WEEK 2 - DEMO CODE

Currency Converter Demo

```
* @author George Nguyen (George.Nguyen@rmit.edu.vn)

public class CurrencyConverter extends MIDlet implements CommandListener, ItemStateListener {

    static final int USD_Rate = 20995;
    static final int GBP_Rate = 35028;
    boolean USD = true;
    private Form myForm;
    private StringItem label;
    private TextField result, txtVND;
    private ChoiceGroup currency;
    private Command convert, exit;

    public CurrencyConverter() {
        //Initialize elements
        myForm = new Form("Currency Converter");

        label = new StringItem(null, "VND Converter ");
        txtVND = new TextField("Input VND", null, 15, TextField.NUMERIC);
        result = new TextField("Result", null, 15, TextField.UNEDITABLE);

        currency = new ChoiceGroup("To Currency", ChoiceGroup.EXCLUSIVE);
        currency.append("USD", null);
        currency.append("GBP", null);

        convert = new Command("Convert", Command.OK, 1);
        exit = new Command("Exit", Command.EXIT, 0);

        //Add to Form
        myForm.append(label);
        myForm.append(txtVND);
        myForm.append(currency);
        myForm.append(result);

        myForm.addCommand(exit);
        myForm.addCommand(convert);

        myForm.setCommandListener(this);
        myForm.setItemStateListener(this);
    }

    public void startApp() {
        Display mDisplay = Display.getDisplay(this);
        mDisplay.setCurrent(myForm);
    }

    public void pauseApp() { }

    public void destroyApp(boolean unconditional) { }

    public void commandAction(Command cmnd, Displayable dsplbl) {
        if (cmnd == convert) {
            if (currency.getSelectedIndex() == 1) {
                USD = false;
            }
            convert();
        }
        if (cmnd == exit) {
            notifyDestroyed();
        }
    }

    public void convert() {
        int rate;
        if (USD) {
            rate = USD_Rate;
        } else {
            rate = GBP_Rate;
        }
        result.setString(String.valueOf(rate * Integer.parseInt(txtVND.getString())));
    }

    public void itemStateChanged(Item item) {
        if (item == currency) {
            USD = !USD;
        }
        convert();
    }
}
```

Canvas Demo

* @author George Nguyen (George.Nguyen@rmit.edu.vn)

```
public class CanvasDemo extends MIDlet {

    public void startApp() {
        MyCanvas cv = new MyCanvas();
        Display mDisplay = Display.getDisplay(this);
        mDisplay.setCurrent(cv);
    }

    public void pauseApp() { }

    public void destroyApp(boolean unconditional) { }
}

class MyCanvas extends Canvas {

    private String keyName = "NULL";
    private int currentKey;

    public MyCanvas() {
        super.setFullScreenMode(true);
    }

    protected void paint(Graphics g) {
        g.fillRect(0, 0, getWidth(), getHeight());
        g.setColor(0xffffff);
        g.drawString(keyName, 0, 0, Graphics.TOP | Graphics.LEFT);

    }

    protected void keyPressed(int keyCode) {
        keyName=getKey(keyCode);
        repaint();
    }

    public String getKey(int keyCode) {
        if (getGameAction(keyCode) != 0) {
            currentKey = getGameAction(keyCode);
            switch (currentKey) {
                case UP:
                    keyName = "UP";
                    break;
                case DOWN:
                    keyName = "DOWN";
                    break;
                case LEFT:
                    keyName = "LEFT";
                    break;
                case RIGHT:
                    keyName = "RIGHT";
                    break;
                case FIRE:
                    keyName = "FIRE";
                    break;
            }
            return "Game Key:"+ keyName;
        }
        return "Normal :"+getKeyName(keyCode);
    }
}
```

Clipping Demo

```
* @author George Nguyen (George.Nguyen@rmit.edu.vn)

public class ClippingDemo extends MIDlet {

    public void startApp() {
        Display mDisplay = Display.getDisplay(this);
        ClippedCanvas cv = new ClippedCanvas();

        mDisplay.setCurrent(cv);
        cv.run();
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }
}

class ClippedCanvas extends Canvas implements Runnable {

    private int width, height, x, y, z, t, u;

    public ClippedCanvas() {
        width = getWidth();
        height = getHeight();
        setFullScreenMode(true);
        x = 255;
        y = 255;
        z = 255;

        t = width;
        u = height;
    }

    protected void paint(Graphics g) {
        g.clipRect(0, 0, t, u);
        g.setColor(x, y, z);
        g.fillRect(0, 0, width, height);

        t -= 5;
        u -= 5;
        y -= 10;
        z -= 10;
    }

    public void run() {
        while (y > 10 && z > 10) {
            try {
                Thread.sleep(200);
            } catch (InterruptedException ex) {
                ex.printStackTrace();
            }

            repaint();
        }
    }
}
```