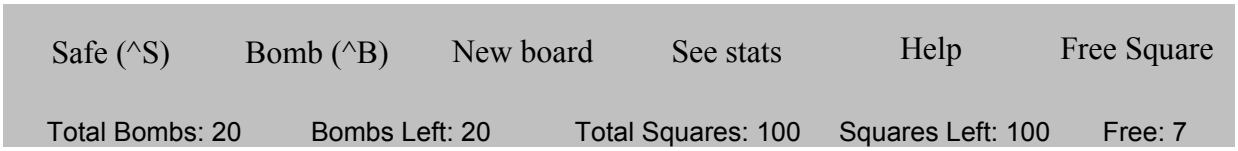

Safe (^S)
Bomb (^B)
New board
See stats
Help
Free Square
Total Bombs: 20
Bombs Left: 20
Total Squares: 100
Squares Left: 100
Free: 7

Return to board

Statistics

Total Squares Checked:	0
Total Squares Left:	100
Total Squares:	100
Bombs Found:	0
Bombs Left:	20
Total Bombs:	20
Safe Squares Found:	0
Safe Squares Left:	80
Total Safe Squares:	80
Free Squares Used:	0
Free Squares Left:	7
Total Free Squares:	7

Games Won:	0
Games Lost:	2

Reset Wc

Set Up Information

Total Columns:	10
Total Rows:	10
Total Squares:	100
Total Bombs:	20
Total Free:	7
Smart Bombs:	No

Game Status: Game is in progress.


ɔn/Lost

Return to board

Quattro Pro Mine Finder

The object of this game is to determine where the m

If you think that a square contains a mine, select it

If you think that a square does not contain a mine,

You win if you either uncover all of the "safe" squar

You lose if you guess wrong about a square, ie: yo "safe", or vice-versa.

Safe squares which have been uncovered show th orthogonal and diagonal directions (from 0 to 8 bor

You can select a CONTIGUOUS block (with the m whole block at once as either "Safe" or "Bomb". Y already marked squares, they are simply ignored. with no mines surrpounding it, simply mark the 3x3 it "safe".

Use the "free square" button to let the computer fin from upper left to lower right for a square with zero tries for a square with one or seven bombs, then tw the game and other ambiguous circumstances.

To start a new game, press the "New board" buttor fields or select "custom" to set up your own mine fi

	Cols	Rows	Bombs	Free Squares	Sm Bombs	Difficulty	Time for new board
Runny	5	5	3	2	Yes	Trivial	5 sec
Over Easy	10	10	20	7	No	Easy	30 sec
Over Medium	27	16	100	20	No	Medium	1.5 min
Over Hard	27	16	200	10	Yes	Hard	2 min
Burned	50	100	2000	100	No	Ridiculous	17 min
Custom				Your own settings			5 sq / sec
Maximums	200	5000	30000	5000	---	---	---

"Smart bombs", when uncovered, show the number of bombs adjacent to them, just like

uncovered "safe" squares.

The "smart bomb" feature can make the game very mechanical, as it allows you to always uncover a rectangular block of the mine field without having to guess, however, an interesti variation using "smart bombs" is to configure a board with mostly bombs, say 80%. Start by assuming a square IS a bomb (or use "Free Square"), and proceed from there.

The "time for new board" is the amount of time it takes my 486/33 to create a new board.

You can interrupt any operation with "Ctrl+Break" without wrecking anything.

When you exit the game, save it to retain the game statistics. In fact, you can save a game progress and continue where you left off when you reload it.

The "Stats" page shows more complete information about the current game than the inform shown on the board, and also number of wins and losses.

The "BombMap" page is a map of where the bombs are. Change the text color to anything but white to view the map. Bombs have values >99. Each number mod 100 is the number of adjacent bombs.

This game contains no startup or exit macros. Speaking of macros, cells on the macro pag marked in blue are unprotected cells for variables. Cells in magenta contain formulas. Cells in bold italics are comments and everything else is code.

