一開始它的默認值是 0。

在 “f. 好感度+=5” 以後，它的值是 5。

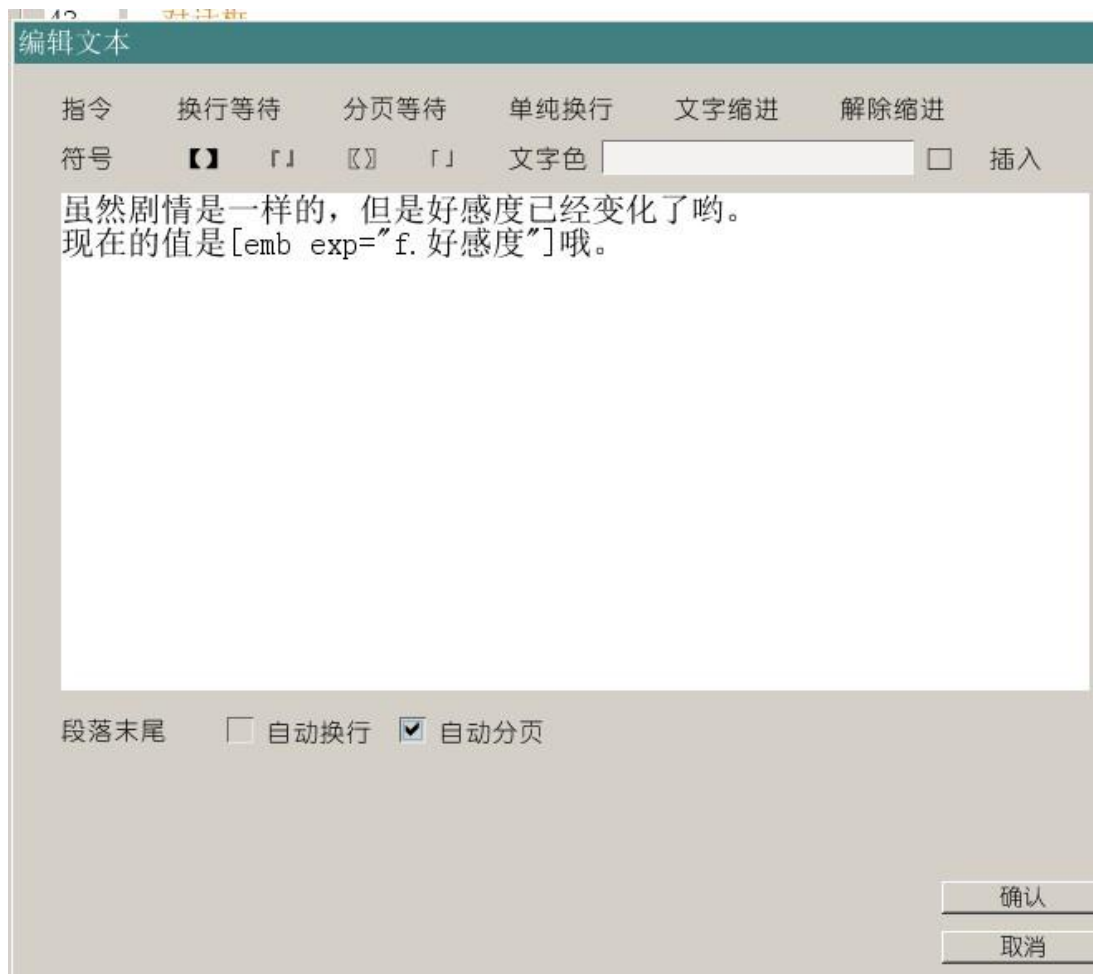
在 “f. 好感度-=5” 以後，它的值是-5。

這個很好理解吧？

#### **（4）讓變數顯示出來**

有時候為了方便調試，或者劇情需要（好感查詢系統？），變數的值需要顯示在遊戲裡。這時候就要用到新的 TJS 式了。而且，我們可以把它插入到對話裡。

比如說這樣：



在 KAG 裡，以@開頭的，或者[]括起來的內容就是一行指令。

包括“指令名”和“屬性”兩個部分。格式是這樣的。

[指令名 屬性 1=值 1 屬性 2=值 2 屬性 3=值 3]

