

MiniCV

bitmap, image processing



라 팀

keokim

minjkim2

hyopark



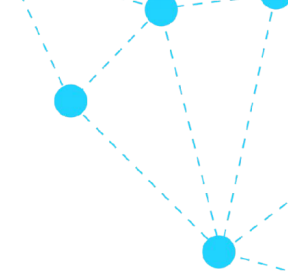


CONTENTS

42서울 2022
EDUTHON

MiniCV

00	개요	_____	3P
01	과제 설명	_____	5P
02	해설	_____	17P
03	목적 및 기대효과	_____	23P
04	발전가능성	_____	27P

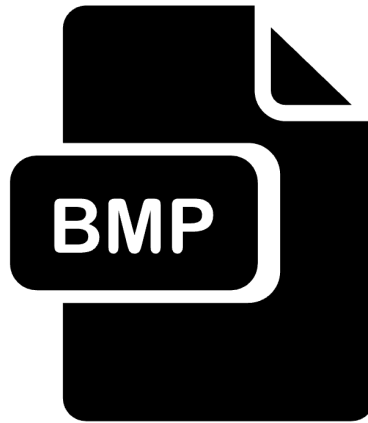


개요

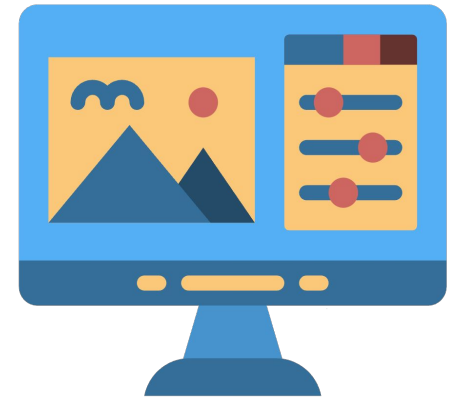




c언어 기반



비트맵 구조



이미지 프로세싱 기초

과제 설명

01



Dash - MiniCV

MiniCV

Summary: bitmap, image processing

Contents

I	Introduction	2
II	General rules	3
III	Mandatory part	4
III.1	Exercise 00: BGB to BGR	4
III.2	Exercise 01: Upside down / Right to Left	6
IV	Bonus part	7

Chapter I

Introduction

Image processing is already used in many fields. Medical field, Transmission and encoding, Robot vision, Pattern recognition etc. Did you know that you can handle bitmap images in C without using OpenCV? This project offers you an opportunity to learn about bitmap structure and image processing basics. We will provide the bitmap image you will use for the project.

The beautiful palace in the picture is Gyeongbokgung Palace in South Korea. Gyeongbokgung Palace was built in 1395, burned down by the war in 1592, and re-built in 1868. Although most of the buildings in the palace have disappeared, the main buildings remain, and it is an important historical site where you can check the appearance of the royal palace in South Korea. However, the image of this beautiful Gyeongbokgung Palace is ruined. You need to restore this image.



Chapter II


General rules

- Your project must be written in according with the Norm. If you have bonus files/ functions, they are included in the norm check and you will receive a 0 if there is a norm error inside.
- Your functions should not quit unexpectedly (segmentation fault, bus error, double free, etc) apart from undefined behaviors. If this happens, your project will be considered non functional and will receive a 0 during the evaluation.
- All heap allocated memory space must be properly freed when necessary. **No leaks will be tolerated.**
- If the subject requires it, you must submit a Makefile which will compile your source files to the required output with the flags -Wall -Wextra and -Werror, and your Makefile must not relink.
- Your Makefile must at least contain the rules \$(NAME), all, clean, fclean, and re.
- We encourage you to create test programs for your project even though this work won't have to be submitted and won't be graded. It will give you a chance to easily test your work and your peers' work. You will find those tests especially useful during your defence. Indeed, during defence, you are free to use your tests and/or the tests of the peer you are evaluating.

Chapter III

Mandatory part

Exercise 00: RGB to BGR

	Exercise : 00
Program name : RGB2BGR	
Turn-in directory : ex00/	
File to turn in : Makefile, *.c */*.c, *.h, */*.h	
Allowed functions : fopen, fread, fwrite, fclose, fprintf, fseek, malloc, free, exit	
Description : Restore the given bmp file to the original image	

the color of the picture has changed.

Write a program that restores the original picture.

Your program must save the **original.bmp** file.

The evaluation is conducted by comparing the original bitmap file with the recovered bitmap file.

Apply the same rule to all subsequent exercises.

You have to use your own bitmap structure.



bitmap file header / bitmap info header
Do you know the -fpack-struct compiler flag?

