



Report CPIT-380

Almas Nogali & Somaya Nimatallah

Section: VAR

Group No. 1



Table of Contents


| | |
|--------------------|----|
| Introduction..... | 3 |
| Text Part | 4 |
| Picture Part | 6 |
| Sound Part | 19 |
| Move Part | 24 |
| Bunes part..... | 29 |
| Conclusion | 30 |





Introduction

Recently, programs and applications - which depend in their presentation of diverse knowledge and experiences on the integration and integration of two or more sensory media in an educational environment - have become one of the recent trends in achieving multiple educational outcomes, and these media often include written text, audio or still images. Or illustrations, animation, maps, etc. Therefore, it is important for the specialists of the Information Technology Department to be familiar with a good amount of information about multimedia. This report presents all the data on the way to solve all the tasks required to solve the requirements of the multimedia technology project.



Text Part



We Create web pages using HTML. Contain three web pages:

- 1- First web page contains Welcome phrase and two links.
- 2- Second web page contain Information about TTF.
- 3- Third web page contain the main stages of Image processing.

**We use link and picture tags and ordered and unordered lists. Also use boldface and block quote tags, and `
` and many more tags to recover all tags that we learned.**

- **Coding part:**

- 1- Home page:**

```
<html>
<head>

<title> Somaya and Almas Project (First part) </title>

</head>
<body>

<h1> <blockquote> <blockquote>WELCOME TO MY PAGE </blockquote></blockquote></h1>
<img src= "1.png" alt= "Welcome pic" >

<ul>

<li><b>TTF </b></li>
<br/>
<a href="file:///C:/part(1)_HTML/TTF-Page2.html"> More Information about TTF.</a>
<br/><br/><br/>

<li> <b>Stages Of Image Processing </b> </li>

<br/>
<a href="file:///C:/part(1)_HTML/Stages-Page3.html"> More Information about stages of image processing.</a>
<br/><br/><br/>

</ul>

</body>
</html>
```

- 2- TTF info. Page:**

```

<html>
<head>

<title> Somaya and Almas Project (First part) TTF Page </title>

</head>
<body>

<br/><br/>
<blockquote><blockquote><a href="file:///C:/part{1}_HTML/Almas,%20Somaya_HTML_PROJECT.html">
<br/><b> Home Page </b></a></blockquote></blockquote>
<br/><br/>

<ol>

<li><b> What is TTF? </b></li>
<br/>
<ul>
<li><b>A file with .ttf extension represents font files based on the TrueType
specifications font technology.</li>
<br/><li><b>it was initially designed and launched by Apple Computer, Inc for Mac OS and was later adopted by Microsoft for Windows OS.</li>
<br/><li><b>TrueType fonts provide highest quality display on computer screens and printers without any dependency on resolution.
All modern applications using fonts are able to work with TTF files.</li>
<br/><li><b>TTF font files are freely available over the internet.</li>
<br/><li><b>A TrueType font file is a binary file that consists of a sequence of concatenated tables,
Each table is a sequence of words and has a name known as Tag,
Each Tag is of uint 32 data type and consists of four characters.
The first table in the file is font directory that gives access to other tables in the font file.
Font data is contained in other tables followed after the font directory table. Since each table is
accessible by its tag, the tables can appear in any
order in the file.</li>
<br/><br/>
</ul>

<img src= "2.png" alt= "ttf taple example" >
<br/><br/><br/><br/>

<li><b> How Can We Create Our Own Font? </b></li>
<br/>
<ol>
<li><b>Create a brief for your font design.</li>
<li><b>Make your fundamental font design choices.</li>
<li><b>Start your font design from scratch.</li>
<li><b>Try designing fonts by hand.</li>
<li><b>Use control characters for your font design.</li>
<li><b>Move to your computer.</li>
<li><b>Choose your software.</li>
<li><b>Draw some letters.</li>
<li><b>Switch to text view mode.</li>
<li><b>Test your font design at line level.</li>
<li><b>Study other typefaces.</li>
<li><b>Scale your font design down.</li>
<li><b>Print your font design.</li>
<li><b>Add special characters to your font design.</li>
<li><b>Explore different styles, weights and widths.</li>
<li><b>Take your font design global.</li>
<li><b>Put your font design to the test.</li>

</ol>

</ol>

</body>
</html>

```

3- Main stages of Image processing page:

```

|
<html>
<head>

<title> Somaya and Almas Project (First part) </title>

</head>
<body>

<br/><br/>
<blockquote><blockquote><a href="file:///C:/part{1}_HTML/Almas,%20Somaya_HTML_PROJECT.html"><b> Home Page </b></a></blockquote></blockquote>
<br/><br/>

<b>Math are the Stages Of Image Processing </b>
<ol>
<li><b>Problem Domain</li>
<li><b>Image Acquisition</li>
<li><b>Image Enhancement</li>
<li><b>Image Restoration</li>
<li><b>Morphological Processing</li>
<li><b>Segmentation</li>
<li><b>Object Recognition</li>
<li><b>Representation & Description</li>
</ol>
<blockquote><li> Additional Processes </li></blockquote>
<ul>
<li><b>Image Compression</li>
<li><b>Colour Image Processing</li>
</ul>
<br/>
<img src= "3.png" alt= "Stages OF Image Processing" >
<br/><br/><br/><br/>

</body>
</html>

```

Picture Part



1- Using Method to make the user be able to browse about a picture to perform the filter on it.

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String file = FileChooser.pickAFile();  
    pic1 = new Picture(file);  
    JLabel1.setIcon(new ImageIcon(pic1.getImage()));  
  
    pic2 = new Picture(pic1);  
}
```

2- Color Change by use slider, to change colors in percentages.

We use slider for each color and based on the value of the slider clicked the desired color of the picture will be changed.

• Coding part:

```
private void jSlider1StateChanged(javax.swing.event.ChangeEvent evt) {  
    //change red  
    int red = jSlider1.getValue();  
  
    Pixel[] pix = pic2.getPixels();  
    for (int i = 0; i < pix.length; i++) {  
        pix[i].setRed(red);  
    }  
    JLabel2.setIcon(new ImageIcon(pic2.getImage()));  
}
```

```
private void jSlider2StateChanged(javax.swing.event.ChangeEvent evt) {  
    //change green  
    int green = jSlider2.getValue();  
    Pixel[] pix = pic2.getPixels();  
    for (int i = 0; i < pix.length; i++) {  
        pix[i].setGreen(green);  
    }  
    JLabel2.setIcon(new ImageIcon(pic2.getImage()));  
}
```

```
private void jSlider3StateChanged(javax.swing.event.ChangeEvent evt) {  
    //change blue  
    int blue = jSlider3.getValue();  
    Pixel[] pix = pic2.getPixels();  
    for (int i = 0; i < pix.length; i++) {  
        pix[i].setBlue(blue);  
    }  
    JLabel2.setIcon(new ImageIcon(pic2.getImage()));  
}
```

3- Copy and Crop (To Crop, use event handlers, e.g., using mouse to select a part of image to crop, instead textboxes.)

This method is to take a part of the picture and display it without unselected part. We apply it by using mouse pressed (to take start x, y) and mouse release to take (End x, y) then, display the part that from start point to end point.

- **Coding part:**

- 1- **Pressed part and Release part:**

```
private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {  
    x1 = evt.getX();  
    y1 = evt.getY();  
    jTextField1.setText(Integer.toString(y1));  
    jTextField2.setText(Integer.toString(x1));  
}
```

```
private void jLabel1MouseReleased(java.awt.event.MouseEvent evt) {  
    x2 = evt.getX();  
    y2 = evt.getY();  
    jTextField3.setText(Integer.toString(x2));  
    jTextField4.setText(Integer.toString(y2));  
}
```

- 2- **Coping part from source to target picture:**

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    //Copy and Crop  
  
    double W = (pic1.getWidth() * 1.00 / jLabel1.getWidth());  
    double H = (pic1.getHeight() * 1.00 / jLabel1.getHeight());  
  
    x1 = (int) (W * x1);  
    x2 = (int) (W * x2);  
    y1 = (int) (H * y1);  
    y2 = (int) (H * y2);  
  
    Picture newPic = new Picture(pic1.getWidth(), pic1.getHeight());  
  
    Pixel sourcePixel = null;  
    Pixel targetPixel = null;  
  
    for (int x = x1, tx = x1; x < x2 && x < pic1.getWidth() && tx < pic1.getWidth(); x++, tx++) {  
        for (int y = y1, ty = y1; y < y2 && y < pic1.getHeight() && ty < pic1.getHeight(); y++, ty++) {  
            sourcePixel = pic1.getPixel(x, y);  
            targetPixel = newPic.getPixel(tx, ty);  
            targetPixel.setColor(sourcePixel.getColor());  
        }  
    }  
  
    pic2 = newPic;  
    jLabel2.setIcon(new ImageIcon(pic2.getImage()));  
}
```

4- Convert RGB to Gray Scale Image (Gray scale images have all components same).

We are applying this method by call (grayscale()) method.

- **Coding part:**

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {  
    //GRAYSCALE  
    pic2.grayscale();  
    jLabel12.setIcon(new ImageIcon(pic2.getImage()));  
}
```

5- Convert Gray scale to Binary Image using thresholding technique.

We are applying this method by take value from user and apply the grayscale method on the picture then by going through the picture pixel by pixel and comparing pixel value with thresholding value to assign while or black color on the pixel.

- **Coding part:**

```
private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {  
    //thresholding  
  
    int n = Integer.valueOf(jTextField5.getText()).intValue();  
  
    Pixel pix = null;  
  
    pic2.grayscale();  
  
    for (int i = 0; i < pic1.getWidth(); i++) {  
        for (int j = 0; j < pic1.getHeight(); j++) {  
            pix = pic2.getPixel(i, j);  
  
            if (pix.getAverage() >= n) {  
                pic2.getPixel(i, j).setColor(Color.WHITE);  
            } else {  
                pic2.getPixel(i, j).setColor(Color.BLACK);  
            }  
        }  
    }  
  
    jLabel12.setIcon(new ImageIcon(pic2.getImage()));  
}
```


6- Blending Two images.

We are applying this method by take other picture and take value from the user that represent blending percentage then, call blending method.

- **Coding part:**

- 1- Browse for the second picture:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    //read picture2  
    String file = FileChooser.pickAFile();  
    pic3 = new Picture(file);  
    jLabel12.setIcon(new ImageIcon(pic3.getImage()));  
}
```

- 2- Blending call method:

```
private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {  
    //blending  
    double blindVal = Double.parseDouble(blind.getText());  
    blindVal = blindVal / 100;  
    jLabel2.setIcon(new ImageIcon((pic2.blendPictures(pic3, blindVal)).getImage()));  
}
```

7- Right and Left Rotations by any degree.

We are applying this method by using slider with spicing by (90) that precent each angle, then based on the slider value we will call the rotating methods.

- **Coding part:**

- 1- Right rotation:

```
private void jSlider4StateChanged(javax.swing.event.ChangeEvent evt) {  
    //right rotation  
    if (jSlider4.getValue() == 0 || jSlider4.getValue() == 360 || jSlider4.getValue() == -360) {  
        jLabel2.setIcon(new ImageIcon(pic2.getImage()));  
        pic2 = new Picture(pic1);  
    } else if (jSlider4.getValue() == 90) {  
        jLabel2.setIcon(new ImageIcon((pic2.rotateRight()).getImage()));  
        pic2 = new Picture(pic1);  
    } else if (jSlider4.getValue() == 180) {  
        jLabel2.setIcon(new ImageIcon(pic2.rotateRight().rotateRight().getImage()));  
        pic2 = new Picture(pic1);  
    } else if (jSlider4.getValue() == 270) {  
        jLabel2.setIcon(new ImageIcon(pic2.rotateRight().rotateRight().rotateRight().getImage()));  
        pic2 = new Picture(pic1);  
    } else if (jSlider4.getValue() == -90) {  
        jLabel2.setIcon(new ImageIcon((pic2.rotateLeft()).getImage()));  
        pic2 = new Picture(pic1);  
    } else if (jSlider4.getValue() == -180) {  
        jLabel2.setIcon(new ImageIcon((pic2.rotateLeft().rotateLeft()).getImage()));  
        pic2 = new Picture(pic1);  
    } else if (jSlider4.getValue() == -270) {  
        jLabel2.setIcon(new ImageIcon((pic2.rotateLeft().rotateLeft().rotateLeft()).getImage()));  
        pic2 = new Picture(pic1);  
    }  
}
```

2- Left rotation:

```
private void jSlider5StateChanged(javax.swing.event.ChangeEvent evt) {  
    //left rotation  
    if (jSlider5.getValue() == 0 || jSlider5.getValue() == 360 || jSlider5.getValue() == -360) {  
        jLabel2.setIcon(new ImageIcon(pic2.getImage()));  
        pic2 = new Picture(pic1);  
    } else if (jSlider5.getValue() == -90) {  
        jLabel2.setIcon(new ImageIcon(pic2.rotateRight().getImage()));  
        pic2 = new Picture(pic1);  
    } else if (jSlider5.getValue() == -180) {  
        jLabel2.setIcon(new ImageIcon(pic2.rotateRight().rotateRight().getImage()));  
        pic2 = new Picture(pic1);  
    } else if (jSlider5.getValue() == -270) {  
        jLabel2.setIcon(new ImageIcon(pic2.rotateRight().rotateRight().rotateRight().getImage()));  
        pic2 = new Picture(pic1);  
    } else if (jSlider5.getValue() == 90) {  
        jLabel2.setIcon(new ImageIcon(pic2.rotateLeft().getImage()));  
        pic2 = new Picture(pic1);  
    } else if (jSlider5.getValue() == 180) {  
        jLabel2.setIcon(new ImageIcon(pic2.rotateLeft().rotateLeft().getImage()));  
        pic2 = new Picture(pic1);  
    } else if (jSlider5.getValue() == 270) {  
        jLabel2.setIcon(new ImageIcon(pic2.rotateLeft().rotateLeft().rotateLeft().getImage()));  
        pic2 = new Picture(pic1);  
    }  
}
```

8- Vertical Reflection and horizontal Reflection.

We are applying this method by calling Vertical and Reflection methods.

- **Coding part:**

```
private void jButton12ActionPerformed(java.awt.event.ActionEvent evt) {  
    //vertical reflect  
    jLabel2.setIcon(new ImageIcon(pic2.mirrorAllVertical().getImage()));  
}  
  
private void jButton13ActionPerformed(java.awt.event.ActionEvent evt) {  
    //horizontal reflect  
    jLabel2.setIcon(new ImageIcon(pic2.mirrorAllHorizontal().getImage()));  
}
```

9- Scaling up or down: Use slider, moving left will scale down and moving right will scale up. You may also use mouse click event instead slider. Must use Generalized equations to compute new positions for target x and target y.

We are applying this method by using a slider and call scaleDown and scaleUp based on slider value.

• Coding part:

```
private void jSlider6StateChanged(javax.swing.event.ChangeEvent evt) {  
    //scaling up and down  
    if (jSlider6.getValue() == 0) {  
        jLabel12.setIcon(new ImageIcon(pic2.getImage()));  
        pic2 = new Picture(pic1);  
    } else if (jSlider6.getValue() < 0) {  
        int n = (jSlider6.getValue()) * -1;  
        jLabel12.setIcon(new ImageIcon((pic2.scaleDown(n)).getImage()));  
        pic2 = new Picture(pic1);  
    } else if (jSlider6.getValue() > 0) {  
        jLabel12.setIcon(new ImageIcon((pic2.scaleUp(jSlider6.getValue())).getImage()));  
        pic2 = new Picture(pic1);  
    }  
}
```

10- Computing histograms (By default, you will compute 3 histograms, one for R, one for G and one for B, for gray scale images, there will be only one histogram). Plotting (Visualize) histograms (By default, you will plot 3 histograms, one for R, one for G and one for B)

We are applying this method by count the values of each color from 0 – 255 then compare the arrays that contain counted values to check if the arrays are equal that means the picture is grayscale picture, so we display one char else we display for each color char.

• Coding part:

```

private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
    //histogram
    int[] Rhis = new int[256];
    int[] Ghis = new int[256];
    int[] Bhis = new int[256];

    for (int i = 0; i < pic2.getWidth(); i++) {
        for (int j = 0; j < pic2.getHeight(); j++) {
            for (int h = 0; h < 256; h++) {
                //Red
                if (pic2.getPixel(i, j).getRed() == h) {
                    Rhis[h]++;
                }

                //green
                if (pic2.getPixel(i, j).getGreen() == h) {
                    Ghis[h]++;
                }

                //blue
                if (pic2.getPixel(i, j).getBlue() == h) {
                    Bhis[h]++;
                }
            }
        }
    }

    System.out.println(Arrays.toString(Rhis));
    System.out.println(Arrays.toString(Ghis));
    System.out.println(Arrays.toString(Bhis));
}

```

```

boolean retval = Arrays.equals(Rhis, Ghis);
boolean retval2 = Arrays.equals(Bhis, Ghis);
boolean retval3 = Arrays.equals(Bhis, Rhis);

if (retval == false || retval2 == false || retval3 == false) {
    int maxR = Rhis[0];
    for (int j = 1; j < Rhis.length; j++) {
        if (Rhis[j] > maxR) {
            maxR = Rhis[j];
        }
    }
    int maxG = Ghis[0];
    for (int j = 1; j < Ghis.length; j++) {
        if (Ghis[j] > maxG) {
            maxG = Ghis[j];
        }
    }
    int maxB = Bhis[0];
    for (int j = 1; j < Bhis.length; j++) {
        if (Bhis[j] > maxB) {
            maxB = Bhis[j];
        }
    }
}

```

```

int numR = maxR / 400;
Picture PR = new Picture(500, 600);
int x = 122, y = 500;
Graphics g = PR.getGraphics();
g.setColor(Color.red);
for (int i = 0; i <= 255; i++) {
    g.drawLine(x, y, x, y - (Rhis[i] / numR));
    x += 1;
}
PR.show();

int numG = maxG / 400;
Picture PG = new Picture(500, 600);
int x1 = 122, y1 = 500;
Graphics gG = PG.getGraphics();
gG.setColor(Color.green);
for (int i = 0; i <= 255; i++) {
    gG.drawLine(x1, y1, x1, y1 - (Ghis[i] / numG));
    x1 += 1;
}
PG.show();

int numB = maxR / 400;
Picture PB = new Picture(500, 600);
int x2 = 122, y2 = 500;
Graphics gB = PB.getGraphics();
gB.setColor(Color.blue);
for (int i = 0; i <= 255; i++) {
    gB.drawLine(x2, y2, x2, y2 - (Bhis[i] / numB));
    x2 += 1;
}
PB.show();

```

```

jTextArea1.setText("Green max value : " + maxG + "\n" + Arrays.toString(Ghis) + "\n"
    + "Blue max value : " + maxB + "\n" + Arrays.toString(Bhis) + "\n"
    + "Red max value : " + maxR + "\n" + Arrays.toString(Rhis) + "\n"
);

} else {

    int maxGr = Rhis[0];
    for (int j = 1; j < Rhis.length; j++) {
        if (Rhis[j] > maxGr) {
            maxGr = Rhis[j];
        }
    }

    int numGr = maxGr / 600;
    Picture PGr = new Picture(500, 600);
    int x = 122, y = 500;
    Graphics gGr = PGr.getGraphics();
    gGr.setColor(Color.gray);
    for (int i = 0; i <= 255; i++) {
        gGr.drawLine(x, y, x, y - (Rhis[i] / numGr));
        x += 1;
    }
    PGr.show();

    jTextArea1.setText("Green max value : " + maxGr + "\n" + Arrays.toString(Ghis) + "\n");
}

```

11- Apply 3 filters (Box, Gaussian, Laplacian, Min, Max, Median).

We apply many filters as follow:

- **Coding part:**
- 1- **grayscale, clear, negative, darkening, lighting and sunset.**

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {  
    //GRAYSCALE  
    pic2.grayscale();  
    jLabel12.setIcon(new ImageIcon(pic2.getImage()));  
}
```

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    // CLEAR  
    pic2 = new Picture(pic1);  
  
    jLabel12.setIcon(new ImageIcon(pic2.getImage()));  
}
```

```
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {  
    //NEGATIVE  
    pic2.negate();  
    jLabel12.setIcon(new ImageIcon(pic2.getImage()));  
}
```

```
private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {  
    //LIGHTNING  
    pic2.lighten();  
    jLabel12.setIcon(new ImageIcon(pic2.getImage()));  
}
```

```
private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {  
    //DARKENING  
    pic2.darken();  
    jLabel12.setIcon(new ImageIcon(pic2.getImage()));  
}
```

```
private void jButton19ActionPerformed(java.awt.event.ActionEvent evt) {  
    //sunset  
    pic2.makeSunset();  
    jLabel12.setIcon(new ImageIcon(pic2.getImage()));  
}
```

2- median: 3x3 filter and take the value of the element in the middle.

```
private void jButton15ActionPerformed(java.awt.event.ActionEvent evt) {  
    //median  
    int[] pixel = new int[9]; //16  
    //pic2 = new Picture(pic1);  
    for (int i = 1; i < pic1.getWidth() - 1; i++) {  
        for (int j = 1; j < pic1.getHeight() - 1; j++) {  
  
            pixel[0] = (int) pic1.getPixel(i - 1, j - 1).getAverage();  
            pixel[1] = (int) pic1.getPixel(i - 1, j + 1).getAverage();  
            pixel[2] = (int) pic1.getPixel(i - 1, j).getAverage();  
  
            pixel[3] = (int) pic1.getPixel(i + 1, j - 1).getAverage();  
            pixel[4] = (int) pic1.getPixel(i + 1, j + 1).getAverage();  
            pixel[5] = (int) pic1.getPixel(i + 1, j).getAverage();  
  
            pixel[6] = (int) pic1.getPixel(i, j - 1).getAverage();  
            pixel[7] = (int) pic1.getPixel(i, j + 1).getAverage();  
            pixel[8] = (int) pic1.getPixel(i, j).getAverage();  
  
            Arrays.sort(pixel);  
  
            pic2.getPixel(i, j).setColor(new Color(pixel[4], pixel[4], pixel[4]));  
        }  
    }  
    jLabel2.setIcon(new ImageIcon(pic2.getImage()));  
}
```

3- Laplacian: by using filter 3 by 3, multiply each matrix value by filter value then sum all values and set the color to the pixel.

```
private void jButton16ActionPerformed(java.awt.event.ActionEvent evt) {  
    //Laplacian  
    int result1;  
    int[] p = new int[9];  
    for (int i = 1; i < pic1.getWidth() - 1; i++) {  
        for (int j = 1; j < pic1.getHeight() - 1; j++) {  
  
            p[0] = (int) pic1.getPixel(i - 1, j - 1).getBlue() * 0;  
            p[1] = (int) pic1.getPixel(i, j - 1).getBlue() * 1;  
            p[2] = (int) pic1.getPixel(i + 1, j - 1).getBlue() * 0;  
            p[3] = (int) pic1.getPixel(i - 1, j).getBlue() * 1;  
            p[4] = (int) pic1.getPixel(i, j).getBlue() * (-4);  
            p[5] = (int) pic1.getPixel(i + 1, j).getBlue() * 1;  
            p[6] = (int) pic1.getPixel(i - 1, j + 1).getBlue() * 0;  
            p[7] = (int) pic1.getPixel(i, j + 1).getBlue() * 1;  
            p[8] = (int) pic1.getPixel(i + 1, j + 1).getBlue() * 0;  
  
            result1 = p[0] + p[1] + p[2] + p[3] + p[4] + p[5] + p[6] + p[7] + p[8];  
  
            if (result1 < 0) {  
                result1 = 0;  
            } else if (result1 > 255) {  
                result1 = 255;  
            }  
            pic2.getPixel(i, j).setColor(new Color(result1, result1, result1));  
        }  
    }  
    jLabel2.setIcon(new ImageIcon(pic2.getImage()));  
}
```

4- Max: compute the value for each color separately and then sort them and set the result in the corresponding pixel.

```
private void jButton20ActionPerformed(java.awt.event.ActionEvent evt) {  
    //Max  
    int[] pixelR = new int[9];  
    int[] pixelG = new int[9];  
    int[] pixelB = new int[9];  
  
    for (int i = 1; i < pic1.getWidth() - 1; i++) {  
        for (int j = 1; j < pic1.getHeight() - 1; j++) {  
  
            pixelR[0] = (int) pic1.getPixel(i - 1, j - 1).getRed();  
            pixelR[1] = (int) pic1.getPixel(i - 1, j + 1).getRed();  
            pixelR[2] = (int) pic1.getPixel(i - 1, j).getRed();  
  
            pixelR[3] = (int) pic1.getPixel(i + 1, j - 1).getRed();  
            pixelR[4] = (int) pic1.getPixel(i + 1, j + 1).getRed();  
            pixelR[5] = (int) pic1.getPixel(i + 1, j).getRed();  
  
            pixelR[6] = (int) pic1.getPixel(i, j - 1).getRed();  
            pixelR[7] = (int) pic1.getPixel(i, j + 1).getRed();  
            pixelR[8] = (int) pic1.getPixel(i, j).getRed();  
            Arrays.sort(pixelR);  
  
            pixelG[0] = (int) pic1.getPixel(i - 1, j - 1).getGreen();  
            pixelG[1] = (int) pic1.getPixel(i - 1, j + 1).getGreen();  
            pixelG[2] = (int) pic1.getPixel(i - 1, j).getGreen();  
  
            pixelG[3] = (int) pic1.getPixel(i + 1, j - 1).getGreen();  
            pixelG[4] = (int) pic1.getPixel(i + 1, j + 1).getGreen();  
            pixelG[5] = (int) pic1.getPixel(i + 1, j).getGreen();  
  
            pixelG[6] = (int) pic1.getPixel(i, j - 1).getGreen();  
            pixelG[7] = (int) pic1.getPixel(i, j + 1).getGreen();  
            pixelG[8] = (int) pic1.getPixel(i, j).getGreen();  
            Arrays.sort(pixelG);  
  
            pixelB[0] = (int) pic1.getPixel(i - 1, j - 1).getBlue();  
            pixelB[1] = (int) pic1.getPixel(i - 1, j + 1).getBlue();  
            pixelB[2] = (int) pic1.getPixel(i - 1, j).getBlue();  
  
            pixelB[3] = (int) pic1.getPixel(i + 1, j - 1).getBlue();  
            pixelB[4] = (int) pic1.getPixel(i + 1, j + 1).getBlue();  
            pixelB[5] = (int) pic1.getPixel(i + 1, j).getBlue();  
  
            pixelB[6] = (int) pic1.getPixel(i, j - 1).getBlue();  
            pixelB[7] = (int) pic1.getPixel(i, j + 1).getBlue();  
            pixelB[8] = (int) pic1.getPixel(i, j).getBlue();  
  
            Arrays.sort(pixelB);  
  
            pic2.getPixel(i, j).setColor(new Color(pixelR[8], pixelG[8], pixelB[8]));  
        }  
    }  
    jLabel12.setIcon(new ImageIcon(pic2.getImage()));  
}
```


- 5- **Average:** sum all values in the filter of 3 x 3 then divide them by 9 and then check if it is less 0 assign to it 0 and more than 255 assign to it 255, then set the values in desired pixel.

