OOPM Mini Project Report.

Problem Statement: Flappy Bird

Objectives:

Implementation of OOP studied in the current Semester in an interactive game. (Flappy Bird)

Introduction:

Flappy Bird was a mobile game developed by Vietnamese video game artist and programmer Dong Nguye. The game is a side-scroller where the player controls a bird, attempting to fly between columns of green pipes without hitting them.

Our project is a minimalized version of the implementation of that game, in which we have implemented the concepts of Object-Oriented Programming and made a simple yet fun UI-based game with the help of awt and swing.

The program detects the input from the user-provided through tapping Spacebar and every time the user taps the Spacebar, it gives a boost to the Y-axis coordinate of the bird. The motive of the game is to pass the maximum number of obstacles (pipes) without colliding.

Design:

Since the game is targeted towards a younger audience, we decided to design it with the theme of our favorite cartoon - ‘Shinchan’. The gameplay, rules, and controls of the game would stay the same as the classic Flappy Bird, just the theme/visuals would differ from the original.

Analysis:

The player controls the bird (head of Shinchan in this case), which moves persistently to the right. The player is tasked with navigating it through pairs of pipes that have equally sized gaps placed at random heights.
The head automatically descends and only ascends when the player taps the Spacebar. Each successful pass through a pair of pipes, award the player one point. Colliding with a pipe ends the gameplay.

TEAM:

Yash Sarang.
Om Gaydhane.
Faayez Muhammad.

Code with Results:

CODE:

```java
package com.flappybird.view;

import com.flappybird.controller.Controller;
import com.flappybird.model.Bird;
import com.flappybird.model.Tube;
import com.flappybird.model.TubeColumn;
import com.flappybird.model.proxy.ProxyImage;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.Rectangle;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import javax.swing.JPanel;
import javax.swing.Timer;

public class Game extends JPanel implements ActionListener {

    private boolean isRunning = false;
    private ProxyImage proxyImage;
    private Image background;
    private Bird bird;
    private TubeColumn tubeColumn;
    private int score;
    private int highScore;

    public Game() {

        proxyImage = new ProxyImage("/assets/background.png");
        background = proxyImage.loadImage().getImage();
        setFocusable(true);
        setDoubleBuffered(false);
        addKeyListener(new GameKeyAdapter());
        Timer timer = new Timer(15, this);
        timer.start();
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        Toolkit.getDefaultToolkit().sync();
        if (isRunning) {
            ////////////////////////////////
            bird.tick();
```

```
            tubeColumn.tick();
            checkColision();
            score++;
            ////////////////////////////////
        }

        repaint();
    }

    @Override
    public void paint(Graphics g) {
        Graphics2D g2 = (Graphics2D) g;
        g2.drawImage(background, 0, 0, null);
        if (isRunning) {
            ////////////////////////////////
            this.bird.render(g2, this);
            this.tubeColumn.render(g2, this);
            g2.setColor(Color.black);
            g.setFont(new Font("Arial", 1, 20));
            g2.drawString("Your score: " + this.tubeColumn.getPoints(), 10, 20);
            ////////////////////////////////
        } else {
            g2.setColor(Color.black);
             g.setFont(new Font("Arial", 1, 20));
            g2.drawString("Press Enter to Start the Game", Window.WIDTH / 2 - 150, Windo
            g2.setColor(Color.black);
            g.setFont(new Font("Arial", 1, 15));
            g2.drawString("", Window.WIDTH - 200, Window.HEIGHT - 50);
        }
        g2.setColor(Color.black);
        g.setFont(new Font("Arial", 1, 20));
        g2.drawString("High Score: " + highScore, Window.WIDTH - 160, 20);

        g.dispose();
    }

    private void restartGame() {
        if (!isRunning) {
            this.isRunning = true;
            this.bird = new Bird(Window.WIDTH / 2, Window.HEIGHT / 2);
            this.tubeColumn = new TubeColumn();
        }
    }

    private void endGame() {
        this.isRunning = false;
        if (this.tubeColumn.getPoints() > highScore) {
```

```
96              this.highScore = this.tubeColumn.getPoints();
97          }
98          this.tubeColumn.setPoints(0);
99
100     }
101
102     private void checkColision() {
103         Rectangle rectBird = this.bird.getBounds();
104         Rectangle rectTube;
105
106         for (int i = 0; i < this.tubeColumn.getTubes().size(); i++) {
107             Tube tempTube = this.tubeColumn.getTubes().get(i);
108             rectTube = tempTube.getBounds();
109             if (rectBird.intersects(rectTube)) {
110                 endGame();
111             }
112         }
113     }
114
115     // Key
116     private class GameKeyAdapter extends KeyAdapter {
117
118         private final Controller controller;
119
120         public GameKeyAdapter() {
121             controller = new Controller();
122         }
123
124         @Override
125         public void keyPressed(KeyEvent e) {
126             if (e.getKeyCode() == KeyEvent.VK_ENTER) {
127                 restartGame();
128             }
129         }
130
131         @Override
132         public void keyReleased(KeyEvent e) {
133             if (isRunning) {
134                 controller.controllerReleased(bird, e);
135             }
136         }
137     }
138 }
139
```

OUTPUT SCREENS: