

**A
MICRO-PROJECT
OF
OOPM (JAVA)
“BOUNCING BALL”**

Name of Group Members

Sarvesh Kadam

Atharva Shelke

Parth Gorathe

Kalpesh Vangujar

Guided By

Mr. Shital Agrawal



DEPARTMENT OF COMPUTER ENGINEERING

DILKAP COLLEGE OF ENGINEERING

2022-2023

SUBMISSION

We, students of first semester of first year diploma in computer engineering humbly submit that we have completed the microproject work as prescribed in this report by own skill and the study in academic session 2022-23, as per the instruction and guidance **Mr. Shital Agrawal**.

The work carried out in the project is our and not copied the report on any appreciable part from any other literature in contravention of the academic ethics.

The teacher has approved our contribution The students associated in the micro-project are:

Name Of Student

Sign

Sarvesh Kadam

.....

Atharva Shelke

.....

Parth Gorathe

.....

Kalpesh Vangujar

.....

Date:-

ACKNOWLEDGEMENT

We wish to avail this opportunity to acknowledge our profound indebtedness and extend our deep sense of gratitude of our profound **Mr. Shital Agrawal** for his valuable guidance, advice and encouragement that has been feel to successful completion of this micro-project.

We hereby express our deep gratitude to our H.O.D. and Hon'ble Principal for his cooperation and help and also the other staff members of the department.

We also thank to library for making us available necessary books for reference. I would like to place on record my sincere thanks to all persons directly or indirectly helped us in completion of this work.

Sarvesh Kadam

BOUNCING BALL MINI PROJECT

Brief Introduction

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. 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. In this mini project we have used JAVA basically to run the code.

Aim of the Mini-Project

This Mini-Project aims at:

1. Create a Bouncing-Ball Program.
2. Create and apply graphics to a ball to bounce.

Action Plan

S. No.	Details of activity	Planned Start date	Planned Finish date	Name of Responsible Team Members
1	Project Survey	05-08-2022	10-10-2022	SARVESH KADAM
2	Gathering Information	07-08-2022	01-09-2022	PARTH GORATHE & KALPESH VANGUJAR
3	Executed	18-09-2022	18-09-2022	ATHARVA SHELKE
4	Create Report	11-10-2022	16-10-2022	SARVESH KADAM & ATHARVA SHELKE

Resources Required

S. No.	Name of Resource / Material	Specification	Quantity
1	Hardware: Computer System	Computer(Intel i3 ,11th GEN), RAM 8GB,SSD 256GB	1
2	Operating System	Windows 7, 10 & 11 (Any one of them).	1
3	Software	VS CODE / Subline text	1

Bouncing Ball Information

Bouncing ball animation in Java using FPanel, is a simple GUI (Graphical User Interface) Animation. Now the question is that how actually they are moving in a square so basically Bouncing ball animation in FPanel we are using a "If Loop" and running this "If Loop" Infinite and changing their Ovals position. Writing a single ball bouncing inside a rectangular container box is straight forward, and can be accomplished with very few lines of codes, as follows

Dissecting BounceThread.java: I assume that you understand Java Graphics programming (AWT/Swing and custom painting), and the multi-threading issues involved.

Construction

In the constructor, we setup the UI components (set the preferred size for the JPanel). We then start a new thread to run the game update (moving the ball). We construct a JFrame as the application's main window. We set our custom JPanel as the content pane for the JFrame. This program, although works, is poor in design (in terms of modularity, reusability and expansibility). Moreover, the collision detection and response algorithm are crude. There is also no timing control.

Source Code

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class BounceThread {
    public static void main(String[] args) {
        JFrame frame = new BounceThreadFrame();
        frame.show();
    }
}

class BounceThreadFrame extends JFrame {
    private JPanel canvas;

    public BounceThreadFrame() {
        setSize(300, 200);
        setTitle("Bounce");

        Container contentPane = getContentPane();
        canvas = new JPanel();
        contentPane.add(canvas, "Center");
    }
}
```

