**COMPUTER PROJECT**

PONG GAME

A Tribute to the original game by Atari



A PROJECT BY: - ARCHAS SRIVASTAVA XI-C

INDEX

1. INFORMATION
2. CODE
3. OUTPUT / GAME IMAGES
4. BIBLIOGRAPHY

INFORMATION

This rebuild of a game presented here is named PONG and it is made using Turbo C++. The game is a Tribute to the original Pong game by the company Atari.

It has a loading screen at first and then takes the user/player to the home screen from where the user can enter the game or read the instructions of the game

Game:

A ping pong ball is on the screen at the start of the game, which then bounces to and from bounce pads placed parallel to each other.

A bounce pad moves in the opposite direction of the other bounce pad.

The objective of the game is to prevent the ball from touching the side borders of the screen by bouncing it away using the bounce pads.

Every instance in which a bounce pad is used, increases the score of the player .The user can play with their friends one by one to see who gets the highest score. The highest scorer wins.

**MAY THE BEST PLAYER WIN.**

CODE

#include<iostream.h>

#include<conio.h>

#include<dos.h>

#include<stdlib.h>

loading()

{ //LOADING SCREEN

textcolor(YELLOW);

\_setcursortype(\_NOCURSOR);

clrscr();

int alan;

for(int g=0;g<=10;g++)

{

alan=((rand()%12)+2)\*100;

sound(alan);

delay(100);

nosound();

}

gotoxy(25,10);

cout<<"Wall-E Production Presents.."<<endl;

delay(500);

gotoxy(34,12);

cout<<" ÉÍ» ";

gotoxy(34,13);

cout<<" ÈÑ¼ ";

gotoxy(34,14);

cout<<" ÖÂ· ";

gotoxy(34,15);

cout<<" Ú×Å×¿ ";

gotoxy(34,16);

cout<<" ³ÓÁ½³ ";

gotoxy(34,17);

cout<<" ÜÜÜÜÜ ";

delay(1000);

gotoxy(15,20);

cout<<"loading:";

for(int o=1;o<=40;o++)

{gotoxy(1,(70+o));

cout<<"±";

delay(50);

}

}

void main()

{

loading();

int a,r,c,h,as,xdir=0,ydir=0,y1=10,y2=10,score1=0,ballx,bally,j=0,i=0,fortnite;

clrscr();

textcolor(CYAN);

cout<<endl<<endl<<endl<<endl;

for(i=1;i<=80;i++)

cout<<"Í"; //MAIN MENU

cout<<endl;

delay(100);

cout<<" "<<" ====== - - - ++++++++++ "<<" "<<endl;

delay(100);

cout<<"º"<<" ° ° - - - + + "<<"º"<<endl;

delay(100);

cout<<"º"<<" ° ° ###### - - - + + "<<"º"<<endl;

delay(100);

cout<<"º"<<" ====== ###### - - - + "<<"º"<<endl;

delay(100);

cout<<"º"<<" ° ###### - - - + +++++ "<<"º"<<endl;

delay(100);

cout<<"º"<<" ° - - - + + "<<"º"<<endl;

delay(100);

cout<<" "<<" ° - -- ++++++++++ "<<" "<<endl;

delay(100);

for(i=1;i<=80;i++)

cout<<"Í";cout<<endl<<endl;

cout<<" By: Archas S.";

cout<<endl<<endl<<endl;

cout<<"For instructions Press i:-"<<endl;

cout<<"Press any other key to continue:-";

fortnite=getch();

if(fortnite=='i'||fortnite=='I')

{

clrscr();

cout<<endl<<" INSTRUCTIONS "<<endl;

cout<<endl<<endl;

cout<<" 1. To move Right bouncepad up press key (w)"<<endl;

cout<<" 2. To move Right bouncepad down press key (s)"<<endl;

cout<<" 3. To exit while playing the game (x)"<<endl;

cout<<endl<<"NOTE:- Moving the right bouncepad up or down will automatically cause the left "<<endl<<"bounceped move down or up respectively.";

cout<<endl<<endl<<endl;

cout<<"Press any key to Continue:-"<<endl;

getch();

}

clrscr();

textbackground(BLACK);

\_setcursortype(\_NOCURSOR);

randomize();

as=random(1); //BALL INITIALIZER

if(as==1)