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    //Averaging  
    for (int i = 1; i < pic1.getWidth() - 1; i++) {  
        for (int j = 1; j < pic1.getHeight() - 1; j++) {  
            int red = 0;  
            int green = 0;  
            int blue = 0;  
  
            red = ((1) * pic1.getPixel(i - 1, j - 1).getRed())  
                + ((1) * pic1.getPixel(i, j - 1).getRed())  
                + ((1) * pic1.getPixel(i + 1, j - 1).getRed())  
                + ((1) * pic1.getPixel(i - 1, j).getRed())  
                + ((-8) * pic1.getPixel(i, j).getRed())  
                + ((1) * pic1.getPixel(i + 1, j).getRed())  
                + ((1) * pic1.getPixel(i - 1, j + 1).getRed())  
                + ((1) * pic1.getPixel(i, j + 1).getRed())  
                + ((1) * pic1.getPixel(i + 1, j + 1).getRed());  
  
            red = red / 9;  
  
            if (red < 0) {  
                red = 0;  
            } else if (red > 255) {  
                red = 255;  
            }  
  
            green = ((1) * pic1.getPixel(i - 1, j - 1).getGreen())  
                + ((1) * pic1.getPixel(i, j - 1).getGreen())  
                + ((1) * pic1.getPixel(i + 1, j - 1).getGreen())  
                + ((1) * pic1.getPixel(i - 1, j).getGreen())  
                + ((-8) * pic1.getPixel(i, j).getGreen())  
                + ((1) * pic1.getPixel(i + 1, j).getGreen())  
                + ((1) * pic1.getPixel(i - 1, j + 1).getGreen())  
                + ((1) * pic1.getPixel(i, j + 1).getGreen())  
                + ((1) * pic1.getPixel(i + 1, j + 1).getGreen());  
  
            green = green / 9;  
  
            if (green < 0) {  
                green = 0;  
            } else if (green > 255) {  
                green = 255;  
            }  
  
            blue = ((1) * pic1.getPixel(i - 1, j - 1).getBlue())  
                + ((1) * pic1.getPixel(i, j - 1).getBlue())  
                + ((1) * pic1.getPixel(i + 1, j - 1).getBlue())  
                + ((1) * pic1.getPixel(i - 1, j).getBlue())  
                + ((-8) * pic1.getPixel(i, j).getBlue())  
                + ((1) * pic1.getPixel(i + 1, j).getBlue())  
                + ((1) * pic1.getPixel(i - 1, j + 1).getBlue())  
                + ((1) * pic1.getPixel(i, j + 1).getBlue())  
                + ((1) * pic1.getPixel(i + 1, j + 1).getBlue());  
  
            blue = blue / 9;  
  
            if (blue < 0) {  
                blue = 0;  
            } else if (blue > 255) {  
                blue = 255;  
            }  
  
            pic2.getPixel(i, j).setColor(new Color(red, green, blue));  
        }  
    }  
  
    jLabel2.setIcon(new ImageIcon(pic2.getImage()));  
}
```

12- Calculate the Construct value.

```
private void jButton18ActionPerformed(java.awt.event.ActionEvent evt) {  
    //Construct  
    //con=max-min/max+min  
  
    Pixel[] pixels = pic1.getPixels();  
  
    int Contrast;  
  
    double max_val = 0;  
    double min_val = pixels[0].getAverage();  
  
    for (int i = 0; i < pixels.length; i++) {  
        Contrast = (int) ((pixels[i].getRed() + pixels[i].getGreen() + pixels[i].getBlue()) / 3);  
  
        if (Contrast > max_val) {  
            max_val = Contrast;  
        }  
        if (Contrast < min_val) {  
            min_val = Contrast;  
        }  
    }  
  
    double contrast = ((max_val - min_val) / (max_val + min_val));  
  
    jTextField6.setText(String.valueOf(contrast));  
}
```

Sound Part



- 1- ChangeVolume using slider. Users specify %age [0, 100] and you will map onto [0 - 255] (Note: please record your own sounds, saying simple words such as 'One', 'Two' and 'Three' etc. Do not use long duration strange sounds during evaluation. This is applied to all sound methods.)

We are applying this method by using slider and change the volume based on the slider value.

- **Coding part:**

```
private void jSlider1StateChanged(javax.swing.event.ChangeEvent evt) {  
  
    Sound sound = new Sound(sound1);  
  
    SoundSample[] SA = sound.getSamples();  
  
    int volume = jSlider1.getValue();  
    //+ve  
    if (volume < OriginalVolume) {  
        for (int i = 0; i < SA.length; i++) {  
            SoundSample sample = SA[i];  
            int v = sample.getValue();  
            int newV = (v * 2);  
            SA[i].setValue(newV);  
        }  
  
        //-ve  
    } else if (volume > OriginalVolume) {  
        for (int i = 0; i < SA.length; i++) {  
            SoundSample sample = SA[i];  
            int v = sample.getValue();  
  
            int newV = (int) (v * 0.5);  
  
            sample.setValue(newV);  
        }  
    }  
    OriginalVolume = volume;  
  
    sound.blockingPlay();  
}
```

2- Clipping a given sound (Define the segment first to be clipped.) get the sampling rate and multiply it with number of seconds will give you total samples in the sound. User will provide starting and ending index of sound through the user interface, to clip the sound.

We are applying this method by call (clip) method after taking the start and end samples and making the clip.

- **Coding part:**

```
private void ClippingActionPerformed(java.awt.event.ActionEvent evt) {  
  
    Sound sound = new Sound(sound1);  
  
    int num1 = Integer.parseInt(Start.getText());  
  
    int num2 = Integer.parseInt(End.getText());  
  
