VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

Department of Computer Engineering

COMPUTER GRAPHICS (CG)

Mini Project Report

On

BRICK GAME

Submitted in partial fulfilment of the requirements of
Second Year Computer Engineering

By
SAKSHEE SAWANT– D7B – 51
SRISHTI VAZIRANI– D7B – 66

Supervisor:
Mrs. Pallavi Saindane

DEPARTMENT OF COMPUTER ENGINEERING V.E.S
INSTITUTE OF TECHNOLOGY

2019-20

TABLE OF CONTENTS

ABSTRACT:

The abstract of this game is to hit the layers of bricks by a ball using keyboard and score points,10 points for hitting each brick.
meanwhile the player has to protect the ball from hitting the lower wall with the help of a paddle. If this happens the player loses.

Keyboard function:
For player :
'^' for moving up
'v' for moving down

INTRODUCTION:-

We have implemented a simple brick game. Here is how simple brick game works. You have layers of bricks and a tossed up ball with which to break the layers. Controlling the momentum of the ball a paddle which you have to move from side to side. When the ball is bounced up by the paddle, it will hit a brick and then the brick is gone. After that the ball will come down, if the paddle misses the ball, game is over. Users can use the keypad to control the position of paddle.

The player gets 10 points rewarded for each slab the ball hits.
At the end of the game , the final points of the player is displayed and a message saying " Sorry! You loose the game .Try again!!!" is displayed.. The basic user interface may be similar to figure
1.

Figure 1: Game User Interface

Requirement Analysis:

Software:C programming language, TURBO C compiler,Graphics library.
Hardware:RAM: 2GB and higher,Hard Disk:40GB and higher,Mouse: 2 button,Monitor

A) LIBRARIES USED:

1) graphics.h:
graphics.h is a header that provides access to a simple graphics library that makes it possible to draw lines, rectangles, ovals, arcs, polygons, images, and strings on a graphical window.

2) stdlib.h:
stdlib.h is the header of the general-purpose standard library of C programming language which includes functions involving memory allocation, process control, conversions, and others.

3) stdio.h:
Input and Output operations can also be performed in C++ using the C Standard Input and Output Library (cstdio, known as stdio.h in the C language). This library uses what are called streams to operate with physical devices such as keyboards, printers, terminals or any other type of files supported by the system. Streams are an abstraction to interact with these in a uniform way; All streams have similar properties independently of the individual characteristics of the physical media they are associated with.

4) conio.h: The conio.h header file used in C programming language contains functions for console input/output. Some of its most commonly used functions are clrscr, getch, getche, kbhit, etc. They can be used to clear screen, change the color of text and background, move text, check whether a key is pressed or not and perform other tasks.

5) dos.h : Dos.h in C: Dos.h header file of C language contains functions for handling interrupts, producing sound, date and time functions etc. It is Borland specific and works in Turbo C compiler. Its functions are: delay,sleep,setdate etc.

6) iostream.h-iostream in c++ is basically an input/output streams library. It includes

functionalities to read and write from streams. You only need to include it if you wish to use

streams. <iostream> declares the objects used to communicate through the standard input and output.

COMPUTER GRAPHICS:

Computer graphics is one of the most exiciting and rapidly growing computer fields. It is also an extremely effective medium for communication between man and computer; a human being can understand the information content of a displayed diagram or perspective view much faster than he can understand a table of numbers or text containing the same information. Thus computer graphics is being used more extensively.

There is a lot of development in hardware and software required to generate images, and nowadays the Cost of hardware and software is dropping rapidly. Due to this, interactive computer graphics is becoming available to more and more people.

Computer graphics started with the display of data on hardcopy plotters and cathode ray tube (CRT) screens soon after the introduction of computers themselves. It has grown to include the creation, storage and manipulation of models and manipulation of models and images of objects. These models come from a diverse and expanding set of fields, and include physical, mathematical, engineering, architectural, and even conceptual structures, natural phenomena, and so on.

Computer graphics today is largely interactive. The user controls the contents, structure and appearance of objects and their displayed images by using input devices, such as a keyboard, mouse, or touch sensitive panel on the screen. The handling of such devices is included in the study of computer graphics, because of the close relationship between the input devices and the display.

IMPLEMENTATION:

CODE:

```
#include"graphics.h"
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<dos.h>
#include<stdlib.h>
int getkey();
void main()
{ int
gd,gm;
gd=DETECT;
initgraph(&gd,&gm,"C:\\TurboC3\\BGI");
cleardevice();
```

```
int ar,xc,yc,xr=0,yr=195,a=1,b=1,c=0,are;
void *bu,*buf; int X=0,Y=0,s,area1;
void *buff1;
rectangle(0,0,50,25);
setfillstyle(6,6); floodfill(2,2,15);
ar=imagesize(0,0,50,25);
bu=malloc(ar);
getimage(0,0,50,25,bu);
putimage(0,0,bu,XOR_PUT);
rectangle(0,0,50,25);
setfillstyle(6,8); floodfill(2,2,15);
are=imagesize(0,0,50,25);
buf=malloc(are);
getimage(0,0,50,25,buf);
putimage(0,0,buf,XOR_PUT);
for(int j=0;j<180;j+=27) for(int
i=0;i<600;i+=52)
putimage(0+i,27+j,bu,XOR_PUT);
putimage(0,27,bu,XOR_PUT);
putimage(572,27,bu,XOR_PUT);
putimage(0,27,buf,XOR_PUT);
putimage(572,27,buf,XOR_PUT);
setcolor(3); rectangle(80,445,159,452);
setfillstyle(1,1); floodfill(82,447,3);
area1=imagesize(80,445,159,452);
buff1=malloc(area1);

getimage(80,445,159,452,buff1);
setcolor(4); line(0,479,640,479);
int area,x=325,y=325,ch,xdirn=-1,ydirn=-1,step;
int maxx,maxy;
void *buff;
setcolor(WHITE);
setfillstyle(SOLID_FILL,RED); circle(350,350,5);
floodfill(350,350,WHITE);
area=imagesize(345,345,355,355);
buff=malloc(area);
getimage(345,345,355,355,buff);
putimage(345,345,buff,XOR_PUT);

while (1)
{
putimage(x, y, buff, XOR_PUT);
delay(15);
putimage(x, y, buff, XOR_PUT);
x=x+(xdirn*2);
y=y+(ydirn*3);
if ( x + 10 - 1 > 640 )
{
```

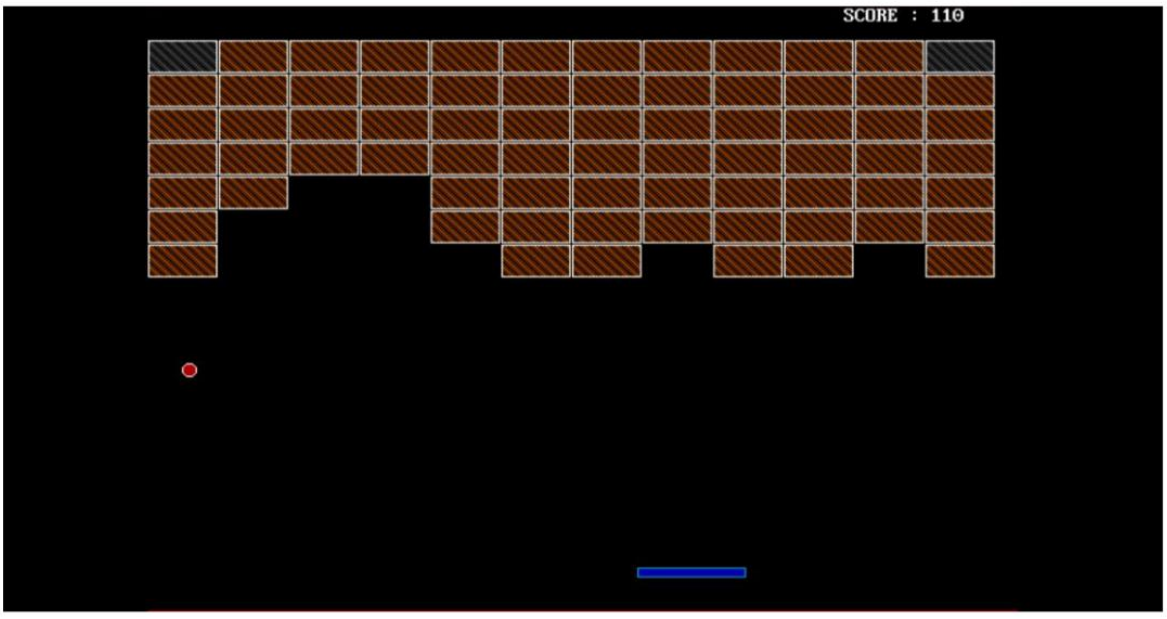
```
xdirn*=-1;
x = 640 - 10 + 1;
} if (x
< 0)
{
xdirn*=-1;
x = 0;
}
if ( y + 10 - 1 > 470 )
{
// ydirn*=-1;
// y = 470 - 10 + 1;
cleardevice();
outtextxy(200,200,"Sorry! You loose the game.");
outtextxy(250,240,"Try Again!!!"); gotoxy(30,8);

cout<<"Total Score : "<<c; delay(5000);
getch(); goto tt;
}
if(c==1020)
{
outtextxy(180,200,"Congrats!");
gotoxy(30,8); cout<<"Total
Score : "<<c; delay(5000);
getch(); goto tt;
}
if ( getpixel(x,y)==1 )
{
sound(500);
delay(15); nosound();
ydirn*=-1;
}
if (getpixel(x,y)==6)
{
sound(100);
delay(50);
nosound();
ydirn*=-1;
xc=x; yc=y;
a=xc/52;
b=(yc/27);
xr=a*52;
yr=b*27;
putimage(xr,yr,bu,XOR_PUT);
c+=10;
}
if (getpixel(x,y)==8)
{
sound(800);
```

```
delay(200); nosound();
ydirn*=-1; xc=x;
yc=y; a=xc/52;
b=(yc/27); xr=a*52;
yr=b*27;
putimage(xr,yr,buf,XOR_PUT);
c+=100;
} if( y
< 0 )

{
ydirn*=-1;
y=0;
}
gotoxy(65,1);
cout<<"SCORE : "<<c;
if( kbhit() )
{
s=getkey();
if(s!=1) {
if(X>480)
{
X=480;
putimage(160+X,445+Y,buff1,XOR_PUT);
}
if(X<-80)
{
X=-80;
putimage(80+X,445+Y,buff1,XOR_PUT);
}
putimage(80+X,445+Y,buff1,XOR_PUT);
//if(s==72) //
Y+=-40;
if(s==75)
X+=-40;
//if(s==80) //
Y+=40;
if(s==77)
X+=40;
putimage(80+X,445+Y,buff1,XOR_PUT);
//cout<<X;
}
if(s==1) { tt:
free(bu);
free(buff);
free(buff1);
closegraph();
exit(0); }
} } free(bu);
```

```
free(buff);
free(buff1);
closegraph();
}

int getkey() {
union REGS j,o;
j.h.ah=0;
int86(22,&j,&o);
return(o.h.ah); }
```

SNAPSHOTS:


Total Score : 20
Sorry! You loose the game.
Try Again!!!


SCORE : 110

CONCLUSION:-

Hence, we have tried to implement "BRICKS GAME" using computer graphics in TURBOC. Some in built functions were used in this implementation which were helpful in implementing this project. While developing this project, vastness and high scope of computer graphics subject was realized. Through this game, we got aware about the various in built functions and the process of building a game.