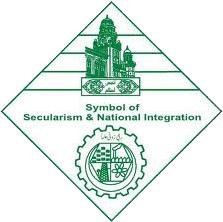
**Anjuman-I-Islam**

**M.H. Saboo Siddik Polytechnic**



COMPUTER GRAPHICS (CGR)

MICROPROJECT

COMPUTER ENGINEERING

DEPARTMENT

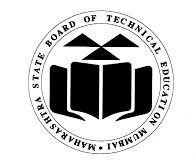
CO-3I

TITLE Fan

YEAR: 2022-23 **Prepared by:**

* 210413 : Mujahid Abdul Qadeer
* 210414 : Patel Areeb
* 210418 : Shaikh M . Hassan
* 210421 : Dhukka Haris

**Guided by**: Prof. Mohd Zaid Sir



**MAHARASHTRA STATE**

**BOARD OF TECHNICAL EDUCATION**

# Certificate

This is to certify that Mr . Mujahid Abdul Qadeer Roll no.

210413 of second semester of Diploma in Computer

Engineering of institute M.H. Saboo Siddik Polytechnic(code:0002) has completed microproject satisfactorily in the subject: CGR (22318) for the academic year 2022-23 as prescribed in the curriculum.

Place: Byculla, Mumbai Enrollment no:2100020120

Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Exam seat no:

## Signature Signature Signature

## Project guide H.O.D Principal

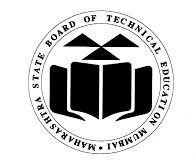


SEAL OF



INSTITUDE





**MAHARASHTRA STATE**

**BOARD OF TECHNICAL EDUCATION**

# Certificate

This is to certify that Mr . Patel Areeb Roll no. 210414 of second semester of Diploma in Computer Engineering of institute M.H. Saboo Siddik Polytechnic(code: 0002) has completed microproject satisfactorily in the subject: CGR (22318) for the academic year 2022-23 as prescribed in the curriculum.

Place: Byculla, Mumbai Enrollment no:2100020078

Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Exam seat no:

**Signature Signature Signature**

## Project guide H.O.D Principal

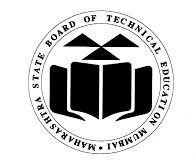


SEAL OF



INSTITUDE





**MAHARASHTRA STATE**

**BOARD OF TECHNICAL EDUCATION**

# Certificate

This is to certify that Mr . Shaikh M . Hassan Roll no.

210418 of second semester of Diploma in Computer

Engineering of institute M.H. Saboo Siddik Polytechnic(code:0002) has completed microproject satisfactorily in the subject: CGR (22318) for the academic year 2022-23 as prescribed in the curriculum.

Place: Byculla, Mumbai Enrollment no:2100020085

Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Exam seat no:

## Signature Signature Signature

## Project guide H.O.D Principal

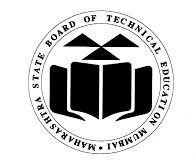


SEAL OF



INSTITUDE





**MAHARASHTRA STATE**

**BOARD OF TECHNICAL EDUCATION**

# Certificate

This is to certify that Mr . Dhukka Haris Roll no.

210411 of second semester of Diploma in Computer

Engineering of institute M.H. Saboo Siddik Polytechnic(code:0002) has completed microproject satisfactorily in the subject: CGR (22318) for the academic year 2022-23 as prescribed in the curriculum.

Place: Byculla, Mumbai Enrollment no:2100020128

Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Exam seat no:

## Signature Signature Signature

## Project guide H.O.D Principal



SEAL OF



INSTITUDE



# Acknowledgment

We wish to express our profound gratitude to our guide Mr. Mohd Zaid Sir who guided us endlessly in the framing and completion of the micro project. He

guided us on all the main points in that micro project. We are indebted to his/her constant encouragement, cooperation, and help. It was his/her enthusiastic

support that helped us in overcoming various obstacles in the micro-project.

