

War-Tic

AiS Challenge
Final Report
April 3, 2002

010
Animas High School

Team Members
Kent Richardson
Mark Cavaliere
Joseph Hernandez
Ryan Richins

Teacher
Mr. Ed Kerr

Table of Contents

Table of Contents	2
Executive Summary	3
Body of Report	4
Source Code	6

Executive Summary

The name of this project is War-Tic. To put it simply, this is a cross between Tic-Tac-Toe and war. I don't know about you, but to me, Tic-Tac-Toe is a great pastime. It's provided hours of fun for my friends, family and I, especially in Algebra class. The company Hasbro has taken many old games, which many people consider outdated, and makes new games out of them. For example, Frogger, a game that was made for consoles like the Atari 2600 and the Commodore 64, has been made into a new game. Both players of the old game and people that hadn't even heard of the old game think that this is a great game.

This is what I planned for this project. Our teacher asked me one day what we were going to do our project on and a couple of classmates and I were playing Tic-Tac-Toe, so the idea jumped into my head: A new version of Tic-Tac-Toe. So I, the lead programmer, went home that night and wrote a code for the original Tic-Tac-Toe. I kept expanding on it and writing in more code, just adding things to the original – graphics, replays, animations, etc..

The goal of this project, I'll be honest with you, isn't to improve society in any way, shape, or form. Basically it is to just have fun. While the game itself isn't the most fun game in the world, making it was really the

fun part. Actually, we learned a lot by making it, too. For example, we became more familiar with the programming language and how to use it efficiently and effectively. Don't get me wrong, the game isn't completely lame, either. It is fun to play and it's something we're proud of.

Body of Report

This project is a game, not a typical application. The problem that it can solve is boredom. It sure did for us, because making it really did keep us busy for a while. Programming this project took quite a while. A few hours a day for two to three months really adds up. It was also fun to play the finished project. Some of you may think, “What can anyone do to Tic-Tac-Toe to make it fun?” Well, I’ll tell you.

First, graphics. Instead of scribbling X’s and O’s on a paper with four lines, there are graphics put in on the computer to represent these scribbles. Second, animations. Animations will make any game fun. Now instead of just bragging to your opponent, you can have the computer do it for you. Last, and most importantly, the battles. Instead of the players picking their spots and that’s the end of the game, the players now compete over the squares.

First off, the players choose a starting position. They start with a fortress, a number of units and resources. The resources can be used to train new units, or to build a new fortress. Special units can gather more resources, while the others are the warriors that conquer other areas. To win, a player has to have three fortresses in a row, defeat the other players

by getting rid of all their units, or achieve a state of peace between the two players. To fully comprehend the concepts of this game, you will have to play it.

The results of this project are: The game is a success. While the real satisfaction in this project is making it, the game is actually fun. We've learned a lot from the experience, and the next project we do will be even better. The main reason for this project's success is that the game actually runs and works. This gives us a sense of pride and satisfaction.

Source Code

```
DECLARE SUB Cheat (Quick, NoTower)
DECLARE SUB Delay (T!)
DECLARE SUB DispTurn (Turn!, HLyne$())
DECLARE SUB CatWin (Winner$)
DECLARE SUB OWin ()
DECLARE SUB Xwin ()
DECLARE SUB Credits ()
DECLARE SUB Dying (Bx!, By!, MapSpot$(), Xp!(), Yp!(), MXp!(), MYp!(),
Heal!(), Id!(), Task!(), Census!(), Units!())
DECLARE SUB Battling (up$, down$, lft$, rite$, esc$, Heal(), Id(),
MapSpot$(), Xp(), Yp(), Task(), NoTower)
DECLARE SUB Battle (Bx!, By!, Starter!, up$, down$, lft$, rite$, esc$,
NoPref!, Heal!(), Id!(), MapSpot$(), Xp!(), Yp!(), NoTower)
DECLARE SUB Ending (Win$)
DECLARE SUB Destination (MapSpot$(), Hall$(), Id!(), ToX!, ToY!)
DECLARE SUB Test (Census!(), Hall$(), Gold!(), Ore!())
DECLARE SUB Warping (up$, down$, lft$, rite$, esc$, MapSpot$(), Xp(),
Yp(), MXp(), MYp(), Hall$(), CondHall(), Team$(), Team, Id(), Heal(),
Task(), Task$(), Ore(), Gold(), Census(), Units(), Grid$(), Glyne,
MoveToX(), MoveToY(), Warper)
DECLARE SUB Training (up$, down$, lft$, rite$)
DECLARE SUB Mining ()
DECLARE SUB Repairing ()
DECLARE SUB Process (up$, down$, lft$, rite$, esc$, MapSpot$(), Xp(),
Yp(), MXp(), MYp(), Hall$(), CondHall(), Team$(), Id(), Heal(), Task(),
Task$(), Ore(), Gold(), Census(), Units(), Grid$(), Glyne, mx, my,
Warper, NoTower)
```

```

DECLARE SUB BuildingHalls (MapSpot$, Gold!(), Ore!(), Id!(), Team$(),
Hall$(), Task!(), CondHall!())
DECLARE SUB Warp (X!, Y!, MapSpot$, Xp!, Yp!, MXp!, MYp!, up$, down$,
lft$, rite$, esc$, mx!, my!, Grid$(), Glyne!, Turn!, Warper!)
DECLARE SUB Upgrade (MapSpot$, Id!(), Task!(), mx!, my!, TaskH!(),
Turn!)
DECLARE SUB Move (X!, Y!, MapSpot$, Xp!, Yp!, MXp!, MYp!, up$, down$,
lft$, rite$, esc$, mx!, my!)
DECLARE SUB Command (up$, down$, lft$, rite$, esc$, MapSpot$, Xp!(),
Yp!(), MXp!(), MYp!(), Hall$(), CondHall!(), Team$(), Turn!, Id!(),
Heal!(), Task!(), Task$(), Ore!(), Gold!(), Census!(), Units!(),
Grid$(), Glyne!, Warper, Fl$, Quick, NoTower) _

DECLARE SUB DrawGrid (Grid$(), Glyne!)
DECLARE SUB FirstFort (up$, down$, lft$, rite$, esc$, MapSpot$(),
Xp!(), Yp!(), MXp!(), MYp!(), Hall$(), CondHall!())
DECLARE SUB GGlyne (Glyne!)
DECLARE SUB Graphics (Grid$(), Glyne!)
DECLARE SUB InSquare (up$, down$, lft$, rite$, esc$, MapSpot$, Xp(),
Yp(), MXp(), MYp(), Hall$(), CondHall(), Team$(), Turn, Id(), Heal(),
Task(), Task$(), Ore(), Gold(), Census(), Units(), Grid$(), Glyne, X,
Y, gone, Warper)
DECLARE SUB Resetter (Team$(), Turn!, Id!(), Heal!(), Task!(), Task$(),
Ore!(), Gold!(), Census!(), Units!(), MapSpot$())
DECLARE SUB Skeleton (up$, down$, lft$, rite$, esc$, Fl$)
DECLARE SUB DrawSquare (up$, down$, lft$, rite$, esc$, MapSpot$,
Xp!(), Yp!(), MXp!(), MYp!(), Hall$(), CondHall!(), Team$(), Turn!,
Id!(), Heal!(), Task!(), Task$(), Ore!(), Gold!(), Census!(), Units!(),
Grid$(), Glyne!, mx!, my!)
CALL Skeleton(up$, down$, lft$, rite$, esc$, Fl$)
CALL GGlyne(Glyne)
' Graphics
B = 1300
S = 1000
S2 = 2000
B2 = 2700
DIM SHARED HLyne$(2, 25)
DIM SHARED Hall(B)
DIM SHARED wbox(B)
DIM SHARED lwbox(S)
DIM SHARED lybox(S)
DIM SHARED pwork(S2)
DIM SHARED prest(S2)
DIM SHARED Capt(S2)
DIM SHARED graf(B2)
DIM SHARED lil(S2)
DIM SHARED bgraf(B2)
DIM SHARED dlil(S2)
DIM SHARED mine(B)
DIM SHARED fgraf(B2)
DIM SHARED Spot(S)
DIM SHARED Grid$(100)
' Gameplay
DIM SHARED MoveToX(100)
DIM SHARED MoveToY(100)
DIM SHARED movefrx(100)
DIM SHARED movefry(100)

```

```

DIM SHARED MoveFMX(100)
DIM SHARED MoveFMY(100)
DIM SHARED Hall$(3, 3)
DIM SHARED Heal(2, 50)
DIM SHARED Task(2, 50)
DIM SHARED TaskH(3, 3)
DIM SHARED Task$(6)
DIM SHARED TaskH$(3)
DIM SHARED Id(2, 50)
DIM SHARED Census(2)
DIM SHARED Units(2, 3, 3)
DIM SHARED Gold(2)
DIM SHARED Ore(2)
DIM SHARED CondHall(3, 3)
DIM SHARED MapSpot$(3, 3, 20, 16)
DIM SHARED Team$(2)
DIM SHARED Xp(2, 50)
DIM SHARED Yp(2, 50)
DIM SHARED MXp(2, 50)
DIM SHARED MYp(2, 50)
DIM SHARED MaxHeal(3)
DIM SHARED Halls(2)
DIM SHARED Win(2)
DIM SHARED Beaten(2)
DIM SHARED Us(2)
DIM SHARED PiF(2)
DIM SHARED WiF(2)
DIM SHARED CiF(2)
DIM SHARED Energy(2, 50)
DIM SHARED MaxDam(3)
DIM SHARED UA(8)
CALL Graphics(Grid$, Glyne)
WipeOldGame:
CALL Resetter(Team$, Turn, Id(), Heal(), Task(), Task$, Ore(),
Gold(), Census(), Units(), MapSpot$())
StartTheGame:
CALL FirstFort(up$, down$, lft$, rite$, esc$, MapSpot$, Xp(), Yp(),
MXp(), MYp(), Hall$, CondHall())
DO
  Turn = 1
  IF Quick = 0 THEN CALL DispTurn(Turn, HLYne$())
  CALL Command(up$, down$, lft$, rite$, esc$, MapSpot$, Xp(), Yp(),
MXp(), MYp(), Hall$, CondHall(), Team$, Turn, Id(), Heal(), Task(),
Task$, Ore(), Gold(), Census(), Units(), Grid$, Glyne, Warper, Fl$,
Quick, NoTower)
  Turn = 2
  IF Quick = 0 THEN CALL DispTurn(Turn, HLYne$())
  CALL Command(up$, down$, lft$, rite$, esc$, MapSpot$, Xp(), Yp(),
MXp(), MYp(), Hall$, CondHall(), Team$, Turn, Id(), Heal(), Task(),
Task$, Ore(), Gold(), Census(), Units(), Grid$, Glyne, Warper, Fl$,
Quick, NoTower)
  CALL Process(up$, down$, lft$, rite$, esc$, MapSpot$, Xp(), Yp(),
MXp(), MYp(), Hall$, CondHall(), Team$, Id(), Heal(), Task(),
Task$, Ore(), Gold(), Census(), Units(), Grid$, Glyne, mx, my,
Warper, NoTower)
  CALL Test(Census(), Hall$, Gold(), Ore())
LOOP

