Table of contents

Abstract

This project involves the creation of an interactive map of Pulchowk Campus using the C programming language and the graphic.h library. The map displays the various buildings and facilities on campus, along with their corresponding locations. The project was developed with the aim of providing an efficient and user-friendly and fun way for new students, visitors, or anyone who wants to explore the campus to easily navigate and locate various buildings and facilities on campus.

The map is implemented using graphics, looping and switch concepts in C programming. The graphic.h library, which is a popular library for graphics in C, was used to create the visual interface for the map. The user interface includes menus and use input from keyboard to navigate the map.

In summary, this project provides an efficient and user-friendly way to navigate the campus of Pulchowk Campus using an interactive map implemented in C programming language using the graphic.h library. The project showcases the use of looping, nested looping and array in C programming, and can be beneficial to anyone who wants to create similar projects in the future.

Acknowledgement

We would like to express our sincere thanks to our supervisor, Dr. Ganesh Gautam and our lab teachers, for their valuable guidance, support, and encouragement throughout the development of this project. We would also like to thank our families, friends and well wishers for their unwavering support and motivation, and for their understanding and patience during our long hours of work. Finally, we would like to acknowledge the support of the Pulchowk Campus for providing us with access to the necessary resources and facilities that made this project possible.

Finally, We would also like to acknowledge the support of the online forums and communities, whose members have generously shared their expertise and insights with me, and whose discussions have enriched my understanding of programming concepts and techniques.

Best regards,

Saurya Shrestha	(078BAS039)
Shubham Ranabhat	(078BAS040)
Saurav Sen Khawas	(078BAS038)
Sandesh Kunwar	(078BAS048)

Introduction

The purpose of this project is to create a map of our Pulchowk Campus using the C programming language and the graphic.h library. The objective of the project is to provide an interactive and visual representation of the campus, allowing students and visitors to easily navigate and locate different buildings and facilities. We chose this project because we wanted to apply our knowledge of C programming and graphics to create a useful and practical tool for our university community plus we were chasing for old school DOS type of user interface. We used Turbo C++, visual studio code IDE and graphics.h library to create the project, and we faced several challenges during the development process. This report will provide an overview of our project design, implementation, testing, and results, as well as our conclusions and areas for future improvement.

General theory

C Programming Language

C is a general-purpose programming language that was developed in the early 1970s by Dennis Ritchie at Bell Labs. It is a low-level language that provides a small set of powerful and flexible constructs for writing efficient and portable programs. C has had a significant influence on the development of many other programming languages, including C++, Java, and Python.

Control statement:

Logical operation is carried out by several symmetrical or logical statements. There are two types of control statement based on their function.

Selective structure:

Selective structures are used when we have a number of situations where we need to change the order of execution of statements based on certain condition. The selective statements make a decision to take the right path before changing the order of execution. C provides the following statements for selective structure:

- If statements
- Switch statements

If statements:

It is a powerful decision-making statement and it is used to control the flow of execution of statements. It is a two-way statement and is used in conjunction with an expression. If statement allows the computer to evaluate the expression first and then on depending whether the value of the expression is true or false it transfers the control to the particular statement. At this point of the program has two paths to follow: one for true condition and other for false condition. The types of if statements are explained below:

- Simple if statement
  The simple if statement is used to conditionally excite a block of code based on whether a test condition is true or false. If the condition is true the block of code is executed, otherwise it is skipped. The syntax of if statement is given below:

```
if (test expression)
{
        statement-block;
}
statement-x;
```

- if else statement
  The if else statement extends the idea of the if statement by specifying another section of code that should be executed only if the condition is false i.e conditional branching. True-block statements are to be executed only if the test expression is true and false block statements to be executed only if the condition is false. The syntax of if else statement is given below:

```
if (test expression)
{
        true block statement;
}
Else
{
        false block statement;
}
```

Switch statement:

C has built in multi way decision statement known as switch. It successively tests the value of an expression against a list of case values (integer or character consonants). When the match is found the statement associated with that case is executed. The syntax of switch expression is given below:

```
switch(expression)
{
    case constant-1:
    block-1;
    break;
    case constant-2:
    block-2;
    break;
   …………
```

```
        ………….

        case constant-2:

        block-n;

        break;

        default:

        default statement:

}
```

LOOPING

Loop causes a section of code to be repeated for a specified number of times or until some condition holds true. When a condition becomes false, the loop terminates and control passes to statement below the loop. Different types of loops are discussed below with their major characteristics and syntax used in C:

- While loop: The while loop specifies that a section of code should be executed while a certain condition holds true. The syntax of while loop is given below:

  ```
  While (test expression)

  {

          body of loop

          (Statement-block)

  }
  ```

- do while statement

  The do while statement is very similar to while statement. It also specifies that a section of code should be executed while a certain condition holds true. the difference between while and do while loop is that while loop test its condition at the top of its loop but do while loop tests its condition at the bottom of loop. In while loop, if the test condition is false, the block of code is skipped. Since condition is tested at the bottom of loop in do while loop, its block of code is always executed at least once. The syntax of do while loop is given below:

```
do
{
        body of loop
}
while (test expression);
```

- for loop

  The for loop is used to execute a block of code for a fixed number of repetitions. Initialization is generally an assignment statement used to set loop control variable. Test expression is a relational expression that determines when loop exits. Update expression defines how the loop variable changes each time when the loop is repeated. The syntax of for loop is given below:

  ```
  for (initialization expression; test expression; update expression)
  {
          body of loop;
  }
  ```

Break statement

The break statement is used to jump out of loop. The break statement terminates the execution of the nearest enclosing loop. Control passes to the statement that follows the terminated statement. in a switch break statement causes the program to execute the next statement after switch.

```
break;
```

Function

A function is a self-contained program segment that carries out some specific well-defined task . Every C program consists of one or functions. Execution of program always begins by carrying out instruction in main. Function makes program significantly easier to understand and maintain. A well written function may be reused in multiple programs. Program that are easier to design, debug and maintain.

Return statement

A function may or may not send back any value to the calling function. If it does, it is through return statement. The called function can only return only one value per call at most. The syntax of return statement is given below:

Return;

Pointer

A pointer is a variable that represents the location (rather than value) of a data item, such as a variable or an array element. Pointers can be used to pass information back and forth between a function and a reference point. Pointer provides a way to return multiple data items from a function via function argument. When a pointer variable is declared, the variable name must be preceded by an aesteric (*).The syntax of a pointer declaration is:

```
 data type *ptar;
```

Structure

It is a heterogeneous user defined data type. It is also called constructed data type. It may contain different data types. Structure can also store non homogenous data type into a single collection. Structure may contain pointer, arrays, or even other structures other than the common data types such as int, float, long int etc. A structure provides a means of grouping variables under a single name for easier handling and identification. It can be defined as new named types. It is a convenient way of grouping several pieces of related information together. Complex hierarchies can be created by nesting structures. Structures may be copied to and assigned. They are also useful in passing groups of logically related data into structures. The declaration of structures is given below:

```
        struct tag
        {
                member 1;
                member 2;
                member n;
        }
```

File

Many applications require that information be written to or read from an auxiliary memory device. Such information is stored on the memory device in the form of a data file. The data files allow us to store information permanently and to access and alter that information whenever necessary.

- Opening a file
  Before performing any input / output operation, file must be opened. While opening file, the following must be specified:

i) The name of file.
ii) The manner in which it should be opened (that for reading, writing, both reading and writing, appending at the end of file, overwriting the file)

iii) when working with a stream-oriented data file, the first step is to establish a buffer area, where information is temporary stored while being transferred between the computer's memory and data-file. The buffer area is established by writing.

FILE *ptvar;

File is a special structure type establishes the buffer area and ptvar is a pointer variable that indicates the beginning of the buffer area the library function fopen is used to open a file .This function is used to open a file .This function is typically written as

ptvar=fopen (file name, file type);

where file name and file type are strings that represent the name of the data file and the manner in which the data file will be utilized.

Finally, a file can be closed at the end of the program. This can be accomplished with the library function fclose. The syntax is simply,

fclose(ptvar);

- Processing a file:
  Most data file application requires that a data file be altered as it is being processed. For example, in an application involving the processing of customer records, it may be desirable to add new records to the file. To delete the existing records, to modify the contents or to rearrange the records.

- File types:
  DOS treats files in two different ways that as binary or text file. Files almost all UNIX systems do not make any distinction between the two. If a file is specified as the binary type, the file I /O functions do not interpret the contents of the file when they read from, or write to the file. But if the file is specified as the text type, the file I / O functions interpret the contents of the file.

Even though we haven't used pointers, structure and file handling, their little description is provided as it comes under C programming language

Header Files

In C programming language, a header file is a file that contains declarations of functions, variables, and data types that are used in other parts of the program. The purpose of header files is to allow multiple source code files to share common functions, variables, and data types, without the need to redefine them in each file.

Header files typically have a ".h" extension and are included in a C program using the "#include" directive. The contents of the header file are then "copied" into the main program at the point of the #include directive, so that the declarations can be used by the main program. While making this project we have use 4 different header files which are mentioned below:

1. stdio.h
   In C programming language, stdio.h is a standard header file that provides functions for input and output operations, such as reading from and writing to files, the console, or other streams. The name "stdio" stands for standard input/output, and this header file is included in most C programs by default.