과제 설명 - ex 00

42서울 2022
EDUTION



Exercise 01: Upside down / Right to left



Exercise : 01

Program name : reverse

Turn-in directory : ez01/

File to turn in : Makefile, *.c, *.h, */*.h


Allowed functions : fopen, fread, fwrite, fclose, fprintf, fseek, malloc, free, exit

Description : Restore the given bmp file to the original image



Bonus part

Exercise 02: Zoom in

	Exercise : 02
Program name : zoom	
Turn-in directory : ex02/	
File to turn in : Makefile, *.c */*.c, *.h, */*.h	
Allowed functions : fopen, fread, fwrite, fclose, fprintf, fseek, malloc, free, exit	
Description : Write a program that magnifies a original bitmap file	


You can pass this dash-project without doing exercise 02.

Don't spend too much time on bonus.

There are many ways to zoom in, but the choice is freedom.

Write a program that stores a given bitmap file to double magnification.

과제 설명 - 제한사항

	Exercise : 02
Program name : zoom	
Turn-in directory : ex02/	
File to turn in : Makefile, *.c */*.c, *.h, */*.h	
Allowed functions : fopen, fread, fwrite, fclose, fprintf, fseek, malloc, free, exit	
Description : Write a program that magnifies a original bitmat file	

비트맵 구조에 대한 이해가 중요

허용 함수를 제한하고 구조체를 직접 구현하여 사용하도록 제한

추가로 **FILE** 스트림을 이용한 입출력을 이용하도록 유도

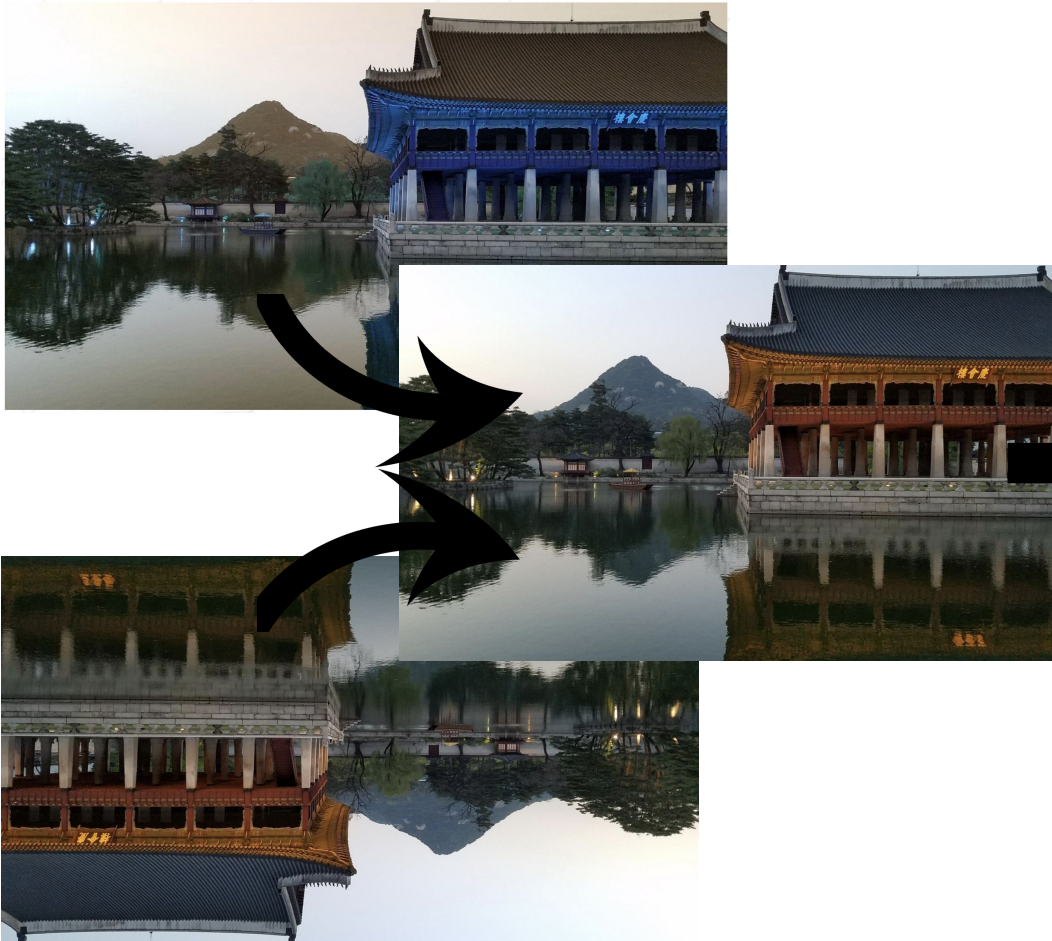


제출 C 파일 기계채점



보너스 파트 동료평가

과제 설명 - 기계채점



```
Minjun ▶ eduton/solve ▶ ./test.sh
=====Norm Test=====
Norm pass
=====ex00=====
ex00 compile pass
ex00 diff pass
=====ex01=====
ex01 compile pass
ex01 diff pass
=====ex02=====
ex02 compile pass
test all passed :)
```


