**扫雷小游戏:**

|  |
| --- |
| REPORT ZRPTEST009.  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* |
|  |