@指令名 屬性 1=值 1 屬性 2=值 2 屬性 3=值 3

這兩種格式都可以使用，但@要求每行只能書寫一個指令。所以在對話裡插入時，用[]的形式會比較容易。

EMB 這個指令就是用來顯示變數的。它有一個叫 EXP 的屬性。EXP 的值則是你填寫的 TJS 表達式。

測試一下。





嗯，完全沒有問題。

順便一說，EMB 在腳本編輯器裡，對應的是這個指令“顯示變數(顯示 TJS 式)”。

假如對話末尾沒有分頁等待之類的符號，在兩段對話中插入這麼一行也可以達到如上文的效果。



```
46 现在的值是
47 ◇显示TJS式 表达式:f. 好感度
48 哦。[w]
49 (到达文件末端)
```

## 五、變數的應用

### (1) 第三種選擇

除了之前說的兩種情況，還有一種常見的劇情設定，就是在選擇之後劇情並沒有

什麼特別的變化，但是到了某個分歧點，突然之前埋下的“地雷”就炸了。

例如說，在第一個事件裡讓角色的好感度下降了，

在出現第二個事件的時候，本來有三個選項可以選，但好感度不夠，於是只顯示了兩個。

這種設定的話，就需要用到條件分歧了。

假設一下接下來就是這種情況。如果 f. 好感度的值 $\geq 5$ ，那麼，你可以多一個選擇。

```
57  □ 准备选项
58  □ 按钮&文字位置 x:0 y:200
59  □ 选项按钮 标签:*planA 选项:方案A
60  □ 按钮&文字位置 x:0 y:300
61  □ 选项按钮 标签:*planB 选项:方案B
62  □ 等待选项
63  (到达文件末端)
```

現在來加這個隱藏的“方案 C”。

## （2）創建條件分歧

在第二個按鈕下面新建一行，戳一下“變數系統-創建分歧”。



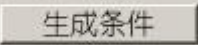


這裡的條件要填什麼呢。雖然可以猜得出來是“f. 好感度 $\geq$ 5”，不過也不能這麼蒙吧。

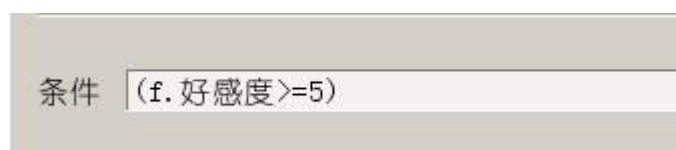
這個文本框後面有個☐。在 THE NVL Maker 裡，這就表示“你基本可以用鼠標來解決問題”。



填寫這樣的東西，應該沒問題吧。

先別急著點確認，這裡需要多一步：。

點一下這個按鈕，查看一下系統自動生成的 TJS 條件式是不是對的。



沒有問題，點下“確認”。



因為沒有其他的分歧情況，也沒有“都不符合”的時候的默認分歧情況，所以就這樣保持原樣，點下確認就好了。

```
44 ◇条件分歧 表达式: (f. 好感度>=5)
45 ◇条件分歧结束
```

接下來要插入的就是第三個選項了。

在“條件分歧”和“條件分歧結束”中間插入新行，填寫按鈕位置和選項按鈕。

```
57 □ 准备选项
58 □ 按钮&文字位置 x:0 y:200
59 □ 选项按钮 标签:*planA 选项:方案A
60 □ 按钮&文字位置 x:0 y:300
61 □ 选项按钮 标签:*planB 选项:方案B
62 ◇条件分歧 表达式: (f. 好感度>=5)
63 □ 按钮&文字位置 x:0 y:400
64 □ 选项按钮 标签:*planC 选项:方案C
65 ◇条件分歧结束
66 □ 等待选项
67 (到达文件末端)
```

不過，記得加入標籤\*planA，\*planB 和\*planC，以及後續的內容~

### （3）測試



現在就只有之前在選擇了一，f. 好感度為 5 的情況下，選項“方案 C”才會顯示。

## 六、擴展知識

### （1）“等於”和“等於等於”不一樣

為什麼給 TJS 條件式增加了特殊的生成器，而普通的 TJS 式就要自己填呢。

因為 TJS 條件式雖然大部分情況下也挺好寫的，但是總會不小心出現各種各樣的錯誤。

比如說，“當 f. 好感度等於 5”和“設定 f. 好感度的值為 5”，這兩個式子一個是條件，一個是執行的內容，當然不會一樣。

條件判斷的時候“等於”是用“==”，兩個等號來表示的。因此，

“當 f. 好感度等於 5”應該寫成“f. 好感度==5”。

而“設定 f. 好感度的值為 5”，才是“f. 好感度=5”。

為了避免寫錯，就專門給“條件”加上了生成器。

這樣也可以寫出更複雜的式子，比如說當 f. 好感度 等於 5 並且 f. HP 大於 10 並且 f. 今天的天空顏色 等於 0x0000FF 的時候，才有方案 C 的選項出現……

