

Práctica de Laboratorio 2

Christofer Fabián Chávez Carazas

Universidad Nacional de San Agustín de Arequipa

Escuela Profesional de Ciencia de la Computación

Computación Gráfica

27 de octubre de 2017

1. Compile y ejecute el siguiente código

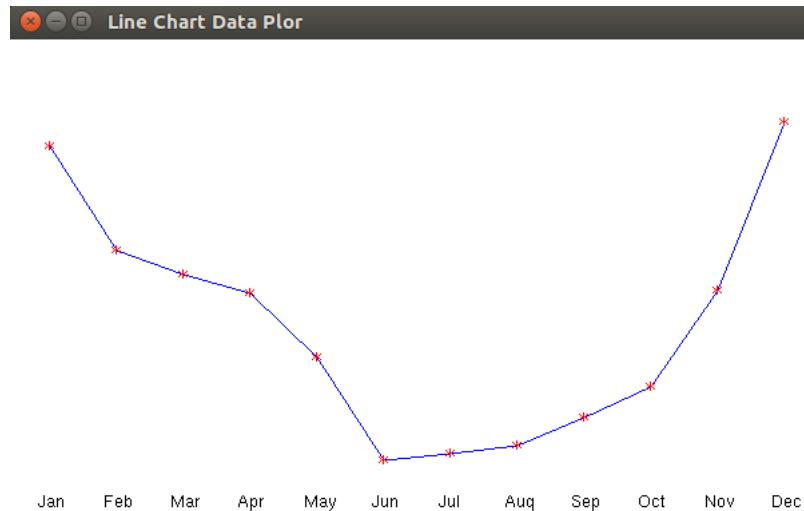


Figura 1: Resultados

2. Explique qué función cumple cada línea de código

```
glBegin(GL_LINE_STRIP);  
    for(k = 0; k < 12; k++){  
        glVertex2i(x + k*50, dataValue[k]);  
    }  
glEnd();
```

En esta parte del código se dibujan las líneas del gráfico. El punto x es el punto de referencia dado al inicio más un desplazamiento. El punto y es el valor del dato actual.

```
for(k = 0; k < 12; k++){  
    glRasterPos2i(xRaster + k*50, dataValue[k] - 4);  
    glutBitmapCharacter ( GLUT_BITMAP_9_BY_15, '*' );  
}
```

En esta parte del código se dibujan los asteriscos que indican los valores de la función. Igual que el anterior, el punto x es el punto de referencia dado antes más un desplazamiento, El punto x es el valor del dato actual más un desplazamiento, para que quede bien alineado con la función.

```
for(month = 0; month < 12; month++){
    glRasterPos2i(xRaster,yRaster);
    for(k = 3*month; k < 3*month + 3; k++){
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_12,label[k]);
    }
    xRaster += 50;
}
```

En esta parte del código se escribe la leyenda de cada valor. Cada leyenda cuenta con 3 caracteres que están guardados en una lista.

3. Cambie la función *lineGraph* por el de *barChart*, luego compile el programa

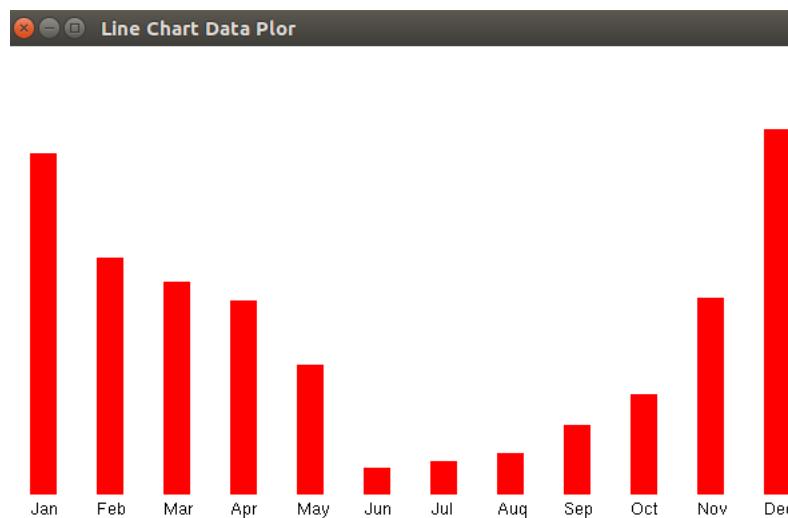


Figura 2: Resultados

4. Construya un programa donde se muestre, en líneas y en barras, las ventanas de un producto en los últimos seis meses del año. EL gráfico debe ser parecido a la siguiente figura. Debe ser posible agregar una línea de tendencia.

El código es una combinación de los dos anteriores.

```
#include <GL/glut.h>

GLsizei winWidth = 600, winHeight = 500;
GLint xRaster = 25, yRaster = 150;
int months = 6;
GLubyte label[18] = {'J','u','l','i','a','u','g','S','e','p',
'O','c','t','N','o','v','D','e','c'};
GLint dataValue[6] = {180,190,220,250,330,450};

void init(void){
    glClearColor(1.0,1.0,1.0,1.0);
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(0.0, 600.0, 0.0, 500.0);
}
```

```

void bartChart(void){
    GLint month, k;

    glClear(GL_COLOR_BUFFER_BIT);

    glColor3f(1.0,0.0,0.0);
    for(k = 0; k < months; k++){
        glRecti(20 + k * 50, 165, 40 + k * 50, dataValue[k]);
    }
    glColor3f(0.0, 0.0, 0.0);
    xRaster = 20;
    for(month = 0; month < months; month++){
        glRasterPos2i(xRaster,yRaster);
        for(k = 3*month; k < 3*month + 3; k++){
            glutBitmapCharacter(GLUT_BITMAP_HELVETICA_12,label[k]);
        }
        xRaster += 50;
    }

    GLint x = 30;
    glColor3f(0.0, 0.0,1.0);
    glBegin(GL_LINE_STRIP);
        for(k = 0; k < months; k++){
            glVertex2i(x + k*50, dataValue[k]);
        }
    glEnd();
    xRaster = 25;
    for(k = 0; k < months; k++){
        glRasterPos2i(xRaster + k*50,dataValue[k] - 4);
        glutBitmapCharacter (GLUT_BITMAP_9_BY_15, '*');
    }
    glFlush();
}

int main(int argc, char **argv){
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition(100, 100);
    glutInitWindowSize(winWidth, winHeight);
    glutCreateWindow("Line Chart Data Plor");

    init();
    glutDisplayFunc(bartChart);

    glutMainLoop();
    return 0;
}

```

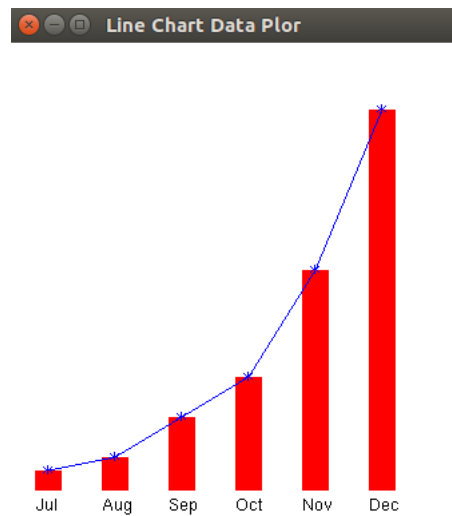


Figura 3: Resultados

5. El siguiente programa construye una figura circular, utilizando la rutina del punto medio para generar el círculo.

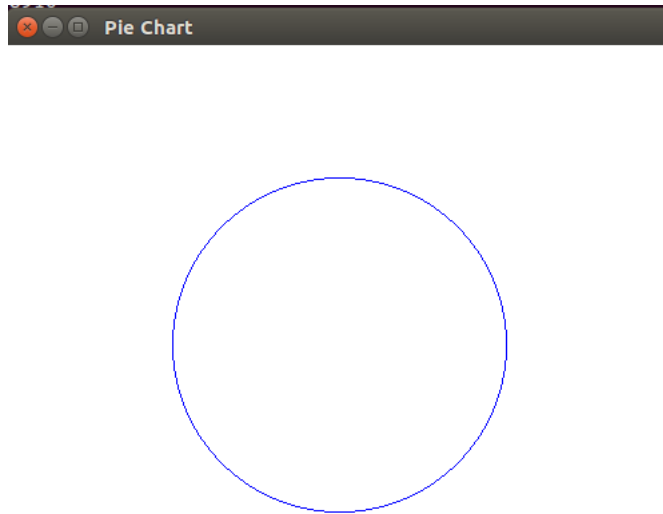


Figura 4: Resultados

6. Las gráficas de sectores circulares se utilizan para mostrar la contribución porcentual de una serie de partes individuales a un todo. Modifique el programa anterior para construir una gráfica de sectores circulares, el resultado debe ser parecido a la siguiente figura.

La única función modificada del código anterior es *pieChart*. Se van acumulando los valores para sacar el siguiente ángulo. Con ese ángulo se saca el seno y el coseno para hallar el punto del círculo donde terminará la línea.

```
void pieChart(void){
    srcPt circCtr, piePt;
    GLint radius = winWidth / 4;
    GLdouble sliceAngle, previousSliceAngle = 0.0;
    GLint k, nSlices = 12;
    GLfloat dataValues[12] = {10.0,7.0,13.0,5.0,13.0,14.0,3.0,16.0,5.0,3.0,17.0,8.0};
    GLfloat total = 0.0;
    GLfloat dataSum = 0.0;
    srcPt secondPoint;
    circCtr.x = winWidth / 2;
    circCtr.y = winHeight / 2;
    for(int i = 0; i < nSlices; i++){
        total += dataValues[i];
    }
    for(int i = 0; i < nSlices; i++){
        dataSum += dataValues[i];
        sliceAngle = getAngle(dataSum, total);
        secondPoint.x = circCtr.x + cos(sliceAngle * PI/180) * radius;
        secondPoint.y = circCtr.y + sin(sliceAngle * PI/180) * radius;
        drawLine(circCtr, secondPoint);
    }
    circleMidPoint(circCtr, radius);
}
```

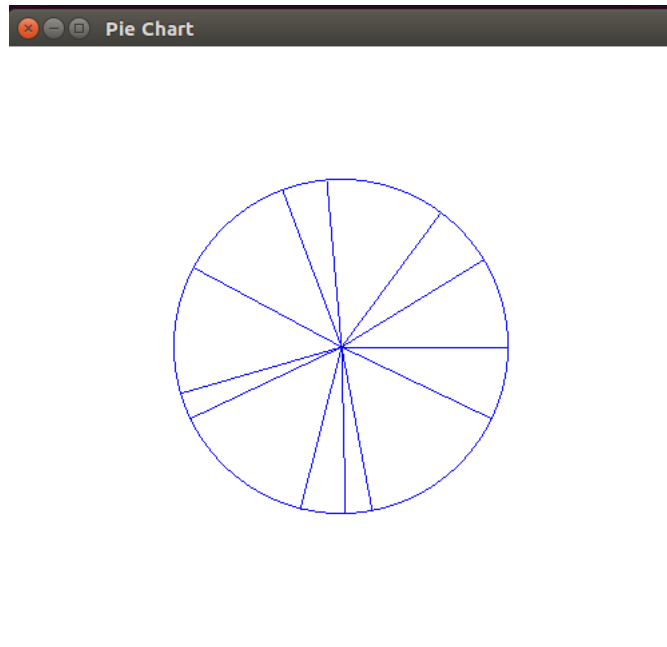


Figura 5: Resultados