```java
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.applet.Applet;

public class RaceApplet extends Applet implements Runnable {
	private final static int NUMRUNNERS = 2;
	private final static int SPACING = 20;
	private Runner[] runners = new Runner[NUMRUNNERS];
	private Thread updateThread = null;

	public void init() {
		String raceType = getParameter("type");
		for (int i = 0; i < NUMRUNNERS; i++) {
			runners[i] = new Runner();
			if (raceType.compareTo("unfair") == 0)
				runners[i].setPriority(i+2);
			else
				runners[i].setPriority(2);
		}
		if (updateThread == null) {
			updateThread = new Thread(this, "Thread Race");
			updateThread.setPriority(NUMRUNNERS+2);
		}
		addMouseListener(new MyAdapter());
	}

	class MyAdapter extends MouseAdapter {
		public void mouseClicked(MouseEvent evt) {
			if (!updateThread.isAlive())
				updateThread.start();
			for (int i = 0; i < NUMRUNNERS; i++) {
				if (!runners[i].isAlive())
					runners[i].start();
			}
		}
	}

	public void paint(Graphics g) {
		g.setColor(Color.lightGray);
		g.fillRect(0, 0, getSize().width, getSize().height);
		g.setColor(Color.black);
		for (int i = 0; i < NUMRUNNERS; i++) {
			int pri = runners[i].getPriority();
			g.drawString(new Integer(pri).toString(), 0, (i+1)*SPACING);
		}
		update(g);
	}

	public void update(Graphics g) {
		for (int i = 0; i < NUMRUNNERS; i++) {
			g.drawLine(SPACING, (i+1)*SPACING,
				SPACING + (runners[i].tick)/1000, (i+1)*SPACING);
		}
	}

	public void run() {
		Thread myThread = Thread.currentThread();
		while (updateThread == myThread) {
			repaint();
			try {
				Thread.sleep(10);
			} catch (InterruptedException e) { }
		}
	}

	public void stop() {
		for (int i = 0; i < NUMRUNNERS; i++) {
			if (runners[i].isAlive())
				runners[i] = null;
		}
		if (updateThread.isAlive())
			updateThread = null;
	}

}
```

```
public class Runner extends Thread {
    public int tick = 1;
    public void run() {
        while (tick < 400000)
            tick++;
    }
}
```

```
public class RaceTest {

    private final static int NUMRUNNERS = 2;

    public static void main(String[] args) {
        SelfishRunner[] runners = new SelfishRunner[NUMRUNNERS];

        for (int i = 0; i < NUMRUNNERS; i++) {
            runners[i] = new SelfishRunner(i);
            runners[i].setPriority(2);
        }
        for (int i = 0; i < NUMRUNNERS; i++)
            runners[i].start();
    }
}
```

```
public class SelfishRunner extends Thread {

    private int tick = 1;
    private int num;

    public SelfishRunner(int num) {
        this.num = num;
    }

    public void run() {
        while (tick < 400000) {
            tick++;
            if ((tick % 50000) == 0)
                System.out.println("Thread #" + num + ", tick = " + tick);
        }
    }
}
```

```
public class RaceTest2 {

    private final static int NUMRUNNERS = 2;

    public static void main(String[] args) {
        PoliteRunner[] runners = new PoliteRunner[NUMRUNNERS];

        for (int i = 0; i < NUMRUNNERS; i++) {
            runners[i] = new PoliteRunner(i);
            runners[i].setPriority(2);
        }
        for (int i = 0; i < NUMRUNNERS; i++)
            runners[i].start();
    }
}
```

```
public class PoliteRunner extends Thread {

    private int tick = 1;
    private int num;

    public PoliteRunner(int num) {
        this.num = num;
    }

    public void run() {
        while (tick < 400000) {
            tick++;
            if ((tick % 50000) == 0) {
                System.out.println("Thread #" + num + ", tick = " + tick);
                yield();
            }
        }
    }
}
```