

DAYANANDA SAGAR UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SCHOOL OF ENGINEERING

DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE - 560068



MINI-PROJECT REPORT ON

“MAN IN RAIN”

2023

**BACHELOR OF TECHNOLOGY
IN**

**COMPUTER SCIENCE & ENGINEERING
(DATA SCIENCE)**

***COURSE:* DESIGN & ANALYSIS OF
ALGORITHMS**

Submitted by:

ABHISHEK A - ENG21DS0002

ABHISHEK N - ENG21DS0003

Submitted to:

**PROF. SHIVAMMA D
DATA SCIENCE
DSU**

DAYANANDA SAGAR UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SCHOOL OF ENGINEERING

DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE - 560068



CERTIFICATE

It is certified that the mini project work entitled “ *MAN IN RAIN* ” has been carried out at *Dayananda Sagar University, Bangalore*, by *ABHISHEK A(ENG21DS0002)*, *ABHISHEK N(ENG21DS0003)*, bonafide students of *fourth Semester, B.tech.* in partial fulfilment for the award of degree in *Bachelor of Technology* in *Computer Science & Engineering, Data Science* during academic year *2023*. It is attested that the report has been updated with all modifications and suggestions designated for internal assessment.

Signature of the TEACHER

Signature of the HOD

ACKNOWLEDGEMENT

Although a project's completion offers a sense of satisfaction, it is never done without acknowledging everyone who contributed to its accomplishment.. We wish to express our profound feelings of gratitude to this great institution of ours DAYANANDA SAGAR UNIVERSITY for providing the excellent facilities.

I am especially thankful to our Chairperson, **Dr. Shaila S G**, for providing necessary departmental facilities, moral support and encouragement. The largest measure of our acknowledgment is reserved for **PROF. SHIVAMMA D**, whose guidance and support made it possible to complete the project work in a timely manner.

I have received a great deal of guidance and co-operation from the staff and I wish to thank all that have directly or indirectly helped me in the successful completion of this MINI PROJECT WORK.

ABHISHEK A - (ENG21DS0002)
ABHISHEK N - (ENG21DS0003)

DECLARATION

I hereby declare that the MINI PROJECT entitled “ MAN IN RAIN ” submitted to Dayananda Sagar University, Bengaluru, is a bonafide record of the work carried out by us under the guidance of PROF. SHIVAMMA D, (DATA SCIENCE) School of Engineering, Dayananda Sagar University, and this work is submitted for the completion of the MINI PROJECT in 4th semester under the *Design And Analysis of Algorithms* course, in Bachelor of Technology in Computer Science and Engineering (data science).

ABHISHEK A - (ENG21DS0002)
ABHISHEK N - (ENG21DS0003)

TABLE OF CONTENTS

CHAPTERS	PAGE NO
1. INTRODUCTION	6
1.1 ABOUT THE PROBLEM	
1.2 ABOUT THE DAA TECHNIQUE	
2. PROBLEM STATEMENT	8
3. LITERATURE REVIEW	9
4. S/W & H/W REQUIREMENTS	10
5. DESIGN	11
5.1 PSEUDOCODE OF ALGORITHM	
5.2 FLOW CHART	
6. IMPLEMENTATION(CODING)	12
7. TESTING	15
8. OUTPUT SCREEN SHOTS	15
9. CONCLUSION	16
10. REFERENCES	17

1. INTRODUCTION:

Rainy days often evoke a sense of calmness and nostalgia. The sound of raindrops hitting the ground and the sight of people seeking shelter under umbrellas create a picturesque scene. In the realm of computer graphics and programming, capturing such moments becomes a fascinating challenge. This mini project report delves into the creation of an animated graphics program titled "Man in Rain," implemented using the C programming language.

1.1 ABOUT THE PROBLEM

The animated graphics program utilizes the power of C programming language and graphics libraries to create an interactive and visually appealing experience for the user. By implementing various graphical elements and employing animation techniques, the program generates a seamless and immersive depiction of a man navigating through a virtual rain shower.

Throughout this report, we will explore the development process of the "Man in Rain" program, discussing the key components, algorithms, and techniques utilized. We will examine the challenges faced during the implementation, the strategies employed to overcome them, and the final outcome of the project. Furthermore, we will discuss the potential applications of this program and reflect upon the knowledge gained through its creation.

