

File created: 10-Jan-86 08:27:39 {PHYLUM}<BOBROW>LISP>BLACKBOX.;7

changes to: (FNS InitializeGuessArray BlackBoxChoices InitializeBallArray NewGame AskQuestion BlackBoxWindowFn  
OnEdge ProbeBallArray FillBox DrawBlackBox RedisplayBox)  
(VARS BLACKBOXCOMS HintTime)

previous date: 3-Jan-86 19:12:57 {PHYLUM}<BOBROW>LISP>BLACKBOX.;5

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS

(\* \* Copyright (c) 1985, 1986 by Xerox Corporation. All rights reserved.)

(RPAQQ **BLACKBOXCOMS**

((VARS BlackBoxSquare HintTime)  
(FNS AskQuestion BlackBox BlackBoxChoices BlackBoxTitle FillBox InitializeBallArray InitializeGuessArray  
LeftAhead LeftTurn MakeBlackBoxWindow MoveAhead MoveBall DrawBlackBox BallAhead BallDownOrUp  
BBBoxNumber BlackBoxWindowFn MakeBallArray NewGame OnEdge ProbeBallArray RedisplayBox RightAhead  
RightTurn SetSquareArray ShowBalls SquareArray))

(RPAQQ **BlackBoxSquare** 40)

(RPAQQ **HintTime** 5000)

(DEFINEQ

**AskQuestion**

[LAMBDA (window xBox yBox) (\* edited: "10-Jan-86 07:15")  
(LET\* ((boxSize (WINDOWPROP window (QUOTE BoxSize)))  
(guessArray (WINDOWPROP window (QUOTE GuessArray)))  
(answer (**ProbeBallArray** (WINDOWPROP window (QUOTE BallArray))  
xBox yBox boxSize)))  
(AND answer (**SetSquareArray** guessArray xBox boxSize yBox  
(if (LISTP answer)  
then (\* use a number)  
(LET [(g (WINDOWPROP window (QUOTE LastGuessNumber)  
(PLUS (WINDOWPROP window (QUOTE LastGuessNumber))  
1])  
(**SetSquareArray** guessArray (CAR answer)  
(CDR answer)  
boxSize g)  
g)  
else answer)))  
answer])

**BlackBox**

[LAMBDA (numRows numBalls) (\* edited: " 3-Jan-86 16:19")  
(\* \* This is a game in which one guesses where balls are hidden)

(if (NOT (NUMBERP numRows))  
then (SETQ numRows 8))  
(if (NOT (NUMBERP numBalls))  
then (SETQ numBalls 4))  
(if (GREATERP numRows 16)  
then (PRINTOUT T "Too big. Using " numRows " rows.")  
(SETQ numRows 16))  
(if (GREATERP numBalls numRows)  
then (PRINTOUT T "You chose too many balls. I will use " numRows T)  
(SETQ numBalls numRows))  
(LET\* ((boxSize (PLUS 2 numRows))  
(boxWidth (TIMES BlackBoxSquare boxSize))  
(boxWindow (**MakeBlackBoxWindow** boxSize boxWidth numBalls)))  
(MOVEW boxWindow 0 0))

