

BÁO CÁO CÁ NHÂN

Nhóm: Game và những con nghiện

Họ và tên: Lê Tấn Lộc

MSSV: 4501104135

Giảng viên hướng dẫn: TS. Nguyễn Đỗ Thái Nguyên

1 Ánh sáng

1.1 Đối với đối tượng có Texture

```
FragColor = texture(texture_diffuse1, TexCoords);
```

Nếu đối tượng có Texture thì sẽ gán thành màu họa tiết có tọa độ họa tiết là `TexCoords` trong mẫu họa tiết `texture_diffuse1`. Nếu điểm đó không có Texture sẽ được trong phần 1.2

1.2 Đối với đối tượng không có Texture

Cách chiếu sáng được kết hợp dựa trên ba mô hình: Ambient lighting, Diffuse lighting, Specular Lighting.

1.2.1 Ambient lighting

Mục đích: Dùng để chiếu sáng toàn cầu (global illumination), nghĩa là ánh sáng được chiếu từ mọi phía và phản xạ lại màu sắc của vật đối tượng đó.

Ý tưởng: Chỉ cần tô màu lên toàn bộ đối tượng với màu đã chọn.

```
float ambientStrength = 0.1;  
vec3 ambient = ambientStrength * lightColor;
```

Trong đó:

`ambientStrength`: Dùng để điều chỉnh cường độ của màu sắc, chỉ số càng nhỏ cường độ sáng càng lớn.

`lightColor`: Màu của nguồn sáng (thang màu RGB)

1.2.2 Diffuse lighting

Mục đích: Để hiển thị những điểm nào gần với nguồn sáng thì màu sắc sẽ sáng hơn, những điểm nào xa nguồn sáng thì màu sắc sẽ tối đi.

Ý Tưởng: Tại mỗi điểm ta sẽ dựng một vector pháp tuyến tại điểm đó sau tính vector chỉ hướng sáng từ điểm đó tới nguồn sáng, góc hợp bởi vector pháp tuyến và vector chỉ hướng sáng càng nhỏ màu sẽ sáng hơn, góc càng lớn mà sẽ càng tối đi.

```
vec3 norm = normalize(Normal);  
vec3 lightDir = normalize(lightPos - FragPos);  
float diff = max(dot(norm, lightDir), 0.0);  
vec3 diffuse = diff * lightColor;
```

Trong đó:

Hàm `normalize()` dùng để chuyển một vector thành vector đơn vị để dễ cho việc tính toán về sau.

`Normal` là vector pháp tuyến. => `norm` vector pháp tuyến đã chuyển về vector đơn vị

`lightPos` là vị trí nguồn sáng.

`FragPos` là vị trí điểm cần chiếu sáng.

`lightDir` là vector chỉ hướng sáng (cũng là vector đơn vị) bằng cách tính hiệu hai vector `lightPos` và `FragPos` (vector hướng từ `FragPos` đến `lightPos`)

`diff` là góc hợp bởi hai vector pháp tuyến (`norm`) và vector chỉ hướng sáng `s(lightDir)` bằng cách tính tích vô hướng hai vector này. Nếu góc này mang giá trị âm thì khó mà xác định cường độ sáng sẽ như thế nào, nên ta lấy cường độ ánh sáng là 0 (không có ánh sáng chiếu tới).

1.2.3 Specular Lighting

Mục đích: Làm nổi bật những điểm nào mà mắt người đang quan sát.

Ý Tưởng: Tại mỗi điểm ta sẽ dựng một vector pháp tuyến tại điểm đó sau tính vector chỉ hướng sáng từ điểm đó tới nguồn sáng, lấy đối xứng vector chỉ hướng sáng qua vector pháp tuyến ta được vector phản xạ, tiếp đó tính vector hướng nhìn, góc được tạo bởi vector phản xạ và vector hướng nhìn càng nhỏ màu sẽ sáng hơn, góc càng lớn mà sẽ càng tối đi.

```
float specularStrength = 0.5;
vec3 viewDir = normalize(viewPos - FragPos);
vec3 reflectDir = reflect(-lightDir, norm);
float spec = pow(max(dot(viewDir, reflectDir), 0.0), 32);
vec3 specular = specularStrength * spec * lightColor; |
```

`specularStrength`: Cường độ sáng.

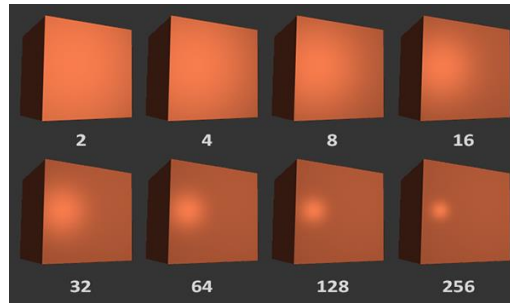
`viewPos`: vị trí quan sát.

`viewDir` vector hướng nhìn

Hàm `reflect()` Dùng để lấy vector phản xạ từ vector chỉ hướng (`lightDir`) sáng qua vector pháp tuyến (`norm`), vì ban đầu vector chỉ hướng sáng đi từ `FragPos` đến `lightPos`, ngược chiều phản xạ nên `lightDir` phải thêm dấu trừ để đổi chiều vector.

`reflectDir` vector phản xạ

`spec` sau khi lấy góc được tạo bởi vector phản xạ (`reflectDir`) và vector hướng nhìn (`viewDir`), cần lũy thừa 32 để mở rộng vùng sáng. Ví dụ bên dưới



1.2.4 Kết hợp

```
vec3 result = (ambient + diffuse + specular) * objectColor;  
FragColor = vec4(result, 1.0);|
```

Khi ta kết hợp ba mô hình Ambient lighting, Diffuse lighting, Specular Lighting. Sẽ được gọi là mô hình Phong.

`FragColor` là kết quả cuối cùng ta nhận được `1.0` chỉ số thang màu alpha (độ trong suốt)

2 Tài liệu tham khảo

[1] LearnOpenGL - Basic Lighting, link: <https://learnopengl.com/Lighting/Basic-Lighting>