Rajdhani
DATE / /

GROUP-B ASSIGNMENT-9 (CG)

Name : Aditya Onkar Patil
Batch : G4 (SE4)
Roll No : 21449
Performance Date : 30/10/2021
Submission Date : 31/10/2021

TITLE : 2-D Transformations.

PROBLEM STATEMENT : write a C++ program to draw 2-D object and perform following basic transforms: scaling, Translation and rotation. Apply the concept of operator overloading.

LEARNING OBJECTIVES : To learn the concepts of 2-D transformations like scaling, translation and rotation as a part of computer graphics.

LEARNING OUTCOMES : After completion of this assignment students will be able to :

1. Implement 2-D transformations of translation, scaling and rotation of 2-D figures.

S/W AND H/W REQUIREMENTS : 64 bit open source Linux or its derivative, C++ programming tools like GCC/G++, Qt Creator, OpenGL etc.

REFERENCES :
1. Programming principles and practice using C++, Bjarne Stroustroup.
2. www.qt.io.

Rajdhani
DATE / /

CONCEPT RELATED THEORY:

2-D Transformations:

Transformation means changing some graphics of a figure by applying rules. when a transformation takes place on a 2-D plane, it's called a 2-D transformation.

Transformations play important role in computer graphics to reposition the graphics on the screen and change their size or orientation.

For translation:

$X = x + t_x$

$Y = y + t_y$, Hence the matrix forms up as

$[x' \; y'] = [x_1 \; y_1] + T$ where "T" is the translational matrix.

For scaling:

$x_2 = x_1 * s$ where "s" is the scaling factor.

$y_2 = y_1 * s$

For Rotating:

this involves rotation of original axes and finding the new coordinates accordingly:

$x_2 = x_1 \cos\theta + y_1 \sin\theta$

$y_2 = x_1 \sin\theta - y_1 \cos\theta$

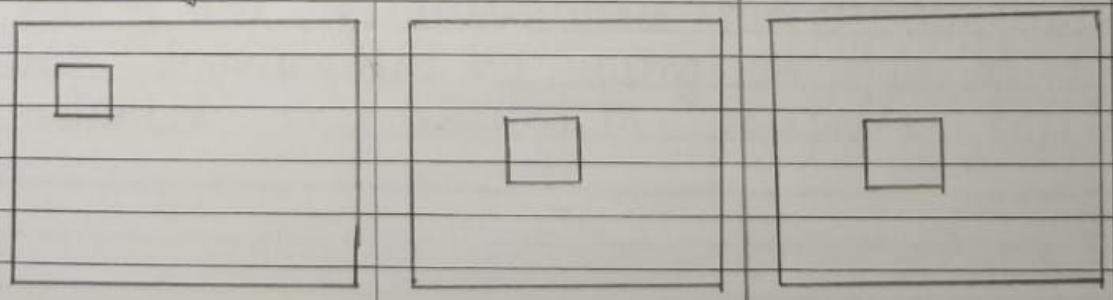
and so on for the respective axes rotations.

Rajdhani
DATE / /

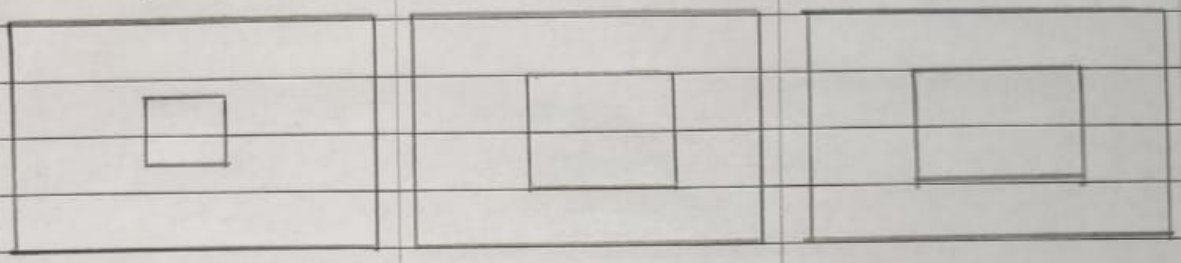
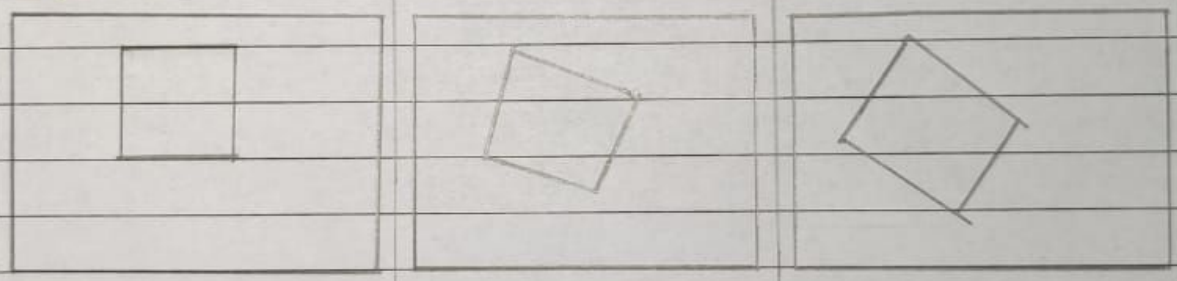
ALGORITHM:

1. Start the program.
2. Define the functions for drawing 2-D figures and translate, scale & rotate them in the mainwindow.cpp.
3. Include their function names in the header file mainwindow.h.
4. Apply concepts of 2-D transforms and operator overloading and make the following 2-D transformations.
5. Design the user interface in mainwindow.ui for taking suitable inputs and showing the required 2-D tranformation outputs.
6. End the program.

TEST CASES:

Sr No.	Description	Expected Output	Actual Output	Status.
1.	Translation: tx=20, ty=20 	Square translation	Square translation	Pass

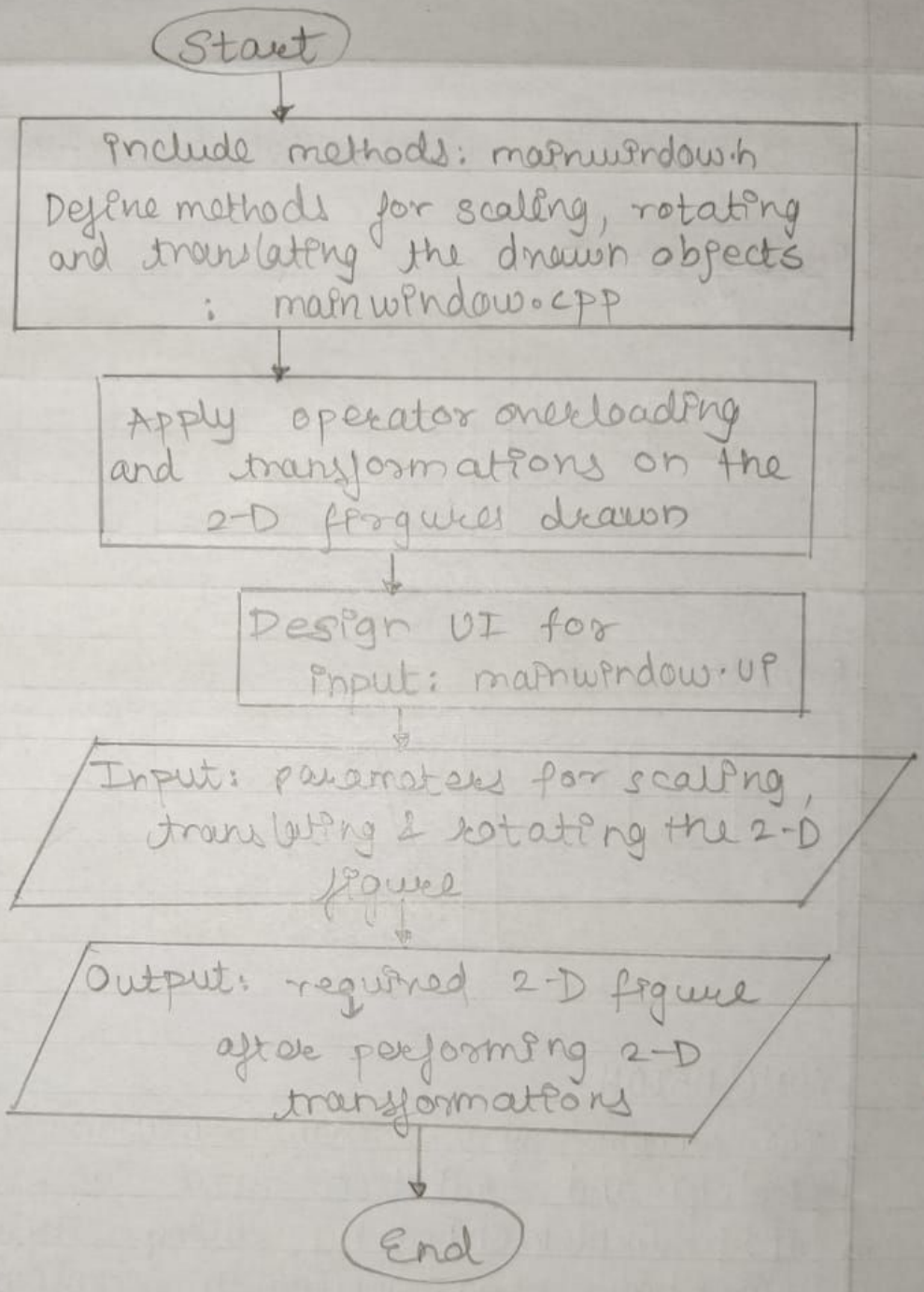
Rajdhani
DATE / /

No.	Description	Expected Output	Actual Output	Status
2.	Scaling : S = 1.3 			Pass
3.	Rotation: 			Pass

CONCLUSION:

The concept of 2-D transformations like translating, scaling and rotation was successfully understood and implemented by using concepts of object oriented programming in computer graphics.

