

사진 조명 시스템 개발

20046 박건호

목차

- ▶ 1. 개요
- ▶ 2. 프로젝트의 목표 필요성
- ▶ 3. 프로젝트 수행 과정
- ▶ 4. 실행 화면
- ▶ 5. 기대효과
- ▶ 6. 느낀점

1. 개요

▶-> 단색인 이미지나 평면느낌의 이미지를
입체적인 느낌이 되도록 바꿔보자!

▶-> 디자인 효과

2. 프로젝트 목표

▶-> 원할때 이미지에 조명을 많이 추가하게 한다.

▶-> 모든 조명의 특성이 모두 나타나도록 코드를 구성한다.

3. 프로젝트 수행 과정

- ▶ lambert light -> 실제로 많은 프로그램에서 쓰고 있는 알고리즘
ex) obj mtl 파일 표현



3. 프로젝트 수행 과정

- ▶ 전체적인 평균 색상을 알려주는 ambient 색깔
- ▶ 물체의 조명의 위치에 따라 부분마다 다른 diffuse 색깔
- > specular 색깔은 생략

식은 google 참고

1. ambient 색깔만 반영하면 최종 색깔은 다음과 같은 식으로 나타난다.

$$\text{ambientcolor} = \text{materialcolor} * \text{ambient},$$

2. diffuse 색깔까지 반영하면 최종 색깔은 다음과 같은 식으로 나타난다.

$$\text{lightvector} = \text{light position} - \text{object position}$$

$$\text{cosine} = \text{dot product}(\text{object normal vector}(0), \text{normalized light vector})$$

$$\text{lam bert factor} = \max(\text{cosine}, 0)$$

$$\text{luminosity} = \frac{1}{1 + \text{distance} * \text{distance}}$$

$$\text{dif fuse color} = \text{material color} * \text{light color} * \text{lam bert factor} * \text{luminosity}$$

$$\text{final color} = \text{ambient color} + \text{dif fuse color}$$

3. 프로젝트 수행 과정

▶ 이미지 불러오기 && 내보내기 ??

-> 픽셀을 이용하자!

PixelReader PixelWriter BufferedReader WritableImage등의
java api이용

3. 프로젝트 수행 과정

- ▶ 이미지의 픽셀을 따오기 위해서 `BufferedReader`라는 클래스를 사용하였다.
- ▶ `BufferedImage buffered= SwingFXUtils.fromFXImage(image, null);`
- ▶ `hasAlphaChannel= buffered.getAlphaRaster() != null;`
- ▶ 그리고 픽셀의 각각 값을 알기 위해서는 `PixelReader`라는 클래스를 만들어야 한다.
- ▶ `pixelReader= image.getPixelReader();`

3. 프로젝트 수행 과정

- ▶ `if(hasAlphaChannel) else`
- ▶ `pixels[pos++] = (int) color.getBrightness(); // Alpha`
- ▶ `else argb+= -16777216; // 255 alpha`
- ▶ `pixels[pos++] = (int) (color.getBlue()*255); // Blue`
- ▶ `pixels[pos++] = (int) (color.getGreen()*255); // Green`
- ▶ `pixels[pos++] = (int) (color.getRed()*255); // Red`

3. 프로젝트 수행 과정

- ▶ `newimage= newWritableImage(width,height);`
- ▶ `write= newimage.getPixelWriter();` <-PixelWriter
- ▶ `for(int i=0;i<width;i++){`
- ▶ `for(int j=0;j<height;j++){`
- ▶ `MakeImage.setPixel(i,j,Color.rgb(255,255,255));`
- ▶ `}`
- ▶ `}`

