

# **MJKP 2020 Rocket Launching**

## **Group Members**

Jomi Susan Mathew  
M Aswin Kishore  
Karthika Sankar  
Parvathy S Thampi

**CODE**

```

1  /*
2      MJKP 2020 ROCKET LAUNCHING
3      -----
4      TEAM MEMBERS
5      -----
6      Jomi Susan Mathew          36
7      M Aswin Kishore            41
8      Karthika Sankar            39
9      Parvathy S Thampi          49
10     -----
11
12     Build on JDK 8.0
13     Java version: 1.8.0_271
14
15     -----
16
17 */
18
19 import java.awt.*;
20 import java.awt.event.*;
21 import java.util.Random;
22 import javax.swing.*;
23
24 class Main extends JFrame implements ActionListener
25 {
26     // Coordinates of rocket
27     static int xRocket = 300;
28     static int yRocket = 300;
29
30     // To determine the number of stars
31     int timer = 5;
32
33     // Importing images
34     Image rocket = Toolkit.getDefaultToolkit().getImage("rocket.png");
35     Image rocket_NoEngine = Toolkit.getDefaultToolkit().getImage("rocket_noengine.png");
36
37     Button liftOff;
38     Label Data;
39
40     boolean Takeoff = false;
41
42     // Threads to repaint the screen and to Increment Timers
43     RepaintThread launch = new RepaintThread();
44     TimerThread LoopTimer = new TimerThread();
45
46     public Main()
47     {
48         Data = new Label("MJKP 2020");
49         Data.setBounds(840, 100, 110, 40);
50         Data.setFont(new Font("Times New Roman", Font.BOLD, 20));
51         Data.setForeground(Color.WHITE);
52         Data.setBackground(Color.BLACK);
53         add(Data);
54
55         liftOff = new Button("START/STOP");
56         liftOff.setBounds(820, 200, 150, 40);
57         liftOff.setFont(new Font("Times New Roman", Font.BOLD, 20));
58         liftOff.setForeground(Color.WHITE);
59         liftOff.setBackground(Color.BLACK);
60         liftOff.addActionListener(this);
61         add(liftOff);
62
63         Background panel = new Background();
64         add(panel);
65
66         setBackground(Color.BLACK);
67         setTitle("MJKP 2020 Rocket Launching");
68         setResizable(false);
69         setSize(1000, 800);
70         setVisible(true);
71
72     }
73
74     public void actionPerformed(ActionEvent e) // Called by pressing the button START/STOP

```

```

75 {
76     if (Takeoff == false)
77     {
78         Takeoff = true;
79         liftOff.setForeground(Color.ORANGE); // Set color of button to Orange
80
81         try
82         {
83             launch.start();
84             LoopTimer.start();
85         }
86         catch (Exception d)
87         {
88         }
89     }
90     else
91     {
92         // Setting Default Values
93         Takeoff = false;
94         liftOff.setForeground(Color.WHITE); // Set color of button to White
95         timer = 10;
96         xRocket = 300;
97         yRocket = 300;
98
99         try
100        {
101            launch.interrupt();
102            LoopTimer.interrupt();
103        }
104        catch (Exception d)
105        {
106        }
107    }
108 }
109
110 // Thread to repaint the Frame
111 class RepaintThread extends Thread
112 {
113     public void run()
114     {
115         while (true)
116         {
117             try
118             {
119                 repaint();
120                 Thread.sleep(17);
121             }
122             catch (Exception e)
123             {
124             }
125         }
126     }
127 }
128
129 // Thread to increment timer in Loop
130 class TimerThread extends Thread
131 {
132     public void run()
133     {
134         while (true)
135         {
136             if (timer >= 300)
137                 timer = 300;
138             else
139                 timer += 1;
140
141             try
142             {
143                 Thread.sleep(400);
144             }
145             catch (Exception e)
146             {
147             }
148         }
149     }

```

```

150     }
151 }
152 }
153
154 class Background extends JPanel implements MouseMotionListener
155 {
156
157     public Background()
158     {
159         addMouseMotionListener(this);
160     }
161
162     @Override
163     public void paintComponent(Graphics g)
164     {
165         Graphics sky = getGraphics();
166
167         if (!Takeoff) // Before TakeOff
168         {
169             g.setColor(Color.BLACK);
170             g.fillRect(0, 0, 800, 800);
171             g.drawImage(rocket_NoEngine, 300, 300, this);
172         }
173         else // After TakeOff
174         {
175             g.setColor(Color.BLACK);
176             g.fillRect(0, 0, 800, 800);
177             g.drawImage(rocket, xRocket, yRocket, this);
178
179             for (int i = 0; i < timer; i++) // Generating Stars Loop
180             {
181                 Random num = new Random(); // To create random X and Y coordinates to set the stars Location
182
183                 sky.setColor(Color.WHITE);
184                 sky.fillOval(num.nextInt(800), num.nextInt(800), 3, 3);
185             }
186
187             try
188             {
189                 Thread.sleep(17);
190             }
191             catch (Exception e)
192             {
193             }
194         }
195     }
196 }
197
198 public void mouseDragged(MouseEvent e)
199 {
200 }
201
202 public void mouseMoved(MouseEvent e)
203 {
204     if (Takeoff)
205     {
206         // Limit is set so that mouse location does not go beyond the Rectangular box
207         if (e.getX() <= 670 && e.getY() <= 700)
208         {
209             xRocket = e.getX() - 100;
210             yRocket = e.getY() - 5;
211         }
212     }
213 }
214 }
215
216 public static void main(String args[])
217 {
218     new Main();
219 }
220 }

```

# SCREENSHOT