We are also thankful to our Principal, HOD, faculty members and classmates of

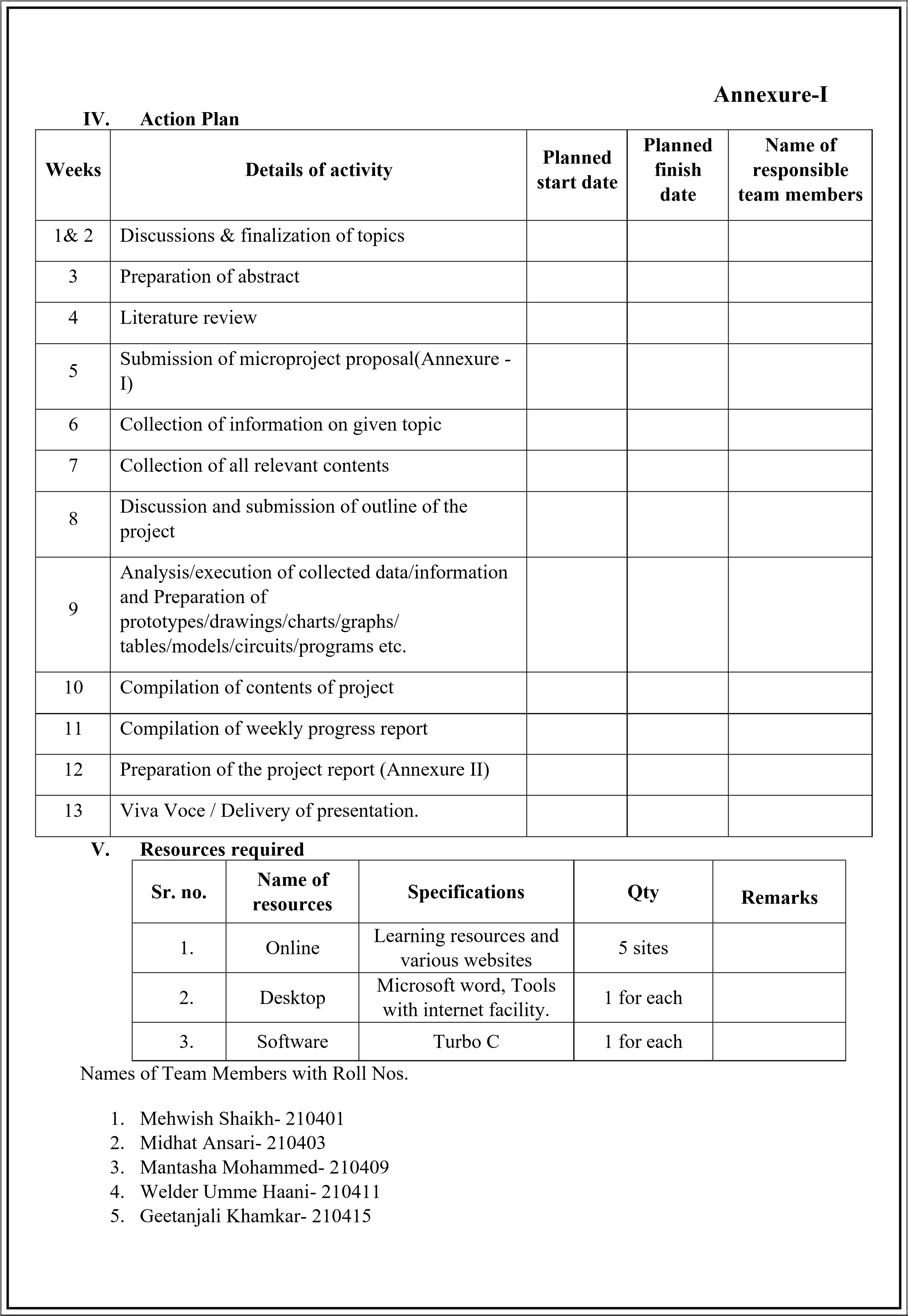
Computer Engineering department for extending their support and motivation in the completion of this micro-project.

Names of Team Members with Roll Nos.

1. Mujahid Abdul Qadeer – 210431
2. Patel Areeb - 210414
3. Shaikh M . Hassan - 210418
4. Dhukka Haris - 210421

|  |  |
| --- | --- |
|  | **Annexure-I *Microproject proposal***  **Title of microproject : Fan** |
| **I.** | **Aims/Benefits of microproject**  Computer graphics help us create realistic images that are easier to see and understand. They can create images that are more appealing to the eye. They can help reduce the time it takes to produce an image. They can help produce images that are more efficient. They can help produce images that are more accurate. |
| **II.** | **Course outcomes addressed** |
|  | * Manipulate visual and geometric information of images. |
|  | * Implement standard algorithm to draw various graphics objects using c program. * Develop a program to create curve using algorithms. |
| **III.** | **Proposed methodology**  1.To search the information about the project. (Collect relevant data from different sources (books/internet/market/suppliers/experts and others through surveys/interviews etc.).   1. To collect all relevant content / materials to complete the project. 2. To prepare the report of micro project. 3. To deliver presentation/ appear for viva-voice |
| i. | Discussion of the given topic among group members. |
| ii. | Literature survey |
| iii. | Submission of project proposal |
| iv. | Analysis of data |
| v. | Work divided among group members |
| vi. | Compilation of content |
| vii. | Representation |
| viii. | Editing the content as per the instructions |
| ix. | Report Preparation  x. Viva and presentation |

|  |
| --- |
|  |



**Annexure-II *Microproject Report***

**Title of Micro-project: Moving Car**

1. **Rationale**

Graphics is an important aspect for Computer Engineering. In this particular subject implementation of algorithms was done. This course consider methods for object designing, transformation , scan conversions visualisation and modelling of real world. Emphasis is on the course on understanding how various elements that underlie computer graphics interact in graphics software systems.

1. **Aims/Benefits of microproject**

Developing programs using core graphical concepts. Computer Graphics is used where a set of images needs to be manipulated or the creation of the image in the form of pixels and is drawn on the computer. Computer Graphics can be used in digital photography, film, entertainment, electronic gadgets, and all other core technologies which are required. It is a vast subject and area in the field of computer science. Computer Graphics can be used in UI design, rendering, geometric objects, animation, and many more.

1. **Course outcomes achieved**

While preparing the microproject, we learned about various algorithms to draw various graphics object and filling it using seed fill algorithms.

1. **Literature review**

While preparing this microproject we have learnt about applications of computer graphics and also implemented some features of computer graphics.

Multiple websites used by us include:

[www.geeksforgeeks.com](http://www.geeksforgeeks.com/)

[www.tutorialspoint.com](http://www.tutorialspoint.com/)

1. **Actual Methodology Followed**

We were assigned with the microproject topic and time was assigned to us to complete the project in 13 weeks. All team members worked together in these 13 weeks together in order to complete this microproject. Data was collected according to our topic. Coding was implemented and finally at last technical report was prepared.

1. **Actual resources used**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. no.** | **Name of resources** | **Specifications** | **Qty** | **Remarks** |
| 1. | Software | Turbo C | 1 for each |  |
| 2. | Websites | [www.geeksforgeeks.com,](http://www.geeksforgeeks.com/) [www.tutorialspoint.com,](http://www.tutorialspoint.com/) | 1 for each |  |

|  |  |
| --- | --- |
| **VII.** | **Outputs of the microproject** |
| **VIII.** | **Skills developed/ Learning outcomes** |
| - | Derive: Derive different possible solutions creatively. |
| - | Data Collection: Collect relevant data from different sources (books/the internet/the market/suppliers/experts and others through surveys/interviews) |
| - | Designing- Designing microproject with minimum required resources and at low cost. |
| - | Teamwork- Learning to work in team and boost individual confidence. |
| - | Time management- Completion of microproject as scheduled. |
| - | Technical writing- Preparing a report of proposed plan and report. |
| - | Presentation and communication skills: Giving working model presentation of the micro project. |
| - | Confidence: Confidently, answer the questions asked about the project. |
| - | Concepts related to creation of object in graphic was understood in detail. |
| - | Improved debugging skills |
|  | **Applications of this microproject**   * **Computer Graphics are used for** an **aided design for engineering and architectural system-** These are used in electrical automobiles, electromechanical, mechanical, electronic devices. For example gears and bolts. * **Computer Art –** MS Paint. * **Presentation Graphics –** It is used to summarize financial statistical scientific or economic data. For example- Bar chart, Line chart. * **Entertainment-** It is used in motion pictures, music videos, television gaming. * **Education and training-** It is used to understand the operations of complex systems. It is also used for specialized system such for framing for captains, pilots and so on. |

Title of microproject : Fan

Course outcomes achieved:

* 1. Write a C program to draw and rotate a fan
  2. Use user defined functions to fill tyres of the car

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Course title : Computer Graphics | Code : 22318 |  |

* 1. Implementing built in functions of graphics.h header file and some user defined functions



**Comments/Suggestions about teamwork/leadership/interpersonal communication**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Name and designation of teacher: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Title of microproject : Rotate a Fan

Course outcomes achieved:

* 1. Write a C program to draw and move a car
  2. Use user defined functions to fill tyres of the car

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Course title : Computer Graphics |  | Code : 22318 |

* 1. Implementing built in functions of graphics.h header file and some user defined functions



**Comments/Suggestions about teamwork/leadership/interpersonal communication**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Name and designation of teacher: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Title of microproject: Rotate Fan

Course outcomes achieved:

* 1. Write a C program to draw and rotate a car
  2. Use user defined functions to fill tyres of the car

|  |  |
| --- | --- |
| Course title : Computer Graphics Code : 22318 |  |
|  |  |
|  |  |

* 1. Implementing built in functions of graphics.h header file and some user defined functions



**Comments/Suggestions about teamwork/leadership/interpersonal communication**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Name and designation of teacher: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |
| --- | --- |
|  |  |
| Course title : Computer Graphics | Code: 22318 |

Title of microproject: Fan

Course outcomes achieved:

* 1. Write a C program to draw and rotate a fan
  2. Use user defined functions to fill tyres of the car
  3. Implementing built in functions of graphics.h header file and some user defined functions



**Comments/Suggestions about teamwork/leadership/interpersonal communication**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Name and designation of teacher: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Dated signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

TOPIC : Fan

### *INTRODUCTION*

In the assigned microproject i.e. Rotate a Fan we have drawn many Fan using in built functions like circle, line, settextstyle, setcolor, outtextxy, etc. Some user defined functions like flood fill, boundary fill, DDA line drawing, Bresenham circle generation concepts was used in our microproject.

We have made a switch case program to rotate a car. It will completely depend on the user as how they want to see the rotate a fan . User will enter the number as per his/her choice. As user enters number 1 fan will display and if user enters 2 then again two choice from user enter either 1 or 2 . if user enter the number 1 rotate the fan without RGB and If user enter the number 2 Fan rotate with RGB other wise user enter the number 3 then exit .

### MAIN PROGRAM

#### ALGORITHM

Step 1: Start

Step 2: Print ‘CGR microproject, topic and name of group members’ using printf(), settextstyle, setcolor, outtextxy functions

Step 3: As user press any key they will move to next page as getch() is used where message is displayed using printf: ‘Press enter to see the car moving’

Step 4: Input choice from user using scanf

Step 5: Using built in functions line(), circle(), rectangle() to draw the car Step 6: Call flood\_fill() and boundary\_fill() user defined functions to fill tyres of a car

