

Report On

ROCKET LAUNCH USING TURBO C GRAPHICS

Submitted in partial fulfilment of the requirements of the Course project in
Semester III of Second Year Computer Engineering

By

Sambit Mazumder (Roll No. 34)

Supervisor
Prof. Akshaya Prabhu



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering



(2023-24)

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

CERTIFICATE

This is to certify that the project entitled “Rocket Launch Using Turbo C Graphics” is a Bonafide work of Sambit Mazumder (Roll No. 34) submitted to the University of Mumbai in partial fulfilment of the requirement for the Course project in semester III of Second Year Computer Engineering.

Supervisor

Prof. Akshaya Prabhu

Internal Examiner

External Examiner

Dr Megha Trivedi
Head of Department

Dr. H.V. Vankudre
Principal

Index

	Pg
1. Abstract	1
2. Description	2
3. Code	5
4 Result	7
5 Conclusion	8
6 Plagiarism	10

Abstract:

This project presents a simulated rocket launch animation implemented in Turbo C graphics, aiming to achieve a high degree of realism within the constraints of this legacy development environment. The program showcases a rocket taking off with a detailed graphical representation.

The rocket's design incorporates a realistic body with lifelike colours, and a nose cone, resembling traditional rocket structures. Fins are added for an authentic appearance. The animation begins with the rocket on the ground and progresses to a liftoff phase, during which flames and exhaust gases emerge, adding an additional layer of realism.

While Turbo C's graphical capabilities are limited, the program maximizes its potential to provide a more realistic rocket launch experience. The project's primary objective is to offer an engaging and visually appealing simulation within the classic Turbo C environment.

Problem Statement:

Design and implement a program to create an animated scene of a rocket launch. The program should use the Turbo C graphics library to produce the animation.

DESCRIPTION:

1

The "Realistic Rocket Launch Animation in Turbo C" project aims to create an engaging and visually impressive simulation of a rocket launch using the Turbo C programming environment. This project leverages the graphic capabilities of Turbo C to provide a detailed and aesthetically appealing representation of a rocket launch, focusing on realism and user engagement.

