



리얼리티 스톤 앱 만들기2

Elly

How To..

1. 삽입

- ① EOI(End Of Image) 바이트(=FF D9) 뒤에 넣기
- ② 파일 헤더(FFE0 - FFEF)에 넣기

2. 변조

LSB를 변경 (RGB 각 끝자리)

- JPEG, BMP 같은 24비트 이미지파일에 적용

변조 - LSB를 변경



Cover pixel: (167, 93, 27)

Secret pixel: (67, 200, 105)

(**10100111**, **01011101**, **00011011**) (**01000011**, **11001000**, **01101001**)



Output pixel: (162, 94, 27) == (101000**10**, 01011**110**, 00011**011**)

코드로 작성해보자

1. 이미지를 RGB값으로 가져오기 (coverImage, secureImage)
2. coverImage를 원하는 bit만큼 RGB값 치환하기
3. 치환한 pixel을 다시 이미지로 만들어서 view에서 보여주기

1. 이미지를 RGB값으로 가져오기

```
class RGBPixelMaker {
    var size: CGSize = CGSize(width: 0, height: 0)
    func makeBinaryPixel(image: UIImage) -> [RGBData] {
        var imageOfRGBData: [RGBData] = []
        size = CGSize(width: image.size.width, height: image.size.height)
        let pixelData = image.cgImage?.dataProvider?.data
        let data: UnsafePointer<UInt8> = CFDataGetBytePtr(pixelData)

        // from image to pixel
        for x in 0..
```

2. coverImage의 RGB값 하위 bit를 치환하기

```
func makebitMixing(imageA: [RGBData], imageB: [RGBData], bit: Int) -> [RGBData] {  
    // 두 이미지 중 작은 값을 기준으로  
    var imageOfMixing: [RGBData] = []  
    let bitMask: CFBit = 11111111  
    for index in 0..  
imageA.count {  
        let resultR = CFBit(imageA[index].r & bitMask << bit | imageB[index].r >> (8 - bit))  
        let resultG = CFBit(imageA[index].g & bitMask << bit | imageB[index].g >> (8 - bit))  
        let resultB = CFBit(imageA[index].b & bitMask << bit | imageB[index].b >> (8 - bit))  
        let resultA = CFBit(imageA[index].a)  
        imageOfMixing.append(RGBData.init(r: resultR, g: resultG, b: resultB, a: resultA))  
    }  
    return imageOfMixing  
}
```

문제점

- 이미지 처리하는 데 너무 오래 걸린다.
- 픽셀을 다시 이미지화 하는 데 에러가 난다.
- 어렵다 π

- $\frac{\pi}{E}$ > _ < -