    Sound sound2 = sound.clip(num1, num2);  
  
    sound2.blockingPlay();  
  
}
```

3- Splicing at least two sounds together keeping the silent zone. (You must implement as many sounds as you want i.e., more than two. In that case, fix the length/duration of silent zone, e.g. 10% of Sampling Rate before splicing. You may create two methods, one for Read/Write from source to Target and other method for silent zone. Once two sounds are spliced, user may be asked to splice for 3rd sound or more. Hint: Consider the output of splicing of S1, and S2 as Source while splicing S3 and so on... Generalized splicing.

We are applying this method by taking two sounds and then create a new sound with length as the two sounds with silent zone, then set the first sound then set the silent zone after that set the second sound. Finally play the generated sound.

- **Coding part:**

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    // Splice  
    Sound sound_tar = new Sound(FileChooser.pickAFile());  
  
    Sound sound2 = new Sound(sound1.getLength() + sound_tar.getLength() + 20000);  
  
    int targetIndex = 0; // the starting place on the target  
  
    int value = 0;  
  
    for (int i = 0; i < sound1.getLength(); i++, targetIndex++) {  
        value = sound1.getSampleValueAt(i);  
        sound2.setSampleValueAt(targetIndex, value);  
    }  
  
    int silentzone = 20000;  
  
    for (int i = 0; i < silentzone; i++, targetIndex++) {  
        sound2.setSampleValueAt(targetIndex, 0);  
    }  
  
    for (int i = 0; i < sound_tar.getLength(); i++, targetIndex++) {  
        value = sound_tar.getSampleValueAt(i);  
        sound2.setSampleValueAt(targetIndex, value);  
    }  
    sound2.play();  
}
```

4- Add a method to spread a sound of two sec over 4 seconds and then squeeze it over 1 second and play at the end. Not only implement a method but also call on GUI. i.e., Frequency changing.

We are applying this method by

- **Coding part:**

```
public Sound Squeeze() {
    Sound ss = new Sound(sound1);
    SoundSample[] sampleSpreaded = ss.getSamples();
    // Squeeze / 4
    Sound squeezed = new Sound((int) sampleSpreaded.length / 2);
    for (double sourceIndex = 0, targetIndex = 0; targetIndex < squeezed.getLength(); sourceIndex += 2, targetIndex++) {
        squeezed.setSampleValueAt((int) targetIndex, sampleSpreaded[((int) sourceIndex)].getValue());
    }
    return squeezed;
}

public Sound spread() {
    Sound ss = new Sound(sound1);
    SoundSample[] sample = ss.getSamples();

    // Spread * 2
    Sound spreaded = new Sound((int) sample.length * 2);
    for (double sourceIndex = 0, targetIndex = 0; targetIndex < spreaded.getLength(); sourceIndex += 0.5, targetIndex++) {
        spreaded.setSampleValueAt((int) targetIndex, sample[((int) sourceIndex)].getValue());
    }
    return spreaded;
}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {

    Sound s1 = spread();
    s1 = Squeeze();

    s1.play();

}
```

5- Apply reversing and mirroring of sounds / Blending Two or more sounds / simple average, weighted average, max, min and median filters on sounds.

We are call some methods to apply different effects on the sound.

- **Coding part:**

1- Reverse method:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

    Sound sound = new Sound(sound1);
    sound.reverse();
    sound.play();

}
```

2- Mirroring method:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    Sound sound = new Sound(sound1);
    sound.mirrorFrontToBack();
    sound.play();

}
```

3- Blending method:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    Sound userSound = new Sound(FileChooser.pickAFile());  
    Sound uSound = new Sound(sound1);  
  
    for (int i = 0; i < uSound.getLength() - 1; i++) {  
        uSound.setSampleValueAt(i, uSound.getSampleValueAt(i) + userSound.getSampleValueAt(i));  
    }  
  
    uSound.play();  
  
}
```

4- Echo method:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    Sound userSound = new Sound(FileChooser.pickAFile());  
    Sound uSound = new Sound(sound1);  
  
    for (int i = 0; i < uSound.getLength() - 1; i++) {  
        uSound.setSampleValueAt(i, uSound.getSampleValueAt(i) + userSound.getSampleValueAt(i));  
    }  
  
    uSound.play();  
  
}
```

Move Part



- 1- Create `RecatangleMovie` method, show it is moving from top right corner to bottom left, curved trajectory. User will select the duration of movies through user interface.

We are applying this method by take from user the number of frames and use it also for the size of the picture and then set color and rectangle on it then add the frame then play them.

- **Coding part:**

```
private void req_moveActionPerformed(java.awt.event.ActionEvent evt) {  
  
    int Frames_num = Integer.valueOf(FramesNum.getText());  
  
    Picture p = null;  
    Graphics g = null;  
  
    FrameSequencer frameSequencer = new FrameSequencer("req_Move");  
    frameSequencer.setShown(true);  
  
    for (int i = 0, j = Frames_num * 5 ; i < Frames_num; i++, j-= 12) {  
  
        p = new Picture(Frames_num * 5, Frames_num * 5);  
  
        g = p.getGraphics();  
        g.setColor(Color.yellow);  
  
        g.fillRect(j , i * 10, 100, 100);  
  
        frameSequencer.addFrame(p);  
    }  
    //-4-  
    frameSequencer.play(Frames_num);  
}
```


2- CreateRecatangleMovie method Show the trajectory of motion when a ball is dropped on the floor. Note that the dropped ball does not bounce back to same height and gradually comes to rest. User will select the duration of movies through user interface. Hint: Sine/Cosine function

We are applying this method by take from user the number of frames and calculate the heigh and start point and by (sin) determine the angle of drop.

• Coding part:

```
private void Dropped_ballActionPerformed(java.awt.event.ActionEvent evt) {  
  
    int framesPerSec = Integer.valueOf(F_num.getText());  
    Picture p = null;  
    Graphics g = null;  
    FrameSequencer frameSequencer = new FrameSequencer("Dropball");  
    int distance = 300;  
    int height = 480;  
    int start = (height - 50) - distance; //130 stop and down  
  
    for (int i = 0; i < framesPerSec; i++) {  
        p = new Picture(640, height); // shap picture  
        g = p.getGraphics();  
        int y = (int) (Math.sin(30 + (i)) * distance) + start; //schot  
        if (y > (height - 50 - 10)) {  
  
            distance = distance / 2;  
            start = (height - 50) - distance;  
            g.setColor(Color.gray);  
            g.drawLine(i * 19, y, 60, 50);  
            g.setColor(Color.GRAY);  
            g.fillOval(i * 19, y, 60, 50);  
        }else{  
            y = (int) (Math.sin(30 + (i)) * distance) + start;  
            g.setColor(Color.gray);  
            g.drawLine(i * 19, y, 50, 50);  
            g.setColor(Color.GRAY);  
            g.fillOval(i * 19, y, 50, 50);  
        }  
        frameSequencer.addFrame(p);  
    }  
    frameSequencer.play(framesPerSec);  
}
```

3- Create Sunset movie – Total change must be made by the user on runtime. User will select the duration of movies through user interface.

We are applying this method by taking the number of frames and use it in the sunset value.

- **Coding part:**

```
private void Sunset_moActionPerformed(java.awt.event.ActionEvent evt) {  
  
    int value=Integer.parseInt(Sun_num.getText());  
  
    String path = FileChooser.pickAFile();  
    Picture beachP = new Picture(path);  
    double amountReduction = 1;  
  
    Picture temp = null;  
  
    FrameSequencer frameSeq= new FrameSequencer("dir");  
  
    // loop creating the frames  
    for (int i = 0; i < value * 15; i++) {  
        temp = new Picture(beachP);  
  
        amountReduction = amountReduction * 0.95;  
  
        if (amountReduction < value) {  
            temp.makeSunset(amountReduction);  
        }  
  
        frameSeq.addFrame(temp);  
    }  
    // play the movie  
    frameSeq.play(value);  
}
```

4- Create Background Subtraction Movie. User will select the duration of movies through user interface.

We are applying this method by take number of frames and three pictures from user then subtract the two pictures to obtain the background pictures by swapBackground method.

- **Coding part:**

```
private void SubtracActionPerformed(java.awt.event.ActionEvent evt) {  
    //background subtraction  
    String file = FileChooser.pickAFile();  
  
    String file2 = FileChooser.pickAFile();  
    Picture oldBG = new Picture(file2);  
  
    String file3 = FileChooser.pickAFile();  
    Picture newBG = new Picture(file3);  
  
    FrameSequencer frameSequencer = new FrameSequencer("run");  
    int framesPerSec = Integer.parseInt(sub.getText());  
  
    for (int i = 0; i < framesPerSec; i++) {  
        pic = new Picture(file);  
        pic.swapBackground(oldBG, newBG, i);  
        frameSequencer.addFrame(pic);  
    }  
  
    frameSequencer.play(framesPerSec);  
}
```

5- Create Edge Detection Movie. User will select the duration of movies through user interface.

We are applying this method by taking from the user number of frames and apply edgeDetection method on the pictures.

- **Coding part:**

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    Picture pic = new Picture(FileChooser.pickAFile());  
    // declare other variables  
    Picture target = null;  
    FrameSequencer frameSequencer = new FrameSequencer("edgedetection");  
  
    int framesPerSec = Integer.parseInt(edge.getText());  
  
    // loop creating the frames  
    for (int i = 0; i < framesPerSec; i++) {  
        target = new Picture(pic);  
        target.edgeDetection(30 - i);  
        frameSequencer.addFrame(target);  
    }  
  
    // play the movie  
    frameSequencer.play(framesPerSec);  
}
```

Bonus part



Move our names with sound:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    Sound sound_move = new Sound(FileChooser.pickAFile());  
    SoundSample [] samples = sound_move.getSamples();  
    SoundSample sam = null;  
  
    int value;  
    int framesPerSec = 100;  
    Picture p = null;  
    Graphics g = null;  
    // -1-  
    FrameSequencer frameSequencer = new FrameSequencer("BONUS");  
    frameSequencer.setShown(true);  
    Font font= new Font("Monospaced", Font.ITALIC, 40);  
  
    for (int i = 0; i < samples.length; i++) {  
  
        p = new Picture(1000,1000); //samples.length  
        g = p.getGraphics();  
        sam = samples[i];  
        value = sam.getValue();  
        g.setFont(font);  
        g.setColor(Color.DARK_GRAY);  
        g.drawString("Somayah & Almas", value+100 , i);  
  
        // -3-  
        frameSequencer.addFrame(p);  
    }  
    // -4-  
    frameSequencer.play(framesPerSec);  
    sound_move.play();  
}
```

Conclusion



Dealing with multimedia is very enjoyable and serves many requirements of life. It is also a very important section in the field of technology. One of the results of our work on this project is our realization of the beauty and expansion of the field of multimedia and the reflection of its importance.

Link project record:

https://drive.google.com/drive/folders/1BAGZI2ihgDR7dnere2ndB7icdH_DZoF2?usp=sharing