```

```

SUB Battle (Bx, By, Starter, up$, down$, lft$, rite$, esc$, NoPref,
Heal(), Id(), MapSpot$(), Xp(), Yp(), NoTower)
  CALL Destination(MapSpot$(), Hall$(), Id(), Bx, By)
  Us(1) = Units(1, Bx, By)
  Us(2) = Units(2, Bx, By)
  IF CondHall(Bx, By) > 0 THEN
    LOCATE 25, 65
    Cond$ = STR$(CondHall(Bx, By)) + "/200"
    Cond$ = RIGHT$(Cond$, LEN(Cond$) - 1)
    IF LEN(Cond$) < 7 THEN Cond$ = "0" + Cond$
    IF LEN(Cond$) < 7 THEN Cond$ = "0" + Cond$
    IF LEN(Cond$) < 7 THEN Cond$ = "0" + Cond$
    PRINT Cond$
  END IF
  FOR z = 1 TO 2
    PiF(z) = 0
    WiF(z) = 0
    CiF(z) = 0
  NEXT
  FOR X = 1 TO 20
    FOR Y = 1 TO 16
      ThisSpot$ = MapSpot$(Bx, By, X, Y)
      IF LEFT$(ThisSpot$, 1) = "X" THEN
        Guy = VAL(RIGHT$(ThisSpot$, 2))
        IF Id(1, Guy) = 1 THEN PiF(1) = PiF(1) + 1: Energy(1, Guy) = 2
        IF Id(1, Guy) = 2 THEN WiF(1) = WiF(1) + 1: Energy(1, Guy) = 4
        IF Id(1, Guy) = 3 THEN CiF(1) = CiF(1) + 1: Energy(1, Guy) = 7
      END IF
      IF LEFT$(ThisSpot$, 1) = "O" THEN
        Guy = VAL(RIGHT$(ThisSpot$, 2))
        IF Id(2, Guy) = 1 THEN PiF(2) = PiF(2) + 1: Energy(2, Guy) = 2
        IF Id(2, Guy) = 2 THEN WiF(2) = WiF(2) + 1: Energy(2, Guy) = 4
        IF Id(2, Guy) = 3 THEN CiF(2) = CiF(2) + 1: Energy(2, Guy) = 7
      END IF
    NEXT
  NEXT
  Fight:
  IF NoPref = 1 THEN Starter = INT(RND * 2) + 1
  Turn = Starter
  Stage = 1
  DO
    IF CondHall(Bx, By) > 0 THEN
      LOCATE 25, 65
      Cond$ = STR$(CondHall(Bx, By)) + "/200"
      Cond$ = RIGHT$(Cond$, LEN(Cond$) - 1)
      IF LEN(Cond$) < 7 THEN Cond$ = "0" + Cond$
      IF LEN(Cond$) < 7 THEN Cond$ = "0" + Cond$
      IF LEN(Cond$) < 7 THEN Cond$ = "0" + Cond$
      PRINT Cond$
    END IF
    IF Hall$(Bx, By) <> "" AND CondHall(Bx, By) < 1 THEN
      LOCATE 25, 65
      PRINT "XXX/XXX"
    END IF
    Going$ = ""
  LINE (519, 0)-(620, 101), 0, BF

```

```

LINE (544, 0)-(570, 26), 1, B
IF Stage = 1 AND Hall$(Bx, By) = Team$(Turn) AND NoTower = 0 THEN
  Going$ = "H"
  LINE (519, 0)-(620, 101), 1, BF
  PUT (520, 1), fgraf(Turn * 1300 - 1299), PSET
END IF
IF Stage = 2 AND CiF(Turn) > 0 OR Stage = 3 AND CiF(Turn) > 0 THEN
  Going$ = Team$(Turn) + "3"
  PUT (545, 1), Capt(Turn * 1000 - 999)
END IF
IF Stage = 4 AND WiF(Turn) > 0 OR Stage = 5 AND WiF(Turn) > 0 THEN
  Going$ = Team$(Turn) + "2"
  PUT (545, 1), lil(Turn * 1000 - 999)
END IF
IF Stage = 6 AND PiF(Turn) > 0 OR Stage = 7 AND PiF(Turn) > 0 THEN
  Going$ = Team$(Turn) + "1"
  PUT (545, 1), pwork(Turn * 1000 - 999)
END IF
IF Stage = 1 AND NoTower = 0 THEN
  FOR Sx = 1 TO 20
    FOR Sy = 1 TO 16
      LookSpot$ = MapSpot$(Bx, By, Sx, Sy)
      IF LEFT$(LookSpot$, 1) <> "M" AND LookSpot$ <> "" AND
LEFT$(LookSpot$, 1) <> "D" AND LEFT$(LookSpot$, 1) <> Hall$(Bx, By)
THEN
        X = Sx
        Y = Sy
      END IF
    NEXT
  NEXT
END IF
IF Stage > 1 AND Going$ <> "" THEN
  FoundOne = 0
  FOR Sx = 1 TO 20
    FOR Sy = 1 TO 16
      LookSpot$ = MapSpot$(Bx, By, Sx, Sy)
      IF LookSpot$ <> "Mine" AND LookSpot$ <> "" AND RIGHT$(LookSpot$,
4) <> "Hall" AND FoundOne = 0 THEN
        IF LEFT$(LookSpot$, 1) = "X" THEN LkT = 1
        IF LEFT$(LookSpot$, 1) = "O" THEN LkT = 2
        LkGuy = VAL(RIGHT$(LookSpot$, 2))
        IF LkT = Turn AND Id(Turn, LkGuy) = VAL(RIGHT$(Going$, 1)) THEN
          X = Sx
          Y = Sy
          FoundOne = 1
        END IF
      END IF
    NEXT
  NEXT
END IF
PUT (X * 25 - 24, Y * 25 - 24), lwbox
DO
  IF Going$ = "" THEN
    PUT (X * 25 - 24, Y * 25 - 24), lwbox
    EXIT DO
  LOCATE 27, 1
  PRINT "      ";

```

```

END IF
what$ = INKEY$
IF what$ = up$ THEN
  PUT (X * 25 - 24, Y * 25 - 24), lwbox
  Y = Y - 1
  IF Y = 0 THEN Y = 16
  PUT (X * 25 - 24, Y * 25 - 24), lwbox
END IF
IF what$ = down$ THEN
  PUT (X * 25 - 24, Y * 25 - 24), lwbox
  Y = Y + 1
  IF Y = 17 THEN Y = 1
  PUT (X * 25 - 24, Y * 25 - 24), lwbox
END IF
IF what$ = lft$ THEN
  PUT (X * 25 - 24, Y * 25 - 24), lwbox
  X = X - 1
  IF X = 0 THEN X = 20
  PUT (X * 25 - 24, Y * 25 - 24), lwbox
END IF
IF what$ = rite$ THEN
  PUT (X * 25 - 24, Y * 25 - 24), lwbox
  X = X + 1
  IF X = 21 THEN X = 1
  PUT (X * 25 - 24, Y * 25 - 24), lwbox
END IF
IF what$ = esc$ THEN
  PUT (X * 25 - 24, Y * 25 - 24), lwbox
  LOCATE 27, 1
  PRINT " ";
  EXIT DO
END IF
IF what$ = " " THEN
  IF Stage = 1 THEN
    PUT (X * 25 - 24, Y * 25 - 24), lybox
    FireSpot$ = MapSpot$(Bx, By, X, Y)
    FireGuy = VAL(RIGHT$(FireSpot$, 2))
    IF LEFT$(FireSpot$, 1) = "X" THEN FireT = 1
    IF LEFT$(FireSpot$, 1) = "O" THEN FireT = 2
    FireDam = INT(RND * 10) + 1
    Heal(FireT, FireGuy) = Heal(FireT, FireGuy) - FireDam
    CALL Delay(1)
    PUT (X * 25 - 24, Y * 25 - 24), lwbox
    PUT (X * 25 - 24, Y * 25 - 24), lybox
    EXIT DO
  END IF
  GoSpot$ = MapSpot$(Bx, By, X, Y)
  GoGuy = VAL(RIGHT$(GoSpot$, 2))
  IF LEFT$(GoSpot$, 1) = "X" THEN GoT = 1
  IF LEFT$(GoSpot$, 1) = "O" THEN GoT = 2
  IF GoT = Turn AND Id(Turn, GoGuy) = VAL(RIGHT$(Going$, 1)) THEN
    LOCATE 27, 1
    Heal$ = STR$(Heal(Turn, GoGuy))
    Heal$ = RIGHT$(Heal$, LEN(Heal$) - 1)
    IF LEN(Heal$) = 1 THEN Heal$ = "0" + Heal$
    Heal$ = Heal$ + "/" + RIGHT$(STR$(MaxHeal(Id(Turn, GoGuy))), 2)
    IF Heal(Turn, GoGuy) < 1 THEN Heal$ = "XX/XX"
  END IF
END IF

```

```

PRINT Heal$
PUT (X * 25 - 24, Y * 25 - 24), lwbox
PUT (X * 25 - 24, Y * 25 - 24), lybox
DO
  IF Energy(Turn, GoGuy) < 1 THEN
    PUT (X * 25 - 24, Y * 25 - 24), lwbox
    LOCATE 27, 1
    PRINT "      ";
    EXIT DO
  END IF
  where$ = INKEY$
  moved = 0
  IF where$ = " " THEN
    PUT (X * 25 - 24, Y * 25 - 24), lwbox
    LOCATE 27, 1
    PRINT "      ";
    EXIT DO
  END IF
  IF where$ = up$ THEN
    IF Y > 1 THEN
      IF MapSpot$(Bx, By, X, Y - 1) = "" THEN
        GET (X * 25 - 24, Y * 25 - 24)-(X * 25, Y * 25), Spot
        LINE (X * 25 - 24, Y * 25 - 24)-(X * 25, Y * 25), 0, BF
        Energy(Turn, GoGuy) = Energy(Turn, GoGuy) - 1
        MapSpot$(Bx, By, X, Y - 1) = MapSpot$(Bx, By, X, Y)
        MapSpot$(Bx, By, X, Y) = ""
        Yp(Turn, GoGuy) = Yp(Turn, GoGuy) - 1
        Y = Y - 1
        PUT (X * 25 - 24, Y * 25 - 24), Spot
        moved = 1
      END IF
      FightSpot$ = MapSpot$(Bx, By, X, Y - 1)
      IF FightSpot$ <> "" AND FightSpot$ <> "Mine" AND moved = 0 THEN
        Energy(Turn, GoGuy) = Energy(Turn, GoGuy) - 5
        FGuy = VAL(RIGHT$(FightSpot$, 2))
        IF LEFT$(FightSpot$, 1) = "X" THEN FTeam = 1
        IF LEFT$(FightSpot$, 1) = "O" THEN FTeam = 2
        Damage = INT(RND * MaxDam(Id(Turn, GoGuy))) + 1
        IF FGuy = 0 THEN CondHall(Bx, By) = CondHall(Bx, By) - Damage
        IF FGuy > 0 THEN Heal(FTeam, FGuy) = Heal(FTeam, FGuy) -
Damage
      END IF
    END IF
  END IF
  IF where$ = down$ THEN
    IF Y < 16 THEN
      IF MapSpot$(Bx, By, X, Y + 1) = "" THEN
        GET (X * 25 - 24, Y * 25 - 24)-(X * 25, Y * 25), Spot
        LINE (X * 25 - 24, Y * 25 - 24)-(X * 25, Y * 25), 0, BF
        Energy(Turn, GoGuy) = Energy(Turn, GoGuy) - 1
        MapSpot$(Bx, By, X, Y + 1) = MapSpot$(Bx, By, X, Y)
        MapSpot$(Bx, By, X, Y) = ""
        Yp(Turn, GoGuy) = Yp(Turn, GoGuy) + 1
        Y = Y + 1
        PUT (X * 25 - 24, Y * 25 - 24), Spot
        moved = 1
      END IF
    END IF
  END IF