{ballx=15;

bally=13;

xdir=+1;}

else

{ballx=65;

bally=15;

xdir=-1;}

while(!kbhit())

{

changed:

gotoxy(5,5);

cout<<"score:-"<<score1;

if(bally==23||bally==2) //TOP AND BOTTOM WALL

{ydir=ydir\*(-1);

sound(200);

delay(1);

nosound();

}

gotoxy(ballx,bally);

cout<<"þ";

for(int l=1;l<=77;l++) //BORDER

{gotoxy(l,1);

cout<<char(220);

gotoxy(l,24);

cout<<char(220);

}

for(int m=1;m<=24;m++)

{gotoxy(1,m);

cout<<char(16);

gotoxy(80,m);

cout<<char(17);

if(ballx==1&&bally==m) //GAME OVER

{delay(100);

sound(100);

delay(100);

nosound();

delay(100);

clrscr();

gotoxy(39,12);

cout<<"OOPS!,You're Out!!";

delay(1000);

clrscr();

gotoxy(39,12);

cout<<"score:- "<<score1;

delay(2000);

goto exit;

}

if(ballx==80&&bally==m) //GAME OVER

{delay(100);

sound(100);

delay(100);

nosound();

delay(100);

clrscr();

gotoxy(39,12);

cout<<"OOPS,You're Out!!";

delay(1000);

clrscr();

gotoxy(39,12);

cout<<endl<<"score:- "<<score1;

delay(2000);

goto exit;

}

}

if(xdir==+1) //ball directive algorithm

{ballx+=1;

if(ydir==+1)

bally++;

if(ydir==-1)

bally--;

if(ydir==0)

bally=bally;

}

if(xdir==-1) //ball directive algorithm

{ballx=ballx-1;

if(ydir==+1)

bally++;

if(ydir==-1)

bally--;

if(ydir==0)

bally=bally;

}

for(i=0;i<=5;i++)

{

if(y2<2||y1>=18) //bouncepad limiter

{y2++;

y1--;

}

if(y2>18||y1<=2)

{y2--;

y1++;

}

gotoxy(65,y1+i);

cout<<char(221);

if(ballx==65)

{ //on hit

if(bally==(y1+i))

{sound(200);

delay(20);

nosound();

score1+=1;

xdir=-1;

if(j==3)

ydir=0;

if(j<3)

ydir=+1;

if(j>3)

ydir=-1; }

}

}

for(j=0;j<=5;j++) // bouncepad maker

{

gotoxy(15,y2+j);

cout<<char(221);

if(ballx==15)

{

//on hit

if(bally==(y2+j))

{sound(200);

delay(20);

nosound();

score1+=1;

xdir=+1;

if(j==3)

ydir=0;

if(j==1||j==2)

ydir=+1;

if(j==4||j==5)

ydir=-1;

}

}

}

delay(90);

clrscr();

} //controller

a=getch();

if(a=='w'||a=='W') //up

{y1--;y2++;}

if(a=='s'||a=='S') //down

{y1++;y2--;}

if(a=='x'||a=='X')

goto exit;

goto changed;

exit:

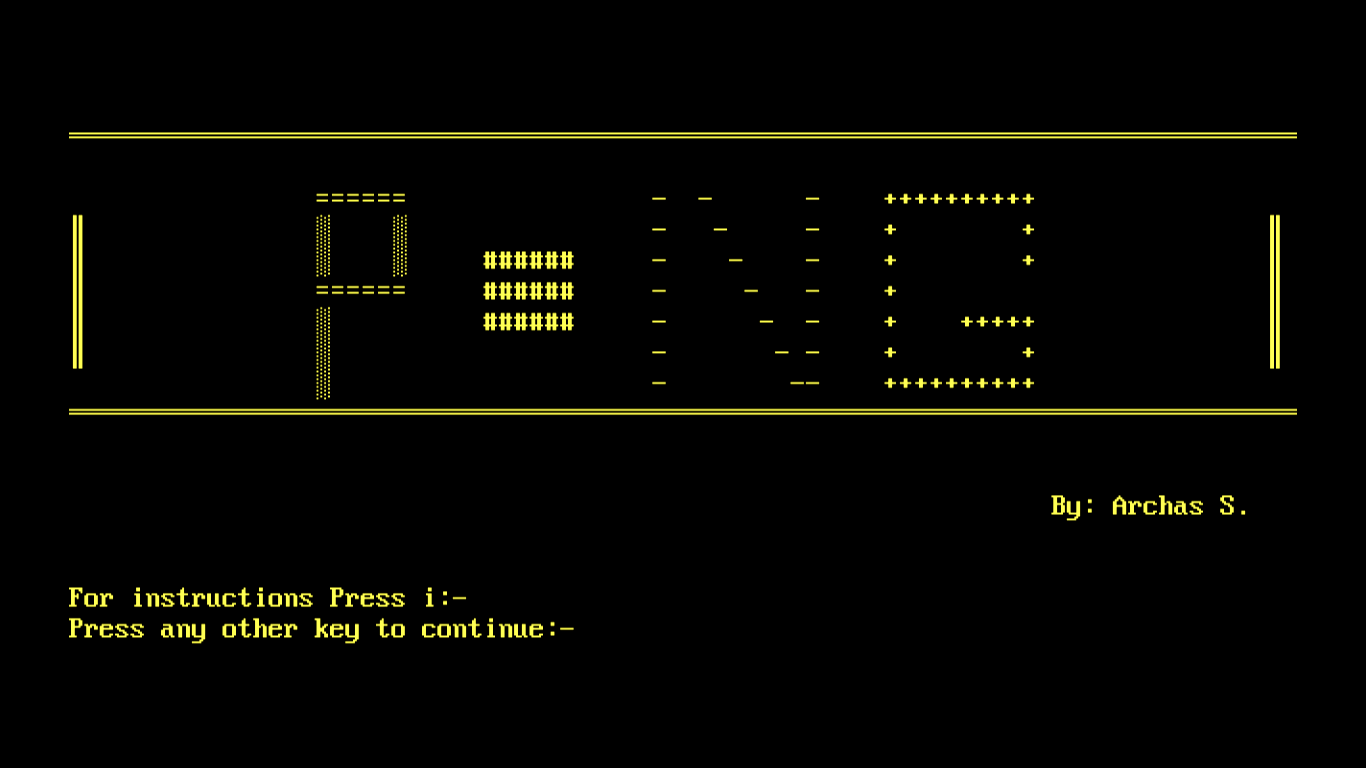
textcolor(WHITE);

