



Student Name	Atharva shevate
SRN No	202201727
Roll No	02
Program	Computer Engineering
Year	Second Year
Division	E1
Subject	Computer Graphics and Gaming Laboratory
Project No	Project 3

Project 3: Brick Crusher

Project Logic:

The project logic involves setting up a graphics window with defined screen dimensions. Once the window is initialized, the program draws an octagon and a circle on the screen. Following this, a menu is displayed to the user, presenting various transformation options such as translation, rotation, scaling, reflection, and shear. The user is prompted to select a transformation and provide any required parameters, such as translation distances, rotation angles, scaling factors, axis of reflection, or shear factors. Upon receiving the user's input, the program executes the selected transformation on the shapes and relevant elements, modifying their positions, orientations, or sizes accordingly. After the transformation is applied, the screen is updated to display the transformed shapes and elements. Optionally, brief delays may be introduced between transformations to create an animation effect, enhancing the visual experience. Finally, the program waits for user input to close the graphics window and terminate the execution, allowing the user to interact with the transformed shapes and explore the effects of different transformations.

Code:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<stdlib.h>
#include<dos.h>

//for circle
float speed=50;
int sb1=700,sb2=100,sb3=1000,sb4=500;
int f1=25,f2=25;
int x=25,y=25,r=12;
int blx=10,bly=10,bls=10,score1=0;
// for bars
int x2=100,y2=340,rndx,randy;
int x3=200,y3=380,rndx3,randy3;
int x4=300,y4=300,rndx4,randy4;
int x5=200,y5=380,rndx5,randy5;
int gdriver=DETECT,gmode;
int c,a,hl=1,ttr=0;
int stack[5]={383,356,329,302,275},j=4,n=5,k=0,bcol=475;
```

```
int
stack2[12]={383,356,329,302,275,248,221,194,167,140,113,86},n2=0,k2=0,bcol2=575,j2=
0;
void move(int a);
void logo(void);
void coddling(int score1);
void randomstar(void);
void won(void);
void gover(void);
void security(void);
void main2(void);
void mouse(void);
// mouse functions prototypes
void initmouse();
void showmp();
void hidemp();
int getmp(int *button,int *x,int *y);
void setmp();
int screen();
void button(int x1,int y1,int x2,int y2,char str[]);
int click(int x1,int y1,int x2,int y2,char str[]);
int press(int,int,int,int);
int unpress(int,int,int,int);
void sound1();
void instruction();

int main (){
initgraph(&gdriver,&gmode,"C:\\\\Turboc3\\\\bgi");
mouse();
}
void logo(){
    cleardevice();
    setbkcolor (0);
    setttextstyle (7,HORIZ_DIR,4);
    setcolor (10);
    rectangle (90,95,540,135);
    rectangle (130,140,500,185);
    setcolor (13);
    outtextxy (120,95,"Vishwakarma University");
    setcolor (9);
    outtextxy (155,140,"Surjitsingh Rajput");
    setttextstyle (8,HORIZ_DIR,2);
    setcolor (14);
    outtextxy (230,430,"press any to continue");
```

```
    getch();
mouse();
}
void mouse(){
    initgraph(&gdriver,&gmode,"c:\\turbo3\\bgi");
    cleardevice();
    initmouse();
    showmp();
    screen();
}
```

```
void initmouse(){
    union REGS i,o;
    i.x.ax=0x0;
    int86(0x33,&i,&o);
}
```

```
void showmp(){
    union REGS i,o;
    i.x.ax=0x1;
    int86(0x33,&i,&o);
}
```

```
void hidemp(){
    union REGS i,o;
    i.x.ax=0x2;
    int86(0x33,&i,&o);
}
```

```
int getmp(int *button,int *x,int *y){
    union REGS i,o;
    i.x.ax=0x3;
    int86(0x33,&i,&o);
    *button=o.x.bx;
    *x=o.x.cx;
    *y=o.x.dx;
    return 0;
}
```

```
void setmp(){
    union REGS i,o;
    i.x.ax=0x4;
    i.x.cx=150;
    i.x.dx=150;
```

