Vidyavardhini's College of Engineering and Technology
Department of Artificial Intelligence & Data Science

Report of Mini-project On

ROTATING FAN

Submitted in partial fulfilment of the requirements of the Mini project in the
Computer Graphics Lab of
Semester III, Second Year Artificial Intelligence & Data Science

by
Pratham Nagar
Rahul Yadav
Priyanka Dhuri
Tanvi Surve

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science Engineering

(A.Y. 2023-24)

Vidyavardhini's College of Engineering and Technology
Department of Artificial Intelligence & Data Science

CERTIFICATE

This is to certify that the Mini Project entitled **"**Rotating Fan**"** is submitted by Pratham Nagar, Rahul Yadav, Priyanka Dhuri, Tanvi Surve for the subject of Computer Graphics Lab in the Department of Artificial Intelligence & Data Science as a record of work done by him/her under our supervision and guidance.

Internal Examiner

Guide
Asst. Prof. Neha Raut

Contents

1. Introduction

1.1 Basic idea of the project

In this project, we have designed a C program to present a rotating fan. This program is created using C programming language. It consists of header files such as graphics.h,stdlib.h,studio.h,conio.h,dos.h.

The stdio.h and conio.h header files are commonly used in C programming language, the graphics.h file is used for animation purpose, the dos.h file is used for the working purpose of delay()function used in the code and stdlib.h file is the header for the General Purpose Standard

1.2 Technology Used

In computer graphics, creating a rotating fan in a 2D environment typically involves using various techniques and technologies, such as programming languages, libraries, and algorithms. Here's a high-level overview of how you can achieve a rotating fan in a 2D computer graphics context:

Programming Language: You would typically use a programming language to create your 2D graphics application. Popular languages for this purpose include JavaScript (with HTML5 Canvas), Python (with libraries like Pygame), or C++ (with libraries like SFML or SDL).

Graphics Library: To handle the rendering and manipulation of 2D graphics, you would use a graphics library or framework. For example, in JavaScript, you can use the HTML5 Canvas API, while in Python, you can use Pygame, and in C++, you can use libraries like SFML or SDL

Transformation: To achieve rotation, you can apply 2D geometric transformations, such as rotation transformations. You'll specify an angle by which to rotate the fan, and the graphics library will handle the transformation of the fan's elements accordingly.

2. Computer Graphics Concepts

2.1 Graphic Functions used

Functions used in the program:

1) **Ellipse() function** : This function can be used to draw different shapes,change colors and many more. Start_angle(stangle) is the starting point of the angle and end_angle(endangle) is the ending point of the angle. The value of angle can vary from 0 to 360 degrees.

2) **Setfillstyle() function** : The header file graphics.h contains setfillstyle()function which sets the current fill pattern and fill color.

3)**Pieslice() function**: This function draws and fills a pie slice with centre at (x,y) and given radius(r). The slice travels from (stangle) and (endangle) which are the starting and ending angles for the pie slice. The angles for pie slice are given in degrees and are measured clockwise.

4) **Delay() function**: This function is used to delay the execution of a command by a certain specified amount of time.

