

Graphics

Lab 1

Name / Ahmed Abdallah Aboeleid

ID / 19015274

CODE :

```
void drawScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0, 0.0, 0.0);
    glLineWidth(1.0); // Default line width.
    /*
    -----set point size below-----
    */
    glPointSize(5.0);
    //-----
    /*
    -----write points drawing logic below-----
    */
    glBegin(GL_POINTS);
        for (int i = 0; i < points.size(); i++)
        {
            glVertex3f(points[i].x, points[i].y, 0.0);
        }
    glEnd();
    //-----
    /*
    -----write lines drawing logic below-----
    */
    if(points.size() > 1){
        glBegin(GL_LINES);
            for (int i = 0; i < points.size()-1; i=i+2)
            {
                glVertex3f(points[i].x, points[i].y, 0.0);
                glVertex3f(points[i+1].x, points[i+1].y, 0.0);
            }
        glEnd();
    }
    //-----
    glFlush();
}
```

- ⇒ In this function, color all points and lines with black color using glColor3f() function
- ⇒ Begin to draw points using points which store in vector points by using for loop

⇒ Then check if vector has more than 2 points to begin drawing lines between every two successive points , a line should be drawn connecting them.

```
// Keyboard input processing routine.
void keyInput(unsigned char key, int x, int y)
{
    switch (key)
    {
        case 27:
            exit(0);
            break;
        case '+':{
            float x_pos = rand() % (int) windowHeight;
            float y_pos = rand() % (int) windowWidth;
            points.push_back(glm::vec3(x_pos,y_pos,0.0));
            glutPostRedisplay();
        }
            break;
        case '-':
            if(points.size() > 0)
                points.pop_back();
            glutPostRedisplay();
            break;
        default:
            break;
    }
}
```

⇒ In this function, we determine action taken when press + or –

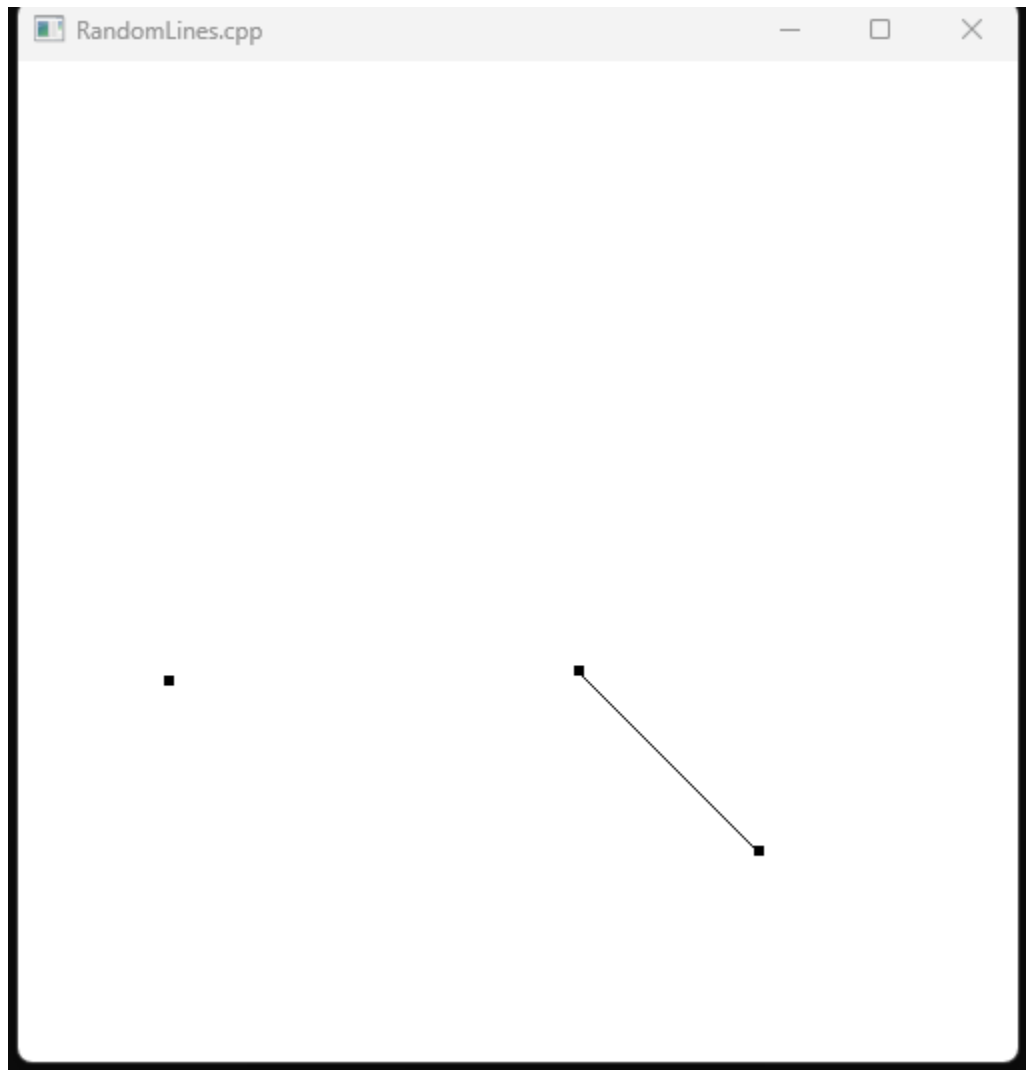
⇒ When user press + , get 2 random values for x and y within application window and add this point in vector to be drawn after that

⇒ When user press - , remove last point store in vector points

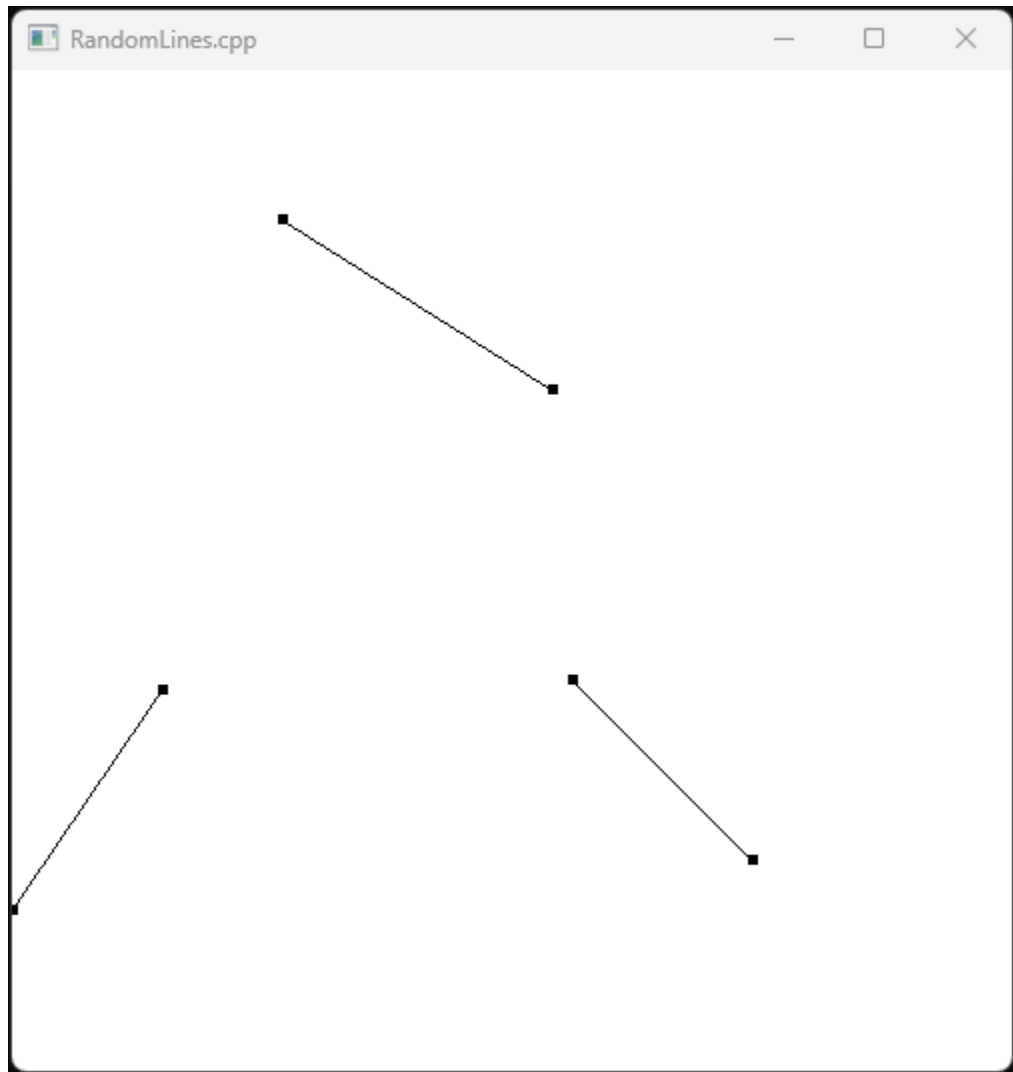
I change code only in this two function which I explain above.