Step 7: If user enters 1

Car will move from left to right till i<=420, increment i by 1

Elseif user enters 2

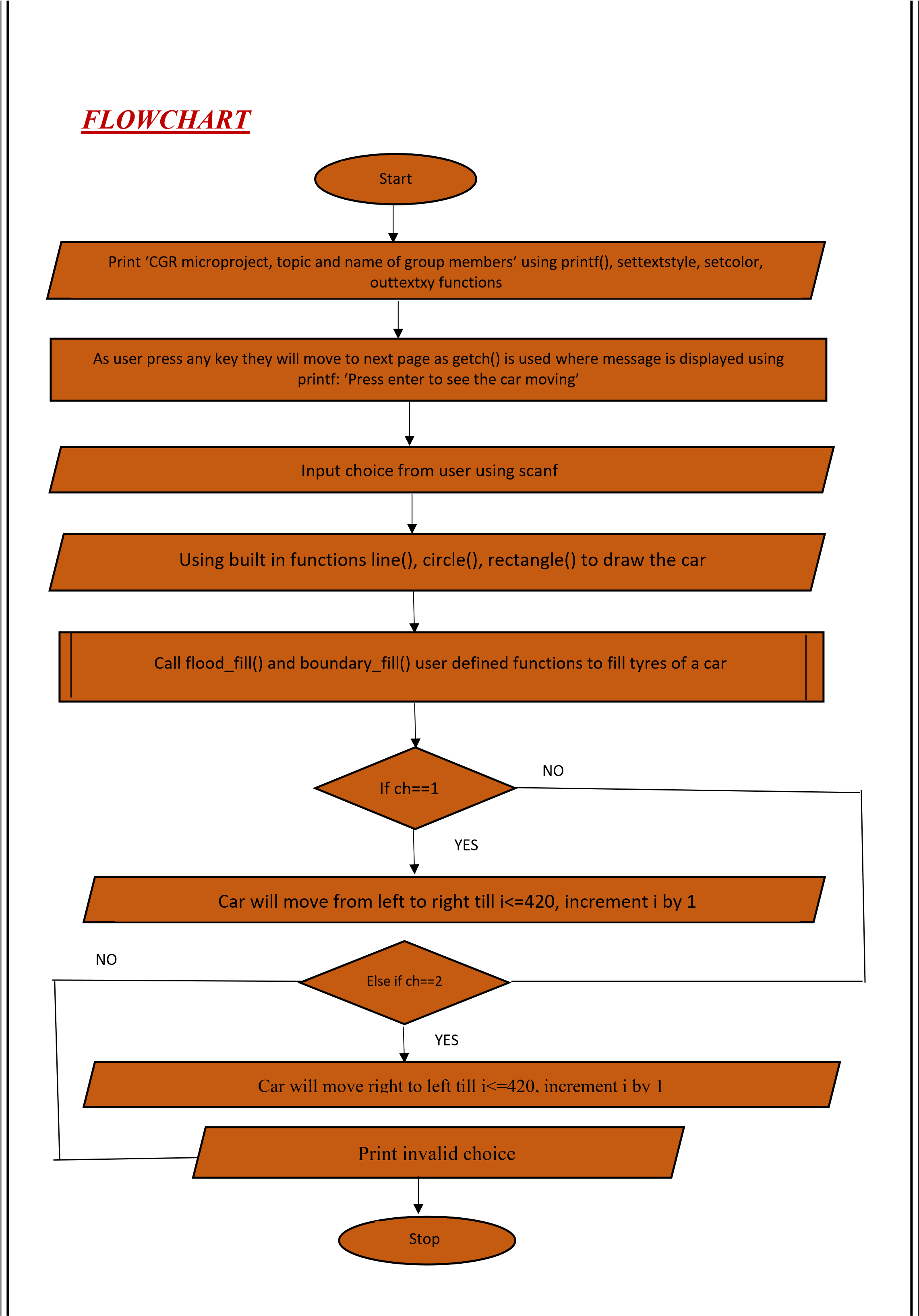
Car will move right to left till i<=420, increment i by 1

Else

Print invalid choice

Step 8: Print ‘Thank you’ using printf(), settextstyle(), setcolor(), etc

Step 9: Stop



• **Polygon filling:**

Filling in the polygon means highlighting all the pixels which lie inside the polygon with any color other than the background color.

There are 2 basic approaches to fill the polygon.

* Seed Fill Approach
* Scan Line Approach

##### • Seed fill approach

In this approach we start from a given seed point known to be inside the polygon and highlight outwards from this point until we encounter the boundary pixels.

There are 2 types of Seed Fill Approach:

* Boundary Fill Algorithm
* Flood Fill Algorithm

##### • Flood fill

1. With the disadvantage of Boundary fill algorithm, one can think that we should include different boundary\_colors in the conditions & then it will work
2. e.g. if((getpixel(x,y)!=b\_color1) && (getpixel(x,y)!=b\_color2) && (getpixel(x,y)!=b\_color3 &&(getpixel(x,y)!=f\_color))).
3. Here the approach and logic is correct but by this method we will make the algorithm less efficient by increasing algorithm’s time & space complexity. 4. In Flood fill, first we initialize the polygons inside pixel’s current color, normally it is referred as OLD\_COLOR or INTERIOR\_COLOR.

5. After initialization, FFA will check whether the seed pixel’s color is equal to OLD\_COLOR? – If yes then it will paint the seed pixel with fill color (NEW\_COLOR) & move to neighboring pixels, – if not then it will do nothing & simply move to the neighboring pixels.

* Algorithm for main function:

Step 1: Start

Step 2: Call circle (xc, yc, r)

Step 3: Call flood-fill (x, y, old\_c, new\_c)

Step 4: Stop

* Flowchart for main function:

S

tart

Call circle(xc,yc,r)

Call

flood\_fill(x,y,oldc,newc)

S

top

* Algorithm for flood fill function

Step 1: Start

Step 2: Initialize old color with black

Step 3: If current pixel=old\_c

Then go to Step 4 Else go to Step 9

Step 4: Paint current pixel with new color

Step 5: Call Flood - fill(x+1,y,old\_c,new\_c)

Step 6: Call Flood - fill(x, y+1, old\_c,new\_c)

Step 7: Call Flood - fill(x-1, y, old\_c,new\_c) Step 8: Call Flood - fill(x, y-1, old\_c,new\_c)

Step 9: Return

Flowchart for Flood fill function:

Stop

Start

Initialize flood\_fill with black

If getpixel =

oldc

Paint pixel(x,y)

Call flood\_fill (x+1,y,oldc,newc)

Call flood\_fill (x+1,y,oldc,newc)

Call flood\_fill (x

-

)

,y,oldc,newc

1

Call flood\_fill (x,y

-

)

,oldc,newc

1

##### 

##### Boundary Fill

• The basic idea of area filling is as:

* Start from any arbitrary known point (seed point) inside polygon, set it to fill colour.

Examine neighboring pixels of seed pixel to check whether boundary pixel is reached.

* If boundary pixels are not reached set fill colour to the pixels (i.e, color those pixels )and continue the process until boundary pixels are reached.

• Here, boundary of area (polygon) is specified in a single colour.

❖ There are two methods for proceeding to neighboring pixels from current test point .

1. 4-connected method, and
2. 8-connected method.

• Boundary fill algorithm can be implemented by using these two methods.

1. **4-connected Method :** In this 4 neighbouring points of a current test point are tested.

These are pixel positions that are right, left, above and below of current pixel as shown in Fig. 2.27. This process will continue until we find a boundary with different color.

1. **8-connected Method :** Here 8 neighbouring pixels of current test pixel are tested. These pixel positions are left, right, above, below and 4-diagonal positions of current pixel as shown in following Fig. 2.28. This process will continue until we find a boundary with different color.

* Main function algorithm

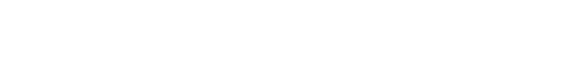
Step 1: Start

Step 2: Call circle (xc, yc, r)

Step 3: Call boundary\_fill (x,y, b\_cod,f\_col) Step 4: Stop.

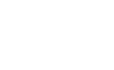
* Flowchart of main function

Start



Call Circle (Xc, Yc, r)

Call Boundary\_fill (X, Y, b\_col, f\_col)



Stop



•

Boundary Fill Algorithm

Step 1: Start

Step 2: Initialize boundary color with WHITE

Step 3: If current pixel color!= boundary color AND new color then goto

step 4 else go to step 9

Step 4: Paint current pixel with new color

Step 5: Call boundary\_fill (x+1,

y, b\_col, f\_col)

Step 6: Call boundary\_fill (x, y+l, b\_col, f\_col)

Step 7: Call boundary\_fill (x

-

, y, b\_col. f\_col

)

1

Step 8: Call boundary\_fill (x, y

-

1

, b\_col. f\_col

)

Step 9: Return.

•

Flowchart of boundary fill

zz

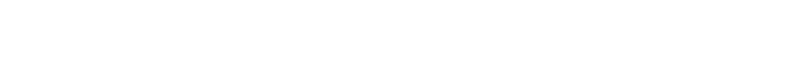


Boundary\_fill

(

X, Y, b\_col, f\_col

)



Initialize b\_col WHITE

If

getpixel (X, Y) !=

b\_col && f\_col



Call Boundary\_fill

X, Y+1, b\_col, f\_col

)

(



Call Boundary\_fill

X

(

-

1

)

, Y, b\_col, f\_col



TRUE

FALSE

Call

Boundary

\_fill (x

-

,y,oldc,newc

1

)

Call

Boundary

\_fill (x,y

-

1

,oldc,newc

)

Stop

* **Description of inbuilt functions:** 
  1. Circle:

The header file graphics.h contains **circle()** function which draws a circle with center at (x, y) and given radius.

**Syntax :**

circle(x, y, radius);

where,

(x, y) is center of the circle.

'radius' is the Radius of the circle.

* 1. Line:

The header file graphics.h contains line() function. Line function is used to draw a line from a point(x1,y1) to point(x2,y2) i.e. (x1,y1) and (x2,y2) are end points of the line.

**Syntax:**

line(int x1, int y1, int x2, int y2);

* 1. Rectangle: **rectangle()** is used to draw a rectangle. Coordinates of left top and right bottom corner are required to draw the rectangle. left specifies the Xcoordinate of top left corner, top specifies the Y-coordinate of top left corner, right specifies the X-coordinate of right bottom corner, bottom specifies the Y-coordinate of right bottom corner.

**Syntax:**

rectangle(int left, int top, int right, int bottom);

* 1. Outtextxy:

**outtextxy()** is a function which displays the text or string at a specified point (x, y) on the screen.

**Syntax :**

void outtextxy(int x, int y, char \*string); where, x, y are coordinates of the point and, third argument contains the address of string to be displayed.

* 1. Setcolor:

The header file graphics.h contains **setcolor()** function which is used to set the current drawing color to the new color.

**Syntax :** void setcolor(int color);

In Graphics, each color is assigned a number. Total number of colors available are 16.

**COLOR** **INT VALUES**

-------------------------------

BLACK 0

BLUE 1

GREEN 2

CYAN 3

RED 4

MAGENTA 5

BROWN 6

LIGHTGRAY 7

DARKGRAY 8

LIGHTBLUE 9

LIGHTGREEN 10

LIGHTCYAN 11

LIGHTRED 12

LIGHTMAGENTA 13

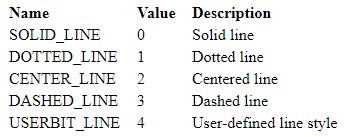
YELLOW 14

WHITE 15

1. Setlinestyle:

**setlinestyle()** function which sets the style for all lines drawn by line, lineto, rectangle, drawpoly, and so on.

**Syntax:** void setlinestyle(int linestyle, unsigned upattern,int thickness); **linestyle** specifies in which of several styles subsequent lines will be drawn (such as solid, dotted, centered, dashed).



1. Settextstyle:

**settextstyle()** function which is used to change the way in which text appears. Using it we can modify the size of text, change direction of text and change the font of text.

**Syntax :**

void settextstyle(int font, int direction, int font\_size);

where, font argument specifies the font of text, Direction can be HORIZ\_DIR (Left to right) or VERT\_DIR (Bottom to top).