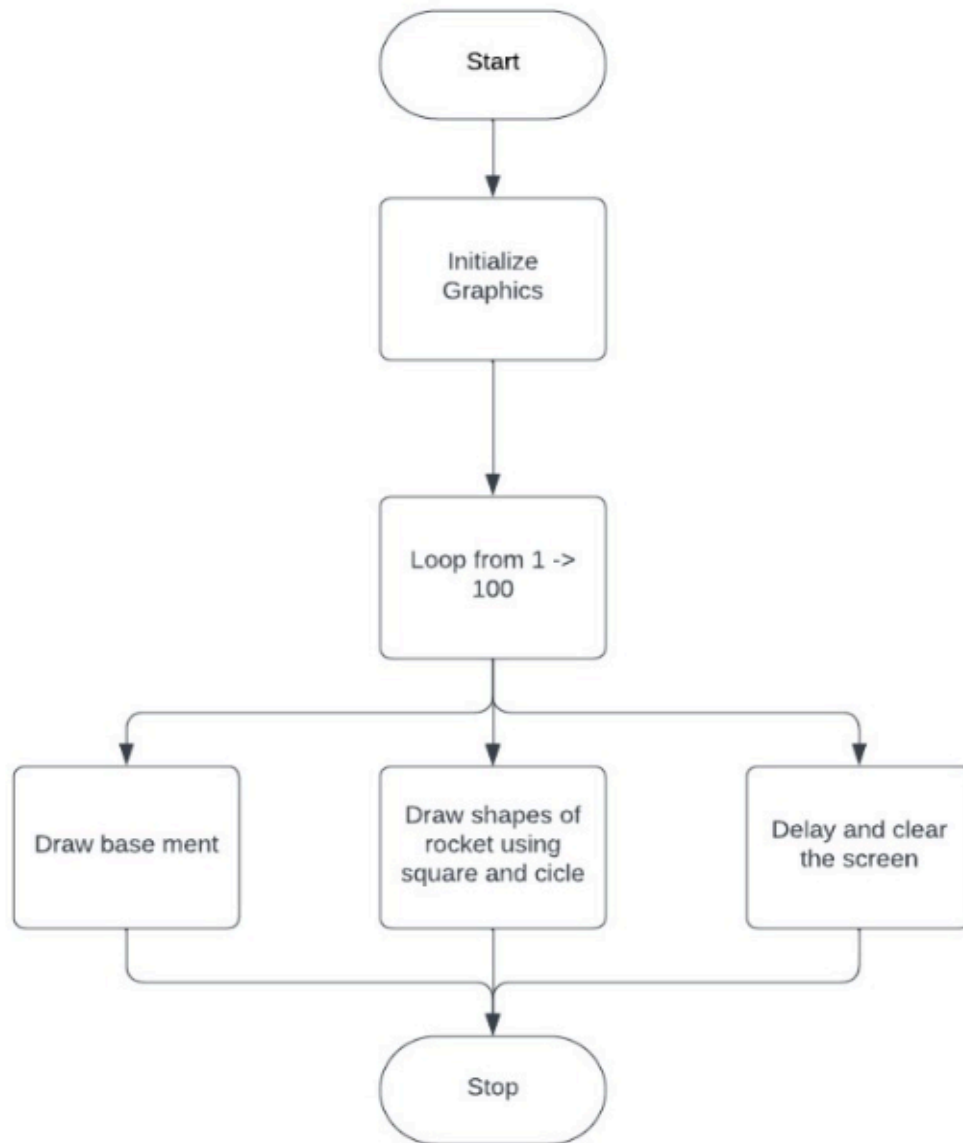
Key Features of the Project:

1. ***Highly Detailed Rocket Design***: The project presents a rocket with a meticulously designed body, nose cone, and fins. The rocket's appearance closely resembles that of a traditional space vehicle.
2. ***Realistic Color Scheme***: The rocket is filled with colours that emulate the realistic appearance of rocket materials. The body is adorned with a light gray shade, the nose cone displays a dark gray hue, and yellow flames realistically depict rocket propulsion.
3. ***Dynamic Animations***: The rocket launch animation progresses through several stages. It begins with the rocket on the ground, then showcases liftoff where flames and exhaust gases emerge, and finally reaches the climax of the launch.
4. ***Aesthetic Appeal***: The project prioritizes aesthetics and provides an engaging visual experience for the user. The rocket's design and the animation sequence are crafted to capture the essence of a rocket launch.

5. ***Turbo C Compatibility***: The project has been tailored to run within the Turbo C environment, making it accessible to those interested in creating visually stimulating programs with older development tools.

The "Realistic Rocket Launch Animation in Turbo C" project offers a combination of graphical artistry and programming skills, pushing the boundaries of what can be achieved in a legacy development environment. This simulation not only showcases the capabilities of Turbo C but also provides an enjoyable and educational experience for those interested in space exploration and computer graphics.

FlowChart:



Code:

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

void main()
{
    int gd = DETECT, gm,i;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

    for (i = 0; i < 350; i++)
    {
        // Basement
        line(225, 450, 225, 430);
        line(425, 450, 425, 430);
        line(225, 430, 425, 430);
        line(225, 450, 425, 450);

        line(275, 428 - i, 305, 395 - i);
        line(375, 428 - i, 345, 395 - i);
        line(275, 428 - i, 375, 428 - i);
        line(305, 395 - i, 345, 395 - i);
        setfillstyle(2, 5);
        floodfill(325, 400 - i, 15);

        line(305, 395 - i, 305, 285 - i);
        line(345, 395 - i, 345, 285 - i);
        line(305, 285 - i, 345, 285 - i);
        setfillstyle(6, 5);
        floodfill(320, 300 - i, 15);

        line(305, 285 - i, 325, 250 - i);
        line(345, 285 - i, 325, 250 - i);
        setfillstyle(1, 4);
        floodfill(315, 270 - i, 15);

        line(305, 370 - i, 285, 370 - i);
```



```
line(285, 370 - i, 305, 340 - i);  
setfillstyle(1, 4);  
floodfill(290, 368 - i, 15);
```

```
line(345, 370 - i, 365, 370 - i);  
line(365, 370 - i, 345, 340 - i);  
setfillstyle(1, 4);  
floodfill(350, 368 - i, 15);
```

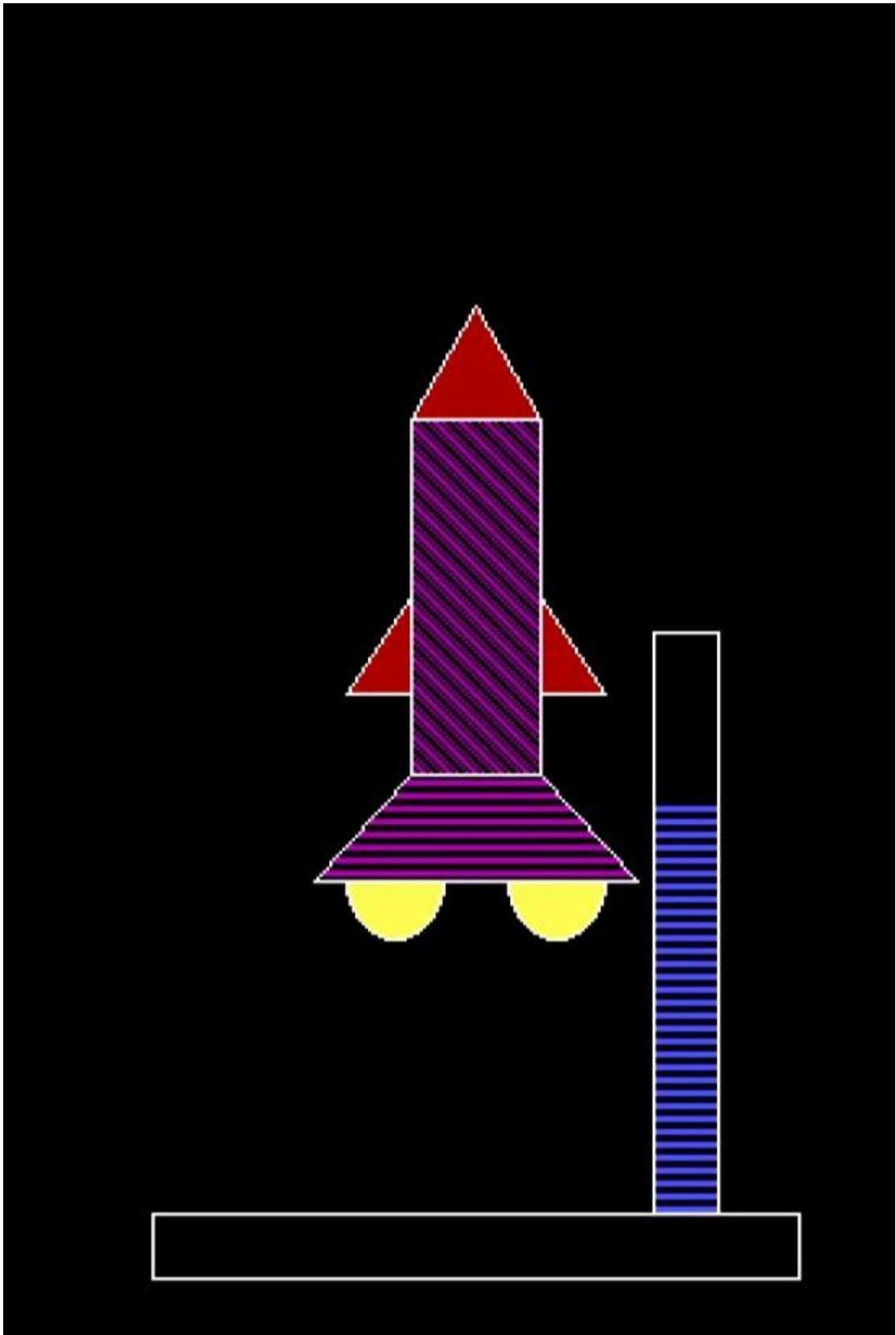
```
if (i > 10)  
{  
    ellipse(300, 428 - i, 180, 360, 15, 18);  
    setfillstyle(1, 14);  
    floodfill(299, 435 - i, 15);  
    ellipse(350, 428-i, 180, 360, 15, 18);  
    setfillstyle(1, 14);  
    floodfill(351, 435 - i, 15);  
}
```

```
line(380, 430, 380, 250);  
line(400, 430, 400, 250);  
line(380, 250, 400, 250);  
setfillstyle(2, 9);  
floodfill(390, 420, 15);
```

```
delay(5);  
cleardevice();  
}
```

```
closegraph();  
}
```

Result:



CONCLUSION:

1. Graphics Environment:

- The code is designed to work in a graphics environment using Turbo C and the BGI (Borland Graphics Interface) graphics library. This environment was commonly used for developing simple 2D graphics applications on older DOS-based systems.

2. Functionality Offered:

- The code creates an animation that appears to be a building or structure.
- It involves drawing and updating various lines and shapes to simulate the animation.
- The animation starts with a basic building shape and gradually modifies it by changing the appearance of its "basement," "windows," and "roof."
- The animation includes the movement of ellipses that might represent some characters or objects (e.g., windows opening or closing).
- The animation continues for 350 frames (controlled by the loop) with a delay of 5 milliseconds between each frame.

3. Character or Object:

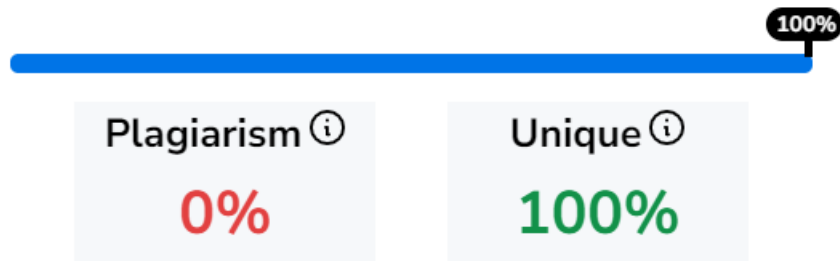
- The main character or object in this animation is a building or structure. It's represented using lines and shapes.
- The basement of the building is drawn using lines, and it appears to have windows.

- There are ellipses drawn that might symbolize objects or characters like windows. These ellipses change in appearance as the animation progresses.

- Additionally, there's an implied movement in some elements of the building (e.g., the "basement" lines and ellipses), suggesting that the building might be undergoing some changes over time.

Overall, the code leverages the Turbo C graphics environment to create a basic animation of a building or structure, demonstrating the capabilities of the BGI graphics library for 2D graphics rendering.

PLAGIARISM:



References:

1. Books on Graphics Programming with Turbo C:
 - "Graphics Programming with Turbo C" by Steve Rimmer.
 - "Graphics Programming in C" by J.H. Tewari.
2. Online Forums and Communities: Platforms like Stack Overflow and programming forums.
3. YouTube Video Tutorials: Video tutorials on platforms like YouTube.
4. Rocket Launch Animations and References: References and videos of actual rocket launches for understanding rocket dynamics.
5. GitHub and Code Repositories: Online code repositories like GitHub with open-source graphics projects.