```
    int86(0x33,&i,&o);
}

int screen(){
    settextstyle(1,HORIZ_DIR,7);
    setcolor(14);
    outtextxy(170,8,"Rapid Roll");
    outtextxy(240,350,"Game");
    setmp();
    button(250,100,400,150,"New Game");
    button(250,150,400,200,"Sound");
    button(250,200,400,250,"Instruction");
    button(250,250,400,300,"Developer");
    button(250,300,400,350,"Exit");
    while(1){
        if(click(250,100,400,150,"New Game")==0){
            hidemp();
            cleardevice();
            main2();
            return 0;
        }

        if(click(250,150,400,200,"Sound")==0){
            hidemp();
            cleardevice();
            sound1();
            return 0;
        }
        if(click(250,200,400,250,"Instruction")==0){
            hidemp();
            instruction();
            return 0;
        }
        if(click(250,250,400,300,"Developer")==0){
            hidemp();
            cleardevice();
            logo();
            return 0;
        }
        if(click(250,300,400,350,"Exit")==0)
            exit(0);
    }
}
```

```
void button(int x1,int y1,int x2,int y2,char str[]){
    int xc,yc,i1=0,l1=0;
    while(i1<strlen(str)){
        l1+=4;
        i1++;
    }
    xc=(x2-x1)/2+x1-l1;
    yc=(y2-y1)/2+y1;
    unpress(x1,y1,x2,y2);
    settextstyle(0,0,0);
    setcolor(11);
    outtextxy(xc,yc,str);
}
```

```
int click(int x1,int y1,int x2,int y2,char str[]){
    int button,x,y;
    int xc,yc,i1=0,l1=0;
    while(i1<strlen(str)){
        l1+=4;
        i1++;
    }
    xc=(x2-x1)/2+x1-l1;
    yc=(y2-y1)/2+y1;
    getmp(&button,&x,&y);
    if( (x>x1 && x<x2) && (y>y1 && y<y2) && button==1){
        hidemp();
        press(x1,y1,x2,y2);
        setcolor(GREEN);
        settextstyle(0,0,0);
        outtextxy(xc,yc,str);
        showmp();
        while((button==1))
            getmp(&button,&x,&y);
        hidemp();
        unpress(x1,y1,x2,y2);
        showmp();
        setcolor(GREEN);
        settextstyle(0,0,0);
        outtextxy(xc,yc,str);
        for(i1=50;i1<500;i1=i1+50){
            delay(10);
            sound(i1+200);
        }
        showmp();
    }
}
```

```
        nosound();
        setcolor(11);
        settextstyle(0,0,0);
        outtextxy(xc,yc,str);
        return 0;
    }else return 1;
}

int unpress(int x1,int y1,int x2,int y2){
    setlinestyle(0,1,1);
    setfillstyle(3,12);
    bar(x1,y1,x2,y2);
    setcolor(WHITE);
    line(x1,y1,x2,y1);
    line(x1,y1,x1,y2);
    line(x1,y2,x2,y2);
    line(x2,y1,x2,y2);
    return 0;
}

int press(int x1,int y1,int x2,int y2){
    setlinestyle(0,1,1);
    setfillstyle(0,3);
    bar(x1,y1,x2,y2);
    setcolor(0);
    line(x1,y1,x2,y1);
    line(x1,y1,x1,y2);
    setcolor(WHITE);
    line(x1,y2,x2,y2);
    line(x2,y1,x2,y2);
    return 0;
}

void sound1(){
    int ssb;
    settextstyle(1,HORIZ_DIR,2);
    outtextxy (200,200,"Press 1 for Sound on");
    outtextxy (200,100,"Press 0 for Sound off");
    scanf ("%d",&ssb);
    if (ssb==1){
        main2();
    }else if (ssb==0){
        sb1=0;
        sb2=0;
    }
}
```

```
        sb3=0;
        sb4=0;
        main2();
    }else
        mouse();
    getch();
}

void instruction(){
    char string[200];
    closegraph(); // Close the graphics window
    initgraph(&gdriver,&gmode,"c:\\turbo3\\bgi"); // Reinitialize the graphics mode
    setbkcolor(BLACK); // Set background color to black
    cleardevice(); // Clear the screen with black background