해설

02

비트맵 , 비트맵 구조

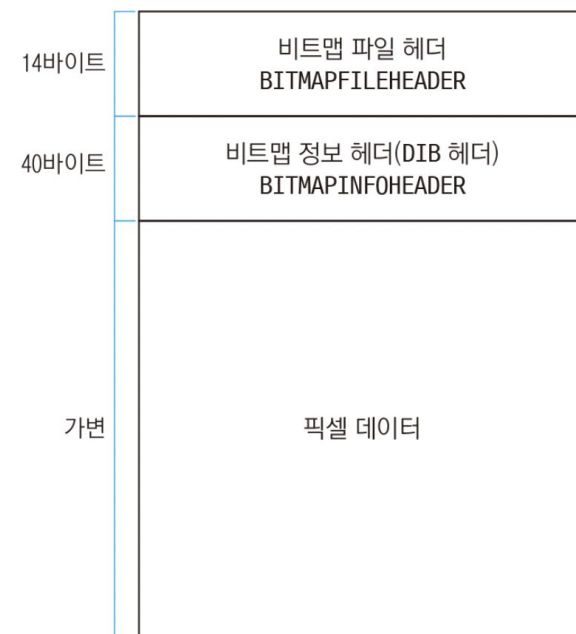
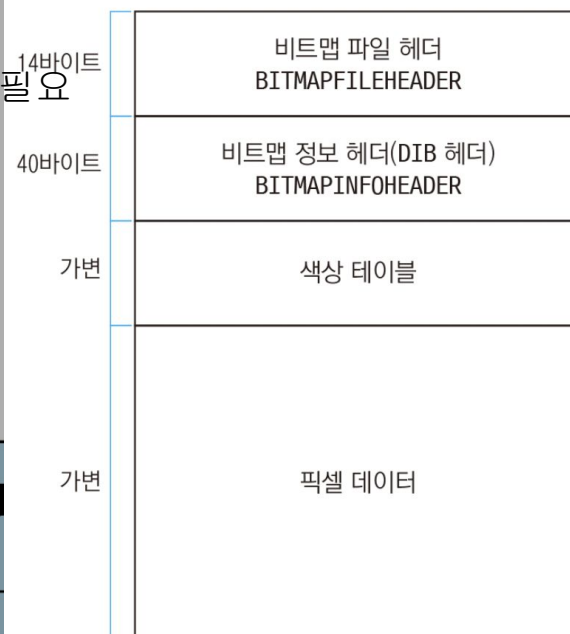
BMP 비트맵의 줄임말

바이너리 형식으로 저장

비트맵 파일의 구조에 대한 지식 필요

가장 많이 쓰는 **24비트** 비트맵

24비트가 한 픽셀



FILE 스트림을 이용한 파일 입출력 제어

파일 출력

'fread()'로 읽은 내용을 비트맵 파일
헤더와 비트맵 정보 헤더 구조체에

저장

'fread()'를 다시 사용하여 이미지
크기만큼 할당한 **image** 변수에

읽음

파일 입력

'fopen()'로 파일을 열고

'fwrite()'로 **image** 변수에 씀

color

비트맵은 1픽셀에 3가지 색상
정보가 **BGR** 순으로 저장

모든 픽셀의 색상 정보를 **RGB**
순으로 수정하면 색상이
좌우반전되어 표현 됨

! **BMP** 파일을 제공하면 프로젝트 진행자는 **BGR**
순으로 색상을 돌려 원래 이미지의 색상을 복구해야함.



상하좌우 반전

이미지의 정보가 담겨있는 `image`
변수 복사

상하반전

`image`를 세로로 반절 나눠
데이터들을 `swap`

좌우반전

`image`를 가로로 반절 나눠
데이터들을 `swap`

! 반전시 색상 정보가 **BGR** 순으로 저장해야 색상이 바뀌지 않음

확대 (scaling)

원본과 똑같은 크기의 이미지
변수에 **N**배만큼 확대하여 저장

원본의 한 픽셀을 **N**의 제곱만큼
복사

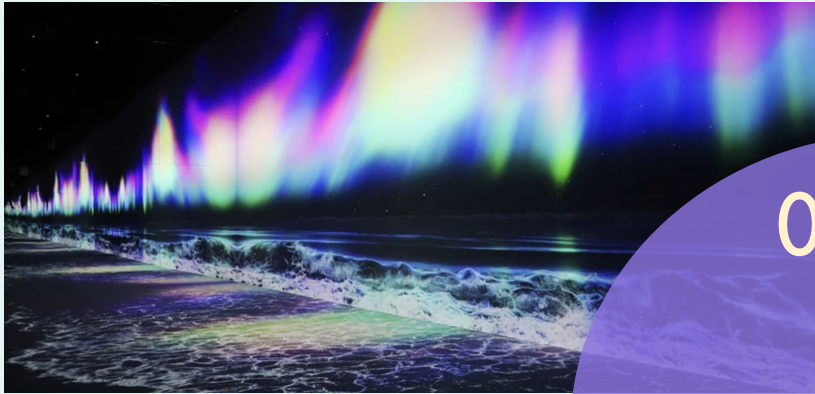
! 비트맵의 특성상 색상 데이터가 반대로
저장



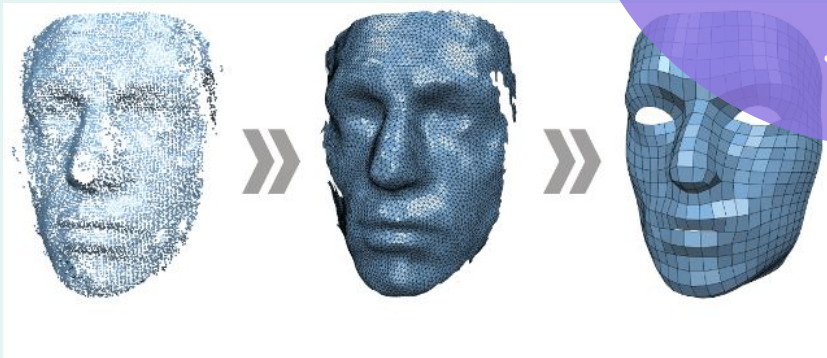
목적 및 기대효과

03

목적 및 기대효과 - 배경



이미지
처리
기술
관심도
증가



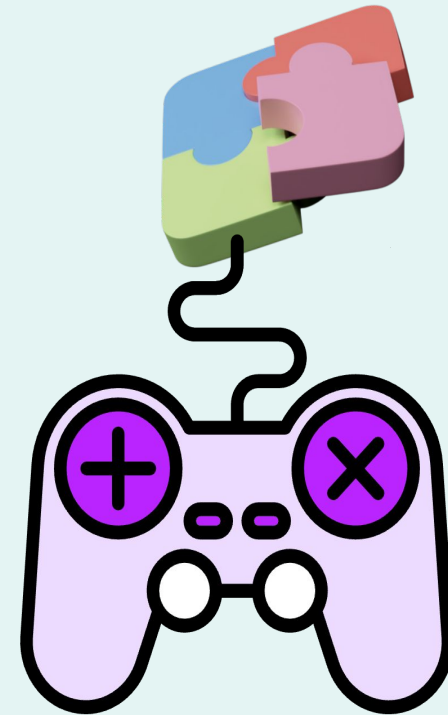
목적 및 기대효과 - 목적



목적 및 기대효과 - 기대효과



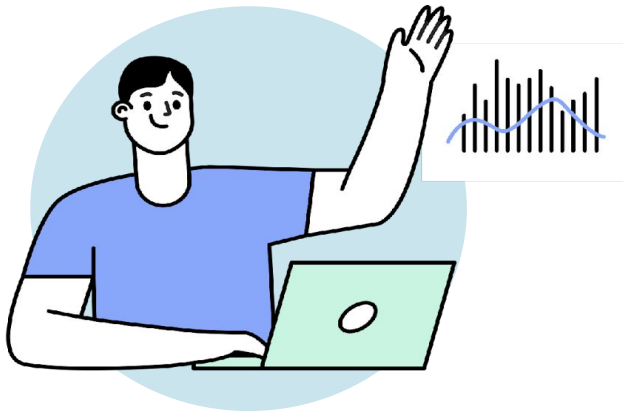
시각적 요소 제공
문제 풀이에 흥미 유발



망가진 이미지를 고치는
게임적 요소

발전 가능성

04



심화 학습

최근점 보간법, ZOI

양선형 보간법 FOI

Bicubic Interpolation

Lanczos Interpolation

+



공통과정

그래픽 과제

Fdf
so long
fract-ol
minRT
cub3d

감사합니다 Q&A

