

Software Project Report

PONG

Edition 6.8

Submitted to :

**Dr.Md. Asif Hossain Khan
Associate Professor, CSEDU**

**Anna Fariha
Lecturer, CSEDU**

Submitted by :

**Md. Sharifuzzaman
Roll : 06**

**Md. Fahim Arefin
Roll : 08**

Course Code : CSE 1211

1st year 2nd semester

Session : 2013-14

**Department of Computer Science and Engineering
University of Dhaka**

Table of Contents

Introduction.....	01
Game Basics.....	01
Game Outline.....	01
Main Features and Flow charts.....	02-05
Special Features.....	06
Graphical User Interface.....	06-10
Library Headers and Functions.....,	11-16
User Defined Headers and functions.....	16-18
Source Code (Main).....	19,20
Source Code (gamefunc.h).....	21-29
Source Code (menufunc.h).....	30-39
Source Code (scorefunc.h).....	40-48
Limitations, conclusion and references.....	49

Pong

Edition 6.8

Introduction:

The project is directly inspired by the legendary double player game - "Pong", we selected it because we wanted to create a simple multiplayer game as our learning basis of Borland Graphics.

Pong is one of the first computer games created, this simple "tennis like" game features two paddles and a ball, the goal is to defeat your opponent by being the first one to gain 10 point, a player gets a point once the opponent misses a ball. The game can be played with two human players. The game was originally developed by Allan Alcorn and released in 1972 by Atari corporations.

Game Basics:

Pong is a two-dimensional sports game that simulates table tennis. The player controls an in-game paddle by moving it vertically across the left side of the screen and can compete against another player controlling a second paddle on the opposing side. Players use the paddles to hit a ball back and forth. The aim is for each player to reach user defined game points before the opponent. Points are earned when the opposition fails to return the ball.

Game Outline :

- Two players control the game by moving the bats vertically on both sides.
- Player 1 uses mouse to move the right hand side bat
- Player 2 uses keyboard for moving the bat on left hand side.
- Players use the bats to hit a ball back and forth.
- Points are earned when one fails to return the ball to the other.
- If a winner has beaten one of the top 10 records, the winner enters his/her name.

Main Features:

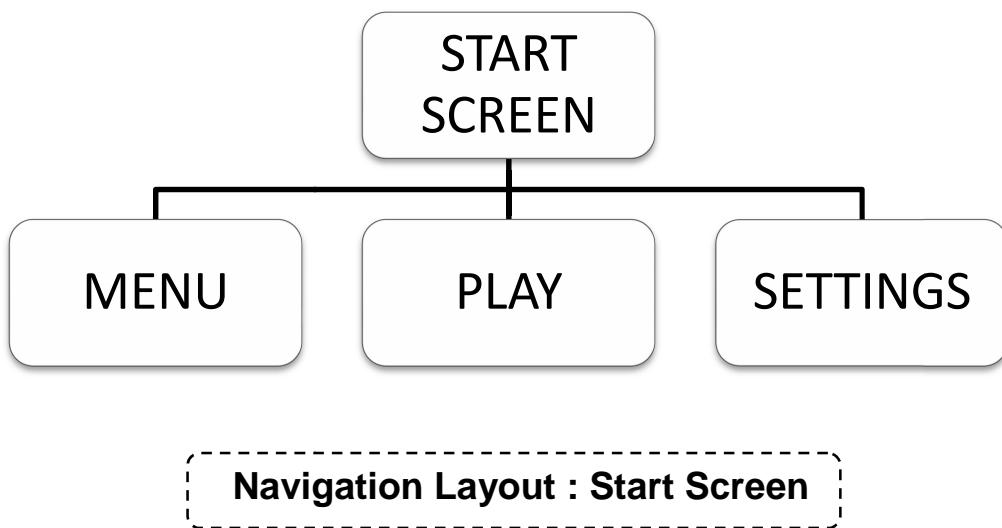
Start Page -

Start page of the game contains three rectangular boxes indicating menu , play and settings - which represent three options.

a) Menu: Menu contains several sub-menus.

b) Play : 'Play' button starts the game in Game Point Mode.

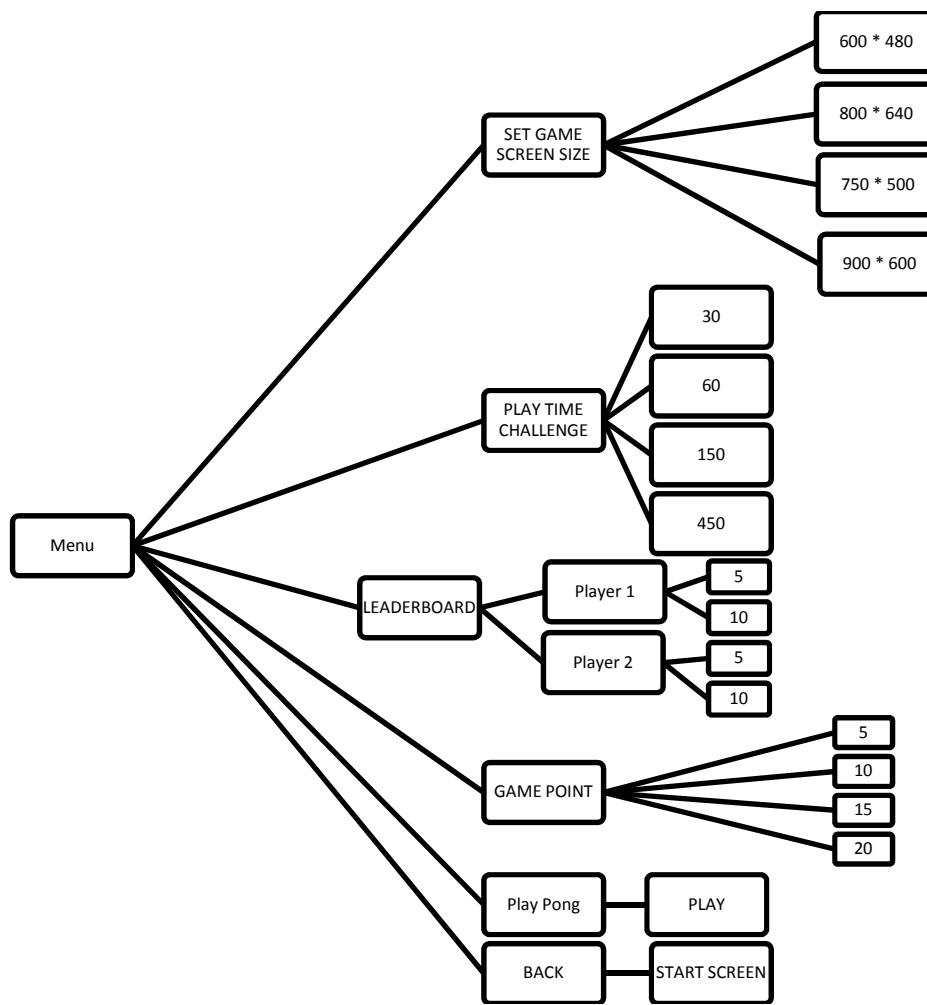
c) Settings: Lets the players change certain settings of the game .



a. Menu:

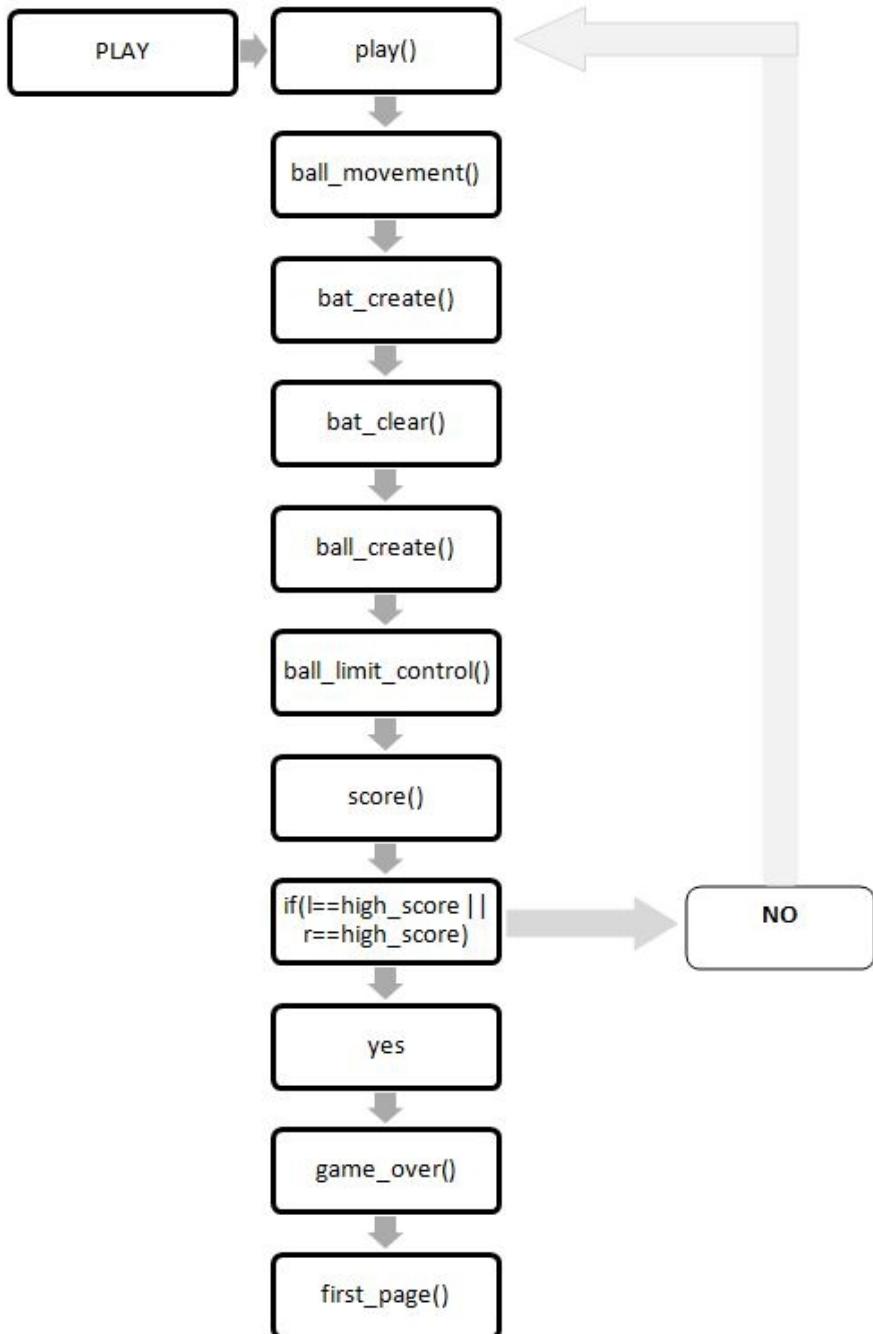
- **Set Game Screen Size :** Players can choose any game screen size of 600*480, 800*640, 750*500 or 900*600 using this option.
- **Play Time Challenge:** A time interval of 30, 60, 150 or 450 seconds can be set using this button. Then the game starts in Time Challenge Mode".

- **Leader Board** : In leader board users can view a rank list of 5 or 10 winning records for both players.
- **Game Point** : This button lets the players choose a "Game Point limit" from 5,10,15 and 20.
- **Play Pong** : The game starts after clicking on this button.
- **Back** : It is for returning to the first page.



Navigation Layout : Menu

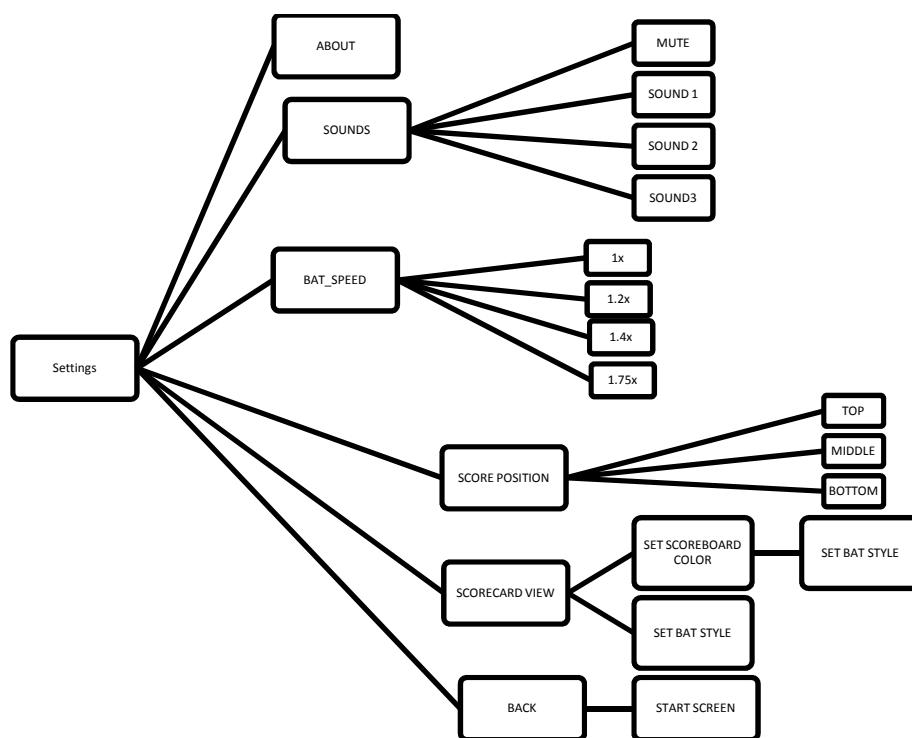
b. Play : The game starts when the user selects this option in start page



Functional Flow chart of Play

c.Settings :

- **About :** Displays a page of trivia about the game
- **Sounds :** We generated a beep sound when a player misses the ball. The user can choose from four options - Mute (No Sound) , Sound 1 (frequency 55 Hz, 80 Hz) , Sound 2 (frequency 85 Hz, 110 Hz), Sound 3 (frequency 100 Hz, 125 Hz) , where the first frequency is generated when the right player misses and the second frequency is generated when the left player misses.
- **Bat Speed :** The players can change the default speed of the bat to 1.2x , 1.4x and 1.75x times the default speed.
- **Set Score Position :** The position of the score can be changed to top, middle or bottom.
- **Scorecard View :** The user can change the color of the scorecard background and the bat style from this option.
- **Back :** This button is for returning to first page



Navigation Layout : Settings

Special Features :

- **Leader board:** The leader board is sorted according to the winning difference. if the difference is equal, then the record where the winner scored more point gets the higher rank.
- **Pause :** By pressing "Esc", players can pause the game. Then they can choose between "Resume" , "Quit" and " Settings" . So they change the game settings while they are playing a game. This menu is keyboard based because the keyboard player needs more time to re-act after the resume.
- **Encrypted Leader Board file :** We encrypted the leader board file by creating it with ".fs" extension, so that a user can't simply open it.
- **Screen Adjustment :** All the menu options are auto adjusted to the screen size of the monitor and the game is always initialized to full screen as we can't resize the window size afterwards in bgi.

Graphical User interface:

Figure 1. Start Page

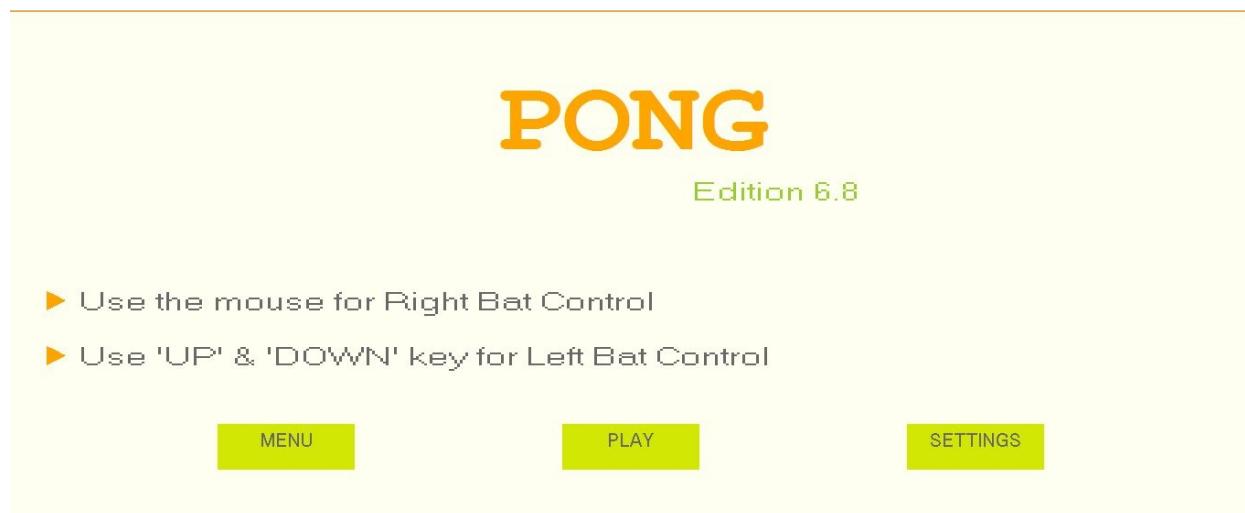


Figure 2. Menu



Figure 3. Settings

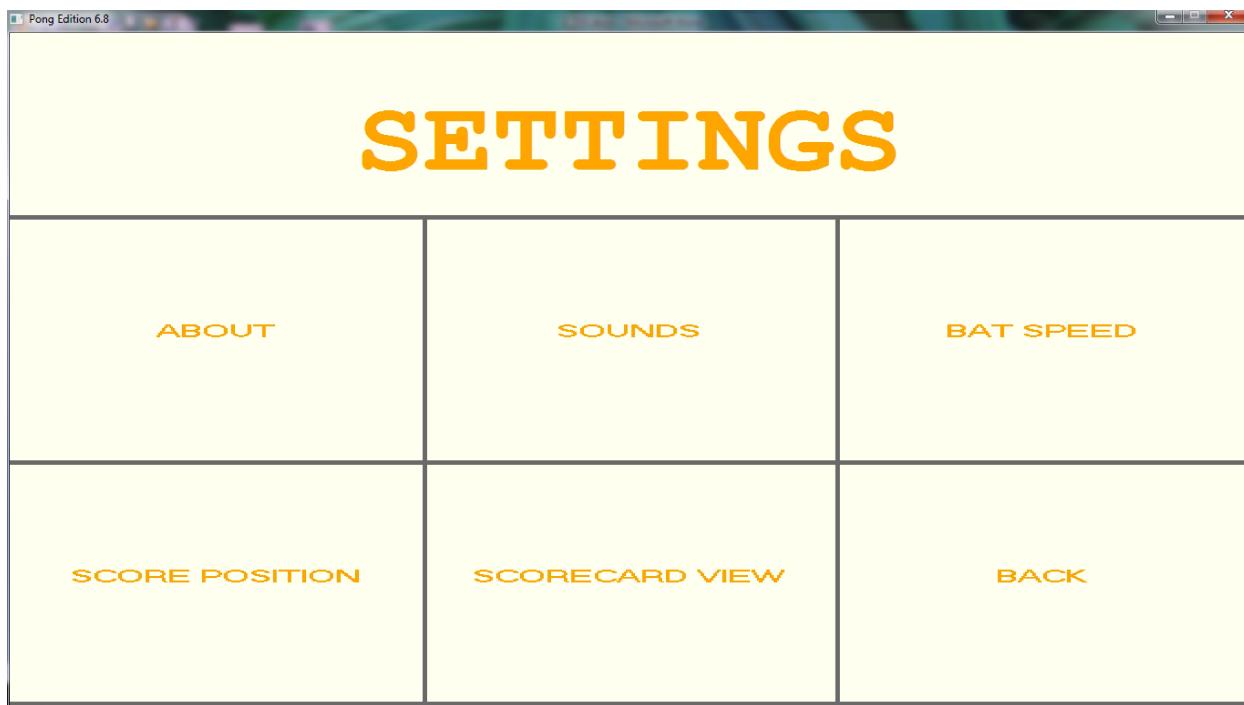


Figure 4 : Submenu (Settings>Sound)



Figure 5 . Scorecard View

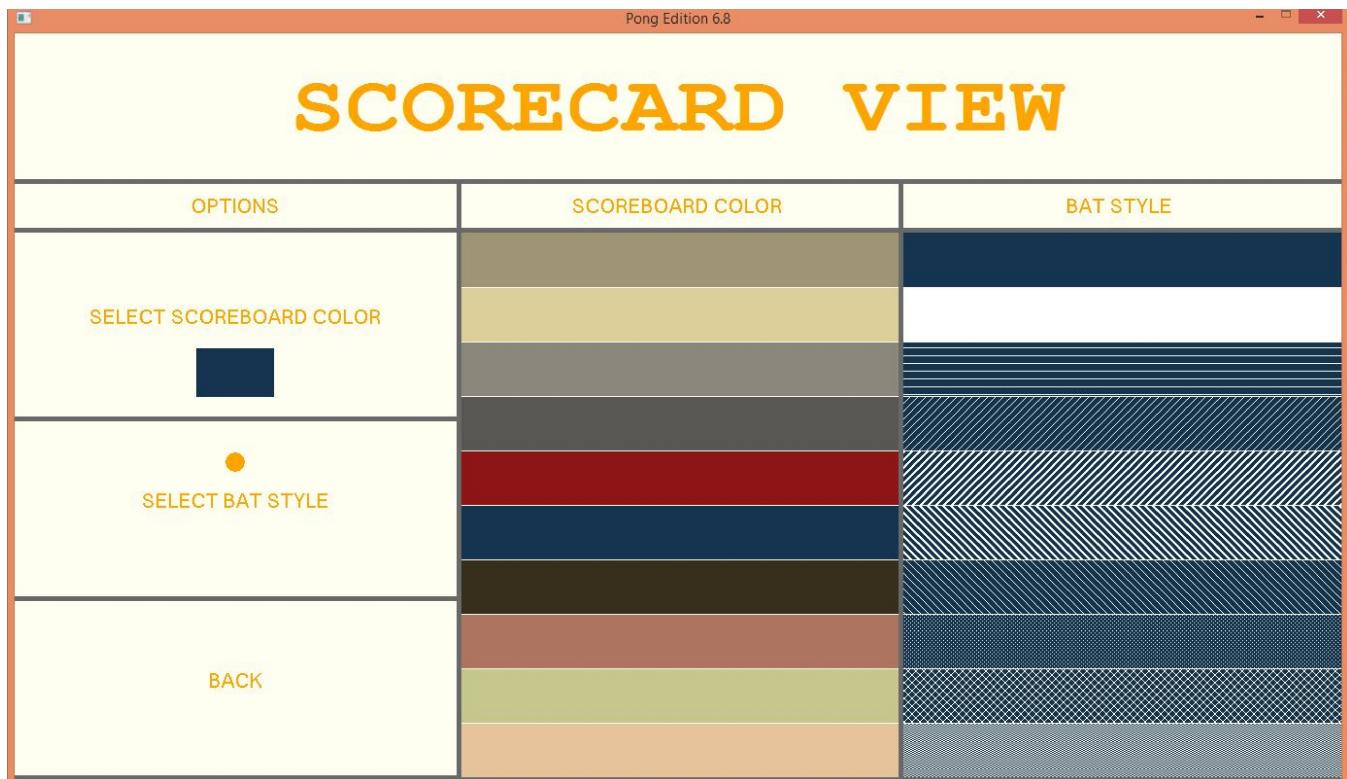


Figure 6. Leaderboard



Figure 7. Gameplay (Pause Screen ; Score_position - Middle)



Figure 8. Gameplay (Score_position - Bottom, Game Screen Size 800 * 640)

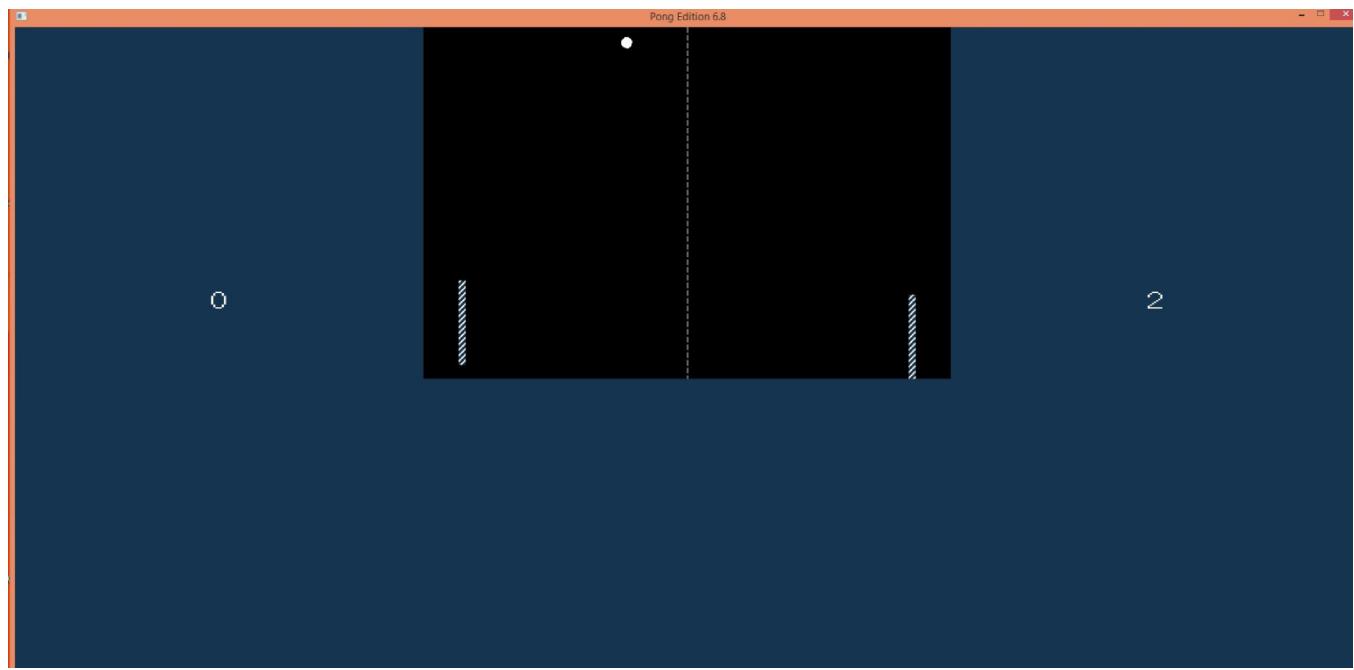


Figure 9. Gameplay (Time Challenge , Score Position - Top, Game Screen Size 900 * 600)



Header Files in the Project :

- Header file stdio.h :

- **printf()** : prints formatted string to a file,it has been used for creating the leaderboard inside the file.

Example of implementation:

```
printf(score_list,"%d %d %s\n",list[i].p2,list[i].p1,list[i].name);
```

- **fscanf()** : reads formatted string from a file,it has been used to read previous record from the score log file.

Example of implementation:

```
fscanf(score_list,"%d %d %[^\n]s",&list[i].p2,&list[i].p1,list[i].name);
```

- **sprintf()** : Writes formatted data to a string, it has been used to print score on graphics window.

Example of implementation: sprintf(left_score,"%d",l);

- **fopen()** : opens the score record file, it has been used to open file for leaderboard.

Example of implementation: File * score_list=fopen("score log.fs","r");

- **fclose()** : closes the previously opened file, it has been used to close file after checking record in read mode.

Example of implementation: fclose(score_list);

- **feof():** Checks if the file has reached the end-of-File, used to determine end of file in score checks.

Example of implementation: if(feof(score_list)) break;

- **fseek()** : moves the file pointer to the beginning of the file, used to read from the start of the file

Example of implementation: fseek(score_list,0L,0);

- **Header file time.h :**

- **time(&x_t)** : It returns the current time and assigns the value to time variable x_t ,it has been used for time challenge

Example of implementation: time(&end_t);

- **difftime(a,b)** : It returns the time interval between a and b, it is actually subtraction, used for creating time counter

Example of implementation: difftime(end_t,start_t);

- **Header file windows.h :**

- **Beep(x,y)** : Beeps a sound of x Hz frequency for y milliseconds. It has been used for generating sound when a player misses the ball.

Example of implementation: Beep(SOUND,500);

- **Header file graphics.h :**

- **bar()** : draws a rectangle and fills it with current fill pattern .It has been used to create the menu buttons.

Example of implementation: bar(0,0,getmaxwidth(), getmaxheight());

- **cleardevice()** : Refreshes the graphical screen. It has been used before creating a new screen after switching from the menu.

Example of implementation:

cleardevice();

leaderboard();

- **clearmouseclick()** : After calling clearmouseclick, for a particular kind of event, the ismouseclick will return false for that kind of event until another such event occurs. It has been used for processing menu click operations properly.
Example of implementation: clearmouseclick(WM_LBUTTONDOWN);
- **COLOR ()**: Returns an integer value for rgb color. It has been used to create rgb colors in the graphics window.
Example of implementation: int bk = COLOR(255,165,50)
- **delay()** : The function pauses the computation for the the specified number of ms. It has been used to keep the displayed contents for a specified time interval.
Example of implementation: delay(100);
- **fillellipse()** : Draws an ellipse using (x,y) as a center point and xradius and yradius as the horizontal and vertical axes, and fills it with the current fill color and fill pattern. It has been used to create the Menu bullets in the about section.
Example of implementation :
fillellipse(screen_left/2,215+(menu_ymid/2),10,10);
- **fillpoly()** : fillpoly draws the outline of a polygon with numpoints points in the current line style and color (just as drawpoly does), then fills the polygon using the current fill pattern and fill color. It has been used to draw the triangular bullets in the about section
Example of implementation : fillpoly(4,points);
- **getch()**:The function reads one character from the keyboard and returns its ASCII value (without waiting for a return key). It has been used to take user input from the keyboard throughout the game.
Example of implementation : input_key = getch();

- **getmaxheight()** : The function returns the maximum height that will fit on the screen when creating a window with initwindow. It has been used to create a full screen window.
Example of implementation: bar(0,0,getmaxwidth(), getmaxheight());
- **getmaxwidth()** : The function returns the maximum width that will fit on the screen when creating a window with initwindow. It has been used to create a full screen window.
Example of implementation: bar(0,0,getmaxwidth(), getmaxheight());
- **getmouseclick()** : This function sets x and y to the pixel coordinates of an unprocessed event of the specified kind. It has been used for processing mouseclick on the menu buttons.
Example of implementation:
getmouseclick(WM_LBUTTONDOWN,x,y);
- **getpixel(x,y)** :It returns the color of the (x,y) pixel, It has been used to determine the selected color in the "Set Background Color" menu.
Example of implementation: getpixel(x,y)
- **ismouseclick()** :_This function returns true if there is an unprocessed mouse event of the specified kind. It has been used to determine mouseclicks in the menu screen.
Example of implementation:
while(!ismouseclick(WM_LBUTTONDOWN));
- **kbhit()** : The function returns true (non-zero) if there is a character in the input buffer ready to read. Otherwise it returns false. In order to work, the user must click in the graphics window. It has been used to check for user input for the game.
Example of implementation: while(!kbhit());

- **outtextxy()**: outtextxy displays a text string in the graphics windows at the given position (x, y), using the current justification settings and the current font, direction, and size. It has been used to print texts on screen.
Example of implementation: outtextxy(screen_left,200+menu_ymid/2, "Pong(1972) by Atari is considered the Mother of Arcade Games");
- **readimagefile()** : This function reads only JPEG and BITMAP image. This has been used for the pause screen.
Example of implementation:
readimagefile("Paused.bmp",0,0,screen_x,screen_y);
- **setbkcolor ()**: setbkcolor sets the background to the color specified by color. It has been used for setting background color of texts
Example of implementation: setbkcolor(GREY);
- **setcolor()** : setcolor sets the current drawing color to color, which can range from 0 to getmaxcolor. It has been used to set text colors.
Example of implementation: setcolor(COLOR(255,165,50));
- **setfillstyle()** : setfillstyle sets the current fill pattern and fill color. It has been used to color the score card background.
Example of implementation:
setfillstyle(SOLID_FILL,COLOR(255,165,0));
- **settextstyle()** : settextstyle sets the text font, the direction in which text is displayed, and the size of the characters. It has been used for setting the text style before printing it on screen.
Example of implementation: settextstyle(FONT,HORIZ_DIR,3);
- **settextjustify()** : Text output after a call to settextjustify is justified around the current position (CP) horizontally and vertically, as specified. It has been used to justify the output texts.
Example of implementation: settextjustify(LEFT_TEXT, TOP_TEXT);

- **textheight()**: The graphics function textheight takes the current font size and multiplication factor, and determines the height of textstring in pixels. It has been used to draw bars according to text size
Example of implementation: textheight("BACK");
- **textwidth()** : The graphics function textwidth takes the string length, current font size, and multiplication factor, and determines the width of textstring in pixels. It has been used to draw bars according to text size
Example of implementation: textwidth("BACK");
- **writeimagefile()** : The function saves a portion of the active page in a BMP file. The filename must end in "BMP" or ".bmp", or it may be NULL. It has been used for pause screen
Example of implementation:
writeimagefile("Paused.bmp",0,0,screen_x,screen_y);

- **Custom Header Files:**

- gameplay.h
- menufunc.h
- scorefunc.h

Header file gameplay.h :

- **void ball_clear()** : Erases the ball before drawing a new one
- **void ball_create()** : Creates the ball
- **void ball_limit_control()** : Checks and maintains the ball inside the playing area
- **void ball_movement()** : Moves the ball along the defined direction
- **void bat_clear()** : Erases the bat on screen

- **void bat_create()** : Draws a new bat on screen
- **void game_over()** : Detects when a game is over and writes the scores accordingly
- **void pause()** : Pauses the game. when Esc is pressed and then creates a keyboard based menu.
- **void play()** : Controls the ball and bat movement and runs the main game play.
- **void start_pong()** : Creates the logo of the game on screen
- **int timel()** : returns 1 when 1 second has elapsed since last check

Example of implementation: if(timel()==1) game_over();

Header file menufunc.h :

- **void about()** : Prints the "about" page
- **void background()** : Draws the background on screen
- **void create_menu()** : Creates the Menu
- **void draw_triangle(int x,int y,int z)** : Used as a bullet icon in the about menu, where (x,y) is the rightmost vertex of the triangle and z is the triangle length

Example of implementation:

drawtriangle(screen_left/2+10,212 +menu_ymid*2,3);

- **void first_page()** : Creates the Start Screen of the Game

- **void menu()** : Takes the mouseclick from the user and sends it to select_menu()
- **void select_menu(int x , int y)** : Gets the co-ordinate of the clicked point from the menu() function and executes the corresponding command. gets the mouseclick co-ordinate(x,y) from the menu() function.

Example of implementation: select_menu(x,y);

- **void settings_menu()**: creates and implements the Settings menu

Header file scorefunc.h :

- **void leaderboard(int n, int p)** : Prints the leader board , here n is the number of player and p is the player number.

Example of implementation: leaderboard(n,p);

- **void score()** : Keeps the score updated

- **void score_card()** : Creates the menu to select screen background color & bat style

- **void score_check(int n, int p)** : sorts the records and creates the leader board, here n is the number of records and p is the player number

Example of implementation: score_check(n,p);

- **void score_menu()** : lets the user create a leader board customized to his preferences

- **void user_input(int us)** : Takes the username as input when a user wins a game, here us is the player number.

Example of implementation: user_input(us);

Source.cpp:

```

#include "graphics.h"
#include<stdio.h>
#include<time.h> /* For Time count */
#include<windows.h> /* for audible sound */

void about();
void background();
void ball_clear();
void ball_create();
void ball_limit_control();
void ball_movement();
void bat_clear();
void bat_create();
void create_menu();
void draw_triangle(int x,int y,int z); // x,y is the point where the sharp end
will be, z+10 is the size of the triangle
void first_page();
void game_over();
void menu();
void pause();
void play();
void score();
void score_card();
void score_check(int n, int p);
void score_menu();
void select_menu(int x , int y);
void settings_menu();
void start_pong();
int timel();
void user_input(int us);
void leaderboard(int n, int p);

/* Screen Controls */

#define screen_top 0 /*top line of the screen.it will always be zero, otherwise we
will have to change all co-ordinates*/
int screen_left=75; /*start of black screen*/
int screen_right=675; /*end of black screen*/
int screen_bottom=480; /*bottom of black screen*/
int screen_end=screen_right+screen_left; /* end line of screen...value of
screen_end should be a multiple of 6 */
int screen_mid=(screen_end)/2; /*middle of the screen, where the net is */

/* GamePlay Settings */

int high_score=10;//set the highest score
#define D_E 15
#define D_H 10
int SPEED=15; // serve speed
int mode=1; // Mode 1 = Tennis, Mode 2 = Time Challenge
int TIME= 30,x=0,timez=0,timey=1; // Time limit for time challenge, timey - timez
=1 means one second has elapsed
time_t start_t, end_t;
char timex[10];

int bat_speed=90; /*speed of the BAT can be changed*/

```

```

#define bat_size 120 /*size of the BAT can be changed*/
#define bat_width 10
#define bat_pos_left screen_left+50
#define bat_pos_right screen_right-60
int ball_y=screen_bottom/2,ball_x=screen_mid,input_key,left_baty,right_baty, f=1,g,
l,r,g_o=0;
int ball_radius=8;
int prex=ball_y+20,prey=ball_y+20;
int dev=SPEED,det,de,i,ball_speed_x=7,ball_speed_y=7;
int bat_pause_y= 0,pausey=0,quit=0,used=0,j[3],elim=0,m=0;

/*Menu */
#define FONT SANS_SERIF_FONT
#define FONT_SIZE 4
#define GREY COLOR(105,105,105)
#define MENU_C COLOR(240,255,240)
#define score_col WHITE
#define ball_col WHITE
int bat_col=WHITE;
int bat_fill=1;
int menu_x1, menu_x2,menu_x3,menu_y1,menu_y2,menu_ymid,menu_ytop;
int SOUND=55; // Sets the sounds when the ball misses the bat
int score_pos=2,score_back=GREY;
int midx,midy,flag,screen_x=getmaxwidth(),screen_y;
int p1[550],p2[550],dif[5][550]; /* Leaderboard array */
char left_score[3],right_score[3],choice,control_l,l_u,l_d;
int left_bat_up, left_bat_down, time_flag=0; // Used for Time-Trial Mode
char top[4][50][100]; /* Generates the Leaderboard */
char username[100];
FILE *score_list=fopen("score log.fs", "a");

#include"menufunc.h"
#include"scorefunc.h"
#include"gameplay.h"

int main( )
{
    l=0,r=0;
    initwindow(getmaxwidth(), getmaxheight(), "Pong Edition 6.8");
    screen_x=get maxx();
    screen_y=get maxy();
    midx = screen_x/ 2;
    midy = screen_y / 2;
    menu_x1=(int)(screen_x/3);
    menu_x2=(int)((2*screen_x)/3);
    menu_x3=screen_x;
    menu_y1=(int)(200+(screen_y-200)/2);
    menu_y2=screen_y;
    menu_ymid=(int) ((screen_y-200)/4);
    menu_ytop=200+ (screen_y-200)/8;
    while(1)
    {
        play();
    }
    system( "pause" );
    return 0;
}

```

gamefunc.h:

```

void ball_create()
{
    ball_clear();
    setcolor(ball_col);
    setfillstyle(SOLID_FILL,ball_col);
    fillellipse(ball_x,ball_y,ball_radius,ball_radius);
    prex=ball_x;
    prey=ball_y;
}

void ball_movement()
{
    if(f==0)                                // if f=0, BALL moves leftward and vice-versa
        ball_x-=ball_speed_x;                // the values of ball_x & ball_y controls
the SPEED & DIRECTION of the BALL along x & y-axis respectively
    else if(f==1)
        ball_x+=ball_speed_x;
    if(g==0)                                //if g=0, BALL moves upward and vice-versa
        ball_y-=ball_speed_y;
    else if(g==1)
        ball_y+=ball_speed_y;
    ball_create();
}

void ball_clear()
{
    setcolor(BLACK);
    setfillstyle(SOLID_FILL,BLACK);
    fillellipse(prex,prey,ball_radius,ball_radius);
    setcolor(score_back);
    setfillstyle(SOLID_FILL,score_back);
    bar(0,screen_bottom,screen_x,screen_bottom+50);
}

void ball_limit_control()
{
    if(ball_x>= bat_pos_left && ball_x -ball_radius <= bat_pos_left+bat_width &&
ball_y + ball_radius >= left_baty && ball_y - ball_radius <= left_baty + bat_size)
    {
        if(ball_x>= bat_pos_left+bat_width/2 && ball_x -ball_radius <=
bat_pos_left+bat_width)
            f=1;
        if(ball_y-ball_radius<=left_baty && g==1) //ball is above the left bat
            g=0;
        else if(ball_y+ball_radius>= left_baty + bat_size && g==0)//ball
is below the left bat
            g=1;
    }
}

```

```

        if(ball_y-ball_radius<=left_baty      ||      ball_y+ball_radius>=
left_baty + bat_size) //corner case
    {
        if(dev>8)
            dev-=2;
        ball_speed_x=10;
        ball_speed_y=6;
    }
    else if(ball_y-ball_radius>left_baty && ball_y+ball_radius <
left_baty + bat_size) //
    {
        ball_speed_x=9;
        ball_speed_y=9;
    }
}

else if(ball_x+ball_radius>= bat_pos_right && ball_x<=
bat_pos_right+bat_width && ball_y + ball_radius >= right_baty && ball_y -
ball_radius <= right_baty + bat_size)
{
    if(ball_x<= bat_pos_right+bat_width/2 && ball_x +ball_radius >=
bat_pos_right)
        f=0;           //ball touches the left bat & direction of the
ball is changed to rightward.
    if(ball_y-ball_radius<=right_baty && g==1) //ball is above the
left bat
        g=0;
    else if(ball_y+ball_radius>= right_baty + bat_size && g==0)//ball
is below the left bat
        g=1;

    if(ball_y-ball_radius<=right_baty      ||      ball_y+ball_radius>=
right_baty + bat_size) //corner case
    {
        if(dev>8)
            dev-=2;
        ball_speed_x=10;
        ball_speed_y=6;
    }
    else if(ball_y-ball_radius>right_baty && ball_y+ball_radius <
right_baty + bat_size) //
    {
        ball_speed_x=9;
        ball_speed_y=9;
    }
}

if(ball_x<= bat_pos_left-20)
{
    f=1;                                // BALL misses the left bat
}

```

```

        dev=SPEED;
        ball_speed_x=10;
        ball_speed_y=10;
        ball_x=screen_mid;                                // BALL starts again
from the mid position
        r++; // one score is added to the right side
        ball_clear();
        if(SOUND!=0)
            Beep(SOUND+25,500);
        else
            delay(500);
    }
    else if(ball_x>=bat_pos_right+bat_width+20)           //similarly BALL misses the
    {                                                       right bat...
        f=1;
        dev=SPEED;
        ball_x=screen_mid;
        l++;
        ball_clear();
        if(SOUND!=0)
            Beep(SOUND,500);
        else
            delay(500);
    }

    if(ball_y-ball_radius<=screen_top)

    {
        g=1; // if BALL touches the upper bound of window, direction is changed
to downward
        ball_speed_x=8;
        ball_speed_x=7;
    }
    else if(ball_y+ball_radius>=screen_bottom)
    {
        g=0;
        ball_speed_x=8;
        ball_speed_x=5;
        setcolor(score_back);
        setfillstyle(SOLID_FILL,score_back);
        bar(0,screen_bottom,screen_end,screen_bottom+50);
    }
}

void bat_clear()
{
    setcolor(BLACK);
    setfillstyle(SOLID_FILL,BLACK);
    bar(bat_pos_left, screen_top, bat_pos_left+bat_width, left_baty);
    bar(bat_pos_left, left_baty+bat_size, bat_pos_left+bat_width, screen_bottom);
}

```

```

bar(bat_pos_right, screen_top, bat_pos_right+bat_width, right_baty);
bar(bat_pos_right, right_baty+bat_size,bat_pos_right+bat_width, screen_bottom);
}

void bat_create()
{
    while(kbhit())                                //takes input for moving Bats
    {
        setcolor(BLACK);
        setfillstyle(SOLID_FILL,BLACK);
        bar(bat_pos_left, left_baty, bat_pos_left+bat_width, left_baty+bat_size);
        input_key=getch();
        if(input_key==27)
        {
            pausey=1;
            pause();
            if(quit==1)
                return;
        }
        if(input_key==left_bat_up && left_baty-bat_size>=0)
            left_baty-=bat_speed;                      //changing
the postion of left bat
        else if(input_key==left_bat_up && left_baty-bat_size<0)
            left_baty=0;
        else                                         if(input_key==left_bat_down &&
left_baty+bat_size+bat_speed>screen_bottom)
            left_baty=screen_bottom-bat_size;
        else   if(input_key==left_bat_down && left_baty+bat_size<=screen_bottom)
//screen limit =700, so last end of bat has to be inside 580
            left_baty+=bat_speed;
        rewind( stdin );
        setcolor(bat_col);
        setfillstyle(bat_fill,bat_col);
        bar(bat_pos_left, left_baty, bat_pos_left+bat_width, left_baty+bat_size);
    }
    if(ismouseclick(WM_MOUSEMOVE))
        right_baty=mousey();
    if(right_baty>=screen_bottom-bat_size)
        right_baty=screen_bottom-bat_size;
    else if (pausey==1)
    {
        right_baty=bat_pause_y;
        pausey=0;
    }
    setcolor(bat_col);
    setfillstyle(bat_fill,bat_col);
    bar(bat_pos_left, left_baty, bat_pos_left+bat_width, left_baty+bat_size);
    bar(bat_pos_right, right_baty, bat_pos_right+bat_width, right_baty+bat_size);
}

```

```

void game_over()
{
    if(l>r)                                //limit of score
    {
        settextjustify(RIGHT_TEXT,CENTER_TEXT);
        setbkcolor(BLACK);
        setcolor(COLOR(30,144,255));
        settextstyle(FONT,HORIZ_DIR,5);
        outtextxy(screen_mid -30,300,"WIN");
    }
    else if(r>l)
    {
        settextjustify(LEFT_TEXT,CENTER_TEXT);
        setbkcolor(BLACK);
        setcolor(COLOR(30,144,255));
        settextstyle(FONT,HORIZ_DIR,5);
        outtextxy(screen_mid +30,300,"WIN");
    }
    else if(l==r)
    {
        settextjustify(CENTER_TEXT,CENTER_TEXT);
        setbkcolor(BLACK);
        setcolor(COLOR(30,144,255));
        settextstyle(FONT,HORIZ_DIR,5);
        outtextxy(screen_mid,300,"TIE");
    }
    g_o=1;
    delay(1000);                            // delay after WIN
    int us;
    if(l>r)
        us=2;
    else if(r>l)
        us=1;
    used=2;
    score_check(10,us);
    printf("%d %d\n",elim,us);
    if(elim!=1)
    {
        score_list=fopen("score log.fs","a");
        user_input(us);
        fprintf(score_list,"%d %d %s\n",l,r,username);
        l=r=0;                                // score 0-0
        SPEED=15;
        used=0;
        int bk= COLOR(255,255,240);
        settextjustify(CENTER_TEXT,CENTER_TEXT);
        setbkcolor(bk);
        setcolor(COLOR(255,165,0));
        settextstyle(BOLD_FONT,HORIZ_DIR,5);
        outtextxy(menu_x2+menu_x1/2,75,"LEADERBOARD");
        setfillstyle(SOLID_FILL,GREY);
    }
}

```

```

        bar(menu_x2,150,screen_x,155);
        bar(menu_x2,150,menu_x2+5,screen_y);
        bar(menu_x2,200,screen_x,205);
        score_check(10,us);
        leaderboard(10,us);
    }

}

void pause()
{
    setcolor(bat_col);
    setfillstyle(bat_fill,bat_col);
    bar(bat_pos_left,left_baty,bat_pos_left+bat_width,left_baty+bat_size);
    bar(bat_pos_right,right_baty,bat_pos_right+bat_width,right_baty+bat_size);
    bat_pause_y=right_baty;
    setbkcolor(score_back);
    settextjustify(CENTER_TEXT, TOP_TEXT);
    setcolor(score_col);
    settextstyle(BOLD_FONT,HORIZ_DIR,6);
    outtextxy(screen_end/2,screen_bottom+2,"GAME PAUSED");

    int color=COLOR(255,255,240);
    int d=(screen_x-450)/4;
    int h=150;
    int x,y;
    setcolor(color);
    setfillstyle(SOLID_FILL,color);
    bar(0,screen_bottom+50,screen_x,screen_bottom+100);
    setbkcolor(color);
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    setcolor(score_back);
    settextstyle(FONT,HORIZ_DIR,3);
    outtextxy(h/2,screen_bottom+80,"PRESS KEY :");
    outtextxy(d+h/2,screen_bottom+80,"QUIT (Q) ");
    outtextxy(2*d+h+h/2,screen_bottom+80,"RESUME (ESC)");
    outtextxy(3*d+2*h+h/2,screen_bottom+80,"SETTINGS (S) ");

    again:
    control_l=getch();

    if(control_l=='q' || control_l=='Q')
        control_l=1;
    else if (control_l=='s' || control_l=='S')
        control_l=2;
    else if (control_l==27)
        control_l=3;
    else
        goto again;
}

```

```

if(control_l==1)
{
    quit=1;
    return;
}
else if(control_l==2)
{
    writeimagefile("Paused.bmp",0,0,screen_x,screen_y);
    settings_menu();
    pausey=0;
    readimagefile("Paused.bmp",0,0,screen_x,screen_y);
    goto again;
}
setfillstyle(SOLID_FILL,score_back);
bar(0,screen_bottom,screen_x,screen_y);
bar(0,0,screen_left,screen_bottom);
bar(screen_right,0,screen_x,screen_bottom);
}

void play()
{
    if(screen_right-screen_left>550)
        det=D_H;
    else
        det=D_E;
    begin:
    first_page();
    delay(100);
    cleardevice();
    setbkcolor(BLACK);
    background();
    prex=screen_left+10,prey=screen_bottom+ball_radius;
    time(&start_t);
    while(1)
    {
        setcolor(WHITE);
        setlinestyle(DASHED_LINE,3,1);
        line(screen_mid,screen_top,screen_mid,screen_bottom);

        //ball_clear();

        ball_movement(); //movement of ball

        bat_create();
        if(quit==1)
            goto gamer;

        bat_clear();

        ball_create();
        delay(dev);

        ball_limit_control();

        score();
    }
}

```

```

        if(l==high_score || r==high_score)
            game_over();
            else if(mode==2)
{
    timel();
    time_t timer=end_t-start_t; // returns the time
between start and current time
    if(timel()==1) // returns 1 when time is up
    {
        game_over();
    }
    timey=timez;//previous timez
    timez=TIME-timer; // countdown timer
    sprintf(timer,"%d",timez);
    // We will print the timer once every second, so
timey-timez==1
    if(score_pos==2 && ( timey-timez==1 || timez==TIME ) )
/* score at middle,so timer at bottom */
{
    setcolor(score_back);
    setfillstyle(SOLID_FILL,score_back);

    bar(0,screen_bottom/2+75,screen_left,screen_bottom);

    bar(screen_right,screen_bottom/2+75,screen_end,screen_bottom);
    setbkcolor(score_back);
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    setcolor(COLOR(255,165,0));
    settextstyle(3,HORIZ_DIR,4);
    outtextxy(screen_left/2,screen_bottom-
100,timer);
    outtextxy(screen_end-
(screen_left/2),screen_bottom-100,timer);
}
else if( timey-timez==1 || timez==TIME ) /* score not
at middle , so timer at Middle */
{
    setcolor(score_back);
    setfillstyle(SOLID_FILL,score_back);
    bar(0,screen_bottom/2-
100,screen_left,screen_bottom/2+100);
    bar(screen_right,screen_bottom/2-
100,screen_end,screen_bottom/2+100);
    setbkcolor(score_back);
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    setcolor(COLOR(255,165,0));
    settextstyle(3,HORIZ_DIR,4);
    outtextxy(screen_left/2,screen_bottom/2,timer);
    outtextxy(screen_end-
(screen_left/2),screen_bottom/2,timer);
}
settextjustify(CENTER_TEXT, CENTER_TEXT);
}
gamer:
if(quit==1)
{
    //last_page();
    quit=0; // Quitting is complete
}

```

```

        l=0,r=0; // initialize the scores to 0
        g_o=1; // so that quit works as Game Over
        ball_x=screen_mid;
    }
    if(g_o==1)
    {
        g_o=0;
        goto begin;
    }
}

void start_pong()
{
    int bk= COLOR(255,255,240);
    setfillstyle(SOLID_FILL,bk);
    bar(0,0,screen_x,screen_y);
    setbkcolor(bk);
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    setcolor(COLOR(255,165,0));
    settextstyle(BOLD_FONT,HORIZ_DIR,10);
    outtextxy(screen_x/2,150,"PONG");

    settextjustify(RIGHT_TEXT, TOP_TEXT);
    setcolor(COLOR(154,205,50));
    settextstyle(FONT,HORIZ_DIR,4);
    outtextxy(screen_x/2+250,180,"Edition 6.8");
    settextjustify(RIGHT_TEXT, TOP_TEXT);
    settextstyle(FONT,HORIZ_DIR,3);
    outtextxy(screen_x-10,screen_y-100,"Designed and Developed By :");
    outtextxy(screen_x-10,screen_y-50,"Md.Shorifuzzaman and Fahim Arefin");
    settextjustify(LEFT_TEXT, TOP_TEXT);
    if(flag==0)
    {
        left_bat_up=72;
        left_bat_down=80;
    }
    flag=1;
}

int timel()
{
    double diff_t;
    int time_flag=0;
    time(&end_t);
    diff_t = difftime(end_t, start_t);
    if(diff_t-TIME==1)
        time_flag=1;
    return time_flag;
}

```

menufunc.h:

```

void about()
{
    int x,y;
    start_pong();
    setcolor(COLOR(255,165,50));
    setfillstyle(SOLID_FILL,COLOR(255,165,0));
    fillellipse(screen_left/2,215+(menu_ymid/2),10,10);
    fillellipse(screen_left/2,215+menu_ymid,10,10);
    fillellipse(screen_left/2,215+(menu_ymid*3)/2,10,10);
    setcolor(COLOR(240,232,200));
    setfillstyle(SOLID_FILL,COLOR(240,232,170));
    draw_triangle(screen_left/2+10,212+menu_ymid*2,3);
    draw_triangle(screen_left/2+10,212+(menu_ymid*5)/2,3);
    draw_triangle(screen_left/2+10,212+menu_ymid*3,3);
    settextjustify(LEFT_TEXT, TOP_TEXT);
    setcolor(GREY);
    settextstyle(FONT,HORIZ_DIR,3);
    outtextxy(screen_left,200+menu_ymid/2,"Pong(1972) by Atari is considered the
Mother of Arcade Games");
    outtextxy(screen_left,200+menu_ymid,"You can set the Game Point from the
Menu");
    outtextxy(screen_left,200+(menu_ymid*3)/2,"Instructions -");
    outtextxy(screen_left+20,200+menu_ymid*2,"Historically,The only instruction
to play the game is: Avoid missing ball for high score.");
    outtextxy(screen_left+20,200+(menu_ymid*5)/2,"Time Challenge: The Higher
scorer in the Time limit wins the game");
    outtextxy(screen_left+20,200+menu_ymid*3,"Press ESC to Pause and Resume the
Game");

    settextjustify(LEFT_TEXT, TOP_TEXT);
    settextstyle(FONT,HORIZ_DIR,3);
    int backx=textwidth("BACK");
    int backy=textheight("BACK");
    setfillstyle(SOLID_FILL,GREY);
    bar(screen_x/2-backx/2-10,200+(7*menu_ymid)/2-
10,screen_x/2+backx/2+10,200+(7*menu_ymid)/2+backy+10);

    setbkcolor(GREY);
    setcolor(WHITE);
    outtextxy(screen_x/2-backx/2,200+(7*menu_ymid)/2,"BACK");
    again:
    clearmouseclick(WM_LBUTTONDOWN);
    while(!ismouseclick(WM_LBUTTONDOWN))
    {
        delay(100);
    }
    getmouseclick(WM_LBUTTONDOWN,x,y);
    if(x>screen_x/2-backx/2-10 && y>200+(7*menu_ymid)/2-10 &&
x<screen_x/2+backx/2+10 && y<200+(7*menu_ymid)/2+backy+10)
}

```

```

        return;
    else
        goto again;
}

void background()
{
    setfillstyle(SOLID_FILL,BLACK);
    bar(0,0,screen_end,screen_bottom);
    setfillstyle(SOLID_FILL,score_back);
    bar(0,screen_top,screen_left,screen_y);
    bar(screen_right,screen_top,screen_x,screen_y);
    bar(0,screen_bottom,screen_right,screen_y);
}

void create_menu()
{
    int bk= COLOR(255,255,240);
    int font=COLOR(255,165,0);
    setfillstyle(SOLID_FILL,bk);
    bar(0,0,screen_x,screen_y);
    setbkcolor(bk);
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    setcolor(font);
    settextstyle(BOLD_FONT,HORIZ_DIR,10);
    outtextxy(midx,150,"MENU");
    setfillstyle(SOLID_FILL,GREY);
    bar(0,200,screen_x,205);
    bar(0,menu_y1,screen_x,menu_y1+5);
    bar(0,screen_y-5,screen_x,screen_y);
    bar(menu_x1,200,menu_x1+5,screen_y);
    bar(menu_x2,200,menu_x2+5,screen_y);
    setcolor(font);
    setbkcolor(bk);
    settextstyle(FONT,HORIZ_DIR,3);

    outtextxy(menu_x1/2,200+menu_ymid,"SET GAME SCREEN SIZE");
    outtextxy(menu_x1+menu_x1/2,200+menu_ymid,"PLAY TIME CHALLENGE");
    outtextxy(menu_x2+menu_x1/2,200+menu_ymid,"LEADERBOARD");
    outtextxy(menu_x1/2,menu_y1+menu_ymid,"GAME POINT");
    outtextxy(menu_x1+menu_x1/2,menu_y1+menu_ymid,"PLAY PONG");
    outtextxy(menu_x2+menu_x1/2,menu_y1+menu_ymid,"BACK");
}

void draw_triangle(int x,int y,int z)
{
    int points[9]={320,150,420,300,250,300,320,150};
    points[0]=x+10+z;
    points[1]=y;
    points[6]=x+10+z;
    points[7]=y;
}

```

```

points[2]=x-z;
points[3]=y+10;
points[4]=x-z;
points[5]=y-10;
setcolor(COLOR(255,165,0));
setfillstyle(SOLID_FILL,COLOR(255,165,0));
fillpoly(4,points);
}

void first_page()
{
    int color=COLOR(210,231,3);
    int d=(screen_x-450)/4;
    int h=150;
    int x,y;
    start_pong();
    draw_triangle(screen_left-30,315,5);
    draw_triangle(screen_left-30,375,5);
    settextjustify(LEFT_TEXT, TOP_TEXT);
    setcolor(GREY);
    settextstyle(FONT,HORIZ_DIR,FONT_SIZE);
    outtextxy(screen_left,300,"Use the mouse for Right Bat Control");
    outtextxy(screen_left,360,"Use 'UP' & 'DOWN' key for Left Bat Control");
    setcolor(color);
    setfillstyle(SOLID_FILL,color);
    //hexagon length 150, height 80
    bar(d,450,d+h,500);
    bar(2*d+h,450,2*d+2*h,500);
    bar(3*d+2*h,450,3*d+3*h,500);
    setbkcolor(color);
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    setcolor(GREY);
    settextstyle(FONT,HORIZ_DIR,3);
    outtextxy(d+h/2,475,"MENU");
    outtextxy(2*d+h+h/2,475,"PLAY");
    outtextxy(3*d+2*h+h/2,475,"SETTINGS");

    starter:
    clearmouseclick(WM_LBUTTONDOWN);
    while(!ismouseclick(WM_LBUTTONDOWN))
    {
        delay(100);
    }
    getmouseclick(WM_LBUTTONDOWN, x, y);

    if(x>d && x<d+h && y>=450 && y<=500)
        control_l=1;
    else if (x>2*d+h && x<(2*d+2*h) && y>=450 && y<=500)
        control_l=2;
    else if(x>3*d+2*h && x<(3*d+3*h) && y>=450 && y<=500)
        control_l=3;
}

```

```

else
    goto starter;
if(control_l==1)
{
    menu();
}
else if(control_l==3)
{
    settings_menu();
}
else
{
    return;
setbkcolor(BLACK);
}

void menu()
{
    int x=0, y=0;
create_menu();
clearmouseclick(WM_LBUTTONDOWN);
while(!ismouseclick(WM_LBUTTONDOWN))
{
    delay(100);
}
getmouseclick(WM_LBUTTONDOWN, x, y);
select_menu(x,y);
}

void select_menu(int x , int y)
{
    int space,wid,hei ;
    int bk= COLOR(255,255,240);
    if((x>0 && x<menu_x1) && (y>200 && y<menu_y1))
    {
        printf("Set game screen\n");//set game screen
        setfillstyle(SOLID_FILL,bk);
        bar(0,205,menu_x1,menu_y1);
        setcolor(GREY);
        setfillstyle(SOLID_FILL,GREY);
        bar(0,200+menu_ymid,menu_x1,200+menu_ymid+5);//The barrier
        bar(menu_x1/2,200,menu_x1/2 + 5,menu_y1);
        setbkcolor(bk);
        setcolor(GREY);
        settextstyle(FONT,HORIZ_DIR,FONT_SIZE);
        outtextxy(menu_x1/4,menu_ytop,"600 * 480");
        outtextxy(menu_x1-menu_x1/4,menu_ytop,"800 * 640");
        outtextxy(menu_x1/4,menu_y1- menu_ymid/2,"750 * 500");
        outtextxy(menu_x1-menu_x1/4,menu_y1- menu_ymid/2,"900 * 600");
        clearmouseclick(WM_LBUTTONDOWN);
        while(!ismouseclick(WM_LBUTTONDOWN))
        {
    }
}

```

```

        delay(100);
    }
    getmouseclick(WM_LBUTTONDOWN, x, y);

    if((x>0 && x<menu_x1/2) && (y>200 && y<200+menu_ymid))/*600 * 480 =
5:4*/
    {
        wid=600;
        hei=480;
    }
    else if((x>0 && x<menu_x1/2) && (y>200+menu_ymid && y<menu_y1))/*800 *
640 = 5:4*/
    {
        wid=800;
        hei=640;
    }
    else if((x>menu_x1/2 && x<menu_x1) && (y>200 && y<200+menu_ymid)) /*750
* 500 = 3:2 */
    {
        wid=750;
        hei=500;
    }
    else if((x>menu_x1/2 && x<menu_x1) && (y>200+menu_ymid &&
y<menu_y1))/*900 * 600 = 3:2*/
    {
        wid=900;
        hei=600;
    }
    space= (screen_x-wid)/2;
    screen_left=space; /*start of black screen*/
    screen_right=screen_left+wid; /*end of black screen*/
    screen_bottom=hei; /*bottom of black screen*/
    screen_end=screen_left+screen_right;
    screen_mid=screen_left+ wid/2;
    ball_y=screen_bottom/2,ball_x=screen_mid;
}
else if((x>menu_x1 && x<menu_x2) && (y>200 && y<menu_y1))
{
    printf("Play TIME CHALLENGE\n");
    mode=2;
    high_score=999;
    setfillstyle(SOLID_FILL,bk);
    bar(menu_x1+5,205,menu_x2,menu_y1);
    setcolor(GREY);
    setfillstyle(SOLID_FILL,GREY);
    bar(menu_x1,200+menu_ymid,menu_x2,200+menu_ymid+5); //The barrier
    bar(menu_x1+menu_x1/2,200,menu_x1+menu_x1/2+5,menu_y1);
    setbkcolor(bk);
    setcolor(GREY);
    settextstyle(FONT,HORIZ_DIR,3);
    outtextxy(menu_x1+menu_x1/4,menu_ytop, "30 seconds");
}

```

```

outtextxy(menu_x2-menu_x1/4,menu_ytop,"60 seconds");
outtextxy(menu_x1+menu_x1/4,menu_y1- menu_ymid/2,"150 seconds");
outtextxy(menu_x2-menu_x1/4,menu_y1- menu_ymid/2,"450 seconds");
int timec=0;
clearmouseclick(WM_LBUTTONDOWN);
while(!ismouseclick(WM_LBUTTONDOWN))
{
    delay(100);
}
getmouseclick(WM_LBUTTONDOWN, x, y);

if((x>menu_x1 && x<menu_x1+menu_x1/2) && (y>200 && y<200+menu_ymid))/*OPTION 1*/
    TIME=30;
else if((x>menu_x1 && x<menu_x1+menu_x1/2) && (y>200+menu_ymid && y<menu_y1))/*OPTION 2*/
    TIME=60;
else if((x>menu_x1+menu_x1/2 && x<menu_x2) && (y>200 && y<200+menu_ymid)) /*OPTION 3*/
    TIME=150;
else if((x>menu_x1+menu_x1/2 && x<menu_x2) && (y>200+menu_ymid && y<menu_y1))/*OPTION 4*/
    TIME=450;
else
    timec=1;
if(timec==0)
{
    time(&start_t);
    return;
}
else if((x>menu_x2 && x<screen_x) && (y>200 && y<menu_y1))
{
    printf("Leaderboard\n");
    score_menu();
}
/* Game Point Menu */
else if((x>0 && x<menu_x1) && (y>menu_y1 && y<screen_y))
{
    printf("GAME POINT\n");
    setfillstyle(SOLID_FILL,bk);
    bar(0,menu_y1+5,menu_x1,menu_y2);
    setcolor(GREY);
    setfillstyle(SOLID_FILL,GREY);
    bar(0,menu_y1+menu_ymid,menu_x1,menu_y1+menu_ymid+5); //The barrier
    bar(menu_x1/2,menu_y1,menu_x1/2 + 5,menu_y2);
    setbkcolor(bk);
    setcolor(GREY);
    settextstyle(FONT,HORIZ_DIR,FONT_SIZE);
    outtextxy(menu_x1/4,menu_y1+ menu_ymid/2,"5");
    outtextxy(menu_x1-menu_x1/4,menu_y1+ menu_ymid/2,"10");
}

```

```

outtextxy(menu_x1/4,menu_y1+3*menu_ymid/2,"15");
outtextxy(menu_x1-menu_x1/4,menu_y1+3*menu_ymid/2,"20");
clearmouseclick(WM_LBUTTONDOWN);
while(!ismouseclick(WM_LBUTTONDOWN))
{
    delay(100);
}
getmouseclick(WM_LBUTTONDOWN, x, y);

if((x>0 && x<menu_x1/2) && (y>menu_y1 && y<menu_y1+menu_ymid))/* Game
Point 5 */
    high_score=5;
else if((x>0 && x<menu_x1/2) && (y>menu_y1+menu_ymid && y<menu_y2))/* Game Point 15 */
    high_score=15;
else if((x>menu_x1/2 && x<menu_x1) && (y>menu_y1 && y<menu_y1+menu_ymid)) /* Game Point 10 */
    high_score=10;
else if((x>menu_x1/2 && x<menu_x1) && (y>menu_y1+menu_ymid && y<menu_y2))/* Game Point 20 */
    high_score=20;
printf("%d\n",high_score);
}
else if((x>menu_x1 && x<menu_x2) && (y>menu_y1 && y<screen_y))
{
    printf("Play\n");
    return;
}
else if((x>menu_x2 && x<screen_x) && (y>menu_y1 && y<screen_y))
{
    first_page();
    return;
}
menu();
}

void settings_menu()
{
    cleardevice();
    int x,y;
    int bk= COLOR(255,255,240);
    int font=COLOR(255,165,0);
    setfillstyle(SOLID_FILL,bk);
    bar(0,0,screen_x,screen_y);
    setbkcolor(bk);
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    setcolor(font);
    settextstyle(BOLD_FONT,HORIZ_DIR,10);
    outtextxy(midx,150,"SETTINGS");
    setfillstyle(SOLID_FILL,GREY);
    bar(0,200,screen_x,205);
}

```

```

bar(0,menu_y1,screen_x,menu_y1+5); // middle horizontal bar
bar(0,screen_y-5,screen_x,screen_y); // bottom bar
bar(menu_x1,200,menu_x1+5,screen_y);
bar(menu_x2,200,menu_x2+5,screen_y);
setcolor(font);
setbkcolor(bk);
settextstyle(FONT,HORIZ_DIR,3);

outtextxy(menu_x1/2,200+menu_ymid,"ABOUT");
outtextxy(menu_x1+menu_x1/2,200+menu_ymid,"SOUNDS");
outtextxy(menu_x2+menu_x1/2,200+menu_ymid,"BAT SPEED");
outtextxy(menu_x1/2,menu_y1+menu_ymid,"SCORE POSITION");
outtextxy(menu_x1+menu_x1/2,menu_y1+menu_ymid,"SCORECARD VIEW");
outtextxy(menu_x2+menu_x1/2,menu_y1+menu_ymid,"BACK");

clearmouseclick(WM_LBUTTONDOWN);
while(!ismouseclick(WM_LBUTTONDOWN))
{
    delay(100);
}
getmouseclick(WM_LBUTTONDOWN, x, y);

if((x>0 && x<menu_x1) && (y>200 && y<menu_y1))
{
    printf("About\n");
    about();
}
else if((x>menu_x1 && x<menu_x2) && (y>200 && y<menu_y1))
{
    printf("SOUNDS\n");
    setfillstyle(SOLID_FILL,bk);
    bar(menu_x1+5,205,menu_x2,menu_y1);
    setcolor(GREY);
    setfillstyle(SOLID_FILL,GREY);
    bar(menu_x1,200+menu_ymid,menu_x2,200+menu_ymid+5); //The barrier
    bar(menu_x1+menu_x1/2,200,menu_x1+menu_x1/2+5,menu_y1);
    setbkcolor(bk);
    setcolor(GREY);
    settextstyle(FONT,HORIZ_DIR,3);
    outtextxy(menu_x1+menu_x1/4,menu_ytop,"MUTE");
    outtextxy(menu_x2-menu_x1/4,menu_ytop,"SOUND 1");
    outtextxy(menu_x1+menu_x1/4,menu_y1- menu_ymid/2,"SOUND 2");
    outtextxy(menu_x2-menu_x1/4,menu_y1- menu_ymid/2,"SOUND 3");
    clearmouseclick(WM_LBUTTONDOWN);
    while(!ismouseclick(WM_LBUTTONDOWN))
    {
        delay(100);
    }
    getmouseclick(WM_LBUTTONDOWN, x, y);
}

```

```

        if((x>menu_x1      &&      x<menu_x1+menu_x1/2)      &&      (y>200      &&
y<200+menu_ymid))/*MUTE*/
            SOUND=0;
        else  if((x>menu_x1  &&  x<menu_x1+menu_x1/2)  &&  (y>200+menu_ymid  &&
y<menu_y1))/*SOUND 1*/
            SOUND=55;
        else  if((x>menu_x1+menu_x1/2  &&  x<menu_x2)  &&  (y>200  &&
y<200+menu_ymid)) /*SOUND 2*/
            SOUND=85;
        else  if((x>menu_x1+menu_x1/2  &&  x<menu_x2)  &&  (y>200+menu_ymid  &&
y<menu_y1))/*SOUND 3*/
            SOUND=150;
    }
else if((x>menu_x2  &&  x<screen_x)  &&  (y>200  &&  y<menu_y1))
{
    printf("BAT SPEED\n");
    setfillstyle(SOLID_FILL,bk);
    bar(menu_x2+5,205,menu_x3,menu_y1);
    setcolor(GREY);
    setfillstyle(SOLID_FILL,GREY);
    bar(menu_x2,200+menu_ymid,menu_x3,200+menu_ymid+5); //The barrier
    bar(menu_x2+menu_x1/2,200,menu_x2+menu_x1/2+5,menu_y1);
    setbkcolor(bk);
    setcolor(GREY);
    settextstyle(FONT,HORIZ_DIR,3);
    outtextxy(menu_x2+menu_x1/4,menu_ytop,"1x");
    outtextxy(menu_x3-menu_x1/4,menu_ytop,"1.2x");
    outtextxy(menu_x2+menu_x1/4,menu_y1- menu_ymid/2,"1.4x");
    outtextxy(menu_x3-menu_x1/4,menu_y1- menu_ymid/2,"1.75x");

    clearmouseclick(WM_LBUTTONDOWN);
    while(!ismouseclick(WM_LBUTTONDOWN))
    {
        delay(100);
    }
    getmouseclick(WM_LBUTTONDOWN, x, y);

    if((x>menu_x2      &&      x<menu_x2+menu_x1/2)      &&      (y>200      &&
y<200+menu_ymid))/*batspeed 100 */
        bat_speed=100;
    else  if((x>menu_x2  &&  x<menu_x2+menu_x1/2)  &&  (y>200+menu_ymid  &&
y<menu_y1))/*batspeed 140 seconds*/
        bat_speed=140;
    else if((x>menu_x2+menu_x1/2  &&  x<menu_x3)  &&  (y>200  &&  200+menu_ymid))
/* batspeed 120 seconds */
        bat_speed=120;
    else  if((x>menu_x2+menu_x1/2  &&  x<menu_x3)  &&  (y>200+menu_ymid  &&
y<menu_y1))/*batspeed 175 seconds */
        bat_speed=175;
}

```

```

    else if((x>0 && x<menu_x1) && (y>menu_y1 && y<screen_y)) // Set Score
Position
{
    int x,y;
    int wid=(screen_y-205)/10;
    int bk= COLOR(255,255,240);
    int font=COLOR(255,165,0);
    setfillstyle(SOLID_FILL,bk);
    bar(0,menu_y1+5,menu_x1,menu_y2-5);
    setcolor(GREY);
    setfillstyle(SOLID_FILL,GREY);
    bar(0,menu_y1+(menu_y2-menu_y1)/3,menu_x1,menu_y1+(menu_y2-
menu_y1)/3+5); //horizontal barrier 1
    bar(0,menu_y1+2*(menu_y2-menu_y1)/3,menu_x1,menu_y1+2*(menu_y2-
menu_y1)/3+5); //horizontal barrier 2
    setcolor(GREY);
    setbkcolor(bk);
    settextstyle(FONT,HORIZ_DIR,FONT_SIZE);
    outtextxy(menu_x1/2,menu_y1+(menu_y2-menu_y1)/6+3,"TOP");
    outtextxy(menu_x1/2,menu_y1+(menu_y2-menu_y1)/2+3,"MIDDLE");
    outtextxy(menu_x1/2,menu_y2-(menu_y2-menu_y1)/6+3,"BOTTOM");
    clearmouseclick(WM_LBUTTONDOWN);
    while(!ismouseclick(WM_LBUTTONDOWN));
    getmouseclick(WM_LBUTTONDOWN, x, y);

    if((x>0 && x<menu_x1) && (y>menu_y1 && y<menu_y1+(menu_y2-
menu_y1)/3))/*TOP */
        score_pos=1;
    else if((x>0 && x<menu_x1) && (y>menu_y1+(menu_y2-menu_y1)/3 &&
y<2*(menu_y2-menu_y1)/3)) /* MIDDLE*/
        score_pos=2;
    else if((x>0 && x<menu_x1) && (y>2*(menu_y2-menu_y1)/3 &&
y<menu_y2))/*BOTTOM*/
        score_pos=3;
    printf("%d\n",score_pos);
}
else if((x>menu_x1 && x<menu_x2) && (y>menu_y1 && y<screen_y))
{
    printf("SCORECARD VIEW\n");
    score_card();
}
else if((x>menu_x2 && x<menu_x3) && (y>menu_y1 && y<screen_y))
{
    printf("BACK\n");
    first_page();
    return;
}
if(pausey!=1)
    settings_menu();
}

```

scorefunc.h:

```

void score()
{
    sprintf(left_score, "%d", l);
    sprintf(right_score, "%d", r);
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    setbkcolor(score_back);
    setcolor(score_col);
    settextstyle(FONT,HORIZ_DIR,5);
    if(score_pos==1) /*score at top */
    {
        outtextxy(screen_left/2,100,left_score);
        outtextxy(screen_end-(screen_left/2),100,right_score);
    }
    else if(score_pos==2) /* score at middle */
    {
        outtextxy(screen_left/2,screen_bottom/2,left_score);
        outtextxy(screen_end-(screen_left/2),screen_bottom/2,right_score);
    }
    else if(score_pos==3) /* score at bottom */
    {
        outtextxy(screen_left/2,screen_bottom-100,left_score);
        outtextxy(screen_end-(screen_left/2),screen_bottom-100,right_score);
    }
}

void score_card()
{
    int x,y;
    int choice;
    int wid=(screen_y-205)/10;
    int bk= COLOR(255,255,240);
    int font=COLOR(255,165,0);
    setfillstyle(SOLID_FILL,bk);
    bar(0,0,screen_x,screen_y);
    setbkcolor(bk);
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    setcolor(font);
    settextstyle(BOLD_FONT,HORIZ_DIR,9);
    outtextxy(screen_x/2,100,"SCORECARD VIEW");
    setfillstyle(SOLID_FILL,GREY);
    bar(0,150,screen_x,155);
    bar(0,200,screen_x,205);
    bar(0,205+(screen_y-200)/3,menu_x1,210+(screen_y-200)/3);
    bar(0,screen_y-(screen_y-200)/3,menu_x1,screen_y+5-(screen_y-200)/3);
    bar(0,screen_y-5,screen_x,screen_y);
}

```

```

bar(menu_x1,150,menu_x1+5,screen_y);
bar(menu_x2,150,menu_x2+5,screen_y);
setcolor(font);
setbkcolor(bk);
settextstyle(FONT,HORIZ_DIR,3);
outtextxy(menu_x1/2,185,"OPTIONS");
outtextxy(menu_x1/2,205+(screen_y-200)/6,"SELECT SCOREBOARD COLOR");
outtextxy(menu_x1/2,205+(screen_y-200)/3+(screen_y-200)/6,"SELECT
STYLE");
outtextxy(menu_x1/2,screen_y-(screen_y-200)/3+(screen_y-200)/6,"BACK");

starter:
clearmouseclick(WM_LBUTTONDOWN);
while(!ismouseclick(WM_LBUTTONDOWN))
{
    delay(100);
}
getmouseclick(WM_LBUTTONDOWN, x, y);

if(x>0 && x<menu_x1 && y>200 && y<205+(screen_y-200)/3)
{
    choice=1;
    setcolor(font);
    setfillstyle(SOLID_FILL,font);
    fillellipse(menu_x1/2,205+(screen_y-200)/12,10,10);
    setbkcolor(bk);
    setcolor(font);
    outtextxy(menu_x1+(menu_x1)/2,185,"SCOREBOARD COLOR");
    setfillstyle(SOLID_FILL,COLOR(157,149,115)); //Beige 100%
    bar(menu_x1+5,205,menu_x2,205+wid);
    setfillstyle(SOLID_FILL,COLOR(221,207,153)); //Sandstone
    bar(menu_x1+5,205+wid+1,menu_x2,205+2*wid);
    setfillstyle(SOLID_FILL,COLOR( 138,136,125)); // Gray 60%
    bar(menu_x1+5,205+2*wid+1,menu_x2,205+wid*3);
    setfillstyle(SOLID_FILL,COLOR(88,87,84)); //Black 80%
    bar(menu_x1+5,205+wid*3+1,menu_x2,205+wid*4);
    setfillstyle(SOLID_FILL,COLOR( 140,21,21)); //Cardinal red
    bar(menu_x1+5,205+wid*4+1,menu_x2,205+wid*5);
    setfillstyle(SOLID_FILL,COLOR(21, 52, 80)); //Elephant Midnight Blue
    bar(menu_x1+5,205+wid*5+1,menu_x2,205+wid*6);
    setfillstyle(SOLID_FILL,COLOR(56, 46, 28)); //Bistre
    bar(menu_x1+5,205+wid*6+1,menu_x2,205+wid*7);
    setfillstyle(SOLID_FILL,COLOR(173, 116, 96) ); //Santa fe
    bar(menu_x1+5,205+wid*7+1,menu_x2,205+wid*8);
    setfillstyle(SOLID_FILL,COLOR(198, 199, 140)); //Pine glade
    bar(menu_x1+5,205+wid*8+1,menu_x2,205+wid*9);
    setfillstyle(SOLID_FILL,COLOR(231, 195, 156)); //Cashmere
    bar(menu_x1+5,205+wid*9+1,menu_x2,205+wid*10);
    clearmouseclick(WM_LBUTTONDOWN);
    while(!ismouseclick(WM_LBUTTONDOWN));
    getmouseclick(WM_LBUTTONDOWN, x, y);
}

```

```
score_back=getpixel(x,y);

setfillstyle(SOLID_FILL,score_back);
bar(menu_x1/2- 40,210+(screen_y-200)/3-75,menu_x1/2+ 40,210+(screen_y-200)/3-25);
setcolor(bk);
setfillstyle(SOLID_FILL,bk);
fillellipse(menu_x1/2,205+(screen_y-200)/12,10,10);
}

if (x>0 && x<menu_x1 && y>205+(screen_y-200)/3 && y<screen_y-(screen_y-200)/3
|| choice==1)
{
    choice=2;
    setcolor(font);
    setfillstyle(SOLID_FILL,font);
    fillellipse(menu_x1/2,205+(screen_y-200)/3+(screen_y-200)/12,10,10);
    setbkcolor(bk);
    setcolor(font);
    outtextxy(menu_x2+(menu_x1)/2,185,"BAT STYLE");
    setbkcolor(score_back);
    setfillstyle(0,bat_col);
    bar(menu_x2+5,205,screen_x,205+wid);
    setfillstyle(1,bat_col);
    bar(menu_x2+5,205+wid+1,screen_x,205+2*wid);
    setfillstyle(2,bat_col);
    bar(menu_x2+5,205+2*wid+1,screen_x,205+wid*3);
    setfillstyle(3,bat_col);
    bar(menu_x2+5,205+wid*3+1,screen_x,205+wid*4);
    setfillstyle(4,bat_col);
    bar(menu_x2+5,205+wid*4+1,screen_x,205+wid*5);
    setfillstyle(5,bat_col);
    bar(menu_x2+5,205+wid*5+1,screen_x,205+wid*6);
    setfillstyle(6,bat_col);
    bar(menu_x2+5,205+wid*6+1,screen_x,205+wid*7);
    setfillstyle(11,bat_col);
    bar(menu_x2+5,205+wid*7+1,screen_x,205+wid*8);
    setfillstyle(8,bat_col);
    bar(menu_x2+5,205+wid*8+1,screen_x,205+wid*9);
    setfillstyle(9,bat_col);
    bar(menu_x2+5,205+wid*9+1,screen_x,205+wid*10);
    clearmouseclick(WM_LBUTTONDOWN);
    while(!ismouseclick(WM_LBUTTONDOWN));
    getmouseclick(WM_LBUTTONDOWN, x, y);
    if(x>=menu_x2+5 && x<=screen_x)
    {
        bat_fill=(y-205-1)/wid;
        if(bat_fill==7)
            bat_fill=11;
    }
    setbkcolor(score_back);
```

```

        setfillstyle(bat_fill,bat_col);
        bar(menu_x1/2-                      40,screen_y+5-(screen_y-200)/3-75,menu_x1/2+
40,screen_y+5-(screen_y-200)/3-25);
        setfillstyle(SOLID_FILL,bk);
        bar(menu_x1+5,155,screen_x-5,screen_y-5);
        setfillstyle(SOLID_FILL,GREY);
        bar(menu_x1+5,200,screen_x,205);
        bar(menu_x2,150,menu_x2+5,screen_y);
        setcolor(bk);
        setfillstyle(SOLID_FILL,bk);
        fillellipse(menu_x1/2,205+(screen_y-200)/3+(screen_y-200)/12,10,10);
        goto starter;
    }
else if (x>0 && x<menu_x1 && y>(screen_y-(screen_y-200)/3) && y<screen_end)
{
    choice=3;
    return ;
}
else if (x>menu_x1)
    goto starter;
}

```

```

void leaderboard(int n, int p)
{
    char scorer[1000];
    int bk= COLOR(255,255,240);
    int font=COLOR(255,165,0);
    setfillstyle(SOLID_FILL,bk);
    bar(menu_x2+5,205,menu_x3,screen_y-5);
    setbkcolor(bk);
    setcolor(font);
    settextstyle(FONT,HORIZ_DIR,3);
    outtextxy(menu_x1,menu_y1,"CLICK ANYWHERE TO RETURN TO MENU");
    sprintf(scorer,"Best %d Results for Player %d",n,p);
    outtextxy(menu_x2+menu_x1/2,185,scorer);
    for(m=0;m<n && m<=j[p];m++)
    {
        outtextxy(menu_x2+menu_x1/2,240+((screen_y-240)/n)*m,top[p][m]);
    }
    r=0;
    l=0;
    //clearmouseclick(WM_LBUTTONDOWN);
    while(!ismouseclick(WM_LBUTTONDOWN));
}

```

```

void score_check(int n, int p)
{
    m=0;
    struct record

```

```

{
    int p1;
    int p2;
    int dif[3];
    char name[100];
} list[550],temp;
int
rec1[1000][20],rec2[1000][20],x1[1000]={0},x2[1000]={0},ind,maxd[3],mind[3],num,las
t=0;
int i,k,y,numx;
fclose(score_list);
score_list=fopen("score log.fs","r"); //Check
fseek(score_list,0L,0);
for(i=0;;i++)
{
    fscanf(score_list,"%d
%[^
\n]s",&list[i].p2,&list[i].p1,list[i].name); //p2 = left, p1=right
    list[i].dif[1]=list[i].p1-list[i].p2;
    list[i].dif[2]=list[i].p2-list[i].p1;
    if(feof(score_list))
    {
        break;
    }
}
fclose(score_list);
score_list=fopen("score log.fs","w");
maxd[1]=-1;
maxd[2]=-1;
mind[1]=1050;
mind[2]=1050;
for(k=0;k<i;k++) // i is the number of total records
{
    if(list[k].dif[1]>=0)
    {
        ind=list[k].dif[1]; //difference for player 1
        rec1[ind][x1[ind]++]=k;
        if(ind >maxd[1])
            maxd[1]=ind;
        if(ind <mind[1])
            mind[1]=ind;
    }
    if(list[k].dif[2]>=0)
    {
        ind=list[k].dif[2];
        rec2[ind][x2[ind]++]=k;
        if(ind >maxd[2])
            maxd[2]=ind;
        if(ind <mind[2])
            mind[2]=ind;
    }
}

```

```

for(k=maxd[1],j[1]=0;k>=mind[1] && j[1]<n;k--)
{
    for(num=0;num<x1[k];num++)
    {
        for(numx=0;numx<x1[k]-1;numx++)
        {
            ind=rec1[k][numx];
            if(list[rec1[k][numx]].p1<list[rec1[k][numx+1]].p1)
            {
                temp=list[rec1[k][numx]];
                list[rec1[k][numx]]=list[rec1[k][numx+1]];
                list[rec1[k][numx+1]]=temp;
            }
        }
    }
    for(num=0;num<x1[k] && j[1]<n;num++,j[1]++)
    {
        ind=rec1[k][num];
        fprintf(score_list,"%d
%s\n",list[ind].p2,list[ind].p1,list[ind].name);
        sprintf(top[1][j[1]],"%s
%d",list[ind].name,list[ind].p1,list[ind].p2);
    }
}

for(k=maxd[2],j[2]=0;k>=mind[2] && j[2]<n;k--)
{
    for(num=0;num<x2[k];num++)
    {
        for(numx=0;numx<x2[k]-1;numx++)
        {
            ind=rec2[k][numx];
            if(list[rec2[k][numx]].p2<list[rec2[k][numx+1]].p2)
            {
                temp=list[rec2[k][numx]];
                list[rec2[k][numx]]=list[rec2[k][numx+1]];
                list[rec2[k][numx+1]]=temp;
            }
        }
    }
    for(num=0;num<x2[k] && j[2]<n;num++,j[2]++)
    {
        ind=rec2[k][num];
        fprintf(score_list,"%d
%s\n",list[ind].p2,list[ind].p1,list[ind].name);
        sprintf(top[2][j[2]],"%s
%d",list[ind].name,list[ind].p2,list[ind].p1);
    }
}

if(used ==2 && p==2 )
{

```

```

        last=list[j[2]-1].dif[2];
        if((l-r)<last)
            elim=1;
        if( j[2]<n)
            elim=0;
    }
else if(used ==2 && p==1 )
{
    last=list[j[1]-1].dif[1];
    printf("%d\n",last);
    if((r-l)<last)
        elim=1;
    if( j[1]<n)
        elim=0;
}
if(j[p]<n)
{
    if(j[p]==0)
        sprintf(top[p][j[p]],"Sorry, But you never won with us");
    else
        sprintf(top[p][j[p]],"Sorry you won only %d times",j[p]);
}
fclose(score_list);
if(used!=2)
    leaderboard(n,p);
}

```

```

void user_input(int us)
{
    char nameinput[100];
    int bk= COLOR(255,255,240);
    setbkcolor(score_back);
    settextjustify(CENTER_TEXT, TOP_TEXT);
    setcolor(score_col);
    settextstyle(BOLD_FONT,HORIZ_DIR,6);
    sprintf(nameinput,"PLAYER %d is the winner",us);
    outtextxy(screen_x/2,screen_bottom+2,nameinput);

    int color=COLOR(255,255,240);
    int d=(screen_x-450)/4;
    int h=150;
    int x,y;
    setcolor(color);
    setfillstyle(SOLID_FILL,color);
    bar(0,screen_bottom+50,screen_x,screen_bottom+100);
    setbkcolor(color);
    settextjustify(LEFT_TEXT, CENTER_TEXT);
    setcolor(score_back);
    settextstyle(FONT,HORIZ_DIR,FONT_SIZE);
    outtextxy(h/2,screen_bottom+80,"ENTER YOUR NAME:");
}

```

```

int l=h/2 + textwidth("ENTER YOUR NAME:") + 20;
int i;
rewind( stdin );
for(i=0;;i++)
{
    username[i]=getch();
    if(username[i]=='\b')
    {
        if(i>0)
        i-=2; // if there is at least two valid letter, we go two spaces back &
then because of the loop, i++.so the last letter gets eliminated
        else if(i==0)
        {
            username[i]=0; // when you don't have any character and you still
press backspace, the string is NULL
            i=i-1; // i=i-1 , so after the loop i=0;
        }
        setfillstyle(SOLID_FILL,color);
        bar(l,screen_bottom+50,l+textwidth(username),screen_bottom+100);
    }
    else if(username[i]=='\r')
        break;
    username[i+1]=0;
    outtextxy(l,screen_bottom+80,username);
}
puts(username);
setfillstyle(SOLID_FILL,bk);
bar(0,0,getmaxwidth(),getmaxheight());
}

```

```

void score_menu()
{
    int x,y,n,p;
    int bk= COLOR(255,255,240);
    int font=COLOR(255,165,0);
    setfillstyle(SOLID_FILL,bk);
    bar(0,0,screen_x,screen_y);
    setbkcolor(bk);
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    setcolor(font);
    settextstyle(BOLD_FONT,HORIZ_DIR,8);
    outtextxy(screen_x/2,100,"BEST RECORDS");
    setfillstyle(SOLID_FILL,GREY);
    bar(0,150,screen_x,155);
    bar(0,200,screen_x,205);
    bar(0,menu_y1,screen_x,menu_y1+5);
    bar(0,screen_y-5,screen_x,screen_y);
    bar(menu_x1,150,menu_x1+5,screen_y);
    bar(menu_x2,150,menu_x2+5,screen_y);
    setcolor(font);
}

```

```

setbkcolor(bk);
settextstyle(FONT,HORIZ_DIR,3);
outtextxy(menu_x1/2,185,"Select Player");
outtextxy(menu_x1/2,200+menu_ymid,"Player 1");
outtextxy(menu_x1/2,menu_y1+menu_ymid,"Player 2");

player_select:
clearmouseclick(WM_LBUTTONDOWN);
while(!ismouseclick(WM_LBUTTONDOWN));
getmouseclick(WM_LBUTTONDOWN, x, y);

if(x>0 && x<menu_x1 && y>200 && y<menu_y1)
{
    p=1;
    setcolor(COLOR(255,165,0));
    setfillstyle(SOLID_FILL,COLOR(255,165,0));
    fillellipse(100,200+menu_ymid-5,10,10);
}
else if (x>0 && x<menu_x1 && y>menu_y1 && y<screen_y)
{
    p=2;
    setcolor(COLOR(255,165,0));
    setfillstyle(SOLID_FILL,COLOR(255,165,0));
    fillellipse(100,menu_y1+menu_ymid-10,10,10);
}
else
    goto player_select;

outtextxy(menu_x1+(menu_x1)/2,185,"Number of Records");
outtextxy(menu_x1+(menu_x1)/2,200+menu_ymid,"5");
outtextxy(menu_x1+(menu_x1)/2,menu_y1+menu_ymid,"10");

record_select:
clearmouseclick(WM_LBUTTONDOWN);
while(!ismouseclick(WM_LBUTTONDOWN));
getmouseclick(WM_LBUTTONDOWN, x, y);

if(x>menu_x1 && y>200 && x<menu_x2 && y< menu_y1)
    n=5;
else if (x>menu_x1 && y>menu_y1 && x<menu_x2 && y<screen_y)
    n=10;
else
    goto record_select;
setfillstyle(SOLID_FILL,bk);
bar(0,155,menu_x2,screen_y-5);
outtextxy(menu_x1,menu_y1,"CLICK ANYWHERE TO RETURN TO MENU");
used=1;
score_check(n,p);
}

```

Limitations:

The delay(int x) function is marked as "CPU Friendly", so the speed of the ball is not constant across all devices

Conclusion:

To ensure that this program works on all Windows devices, the .exe file was built using Release Mode on Visual Studio 2010.

As Pong is considered as the mother of all arcade games, we tried to stick to the heritage of this game and made the maximum usage of Borland Graphics Interface and we tried to negate most of its drawbacks.. We made minor tweaks to the original game to have an original feel throughout the game and added a bunch of customization options for the user so that they can set the settings according their preferences.

It has been a memorable experience to make a game of our own and it is far more special because it is the first user-oriented software we ever made.

We've learnt a lot from this experience and we look forward to implement our experience in the future projects.

References:

This project has been made according to the suggestions of our honorable teachers and is inspired by the historic "Pong Game". We used the online resources to learn about the bgi graphics environment and troubleshoot the program.

We would specially thank our honorable teachers - Dr.Md. Asif Hossain Khan sir and Anna Fariha madam, for their continuous guidance and suggestions.