```

JPanel p = new JPanel();
addButton(p, "Start", new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        Ball b = new Ball(canvas);
        b.start();
        return;
    }
});
addButton(p, "Close", new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        canvas.setVisible(false);
        System.exit(0);
    }
});
contentPane.add(p, "South");
}

public void addButton(Container c, String title,
ActionListener a) {
    JButton b = new JButton(title);
    c.add(b);

```

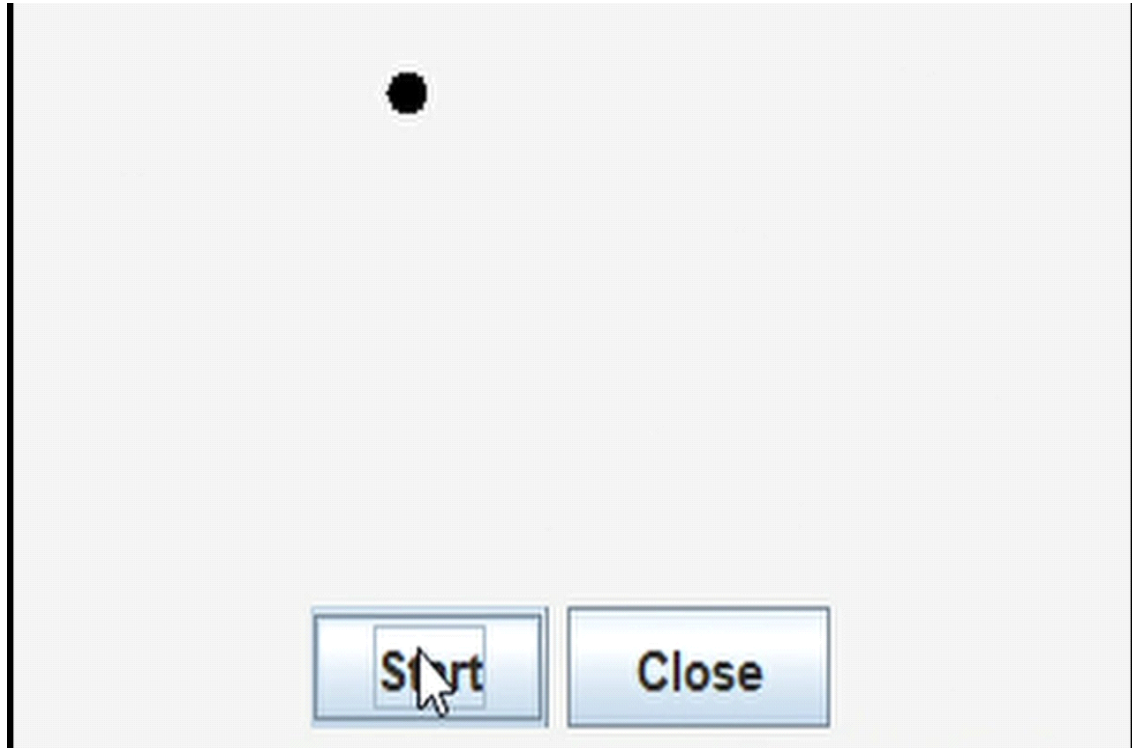
```
        b.addActionListener(a);
    }
}

class Ball extends Thread {
    private JPanel box;
    private static final int XSIZE = 10;
    private static final int YSIZE = 10;
    private int x = 0;
    private int y = 0;
    private int dx = 2;
    private int dy = 2;
    public Ball(JPanel b) {
        box = b;
    }
    public void draw() {
        Graphics g = box.getGraphics();
        g.fillOval(x, y, XSIZE, YSIZE);
        g.dispose();
    }
}
```

```
public void move() {  
    if (!box.isVisible()) return;  
    Graphics g = box.getGraphics();  
    g.setXORMode(box.getBackground());  
    g.fillOval(x, y, XSIZE, YSIZE);  
    x += dx;  
    y += dy;  
    Dimension d = box.getSize();  
    if (x < 0) {  
        x = 0;  
        dx = -dx;  
    }  
    if (x + XSIZE >= d.width) {  
        x = d.width - XSIZE;  
        dx = -dx;  
    }  
    if (y < 0) {  
        y = 0;  
        dy = -dy;  
    }  
}
```

```
}  
if (y + YSIZE >= d.height) {  
    y = d.height - YSIZE;  
    dy = -dy;  
}  
g.fillOval(x, y, XSIZE, YSIZE);  
g.dispose();  
}  
public void run() {  
    try {  
        draw();  
        while (true) {  
            move();  
            sleep(5);  
        }  
    } catch (Exception e) {}  
}  
}
```

Output



Conclusion

- Skill Developed / learning out of this Mini-Project
- Discipline knowledge: Apply Computer engineering discipline- specific knowledge to solve core computer engineering related problem.
- Experiments and practice: Plan to perform experiments and practices to use the results to solve broad-based Computer engineering problems.
- Engineering tools: Apply relevant Computer technologies and tools with an understanding of the limitations.
- Communication: Communication effectively in oral and written form.