

THE MAD PAINTER

CS BOARD PROJECT 2017-18

'The Mad Painter' provides a platform for you to exercise your heart's content and draw all you like.

A

BONAFIDE CERTIFICATE

Certified to be the bonafide record of work done
by Master **AKASH.S** of class **XII – B** in **PADMA SESHADRI BALA
BHAVAN SR. SEC. SCHOOL, CHENNAI.**

During the year : 2017-18

Date : / / 2018

P.G.T. in Computer Science

Chennai

Submitted for All – India Senior Secondary
Practical held in COMPUTER SCIENCE at PSBB, NUNGAMBAKKAM
CHENNAI.

Date : / / 2018

Examiner

Seal

INDEX

S.N O		PAGE NO.
1	SYNOPSIS	3
2	FLOWCHART	4
3	ALGORITHM	7
4	PROGRAM CODE	17
5	SAMPLE SCREENSHOTS	33
6	LIMITATIONS OF PROJECT	36
7	SCOPE FOR IMPROVEMENT	36
8	BIBILIOGRAPHY	37

SYNOPSIS

Do you remember that time as a kid when you used to scribble on every square inch of space you got? Well, here's an application to bring back that kid in you! 'The Mad Painter' provides a platform for you to exercise your heart's content and draw all you like. But it doesn't end there. It can also be used as a teaching aid. The scope of drawing is quite limited, but the mere fact that whatever you draw actually makes a difference on the screen is in itself satisfying.

The application is fully facilitated with the usage of a working mouse. The limited memory that C++ offers proves to be a problem, but it has been well overcome. The screen consists of a number of buttons. On clicking them, you can achieve the full potential of the program, i.e. to say, you can draw!

There is even a provision to change colour. As much as it might resemble Microsoft Paint, with all due respect I leave the rights of the initial idea with them. This is a mere stimulation of it in a much lower scale.

The application provides facilities to draw basic shapes such as a line, an ellipse, a circle, and any polygon. In addition, it contains a feature that brings up a graph sheet on-screen. Such additional features can help calculate the area of the shapes you drew, and so on. On a higher level, it can enable you to integrate functions and calculate the value of pi.

The application also uses data file handling. This lies in the core of the program – the Undo function! Every shape drawn on the screen is recorded with the help of a temporary file. In the case of the need to undo, all the content of this file is copied into a second temporary file excluding the last entry. This second temporary file is then set as default. All the data on the screen is cleared and every move of the user is reconstructed step by step.

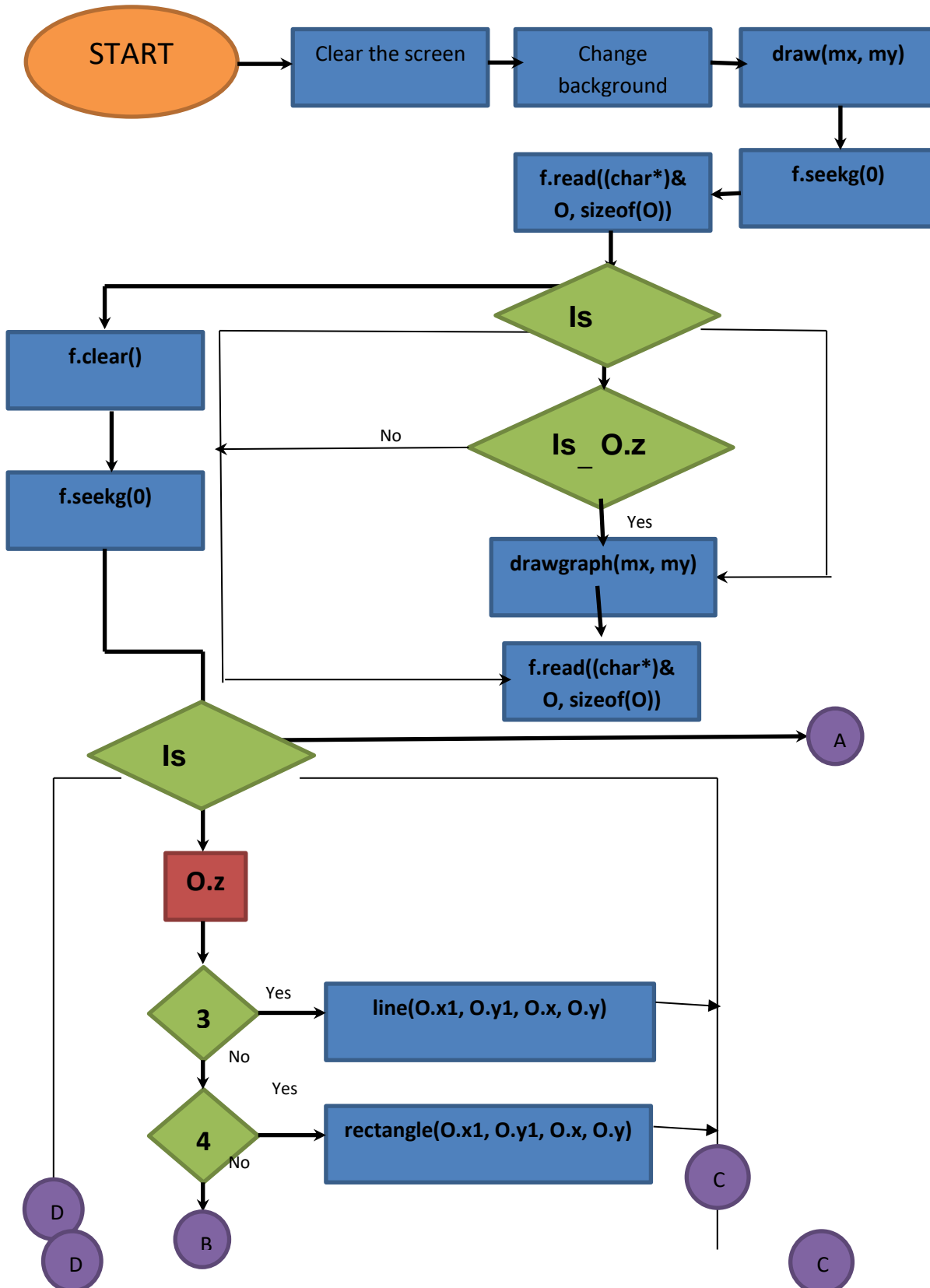
Data files are also used in other ways. They are used to Save the file to the hard disk and Load a file from the disk. This ensures that your memories can be saved and provides you an opportunity to refer to it anytime again. What more could one ask for from such a program?

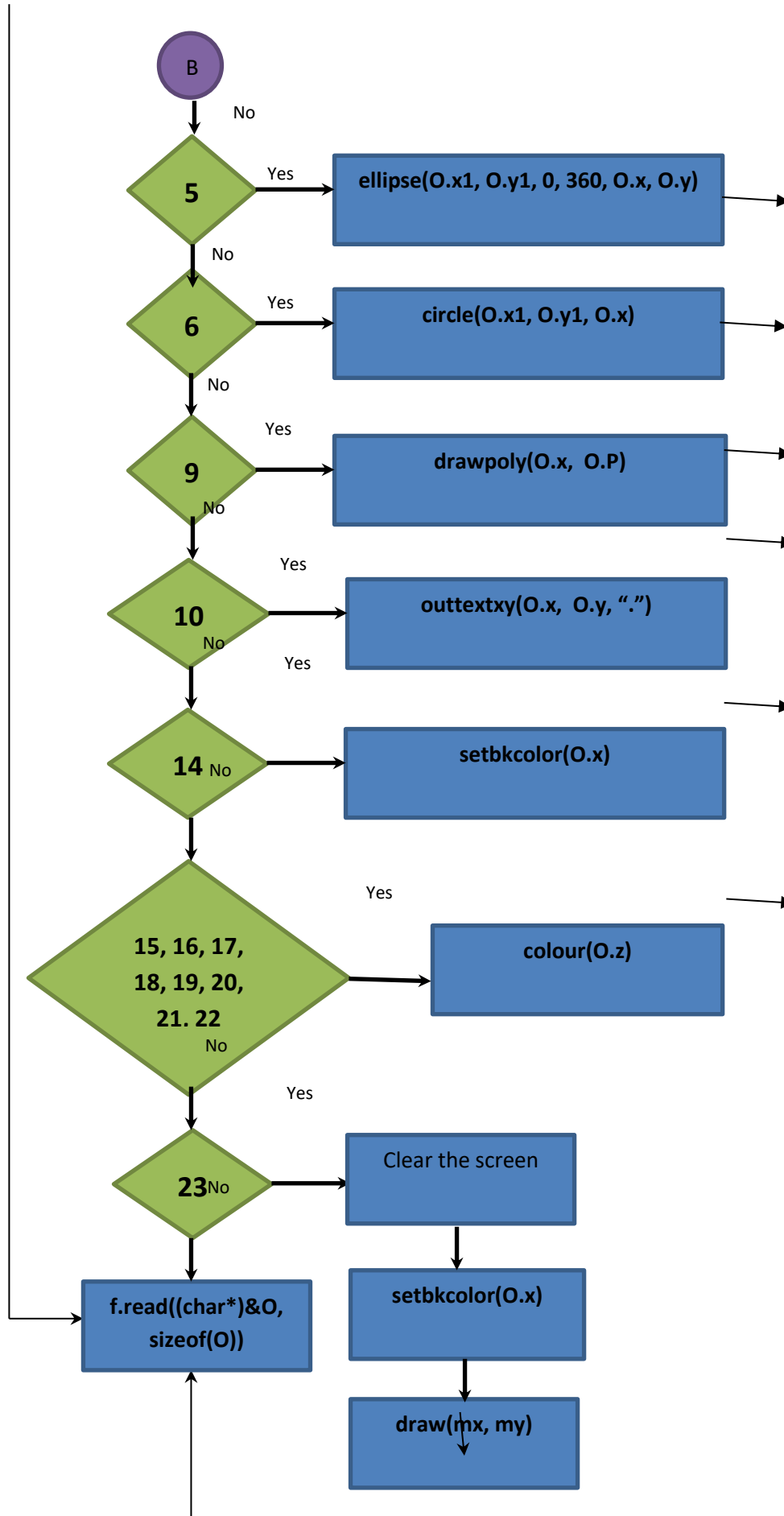
Sadly, that is the conclusion of all the superhuman abilities that the application possesses, but the scope is unlimited, for, if the whole of the initial windows operating system could have been coded from C++, there is still a long way to go for the application. It will reach market in a few years in its final form with its true potential.

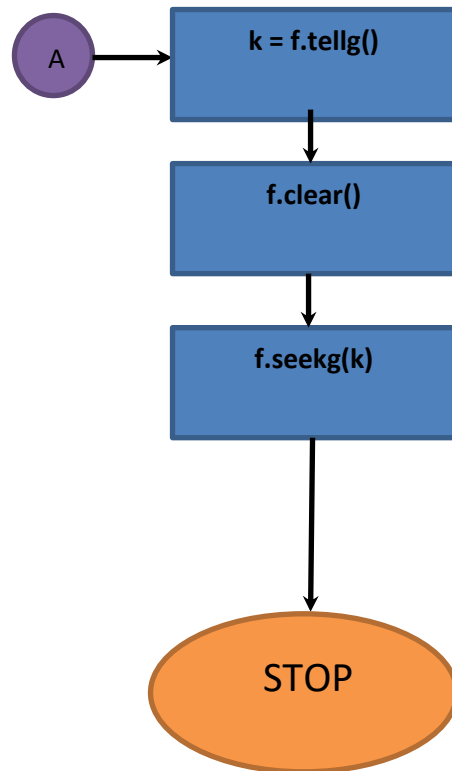
With the removal of MS-paint as a mandatory application in Windows – 10, this application might then soon take its place. The future is yet to be decided; this application is merely leading the way!!

FLOWCHART

Function: redo(fstream &f, Obj &O, int &mx, int &my)







ALGORITHM

Define an inbuilt union REGS with identifiers in and out. This helps call interrupt functions using C++.

Define a structure Obj with

data members - int z, x1, y1, x, y, and int array P
member function - default constructor.

ALGORITHM FOR DEFAULT CONSTRUCTOR OF OBJ:

1. Start
2. Assign all data members to 0.
3. Stop

ALGORITHM FOR FUNCTION `callmouse()` THAT RETURNS AN INT:

1. Start
2. `in.x.ax=1`. It gets the mouse's current status (active/inactive).
3. Call interrupt 86, i.e., `int86()` with interrupt number 51 (that of the mouse), and in and out passed by address.
4. Return value 1 to show successful calling of function.
5. Stop

ALGORITHM FOR FUNCTION `mouseposi()` WITH `xpos`, `ypos` AND `click` PASSED BY REFERENCE:

1. Start
2. `in.x.ax=3`. It is used to get the mouse position.
3. Call interrupt 86, i.e., `int86()` with interrupt number 51 (that of the mouse), and in and out passed by address.
4. `click = out.x.bx`. Using the value of click, it can be identified which button of the mouse is pressed.
 - i. If `cl=0`, no button of the mouse has been clicked.
 - ii. If `cl=1`, the left button has been clicked.
 - iii. If `cl=2`, the right button has been clicked.
 - iv. If `cl=3`, the scroll button has been clicked.
5. `xpos = out.x.cx`.
6. `ypos = out.x.dx`. The position of the mouse is stored.
7. Stop

ALGORITHM FOR FUNCTION `mousehide()` THAT RETURNS AN INT:

1. Start
2. `in.x.ax=2`. It is used to hide the mouse pointer.
3. Call interrupt 86, i.e., `int86()` with interrupt number 51 (that of the mouse), and in and out passed by address.
4. Return value 1 to show successful calling of function.
5. Stop

ALGORITHM FOR FUNCTION `setposi()` WITH `xpos` AND `ypos` PASSED BY REFERENCE:

1. Start
2. `in.x.ax=4`. It is used to display the mouse position.
3. `in.x.cx = xpos`
4. `in.x.dx = ypos`. The coordinates on the screen are assigned.
5. Call interrupt 86, i.e., `int86()` with interrupt number 51 (that of the mouse), and `in` and `out` passed by address.
6. Stop

ALGORITHM FOR FUNCTION `mtrav()` WITH `x`, `y`, `cl` AND `z` PASSED BY REFERENCE:

1. Start
2. As long as `cl` is 0, i.e., nothing has been clicked yet, repeat steps 3 and 4.
3. Call `mouseposi()` with `x`, `y` and `cl`.
4. Display the coordinates of the mouse in a corner of the screen.
5. Stop

ALGORITHM FOR FUNCTION `undo()` WITH FSTREAM OBJECT `f` AND AN OBJECT `O` OF STRUCT `OBJ` PASSED BY REFERENCE:

1. Start
2. `int m = f.tellg() - sizeof(O)`. It stores the size of the resultant file.
3. Open a binary file "`tmp2.dat`" using an `fstream` object `g` opened in - in, out mode.
4. As long as `f.tellg() < m`, repeat steps 5 and 6.
5. Read into `O`, data from file `f`.
6. Write `O` into the file `g`.
7. Close both the files.
8. Remove the file "`tmp.dat`" and rename "`tmp2.dat`" to "`tmp.dat`".
9. Reopen "`tmp.dat`" in file `f`.
10. Stop

ALGORITHM FOR FUNCTION `drawgraph()` WITH `mx` AND `my` PASSED BY REFERENCE:

1. Start
2. Draw horizontal lines separated by `my/15` till it reaches `my-64`
3. Draw vertical lines separated by `mx/15` till it reaches `mx`
4. Stop

ALGORITHM FOR FUNCTION `draw()` WITH `mx` AND `my` PASSED BY REFERENCE:

1. Start

2. Draw a 90x25 pixels button "Circle" in the bottom right corner.
3. Draw a 90x25 pixels button "Ellipse" in the bottom, to the left of "Circle".
4. Draw a 90x25 pixels button "Rectangle" in the bottom, to the left of "Ellipse".
5. Draw a 90x25 pixels button "Line" in the bottom, to the left of "Rectangle".
6. Draw a 90x25 pixels button "UNDO" in the bottom, to the left of "Line".
7. Draw a 90x25 pixels button "Exit" in the bottom left corner.
8. Draw a 90x25 pixels button "Open Graph" above "Circle".
9. Draw a 90x25 pixels button "Close Graph" to the left of "Open Graph".
10. Draw a 90x25 pixels button "Polygon" to the left of "Close Graph".
11. Draw a 90x25 pixels button "Free Hand" to the left of "Polygon".
12. Draw a 90x25 pixels button "Save File" to the left of "Free Hand".
13. Draw a 90x25 pixels button "Load File" above "Exit".
14. Draw a 90x25 pixels button "HELP" in the top right corner.
15. Draw a 90x25 pixels button "Colour" to the left of "HELP".
16. Draw a 20x15 pixels button "+" to the left of "Colour".
17. Draw a 15x10 pixels white coloured button "WHITE" to the left, above "Load File".
18. Draw a 15x10 pixels red coloured button "RED" to the left, above "WHITE".
19. Draw a 15x10 pixels blue coloured button "BLUE" to the left, above "RED".
20. Draw a 15x10 pixels green coloured button "GREEN" to the left, above "BLUE".
21. Draw a 15x10 pixels dark gray coloured button "DARKGRAY" to the left, above "GREEN".
22. Draw a 15x10 pixels brown coloured button "BROWN" to the left, above "DARKGRAY".
23. Draw a 15x10 pixels yellow coloured button "YELLOW" to the left, above "BROWN".
24. Draw a 15x10 pixels magenta coloured button "MAGENTA" to the left, above "YELLOW".
25. Display the names of the buttons on them as prompts.
26. Stop

ALGORITHM FOR FUNCTION `colour()` WITH `col` PASSED BY VALUE:

1. Start
2. If `col` is 15, set the colour to white.
3. If `col` is 16, set the colour to red.
4. If `col` is 17, set the colour to blue.

5. If col is 18, set the colour to green.
6. If col is 19, set the colour to dark gray.
7. If col is 20, set the colour to brown.
8. If col is 21, set the colour to yellow.
9. If col is 22, set the colour to magenta.
10. Stop

ALGORITHM FOR FUNCTION `redo()` WITH FSTREAM OBJECT `f`, AN OBJECT `O` OF STRUCT OBJ, `mx` AND `my` PASSED BY REFERENCE:

1. Start
2. Clear the screen.
3. Set the background colour to cyan.
4. Call `draw()` with `mx` and `my`.
5. Go to the beginning of the file.
6. Read into `O`, data from file `f`.
7. As long as the end of file is not reached, repeat steps 8 and 9.
8. If `O.z` is 7,
 - i) Call `mousehide()`
 - ii) Call `drawgraph()` with `mx` and `my`
 - iii) Call `callmouse()`
9. Read into `O`, data from file `f`.
10. Call `f.clear()`
11. Go to the beginning of the file.
12. As long as the end of file is not reached, repeat steps 13 and 22.
13. If `O.z` is 3, call `line()` with `O.x1`, `O.y1`, `O.x` and `O.y`.
14. If `O.z` is 4, call `rectangle()` with `O.x1`, `O.y1`, `O.x` and `O.y`.
15. If `O.z` is 5, call `ellipse()` with `O.x1`, `O.y1`, 0, 360, `O.x` and `O.y`.
16. If `O.z` is 6, call `circle()` with `O.x1`, `O.y1` and `O.x`.
17. If `O.z` is 9, call `drawpoly()` with `O.x` and `O.P`.
18. If `O.z` is 10, display a "." at (`O.x`, `O.y`).
19. If `O.z` is 14, call `setbkcolor()` with `O.x`.
20. If `O.z` is from 15 to 22, call `colour()` with `O.z`.
21. If `O.z` is 23,
 - i) Clear the screen
 - ii) Call `draw()` with `mx` and `my`
22. Read into `O`, data from file `f`.
23. `k = f.tellg()`
24. Call `f.clear()`
25. Call `f.seekg()` with `k`
26. Stop

ALGORITHM FOR `main()`:

1. Start
2. Open graphics drivers.

3. `mx = getmaxx()`
4. `my = getmaxy()`
5. Call `setbkcolor()` with 3.
6. Call `remove()` with "tmp.dat" to remove any such existing file.
7. Open "tmp.dat" using a `fstream` object `f` in - in, out, and binary mode.
8. Call `intro()`.
9. `int a = 100, b = 370, col = 15`
10. Call `setposi()` with `a` and `b`.
11. Declare `x`, `y`, `cl` and `z`. `z=0`
12. Call `callmouse()`
13. Call `draw()` with `mx` and `my`.
14. Call `mouseposi()` with `x`, `y`, and `cl`.
15. If `cl = 1`,
 - i) `a = x`
 - ii) `b = y`
 - iii) `z` is assigned its value based on the coordinates of the button clicked. This is used to uniquely identify each button.

Coordinates	Assignments
Exit	<code>z=1</code>
UNDO	<code>z=2</code>
Line	<code>z=3</code>
Rectangle	<code>z=4</code>
Ellipse	<code>z=5</code>
Circle	<code>z=6</code>
Open Graph	<code>z=7</code>
Close Graph	<code>z=8</code>
Polygon	<code>z=9</code>
Free Hand	<code>z=10</code>
Save File	<code>z=11</code>
Load File	<code>z=12</code>
HELP	<code>z=13</code>
Colour	<code>z=14</code>
WHITE	<code>z=15</code>

RED	z=16
BLUE	z=17
GREEN	z=18
DARKGRAY	z=19
BROWN	z=20
YELLOW	z=21
MAGENTA	z=22
+	z=23
--other--	z=cl=0

16. Display the mouse position (x,y).
17. Display the click status (cl). If cl becomes 1, click is successful.
18. Repeat steps 16 to 17 as long as cl is 0.
19. cl = 0
20. Create an object O of structure Obj
21. O.z = z
22. Declare x1, y1
23. Delay 0.5 second. This is so that one click is not counted as many (click identification buffer time).
24. If z is 1,
 exit.
25. if z is 2,
 i) Call undo() with f and O
 ii) Call redo() with f, O, mx and my
26. If z is 3,
 i) Call mtrav() with x, y, cl and z
 ii) x1 = x
 iii) y1 = y
 iv) cl = 0
 v) Call mtrav() with x, y, cl and z
 vi) Call colour() with col
 vii) Call line() with x1, y1, x and y
 viii) O.x1 = x1
 ix) O.y1 = y1
 x) O.x = x
 xi) O.y = y
 xii) Write O into file f
27. If z is 4,
 i) Call mtrav() with x, y, cl and z
 ii) x1 = x

```

    iii)  y1 = y
    iv)   cl = 0
    v)    Call mtrav() with x, y, cl and z
    vi)   Call colour() with col
    vii)  Call rectangle() with x1, y1, x and y
    viii) O.x1 = x1
    ix)   O.y1 = y1
    x)    O.x = x
    xi)   O.y = y
    xii)  Write O into file f
28.  If z is 5,
    i)    Call mtrav() with x, y, cl and z
    ii)   Declare x2, y2
    iii)  x1 = x
    iv)   y1 = y
    v)    cl = 0
    vi)   Call mtrav() with x, y, cl and z
    vii)  x2 = |x1-x|
    viii) Call mtrav() with x, y, cl and z
    ix)   y2 = |y1-y|
    x)    Call colour() with col
    xi)   Call ellipse() with x1, y1, 0, 360, x2 and y2
    xii)  O.x1 = x1
    xiii) O.y1 = y1
    xiv)  O.x = x2
    xv)   O.y = y2
    xvi)  Write O into file f
29.  If z is 6,
    i)    Call mtrav() with x, y, cl and z
    ii)   Declare xr
    iii)  x1 = x
    iv)   y1 = y
    v)    cl = 0
    vi)   Call mtrav() with x, y, cl and z
    vii)   $xr = ((x-x1)^2 + (y-y1)^2)^{1/2}$ 
    viii) Call colour() with col
    ix)   Call circle() with x1, y1, xr
    x)    O.x1 = x1
    xi)   O.y1 = y1
    xii)  O.x = xr
    xiii) Write O into file f
30.  If z is 7,
    i)    Write O into the file
    ii)   Call redo() with f, O, mx and my
31.  If z is 8,
    i)    Declare m
    ii)   m = f.tellg()

```

- iii) Open "tmp2.dat" using a fstream object g in - in, out, and binary mode.
 - iv) Call f.seekg() with 0.
 - v) As long a f.tellg() < m,
 - a) Read into O, data from the file f
 - b) If O.z is not 7, write O to file g
 - vi) Close both the files.
 - vii) Remove the file "tmp.dat" and rename "tmp2.dat" to "tmp.dat".
 - viii) Reopen "tmp.dat" in file f.
 - ix) Call redo() with f, O, mx and my
 - x) Write O into file f
32. If z is 9,
- i) Clear the screen
 - ii) Close Graphics mode
 - iii) Declare n
 - iv) Prompt
 - Number of points:
 - v) Read in number into n
 - vi) If col is not between 3 and 15, prompt - "INVALID" and repeat steps iv and v till a valid input is obtained.
 - vii) Open Graphics mode.
 - viii) Call redo() with f, O, mx and my
 - ix) Declare P[30]
 - x) i=0.
 - xi) Until i<n,
 - a) Call mtrav() with x, y, cl and z
 - b) cl=0
 - c) P[2*i + 0]=x
 - d) P[2*i + 1]=y
 - e) i=i+1
 - xii) Call drawpoly() with n+1 and P
 - xiii) O.x = n+1
 - xiv) i=0.
 - xv) Until i<2*n,
 - a) O.P[i] = P[i]
 - b) i=i+1
 - xvi) Write O into the file f
33. If z is 10,
- i) Call mtrav() with x, y, cl and z
 - ii) As long as cl=1, repeat steps iii to
 - iii) Call colour() with col
 - iv) Call mouseposi with x, y and cl
 - v) Display a "." at (x-1, y-1)
 - vi) Declare flag
 - vii) flag = 0
 - viii) Call f.seekg() with 0 and ios::end

```

ix)    If f.tellg() = 0,
        a) O.x = x
        b) O.y = y
        c) flag = 1
x)     Otherwise,
        a) Call f.seekg() with sizeof(O) and
           ios::end
        b) Read into O, data from file f.
        c) If O.x is not x and O.y is not y,
           O.x=x
           O.y=y
           flag=1
xi)    Write O into file f
34.    If z is 11,
        i)    Clear the screen
        ii)   Close Graphics mode
        iii)  Declare S[15]
        iv)   Prompt:
              Name of File:
        v)    Read in character array into S.
        vi)   If string length of S is greater than 8, prompt -
              "INVALID" and repeat steps iv and v till a valid
              input is obtained.
        vii)  Add ".dat" to the end of string S.
        viii) Open file S using a fstream object g in - in, out,
              and binary mode.
        ix)   Declare k, i
        x)    k=f.tellg()-1
        xi)   i=0
        xii)  As long i < k,
              a) Write O into the file g.
              b) Read into O, data from the file f.
              c) i=i+sizeof(O)
        xiii) O.z=-1
        xiiii) Write O into the file g.
        xv)   Close the file g.
        xvi)  Open Graphics mode.
        xvii) Call redo() with f, O, mx and my
35.    If z is 12,
        i)    Clear the screen
        ii)   Close Graphics mode
        iii)  Declare S[15]
        iv)   Prompt:
              Name of File:
        v)    Read in character array into S.
        vi)   Add ".dat" to the end of string S.
        vii)  Open file S using a fstream object g in - in, out,
              and binary mode.

```


- viii) Close f
- ix) Remove "tmp.dat"
- x) Open file "tmp.dat" using f in - in, out, and binary mode.
- xi) Read into O, data from the file g.
- xii) As long O.z is not -1,
 - a) Write O into the file f.
 - b) Read into O, data from the file g.
- xii) Close file g
- xiii) Declare k
- xiv) k=f.tellg()-1
- xv) Call f.clear()
- xvi) Call f.seekg() with k.
- xvii) Open Graphics mode.
- xviii) Call redo() with f, O, mx and my

36. If z is 13,

- i) Clear the screen
- ii) Close Graphics mode
- iii) Declare K[50]
- iv) Open file "projhelp.dat" using an ifstream object h.
- v) Read into K data from file h, either up to 50 characters or up to a "\n".
- vi) As long as h.eof() = 0,
 - a) Display K.
 - b) Read into K data from file h, either up to 50 characters or up to a "\n".
- vii) Open Graphics mode.
- viii) Call redo() with f, O, mx and my

37. If z is 14,

- i) Clear the screen
- ii) Close Graphics mode
- iii) Declare col
- iv) Display a Menu

Black	0
Blue	1
Green	2
Cyan	3
Red	4
Magenta	5
Brown	6
Light Gray	7

Choose your Colour:
- v) Read in choice into col
- vi) If col is not between 0 and 7, prompt - "INVALID" and repeat steps iv and v till a valid input is obtained.

- vii) Open Graphics mode.
 - viii) Call redo() with f, O, mx and my
 - ix) Call setbkcolor() with col
 - x) O.x = col
 - xi) Write O into the file f
38. If z is between 15 and 22,
- i) col = z
 - ii) Write O into file f
 - iii) Call colour() with col
39. If z is 23,
- i) Clear the screen
 - ii) Call setbkcolor() with 3
 - iii) Call callmouse()
 - iv) Call draw() with mx and my
 - v) Write O into file f
40. Repeat steps 12 to 39 as long as kbhit() is 0
41. Stop

PROGRAM CODE

```
#include<graphics.h>
#include<fstream.h>
#include<stdio.h>
#include<conio.h>
#include<dos.h>
#include<string.h>
#include<math.h>

union REGS in, out;

struct Obj {
    int z;
    int x1, y1, x, y;
    int P[30];
    Obj() {
        z = x1 = y1 = x = y = 0;
    }
};

int callmouse() {
    in.x.ax=1;
    int86(51, &in, &out);
    return 1; }

void mouseposi(int &xpos, int &ypos, int &click) {
    in.x.ax=3;
    int86(51, &in, &out);
    click = out.x.bx;
    xpos=out.x.cx;
    ypos=out.x.dx; }
```

```

int mousehide() {
    in.x.ax=2;
    int86(51, &in, &out);
    return 1; }

void setposi(int &xpos, int &ypos) {
    in.x.ax=4;
    in.x.cx=xpos;
    in.x.dx=ypos;
    int86(51, &in, &out); }

void mtrav(int &x, int &y, int &cl, int z) {
    while(!cl) {
        mouseposi(x,y,cl);

        setcolor(RED);
        gotoxy(1, 2);
        printf("\nmouse position is: %d, %d", x, y);
        printf("\nclick %d", cl);
        //printf("\nz is %d", z);

    } }

void undo(fstream &f, Obj &O) {
    int m = f.tellg() - sizeof(O);
    fstream g("tmp2.dat", ios::binary| ios::in| ios::out);

    f.seekg(0);

    while(f.tellg() < m) {
        f.read((char*)&O, sizeof(O));
        g.write((char*)&O, sizeof(O));
    }

    f.close(); g.close();
    remove("tmp.dat");
    rename("tmp2.dat", "tmp.dat");

    f.open("tmp.dat", ios::binary| ios::in| ios::out);
}

void drawgraph(int &mx, int &my) {
    for(int i=0; i<mx; i+=mx/15) {
        setlinestyle(0, 1, 500);
        setcolor(4);
        line(i, 0, i, my-64);
        for(int j=i+mx/150; j<i+mx/15; j+=mx/150) {
            setcolor(3);
            setlinestyle(1, 1, 5);
            line(j, 0, j, my-64);
        }
    }
    for(i=0; i<my-64; i+=my/15) {
        setlinestyle(0, 1, 500);
        setcolor(4);