Sample runs :

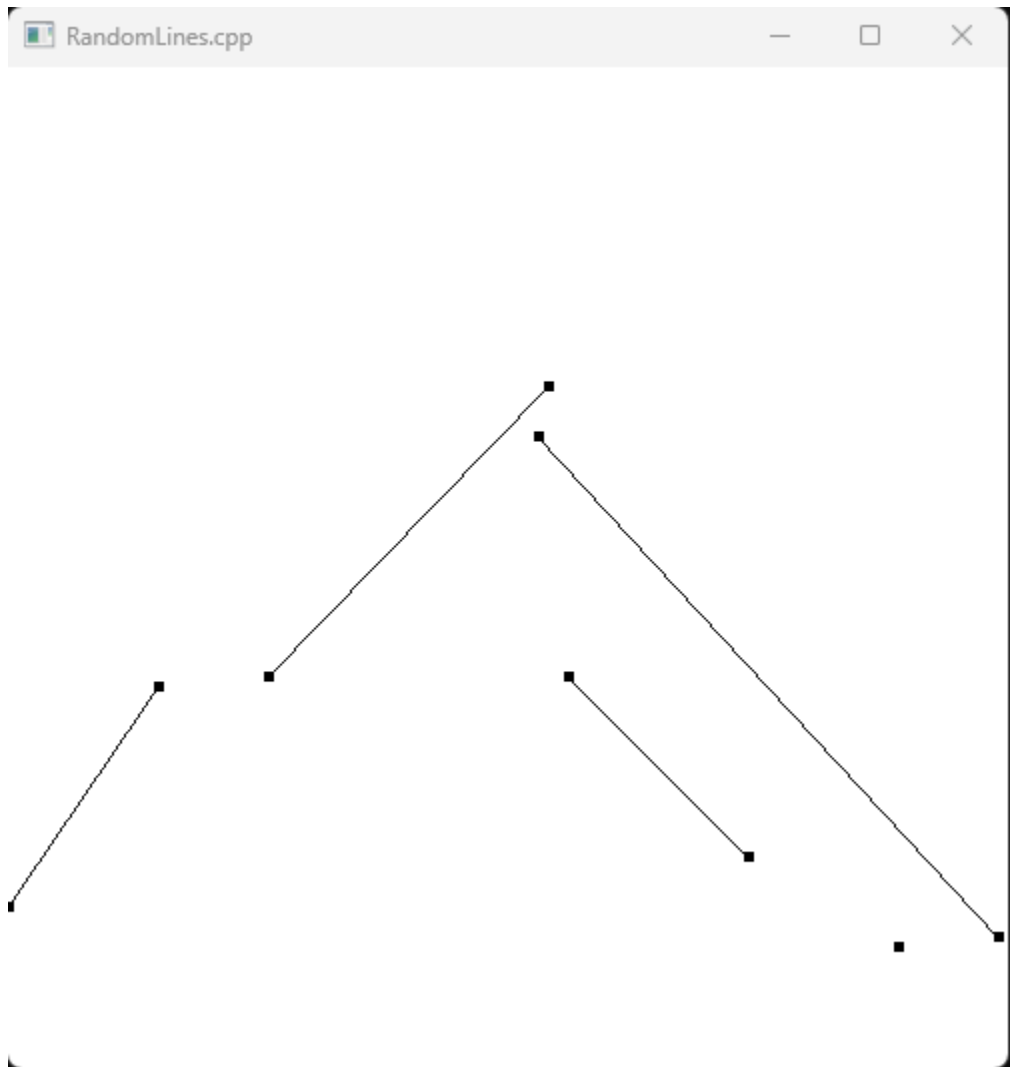
⇒ After adding odd number of points



⇒ After adding even number of points



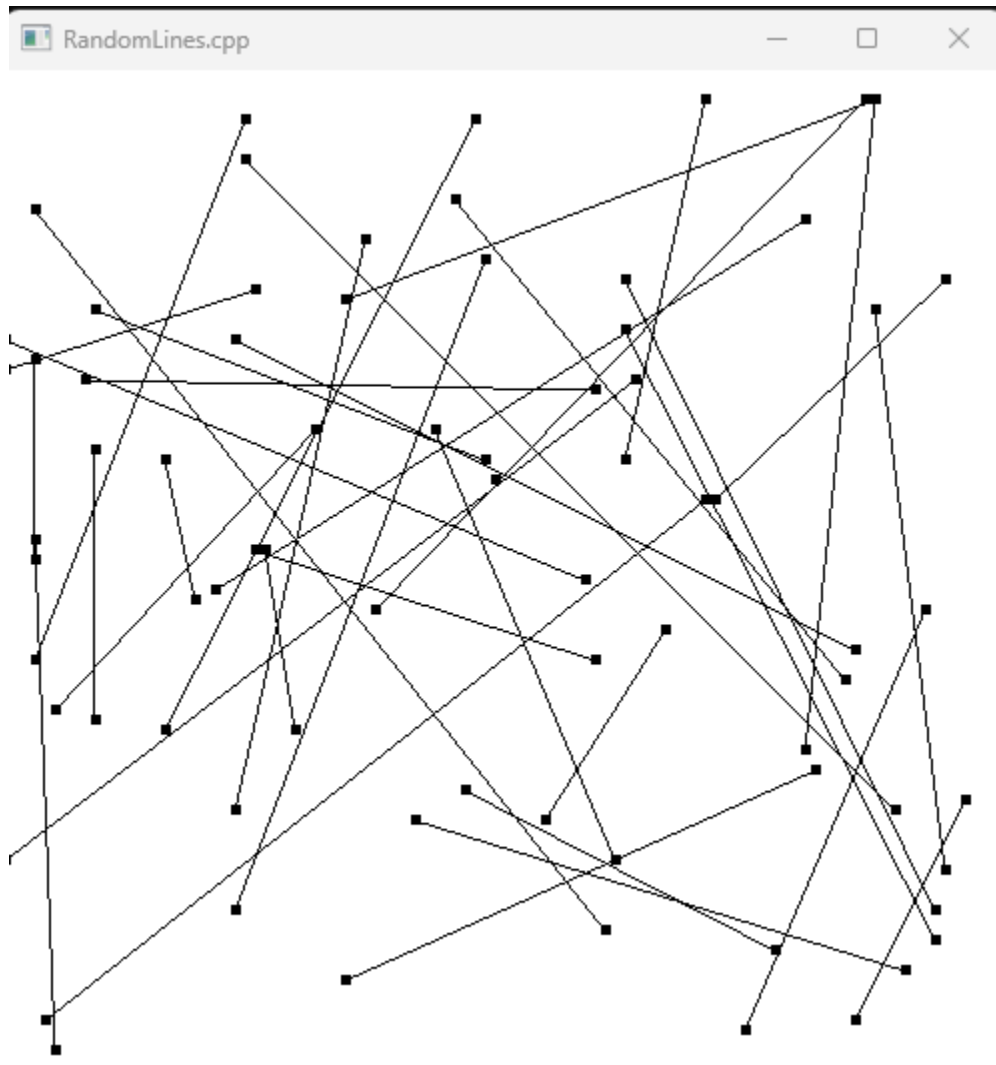
⇒ After removing some points and adding new ones



⇒ After removing All points



⇒ Then adding many points again



⇒ Then resizing window application