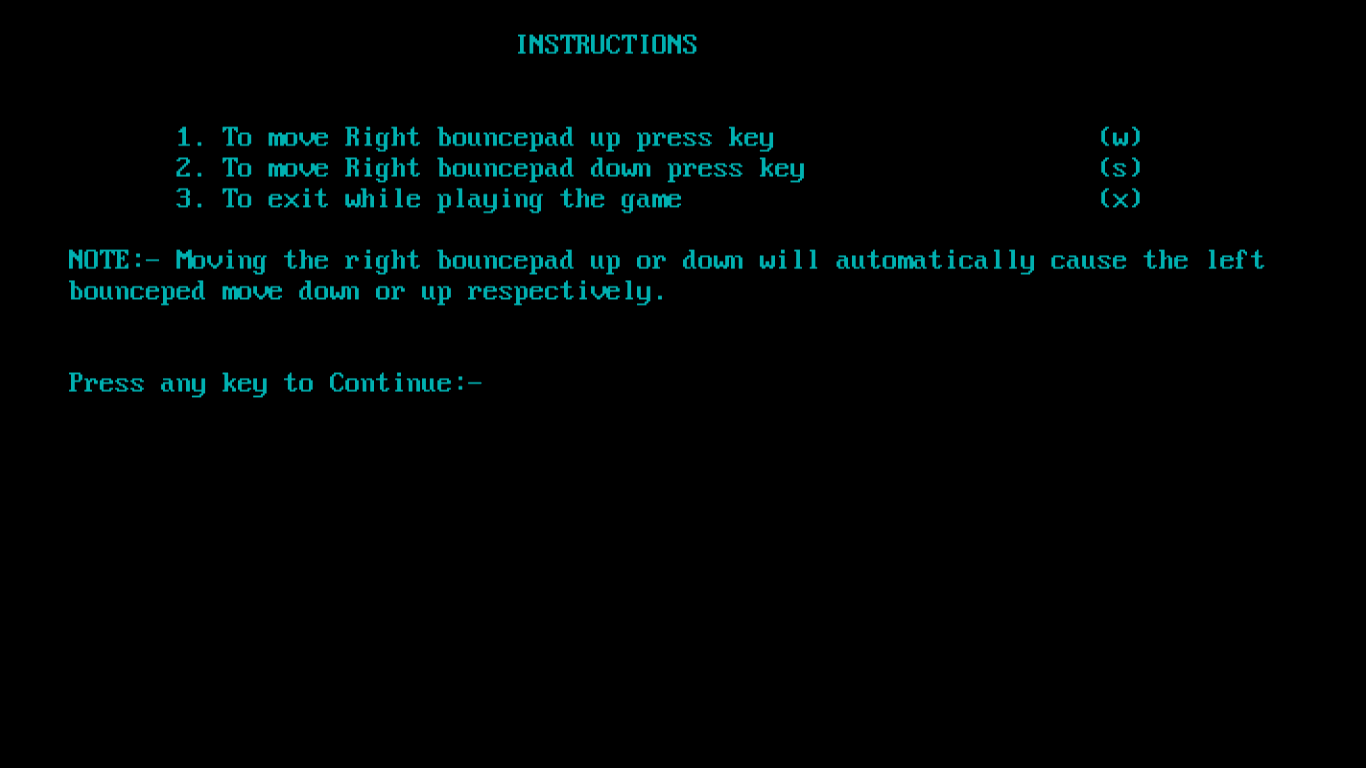
} //END OF CODE

OUTPUT/ GAME

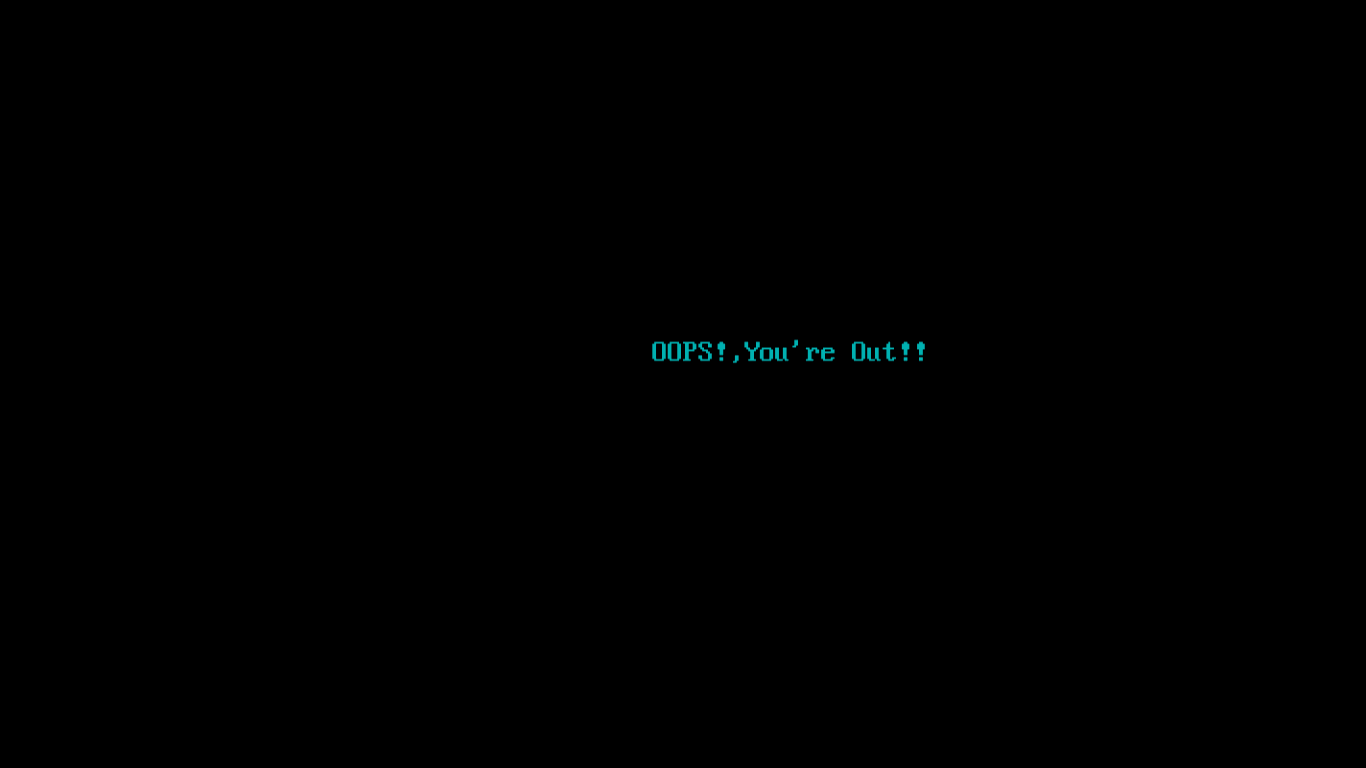
IMAGES