**SOURCE CODE:**

#include <dos.h>

#include <math.h>

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <graphics.h>

#include <string.h>

void drawCircle(int xc, int yc, int x, int y)

{

putpixel(xc + x, yc + y, LIGHTRED);

putpixel(xc - x, yc + y, LIGHTRED);

putpixel(xc + x, yc - y, LIGHTRED);

putpixel(xc - x, yc - y, LIGHTRED);

putpixel(xc + y, yc + x, LIGHTRED);

putpixel(xc - y, yc + x, LIGHTRED);

putpixel(xc + y, yc - x, LIGHTRED);

putpixel(xc - y, yc - x, LIGHTRED);

}

void flood(int x, int y, int oc, int nc)

{

if (getpixel(x, y) == oc)

{

putpixel(x, y, nc);

flood(x + 1, y, oc, nc);

flood(x, y + 1, oc, nc);

flood(x - 1, y, oc, nc);

flood(x, y - 1, oc, nc);

}

}

void sixwing(int i)

{

setfillstyle(1, 7);

pieslice(300, 178, 180 + i, 210 + i, 80); // Incrementing starting and ending angles by adding i

floodfill(300, 178, 15);

setfillstyle(1, 9);

pieslice(300, 178, 185 + i, 205 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 110 + i, 140 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 9);

pieslice(300, 178, 115 + i, 135 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 0 + i, 30 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 9);

pieslice(300, 178, 5 + i, 25 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 80 + i, 50 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 9);

pieslice(300, 178, 75 + i, 55 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 290 + i, 320 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 9);

pieslice(300, 178, 295 + i, 315 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 235 + i, 265 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 9);

pieslice(300, 178, 240 + i, 260 + i, 70);

floodfill(300, 178, 15);

}

void fan\_blades()

{

// fan blades

//

setbkcolor(BLACK);

// fan blade body

setcolor(12);

line(300, 160, 300, 90);

line(305, 160, 330, 95);

line(310, 163, 352, 110);

line(315, 168, 370, 125);

line(320, 175, 385, 150);

line(320, 180, 390, 180);

line(320, 185, 385, 212);

line(317, 190, 370, 237);

line(312, 195, 350, 255);

line(305, 200, 325, 267);

line(300, 200, 300, 270);

line(295, 200, 275, 267);

line(288, 195, 250, 255);

line(283, 190, 230, 237);

line(280, 185, 215, 212);

line(210, 180, 280, 180);

line(280, 175, 215, 150);

line(285, 168, 230, 125);

line(290, 163, 250, 105);

line(295, 160, 275, 95);

}

void fc()

{

setcolor(9);

circle(300, 180, 10);

setcolor(7);

line(300, 170, 300, 130);

line(305, 170, 315, 133);

line(308, 171, 325, 138);

line(311, 173, 340, 148);

line(311, 175, 345, 163);

line(311, 180, 350, 180);

putpixel(285, 135, CYAN);

line(312, 185, 345, 200); //

line(312, 186, 338, 210); //

line(306, 190, 330, 220); //

line(305, 190, 315, 228); //

line(300, 190, 300, 230); //

line(295, 190, 285, 228); //

line(293, 190, 273, 223); //

line(291, 185, 262, 210); //

line(290, 185, 255, 200); //

line(250, 180, 290, 180); //

line(290, 175, 255, 160); //

line(290, 172, 260, 150); //

line(292, 171, 270, 140); //

line(295, 170, 285, 135);

setcolor(9);

circle(300, 180, 90);

}

void loading()

{

int gd = DETECT, gm, mx, my, i, ch, a, b;

initgraph(&gd, &gm, "C:\\Turboc3\\BGI");

mx = getmaxx() / 2;

my = getmaxy() / 2;

settextstyle(7, 0, 4);

outtextxy(175, 37, "FAN ROTATIONS");

outtextxy(105, 70, "IN COMPUTER GRAPHICS");

settextstyle(3, 0, 1);

outtextxy(220, 110, "GUIDED BY MR.KHAN MOHAMMED ZAID");

settextstyle(1, 0, 3);

outtextxy(10, 140, "PREPARED BY");

settextstyle(1, 0, 4);

outtextxy(10, 165, "210418 SHAIKH HASSAAN");

outtextxy(10, 195, "210413 ADBUL QADEER");

outtextxy(10, 225, "210414 AREEB PATEL");

outtextxy(10, 255, "210421 DHUKKA HARIS");

setcolor(YELLOW);

// creating circle and fill it with

// yellow color using floodfill.

circle(50, 70, 40);

circle(580, 70, 40);

setfillstyle(SOLID\_FILL, YELLOW);

// floodfill(50, 70, YELLOW);

// Set color of background to black

setfillstyle(SOLID\_FILL, BLACK);

// Use fill ellipse for creating eyes

fillellipse(60, 55, 2, 6);

fillellipse(40, 55, 2, 6);

fillellipse(570, 55, 2, 6);

fillellipse(590, 55, 2, 6);

// Use ellipse for creating mouth

ellipse(50, 70, 205, 335, 20, 9);

ellipse(50, 70, 205, 335, 20, 10);

ellipse(50, 70, 205, 335, 20, 11);

ellipse(580, 70, 205, 335, 20, 9);

ellipse(580, 70, 205, 335, 20, 10);

ellipse(580, 70, 205, 335, 20, 11);

settextstyle(1, 0, 3);

outtextxy(100, 310, "press any key to continue");

// outtextxy(100,335,"2:EXIT");

// scanf("%d",&ch);

// switch(ch)

// {

// case 1:

a = getch();

b = a;

if (b <= 255)

{

rectangle(100, 410, 540, 450);

settextstyle(8, 0, 3);

outtextxy(260, 370, "LOADING!!");

while (!kbhit())

{

for (i = 0; i < 440; i++)

{

if (kbhit())

break;

setlinestyle(3, 0, 1);

line(100 + i, 410, 100 + i, 450);

delay(10);

}

}

}

getch();

cleardevice();

closegraph();

}

void fan()

{

setcolor(9);

circle(300, 180, 50);

line(280, 270, 280, 340);

line(320, 270, 320, 340);

arc(270, 340, 270, 360, 10);

arc(330, 340, 180, 270, 10);

line(270, 350, 270, 370);

line(330, 350, 330, 370);

ellipse(300, 377, 120, 360, 60, 30);

ellipse(300, 377, 0, 60, 60, 30);

ellipse(300, 367, 190, 350, 60, 30);

ellipse(300, 360, 220, 320, 40, 15);

setcolor(WHITE);

rectangle(285, 355, 295, 365);

rectangle(305, 355, 315, 365);

flood(290, 360, BLACK, WHITE);

flood(310, 360, BLACK, WHITE);

flood(300, 377, BLACK, 12);

}

void main()

{

int gd = DETECT, gm, xc = 300, yc = 180, r2 = 90, ch1, rch, ch2, i = 0, j = 0;

int stangle1 = -45, endangle1 = 0, radius = 100;

int stangle2 = 135, endangle2 = 180;

loading();

initgraph(&gd, &gm, "C:\\TurboC3\\BGI");

A:

printf("1: Rotate the fan \n");

printf("2: Exit\n");

sixwing(i);

fan\_blades();

fc();

fan();

printf("Enter the choice : ");

scanf("%d", &ch1);

switch (ch1)

{

case 2:

// circleBres(xc,yc,r2);

break;

case 1:

printf("1: Clockwise \n");

printf("2: AntiClockwise \n");

printf("3; Exit \n");

printf("Enter the Rotation Method: ");

scanf("%d", &rch);

switch (rch)

{

case 1:

printf("Do you want RGB in Your Fan?\n");

printf("Press 1 for yes: \n");

printf("Press 2 for no: \n");

scanf("%d", &ch2);

switch (ch2)

{

case 1:

printf("RGB codings stay tuned");

/\* for clockwise \*/

// here the regulator

// switch case according to regulator

while (!kbhit())

{

for (i = 0; i <= 80; i = i + 40)

{

if (kbhit())

goto A;

// RED

setfillstyle(1, 7);

pieslice(300, 178, 180 + i, 210 + i, 80); // Incrementing starting and ending angles by adding i

floodfill(300, 178, 15);

setfillstyle(1, 4);

pieslice(300, 178, 185 + i, 205 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 110 + i, 140 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 4);

pieslice(300, 178, 115 + i, 135 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 0 + i, 30 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 4);

pieslice(300, 178, 5 + i, 25 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 80 + i, 50 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 4);

pieslice(300, 178, 75 + i, 55 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 290 + i, 320 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 4);

pieslice(300, 178, 295 + i, 315 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 235 + i, 265 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 4);

pieslice(300, 178, 240 + i, 260 + i, 70);

floodfill(300, 178, 15);

fan\_blades();

fc();

fan();

delay(10);

cleardevice();

}

// BLUE

for (i = 0; i <= 80; i = i + 40)

{

if (kbhit())

goto A;

setfillstyle(1, 7);

pieslice(300, 178, 180 + i, 210 + i, 80); // Incrementing starting and ending angles by adding i

floodfill(300, 178, 15);

setfillstyle(1, 1);

pieslice(300, 178, 185 + i, 205 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 110 + i, 140 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 1);

pieslice(300, 178, 115 + i, 135 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 0 + i, 30 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 1);

pieslice(300, 178, 5 + i, 25 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 80 + i, 50 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 1);

pieslice(300, 178, 75 + i, 55 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 290 + i, 320 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 1);

pieslice(300, 178, 295 + i, 315 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 235 + i, 265 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 1);

pieslice(300, 178, 240 + i, 260 + i, 70);

floodfill(300, 178, 15);

fan\_blades();

fc();

fan();

delay(10);

cleardevice();

}

// GREEN

for (i = 0; i <= 80; i = i + 40)

{

if (kbhit())

goto A;

setfillstyle(1, 7);

pieslice(300, 178, 180 + i, 210 + i, 80); // Incrementing starting and ending angles by adding i

floodfill(300, 178, 15);

setfillstyle(1, 2);

pieslice(300, 178, 185 + i, 205 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 110 + i, 140 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 2);

pieslice(300, 178, 115 + i, 135 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 0 + i, 30 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 2);

pieslice(300, 178, 5 + i, 25 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 80 + i, 50 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 2);

pieslice(300, 178, 75 + i, 55 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 290 + i, 320 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 2);

pieslice(300, 178, 295 + i, 315 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 235 + i, 265 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 2);

pieslice(300, 178, 240 + i, 260 + i, 70);

floodfill(300, 178, 15);

fan\_blades();

fc();

fan();

delay(10);

cleardevice();

}

for (i = 0; i <= 80; i = i + 40)

{

if (kbhit())

goto A;

setfillstyle(1, 7);

pieslice(300, 178, 180 + i, 210 + i, 80); // Incrementing starting and ending angles by adding i

floodfill(300, 178, 15);

setfillstyle(1, i % 15);

pieslice(300, 178, 185 + i, 205 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 110 + i, 140 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, i % 15);

pieslice(300, 178, 115 + i, 135 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 0 + i, 30 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, i % 15);

pieslice(300, 178, 5 + i, 25 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 80 + i, 50 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, i % 15);

pieslice(300, 178, 75 + i, 55 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 290 + i, 320 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, i % 15);

pieslice(300, 178, 295 + i, 315 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, i % 15);

pieslice(300, 178, 235 + i, 265 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, i % 15);

pieslice(300, 178, 240 + i, 260 + i, 70);

floodfill(300, 178, 15);

fan\_blades();

fc();

fan();

delay(10);

cleardevice();

}

}

break;

case 2:

printf("Rotate without RGB");

while (!kbhit())

{

for (i = 0; i <= 80; i = i + 40)

{

if (kbhit())

goto A;

sixwing(i);

fan\_blades();

fc();

fan();

delay(10);

cleardevice();

}

}

/\* for clockwise \*/

// here the regulator

// switch case according to regulator

}

break;

case 2:

printf("Do you want RGB in Your Fan?\n");

printf("Press 1 for yes: \n");

printf("Press 2 for no: \n");

scanf("%d", &ch2);

switch (ch2)

{

case 1:

printf("RGB codings stay tuned");

/\* for anticlockwise \*/

// here the regulator

// switch case according to regulator

while (!kbhit())

{

for (i = 0; i <= 40; i = i + 20)

{

if (kbhit())

goto A;

// RED

setfillstyle(1, 7);

pieslice(300, 178, 180 + i, 210 + i, 80); // Incrementing starting and ending angles by adding i

floodfill(300, 178, 15);

setfillstyle(1, 4);

pieslice(300, 178, 185 + i, 205 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 110 + i, 140 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 4);

pieslice(300, 178, 115 + i, 135 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 0 + i, 30 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 4);

pieslice(300, 178, 5 + i, 25 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 80 + i, 50 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 4);

pieslice(300, 178, 75 + i, 55 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 290 + i, 320 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 4);

pieslice(300, 178, 295 + i, 315 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 235 + i, 265 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 4);

pieslice(300, 178, 240 + i, 260 + i, 70);

floodfill(300, 178, 15);

fan\_blades();

fc();

fan();

delay(10);

cleardevice();

}

// BLUE

for (i = 0; i <= 40; i = i + 20)

{

if (kbhit())

goto A;

setfillstyle(1, 7);

pieslice(300, 178, 180 + i, 210 + i, 80); // Incrementing starting and ending angles by adding i

floodfill(300, 178, 15);

setfillstyle(1, 1);

pieslice(300, 178, 185 + i, 205 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 110 + i, 140 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 1);

pieslice(300, 178, 115 + i, 135 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 0 + i, 30 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 1);

pieslice(300, 178, 5 + i, 25 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 80 + i, 50 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 1);

pieslice(300, 178, 75 + i, 55 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 290 + i, 320 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 1);

pieslice(300, 178, 295 + i, 315 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 235 + i, 265 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 1);

pieslice(300, 178, 240 + i, 260 + i, 70);

floodfill(300, 178, 15);

fan\_blades();

fc();

fan();

delay(10);

cleardevice();

}

// GREEN

for (i = 0; i <= 40; i = i + 20)

{

if (kbhit())

goto A;

setfillstyle(1, 7);

pieslice(300, 178, 180 + i, 210 + i, 80); // Incrementing starting and ending angles by adding i

floodfill(300, 178, 15);

setfillstyle(1, 2);

pieslice(300, 178, 185 + i, 205 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 110 + i, 140 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 2);

pieslice(300, 178, 115 + i, 135 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 0 + i, 30 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 2);

pieslice(300, 178, 5 + i, 25 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 80 + i, 50 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 2);

pieslice(300, 178, 75 + i, 55 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 290 + i, 320 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 2);

pieslice(300, 178, 295 + i, 315 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 235 + i, 265 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, 2);

pieslice(300, 178, 240 + i, 260 + i, 70);

floodfill(300, 178, 15);

fan\_blades();

fc();

fan();

delay(10);

cleardevice();

}

for (i = 0; i <= 40; i = i + 20)

{

if (kbhit())

goto A;

setfillstyle(1, 7);

pieslice(300, 178, 180 + i, 210 + i, 80); // Incrementing starting and ending angles by adding i

floodfill(300, 178, 15);

setfillstyle(1, i % 15);

pieslice(300, 178, 185 + i, 205 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 110 + i, 140 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, i % 15);

pieslice(300, 178, 115 + i, 135 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 0 + i, 30 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, i % 15);

pieslice(300, 178, 5 + i, 25 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 80 + i, 50 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, i % 15);

pieslice(300, 178, 75 + i, 55 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, 7);

pieslice(300, 178, 290 + i, 320 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, i % 15);

pieslice(300, 178, 295 + i, 315 + i, 70);

floodfill(300, 178, 15);

setfillstyle(1, i % 15);

pieslice(300, 178, 235 + i, 265 + i, 80);

floodfill(300, 178, 15);

setfillstyle(1, i % 15);

pieslice(300, 178, 240 + i, 260 + i, 70);

floodfill(300, 178, 15);

fan\_blades();

fc();

fan();

delay(10);

cleardevice();

}

}

break;

case 2:

while (!kbhit())

{

for (i = 0; i <= 40; i = i + 20)

{

if (kbhit())

goto A;

sixwing(i);

fan\_blades();

fc();

fan();

delay(5);

cleardevice();

}

}

/\* for anticlockwise \*/

// here the regulator

// switch case according to regulator

}

break;

case 3:

exit(0);

}

break;

case 3:

exit(0);

}

getch();

closegraph();

}

**OUTPUTS:**

