The following script sample illustrate how to you can present randomized items, with for each item presented, a presentation of “sub-items”, themselves randomized.
The first section is the script with comments.
The second section illustrate the same process, showing list content and indexes.

1. Eye and Pen script (E&P v2.0)

```
; Randomize blocs of items, next randomize sub-blocs for each item
; The trick is to add sub-items at the end of items list, each time we process an item of the list.

; First, load the « main » item list
LoadList(MyItemList.txt)

; save its size in a label’s counter (i.e. a variable)
:MyListSize
SetLabelCounter(MyListSize,%M%)

; we will later use this list size to loop through the items
 RandomizeList

; Declare two labels to keep track of sub-items first and last index in list
:Start
:End

; Here we start looping through items in list
:MyLoop
    ; HERE Perform something with Item in list, for example
    DisplayPic(%L:MyLoop%.bmp,-1,-1,-1)

    ; Add 1st « sub-item » at the end of the list
    AddToList(%L:MyLoop%_word1)

    ; We will memorize this position in list in « Start » (this is the actual end of the list).
    ; We retrieve the index of the end of the with the keyword %M%, which returns the list size
    SetLabelCounter(Start,%M%)

    ; Next we add some other sub-items at the end of the list
    AddToList(%L:MyLoop%_word2)
    AddToList(%L:MyLoop%_word3)

    ; We will memorize this new position in list in « End » (this is the actual end of the list),
    ; using the same technique as above
    SetLabelCounter(End,%M%)

    ; Randomize the 3 sub-items we added (list indexes are from « Start" to « End »)
    RandomizeListRange(%I:Start%,%I:End%)

    ; We will now loop through the 3 sub-items. The loop will go from list index « Start » to « end ».
    SetLabelCounter(MySubLoop,%I:Start%)

    ; Since MySubLoop’s Label counter will be automatically incremented by 1 when the script
    ; interpreter will read it, we subtract 1 before, to compensate and thus keep a correct start index.
    SubtractFromLabelCounter(MySubLoop,1)
```

```
:MySubLoop

    ; HERE Perform some task
    DisplayPic(%L:MySubLoop%.bmp,-1,-1,-1)

  LoopIfLabelIsBelow(MySubLoop,%I:End%,FALSE)
LoopIfLabelIsBelow(MyLoop,%I:MyListSize%,FALSE)
```

2. Illustrated explanation

After LoadList(MyItemList.txt), the list will look something like below:

Index	**Item**
1	MyItem1
2	MyItem2
3	MyItem3
4	MyItem4

MyListSize's counter will be 4, since the list contains 4 items (retrieved with %M%).
The main loop (MyLoop) will go from list item 1 to 4.
When processing each item in list we will do the following.
Add 3 sub-items at the end of the list…

On first pass:

	Index	Item	
MyLoop →	1	MyItem1	
	2	MyItem2	
	3	MyItem3	
	4	MyItem4	← MyListSize
MySubLoop →	5	MyItem1_word1	← Start
	6	MyItem1_word2	
	7	MyItem1_word3	← End

On second pass:

	Index	Item	
	1	MyItem1	
MyLoop →	2	MyItem2	
	3	MyItem3	
	4	MyItem4	← MyListSize
	5	MyItem1_word1	
	6	MyItem1_word2	
	7	MyItem1_word3	
MySubLoop →	8	MyItem2_word1	← Start
	9	MyItem2_word2	
	10	MyItem2_word3	← End

Etc.