```

```

FightSpot$ = MapSpot$(Bx, By, X, Y + 1)
IF FightSpot$ <> "" AND FightSpot$ <> "Mine" AND moved = 0 THEN
  Energy(Turn, GoGuy) = Energy(Turn, GoGuy) - 5
  FGuy = VAL(RIGHT$(FightSpot$, 2))
  IF LEFT$(FightSpot$, 1) = "X" THEN FTeam = 1
  IF LEFT$(FightSpot$, 1) = "O" THEN FTeam = 2
  Damage = INT(RND * MaxDam(Id(Turn, GoGuy))) + 1
  IF FGuy = 0 THEN CondHall(Bx, By) = CondHall(Bx, By) - Damage
  IF FGuy > 0 THEN Heal(FTeam, FGuy) = Heal(FTeam, FGuy) -
Damage
  END IF
END IF
END IF
IF where$ = lft$ THEN
  IF X > 1 THEN
    IF MapSpot$(Bx, By, X - 1, Y) = "" THEN
      GET (X * 25 - 24, Y * 25 - 24)-(X * 25, Y * 25), Spot
      LINE (X * 25 - 24, Y * 25 - 24)-(X * 25, Y * 25), 0, BF
      Energy(Turn, GoGuy) = Energy(Turn, GoGuy) - 1
      MapSpot$(Bx, By, X - 1, Y) = MapSpot$(Bx, By, X, Y)
      MapSpot$(Bx, By, X, Y) = ""
      Xp(Turn, GoGuy) = Xp(Turn, GoGuy) - 1
      X = X - 1
      PUT (X * 25 - 24, Y * 25 - 24), Spot
      moved = 1
    END IF
    FightSpot$ = MapSpot$(Bx, By, X - 1, Y)
    IF FightSpot$ <> "" AND FightSpot$ <> "Mine" AND moved = 0 THEN
      Energy(Turn, GoGuy) = Energy(Turn, GoGuy) - 5
      FGuy = VAL(RIGHT$(FightSpot$, 2))
      IF LEFT$(FightSpot$, 1) = "X" THEN FTeam = 1
      IF LEFT$(FightSpot$, 1) = "O" THEN FTeam = 2
      Damage = INT(RND * MaxDam(Id(Turn, GoGuy))) + 1
      IF FGuy = 0 THEN CondHall(Bx, By) = CondHall(Bx, By) - Damage
      IF FGuy > 0 THEN Heal(FTeam, FGuy) = Heal(FTeam, FGuy) -
Damage
    END IF
  END IF
END IF
IF where$ = rite$ THEN
  IF Y < 16 THEN
    IF MapSpot$(Bx, By, X + 1, Y) = "" THEN
      GET (X * 25 - 24, Y * 25 - 24)-(X * 25, Y * 25), Spot
      LINE (X * 25 - 24, Y * 25 - 24)-(X * 25, Y * 25), 0, BF
      Energy(Turn, GoGuy) = Energy(Turn, GoGuy) - 1
      MapSpot$(Bx, By, X + 1, Y) = MapSpot$(Bx, By, X, Y)
      MapSpot$(Bx, By, X, Y) = ""
      Xp(Turn, GoGuy) = Xp(Turn, GoGuy) + 1
      X = X + 1
      PUT (X * 25 - 24, Y * 25 - 24), Spot
      moved = 1
    END IF
    FightSpot$ = MapSpot$(Bx, By, X + 1, Y)
    IF FightSpot$ <> "" AND FightSpot$ <> "Mine" AND moved = 0 THEN
      Energy(Turn, GoGuy) = Energy(Turn, GoGuy) - 5
      FGuy = VAL(RIGHT$(FightSpot$, 2))
      IF LEFT$(FightSpot$, 1) = "X" THEN FTeam = 1

```

```

        IF LEFT$(FightSpot$, 1) = "O" THEN FTeam = 2
        Damage = INT(RND * MaxDam(Id(Turn, GoGuy))) + 1
        IF FGuy = 0 THEN CondHall(Bx, By) = CondHall(Bx, By) - Damage
        IF FGuy > 0 THEN Heal(FTeam, FGuy) = Heal(FTeam, FGuy) -
Damage
        END IF
        END IF
        END IF
        LOOP
        PUT (X * 25 - 24, Y * 25 - 24), lybox
        END IF
        END IF
        LOOP
        IF Stage > 1 THEN Turn = Turn + 1
        IF Turn = 3 THEN Turn = 1
        Stage = Stage + 1
        IF Stage > 7 THEN
        FOR xx = 1 TO 20
        FOR yy = 1 TO 16
        Sp$ = MapSpot$(Bx, By, xx, yy)
        IF LEFT$(Sp$, 4) <> "Dead" AND Sp$ <> "" AND Sp$ <> "Mine" AND
VAL(RIGHT$(Sp$, 2)) > 0 THEN
        Man = VAL(RIGHT$(Sp$, 2))
        IF LEFT$(Sp$, 1) = "X" THEN Side = 1
        IF LEFT$(Sp$, 1) = "O" THEN Side = 2
        IF Heal(Side, Man) < 1 AND Id(Side, Man) <> 4 THEN
        PUT (xx * 25 - 24, yy * 25 - 24), dlil(Side * 1000 - 999), PSET
        MapSpot$(Bx, By, xx, yy) = "Dead" + Sp$
        Energy(Side, Man) = 0
        IF Id(Side, Man) = 1 THEN PiF(Side) = PiF(Side) - 1
        IF Id(Side, Man) = 2 THEN WiF(Side) = WiF(Side) - 1
        IF Id(Side, Man) = 3 THEN CiF(Side) = CiF(Side) - 1
        Us(Side) = Us(Side) - 1
        Id(Side, Man) = 4
        ELSE
        IF Id(Side, Man) = 1 THEN Energy(Side, Man) = 2
        IF Id(Side, Man) = 2 THEN Energy(Side, Man) = 4
        IF Id(Side, Man) = 3 THEN Energy(Side, Man) = 7
        END IF
        END IF
        NEXT
        NEXT
        IF CondHall(Bx, By) < 1 AND Hall$(Bx, By) <> "" THEN
        Hall$(Bx, By) = ""
        FOR KHX = 7 TO 10
        FOR KHY = 6 TO 9
        MapSpot$(Bx, By, KHX, KHY) = ""
        NEXT
        NEXT
        LINE (151, 126)-(250, 225), 0, BF
        END IF
        IF Us(1) < 1 AND Hall$(Bx, By) <> "X" THEN EXIT SUB
        IF Us(2) < 1 AND Hall$(Bx, By) <> "O" THEN EXIT SUB
        Stage = 1
        END IF
        LOOP
    END SUB

```

```

SUB Battling (up$, down$, lft$, rite$, esc$, Heal(), Id(), MapSpot$,
Xp(), Yp(), Task(), NoTower)
  FOR X = 1 TO 3
  FOR Y = 1 TO 3
    Fight = 0
    Starter = 0
    IF Units(1, X, Y) > 0 AND Units(2, X, Y) > 0 AND Hall$(X, Y) = ""
THEN
      Fight = 1
      IF Units(1, X, Y) > Units(2, X, Y) THEN Starter = 1
      IF Units(1, X, Y) < Units(2, X, Y) THEN Starter = 2
      IF Units(1, X, Y) = Units(2, X, Y) THEN NoPref = 1
      END IF
      IF Units(1, X, Y) > 0 AND Hall$(X, Y) = "O" THEN Fight = 1: Starter =
2
      IF Units(2, X, Y) > 0 AND Hall$(X, Y) = "X" THEN Fight = 1: Starter =
1
      IF Fight = 1 THEN
        CALL Battle(X, Y, Starter, up$, down$, lft$, rite$, esc$, NoPref,
Heal(), Id(), MapSpot$, Xp(), Yp(), NoTower)
        CALL Dying(X, Y, MapSpot$, Xp(), Yp(), MXp(), MYp(), Heal(), Id(),
Task(), Census(), Units())
      END IF
    NEXT
  NEXT
END SUB

```

```

SUB BuildingHalls (MapSpot$, Gold(), Ore(), Id(), Team$, Hall$,
Task(), CondHall())
  FOR X = 1 TO 3
  FOR Y = 1 TO 3
    IF Hall$(X, Y) <> Team$(1) AND Hall$(X, Y) <> Team$(2) THEN
      FOR xx = 1 TO 20
      FOR yy = 1 TO 16
        ThisSpot$ = MapSpot$(X, Y, xx, yy)
        IF ThisSpot$ <> "" AND ThisSpot$ <> "Mine" THEN
          IF LEFT$(ThisSpot$, 1) = "X" THEN Team = 1
          IF LEFT$(ThisSpot$, 2) = "O" THEN Team = 2
          ThisUnit = VAL(RIGHT$(ThisSpot$, 2))
          END IF
          IF Id(Team, ThisUnit) = 1 AND Task(Team, ThisUnit) = 6 THEN
            IF Ore(Team) >= 100 AND Gold(Team) >= 75 THEN
              Ore(Team) = Ore(Team) - 100
              Gold(Team) = Gold(Team) - 75
              IF Hall$(X, Y) <> "" THEN
                Stage = VAL(RIGHT$(Hall$(X, Y), 1))
                Stage = Stage + 1
                Hall$(X, Y) = Team$(Team) + STR$(Stage)
                IF Stage = 4 THEN
                  Hall$(X, Y) = Team$(Team)
                  Task(Team, ThisUnit) = 0
                  CondHall(X, Y) = 200
                END IF
              END IF
            IF Hall$(X, Y) = "" THEN Hall$(X, Y) = Team$(Team) + "1"
            ThisUnit = 0
          END IF
        END IF
      END IF
    END IF
  END IF

```

```

        ThisSpot$ = ""
    END IF
END IF
NEXT
NEXT
END IF
NEXT
NEXT
END SUB

SUB CatWin (Winner$)
CLS
IF Winner$ = "Cat" THEN X = 30: Y = 180: ST = 30
IF Winner$ = "Peace" THEN X = 180: Y = 30: ST = -30
FOR March = X TO Y STEP ST
    FOR Units = 10 TO 350 STEP 110
        PUT (March, Units), graf(1)
        PUT (540 - March, Units), graf(1301)
    NEXT
    CALL Delay(.5)
    FOR Units = 10 TO 350 STEP 110
        PUT (March, Units), graf(1)
        PUT (540 - March, Units), graf(1301)
    NEXT
NEXT
IF Winner$ = "Peace" THEN CALL Delay(1): EXIT SUB
FOR Units = 10 TO 350 STEP 110
    PUT (210, Units), graf(1)
    PUT (330, Units), graf(1301)
NEXT
FOR UAlive = 1 TO 8
    UA(UAlive) = 1
NEXT
alldead = 0
DO
    Killer = INT(RND * 8) + 1
    IF UA(Killer) = 1 THEN
        UA(Killer) = 0
        IF Killer < 5 THEN Kill$ = "X"
        IF Killer > 4 THEN Kill$ = "O": Killer = Killer - 4
        IF Kill$ = "X" THEN PUT (210, (Killer - 1) * 110 + 10), bgraf(1301),
PSET
        IF Kill$ = "O" THEN PUT (330, (Killer - 1) * 110 + 10), bgraf(1),
PSET
        CALL Delay(.5)
    END IF
    alldead = 1
    FOR dedder = 1 TO 8
        IF UA(dedder) = 1 THEN alldead = 0
    NEXT
LOOP WHILE alldead = 0
END SUB

SUB Cheat (Quick, NoTower)
CLS
INPUT "Cheat"; Cheet$
IF Cheet$ = "X Rich" THEN Gold(1) = 5000

```

```

IF Cheet$ = "O Rich" THEN Gold(2) = 5000
IF Cheet$ = "X Rocks" THEN Ore(1) = 5000
IF Cheet$ = "O Rocks" THEN Ore(2) = 5000
IF Cheet$ = "X Plague" THEN
  FOR Sicken = 1 TO Census(1)
    Heal(1, Sicken) = 1
  NEXT
END IF
IF Cheet$ = "O Plague" THEN
  FOR Sicken = 1 TO Census(2)
    Heal(2, Sicken) = 1
  NEXT
END IF
IF Cheet$ = "Quickening" THEN Quick = 1
IF Cheet$ = "Towerless" THEN NoTower = 1
IF Cheet$ = "Death To All" THEN
  FOR Team = 1 TO 2
    Census(Team) = 0
    FOR X = 1 TO 3
      FOR Y = 1 TO 3
        Units(Team, X, Y) = 0
        Hall$(X, Y) = ""
      NEXT
    NEXT
  NEXT
END IF
END SUB

SUB Command (up$, down$, lft$, rite$, esc$, MapSpot$, Xp(), Yp(),
MXp(), MYp(), Hall$, CondHall(), Team$, Turn, Id(), Heal(), Task(),
Task$, Ore(), Gold(), Census(), Units(), Grid$, Glyne, Warper, Fl$,
Quick, NoTower)
Command:
CLS
CALL DrawGrid(Grid$, Glyne)
PUT (1, 454), lil((Turn - 1) * 1000 + 1)
X = 1
Y = 1
FoundOne = 0
FOR Sx = 1 TO 3
  FOR Sy = 1 TO 3
    IF FoundOne = 0 AND Hall$(Sx, Sy) = Team$(Turn) THEN X = Sx: Y = Sy:
FoundOne = 1
    IF FoundOne = 0 AND Units(Turn, Sx, Sy) > 0 THEN X = Sx: Y = Sy:
FoundOne = 1
  NEXT
NEXT
PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), wbox
DO
  what$ = INKEY$
  IF what$ = Fl$ THEN
    CALL Cheat(Quick, NoTower)
    GOTO Command
  END IF
  IF what$ = up$ THEN
    PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), wbox
    Y = Y - 1

