B.Sc. Computer Science
3.2

ICS 2311 Computer Graphics

SCT211-0848/2018 - JANY MUONG
SCT211-0079/2022 - JORAM KIREKI
SCT211-0003/2022 - JOSPHAT WAWERU
SCT211-0535/2022 - AKECH ATEM

SUBMITTED: **April 9th, 2025**

G7

ICS 2311: COMPUTER GRAPHICS – OPENGL GROUPWORK

CAT – WRITE UP:

mail to: jkuatnotes7@gmail.com
subject: GROUP 7: ICS 2311

BACKGROUND CONTEXT:

Group work using OpenGL/GLUT/GLEW.
Working on question 7 from the psets

INFORMATION:

This write-up contains our solutions for questions 7, captured in screenshots and the code for each part (**a** and **b**) of the question are at the end of the screenshots. They can be used to reconstruct the OpenGL/C code as well as the Python equivalents if opening the zip files fails. The same code is also in the zip archive.

GitHub Link:
Here is the directory that hosts our solution code(**C** + **Python**): GitHub-Repo-Link

Presentation/Slide Link: canva-g7-opengl-presentation

Question 7:
A survey was carried out in Gachororo about youth preference on fruits. 150 youth were interviewed about their fruits of preference as follows:

Fruit:	Ovacado	Orange	Banana	Kiwifruit	Mangos	Grapes
People:	36	41	19	28	30	16

QUESTION 7 – SOLUTION

We are being tested for these core concepts:

1. **OpenGL:** create basic 2D visualizations using OpenGL, **coordinate systems** and transformations, output primitives (lines, rectangles, text)
2. **Miscellaneous**: creating accurate bar charts from raw data, proper **scaling** of data to fit display window, effective labeling of chart elements
3. **Solution**:

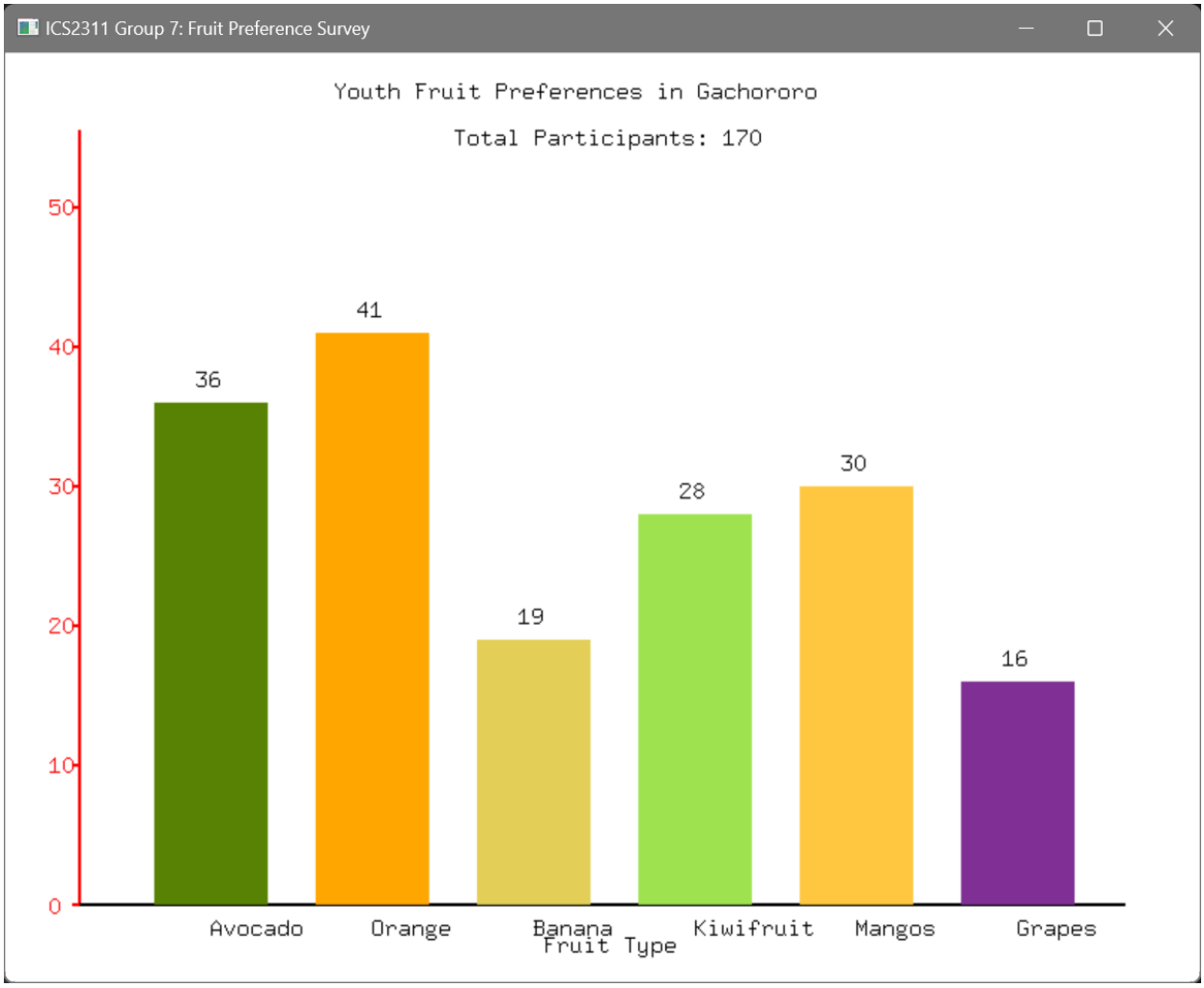
- **Part (a)**: scaling of data points to fill display area, color mapping (bars colored to match actual fruits), axis labeling with specific color requirements (x-axis black, y-axis red)
- **Part (b)**: coordinate system manipulation, translate/offset graphical elements, maintaining all chart functionality while changing origin point.

a) First Part: Response/OpenGL C:

```
merou@HP MINGW64 ~/cs32/ics2311/gachororo_fruit
$ gcc a_fruit_bars.c -o a_fruit_bars  -lglew32 -lfreeglut -
lopengl32 -lglu32

merou@HP MINGW64 ~/cs32/ics2311/gachororo_fruit
$ ./a_fruit_bars
```

OpenGL Display

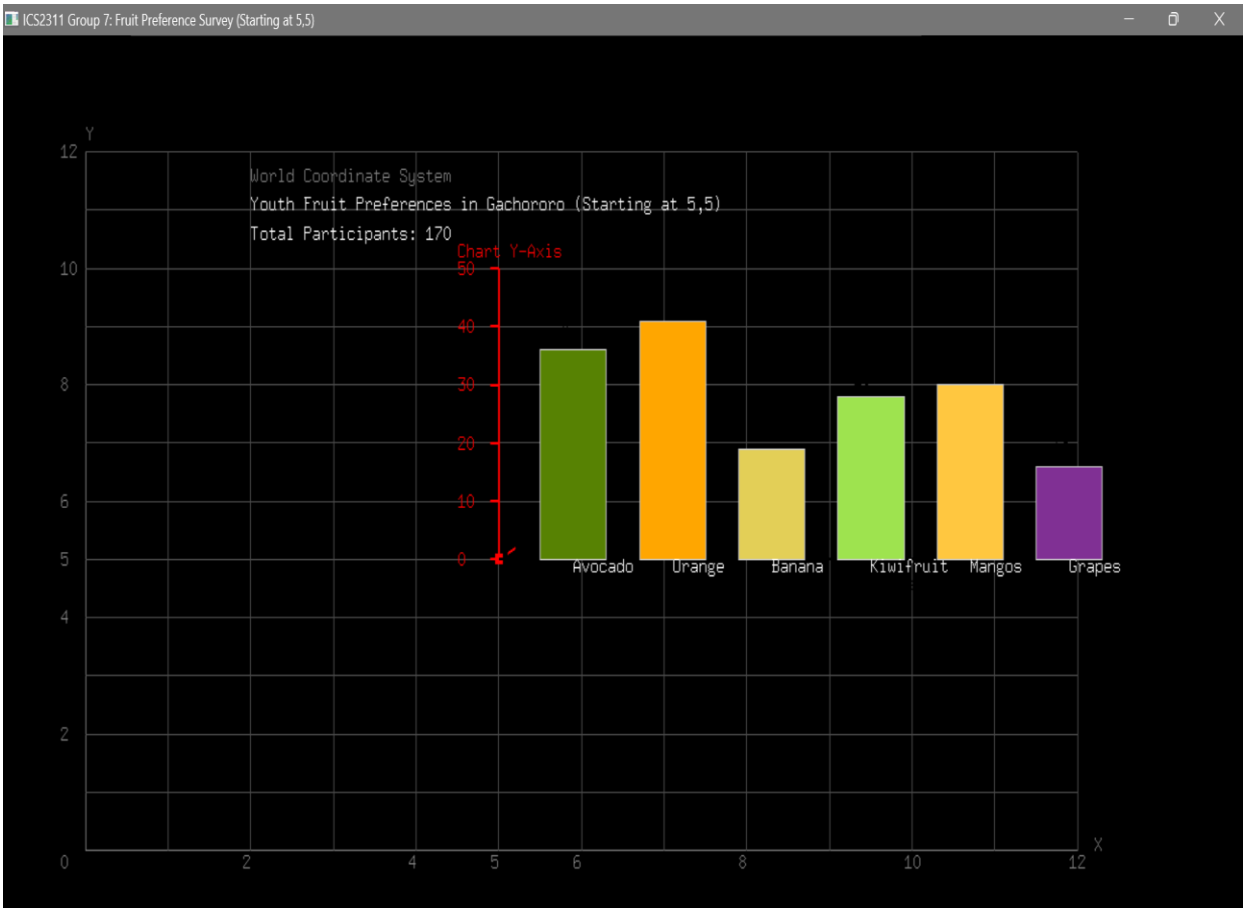


b) SECOND PART: Response/OpenGL C:

This is meant to show transformations ie translation that starts at origin (5, 5). We have included a second coordinate systems for comparison.

```
merou@HP MINGW64 ~/cs32/ics2311/gachororo_fruit
$ gcc b_fruit_bars.c -o b_fruit_bars  -lglew32 -lfreeglut -lopengl32 -
lglu32

merou@HP MINGW64 ~/cs32/ics2311/gachororo_fruit
$ ./b_fruit_bars
```



EQUIVALENT OPENGL SOLUTION IN PYTHON

The equivalent Python Libraries in Python can be installed this way in a terminal using Pip Package Manager. The libraries/modules to install are: **`PyOpenGL PyOpenGL_accelerate PyGLM`**

```
pip install PyOpenGL PyOpenGL_accelerate PyGLM
```

FIRST PART:

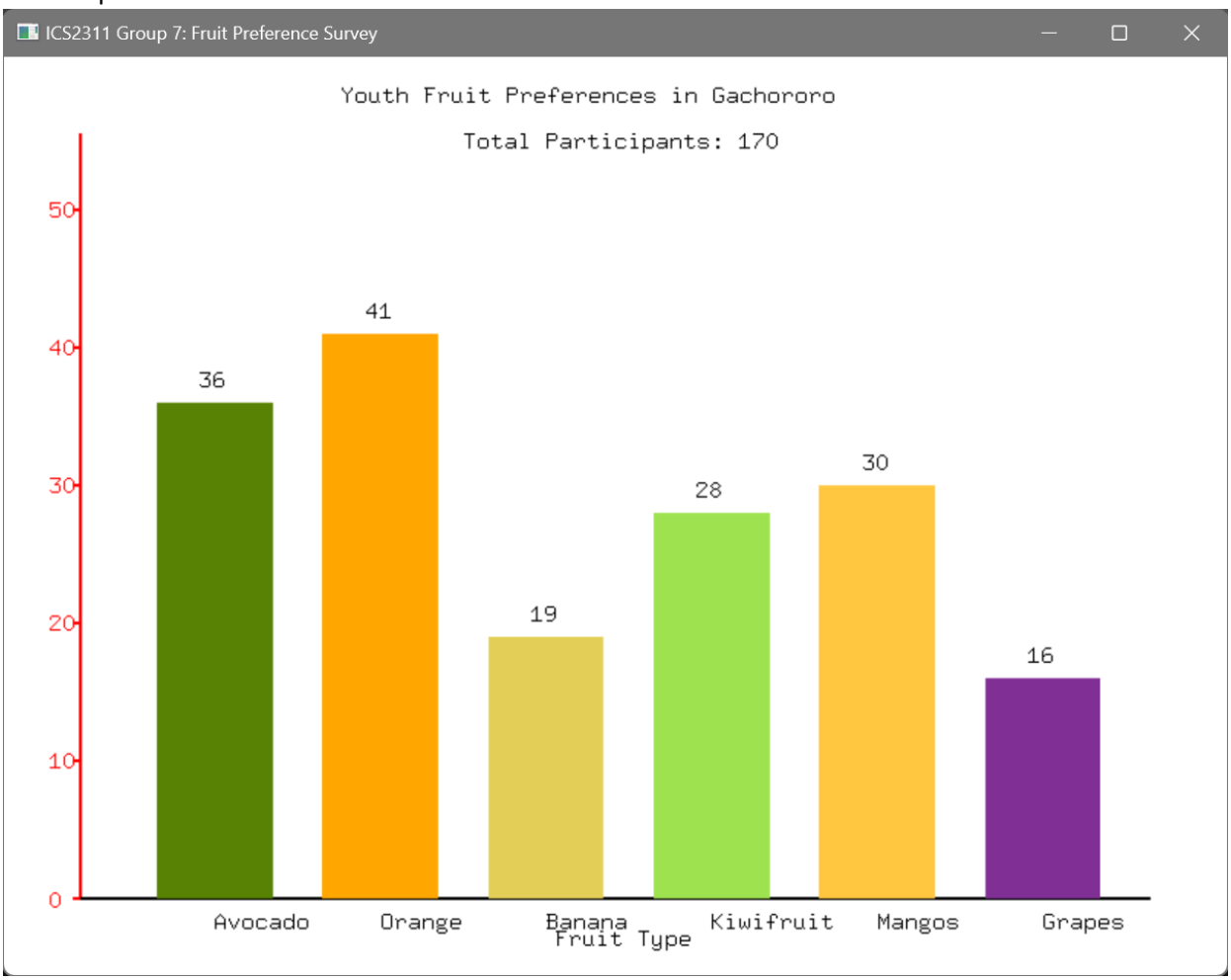
merou@HP MINGW64 ~/cs_jkuat/cs_notes/cs32mu-o/ICS2311-Computer_Graphics/ics2311-cg-CAT/opengl_groupwork

$ python a_fruit_bars.py

freeglut (a_fruit_bars.py): fgPlatformInitialize: CreateDC failed, Screen size info may be incorrect

This is quite likely caused by a bad '-display' parameter

The Output is this:



G7

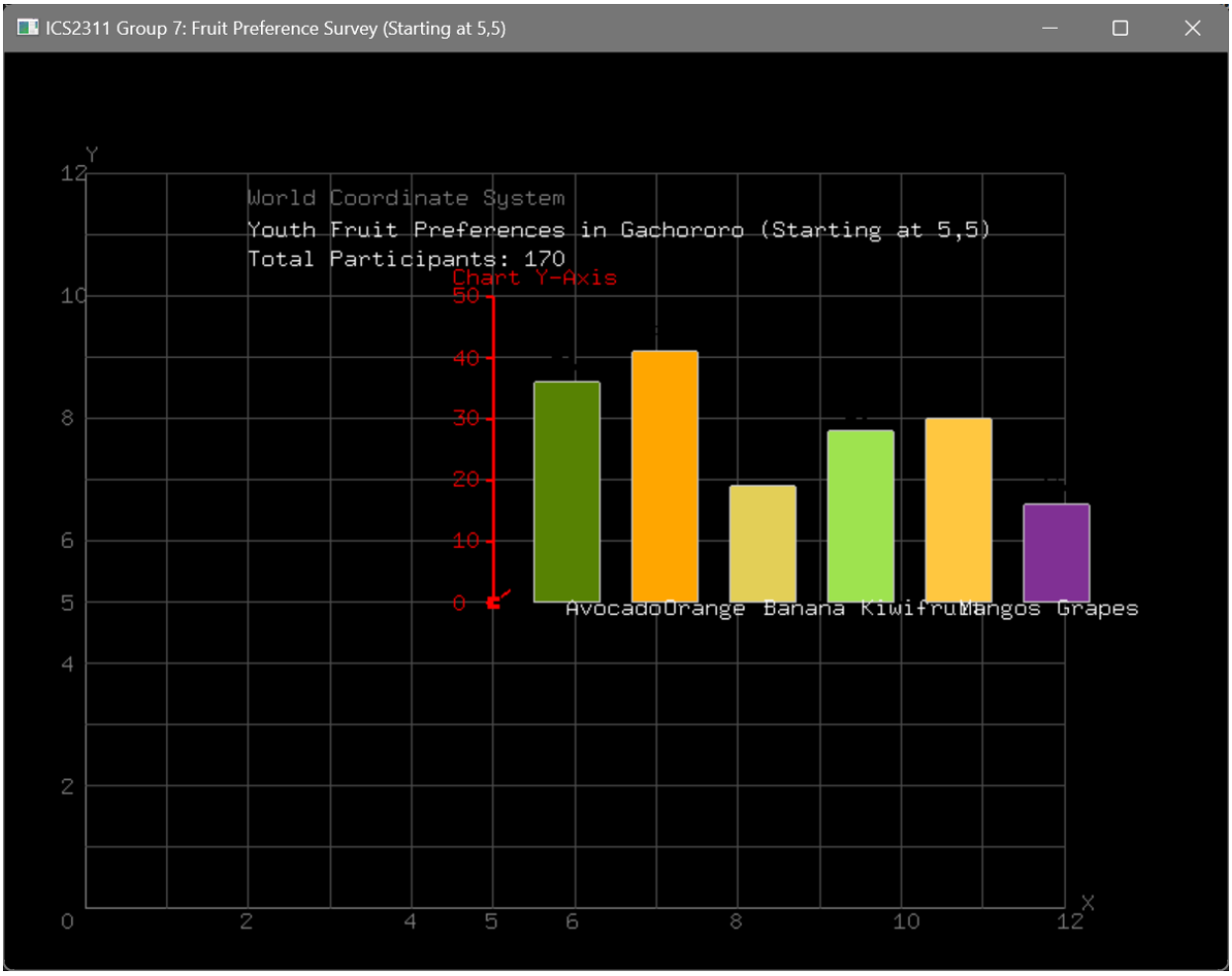
SECOND PART:

merou@HP MINGW64 ~/cs_jkuat/cs_notes/cs32mu-o/ICS2311-Computer_Graphics/ics2311-cg-CAT/opengl_groupwork

$ python b_fruit_bars.py

freeglut (b_fruit_bars.py): fgPlatformInitialize: CreateDC failed, Screen size info may be incorrect

This is quite likely caused by a bad '-display' parameter



The

G7

OpenGL/ C Code:

Part A:

```
// ICS2311 Group 7 - fruit preference bar chart
// a_fruit_bars.c - program to display a bar chart of youth fruit preferences in gachororo

// MEMBERS:
/*
* SCT211-0848/2018 - Jany Muong
* SCT211-0079/2022 - Joram Kireki
* SCT211-0003/2022 - Josphat Waweru Thumi
* SCT211-0535/2022 - Akech Atem
*/

#include <GL/glew.h>
#include <GL/freeglut.h>
#include <stdio.h>
#include <math.h>
#include <string.h>

// structure to hold fruit data
typedef struct {
    char name[20];
    int count;
    float color[3]; // RGB values
} FruitData;

// global array of fruit data
FruitData fruits[] = {
    {"Avocado", 36, {0.34, 0.51, 0.01}}, // dark green
    {"Orange", 41, {1.00, 0.65, 0.00}}, // orange color
    {"Banana", 19, {0.89, 0.81, 0.34}}, // yellow
    {"Kiwifruit", 28, {0.62, 0.89, 0.31}}, // kiwi green
    {"Mangos", 30, {1.00, 0.78, 0.25}}, // mango orange
    {"Grapes", 16, {0.50, 0.19, 0.58}} // purple
};
const int numFruits = 6;
int windowWidth = 800, windowHeight = 600;

// reshape function to handle window resizing
void reshape(int width, int height) {
    windowWidth = width;
    windowHeight = height;
```

```
    glViewport(0, 0, width, height);
    glutPostRedisplay(); // trigger redraw when window is resized
}

// world coordinate reference system to use
void drawCoordinateSystem() {
    // draw y-axis (red)
    glColor3f(1.0, 0.0, 0.0);
    glLineWidth(2.0);
    glBegin(GL_LINES);
        glVertex2f(50.0, 50.0);
        glVertex2f(50.0, windowHeight - 50.0);
    glEnd();

    // draw x-axis (black)
    glColor3f(0.0, 0.0, 0.0);
    glBegin(GL_LINES);
        glVertex2f(50.0, 50.0);
        glVertex2f(windowWidth - 50.0, 50.0);
    glEnd();
}

// function to draw text in the scene
void drawText(const char* text, float x, float y, float r, float g, float b) {
    glColor3f(r, g, b);
    glRasterPos2f(x, y);
    for(const char* c = text; *c != '\0'; c++) {
        glutBitmapCharacter(GLUT_BITMAP_9_BY_15, *c);
    }
}

void display() {
    glClear(GL_COLOR_BUFFER_BIT);

    // set up projection
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0, windowWidth, 0, windowHeight);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    // draw coordinate system (just the axes)
    drawCoordinateSystem();
```