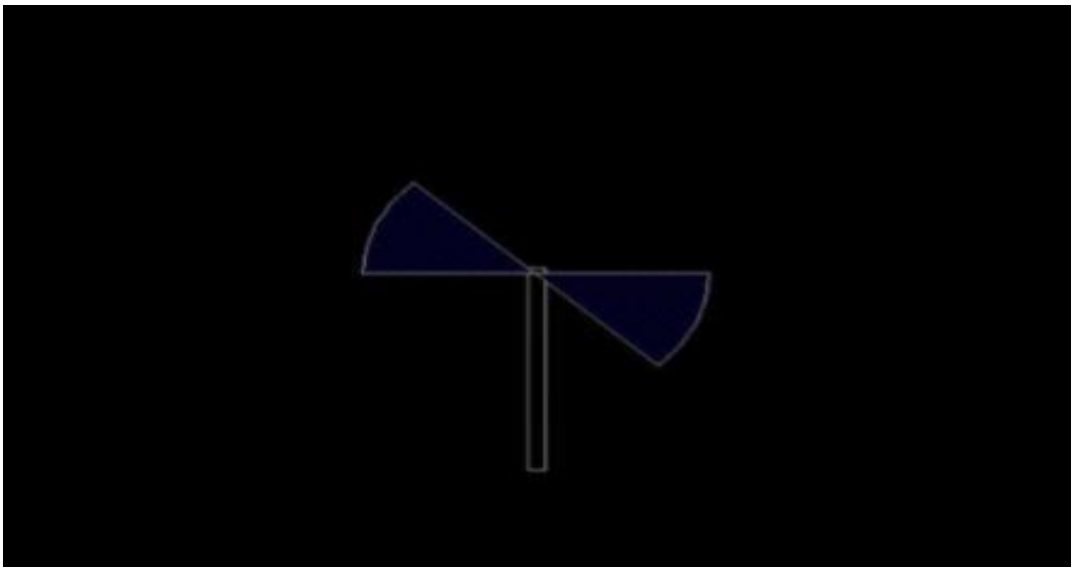
3. Code and Result

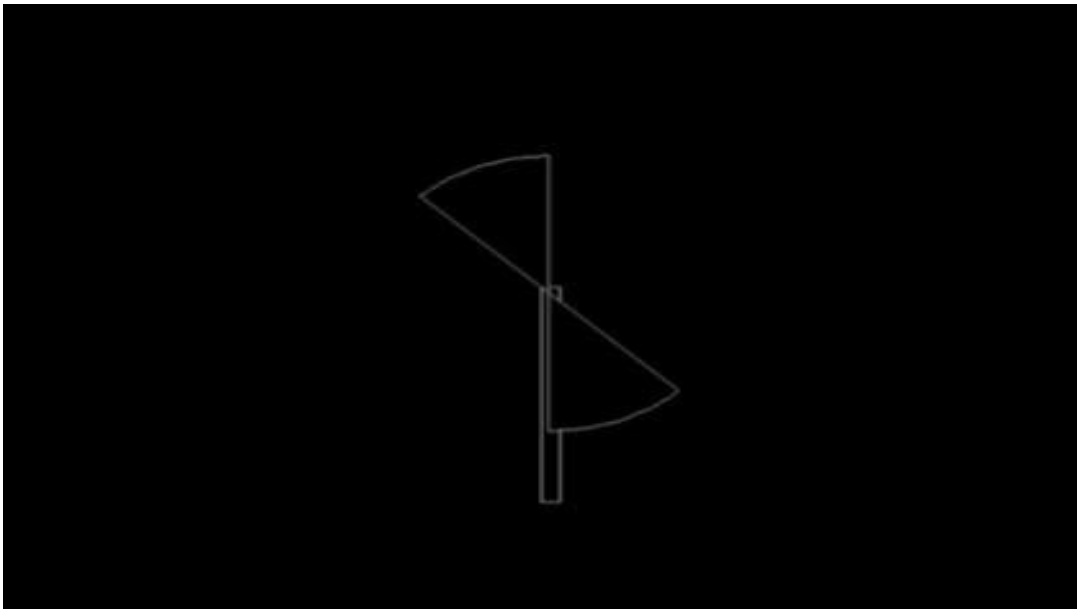
3.1 Code:

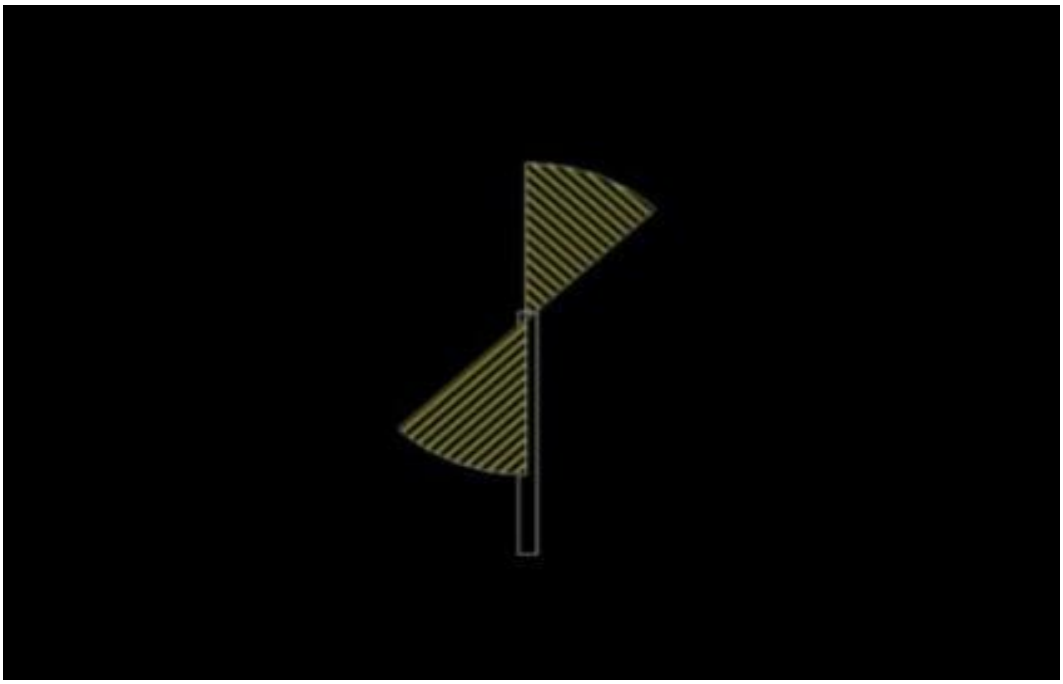
```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <dos.h>
int main(void) {
/* request auto detection */
int gdriver = DETECT, gmode, errorcode; int i, midx, midy;
int stangle1 = -45, endangle1 = 0, radius = 100; int stangle2 = 135, endangle2 = 180;
/* initialize graphics and local variables */
initgraph(&gdriver, &gmode, "C:/TURBOC3/BGI");
/* read result of initialization */ errorcode = graphresult(); if (errorcode != grOk) { /* an error occurred */
printf("Graphics error: %s\n", grapherrormsg(errorcode));
printf("Press any key to halt:");
getch();
exit(1); /* terminate with an error code */
}
/* mid position of x in x-axis */ midx = getmaxx() / 2; /* mid position of y in y-axis */ midy = getmaxy()/ 2;
for (i = 0; i < 400; i++) {
/* start and end angle of fan's first wing */ if (endangle1 == 360) { stangle1 = -45; endangle1 = 0;
} /* start and end angle of fan's second wing */ if (endangle2 == 360) { stangle2 = -45; endangle2 = 0; } /* clears graphic device */ cleardevice(); stangle1 = stangle1 + 45; stangle2 = stangle2 + 45; endangle1 = endangle1 + 45; endangle2 = endangle2 + 45;
/* fan stand */
```

```
rectangle(midx - 5, midy - 5, midx + 5, midy + 150);
/* draws first wing of fan */ setfillstyle(SLASH_FILL, i % 15);
pieslice(midx, midy, stangle1, endangle1, radius);
/* draws second wing of fan */ setfillstyle(BKSLASH_FILL, i % 15); pieslice(midx, midy,
stangle2, endangle2, radius);
/* sleep for 40 millisecond */
delay(75);
}
/* clean up */
getch();
/* deallocate memory allocated for graphic screen */ closegraph();
return 0;
}
```

3.2 Snapshots of Result







4. Conclusion

Through this project, we were able to investigate basic ideas in computer graphics, such as animation, rendering, and transformations. We improved our C programming abilities and developed a deeper understanding of 2D and 3D graphics by implementing a rotating fan. It's evidence of the inventiveness and technical mastery possible in computer graphics.

Video of the Output:

Youtube video link –

https://youtu.be/T1lNRpIl6MU?si=hOL6vncavJVf1Gv4