1.2.ABOUT THE DAA TECHNIQUE

Combining the creative aspects of animation with the technical intricacies of programming, the "Man in Rain" project serves as a testament to the versatility and power of the C programming language. It not only provides a platform for artistic expression but also enhances the understanding of fundamental programming concepts within the domain of computer graphics.

In the following sections, we will delve into the various components of this project, detailing the algorithms used to render the rain, animate the figure, and create an engaging user experience. We will also provide insights into the challenges encountered and the approaches employed to tackle them, thereby shedding light on the thought process and problem-solving abilities required in the realm of graphics programming.

Let us now embark on the journey of the "Man in Rain" project, exploring the beauty of a rainy day through the lens of computer graphics and program

2. PROBLEM STATEMENT

The objective of this project is to simulate a realistic rain scenario by employing fundamental graphics concepts and algorithms. The program features an animated figure of a man walking in the rain while holding an umbrella to shield himself from the downpour. Through careful design and coding, this project aims to bring the essence of a rainy day to life on the computer screen.

The program should incorporate the following key features:

1.Rain Simulation: The program should generate a visually convincing rain effect, with raindrops falling from the top of the screen. The raindrops should exhibit realistic motion, including the appearance of random falling patterns and collision with objects.

2.Animated Figure: The program should feature an animated figure of a man walking across the screen. The figure should exhibit fluid motion and realistic walking behavior. The man should be equipped with an umbrella, which should be properly positioned and move in synchronization with the figure's motion.

3.Graphics Rendering: The program should employ appropriate graphics rendering techniques to ensure smooth animation and seamless integration of all visual elements. The graphics should be rendered efficiently, minimizing flickering or other artifacts that may disrupt the user experience.

4.User Interaction: The program should provide basic user interaction capabilities, allowing the user to control certain aspects of the animation. For example, the user should be able to start and stop the rain,

5.Error Handling: The program should implement appropriate error handling mechanisms to handle exceptional scenarios, such as memory allocation failures or unexpected user input. It should gracefully handle errors, providing meaningful error messages to the user and avoiding crashes or instability.

By addressing these requirements, the "Man in Rain" animated graphics program will provide an engaging and immersive experience, showcasing the capabilities of the C programming language and graphics libraries.

3. LITERATURE REVIEW

Paper Name	Author Name	Limitations
A man walking in the rain	K. Salma	Not included rainbow

4. SOFTWARE & HARDWARE REQUIREMENTS

Software Requirements:

- Operating System: Any modern OS compatible with C programming.
- C Compiler: GCC, Visual C++, or Clang.
- Graphics Libraries: OpenGL, SDL, or Allegro

.

Hardware requirements:

- Processor: Modern processor compatible with the chosen OS and C compiler.
- Memory (RAM): Minimum 4 GB RAM.
- Graphics Card: Capable of 2D rendering and hardware acceleration.
- Display: Standard display with a minimum resolution of 1024x768 pixels.
- Please note that these are general requirements, and the specific software and hardware needs may vary depending on the chosen graphics libraries and the complexity of your program. Adjust the requirements accordingly based on your specific implementation.

5. DESIGN

5.1. PSEUDOCODE OF ALGORITHM

- i. Import necessary libraries (conio.h, graphics.h, stdio.h).
- ii. Define constants: ScreenWidth as getmaxx(), ScreenHeight as getmaxy(), and GroundY as ScreenHeight * 0.75.
- iii. Initialize ldisp variable to 0.
- iv. Create the hut function to draw a hut using the graphics library.
- v. Create the DrawManAndUmbrella function that takes x and ldisp as parameters and uses the graphics library to draw a man with an umbrella.
- vi. Create the Rain function that takes x as a parameter and uses the graphics library to simulate rainfall.
- vii. Create the rainbow function.
- viii. Declare variables x, y, and i.
- ix. Draw a circle for the sun using the graphics library.
- x. Use a loop to draw the rainbow animation.
- xi. Delay for animation effect.
- xii. Get user input to exit the loop.
- xiii. In the main function:
- xiv. Declare variables gd, gm, and x.
- xv. Initialize the graphics mode using initgraph.
- xvi. Create a loop until a key is pressed.
- xvii. Call the hut function.
- xviii. Draw the sun using the graphics library.
- xix. Draw the ground line.
- xx. Call the Rain function.
- xxi. Update ldisp using modulo operator.
- xxii. Call the DrawManAndUmbrella function.
- xxiii. Delay for animation effect.
- xxiv. Clear the screen.
- xxv. Update x using modulo operator.
- xxvi. Update ldisp using modulo operator.
- xxvii. Call the DrawManAndUmbrella function.
- xxviii. Call the rainbow function.
- xxix. Get user input to exit the program.
- xxx. Note: This algorithm is a representation of the code provided. The code itself uses the Turbo C graphics library, which may not be supported in modern systems. You may need to modify the code or use alternative graphics libraries to run it successfully.

6.IMPLEMENTATION(CODING):

```
#include <conio.h>
#include <graphics.h>
#include <stdio.h>
#define ScreenWidth getmaxx()
#define ScreenHeight getmaxy()
#define GroundY ScreenHeight * 0.75
int ldisp = 0;

// Creating a hut
void hut()
{
    setcolor(WHITE);
    rectangle(150, 180, 250, 300);
    rectangle(250, 180, 420, 300);
    rectangle(180, 250, 220, 300);

    line(200, 100, 150, 180);
    line(200, 100, 250, 180);
    line(200, 100, 370, 100);
    line(370, 100, 420, 180);

    setfillstyle(SOLID_FILL, BROWN);
    floodfill(152, 182, WHITE);
    floodfill(252, 182, WHITE);
    setfillstyle(SLASH_FILL, BLUE);
    floodfill(182, 252, WHITE);
    setfillstyle(HATCH_FILL, GREEN);
    floodfill(200, 105, WHITE);
    floodfill(210, 105, WHITE);
}

// Drawing a Man with
// an umbrella
void DrawManAndUmbrella(int x,
                        int ldisp)
{
    circle(x, GroundY - 90, 10);
    line(x, GroundY - 80, x,
        GroundY - 30);
    line(x, GroundY - 70,
        x + 10, GroundY - 60);
    line(x, GroundY - 65, x + 10,
        GroundY - 55);
    line(x + 10, GroundY - 60,
        x + 20, GroundY - 70);
```

```
    line(x + 10, GroundY - 55,
        x + 20, GroundY - 70);

    line(x, GroundY - 30,
        x + ldisp, GroundY);
    line(x, GroundY - 30,
        x - ldisp, GroundY);

    pieslice(x + 20, GroundY - 120,
        0, 180, 40);
    line(x + 20, GroundY - 120,
        x + 20, GroundY - 70);
}

// Creating the Rainfall
void Rain(int x)
{
    int i, rx, ry;
    for (i = 0; i < 400; i++)
    {
        rx = rand() % ScreenWidth;
        ry = rand() % ScreenHeight;
        if (ry < GroundY - 4)
        {
            if (ry < GroundY - 120 ||
                (ry > GroundY - 120 &&
                 (rx < x - 20 ||
                  rx > x + 60)))
                line(rx, ry,
                    rx + 0.5, ry + 4);
        }
    }
}

// Creating the rainbow
void rainbow()
{
    int x, y, i;

    circle(ScreenWidth - 100,
        50, 30);
    setfillstyle(SOLID_FILL,
        YELLOW);
    floodfill(ScreenWidth - 100,
        50, WHITE);

    ldisp = (ldisp + 2) % 20;
    DrawManAndUmbrella(x, ldisp);
    hut();
    x = getmaxx() / 5;
    y = getmaxy() / 5;
```

```
for (i = 30; i < 100; i++)
{
    // for animation
    delay(50);

    setcolor(i / 10);

    arc(x, y, 0, 180, i - 10);
}
getch();