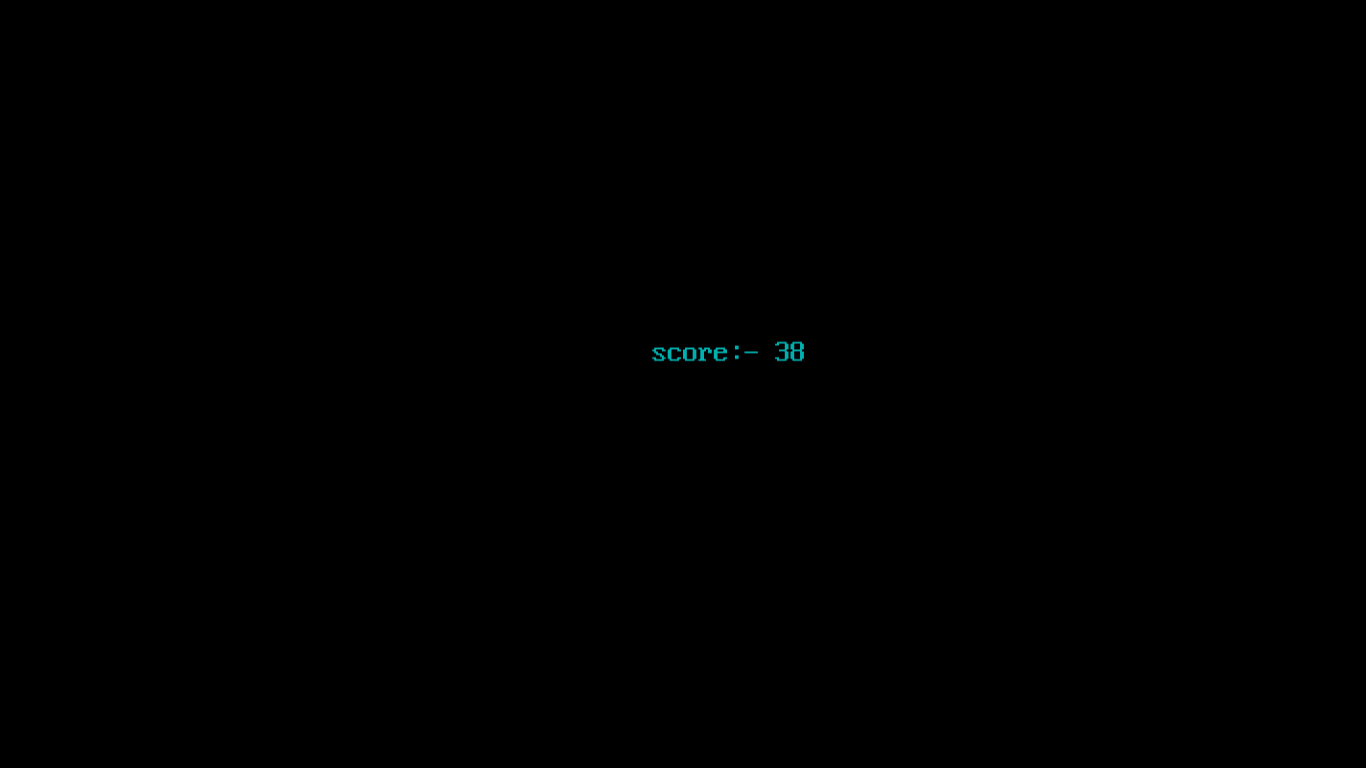












BIBLIOGRAPHY

* SUMITA ARORA CLASS XI BOOK
* TURBO C++ HELP DESK
* ATARI GAME COMPANY