//Run in c compiler. This is the code of ball brick game which i have tried.

```
#include <stdio.h>

#include <conio.h>

#include <time.h>

void delay(int n)

{

  int ms=10000*n;

 clock_t st=clock();   //time delay

 while(clock()<st+ms);

  }

int main()

{
int row,col,count;
printf("enter size of matrix \n");   //here we are taking input from user about the size of matrix
scanf ("%d %d",&row,&col);
printf("size of matrix is %d x %d\n",row,col);


int i,j,l1=1,temp1=0,temp2=0,temp3=0,temp4=0,cc=0;

char a[row][col],choice='q',move,type;
int m=row/2,n=col/2,posx,posy;//posx means position in x direction and posy means position in y
direction  at which type will be entered
int x=n;
for(i=1;i<=row;i++){
for(j=1;j<=col;j++)
{
printf("enter brick position and brick type\n");
scanf("%d%d%c",&posx,&posy,&type);// here input is taking from user and type of bricks  also
a[posx][posy]=type;
}
```

```
}
printf("Enter ball count\n"); // means chances or life for playing the game
scanf("%d",&count);
again:

//initialisation of background

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

 {

for(j=1;j<=col;j++)

 {

if(i==1)
{
a[i][col]='W';
}

if(i==row)
{
a[i][col]='G';
}

if(j==1 || j==col)
{
a[i][j]='W';
}

a[m][n]='=';



//updated version

update:

clrscr();



//for player's pad or slider
```

```
a[row][x]='o';


//for assignment of movement of ball
//1) for left-up movement (if choice is q)
if(choice=='q'&&m!=1&&n!=1)
{
m--;n--;
a[m][n]='='; //this will show movement of ball or path of ball striking the brick or walls

 }
//2) for right-up movement (if choice is w)
else if(choice=='w'&&m!=1&&n!=col-1)
{
m--;n++;
a[m][n]='=';


 }
//3) for left-down movement (if choice is e)
else if(choice=='e'&&m!=row-1&&n!=1)
{
m++;n--;
a[m][n]='=';
```

```
 }
//4) for right-down movement (if choice is r)
else if(choice=='r'&&m!=row-1&&n!=col-1)
{
m++;n++;
a[m][n]='=';

 }

temp3=m;
temp4=n;

//For executing figure
for(i=1;i<=row;i++)
 {
for(j=1;j<=col;j++)
 {
printf("%c",a[i][j]);
 }
printf("\n");
 }
```

```
//for hitting pad

move=getch();



//1) for left movement of slider

if((move=='a'||move=='s')&&x!=1)

 {

x--;



  }
//2) for right movement of slider
if((move=='k'||move=='l')&&x!=col)

  {

x++;



   }



//for movement of ball

//critical condition - [NW to WS]

if((m==1&&temp3==temp1-1&&temp4==temp2-1))

 {

 choice='e';
```

```
 }
//critical condition - [NE to SE]
else if((m==1&&temp3==temp1-1&&temp4==temp2+1))
 {
 choice='r';
 }
//critical condition - [SE to NE]
else if((m==14&&temp3==temp1+1&&temp4==temp2+1))
 {
 choice='w';
 }
//critical condition - [WS to NW]
else if((m==14&&temp3==temp1+1&&temp4==temp2-1))
 {
 choice='q';
 }
//critical condition - [NW to NE]
else if((n==1&&temp3==temp1-1&&temp4==temp2-1))
 {
 choice='w';
 }
//critical condition - [WS to SE]
```

```
else if((n==1&&temp3==temp1+1&&temp4==temp2-1))
 {
 choice='r';
 }
//critical condition - [SE to WS]
else if((n==47&&temp3==temp1+1&&temp4==temp2+1))
 {
 choice='e';
 }
//critical condition - [NW to NE]
else if((n==47&&temp3==temp1-1&&temp4==temp2+1))
 {
 choice='q';
 }
//critical condition - [right bottom corner]
else if(m==row-1&&n==col-1)
 {
 choice='q';
 }
//critical condition - [left bottom corner]
else if(m==row-1&&n==1)
```

```
{
 choice='w';
 }
//critical condition - [up right corner]
else if(m==1&&n==col-1)
 {
 choice='e';
 }
//critical condition - [left top corner]
else if(m==1&&n==1)
 {
 choice='r';
 }


//previous move of ball
temp1=temp3;
temp2=temp4;
x=x+col-1;
//game ending
if(temp3==row-1&&(temp4==x-(col-1)||temp4==x-(col-2)||temp4==x-(col-3)||temp4==x-(col-4)||te
mp4==x-(col-5)||temp4==x-col||temp4==x-(col+1)||temp4==x-(col+2)||temp4==x-(col+3)))
 {
```

```
 }

else if(temp3==row-1)

{

 goto gameend;

}

x=x-(col-1);

goto update;

gameend:
count--;
printf("Ball count is %d\n",count);
clrscr();


if(count>0)

{

x=col/2,m=row/2,n=col/2,l1=1,temp1=0,temp2=0,temp3=0,temp4=0,cc=0,choice='q';

goto again;

}

else

{printf("GAMEOVER\n");

 goto gameend;

}

return 0;

}
```