FLOWCHART:


Start
Include methods: mainwindow.h
Define methods for scaling, rotating and translating the drawn objects
: mainwindow.cpp
Apply operator overloading and transformations on the 2-D figures drawn
Design UI for
input: mainwindow.ui
Input: parameters for scaling, translating & rotating the 2-D figure
Output: required 2-D figure after performing 2-D transformations
End

mainwindow.h @ 2-D_transform - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- 2-D_transform
  - 2-D_transform.pro
  - Headers
    - mainwindow.h
  - Sources
  - Forms

mainwindow.h <Select Symbol> Windows (CRLF) Line: 1, Col: 1

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    void LineDraw(float,float,float,float);
    void Scaler(float);
    void Translator(int,int);
    void Rotator(float);
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private slots:
    void on_pushButton_clicked();

    void on_pushButton_2_clicked();

    void on_pushButton_3_clicked();

    void on_pushButton_4_clicked();

private:
    Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H
```

Welcome Edit Design Debug Projects Help

2-D...orm Debug

Type to locate (Ctrl... 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

main.cpp @ 2-D_transform - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- 2-D_transform
  - 2-D_transform.pro
  - Headers
    - mainwindow.h
  - Sources
    - main.cpp
    - mainwindow.cpp
  - Forms

main.cpp <No Symbols> Windows (CRLF) Line: 1, Col: 1

```cpp
#include "mainwindow.h"

#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}
```

Welcome Edit Design Debug Projects Help

2-D...orm Debug

Type to locate (Ctrl... 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

mainwindow.cpp @ 2-D_transform - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- 2-D_transform
  - 2-D_transform.pro
  - Headers
    - mainwindow.h
  - Sources
    - main.cpp
    - mainwindow.cpp
  - Forms

mainwindow.cpp

```cpp
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include<math.h>

int coordarr[20][4];
int lenarr[20];
QImage img(900,400,QImage::Format_RGB888);
int count=0;
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::LineDraw(float x1, float y1, float x2, float y2){
    float dx, dy, length, Xinc, Yinc;
    dx = (x2 - x1); dy = (y2 - y1);
    if (abs(dx)>abs(dy)){
        length = abs(dx);
    }
    else{
        length = abs(dy);
    }
    Xinc = dx/length; Yinc = dy/length;
    for(int i = 0;i <=length; i++){
       img.setPixel(x1,y1,qRgb(225,225,225));
       x1 = x1 + Xinc; y1 = y1 + Yinc;
    }
}



void MainWindow::Scaler(float scaler){
    for (int x = 0; x < 900; ++x)
```

Welcome | Edit | Design | Debug | Projects | Help

2-D...orm | Debug

Type to locate (Ctrl... | 1 Issues | 2 Search Results | 3 Application Output | 4 Compile Output | 5 QML Debugger Console | 6 General Messages | 8 Test Results

Windows (CRLF) | Line: 1, Col: 1

mainwindow.cpp @ 2-D_transform - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- 2-D_transform
  - 2-D_transform.pro
  - Headers
    - mainwindow.h
  - Sources
    - main.cpp
    - mainwindow.cpp
  - Forms

mainwindow.cpp

```cpp
        x1 = x1 + Xinc; y1 = y1 + Yinc;
    }
}



void MainWindow::Scaler(float scaler){
    for (int x = 0; x < 900; ++x)
        {
            for (int y = 0; y < 400; ++y)
            {
                img.setPixel(x, y, qRgb(0, 0, 0));
            }
        }


