A
Project Report
On

”Program for Bouncing Ball Animation in C”

Submitted By

Ms. Sonawane Vaishnavi [206] **Ms. Temgar Shravani [214]**

Under the guidance of

Prof. Amol Deokate

Department of Computer Technology
Sanjivani Rural Education Society’s
Sanjivani K.B.P. Polytechnic
Kopargaon – 423601, Dist.-Ahmadnagar

2022-2023

Sanjivani Rural Education Society's

Sanjivani K.B.P. Polytechnic

Department of Computer Technology

CERTIFICATE

This is to certify that the project report entitled

”Program for Bouncing Ball Animation in C”

Submitted by

Ms. Sonawane Vaishnavi [206] **Ms. Temgar Shravani [214]**

Under our supervision and guidance for partial fulfillment of the requirement for Diploma in Computer Technology affiliated to Maharashtra State Board of Technical Education, Mumbai.

For academic year
2022-2023

Prof. Amol Deokate
Project Guide

Mr. G.N. Jorvekar
Head of Department

Acknowledgement

First and the foremost we, express my deep sense of gratitude, sincere thanks and deep sense of appreciation to Project Guide Prof. Amol Deokate, Department of Computer Technology,Sanjivani K.B.P. Polytechnic, Kopargaon. Your availability at any time throughout the year, valuable guidance, opinion, view, comments, critics, encouragement, and support tremendously boosted this project work.

Lots of thanks to Mr. G. N. Jorvekar, Head Of Department Computer Technology Department, for providing us the best support we ever had. We like to express my sincere gratitude to Mr. A.R. Mirikar, Principal, Sanjivani K. B. P. Polytechnic, Kopargaon for providing a great platform to complete the project within the scheduled time.

We are also thankful to all the faculty members, Computer Technology Department, Sanjivani K. B. P. Polytechnic, Kopargaon for giving comments for improvement of work, encouragement and help during completion of the Project.

Last but not the least, we should say thanks from our bottom of heart to my Family and Friends for their never ending love, help, and support in so many ways through all this time.

Ms. Sonawane Vaishnavi
Ms. Temgar SHravani

Index

Sr. No.	Contents	Page No.
1.	Introduction	5
2.	Source Code	7
3.	Flowchart	9
4.	Source Code	11
5.	Output	13
6.	Conclusion	15
7.	Overview	17
8.	Reference	18

1. Introduction

A bouncing ball is one of the simplest physics simulations yet provides a novice graphics programmer with a host of useful experience. The student creates a ball that bounces off the walls of a box, adds a bat, then gamifies the whole experience into a simple block-out game.

The assignment is designed as the first significant piece of programming on a 2D computer graphics course. It is designed to be accessible to students who have taken an introductory programming course and who have physics and algebra to the level of a high-school graduate.

In computer graphics, use graphics.h which provide direct functions to draw different coordinate shapes (like circle, rectangle etc). By using these functions we can draw different objects like car, hut, trees, etc. In this program, we will draw a moving car using line and circles.

This bouncing ball program in C teaches you how to create a simple bouncing ball game using computer programs. In the console interface, the ball keeps moving. It reverses course when it comes into contact with the border. The ball is caught by the control stick, and the score is increased by one. The game is over if the ball is not caught.

Controlling the ball's movement direction via a variable is crucial to the effect of the ball movement. This variable changes depending on the boundary encountered and its current value, therefore the direction of flight is in the exact opposite direction.

Then, to conclude the game, set the ball to only hit the lower boundary where the stick is positioned, rather than touching the lower boundary where the stick is located.

Bouncing Ball Game Code in C Language it include a free Download Source Code. Simply locate the downloadable source code below and click the “Download Now” button. To start creating a Calculator in C Language, make sure that you have a Code Blocks or any platform of C installed in your computer.

Computer graphics are pictures and films created using computers. Usually, the term refers to computer-generated image data created with the help of specialized graphical hardware and software.

It is a vast and recently developed area of computer science. The phrase was coined in 1960, by computer graphics researchers Verne Hudson and William Fetter of Boeing. It is often abbreviated as CG, though sometimes erroneously referred to as computer-generated imagery (CGI).

Some topics in computer graphics include user interface design,sprite graphics,vector graphics,3D modeling,shaders,GPU design,implicit surface visualization with ray tracing, and computer vision, among others.

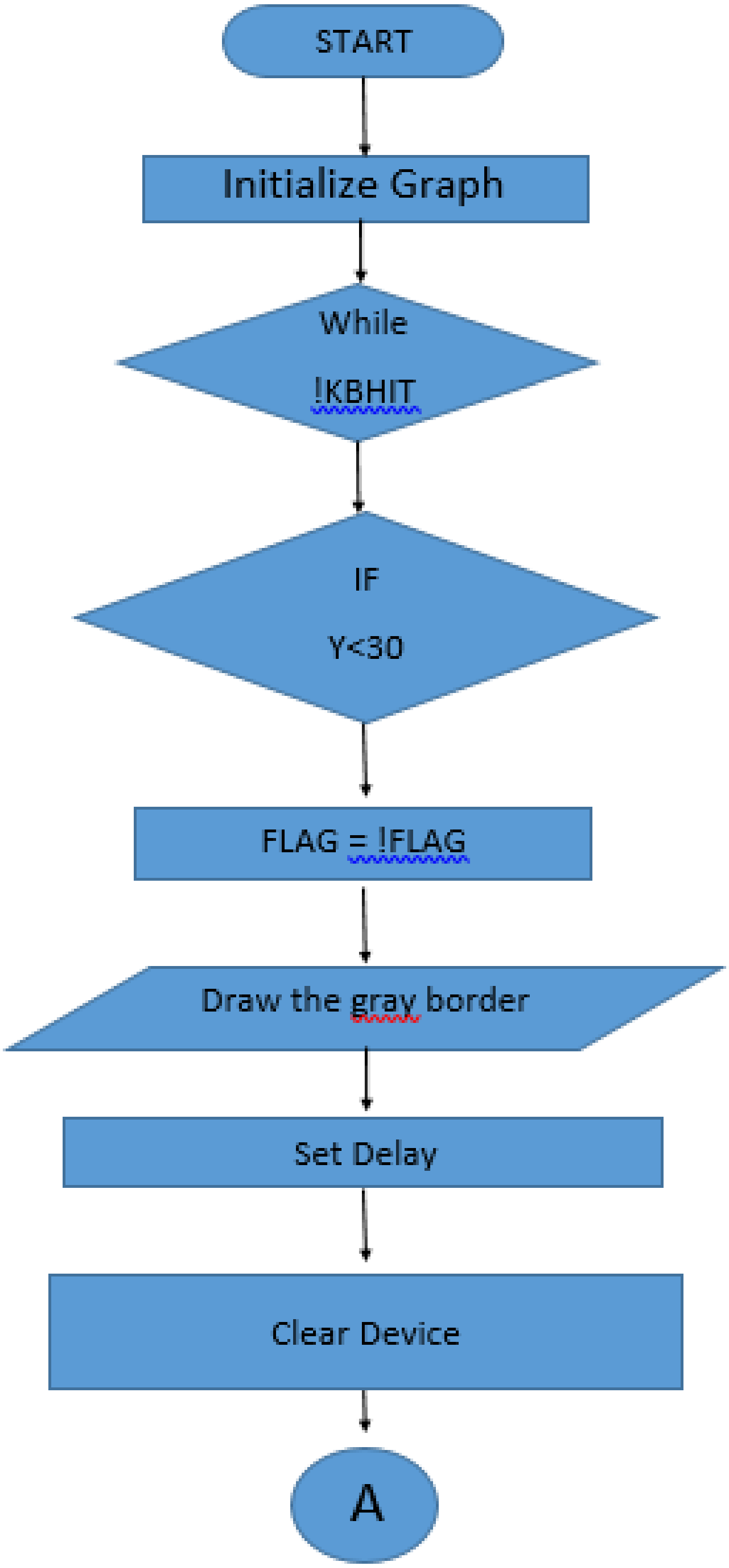
Computer graphics is responsible for displaying art and image data effectively and meaningfully to the consumer. It is also used for processing image data received from the physical world. Computer graphics development has had a significant impact on many types of media and has revolutionized animation,movies,advertising,video games, and graphic design in general.