    // Additional instructions
    printf("1) There are two keys to move the Ball: 'Left cursor movement key' for left
and 'Right cursor movement key' for right.\n");
    printf("2) There is a limitation of the Ball's movement from right to left to the
rectangle lines.\n");
    printf("3) Caution: If your Ball touches the top or bottom of the rectangle lines, your
Bonus Ball count will decrease.\n");
    printf("4) There is a 'Stack' for keeping Bonus Balls with green color on the right
side, and there is also a 'Queue' for putting Score Balls with yellow color. One Ball can
be obtained with 10 score.\n");
    printf("5) After getting 10, 20, 30, and so on, score, the speed of the Game will
increase respectively.\n");

    printf("\n\nPress any key to go back...");
    getch();
    mouse();
}

void main2(){
    int gdriver=DETECT,gmode;
    initgraph(&gdriver,&gmode,"c:\\turbo3\\bgi");
    cleardevice();
    rndx+=x2;
    rndy+=y2;

    rndx3+=x3;
    rndy3+=y3;
```

```
rndx4+=x4;
rndy4+=y4;
```

```
rndx5+=x5;
rndy5+=y5;
```

```
randomstar();
```

```
coding(score1);
while((c=getch())!='\033'){
    //right
    if (c=='\115') a=1;
    //left
    if (c=='\113') a=2;
    move (a);
}
```

```
}
```

```
void move (int a){
    void score(int f1,int f2,int ttr);
    while(!kbhit()){
        //clear remaining bars
        setfillstyle(0,0);
        bar (rndx5-198,rndy5-378,rndx5+198,rndy5-365);

        //bar coding
        setfillstyle(0,0);
        bar (rndx-25,rndy-3,rndx+25,rndy+3);
        rndy=rndy-1;
        setfillstyle(1,2);
        bar (rndx-25,rndy-3,rndx+25,rndy+3);
        if (rndy==10){
            rndy+=y2;
        }
        setfillstyle(0,0);
        bar (rndx3-25,rndy3-3,rndx3+25,rndy3+3);
        rndy3=rndy3-1;
        setfillstyle(1,2);
        bar (rndx3-25,rndy3-3,rndx3+25,rndy3+3);
        if (rndy3==10){
            rndy3+=y3;
        }
        setfillstyle(0,0);
        bar (rndx4-25,rndy4-3,rndx4+25,rndy4+3);
```

```
    rndy4=rndy4-1;
    setfillstyle(1,2);
    bar (rndx4-25,rndy4-3,rndx4+25,rndy4+3);
    if (rndy4==10){
        rndy4=+y4;
    }

    // food coddling
    setfillstyle(5,14);
    bar3d(blx,bly,blx+bls,bly+bls,0,0);
    //circle coddling
    setcolor (WHITE);
    circle(f1,f2,r);
    setfillstyle(SOLID_FILL,BLACK);
    floodfill(f1,f2,BLACK);

    //clear remaining circle
    bar (rndx5-198,rndy5+18,rndx5+198,rndy5-6);

    setcolor (WHITE);
    circle(f1,f2,r);
    setfillstyle(SOLID_FILL,RED);
    floodfill(f1,f2,WHITE);
    setcolor(0);

    if ((f2==rndy-18) && (f1==75 || f1==85 || f1==95 || f1==105 || f1==115 || f1==125))
    {
        f2=f2-1;
    }
    else if ((f2==rndy4-18) && (f1==275 || f1==285 || f1==295 || f1==305 || f1==315 ||
f1==325))
    {
        f2=f2-1;
    }
    else if ((f2==rndy3-18)&& (f1==175 || f1==185 || f1==195 || f1==205 || f1==215 ||
f1==225)){
        f2=f2-1;
    }else
        f2=f2+1;
    delay(speed);
    score (f1,f2,ttr);
    if (f2==387)
    {
        f2=25;
```

```
ttr++;
}
else if (f2==14){
    f2=25+1;
    ttr++;
}

}
if(a==1)f1=f1+5;
if(a==2)f1=f1-5;

if (f1==10) f1=20;
if (f1==390) f1=385;
}

void score (int f1,int f2,int ttr){
    while (k<n-ttr){
        setcolor (WHITE);
        circle(bcol,stack[k],r);
        setfillstyle(SOLID_FILL,GREEN);
        floodfill(bcol,stack[k],WHITE);
        k++;
    }
    if (f2==383 || f2==14){
        setcolor (BLACK);
        circle(bcol,stack[j],r);
        setfillstyle(SOLID_FILL,BLACK);
        floodfill(bcol,stack[j],BLACK);
        j--;
        sound(sb4);
        delay(50);
        nosound();
        if (j==-1){
            sound(sb2);
            delay(100);
            nosound();
            gover();
        }
    }
    if ((f2==bly || f2+11==bly || f2-11==bly) && (f1==blx+5 || f1+11==blx+5 || f1-11==blx+5)){
        sound(sb3);
        delay(50);
        score1+=2;
    }
}
```

```
    nosound();
    coding (score1);
if (score1==10){
    j2++;
    n2=j2;
    sound(sb1);
    delay(70);
    nosound();
    speed=speed-3.75;
}else if (score1==20){
    j2++;
    n2=j2;
    sound(sb1);
    delay(70);
    nosound();
    speed=speed-3.75;
}else if (score1==30){
    j2++;
    n2=j2;
    sound(sb1);
    delay(70);
    nosound();
    speed=speed-3.75;
}else if (score1==40){
    j2++;
    n2=j2;
    sound(sb1);
    delay(70);
    nosound();
    speed=speed-3.75;
}else if (score1==50){
    j2++;
    n2=j2;
    sound(sb1);
    delay(70);
    nosound();
    speed=speed-3.75;
}else if (score1==60){
    j2++;
    n2=j2;
    sound(sb1);
    delay(70);
    nosound();
    speed=speed-3.75;
```

```
}else if (score1==70){
    j2++;
    n2=j2;
    sound(sb1);
    delay(70);
    nosound();
    speed=speed-3.75;
}else if (score1==80){
    j2++;
    n2=j2;
    sound(sb1);
    delay(70);
    nosound();
    speed=speed-3.75;
}else if (score1==90){
    j2++;
    n2=j2;
    sound(sb1);
    delay(70);
    nosound();
    speed=speed-3.75;
}else if (score1==100){
    j2++;
    n2=j2;
    sound(sb1);
    delay(70);
    nosound();
    speed=speed-3.75;
}else if (score1==110){
    j2++;
    n2=j2;
    sound(sb1);
    delay(70);
    nosound();
    speed=speed-3.75;
}else if (score1==120){
    j2++;
    n2=j2;
    sound(sb1);
    delay(100);
    nosound();
    speed=speed-3.75;
}else if (score1>120){
    won ();
```

```
    }
    randomstar();
    }

    while (k2<n2){
        setcolor (WHITE);
        circle(bcol2,stack2[k2],r);
        setfillstyle(SOLID_FILL,YELLOW);
        floodfill(bcol2,stack2[k2],WHITE);
        k2++;
    }
}

void randomstar(){
    int ic=0,pr=0;
    randomize();
    ic=random(10000)%40;
    ic=ic*bls;
    pr=random(10000)%40;
    pr=pr*bls;
    blx=ic;
    bly=pr;

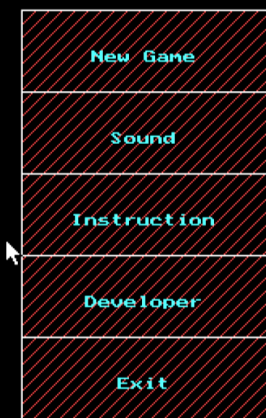
    if ((bly<30) || (bly>350)){
        bly=40;
    }
    if ((blx<30) || (blx>383)){
        blx=20;
    }
}

