

AAC - Ascii Art Converter

No More Double

조다니엘, 박준영

Sogang University

CSE2035/AIE2051

2023년 12월 3일



목차

① 왜 만들었는가?

② 무엇으로 만들었는가?

- eg

- `img2ascii`

- 기타

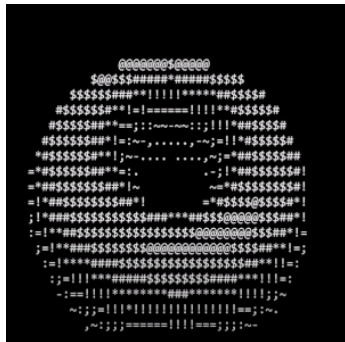
③ 어떻게 동작하는가?

- tone-based 방식

- structure-based 방식

왜 만들었는가?

- AAC(악)이라는 이름이 너무 찰져서
- YouTube에서 donut.c라는 아스키 아트를 봤는데 멋있어서¹



¹https://www.youtube.com/watch?v=DEqXNfs_HhY

무엇으로 만들었는가?

- ① 커스텀 라이브러리: `eg`
- ② 메인 루틴: `img2ascii`
- ③ 기타: GitHub, Valgrind, ...

EasypnG라는 뜻

libpng + **Eigen**으로 구성

이미지 입출력, 변환, 연산 등에 필요한 클래스를 내장

- egLoader: 이미지 입출력
- egMath: 변환에 필요한 연산
- egProcessing: 이미지 변환
- 등등... / egTypes나 egExceptions 같은 사소한 건 빼기

eg 라이브러리를 활용하여 이미지를 처리하는 메인 루틴

tone-based 방식과 structure-based 방식으로 나뉨

(tone-based 방식 결과물 사진 한 장, structure-based 방식
결과물 사진 한 장 넣기)

- GitHub: 소스 코드 버전 관리 플랫폼
(Github 이미지 넣기)
- Valgrind: 메모리 누수 탐지 라이브러리
(Valgrind 메모리 누수 탐지 cmd창 이미지 넣기)

어떻게 동작하는가?

- tone-based 방식 - easy
- structure-based 방식 - EXTREMELY DIFFICULT

tone-based 방식

- ① 이미지를 3차원 텐서 ($width, height, rgba$) 형태로 저장
- ② 각 픽셀별 red, green, blue 값의 평균을 구하여 밝기 도출
- ③ 밝기 정보를 이용하여 이미지를 회색조로 변환
- ④ 밝기에 해당하는 아스키 문자를 출력

결과

(처음 input 사진 한 장 → 아스키 아트 사진 한 장)

structure-based 방식

- ① 이미지를 3차원 텐서 (*width, height, rgba*) 형태로 저장
- ② 알고리즘 1 적용
- ③ 알고리즘 2 적용
- ④ 알고리즘 3 적용
- ⑤ 알고리즘 4 적용
- ⑥ 알고리즘 5 적용
- ⑦ 알고리즘 6을 적용하여 아스키 문자를 선택
- ⑧ 결과 출력

알고리즘 1

- ① 알고리즘 1 내용 설명 1 (1줄씩)
- ② 알고리즘 1 내용 설명 2 (1줄씩)

(알고리즘 1로 이미지가 어떻게 변했는지
before & after 사진 넣기)

(나머지 알고리즘도 이런 식으로 반복하면 될 듯)

결과

(처음 input 사진 한 장 → 아스키 아트 사진 한 장)

참고 자료

① 자료 1

② 자료 2

③ 자료 3

(인용 양식 지켜서 쓰면 교수님 호감 → 대학원생으로 바로 납치)

감사합니다.