

```
<!-------#Matrix.java-----!>
```

```
package com.example.matrix;
```

```
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.util.AttributeSet;
import android.view.View;
```

```
import java.util.Random;
```

```
public class Matrix extends View {
```

```
    int width = 1000000; //default initial width
    int height = 100; //default initial height
    Canvas canvas = null; //default canvas
    Bitmap canvasBitmap; //Bitmap used to create the canvas
    int fontSize = 15; //font size of the text which will fall
    int columnSize = width/fontSize; //column size ; no of digit required to fill the screen
    int parentWidth;
    String text = "MATRIXRAIN"; // Text which need to be drawn
    char[] textChar = text.toCharArray(); // split the character of the text
    int textLength = textChar.length; //length of the length text
    Random rand = new Random(); //random generater
```

```
    int[] textPosition; // contain the position which will help to draw the text
```

```
    public Matrix(Context context) {
        super(context);
```

```
    }
```

```
    public Matrix(Context context, AttributeSet attrs) {
        super(context, attrs);
    }
```

```
    void drawText()
```

```
    {
```

```
        //Set up the paint
```

```
        Paint paint = new Paint();
```

```
        paint.setStyle(Paint.Style.FILL);
```

```
        paint.setColor(Color.GREEN);
```

```
        paint.setTextSize(15);
```

```
        //loop and paint
```

```
        for(int i =0 ;i<textPosition.length;i++)
```

```
        {
```

```
            // draw the text at the random position
```

```
            canvas.drawText(""+textChar[rand.nextInt(textLength)+0],i*fontSize,textPosition[i]*fontSize,paint);
```

```
            // check if text has reached bottom or not
```

```
            if(textPosition[i]*fontSize > height && Math.random() > 0.975)
```

```
                textPosition[i] = 0; // change text position to zero when 0 when text is at the bottom
```

```
            textPosition[i]++; //increment the position array
```

```
        }
```

```
    }
```

```
    public void canvasDraw()
```

```
    {
```

```
        //set the paint for the canvas
```

```
        Paint paint = new Paint();
```

```
        paint.setColor(Color.BLACK);
```

```
        paint.setAlpha(5);
```

```
        paint.setStyle(Paint.Style.FILL);
```

```
        canvas.drawRect(0, 0, width, height, paint); //draw rect to clear the canvas
```

```
        drawText(); // draw the canvas
```

```
    }
```

```
    //function responsonible for draw calls
```

```
    @Override
```

```
    protected void onDraw(Canvas canvas) {
```

```
        super.onDraw(canvas);
```

```
        Paint paint = new Paint();
```

```
        paint.setColor(Color.BLACK);
```

```
        canvas.drawBitmap(canvasBitmap,0,0,paint); //draw the bitmap to canvas
```

```
        canvasDraw(); // call the draw command
```

```
        //Redraw the canvas
```

```
        invalidate();
```

```
    }
```

```
    //set the height and width of the canvas according to the screen size
```

```
    @Override
```

```
    protected void onSizeChanged(int w, int h, int oldw, int oldh) {
```

```
        width= w;
```

```
        height = h;
```

```
        super.onSizeChanged(w, h, oldw, oldh);
```

```
        //create a Bitmap
```

```
        canvasBitmap = Bitmap.createBitmap(width, height, Bitmap.Config.ARGB_8888);
```

```
        canvas = new Canvas(canvasBitmap); //set the canvas
```

```
        // init paint with black rectangle
```

```
        Paint paint = new Paint();
```

```

        paint.setColor(Color.BLACK);
        paint.setAlpha(255); //set the alpha
        paint.setStyle(Paint.Style.FILL);
        canvas.drawRect(0, 0, width, height, paint);

        columnSize = width/fontSize;
        //initialise the textposiotn to zero
        textPosition = new int[columnSize+1]; //add one more drop
        for(int x = 0; x < columnSize; x++)
            textPosition[x] = 1;
    }
}

<!-------#MainActivity-----!>

Use MatrixView In MainActivity

package com.example.matrix;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Without Any Xml Id By Only Java Source
        Matrix matrixView = new Matrix(this);
        layout_name.addView(matrixView); // Replace With Actual Layout Id | Layout Names($$layout_name$$)-----!
    }
}

```