**BlackBoxChoices**

[LAMBDA (window) (\* edited: "10-Jan-86 08:25")  
(LET [(ballArray (WINDOWPROP window (QUOTE BallArray)))  
(guessArray (WINDOWPROP window (QUOTE GuessArray))  
(SELECTQ [MENU (**create** MENU  
ITEMS \_ (QUOTE (ShowCorrect ShowAll NewGame ("New Game Using Balls Shown"  
(QUOTE NewGameFrom)  
"Use the balls shown to initialize  
game")  
ChangeNumberOfBalls ChangeNumberOfRows])  
(ShowAll (**ShowBalls** window ballArray guessArray T)  
(DISMISS HintTime)  
(REDISPLAYW window))  
(ShowCorrect (**ShowBalls** window ballArray guessArray NIL)  
(DISMISS HintTime)  
(REDISPLAYW window))  
(NewGame (**NewGame** window))

```

(NewGameFrom (NewGame window NIL (BallsDisplayed window))
(ChangeNumberOfBalls
  (LET ((numRows (DIFFERENCE (WINDOWPROP window (QUOTE BoxSize))
    2))
    (numBalls (RNUMBER "How many hidden balls")))
    (WINDOWPROP window (QUOTE NumBalls)
      (COND
        ((GREATERP 1 numBalls)
         (PRINTOUT T .FONT (HELVETICA 18)
           "You need to hide some balls." T)
         (WINDOWPROP window (QUOTE NumBalls)))
        ((GREATERP numBalls numRows)
         (PRINTOUT T .FONT (HELVETICA 18)
           "You are asking for too many balls. Using " numRows T)
         numRows)
        (T numBalls)))
      (WINDOWPROP window (QUOTE TITLE)
        (CONCAT "Black Box with " numBalls " balls")))
  (NewGame window))
(ChangeNumberOfRows
  [LET ((numRows (RNUMBER "How many rows?"))
    (CLOSEF window)
    (BlackBox numRows (WINDOWPROP window (QUOTE NumBalls)]
  NIL])

```

**(BlackBoxTitle**

```

[LAMBDA (numBalls) (* edited: "30-Dec-85 17:51")
  (CONCAT "Black Box with " numBalls " balls -- Click Here For Help")]

```

**(FillBox**

```

[LAMBDA (window x y symbol) (* edited: "10-Jan-86 07:59")
  (LET ((xPos (TIMES x BlackBoxSquare))
    (yPos (TIMES y BlackBoxSquare))
    (SELECTQ symbol
      (Black (DSPFILL (CREATEREGION xPos yPos BlackBoxSquare BlackBoxSquare)
        BLACKSHADE NIL window))
      (Ball (FILLCIRCLE (PLUS xPos (TIMES 0.5 BlackBoxSquare))
        (PLUS yPos (TIMES 0.5 BlackBoxSquare))
        (TIMES 0.45 BlackBoxSquare)
        BLACKSHADE window))
      (NIL (DSPFILL (CREATEREGION (PLUS xPos 2)
        (PLUS yPos 2)
        (DIFFERENCE BlackBoxSquare 2)
        (DIFFERENCE BlackBoxSquare 2))
        WHITESHADE
        (QUOTE REPLACE)
        window))
      (PROGN (MOVETO (PLUS xPos (TIMES 0.25 BlackBoxSquare))
        (PLUS yPos (TIMES 0.25 BlackBoxSquare))
        window)
        (PRIN1 symbol window]))

```

**(InitializeBallArray**

```

[LAMBDA (array numBalls boxSize ballPositions) (* edited: "10-Jan-86 08:27")
  (for I from 0 to (SUB1 (ARRAYSIZE array)) do (SETA array I NIL))
  (for position in ballPositions do (SetSquareArray array (CAR position)
    (CDR position)
    boxSize
    (QUOTE Ball)))
  (for i from (LENGTH ballPositions) to (SUB1 numBalls)
    do (PROG (randX randY)
      doAgain
        (SETQ randX (RAND 1 (DIFFERENCE boxSize 2)))
        (SETQ randY (RAND 1 (DIFFERENCE boxSize 2)))
        (if (SquareArray array randX randY boxSize)
          then (GO doAgain)
          else (SetSquareArray array randX randY boxSize (QUOTE Ball))

```

**(InitializeGuessArray**

```

[LAMBDA (guessArray boxSize) (* edited: "10-Jan-86 08:06")
  (LET ((maxIndex (SUB1 boxSize))
    (for i from 0 to maxIndex
      do (for j from 0 to maxIndex do (SetSquareArray guessArray i j boxSize
        (if [OR (AND (EQ i 0)
          (OR (EQ j 0)
            (EQ j maxIndex)))
          (AND (EQ i maxIndex)
            (OR (EQ j 0)
              (EQ j maxIndex)]
          then (QUOTE Black)]

