

Applies To:

SAP R/3, ABAP/4

Summary

This program illustrates some of the selection-screen design features, simple use of field symbols and the various events associated with a report program. And one good point about this program is, if one feels bored reading this, just press F8 and enjoy a nice refreshing memory test Game.

By: Renjith John Andrews

Company and Title: Wipro Technologies, Associate Consultant

Date: 03 Mar 2005

The Program

This program illustrates some of the selection-screen design features, simple use of field symbols and the various events associated with a report program. And one good point about this program is, if one feels bored reading this, just press F8 and enjoy a nice refreshing memory test Game.

```
REPORT zrja_memorygame_forblog .

INCLUDE <icon>.

TABLES : sscrfields.

*** internal table for storing the list of icons temporarily
DATA : BEGIN OF i_icons OCCURS 1000,
        id LIKE icon-id,
      END OF i_icons.

*** internal table for storing the 32 icons/pictures
DATA : BEGIN OF i_iconlist OCCURS 32,
        id LIKE icon-id,
      END OF i_iconlist.

*** internal table for storing the icons in the 64 positions
DATA : BEGIN OF i_layout OCCURS 0,
        row(1) TYPE n,
        col(1) TYPE n,
        id LIKE icon-id,
      END OF i_layout.

*** data declaration

DATA : w_layout LIKE i_layout.
DATA : ran_no TYPE i.
DATA : len TYPE i.
DATA : subrc LIKE sy-subrc.
DATA : count TYPE i.
DATA : name(10).
```

```
DATA : flg_one.  
DATA : flg_two.  
DATA : icon_one LIKE icon-id.  
DATA : icon_two LIKE icon-id.  
DATA : b_one(5).  
DATA : b_two(5).  
DATA : current_ucomm LIKE sscrfields-ucomm.  
FIELD-SYMBOLS : <fs> TYPE ANY.
```

```
**** selection-screen declaration.
```

```
*** keeping the button names and its user-command same will  
*** make things easier later....
```

```
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME NO INTERVALS.  
SELECTION-SCREEN SKIP 1.  
SELECTION-SCREEN PUSHBUTTON /2(4) pb11 USER-COMMAND pb11.  
SELECTION-SCREEN PUSHBUTTON 9(4) pb12 USER-COMMAND pb12.  
SELECTION-SCREEN PUSHBUTTON 16(4) pb13 USER-COMMAND pb13.  
SELECTION-SCREEN PUSHBUTTON 23(4) pb14 USER-COMMAND pb14.  
SELECTION-SCREEN PUSHBUTTON 30(4) pb15 USER-COMMAND pb15.  
SELECTION-SCREEN PUSHBUTTON 37(4) pb16 USER-COMMAND pb16.  
SELECTION-SCREEN PUSHBUTTON 44(4) pb17 USER-COMMAND pb17.  
SELECTION-SCREEN PUSHBUTTON 51(4) pb18 USER-COMMAND pb18.
```

```
SELECTION-SCREEN SKIP 1.  
SELECTION-SCREEN PUSHBUTTON /2(4) pb21 USER-COMMAND pb21.  
SELECTION-SCREEN PUSHBUTTON 9(4) pb22 USER-COMMAND pb22.  
SELECTION-SCREEN PUSHBUTTON 16(4) pb23 USER-COMMAND pb23.  
SELECTION-SCREEN PUSHBUTTON 23(4) pb24 USER-COMMAND pb24.  
SELECTION-SCREEN PUSHBUTTON 30(4) pb25 USER-COMMAND pb25.  
SELECTION-SCREEN PUSHBUTTON 37(4) pb26 USER-COMMAND pb26.  
SELECTION-SCREEN PUSHBUTTON 44(4) pb27 USER-COMMAND pb27.  
SELECTION-SCREEN PUSHBUTTON 51(4) pb28 USER-COMMAND pb28.
```

SELECTION-SCREEN SKIP 1.

SELECTION-SCREEN PUSHBUTTON /2(4) pb31 USER-COMMAND pb31.

SELECTION-SCREEN PUSHBUTTON 9(4) pb32 USER-COMMAND pb32.

SELECTION-SCREEN PUSHBUTTON 16(4) pb33 USER-COMMAND pb33.

SELECTION-SCREEN PUSHBUTTON 23(4) pb34 USER-COMMAND pb34.

SELECTION-SCREEN PUSHBUTTON 30(4) pb35 USER-COMMAND pb35.

