'\*\*\* Ant\_Trap.Jb - Game to box in Ant \*\*\*

'\*\*\* Code placed in the public domain by AltBas of the Just BASIC forums \*\*\*

'\* Ant artificial intelligence is an oxymoron \*

'\* Maybe make the ant seek out open corners to squeeze out of a closing box easier. \*

NOMAINWIN

WindowWidth = 620

WindowHeight = 730

gNorth = 0 : gSouth = 31 : gWest = 0 : gEast = 31

dim F(1, 1) '\* Playing field "map"; a wall is -1 \*

gosub [MakeAnt]

BUTTON #at.bNew "New Game", [Start], UL, 5, 5, 80, 25

BUTTON #at.bQuit "Quit", [Done], UL, 520, 5, 80, 25

STATICTEXT #at "Try to box the ant in by connecting cells", 150, 8, 350, 25

STATICTEXT #at.stMsg "", 200, 640, 230, 30

STATICTEXT #at.stMov "", 10, 640, 150, 30

GRAPHICBOX #at.gb 5, 35, 600, 600

open "Ant Trap" for window as #at

#at "trapclose [Done]"

#at.stMsg "!font 14"

[Start] '\* Draw grid \*

#at.stMsg "Waiting Your Move"

#at.stMov "Moves: 0"

#at.gb "when leftButtonUp [PlayerClick]"

redim F(gEast - 1, gSouth - 1)

AntTrap = 0

Moves = 0 '\* Clear counter \*

#at.gb "cls"

#at.gb "down"

for x = 20 to 580 step 20

#at.gb "line ";x;" 0 ";x;" 600"

next x

for y = 20 to 580 step 20

#at.gb "line 0 ";y;" 600 ";y

next y

#at.gb "flush"

'\* Ant starts in inner 10 x 10 grid, so doesn't start trapped \*

AX = int(rnd(a) \* 10) + 10 '\* Assign random Ant location \*

AY = int(rnd(a) \* 10) + 10

F(AX, AY) = -2

gosub [DrawAnt]

wait

[Done]

#at.gb "when leftButtonUp" '\* Turn off event \*

unloadbmp "ANT"

#at.gb "cls"

close #at

END

[PlayerClick]

PX = int(MouseX / 20) + 1

PY = int(MouseY / 20) + 1

select case F(PX, PY)

case -2 : NOTICE "Whoa!";chr$(13); "You can't click ON the Ant!";chr$(13);"Try again!" : wait

case -1 : NOTICE "Whoa!";chr$(13); "You can't click ON your Wall!";chr$(13);"Try again!" : wait

case 0 : gosub [DrawBrick] : Moves = Moves + 1

#at.stMov "Moves: ";using("###", Moves)

#at.stMsg "Ant Moving..."

gosub [AntMove]

if AntTrap = -1 then [GameOver]

end select

#at.stMsg "Waiting Your Move"

wait

[GameOver]

#at.stMsg "Congratulations!"

NOTICE "Congratulations!"; chr$(13); "You have trapped the Ant!"

#at.gb "when leftButtonUp" '\* Turn off event \*

wait

[AntMove]

'\* 0=N; 1=NE; 2=E; 3=SE; 4=S; 5=SW; 6=W; 7=NW \*

'\* The Ant can't pass on its move \*

'\* It may take awhile to find a single move... \*

IsMove = 0

AM = int(rnd(1) \* 8)

select case AM

case 0 '\* North \*

if AY - 1 > gNorth then

if F(AX, AY - 1) = -1 then

gosub [ChkWall]

if IsMove = -1 then [AntMove] '\* Try again \*

AntTrap = -1

else

nx = AX : ny = AY - 1

end if

else

goto [AntMove] '\* Hit wall - Try again \*

end if

case 1 '\* North-East \*

if AY - 1 > gNorth and AX + 1 < gEast then

if F(AX + 1, AY - 1) = -1 then

gosub [ChkWall]

if IsMove = -1 then [AntMove] '\* Try again \*

AntTrap = -1

else

nx = AX + 1 : ny = AY - 1

end if

else

goto [AntMove] '\* Hit wall - Try again \*

end if

case 2 '\* East \*

if AX + 1 < gEast then

if F(AX + 1, AY) = -1 then

gosub [ChkWall]