```

**(LeftAhead**

```
[LAMBDA (array x y boxSize direction)
  (EQ (QUOTE Ball)
    (SquareArray array (SELECTQ direction
      ((L U)
        (SUB1 x))
      ((D R)
        (ADD1 x))
      x)
    (SELECTQ direction
      ((L D)
        (SUB1 y))
      ((R U)
        (ADD1 y))
      y)
    boxSize])
  (* edited: "29-Dec-85 18:17")
```

**(LeftTurn**

```
[LAMBDA (direction)
  (SELECTQ direction
    (U (QUOTE L))
    (R (QUOTE U))
    (D (QUOTE R))
    (L (QUOTE D))
    (ERROR "Bad Direction" direction))
  (* edited: "29-Dec-85 18:08")
```

**(MakeBlackBoxWindow**

```
[LAMBDA (boxSize boxWidth numBalls)
  (* * Draw the window, and install a buttonFunction that will make the right moves for the game)
  (LET ((window (CREATEW (CREATEREGION 0 0 (WIDTHIFWINDOW boxWidth 4)
    (HEIGHTIFWINDOW boxWidth T 4))
    (BlackBoxTitle numBalls)
    4)))
    (DSPFONT (FONTCREATE (QUOTE (HELVETICA 18 BOLD)))
      window)
    (WINDOWPROP window (QUOTE BoxWidth)
      boxWidth)
    (WINDOWPROP window (QUOTE BoxSize)
      boxSize)
    (WINDOWPROP window (QUOTE NumBalls)
      numBalls)
    (WINDOWPROP window (QUOTE REPAINTFN)
      (QUOTE DrawBlackBox))
    (WINDOWPROP window (QUOTE BUTTONEVENTFN)
      (QUOTE BlackBoxWindowFn))
    (NewGame window boxSize)
    window])
  (* edited: " 3-Jan-86 18:51")
```

**(MoveAhead**

```
[LAMBDA (array x y boxSize direction)
  (MoveBall array (SELECTQ direction
    (L (SUB1 x))
    (R (ADD1 x))
    x)
    (SELECTQ direction
      (D (SUB1 y))
      (U (ADD1 y))
      y)
    boxSize direction))
  (* edited: "29-Dec-85 18:03")
```

**(MoveBall**

```
[LAMBDA (array xPos yPos boxSize direction)
  (LET ((edge (OnEdge xPos yPos boxSize)))
    (if edge
      then
        (* Coming Out)
        (CONS xPos yPos)
      elseif (BallAhead array xPos yPos boxSize direction)
        then (QUOTE H)
      elseif (LeftAhead array xPos yPos boxSize direction)
        then (if (RightAhead array xPos yPos boxSize direction)
          then (QUOTE R)
          else (MoveAhead array xPos yPos boxSize (RightTurn direction)))
      elseif (RightAhead array xPos yPos boxSize direction)
        then (MoveAhead array xPos yPos boxSize (LeftTurn direction))
      else (MoveAhead array xPos yPos boxSize direction))
  (* edited: "29-Dec-85 18:22")
```

**(DrawBlackBox**

```
[LAMBDA (window)
  (LET* [(boxSize (WINDOWPROP window (QUOTE BoxSize)))
    (boxWidth (WINDOWPROP window (QUOTE BoxWidth)))
    (lastLinePos (DIFFERENCE boxWidth BlackBoxSquare))
  (* edited: "10-Jan-86 07:40")
```

```

(guessArray (WINDOWPROP window (QUOTE GuessArray]
(for bottom from BlackBoxSquare by BlackBoxSquare to boxWidth
  do (DRAWLINE 0 bottom boxWidth bottom (if (OR (EQ bottom BlackBoxSquare)
                                                (EQ bottom lastLinePos))
      then 4
      else 2)
      NIL window)
      (DRAWLINE bottom 0 bottom boxWidth (if (OR (EQ bottom BlackBoxSquare)
                                                  (EQ bottom lastLinePos))
      then 4
      else 2)
      NIL window))
(for xPos from 0 to (SUB1 boxSize) do (for yPos from 0 to (SUB1 boxSize)
  do (FillBox window xPos yPos (SquareArray guessArray xPos yPos
      boxSize])

```

**(BallAhead**

```

[LAMBDA (array x y boxSize direction) (* edited: "29-Dec-85 17:29")
  (EQ (QUOTE Ball)
    (SquareArray array (SELECTQ direction
      (L (SUB1 x))
      (R (ADD1 x))
      x)
    (SELECTQ direction
      (D (SUB1 y))
      (U (ADD1 y))
      y)
    boxSize])

```

**(BallDownOrUp**

```

[LAMBDA (window xBox yBox) (* edited: "29-Dec-85 14:33")
  (LET* [(array (WINDOWPROP window (QUOTE GuessArray))
    (boxSize (WINDOWPROP window (QUOTE BoxSize]
    (SetSquareArray array xBox yBox boxSize (if (SquareArray array xBox yBox boxSize)
      then NIL
      else (QUOTE Ball])

```

**(BBBoxNumber**

```

[LAMBDA (window place) (* dbg: "25-Dec-85 16:54")
  (IQUOTIENT place BlackBoxSquare])

```

**(BlackBoxWindowFn**

```

[LAMBDA (window) (* edited: "10-Jan-86 07:49")
  (LET [(buttons (DECODEBUTTONS))
    (xBox (BBBoxNumber window (LASTMOUSEX window)))
    (yBox (BBBoxNumber window (LASTMOUSEY window)))
    (lastRow (SUB1 (WINDOWPROP window (QUOTE BoxSize]
    (if (EQ yBox (WINDOWPROP window (QUOTE BoxSize))
      then (AND buttons (BlackBoxChoices window))
      elseif buttons
      elseif (OR (EQ xBox 0)
        (EQ yBox 0)
        (EQ xBox lastRow)
        (EQ yBox lastRow))
      then [LET ((answer (AskQuestion window xBox yBox))
        (RedisplayBox window xBox yBox)
        (AND (LISTP answer)
          (RedisplayBox window (CAR answer)
            (CDR answer]
      else (BallDownOrUp window xBox yBox)
        (RedisplayBox window xBox yBox])

```

**(MakeBallArray**

```

[LAMBDA (numBalls) (* edited: "29-Dec-85 14:15")
  (ARRAY (TIMES numBalls numBalls)
    NIL NIL 0])

```

**(NewGame**

```

[LAMBDA (window boxSize ballPositions) (* edited: "10-Jan-86 08:27")
  [OR boxSize (SETQ boxSize (WINDOWPROP window (QUOTE BoxSize]
  (LET [(ballArray (OR (WINDOWPROP window (QUOTE BallArray))
    (LET ((V (MakeBallArray boxSize))
      (WINDOWPROP window (QUOTE BallArray)
        V))
    V))
    (guessArray (OR (WINDOWPROP window (QUOTE GuessArray))
      (LET ((V (MakeBallArray boxSize))
        (WINDOWPROP window (QUOTE GuessArray)
          V))
      V]
  (InitializeGuessArray guessArray boxSize ballPositions)

```

```
(InitializeBallArray ballArray (WINDOWPROP window (QUOTE NumBalls))
  boxSize ballPositions)
(WINDOWPROP window (QUOTE LastGuessNumber)
  1)
(REDISPLAYW window])
```

(OnEdge

```
[LAMBDA (x y boxSize) (* edited: "10-Jan-86 06:01")
  (if (EQ y 0)
    then (QUOTE U)
    elseif (EQ x 0)
    then (QUOTE R)
    elseif (EQ y (SUB1 boxSize))
    then (QUOTE D)
    elseif (EQ x (SUB1 boxSize))
    then (QUOTE L)
    else NIL])
```

