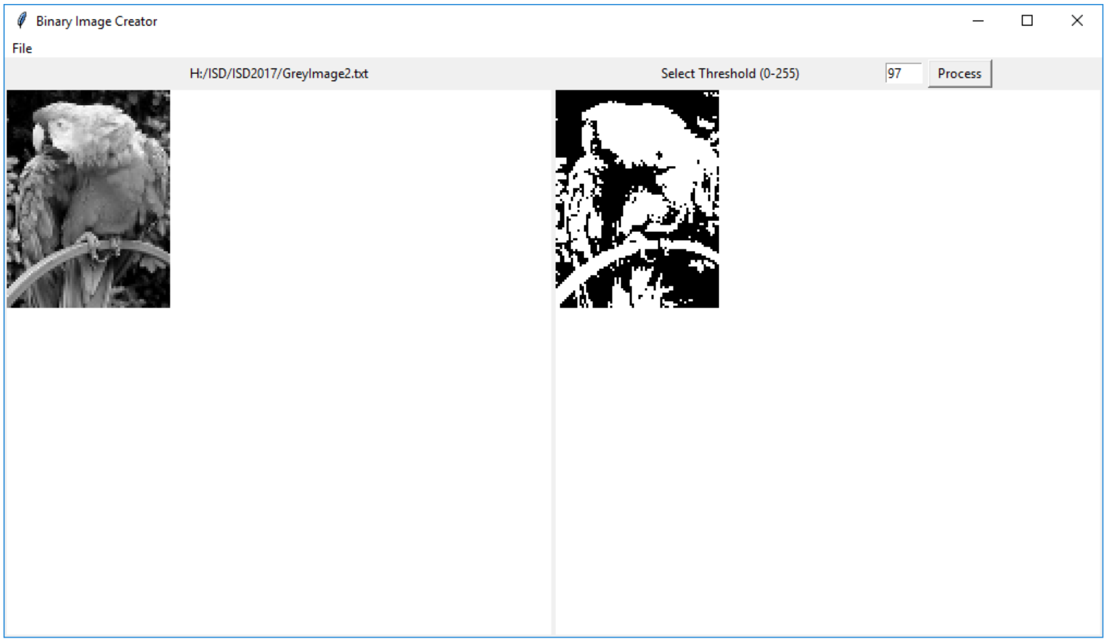
ISD Term 2 Coursework 2

- Deadline for submission: **Monday 9th April at 17:55**. Please read the *Submitting the assignment* section at the end of this document.
- Second deadline (mark will be capped to pass mark unless there are mitigating circumstances) is Monday 23rd April 17:55.
- Any submissions uploaded after the 2nd deadline will be capped at zero marks.

The Exercise

Image binarisation is a process used to simplify images for further investigation or simply to reduce the size of an image file. It simply replaces each pixel in an image with one of two possible colours (usually either black or white).

The object of this coursework is for you to write an extensible Graphical User Interface (GUI) that can allow a user to load image data, in the form of text files, and perform image binarisation and to save the results.



H:/ISD/ISD2017/GreyImage2.txt

Select Threshold (0-255) 97

Figure 1: User interface in action

1 The Image data file formats

There are 3 image file formats to consider in this exercise. All of them are text based and the first line in the file is the name of the format used.

1. Greyscale Image
   Each line of data contains information about one pixel.
   It has the form x, y, v. Where x, y are the coordinates of the pixel and v is the intensity of the pixel. The intensity ranges from 0 (black) all the way to 255 (white).
   Below is the first 5 lines of the file `GreyImage.txt`.

   ```
   Greyscale Image
   61, 85, 2
   69, 8, 21
   39, 55, 125
   46, 37, 92
   ```

2. Colour Image
   Each line of data contains information about one pixel.
   It has the form x, y, r, g, b. Where x, y are the coordinates of the pixel and r, g, b are the red, green and blue components of the colour.
   Below is the first 5 lines of the file `ColourImage.txt`.

   ```
   Colour Image
   117, 69, 76, 85, 66
   7, 126, 82, 63, 56
   81, 16, 132, 146, 95
   13, 178, 30, 16, 39
   ```

Your program must be able to *read* in files using the above 2 formats. Your program must also be able to *write* a file in the following Binary Image format:

1. Binary Image
   This format has the form x, y, b. Where x, y are the coordinates of the pixel and b is either 0 or 1. Below is the first 5 lines of `GreyImageBinaryOutput.txt`.

   ```
   Binary Image
   61,85,0
   69,8,0
   39,55,1
   46,37,0
   ```

2 Binarisation Methods

For a user selected threshold, t. The binarisation process works as follows:

- For Greyscale Images
  if $v < t$ then the output is 0 else it is 1.

- For Colour Images
  if $\frac{r+g+b}{3} < t$ then the output is 0 else it is 1.

Automatically selecting a threshold

Your program is required to suggest a threshold to use. This is calculated in the following way:

- For Greyscale Images
  t is the average of all the intensity values (v) from the image.

- For Colour Images
  The intensity of a pixel is $\frac{r+g+b}{3}$. t is the average of all these computed intensity values from the image.

The suggested threshold is `int(t)`, (since t is likely not to be a whole number).

3 Coding the solution

Your solution must follow the the specification below:

Display

For the display use 2 `canvas` widgets, one to show the input image and one to show the result of the binarisation. The pixels in the image are to be displayed using the `create_rectangle` method.

Above the leftmost canvas have a `label` that will display the full file name of any succesfully loaded file. Above the rightmost canvas have another `label` with content `Select Threshold (0-255)` an `entry box` and finally a `button`.

Have a `menu` item called `File` which has sub-items `load` and `save`, each link to the relevant file manager window.

Once a file has been succefully loaded, its name will be displayed in the leftmost label. The suggested threshold will be computed and placed in the `entry box`. When the process `button` is pressed, the binaristaion will occur using the threshold value in the `entry box`. The result will be displayed in the rightmost canvas.

To successfully complete this coursework you will need to investigate the functionality available in tkinter, specifically research the `canvas` widget. The last slide (49) on

http://wiki.scinethpc.ca/wiki/images/9/9c/Tkinter.pdf

gives you links to resources, be sure to use Python 3 versions.

4 Code Provided

A structure to your program is already provided for you to download. You must use the structure provided and follow any instructions provided in these files.

- `BinaryConverter.py`. This class is your GUI. You can run it and it will create a small window. Read through it carefully. You have been provided with working code to display pixels onto a canvas. The colour a `create_rectangle` requires is a `String` that follows a specific format. I have also provided a method called `_determineColorValue()` for each class of image which will create this `String` for you.

- `GUIconnect.py`. An abstract class which is the parent to both `ColourImage.py` and `GreyScaleImage.py`. Read the comments in the method definitions carefully. These provide further instructions of what to write.

- `ColourImage.py` and `GreyScaleImage.py` are where you put code specific to each file format.

- `BinaryImage.py`. This class stores the result of the binarisation process and the code for the file format for saving to.

Hint: `ColourImage.py`, `GreyScaleImage.py` and `BinaryImage.py` are missing constructors. You will need to implement them.

Note: You may only `import` from `tkinter`. You must not `import` any other libraries.

Extensibility and Polymorphism

The Graphical User Interface should not be too tightly coded to binarisation approaches and their file formats. In fact we aim to have a situation that if a programmer wishes to include their own file format and binarisation approach they could include it with minimal effort. In addition if the programmer wishes to change the file format for saving the binary image, the code for your GUI (`BinaryConverter`) should not need to be ammended at all.

To achieve this, all image types are implemented as classes that inherit from the abstract class `GUIconnect` (an abstract class has method definitions but with no usable method bodies). Your Graphical User Interface should communicate with these classes through the methods defined in that abstract class.

Within your Graphical User Interface use polymorphism to make the inclusion of another image file class as simple as possible.

You may always assume that the first line in an image file is a string containing the file type.

Preliminaries

Download from Moodle BinaryConverter.py, GUIconnect.py, ColourImage.py, GreyScaleImage.py and BinaryImage.py . Place them all in the **same** directory of your choosing. Read through this code and understand the structure.

Finally download the test data:
GreyImage.txt, ColourImage.txt, GreyImageBinaryOutput.txt, ColourImageBinaryOutput.txt.

- For GreyImage.txt, the suggested threshold should be 97 and the file output after saving the result is in GreyImageBinaryOutput.txt.
- For ColourImage.txt, the suggested threshold should be 70 and the file output after saving the result is in ColourImageBinaryOutput.txt.

Marking Scheme

This coursework is marked out of 100 as follows:

1. **20 marks**. Producing a Graphical User Interface that fits the specification described above and is extensible. Using object oriented programming and in particular polymorphism within BinaryConverter.
2. **10 marks**. Robustness of your user interface to users interacting with it, e.g. your program should not crash if the user types in a name instead of a number for the parameter.
3. **20 marks**. Correctly implementing class functionality for BinaryImage.
4. **20 marks**. Correctly implementing class functionality for GreyImage.
5. **10 marks**. Correctly implementing class functionality for ColourImage.
6. **15 marks**. Code Quality. Marks will be awarded for the clarity of your code, this includes the following:
    - Comments. See function comments in Section 5.2
    - All Classes will need to be commented appropriately see Slide 21 of Chapter 9 (Writing the Public Interface).
    - Appropriate variable, function and argument names.
    - Any hard-coded strings that are used in multiple locations should be defined as constants.
7. **5 marks**. Correctly following the assignment submission instructions.

Submitting the assignment:

1. Always keep backup copies of all assignments. If your assignment gets lost, a backup copy will make things easier for you.
2. Include comments in your source code.
3. Include in a comment at the top of each source code file, your name, the name of the programme you are taking (e.g., MSc IT, etc.), and the submission date.
4. Submit one **zip** file containing only your `BinaryConverter.py`, `ColourImage.py`, `GreyScaleImage.py` and `BinaryImage.py` . Also include the file `GUIconnect.py`.
5. To submit your file, follow the Upload Submission for Term2 - Coursework 2 link on the ISD Moodle page.