**Shader in Unity**

# I/ Những thuật ngữ sử dụng nhiều

## 1. Shader

Theo như trong tài liệu của Unity3D thì Shader là các đoạn script nhỏ có chứa những cái tính toán về toán học, thuật toán để tính toán màu sắc cho từng pixel rendered, dựa trên ánh sáng và những tùy chỉnh Material.

Unity có hỗ trợ 3 loại Shader. Nhưng chỉ có 2 loại thường được sử dụng đó là Suface Shader và Fragment and Vertex Shaders, loại còn lại là Fixed Function Shader, nhưng loại này giờ hầu như không còn được sử dụng nữa bởi vì nó đã quá lỗi thời so với các loại Shader đang sử dụng hiện giờ.

## 2. Shader Lab

Tất cả các file Shader trong Unity đều được khai báo bằng một ngôn ngữ gọi là ShaderLab. Trong ShaderLab có các cú pháp cơ bản để định nghĩa nên những mô tả khác nhau cho Shader. Ví dụ như là các thuộc tính của Shader sẽ được hiển thị trong material trên Inspector mà bạn hay nhìn thấy.

## 3. Cg program

CG là viết tắt của C for Graphics. Nó là ngôn ngữ shading bậc cao đã được phát triển bởi Nvidia phối hợp với Microsoft cho lập trình Vertex và pixel Shader. CG thì base trên ngôn ngữ C, mặc dù chúng có cùng cú pháp nhưng một vài tính năng của C đã được sửa đổi và các kiểu dữ liệu mới được thêm vào sao cho CG phù hợp hơn với các đơn vị xử lý đồ họa lập trình. Ngôn ngữ này chỉ phù hợp phù hợp cho lập trình GPU và không như một ngôn ngữ lập trình chung. Biên dịch Cg có kết quả đầu ra DirectX hoặc OpenGL.

# 4. Material là gì?

Như nào nhỉ? Mọi người có thể hiểu như thế này nhé: Một Material là một asset có thể đưa vào một mesh để điều khiển cái cách nhìn nhận trực quan của cảnh, làm cho ảnh trong máy có thể mô tả được giống nhất với ý định của người tạo cảnh. Hoặc nghĩ thoáng hơn, liên hệ thực tế hơn chat thì nó như kiểu là sơn được sơn lên cho một đối tượng, hoặc là như cấu trúc 1 bức tường ban đầu chỉ toàn là cốt thép, xương xẩu thì nó như là xi măng để chat lên đó rồi mình nhìn thấy được nó ra một hình hài, màu sắc một bức tường. À đó lại còn khái niệm về Mesh ở trên nữa. Hầy, nếu như với một người mới tiếp cận 3D thì đúng là cần nắm được kha kha những hiểu biết về những cái này. Nhiều thật, nhưng dần rồi sẽ quen. Cái khách thông biết thế nào nhưng trong Unity thì Material có liên hệ khá mật thiết với Shader đấy. Vừa làm các ví dụ bạn vừa cảm nghiệm và đúc rút nhé.

# 5. Mesh

Như nghĩa của nó. Nó là tập hợp các đỉnh, cạnh liên kết với nhau tạo nên một đối tượng thống nhất.

Nhiều khái niệm quá cũng chán. Chúng ta tới luôn Shader thôi.

# II/ Cấu trúc Shader

Về tổng quan thì nó trông như sau:

```
Shader "MyShader"
{
    Properties
    {
        // Phần khai báo thuộc tính Shader(texture, color, parameter...)
    }

    SubShader
    {
        // Đây là phần chính của Shader. Tại đây bạn có thể viết Shader theo
        // kiểu surface shader hoặc vertex and fragment shader hoặc là fixed function shader
        // Tôi khuyến cáo không nên dùng fixed function nữa bởi vì nó thành đồ cổ rồi. Giờ chắc chẳng support nhiều nữa
    }
}
```

Trong một file Shader bạn có thể định nghĩa nhiều SubShader khác nhau. Chúng có chứa những ủy nhiệm công việc trực tiếp cho GPU, Unity sẽ duyệt, thực thi lần lượt từng cái một cho tới khi nó tìm được một cái tương thích với card đồ họa của bạn. Nó khá là hay trong việc lập trình đa nền tảng :x

## Properties

Thuộc tính của Shader, bạn có thể tưởng tượng nó giống như trường public của C# trong Unity vậy. Nó sẽ hiển thị lên Inspector và bạn có thể tùy chỉnh các thuộc tính của nó ngay trên editor và nhìn thấy luôn những thay đổi của nó.

Dưới đây là tổng hợp khai báo các thuộc tính cơ bản của Shader mà mình sưu tầm được từ Unity tutorial.

```
Properties
{
    _Texture ("Texture", 2D) = "white" {}
    _NormalMap ("Normal map", 2D) = "bump" {}

    _Int ("Integer", Int) = 2
```

```
    _Float ("Float", Float) = 1.5
    _Range ("Range", Range(0.0, 1.0)) = 0.5

    _Color ("color", Color) = (1, 0, 0, 1)    // (R, G, B, A)
    _Vector ("Vector4", Vector) = (0, 0, 0, 0) // (x, y, z, w)
}
```

Với chỉ như trên thì thật sự nó chưa sử dụng được. Các giá trị từ các thuộc tính trên thực tế sẽ được xử lý trong SubShader cơ.

```
SubShader
{
    exampler2D _Texture;
    exmpler2D _NormalMap;

    int _Int;
    float _Float;
    float _Range;
    half4 _Color;
    float4 _Vector;


}
```