2. Algorithm

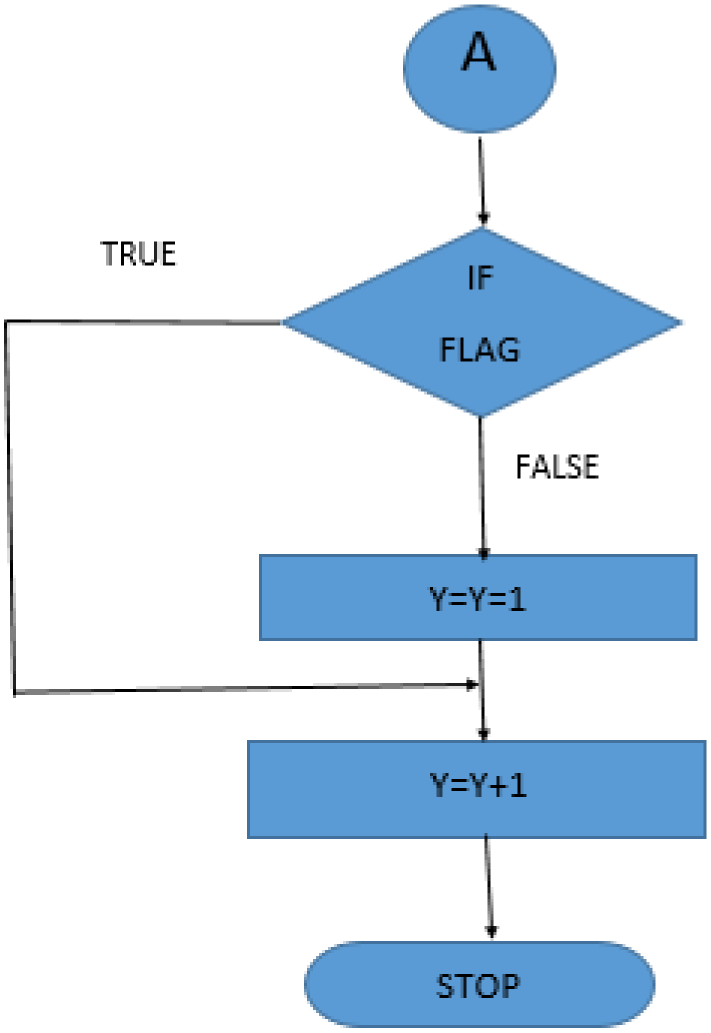
Algorithm:

```
Step 1: Start

Step 2: Initialize Graph

Step 3: Repeat following steps while !kbhit

Step 4: IF Y >= getmaxy OR Y <=30 THEN
          a. FLAG = not FLAG
          b.Draw the Grey border

Step 5: Set delay

Step 6: clear device

Step 7: IF flag THEN
        a. Y=Y+1

Step 8: ELSE
        Y=Y-5

Step 9: Stop
```

3. Flowchart


START
Initialize Graph
While
!KBHIT
IF
Y<30
FLAG = !FLAG
Draw the gray border
Set Delay
Clear Device
A

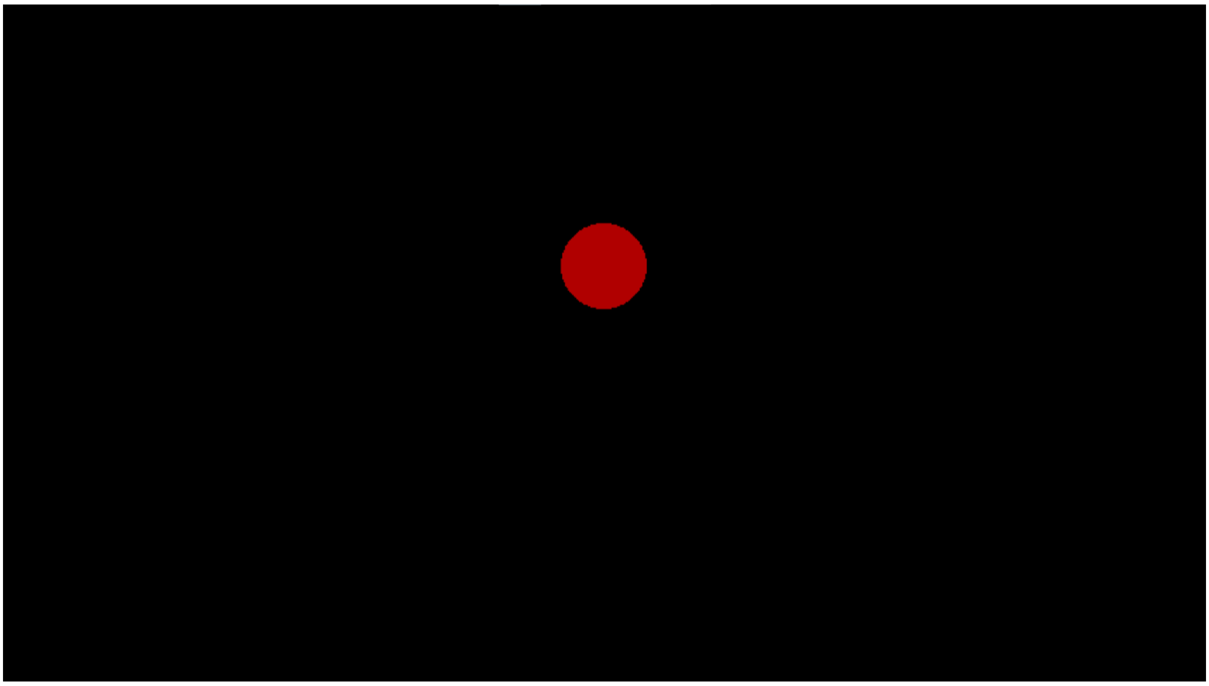
I

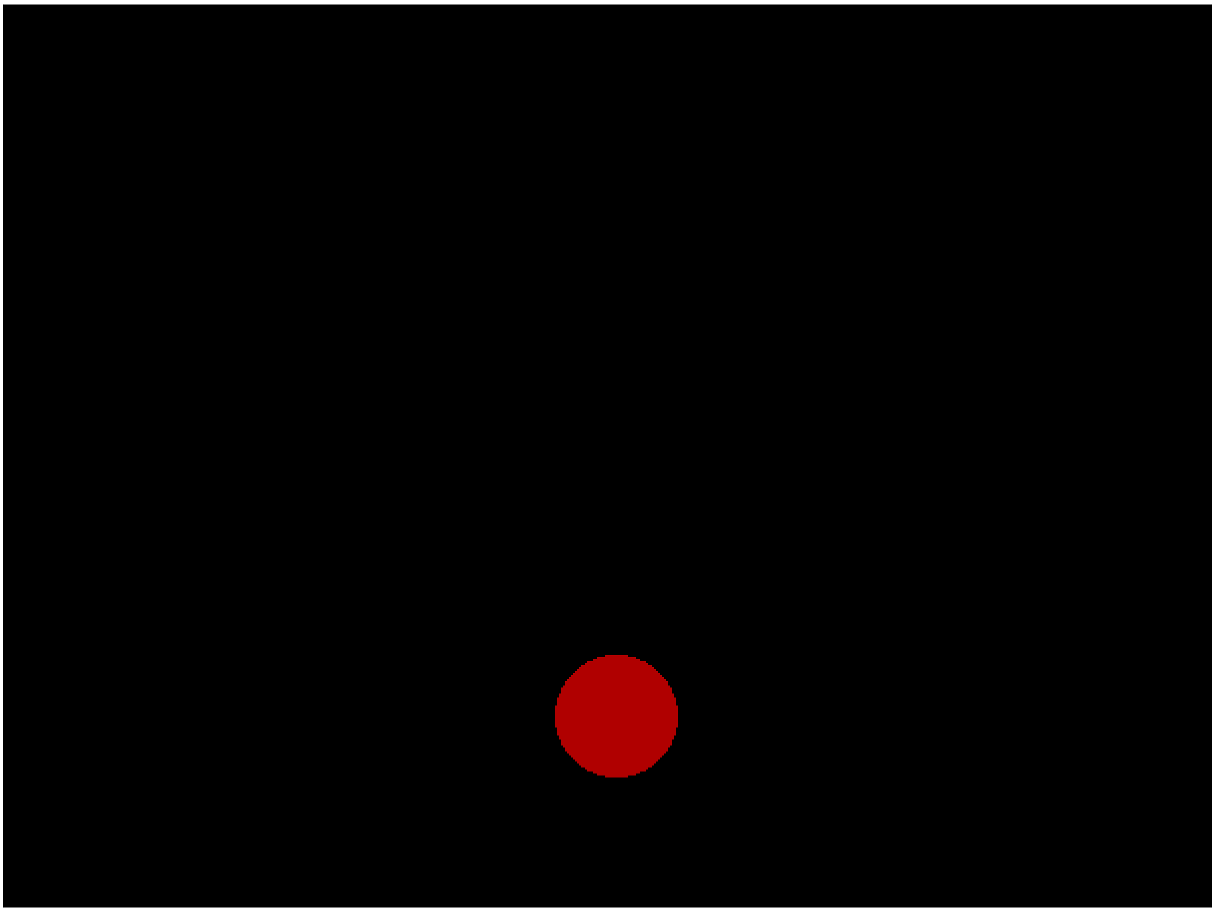


4. Source Code

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

int main() {
 int gd = DETECT, gm;
 int i, x, y, flag=0;
 initgraph(&gd, &gm, "C:\\TC\\BGI");

 /* get mid positions in x and y-axis */
 x = getmaxx()/2;
 y = 30;


 while (!kbhit()) {
  if(y >= getmaxy()-30 || y <= 30)
    flag = !flag;
    /* draws the gray board */
    setcolor(RED);
    setfillstyle(SOLID_FILL, RED);
    circle(x, y, 30);
    floodfill(x, y, RED);

 /* delay for 50 milli seconds */
 delay(50);
 /* clears screen */
 cleardevice();
 if(flag){
    y = y + 5;
 } else {
    y = y - 5;
 }
   }

   getch();
   closegraph();
   return 0;
}
```

5. Output





6. Conclusion

Computer graphics are pictures and films created using computers. Usually, the term refers to computer-generated image data created with the help of specialized graphical hardware and software. It is a vast and recently developed area of computer science.

The phrase was coined in 1960, by computer graphics researchers Verne Hudson and William Fetter of Boeing. It is often abbreviated as CG, though sometimes erroneously referred to as computer-generated imagery (CGI).

Some topics in computer graphics include user interface design, sprite graphics, vector graphics, 3D modeling, shaders, GPU design, implicit surface visualization with ray tracing, and computer vision, among others.

Computer graphics is responsible for displaying art and image data effectively and meaningfully to the consumer. It is also used for processing image data received from the physical world. Computer graphics development has had a significant impact on many types of media and has revolutionized animation, movies, advertising, video games, and graphic design in general.

The design of the project involves three different and mostly independent computations.

First, designing the simulation of the horizontal position over time, this in reality should just be a rather slow asymptotic growth, where the ball starts accelerating relatively fast in the beginning and slows down due to the forces of friction, and restitution (elasticity of the ball), but the actual design here will be different.

Second, the vertical motion of the ball governed by three coefficients of acceleration, gravity, elasticity or deformability of the ball and the coefficient of air resistance and. Finally, the projecting a ball on the screen of an oscilloscope for visual demonstration.

The design of the ball's horizontal position was changed several times, the earlier design involved restricting the ball in small an area where there are walls on both sides, and the ball bounces back and forth. Figure 1 shows what the position of the ball would look like over multiple steps.

A bouncing ball is one of the simplest physics simulations yet provides a novice graphics programmer with a host of useful experience. The student creates a ball that bounces off the walls of a box, adds a bat, then gamifies the whole experience into a simple block-out game. The assignment is designed as the first significant piece of programming on a 2D computer graphics course. It is designed to be accessible to students who have taken an introductory programming course and who have physics and algebra to the level of a high-school graduate.

This bouncing ball program in C teaches you how to create a simple bouncing ball game using computer programs. In the console interface, the ball keeps moving. It reverses course when it comes into contact with the border. The ball is caught by the control stick, and the score is increased by one. The game is over if the ball is not caught.

Controlling the ball's movement direction via a variable is crucial to the effect of the ball movement. This variable changes depending on the boundary encountered and its current value, therefore the direction of flight is in the exact opposite direction.

Then, to conclude the game, set the ball to only hit the lower boundary where the stick is positioned, rather than touching the lower boundary where the stick is located.

7. Overview

In this program, we first draw a red color ball on screen having center at (x, y) and then erases it using cleardevice function. We again draw this ball at center (x, y + 5), or (x, y - 5) depending upon whether ball is moving down or up. This will look like a bouncing ball. We will repeat above steps until user press any key on keyboard.

This bouncing ball program in C teaches you how to create a simple bouncing ball game using computer programs. In the console interface, the ball keeps moving. It reverses course when it comes into contact with the border. The ball is caught by the control stick, and the score is increased by one. The game is over if the ball is not caught.

Controlling the ball's movement direction via a variable is crucial to the effect of the ball movement. This variable changes depending on the boundary encountered and its current value, therefore the direction of flight is in the exact opposite direction.

Then, to conclude the game, set the ball to only hit the lower boundary where the stick is positioned, rather than touching the lower boundary where the stick is located.

The goal of this project was to design and build a small analog computer called "The Bouncing Ball Simulator".

The Bouncing Ball Simulator was first designed by a German Company called Telefunken. The Telefunken design was aimed for existed general purpose analog computers. Since then few people tried to design them with stand-alone op-amps to my knowledge.

The Bouncing Ball Simulator calculates the position of a ball falling from some initial height and bouncing due to the Newton's law of motion.

What makes this an analog problem is the fact that you can simulate the horizontal and the vertical motion of the ball independently.

That feature of the problem fits analog because analog computations are done in parallel. Furthermore, analog computers above all exist because of the power to integrate; hence the problems solved are almost always a set of differential equations, and this problem is in fact a set of well-known differential equations

8. Reference

Sr. No.	Title of Book	Author	Publication
1.	Computer Graphics	Donald Hearn, Baker M. Pauline	Pearson Education, New-Delhi June 2012, ISBN: 978-1259004612
2.	Computer Graphics	Dr. Chopra Rajiv	S.Chand 2016, New Delhi , ISBN: 978-1259004612
3.	Computer Graphics	Foly James	Pearson Education, New-Delhi June 2014, ISBN: 978-1259004399526