## （2）“變數”不一定是“數”

之前已經說過了，變數的值除了數字，還可以有其他的形式。

例如說，字符串。

假如你需要讓玩家輸入自己定義的角色姓名，這個姓名一定也是存儲在一個變數裡的。

在吉裡吉裡中，字符串是用雙引號””（注意不是 “ ”）括起來的文字。並且也可以保存在變數裡。

例如你可以設定選擇按鈕：

選擇“方案 A”的時候，f. 方案=”方案 A”。

選擇“方案 B”的時候，f. 方案=”方案 B”。

選擇“方案 C”的時候，f. 方案=”方案 C”。

最後在對話裡說：

選擇了[emb exp=”f. 方案”]。

寫成什麼樣都沒關係，不要忘記雙引號哟。

## （3）加不加引號？

除了給字符串設置值以外，剛剛用到的 EMB 指令什麼的，後面的 exp=也跟著雙引號。這是在告訴系統，雙引號之內的內容是連續的。

通常來說，除了字符串以外的東西加不加雙引號並不是很重要。但是……

### (1) 屬性的內容包含空格的情況

假如你的文件名有空間，例如 圖片 abc.png 這種時候，寫成

```
[bg storage=圖片 abc]
```

就會被當成是顯示一張名叫“圖片”的文件，而不是“圖片 abc”。這時候的結果很可能就是報錯找不到圖片。

### (2) 值為數字的情況

假如你是在操作或者比較變量。用於數字的時候，應該寫成：

@eval exp="f.好感度=5"，或者@if exp="f.好感度==5"這樣的形式，數字周圍是不能加引號的，否則就會被認為是字符串。

### (3) 外面已經有一層雙引號的情況

在 TJS 式內，也就是@eval exp=""裡面的內容時，由於 TJS 式本身是要用雙引號括起來的，就不能再重複使用了。

所以如果要給字符串賦值，請寫成這樣的形式：

```
@eval exp="f.星期幾='星期五'"，用單引號'代替雙引號。
```

或者寫成：

```
@eval exp="f.星期幾=\"星期五\""，用\"來代替雙引號。
```

THE NVL Maker 會為所有合適的屬性都加上雙引號或者單引號，但自己撰寫 KAG 腳本和 TJS 式的時候，單、雙引號的使用請額外注意。

## （4）全域變數的使用：初次打開遊戲與二周目

通常情況下，使用 f. 變數來應付遊戲裡的大部分情況就足夠了。但是，假如需要製作“整個遊戲的公共設置”這樣的功能，就要用到全域變數了。

所謂全域變數，就是會被系統保存，但是不會受到存檔、讀檔操作影響的變數。

（例如玩家打開遊戲，但是沒有做任何一次保存，只要腳本裡對全域變數進行了操作，依然會被記錄下來。）

例如說，我們需要遊戲在第一次執行的時候播放片頭動畫，之後就不再播放。

或者需要給標題畫面增加一個按鈕，只有在通關以後才出現。

類似這兩種情況就要使用到全域變數。

初次執行遊戲播放片頭動畫（以後執行遊戲就不再播放）的代碼示例：

```
;假如變數 sf.first 為 0，則意味著從來沒有播放過動畫
[if exp="sf.first==0"]

;播放片頭動畫的代碼
[video width="1024" left="0" visible="true" height="768" top="0"]
[playvideo storage="op.mpg"]
[wv canskip=false]

;執行到這裡，說明動畫已經播放完了

;接下來操作變數 sf.first 為 1，系統將會記錄這個值
[eval exp="sf.first=1"]

;下次打開遊戲的時候，因為 sf.first 已經不再是 0 而是 1 了，因此從上面的
if 到這個 endif 中間的代碼都不會再執行
[endif]
```