(ProbeBallArray

```
[LAMBDA (array xPos yPos boxSize) (* edited: "10-Jan-86 07:06")
  (* Returns NIL if at corner, H, R, or for a detour a dotted pair of final position for x and y)

  (LET* [(lastIndex (SUB1 boxSize))
    [atCorner (OR (AND (EQ xBox 0)
      (OR (EQ yBox 0)
        (EQ yBox lastIndex)))
      (AND (EQ xBox lastIndex)
        (OR (EQ yBox 0)
          (EQ yBox lastIndex))
      (direction (AND (NOT atCorner)
        (OnEdge xPos yPos boxSize)
      (AND direction (if (BallAhead array xPos yPos boxSize direction)
        then (QUOTE H)
        elseif (OR (LeftAhead array xPos yPos boxSize direction)
          (RightAhead array xPos yPos boxSize direction))
        then (QUOTE R)
        else (MoveAhead array xPos yPos boxSize direction))])
```

(RedisplayBox

```
[LAMBDA (window xBox yBox) (* edited: "10-Jan-86 07:48")
  (FillBox window xBox yBox (SquareArray (WINDOWPROP window (QUOTE GuessArray))
    xBox yBox (WINDOWPROP window (QUOTE BoxSize))])
```

(RightAhead

```
[LAMBDA (array x y boxSize direction) (* edited: "29-Dec-85 17:35")
  (EQ (QUOTE Ball)
    (SquareArray array (SELECTQ direction
      ((D L)
        (SUB1 x))
      ((U R)
        (ADD1 x))
      x)
    (SELECTQ direction
      ((R D)
        (SUB1 y))
      ((L U)
        (ADD1 y))
      y)
    boxSize])
```

(RightTurn

```
[LAMBDA (direction) (* edited: "29-Dec-85 18:07")
  (SELECTQ direction
    (U (QUOTE R))
    (R (QUOTE D))
    (D (QUOTE L))
    (L (QUOTE U))
    (ERROR "Bad Direction" direction])
```

(SetSquareArray

```
[LAMBDA (array x y rowSize newValue) (* edited: "29-Dec-85 13:27")
  (SETA array (PLUS x (TIMES rowSize y))
    newValue])
```

(ShowBalls

```
[LAMBDA (window ballArray gameArray showAll) (* edited: "29-Dec-85 17:00")
  (LET [(boxSize (WINDOWPROP window (QUOTE BoxSize))
    (for xPos from 0 to (SUB1 boxSize)
      do (for yPos from 0 to (SUB1 boxSize)
```

```
do (if (AND (EQ (QUOTE Ball)
                (SquareArray ballArray xPos yPos boxSize))
        (OR showAll (SquareArray gameArray xPos yPos boxSize)))
    then (DSPFILL (CREATEREGION (PLUS (TIMES xPos BlackBoxSquare)
                                     2)
                                   (PLUS (TIMES yPos BlackBoxSquare)
                                         2)
                                   (DIFFERENCE BlackBoxSquare 2)
                                   (DIFFERENCE BlackBoxSquare 2))
              BLACKSHADE
              (QUOTE INVERT)
              window])
```

**(SquareArray**

```
[LAMBDA (array x y rowSize)
  (ELT array (PLUS x (TIMES rowSize y))
```

(\* edited: "29-Dec-85 13:26")

)

(PUTPROPS **BLACKBOX COPYRIGHT** ("Xerox Corporation" 1985 1986))

---

**FUNCTION INDEX**

AskQuestion .....	1	BlackBoxWindowFn .....	4	MakeBallArray .....	4	RedisplayBox .....	5
BallAhead .....	4	DrawBlackBox .....	3	MakeBlackBoxWindow .....	3	RightAhead .....	5
BallDownOrUp .....	4	FillBox .....	2	MoveAhead .....	3	RightTurn .....	5
BBBoxNumber .....	4	InitializeBallArray .....	2	MoveBall .....	3	SetSquareArray .....	5
BlackBox .....	1	InitializeGuessArray .....	2	NewGame .....	4	ShowBalls .....	5
BlackBoxChoices .....	1	LeftAhead .....	2	OnEdge .....	5	SquareArray .....	6
BlackBoxTitle .....	2	LeftTurn .....	3	ProbeBallArray .....	5		

---

**VARIABLE INDEX**

BlackBoxSquare .....	1	HintTime .....	1
----------------------	---	----------------	---

---