CS105 Exemption Project 2016

Damyan Zdravkov Kalev

201542606

BSc Computer Science

Except where explicitly marked to the contrary by quotation or reference, what follows is all my own work. I have not knowingly allowed others to copy my work.

Explanation of the original ImageViewer system

The ImageViewer class creates the GUI and initializes other necessary components/

* In its constructor it initializes currentImage and the filters and calls the makeFrame() method to create the swing frame.
* private void openFile() : the method opens a filechooser to select an image, then loads it with the loadImage () method of the ImageFileManager class and after that displays it.
* private void close(): closes the current image.
* private void saveAs(): saves the current image to a file using the saveImage() method of the ImageFileManager class.
* private void quit(): exits the application.
* private void applyFilter(Filter filter): takes as a parameter a filter object and applies it to the current image.
* private void showAbout()creates a JOptionPane and displays information about the application.
* private void makeLarger():doubles the size of the current image using nested for-loops and the OFImage methods getPixel() and setPixel().
* private void makeSmaller(): similar to make larger, but makes the current image twice as smaller.
* private void showFilename(String filename): shows the filename if a file(image) has been loaded
* private void showStatus(String text): shows a message in the status bar
* private void setButtonsEnabled(boolean status): enables or disables the toolbar buttons
* private List<Filter> createFilters(): creates a list with the existing filters
* private void makeFrame(): creates the GUI
* private void makeMenuBar(JFrame frame): creates the menu bar of the main frame

The ImagePanel class extends JComponent with the functionality of displaying an OFImage on the surface.

The OFImage class extends BufferedImage. It defines an image in Objects First format. It has two constructors: one with parameter BufferedImage and one with parameters width and height(both of type integer).It also has two methods public void setPixel(int x, int y, Color col) and public Color getPixel(int x, int y) which are used to modify the current image(mostly from the filters).

The ImageFileManager class provides static methods for loading and saving images.

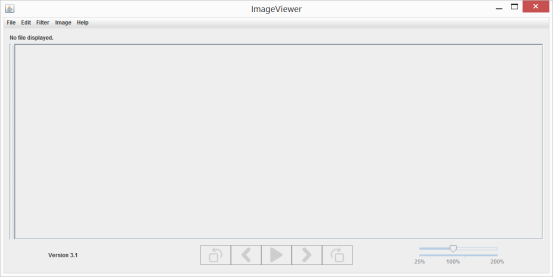
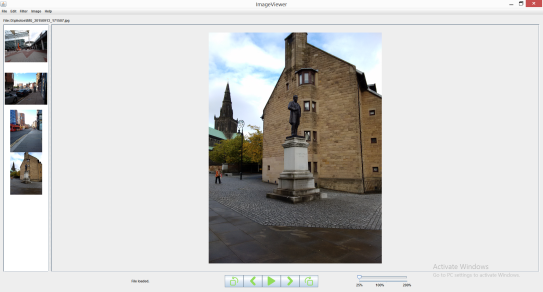
The ImageDriver class provides the main method from which the application is started.

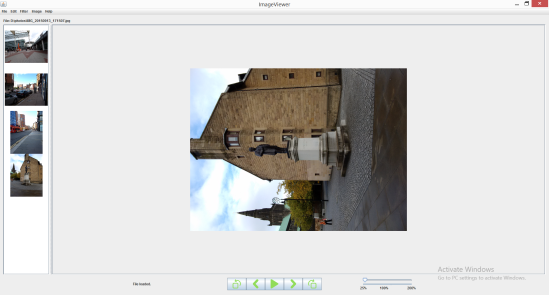
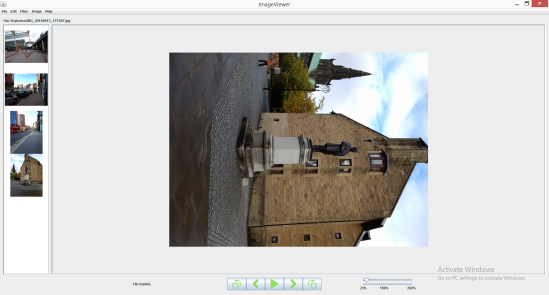
The Filter class is an abstract superclass for all filter classes.

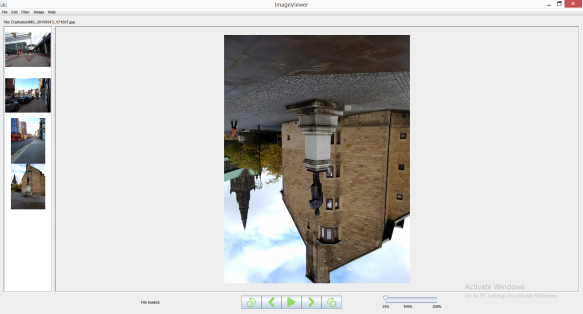
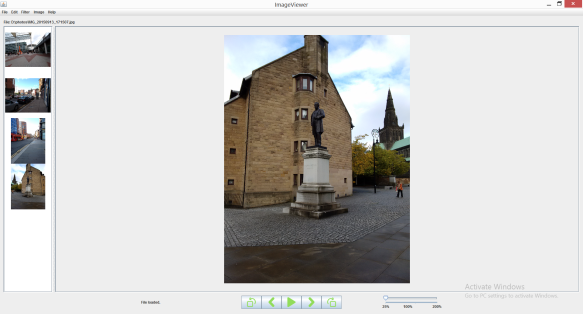
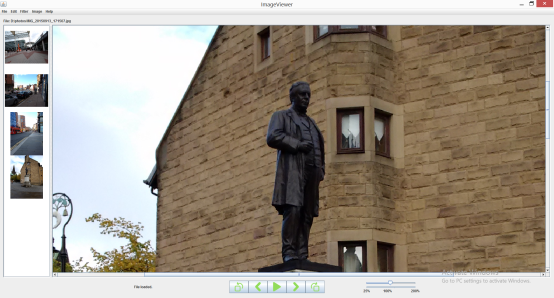
Filter classes:

* Threshold – changes the colors of the pixels in the image to black, white or gray according to their brightness
* Solarize – creates a solarization effect
* Smooth – reduces sharp edges and pixelization
* Pixelize – opposite to smooth creates or increases pixelization
* Mirror – flips the image vertically
* Lighter – increases the brightness of the image
* Invert – inverts the colors of the pixels in the image
* GrayScale – removes colors from the image
* FishEye – creates and effect similar to fisheye camera lens
* Edge – detects the edges in the image and highlights them
* Darker – reduces the brightness of the image

Summary

* Added JScrollPane
* Added JSplitPane
* Handling multiple images
* Added preview pane with thumbnails for the loaded pictures
* Added Undo/Redo functionality
* Slideshow
* Added rotate left/right, flip horizontal/vertical functionality
* Added resize function
* Added reload function
* Disabled menu items when no image is displayed
* Disabled undo/redo when not possible
* Added zoom functionality





Changes and improvements

Removed:

* MirrorFilter class
* private void makeSmaller () from ImageViewer class
* private void makeLarger() from ImageViewer class
* public void clearImage() from ImagePanel class

Changed:

* ImagePanel class
  + public void setImage(OFImage image): the image is scaled in order to fit into the ImagePanel
  + public void paintComponent(Graphics g) : the image is displayed in the center of the panel, instead of the top left corner
* ImageViewer class
  + The constructor: an object of type ImageAlbum is initialized; an ArrayList<String> that will contain the absolute paths of all images is initialized; Vector<ImageIcon> that will hold all thumbnails is initialized
  + private void openFile(): after the image is loaded it is stored as well as its absolute path and resized version(thumbnail), the menu items are enabled
  + private void close(): the image as well as its filepath and thumbnail are deleted and if other images are loaded the previous one is displayed. If not – the menu items and buttons are disabled and nothing is displayed
  + private void applyFilter(Filter filter): a copy of currentImage is created and the filter is applied to it(because of problems with the undo/redo functionality). Then thecpy is assigned to the currentImages and the currentImage is added to the album and displayed; the undo/redo menu items are enabled/disabled.
  + private void setButtonsEnabled(boolean status): enables or disables the following buttons: rotateLeftButton, rotateRightButton, prevButton, nextButton, slideshowButton and the JSlider zoomSlider
  + private void makeFrame(): To the JSplitPane splitPane are added two JScrollPanes: scrollPane and previewPane. To scrollPane is added the imagePanel, while to the previewPane is added JList<ImageIcon> list, which is populated by the Vector<ImageIcon> thumbnails. The ImageIcons are centered using DefaultListCellRenderer. The buttons are created and added to JLabel buttons, the zoomSlider is created and added to JLabel zoomSliderPanel and the statusLabel is created. The three labels are then added to the toolbar using GridLayout manager.
  + private void makeMenuBar(JFrame frame):
    - Edit menu added with menu items: undo, redo, reload
    - Image menu added with menu items: resize, rotate 90, rotate 270, flip horizontal, flip vertical

Added:

* Libraries: Scalr library is used for resizing images in the project
* Classes
  + History class: provides the undo/redo functionality: In the constructor two LinkedList<OFImage> (undoHistory and redoHistory)are inisialized. Every time the current image is changed, it is added to the undoHistory list. When an undo is executed the last added image is removed from the undoHistory and added to the redoHistory. If a redo is executed, the opposite happens. Meanwhile two indexes(undoIndex and redoIndex) are constantly updated to track the number of elements in the lists. They are used in the isUndoable() and isRedoable() method, which return true if undo or redo accordingly are possible.
  + AlbumImage: creates a History object for every image loaded into the ImageViewer and constantly updates it, providing the multiple images functionality. Also contains the method public ImageIcon createThumbnail(OFImage image) which creates the thumbnails for the previewPane.
* Methods(all in the ImageViewer class)
  + private void undo(): if possible changes the currentImage to the previous and disables/enables the undo, redo menu items
  + private void redo(): if possible changes the currentImage to the next one and disables/enables the undo, redo menu items
  + private void zoom(): gets the current value from the zoomSlider. Then using the public void setImage(OFImage image, double scale) from ImagePanel resizes the ImagePanel and the panelImage using the value. After that it calculates the center of the viewport of the scrollPane and displays the ImagePanel in the center of the scrollPane.
  + private void resize() : creates JOptionPane with two JLabels and two JTextFields inside(they are added to JPanel resizePanel and after that the panel is added to the JOptionPane). The input of two Strings is checked with regular expression if it contains integers and if yes used to resize the currentImage.
  + private void flipHorizontal(): the method is identical to the one in the mirror filter(the reason why the mirror filter was removed).After the algorithm is finished the changed image is added to the album and undo/redo are updated.
  + private void flipVertical(): the method is similar to the flipHorizontal but the center axis is in the middle of the height of the image(not the width like in flipHorizontal).After the algorithm is finished the changed image is added to the album and undo/redo are updated.
  + private void rotateRight(): the method uses two nested if statements to iterate through the pixels in the image and the getPixel() and setPixel() methods in OFImage. The width of the rotated image is equal to the heigh of the original image minus the value of the current y coordinate and minus 1.The height of the rotated image is equal to the current value of the x coordinate. After the algorithm is finished the changed image is added to the album and undo/redo are updated.
  + private void rotateLeft(): quite similar rotateRight()but the width of the rotated image is equal to the current y coordinate and the height is equal to the original’s width minus the current x coordinate minus 1. After the algorithm is finished the changed image is added to the album and undo/redo are updated.
  + private void setMenuItemsEnabled(boolean status): the method enables the menu items when an image is loaded and disables them if it is not.
  + private void setUndoRedoEnabled(): uses the values returned by the isUndoable() and isRedoable() methods to determine if the actions are possible and disables/ enables the undo, redo menu items accordingly.
  + private void makeSlideshowFrame(): creates a new JFrame. Then sets the layout of the frame’s content pane to GridLayout and adds ImagePanel to it. Then the frame is set to full screen, the default close operation is set to dispose on close and the frame is set visible. A timer is created (action listener is added in the constructor). In the actionPerformed method the images are loaded using the absolute addresses in the filepaths array list, then resized to fit the screen and added to the image panel.
  + The ImageViewer class implements the ListSelectionListener and it’s method public void valueChanged(ListSelectionEvent arg0) defines what happens when one of the thumbnails in the previewPane is clicked(one of the ImageIcons in the JList is selected).

Used lessons from the course:

* Arrays and For Loops – the rotate and flip methods
* Arrays of Objects – in ImageAlbum(the album ArrayList<History>)
* Iteration – in the Timer in makeSlideshowFrame()
* Using Library Classes – all the swing components, using the Scalr external library
* Testing and Debugging
* Inheritance – in ImagePanel the class extends JLabel
* Graphical User Interface – all the swing components
* Maps, Sets, Tokenizing strings, Constants – Hashtable<Integer, JLabel> for the zoomSlider label table
* Inheritance-overriding - ImageViewer class implements ListSelectionListener
* Exceptions and File I/O - in the reload method and in the Timer in the slideshow frame