```

```

        line(0, i, mx, i);
        for(int j=i+my/150; j<i+my/15; j+=my/150) {
            setcolor(2);
            setlinestyle(1, 1, 5);
            line(0, j, mx, j);
        }
    }
}

inline void draw(int &mx, int &my) {
    /* draw the buttons */
    bar(mx-100, my-35, mx-10, my-10);
        outtextxy(mx-70, my-30, "Circle");
    bar(mx-200, my-35, mx-110, my-10);
        outtextxy(mx-175, my-30, "Ellipse");
    bar(mx-300, my-35, mx-210, my-10);
        outtextxy(mx-290, my-30, "Rectangle");
    bar(mx-400, my-35, mx-310, my-10);
        outtextxy(mx-370, my-30, "Line");
    bar(mx-500, my-35, mx-410, my-10);
        outtextxy(mx-470, my-30, "UNDO");
    bar(mx-600, my-35, mx-510, my-10);
        outtextxy(mx-570, my-30, "Exit");
    bar(mx-100, my-65, mx-10, my-40);
        outtextxy(mx-90, my-60, "Open Graph");
    bar(mx-200, my-65, mx-110, my-40);
        outtextxy(mx-196, my-60, "Close Graph");
    bar(mx-300, my-65, mx-210, my-40);
        outtextxy(mx-290, my-60, "Polygon");
    bar(mx-400, my-65, mx-310, my-40);
        outtextxy(mx-390, my-60, "Free Hand");
    bar(mx-500, my-65, mx-410, my-40);
        outtextxy(mx-490, my-60, "Save File");
    bar(mx-600, my-65, mx-510, my-40);
        outtextxy(mx-590, my-60, "Load File");
    bar(mx-100, 40, mx-10, 65);
        outtextxy(mx-90, 50, "HELP");
    bar(mx-200, 40, mx-110, 65);
        outtextxy(mx-190, 50, "Colour");
    bar(mx-230, 40, mx-210, 55);
        outtextxy(mx-222, 45, "+");
    setfillstyle(1, WHITE);
    bar(mx-600, my-90, mx-585, my-80);
    setfillstyle(1, RED);
    bar(mx-600, my-100, mx-585, my-90);
    setfillstyle(1, BLUE);
    bar(mx-600, my-110, mx-585, my-100);
    setfillstyle(1, GREEN);
    bar(mx-600, my-120, mx-585, my-110);
    setfillstyle(1, DARKGRAY);
    bar(mx-600, my-130, mx-585, my-120);
    setfillstyle(1, BROWN);
    bar(mx-600, my-140, mx-585, my-130);
    setfillstyle(1, YELLOW);
    bar(mx-600, my-150, mx-585, my-140);
    setfillstyle(1, MAGENTA);
    bar(mx-600, my-160, mx-585, my-150);
}

```

```

        setfillstyle(1,WHITE);
    }

void colour(int col) {
    if(col==15)    setcolor(WHITE);
    if(col==16)    setcolor(RED);
    if(col==17)    setcolor(BLUE);
    if(col==18)    setcolor(GREEN);
    if(col==19)    setcolor(DARKGRAY);
    if(col==20)    setcolor(BROWN);
    if(col==21)    setcolor(YELLOW);
    if(col==22)    setcolor(MAGENTA);
}

void redo(fstream &f, Obj &O, int &mx, int &my) {
    cleardevice();
    setbkcolor(CYAN);
    draw(mx, my);

    f.seekg(0);
    f.read((char*)&O, sizeof(O));

    setcolor(WHITE);
    while(!f.eof()) {
        if(O.z==7) {
            mousehide();
            drawgraph(mx, my);
            callmouse();
            break; }
        f.read((char*)&O, sizeof(O));
    }
    f.clear();
    f.seekg(0);
    setcolor(WHITE);

    while(!f.eof()) {
        switch(O.z) {
            case 3:line(O.x1, O.y1, O.x, O.y);
                break;
            case 4:rectangle(O.x1, O.y1, O.x, O.y);
                break;
            case 5:ellipse(O.x1, O.y1,0, 360, O.x, O.y);
                break;
            case 6:circle(O.x1, O.y1, O.x);
                break;
            case 9:drawpoly(O.x, O.P);
                break;
            case 10:outtextxy(O.x, O.y, ".");
                break;
            case 14:setbkcolor(O.x);
                break;
            case 15:
            case 16:
            case 17:
            case 18:
            case 19:

```

```

        case 20:
        case 21:
        case 22: colour(O.z);
            break;
        case 23: cleardevice();
            setbkcolor(3);
            setfillstyle(SOLID_FILL, 15);
            draw(mx, my);
            break;
    }
    f.read((char*)&O, sizeof(O));
}
int k = f.tellg();
f.clear();
f.seekg(k); }

void intro() {
    int midx=300, midy = 200;
    setbkcolor(0);
    settextstyle(SANS_SERIF_FONT, HORIZ_DIR, 1);
    setusercharsize(1,2,3,4);
    for(int k = 1; k<=10 && kbhit() == 0; k++) {
        if(k != 4 && k!= 5) setfillstyle(k,WHITE +k);
        else setfillstyle(k-2, WHITE + k);
        for(int i = 0; i <= 50; i++) {
            fillellipse(midx, midy,100 +i,50 +i);
            delay(10); }
        setcolor(RED + k);

switch(k) {
    case 1:
    case 7: outtextxy(midx - 25, midy, "WELCOME");
        break;
    case 2:
    case 8: outtextxy(midx , midy, "TO");
        break;
    case 3: outtextxy(midx - 85, midy + 10, "Paint to your
hearts content");
    case 9: setcolor(WHITE);outtextxy(midx - 25, midy - 15,
"MAD PAINTER");
        delay(1500);
        break;
    case 4: setcolor(YELLOW); outtextxy(midx - 25, midy,
"PROEJCT");
        break;
    case 5: outtextxy(midx , midy, "BY");
        break;
    case 6: outtextxy(midx-25 , midy, "AKASH S");
        break;
    case 10:outtextxy(midx - 75, midy - 10, "THIS IS TOP
SECRET.");
        outtextxy(midx - 140, midy + 10, "SO PUT YOUR NEXT
WORDS IN COLOUR :>");
        break;
    }
    delay(500);
}

```

```

    }

    delay(1500);
    cleardevice();

    setbkcolor(BLACK);
    setcolor(14);

    settextstyle(DEFAULT_FONT, HORIZ_DIR, 7);
    setusercharsize(3,1,4,3);
    outtextxy(220, 100, "MAD");
    outtextxy(130, 300, "PAINTER");

    settextstyle(DEFAULT_FONT, HORIZ_DIR, 2);
    setusercharsize(2,1,3,2);
    setcolor(15);
    outtextxy(220, 400, "Akash S");
    setcolor(11);
    outtextxy(245, 415, "12 B");
    getch();
    cleardevice();
    setfillstyle(1, WHITE);
}

int main() {

    int gdriver = DETECT, gmode, errorcode;
    int mx, my, i;

    initgraph(&gdriver, &gmode, "C:\\\\turbo3\\\\bgi");

    errorcode = graphresult();
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        return 1;
    }

    mx=getmaxx();
    my=getmaxy();

    setbkcolor(2);

    i=SOLID_FILL;
    setfillstyle(i, 15);

    remove("tmp.dat");
    fstream f("tmp.dat", ios::binary| ios::in| ios::out);
    if (!f) { printf("tmp.dat not found!!!"); return 0; }

    intro();
    setbkcolor(3);

    settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);

```

```

setusercharsize(1,1,1,1);

int a,b,col = 15;

a=100; b=370;
setposi(a,b);

do{

setcolor(RED);
int x, y, cl, z=0;

callmouse();
draw(mx, my);

do {
    mouseposi(x, y, cl);
    if(cl==1) {
        a=x; b=y;
        if(y > my-35 && y < my-10) {
            if(x > mx-600 && x < mx-510) z=1;
            if(x > mx-500 && x < mx-410) z=2;
            if(x > mx-400 && x < mx-310) z=3;
            if(x > mx-300 && x < mx-210) z=4;
            if(x > mx-200 && x < mx-110) z=5;
            if(x > mx-100 && x < mx-10) z=6; }

            else if(y > my-65 && y < my-40) {
                if(x > mx-600 && x < mx-510) z=12;
                if(x > mx-500 && x < mx-410) z=11;
                if(x > mx-400 && x < mx-310) z=10;
                if(x > mx-300 && x < mx-210) z=9;
                if(x > mx-200 && x < mx-110) z=8;
                if(x > mx-100 && x < mx-10) z=7; }

            else if(y > 40 && y < 65) {
                if(x > mx-300 && x < mx-210) z=23;
                if(x > mx-200 && x < mx-110) z=14;
                if(x > mx-100 && x < mx-10) z=13;
                }

            else if(x > my-600 && x < mx-585) {
                if(y > my-90 && y < my-80) z=15;
                if(y > my-100 && y < my-90) z=16;
                if(y > my-110 && y < my-100) z=17;
                if(y > my-120 && y < my-110) z=18;
                if(y > my-130 && y < my-120) z=19;
                if(y > my-140 && y < my-130) z=20;
                if(y > my-150 && y < my-140) z=21;
                if(y > my-160 && y < my-150) z=22; }

            else z = cl = 0;
        }

        setcolor(RED);
    }
}