G7

```
    // draw title
    drawText("Youth Fruit Preferences in Gachororo", windowWidth/2 - 180,
windowHeight - 30, 0.0, 0.0, 0.0);
    drawText("Total Participants: 170", windowWidth/2 - 100, windowHeight - 60,
0.0, 0.0, 0.0);

    // draw axes labels
    drawText("Fruit Type", windowWidth/2 - 40, 20, 0.0, 0.0, 0.0); // black x-
axis
    // drawText("Number of People", 10, windowHeight/2 - 80, 1.0, 0.0, 0.0); //
red y-axis

    // calculate dimensions
    float barWidth = (windowWidth - 150) / numFruits * 0.7;
    float barSpacing = (windowWidth - 150) / numFruits * 0.3;
    float maxBarHeight = windowHeight - 150;
    float startX = 100.0;

    // draw bars
    for(int i = 0; i < numFruits; i++) {
        float barHeight = (float)fruits[i].count / 50 * maxBarHeight;

        // draw bar with fruit color
        glColor3fv(fruits[i].color);
        glBegin(GL_QUADS);
            glVertex2f(startX, 50.0);
            glVertex2f(startX + barWidth, 50.0);
            glVertex2f(startX + barWidth, 50.0 + barHeight);
            glVertex2f(startX, 50.0 + barHeight);
        glEnd();

        // draw value
        char label[10];
        sprintf(label, "%d", fruits[i].count);
        drawText(label, startX + barWidth/2 - 10, 60 + barHeight, 0.0, 0.0,
0.0);

        // draw fruit name
        glPushMatrix();
        glTranslatef(startX + barWidth/2, 30, 0);
        glRotatef(45, 0, 0, 1);
        drawText(fruits[i].name, 0, 0, 0.0, 0.0, 0.0);
        glPopMatrix();
```

G7

```
        startX += barWidth + barSpacing;
    }

    // draw the fruit name below the bar in BLACK
    for(int i = 0; i <= 50; i += 10) {
        float y = 50 + (float)i / 50 * maxBarHeight;
        glColor3f(1.0, 0.0, 0.0);
        glBegin(GL_LINES);
            glVertex2f(45, y);
            glVertex2f(50, y);
        glEnd();
        char str[10];
        sprintf(str, "%d", i);
        drawText(str, 30, y - 5, 1.0, 0.0, 0.0);
    }

    glFlush();
}

void init() {
    glClearColor(1.0, 1.0, 1.0, 1.0);
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(windowWidth, windowHeight);
    glutCreateWindow("ICS2311 Group 7: Fruit Preference Survey");

    glutReshapeFunc(reshape); // for proper scaling to all windo sizes
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```

PART B:

```
// ICS2311 Group 7 - fruit preference bar chart starting at point (5,5)
// b_fruit_bars.c - program to display a bar chart of youth fruit preferences in
gachororo

// MEMBERS:
```

G7

```c
/*
* SCT211-0848/2018 - Jany Muong
* SCT211-0079/2022 - Joram Kireki
* SCT211-0003/2022 - Josphat Waweru Thumi
* SCT211-0535/2022 - Akech Atem
*/

#include <GL/glew.h>
#include <GL/freeglut.h>
#include <stdio.h>
#include <math.h>
#include <string.h>

// structure to hold fruit data
typedef struct {
    char name[20];
    int count;
    float color[3]; // RGB values
} FruitData;

// global array of fruit data - using the same colors as part A
FruitData fruits[] = {
    {"Avocado", 36, {0.34, 0.51, 0.01}}, // dark green
    {"Orange", 41, {1.00, 0.65, 0.00}}, // orange color
    {"Banana", 19, {0.89, 0.81, 0.34}}, // yellow
    {"Kiwifruit", 28, {0.62, 0.89, 0.31}}, // kiwi green
    {"Mangos", 30, {1.00, 0.78, 0.25}}, // mango orange
    {"Grapes", 16, {0.50, 0.19, 0.58}} // purple
};

const int numFruits = 6;
int windowWidth = 800, windowHeight = 600;

// offset for the chart (starting point at (5,5))
const float xOffset = 5.0;
const float yOffset = 5.0;

// reshape function to handle window resizing
void reshape(int width, int height) {
    windowWidth = width;
    windowHeight = height;
    glViewport(0, 0, width, height);
    glutPostRedisplay(); // trigger redraw when window is resized
}
```

G7

```
// function to draw text in the scene
void drawText(const char* text, float x, float y, float r, float g, float b) {
    glColor3f(r, g, b);
    glRasterPos2f(x, y);
    for(const char* c = text; *c != '\0'; c++) {
        glutBitmapCharacter(GLUT_BITMAP_9_BY_15, *c);
    }
}

// display coordinate system to show where (5,5) is located
void drawBaseCoordinateSystem() {
    // draw main world axes
    glColor3f(0.5, 0.5, 0.5);  // gray color for base axes
    glLineWidth(1.0);
    glBegin(GL_LINES);
        glVertex2f(0.0, 0.0); glVertex2f(12.0, 0.0); // x-axis
        glVertex2f(0.0, 0.0); glVertex2f(0.0, 12.0); // y-axis
    glEnd();

    // label the axes of the base coordinate system
    drawText("0", -0.3, -0.3, 0.5, 0.5, 0.5);  // Origin
    drawText("X", 12.2, 0.0, 0.5, 0.5, 0.5);   // X-axis
    drawText("Y", 0.0, 12.2, 0.5, 0.5, 0.5);   // Y-axis
    drawText("World Coordinate System", 2.0, 11.5, 0.5, 0.5, 0.5); // title

    // grid lines (lighter)
    glColor3f(0.3, 0.3, 0.3);
    glLineWidth(0.5);
    glBegin(GL_LINES);
    for (int i = 1; i <= 12; i++) {
        // only draw grid lines within our view
        if (i <= 12) {
            glVertex2f(i, 0.0); glVertex2f(i, 12.0); // vertical
            glVertex2f(0.0, i); glVertex2f(12.0, i); // horizontal
        }
    }
    glEnd();

    // label key grid points
    for (int i = 1; i <= 12; i += 1) {
        if (i % 2 == 0 || i == 5) { // only label even numbers and 5
            char num[5];
            sprintf(num, "%d", i);
```

```
            drawText(num, i - 0.1, -0.3, 0.5, 0.5, 0.5);  // x-axis labels
            drawText(num, -0.3, i - 0.1, 0.5, 0.5, 0.5);  // y-axis labels
        }
    }