- Sprint();
  Sprint stands for string print . Instead of printing on console , it stores output on char buffer which are specified in sprintf.
  Syntax: sprint(char *str, const char *str,…);
- conio.h
  conio.h is a header file that provides functions for console input/output operations in C programming language. The name "conio" stands for console input/output.
  conio.h is not a part of the standard C library, but it is widely available on various platforms, including DOS, Windows, and some Unix-based systems. It provides a set of functions that allow programmers to perform console input/output operations, such as reading keyboard input and displaying text on the console.
  Some of the used functions in conio.h for this project include:
  - getch()
    getch() is a function provided by conio.h header file in C programming language. It is used to read a single character of input from the keyboard without displaying it on the screen.
    The getch() function reads a single character from the keyboard buffer and returns its ASCII value as an integer. The function does not wait for the user to press the Enter key, so it can be used to read single keystrokes or to implement interactive menus and other console-based applications.
  - kbhit()

kbhit() is a function provided by the conio.h header file in C programming language. It is used to determine if a key has been pressed on the keyboard or not, without actually waiting for the user to press a key.
The kbhit() function returns a non-zero value (true) if a key has been pressed on the keyboard, and a zero value (false) otherwise

2. dos.h
The dos.h header file is a header file in C that provides access to some functions and variables that deal with low-level operations in a DOS environment. It is a non-standard header file that was commonly used in MS-DOS and early Windows programming.

3. graphic.h
graphics.h is a C programming language header file that contains functions for creating and manipulating graphical images on the screen. It is a non-standard header file and is not part of the standard C library.
graphics.h provides a set of functions for drawing basic geometric shapes like lines, circles, rectangles, and polygons on the screen, as well as more advanced features like text fonts and image rendering, color manipulation, and input handling.
Some of the functions used in graphic.h for this project includes:

- graphdriver= …
  This statement is used to assign a graphic driver that the program is going to use. Putting the value to DETECT will allow the computer to search for adapter and use it for the program.

  Syntax: int graphdriver = DETECT;

- graphmode=....
  This statement is used to declare a mode the adapter is going to run on.
  Syntax: int graphmode;
- initgraph(....)
  The default mode of display devices is in text mode when we boot a computer. It is necessary to switch to graphics mode to display output of graphics program. The function initgraph is used to initialize graphics hardware. Three arguments needs to be passed to this function. The arguments are driver, mode, and path of driver.
  Syntax: initgraph(&graphdriver,&graphmode,"C:\\TURBOC3\\BGI");
- closegraph();
  This function is used to exit the graphics mode.
- line(int a,int b,int c,int d);

This function is used to display a line on the screen. (a,b) and (c,d) are the two end points of a line.
Syntax: line(30,40,70,90);

- rectangle(int a,int b,int c,int d);
  This function is used to display a rectangle on the screen. (a,b) and (c,d) represents the coordinates of leading diagnol.
  Syntax :(10,20,40,50);
- circle(int h,int k,int r);
  This function is used to display a cirlce on the screen. (h,k) represents the center of the circle and r represents the radius.
  circle(100,100,5);
- arc(int h,int k,int stangle,int enangle,int r);
  This function is used to display an arc on the screen. (h,k) reprenents the center, stangle and enangle represents starting and ending angle and r represents radius. The arc is drawn in counterclockwise direction.
  Syntax: arc(30,60,90,270,10);
- ellipse(int h,int k,int stangle,int enangle,int r1,int r2);
  This function is used to display an ellipse on the screen. (h,k) represents the centre of ellipse, stangle and enangle represents start angle and end angle, r1 and r2 represents radius in x and y axes. The ellipse is drawn counterclockwise.
  Syntax: ellipse(40,50,0,360,5,10);
- outtextxy(int x,int y,"Your text here");
  This function is used to display a text at specific part of the display screen.
  Syntax: outtextxy(50,90,"C PROGRAMMING")
- settextstyle(int a, int b, int c);
  This function is used to change the font and font size of a text. Three arguments a, b and c represents the font, orientation and font size.
  Syntax: settextstyle(4,0,4);
- setlinestyle(int a, int b, int c);
  This function is used to change the appearence of a line. Three arguments a,b and c represents type of line, orientation and width of line.
  Syntax: setlinestyle(0,0,3);
- setcolor(int a);
  This function is used to select the drawing colour. There are variety of colours that can be accessed by their unique number.
  Syntax: setcolor(3);
- setbkcolour(int a);
  This function is used to change the colour of background display. The default is black.

Syntax:setblcolour(4);

- bar(int a ,in b, int c, int d);
  This function is used to draw a filled rectangular bar.
- cleardevice();
  This function is used to clear the display in graphics mode.

Project Design

Our project's user interface design is based on the use of graphics functions provided by the graphic.h library, such as the line(), rectangle(), arc() and circle() functions. We also implemented keyboard input handling using the getch() function. We used a simple algorithm to calculate the position of the buildings and facilities on the map based on their coordinates and dimensions. Overall, the design of our project aims to be simple, modular, and scalable, so that it can be easily extended or modified in the future.

Algorithm

- For WlcScreen function

```
Step 1   :   Start
Step 2   :   Declare integers progress=0 and width and character text
Step 3   :   Display "LOADING..." at (270,185)
Step 4   :   Display rectangle with the coordinates (200,200,400,225)
Step 5   :   width = progress * 2
Step 6   :   Change fill color to red
Step 7   :   Fill in the rectangle up-to (201,201,200 + width,224)
Step 8   :   Display progress % at (290,210)
Step 9   :   Delay next iteration by 50 milliseconds
Step 10  :   Add 1 to progress
Step 11  :   Run the loop till progress <= 100
Step 12  :   Stop
```

- For Map Function

```
Step 1   :   Start
Step 2   :   Display" MAP OF IOE PULCHOWK"
Step 3   :   Draw various lines resembling outline of map
Step 4   :   Call love garden function
                 Call white house function
                 …………………………………
                 ………………………………...
                 Call hostel function
Step 5   :   Draw remaining parts of map
Step 6   :   Stop
```

- For swich function

```
Step 1   :   Start
Step 2   :   Initialize a variable
Step 3   :   Take input from user
Step 4   :   Is input=esc then
             Break
             Else,
             case a
             case b
             case c
             …….
             …….
```

```
                case y
                case z
Step 5   :      Display output according to case
Step 6   :      End
```

- For Endscreen Function

```
Step 1   :      Start.
Step 2   :      Display "Thank you for visiting"
Step 3   :      Display "Credits"
Step 4   :      Display "Shubham Ranabhat, Saurya Shrestha, Saurav Sen Khawas,
                Sandesh Kunwar"
Step 5   :      Display "078BAS040, 078BAS039, 078BAS038, 078BAS048"
Step 6   :      Display "source code is available on:"
Step 7   :      Display GitHub link
Step 8   :      Stop
```

- For main function

```
Step 1   :      Start
Step 2   :      Call welcome screen
Step 3   :      Wait for input
Step 4   :      Clear display
Step 5   :      Call map
Step 6   :      Call switch
Step 7   :      Clear display ()
Step 8   :      Call end screen
Step 9   :      Wait for input
Step 10  :      Stop
```

- For Fix function

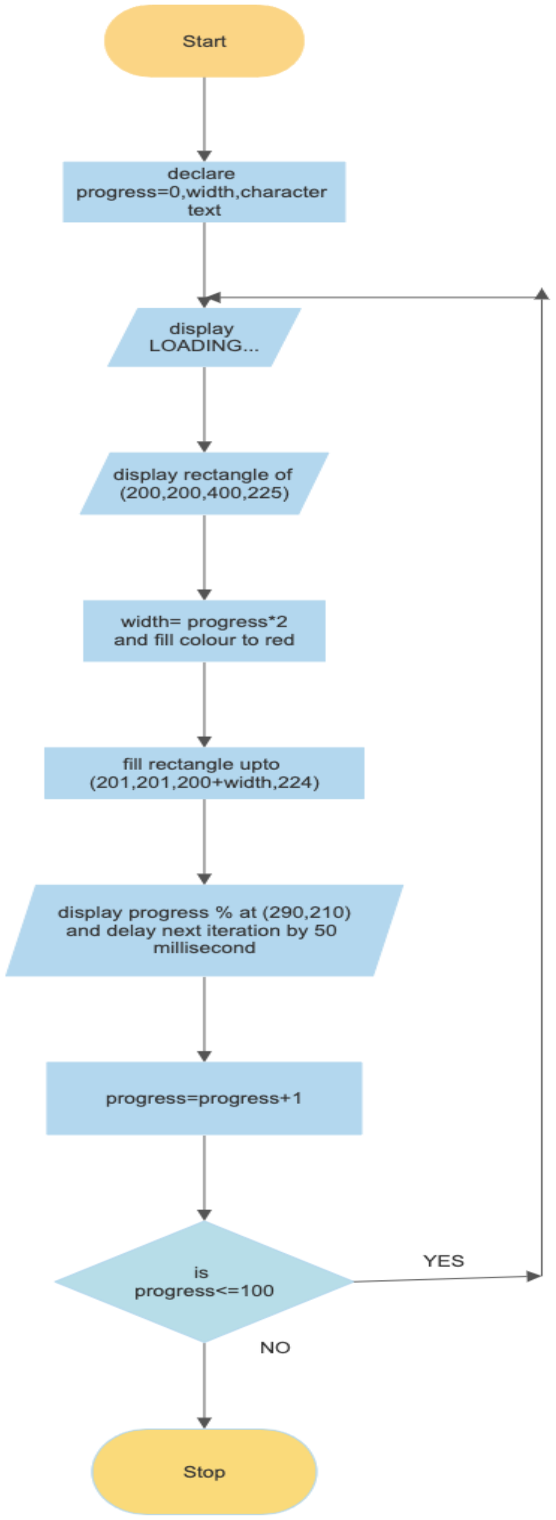
Step 1	:	Start
Step 2	:	clear screen
Step 3	:	Display Use your keyboard to select destination on your map
Step 4	:	End

Function such as Lovegarden, Whitehouse, ….. ….. ……, Girls, Staff all are based on similar algorithm which is presented below:

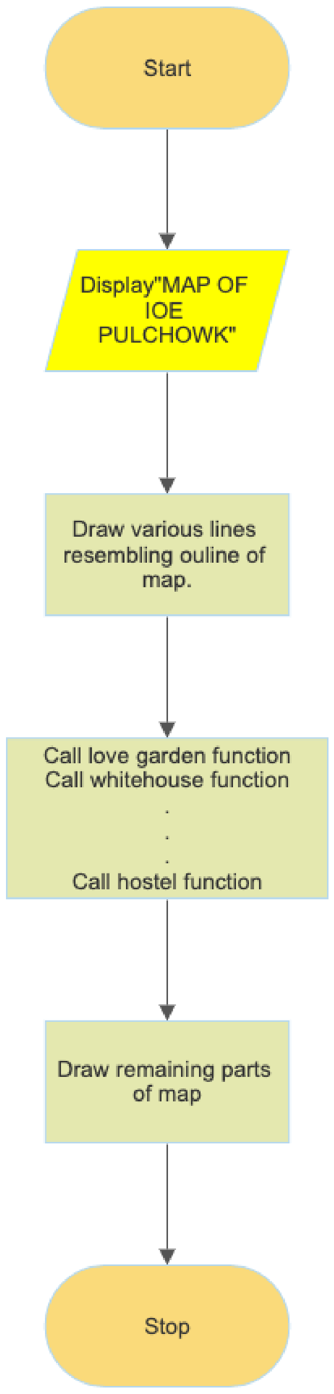
Step 1	:	Start
Step 2	:	Draw the resembling shape of building/ object
Step 3	:	Print the corresponding alphabet
Step 4	:	Clear display

Flowchart

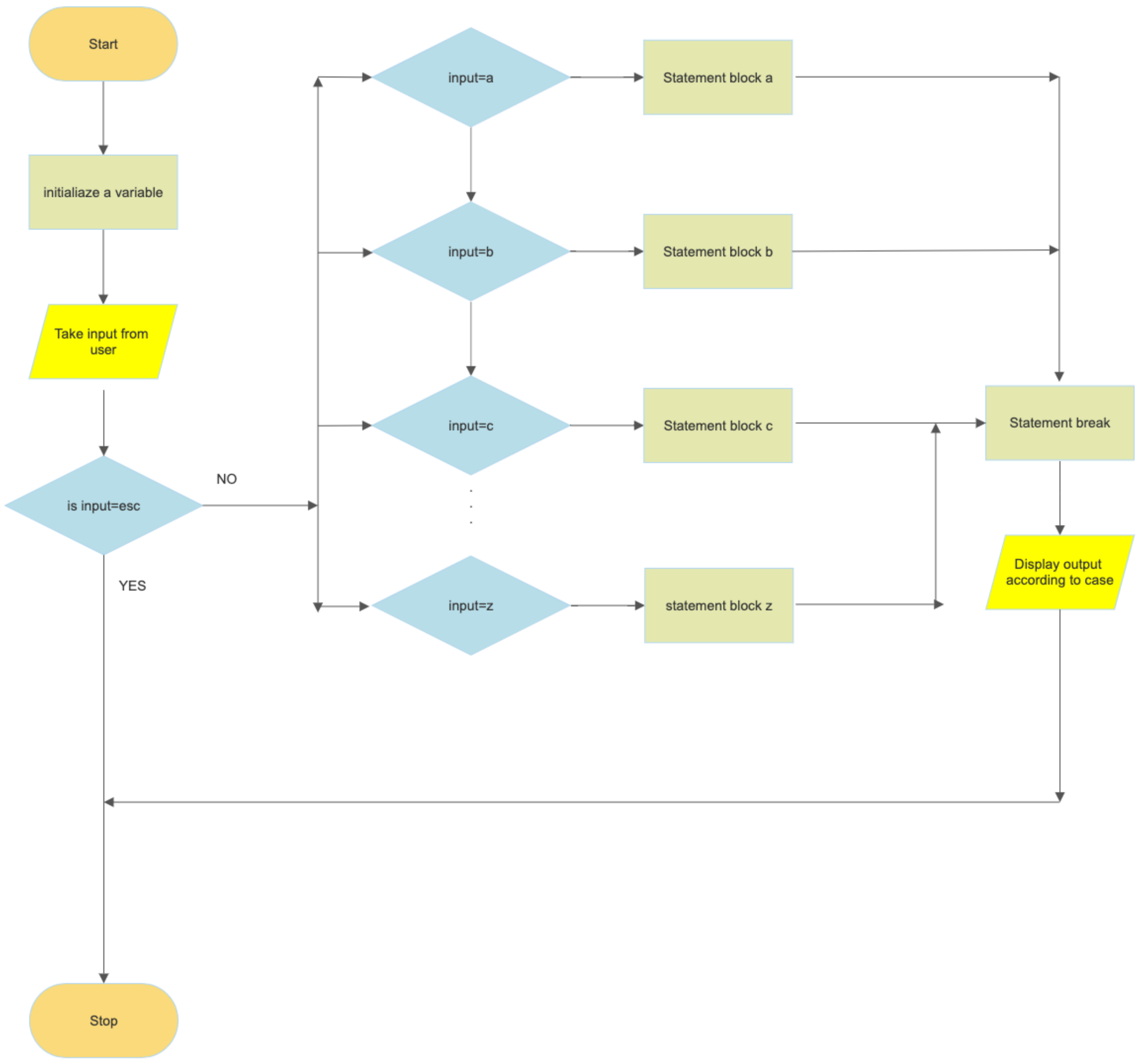
- For WlcScreen function


Start
declare progress=0,width,character text
display LOADING...
display rectangle of (200,200,400,225)
width= progress*2 and fill colour to red
fill rectangle upto (201,201,200+width,224)
display progress % at (290,210) and delay next iteration by 50 millisecond
progress=progress+1
is progress<=100
YES
NO
Stop

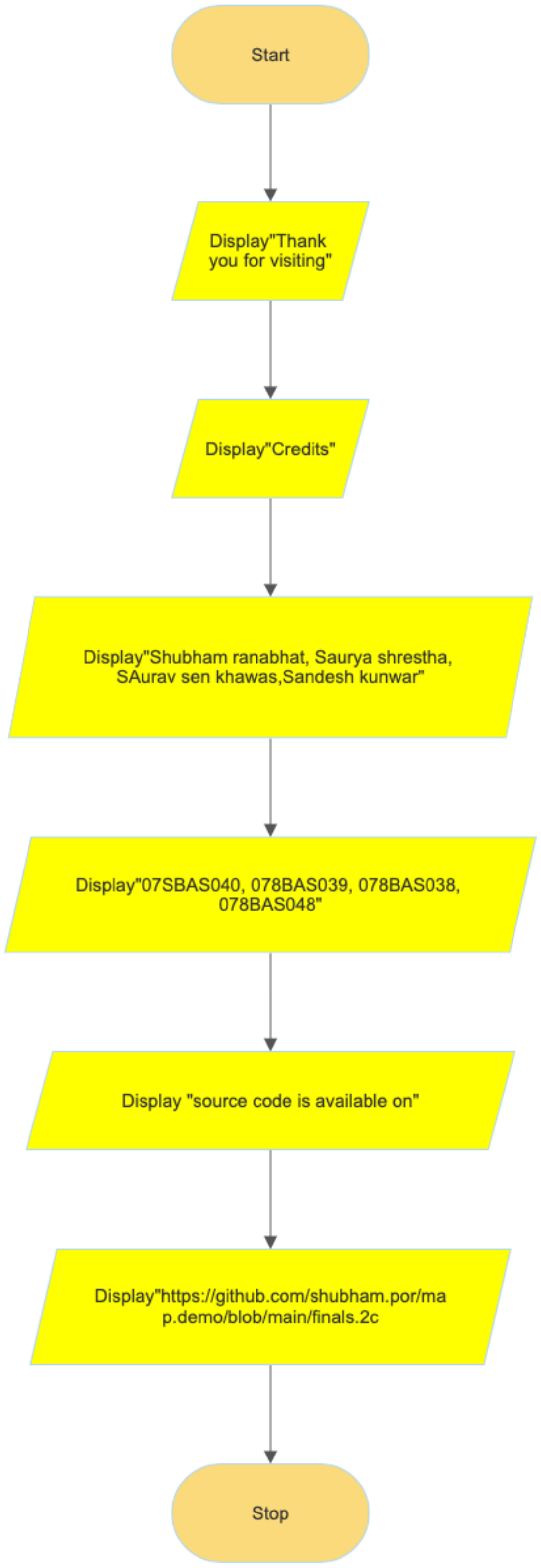
- For Map function


Start
Display"MAP OF IOE PULCHOWK"
Draw various lines resembling ouline of map.
Call love garden function
Call whitehouse function
.
.
.
Call hostel function
Draw remaining parts of map
Stop

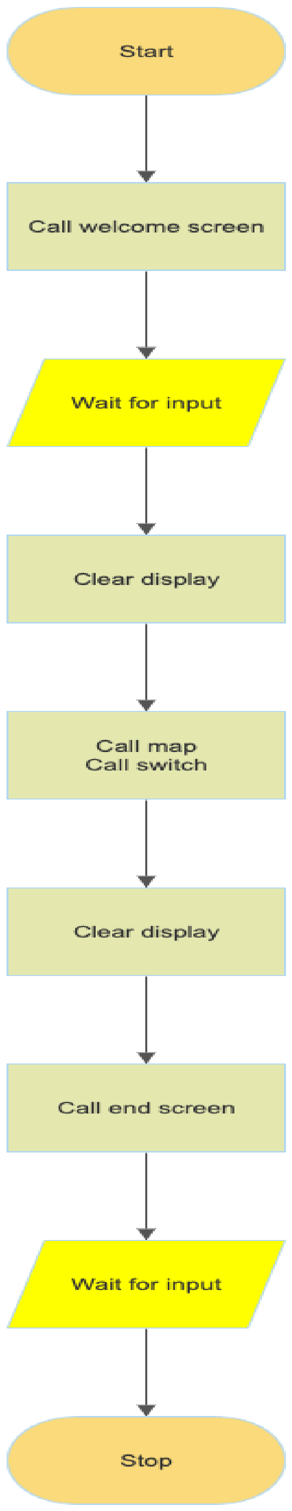
- For swich function



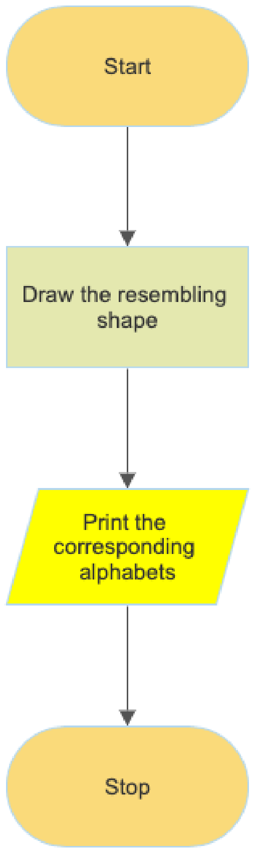
- For EndScreen function


Start
Display"Thank you for visiting"
Display"Credits"
Display"Shubham ranabhat, Saurya shrestha, SAurav sen khawas,Sandesh kunwar"
Display"07SBAS040, 078BAS039, 078BAS038, 078BAS048"
Display "source code is available on"
Display"https://github.com/shubham.por/map.demo/blob/main/finals.2c
Stop

- For Main function



Function such as Lovegarden, Whitehouse, ….. ….. ……, Girls, Staff all are based on similar flowchart which are drawn below:



Source code

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

void WlcScreen()
{
        int progress=0, width;
        char text[10];

   while (progress <= 100)
   {
                // Set the background color
                setbkcolor(DARKGRAY);
                cleardevice();
                outtextxy(270,185, "LOADING...");
                //border of the loading bar
                rectangle(200, 200, 400, 225);
                //Calculate the width of the loading bar based on progress
                width = (progress * 2);
                //loading bar
                setfillstyle(SOLID_FILL, RED);
                bar(201, 201, 200 + width, 224);
                //text to show the percentage of the loading bar
                sprintf(text, "%d%%", progress);
                outtextxy(290, 210, text);
                rectangle(200,200,400,225);
                // Delay for a short period of time to show the progress
                delay(50);
                progress++;
   }
   cleardevice();
        settextstyle(4,0,4);
        outtextxy(125,160,"A Guide to IOE Pulchowk");
        settextstyle(3,0,3);
        outtextxy(170,225,"Press any key to continue");
        settextstyle(0,0,1);
        outtextxy(10,465 ,"A mini project developed by team AXC");
}
void LoveGarden()
{
        setlinestyle(1,0,0);
        line(160,290,180,280);
        line(160,290,180,300);
        arc(180,285,270,300,5);
        arc(180,285,335,0,5);
        arc(180,285,25,60,5);
        arc(180,285,80,90,5);
        arc(180,295,270,300,5);
        arc(180,295,335,0,5);
        arc(180,295,25,60,5);
        outtextxy(172,286,"A");
}
void Whitehouse()
{
        setlinestyle(0,0,0);
        line(200,295,240,295);
        line(200,295,200,280);
        line(200,280,225,280);
        line(225,280,225,288);
        line(225,288,240,288);
        line(240,288,240,295);
        outtextxy(205,285,"B");
}
void Architecture()
{
        setlinestyle(0,0,0);
        line(190,325,230,325);
        line(190,325,190,320);
        line(190,320,180,320);
        line(180,320,180,315);
        line(180,315,200,315);
        line(200,315,200,300);
        line(200,300,215,300);
        line(215,300,215,315);
        line(215,315,230,315);
        line(230,315,230,325);
        outtextxy(205,310,"C");
}
void MasterHostel()
{
        setlinestyle(0,0,0);
        rectangle(265,300,253,313);
        outtextxy(256,304,"D");
}
void Hydrolab()
{
        setlinestyle(0,0,0);
        rectangle(265,315,253,327);
        outtextxy(256,318,"E");
}
void Cafe()
{
        setlinestyle(0,0,0);
        line(267,275,257,275);
        line(267,275,267,265);
        line(262,265,257,265);
        line(262,265,262,260);
        line(262,260,267,260);
        line(267,260,267,265);
        line(257,265,257,275);
        outtextxy(259,267,"F");
}
void RoboticsClub()
{
        setlinestyle(0,0,0);
        line(205,270,217,270);
        line(217,270,217,253);
        line(217,253,205,253);
        line(205,253,205,270);
        outtextxy(208,258,"G");
}
void Electronic()
{
        setlinestyle(0,0,0);
        line(182,270,182,263);
        line(182,263,185,263);
        line(185,263,185,258);
        line(185,258,182,258);
        line(182,258,182,235);
        line(182,235,165,235);
        line(165,235,165,233);
        line(165,233,163,233);
        line(163,233,161,233);
        line(161,233,161,245);
        line(161,245,163,245);
        line(163,245,163,245);
        line(163,245,165,245);
        line(165,245,165,239);
        line(165,239,172,239);
        line(172,239,172,270);
        line(172,270,182,270);
        outtextxy(174,250,"H");
}
void Cit()
{
        setlinestyle(0,0,0);
        rectangle(205,235,217,245);
        outtextxy(208,237,"I");
}
void Library()
{
        setlinestyle(0,0,0);
        outtextxy(243,235,"J");
        line(240,243,252,243);
        line(240,243,240,245);
        line(240,245,227,245);
        line(227,245,227,237);
        line(227,237,240,237);
        line(240,237,240,230);
        line(240,230,252,230);
        line(252,230,252,243);
}
void Ictc()
```

```
{
        setlinestyle(0,0,0);
        outtextxy(208,212,"K");
        line(207,208,207,211);
        line(207,211,204,211);
        line(204,211,204,220);
        line(204,220,210,220);
        line(210,220,216,220);
        line(216,220,218,220);
        line(218,220,218,213);
}
void Workshop()
{
        setlinestyle(0,0,0);
        line(252,270,252,253);
        line(252,270,240,270);
        line(252,253,240,253);
        line(240,253,240,270);
        outtextxy(244,260,"L");
}
void Mechanical()
{
        setlinestyle(0,0,0);
        line(240,190,242,192);
        line(242,192,240,194);
        line(240,194,242,196);
        line(240,190,242,188);
        line(242,188,244,190);
        line(244,190,254,180);
        line(254,180,249,175);
        line(249,175,256,168);
        line(256,168,262,174);
        line(262,174,264,172);
        line(264,172,272,180);
        line(272,180,250,202);
        line(250,202,242,196);
        outtextxy(257,177,"M");
}
void Civil()
```

```
{
        setlinestyle(0,0,0);
        line(205,185,205,178);
        line(205,178,213,178);
        line(213,178,213,180);
        line(205,185,213,185);
        line(213,185,213,190);
        line(213,180,228,180);
        line(228,180,228,190);
        line(228,190,213,190);
        outtextxy(217,182,"N");
}
void Chemical()
{
        setlinestyle(0,0,0);
        line(250,155,258,155);
        line(258,155,258,160);
        line(258,160,273,160);
        line(250,155,250,148);
        line(250,148,260,148);
        line(260,148,260,150);
        line(260,150,273,150);
        line(273,150,273,160);
        outtextxy(261,152,"O");
}
void Incubation()
{
        setlinestyle(0,0,0);
        line(243,130,255,133);
        line(255,133,257,120);
        line(257,120,246,120);
        line(245,120,243,130);
        outtextxy(248,123,"P");
}
void Zero()
{
        setlinestyle(0,0,0);
        arc(274,120,240,10,4);
        arc(271,127,50,270,3);
        line(271,130,283,133);
        line(283,133,286,136);
        line(286,136,288,134);
        line(290,120,288,130);
        line(278,118,290,120);
        line(288,130,288,134);
        outtextxy(278,123,"Q");
}
void Heavy()
{
        setlinestyle(0,0,0);
        line(295,195,305,201);
        line(305,201,290,216);
        line(290,216,282,210);
        line(282,210,295,195);
        outtextxy(293,200,"R");
}
void Aero()
{
        setlinestyle(0,0,0);
        line(293,180,295,185);
        line(295,185,300,183);
        line(300,183,303,190);
        line(303,190,310,186);
        line(310,186,307,179);
        line(307,179,316,175);
        line(316,175,308,167);
        line(308,167,302,170);
        line(302,170,303,172);
        outtextxy(305,170,"S");
        line(303,172,292,179);
}
void Ground()
{
        setlinestyle(0,0,0);
        line(330,80,410,110);
        line(330,80,310,130);
        line(310,130,340,140);
        line(340,140,342,132);
        line(342,132,400,150);
        line(400,150,410,110);
        //setfillstyle(1,2);
    //  floodfill(334,114,WHITE);
        settextstyle(0,0,2);
        outtextxy(360,110,"T");
        settextstyle(0,0,0);
}
void Carpentary()
{
        line(265,295,250,295);
        line(265,295,265,280);
        line(250,295,250,280);
        line(250,280,265,280);
        outtextxy(255,285,"U");
}
void Basket()
{
        setlinestyle(0,0,0);
        rectangle(395,175,415,185);
        outtextxy(402,177,"V");
}
void Boys()
{
        setlinestyle(0,0,0);
        line(455,161,465,164);
        line(465,164,467,154);
        line(455,160,457,154);
        line(467,154,459,152);
        line(459,152,461,142);
        line(457,154,447,152);
        line(461,142,453,140);
        line(447,152,449,142);
        line(453,140,455,130);
        line(449,142,439,140);
        line(455,130,443,128);
        line(439,140,443,128);
        line(446,132,451,133);
        line(446,132,444,137);
```

