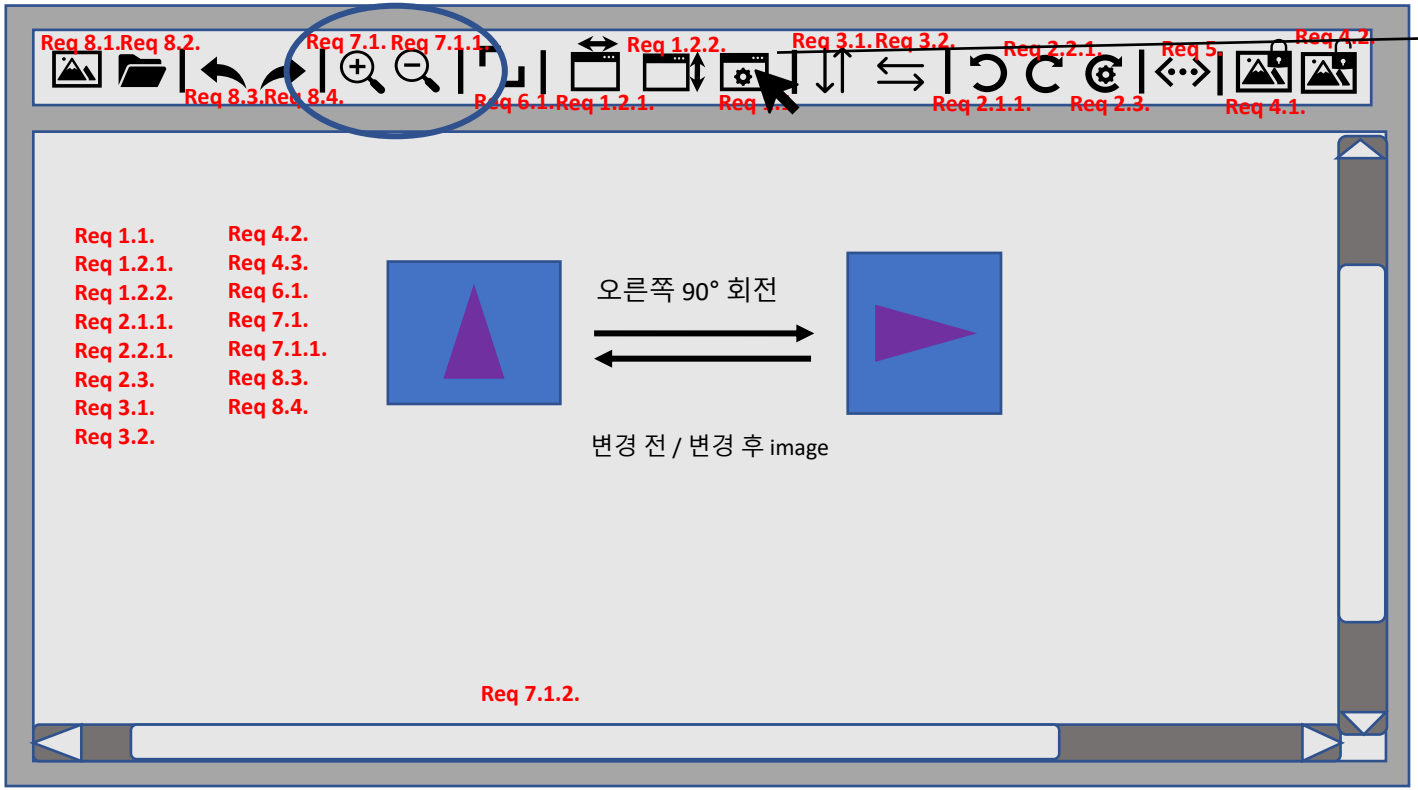
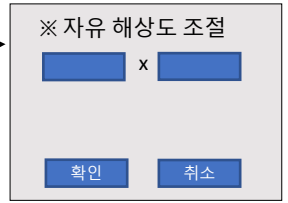
5조

Another Photo Editor

201411257 강정모
201411307 이한강
201611284 이유진

#1. 최종 산출물 모양.





위와 같이
기능 선택 시
별도의 창으로
상세 입력 필드 출력

항목 별
세부 내용

Req 1.1.
Req 1.2.1.
Req 1.2.2.
Req 2.3.
Req 4.1.
Req 5.1.
Req 5.2.
Req 5.3.
Req 5.4.
Req 5.5.
Req 5.6.
Req 8.1.
Req 8.2.

1. 실시간으로 변경이 불가 (image 편집 → 저장 → 저장한 image 불러와서 출력)
2. 기능 선택 시 옵션 입력 창과 출력 창 분리
3. 창 확대 / 축소 기능의 필요성

#2. 2차 구현물 데모

(1) JPEG decoder

Users > jungmo_k > 20_1 > g_project > construct > JPEGDecoder.py > JPEGDecoder

```
77
78  >   def JFIFParsing(self): ⋯
103
104 >   def segmentParsingAndDecoding(self): ⋯
315
316 >   def SOSParser(self): ⋯
358
359 >   def SOF0parser(self): ⋯
450
451 >   def HuffmanParser(self) : ⋯
496
497 >   def QuantumTable(self): ⋯
530
531 >   def DRIParser(self): ⋯
539
540 >   def DCcalculation(self, bit, SSSS): ⋯
557
558 >   def ACcalculation(self, bit, SSSS): ⋯
573
574 >   def makeMcuBlock(self, dc, runlist): ⋯
641
642 >   def selectMcuType(self, scan_component_id_list): ⋯
674
675 >   def RunLengthDecoder(self, prev_dc, componentID): ⋯
752
753 >   def postProcessing(self, array, componentID): ⋯
794
```

Performance 문제

→ Huffman table에 hash table 적용 & RST block 단위로 multithread 적용

→ 이미 dictionary 에서 hash table 적용 시 첫번째는 크게 영향 없을 것…

#2. 2차 구현물 데모

(2) SHA-256

```
class SHA :…

class Operation :

    def ch(self, x, y, z):…

    def maj(self, x, y, z):…

    def sig0(self, x):…

    def sig1(self, x):…

    def omega0(self, x):…

    def omega1(self, x):…

    def add(self, x, y):…

    def padding(self, x):…

    def parsing(self, x, block):…

    def ror(self, x, n):…
```

```
(base) jungmo_k@JungMo-K-MacBookPro ~ % /usr/local/anaconda3/bin/
abc
BA7816BF8F01CFEA414140DE5DAE2223B00361A396177A9CB410FF61F20015AD
(base) jungmo_k@JungMo-K-MacBookPro ~ %
```

여기 SHA256 해시하고자하는 텍스트를 붙여 넣습니다 :

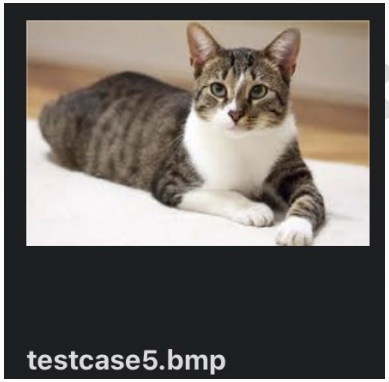
abc

당신의 SHA256 메시지 여기에서 소화 복사합니다.

BA7816BF8F01CFEA414140DE5DAE2223B00361A396177A9CB410FF61F20015AD

#2. 2차 구현물 데모

(3) 잘라내기, 자유 각도 회전, 파일 변환

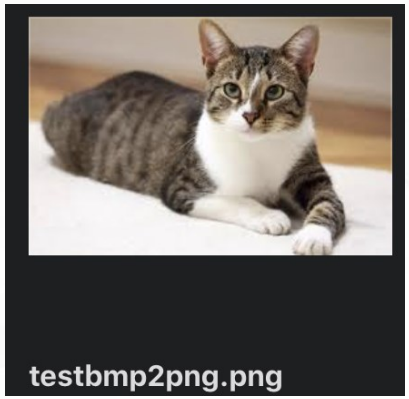


일반:
종류: Windows BMP 이미지
크기: 151,562바이트(디스크에 156KB 있음)
위치: Macintosh HD ▸ 사용자 ▸ jungmo_k ▸ test
생성일: 2020년 5월 16일 토요일 오후 5:18
수정일: 2020년 5월 16일 토요일 오후 5:18

원판 파일
잠금

추가 정보:
규격: 277×182
색상 공간: RGB
알파 채널: 아니요

이름 및 확장자:
testcase5.bmp
확장자 가리기



일반:
종류: PNG 이미지
크기: 85,210바이트(디스크에 86KB 있음)
위치: Macintosh HD ▸ 사용자 ▸ jungmo_k
생성일: 2020년 6월 3일 수요일 오후 10:05
수정일: 2020년 6월 3일 수요일 오후 10:15

원판 파일
잠금

추가 정보:
규격: 277×182
색상 공간: RGB
알파 채널: 아니요

이름 및 확장자:
testbmp2png.png
확장자 가리기

#3. System Test / Pass & Fail

Testcase Set :

width * height

(1) 800 * 600 jpg color image
(2) 1123 * 729 jpg color image
(3) 304 * 300 png color image
(4) 512 * 512 png color image
(5) 277 * 182 bmp color image
(6) 640 * 559 bmp color image

Spec	input	Expected Output	Pass / Fail	comment
1.1	(3), 가로 520, 세로 230 해상도	화면에 520 * 230 해상도를 가진 png color image 출력	P	구현 완료
1.2.1.	(2), 가로 1600 해상도	화면에 1600 * 1039 해상도를 가진 jpg color image 출력	P	구현 완료
1.2.2.	(4), 세로 539 해상도	화면에 539 * 539 해상도를 가진 png color image 출력	P	구현 완료
2.1.	(6), 왼쪽 회전 1번	화면에 왼쪽으로 90° 회전된 559 * 640 bmp color image 출력	P	구현 완료
2.2.	(5), 오른쪽 회전 3번	화면에 오른쪽으로 270° 회전된 182 * 227 bmp color image 출력	P	구현 완료
2.3.	(4), 왼쪽 회전 27°	화면에 왼쪽으로 27° 회전된 381 * 381 png color image 출력	P*	구현 완료* (513 * 513 png color image 출력)
3.1.	(1), 상하 대칭	화면에 상하 대칭된 800 * 600 jpg color image 출력	P	구현 완료
3.2.	(5), 좌우 대칭	화면에 좌우 대칭 된 277 * 182 bmp color image 출력	P	구현 완료
4.1.	(2), hello, world! 입력	다른 image editor에서는 열리지 않는 암호화된 (2) image file	△	현재 AES 256 구현 중
4.2.	4.1.의 expected output, hello, world! 입력	다른 image editor에서도 열리는 복호화된 (2) image file	△	현재 AES 256 구현 중
5.1.	(1), jpg → bmp 선택	800 * 600 해상도의 bmp color image	P	bmp encoding 구현 완료 jpg decoding 구현 완료
5.2.	(2), jpg → png 선택	1123 * 729 해상도의 png color image	P	jpg decoding 구현 완료 png encoding 구현 완료

#3. System Test / Pass & Fail

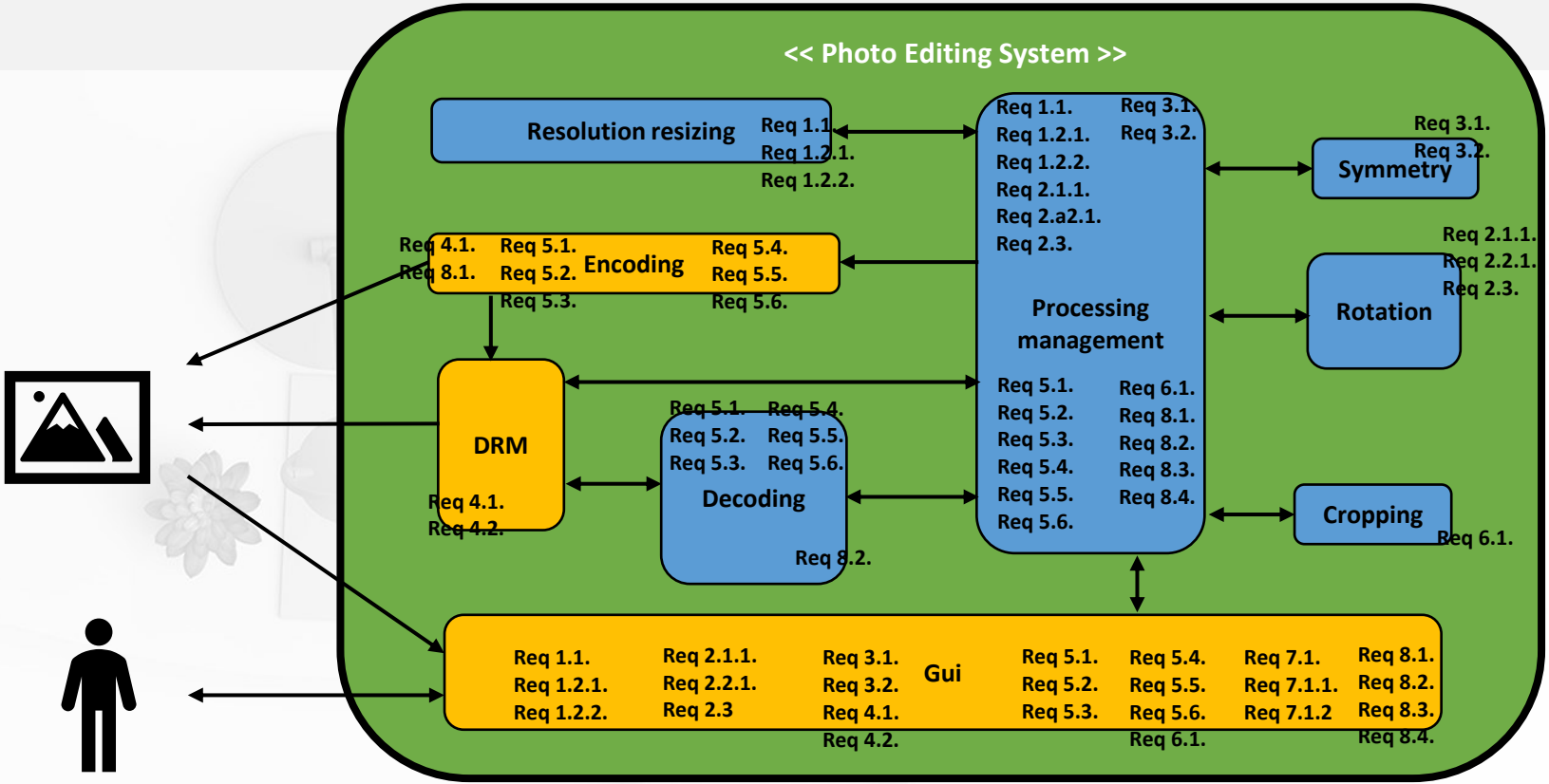
Testcase Set :

width * height

(1) 800 * 600 jpg color image
(2) 1123 * 729 jpg color image
(3) 304 * 300 png color image
(4) 512 * 512 png color image
(5) 277 * 182 bmp color image
(6) 640 * 559 bmp color image

Spec	input	Expected Output	Pass / Fail	comment
5.3.	(3), png → bmp 선택	304 * 300 해상도의 bmp color image	P	bmp encoding 구현 완료
5.4.	(4), png → jpg 선택	512 * 512 해상도의 jpg color image	F	png decoding 구현 완료 jpg encoding 구현 중
5.5.	(5), bmp → png 선택	277 * 182 해상도의 png color image	P	png encoding 구현 완료
5.6.	(6), bmp → jpg 선택	640 * 559 해상도의 jpg color image	F	bmp decoding 구현 완료 jpg encoding 구현 중
6.1.	(1), (1) 내부를 마우스 커서로 drag 한 영역	화면에 Drag한 영역 만큼의 해상도를 가지는 (1) 내부의 jpg color image 출력	△	cropping 구현 완료 GUI 구현 중
7.1.	(3)을 띄운 출력 창 선택 3번 확대	화면에 (3)이 300% (3배) 확대된 image 출력	F	GUI 구현 중
7.1.1.	키보드 → 화살표	확대된 image가 출력 창 크기를 초과하는 경우 확대된 (3)의 오른쪽 부분으로 이동하여 출력	F	GUI 구현 중
7.1.2.	(3)을 3번 확대한 출력 창 선택 1번 축소	화면에 (3)이 200% (2배) 확대된 Image 출력 (확대 단계 1단계 감소)	F	GUI 구현 중
8.1.	4.2. expected output, 원하는 directory, file name, file extension	4.2.의 expected output을 지정한 directory에 지정한 file extenstion format과 file name으로 저장	△	GUI 구현 중 AES / JPEG encoder 구현 중 저장 기능은 구현 완료
8.2.	(6), (6) file이 있는 directory 경로	(6) Image file을 창에 출력	△	GUI 구현 중 불러오기 기능은 구현 완료
8.3.	6.1. expected output	(1) 출력 (expected output 잘못 기입)	△	GUI 구현 중 기능은 구현 완료
8.4.	8.3. expected output	화면에 6.1. expected output	△	GUI 구현 중 기능은 구현 완료

#4. Component / Architecture Diagram



#5. Pass & Fail Revision + Final iteration

Testcase Set : width * height

(1) 800 * 600 jpg color image
(2) 1123 * 729 jpg color image
(3) 304 * 300 png color image
(4) 512 * 512 png color image
(5) 277 * 182 bmp color image
(6) 640 * 559 bmp color image

Spec	input	Expected Output	Deadline Pass / Fail	Comment & revision
1.1	(3), 가로 520, 세로 230 해상도	화면에 520 * 230 해상도를 가진 png color image 출력	P	구현 완료
1.2.1.	(2), 가로 1600 해상도	화면에 1600 * 1039 해상도를 가진 jpg color image 출력	P	구현 완료
1.2.2.	(4), 세로 539 해상도	화면에 539 * 539 해상도를 가진 png color image 출력	P	구현 완료
2.1.	(6), 왼쪽 회전 1번	화면에 왼쪽으로 90° 회전된 559 * 640 bmp color image 출력	P	구현 완료
2.2.	(5), 오른쪽 회전 3번	화면에 오른쪽으로 270° 회전된 182 * 227 bmp color image 출력	P	구현 완료
2.3.	(4), 왼쪽 회전 27°	화면에 왼쪽으로 27° 회전된 381 * 381 png color image 출력	P	Cropping 을 이용하여 각도에 맞는 size로 자르기 고려
3.1.	(1), 상하 대칭	화면에 상하 대칭된 800 * 600 jpg color image 출력	P	구현 완료
3.2.	(5), 좌우 대칭	화면에 좌우 대칭 된 277 * 182 bmp color image 출력	P	구현 완료
4.1.	(2), hello, world! 입력	다른 image editor에서는 열리지 않는 암호화된 (2) image file	6.10	현재 AES 256 구현 중
4.2.	4.1.의 expected output, hello, world! 입력	다른 image editor에서도 열리는 복호화된 (2) image file	6.10	현재 AES 256 구현 중
5.1.	(1), jpg → bmp 선택	800 * 600 해상도의 bmp color image	P	bmp encoding 구현 완료 jpg decoding 구현 완료
5.2.	(2), jpg → png 선택	1123 * 729 해상도의 png color image	P	jpg decoding 구현 완료 png encoding 구현 완료

#5. Pass & Fail Revision + Final iteration

Testcase Set : width * height

(1) 800 * 600 jpg color image
(2) 1123 * 729 jpg color image
(3) 304 * 300 png color image
(4) 512 * 512 png color image
(5) 277 * 182 bmp color image
(6) 640 * 559 bmp color image

Spec	input	Expected Output	Deadline Pass / Fail	Comment & revision
5.3.	(3), png → bmp 선택	304 * 300 해상도의 bmp color image	P	bmp encoding 구현 완료
5.4.	(4), png → jpg 선택	512 * 512 해상도의 jpg color image	6.10	png decoding 구현 완료 jpg encoding 구현 중
5.5.	(5), bmp → png 선택	277 * 182 해상도의 png color image	P	png encoding 구현 완료
5.6.	(6), bmp → jpg 선택	640 * 559 해상도의 jpg color image	6.10	bmp decoding 구현 완료 jpg encoding 구현 중
6.1.	(1), (1) 내부를 마우스 커서로 drag 한 영역	화면에 Drag한 영역 만큼의 해상도를 가지는 (1) 내부의 jpg color image 출력	△	cropping 구현 완료 PyQT 상에서 제공되지 않을 시 별도의 방식으로 영역 입력 받을 예정
7.1.	(3)을 띄운 출력 창 선택 3번 확대	화면에 (3)이 300% (3배) 확대된 image 출력	???	PyQT 상에서 해당 기능 제공 X → resolution resizing으로 제공 (따라서 req 내용 변경 여지 있음)
7.1.1.	키보드 → 화살표	확대된 image가 출력 창 크기를 초과하는 경우 확대된 (3)의 오른쪽 부분으로 이동하여 출력	???	해당 기능이 PyQT 상에서 제공되지 않을 시 마우스로 스크롤 바 이동
7.1.2.	(3)을 3번 확대한 출력 창 선택 1번 축소	화면에 (3)이 200% (2배) 확대된 Image 출력 (확대 단계 1단계 감소)	???	PyQT 상에서 해당 기능 제공 X → resolution resizing으로 제공 (따라서 req 내용 변경 여지 있음)
8.1.	4.2. expected output, 원하는 directory, file name, file extension	4.2.의 expected output을 지정한 directory에 지정한 file extenstion format과 file name으로 저장	△	GUI 구현 중 AES / JPEG encoder 구현 중 저장 기능은 구현 완료
8.2.	(6), (6) file이 있는 directory 경로	(6) Image file을 창에 출력	△	GUI 구현 중 불러오기 기능은 구현 완료
8.3.	6.1. expected output	(1) 출력 (expected output 잘못 기입)	△	GUI 구현 중 / 기능은 구현 완료
8.4.	8.3. expected output	화면에 6.1. expected output	△	GUI 구현 중 / 기능은 구현 완료