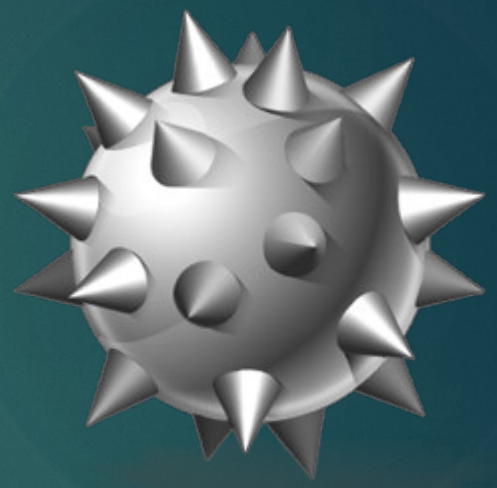
# MINESWEEPER



# The initialization

- Declaring a 2D array , the  beginning of the program

- #!/bin/bash
- 
- clear
- n=10
- m=10
- mine=15
- mine2=$mine
- i=0
- j=0
- k=0
- r=0
- declare -A board
- declare -A fore
- for((i=0;i<n;i++))

- do
- for((j=0;j<m;j++))
- do
- board[$i,$j]=0
- fore[$i,$j]=0
- done
- done
- i=0
- j=0
- d=0

# Creating the mine field

```
drawBoard()
{
#place mines
for((i=0;i<n;i++))
do
    for((j=0;j<m;j++))
    do
    r=$(shuf -i 1-100 -n 1)
    if ((board[$i,$j] !=-1))
    then
    if ((r<=5&&mine!=0))
    then
    board[$i,$j]=-1
    ((mine--))
    else
    board[$i,$j]=0
    fi
```

```
fi
        if ((i==n-1&&j==m-1&&mine!=0))
        then
        i=0
        j=0
        fi
        done
done
#update board
for((i=0;i<n;i++))
do
for((j=0;j<m;j++))
do
if ((board[$i,$j] == -1))
then
```

```
for((x=-1;x<=1;x++))
    do
    for((y=-1;y<=1;y++))
    do
    d=$((i+x))
    e=$((j+y))
    if ((d>-1 && d<n && e>-1 && e<m && board[$d,$e] != -1))
    then
    ((board[$d,$e]++))
    fi
    done
    done

fi
done
done
}
```

# To test the program

```
alpha()
{
#display-test
clear
echo
echo -e "\t\t\t\t    \033[32m$n X $m GRID ----- $mine2 MINEs\033[0m \n"
echo
echo -ne " \t\t\t \033[33mcol\033[0m\t "
for((i=0;i<m;i++))
do
echo -ne " \033[33m$i\033[0m  "
done
echo
echo
for((i=0;i<n;i++))
do
echo -ne "\t\t\t\033[33mrow $i\033[0m \t "
```

```
for((j=0;j<m;j++))
do
if((board[$i,$j]==0))
then
/bin/echo -ne "\e[0;37m . \e[0m "
elif((board[$i,$j]==1))
then
/bin/echo -ne "\e[1;34m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==2))
then
/bin/echo -ne "\e[1;32m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==3))
then
/bin/echo -ne "\e[1;35m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==4))
then
/bin/echo -ne "\e[1;33m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==5))
then
/bin/echo -ne "\e[0;34m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==6))
then
/bin/echo -ne "\e[0;32m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==7))
then
```

```
/bin/echo -ne "\e[0;35m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==8))
then
/bin/echo -ne "\e[0;33m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==-1))
then
/bin/echo -ne " \e[36;7m#\e[0m  "
fi
done
echo
echo
done
echo
}
```

# To display the board after each move

```
display()
{
#display-actual
clear
echo
echo -e "\t\t\t\t    \033[32m$n X $m GRID ----- $mine2 MINEs\033[0m \n"
echo
echo -ne " \t\t\t \033[33mcol\033[0m\t "
for((i=0;i<m;i++))
do
echo -ne " \033[33m$i\033[0m  "
done
echo
echo
for((i=0;i<n;i++))
do
echo -ne "\t\t\t\033[33mrow $i\033[0m \t "
```

```
for((j=0;j<m;j++))
do
if ((fore[$i,$j]==1))
then
if((board[$i,$j]==0))
then
/bin/echo -ne "\e[0;37m . \e[0m "
elif((board[$i,$j]==1))
then
/bin/echo -ne "\e[1;34m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==2))
then
/bin/echo -ne "\e[1;32m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==3))
then
/bin/echo -ne "\e[1;35m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==4))
then
/bin/echo -ne "\e[1;33m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==5))
then
```

```
/bin/echo -ne "\e[0;34m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==6))
then
/bin/echo -ne "\e[0;32m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==7))
then
/bin/echo -ne "\e[0;35m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==8))
then
/bin/echo -ne "\e[0;33m ${board[$i,$j]} \e[0m "
elif((board[$i,$j]==-1))
then
/bin/echo -ne "\e[1;31m * \e[0m "
fi
else
echo -ne " #  "
fi
done
echo
echo
done
echo
}
```

# To reset game values to initial value

- refresh()
- {
- for((i=0;i<n;i++))
- do
- for((j=0;j<m;j++))
- do
- board[$i,$j]=0
- fore[$i,$j]=0
- done
- done
- n=10
- m=10
- mine=15
- i=0
- j=0
- k=0
- r=0
- x=0
- y=0
- d=0
- e=0
- }

# Display losing scenario

- gameover()
- {
- for((i=0;i<n;i++))
- do
- for((j=0;j<m;j++))
- do
- if ((board[$i,$j] == -1))
- then
- fore[$i,$j]=1
- fi
- done
- done
- display
- echo -e "\e[0;31m \t\t\t\t All the mines have exploded. \e[0m"
- echo -e "\e[0;31m \t\t\t You could not survive the explosion. \e[0m"

- echo -e "\e[0;31m \t\t\t\t\t  Game Over \e[0m\n\n\n\n\n\n\n"
- echo -e "\e[0;36m \t\t\t\t  Press [ ENTER ] to continue. \e[0m\n\n"
- read
- }

# Display winning scenario

```
checkwin()
{
win=0
fmin=0
for((q1=0;q1<n;q1++))
do
for((q2=0;q2<m;q2++))
do
if ((fore[$q1,$q2] == 1))
then
if ((board[$q1,$q2] == -1))
then
fmin=1
fi
((win++))
fi
```

```
done
done
if ((fmin==1))
then
gameover
i=$((m*n))
fi
if ((win==m*n-mine2))
then
echo -e "\e[0;32m \t\t\t\t     You Have Won The Game !!!   \e[0m"
echo -e "\e[0;33m \t\t\t\t       Avoided all the mines.    \e[0m"
echo -e "\e[0;36m \t\t\t\t         All mines defused.      \e[0m\n\n"
echo -e "\e[0;36m \t\t\t\t    Press [ ENTER ] to continue. \e[0m\n\n"
read
i=$((m*n))
fi
}
```

# An alternate to win the game

- backdoor()
- {
- alpha
- echo -e "\e[0;32m \t\t\t\t  You Have Won The Game !!!  \e[0m"
- echo -e "\e[0;33m \t\t\t\t  Avoided all the mines.  \e[0m"
- echo -e "\e[0;36m \t\t\t\t  All mines defused.  \e[0m\n\n"
- echo -e "\e[0;36m \t\t\t\t  Press [ ENTER ] to continue. \e[0m\n\n"
- read
- i=$((m*n))
- }

# Initiates the changes to the board after each move

- chain()
- {
- i=$1
- j=$2
- fore[$i,$j]=1
- if ((board[$i,$j] == 0))
- then
- reveal $i $j
- revzero
- fi
- }

# Reveals the surrounding 8 neighbours of given tile

```
reveal()
{
i=$1
j=$2
fore[$i,$j]=1
for((x=-1;x<=1;x++))
do
for((y=-1;y<=1;y++))
do
    d=$((i+x))
    e=$((j+y))
    if ((x!=0 || y!=0))
    then
    if ((d>-1 && d<n && e>-1 && e<m && fore[$d,$e] == 0))
    then
    fore[$d,$e]=1
    fi
    fi
done
done
}
```

# Reveals the chain of zeros and their neighbors

```
revzero()
{
for((i=0;i<n;i++))
do
for((j=0;j<m;j++))
do
k=1
for((x=-1;x<=1;x++))
do
for((y=-1;y<=1;y++))
do
    d=$((i+x))
    e=$((j+y))
    if ((x!=0 || y!=0))
    then
    if ((d>-1 && d<n && e>-1 && e<m && fore[$d,$e] == 0))
    then
    k=0
    fi
    fi
done
done
```

```
if ((fore[$i,$j] == 1 && board[$i,$j] == 0 && k==0))
then
reveal $i $j
i=0
j=0
fi
done
done
}
```

# To give the instructions

- instructions()
- {
- fiv="\e[0;34m5\e[0m"
- sta="\e[1;31m*\e[0m"
- clear
- echo -e "\n\n\t\t\t\t    \e[5;32;40m Welcome  to  MINE MANIA \e[m\n"
- echo -e "\n\n\t\e[0;36mInstructions:\e[0m\n"
- echo -e "\tYou are a \e[0;33mSoldier\e[0m stranded in a minefield.Your mission is to navigate"
- echo -e "\tthrough it without setting off any mines.\n"
- echo -e "\tYou know that:"
- echo -e "\tThe field is a \e[0;33m10X10 grid\e[0m and there are \e[0;33m15 mines\e[0m.\n"

```
echo -e "\tThe purpose of the game is to open all the tiles of the field which do not contain"
echo -e "\ta mine. You lose if you set off a mine tile.\n"
echo -e "\tEvery non-mine tile you open will tell you the total number of mines in the eight"
echo -e "\tneighboring tiles.\n"
echo -e "\tFor example,\n"
echo -e "\t     #  #  #     $sta  #  $sta       $sta  #  $sta       $sta  #  #"
echo -e "\t     #  $fiv  #   =  $sta  $fiv  $sta   or   $sta  $fiv  #   or   $sta  $fiv  #"
echo -e "\t     #  #  #     #  $sta  #       $sta  #  $sta       $sta  $sta  $sta"
echo -e "\n"
echo -e "\tOnce you are sure that a tile contains a mine, do not open it and uncover all the"
echo -e "\trest of the tiles.\n"
```

- echo -e "\tEnter the X co-ordinate and Y co-ordinate of the tile to open it.\n"
- echo -e "\t\e[31mCAUTION\e[0m: Pressing [Enter] without co-ordinate input results in random selection."
- echo -e "\tIf you uncover a mine then the game terminates."
- echo -e "\n\n\t\t\t\t\t\e[5;32;40m  Happy Hunting ! \e[m\n\n\n\n"
- echo -e "\e[0;36m \t\t\t\t\t Press [ ENTER ] to continue. \e[0m\n\n"
- read
- }

# To begin a new game

```
newgame()
{
refresh
drawBoard
#alpha
display
for((i=0;i<m*n-mine;i++))
do
    x=500
    y=500
    read -p "   X co-ordinate : " x
    read -p "   Y co-ordinate : " y
    if [ "$x" == "mine" -a "$y" == "mania" ]
    then
    backdoor
    #i=$((m*n))
```

```
elif ((x>-1 && x<n && y>-1 && y<m))
    then
    chain $x $y
    #if ((board[$x,$y]==-1))
    #then
    #gameover
    #i=$((m*n))
    #else
    display
    #fi
    else
    echo -e "\e[0;31m \t\t\t Invalid Position \e[0m "
    fi
    checkwin
done
}
```

# The credits

```
rollcredits()
{
clear
echo -e "\n\n\n\n\n\n"
echo -e "\e[0;32m \t\t\t\t        See you next time.\e[0m\n"
echo -e "\e[1;34m \t\t\t\t     Thank You For Playing !!!\e[0m\n"
echo -e "\n\n\n\n\n\n\n\n\n\n"
echo -e "\e[1;34m \t\t\t\t\t Game created by:\e[0m\n\n"
echo -e "\e[0;32m \t\t\t\t\t      Sachin      \e[0m\n"
echo -e "\e[0;34m \t\t\t\t\t    Ramprakash    \e[0m\n"
echo -e "\e[0;33m \t\t\t\t\t    Abhishek K    \e[0m\n"
echo -e "\e[0;31m \t\t\t\t\t    Abhishek S    \e[0m\n\n\n\n\n\n\n\n\n"
echo -e "\e[0;36m \t\t\t\t      Press [ ENTER ] to exit. \e[0m\n\n"
read
echo
clear
}
```

# Main method

- #main method starts here
- p1='y'
- for (( ;(p1 == 'y') || (p1 == 'Y'); ))
- do
- instructions
- newgame
- echo -ne "\t\t\t  Would you like to play again? (y/n) "
- read p1
- done
- rollcredits
- clear