The only variables used are the bomb map, which also contains the adjacency information and some work variables.

This game does not change any application properties of Quattro Pro.

This spreadsheet was programmed by:

Robert M. Peffer
The Crane-Peffer
1021 Shelter Cove Road
P.O. Box 247
Whitethorn, CA 95589
(707) 986-7811
CompuServe ID: 70621, 1702

mines are in a mine field.

and then press the ¨Bomb¨ button (or Ctrl+B).

select it and then press ¨Safe¨ (or Ctrl+S).

res or if you uncover all of the bombs.

u declare a square which has a bomb to be

e number of bombs adjacent to it, along
nbs possible).

ouse or SHIFT + arrow keys) and mark the
ou can allow the selected block to cover
For instance, if you uncover a ¨0¨ square,
block which surrounds the ¨0¨ and mark

d a ¨safe¨ square for you. The computer searches
or eight bombs surrounding it. Failing that it then
wo or six bombs, etc. Use this feature when starting

n. You can select from one of the presized mine
eld. the predefined mine fields are as follows:

ng

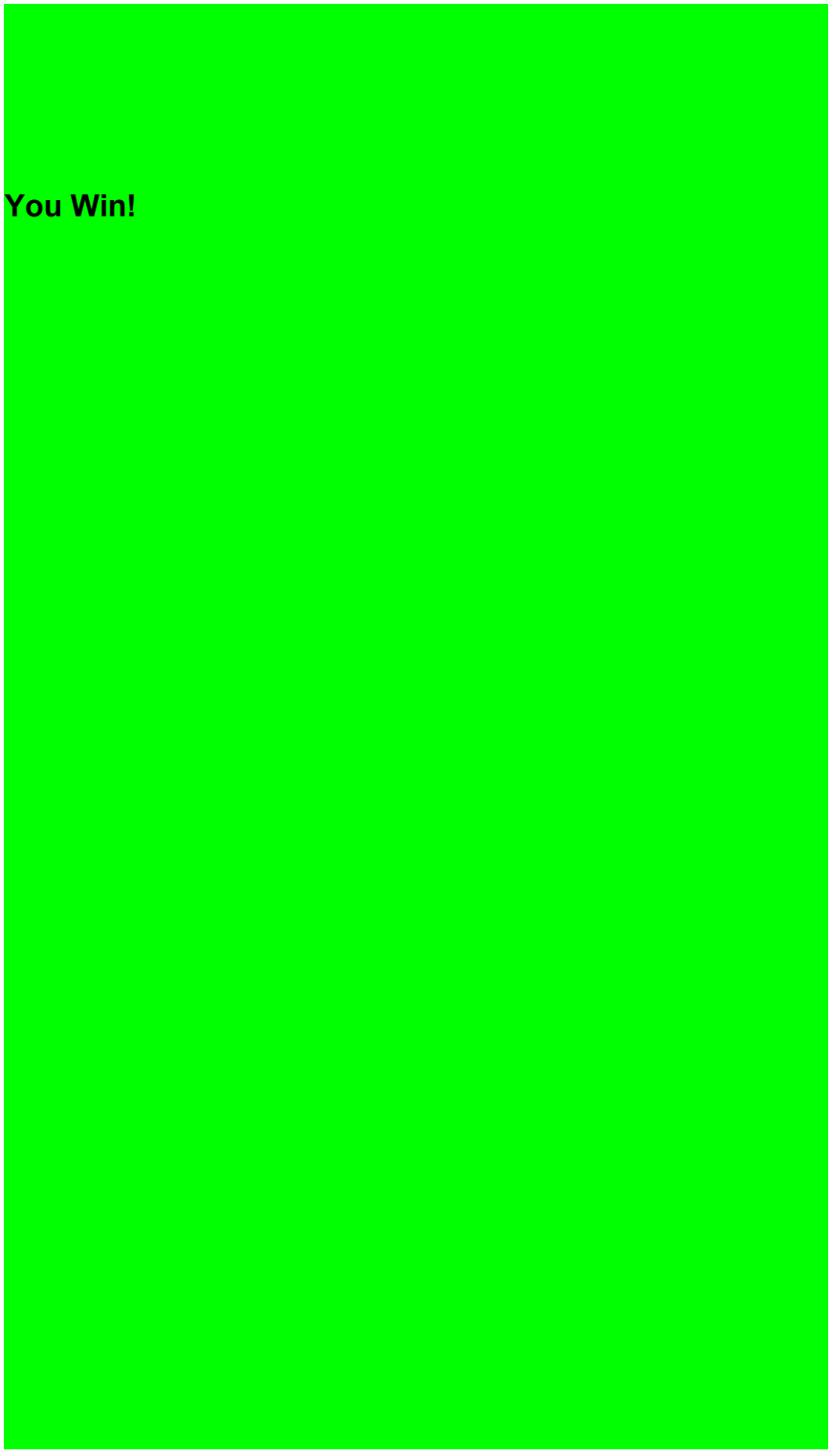
e in

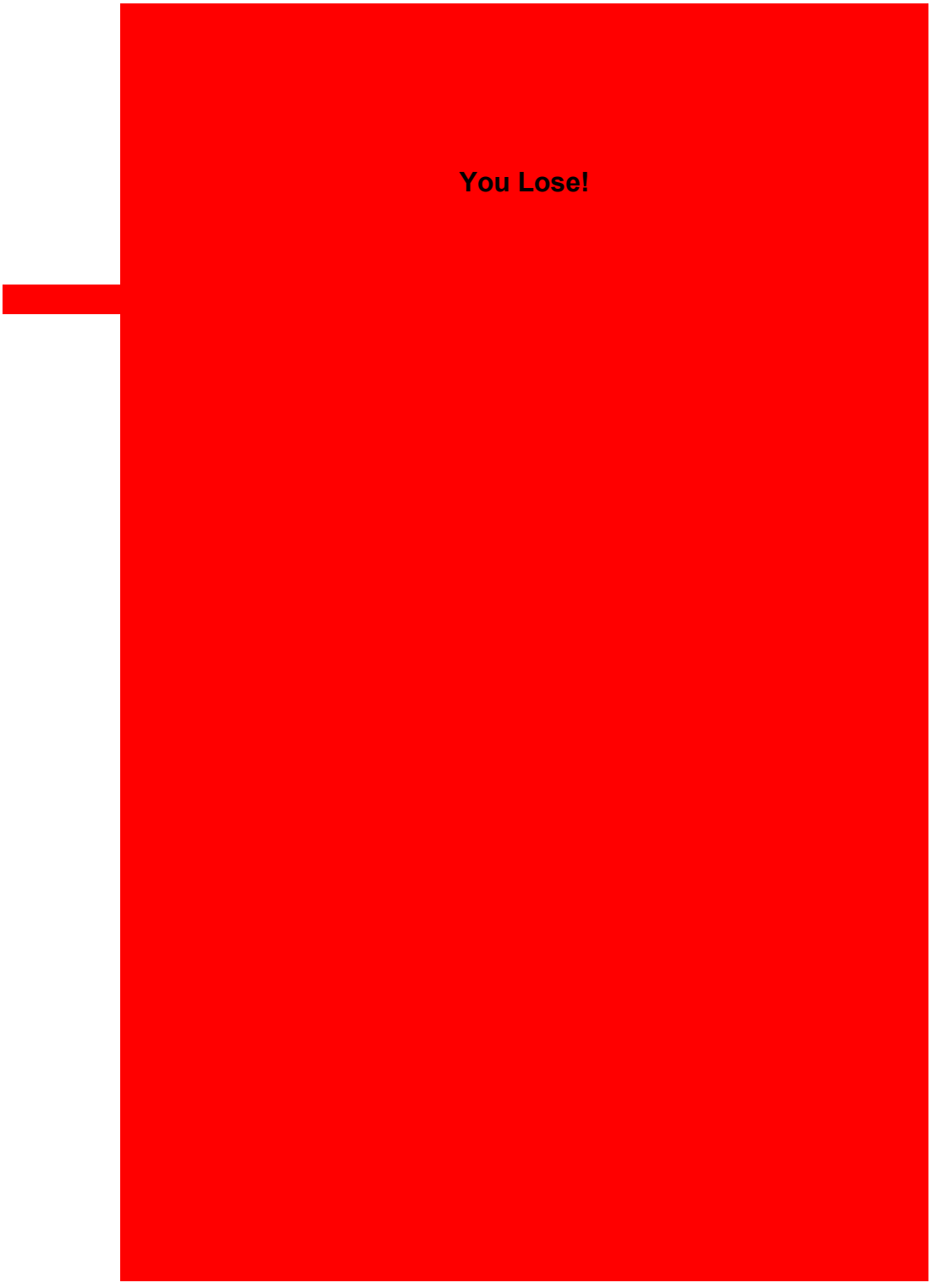
ıation

ge

:, so long as
ticular way,
xpressed or

Return to board See stats




You Lose!

Macros

Values returned from LevelDlg

DlgLevel	1	***Playing level selected by user***

Values returned from SetupDlg (block is called SetupVars)

DlgCols	10	***Number of rows in bomb field - 1***
DlgRows	10	***Number of columns in bomb field - 1***
MaxBombs	20	***Number of bombs in bomb field***
MaxFree	7	***Number of free squares allowed***
SmartBombs	0	***Smart bombs? 1=yes***

Values returned from LoseDlg

LoseOpt	0	***Lose option selected by user***

Other variables

MaxRow	9	***Number of rows in bomb field - 1***
MaxCol	9	***Number of columns in bomb field - 1***
DlgCls	1	***Dialog box close flag... 1=Ok button***
WrkRow	10	***Current row***
WrkCol	0	***Current column***
BombCtr	100	***Number of bombs laid***
CtrNo	0	***Number of safe squares revealed***
CtrYes	0	***Number of bombs revealed***
CtrFree	0	***Number of free squares used***
Done	0	***Game status: 0=playing, 1=lost, 2=won, 3=no board***
SelBlock	Board:B2	***Block selected by user for checking***
SelRow	0	***Current row of SelBlock being checked***
SelCol	0	***Current column of SelBlock being checked***
FreeRow	0	***Current row of Free Square search***
FreeCol	0	***Current Column of Free Square search***
FreeBombs	0	***Current number of bombs for Free Square search***
GamesWon	0	***Number of games won***
GamesLost	2	***Number of games lost***