if IsMove = -1 then [AntMove] '\* Try again \*

AntTrap = -1

else

nx = AX + 1 : ny = AY

end if

else

goto [AntMove] '\* Try again \*

end if

case 3 '\* South-East \*

if AY + 1 < gSouth and AX + 1 < gEast then

if F(AX + 1, AY + 1) = -1 then

gosub [ChkWall]

if IsMove = -1 then [AntMove] '\* Try again \*

AntTrap = -1

else

nx = AX + 1 : ny = AY + 1

end if

else

goto [AntMove] '\* Hit wall - Try again \*

end if

case 4 '\* South \*

if AY + 1 < gSouth then

if F(AX, AY + 1) = -1 then

gosub [ChkWall]

if IsMove = -1 then [AntMove] '\* Try again \*

AntTrap = -1

else

nx = AX : ny = AY + 1

end if

else

goto [AntMove] '\* Hit wall - Try again \*

end if

case 5 '\* South-West \*

if AY + 1 < gSouth and AX - 1 > gWest then

if F(AX - 1, AY + 1) = -1 then

gosub [ChkWall]

if IsMove = -1 then [AntMove] '\* Try again \*

AntTrap = -1

else

nx = AX - 1 : ny = AY + 1

end if

else

goto [AntMove] '\* Hit wall - Try again \*

end if

case 6 '\* West \*

if AX - 1 > gWest then

if F(AX - 1, AY) = -1 then

gosub [ChkWall]

if IsMove = -1 then [AntMove] '\* Try again \*

AntTrap = -1

else

nx = AX - 1 : ny = AY

end if

else

goto [AntMove] '\* Hit wall - Try again \*

end if

case 7 '\* North-West \*

if AY - 1 > gNorth and AX - 1 > gWest then

if F(AX - 1, AY - 1) = -1 then

gosub [ChkWall]

if IsMove = -1 then [AntMove] '\* Try again \*

AntTrap = -1

else

nx = AX - 1 : ny = AY - 1

end if

else

goto [AntMove] '\* Hit wall - Try again \*

end if

end select

if AntTrap = 0 then

gosub [EraseAnt]

AX = nx : AY = ny

gosub [DrawAnt]

end if

RETURN

[ChkWall] '\* Look for a move \*

IsMove = 0

for x = -1 to 1

for y = -1 to 1

if ((AX + x > gWest) AND (AX + x < gEast)) AND ((AY + y > gNorth) AND (AY + y < gSouth)) then

if F(AX + x, AY + y) = 0 then IsMove = -1 : exit for '\* Empty cell \*

end if

next y

if IsMove = -1 then exit for

next x

RETURN

[DrawAnt]

if F(AX, AY) = -1 then NOTICE "Error..."

x = (AX - 1) \* 20 + 1

y = (AY - 1) \* 20 + 1

#at.gb "drawbmp ANT ";x;" ";y

#at.gb "flush"

F(AX, AY) = -2

RETURN

[EraseAnt]

#at.gb "backcolor white"

#at.gb "color white"

x = (AX - 1) \* 20 + 1

y = (AY - 1) \* 20 + 1

#at.gb "place ";x;" ";y

#at.gb "boxfilled "; x + 19;" "; y + 19

#at.gb "flush"

F(AX, AY) = 0

RETURN

[DrawBrick]

#at.gb "backcolor darkgray"

#at.gb "color darkgray"

x = (PX - 1) \* 20 + 1

y = (PY - 1) \* 20 + 1

#at.gb "place ";x;" ";y

#at.gb "boxfilled "; x + 19;" "; y + 19

#at.gb "flush"

F(PX, PY) = -1

RETURN

[MakeAnt]

data "1111111111111111111"

data "11111111 1111111"

data "1 11111 1111 1"

data "1 111 11 1"

data "11 111 11 11"

data "111 111 11 11"

data "111 111 11 111"

data "1 1 111 1"

data "11 11 11 11"

data "111 1 1 111"

data "111 111"

data "111111 111111"

data "111111 111111"

data "1111 1111"

data "111 111"

data "111 1 1 111"

data "111 1 1 111"

data "11 111 111 11"

data "1 111111111111111 1"

restore [MakeAnt]

open "Test" for graphics\_nf\_nsb as #at

#at "down ; color RED"

for y = 0 to 18

read a$

for x = 0 to 18

if mid$(a$, x+1, 1) = "1" then #at "set ";x;" ";y

next x

next y

#at "flush"

#at, "up"

#at "getbmp ANT 0 0 19 19"

#at "cls"

close #at

RETURN