**LAB : 3**

**OBJECTIVE :**

Write an application that draws basic graphical primitives on the screen

**Requrements :**

(a) Windows PC (Windows 7/8/10) / Mac

(b) JDK 1.5

(c) Java Wireless Toolkit 2.5.2

**Implementation :**

**GraphicsPremitives.java :**

import javax.microedition.lcdui.\*;

import javax.microedition.midlet.\*;

import javax.microedition.io.\*;

import java.io.\* ;

import java.lang.\*;

public class GraphicsPremitives extends MIDlet implements CommandListener {

Display display;

Form form;

List menu;

Ticker ticker;

static final Command backCommand = new Command("Back", Command.BACK, 0);

static final Command exitCommand = new Command("Exit", Command.STOP, 1);

String currentlyAt;

public GraphicsPremitives() {

super();

}

public void startApp() throws MIDletStateChangeException {

display = Display.getDisplay(this);

menu = new List("LAB 3", Choice.IMPLICIT);

menu.append("1. Line", null);

menu.append("2. Rectangle", null);

menu.append("3. Rounded Rectangle", null);

menu.append("4. Circle", null);

menu.append("5. Ellipse", null);

menu.append("6. Arc", null);

menu.append("7. Triangle", null);

menu.addCommand(exitCommand);

menu.setCommandListener(this);

ticker = new Ticker("18124004 : Lab 3 - Graphics Premitives");

menu.setTicker(ticker);

mainMenu();

}

void mainMenu() {

display.setCurrent(menu);

currentlyAt = "main";

}

public void pauseApp() {

display = null;

form = null;

ticker = null;

menu = null;

currentlyAt = null;

}

public void destroyApp(boolean unconditional) {

notifyDestroyed();

}

public void commandAction(Command cm, Displayable ds) {

String label = cm.getLabel();

if (label.equals("Exit")) {

destroyApp(true);

} else if (label.equals("Back")) {

mainMenu();

} else {

List down = (List)display.getCurrent();

switch (down.getSelectedIndex()) {

case 0: drawLine(); break;

case 1: drawRect(); break;

case 2: drawRoundRect(); break;

case 3: drawCirc(); break;

case 4: drawOval(); break;

case 5: drawArc(); break;

case 6: drawTri(); break;

}

}

}

public void drawLine() {

graphicsCanvas c = new graphicsCanvas(0);

c.addCommand(backCommand);

c.setCommandListener(this);

display.setCurrent(c);

currentlyAt = "Line";

}

public void drawRect() {

graphicsCanvas c = new graphicsCanvas(1);

c.addCommand(backCommand);

c.setCommandListener(this);

display.setCurrent(c);

currentlyAt = "Rectangle";

}

public void drawRoundRect() {

graphicsCanvas c = new graphicsCanvas(2);

c.addCommand(backCommand);

c.setCommandListener(this);

display.setCurrent(c);

currentlyAt = "RoundedRect";

}

public void drawCirc() {

graphicsCanvas c = new graphicsCanvas(3);

c.addCommand(backCommand);

c.setCommandListener(this);

display.setCurrent(c);

currentlyAt = "Circle";

}

public void drawOval() {

graphicsCanvas c = new graphicsCanvas(4);

c.addCommand(backCommand);

c.setCommandListener(this);

display.setCurrent(c);

currentlyAt = "Oval";

}

public void drawArc() {

graphicsCanvas c = new graphicsCanvas(5);

c.addCommand(backCommand);

c.setCommandListener(this);

display.setCurrent(c);

currentlyAt = "Arc";

}

public void drawTri() {

graphicsCanvas c = new graphicsCanvas(6);

c.addCommand(backCommand);

c.setCommandListener(this);

display.setCurrent(c);

currentlyAt = "Triangle";

}

}

class graphicsCanvas extends Canvas {

int choice;

public graphicsCanvas (int i) {

super();

choice = i;

}

public void paint(Graphics g) {

g.setColor(0xffffff);

g.fillRect(0, 0, getWidth(), getHeight());

g.setColor(0x0000ff);

if (choice == 0) { //line

g.drawLine(50, 20, 100, 200);

} else if (choice == 1) { //rectangle

g.drawRect(20, 20, 100, 120);

} else if (choice == 2) { //rounded rectangle

g.drawRoundRect(20, 20, 100, 120, 20, 20);

} else if (choice == 3) { //circle

g.drawArc(50, 50, 50, 50, 0, 360);

} else if (choice == 4) { //ellipse

g.drawArc(50, 50, 100, 50, 0, 360);

} else if (choice == 5) { //arc

g.drawArc(50, 50, 100, 100, 30, 200);

} else if (choice == 6) { //triangle

g.fillTriangle(20, 20, 160, 40, 120, 20);

} else {

g.setFont(Font.getFont(Font.FACE\_SYSTEM, Font.STYLE\_BOLD, Font.SIZE\_MEDIUM));

g.drawString("ERROR: UNIDENTIFIED SHAPE CHOICE", 0, 30, g.LEFT | g.TOP);

}

}

}

**Output :**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |