2018 여름 JAVA 스터디

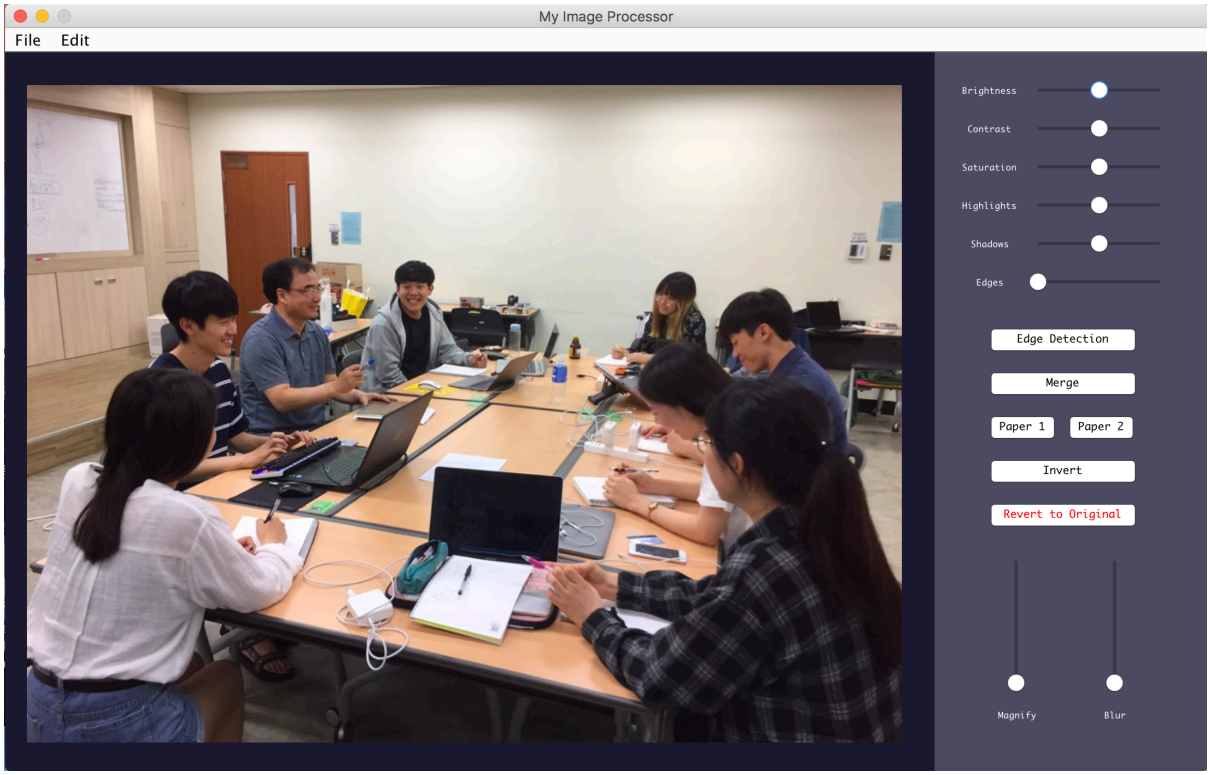
Round 4: Image Processor

6. 28 – 6.30

21700581 이지현

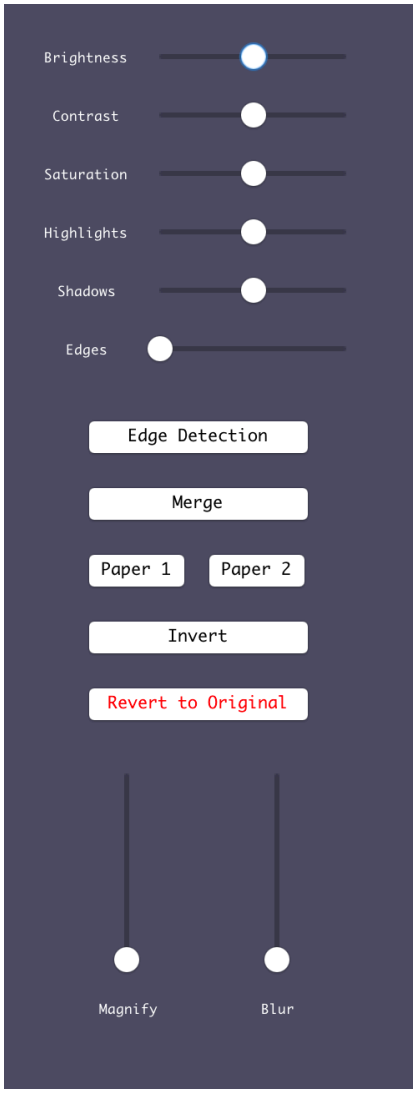
I. User's Manuel

i) 기본 인터페이스



: 오른쪽 컨트롤 패널에서 사진을 조정할 수 있고, 마우스로 사진을 클릭하면 원본 이미지를 볼 수 있습니다. 클릭을 떼면 다시 현재 이미지로 돌아갑니다.

ii) 컨트롤 바



(1) Brightness

: 사진의 밝기를 조정합니다.

(2) Contrast

: 사진의 대비를 조정합니다.

(3) Saturation

: 사진의 채도를 조정합니다.

(4) Highlights

: 사진의 Highlights (밝은 부분) 을 더 밝거나, 어둡게 조정합니다.

(5) Shadows

: 사진의 Shadows (어두운 부분) 을 더 어둡거나, 밝게 조정합니다.

(6) Edges

: 사진의 윤곽선을 부각시켜 사진이 더 또렷하게 보이게 해줍니다.

(7) Edge Detection

: 사진의 경계선 추출 결과를 보여줍니다.

(8) Merge

: 두 개의 사진을 합성합니다. 가로 / 세로 합성과 합성의 경계선 비율을 설정할 수 있습니다.

(9) Paper 1 / Paper 2

: 사진에 종이 질감을 입혀줍니다.

(10) Invert

: 사진을 반전시킵니다.

(11) Revert to Original

: 사진을 원본으로 되돌립니다.

(12) Magnify

: 돋보기 모드를 킵니다. 확대 비율을 조정할 수 있습니다.

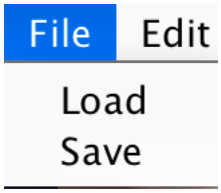
(13) Blur

: 사진의 일정 부분을 흐릿하게 만드는 지우개 모드를 킵니다. 뾰루지 제거 등에 사용될 수 있습니다.

iii) 메뉴

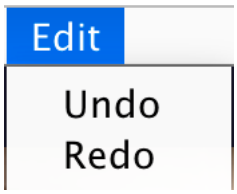
(1) File

: 새로운 파일을 불러오거나, 현재 작업중인 사진을 png 파일로 저장할 수 있습니다.



(2) Edit

: 사진에 가한 조정을 Undo / Redo 할 수 있습니다.



II. Technical Reference

1) Class Description

(1) Main

: 본 프로그램의 메인 함수입니다. *Frame_Main* 클래스의 객체를 생성하여 줍니다.

(2) Frame_Main

: JFrame 을 상속하며, 본 프로그램의 유일한 프레임 클래스입니다.

(3) Panel_Image

: JPanel 을 상속하며, 파일로부터 로드해온 이미지나, 유저의 조정을 통해 변환된 이미지를 화면에 출력해주는 패널입니다.

(4) Panel_Control

: JPanel 을 상속하며, 유저 인터페이스 오른쪽의 컨트롤 슬라이더 / 버튼들을 포함하는 패널입니다.

(5) Button_Combine, Button_Edge, Button_Invert, Button_Paper, Button_Paper2, Button_Revert

: JButton 을 상속하며, 화면에 출력되는 사진을 조정하는 버튼들입니다. *Panel_Control* 에서 Instance Variable 로 이 버튼 클래스들의 객체를 가지고 있습니다.

(6) Slider_Blur, Slider_Brightness, Slider_Contrast, Slider_Edges, Slider_Highlights, Slider_Magnifying, Slider_Saturation, Slider_Shadows

: JSlider 를 상속하며, 화면에 출력되는 사진을 조정하는 버튼들입니다. *Panel_Control* 에서 Instance Variable 로 이 슬라이더 클래스들의 객체를 가지고 있습니다.

2) General Flow

(1) 메인 함수에서 프로그램의 뼈대가 될 프레임 클래스 (*Frame_Main*) 의 객체를 생성합니다. *Frame_Main* 클래스의 생성자에는 *Panel_Image* 클래스의 객체가 초기 Content Pane 으로 설정되어 있고, Panel_Control 의 객체가 그 위에 추가되어 있어 기본 유저 인터페이스를 구현합니다.

(2) 유저가 메뉴 바의 File > Load 를 클릭하면, 파일로부터 이미지를 불러와 BufferedImage 로 저장한 후, 저장된 BufferedImage 를 *Vector<BufferedImage> history* 에 추가합니다 (후에 Undo / Redo 기능 구현에 사용).
또한, 이미지의 각 픽셀 RGB 값을 Integer 타입의 3 차원 배열에 저장해 놓습니다.

(3) 유저가 컨트롤 패널의 버튼을 누르거나 슬라이더를 조정할 경우, 그림의 RGB 값이 저장되었던 Integer 타입의 배열을 읽어들여 그 값을 조정한 뒤, 출력할 때에는 그 픽셀 값들을 다시 BufferedImage 로 변환한 뒤 출력합니다. 이는 사진 조정으로 인한 RGB 값의 손실을 막기 위함인데, Integer 타입의 배열과 BufferedImage 사이의 변환이 빈번하게 이루어져 다소 비효율적이라 생각됩니다. 더 효율적인 알고리즘을 찾아서 사용해도 좋을 것 같습니다.

2) Algorithms

(1) Brightness

: 각 픽셀의 RGB 값에 슬라이더로 받은 "factor" 의 값을 더해줍니다.

(2) Contrast

: 각 픽셀의 RGB 값에 슬라이더로 받은 "factor" 의 값을 곱해주고, 그로 인한 밝기 손실을 조정해줍니다.

(3) Saturation

: "factor" 의 값이 음수이면 각 픽셀 RGB 값 사이의 차이가 줄어들게, "factor" 의 값이 양수이면 각 픽셀 RGB 값 사이의 차이가 벌어지게 합니다.

(4) Highlights

: RGB 값이 비교적 높은 픽셀들의 밝기를 조정합니다.

(5) Shadows

: RGB 값이 비교적 낮은 픽셀들의 밝기를 조정합니다.

(6) Edge Detection

: 현재 픽셀의 RGB 값의 sum 을 오른쪽, 아래쪽에 위치한 두 픽셀의 sum 과 각각 비교합니다. 둘 중 하나라도 그 값의 차이가 threshold value 를 넘으면, 경계선으로 판단합니다.

(7) Edges

: (6) Edge Detection 의 방법으로 판단한 경계선 픽셀의 밝기 값을 조정합니다.

(8) Merge

: 우선, 유저에게 사진을 가로 / 세로 방향으로 합성할 것인지, 합성 사진의 경계선은 어디로 둘 건지 입력을 받습니다. 유저의 입력으로 받은 경계선을 기점으로 한 쪽은 "image 1" 의 픽셀 RGB 값, 다른 쪽은 "image 2" 의 픽셀 RGB 값을 가진 새로운 사진을 생성합니다.

두 사진의 경계선 부분은 "image 1" 과 "image 2" 의 값을 9:1 → … → 7:3 → … → 5:5 → … → 3:7 → … → 1:9 비율로 가지고 있는 픽셀로 넣어, 사진이 자연스럽게 변하는 듯한 효과를 주려고 했습니다.

(9) Paper 1 / Paper 2

: 현재 유저가 작업중인 사진의 픽셀과, Paper (구겨진 종이, 벽돌 무늬 종이) 사진의 같은 위치의 픽셀을 6:4 비율로 섞은 새로운 픽셀을 가진 사진을 생성합니다. 종이 질감이 더욱 도드라지도록 밝기와 채도도 조금 조정합니다.

(10) Invert

: 사진의 RGB 값을 모두 역전시킵니다.

(11) Revert to Original

: 현재 작업중인 사진의 RGB 값을 초기 값으로 되돌립니다.

(12) Magnify

: 마우스 위치를 받아, 그 부분의 이미지를 확대합니다.

```
// draw magnified area
if ((int)Math.abs(mouse_x - i) <= frame_radius - 6 && (int)Math.abs(mouse_y - j) <= frame_radius - 6) {
    r = current_pixels[(int)((i + mouse_x) / factor)][(int)((j + mouse_y) / factor)][0];
    g = current_pixels[(int)((i + mouse_x) / factor)][(int)((j + mouse_y) / factor)][1];
    b = current_pixels[(int)((i + mouse_x) / factor)][(int)((j + mouse_y) / factor)][2];
}
```

(13) Blur

: 마우스의 위치를 받아, 그 부분의 픽셀을 blur 처리 합니다.

(Blur: 해당 픽셀의 RGB 값을 "자신, 왼쪽 픽셀, 오른쪽 픽셀, 위에 위치한 픽셀, 아래에 위치한 픽셀" 총 다섯 픽셀의 RGB 값 평균으로 지정합니다.)