    for (int i=0;i<=count;i++){
        float nx1,nx2,ny1,ny2;
        nx1= (coordarr[i][0]*(scaler+2) - (scaler*coordarr[i][2]))/2;
        ny1= (coordarr[i][1]*(scaler+2) - (scaler*coordarr[i][3]))/2;
        nx2= (coordarr[i][2]*(scaler+2) - (scaler*coordarr[i][0]))/2;
        ny2= (coordarr[i][3]*(scaler+2) - (scaler*coordarr[i][1]))/2;
        coordarr[i][0]=nx1;
        coordarr[i][1]=ny1;
        coordarr[i][2]=nx2;
        coordarr[i][3]=ny2;
    }
    LineDraw(coordarr[0][0],coordarr[0][1]+float(scaler*(lenarr[0]/2)),coordarr[0][2],coordarr[0][3]+float(scaler*(lenarr[0]/2)));
    LineDraw(coordarr[1][0]+float(scaler*(lenarr[1]/2)),coordarr[1][1],coordarr[1][2]+float(scaler*(lenarr[1]/2)),coordarr[1][3]);
    LineDraw(coordarr[2][0],coordarr[2][1]-float(scaler*(lenarr[2]/2)),coordarr[2][2],coordarr[2][3]-float(scaler*(lenarr[2]/2)));
    LineDraw(coordarr[3][0]-float(scaler*(lenarr[3]/2)),coordarr[3][1],coordarr[3][2]-float(scaler*(lenarr[3]/2)),coordarr[3][3]);
    //  for (int j=0;j<=count;j++) {
     //   LineDraw(coordarr[j][0],coordarr[j][1],coordarr[j][2],coordarr[j][3]);
    //}
    ui->label->setPixmap(QPixmap::fromImage(img));

}

void MainWindow::Rotator(float angle){
    for (int x = 0; x < 900; ++x)
```

Windows (CRLF) | Line: 1, Col: 1

Welcome | Edit | Design | Debug | Projects | Help

2-D...orm | Debug

Type to locate (Ctrl... | 1 Issues | 2 Search Results | 3 Application Output | 4 Compile Output | 5 QML Debugger Console | 6 General Messages | 8 Test Results

mainwindow.cpp @ 2-D_transform - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- 2-D_transform
  - 2-D_transform.pro
  - Headers
    - mainwindow.h
  - Sources
    - main.cpp
    - mainwindow.cpp
  - Forms

mainwindow.cpp | <Select Symbol> | Windows (CRLF) | Line: 1, Col: 1

```cpp
     //   LineDraw(coordarr[j][0],coordarr[j][1],coordarr[j][2],coordarr[j][3]);
    //}
    ui->label->setPixmap(QPixmap::fromImage(img));

}

void MainWindow::Rotator(float angle){
    for (int x = 0; x < 900; ++x)
        {
            for (int y = 0; y < 400; ++y)
            {
                img.setPixel(x, y, qRgb(0, 0, 0));
            }
        }
      //  coordarr[0][2]=(coordarr[0][2]*cos(angle)) + (coordarr[0][3]*sin(angle));
       // coordarr[0][3]=(coordarr[0][2]*sin(angle)) - (coordarr[0][3]*cos(angle));
    for (int i=0;i<count;i++){
        float nwx1,nwx2,nwy1,nwy2;

        nwx1= (coordarr[i][0]*qFastCos(angle)) + (coordarr[i][1]*qFastSin(angle));
        nwx2= (coordarr[i][2]*qFastCos(angle)) + (coordarr[i][3]*qFastSin(angle));
        nwy1= (coordarr[i][0]*qFastSin(angle)) - (coordarr[i][1]*qFastCos(angle));
        nwy2= (coordarr[i][2]*qFastSin(angle)) - (coordarr[i][3]*qFastCos(angle));
        coordarr[i][0]=nwx1;
        coordarr[i][1]=nwy1;
        coordarr[i][2]=nwx2;
        coordarr[i][3]=nwy2;
    }
    for (int j=0;j<count;j++) {
        LineDraw(coordarr[j][0],coordarr[j][1]+300,coordarr[j][2],coordarr[j][3]+300);
    }
    ui->label->setPixmap(QPixmap::fromImage(img));

}
void MainWindow::Translator(int tx,int ty){
    for (int x = 0; x < 900; ++x)
        {
            for (int y = 0; y < 400; ++y)
            {
                img.setPixel(x, y, qRgb(0, 0, 0));
```

Welcome | Edit | Design | Debug | Projects | Help

2-D...orm | Debug

Type to locate (Ctrl... | 1 Issues | 2 Search Results | 3 Application Output | 4 Compile Output | 5 QML Debugger Console | 6 General Messages | 8 Test Results

mainwindow.cpp @ 2-D_transform - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- 2-D_transform
  - 2-D_transform.pro
  - Headers
    - mainwindow.h
  - Sources
    - main.cpp
    - mainwindow.cpp
  - Forms

mainwindow.cpp | <Select Symbol> | Windows (CRLF) | Line: 1, Col: 1

```cpp
    }
    for (int j=0;j<count;j++) {
        LineDraw(coordarr[j][0],coordarr[j][1],coordarr[j][2],coordarr[j][3]);

    }
    ui->label->setPixmap(QPixmap::fromImage(img));
}
void MainWindow::on_pushButton_clicked()                //DRAWING
{
    int x1 =ui->textEdit->toPlainText().toFloat();
    int y1 =ui->textEdit_3->toPlainText().toFloat();
    int x2 =ui->textEdit_2->toPlainText().toFloat();
    int y2 =ui->textEdit_4->toPlainText().toFloat();
// take these coordinates and store them in the array for translating,scaling and rotating
    coordarr[count][0]=x1;
    coordarr[count][1]=y1;
    coordarr[count][2]=x2;
    coordarr[count][3]=y2;
    lenarr[count]= sqrt(((x2-x1)*(x2-x1))+((y2-y1)*(y2-y1)));
    count++;
    LineDraw(x1,y1,x2,y2);
    ui->label->setPixmap(QPixmap::fromImage(img));
}


void MainWindow::on_pushButton_2_clicked()            //SCALING
{
// clear the screen and draw the same figure by using coordinates from array by multiplying scaling factor
    float scale= ui->textEdit_5->toPlainText().toDouble();
    Scaler(scale);

}


void MainWindow::on_pushButton_3_clicked()                //TRANSLATION
{
// clear the screen and draw the same figure by using coordinates from array by adding the translating factor
    int tx= ui->textEdit_6->toPlainText().toFloat();
    int ty= ui->textEdit_7->toPlainText().toFloat();
    Translator(tx,ty);
```

Welcome | Edit | Design | Debug | Projects | Help

2-D...orm | Debug

Type to locate (Ctrl... | 1 Issues | 2 Search Results | 3 Application Output | 4 Compile Output | 5 QML Debugger Console | 6 General Messages | 8 Test Results

mainwindow.cpp @ 2-D_transform - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

