Name:	Mustafa Chowdhury
Student access ID:	Ge3306
Date (MM/DD/YYYY):	04/12/2019
Group Number (if any):	2

Title of the change request	
Sources:	**Source#1:** https://doc.qt.io/qt-5.11/qpointf.html **Source#2:** https://doc.qt.io/archives/qt-4.8/qpolygon.html

1. Change Request and concepts*:* (10 points)

Change Request: "Add a new tool to draw regular or irregular Pentagons. After writing the pentagon, it should automatically stamp user specified name in the middle of that pentagon with a font size appropriate to the size of the pentagon drawn. The user should be able to choose regular or irregular pentagon. This new tool should work exactly like any other tools in the easyPaint application. You must use proper icons and shortcut for this selection instrument."

From the change request, I identified following concept:

1. Add
2. Tool
3. Draw
4. regular or irregular pentagon
5. Instrument

Description:

1) Add: is used to communicate with programmer. So, it is an irrelevant concept.
2) Tool: some form of the tool already implemented into this program, therefore it is a relevant concept.
3) Draw: some form of the tool already implemented into this program, therefore it is a relevant concept.
4) Regular or irregular pentagon: is the new concept that need to be implemented, therefore it is an external concept.
5) Instrument: some form of the tool already implemented into this program, therefore it is a relevant concept.

Table 1 Significant Concepts and their details

SN#	Concept Name	Details of how your extracted this concept.
CON1	Instrument	Some form of the tool already implemented into this program, therefore it is a relevant concept.
CON2	Tool	Some form of the tool already implemented into this program, therefore it is a relevant concept.
CON3	Draw	Some form of the tool already implemented into this program,

		therefore it is a relevant concept.

2. Functional requirements: (10 points)

Table 2 Functional Requirements

Requirement#	Functional Requirement Details
FR1	Regular and Irregular pentagon icons shall be displayed in tool bar area.
FR2	Based on user selection, the pentagon shape shall draw on the canvas.
FR3	User can save a specified text inside of the pentagon after finished drawing.

3. Concept Location:

.

Methodology: (5 points)

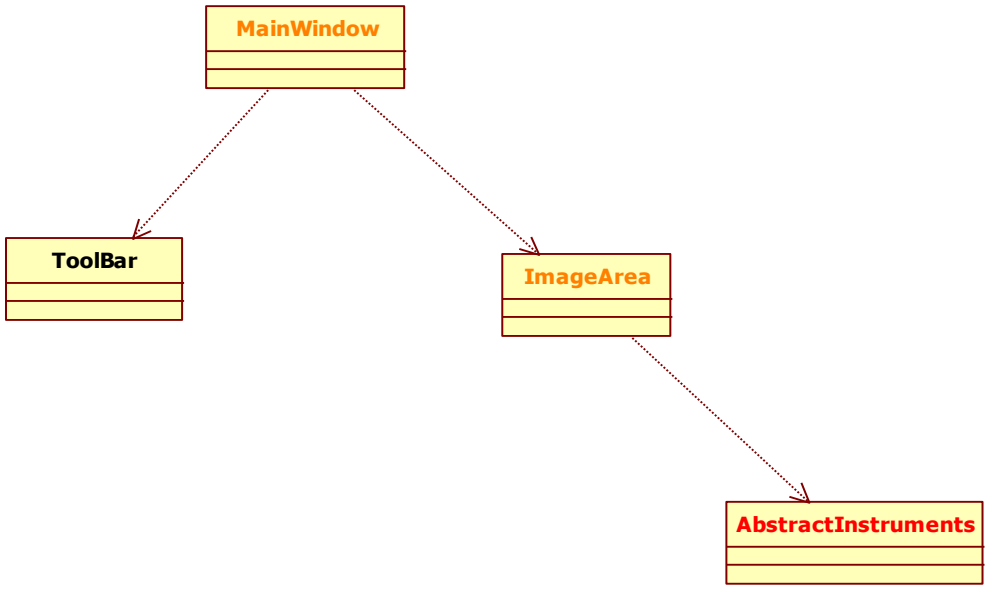
I started using dependency search from MainWindow and looking for all the supplier slice of MainWindow to check for concept location. I keep looking for supplier, supplier slice to get the concept location. I found the Instrument classes that uses for already defined rectangle instrument. Because, the change request demands two completely new instrument functionality, I crated two instrument class derived from AbstractInstrument class.

3.1 Dependency Search (Use this section if you have used dependency search) (7 points)

Table 3 if Dependency Search is used

Class/file name	Tool used	Mark	Explanation
MainWindow	Go to Definition/Declaration	Propagating	Staring point for searching significant concept, but MainWindow doesn't have the concept location.
Toolbar	Go to Definition/Declaration	Unchanged	No concept related location is found. So, go back to the check mainWindow's another supplier
ImageArea	Go to Definition/Declaration	Propagating	AbstractInstrument is the supplier of ImageArea, therefore ImageArea just propagating location
AbstractInstrument	Go to Definition/Declaration	Located	AbstractInstrument is client of some predefined instrument class. Therefore, concept location is found. Need to be incorporating new supplier classes to fulfill the responsibilities of the client AbstractInstrument for this change request.

Partial Class Dependency Graph (UML): (3 points)



3.2 Grep Search (Use this section if you have used grep search) (10 points)

Table 4 If Grep Search is used

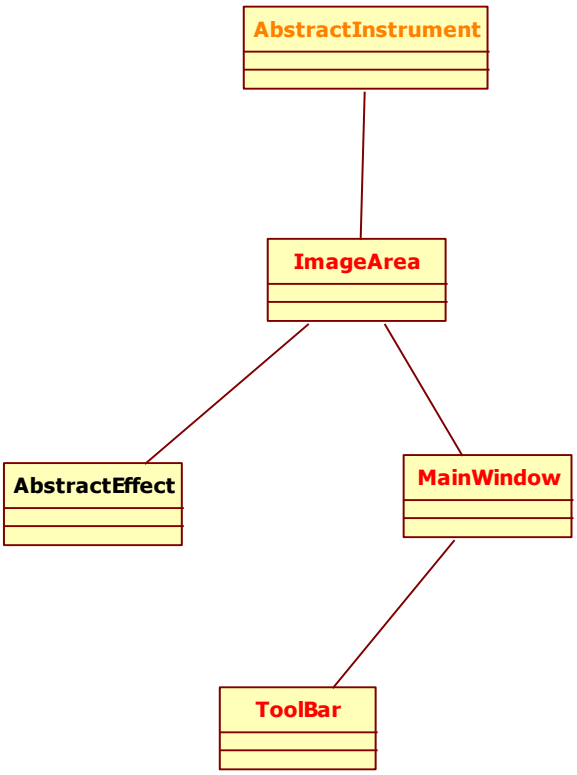
Concept	Query	#Results	Target class/file	Tool used	Explanation

4. Impact Analysis: (10 points)

Table 5 The list of all the classes visited during impact analysis

Class name	Tool used	Mark	Explanation
AbstractInstrument	Go to Definition/ Declaration	Propagating	New supplier class is created, but this class just propagating information to other class
ImageArea	View all references	Impacted	Instrument handler need to be updated to handle the instrument, therefore, ImageArea will be impacted by the change.
AbstractEffect	View all references	Unchanged	Update is not needed.
MainWindow	View all references	Impacted	Instrument icon and action need to implement in this class.
ToolBar	View all references	Impacted	Tool button for new instrument need to be implemented.

Partial class interaction graph (use starUML): (5 points)



5. Prefactoring: (5 points)

Table 6 Prefactoring Code Files

File Name	Refactoring Issue	Lines of Code		
		Added	Deleted	Total

N/A

6. Actualization: (10 points)

Table 7 Actualization Summary

Code Files					
Visited#	**Changed#**	**Added#**	**Propagating#**	**Unchanged#**	**Added to Changed Set#**
5	3	3	1	1	3

Table 8 Actualization Code Files

File Name	Task	Lines of Code		
		Added	Deleted	Total
Regularpentagoninstrument.h	Added a new class derived from AbstractInstrument class with new functionalities	38	0	38
RegularPentagonInstrument.cpp	Implemented functionalities of header file	172	0	172
Irregularpentagoninstrument.h	Added a new class derived from AbstractInstrument class with new functionalities	38	0	38
IrregularPentagonInstrument.cpp	Added a new class derived from AbstractInstrument class	168	0	168
ImageArea.cpp	Instrument handler for regular and irregular pentagon is added.	11	0	11
MainWindow.cpp	Action of regular and irregular pentagon is	15	0	15

	updated			
Toolbar.h	Variable for the pentagon created	1	0	1
ToolBar.cpp	Layout for the regular and irregular pentagon's icon implemented	7	0	7
EasyPaintEnumType.h	Enum type for regular and irregular pentagon is added	3	0	3
Resource.qrc	Image for icon is added	2	0	2

7. Postfactoring: (5 points)

Table 9 Postfactoring Code Files

File Name	Refactoring Issue	Lines of Code		
		Added	Deleted	Total

N/A

8. Verification: (15 points)

Only functional testing is performed to check for the functionality of two new instrument I added.

Table 10 Statement Verification Summary

File Name	Coverage of Application			Unit Test Failed (Just indicate the SN#)	Bugs Found
	Total statements added	Total statements covered	Statement coverage %		
Example: TestMe.cpp	7	5	71%	UT#1	0

Unit Test Case Details:

N/A

Functional Test Case Details:

After implementing the change request, I draw the regular and irregular pentagon and filled with color red. The space is successfully drawn and filled with color. Also, as soon as I finished the drawing, a dialogue box asked for an input text. At first, I canceled it, therefore no text is saved inside of the pentagon. Afterward, I drew the pentagon again but save a text as my name "mustafa" and its worked.

I also check for the old functionality after modification. And all the other functions working properly and does not impacted by the change.

9. Highlighted Source Code: (10 points)

RegularPentagonInstrument

```
// mustafa
// regular pentagon
#ifndef RegularPentagonInstrument_H
#define RegularPentagonInstrument_H

#include "abstractinstrument.h"

#include <QtCore/QObject>

/**
* @brief Concave Pentagon instrument class.
*
*/
class RegularPentagonInstrument : public AbstractInstrument
{
	Q_OBJECT
public:
	explicit RegularPentagonInstrument(QObject *parent = 0);

	void mousePressEvent(QMouseEvent *event, ImageArea &imageArea);
	void mouseMoveEvent(QMouseEvent *event, ImageArea &imageArea);
	void mouseReleaseEvent(QMouseEvent *event, ImageArea &imageArea);
	void textDialog(ImageArea &imageArea);
	void paintText(ImageArea &imageArea);

protected:
	void paint(ImageArea &imageArea, bool isSecondaryColor = false, bool
additionalFlag = false);
private:
	QString mText;
signals:
	void sendCloseTextDialog();
private slots:
	void updateText(ImageArea *imageArea, QString textString);
	void cancel(ImageArea *imageArea);

};

#endif
// Mustafa
// regular pentagon
#include "RegularPentagonInstrument.h"
#include "../imagearea.h"
#include "../datasingleton.h"

#include <QPen>
#include <QPainter>
#include <QPolygon>
#include <QVector>
#include <QPointF>
#include "../dialogs/textdialog.h"

RegularPentagonInstrument::RegularPentagonInstrument(QObject *parent) :
```

```
        AbstractInstrument(parent)
{
}
void RegularPentagonInstrument::mousePressEvent(QMouseEvent * event, ImageArea &
imageArea)
{
        if (event->button() == Qt::LeftButton || event->button() == Qt::RightButton)
        {
                mStartPoint = mEndPoint = event->pos();
                imageArea.setIsPaint(true);
                mImageCopy = *imageArea.getImage();
                makeUndoCommand(imageArea);
        }
}

void RegularPentagonInstrument::mouseMoveEvent(QMouseEvent * event, ImageArea & imageArea)
{
        if (imageArea.isPaint())
        {
                mEndPoint = event->pos();
                imageArea.setImage(mImageCopy);
                if (event->buttons() & Qt::LeftButton)
                {
                        paint(imageArea, false);
                }
                else if (event->buttons() & Qt::RightButton)
                {
                        paint(imageArea, true);
                }
        }
}

void RegularPentagonInstrument::mouseReleaseEvent(QMouseEvent * event, ImageArea &
imageArea)
{
        if (imageArea.isPaint())
        {
                textDialog(imageArea);
                imageArea.setImage(mImageCopy);
                if (event->button() == Qt::LeftButton)
                {
                        paint(imageArea, false);
                }
                else if (event->button() == Qt::RightButton)
                {
                        paint(imageArea, true);
                }
                imageArea.setIsPaint(false);
        }
}

//: Text Dialog pop up
void RegularPentagonInstrument::textDialog(ImageArea & imageArea)
{
        TextDialog *td = new TextDialog(mText, &imageArea);
        connect(td, SIGNAL(textChanged(ImageArea *, QString)), this,
SLOT(updateText(ImageArea *, QString)));
        connect(this, SIGNAL(sendCloseTextDialog()), td, SLOT(accept()));
```

```
        connect(td, SIGNAL(canceled(ImageArea *)), this, SLOT(cancel(ImageArea *)));
        td->setAttribute(Qt::WA_DeleteOnClose);
        td->show();
}

//: Paints text on the screen
void RegularPentagonInstrument::paintText(ImageArea & imageArea)
{
        QPainter painter(imageArea.getImage());
        painter.setPen(QPen(DataSingleton::Instance()->getPrimaryColor()));
        painter.setFont(DataSingleton::Instance()->getTextFont());
        painter.drawText(QRect(mStartPoint, mEndPoint), mText);
        painter.end();
        imageArea.setEdited(true);
        imageArea.update();
}

//: Paints the pentagon on the screen
void RegularPentagonInstrument::paint(ImageArea & imageArea, bool isSecondaryColor, bool
additionalFlag)
{
        QPainter painter(imageArea.getImage());
        painter.setPen(QPen(DataSingleton::Instance()->getPrimaryColor(),
                DataSingleton::Instance()->getPenSize() * imageArea.getZoomFactor(),
                Qt::SolidLine, Qt::RoundCap, Qt::RoundJoin));
        if (isSecondaryColor)
        {
                painter.setBrush(QBrush(DataSingleton::Instance()->getSecondaryColor()));
        }
        if (mStartPoint != mEndPoint)
        {
                // get the sizw
                int size = mEndPoint.x() - mStartPoint.x();
                //get half size
                int hSize = (size / 2);

                // pentagon have six point
                QPointF point[6];

                //for 1st point is not updated
                point[0].setX(mStartPoint.x());
                point[0].setY(mStartPoint.y());

                //for 2nd point x move from left and y goes down
                //therefore, an angle shape c
                point[1].setX(point[0].x() + size);
                point[1].setY(point[0].y() + size);

                //for 3rd point x moves half size to left and the y goes down
                point[2].setX(point[1].x() - hSize);
                point[2].setY(point[1].y() + size);

                //for 4th point x moves left from point 3 and the y doest not move
                point[3].setX(point[2].x() - size);
                point[3].setY(point[2].y());

                // for 5th point x moves half size from point 4 and the y goes up
                point[4].setX(point[3].x() - hSize);
```

```
                point[4].setY(point[1].y());

                //for 6th upadted sames point 1
                point[5].setX(point[0].x());
                point[5].setY(point[0].y());

                // create polygon
                QPolygon poly;
                poly << point[0].toPoint();
                poly << point[1].toPoint();
                poly << point[2].toPoint();
                poly << point[3].toPoint();
                poly << point[4].toPoint();
                poly << point[5].toPoint();

                // set point for polygon
                poly.setPoint(0, point[0].toPoint());

                // draw the polygon
                QPainterPath path;
                path.addPolygon(QPolygon(poly));
                painter.drawPath(path);

        }
        imageArea.setEdited(true);

        painter.end();

        imageArea.update();
}

//get user text after shape drew
void RegularPentagonInstrument::updateText(ImageArea *imageArea, QString textString)
{
        mText = textString;
        imageArea->setImage(mImageCopy);
        paintText(*imageArea);
        paint(*imageArea);
}

//if user cancel the dialog box
void RegularPentagonInstrument::cancel(ImageArea * imageArea)
{
        mText = QString();
}
```

IrregularPentagonInstrument

```
// Mustafa
// irregular pentagon

#ifndef IrregularPentagonInstrument_H
#define IrregularPentagonInstrument_H

#include "abstractinstrument.h"

#include <QtCore/QObject>

/**
* @brief Concave Pentagon instrument class.
*
*/
class IrregularPentagonInstrument : public AbstractInstrument
{
	Q_OBJECT
public:
	explicit IrregularPentagonInstrument(QObject *parent = 0);

	void mousePressEvent(QMouseEvent *event, ImageArea &imageArea);
	void mouseMoveEvent(QMouseEvent *event, ImageArea &imageArea);
	void mouseReleaseEvent(QMouseEvent *event, ImageArea &imageArea);
	void textDialog(ImageArea &imageArea);
	void paintText(ImageArea &imageArea);

protected:
	void paint(ImageArea &imageArea, bool isSecondaryColor = false, bool
additionalFlag = false);
private:
	QString mText;
signals:
	void sendCloseTextDialog();
private slots:
	void updateText(ImageArea *imageArea, QString textString);
	void cancel(ImageArea *imageArea);

};

#endif

// mustafa
// irregular pentagon
#include "IrregularPentagonInstrument.h"
#include "../imagearea.h"
#include "../datasingleton.h"

#include <QPen>
#include <QPainter>
#include "../dialogs/textdialog.h"

IrregularPentagonInstrument::IrregularPentagonInstrument(QObject *parent) :
	AbstractInstrument(parent)
{
}
```

```
void IrregularPentagonInstrument::mousePressEvent(QMouseEvent * event, ImageArea &
imageArea)
{
	if (event->button() == Qt::LeftButton || event->button() == Qt::RightButton)
	{
		mStartPoint = mEndPoint = event->pos();
		imageArea.setIsPaint(true);
		mImageCopy = *imageArea.getImage();
		makeUndoCommand(imageArea);
	}
}

void IrregularPentagonInstrument::mouseMoveEvent(QMouseEvent * event, ImageArea &
imageArea)
{
	if (imageArea.isPaint())
	{
		mEndPoint = event->pos();
		imageArea.setImage(mImageCopy);
		if (event->buttons() & Qt::LeftButton)
		{
			paint(imageArea, false);
		}
		else if (event->buttons() & Qt::RightButton)
		{
			paint(imageArea, true);
		}
	}
}

void IrregularPentagonInstrument::mouseReleaseEvent(QMouseEvent * event, ImageArea &
imageArea)
{
	if (imageArea.isPaint())
	{
		textDialog(imageArea);
		imageArea.setImage(mImageCopy);
		if (event->button() == Qt::LeftButton)
		{
			paint(imageArea, false);
		}
		else if (event->button() == Qt::RightButton)
		{
			paint(imageArea, true);
		}
		imageArea.setIsPaint(false);
	}
}

//: Text Dialog pop up
void IrregularPentagonInstrument::textDialog(ImageArea & imageArea)
{
	TextDialog *td = new TextDialog(mText, &imageArea);
	connect(td, SIGNAL(textChanged(ImageArea *, QString)), this,
SLOT(updateText(ImageArea *, QString)));
	connect(this, SIGNAL(sendCloseTextDialog()), td, SLOT(accept()));
	connect(td, SIGNAL(canceled(ImageArea *)), this, SLOT(cancel(ImageArea *)));
	td->setAttribute(Qt::WA_DeleteOnClose);
```

```
	td->show();
}

//: Paints text on the screen
void IrregularPentagonInstrument::paintText(ImageArea & imageArea)
{
	QPainter painter(imageArea.getImage());
	painter.setPen(QPen(DataSingleton::Instance()->getPrimaryColor()));
	painter.setFont(DataSingleton::Instance()->getTextFont());
	painter.drawText(QRect(mStartPoint, mEndPoint), mText);
	painter.end();
	imageArea.setEdited(true);
	imageArea.update();
}

//: Paints the pentagon on the screen
void IrregularPentagonInstrument::paint(ImageArea & imageArea, bool isSecondaryColor,
bool additionalFlag)
{
	QPainter painter(imageArea.getImage());
	painter.setPen(QPen(DataSingleton::Instance()->getPrimaryColor(),
		DataSingleton::Instance()->getPenSize() * imageArea.getZoomFactor(),
		Qt::SolidLine, Qt::RoundCap, Qt::RoundJoin));
	if (isSecondaryColor)
	{
		painter.setBrush(QBrush(DataSingleton::Instance()->getSecondaryColor()));
	}
	if (mStartPoint != mEndPoint)
	{
		// pentagon have six point
		QPointF point[6];
		// get the size
		int size = mEndPoint.rx() - mStartPoint.rx();

		// 1st point from where drawing will start
		point[0].setX(mStartPoint.x());
		point[0].setY(mStartPoint.y());

		// 2nd point x goes right from point 0 and the y goes down
		// therefore, angle shape created
		point[1].setX(mStartPoint.x() + size);
		point[1].setY(mStartPoint.y() + size);

		//3rd point x stay as same point and y goes down straight
		point[2].setX(point[1].x());
		point[2].setY(point[1].y() + size);

		//4t point x moves from point 2 to left and y point to same as x
		// therefore straight line will be created from point 3 to 4
		point[3].setX(point[2].x() - (size+size));
		point[3].setY(point[2].x());

		// 5th point x doesn't updated and y goes up from 3
		// there a straigh vertical line created
		point[4].setX(point[3].x());
		point[4].setY(point[1].y());

		//6th point is same as point 1
```

```
                point[5].setX(point[0].x());
                point[5].setY(point[0].y());

                // create a QPolygon object
                QPolygon poly;
                poly << point[0].toPoint();
                poly << point[1].toPoint();
                poly << point[2].toPoint();
                poly << point[3].toPoint();
                poly << point[4].toPoint();
                poly << point[5].toPoint();

                // setup the set point
                poly.setPoint(5, point[0].toPoint());

                //draw the polygo
                QPainterPath path;
                path.addPolygon(QPolygon(poly));
                painter.drawPath(path);
        }

        imageArea.setEdited(true);

        painter.end();

        imageArea.update();
}

// set text for shape by showing textDialoug
void IrregularPentagonInstrument::updateText(ImageArea *imageArea, QString textString)
{
        mText = textString;
        imageArea->setImage(mImageCopy);
        paintText(*imageArea);
        paint(*imageArea);
}

// if user cancel text dialoug
void IrregularPentagonInstrument::cancel(ImageArea * imageArea)
{
        mText = QString();
}
```

Rest of the modification is showed by git diff:

```
musta@DESKTOP-ST5QVK3 MINGW64
~/Desktop/CurrentVersion/csc4111w19grp2/Project (master)
$ git diff
diff --git a/Project/CMakeLists.txt b/Project/CMakeLists.txt
index ded8860..246df9e 100644
--- a/Project/CMakeLists.txt
+++ b/Project/CMakeLists.txt
@@ -55,7 +55,9 @@ set (HEADERS
     sources/instruments/curvelineinstrument.h
     sources/instruments/textinstrument.h
     sources/instruments/convexpentagoninstrument.h
+    sources/instruments/irregularpentagoninstrument.h
+    sources/instruments/regularpentagoninstrument.h)

 #------- sources --------
 set (SOURCES
@@ -99,7 +101,9 @@ set (SOURCES
     sources/instruments/curvelineinstrument.cpp
     sources/instruments/textinstrument.cpp
     sources/instruments/concavepentagoninstrument.cpp
+    sources/instruments/irregularpentagoninstrument.cpp
+    sources/instruments/regularpentagoninstrument.cpp)
:...skipping...
diff --git a/Project/CMakeLists.txt b/Project/CMakeLists.txt
index ded8860..246df9e 100644
--- a/Project/CMakeLists.txt
+++ b/Project/CMakeLists.txt
@@ -55,7 +55,9 @@ set (HEADERS
     sources/instruments/curvelineinstrument.h
     sources/instruments/textinstrument.h
     sources/instruments/convexpentagoninstrument.h
+    sources/instruments/irregularpentagoninstrument.h
+    sources/instruments/regularpentagoninstrument.h)

 #------- sources --------
 set (SOURCES
@@ -99,7 +101,9 @@ set (SOURCES
     sources/instruments/curvelineinstrument.cpp
     sources/instruments/textinstrument.cpp
     sources/instruments/concavepentagoninstrument.cpp
+    sources/instruments/irregularpentagoninstrument.cpp
+    sources/instruments/regularpentagoninstrument.cpp)

 #------- resources -------
 set (RESOURCE_PATH
diff --git a/Project/sources/easypaintenums.h
b/Project/sources/easypaintenums.h
index 1323675..a506132 100644
--- a/Project/sources/easypaintenums.h
+++ b/Project/sources/easypaintenums.h
@@ -45,6 +45,11 @@ typedef enum
         //JUSTIN: Added enum for new pentagon instruments
         CONVEXPENTAGON,
         CONCAVEPENTAGON,
+
```

```
+       // mustafa
+       REGULARPENTAGON,
+       IRREGULARPENTAGON,
+
     ELLIPSE,
     CURVELINE,
     TEXT,
diff --git a/Project/sources/imagearea.cpp
b/Project/sources/imagearea.cpp
index f08cb07..09cc4c5 100644
--- a/Project/sources/imagearea.cpp
+++ b/Project/sources/imagearea.cpp
@@ -42,6 +42,9 @@
 #include "instruments/selectioninstrument.h"
 #include "instruments/curvelineinstrument.h"
 #include "instruments/textinstrument.h"
+#include "instruments/regularpentagoninstrument.h"
+#include "instruments/irregularpentagoninstrument.h"
+
 #include "dialogs/resizedialog.h"

 #include "effects/abstracteffect.h"
@@ -155,6 +158,12 @@ ImageArea::ImageArea(const bool &isOpen, const
QString &filePath, QWidget *paren
        mInstrumentsHandlers[CONVEXPENTAGON] = new
ConvexPentagonInstrument(this);
        mInstrumentsHandlers[CONCAVEPENTAGON] = new
ConcavePentagonInstrument(this);

+       //Mustafa-> regular & irregular instrument halder
+       mInstrumentsHandlers[REGULARPENTAGON] = new
RegularPentagonInstrument(this);
+       mInstrumentsHandlers[IRREGULARPENTAGON] = new
IrregularPentagonInstrument(this);
+
+       //
+
     // Effects handlers
     mEffectsHandlers.fill(0, (int)EFFECTS_COUNT);
     mEffectsHandlers[NEGATIVE] = new NegativeEffect(this);
@@ -486,6 +495,8 @@ void ImageArea::restoreCursor()
        case RECTANGLE: case ELLIPSE: case LINE: case CURVELINE: case
TEXT:
                //JUSTIN: Added concave and convex pentagon cursor
movement
        case CONCAVEPENTAGON: case CONVEXPENTAGON:
+               // Mustafa
+       case REGULARPENTAGON: case IRREGULARPENTAGON:
         mCurrentCursor = new QCursor(Qt::CrossCursor);
         setCursor(*mCurrentCursor);
         break;
@@ -513,6 +524,8 @@ void ImageArea::drawCursor()
                //JUSTIN: added convex and concave pentagon draw
cursor movements
        case FILL: case RECTANGLE: case CONCAVEPENTAGON: case
CONVEXPENTAGON: case ELLIPSE:
        case CURSOR: case INSTRUMENTS_COUNT: case CURVELINE: case TEXT:
+               //MUSTAFA
```

```
+       case REGULARPENTAGON: case IRREGULARPENTAGON:
         break;
     case PEN: case ERASER:
         mPixmap->fill(QColor(0, 0, 0, 0));
@@ -525,6 +538,8 @@ void ImageArea::drawCursor()
     case FILL: case RECTANGLE: case ELLIPSE: case CURSOR: case
INSTRUMENTS_COUNT:
                 //JUSTIN: Added convex and concave pentagon cursor
movements
        case CURVELINE: case TEXT: case CONVEXPENTAGON: case
CONCAVEPENTAGON:
+               //mustafa
+       case REGULARPENTAGON: case IRREGULARPENTAGON:
         break;
     case PEN:
         if(mRightButtonPressed)
diff --git a/Project/sources/mainwindow.cpp
b/Project/sources/mainwindow.cpp
index ef7b5ce..aed32ae 100644
--- a/Project/sources/mainwindow.cpp
+++ b/Project/sources/mainwindow.cpp
@@ -341,6 +341,24 @@ void MainWindow::initializeMainMenu()
        mInstrumentsMenu->addAction(mConcavePentagonAction);
        mInstrumentsActMap.insert(CONCAVEPENTAGON,
mConcavePentagonAction);

+       // Mustafa, Regular & Irregular instrument added below
+       QAction *mRegularPentagonAction = new QAction(tr("Regular
Pentagon"), this);
+       mRegularPentagonAction->setCheckable(true);
+       mRegularPentagonAction->setIcon(QIcon(":/media/instruments-
icons/regularPentagon.png"));
+       connect(mRegularPentagonAction, SIGNAL(triggered(bool)), this,
SLOT(instumentsAct(bool)));
+       //
+       mInstrumentsMenu->addAction(mRegularPentagonAction);
+       mInstrumentsActMap.insert(REGULARPENTAGON,
mRegularPentagonAction);
+
+       QAction *mIrregularPentagonAction = new QAction(tr("Irregular
Pentagon"), this);
+       mIrregularPentagonAction->setCheckable(true);
+       mIrregularPentagonAction->setIcon(QIcon(":/media/instruments-
icons/irregularPentagon.png"));
+       connect(mIrregularPentagonAction, SIGNAL(triggered(bool)),
this, SLOT(instumentsAct(bool)));
+       mInstrumentsMenu->addAction(mIrregularPentagonAction);
+       mInstrumentsActMap.insert(IRREGULARPENTAGON,
mIrregularPentagonAction);
+
+
+
        //01/06/2019 Macam added for CSC4110/CSC4111 project/change
request
        mSelInstruMenu = menuBar()->addMenu(tr("Selection
Instruments"));
```

```
diff --git a/Project/sources/resources.qrc
b/Project/sources/resources.qrc
index 912be09..6ab45f9 100644
--- a/Project/sources/resources.qrc
+++ b/Project/sources/resources.qrc
@@ -17,6 +17,8 @@
         <file>media/instruments-icons/cursor_fill.png</file>
         <file>media/instruments-icons/convexPentagon.png</file>
         <file>media/instruments-icons/concavePentagon.png</file>
+        <file>media/instruments-icons/regularPentagon.png</file>
+        <file>media/instruments-icons/irregularPentagon.png</file>
         <file>media/textures/transparent.jpg</file>
         <file>media/actions-icons/application-exit.png</file>
         <file>media/actions-icons/document-new.png</file>
diff --git a/Project/sources/widgets/toolbar.cpp
b/Project/sources/widgets/toolbar.cpp
index c1b0f53..987abce 100644
--- a/Project/sources/widgets/toolbar.cpp
+++ b/Project/sources/widgets/toolbar.cpp
@@ -65,6 +65,10 @@ void ToolBar::initializeItems()
        //JUSTIN: added new tool button declarations for concave and
convez pentagons
        mConcavePentagonButton =
createToolButton(mActMap[CONCAVEPENTAGON]);
        mConvexPentagonButton =
createToolButton(mActMap[CONVEXPENTAGON]);
+       //Mustafa
+       mRegularPentagonButton =
createToolButton(mActMap[REGULARPENTAGON]);
+       mIrregularPentagonButton =
createToolButton(mActMap[IRREGULARPENTAGON]);
+
     mEllipseButton = createToolButton(mActMap[ELLIPSE]);
     mCurveButton = createToolButton(mActMap[CURVELINE]);
     mTextButton = createToolButton(mActMap[TEXT]);
@@ -86,6 +90,10 @@ void ToolBar::initializeItems()
        bLayout->addWidget(mConcavePentagonButton, 6, 0);
        bLayout->addWidget(mConvexPentagonButton, 6, 1);

+       //mustafa
+       bLayout->addWidget(mRegularPentagonButton, 7, 0);
+       bLayout->addWidget(mIrregularPentagonButton, 7, 1);
+
     QWidget *bWidget = new QWidget();
     bWidget->setLayout(bLayout);

diff --git a/Project/sources/widgets/toolbar.h
b/Project/sources/widgets/toolbar.h
index 5805333..09773a0 100644
--- a/Project/sources/widgets/toolbar.h
+++ b/Project/sources/widgets/toolbar.h
@@ -65,7 +65,9 @@ private:
     QToolButton *mCursorButton, *mEraserButton, *mPenButton,
*mLineButton,
                 *mColorPickerButton, *mMagnifierButton, *mSprayButton,
*mFillButton,
                 *mRectangleButton, *mEllipseButton, *mCurveButton,
*mTextButton,
```

```
                                // mustafa
+                                  *mRegularPentagonButton,
*mIrregularPentagonButton;
     ColorChooser *mPColorChooser, *mSColorChooser;
     bool mPrevInstrumentSetted;
     const QMap<InstrumentsEnum, QAction*> &mActMap;
```