```

```

    IF Y = 0 THEN Y = 3
    PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), wbox
END IF
IF what$ = down$ THEN
    PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), wbox
    Y = Y + 1
    IF Y = 4 THEN Y = 1
    PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), wbox
END IF
IF what$ = lft$ THEN
    PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), wbox
    X = X - 1
    IF X = 0 THEN X = 3
    PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), wbox
END IF
IF what$ = rite$ THEN
    PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), wbox
    X = X + 1
    IF X = 4 THEN X = 1
    PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), wbox
END IF
IF what$ = " " THEN
    DO
        CALL InSquare(up$, down$, lft$, rite$, esc$, MapSpot$(), Xp(),
Yp(), MXp(), MYp(), Hall$(), CondHall(), Team$(), Turn, Id(), Heal(),
Task(), Task$(), Ore(), Gold(), Census(), Units(), Grid$(), Glyne, X,
Y, gone, Warper)
        IF gone = 1 THEN EXIT DO
    LOOP
    GOTO Command
END IF
IF LCASE$(what$) = "q" THEN
    END
END IF
IF what$ = esc$ THEN EXIT SUB
LOOP
END SUB

```

```

SUB Credits
CLS
CALL Delay(2)
LOCATE 15, 26
PRINT "Created by Kent Richardson"
CALL Delay(3)
LOCATE 17, 29
PRINT "Press any key to quit"
CALL Delay(1)
DO
LOOP WHILE INKEY$ = ""
END
END SUB

```

```

SUB Delay (T)
N = TIMER
DO
    IF INKEY$ = " " THEN EXIT DO
LOOP WHILE TIMER < N + T

```

```

END SUB

SUB Destination (MapSpot$, Hall$, Id(), ToX, ToY)
  CLS
  PUT (326, 76), mine
  THere = 0
  IF Hall$(ToX, ToY) = "X" THEN THere = 1
  IF Hall$(ToX, ToY) = "O" THEN THere = 2
  IF THere > 0 THEN PUT (151, 126), fgraf(THere * 1300 - 1299)
  LINE (0, 0)-(501, 401), 1, B
  FOR X = 1 TO 20
  FOR Y = 1 TO 16
    ThisSpot$ = MapSpot$(ToX, ToY, X, Y)
    IF ThisSpot$ <> "" THEN
      IF LEFT$(ThisSpot$, 1) = "X" THEN TTeam = 1
      IF LEFT$(ThisSpot$, 1) = "O" THEN TTeam = 2
      Guy = VAL(RIGHT$(ThisSpot$, 2))
      IF Id(TTeam, Guy) = 1 THEN PUT (X * 25 - 24, Y * 25 - 24),
pwork(TTeam * 1000 - 999)
      IF Id(TTeam, Guy) = 2 THEN PUT (X * 25 - 24, Y * 25 - 24), lil(TTeam
* 1000 - 999)
      IF Id(TTeam, Guy) = 3 THEN PUT (X * 25 - 24, Y * 25 - 24),
Capt(TTeam * 1000 - 999)
    END IF
  NEXT
NEXT
END SUB

SUB DispTurn (Turn, HLyne$())
  CLS
  FOR HY = 1 TO 25
  FOR HX = 1 TO 25
    LINE ((HX - 1) * 8 + 221, (HY - 1) * 8 + 141)-(HX * 8 + 220, HY * 8
+ 220), VAL(MID$(HLyne$(Turn, HY), HX, 1)), BF
  NEXT
NEXT
LINE (220, 340)-(420, 420), 0, BF
CALL Delay(3)
END SUB

SUB DrawGrid (Grid$, Glyne)
  FOR a = 1 TO Glyne
  DRAW Grid$(a)
NEXT
  FOR X = 1 TO 3
  FOR Y = 1 TO 3
    Squaredone = 0
    IF Hall$(X, Y) = "X" THEN PUT ((X - 1) * 150 + 121, (Y - 1) * 150 +
41), fgraf(1): Squaredone = 1
    IF Hall$(X, Y) = "O" THEN PUT ((X - 1) * 150 + 121, (Y - 1) * 150 +
41), fgraf(1301): Squaredone = 1
    IF Squaredone = 0 THEN
      FOR T = 1 TO 2
        IF Units(T, X, Y) > 0 THEN PUT ((X - 1) * 150 + 121, (Y - 1) * 150
+ 41), graf((T - 1) * 1300 + 1)
      NEXT
    END IF
  NEXT
END SUB

```

```

NEXT
NEXT
END SUB

SUB DrawSquare (up$, down$, lft$, rite$, esc$, MapSpot$, Xp(), Yp(),
MXp(), MYp(), Hall$, CondHall(), Team$, Turn, Id(), Heal(), Task(),
Task$, Ore(), Gold(), Census(), Units(), Grid$, Glyne, mx, my)
  T$ = Team$(Turn)
  IF T$ = Hall$(mx, my) OR Units(Turn, mx, my) > 0 THEN
    CLS
    LINE (0, 0)-(501, 401), 1, B
    PUT (326, 76), mine
    IF Hall$(mx, my) = "X" OR Hall$(mx, my) = "O" THEN PUT (151, 126),
fgraf((Turn - 1) * 1300 + 1)
    LOCATE 25, 64
    Cond$ = STR$(CondHall(mx, my)) + "/200"
    IF CondHall(mx, my) > 0 THEN PRINT Cond$
    LOCATE 27, 59
    PRINT "Gold:"; Gold(Turn);
    LOCATE 28, 59
    PRINT " Ore:"; Ore(Turn);
    FOR Sx = 1 TO 20
      FOR Sy = 1 TO 16
        ThisSpot$ = MapSpot$(mx, my, Sx, Sy)
        IF ThisSpot$ <> "" THEN
          IF ThisSpot$ <> "Mine" AND RIGHT$(ThisSpot$, 4) <> "Hall" THEN
            ThisUnit = VAL(RIGHT$(ThisSpot$, 2))
            IF Id(Turn, ThisUnit) = 1 THEN
              IF Task(Turn, ThisUnit) < 3 THEN PUT ((Sx - 1) * 25 + 1, (Sy -
1) * 25 + 1), prest((Turn - 1) * 1000 + 1)
              IF Task(Turn, ThisUnit) > 2 THEN PUT ((Sx - 1) * 25 + 1, (Sy -
1) * 25 + 1), pwork((Turn - 1) * 1000 + 1)
            END IF
            IF Id(Turn, VAL(RIGHT$(ThisSpot$, 2))) = 2 THEN PUT ((Sx - 1) *
25 + 1, (Sy - 1) * 25 + 1), lil((Turn - 1) * 1000 + 1)
            IF Id(Turn, VAL(RIGHT$(ThisSpot$, 2))) = 3 THEN PUT ((Sx - 1) *
25 + 1, (Sy - 1) * 25 + 1), Capt((Turn - 1) * 1000 + 1)
          END IF
        END IF
      NEXT
    NEXT
  END IF
END SUB

SUB Dying (Bx, By, MapSpot$, Xp(), Yp(), MXp(), MYp(), Heal(), Id(),
Task(), Census(), Units())
  FOR X = 1 TO 20
    FOR Y = 1 TO 16
      ThisSpot$ = MapSpot$(Bx, By, X, Y)
      IF LEFT$(ThisSpot$, 4) = "Dead" THEN
        ThisSpot$ = RIGHT$(ThisSpot$, 4)
        IF LEFT$(ThisSpot$, 1) = "X" THEN T = 1
        IF LEFT$(ThisSpot$, 1) = "O" THEN T = 2
        DGuy = VAL(RIGHT$(ThisSpot$, 2))
        IF DGuy >= Census(T) THEN
          Census(T) = Census(T) - 1
          Units(T, Bx, By) = Units(T, Bx, By) - 1
        END IF
      END IF
    NEXT
  NEXT
END SUB

```

```

    MapSpot$(Bx, By, X, Y) = ""
    Xp(T, DGuy) = 0
    Yp(T, DGuy) = 0
    MXp(T, DGuy) = 0
    MYp(T, DGuy) = 0
    Heal(T, DGuy) = 0
    Task(T, DGuy) = 0
    Id(T, DGuy) = 0
ELSE
    Units(T, Bx, By) = Units(T, Bx, By) - 1
    MapSpot$(MXp(T, DGuy), MYp(T, DGuy), Xp(T, DGuy), Yp(T, DGuy)) =
""
    FOR P = DGuy TO Census(T) - 1
        Xp(T, P) = Xp(T, P + 1)
        Yp(T, P) = Yp(T, P + 1)
        MXp(T, P) = MXp(T, P + 1)
        MYp(T, P) = MYp(T, P + 1)
        Heal(T, P) = Heal(T, P + 1)
        Task(T, P) = Task(T, P + 1)
        Id(T, P) = Id(T, P + 1)
    NEXT
    C = Census(T)
    Xp(T, Census(T)) = 0
    Yp(T, Census(T)) = 0
    MXp(T, Census(T)) = 0
    MYp(T, Census(T)) = 0
    Heal(T, Census(T)) = 0
    Task(T, Census(T)) = 0
    Id(T, Census(T)) = 0
    Census(T) = Census(T) - 1
    FOR MS = 1 TO Census(T)
        U = MS
        Man$ = RIGHT$(STR$(U), LEN(STR$(U)) - 1)
        IF LEN(Man$) = 1 THEN Man$ = "0" + Man$
        MapSpot$(MXp(T, U), MYp(T, U), Xp(T, U), Yp(T, U)) = Team$(T) + "
" + Man$
        IF Heal(T, U) < 1 THEN MapSpot$(MXp(T, U), MYp(T, U), Xp(T, U),
Yp(T, U)) = "Dead" + MapSpot$(MXp(T, U), MYp(T, U), Xp(T, U), Yp(T, U))
    NEXT
    END IF
    END IF
    NEXT
    NEXT
END SUB

SUB FirstFort (up$, down$, lft$, rite$, esc$, MapSpot$, Xp(), Yp(),
MXp(), MYp(), Hall$, CondHall())
    Turn = 1
    Choose:
    LOCATE 1, 24
    PRINT Team$(Turn); " Commander, choose your home base"
    X = 1
    Y = 1
    IF X = MXp(1, 1) AND Y = MYp(1, 1) THEN X = 2
    DO
        PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), graf((Turn - 1) * 1300
+ 1)

```

```

DO
  what$ = INKEY$
  IF what$ = up$ THEN
    PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), graf((Turn - 1) *
1300 + 1)
    Y = Y - 1
    IF Y = 0 THEN Y = 3
    IF Y = 2 AND X = 2 THEN Y = 1
    EXIT DO
  END IF
  IF what$ = down$ THEN
    PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), graf((Turn - 1) *
1300 + 1)
    Y = Y + 1
    IF Y = 4 THEN Y = 1
    IF Y = 2 AND X = 2 THEN Y = 3
    EXIT DO
  END IF
  IF what$ = lft$ THEN
    PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), graf((Turn - 1) *
1300 + 1)
    X = X - 1
    IF X = 0 THEN X = 3
    IF X = 2 AND Y = 2 THEN X = 1
    EXIT DO
  END IF
  IF what$ = rite$ THEN
    PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), graf((Turn - 1) *
1300 + 1)
    X = X + 1
    IF X = 4 THEN X = 1
    IF X = 2 AND Y = 2 THEN X = 3
    EXIT DO
  END IF
  OK = 1
  IF X = MXp(1, 1) AND Y = MYp(1, 1) THEN OK = 0
  IF what$ = " " AND OK = 1 THEN
    T$ = Team$(Turn)
    Hall$(X, Y) = T$
    Units(Turn, X, Y) = 3
    CondHall(X, Y) = 200
    FOR HX = 7 TO 10
      FOR HY = 6 TO 9
        MapSpot$(X, Y, HX, HY) = T$ + " Hall"
      NEXT
    NEXT
    Xp(Turn, 1) = 13
    Yp(Turn, 1) = 4
    MapSpot$(X, Y, 13, 4) = T$ + " 01"
    Xp(Turn, 2) = 13
    Yp(Turn, 2) = 8
    MapSpot$(X, Y, 13, 8) = T$ + " 02"
    Xp(Turn, 3) = 11
    Yp(Turn, 3) = 7
    MapSpot$(X, Y, 11, 7) = T$ + " 03"
    FOR m = 1 TO 3
      MXp(Turn, m) = X

