#!/bin/bash

#---------------Define--------------

ECHO="echo -ne"

ESC="\033["

OK=0

FALSE=1

#--------------Variable--------------

#ANSI ESC action

FLASH=5

REV=7

#color

NULL=0

BLACK=30

RED=31

GREEN=32

ORANGE=33

BLUE=34

PURPLE=35

SBLUE=36

GREY=37

#back color

BBLACK=40

BRED=41

BGREEN=42

BORANGE=43

BBLUE=44

BPURPLE=45

BSBLUE=46

BGREY=47

MINE='@'

FLAG='F'

NUL=' '

SHADOW='X'

X=0

Y=0

CurX=1 #cur's X

CurY=1 #cur's Y

OCurX=1 #old cur's X

OCurY=1 #old cur's Y

MCount=0 #count mine

FCount=0 #count flag

SCount=0 #count shadow

MXYp=0 #MXY Array's ptr

#---------------Array----------------

#if ${XY[]} == M { mine }

#if ${XY[]} == F { flag }

#if ${XY[]} == N { null }

#if ${XY[]} == S { shadow }

#if ${XY[]} == [1-8] { tip\_num }

#${XY[]} init in XYInit(i)

MXY[0]=""

#--------------Function--------------

function SttyInit ()

{

stty\_save=$(stty -g) #backup stty

clear

trap "GameExit;" 2 15

stty -echo

$ECHO "${ESC}?25l" #hidden cursor

return $OK

}

function GameExit ()

{

stty $stty\_save

stty echo

clear

trap 2 15

$ECHO "${ESC}?25h${ESC}0;0H${ESC}0m"

exit $OK

}

#print help

function Help ()

{

msg="Move:w s a d Dig:j Flag:f NewGame:n Exit:x --CopyRight-- -2005-10-28 BitBull--"

$ECHO "${ESC}${REV};${RED}m${ESC}24;1H${msg}${ESC}${NULL}m"

return $OK

}

#print dialog window in screen

function PMsg ()

{

local title="$1" content="$2" greeting="$3"

$ECHO "${ESC}${RED}m"

$ECHO "${ESC}11;20H ------------------------------------------- "

$ECHO "${ESC}12;20H| ======>$title<====== |"

$ECHO "${ESC}13;20H| $content |"

$ECHO "${ESC}14;20H| ======>$greeting<====== |"

$ECHO "${ESC}15;20H ------------------------------------------- "

$ECHO "${ESC}${NULL}m"

return $OK

}

#print menu and player choose level,then ${X,Y,MCount,FCount,SCount} init

function Menu ()

{

local key

$ECHO "${ESC}6;1H${ESC}${RED}m"

cat<<MENUEND

+++++++++++++++++++++++++++++

+ (1) Easy +

+ (2) Normal +

+ (3) Hardly +

+ (4) Exit +

+++++++++++++++++++++++++++++

MENUEND

$ECHO "${ESC}${NULL}m"

while read -s -n 1 key

do

case $key in

1) X=10;Y=10;MCount=10;FCount=10;SCount=100;break

;;

2) X=20;Y=14;MCount=28;FCount=28;SCount=280;break

;;

3) X=36;Y=18;MCount=65;FCount=65;SCount=648;break

;;

4) GameExit

;;

esac

done

return $OK

}

#receive CurX CurY,put it into XY[CurX+X\*(CurY-1))]

#if $# == 3;write into XY[]

#if $# == 2;read from XY[]

function XYFormat ()

{

local XTmp=$1 YTmp=$2

if [[ $# -eq 3 ]]

then XY[$XTmp+$X\*($YTmp-1)]=$3

else echo ${XY[$XTmp+$X\*($YTmp-1)]}

fi

return $OK

}

function DrawInit ()

{

local DIline DIline2

DIline=$( for (( i=1; i<$((X\*2)); i++ )) do $ECHO '-';done )

DIline2=$( for (( i=0; i<X; i++ )) do $ECHO "|${ESC}${SBLUE}mX${ESC}${NULL}m";done )

clear

Help

$ECHO "${ESC}1;1H+${DIline}+"

for (( i=0; i<Y; i++ ))

do

$ECHO "${ESC}$((i+2));1H${DIline2}|"

done

$ECHO "${ESC}$((Y+2));1H+${DIline}+"

return $OK

}

#${XY[\*]}=S

function XYInit ()

{

for (( i=1; i<=$X; i++ ))

do

for (( j=1; j<=$Y; j++ ))

do

XYFormat $i $j S

done

done

return $OK

}

#check X Y

function CheckXY ()

{

local XYTmp="$1 $2"

for(( i=0; i<MXYp; i++ ))

do

if [[ "${MXY[i]}" == "$XYTmp" ]]

then return $FALSE

fi

done

return $OK

}

#RANDOM mine's X Y

function XYRand ()

{

local XTmp YTmp

for(( i=0; i<MCount; i++ ))

do

while :

do

XTmp=$(( RANDOM % ( X - 1 ) + 1 ))

YTmp=$(( RANDOM % ( Y - 1 ) + 1 ))

CheckXY $XTmp $YTmp

if [[ "$?" == "$OK" ]]

then

XYFormat $XTmp $YTmp M

MXY[i]="$XTmp $YTmp"

(( ++MXYp ))

break

else continue

fi

done

done

return $OK

}

#DEBUG

# print ${XY[\*]} into ./mine.tmp

#you can read mine.tmp to know where is mine,xixi~~:)

#M is mine

function DEBUGPXY ()

{

rm mine.tmp>/dev/null 2>&1

for(( i=1; i<=$Y; i++ ))

do

for(( j=1; j<=$X; j++))

do

$ECHO "$(XYFormat $j $i)">>mine.tmp

done

$ECHO "\n">>mine.tmp

done

return $OK

}

#move cur

#usage:CurMov [UP|DOWN|LEFT|RIGHT]

function CurMov ()

{

local direction=$1 Xmin=1 Ymin=1 Xmax=$X Ymax=$Y

OCurX=$CurX

OCurY=$CurY

case $direction in

"UP") if [[ $CurY -gt $Ymin ]];then (( CurY-- ));fi

;;

"DOWN") if [[ $CurY -lt $Ymax ]];then (( CurY++ ));fi

;;

"LEFT") if [[ $CurX -gt $Xmin ]];then (( CurX-- ));fi

;;

"RIGHT")if [[ $CurX -lt $Xmax ]];then (( CurX++ ));fi

;;

esac

if [[ $CurX != $OCurX || $CurY != $OCurY ]]

then DrawPoint $CurX $CurY CUR

fi

return $OK

}

#display point

#include cur,flag,mine,shadow,nul,tip [1-8]

function DrawPoint ()

{

local TCurX=$(( $1 \* 2 )) TCurY=$(( $2 + 1 )) Type=$3

local TOCurX=$(( OCurX \* 2 )) TOCurY=$(( OCurY + 1 ))

local colr=0 osign=0 sign=0

case $Type in

"CUR")

case $(XYFormat $OCurX $OCurY) in

F) colr=$PURPLE;osign=$FLAG;;

N) colr=$NULL;osign=$NUL;;

[1-8]) colr=$ORANGE;osign=$(XYFormat $OCurX $OCurY);;

[SM]) colr=$SBLUE;osign=$SHADOW;;

esac

case $(XYFormat $CurX $CurY) in

F) sign=$FLAG;;

N) sign=$NUL;;

[1-8]) sign=$(XYFormat $CurX $CurY);;

[SM]) sign=$SHADOW;;

esac

$ECHO "${ESC}${colr}m${ESC}${TOCurY};${TOCurX}H${osign}${ESC}${NULL}m"

$ECHO "${ESC}${REV};${FLASH};${ORANGE}m${ESC}${TCurY};${TCurX}H${sign}${ESC}${NULL}m"

;;

"SHADOW")

$ECHO "${ESC}${SBLUE}m${ESC}${TCurY};${TCurX}H${SHADOW}${ESC}${NULL}m"

;;

"MINE")

$ECHO "${ESC}${REV};${RED}m${ESC}${TCurY};${TCurX}H${MINE}${ESC}${NULL}m"

;;

"FLAG")

$ECHO "${ESC}${TCurY};${TCurX}H${ESC}${PURPLE}m${FLAG}${ESC}${NULL}m"

;;

[1-8])

$ECHO "${ESC}${TCurY};${TCurX}H${ESC}${ORANGE}m${Type}${ESC}${NULL}m"

;;

"NUL")

$ECHO "${ESC}${TCurY};${TCurX}H${NUL}"

esac

return $OK

}

#check xy

function Loop ()

{

local XYTmp="$1 $2"

for (( i=0; i<MXYp; i++ ))

do

if [[ "$XYTmp" == "${MXY[i]}" ]]

then $ECHO 1

fi

done

return $OK

}

#count around mine

#A B C

#D X E

#F G H

#return mine's number

function CountM ()

{

local Xmin=1 Ymin=1 Xmax=$X Ymax=$Y minecount=0 n=0

#A

if [[ ( $CurX -gt $Xmin ) && ( $CurY -gt $Ymin ) ]]

then

n=$( Loop $((CurX-1)) $((CurY-1)) )

(( minecount += n ))

n=0

fi

#B

if [[ $CurY -gt $Ymin ]]

then

n=$( Loop $CurX $((CurY-1)) )

(( minecount += n ))

n=0

fi

#C

if [[ ( $CurX -lt $Xmax ) && ( $CurY -gt $Ymin ) ]]

then

n=$( Loop $((CurX+1)) $((CurY-1)) )

(( minecount += n ))

n=0

fi

#D

if [[ $CurX -gt $Xmin ]]

then

n=$( Loop $((CurX-1)) $CurY )

(( minecount += n ))

n=0

fi

#E

if [[ $CurX -lt $Xmax ]]

then

n=$( Loop $((CurX+1)) $CurY )

(( minecount += n ))

n=0

fi

#F

if [[ ( $CurX -gt $Xmin ) && ( $CurY -lt $Ymax ) ]]

then

n=$( Loop $((CurX-1)) $((CurY+1)) )

(( minecount += n ))

n=0

fi

#G

if [[ $CurY -lt $Ymax ]]

then

n=$( Loop $CurX $((CurY+1)) )

(( minecount += n ))

n=0

fi

#H

if [[ ( $CurX -lt $Xmax ) && ( $CurY -lt $Ymax ) ]]

then

n=$( Loop $((CurX+1)) $((CurY+1)) )

(( minecount += n ))

n=0

fi

return $minecount

}

#dig

#if mine ,gameover

#else tip around mine's number

function Dig ()

{

local key minenum=0

case $(XYFormat $CurX $CurY) in

M)

DrawPoint $CurX $CurY MINE

read -s -n 1 key

GameOver "Game Over"

;;

S)

CountM

minenum=$?

if [[ $minenum -eq $NULL ]]

then

XYFormat $CurX $CurY N

DrawPoint $CurX $CurY NUL

else

XYFormat $CurX $CurY $minenum

DrawPoint $CurX $CurY $minenum

fi

(( SCount-- ))

if [[ $SCount -eq $MCount ]]

then GameOver "Well Done"

fi

;;

esac

DrawPoint $CurX $CurY CUR

return $OK

}

#draw flag's number

function DrawFCount ()

{

$ECHO "${ESC}22;34H${ESC};${PURPLE}mFLAG=${FCount} ${ESC}${NULL}m"

}

#sign mine

function Flag ()

{

local XYTmp="$CurX $CurY";stat=$FALSE

case $(XYFormat $CurX $CurY) in

F)

for (( i=1; i<MXYp; i++ ))

do

if [[ "${MXY[i]}" == "$XYTmp" ]]

then XYFormat $CurX $CurY M;stat=$OK;break

fi

done

if [[ $stat == $FALSE ]]

then XYFormat $CurX $CurY S

fi

DrawPoint $CurX $CurY SHADOW

(( FCount++ ))

DrawFCount

;;

[SM])

if [[ $FCount -eq $NULL ]]

then return $FALSE

fi

DrawPoint $CurX $CurY FLAG

XYFormat $CurX $CurY F

(( FCount-- ))

DrawFCount

;;

esac

DrawPoint $CurX $CurY CUR

return $OK

}

function GameOver ()

{

local key msgtitle=$1

PMsg "$msgtitle" "Do you want replay?<y/n>" "Thank You"

while read -s -n 1 key

do

case $key in

[yY]) exec $(dirname $0)/$(basename $0);;

[nN]) GameExit;;

\*) continue;;

esac

done

return $OK

}

#main

#drawscreen and control

function Main ()

{

local key

XYInit

XYRand

############################

# if you enable DEBUGPXY,

#you can know where is mine

# DEBUGPXY #delete this line's #

#then cat ./mine.tmp

############################

DrawPoint $CurX $CurY CUR

DrawFCount

while read -s -n 1 key

do

case $key in

[wW]) CurMov UP;;

[sS]) CurMov DOWN;;

[aA]) CurMov LEFT;;

[dD]) CurMov RIGHT;;

[jJ]) Dig;;

[fF]) Flag;;

[nN]) exec $(dirname $0)/$(basename $0);;

[xX]) GameExit;;

esac

done

return $OK

}

#---------------Main-----------------

SttyInit

Menu #X Y MCount FCount SCount OK!

DrawInit

Main