1 Wstęp teoretyczny

Vertex shader jest etapem potoku renderowania odpowiedzialnym za przetwarzanie wierzchołków obiektu przed rasteryzacją. Jego podstawowym zadaniem jest wyznaczenie końcowego położenia każdego wierzchołka w przestrzeni obrazu, jednak w praktyce może on również modyfikować geometrię modelu. Oznacza to, że shader wierzchołków może zostać wykorzystany nie tylko do transformacji obiektu, ale także do realizacji prostych animacji i deformacji powierzchni.

Animacja w vertex shaderze polega na zmianie położenia wierzchołków w funkcji czasu. W najprostszym przypadku można przesuwać wierzchołki wzdłuż jednej osi, na przykład osi `Y`, wykorzystując funkcje okresowe, takie jak sinus. Dzięki temu możliwe jest uzyskanie efektu falowania, pulsowania lub drgania powierzchni. Istotne jest przy tym, że deformacja dotyczy bezpośrednio geometrii obiektu, a nie jedynie jego koloru lub tekstury.

Końcowy wygląd animacji zależy najczęściej od trzech czynników: czasu, położenia wierzchołka oraz parametrów sterujących efektem. Czas pozwala wprowadzić zmianę dynamiczną, położenie wierzchołka umożliwia zróżnicowanie deformacji w różnych częściach obiektu, natomiast parametry takie jak amplituda i częstotliwość pozwalają kontrolować siłę oraz tempo animacji. W efekcie można uzyskać zarówno prosty ruch całego obiektu, jak i bardziej złożoną deformację powierzchni.

W praktyce animacja oparta na vertex shaderze jest szczególnie użyteczna w przypadku efektów takich jak fale, trawa, tkaniny, drgania obiektów czy stylizowane deformacje. Należy jednak pamiętać, że jakość efektu zależy od liczby wierzchołków w siatce. Zbyt mała gęstość geometrii ogranicza możliwości deformacji, ponieważ shader może przemieszczać jedynie istniejące wierzchołki, a nie tworzy nowych punktów na powierzchni.

Przykładowa modyfikacja położenia wierzchołka może wyglądać następująco:

```
float wave = sin(v.vertex.x * _Frequency + _Time.y) * _Amplitude;
v.vertex.y += wave;
```

W powyższym fragmencie wartość przesunięcia zależy od współrzędnej `x` wierzchołka, czasu oraz dwóch parametrów sterujących. Taki zapis powoduje, że różne fragmenty siatki poruszają się z różnym przesunięciem, co daje efekt fali.

Po obliczeniu nowego położenia wierzchołka należy przekształcić je do przestrzeni obrazu:

```
o.pos = UnityObjectToClipPos(v.vertex);
```

W ten sposób zmodyfikowana geometria zostaje przekazana do dalszych etapów renderowania. Dzięki temu vertex shader może pełnić nie tylko rolę transformacyjną, ale również funkcję prostego narzędzia animacji proceduralnej.

2 Zadania

Zadanie 1

Zaimplementuj prostą animację geometrii w vertex shaderze, w której wierzchołki obiektu będą okresowo przesuwane w osi pionowej. W tym celu wykorzystaj funkcję sinus oraz parametr czasu. Efekt powinien przypominać falowanie powierzchni.

Zadanie 2

Wygeneruj obszar trawy, w którym każde źdźbło jest reprezentowane przez pojedynczy trójkąt. Rozmieść źdźbła losowo na wybranej powierzchni, z uwzględnieniem niewielkich różnic położenia, wysokości lub orientacji. Następnie zaimplementuj w vertex shaderze animację imitującą ruch trawy na wietrze. Spraw, aby górne części źdźbeł poruszały się wyraźniej niż ich podstawy, a ruch poszczególnych elementów nie był identyczny.