    // highlight the starting point (5,5) with a red dot
    glColor3f(1.0, 0.0, 0.0);
    glPointSize(8.0);
    glBegin(GL_POINTS);
        glVertex2f(xOffset, yOffset);
    glEnd();

    // label for (5,5) point with arrow
    // drawText("(5,5)", xOffset + 0.2, yOffset + 0.2, 1.0, 0.0, 0.0);

    // draw arrow pointing to (5,5)
    glColor3f(1.0, 0.0, 0.0);
    glLineWidth(1.5);
    glBegin(GL_LINES);
        glVertex2f(xOffset + 0.1, yOffset + 0.1);
        glVertex2f(xOffset + 0.2, yOffset + 0.2);
    glEnd();
}

// draw the actual chart axes starting at (5,5)
void drawChartAxes() {
    // draw y-axis (red) - same as part A
    glColor3f(1.0, 0.0, 0.0);
    glLineWidth(2.0);
    glBegin(GL_LINES);
        glVertex2f(xOffset, yOffset);
        glVertex2f(xOffset, yOffset + 5.0);
    glEnd();

    // draw x-axis (black) - same as part A
    glColor3f(0.0, 0.0, 0.0);
    glBegin(GL_LINES);
        glVertex2f(xOffset, yOffset);
        glVertex2f(xOffset + 8.0, yOffset);
    glEnd();

    // Label the axes of the chart coordinate system
    drawText("Chart X-Axis", xOffset + 8.2, yOffset, 0.0, 0.0, 0.0);
    drawText("Chart Y-Axis", xOffset - 0.5, yOffset + 5.2, 1.0, 0.0, 0.0);
```

```
    // Add y-axis ticks and labels (red)
    glColor3f(1.0, 0.0, 0.0);
    for (int i = 0; i <= 50; i += 10) {
        float y = yOffset + ((float)i / 50.0) * 5.0;
        glBegin(GL_LINES);
            glVertex2f(xOffset - 0.1, y);
            glVertex2f(xOffset, y);
        glEnd();

        char str[10];
        sprintf(str, "%d", i);
        glRasterPos2f(xOffset - 0.5, y - 0.1);
        for (char* c = str; *c != '\0'; c++)
            glutBitmapCharacter(GLUT_BITMAP_9_BY_15, *c);
    }
}

void display() {
    glClear(GL_COLOR_BUFFER_BIT);

    // draw the base coordinate system to show where (5,5) is
    drawBaseCoordinateSystem();

    // draw the chart axes starting at (5,5)
    drawChartAxes();

    // draw title
    drawText("Youth Fruit Preferences in Gachororo (Starting at 5,5)", 2.0,
11.0, 1.0, 1.0, 1.0);
    drawText("Total Participants: 170", 2.0, 10.5, 1.0, 1.0, 1.0);

    // draw axes labels
    drawText("Fruit Type", xOffset + 4.0, yOffset - 0.5, 0.0, 0.0, 0.0); //
black x-axis

    // calculate dimensions for the bars
    float barWidth = 0.8;
    float barSpacing = 0.4;
    float maxBarHeight = 5.0;  // maximum height based on y-axis scale
    float startX = xOffset + 0.5;

    // draw bars - ensure they're all visible within our viewport
    for(int i = 0; i < numFruits; i++) {
```

G7

```
        // calculate bar height proportional to the max value (50)
        float barHeight = (float)fruits[i].count / 50.0 * maxBarHeight;

        // ensure minimum height for visibility
        if(barHeight < 0.1) barHeight = 0.1;

        // draw bar with fruit color
        glColor3fv(fruits[i].color);
        glBegin(GL_QUADS);
            glVertex2f(startX, yOffset);
            glVertex2f(startX + barWidth, yOffset);
            glVertex2f(startX + barWidth, yOffset + barHeight);
            glVertex2f(startX, yOffset + barHeight);
        glEnd();

        // draw outline around bar for better visibility
        glColor3f(0.8, 0.8, 0.8);
        glLineWidth(1.0);
        glBegin(GL_LINE_LOOP);
            glVertex2f(startX, yOffset);
            glVertex2f(startX + barWidth, yOffset);
            glVertex2f(startX + barWidth, yOffset + barHeight);
            glVertex2f(startX, yOffset + barHeight);
        glEnd();

        // draw value at top of bar
        char label[10];
        sprintf(label, "%d", fruits[i].count);
        drawText(label, startX + barWidth/2 - 0.2, yOffset + barHeight + 0.2,
0.0, 0.0, 0.0);

        // draw fruit name below bar (rotated like in part A)
        glPushMatrix();
        glTranslatef(startX + barWidth/2, yOffset - 0.2, 0);
        glRotatef(-45, 0, 0, 1);
        drawText(fruits[i].name, 0, 0, 1.0, 1.0, 1.0);
        glPopMatrix();

        startX += barWidth + barSpacing;
    }

    glFlush();
}
```

```
void init() {
    glClearColor(0.0, 0.0, 0.0, 1.0); // black background
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    // extended viewport to ensure all bars are visible
    // (so that transformation does not affect visibility of bars)
    gluOrtho2D(-1, 14, -1, 14);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(windowWidth, windowHeight);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("ICS2311 Group 7: Fruit Preference Survey (Starting at
5,5)");

    glewInit();
    init();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMainLoop();
    return (0);
}
```

G7

OpenGL/ Python Code:

Part A: a_fruit_bars.py

```
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *
import sys

fruits = [
    {"name": "Avocado", "count": 36, "color": [0.34, 0.51, 0.01]},
    {"name": "Orange", "count": 41, "color": [1.00, 0.65, 0.00]},
    {"name": "Banana", "count": 19, "color": [0.89, 0.81, 0.34]},
    {"name": "Kiwifruit", "count": 28, "color": [0.62, 0.89, 0.31]},
    {"name": "Mangos", "count": 30, "color": [1.00, 0.78, 0.25]},
    {"name": "Grapes", "count": 16, "color": [0.50, 0.19, 0.58]}
]
num_fruits = len(fruits)
window_width, window_height = 800, 600

def reshape(width, height):
    global window_width, window_height
    window_width, window_height = width, height
    glViewport(0, 0, width, height)
    glutPostRedisplay()

def draw_text(text, x, y, r, g, b):
    glColor3f(r, g, b)
    glRasterPos2f(x, y)
    for c in text:
        glutBitmapCharacter(GLUT_BITMAP_9_BY_15, ord(c))

def display():
    glClear(GL_COLOR_BUFFER_BIT)

    # Set up projection
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluOrtho2D(0, window_width, 0, window_height)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()

    # Draw title
```

```
    draw_text("Youth Fruit Preferences in Gachororo", window_width/2 - 180,
window_height - 30, 0.0, 0.0, 0.0)
    draw_text("Total Participants: 170", window_width/2 - 100, window_height -
60, 0.0, 0.0, 0.0)

    # Draw axes
    glColor3f(1.0, 0.0, 0.0)  # Red Y-axis
    glLineWidth(2.0)
    glBegin(GL_LINES)
    glVertex2f(50.0, 50.0)
    glVertex2f(50.0, window_height - 50.0)
    glEnd()

    glColor3f(0.0, 0.0, 0.0)  # Black X-axis
    glBegin(GL_LINES)
    glVertex2f(50.0, 50.0)
    glVertex2f(window_width - 50.0, 50.0)
    glEnd()

    draw_text("Fruit Type", window_width/2 - 40, 20, 0.0, 0.0, 0.0)

    # Calculate dimensions
    bar_width = (window_width - 150) / num_fruits * 0.7
    bar_spacing = (window_width - 150) / num_fruits * 0.3
    max_bar_height = window_height - 150
    start_x = 100.0

    # Draw bars
    for fruit in fruits:
        bar_height = (fruit["count"] / 50) * max_bar_height

        # Draw bar
        glColor3fv(fruit["color"])
        glBegin(GL_QUADS)
        glVertex2f(start_x, 50.0)
        glVertex2f(start_x + bar_width, 50.0)
        glVertex2f(start_x + bar_width, 50.0 + bar_height)
        glVertex2f(start_x, 50.0 + bar_height)
        glEnd()

        # Draw value
        draw_text(str(fruit["count"]), start_x + bar_width/2 - 10, 60 +
bar_height, 0.0, 0.0, 0.0)
```

```
        # Draw fruit name (rotated)
        glPushMatrix()
        glTranslatef(start_x + bar_width/2, 30, 0)
        glRotatef(45, 0, 0, 1)
        draw_text(fruit["name"], 0, 0, 0.0, 0.0, 0.0)
        glPopMatrix()

        start_x += bar_width + bar_spacing

    # Draw y-axis labels
    for i in range(0, 51, 10):
        y = 50 + (i / 50) * max_bar_height
        glColor3f(1.0, 0.0, 0.0)
        glBegin(GL_LINES)
        glVertex2f(45, y)
        glVertex2f(50, y)
        glEnd()
        draw_text(str(i), 30, y - 5, 1.0, 0.0, 0.0)

    glFlush()

def init():
    glClearColor(1.0, 1.0, 1.0, 1.0)

def main():
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
    glutInitWindowSize(window_width, window_height)
    glutCreateWindow(b"ICS2311 Group 7: Fruit Preference Survey")
    glutReshapeFunc(reshape)
    init()
    glutDisplayFunc(display)
    glutMainLoop()

if __name__ == "__main__":
    main()
```

PART B: b_fruit_bars.py

```
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *
import sys
```

G7

```
# Fruit data - same as part A
fruits = [
    {"name": "Avocado", "count": 36, "color": [0.34, 0.51, 0.01]},  # dark green
    {"name": "Orange", "count": 41, "color": [1.00, 0.65, 0.00]},   # orange
color
    {"name": "Banana", "count": 19, "color": [0.89, 0.81, 0.34]},   # yellow
    {"name": "Kiwifruit", "count": 28, "color": [0.62, 0.89, 0.31]}, # kiwi
green
    {"name": "Mangos", "count": 30, "color": [1.00, 0.78, 0.25]},    # mango
orange
    {"name": "Grapes", "count": 16, "color": [0.50, 0.19, 0.58]}     # purple
]
num_fruits = len(fruits)
window_width, window_height = 800, 600

# Offset for the chart (starting point at (5,5))
x_offset = 5.0
y_offset = 5.0

def reshape(width, height):
    global window_width, window_height
    window_width, window_height = width, height
    glViewport(0, 0, width, height)
    glutPostRedisplay()

def draw_text(text, x, y, r, g, b):
    glColor3f(r, g, b)
    glRasterPos2f(x, y)
    for c in text:
        glutBitmapCharacter(GLUT_BITMAP_9_BY_15, ord(c))

def draw_base_coordinate_system():
    # Draw main world axes
    glColor3f(0.5, 0.5, 0.5)  # gray color for base axes
    glLineWidth(1.0)
    glBegin(GL_LINES)
    glVertex2f(0.0, 0.0); glVertex2f(12.0, 0.0)  # x-axis
    glVertex2f(0.0, 0.0); glVertex2f(0.0, 12.0)  # y-axis
    glEnd()

    # Label the axes
    draw_text("0", -0.3, -0.3, 0.5, 0.5, 0.5)  # Origin
    draw_text("X", 12.2, 0.0, 0.5, 0.5, 0.5)   # X-axis
    draw_text("Y", 0.0, 12.2, 0.5, 0.5, 0.5)   # Y-axis
```

