

# **Car Moving in a linear Path**

Submitted in partial fulfillment of the requirements  
of degree of

## **S. E. Computer Engineering**

By

<b>Jaimin Solanki</b>	<b>33</b>	<b>192110</b>
<b>Rohan Tondlekar</b>	<b>39</b>	<b>192116</b>
<b>Keegan Vaz</b>	<b>42</b>	<b>192120</b>
<b>Alex Victor</b>	<b>43</b>	<b>192121</b>

Guide:

**Mr. Rupesh Mishra**  
Assistant Professor



Department of Computer Engineering  
St. Francis Institute of Technology  
(Engineering College)

University of Mumbai  
2020-2021

## **CERTIFICATE**

This is to certify that the project entitled“**Car Moving in a Linear Path**”is a bonafide work of”**Jaimin Solanki(33), Rohan Tondlekar(39), Keegan Vaz(42), Alex Victor(43)**” submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of S.E. in Computer Engineering

**(Name and sign)**  
**Guide**

**(Name and sign)**  
**Head of Department**

**(Name and sign)**  
**Principal**

# Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

-----  
(Signature)

Jaimin Solanki(33)  
Rohan Tondlekar(39)  
Keegan Vaz(42)  
Alex Victor(43)

-----  
(Name of student and Roll No.)

Date:

## **Abstract**

**Our project is a demonstration in which the car would be moving in a straight line from left to right on the screen , we have different functions like Line, Rectangle, Circle, Flood Fill ,Boundary Fill, etc. We have tried to implement the topics in CG which we learnt this semester. Our project can be used for educational purposes such as in Explaining Linear Motion of a Car with little modification as per user.**

## Contents

Chapter		Contents	Page No.
1		<b>INTRODUCTION:</b> Give at least two to three sentences about your project.	
	1.1	<b>Description (<i>Brief description of project</i>)</b> The main functionality of the project should be explained in brief	
	1.2	<b>Problem Formulation (<i>Explain the problem</i>)</b>	
	1.3	<b>Motivation (<i>need of the project</i>):</b> List the various approaches along with its drawbacks for solving the problem and briefly explain the approach used for your project.	
	1.4	<b>Proposed Solution:</b> Explain the method/technique used for solving the problem and how it overcomes the drawbacks mentioned under heading 1.3. Also explain how the project is going to help end users.	
	1.5	<b>Scope of the project (<i>scale/range of your project</i>):</b> This can be in terms of domain or application related constraints/limitations.	
2		<b>REVIEW OF LITERATURE (<i>include at least 3IEEE or similar reputed technical papers as reference</i>)</b> Should be atleast 2 pages which gives the ideas referenced by the reference papers. Mark the references wherever appropriate. (Note: - Please don't write the paper titles and the abstract of papers.)	
3		<b>SYSTEM ANALYSIS</b>	
	3.1	<b>Functional Requirements(<i>write requirements of the project</i>)</b> Should follow the IEEE SRS format	
	3.2	<b>Non Functional Requirements</b> Should follow the IEEE SRS format	
	3.3	<b>Specific Requirements (<i>Hardware and software requirements</i>)</b>	
	3.4	<b>Use-Case Diagrams and description</b> (Application development projects use-case is mandatory)	
4		<b>ANALYSIS MODELING</b>	
	4.1	<b>Data Modeling (<i>E-R Model if any with its associated Data dictionary</i>)</b> Applicable for those applications which are dependent on data storage and retrieval. ER Diagram normalized till the third normal form accompanied by the respective data dictionary table should be included	
	4.2	<b>Activity Diagrams / Class Diagram</b> Depending on the type of your project you may include any of the diagrams.	
	4.3	<b>Functional Modeling (DFDs <i>with specifications</i>)</b> mandatory for all projects	

	<b>4.4</b>	<b>TimeLine Chart</b> <i>(For semester 7and 8)</i>	
<b>5</b>		<b>DESIGN</b>	
	<b>5.1</b>	<b>Architectural Design</b> <i>(Project Flow /architecturewith description)</i>	
	<b>5.2</b>	<b>User Interface Design</b> GUI for your project	
<b>6</b>		<b>IMPLEMENTATION</b> (if any)	
	<b>6.1</b>	<b>Algorithms / Methods Used</b> Mention your algorithms if any or any methodology used.	
	<b>6.2</b>	<b>Working of the project</b> <i>(code for mentioned algorithms)</i>	
<b>7</b>		<b>CONCLUSIONS</b>	

Appendix

References

Publications by your group (if any)

Acknowledgments

## **Introduction**

Our Project is an animation of a Car Performing Linear motion in a straight line from left to right on the screen. In this we used different functions to create different parts of the car and for the movement of the car from left to right.

## **Problem Definition**

Applications of Computer Graphics were implemented to show the design of the car and animation of the car from the left to the right of the screen. Also, different shapes and line functions were used to design the car.

## **Scope**

To Demonstrate Linear Motion along a straight path. This output could be used for Educational Purposes like animated videos explaining physics concepts like speed and velocity.

# Implementation

## Code:

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

int main()
{
    int gd=DETECT,gm, i, maxx, cy;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    setbkcolor(WHITE);
    setcolor(RED);
    maxx = getmaxx();
    cy = getmaxy()/2;
    for(i=0;i<maxx-140;i++)
    {
        cleardevice();
        line(0+i,cy-20, 0+i, cy+15);
        line(0+i, cy-20, 25+i, cy-20);
        line(25+i, cy-20, 40+i, cy-70);
        line(40+i, cy-70, 100+i, cy-70);
        line(100+i, cy-70, 115+i, cy-20);
        line(115+i, cy-20, 140+i, cy-20);
    }
}
```



```
line(0+i, cy+15, 18+i, cy+15);
circle(28+i, cy+15, 10);
circle(28+i, cy+15, 7);
circle(28+i, cy+15, 5);
line(38+i, cy+15, 102+i, cy+15);
circle(112+i, cy+15, 10);
circle(112+i, cy+15, 7);
circle(112+i, cy+15, 5);
line(122+i, cy+15, 140+i, cy+15);
line(140+i, cy+15, 140+i, cy-20);
rectangle(50+i, cy-62, 90+i, cy-30);
setfillstyle(1, BLUE);
floodfill(5+i, cy-15, RED);
setfillstyle(1, LIGHTBLUE);
floodfill(52+i, cy-60, RED);
delay(70);
}
getch();
closegraph();
return 0;
}
```

## **Functions Used:**

- Line
- getmaxx()
- Rectangle
- BoundaryFill
- FloodFill
- Circle

## **Results(Snapshots)**

Video Link:

<https://github.com/Keegan-CalibrantTech/Linear-Car-Movement/blob/master/Final%20cut.mp4>

## **Conclusion**

We are able to showcase a car moving from left to right in a straight line on the screen by implementing the CG Topics Learned in this Semester.

## **References**

<https://www.geeksforgeeks.org/introduction-to-computer-graphics/>