```

```

        MYp(Turn, m) = Y
    NEXT
    PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), graf((Turn - 1) *
1300 + 1)
    PUT ((X - 1) * 150 + 121, (Y - 1) * 150 + 41), fgraf((Turn - 1) *
1300 + 1)
    Turn = Turn + 1
    IF Turn = 3 THEN EXIT SUB
    GOTO Choose
END IF
LOOP
LOOP
END SUB

```

```

SUB GGlyne (Glyne)
    OPEN "mscreen.dat" FOR INPUT AS #1
    DO
        Glyne = Glyne + 1
        LINE INPUT #1, a$
        LOOP WHILE NOT EOF(1)
    CLOSE #1
    Glyne = Glyne - 150
END SUB

```

```

SUB Graphics (Grid$, Glyne)
    OPEN "mscreen.dat" FOR INPUT AS #1
    FOR thing = 1 TO 6
        FOR GY = 1 TO 25
            INPUT #1, Lyne$
            IF thing < 3 THEN HLyne$(thing, GY) = Lyne$
            FOR GX = 1 TO 25
                PSET (GX + 100, GY + 100), VAL(MID$(Lyne$, GX, 1))
                LINE ((GX - 1) * 4 + 1, (GY - 1) * 4 + 1)-(GX * 4, GY * 4),
VAL(MID$(Lyne$, GX, 1)), BF
            NEXT
        NEXT
        IF thing = 3 THEN GET (101, 101)-(125, 125), dlil(1001)
        IF thing = 4 THEN GET (101, 101)-(125, 125), dlil(1)
        IF thing = 1 THEN GET (1, 1)-(100, 100), graf(1)
        IF thing = 2 THEN GET (1, 1)-(100, 100), graf(1301)
        IF thing = 3 THEN GET (1, 1)-(100, 100), bgraf(1)
        IF thing = 4 THEN GET (1, 1)-(100, 100), bgraf(1301)
        IF thing = 5 THEN GET (1, 1)-(100, 100), fgraf(1)
        IF thing = 6 THEN GET (1, 1)-(100, 100), fgraf(1301)
    NEXT
    CLS
    FOR Grid = 1 TO Glyne
        LINE INPUT #1, Grid$(Grid)
        DRAW Grid$(Grid)
    NEXT
    CLOSE #1
    OPEN "bfield.dat" FOR INPUT AS #1
    FOR thing = 1 TO 10
        FOR GY = 1 TO 25
            INPUT #1, Lyne$
            FOR GX = 1 TO 25
                PSET (GX + 100, GY + 100), VAL(MID$(Lyne$, GX, 1))
            NEXT
        NEXT
    NEXT
    CLOSE #1

```

```

        LINE ((GX - 1) * 4 + 1, (GY - 1) * 4 + 1)-(GX * 4, GY * 4),
VAL(MID$(Lyne$, GX, 1)), BF
    NEXT
NEXT
    IF thing = 1 THEN GET (101, 101)-(125, 125), pwork(1)
    IF thing = 2 THEN GET (101, 101)-(125, 125), pwork(1001)
    IF thing = 3 THEN GET (101, 101)-(125, 125), prest(1)
    IF thing = 4 THEN GET (101, 101)-(125, 125), prest(1001)
    IF thing = 5 THEN GET (101, 101)-(125, 125), lil(1)
    IF thing = 6 THEN GET (101, 101)-(125, 125), lil(1001)
    IF thing = 7 THEN GET (101, 101)-(125, 125), Capt(1)
    IF thing = 8 THEN GET (101, 101)-(125, 125), Capt(1001)
    IF thing = 9 THEN GET (1, 1)-(100, 100), Hall
    IF thing = 10 THEN GET (1, 1)-(100, 100), mine
NEXT
CLOSE #1
LINE (101, 101)-(125, 125), 0, BF
LINE (1, 1)-(100, 100), 15, BF
LINE (16, 16)-(84, 84), 0, BF
GET (1, 1)-(100, 100), wbox
LINE (1, 1)-(100, 100), 0, BF
LINE (1, 1)-(25, 25), 15, BF
LINE (3, 3)-(23, 23), 0, BF
GET (1, 1)-(25, 25), lwbox
DRAW "BM2,2 p14,0"
GET (1, 1)-(25, 25), lybox
LINE (1, 1)-(25, 25), 0, BF
END SUB

SUB InSquare (up$, down$, lft$, rite$, esc$, MapSpot$, Xp(), Yp(),
MXp(), MYp(), Hall$, CondHall(), Team$, Turn, Id(), Heal(), Task(),
Task$, Ore(), Gold(), Census(), Units(), Grid$, Glyne, mx, my,
gone, Warper)
    X = 1
    Y = 1
    gone = 0
    OK = 0
    IF Units(Turn, mx, my) > 0 OR Hall$(mx, my) = Team$(Turn) THEN OK = 1
    IF OK = 1 THEN
        gone = 1
        CALL DrawSquare(up$, down$, lft$, rite$, esc$, MapSpot$, Xp(),
Yp(), MXp(), MYp(), Hall$, CondHall(), Team$, Turn, Id(), Heal(),
Task(), Task$, Ore(), Gold(), Census(), Units(), Grid$, Glyne, mx,
my)
        PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lwbox
        ThisSpot$ = MapSpot$(mx, my, X, Y)
        IF ThisSpot$ <> "" THEN
            LOCATE 29, 1
            PRINT Task$(Task(Turn, VAL(RIGHT$(ThisSpot$, 2))));
            LOCATE 27, 1
            Heal$ = RIGHT$(STR$(Heal(Turn, VAL(RIGHT$(ThisSpot$, 2)))),
LEN(STR$(Heal(Turn, VAL(RIGHT$(ThisSpot$, 2))))) - 1)
            IF LEN(Heal$) = 1 THEN Heal$ = "0" + Heal$
            Heal$ = Heal$ + "/" + RIGHT$(STR$(MaxHeal(Id(Turn,
VAL(RIGHT$(ThisSpot$, 2))))) , 2)
            PRINT Heal$
        END IF
    END IF

```

```

LOCATE 1, 64
PRINT "Commands:";
LOCATE 3, 64
PRINT "M) Move";
LOCATE 4, 64
PRINT "W) Warp";
LOCATE 5, 64
PRINT "G) Harvest Gold"
LOCATE 6, 64
PRINT "O) Harvest Ore"
LOCATE 7, 64
IF LEN(Hall$(mx, my)) = 1 THEN PRINT "V) View Hall"
LOCATE 8, 64
IF CondHall(mx, my) < 200 AND LEN(Hall$(mx, my)) = 1 THEN PRINT "R)
Repair Hall"
IF LEN(Hall$(mx, my)) <> 1 THEN PRINT "B) Build Hall"
DO
  ThisSpot$ = MapSpot$(mx, my, X, Y)
  what$ = INKEY$
  IF what$ = esc$ THEN EXIT DO
  IF what$ = up$ THEN
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lwbox
    DO
      Y = Y - 1
      IF Y = 0 THEN Y = 16
      ThisSpot$ = MapSpot$(mx, my, X, Y)
      IF Y > 0 AND RIGHT$(ThisSpot$, 4) <> "Hall" AND ThisSpot$ <>
"Mine" THEN
        PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lwbox
        IF ThisSpot$ <> "" THEN
          LOCATE 29, 1
          PRINT Task$(Task(Turn, VAL(RIGHT$(ThisSpot$, 2))));
          LOCATE 27, 1
          Heal$ = RIGHT$(STR$(Heal(Turn, VAL(RIGHT$(ThisSpot$, 2)))))
LEN(STR$(Heal(Turn, VAL(RIGHT$(ThisSpot$, 2))))) - 1)
          IF LEN(Heal$) = 1 THEN Heal$ = "0" + Heal$
          Heal$ = Heal$ + "/" + RIGHT$(STR$(MaxHeal(Id(Turn,
VAL(RIGHT$(ThisSpot$, 2)))))
          PRINT Heal$
        END IF
        IF ThisSpot$ = "" THEN
          LOCATE 29, 1
          PRINT " ";
          LOCATE 27, 1
          PRINT " ";
        END IF
      EXIT DO
    END IF
  LOOP
END IF
IF what$ = down$ THEN
  PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lwbox
  DO
    Y = Y + 1
    IF Y = 17 THEN Y = 1
    ThisSpot$ = MapSpot$(mx, my, X, Y)

```

```

    IF Y < 17 AND RIGHT$(ThisSpot$, 4) <> "Hall" AND ThisSpot$ <>
"Mine" THEN
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lwbox
    IF ThisSpot$ <> "" THEN
        LOCATE 29, 1
        PRINT Task$(Task(Turn, VAL(RIGHT$(ThisSpot$, 2))));
        LOCATE 27, 1
        Heal$ = RIGHT$(STR$(Heal(Turn, VAL(RIGHT$(ThisSpot$, 2)))),
LEN(STR$(Heal(Turn, VAL(RIGHT$(ThisSpot$, 2))))) - 1)
        IF LEN(Heal$) = 1 THEN Heal$ = "0" + Heal$
        Heal$ = Heal$ + "/" + RIGHT$(STR$(MaxHeal(Id(Turn,
VAL(RIGHT$(ThisSpot$, 2))))) , 2)
        PRINT Heal$
    END IF
    IF ThisSpot$ = "" THEN
        LOCATE 29, 1
        PRINT " ";
        LOCATE 27, 1
        PRINT " ";
    END IF
    EXIT DO
    END IF
    LOOP
    END IF
    IF what$ = lft$ THEN
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lwbox
    DO
        X = X - 1
        IF X = 0 THEN X = 20
        ThisSpot$ = MapSpot$(mx, my, X, Y)
        IF X > 0 AND RIGHT$(ThisSpot$, 4) <> "Hall" AND ThisSpot$ <>
"Mine" THEN
            PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lwbox
            IF ThisSpot$ <> "" THEN
                LOCATE 29, 1
                PRINT Task$(Task(Turn, VAL(RIGHT$(ThisSpot$, 2))));
                LOCATE 27, 1
                Heal$ = RIGHT$(STR$(Heal(Turn, VAL(RIGHT$(ThisSpot$, 2)))),
LEN(STR$(Heal(Turn, VAL(RIGHT$(ThisSpot$, 2))))) - 1)
                IF LEN(Heal$) = 1 THEN Heal$ = "0" + Heal$
                Heal$ = Heal$ + "/" + RIGHT$(STR$(MaxHeal(Id(Turn,
VAL(RIGHT$(ThisSpot$, 2))))) , 2)
                PRINT Heal$
            END IF
            IF ThisSpot$ = "" THEN
                LOCATE 29, 1
                PRINT " ";
                LOCATE 27, 1
                PRINT " ";
            END IF
            EXIT DO
        END IF
    LOOP
    END IF
    IF what$ = rite$ THEN
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lwbox
    DO

