Page:
Date:

Assignment 8

Class - SE IV Roll No - 21430
Batch - F4
D.O.P - 23/10/2020 DOS - 29/10/2020

Title :- Basic 2D transformation.

Problem Statement :-
Write C++ program to draw 2D object & perform following task transformation scaling, transformation, rotation. apply Concept of operator overloading.

S/W requirement :-
1. QT creator
2 Windows 10.

H/W requirement :-
1. 4 Gb ram.

Theory :-
Transformation means changing 2D graphic into something else by applying rules. We can have various types of transformation as translation. Scaling updrown, rotation, shearing, reflection etc. When a transformation takes place on a 2D plane it is called 2D transformation. Transformation play an important role in computer graphics to reposition the graphics on screen and change

their size or orientation. Translation, scaling and rotation are basic transformation.

Translation is the process of changing the co-ordinates of 2D objects by adding translation co-ordinate.

Basic Formula:

Resultant points = Currents pts x translating matrix.

$$\begin{bmatrix} Rx_1 & Ry_1 & 1 \\ Rx_2 & Ry_2 & 1 \\ \vdots & \vdots & 1 \\ Rx_n & Ry_n & \end{bmatrix} = \begin{bmatrix} Px_1 & Py_1 & 1 \\ Px_2 & Py_2 & 1 \\ \vdots & \vdots & 1 \\ Px_n & Py_n & \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ tx & ty & 1 \end{bmatrix}$$

Scaling -:

This is the process of magnifying or diminishing the 2D object by given factors. These are casual types of scaling. Symmetric / unsymmetric, magnifical, dimensionary etc.

$$\begin{bmatrix} Rx_1 & Ry_1 & 1 \\ Rx_2 & Ry_2 & 1 \\ \vdots & \vdots & 1 \\ Rx_n & Ry_n & 1 \end{bmatrix} = \begin{bmatrix} Px_1 & Py_1 & 1 \\ Px_2 & Py_2 & 1 \\ \vdots & \vdots & 1 \\ Px_n & Py_n & 1 \end{bmatrix} \times \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation-1

Rotation is the process of rotating a 2D object about the origin usually, rotation about a particular point in need that can be done by the object to origin then rotate it and then translate it to the origin. The rotation is done as follow.

$$\begin{bmatrix} Rx_1 & Ry_1 & 1 \\ Rx_2 & Ry_2 & 1 \\ \vdots & \vdots & 1 \\ Rx_n & Ry_n & 1 \end{bmatrix} = \begin{bmatrix} Px_1 & Py_1 & 1 \\ Px_2 & Py_2 & 1 \\ \vdots & \vdots & 1 \\ Px_n & Py_n & 1 \end{bmatrix} \times \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Algorithm / Psudocode:-

```
class Matrix
    double mat[4][3];
    int n;
    operator *(matrix M)

void multiply matrix( int a[ ][3], int b[3][b], int M2)
{   int m2 = 3, n2 = 3;
    for( i=0; i<n; i++)
    {
        for(j=0; j<n2; j++)
        {
            int sum=0;
            int k;
            for( k=0; i<M2; k++)
            {
                sum= mat[i][k] + mat2[k][j];
```

```
            K[i][j] = sum;
        }
    }
    return K;
}
Translation(p)
{
    tx = input(1);
    ty = input(2);
    matrix T(tx, ty);
    Matrix R = P * T;
}
Rotation(matrix P)
{
    Angle = input();
    Matrix R = { (cosθ, sinθ, 0),
                 (-sinθ, cosθ, 0),
                 (0, 0, 1), };
    Matrix Res = P * R;
}
Scaling(matrix P)
{
    Sx = input();
    Sy = input2();
    Matrix S = { (Sx, 0, 0),
                 (0, Sy, 0),
                 (0, 0, 1) };
    Matrix Res = S * P;
}
```

* Dry Run :-

i) Translation

$P = (4, 6)$

$T_x = 2 \;,\; T_y = 3$

$$\begin{bmatrix} 4 & 6 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 9 & 1 \end{bmatrix}$$

Res = (6, 9)

ii) Scaling

$P = (4, 6)$

$S_x = 2 \;,\; S_y = 2$

$$\begin{bmatrix} 4 & 6 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 8 & 12 & 1 \end{bmatrix}$$

Res = (8, 12)