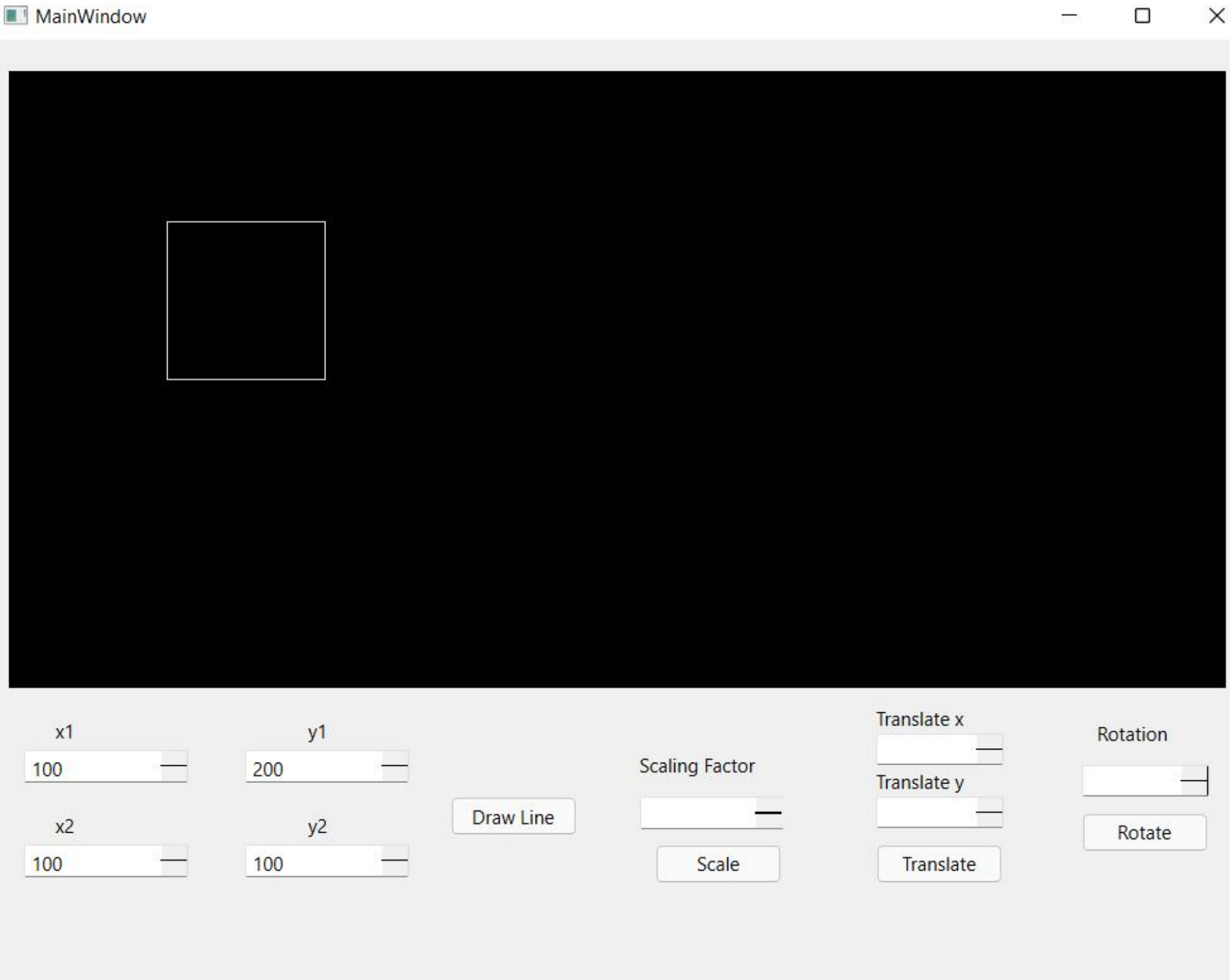
Projects

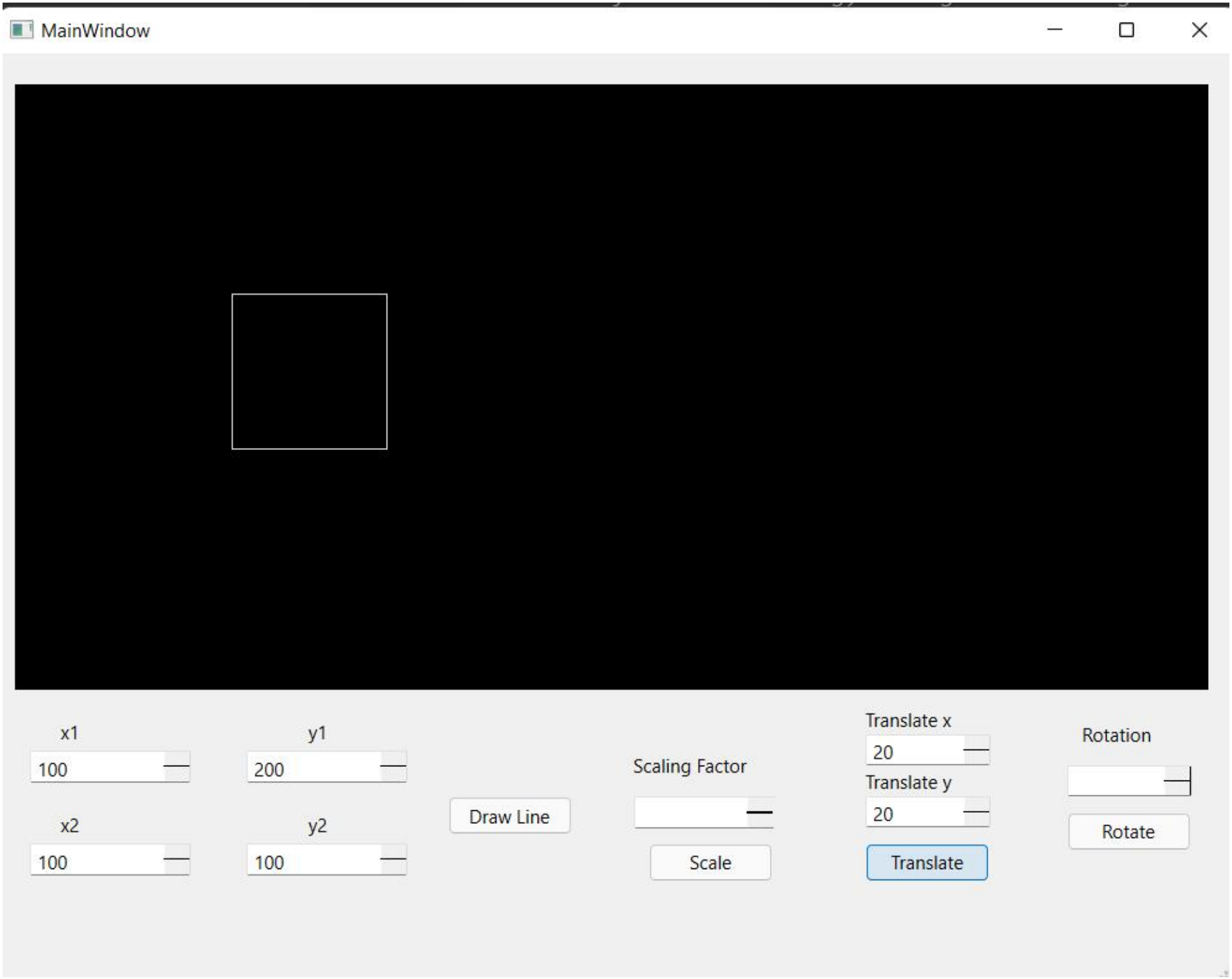
- 2-D_transform
  - 2-D_transform.pro
  - Headers
    - mainwindow.h
  - Sources
    - main.cpp
    - mainwindow.cpp
  - Forms

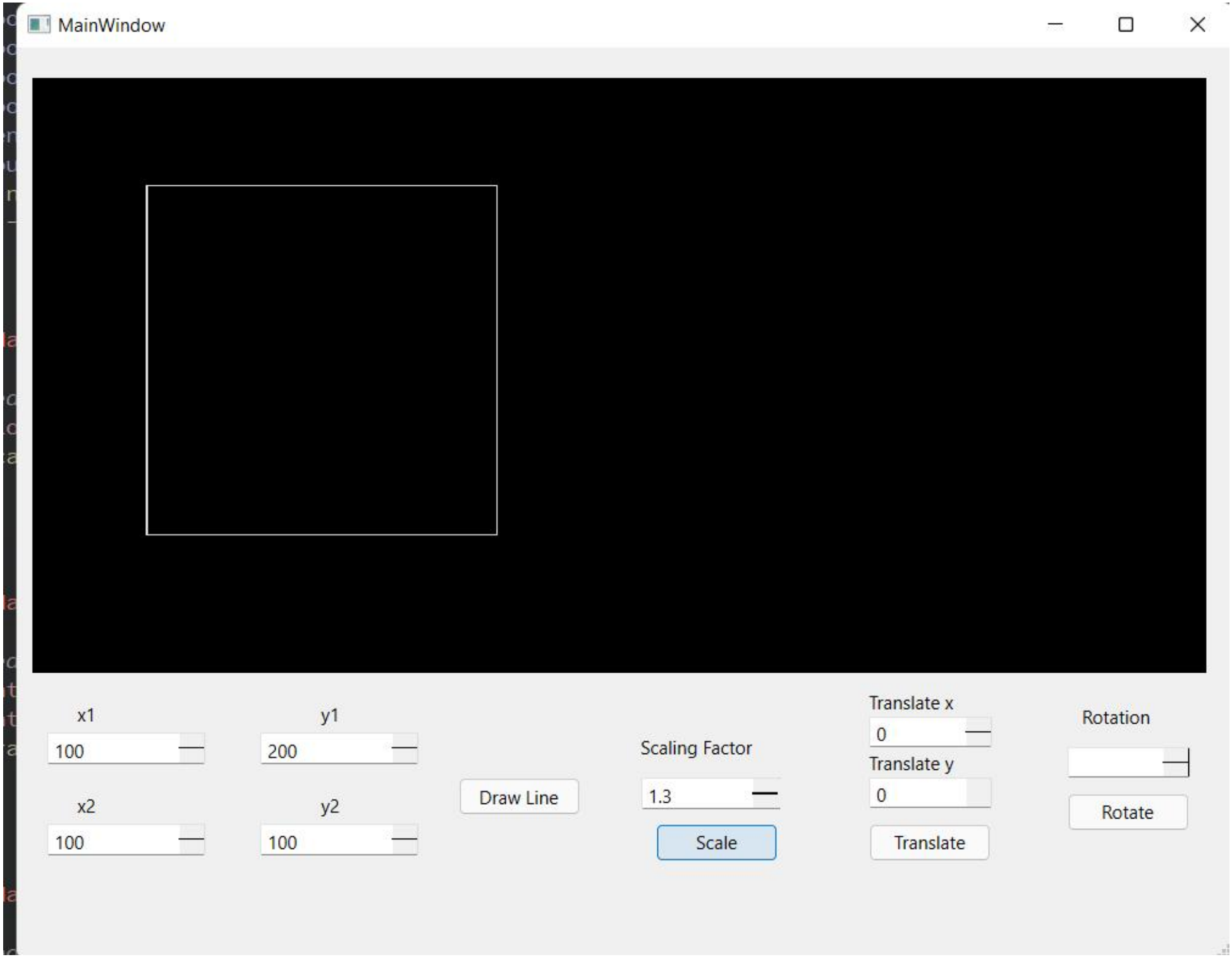
mainwindow.cpp

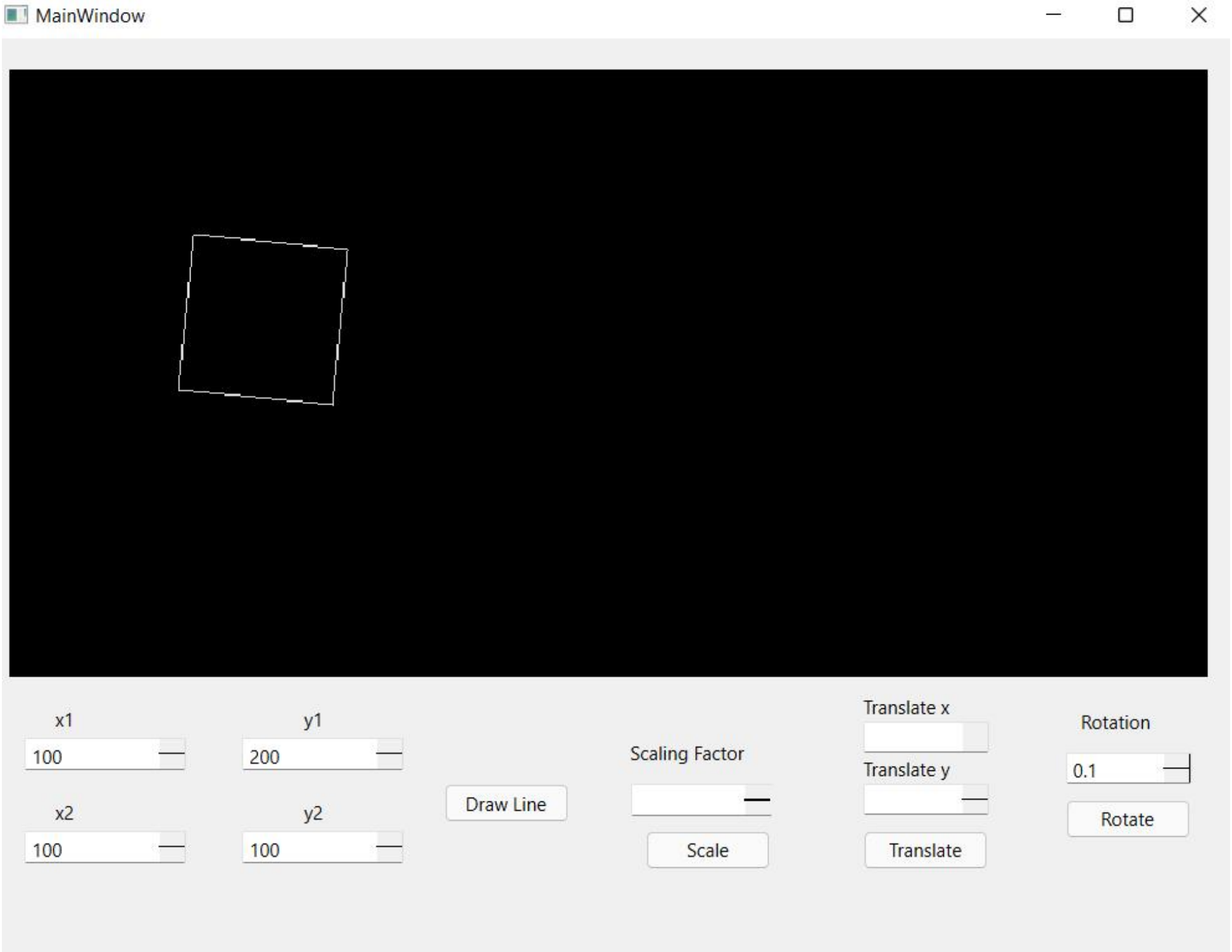
```cpp
    int y2 =ui->textEdit_4->toPlainText().toFloat();
// take these coordinates and store them in the array for translating,scaling and rotating
    coordarr[count][0]=x1;
    coordarr[count][1]=y1;
    coordarr[count][2]=x2;
    coordarr[count][3]=y2;
    lenarr[count]= sqrt(((x2-x1)*(x2-x1))+((y2-y1)*(y2-y1)));
    count++;
    LineDraw(x1,y1,x2,y2);
    ui->label->setPixmap(QPixmap::fromImage(img));
}


void MainWindow::on_pushButton_2_clicked()            //SCALING
{
// clear the screen and draw the same figure by using coordinates from array by multiplying scaling factor
    float scale= ui->textEdit_5->toPlainText().toDouble();
    Scaler(scale);

}


void MainWindow::on_pushButton_3_clicked()               //TRANSLATION
{
// clear the screen and draw the same figure by using coordinates from array by adding the translating factor
    int tx= ui->textEdit_6->toPlainText().toFloat();
    int ty= ui->textEdit_7->toPlainText().toFloat();
    Translator(tx,ty);

}


void MainWindow::on_pushButton_4_clicked()              //ROTATION
{
// clear the screen and draw the same figure by using coordinates from array by using rotating factor
    float angle= ui->textEdit_8->toPlainText().toFloat();
    Rotator(angle);
}
```

Type to locate (Ctrl... 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results


MainWindow
x1
100
y1
200
x2
100
y2
100
Draw Line
Scaling Factor
Scale
Translate x
Translate y
Translate
Rotation
Rotate


MainWindow
x1
100
y1
200
x2
100
y2
100
Draw Line
Scaling Factor
Scale
Translate x
20
Translate y
20
Translate
Rotation
Rotate


MainWindow
x1
100
y1
200
x2
100
y2
100
Draw Line
Scaling Factor
1.3
Scale
Translate x
0
Translate y
0
Translate
Rotation
Rotate


MainWindow
x1
100
y1
200
x2
100
y2
100
Draw Line
Scaling Factor
Scale
Translate x
Translate y
Translate
Rotation
0.1
Rotate