```
	line(444,137,450,138);
	line(451,133,450,138);
	outtextxy(451,144,"W");
}
void TtHall()
{
	rectangle(260,212,275,222);
	outtextxy(264,214,"X");
}
void Girls()
{
	setlinestyle(0,0,0);
	setlinestyle(0,0,0);
	rectangle(410,90,425,100);
	outtextxy(414,92,"Y");
}
void Staff()
{
	setlinestyle(0,0,0);
	rectangle(440,70,455,75);
	rectangle(460,80,475,85);
	rectangle(480,90,495,95);
	rectangle(485,80,500,85);
	rectangle(435,85,450,90);
	line(430,70,430,100);
	line(430,100,500,100);
	line(500,100,510,70);
	outtextxy(460,90,"Z");
}

void Map()
{
	settextstyle(4,0,4);
	outtextxy(120,0,"MAP OF IOE PULCHOWK'");

	settextstyle(0,0,0);
//Outline of Map
	line(0,40,1000,40);
	line(0,350,1000,350);
```

```
            line(0,376,1000,376);
            rectangle(0,0,639,476);
            setlinestyle(0,0,3);
// ICTC section
             setcolor(BROWN);
             line(150,330,270,330);
             line(270,330,270,300);
             line(270,300,270,300);
             line(150,330,150,300);
             line(150,280,150,220);
             line(270,300,270,237);
             line(150,220,160,210);
             line(160,210,190,220);
             line(190,220,205,205);
             line(205,205,230,217);
             line(270,237,243,224);
// Aerospace section
             line(210,200,240,215);
             line(249,219,270,230);
             line(210,200,190,185);
             line(190,185,205,170);
             line(205,170,210,160);
             line(210,160,200,150);
             line(200,150,230,135);
             line(230,135,240,105);
             line(240,105,290,100);
             line(290,100,300,60);
             line(300,60,330,50);
             line(330,50,350,60);
             line(350,60,365,75);
             line(365,75,385,71);
             line(385,71,430,71);
             line(430,71,435,65);
             line(435,65,485,75);
             line(485,75,510,70);
             line(270,230,310,220);
             line(310,220,307,205);
             line(307,205,335,190);
             line(335,190,410,200);
             line(410,200,413,190);
             line(413,190,460,195);
             line(460,195,510,175);
             line(510,175,520,120);
             line(520,120,510,70);
             setcolor(WHITE);
// Outline complete
//buildings
            LoveGarden();
            Whitehouse();
            Architecture();
            MasterHostel();
            Hydrolab();
            Cafe();
            RoboticsClub();
            Electronic();
            Cit();
            Library();
            Ictc();
            Workshop();
            Mechanical();
            Civil();
            Chemical();
            Incubation();
            Zero();
            Heavy();
            Aero();
            Ground();
            Carpentary();
            Basket();
            Boys();
            TtHall();
            Girls();
            Staff();
// see saw
            ellipse(223,261,270,315,6,6);
            ellipse(223,261,345,0,6,6);
            ellipse(223,261,35,70,6,6);
            ellipse(223,261,90,135,6,6);
```

```
        ellipse(223,261,150,175,6,6);
        ellipse(223,261,200,225,6,6);
        ellipse(223,261,245,260,6,6);
//Road
        setcolor(LIGHTGRAY);
        line(150,287,180,277);
        line(150,282,180,272);
        line(180,272,190,272);
        line(180,277,190,277);
        line(190,272,190,235);
        line(150,293,180,303);
        line(150,298,180,308);
        line(150,293,150,298);
        line(180,308,198,308);
        line(180,303,190,303);
        line(190,277,190,303);
        line(198,277,198,308);
        line(198,272,232,272);
        line(198,277,240,277);
        line(150,287,150,282);
        line(198,252,198,272);
        line(198,251,232,251);
        line(237,251,250,251);
        line(250,251,250,248);
        line(232,251,232,272);
        line(237,251,237,272);
        line(237,272,255,272);
        line(255,272,255,277);
        line(242,277,198,277);
        line(242,277,242,295);
        line(242,295,247,295);
        line(247,295,247,277);
        line(247,277,255,277);
        line(198,248,198,235);
        line(198,248,250,248);
        line(190,235,185,230);
        line(185,230,160,230);
        line(160,226,185,226);
        line(185,226,190,230);
        line(190,230,230,220);
        line(198,235,240,225);
        line(230,220,240,225);
        line(160,226,160,230);
        setcolor(WHITE);
        setlinestyle(3,0,1);
        line(150,285,180,275);
        line(182,275,260,275);
        line(194,272,194,235);
        line(194,232,235,223);
        line(194,277,194,300);
        line(150,295,180,306);
        line(180,306,196,306);
        line(244,275,244,290);
        line(160,228,185,228);
        setlinestyle(2,0,0);
        line(198,250,250,250);
        line(235,253,235,270);
        setlinestyle(0,0,0);
//Soil Testing lab
        rectangle(160,217,167,223);
//Other side of ICTC
//Water pond
        setfillstyle(1,BLUE);
        circle(310,110,3);
        floodfill(310,110,WHITE);
//road
        setcolor(WHITE);
        setlinestyle(2,0,2);
        line(247,216,309,154);
        setcolor(LIGHTGRAY);
        setlinestyle(0,0,0);
        line(245,214,250,217);
        line(250,217,312,155);
        line(245,214,306,153);
        line(306,153,241,140);
        line(241,140,244,135);
        setlinestyle(2,0,0);
        setcolor(WHITE);
```

```
line(245,138,310,150);
setcolor(LIGHTGRAY);
setlinestyle(0,0,0);
line(244,135,314,148);
line(314,148,324,152);
line(312,155,324,160);
setcolor(WHITE);
ellipse(340,156,180,220,16,2);
ellipse(340,156,240,260,16,2);
ellipse(340,156,280,300,16,2);
setcolor(LIGHTGRAY);
ellipse(340,152,180,0,16,2);
ellipse(340,160,180,0,16,2);
setcolor(WHITE);
ellipse(356,166,0,20,48,10);
ellipse(356,166,30,50,48,10);
ellipse(356,166,60,65,48,10);
ellipse(356,166,70,80,48,10);
ellipse(356,166,85,90,48,10);
setcolor(LIGHTGRAY);
ellipse(356,170,0,90,48,10);
ellipse(356,162,0,90,48,10);
ellipse(404,152,270,0,16,10);
ellipse(404,160,270,315,16,10);
setcolor(LIGHTGRAY);
line(415,167,440,174);
line(440,174,443,170);
line(443,170,425,164);
setcolor(WHITE);
setlinestyle(2,0,1);
line(422,160,433,112);
setcolor(LIGHTGRAY);
setlinestyle(0,0,0);
setcolor(LIGHTGRAY);
line(425,164,435,115);
line(420,152,430,110);
arc(435,110,90,180,5);
arc(440,115,90,180,5);
line(435,105,510,105);
```

```
        line(440,109,510,109);
        line(510,105,510,109);
        setcolor(WHITE);
        setlinestyle(3,0,1);
        line(435,107,505,107);
        setlinestyle(0,0,0);
// the building i dont know about
        line(445,124,453,125);
        line(445,124,447,118);
        line(453,124,455,119);
        line(447,118,455,119);
//campus mess
        line(460,183,463,173);
        line(460,183,453,181);
        line(463,173,456,171);
        line(456,171,453,181);
//Gym
        line(445,167,453,169);
        line(453,169,454,166);
        line(445,167,446,164);
        line(454,166,446,164);
        settextstyle(0,0,1);
//Path of mechanical
        setlinestyle(1,0,3);
        setcolor(YELLOW);
        line(247,207,230,190);
        line(233,190,233,140);
        line(233,173,247,173);
        line(247,173,247,163);
        line(247,163,233,163);
        line(247,163,290,163);
        line(233,177,215,177);
        setlinestyle(0,0,0);
        setcolor(WHITE);
//green area around chemical
        setcolor(10);
        line(230,170,220,173);
        line(220,173,210,170);
        line(210,170,213,163);
```

```
            line(213,163,213,158);
            line(213,158,215,150);
            line(215,150,225,147);
            line(225,147,230,170);
            setfillstyle(1,10);
            floodfill(219,154,10);
//green area architecture
            line(170,325,170,310);
            line(170,310,155,307);
            line(155,307,155,325);
            line(155,325,170,325);
            floodfill(159,320,10);
//green area electronics engineering
            setcolor(10);
            rectangle(160,250,167,270);
            floodfill(164,254,10);
//green area around basketball court
            line(380,170,385,190);
            line(380,170,365,167);
            line(365,167,353,170);
            line(353,170,343,167);
            line(343,167,335,171);
            line(335,171,327,187);
            line(327,187,335,185);
            line(335,185,385,190);
            floodfill(380,185,10);
//green area behind mess
            line(470,185,475,150);
            line(475,150,470,120);
            line(470,120,510,113);
            line(510,113,512,140);
            line(512,140,506,170);
            line(506,170,470,185);
            floodfill(500,160,10);
//green area around ground
            line(300,95,310,100);
            line(300,95,300,70);
            line(300,70,325,57);
            line(325,57,340,63);
```

```
        line(340,63,342,74);
        line(342,74,330,74);
        line(330,74,310,100);
        floodfill(310,70,10);
//green area around incubtion
        line(243,110,270,108);
        line(270,108,272,115);
        line(243,110,240,116);
        line(240,116,272,115);
        floodfill(250,115,10);
        setcolor(WHITE);
//tt
        line(430,162,440,165);
        line(440,165,443,156);
        line(430,162,433,153);
        line(433,153,443,156);
        outtextxy(470,465,"Press ESC key to quit");
}
void Fix()
{
        cleardevice();
        Map();
        settextstyle(7,0,2);
        outtextxy(10,348,"Use your keyboard to select your destination on the map.");
        settextstyle(0,0,0);
        setcolor(LIGHTCYAN);

}
void Swich()
{
        char c;

        while (1)
         {
        if (kbhit)
                {
                        if ((int)c == 27)
                           break;
                        settextstyle(7,0,2);
```

```
		outtextxy(10,348,"Use your keyboard to select your destination on the
map.");
		c = getch();
		switch (c)
		{
	case 'a':
		Fix();

		LoveGarden();
		settextstyle(7,0,2);
		outtextxy(10,375,"Love Garden");
		setcolor(WHITE);

		break;
	case 'b':
		Fix();

		Whitehouse();
		settextstyle(7,0,2);
		outtextxy(10,375,"Office of the Dean");
		setcolor(WHITE);
		break;
	case 'c':
		Fix();
		Architecture();
		settextstyle(7,0,2);
		outtextxy(10,375,"Department of Architecture");
		setcolor(WHITE);
		break;
	case 'd':
		Fix();
		MasterHostel();
		settextstyle(7,0,2);
		outtextxy(10,375,"M. Sc Hostel");
		setcolor(WHITE);
		break;
	case 'e':
		Fix();
		Hydrolab();
```

```
			settextstyle(7,0,2);
			outtextxy(10,375,"Hydro Lab");
			setcolor(WHITE);
			break;
		case 'f':
			Fix();
			Cafe();
			settextstyle(7,0,2);
			outtextxy(10,375,"Campus Cafeteria");
			setcolor(WHITE);
			break;
		case 'g':
			Fix();
			RoboticsClub();
			settextstyle(7,0,2);
			outtextxy(10,375,"Robotics Club");
			setcolor(WHITE);
			break;
		case 'h':
			Fix();
			Electronic();
			settextstyle(7,0,1);
			outtextxy(10,375,"Department of Electrical, Electronics & Computer
Engineering");
			setcolor(WHITE);
			break;
		case 'i':
			Fix();
			Cit();
			settextstyle(7,0,2);
			outtextxy(10,375,"CIT Building");
			setcolor(WHITE);
			break;
		case 'j':
			Fix();
			Library();
			settextstyle(7,0,2);
			outtextxy(10,375,"Library Block");
			setcolor(WHITE);
```

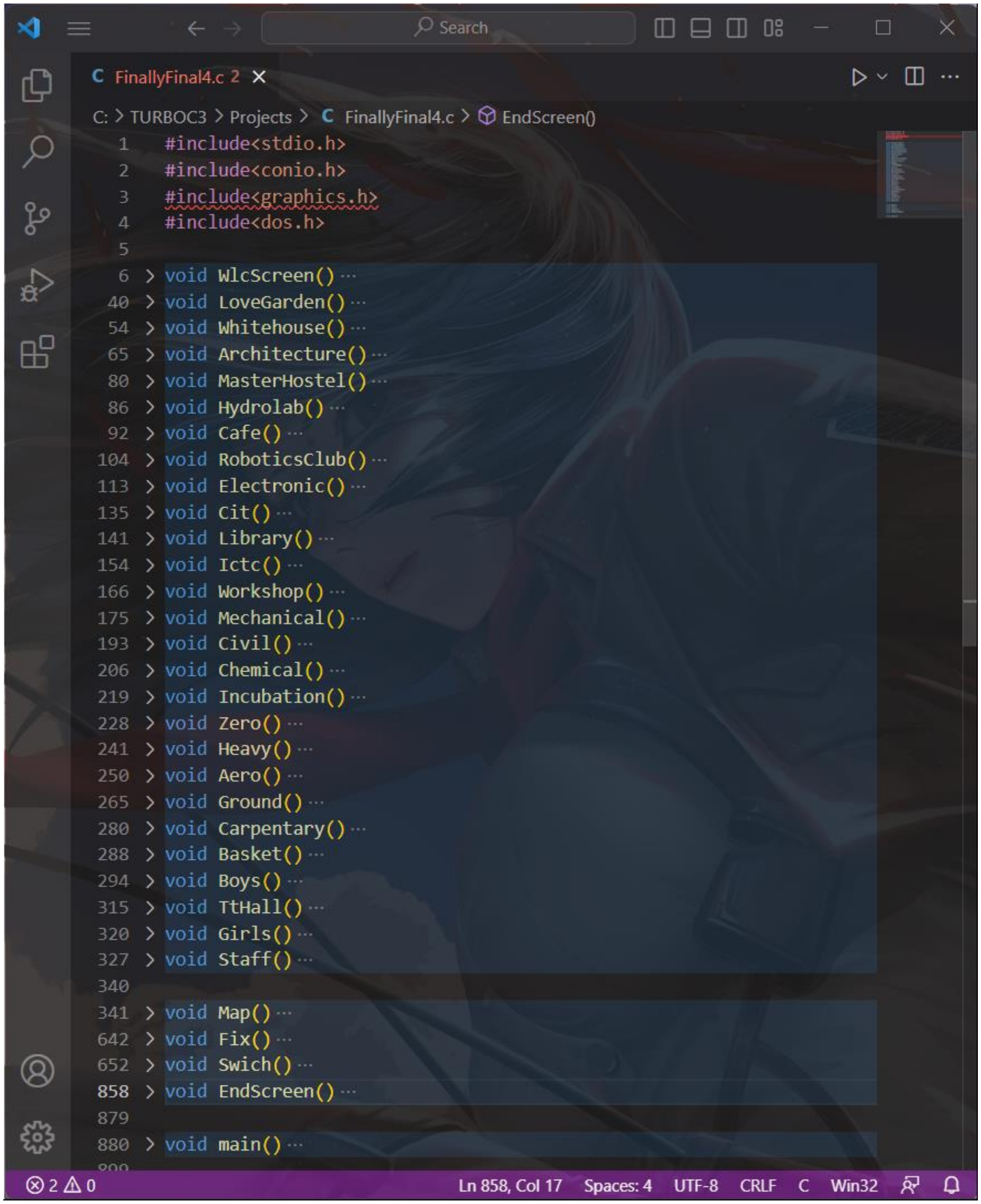
```
                break;
        case 'k':
                Fix();
                Ictc();
                settextstyle(7,0,2);
                outtextxy(10,375,"ICTC Building");
                setcolor(WHITE);
                break;
        case 'l':
                Fix();
                Workshop();
                settextstyle(7,0,2);
                outtextxy(10,375,"Workshop and Engine Lab");
                setcolor(WHITE);
                break;
        case 'm':
                Fix();
                Mechanical();
                settextstyle(7,0,2);
                outtextxy(10,375,"Department of Mechanical Engineering");
                setcolor(WHITE);
                break;
        case 'n':
                Fix();
                Civil();
                settextstyle(7,0,2);
                outtextxy(10,375,"Department of Civil Engineering");
                setcolor(WHITE);
                break;
        case 'o':
                Fix();
                Chemical();
                settextstyle(7,0,2);
                outtextxy(10,375,"Department    of    Applied    Sciences    &    Chemical
engineering");
                setcolor(WHITE);
                break;
        case 'p':
                Fix();
                Incubation();
                settextstyle(7,0,2);
                outtextxy(10,375,"Incubation, Innovation and Entrepreneurship Center");
                setcolor(WHITE);
                break;
        case 'q':
                Fix();
                Zero();
                settextstyle(7,0,2);
                outtextxy(10,375,"Centre for Energy Studies");
                setcolor(WHITE);
                break;
        case 'r':
                Fix();
                Heavy();
                settextstyle(7,0,2);
                outtextxy(10,375,"Heavy Lab");
                setcolor(WHITE);
                break;
        case 's':
                Fix();
                Aero();
                settextstyle(7,0,2);
                outtextxy(10,375,"Department of Aerospace engineering");
                setcolor(WHITE);
                break;
        case 't':
                Fix();
                Ground();
                settextstyle(7,0,2);
                outtextxy(10,375,"Football and Cricket Ground");
                setcolor(WHITE);
                break;
        case 'u':
                Fix();
                Carpentary();
                settextstyle(7,0,2);
                outtextxy(10,375,"Department of Carpentary");
                setcolor(WHITE);
```

```
			break;
		case 'v':
			Fix();
			Basket();
			settextstyle(7,0,2);
			outtextxy(10,375,"Basketball Court");
			setcolor(WHITE);
			break;
		case 'w':
			Fix();
			Boys();
			settextstyle(7,0,2);
			outtextxy(10,375,"Boy's Hostel");
			setcolor(WHITE);
			break;
		case 'x':
			Fix();
			TtHall();
			settextstyle(7,0,2);
			outtextxy(10,375,"Guard Post");
			setcolor(WHITE);
			break;
		case 'y':
			Fix();
			Girls();
			settextstyle(7,0,2);
			outtextxy(10,375,"Girl's Hostel");
			setcolor(WHITE);
			break;
		case 'z':
			Fix();
			Staff();
			settextstyle(7,0,2);
			outtextxy(10,375,"IOE Staff Quarter");
			setcolor(WHITE);
			break;
		default:
			break;
			}
```

```
                    }
           }
}
void EndScreen()
{
          settextstyle(4,0,5);
          outtextxy(120,150,"Thanks for your visit.");
          settextstyle(5,0,5);
          outtextxy(260,290,"Credits");
          settextstyle(1,0,3);
          outtextxy(60,350,"Shubham Ranabhat");
          outtextxy(350,350,"Saurya Shrestha");
          settextstyle(1,0,2);
          outtextxy(60,400,"Sandesh Kumar");
          outtextxy(350,400,"Saurav Sen Khawas");
          settextstyle(6,0,1);
          outtextxy(60,370,"078BAS040");
          outtextxy(350,370,"078BAS039");
          outtextxy(60,417,"078BAS048");
          outtextxy(350,417,"078BAS038");
          settextstyle(0,0,1);
          outtextxy(10,5,"source code is available on");
          outtextxy(10,15," https://github.com/shubham-per/Map-mini-project");
}

void main()
{
          int graphdriver=DETECT, graphmode;
          initgraph(&graphdriver,&graphmode,"C:\\TURBOC3\\BGI");
          cleardevice();

          WlcScreen();
          getch();
          cleardevice();

          Map();
          Swich();

          cleardevice();
```

```
        EndScreen();

        getch();
        closegraph();
}
```

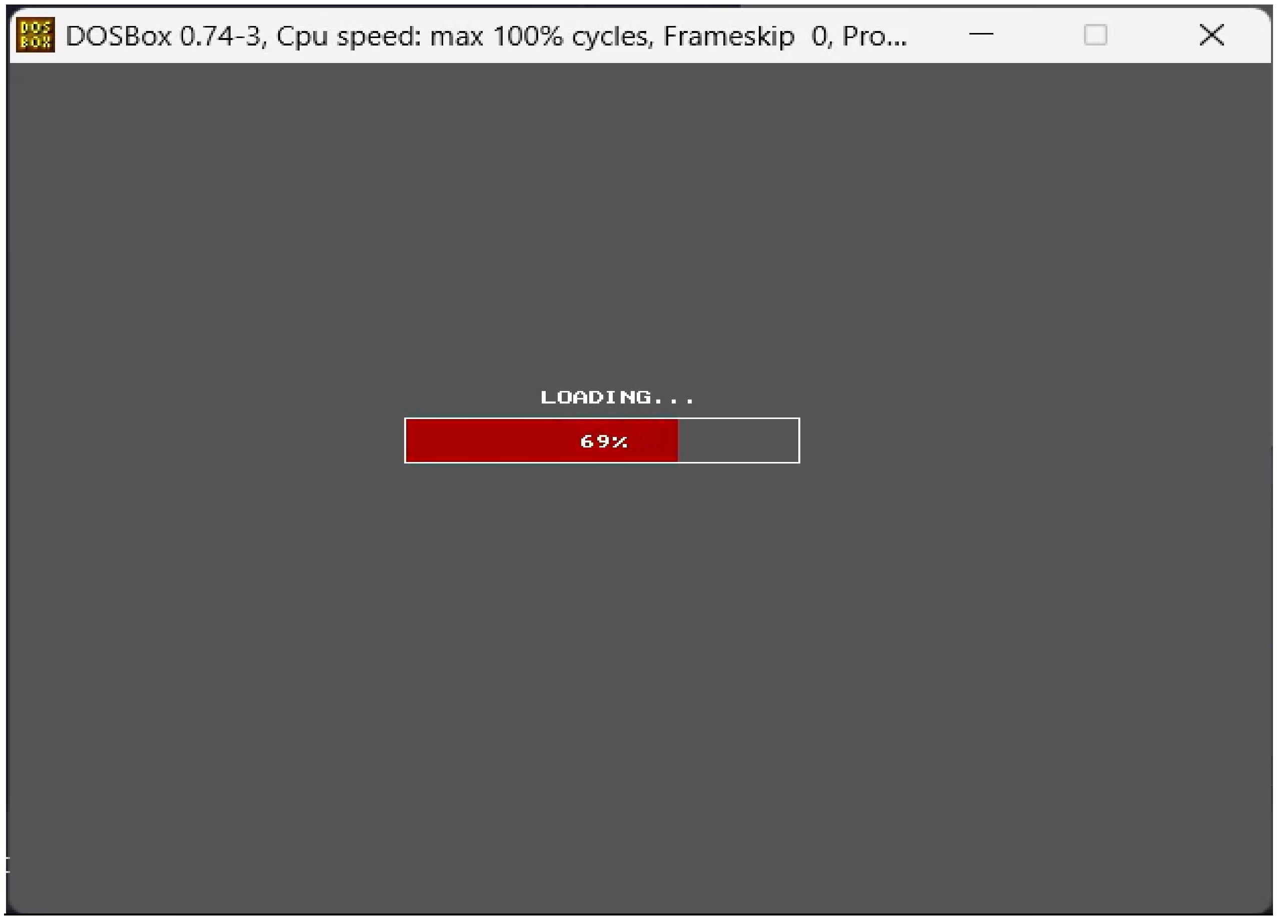
Simpler version of our source code

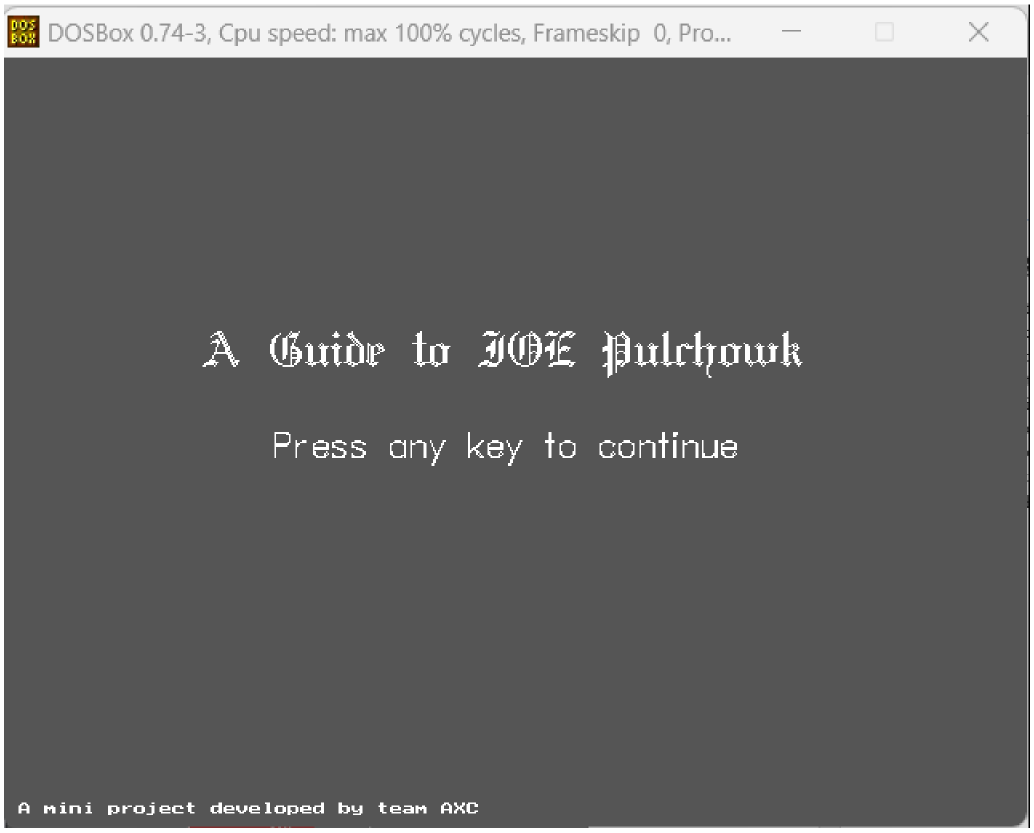


```c
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>

void WlcScreen() ···
void LoveGarden() ···
void Whitehouse() ···
void Architecture() ···
void MasterHostel() ···
void Hydrolab() ···
void Cafe() ···
void RoboticsClub() ···
void Electronic() ···
void Cit() ···
void Library() ···
void Ictc() ···
void Workshop() ···
void Mechanical() ···
void Civil() ···
void Chemical() ···
void Incubation() ···
void Zero() ···
void Heavy() ···
void Aero() ···
void Ground() ···
void Carpentary() ···
void Basket() ···
void Boys() ···
void TtHall() ···
void Girls() ···
void Staff() ···

void Map() ···
void Fix() ···
void Swich() ···
void EndScreen() ···

void main() ···
```

Implementation

To implement our project, we used several programming techniques, including the use of arrays to make loading screen at the start of our program. We also used the graphic.h library to create the user interface, and the getch() function to handle keyboard input. We organized our code into several modules, each responsible for a specific aspect of the project, such as switching data, user interface, and input handling. We used header files to manage dependencies between modules, and to provide a clean and readable interface for the main function. During the development process, we faced several challenges, such as dealing with graphics bugs. However, we were able to overcome these challenges by carefully testing and debugging our code. The overall structure of our code is based on a main event loop, which listens for user input and updates the screen accordingly. We also included initialization routines to set up the user interface and the data structures. Overall, the implementation of our project aims to be efficient, readable, and maintainable, so that it can be easily extended or modified in the future.

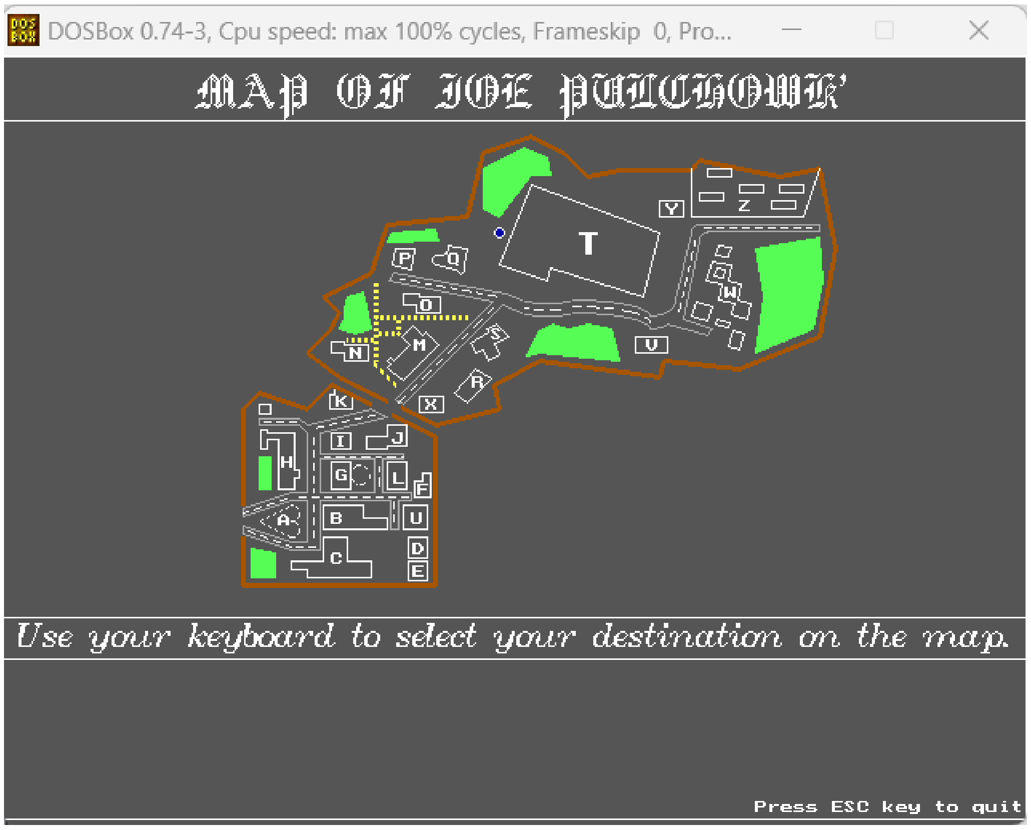

DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Pro...
LOADING...
69%


DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Pro...

A Guide to IOE Pulchowk

Press any key to continue

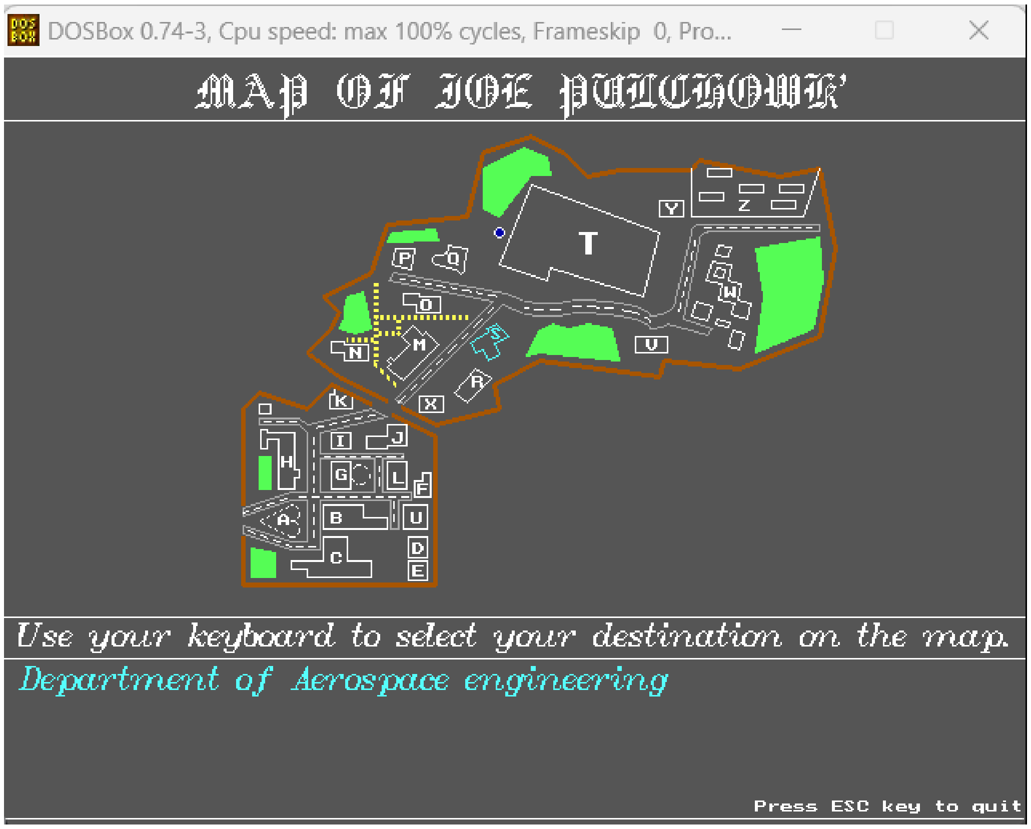
A mini project developed by team AXC


DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Pro...

MAP OF IOE PULCHOWK

Use your keyboard to select your destination on the map.

Press ESC key to quit

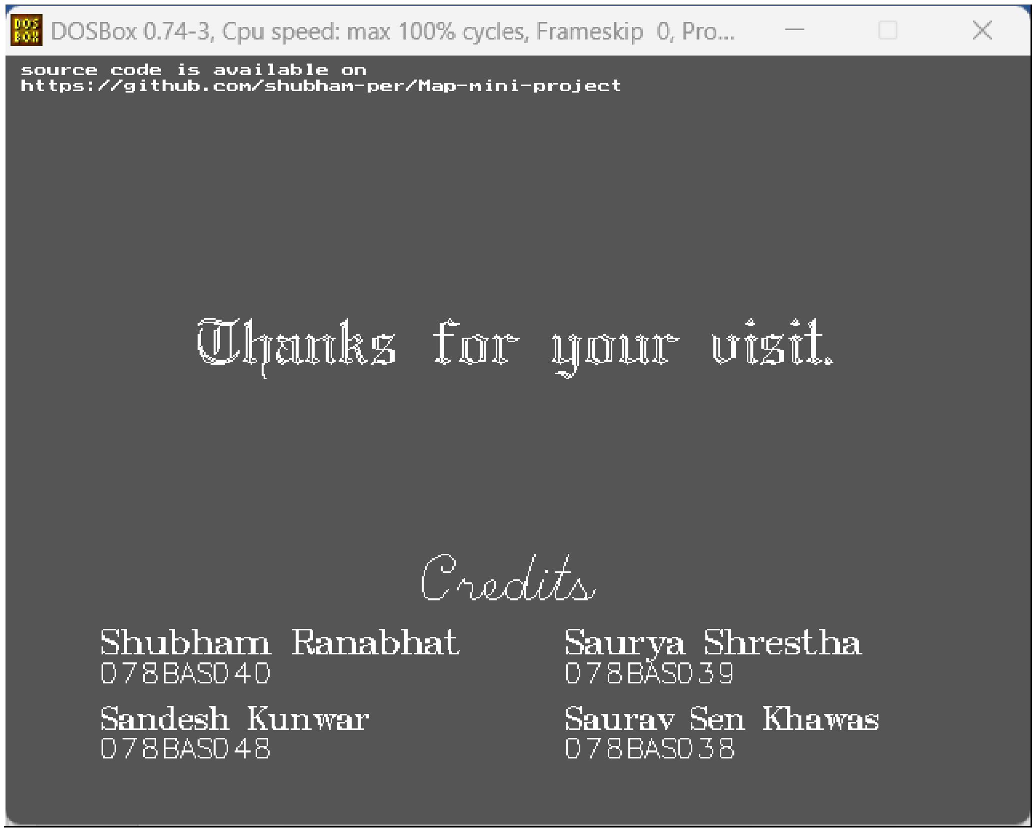

DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Pro...

MAP OF IOE PULCHOWK

Use your keyboard to select your destination on the map.

Department of Aerospace engineering

Press ESC key to quit


DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Pro...

```
source code is available on
https://github.com/shubham-per/Map-mini-project
```

Thanks for your visit.

Credits

Shubham Ranabhat
078BAS040

Saurya Shrestha
078BAS039

Sandesh Kunwar
078BAS048

Saurav Sen Khawas
078BAS038

Testing

To ensure that our project is functional and meets the requirements, we used manual testing as with given time constrain as automated testing was bit overwhelming. We created a set of test cases that cover all the main features of the project, including the display of the map, the handling of user input, and the updating of the screen. Each test case includes a set of input data and the expected output, which we verified manually. During the testing process, we found several bugs and errors, such as incorrect position of buildings and facilities, and incorrect handling of user input. However, we were able to fix these bugs by carefully reviewing our code and testing it again. The final results of our testing show that our project is functional and meets the requirements, and that it is robust enough to handle various input scenarios.

Result and Conclusion

In conclusion, our project has achieved its goals of creating a map display of the Pulchowk campus using the graphics.h library and handling user input. We were able to create a user-friendly interface that allows users to navigate the map and to view information about the buildings and facilities. However, our project also has some limitations, such as the performance issues that we encountered when rendering large maps. In addition, we were unable to implement some advanced features, such as search and navigation functions, due to time and resource constraints. Despite these limitations, our project has important implications for the college community, as it provides a useful tool for students, faculty, and staff to navigate the campus and to find important locations. It also has potential implications for the field of computer science, as it demonstrates the use of graphics libraries to create interactive interfaces. In the future, if possible we plan to further develop and improve our project by implementing advanced features and optimizing performance. We also plan to explore other technologies and programming languages that could enhance the functionality and usability of our project.