```



```

        gotoxy(1, 2);
        printf("\nmouse position is: %d, %d", x, y);
        printf("\nclick %d", cl);
        //printf("\nz is %d", z);
    }while(!cl);
    cl=0;
    int x1, y1;
    delay(500);
    setcolor(BLUE);

    Obj O;
    O.z=z;

    if(z==1) {
        return 1;
    }

    if(z==2) {
        undo(f, O);
        redo(f, O, mx, my);
    }

    if(z==3) {
        mousehide();
        outtextxy(mx-370, my-30, "Line");
        callmouse();

        mtrav(x, y, cl, z);
        x1 = x;    y1 = y;
        cl=0;
        delay(500);

        mtrav(x, y, cl, z);
        colour(col);
        line(x1, y1, x, y);

        O.x1 = x1;    O.y1 = y1;
        O.x = x;      O.y = y;

        f.write((char*)&O, sizeof(O));
    }

    if(z==4) {
        mousehide();
        outtextxy(mx-290, my-30, "Rectangle");
        callmouse();

        mtrav(x, y, cl, z);
        x1 = x;    y1 = y;
        cl=0;
        delay(500);

        mtrav(x, y, cl, z);

        mousehide();
        colour(col);
    }

```

```

rectangle(x1, y1, x, y);
callmouse();

O.x1 = x1;    O.y1 = y1;
O.x = x;      O.y = y;

f.write((char*)&O, sizeof(O));
}

if(z==5) {
    mousehide();
    outtextxy(mx-175, my-30, "Ellipse");
    callmouse();

    mtrav(x, y, cl, z);
    x1 = x;    y1 = y;
    cl=0;
    delay(500);

    mtrav(x, y, cl, z);
    int x2 = fabs(x-x1);
    cl=0;
    delay(500);

    mtrav(x, y, cl, z);
    int y2 = fabs(y1-y);

    mousehide();
    colour(col);
    ellipse(x1, y1, 0, 360, x2, y2);
    callmouse();

    O.x1 = x1;    O.y1 = y1;
    O.x = x2;      O.y = y2;

    f.write((char*)&O, sizeof(O));
}

if(z==6) {
    mousehide();
    outtextxy(mx-70, my-30, "Circle");
    callmouse();

    mtrav(x, y, cl, z);
    x1 = x;    y1 = y;
    cl=0;
    delay(500);

    mtrav(x, y, cl, z);
    int xr=sqrt(pow(x-x1,2) + pow(y-y1,2));

    mousehide();
    colour(col);
    circle(x1, y1, xr);
    callmouse();
}

```

```

O.x1 = x1;    O.y1 = y1;
O.x = xr;

f.write((char*)&O, sizeof(O));
}

if(z==7) {
    mousehide();
    outtextxy(mx-90, my-60, "Open Graph");
    callmouse();
    delay(500);

    f.write((char*)&O, sizeof(O));
    redo(f, O, mx, my);
}

if(z==8) {
    mousehide();
    outtextxy(mx-196, my-60, "Close Graph");
    callmouse();

    int m = f.tellg();
    fstream g("tmp2.dat", ios::binary| ios::in| ios::out);

    f.seekg(0);

    while(f.tellg() < m) {
        f.read((char*)&O, sizeof(O));
        if(O.z != 7)
            g.write((char*)&O, sizeof(O));
    }

    f.close(); g.close();
    remove("tmp.dat");
    rename("tmp2.dat", "tmp.dat");

    fstream f("tmp.dat", ios::binary| ios::in| ios::out);
    if(!f) {
        cout<<"ERROR IN OPENING FILE. PLEASE RETRY LATER.";
        getch();
        return 0;}

    redo(f, O, mx, my);
}

if(z==9) {
    mousehide();
    outtextxy(mx-290, my-60, "Polygon");
    callmouse();
    delay(500);

    cleardevice();
    closegraph();

    int n;

```

```

do {
    cout<<"Number of points: ";
    cin>>n;

    if(n>15 || n<3) {
        cout<<"PLEASE ENTER A VALID NUMBER OF POINTS BETWEEN
3 AND 15";
        getch(); }

    }while(n > 15 || n <3);

    initgraph(&gdriver, &gmode, "C:\\\\turbo3\\bgi");
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    {
        printf("Graphics error: %s\n",
grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        return 1; /* terminate with an error code */
    }

    redo(f, O, mx, my);

    callmouse();

    int P[30];
    for(int i=0; i<n; i++) {
        mtrav(x, y, cl, z);
        cl=0;
        delay(500);
        P[2*i + 0]= x;
        P[2*i + 1]= y;
    }
    P[2*i + 0]= P[0];
    P[2*i + 1]= P[1];

    mousehide();
    colour(col);
    drawpoly(n+1, P);
    callmouse();

    O.x = n+1;
    for(i = 0; i<2*n; i++) O.P[i] = P[i];
    f.write((char*)&O, sizeof(O));

}

if(z==10) {
    mousehide();
    outtextxy(mx-390, my-60, "Free Hand");
    callmouse();

    mtrav(x, y, cl, z);

    while(cl == 1) {

```

```

        colour(col);
        mouseposi(x,y,cl);
        mousehide();

        outtextxy(x-1, y-1, ".");
        callmouse();

        setcolor(RED);
        gotoxy(1, 2);
        printf("\nmouse position is: %d, %d", x, y);
        printf("\nclick %d", cl);
        //printf("\nz is %d", z);

        O.z = 10;
        int flag =0;
        f.seekg(0, ios::end);
        if ( f.tellg() ==0) {
            printf("%d", O.x);
            O.x = x; O.y = y;
            flag =1;}

        else
        { f.seekg(sizeof(O), ios::end);
          f.read((char*)&O, sizeof(O));
            if((O.x != x && O.y != y)) {
                O.x = x; O.y = y;
                flag=1;
            }
        }
        if(flag==1)
        {
            f.close();
            f.open("tmp.dat",ios::app|ios::binary);
            f.write((char*)&O, sizeof(O));
            f.close();

f.open("tmp.dat",ios::in|ios::out|ios::binary);

        }
    }

    if(z==11) {
        mousehide();
        outtextxy(mx-490, my-60, "Save File");
        callmouse();
        delay(500);

        cleardevice();
        closegraph();

        char S[15];

        do {
            clrscr();
            cout<<"Name of file: ";

```

```

cin>>S;

if(strlen(S) > 8) {
    cout<<"\nINVALID NAME. PLEASE ENTER ONLY MAXIMUM OF
8 CHARACTERS.";
    getch(); }

}while(strlen(S) > 8);

strcat(S, ".dat");

fstream g(S, ios::binary| ios::in| ios::out| ios::trunc);
if(!g) {
    cout<<"ERROR IN OPENING FILE. PLEASE RETRY LATER.";
    getch();
    return 0; }

int k = f.tellg() - 1, i=0;

f.clear(); f.seekg(0);
f.read((char*)&O, sizeof(O));
while(i<k) {
    g.write((char*)&O, sizeof(O));
    f.read((char*)&O, sizeof(O));
    i+=sizeof(O);
}

O.z = -1;
g.write((char*)&O, sizeof(O));
g.close();

f.clear();
f.seekg(k);

initgraph(&gdriver, &gmode, "C:\\\\turbo3\\bgi");
errorcode = graphresult();
if (errorcode != grOk)
{
    printf("Graphics error: %s\n",
grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    return 1; /* terminate with an error code */
}

redo(f, O, mx, my);
}

if(z==12) {
    mousehide();
    outtextxy(mx-590, my-60, "Load File");
    callmouse();

    delay(500);

    cleardevice();

```

```

closegraph();

char S[14];
cout<<"Name of file: ";
cin>>S;

strcat(S, ".dat");

fstream g(S, ios::binary| ios::in| ios::out);
if(!g) {
    cout<<"ERROR IN OPENING FILE. PLEASE RETRY LATER.";
    getch();
    return 0;}

f.close(); remove("tmp.dat");
f.open("tmp.dat", ios::binary| ios::in| ios::out);

g.read((char*)&O, sizeof(O));
while(O.z != -1) {
    f.write((char*)&O, sizeof(O));
    g.read((char*)&O, sizeof(O));
}
g.close();

int k = f.tellg() - 1;
f.clear();
f.seekg(k);

initgraph(&gdriver, &gmode, "C:\\\\turbo3\\bgi");
errorcode = graphresult();
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n",
grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    return 1; /* terminate with an error code */
}

redo(f, O, mx, my);
}

if(z==13) {
    mousehide();
    outtextxy(mx-90, 50, "HELP");
    callmouse();
    delay(500);

    cleardevice();
    closegraph();

    char K[50];

    ifstream h("projhelp.txt");
    if(!h) {
        cout<<"ERROR IN OPENING FILE. PLEASE RETRY LATER.";

```

```

        getch();
        return 0;}

h.getline(K, 50, '\n');
while(!h.eof()) {
    cout<<K<<"\n";
    h.getline(K, 50, '\n'); }
cout<<"Press any key to exit.";
getch();

initgraph(&gdriver, &gmode, "C:\\\\turbo3\\bgi");
errorcode = graphresult();
if (errorcode != grOk) /* an error occurred */
{
    printf("Graphics error: %s\n",
grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    return 1; /* terminate with an error code */
}

redo(f, 0, mx, my);
}

if(z==14) {
    mousehide();
    outtextxy(mx-190, 50, "Colour");
    callmouse();
    delay(500);

    cleardevice();
    closegraph();

    char col;

    do {
        clrscr();
        cout<<"\n\n\t\t\tMENU          "
            <<"\nBLACK          3    0    "
            <<"\nBLUE           3    1    "
            <<"\nGREEN           3    2    "
            <<"\nCYAN           3    3    "
            <<"\nRED            3    4    "
            <<"\nMAGENTA        3    5    "
            <<"\nBROWN         3    6    "
            <<"\nLIGHTGRAY      3    7    ";
        cout<<"\nChoose your colour: ";
        cin>>col;

        if(col > '7' || col < '0') {
            cout<<"\nPLEASE ENTER A VALID COLOUR NUMBER";
            getch(); }

    }while(col > '7' || col < '0');
}

```



```

        cout<<"\n Your change will be made when you press any
key. ";

        getch();

        initgraph(&gdriver, &gmode, "C:\\\\turbo3\\\\bgi");
        redo(f, 0, mx, my);
        setbkcolor(col-(int)'0');

        O.x = col - (int)'0';
        f.write((char*)&O, sizeof(O));
        f.close();
    }

    if(z>=15 && z<=22) {
        col = z;
        f.write((char*)&O, sizeof(O));
        colour(col);
    }

    if(z==23) {
        outtextxy(mx-222, 45, "+");
        delay(500);
        cleardevice();
        setbkcolor(3);

        i=SOLID_FILL;
        setfillstyle(i, 15);

        setcolor(RED);
        callmouse();
        draw(mx, my);
        f.write((char*)&O, sizeof(O));
    }

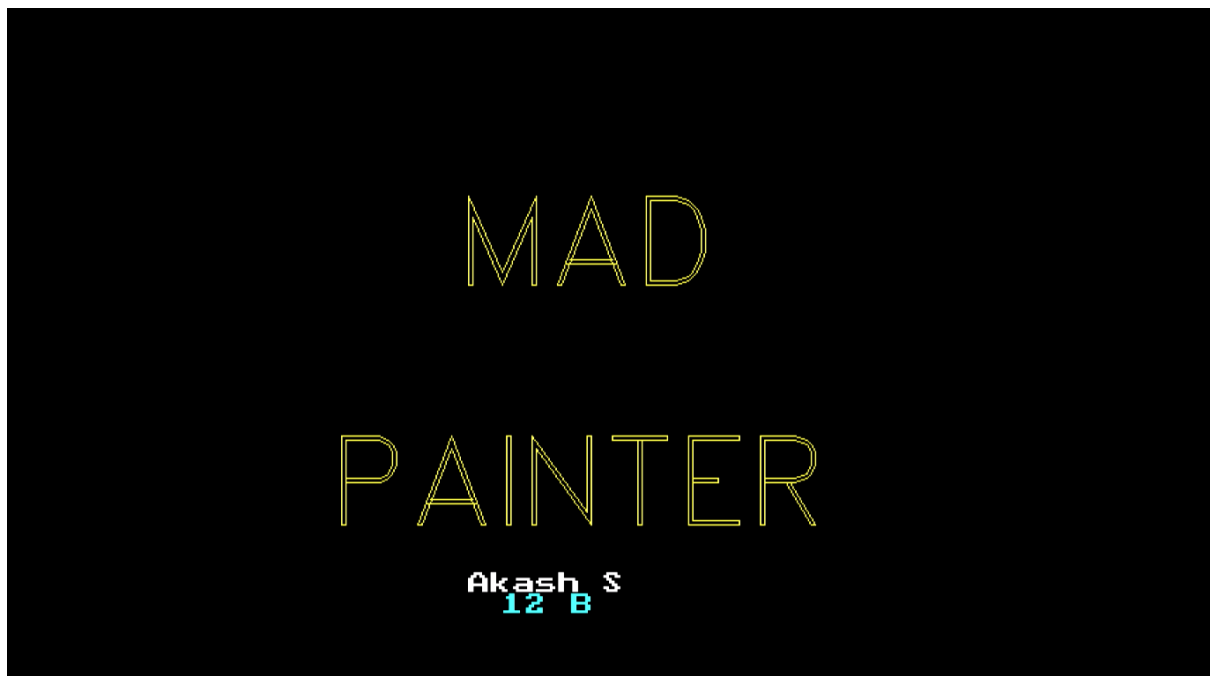
    delete O.P;
}while(!kbhit());

cleardevice();
closegraph();
f.close();
return 1;
}

```

SAMPLE SCREENSHOTS

These are a few of the output screens.



Help...

1. Line: Click on the two end points of the line that you would like to draw.
2. Circle: Click on the centre and the circle and any point on the circumference.
3. Ellipse: Click on the centre. Then on how much you want to extend it to the right and then on how much you want to extend it above.
4. Rectangle: Click on any two opposite corners.
5. Undo: Enables you to undo your last action.
6. Open Graph: Opens a graph sheet on the screen.

7. Close Graph: Closes a graph sheet that has been opened on the screen.

8. Free Hand: Allows you to draw to your hearts content on the screen as long as the mouse is clicked.

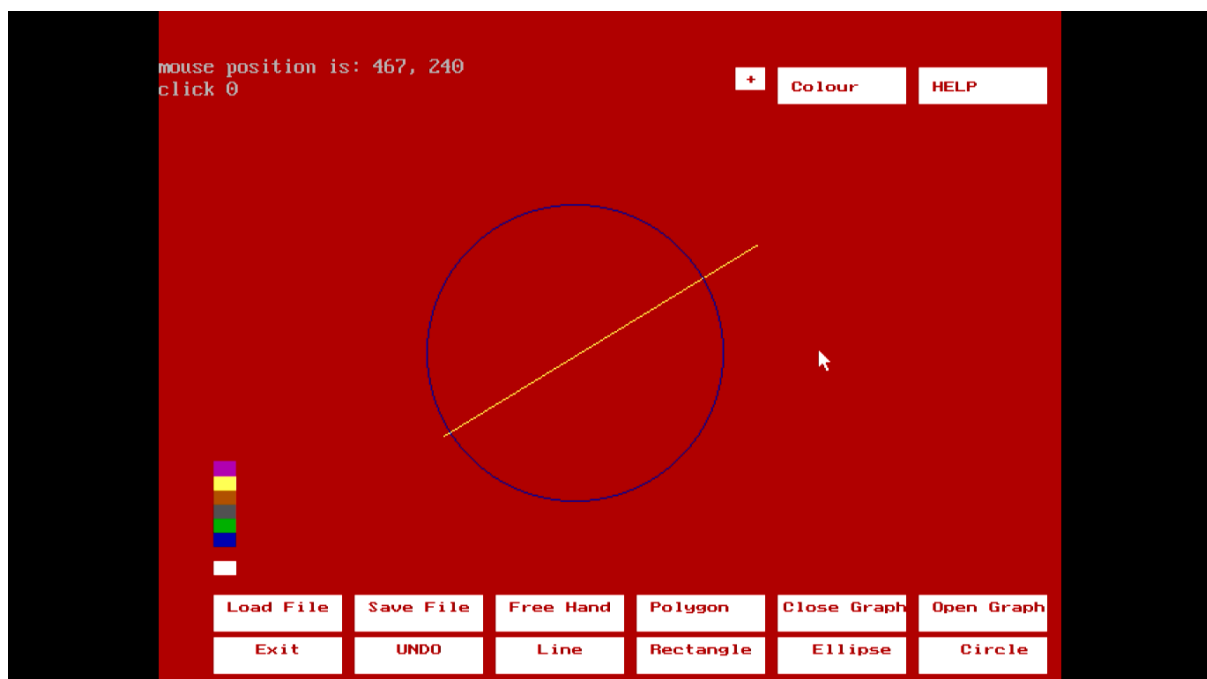
9. Polygon: Takes in number of points and allows you to click that many times on the screen. Draws a polygon connecting all those points.

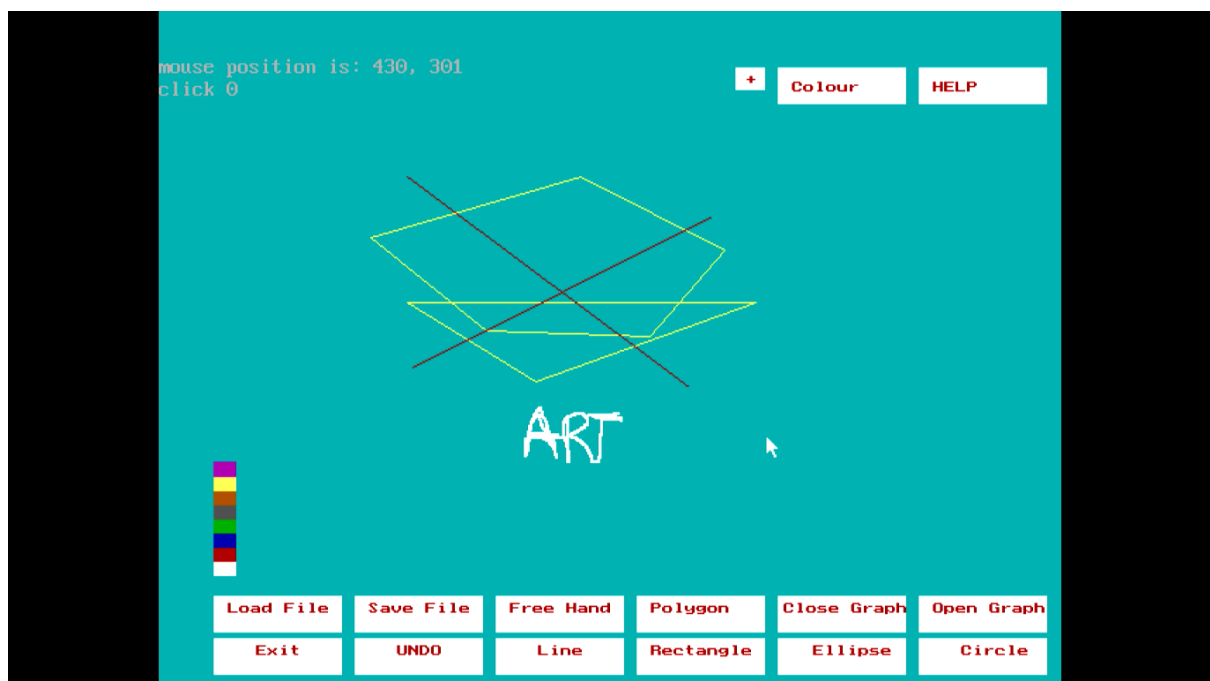
10. Save File: Allows you to save the file .

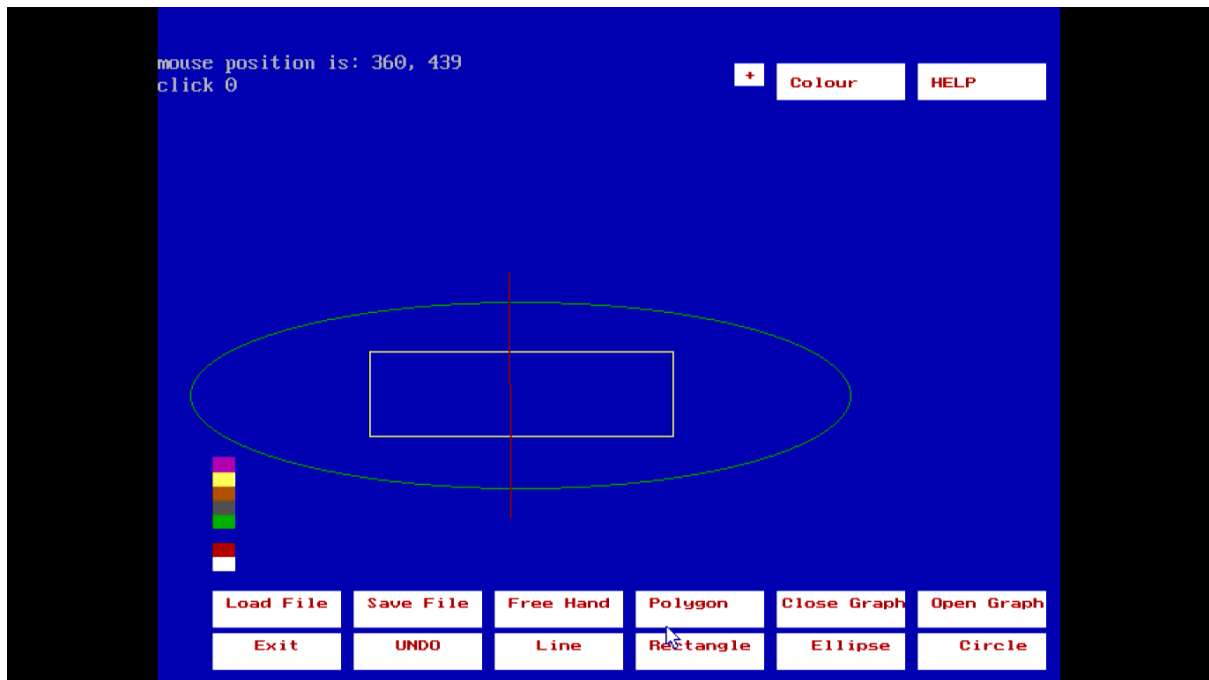
11. Load File: Allows you to load the file.

12. Colour: Allows you to set the bk colour.

Press any key to exit.







LIMITATIONS OF MY PROJECT

My project encompasses many aspects of data file handling and graphics in Turbo C++. However, it still faces some drawbacks.

The first one is a limitation of C++. Due to some unforeseen error which I suppose has its roots in memory overloading or hogging the RAM by calling too many interrupts, the same script works for the first few loops and then may malfunction after some three to four minutes into the program.

The second one is the problem of a static screen. You cannot move any of the sprites on the screen once they are drawn. It is not possible to scroll infinitely, or zoom in and out. However, new pages can be created if need be.

The third, is the need to re-click the function you want to work with after every use.

SCOPE FOR IMPROVEMENT

My project still has a long way to go.

The first scope for improvement lies in clearly defining the objects on the screen and separating the workspace from the toolbars. Then even the need to re-click the function every time can be avoided.

The second scope lies in the number of features it has to offer, like the textbox tool, the select tool or even the clipboard that helps to copy a feature and paste it. This can be marginally improved even under the restrictions of C++.

The page scroll and zoom can also be added with some additional programming.

BIBILIOGRAPHY

These are a few of the sites that were of great help to me.

<https://www.go4expert.com/articles/mouse-programming-c-t21153/>

<https://stackoverflow.com/questions/22259936/c-move-mouse-in-windows-using-setcursorpos>

<https://www.youtube.com/watch?v=J2sCngMdWEA>

... and the HELP Index in Turbo C++.