Generate the board

```
GenBoard    {}
            {DoDialog LevelDlg,DlgCls,DlgLevel}
            {If DlgCls<>1}{Branch GenBoard900}
            {If DlgLevel=5}{Branch GenBoard001}
            {Let DlgCols,@CHOOSE(DlgLevel,5,10,27,27,50)}
            {Let DlgRows,@CHOOSE(DlgLevel,5,10,16,16,100)}
            {Let MaxBombs,@CHOOSE(DlgLevel,3,20,100,200,2000)}
            {Let MaxFree,@CHOOSE(DlgLevel,2,7,20,10,100)}
            {Let SmartBombs,@CHOOSE(DlgLevel,1,0,0,1,0)}
            {Branch GenBoard002}

GenBoard001 {DoDialog SetupDlg,DlgCls,SetupVars}
            {If DlgCls<>1}{Branch GenBoard900}
```

```
                {Branch GenBoard002}

GenBoard002     {}
                {ReCalc MaxRow}{ReCalc MaxCol}
                {Blank Board:}{SetObjectProperty "Board:.Shading","3,1,Blend7"}
                {OnError GenBoard800}
                {Indicate "WAIT"}{WaitCursorOn}
                {Let BombCtr,0}
                {Blank BombMap:}{Let Done,3}{Let SelBlock,"Board:B2"}
GenBoard004     {Let WrkRow,@INT(@RAND*MaxRow+0.5)}
                {Let WrkCol,@INT(@RAND*MaxCol+0.5)}
                {If @CELLINDEX("type",BombMap:,WrkCol,WrkRow)<>"b"}{Branch GenBoard004}
                {Put BombMap:,WrkCol,WrkRow,100}
                {Let BombCtr,+BombCtr+1}
                {ReCalc GenBoard005}
GenBoard005     {SelectBlock Board:J7}&apos;250.0% done~
                {If BombCtr<MaxBombs}{Branch GenBoard004}
                {}
                {Let WrkRow,0}{Let WrkCol,0}
GenBoard020     {Let BombCtr,@CELLINDEX("contents",BombMap:,WrkCol,WrkRow)}
                {If WrkCol>0#AND#WrkRow>0#AND#@CELLINDEX("contents",BombMap:,WrkCol-1
                {If WrkCol>0#AND#@CELLINDEX("contents",BombMap:,WrkCol-1,WrkRow)>=100}{I
                {If WrkCol>0#AND#WrkRow<MaxRow#AND#@CELLINDEX("contents",BombMap:,W
                {If WrkRow>0#AND#@CELLINDEX("contents",BombMap:,WrkCol,WrkRow-1)>=100}
                {If WrkRow<MaxRow#AND#@CELLINDEX("contents",BombMap:,WrkCol,WrkRow+1
                {If WrkCol<MaxCol#AND#WrkRow>0#AND#@CELLINDEX("contents",BombMap:,Wr
                {If WrkCol<MaxCol#AND#@CELLINDEX("contents",BombMap:,WrkCol+1,WrkRow)>
                {If WrkCol<MaxCol#AND#WrkRow<MaxRow#AND#@CELLINDEX("contents",BombM
                {Put BombMap:,WrkCol,WrkRow,+BombCtr}
                {Let WrkCol,+WrkCol+1}
                {If WrkCol<=MaxCol}{Branch GenBoard020}
                {ReCalc GenBoard030}
GenBoard030     {SelectBlock Board:J7}&apos;105.6% done~
                {Let WrkRow,+WrkRow+1}
                {Let WrkCol,0}
                {If WrkRow<=MaxRow}{Branch GenBoard020}
                {}
                {Let CtrNo,0}{Let CtrYes,0}
                {Let CtrFree,0}{Let FreeRow,0}{Let FreeCol,0}{Let FreeBombs,0}
                {}
                {Blank Board:}
                {SelectBlock Board:B1}+"Total Bombs: "&@STRING(MaxBombs,0)~
                {SelectBlock Board:H1}+"Bombs Left: "&@STRING(MaxBombs-CtrYes,0)~
                {SelectBlock Board:N1}+"Total Squares: "&@STRING(DlgRows*DlgCols,0)~
                {SelectBlock Board:T1}+"Squares Left: "&@STRING(DlgRows*DlgCols-CtrNo-CtrYes
                {SelectBlock Board:Z1}+"Free: "&@STRING(MaxFree-CtrFree,0)~
                {}
                {ReCalc GenBoard200}
GenBoard200       #NAME?
```

```
                {Let Done,0}
                {WaitCursorOff}{Indicate}
                {OnError}
                {SelectBlock Board:B2}
                {WindowMaximize}

GenBoard800     {}
                {Blank BombMap:}{Blank Board:}
                {WaitCursorOff}{Indicate}

GenBoard900     {}
```

Check for safe squares

```
\S              {}
CkNo            {}
                {If Done<>0}{Branch CkNo099}
                {Let SelBlock,@PROPERTY("Active_Block.Selection")}
                {Let SelCol,0}{Let SelRow,0}
CkNo005         {Let WrkCol,@CELLINDEX("col",@@(SelBlock),SelCol,SelRow)-2}
                {Let WrkRow,@CELLINDEX("row",@@(SelBlock),SelCol,SelRow)-2}
                {If WrkCol<0#OR#WrkCol>MaxCol}{Branch CkNo099}
                {If WrkRow<0#OR#WrkRow>MaxRow}{Branch CkNo099}
                {If @CELLINDEX("type",Board:,WrkCol+1,WrkRow+1)<>"b"}{Branch CkNo099}
                {If @CELLINDEX("contents",BombMap:,WrkCol,WrkRow)>=100}{Branch CkNo500}
                {Put Board:,WrkCol+1,WrkRow+1,@CELLINDEX("contents",BombMap:,WrkCol,WrkF
                {ReCalc CkNo010}
CkNo010             #NAME?
                {Let CtrNo,+CtrNo+1}
CkNo099         {Let SelCol,+SelCol+1}
                {If SelCol<@COLS(@@(SelBlock))}{Branch CkNo005}
                {Let SelCol,0}
                {Let SelRow,+SelRow+1}
                {If SelRow<@Rows(@@(SelBlock))}{Branch CkNo005}
                {If CtrNo=DlgRows*DlgCols-MaxBombs}{Branch Win}
```

User thought a bomb was a safe square

```
CkNo500         {}
                {ReCalc CkNo510}
CkNo510             #NAME?
                {SetObjectProperty "Active_Block.Shading","0,4,Blend4"}
                X~
                {Branch Lose}
```

Check for bombs

```
\B              {}
CkYes           {}
                {If Done<>0}{Branch CkYes099}
                {Let SelBlock,@PROPERTY("Active_Block.Selection")}
                {Let SelCol,0}{Let SelRow,0}
```

```
CkYes005    {Let WrkCol,@CELLINDEX("col",@@(SelBlock),SelCol,SelRow)-2}
            {Let WrkRow,@CELLINDEX("row",@@(SelBlock),SelCol,SelRow)-2}
            {If WrkCol<0#OR#WrkCol>MaxCol}{Branch CkYes099}
            {If WrkRow<0#OR#WrkRow>MaxRow}{Branch CkYes099}
            {If @CELLINDEX("type",Board:,WrkCol+1,WrkRow+1)<>"b"}{Branch CkYes099}
            {If @CELLINDEX("contents",BombMap:,WrkCol,WrkRow)<100}{Branch Ckyes500}
            {If SmartBombs=0}{Put Board:,WrkCol+1,WrkRow+1,"Q"}
            {If SmartBombs=1}{Put Board:,WrkCol+1,WrkRow+1,@CELLINDEX("contents",Bomb
            {ReCalc CkYes010}
CkYes010      #NAME?
            {Let CtrYes,+CtrYes+1}
CkYes099    {Let SelCol,+SelCol+1}
            {If SelCol<@COLS(@@(SelBlock))}{Branch CkYes005}
            {Let SelCol,0}
            {Let SelRow,+SelRow+1}
            {If SelRow<@Rows(@@(SelBlock))}{Branch CkYes005}
            {If CtrYes=MaxBombs}{Branch Win}
```

User thought a safe square was a bomb

```
CkYes500    {}
            {ReCalc CkYes510}
CkYes510      #NAME?
            {SetObjectProperty "Active_Block.Shading","0,5,Blend4"}
            X~
            {Branch Lose}
```

User got it wrong

```
Lose        {}
            {Beep 4}{Beep 3}{Beep 2}{Beep 1}
            {EditGoto YouLose:A1}{EditGoto YouLose:B2}
Lose002     {DoDialog LoseDlg,DlgCls,LoseOpt}
            {If DlgCls<>1}{Branch Lose002}
            {If LoseOpt=0}{Branch Lose200}
            {If LoseOpt=2}{Branch Lose300}
            {Let GamesLost,+GamesLost+1}
            {Let YouLose:B6,"Please wait for the computer to find the mines..."}
            {SetObjectProperty "YouLose:B6.Alignment","Center"}
            {Let YouLose:B8,"(Press Ctrl+Break to interrupt the search)"}
            {SetObjectProperty "YouLose:B8.Alignment","Center"}
            {Let Done,1}
            {OnError Lose800}
            {Indicate "WAIT"}{WaitCursorOn}
            {Let WrkCol,0}{Let WrkRow,0}
Lose010     {If @CELLINDEX("contents",BombMap:,WrkCol,WrkRow)<100}{Branch Lose099}
            {If @CELLINDEX("type",Board:,WrkCol+1,WrkRow+1)<>"b"}{Branch Lose099}
            {If SmartBombs=0}{Put Board:,WrkCol+1,WrkRow+1,"Q"}
            {If SmartBombs=1}{Put Board:,WrkCol+1,WrkRow+1,@CELLINDEX("contents",Bomb
            {ReCalc Lose020}
Lose020       #NAME?
```

```
Lose099         {Let WrkCol,+WrkCol+1}
                {If WrkCol<=MaxCol}{Branch Lose010}
                {Let WrkRow,+WrkRow+1}
                {Let WrkCol,0}
                {If WrkRow<=MaxRow}{Branch Lose010}
                {WaitCursorOff}{Indicate}
                {OnError}
                {Let YouLose:B6,""}{Let YouLose:B8,""}
                {GoBoard}

Lose200         {}
                {Let GamesLost,+GamesLost+1}
                {Let Done,1}
                {GoBoard}

Lose300         {}
                {ReCalc Lose310}
Lose310             #NAME?
                {Del}
                {If @CELLINDEX("contents",BombMap:,WrkCol,WrkRow)>=100}{Branch Lose350}
                {Put Board:,WrkCol+1,WrkRow+1,@CELLINDEX("contents",BombMap:,WrkCol,WrkF
Block.Shading","3,5,Blend7"}
                {Let CtrNo,+CtrNo+1}
                {If CtrNo=DlgRows*DlgCols-MaxBombs}{Branch Win}

Lose350         {}
                {If SmartBombs=0}{Put Board:,WrkCol+1,WrkRow+1,"Q"}
                {If SmartBombs=1}{Put Board:,WrkCol+1,WrkRow+1,@CELLINDEX("contents",Bomb
Block.Shading","3,4,Blend7"}
                {Let CtrYes,+CtrYes+1}
                {If CtrYes=MaxBombs}{Branch Win}

Lose800         {}
                {Let YouLose:B6,""}{Let YouLose:B8,""}
                {WaitCursorOff}{Indicate}
                {GoBoard}
```

User found all safe squares or all bombs

```
Win             {}
                {Beep 1}{Beep 4}{Beep 1}{Beep 4}
                {Let GamesWon,+GamesWon+1}
                {Let Done,2}
                {EditGoto YouWin:A1}{EditGoto YouWin:B2}
```

Show statistics page

```
GoStats         {}
                {Let WrkCol,@CELLPOINTER("col")-2}
                {Let WrkRow,@CELLPOINTER("row")-2}
                {EditGoto Stats:A1}
```

Return from statistics page

```
GoBoard         {}
                {SelectBlock Board:A1}
                {ReCalc GoBoard010}
GoBoard010      {SelectBlock Board:B2}
```

Show help screen

```
GoHelp          {}
                {Let WrkCol,@CELLPOINTER("col")-2}
                {Let WrkRow,@CELLPOINTER("row")-2}
                {EditGoto Help:A1}

DoReset         {}
                {Let GamesWon,0}
                {Let GamesLost,0}
```

Show a free square

```
DoFree          {}
                {If Done<>0}{Branch DoFree900}
                {If CtrFree=MaxFree}{Branch DoFree900}
                {OnError DoFree800}
                {Indicate "WAIT"}{WaitCursorOn}
DoFree010       {If @CELLINDEX("type",Board:,FreeCol+1,FreeRow+1)<>"b"}{Branch DoFree020}
                {If @CELLINDEX("contents",BombMap:,FreeCol,FreeRow)=FreeBombs}{Branch DoF
                {If FreeRow>0#AND#FreeRow<MaxRow#AND#FreeCol>0#AND#FreeCol<MaxCol#A
                {If (FreeRow=0#OR#FreeRow=MaxRow)#AND#FreeCol>0#AND#FreeCol<MaxCol#A
                {If (FreeCol=0#OR#FreeCol=MaxCol)#AND#FreeRow>0#AND#FreeRow<MaxRow#A
                {If (FreeRow=0#OR#FreeRow=MaxRow)#AND#(FreeCol=0#OR#FreeCol=MaxCol)#A
DoFree020       {Let FreeCol,+FreeCol+1}
                {If FreeCol<=MaxCol}{Branch DoFree010}
                {Let FreeRow,+FreeRow+1}
                {Let FreeCol,0}
                {If FreeRow<=MaxRow}{Branch DoFree010}
                {Let FreeBombs,+FreeBombs+1}
                {Let FreeRow,0}
                {If FreeBombs<=8}{Branch DoFree010}
                {WaitCursorOff}{Indicate}
                {OnError}


DoFree200       {}
                {Put Board:,FreeCol+1,FreeRow+1,@CELLINDEX("contents",BombMap:,FreeCol,Fre
                {Let CtrNo,+CtrNo+1}
                {Let CtrFree,+CtrFree+1}
                {ReCalc DoFree210}
DoFree210           #NAME?
                {SetObjectProperty "Active_Block.Shading","3,5,Blend7"}
                {WaitCursorOff}{Indicate}
```

```
                {OnError}
                {If CtrNo=DlgRows*DlgCols-MaxBombs}{Branch Win}

DoFree800       {}
                {WaitCursorOff}{Indicate}

DoFree900       {}
```

,WrkRow-1)>=100}{Let BombCtr,+BombCtr+1}
Let BombCtr,+BombCtr+1}
/rkCol-1,WrkRow+1)>=100}{Let BombCtr,+BombCtr+1}
{Let BombCtr,+BombCtr+1}
)>=100}{Let BombCtr,+BombCtr+1}
kCol+1,WrkRow-1)>=100}{Let BombCtr,+BombCtr+1}
=100}{Let BombCtr,+BombCtr+1}
Map:,WrkCol+1,WrkRow+1)>=100}{Let BombCtr,+BombCtr+1}

,0)~

{(wo

›Map:,WrkCol,WrkRow)-100}

›Map:,WrkCol,WrkRow)-100}

ɾow)}

›Map:,WrkCol,WrkRow)-100}

ree200}
ND#@CELLINDEX("contents",BombMap:,FreeCol,FreeRow)=8-FreeBombs}{Branch DoFree200}
ND#@CELLINDEX("contents",BombMap:,FreeCol,FreeRow)=5-FreeBombs}{Branch DoFree200}
ND#@CELLINDEX("contents",BombMap:,FreeCol,FreeRow)=5-FreeBombs}{Branch DoFree200}
ND#@CELLINDEX("contents",BombMap:,FreeCol,FreeRow)=3-FreeBombs}{Branch DoFree200}

eRow)}

```
}
}
}
}
```

100	1	2	101	2	1	100	2
1	1	2	101	2	1	1	2
0	0	1	1	2	1	1	1
0	0	0	1	2	101	1	0
1	2	2	2	101	3	2	1
1	101	102	4	4	4	101	1
1	2	3	102	103	103	3	1
1	1	2	3	5	103	3	1
100	1	1	100	2	2	101	1
1	1	1	1	1	1	1	1

1	1
100	1
1	1
1	1
1	100
1	1
1	1
1	100
2	2
1	100