}

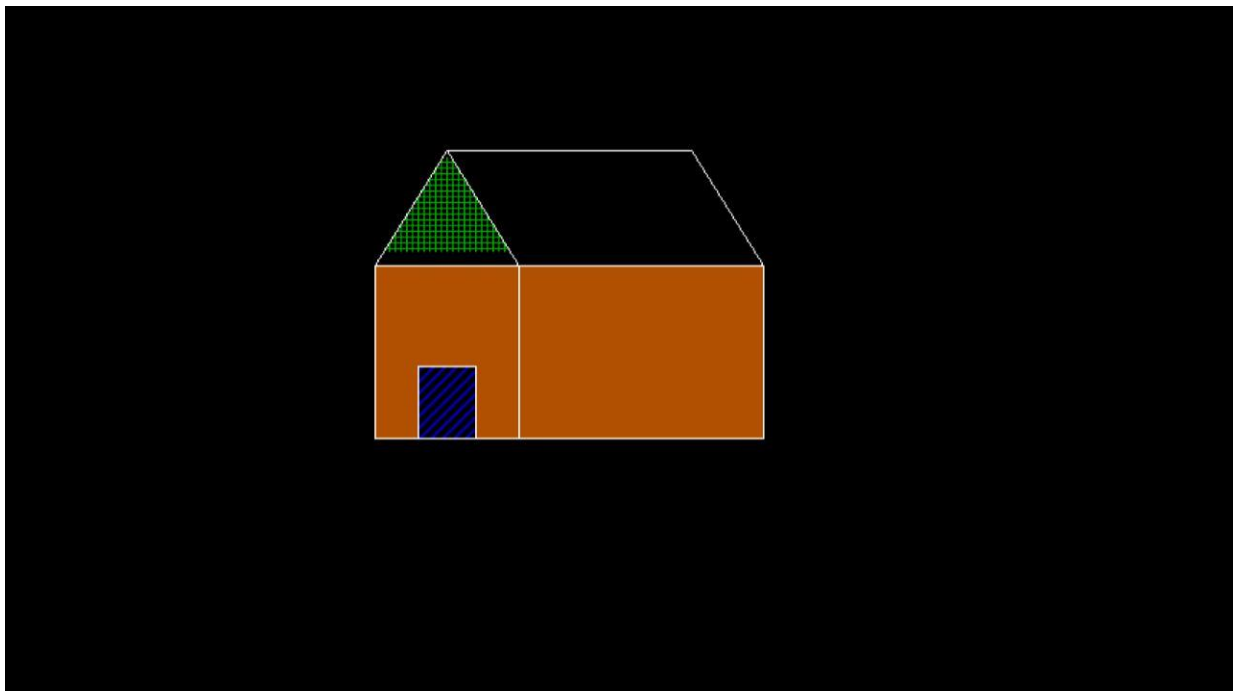
// Driver code
void main()
{
    int gd = DETECT, gm, x = 0;

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

    // executes till any key
    // is pressed
    while (!kbhit())
    {
        hut();
        circle(ScreenWidth - 100,
            50, 30);
        setfillstyle(SOLID_FILL,
            YELLOW);
        floodfill(ScreenWidth - 100,
            50, WHITE);
        line(0, GroundY, ScreenWidth,
            GroundY);
        Rain(x);

        ldisp = (ldisp + 2) % 20;
        DrawManAndUmbrella(x, ldisp);
        delay(20);
        cleardevice();
        x = (x + 2) % ScreenWidth;
    }
    // if the key is pressed the
    // rain stops, rainbow appears
    ldisp = (ldisp + 2) % 20;
    DrawManAndUmbrella(x, ldisp);
    rainbow();
    getch();
}
```

®

7.TESTING & 8. OUTPUT SCREEN SHOTS ::

9. CONCLUSION

In conclusion, the "Man in Rain" animated graphics program has successfully achieved its objective of simulating a realistic rain scenario and creating an immersive user experience. By utilizing the C programming language and graphics libraries such as OpenGL, SDL, or Allegro, this project has demonstrated the fusion of artistry and technical expertise required in computer graphics programming.

Through the implementation of features like rain simulation, an animated figure, graphics rendering, and user interaction, the program has brought the essence of a rainy day to life on the computer screen. Realistic raindrops, synchronized motion of the figure, and efficient graphics rendering contribute to the overall visual appeal of the program.

The project has expanded our understanding of fundamental graphics concepts, animation techniques, and user interaction. It has also highlighted the versatility and power of the C programming language in the domain of computer graphics.

In summary, the "Man in Rain" program showcases the creative possibilities within the realm of computer graphics. It has provided valuable insights into graphics programming and has been an enlightening journey that enhances programming skills while appreciating the intricacies of computer-generated visuals.

10.REFERENCES

- Donald, Hearn. Computer Graphics. Pearson Education: 2005 , Pearson Education
- www.learnsteert.com
- <http://www.sourcecodesworld.com>