void coddling (int score1){
    setcolor (WHITE);
    rectangle(0,0,400,400);
    gotoxy (60,5);
    circle (480,70,30);
    setcolor (WHITE);
    printf ("%d ",score1);
//Bonus
    setcolor (7);
    outtextxy (455,165,"BONUS");
    setcolor (WHITE);
    line (500,200,500,400);line (450,200,450,400);line (450,400,500,400);
//Score
    setcolor (7);
```

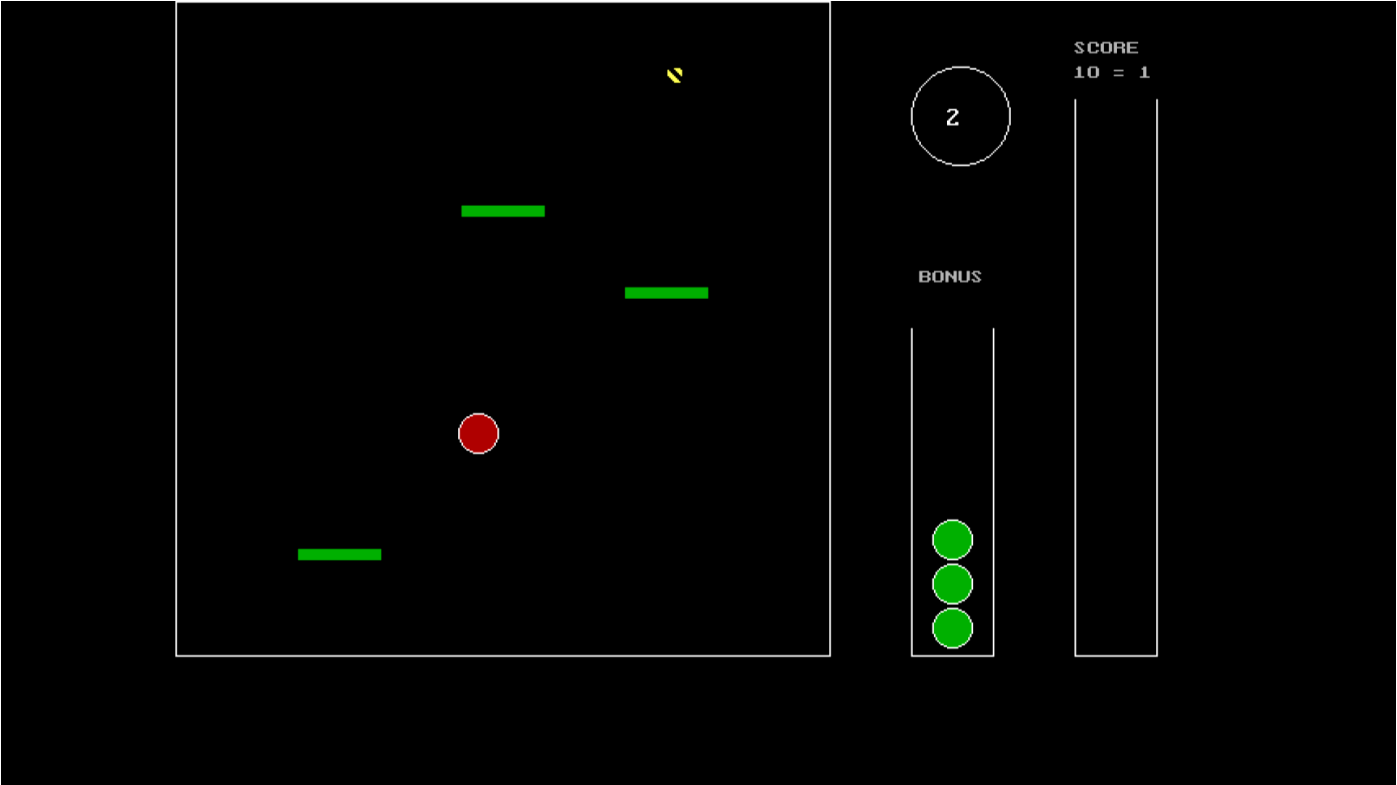
```
    outtextxy (550,25,"SCORE");
    outtextxy (550,40,"10 = 1");
    setcolor (WHITE);
    line (600,60,600,400);line (550,60,550,400);line (550,400,600,400);
}

void won (){
    cleardevice();
    rectangle (40,100,100,135);line (40,100,100,135);line (40,135,100,100);
    settextstyle (7,HORIZ_DIR,5);setcolor (GREEN);
    outtextxy (135,150,"Congratulation!");
    outtextxy (150,200,"You won the GAME");
    settextstyle (7,HORIZ_DIR,1);setcolor (YELLOW);
    outtextxy (300,300,"By: Surjitsingh Rajput");
    setcolor (7);
getch();
exit(0);
}
void gover(){
cleardevice();
    rectangle (40,100,100,135);line (40,100,100,135);line (40,135,100,100);
    settextstyle (7,HORIZ_DIR,5);setcolor (RED);
    outtextxy (200,200,"GAME OVER");
    settextstyle (7,HORIZ_DIR,1);setcolor (YELLOW);
    outtextxy (300,300,"By: Surjitsingh Rajput");
getch();
mouse();
return 0;
}Output:
```

Rapid Roll



Game





Recording

2024-04-25 130127.m

Conclusion:

In conclusion, the project illustrates the versatile application of 2D transformations and filling algorithms in computer graphics. Through transformations like translation, rotation, scaling, reflection, and shear, applied to a polygon and an emoji, it showcases the dynamic manipulation of shapes. Filling algorithms such as flood fill and scanline fill enhance the visual appeal by adding color and detail. The project encourages experimentation and creativity while highlighting the synergy between algorithms and visual expression, making it a compelling exploration of digital design.



VISHWAKARMA
UNIVERSITY
Maximising Human Potential