

# 1 Wstęp teoretyczny

Efekty zależne od kąta patrzenia opierają się na zależności pomiędzy kierunkiem obserwacji a orientacją powierzchni obiektu. W praktyce oznacza to, że wygląd materiału może zmieniać się w zależności od tego, pod jakim kątem obserwator patrzy na daną powierzchnię. Zjawisko to jest często wykorzystywane do uzyskiwania efektów poświaty na krawędziach, stylizowanych obrysów, osłon energetycznych lub powierzchni o właściwościach zbliżonych do szkła, wody czy metalu.

Jednym z najczęściej stosowanych efektów tego typu jest efekt Fresnela. Opiera się on na porównaniu normalnej powierzchni z kierunkiem widzenia. Gdy obserwator patrzy na powierzchnię niemal prostopadle, efekt jest słabszy, natomiast przy obserwacji pod kątem stycznym staje się wyraźniejszy. W rezultacie krawędzie obiektu mogą być jaśniejsze lub bardziej nasycone niż jego część środkowa.

W shaderze efekt ten uzyskuje się zwykle przez obliczenie iloczynu skalarnego pomiędzy znormalizowaną normalną powierzchni a znormalizowanym wektorem widzenia. Odpowiednie przekształcenie otrzymanej wartości pozwala sterować intensywnością efektu. Dzięki temu możliwe jest tworzenie materiałów reagujących na pozycję kamery bez konieczności modyfikowania geometrii obiektu.

## 2 Zadania

### Zadanie 1

Napisz shader, w którym wartość efektu Fresnela zostanie obliczona na podstawie iloczynu skalarnego pomiędzy znormalizowaną normalną powierzchni a znormalizowanym wektorem widzenia. Następnie wykorzystaj tę wartość do sterowania jasnością koloru obiektu. Spraw, aby krawędzie modelu były wyraźnie jaśniejsze niż jego środkowa część.

```
float3 N = normalize(i.worldNormal);
float3 V = normalize(_WorldSpaceCameraPos - i.worldPos);
float fresnel = 1.0 - saturate(dot(N, V));
```

### Zadanie 2

Rozszerz rozwiązanie z Zadania 1 o dwa parametry materiału: kolor efektu oraz wykładnik odpowiedzialny za ostrość przejścia. Następnie dobrać wartości parametrów tak, aby uzyskać dwa różne rezultaty: miękką poświatę na krawędziach oraz wąski, wyraźny obrys. W obu przypadkach wykorzystaj ten sam shader.

```
float fresnel = 1.0 - saturate(dot(N, V));
fresnel = pow(fresnel, _Power);
```

### Zadanie 3

Zmodyfikuj shader tak, aby końcowy kolor obiektu był wynikiem połączenia koloru bazowego i efektu Fresnela. Środkowa część obiektu ma zachować kolor bazowy, natomiast efekt zależny od kąta patrzenia ma wzmacniać wyłącznie fragmenty krawędziowe. Sprawdź działanie shadera dla co najmniej dwóch różnych obiektów 3D i porównaj widoczność efektu.

```
fixed3 finalColor = _BaseColor.rgb + _FresnelColor.rgb * fresnel;
```

### Zadanie 4

Rozszerz shader o animację intensywności efektu w czasie. Zmiana ma być okresowa i powinna prowadzić do uzyskania wrażenia pulsującej poświaty. Animacja nie może zmieniać geometrii obiektu, a jedynie wartość efektu obliczanego we fragmencie shaderze.

```
float pulse = 0.5 + 0.5 * sin(_Time.y * _Speed);  
float animated = fresnel * pulse;
```

### Zadanie 5

Przygotuj materiał imitujący osłonę energetyczną lub pole siłowe. W tym celu połącz efekt Fresnela z częściową przezroczystością albo emisją koloru. Końcowy rezultat powinien wyraźnie reagować na zmianę kąta patrzenia i przypominać stylizowany efekt stosowany w grach komputerowych.

```
fixed4 col;  
col.rgb = _BaseColor.rgb + _FresnelColor.rgb * fresnel;  
col.a = fresnel;
```