3. 프로젝트 수행 과정

light list

new

delete

Before

After

load

save

light property

exit

lightX

lightY

lightZ

power

light color

lightR

lightG

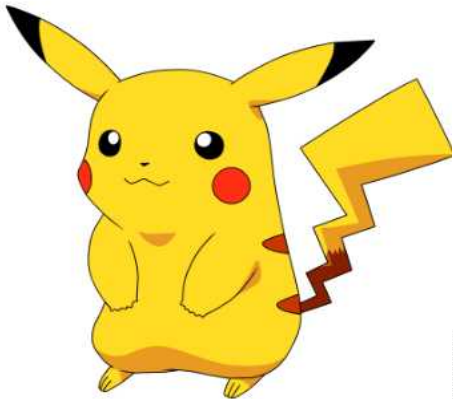
lightB

ambient

3. 프로젝트 수행 과정

- ▶ load: 이미지를 파일탐색기를 이용하여 불러온다
- ▶ save: 옆에 있는 TextField 의 이름으로 png파일로 조명이 반영된 이미지가 저장된다.
- ▶ lightlist: 조명을 추가하고 제거할 수 있다.
- ▶ 이 list에서 설정을 바꿀 조명을 선택하고 lightproperty에서 값을 변하게 하면 조명에 대한 정보가 저장된다.

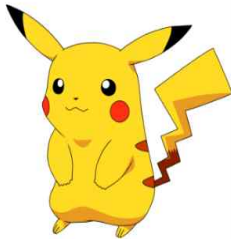
4. 실행 화면



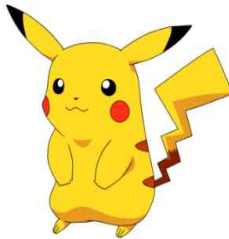
이미지 불러오기

light list

Before



After



light property

lightX

lightY

lightZ

power

light color

lightR

lightG

lightB

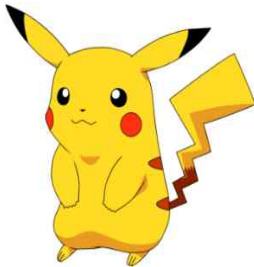
ambient

조명 1개 추가

light list new delete

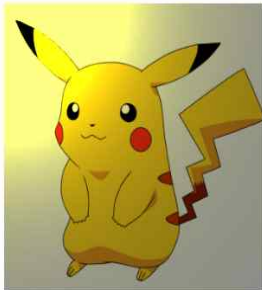
light1

Before



load

After



save

light property exit

lightX 0

lightY 0

lightZ 100

power

light color

lightR 255

lightG 255

lightB 0

ambient

조명 많이 추가

light list

new

delete

light1

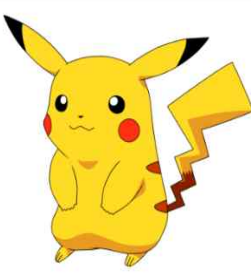
light2

light3

light4

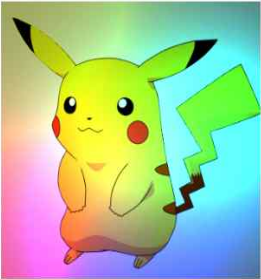
light5

Before



load

After



save

light property

exit

lightX

150

lightY

200

lightZ

100

power

light color

lightR

255

lightG

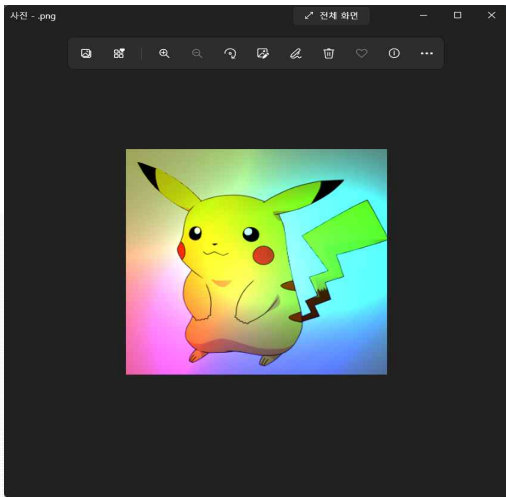
255

lightB

255

ambient

저장 화면



감사합니다.