iii) Rotation

$P = (1, 1)$

$\theta = 90°$

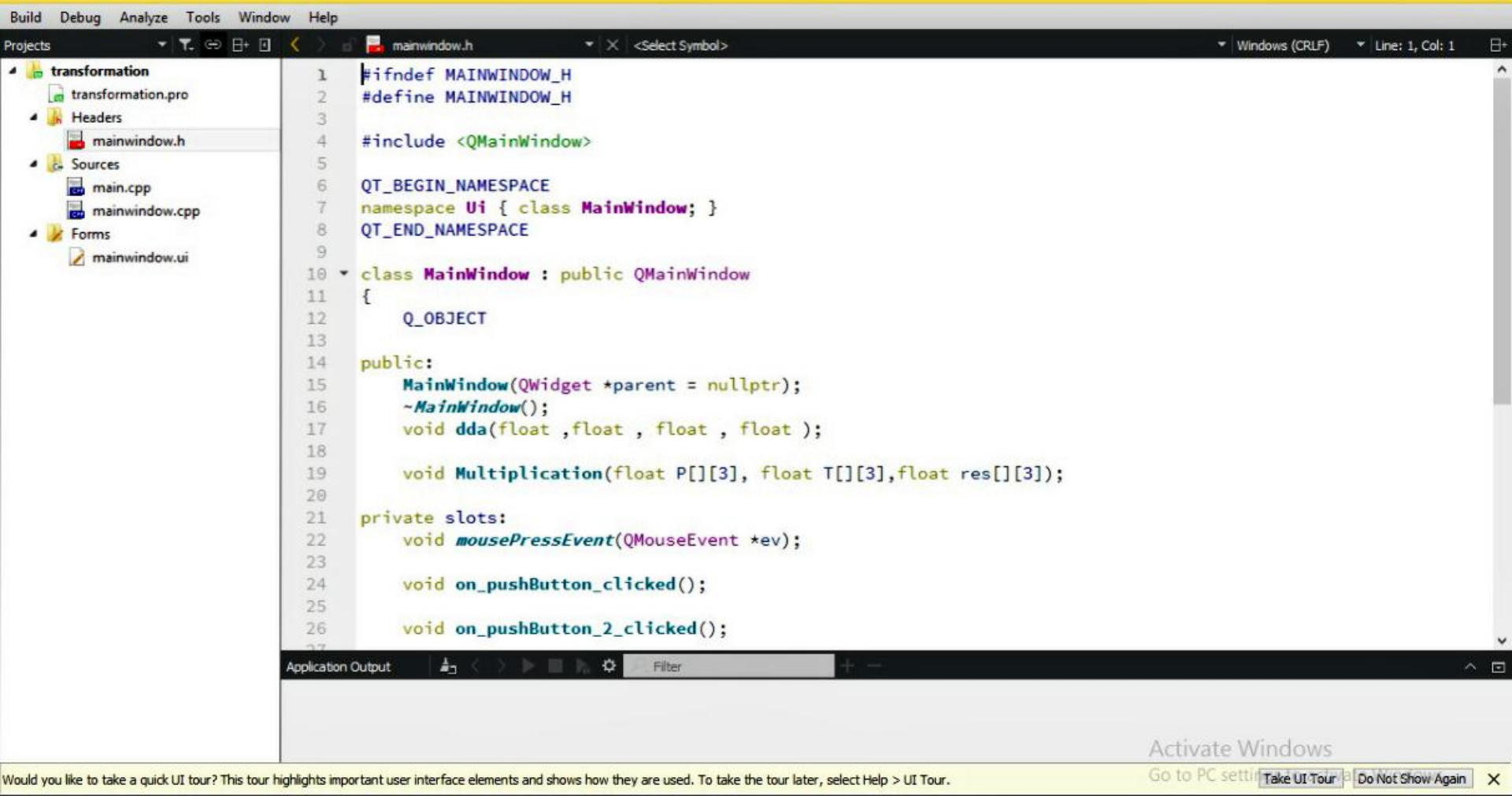
$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \cos 90 & \sin 90 & 0 \\ -\sin 90 & \cos 90 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 \end{bmatrix}$$

Res = (-1, 1)

Page:
Date: / /

Conclusion ::

The 2D transformation have done Successfully implemented and applied on triangle and rhombus.


Build Debug Analyze Tools Window Help
Projects
transformation
transformation.pro
Headers
mainwindow.h
Sources
main.cpp
mainwindow.cpp
Forms
mainwindow.ui
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:
MainWindow(QWidget *parent = nullptr);
~MainWindow();
void dda(float ,float , float , float);
void Multiplication(float P[][3], float T[][3],float res[][3]);
private slots:
void mousePressEvent(QMouseEvent *ev);
void on_pushButton_clicked();
void on_pushButton_2_clicked();
Application Output
Filter
Activate Windows
Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.
Take UI Tour
Do Not Show Again

File Edit Build Debug Analyze Tools Window Help

Projects

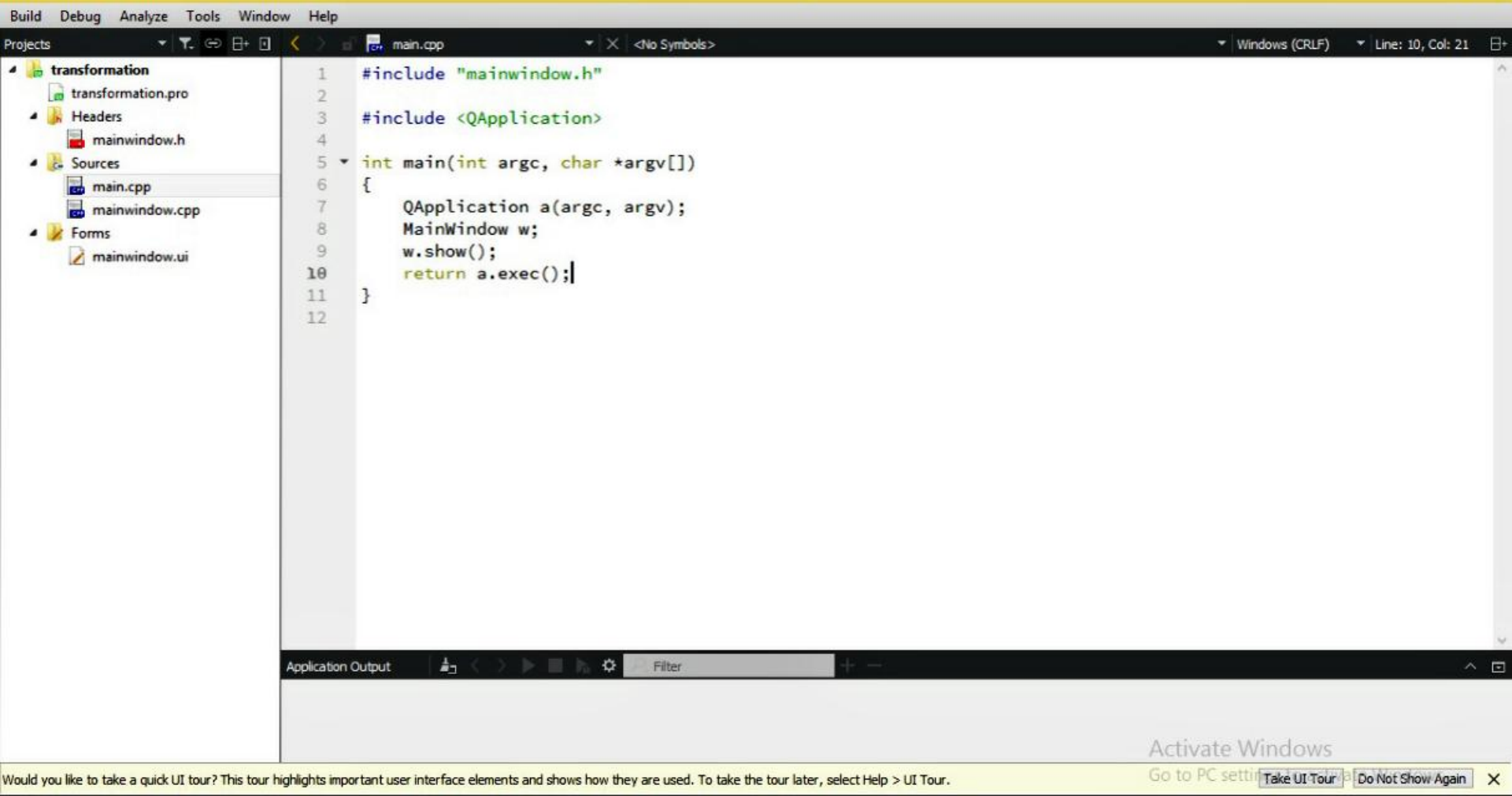
- transformation
  - transformation.pro
  - Headers
    - mainwindow.h
  - Sources
    - main.cpp
    - mainwindow.cpp
  - Forms
    - mainwindow.ui

