

1 Wstęp teoretyczny

Animacja w shaderze fragmentów polega na wyznaczeniu koloru każdego piksela na podstawie jego położenia oraz czasu. W przeciwieństwie do vertex shadera nie zmienia się tutaj geometria obiektu, lecz bezpośrednio obraz wynikowy. Dzięki temu możliwe jest tworzenie dynamicznych wzorów, fal, przejść kolorów i prostych efektów proceduralnych określanych jako shader art lub art coding.

Podstawą tego podejścia jest wykorzystanie współrzędnych fragmentu oraz operacji matematycznych, takich jak przesunięcie, skalowanie, obliczanie odległości czy funkcje trygonometryczne. Dodanie parametru czasu pozwala animować uzyskany obraz. Technika ta umożliwia tworzenie efektów wizualnych bez użycia złożonej geometrii i tekstur.

2 Zadania

Zadanie 1

Zaimplementuj w shaderze fragmentów efekt proceduralnych okręgów widocznych na powierzchni obiektu 2D lub 3D. Okręgi mają być wyznaczone wyłącznie na podstawie współrzędnych UV, bez użycia tekstury. Rozwiązanie powinno pozwalać na wygenerowanie co najmniej jednego wyraźnie widocznego okręgu.

```
float2 uv = i.uv - 0.5;
float d = length(uv);
float circle = step(d, 0.2);
```

Zadanie 2

Rozszerz rozwiązanie z Zadania 1 w taki sposób, aby wygenerowane okręgi były animowane w czasie. Animacja powinna powodować ich rozszerzanie się od środka na zewnątrz albo kurczenie ku środkowi. Efekt powinien być uzyskany wyłącznie przez modyfikację obliczeń w shaderze fragmentów.

```
float2 uv = i.uv - 0.5;
float d = length(uv);
float r = frac(_Time.y * 0.2);
float ring = step(abs(d - r), 0.02);
```

Zadanie 3

Wykorzystując operacje na współrzędnych oraz parametr czasu, zaimplementuj prosty efekt stylizowanej powierzchni wody. Efekt powinien sprawiać wrażenie ruchu fal lub zaburzeń na powierzchni. Rozwiązanie może opierać się na okresowej zmianie koloru, jasności albo położenia wzoru proceduralnego, bez konieczności odwzorowania fizycznie poprawnego ruchu cieczy.

```
float wave = sin(i.uv.x * 20.0 + _Time.y * 2.0);
float value = 0.5 + 0.5 * wave;
```