```
    draw_text("World Coordinate System", 2.0, 11.5, 0.5, 0.5, 0.5)

    # Grid lines (lighter)
    glColor3f(0.3, 0.3, 0.3)
    glLineWidth(0.5)
    glBegin(GL_LINES)
    for i in range(1, 13):
        glVertex2f(i, 0.0); glVertex2f(i, 12.0)  # vertical
        glVertex2f(0.0, i); glVertex2f(12.0, i)  # horizontal
    glEnd()

    # Label key grid points
    for i in range(1, 13):
        if i % 2 == 0 or i == 5:  # only label even numbers and 5
            draw_text(str(i), i - 0.1, -0.3, 0.5, 0.5, 0.5)  # x-axis labels
            draw_text(str(i), -0.3, i - 0.1, 0.5, 0.5, 0.5)  # y-axis labels

    # Highlight the starting point (5,5) with a red dot
    glColor3f(1.0, 0.0, 0.0)
    glPointSize(8.0)
    glBegin(GL_POINTS)
    glVertex2f(x_offset, y_offset)
    glEnd()

    # Draw arrow pointing to (5,5)
    glColor3f(1.0, 0.0, 0.0)
    glLineWidth(1.5)
    glBegin(GL_LINES)
    glVertex2f(x_offset + 0.1, y_offset + 0.1)
    glVertex2f(x_offset + 0.2, y_offset + 0.2)
    glEnd()

def draw_chart_axes():
    # Draw y-axis (red)
    glColor3f(1.0, 0.0, 0.0)
    glLineWidth(2.0)
    glBegin(GL_LINES)
    glVertex2f(x_offset, y_offset)
    glVertex2f(x_offset, y_offset + 5.0)
    glEnd()

    # Draw x-axis (black)
    glColor3f(0.0, 0.0, 0.0)
    glBegin(GL_LINES)
```

```
    glVertex2f(x_offset, y_offset)
    glVertex2f(x_offset + 8.0, y_offset)
    glEnd()

    # Label the axes
    draw_text("Chart X-Axis", x_offset + 8.2, y_offset, 0.0, 0.0, 0.0)
    draw_text("Chart Y-Axis", x_offset - 0.5, y_offset + 5.2, 1.0, 0.0, 0.0)

    # Add y-axis ticks and labels (red)
    glColor3f(1.0, 0.0, 0.0)
    for i in range(0, 51, 10):
        y = y_offset + (i / 50.0) * 5.0
        glBegin(GL_LINES)
        glVertex2f(x_offset - 0.1, y)
        glVertex2f(x_offset, y)
        glEnd()
        draw_text(str(i), x_offset - 0.5, y - 0.1, 1.0, 0.0, 0.0)

def display():
    glClear(GL_COLOR_BUFFER_BIT)

    # Draw the base coordinate system to show where (5,5) is
    draw_base_coordinate_system()

    # Draw the chart axes starting at (5,5)
    draw_chart_axes()

    # Draw title
    draw_text("Youth Fruit Preferences in Gachororo (Starting at 5,5)", 2.0,
11.0, 1.0, 1.0, 1.0)
    draw_text("Total Participants: 170", 2.0, 10.5, 1.0, 1.0, 1.0)

    # Draw axes labels
    draw_text("Fruit Type", x_offset + 4.0, y_offset - 0.5, 0.0, 0.0, 0.0)

    # Calculate dimensions for the bars
    bar_width = 0.8
    bar_spacing = 0.4
    max_bar_height = 5.0  # maximum height based on y-axis scale
    start_x = x_offset + 0.5

    # Draw bars
    for fruit in fruits:
        bar_height = (fruit["count"] / 50.0) * max_bar_height
```

G7

```
        # Ensure minimum height for visibility
        if bar_height < 0.1:
            bar_height = 0.1

        # Draw bar with fruit color
        glColor3fv(fruit["color"])
        glBegin(GL_QUADS)
        glVertex2f(start_x, y_offset)
        glVertex2f(start_x + bar_width, y_offset)
        glVertex2f(start_x + bar_width, y_offset + bar_height)
        glVertex2f(start_x, y_offset + bar_height)
        glEnd()

        # Draw outline around bar
        glColor3f(0.8, 0.8, 0.8)
        glLineWidth(1.0)
        glBegin(GL_LINE_LOOP)
        glVertex2f(start_x, y_offset)
        glVertex2f(start_x + bar_width, y_offset)
        glVertex2f(start_x + bar_width, y_offset + bar_height)
        glVertex2f(start_x, y_offset + bar_height)
        glEnd()

        # Draw value at top of bar
        draw_text(str(fruit["count"]), start_x + bar_width/2 - 0.2, y_offset +
bar_height + 0.2, 0.0, 0.0, 0.0)

        # Draw fruit name below bar (rotated)
        glPushMatrix()
        glTranslatef(start_x + bar_width/2, y_offset - 0.2, 0)
        glRotatef(-45, 0, 0, 1)
        draw_text(fruit["name"], 0, 0, 1.0, 1.0, 1.0)
        glPopMatrix()

        start_x += bar_width + bar_spacing

    glFlush()

def init():
    glClearColor(0.0, 0.0, 0.0, 1.0)  # black background
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluOrtho2D(-1, 14, -1, 14)  # extended viewport
```

```
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()

def main():
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
    glutInitWindowSize(window_width, window_height)
    glutInitWindowPosition(100, 100)
    glutCreateWindow(b"ICS2311 Group 7: Fruit Preference Survey (Starting at
5,5)")

    init()
    glutDisplayFunc(display)
    glutReshapeFunc(reshape)
    glutMainLoop()

if __name__ == "__main__":
    main()
```

END FILE: