**FLUTTER**



**Contents:**

Page number

1. Introduction ……………………………….............. 3
2. Project Goal …………………………………………… 3
3. The Game…...………………………………………….. 3
4. Rules…………..………………………………………….. 4
5. User Defined Functions…………………………… 6
6. Header Files Used in The Project…………….. 7
7. Algorithm……………………………………………….. 8
8. Flow Chart………………………………………………. 9
9. Source Code…………………………………………… 10
10. Achievement………………………………………….. 21
11. Limitations…………………………………………….. 21
12. Conclusions……………………………………………. 21

**Introduction:**

Here we made a simple colorful game using BGI ( Boreland Graphics Interface ) and named it ‘ flutter’. Microsoft Visual studio 2010 compiler is used in the project. The code is written in C.

**Project Goal:**

The goal of this project was to create a game using the BGI library. The game is written in C. Mainly we wanted to implement our knowledge in a practical case and have some fun.

**The Game:**

In the game ‘flutter’ 3 levels are designed. In level-I , flowers of two different colors falling downward fastly . the players have to concentrate on which flowers he has to click.

On level-ll it is harder than as he has to click only on the center and in last level we use keyboard input. Players has to manage control over the butterfly height.

**RULES:**

1. In 1st level, flowers are coming randomly in downward direction, we have to click on the light red colored flower to score. Click on pink flower will lead to game over. we have to attain at least 10 points in 20 second for completing 1st level.

Here is a figure that shows the features of level 1.



Fig : screenshot of level 1.

2. In second level randomly 4 flowers will be generated in the screen. We have to click on the center of at least 2 flowers. In this way at least 5 points have to achieve for completing level II.

Here is a figure that shows the features of level 2.



Fig : screenshot of level 2

3.We have to save the butterfly from the obstacles coming to its way and if we manage to take it to its destiny we will win the game.

Here is a figure that shows the features of level 3.



Fig: screenshot of level 3

**6. User-Defined Functions:**

void starting():displays the menu.

void leaves():the co-ordinates of flower are sent to another function named ‘flower’.

void flower(): flowers are generated in the screen on the given position.

void butterfly(int x, int y): create butterfly at mouse click on the light red flower. x and y are the co-ordinates of the butterfly.

void lvl2(int marks) : level II starts. Here the marks is the score of level-I.

void leaves1(int nx, int ny): Sends the co-ordinates to the function ‘flower-1’. Here nx and ny are the co-ordinates.

void flower1(int n,int i): create flowers for level II. n and i are the co-ordinates which forms the flowers.

void butterfly1(): creates butterfly at mouseclick on flower.

void score(): Showed the score on the left corner of the screen

void writename(int scr): take the name of the player as input. Here scr is the final score.

void dashboard(): show the leadership board which consists of 1st 7 best scorer.

void score\_again(): shows the score after refreshing the screen.

void time(): counts time.

void menu(): play,instruction,dashboard,exit are shown and can be gone to those function through this.

void instructions(): Instruction of levels are given there.

void exit(): game exits using this function.

**Header Files Used in The Project:**

stdio.h:

This header file is used for these functions:

scanf() : Takes input.

sprint() : prints on the window.

string.h:

This header file is used for these functions:

strcpy() : copies string.

graphics.h:

This header file is used for these functions:

setcolor(): Sets the current color. Here we used this function to color the texts that we used and the border of flowers and butterfly.

settextstyle(): Imposes the style, direction and size of a text.

outtextxy(): Shows texts on a given coordinate.

setfillstyle(): Imposes the color and pattern of filling. We used this function to add some extra colors on our flowers and butterflies and the bars.

bar(): Draws a solid rectangle.

setbkcolor(): Sets current background color.

pieslice() : draws a shape like a slice of pie. Mainly we used it to draw the butterfly.

readimagefile() : puts image on the screen. We used it to make the background more colorful and interesting.

fillellipse() : draws a solid ellipse.

getmouseclick() : This function sets x and y to the pixel coordinates of an unprocessed event of the specified kind. We used it to deal with every kind of functions to do their work.

getpixel() : getpixel gets the color of the pixel located at (x,y).

ismouseclick() : This function returns true if there is an unprocessed mouse event of the specified kind.

delay() : The function pauses the computation for the specified number of milliseconds. Weused it whenever we needed some extra time.

circle: circle draws a circle in the current drawing color with its center at (x,y) and the radius given by radius.

kbhit():True (non-zero) if there is a character in the input buffer, otherwise false.

getch():The function reads one character from the keyboard and returns its ASCII value (without waiting for a return key). We used it in level 3 to make the butterfly fly and when we needed to enter our name.

**Algorithm:**

**Level-I:**

Level-I is designed by taking values from rand() function. Then those values are used as the x co-ordinates of the flowers. 20 flowers are created behind the screen and those are falling downwards by changing the ordinate of y-axis in a loop. The loop continues till certain time limit. Score are also counted at the same time. In each loop the screen is refreshed by calling readimage () function again and again. Mouse click is also checked in every loop. And when any click is detected, score () function is called, where it is verified that is the click is done on the light red rose, if so score is increased by one, else the wrong click makes the game over.

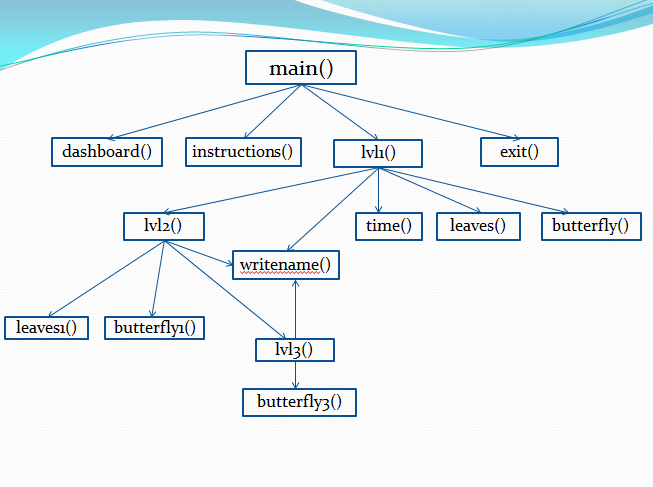
**Level-II :**

In level-II rand() function is used again to take the co- ordinate of 4 flowers at a time. Distance between each flower and the corner of the screen is counted. Then each flower is moved to the longest path in the screen by changing co-ordinates. Mouse click is checked in every loop. During the time when the all the 4 flowers don’t cross the screen, we have to click at least 2 flowers in their centers. Score will increase in each write click.

**Level-III :**

For level - III bars with random heights are designed. Bars are moved from the left corner to the right by changing the co-ordinate in a loop. Butterfly’s moving is controlled by taking input from player, it space button is clicked butterfly is going upward by else it falls downward. If any co-ordinate of butterfly and bars are matched the game will be over.

**Flow Chart:**

****

**Source Code :**

// my header.h

#include<stdio.h>

#include"graphics.h"

void lvl2(int marks);

void flower();

void leaves();

void flower1(int n,int i);

void leaves1(int nx,int ny);

void hide\_leaves(int nx,int ny);

void starting();

void butterfly1(int x,int y);

void butterfly();

void writename(int scr);

void dashboard();

void instruction();

void score\_again();

void lvl3();

//menu or starting

#include"graphics.h"

#include"myheader.h"

void starting()

{

readimagefile("menu.jpg",0,0,1000,700);

char s[10000];

setcolor(WHITE);

settextstyle(0,HORIZ\_DIR,3);

outtextxy(800,300,"DASHBOARD");

outtextxy(500, 50, "WElCOME :D");

outtextxy(400, 300, "PLAY");

outtextxy(800,400,"EXIT");

outtextxy(400,400,"INSTRUCTION");

int x,y;

while(!ismouseclick(WM\_LBUTTONDOWN))

{

delay(100);

}

getmouseclick(WM\_LBUTTONDOWN,x,y);

printf("%d\n%d",x,y);

if(x>=800 && x<=1000 && y>=300 && y<=350)

{

dashboard();

}

}

//instruction:

// leadership board

#include"graphics.h"

#include<stdio.h>

#include"myheader.h"

void dashboard()

{

char tempo[100];

struct file

{

char name[100];

int scr;

} listo[8];

int p1=1,i,j;

FILE \*m1;

m1=fopen("rank.txt","r");

for(i=0; i<=6; i++)

{

fscanf(m1,"%s%d",listo[i].name,&listo[i].scr);

}

readimagefile("lb4.jpg",0,0,1000,700);

setcolor(BLUE);

setbkcolor(15);

settextstyle(0,HORIZ\_DIR,3);

for(j=0; j<7; j++)

{

if(j==0)

{

sprintf(tempo,"%s",listo[j].name);

outtextxy(280,200,tempo);

sprintf(tempo,"%d",listo[j].scr);

outtextxy(580,200,tempo);

p1=p1+4;

}

else

{

sprintf(tempo,"%s",listo[j].name);

outtextxy(280,50\*p1,tempo);

sprintf(tempo,"%d",listo[j].scr);

outtextxy(580,50\*p1,tempo);

p1=p1+1;

}

}

setcolor(WHITE);

settextstyle(0,HORIZ\_DIR,3);

outtextxy(1000,350,"RETURN");

int x,y;

getmouseclick(WM\_LBUTTONDOWN,x,y);

while(!(x>=1000 && x<=1100 && y>=350 && y<=400))

{

delay(100);

getmouseclick(WM\_LBUTTONDOWN,x,y);

}

starting();

}

// level-I and main()

#include "graphics.h"

#include"myheader.h"

int x,y,i,j,p=0,l,z,scr=0,flag=0,n,t=1;

char s[20],b;

void readimagefile(

const char\* title,

int left, int top, int right, int bottom

);

void flower(int n,int i,int b)

{

pieslice(n,i,45,107,80);

pieslice(n,i,117,179,80);

pieslice(n,i,189,251,80);

pieslice(n,i,261,323,80);

pieslice(n,i,333,35,80);

setfillstyle(SOLID\_FILL,b);

fillellipse(n,i,20,20);

}

void score\_again()

{

setcolor(15);

sprintf(s,"SCORE: %d",scr);

outtextxy(10, 10, s);

}

void butterfly()

{

setfillstyle(SOLID\_FILL,COLOR(43,51,162));

pieslice(x,y,110,250,80);

pieslice(x,y,290,70,80);

setfillstyle(SOLID\_FILL,COLOR(79,88,208));

pieslice(x,y,110,250,60);

pieslice(x,y,290,70,60);

setfillstyle(SOLID\_FILL,COLOR(44,44,237));

pieslice(x,y,110,250,45);

pieslice(x,y,290,70,45);

setfillstyle(SOLID\_FILL,COLOR(0,0,196));

pieslice(x,y,110,250,30);

pieslice(x,y,290,70,30);

setcolor(1);

setfillstyle(SOLID\_FILL,COLOR(9,1,52));

fillellipse(x,y,8,35);

fillellipse(x,y-37,9,9);

}

void score()

{

getmouseclick(WM\_LBUTTONDOWN, x, y);

z=getpixel(x,y);

printf("%d ",z);

if(z==12)

{

scr++;

setcolor(15);

sprintf(s,"SCORE: %d",scr);

outtextxy(10, 10, s);

setcolor(9);

butterfly();

delay(200);

}

else if(z==13)

{

flag=1;

}

}

void leaves()

{

//flower

setcolor(15);

setfillstyle(SOLID\_FILL,13);

flower(n,50+i,4);

setfillstyle(SOLID\_FILL,12);

flower(n+300,50+i-j,4);

setfillstyle(SOLID\_FILL,13);

flower(n,50+i-2\*j,4);

setfillstyle(SOLID\_FILL,12);

flower(n+350,50+i-3\*j,4);

setfillstyle(SOLID\_FILL,13);

flower(n+450,50+i-4\*j,4);

setfillstyle(SOLID\_FILL,12);

flower(n+150,50+i-5\*j,4);

setfillstyle(SOLID\_FILL,12);

flower(n+600,50+i-6\*j,4);

setfillstyle(SOLID\_FILL,13);

flower(n+150,50+i-7\*j,4);

setfillstyle(SOLID\_FILL,12);

flower(n+300,50+i-8\*j,4);

setfillstyle(SOLID\_FILL,13);

flower(n+600,50+i-9\*j,4);

setfillstyle(SOLID\_FILL,12);

flower(n+450,50+i-10\*j,4);

setfillstyle(SOLID\_FILL,12);

flower(n+150,50+i-11\*j,4);

setfillstyle(SOLID\_FILL,13);

flower(n+450,50+i-12\*j,4);

setfillstyle(SOLID\_FILL,13);

flower(n+300,50+i-13\*j,4);

setfillstyle(SOLID\_FILL,12);

flower(n+150,50+i-14\*j,4);

setfillstyle(SOLID\_FILL,12);

flower(n+600,50+i-15\*j,4);

setfillstyle(SOLID\_FILL,13);

flower(n+300,50+i-16\*j,4);

setfillstyle(SOLID\_FILL,12);

flower(n+150,50+i-17\*j,4);

setfillstyle(SOLID\_FILL,12);

flower(n+750,50+i-18\*j,4);

setfillstyle(SOLID\_FILL,13);

flower(n+900,50+i-19\*j,4);

setfillstyle(SOLID\_FILL,12);

flower(n+150,50+i-20\*j,4);

}

void time()

{

setcolor(WHITE);

settextstyle(0,HORIZ\_DIR,3);

sprintf(s,"TIME: %d",t);

outtextxy(750,10,s);

}

int main()

{

initwindow(1000,700, "butterfly butterfly");

readimagefile("flutter3.jpg",0,0,1000,700);

delay(1000);

starting();

setcolor(WHITE);

settextstyle(0,HORIZ\_DIR,3);

sprintf(s,"SCORE: %d",scr);

outtextxy(10, 10, s);

setfillstyle(SOLID\_FILL,9);

bar(0,0,1400,700);

for(; flag!=1;)

{

for(i=0,j=100; i<=2600; i+=50,p++)

{

readimagefile("c.jpg",0,0,1000,700);

score\_again();

time();

leaves();

delay(50);

if(ismouseclick(WM\_LBUTTONDOWN))

{

score();

if(flag==1)

break;

}

if(p%5==0)

t++;

if(t==10)

{

flag=1;

break;

}

}

n=rand()%250;

}

if(scr>=5)

{

readimagefile("lb1.jpg",0,0,1000,700);

settextstyle(4,HORIZ\_DIR,8);

setbkcolor(COLOR(218,174,111));

outtextxy(280, 290, "LEVEL 2");

delay(3000);

lvl2(scr);

}

else

{

readimagefile("8.jpg",0,0,1000,700);

setcolor(0);

setbkcolor(WHITE);

settextstyle(4,HORIZ\_DIR,7);

outtextxy(190, 250, "Game over :(");

delay(3000);

writename(scr);

}

while (!kbhit( ))

{

delay(200);

}

return 0;

}

// level –II:

#include "graphics.h"

#include"myheader.h"

void butterfly1(int x,int y)

{

setfillstyle(SOLID\_FILL,COLOR(43,51,162));

pieslice(x,y,110,250,80);

pieslice(x,y,290,70,80);

setfillstyle(SOLID\_FILL,COLOR(79,88,208));

pieslice(x,y,110,250,60);

pieslice(x,y,290,70,60);

setfillstyle(SOLID\_FILL,COLOR(44,44,237));

pieslice(x,y,110,250,45);

pieslice(x,y,290,70,45);

setfillstyle(SOLID\_FILL,COLOR(0,0,196));

pieslice(x,y,110,250,30);

pieslice(x,y,290,70,30);

setcolor(1);

setfillstyle(SOLID\_FILL,COLOR(9,1,52));

fillellipse(x,y,8,35);

fillellipse(x,y-37,9,9);

}

void lvl2(int marks)

{

int x,y,i,j,p=0,scr=0,z=0,flag=0,n,x1,x2,x3,x4,p1,p2,p3,p4,f1,f2,f3,f4,c1,c2,c3,c4,i1=0,pscr=0;

char s[20],b;

readimagefile("cloud.jpg",0,0,1000,700);

for(;;)

{

f1=0;

f2=0;

f3=0;

f4=0;

p1=270;

p2=270;

p3=420;

p4=420;

pscr=0;

x1=rand()%1000;

//c1=rand()%2;

if(x1<=500)

f1=1;

x2=rand()%1000;

//c2=rand()%2;

if(x2<=500)

f2=1;

x3=rand()%1000;

//c3=rand()%2;

if(x3<=500)

f3=1;

x4=rand()%1000;

//c4=rand()%2;

if(x4<=500)

f4=1;

for(; p1<=700 && p2<=700 && p3>=0 && p4>=0 ;)

{

readimagefile("cloud.jpg",0,0,1000,700);

setcolor(15);

settextstyle(4,HORIZ\_DIR,5);

sprintf(s,"SCORE: %d",scr);

outtextxy(10, 10, s);

if(f1==1)

{

leaves1(x1,p1);

x1=x1+5;

p1=p1+5;

}

else if(f1==0)

{

leaves1(x1,p1);

x1=x1-5;

p1=p1+5;

}

if(f2==1)

{

leaves1(x2,p2);

x2=x2+5;

p2=p2+5;

}

else if(f2==0)

{

leaves1(x2,p2);

x2=x2-5;

p2=p2+5;

}

if(f3==1)

{

leaves1(x3,p3);

x3=x3+5;

p3=p3-5;

}

else if(f3==0)

{

leaves1(x3,p3);

x3=x3-5;

p3=p3-5;

}

if(f4==1)

{

leaves1(x4,p4);

x4=x4+5;

p4=p4-5;

}

else if(f4==0)

{

leaves1(x4,p4);

x4=x4-5;

p4=p4-5;

}

delay(20);

if(ismouseclick(WM\_LBUTTONDOWN))

{

getmouseclick(WM\_LBUTTONDOWN, x, y);

z=getpixel(x,y);

if((x>=x1-15&&x<=x1+15)&&(y>=p1-15&&y<=p1+15)&&z==4)

{

f1=2;

x1=-40;

}

else if((x>=x2-15&&x<=x2+15)&&(y>=p2-15&&y<=p2+15)&&z==4)

{

f2=2;

x2=-40;

}

else if((x>=x3-15&&x<=x3+15)&&(y>=p3-15&&y<=p3+15)&&z==4)

{

f3=2;

x3=-40;

}

else if((x>=x4-15&&x<=x4+15)&&(y>=p4-15&&y<=p4+15)&&z==4)

{

f4=2;

x4=-40;

}

if((f1==2||f2==2||f3==2||f4==2)&&z==4)

{

scr++;

pscr++;

//setcolor(10);

butterfly1(x,y);

delay(100);

}

}

if(f1==2&&f2==2&&f3==2&&f4==2)

break;

}

if(pscr<2)

break;

}

lvl3();

readimagefile("8.jpg",0,0,1000,700);

setcolor(1);

setbkcolor(WHITE);

settextstyle(0,HORIZ\_DIR,7);

outtextxy(190, 250, "Game over :(");

delay(3000);

writename(scr+marks);

//dashboard();

while (!kbhit( ))

{

delay(200);

}

}

**Achievement :**

We have learnt and implemented many important features of C language like FILE, Module system coding, function, structure. We also learnt working with graphics and create a colorful game just like we wanted using BGI.

**Limitations:**

1. With BGI function we can’t make the graphics of our game attractive.
2. And delay () function sometimes doesn’t work properly so we have to face the screen blink.
3. When player click on same flower for several times, score increases with every click.

**Conclusion:**

As our first software project, we have learnt a lot of things. And working in graphics is really enjoyable. Though we can’t make the game perfect but we think it will be an enjoyable game.