```

```

X = X + 1
IF X = 21 THEN X = 1
ThisSpot$ = MapSpot$(mx, my, X, Y)
IF X < 21 AND RIGHT$(ThisSpot$, 4) <> "Hall" AND ThisSpot$ <>
"Mine" THEN
  PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lwbox
  IF ThisSpot$ <> "" THEN
    LOCATE 29, 1
    PRINT Task$(Task(Turn, VAL(RIGHT$(ThisSpot$, 2))));
    LOCATE 27, 1
    Heal$ = RIGHT$(STR$(Heal(Turn, VAL(RIGHT$(ThisSpot$, 2))),
LEN(STR$(Heal(Turn, VAL(RIGHT$(ThisSpot$, 2)))) - 1)
    IF LEN(Heal$) = 1 THEN Heal$ = "0" + Heal$
    Heal$ = Heal$ + "/" + RIGHT$(STR$(MaxHeal(Id(Turn,
VAL(RIGHT$(ThisSpot$, 2))))), 2)
    PRINT Heal$
  END IF
  IF ThisSpot$ = "" THEN
    LOCATE 29, 1
    PRINT " ";
    LOCATE 27, 1
    PRINT " ";
  END IF
  EXIT DO
END IF
LOOP
END IF
what$ = LCASE$(what$)
IF what$ = "b" AND Task(Turn, VAL(RIGHT$(ThisSpot$, 2))) <> 1 AND
Id(Turn, VAL(RIGHT$(ThisSpot$, 2))) = 1 AND LEN(Hall$(mx, my)) <> 1
THEN
  Task(Turn, VAL(RIGHT$(ThisSpot$, 2))) = 6
  LOCATE 29, 1
  PRINT Task$(Task(Turn, VAL(RIGHT$(ThisSpot$, 2))));
END IF
IF what$ = "r" AND Task(Turn, VAL(RIGHT$(ThisSpot$, 2))) <> 1 AND
Id(Turn, VAL(RIGHT$(ThisSpot$, 2))) = 1 AND CondHall(mx, my) < 200 AND
LEN(Hall$(mx, my)) = 1 THEN
  Task(Turn, VAL(RIGHT$(ThisSpot$, 2))) = 3
  LOCATE 29, 1
  PRINT Task$(Task(Turn, VAL(RIGHT$(ThisSpot$, 2))));
END IF
IF what$ = "g" AND LEN(Hall$(mx, my)) = 1 AND Task(Turn,
VAL(RIGHT$(ThisSpot$, 2))) <> 1 AND Id(Turn, VAL(RIGHT$(ThisSpot$, 2)))
= 1 THEN
  Task(Turn, VAL(RIGHT$(ThisSpot$, 2))) = 4
  LOCATE 29, 1
  PRINT Task$(Task(Turn, VAL(RIGHT$(ThisSpot$, 2))));
  PUT (X * 25 - 24, Y * 25 - 24), lwbox, PSET
  PUT (X * 25 - 24, Y * 25 - 24), pwork(Turn * 1000 - 999)
END IF
IF what$ = "o" AND LEN(Hall$(mx, my)) = 1 AND Task(Turn,
VAL(RIGHT$(ThisSpot$, 2))) <> 1 AND Id(Turn, VAL(RIGHT$(ThisSpot$, 2)))
= 1 THEN
  Task(Turn, VAL(RIGHT$(ThisSpot$, 2))) = 5
  LOCATE 29, 1
  PRINT Task$(Task(Turn, VAL(RIGHT$(ThisSpot$, 2))));

```

```

    PUT (X * 25 - 24, Y * 25 - 24), lwbox, PSET
    PUT (X * 25 - 24, Y * 25 - 24), pwork(Turn * 1000 - 999)
END IF
IF what$ = "m" AND Task(Turn, VAL(RIGHT$(ThisSpot$, 2))) <> 1 AND
ThisSpot$ <> "" THEN
    CALL Move(X, Y, MapSpot$(), Xp, Yp, MXp, MYp, up$, down$, lft$,
rite$, esc$, mx, my)
END IF
IF what$ = "s" AND Task(Turn, VAL(RIGHT$(ThisSpot$, 2))) <> 1 AND
Task(Turn, VAL(RIGHT$(ThisSpot$, 2))) <> 2 AND ThisSpot$ <> "" THEN
    Task(Turn, VAL(RIGHT$(ThisSpot$, 2))) = 0
    LOCATE 29, 1
    PRINT Task$(Task(Turn, VAL(RIGHT$(ThisSpot$, 2))));
    IF Id(Turn, VAL(RIGHT$(ThisSpot$, 2))) = 1 THEN
        PUT (X * 25 - 24, Y * 25 - 24), lwbox, PSET
        PUT (X * 25 - 24, Y * 25 - 24), prest(Turn * 1000 - 999)
    END IF
END IF
IF what$ = "s" AND Task(Turn, VAL(RIGHT$(ThisSpot$, 2))) = 2 THEN
    Task(Turn, VAL(RIGHT$(ThisSpot$, 2))) = 0
    LOCATE 29, 1
    PRINT Task$(Task(Turn, VAL(RIGHT$(ThisSpot$, 2))));
    TaskH(mx, my) = 0
END IF
IF what$ = "w" AND Task(Turn, VAL(RIGHT$(ThisSpot$, 2))) <> 1 AND
ThisSpot$ <> "" THEN
    CALL Warp(X, Y, MapSpot$(), Xp, Yp, MXp, MYp, up$, down$, lft$,
rite$, esc$, mx, my, Grid$(), Glyne, Turn, Warper)
    gone = 0
    EXIT SUB
END IF
IF what$ = "v" AND LEN(Hall$(mx, my)) = 1 THEN
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lwbox
    LOCATE 3, 64
    PRINT "P) Train Peasant";
    LOCATE 4, 64
    PRINT "W) Train Warrior";
    LOCATE 5, 64
    PRINT "U) Upgrade Unit";
    LOCATE 6, 64
    PRINT "S) Stop          ";
    LOCATE 7, 64
    PRINT "                  ";
    LOCATE 8, 64
    PRINT "D) Done          ";
    DO
        HWhat$ = LCASE$(INKEY$)
        IF HWhat$ = "p" THEN TaskH(mx, my) = 1
        IF HWhat$ = "w" THEN TaskH(mx, my) = 2
        IF HWhat$ = "u" THEN
            CALL Upgrade(MapSpot$(), Id(), Task(), mx, my, TaskH(), Turn)
        END IF
        IF HWhat$ = "d" THEN EXIT DO
        IF HWhat$ = "s" THEN
            IF TaskH(mx, my) = 3 THEN
                FOR zy = 1 TO 16
                    FOR zx = 1 TO 20

```

```

        Dude = VAL(RIGHT$(MapSpot$(mx, my, zx, zy), 2))
        IF Task(Turn, Dude) = 2 THEN Task(Turn, Dude) = 0
    NEXT
    NEXT
    END IF
    TaskH(mx, my) = 0
    END IF
    LOCATE 29, 1
    PRINT TaskH$(TaskH(mx, my));
    LOOP
    gone = 0
    EXIT SUB
    END IF
    LOOP
    END IF
    IF OK = 0 THEN gone = 1
END SUB

SUB Mining
    FOR X = 1 TO 3
    FOR Y = 1 TO 3
    FOR z = 1 TO 2
        IF Units(z, X, Y) > 0 THEN
            FOR xx = 1 TO 20
            FOR yy = 1 TO 16
                ThisSpot$ = MapSpot$(X, Y, xx, yy)
                ThisUnit = VAL(RIGHT$(ThisSpot$, 2))
                IF Id(z, ThisUnit) = 1 AND Task(z, ThisUnit) = 4 THEN Gold(z) =
Gold(z) + 10
                IF Id(z, ThisUnit) = 1 AND Task(z, ThisUnit) = 5 THEN Ore(z) =
Ore(z) + 5
            NEXT
            NEXT
        END IF
    NEXT
    NEXT
    NEXT
    NEXT
END SUB

SUB Move (X, Y, MapSpot$(), Xp, Yp, MXp, MYp, up$, down$, lft$, rite$,
esc$, mx, my)
    fx = X
    fy = Y
    DO
        X = X + 1
        IF X = 21 THEN Y = Y + 1: X = 1
        IF Y = 17 THEN Y = 1
        ThisSpot$ = MapSpot$(mx, my, X, Y)
        IF ThisSpot$ = "" THEN EXIT DO
    LOOP
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lybox
    DO
        what$ = INKEY$
        IF what$ = up$ THEN
            PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lybox
            DO
                Y = Y - 1

```

```

    IF Y = 0 THEN Y = 16
    IF MapSpot$(mx, my, X, Y) = "" THEN EXIT DO
    LOOP
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lybox
END IF
IF what$ = down$ THEN
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lybox
    DO
        Y = Y + 1
        IF Y = 17 THEN Y = 1
        IF MapSpot$(mx, my, X, Y) = "" THEN EXIT DO
    LOOP
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lybox
END IF
IF what$ = lft$ THEN
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lybox
    DO
        X = X - 1
        IF X = 0 THEN X = 20
        IF MapSpot$(mx, my, X, Y) = "" THEN EXIT DO
    LOOP
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lybox
END IF
IF what$ = rite$ THEN
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lybox
    DO
        X = X + 1
        IF X = 21 THEN X = 1
        IF MapSpot$(mx, my, X, Y) = "" THEN EXIT DO
    LOOP
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lybox
END IF
IF what$ = esc$ THEN
    PUT (X * 25 - 24, Y * 25 - 24), lybox
    PUT (X * 25 - 24, Y * 25 - 24), lwbox
    PUT (fx * 25 - 24, fy * 25 - 24), lwbox
    EXIT SUB
END IF
IF what$ = " " THEN
    MapSpot$(mx, my, X, Y) = MapSpot$(mx, my, fx, fy)
    MapSpot$(mx, my, fx, fy) = ""
    Guy = VAL(RIGHT$(MapSpot$(mx, my, X, Y), 2))
    Xp(Turn, Guy) = X
    Yp(Turn, Guy) = Y
    PUT ((fx - 1) * 25 + 1, (fy - 1) * 25 + 1), lwbox
    GET ((fx - 1) * 25 + 1, (fy - 1) * 25 + 1)-(fx * 25, fy * 25), Spot
    PUT ((fx - 1) * 25 + 1, (fy - 1) * 25 + 1), lwbox, PSET
    PUT ((fx - 1) * 25 + 1, (fy - 1) * 25 + 1), lwbox
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lybox
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), Spot
    PUT ((X - 1) * 25 + 1, (Y - 1) * 25 + 1), lwbox
    EXIT SUB
END IF
LOOP
END SUB

SUB OWin

```

```

CLS
PUT (300, 220), lil(1001)
FOR chase = 1 TO 201 STEP 25
  PUT (chase, 175), graf(1)
  CALL Delay(.5)
  PUT (chase, 175), graf(1)
NEXT
PUT (225, 175), graf(1)
PUT (300, 220), lil(1001)
PUT (315, 220), lil(1001)
CALL Delay(.5)
PUT (225, 175), graf(1)
PUT (315, 220), lil(1001)
PUT (250, 175), graf(1)
PUT (330, 220), lil(1001)
CALL Delay(.5)
PUT (250, 175), graf(1)
PUT (330, 220), lil(1001)
PUT (275, 175), graf(1)
PUT (400, 220), lil(1001)
CALL Delay(.5)
PUT (275, 175), graf(1)
PUT (400, 220), lil(1001)
PUT (285, 175), graf(1)
PUT (475, 220), lil(1001)
CALL Delay(.5)
PUT (475, 220), lil(1001)
PUT (550, 220), lil(1001)
CALL Delay(.5)
PUT (550, 220), lil(1001)
CALL Delay(1)
PUT (285, 175), graf(1)
PUT (275, 175), graf(1)
CALL Delay(.5)
PUT (275, 175), graf(1)
PUT (265, 175), graf(1)
CALL Delay(.5)
PUT (265, 175), graf(1)
PUT (255, 175), graf(1)
FOR ohboy = 174 TO 74 STEP -25
  PUT (255 - ohboy, 175 - ohboy), graf(1301)
  PUT (255 + ohboy, 175 - ohboy), graf(1301)
  PUT (255 - ohboy, 175 + ohboy), graf(1301)
  PUT (255 + ohboy, 175 + ohboy), graf(1301)
  CALL Delay(.5)
  PUT (255 - ohboy, 175 - ohboy), graf(1301)
  PUT (255 + ohboy, 175 - ohboy), graf(1301)
  PUT (255 - ohboy, 175 + ohboy), graf(1301)
  PUT (255 + ohboy, 175 + ohboy), graf(1301)
NEXT
CLS
PUT (255, 175), bgraf(1301)
CALL Delay(.5)
FOR happyohs = 1 TO 6
  PUT ((happyohs - 1) * 100 + 1, 50), lil(1001)
  PUT ((happyohs - 1) * 75 + 1, 150), lil(1001)
  PUT ((happyohs - 1) * 90 + 1, 250), lil(1001)

```

```

    PUT ((happyohs - 1) * 105 + 1, 350), lil(1001)
    CALL Delay(.5)
    PUT ((happyohs - 1) * 100 + 1, 50), lil(1001)
    PUT ((happyohs - 1) * 75 + 1, 150), lil(1001)
    PUT ((happyohs - 1) * 90 + 1, 250), lil(1001)
    PUT ((happyohs - 1) * 105 + 1, 350), lil(1001)
NEXT
'PUT (5 * 75 + 1, 150), lil(1001)
'CALL Delay(.5)
'PUT (5 * 75 + 1, 150), lil(1001)
'PUT (4 * 75 + 40, 150), lil(1001)
'CALL Delay(.5)
'PUT (4 * 75 + 40, 150), lil(1001)
'PUT (4 * 75, 150), lil(1001)
'CALL Delay(.5)
'PUT (4 * 75, 150), lil(1001)
'PUT (4 * 75, 175), lil(1001)
'CALL Delay(.5)
'FOR pee = 1 TO 5
' CIRCLE (315, 205), pee, 14
'NEXT
'CALL Delay(.5)
'PUT (4 * 75, 175), lil(1001)
'PUT (5 * 75, 150), lil(1001)
'CALL Delay(.5)
'PUT (5 * 75, 150), lil(1001)
'PUT (6 * 75, 150), lil(1001)
'CALL Delay(.5)
'PUT (6 * 75, 150), lil(1001)
'PUT (7 * 75, 150), lil(1001)
'CALL Delay(.5)
END SUB

SUB Process (up$, down$, lft$, rite$, esc$, MapSpot$, Xp(), Yp(),
MXp(), MYp(), Hall$, CondHall(), Team$, Id(), Heal(), Task(),
Task$, Ore(), Gold(), Census(), Units(), Grid$, Glyne, mx, my,
Warper, NoTower)
    CALL Training(up$, down$, lft$, rite$)
    CALL Warping(up$, down$, lft$, rite$, esc$, MapSpot$, Xp(), Yp(),
MXp(), MYp(), Hall$, CondHall(), Team$, Team, Id(), Heal(), Task(),
Task$, Ore(), Gold(), Census(), Units(), Grid$, Glyne, MoveToX(),
MoveToY(), Warper)
    CALL Battling(up$, down$, lft$, rite$, esc$, Heal(), Id(), MapSpot$,
Xp(), Yp(), Task(), NoTower)
    CALL Repairing
    CALL BuildingHalls(MapSpot$, Gold(), Ore(), Id(), Team$, Hall$,
Task(), CondHall())
    CALL Mining
END SUB

SUB Repairing
FOR X = 1 TO 3
FOR Y = 1 TO 3
    IF 0 < CondHall(X, Y) AND CondHall(X, Y) < 200 THEN
        FOR z = 1 TO 2
            IF Team$(z) = Hall$(X, Y) THEN ZZ = z
        NEXT
    NEXT
NEXT

```

```

FOR xx = 1 TO 20
FOR yy = 1 TO 16
  ThisSpot$ = MapSpot$(X, Y, xx, yy)
  ThisUnit = VAL(RIGHT$(ThisSpot$, 2))
  IF Id(ZZ, ThisUnit) = 1 AND Task(ZZ, ThisUnit) = 3 AND CondHall(X,
Y) < 200 THEN
    IF Gold(ZZ) >= 10 AND Ore(ZZ) >= 25 THEN
      Gold(ZZ) = Gold(ZZ) - 10
      Ore(ZZ) = Ore(ZZ) - 25
      CondHall(X, Y) = CondHall(X, Y) + 5
      IF CondHall(X, Y) > 200 THEN CondHall(X, Y) = 200
    END IF
  END IF
  IF CondHall(X, Y) = 200 THEN
    FOR XXX = 1 TO 20
    FOR YYY = 1 TO 16
      TisUnt = VAL(RIGHT$(MapSpot$(X, Y, XXX, YYY), 2))
      IF Task(ZZ, TisUnt) = 3 THEN Task(ZZ, TisUnt) = 0
    NEXT
  NEXT
  END IF
NEXT
NEXT
END IF
NEXT
NEXT
END SUB

```

```

SUB Resetter (Team$(), Turn, Id(), Heal(), Task(), Task$(), Ore(),
Gold(), Census(), Units(), MapSpot$())
Team$(1) = "X"
Team$(2) = "O"
Turn = 1
FOR Tem = 1 TO 2
  Id(Tem, 1) = 3
  Heal(Tem, 1) = 50
  Task(Tem, 1) = 0
  Id(Tem, 2) = 1
  Heal(Tem, 2) = 20
  Task(Tem, 2) = 0
  Id(Tem, 3) = 1
  Heal(Tem, 3) = 20
  Task(Tem, 3) = 0
  Ore(Tem) = 50
  Gold(Tem) = 50
  Census(Tem) = 3
NEXT
FOR Bx = 1 TO 3
  FOR By = 1 TO 3
    FOR Tem = 1 TO 2
      Units(Tem, Bx, By) = 0
    NEXT
  FOR lx = 1 TO 20
    FOR ly = 1 TO 16
      MapSpot$(Bx, By, lx, ly) = ""
    NEXT
  NEXT
NEXT

```

```

FOR mx = 14 TO 17
  FOR my = 5 TO 7
    MapSpot$(Bx, By, mx, my) = "Mine"
  NEXT
NEXT
NEXT
NEXT
MaxHeal(1) = 20
MaxHeal(2) = 30
MaxHeal(3) = 50
MaxDam(1) = 4
MaxDam(2) = 7
MaxDam(3) = 10
Task$(0) = "Stopped"
Task$(1) = "Warping"
Task$(2) = "Upgrading to Captain"
Task$(3) = "Repairing the Hall"
Task$(4) = "Harvesting Gold"
Task$(5) = "Harvesting Ore"
Task$(6) = "Building a Hall"
TaskH$(0) = "Stopped"
TaskH$(1) = "Training a peasant"
TaskH$(2) = "Training a warrior"
TaskH$(3) = "Upgrading a warrior"
END SUB

SUB Skeleton (up$, down$, lft$, rite$, esc$, F1$)
SCREEN 12
RANDOMIZE TIMER
OPEN "Dirction.dat" FOR RANDOM AS #1
  GET #1, 1, up$
  GET #1, 2, down$
  GET #1, 3, lft$
  GET #1, 4, rite$
  GET #1, 5, esc$
  GET #1, 6, F1$
CLOSE #1
END SUB

SUB Test (Census(), Hall$(), Gold(), Ore())
Winner$ = ""
Win(1) = 0
Win(2) = 0
FOR Team = 1 TO 2
  T$ = Team$(Team)
  IF Hall$(1, 1) = T$ AND Hall$(1, 2) = T$ AND Hall$(1, 3) = T$ THEN
Win(Team) = 1
  IF Hall$(2, 1) = T$ AND Hall$(2, 2) = T$ AND Hall$(2, 3) = T$ THEN
Win(Team) = 1
  IF Hall$(3, 1) = T$ AND Hall$(3, 2) = T$ AND Hall$(3, 3) = T$ THEN
Win(Team) = 1
  IF Hall$(1, 1) = T$ AND Hall$(2, 1) = T$ AND Hall$(3, 1) = T$ THEN
Win(Team) = 1
  IF Hall$(1, 2) = T$ AND Hall$(2, 2) = T$ AND Hall$(3, 2) = T$ THEN
Win(Team) = 1
  IF Hall$(1, 3) = T$ AND Hall$(2, 3) = T$ AND Hall$(3, 3) = T$ THEN
Win(Team) = 1

```

```

    IF Hall$(1, 1) = T$ AND Hall$(2, 2) = T$ AND Hall$(3, 3) = T$ THEN
Win(Team) = 1
    IF Hall$(1, 3) = T$ AND Hall$(2, 2) = T$ AND Hall$(3, 1) = T$ THEN
Win(Team) = 1
    NEXT
    IF Win(1) = 1 AND Win(2) = 1 THEN Winner$ = "Peace"
    IF Win(1) = 1 AND Win(2) = 0 THEN Winner$ = "X"
    IF Win(1) = 0 AND Win(2) = 1 THEN Winner$ = "O"
    Halls(1) = 0
    Halls(2) = 0
    FOR X = 1 TO 3
    FOR Y = 1 TO 3
        IF Hall$(X, Y) = "X" THEN Halls(1) = Halls(1) + 1
        IF Hall$(X, Y) = "O" THEN Halls(2) = Halls(2) + 1
    NEXT
    NEXT
    FOR Team = 1 TO 2
        Beaten(Team) = 0
        IF Census(Team) = 0 AND Halls(Team) = 0 THEN Beaten(Team) = 1
        IF Census(Team) = 0 AND Gold(Team) < 20 THEN Beaten(Team) = 1
    NEXT
    IF Beaten(1) = 1 AND Beaten(2) = 1 THEN Winner$ = "Cat"
    IF Beaten(1) = 1 AND Beaten(2) = 0 THEN Winner$ = "O"
    IF Beaten(1) = 0 AND Beaten(2) = 1 THEN Winner$ = "X"
    IF Winner$ = "X" THEN CALL Xwin
    IF Winner$ = "O" THEN CALL Owin
    IF Winner$ = "Cat" OR Winner$ = "Peace" THEN CALL CatWin(Winner$)
    IF Winner$ <> "" THEN CALL Credits
END SUB

SUB Training (up$, down$, lft$, rite$)
FOR X = 1 TO 3
FOR Y = 1 TO 3
    Team = 0
    IF Hall$(X, Y) = "X" THEN Team = 1
    IF Hall$(X, Y) = "O" THEN Team = 2
    IF TaskH(X, Y) > 0 AND TaskH(X, Y) < 3 AND Census(Team) <= 50 THEN
        NeededG = 0
        NeededO = 0
        IF TaskH(X, Y) = 1 THEN
            NeededG = 20
            NeededO = 0
        END IF
        IF TaskH(X, Y) = 2 THEN
            PUT (615, 0), lil(Team * 1000 - 999)
            NeededG = 40
            NeededO = 25
        END IF
        IF NeededG > Gold(Team) OR NeededO > Ore(Team) THEN GOTO Progress
        Census(Team) = Census(Team) + 1
        Units(Team, X, Y) = Units(Team, X, Y) + 1
        CALL DrawSquare(up$, down$, lft$, rite$, esc$, MapSpot$, Xp(),
Yp(), MXp(), MYp(), Hall$, CondHall(), Team$, Team, Id(), Heal(),
Task(), Task$, Ore(), Gold(), Census(), Units(), Grid$, Glyne, X,
Y)
        IF TaskH(X, Y) = 1 THEN PUT (615, 0), prest(Team * 1000 - 999)
        IF TaskH(X, Y) = 2 THEN PUT (615, 0), lil(Team * 1000 - 999)

```

```

xx = 0
yy = 1
DO
  xx = xx + 1
  IF xx = 21 THEN xx = 1: yy = yy + 1
  IF MapSpot$(X, Y, xx, yy) = "" THEN EXIT DO
LOOP
PUT (xx * 25 - 24, yy * 25 - 24), lybox
PUT (615, 0), lwbox
DO
  what$ = INKEY$
  IF what$ = up$ THEN
    PUT (xx * 25 - 24, yy * 25 - 24), lybox
    DO
      YY = yy - 1
      IF yy = 0 THEN yy = 16
      IF MapSpot$(X, Y, xx, yy) = "" THEN EXIT DO
    LOOP
    PUT (xx * 25 - 24, yy * 25 - 24), lybox
  END IF
  IF what$ = down$ THEN
    PUT (xx * 25 - 24, yy * 25 - 24), lybox
    DO
      yy = yy + 1
      IF yy = 17 THEN yy = 1
      IF MapSpot$(X, Y, xx, yy) = "" THEN EXIT DO
    LOOP
    PUT (xx * 25 - 24, yy * 25 - 24), lybox
  END IF
  IF what$ = lft$ THEN
    PUT (xx * 25 - 24, yy * 25 - 24), lybox
    DO
      xx = xx - 1
      IF xx = 0 THEN xx = 20
      IF MapSpot$(X, Y, xx, yy) = "" THEN EXIT DO
    LOOP
    PUT (xx * 25 - 24, yy * 25 - 24), lybox
  END IF
  IF what$ = rite$ THEN
    PUT (xx * 25 - 24, yy * 25 - 24), lybox
    DO
      xx = xx + 1
      IF xx = 21 THEN xx = 1
      IF MapSpot$(X, Y, xx, yy) = "" THEN EXIT DO
    LOOP
    PUT (xx * 25 - 24, yy * 25 - 24), lybox
  END IF
  IF what$ = " " THEN
    Xp(Team, Census(Team)) = xx
    Yp(Team, Census(Team)) = yy
    MXp(Team, Census(Team)) = X
    MYp(Team, Census(Team)) = Y
    Gold(Team) = Gold(Team) - NeededG
    Ore(Team) = Ore(Team) - NeededO
    Id(Team, Census(Team)) = TaskH(X, Y)
    Heal(Team, Census(Team)) = MaxHeal(TaskH(X, Y))
    Guy$ = STR$(Census(Team))

```

```

        IF LEN(Guy$) = 2 THEN Guy$ = "0" + Guy$
        MapSpot$(X, Y, xx, yy) = Team$(Team) + Guy$
        TaskH(X, Y) = 0
        EXIT DO
    END IF
LOOP
END IF
IF TaskH(X, Y) = 3 AND Gold(Team) >= 20 AND Ore(Team) >= 20 THEN
    FOR xx = 1 TO 20
    FOR yy = 1 TO 16
        ThisSpot$ = MapSpot$(X, Y, xx, yy)
        Guy = VAL(RIGHT$(ThisSpot$, 2))
        IF Task(Team, Guy) = 2 THEN
            Task(Team, Guy) = 0
            TaskH(X, Y) = 0
            Gold(Team) = Gold(Team) - 20
            Ore(Team) = Ore(Team) - 20
            Heal(Team, Guy) = Heal(Team, Guy) + 20
            Id(Team, Guy) = 3
        END IF
    NEXT
NEXT
END IF
Progress:
NEXT
NEXT
END SUB

SUB Upgrade (MapSpot$( ), Id( ), Task( ), mx, my, TaskH( ), Turn)
    FOR Y = 1 TO 16
    FOR X = 1 TO 20
        ThisSpot$ = MapSpot$(mx, my, X, Y)
        ThisUnit = VAL(RIGHT$(ThisSpot$, 2))
        IF Id(Turn, ThisUnit) = 2 AND Task(Turn, ThisUnit) = 0 THEN
            PUT (X * 25 - 24, Y * 25 - 24), lybox
            DO
                what$ = LCASE$(INKEY$)
                IF what$ = " " THEN
                    Task(Turn, ThisUnit) = 2
                    TaskH(mx, my) = 3
                    PUT (X * 25 - 24, Y * 25 - 24), lwbox
                    EXIT SUB
                END IF
                IF what$ <> "" THEN
                    PUT (X * 25 - 24, Y * 25 - 24), lybox
                    EXIT DO
                END IF
            LOOP
        END IF
    NEXT
NEXT
END SUB

SUB Warp (X, Y, MapSpot$( ), Xp, Yp, MXp, MYp, up$, down$, lft$, rite$,
esc$, mx, my, Grid$( ), Glyne, Turn, Warper)
    CLS
    CALL DrawGrid(Grid$( ), Glyne)

```

```

xx = 1
yy = 1
PUT ((xx - 1) * 150 + 121, (yy - 1) * 150 + 41), wbox
DO
  what$ = INKEY$
  IF what$ <> "" THEN PUT ((xx - 1) * 150 + 121, (yy - 1) * 150 + 41),
wbox
  IF what$ = up$ THEN yy = yy - 1
  IF what$ = down$ THEN yy = yy + 1
  IF what$ = lft$ THEN xx = xx - 1
  IF what$ = rite$ THEN xx = xx + 1
  IF yy = 0 THEN yy = 3
  IF yy = 4 THEN yy = 1
  IF xx = 0 THEN xx = 3
  IF xx = 4 THEN xx = 1
  IF what$ <> "" THEN PUT ((xx - 1) * 150 + 121, (yy - 1) * 150 + 41),
wbox
  IF what$ = " " THEN
    IF xx = mx AND yy = my THEN EXIT SUB
    Warper = Warper + 1
    Guy = VAL(RIGHT$(MapSpot$(mx, my, X, Y), 2))
    Task(Turn, Guy) = 1
    MoveToX(Warper) = xx
    MoveToY(Warper) = yy
    MoveFMX(Warper) = mx
    MoveFMY(Warper) = my
    movefrx(Warper) = X
    movefry(Warper) = Y
    EXIT DO
  END IF
LOOP
END SUB

SUB Warping (up$, down$, lft$, rite$, esc$, MapSpot$, Xp(), Yp(),
MXp(), MYp(), Hall$, CondHall(), Team$, Team, Id(), Heal(), Task(),
Task$, Ore(), Gold(), Census(), Units(), Grid$, Glyne, MoveToX(),
MoveToY(), Warper)
  FOR W = 1 TO Warper
    From$ = MapSpot$(MoveFMX(W), MoveFMY(W), movefrx(W), movefry(W))
    GuyWarping = VAL(RIGHT$(From$, 2))
    IF LEFT$(From$, 1) = "X" THEN Team = 1
    IF LEFT$(From$, 1) = "O" THEN Team = 2
    CALL Destination(MapSpot$, Hall$, Id(), MoveToX(W), MoveToY(W))
    PUT (615, 0), lwbox
    IF Id(Team, GuyWarping) = 1 THEN PUT (615, 0), prest(Team * 1000 -
999)
    IF Id(Team, GuyWarping) = 2 THEN PUT (615, 0), lil(Team * 1000 - 999)
    IF Id(Team, GuyWarping) = 3 THEN PUT (615, 0), Capt(Team * 1000 -
999)
    X = 0
    Y = 1
    DO
      X = X + 1
      IF X = 21 THEN X = 1: Y = Y + 1
      IF MapSpot$(MoveToX(W), MoveToY(W), X, Y) = "" THEN EXIT DO
    LOOP
    PUT (X * 25 - 24, Y * 25 - 24), lybox

```

```

DO
  what$ = INKEY$
  IF what$ = up$ THEN
    PUT (X * 25 - 24, Y * 25 - 24), lybox
    DO
      Y = Y - 1
      IF Y = 0 THEN Y = 16
      IF MapSpot$(MoveToX(W), MoveToY(W), X, Y) = "" THEN EXIT DO
    LOOP
    PUT (X * 25 - 24, Y * 25 - 24), lybox
  END IF
  IF what$ = down$ THEN
    PUT (X * 25 - 24, Y * 25 - 24), lybox
    DO
      Y = Y + 1
      IF Y = 17 THEN Y = 1
      IF MapSpot$(MoveToX(W), MoveToY(W), X, Y) = "" THEN EXIT DO
    LOOP
    PUT (X * 25 - 24, Y * 25 - 24), lybox
  END IF
  IF what$ = lft$ THEN
    PUT (X * 25 - 24, Y * 25 - 24), lybox
    DO
      X = X - 1
      IF X = 0 THEN X = 20
      IF MapSpot$(MoveToX(W), MoveToY(W), X, Y) = "" THEN EXIT DO
    LOOP
    PUT (X * 25 - 24, Y * 25 - 24), lybox
  END IF
  IF what$ = rite$ THEN
    PUT (X * 25 - 24, Y * 25 - 24), lybox
    DO
      X = X + 1
      IF X = 21 THEN X = 1
      IF MapSpot$(MoveToX(W), MoveToY(W), X, Y) = "" THEN EXIT DO
    LOOP
    PUT (X * 25 - 24, Y * 25 - 24), lybox
  END IF
  IF what$ = " " THEN
    Task(Team, GuyWarping) = 0
    Xp(Team, GuyWarping) = X
    Yp(Team, GuyWarping) = Y
    MXp(Team, GuyWarping) = MoveToX(W)
    MYp(Team, GuyWarping) = MoveToY(W)
    MapSpot$(MoveToX(W), MoveToY(W), X, Y) = From$
    MapSpot$(MoveFMX(W), MoveFMY(W), movefrx(W), movefry(W)) = ""
    Units(Team, MoveFMX(W), MoveFMY(W)) = Units(Team, MoveFMX(W),
MoveFMY(W)) - 1
    Units(Team, MoveToX(W), MoveToY(W)) = Units(Team, MoveToX(W),
MoveToY(W)) + 1
    EXIT DO
  END IF
  LOOP
NEXT
Warper = 0
END SUB

```

```

SUB Xwin
CLS
CX = 320
CY = 240
PUT (CX - 12, CY - 12), prest(1)
FOR a = 225 TO 100 STEP -12.5
  PUT (CX - a - 12, CY - a - 12), lil(1001)
  PUT (CX + a - 12, CY - a - 12), lil(1001)
  PUT (CX - a - 12, CY + a - 12), lil(1001)
  PUT (CX + a - 12, CY + a - 12), lil(1001)
  CALL Delay(.25)
  PUT (CX - a - 12, CY - a - 12), lil(1001)
  PUT (CX + a - 12, CY - a - 12), lil(1001)
  PUT (CX - a - 12, CY + a - 12), lil(1001)
  PUT (CX + a - 12, CY + a - 12), lil(1001)
  IF a = 125 THEN PUT (CX - 12, CY - 12), pwork(1), PSET
NEXT
FOR a = 87.5 TO 25 STEP -12.5
  PUT (CX - a - 12, CY - a - 12), lil(1001)
  PUT (CX + a - 12, CY - a - 12), lil(1001)
  PUT (CX - a - 12, CY + a - 12), lil(1001)
  PUT (CX + a - 12, CY + a - 12), lil(1001)
  PUT (CX - 50, CY - 50 - a * 2), graf(1)
  CALL Delay(.125)
  PUT (CX - 50, CY - a * 2 - 50), graf(1)
  PUT (CX - 50, CY - a * 2 - 25), graf(1)
  CALL Delay(.125)
  PUT (CX - 50, CY - a * 2 - 25), graf(1)
  PUT (CX - a - 12, CY - a - 12), lil(1001)
  PUT (CX + a - 12, CY - a - 12), lil(1001)
  PUT (CX - a - 12, CY + a - 12), lil(1001)
  PUT (CX + a - 12, CY + a - 12), lil(1001)
NEXT
PUT (CX - 25 - 12.5, CY - 25 - 12.5), lil(1001)
PUT (CX + 25 - 12.5, CY - 25 - 12.5), lil(1001)
PUT (CX - 25 - 12.5, CY + 25 - 12.5), lil(1001)
PUT (CX + 25 - 12.5, CY + 25 - 12.5), lil(1001)
PUT (CX - 50, CY - 50), graf(1)
CALL Delay(1)
PUT (CX - 50, CY - 50), graf(1)
PUT (CX - 25 - 12.5, CY - 25 - 12.5), dlil(1001), PSET
PUT (CX + 25 - 12.5, CY - 25 - 12.5), dlil(1001), PSET
PUT (CX - 25 - 12.5, CY + 25 - 12.5), dlil(1001), PSET
PUT (CX + 25 - 12.5, CY + 25 - 12.5), dlil(1001), PSET
PUT (CX + 50, CY - 150), graf(1)
CALL Delay(.5)
PUT (CX + 50, CY - 150), graf(1)
PUT (CX + 100, CY - 50), graf(1)
PUT (CX - 12, CY - 12), prest(1), PSET
CALL Delay(.5)
PUT (CX + 100, CY - 50), graf(1)
PUT (CX - 12, CY - 12), prest(1)
PUT (CX + 150, CY - 50), graf(1)
PUT (CX + 50, CY - 25), prest(1)
CALL Delay(.5)
PUT (CX + 150, CY - 50), graf(1)
PUT (CX + 50, CY - 25), prest(1)

```

```
PUT (CX + 200, CY - 50), graf(1)
PUT (CX + 75, CY - 25), prest(1)
CALL Delay(.5)
PUT (CX + 200, CY - 50), graf(1)
PUT (CX + 75, CY - 25), prest(1)
PUT (CX + 100, CY - 25), prest(1)
CALL Delay(.5)
PUT (CX + 100, CY - 25), prest(1)
PUT (CX + 125, CY - 25), prest(1)
CALL Delay(.5)
PUT (CX + 125, CY - 25), prest(1)
PUT (CX + 150, CY - 25), prest(1)
CALL Delay(.5)
PUT (CX + 150, CY - 25), prest(1)
PUT (CX + 175, CY - 25), prest(1)
CALL Delay(.5)
PUT (CX + 175, CY - 25), prest(1)
PUT (CX + 200, CY - 25), prest(1)
CALL Delay(1)
END SUB
```