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

```
19      void Multiplication(float P[][3], float T[][3],float res[][3]);
20
21  private slots:
22      void mousePressEvent(QMouseEvent *ev);
23
24      void on_pushButton_clicked();
25
26      void on_pushButton_2_clicked();
27
28      void on_pushButton_3_clicked();
29
30      void on_pushButton_4_clicked();
31
32      void on_pushButton_5_clicked();
33
34  private:
35      Ui::MainWindow *ui;
36      float x1,y1,x2,y2,tx,ty,sx,sy,x;
37      float a[20],b[20];
38      int ver;
39      bool start;
40  };
41  #endif // MAINWINDOW_H
42
43
44
```

Application Output | Filter

Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour. Take UI Tour | Do Not Show Again


Build Debug Analyze Tools Window Help
Projects
main.cpp
<No Symbols>
Windows (CRLF)
Line: 10, Col: 21
transformation
transformation.pro
Headers
mainwindow.h
Sources
main.cpp
mainwindow.cpp
Forms
mainwindow.ui
#include "mainwindow.h"
#include <QApplication>
int main(int argc, char *argv[])
{
QApplication a(argc, argv);
MainWindow w;
w.show();
return a.exec();
}
Application Output
Filter
Activate Windows
Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.
Take UI Tour
Do Not Show Again

Build Debug Analyze Tools Window Help

Projects

- transformation
  - transformation.pro
  - Headers
    - mainwindow.h
  - Sources
    - main.cpp
    - mainwindow.cpp
  - Forms
    - mainwindow.ui

mainwindow.cpp | MainWindow::mousePressEvent(QMouseEvent *) -> void | Windows (CRLF) | Line: 98, Col: 26

```cpp
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include<iostream>
#include<QMessageBox>
#include<math.h>
#include<QMouseEvent>
using namespace std;

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    dda(-249,0,249,0);
    dda(0,-249,0,249);
    start = false;
    ver = 0;
}

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

QImage img(500,500,QImage::Format_RGB888);

class Matrices
```

Application Output | Filter

Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour. Take UI Tour | Do Not Show Again

Build Debug Analyze Tools Window Help

Projects

- transformation
  - transformation.pro
  - Headers
    - mainwindow.h
  - Sources
    - main.cpp
    - mainwindow.cpp
  - Forms
    - mainwindow.ui

mainwindow.cpp | MainWindow::mousePressEvent(QMouseEvent *) -> void | Windows (CRLF) | Line: 98, Col: 26

```
24
25  QImage img(500,500,QImage::Format_RGB888);
26
27  class Matrices
28  {
29      public:
30          float a[2][3];
31          float b[3][3];
32          Matrices operator *(Matrices ob)
33          {
34              Matrices temp;
35              for(int i=0; i<2; i++)
36              {
37                  for (int j=0; j<3; j++)
38                  {
39                      temp.a[i][j] = 0;
40                      for (int k=0; k<3; k++)
41                      {
42                          temp.a[i][j] += a[i][k] * ob.b[k][j];
43                      }
44                  }
45              }
46              return temp;
47          }
48  };
49
50  void MainWindow::dda(float x1, float y1, float x2, float y2)
```

Application Output | Filter

Build Debug Analyze Tools Window Help

Projects

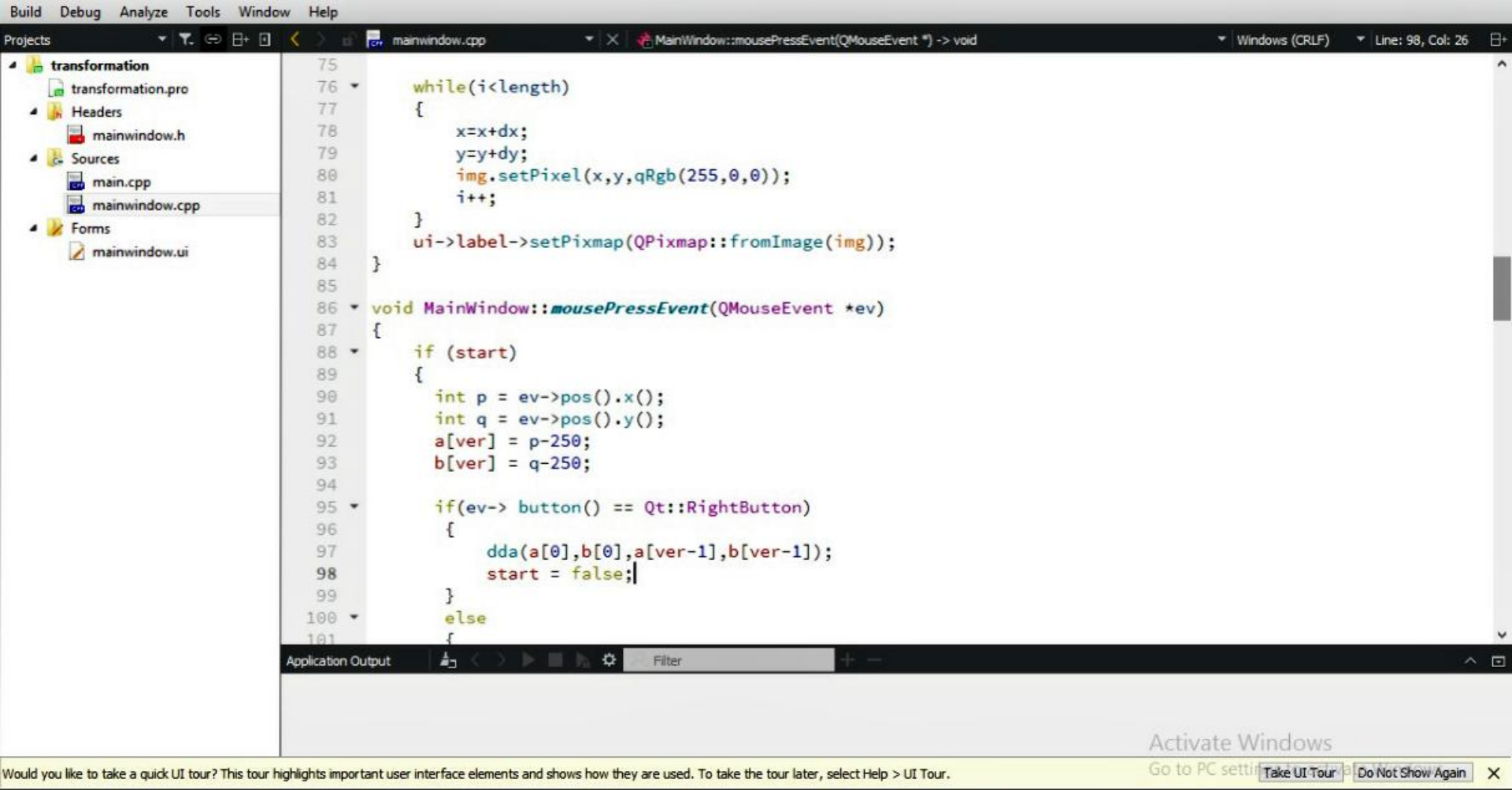
- transformation
  - transformation.pro
  - Headers
    - mainwindow.h
  - Sources
    - main.cpp
    - mainwindow.cpp
  - Forms
    - mainwindow.ui

mainwindow.cpp | MainWindow::mousePressEvent(QMouseEvent *) -> void | Windows (CRLF) | Line: 98, Col: 26

```
50  void MainWindow::dda(float x1, float y1, float x2, float y2)
51  {
52      x1 += 250;
53      y1 += 250;
54      x2 += 250;
55      y2 += 250;
56
57      float dx,dy,length,x,y;
58      dx=x2-x1;
59      dy=y2-y1;
60
61      if(abs(dx)>abs(dy))
62      {
63          length=abs(dx);
64      }
65      else
66      {
67          length=abs(dy);
68      }
69      dx=dx/length;
70      dy=dy/length;
71      x=x1;
72      y=y1;
73      int i=0;
74      img.setPixel(x,y,qRgb(255,0,0));
75
76      while(i<length)
```

Application Output | Filter

Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour. Take UI Tour | Do Not Show Again


Build Debug Analyze Tools Window Help
Projects
transformation
transformation.pro
Headers
mainwindow.h
Sources
main.cpp
mainwindow.cpp
Forms
mainwindow.ui
mainwindow.cpp
MainWindow::mousePressEvent(QMouseEvent *) -> void
Windows (CRLF)
Line: 98, Col: 26
while(i<length)
{
x=x+dx;
y=y+dy;
img.setPixel(x,y,qRgb(255,0,0));
i++;
}
ui->label->setPixmap(QPixmap::fromImage(img));
}
void MainWindow::mousePressEvent(QMouseEvent *ev)
{
if (start)
{
int p = ev->pos().x();
int q = ev->pos().y();
a[ver] = p-250;
b[ver] = q-250;
if(ev-> button() == Qt::RightButton)
{
dda(a[0],b[0],a[ver-1],b[ver-1]);
start = false;
}
else
{
Application Output
Filter
Activate Windows
Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.
Take UI Tour
Do Not Show Again

Build Debug Analyze Tools Window Help

Projects

- transformation
  - transformation.pro
  - Headers
    - mainwindow.h
  - Sources
    - main.cpp
    - mainwindow.cpp
  - Forms
    - mainwindow.ui

mainwindow.cpp | MainWindow::mousePressEvent(QMouseEvent *) -> void | Windows (CRLF) | Line: 98, Col: 26

```
101         {
102             if (ver>0)
103             {
104                 dda(a[ver], b[ver], a[ver-1],b[ver-1]);
105             }
106          }
107          ver++;
108
109     }
110     a[ver] = a[0];
111     b[ver] = b[0];
112 }
113
114
115 void MainWindow::on_pushButton_4_clicked()
116 {
117     start = true;
118 }
119
120 void MainWindow::on_pushButton_clicked()
121 {
122     tx = ui->textEdit->toPlainText().toFloat();
123     ty = ui->textEdit_2->toPlainText().toFloat();
124
125     for(int i=0; i<ver; i++)
126     {
127
```

Application Output | Filter

Build Debug Analyze Tools Window Help

Projects

- transformation
  - transformation.pro
  - Headers
    - mainwindow.h
  - Sources
    - main.cpp
    - mainwindow.cpp
  - Forms
    - mainwindow.ui

mainwindow.cpp | MainWindow::mousePressEvent(QMouseEvent *) -> void | Windows (CRLF) | Line: 98, Col: 26

```
127
128         Matrices A,B,result;
129         float P[2][3] = {{a[i],b[i],1},{a[i+1],b[i+1],1}};
130         float T[3][3] = {{1,0,0},{0,1,0},{tx,ty,1}};
131
132         for (int i=0; i<2; i++)
133         {
134             for (int j=0; j<3; j++)
135             {
136                 A.a[i][j] = P[i][j];
137             }
138         }
139         for (int i=0; i<3; i++)
140         {
141             for (int j=0; j<3; j++)
142             {
143                 B.b[i][j] = T[i][j];
144             }
145         }
146
147         result = A*B;
148         dda(result.a[0][0], result.a[0][1], result.a[1][0], result.a[1][1]);
149     }
150 }
151
152 void MainWindow::on_pushButton_2_clicked()
153 {
```

Application Output | Filter

Build Debug Analyze Tools Window Help

Projects

- transformation
  - transformation.pro
  - Headers
    - mainwindow.h
  - Sources
    - main.cpp
    - mainwindow.cpp
  - Forms
    - mainwindow.ui

mainwindow.cpp | MainWindow::mousePressEvent(QMouseEvent *) -> void | Windows (CRLF) | Line: 98, Col: 26

```
153  {
154      sx = ui->textEdit_3->toPlainText().toFloat();
155      sy = ui->textEdit_4->toPlainText().toFloat();
156
157      for(int i=0; i<ver; i++)
158      {
159          Matrices A,B,result;
160          float P[2][3] = {{a[i],b[i],1},{a[i+1],b[i+1],1}};
161          float S[3][3] = {{sx,0,0},{0,sy,0},{0,0,1}};
162
163          for (int i=0; i<2; i++)
164          {
165              for (int j=0; j<3; j++)
166              {
167                  A.a[i][j] = P[i][j];
168              }
169          }
170          for (int i=0; i<3; i++)
171          {
172              for (int j=0; j<3; j++)
173              {
174                  B.b[i][j] = S[i][j];
175              }
176          }
177
178          result = A*B;
179          dda(result.a[0][0], result.a[0][1], result.a[1][0], result.a[1][1]);
```

Application Output | Filter

Build Debug Analyze Tools Window Help

Projects

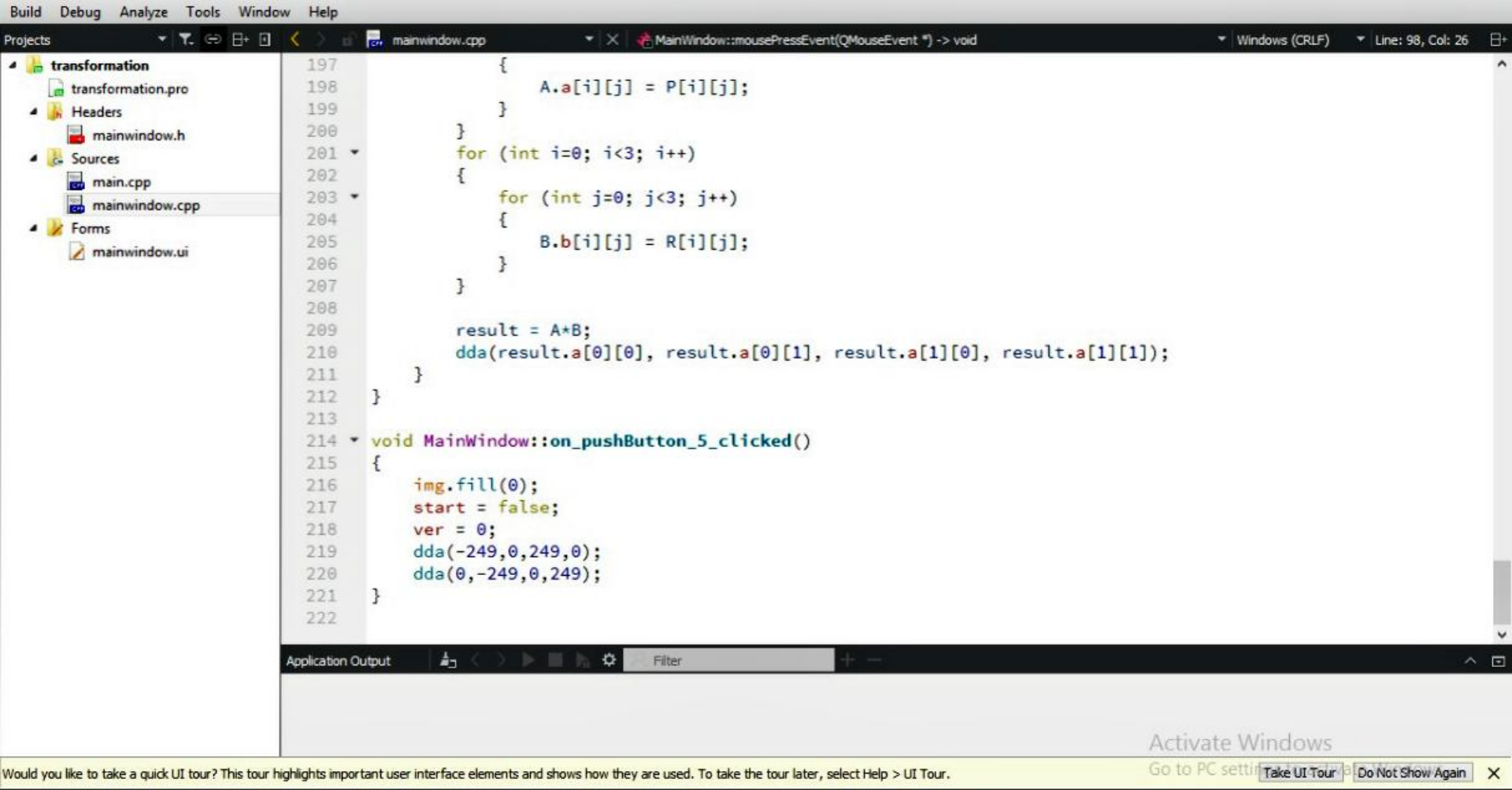
- transformation
  - transformation.pro
  - Headers
    - mainwindow.h
  - Sources
    - main.cpp
    - mainwindow.cpp
  - Forms
    - mainwindow.ui

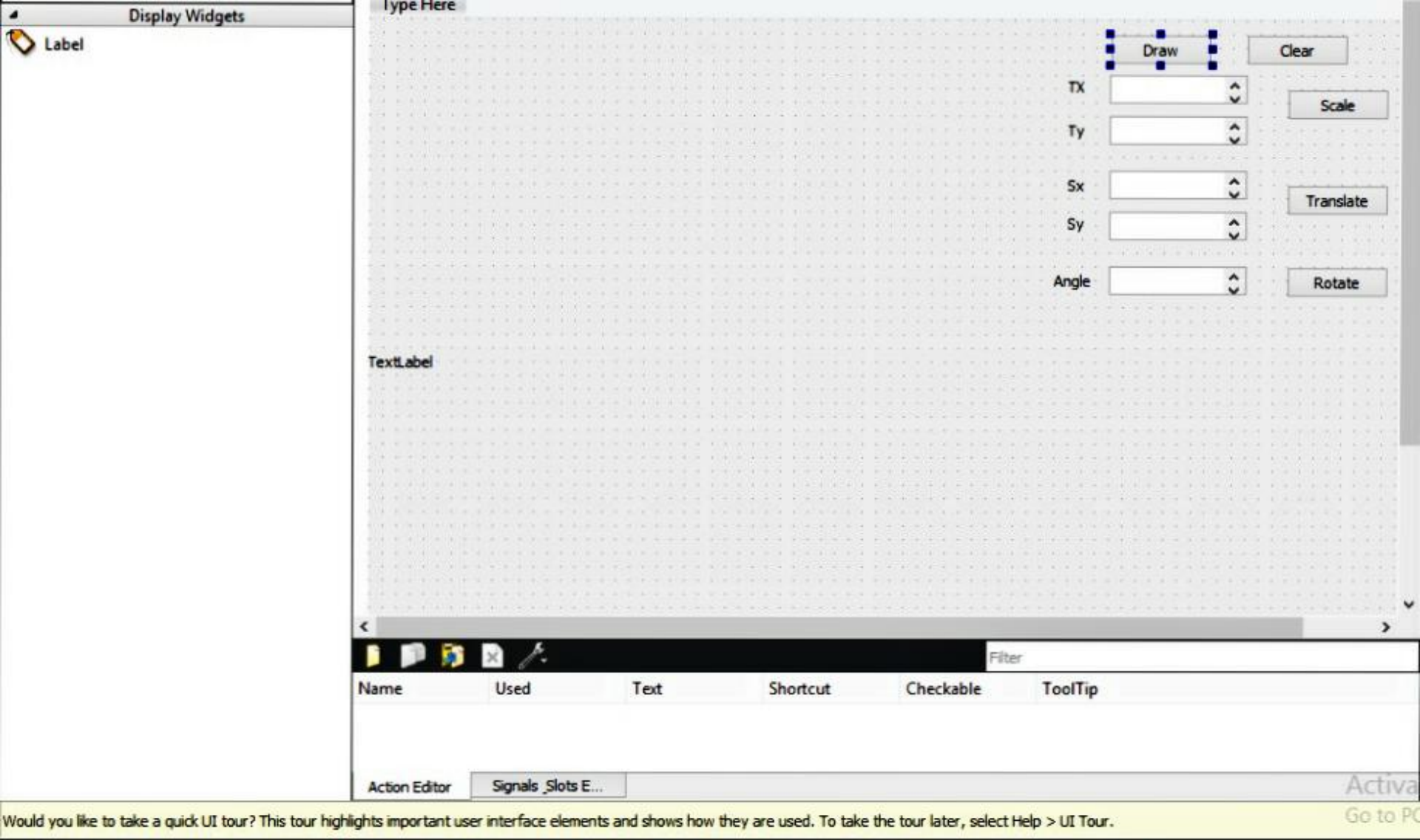
mainwindow.cpp | MainWindow::mousePressEvent(QMouseEvent *) -> void | Windows (CRLF) | Line: 98, Col: 26

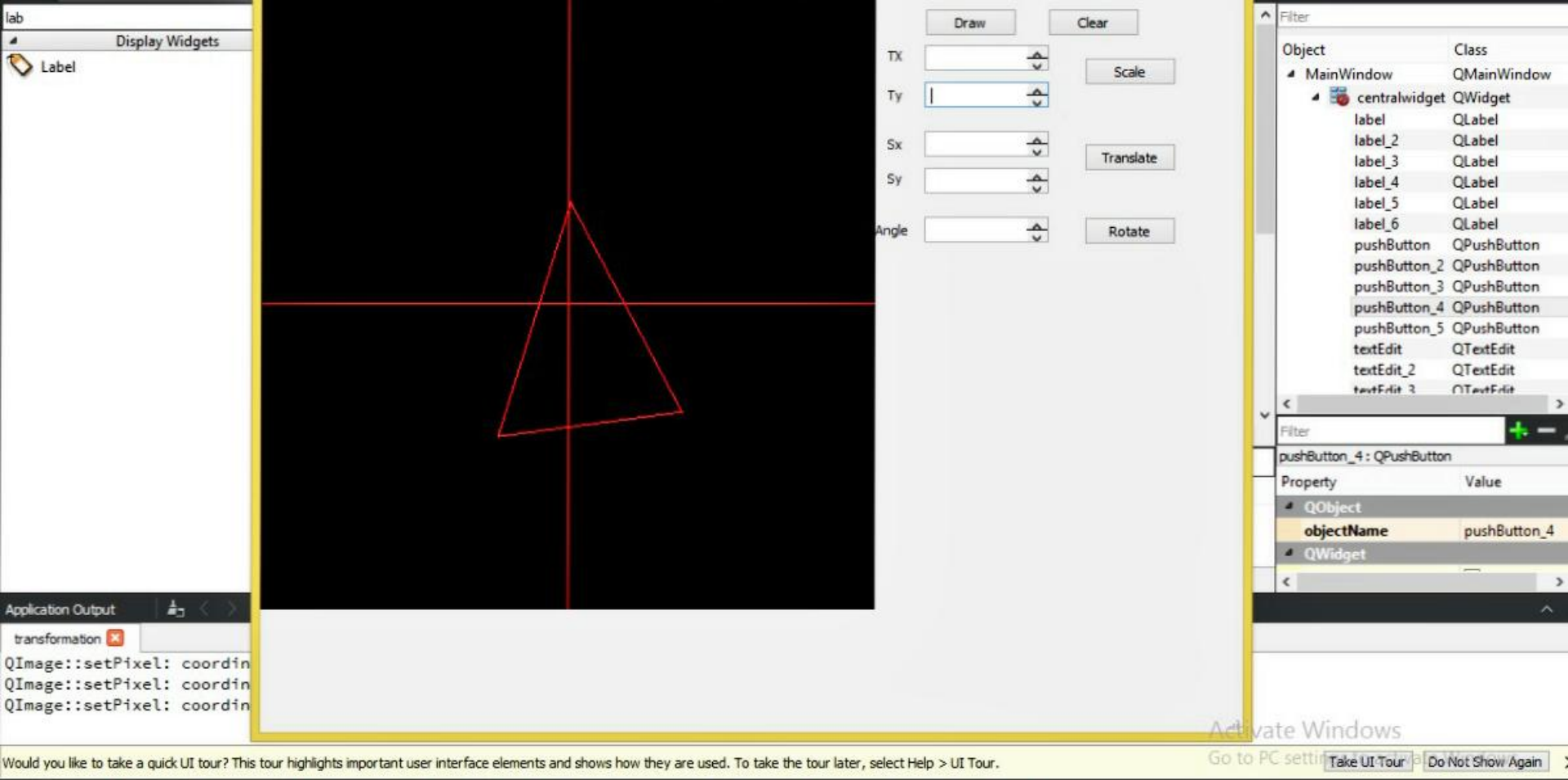
```
180     }
181 }
182
183 void MainWindow::on_pushButton_3_clicked()
184 {
185     x = ui->textEdit_5->toPlainText().toFloat();
186     x *= 3.142/180;
187
188     for(int i=0; i<ver; i++)
189     {
190         Matrices A,B,result;
191         float P[2][3] = {{a[i],b[i],1},{a[i+1],b[i+1],1}};
192         float R[3][3] = {{cos(x),(-sin(x)),0},{sin(x),cos(x),0},{0,0,1}};
193
194         for (int i=0; i<2; i++)
195         {
196             for (int j=0; j<3; j++)
197             {
198                 A.a[i][j] = P[i][j];
199             }
200         }
201         for (int i=0; i<3; i++)
202         {
203             for (int j=0; j<3; j++)
204             {
205                 B.b[i][j] = R[i][j];
206             }
```

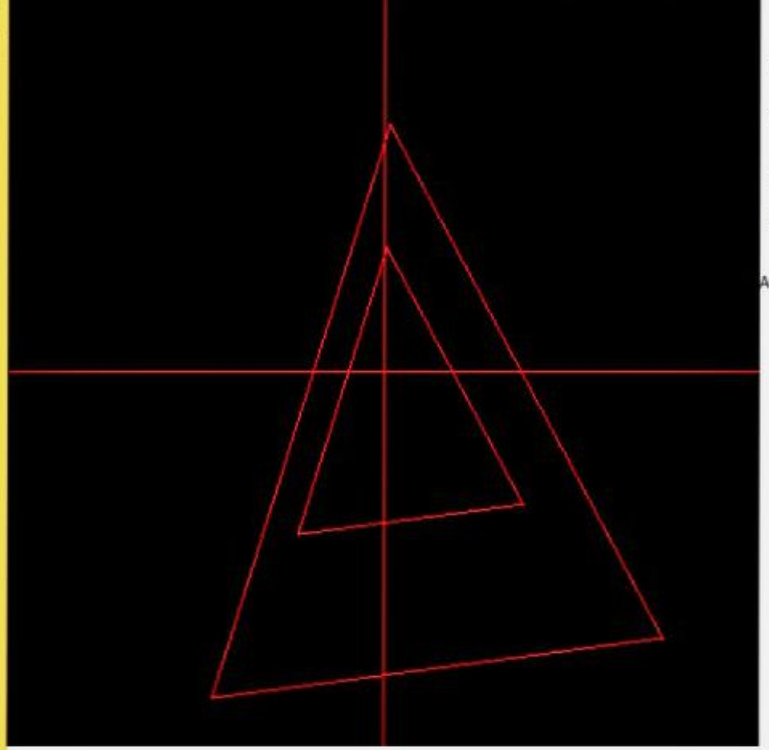
Application Output | Filter

Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour. Take UI Tour | Do Not Show Again

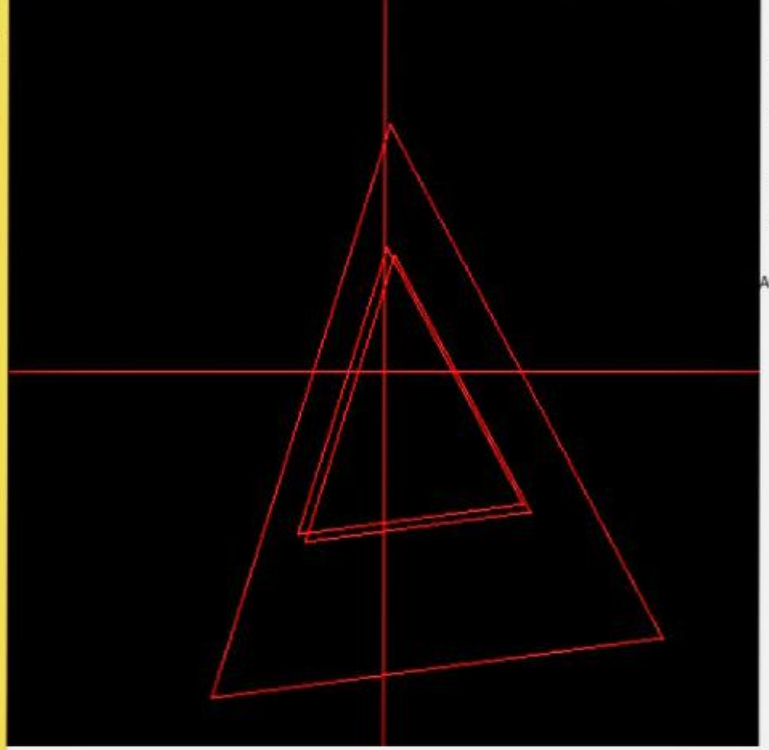

Build Debug Analyze Tools Window Help
Projects
transformation
transformation.pro
Headers
mainwindow.h
Sources
main.cpp
mainwindow.cpp
Forms
mainwindow.ui
mainwindow.cpp
MainWindow::mousePressEvent(QMouseEvent *) -> void
Windows (CRLF)
Line: 98, Col: 26
197 {
198 A.a[i][j] = P[i][j];
199 }
200 }
201 for (int i=0; i<3; i++)
202 {
203 for (int j=0; j<3; j++)
204 {
205 B.b[i][j] = R[i][j];
206 }
207 }
208
209 result = A*B;
210 dda(result.a[0][0], result.a[0][1], result.a[1][0], result.a[1][1]);
211 }
212 }
213
214 void MainWindow::on_pushButton_5_clicked()
215 {
216 img.fill(0);
217 start = false;
218 ver = 0;
219 dda(-249,0,249,0);
220 dda(0,-249,0,249);
221 }
222
Application Output
Filter
Activate Windows
Go to PC settings to activate Windows.
Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.
Take UI Tour
Do Not Show Again


Display Widgets
Label
Type Here
Draw
Clear
TX
Ty
Sx
Sy
Angle
Scale
Translate
Rotate
TextLabel
Filter
Name
Used
Text
Shortcut
Checkable
ToolTip
Action Editor
Signals _Slots E...
Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.


lab
Display Widgets
Label
Draw
Clear
TX
Ty
Sx
Sy
Angle
Scale
Translate
Rotate
Filter
Object
Class
MainWindow
QMainWindow
centralwidget
QWidget
label
QLabel
label_2
QLabel
label_3
QLabel
label_4
QLabel
label_5
QLabel
label_6
QLabel
pushButton
QPushButton
pushButton_2
QPushButton
pushButton_3
QPushButton
pushButton_4
QPushButton
pushButton_5
QPushButton
textEdit
QTextEdit
textEdit_2
QTextEdit
Filter
pushButton_4 : QPushButton
Property
Value
QObject
objectName
pushButton_4
QWidget
Application Output
transformation
QImage::setPixel: coordin
QImage::setPixel: coordin
QImage::setPixel: coordin
Activate Windows
Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.
Take UI Tour
Do Not Show Again



Draw
Clear
TX
2
Scale
Ty
2
Sx
Translate
Sy
Angle
Rotate
lab
Display Widgets
Label
Filter
Object
Class
MainWindow
QMainWindow
centralwidget
QWidget
label
QLabel
label_2
QLabel
label_3
QLabel
label_4
QLabel
label_5
QLabel
label_6
QLabel
pushButton
QPushButton
pushButton_2
QPushButton
pushButton_3
QPushButton
pushButton_4
QPushButton
pushButton_5
QPushButton
textEdit
QTextEdit
textEdit_2
QTextEdit
Filter
pushButton_4 : QPushButton
Property
Value
QObject
objectName
pushButton_4
QWidget
Application Output
transformation
QImage::setPixel: coordin
QImage::setPixel: coordin
QImage::setPixel: coordin
Activate Windows
Go to PC settings to activate Windows.
Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.
Take UI Tour
Do Not Show Again



Draw
Clear
TX 2
Ty 2
Sx 5
Sy 5
Angle
Scale
Translate
Rotate

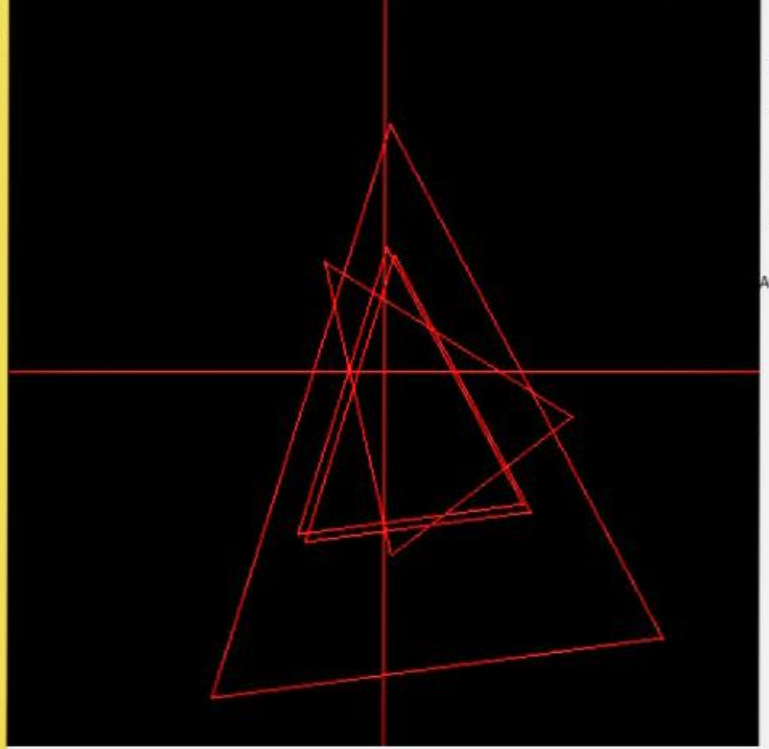
lab
Display Widgets
Label

Filter
Object | Class
MainWindow | QMainWindow
centralwidget | QWidget
label | QLabel
label_2 | QLabel
label_3 | QLabel
label_4 | QLabel
label_5 | QLabel
label_6 | QLabel
pushButton | QPushButton
pushButton_2 | QPushButton
pushButton_3 | QPushButton
pushButton_4 | QPushButton
pushButton_5 | QPushButton
textEdit | QTextEdit
textEdit_2 | QTextEdit

Filter
pushButton_4 : QPushButton
Property | Value
QObject
objectName | pushButton_4
QWidget

Application Output
transformation
QImage::setPixel: coordin
QImage::setPixel: coordin
QImage::setPixel: coordin

Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.
Take UI Tour
Do Not Show Again



Draw

Clear

TX 2

Ty 2

Scale

Sx 5

Sy 5

Translate

Angle 30

Rotate

lab

Display Widgets

Label

Application Output

transformation

QImage::setPixel: coordin

QImage::setPixel: coordin

QImage::setPixel: coordin

Filter

Object	Class
MainWindow	QMainWindow
centralwidget	QWidget
label	QLabel
label_2	QLabel
label_3	QLabel
label_4	QLabel
label_5	QLabel
label_6	QLabel
pushButton	QPushButton
pushButton_2	QPushButton
pushButton_3	QPushButton
pushButton_4	QPushButton
pushButton_5	QPushButton
textEdit	QTextEdit
textEdit_2	QTextEdit

Filter

pushButton_4 : QPushButton

Property	Value
QObject	
objectName	pushButton_4
QWidget	

Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.

Take UI Tour

Do Not Show Again