***National University of Computer and Emerging Sciences***



# Game Proposal

*For*

*Computer Organization and Assembly Language*

|  |  |
| --- | --- |
| Course Instructor(s) Lab Instructor(s) | Mam Asra Ashraf  Sir Uzair Tahir |
| Semester | Fall 2019 |

**Department of Computer Science**

**Team Name:**

The KillerZ

**Game:**

 Bomber Battle

**Team Members:**

* M. Usama Butt (18F-0428)
* Hafiz Attiq Ur Rehman (18F-0174)
* Muhammad Numan (18F-0414)

## Bomber Battle

**Description:**

In bomber game, you will move through multi levels. The general goal throughout the series is to complete the levels by strategically placing bombs in order to kill enemies and destroy obstacles.

Exploding bombs can set off other bombs kill or injure enemies and destroy obstacles.

You have to place bombs to destroy brick to find this. Bomb will automatically explode after several seconds or manually explode when you get blow up control. You will lose the blow up control if you die. You can get this one in some next level.

Be careful bombs can also destroy yourself if you in exploding range.

**Features:**

* Beautiful and stunning game.
* Attractive Bomber games play.
* You will control the hero to get through many game levels from easy to difficult.
* Try to find out Power Up item in each level to improve your skill.

**Workout:**

My group member will work on different functions and I will work on the remaining things. Moreover, both of us equally contributed in our project.

**Code:**

include Irvine32.inc

.data

blockArray dd 81 dup('0')

minearray db 81 dup('0')

playerarray db 81 dup ('#')

playername db "What is your Name :",0

countmines db 0

totalindex db 81

cornertemp dd ?

minetemp dd ?

score db 0

remain db 0

row dd 0

col dd 0

index dd 9

got dd ?

win db 1

scr db ' Score =',0

rsize =0

player db "player is at following index :",0

lgame db "You are lose & mines destoyed you!!!!",0

tot db "Total Block :",0

remin db "Remaining block :",0

bom db "Mines :",0

winner db "You Win the game... Final score is :",0

.code

main proc

;call Randomize

;call Rand2

;call block

;mov ecx,25

call initialblock

call mineblock

call checks

call mineblockprint

call count

mov ecx,100

mov bl,totalindex

sub bl,countmines

mov remain,bl

ford:

mov edx,offset tot

call writestring

mov al,totalindex

call writedec

call crlf

mov edx,offset remin

call writestring

mov al,remain

call writedec

call crlf

mov edx,offset bom

call writestring

mov al,countmines

call writedec

call crlf

call movement

call crlf

mov edx,offset player

call writestring

mov eax,index

;call writedec

call crlf

mov edx,offset scr

call writestring

mov al,score

call writedec

call crlf

cmp win,0

jz lose

call crlf

call crlf

call crlf

call blockprint

;call mineblockprint

mov bl,countmines

cmp remain,bl

jz wins

jmp ford

wins:

mov edx,offset winner

call writestring

mov al,score

call writedec

lose:

mov edx,offset lgame

call writestring

final:

call readchar

exit

main endp

initialblock proc

mov ecx,81

mov al, 9

mov esi,offset blockarray

initialblockloop:

mov [esi],al

add esi,1

inc al

loop initialblockloop

ret

initialblock endp

blockprint proc

mov ecx,81

mov ebx,0

mov eax,0

mov esi,OFFSET blockarray ; table offset

mov edi,0

mov al,9

call writechar

mov al,'0'

call writechar

loop1:

mov al,[esi]

call writedec

mov al,'='

call writechar

mov al,playerarray[edi]

call writechar

mov al,32

call writechar

call writechar

inc ebx

cmp ebx,9

jz zero

jmp next

zero:

mov ebx,0

call crlf

call crlf

mov al,9

call writechar

next:

inc esi

inc edi

loop loop1

ret

blockprint endp

mineblock proc

call Randomize

mov ecx,15

mov esi,0

L1:

mov eax,88 ; values 0-1010

call Randomrange

add eax,1

mov esi,eax

mov ebx,9

div bl

cmp ah,9

jz noway

jmp yesway

noway:

dec esi

yesway:

cmp ah,0

jz rem

jmp quo

rem:

inc esi

quo:

mov mineArray[esi],'M'

loop L1

ret

mineblock endp

mineblockprint proc

mov ecx,81

mov ebx,0

mov eax,0

mov esi,OFFSET minearray ; table offset

loop1:

mov al,[esi]

call writechar

mov al,32

call writechar

inc ebx

cmp ebx,9

jz zero

jmp next

zero:

mov ebx,0

call crlf

call crlf

next:

inc esi

loop loop1

ret

mineblockprint endp

checks proc

mov esi,0

mov ecx,81

loop2:

cmp minearray[esi],'M'

jz minefind

jmp next

minefind:

mov minetemp,esi

inc esi

cmp minearray[esi],'M'

jz n1

add minearray[esi],1

n1:

mov esi,minetemp

dec esi

cmp minearray[esi],'M'

jz n2

add minearray[esi],1

n2:

mov esi,minetemp

cmp esi,9

jg ok1

jmp next1

ok1:

sub esi,9

cmp minearray[esi],'M'

jz n3

add minearray[esi],1

n3:

dec esi

cmp minearray[esi],'M'

jz n4

add minearray[esi],1

n4:

add esi,2

cmp minearray[esi],'M'

jz n5

add minearray[esi],1

n5:

mov esi,minetemp

next1:

cmp esi,81

jl ok2

jmp next2

ok2:

add esi,9

cmp minearray[esi],'M'

jz n6

add minearray[esi],1

n6:

dec esi

cmp minearray[esi],'M'

jz n7

add minearray[esi],1

n7:

add esi,2

cmp minearray[esi],'M'

jz n8

add minearray[esi],1

n8:

mov esi,minetemp

next2:

next:

inc esi

dec ecx

cmp ecx,0

jnz loop2

ret

checks endp

;gotplayer proc

;call readchar

;cmp al,'w'

;jz up

;jmp nexttop

;up:

;cmp row,0

;jz top

;jmp nottop

;top:

;mov row,10

;add index,72

;jmp nexttop

;nottop:

;inc row

;add index,9

;nexttop:

;call readchar

;cmp al,'s'

;jz down

;jmp nextbottom

;down:

;cmp row,9

;jz bottom

;jmp notbottom

;bottom:

;mov row,0

;sub index,72

;jmp nextbottom

;notbottom:

;dec row

;sub index,9

;nextbottom:;

;call readchar

;cmp al,'d'

;jz right

;jmp nextright

;right:

;inc col

;cmp col,9

;jz rightzero

;jmp notrightzero

;rightzero:

;mov col,0

;cmp row,10

;jz rightdone

;jmp noright

;rightdone:

;mov row,0

;noright:

;inc row

;notrightzero:

;inc index

;nextright:

;mov edx,offset player

;call writestring

;call crlf

;mov edx,offset rowno

;call writestring

;mov al,row

;call writechar

;call crlf

;mov edx,offset colno

;mov al,col

;call writechar

;ret

;gotplayer endp

movement proc

mov edx,offset player

call writestring

mov eax,index

mov got,eax

call writedec

call crlf

mov eax,got

;call writedec

;

call readchar

cmp al,'w'

jz up1

jmp noup

up1:

cmp index,18

jl sup

jmp nsup

sup:

add index,81

mov eax,index

;

call writedec

nsup:

sub index,9

mov eax,index

;

call writedec

noup:

cmp al,'s'

jz down1

jmp nodown

down1:

cmp index,81

jl sdown

sub index,72

mov eax,index

;

call writedec

jmp nodown

sdown:

add index,9

mov eax,index

;

call writedec

nodown:

cmp al,'d'

jz right

jmp notright

right:

inc index

cmp index,89

jg zeroyes

jmp notright

zeroyes:

mov index,9

notright:

cmp al,'a'

jz left

jmp notleft

left:

dec index

cmp index,9

jl yesleft

jmp notleft

yesleft:

mov index,89

notleft:

cmp al,'o'

jz dom

jmp noopen

dom:

mov edi,0

mov ebx,0

mov ebx,got

mov esi, got

sub esi,9

cmp minearray[esi],'M'

jz lose

loopopen:

cmp minearray[esi],'M'

jz notlose

;call crlf

;mov eax,got

;call writedec

mov bl,minearray[esi]

mov edi,esi

;sub bl,48

call crlf

mov eax,ebx

;call writedec

mov playerarray[esi],bl

add edi,9

mov bh,minearray[edi]

cmp bh,'M'

jz yesmin

jmp nomin

nomin:

mov playerarray[edi],bh

dec remain

inc score

yesmin:

sub edi,9

inc score

dec remain

inc esi

inc edi

jmp loopopen

notlose:

jmp noopen

lose:

call crlf

mov win,0

noopen:

call crlf

ret

movement endp

count proc

mov esi,0

mov ecx,81

loopmine:

cmp minearray[esi],'M'

jz nomine

jmp you

nomine:

inc countmines

you:

inc esi

loop loopmine

ret

count endp

end main