SELECTION-SCREEN PUSHBUTTON 37(4) pb36 USER-COMMAND pb36.

SELECTION-SCREEN PUSHBUTTON 44(4) pb37 USER-COMMAND pb37.

SELECTION-SCREEN PUSHBUTTON 51(4) pb38 USER-COMMAND pb38.

SELECTION-SCREEN SKIP 1.

SELECTION-SCREEN PUSHBUTTON /2(4) pb41 USER-COMMAND pb41.

SELECTION-SCREEN PUSHBUTTON 9(4) pb42 USER-COMMAND pb42.

SELECTION-SCREEN PUSHBUTTON 16(4) pb43 USER-COMMAND pb43.

SELECTION-SCREEN PUSHBUTTON 23(4) pb44 USER-COMMAND pb44.

SELECTION-SCREEN PUSHBUTTON 30(4) pb45 USER-COMMAND pb45.

SELECTION-SCREEN PUSHBUTTON 37(4) pb46 USER-COMMAND pb46.

SELECTION-SCREEN PUSHBUTTON 44(4) pb47 USER-COMMAND pb47.

SELECTION-SCREEN PUSHBUTTON 51(4) pb48 USER-COMMAND pb48.

SELECTION-SCREEN SKIP 1.

SELECTION-SCREEN PUSHBUTTON /2(4) pb51 USER-COMMAND pb51.

SELECTION-SCREEN PUSHBUTTON 9(4) pb52 USER-COMMAND pb52.

SELECTION-SCREEN PUSHBUTTON 16(4) pb53 USER-COMMAND pb53.

SELECTION-SCREEN PUSHBUTTON 23(4) pb54 USER-COMMAND pb54.

SELECTION-SCREEN PUSHBUTTON 30(4) pb55 USER-COMMAND pb55.

SELECTION-SCREEN PUSHBUTTON 37(4) pb56 USER-COMMAND pb56.

SELECTION-SCREEN PUSHBUTTON 44(4) pb57 USER-COMMAND pb57.

SELECTION-SCREEN PUSHBUTTON 51(4) pb58 USER-COMMAND pb58.

SELECTION-SCREEN SKIP 1.

SELECTION-SCREEN PUSHBUTTON /2(4) pb61 USER-COMMAND pb61.

SELECTION-SCREEN PUSHBUTTON 9(4) pb62 USER-COMMAND pb62.

SELECTION-SCREEN PUSHBUTTON 16(4) pb63 USER-COMMAND pb63.

```
SELECTION-SCREEN PUSHBUTTON 23(4) pb64 USER-COMMAND pb64.
SELECTION-SCREEN PUSHBUTTON 30(4) pb65 USER-COMMAND pb65.
SELECTION-SCREEN PUSHBUTTON 37(4) pb66 USER-COMMAND pb66.
SELECTION-SCREEN PUSHBUTTON 44(4) pb67 USER-COMMAND pb67.
SELECTION-SCREEN PUSHBUTTON 51(4) pb68 USER-COMMAND pb68.
```

```
SELECTION-SCREEN SKIP 1.
SELECTION-SCREEN PUSHBUTTON /2(4) pb71 USER-COMMAND pb71.
SELECTION-SCREEN PUSHBUTTON 9(4) pb72 USER-COMMAND pb72.
SELECTION-SCREEN PUSHBUTTON 16(4) pb73 USER-COMMAND pb73.
SELECTION-SCREEN PUSHBUTTON 23(4) pb74 USER-COMMAND pb74.
SELECTION-SCREEN PUSHBUTTON 30(4) pb75 USER-COMMAND pb75.
SELECTION-SCREEN PUSHBUTTON 37(4) pb76 USER-COMMAND pb76.
SELECTION-SCREEN PUSHBUTTON 44(4) pb77 USER-COMMAND pb77.
SELECTION-SCREEN PUSHBUTTON 51(4) pb78 USER-COMMAND pb78.
```

```
SELECTION-SCREEN SKIP 1.
SELECTION-SCREEN PUSHBUTTON /2(4) pb81 USER-COMMAND pb81.
SELECTION-SCREEN PUSHBUTTON 9(4) pb82 USER-COMMAND pb82.
SELECTION-SCREEN PUSHBUTTON 16(4) pb83 USER-COMMAND pb83.
SELECTION-SCREEN PUSHBUTTON 23(4) pb84 USER-COMMAND pb84.
SELECTION-SCREEN PUSHBUTTON 30(4) pb85 USER-COMMAND pb85.
SELECTION-SCREEN PUSHBUTTON 37(4) pb86 USER-COMMAND pb86.
SELECTION-SCREEN PUSHBUTTON 44(4) pb87 USER-COMMAND pb87.
SELECTION-SCREEN PUSHBUTTON 51(4) pb88 USER-COMMAND pb88.
```

```
SELECTION-SCREEN SKIP 1.
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT 2(15) no_tries.
PARAMETERS : p_tries TYPE i MODIF ID hid VISIBLE LENGTH 3.
SELECTION-SCREEN END OF LINE.
```

```
SELECTION-SCREEN END OF BLOCK b1.
```

```
INITIALIZATION.
```

```
    PERFORM do_initialization.
```

```
AT SELECTION-SCREEN OUTPUT.
```

```
    IF i_layout[] IS INITIAL.
```

```
        MESSAGE i000(z01) WITH 'You have WON !!!'.
```

```
    ENDIF.
```

```
*** make the count field display only
```

```
    LOOP AT SCREEN.
```

```
        IF screen-group1 = 'HID'.
```

```
            screen-input = 0.
```

```
            MODIFY SCREEN.
```

```
        ENDIF.
```

```
    ENDLOOP.
```

```
AT SELECTION-SCREEN.
```

```
*** make sure it is one of the buttons that is clicked and
```

```
*** not clicking the same button again
```

```
*** exception is if 2 are open and you want to keep the 2nd one open
```

```
*** you can do so....
```

```
    CHECK sscrfields-ucomm CS 'PB' AND
```

```
        ( sscrfields-ucomm <> current_ucomm OR
```

```
          NOT flg_two IS INITIAL ).
```

```
    current_ucomm = sscrfields-ucomm.
```

```
*** making sure again that a button is clicked and get the icon
```

```
*** hidden in the clicked button.
```

```
    READ TABLE i_layout WITH KEY row = sscrfields-ucomm+2(1)
```

```
        col = sscrfields-ucomm+3(1).
```

```
CHECK sy-subrc = 0.

PERFORM validate.

*&-----*
*&      Form  validate
*&-----*
* checking whether its the 1st or the 2nd button click, assign the ucomm
* to the field-symbol. The trick is the buttons have been named same as
* its ucomm, so assigning it gives the value of the button.
*-----*
* --> p1      text
* <-- p2      text
*-----*

FORM validate.
  IF NOT flg_two IS INITIAL.
    ASSIGN (b_one) TO <fs>.
    <fs> = '@5F@'.
    ASSIGN (b_two) TO <fs>.
    <fs> = '@5F@'.
    CLEAR : flg_one,flg_two,
            icon_one,icon_two,
            b_one,b_two.
  ENDIF.
  IF flg_one IS INITIAL OR
    flg_two IS INITIAL.
    ASSIGN (sscrfields-ucomm) TO <fs>.
    IF sy-subrc = 0.
      <fs> = i_layout-id.
    ENDIF.
    IF flg_one IS INITIAL.
      flg_one = 'X'.
      icon_one = i_layout-id.
      b_one = sscrfields-ucomm.
    ELSEIF flg_two IS INITIAL.
      flg_two = 'X'.
```

```
        icon_two = i_layout-id.
        b_two = sscrfields-ucomm.
    ENDIF.
ENDIF.
IF icon_one = icon_two.
    p_tries = p_tries + 1.
    DELETE i_layout WHERE id = icon_one.
    CLEAR : flg_one,flg_two,icon_one,icon_two.
ENDIF.
IF NOT flg_two IS INITIAL.
    p_tries = p_tries + 1.
ENDIF.

ENDFORM.                " validate

*&-----*
*&      Form  do_initialization
*&-----*
* Make all the buttons show blank space. Also select, the list of
* all icons from the standard table and select 32 from it randomly. Once
* the 32 icons are selected, lay it out in pairs on the 8 X 8 board of
* buttons.
*-----*
*-----*
FORM do_initialization.

    no_tries = 'No. of Tries'.

    pb11 = pb12 = pb13 = pb14 = pb15 = pb16 = pb17 = pb18 = icon_space.

    pb21 = pb22 = pb23 = pb24 = pb25 = pb26 = pb27 = pb28 = icon_space.

    pb31 = pb32 = pb33 = pb34 = pb35 = pb36 = pb37 = pb38 = icon_space.

    pb41 = pb42 = pb43 = pb44 = pb45 = pb46 = pb47 = pb48 = icon_space.
```

```
pb51 = pb52 = pb53 = pb54 = pb55 = pb56 = pb57 = pb58 = icon_space.
```

```
pb61 = pb62 = pb63 = pb64 = pb65 = pb66 = pb67 = pb68 = icon_space.
```

```
pb71 = pb72 = pb73 = pb74 = pb75 = pb76 = pb77 = pb78 = icon_space.
```

```
pb81 = pb82 = pb83 = pb84 = pb85 = pb86 = pb87 = pb88 = icon_space.
```

```
SELECT id FROM icon INTO TABLE i_icons
      WHERE NOT id IN ('@5F@','@E5@','@00@','@E6@','@E7@',
                      '@EI@','@GE@','@GG@','@JX@','@JY@',
                      '@JZ@','@K0@').
```

```
DESCRIBE TABLE i_icons LINES len.
```

```
CLEAR : subrc.
```

```
WHILE subrc < 32.
```

```
  CALL FUNCTION 'QF05_RANDOM_INTEGER'
```

```
    EXPORTING
```

```
      ran_int_max = len
```

```
      ran_int_min = 1
```

```
    IMPORTING
```

```
      ran_int = ran_no
```

```
    EXCEPTIONS
```

```
      invalid_input = 1
```

```
      OTHERS = 2.
```

```
  READ TABLE i_icons INDEX ran_no.
```

```
  READ TABLE i_iconlist WITH KEY id = i_icons-id.
```

```
  IF sy-subrc <> 0.
```

```
    i_iconlist-id = i_icons-id.
```

```
    APPEND i_iconlist.
```

```
  ENDIF.
```

```
  DESCRIBE TABLE i_iconlist LINES subrc.
```

```
ENDWHILE.
```

```
REFRESH : i_icons.
```

```
CLEAR : subrc.
```

```
WHILE subrc < 64.
  IF subrc <> 0.
    READ TABLE i_layout INDEX subrc.
    IF i_layout-col = 8.
      i_layout-col = 1.
      i_layout-row = i_layout-row + 1.
    ELSE.
      i_layout-col = i_layout-col + 1.
    ENDIF.
  ELSE.
    i_layout-row = 1.
    i_layout-col = 1.
  ENDIF.
  CLEAR : count,i_iconlist.
  CALL FUNCTION 'QF05_RANDOM_INTEGER'
    EXPORTING
      ran_int_max   = 32
      ran_int_min   = 1
    IMPORTING
      ran_int       = ran_no
    EXCEPTIONS
      invalid_input = 1
      OTHERS        = 2.
  READ TABLE i_iconlist INDEX ran_no.
  LOOP AT i_layout INTO w_layout WHERE id = i_iconlist-id.
    count = count + 1.
  ENDLOOP.
  IF count < 2.
    i_layout-id = i_iconlist-id.
    APPEND i_layout.
  ENDIF.
  CLEAR : i_layout.
  DESCRIBE TABLE i_layout LINES subrc.
ENDWHILE.

CLEAR : flg_one,flg_two,icon_one,icon_two, current_ucomm.
```

```
ENDFORM.                                " do_initialization
```

And finally, how to play the game. Identify pairs of picture by consecutive clicks on buttons hiding the same picture. If consecutive clicks do not reveal the same pictures, those are hidden again. Try and identify all the pairs with the minimum no. of clicks.

And some stats to spice things up. I was able to identify all 32 pairs in a best of 53 tries. !!!!

Disclaimer & Liability Notice

This document may discuss sample coding, which does not include official interfaces and therefore is not supported. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing of the code and methods suggested here, and anyone using these methods, is doing it under his/her own responsibility.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of the technical article, including any liability resulting from incompatibility between the content of the technical article and the materials and services offered by SAP. You agree that you will not hold SAP responsible or liable with respect to the content of the Technical Article or seek to do so.

Author Bio

Renjith John Andrews is an Associate Consultant working with Wipro Technologies, India.

Copyright © 2005 SAP AG, Inc. All Rights Reserved. SAP, mySAP, mySAP.com, xApps, xApp, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product, service names, trademarks and registered trademarks mentioned are the trademarks of their respective owners.