**Perlin Noise**

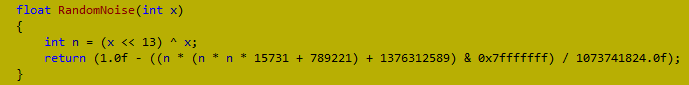
**一、简介**

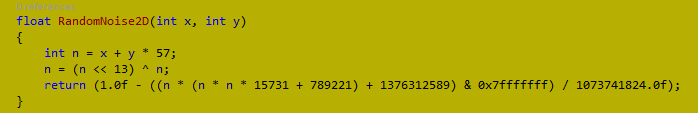
Perlin噪音是一个非常重要的工具，它能够生成类似于所有自然现象的纹理——云、大理石、岩石、木头以及介于期间的所有现象。Perlin 噪音是Ken Perlin于1985年提出。Perlin噪音本质上相当于一个随机数生成器，不过与普通的随机数生成器不同的是Perlin噪音使用一个整数作为种子返回一个随机数，如果种子相同，那么两次会产生相同的随机数，相同情况下，普通的随机数生成器可能会生成不同随机数，所以也可以认为Perlin噪音是“伪随机数”的。

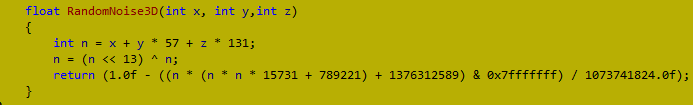
**二、噪音函数**

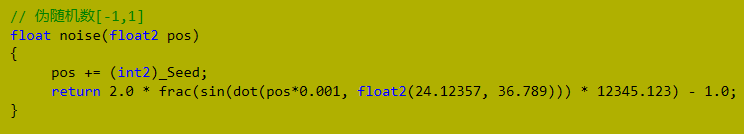
为了创建一个Perlin噪音函数，需要两个函数，一个是随机噪音函数，另一个是插值函数。

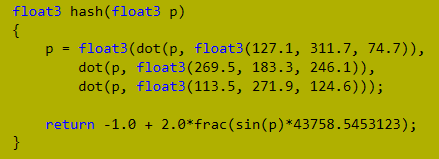
下面是不同形式的**随机噪音函数，**通过一个坐标值（一维、二维或是三维），输出一个随机数，只要坐标值不变，随机数就不变，这很重要。





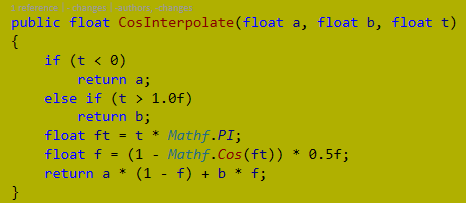






下面是几种**插值函数，**通过t（取值范围[0,1]，t = 0时，返回a，t=1时，返回b）在a和b之间进行插值：

1、余弦插值

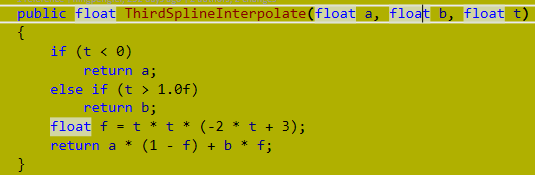






所以余弦插值函数是**一阶连续**的。

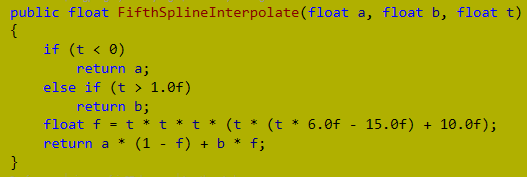
1. 三次样条曲线插值





所以三次样条曲线插值函数也是**一阶连续**的。

3、五次样条曲线插值



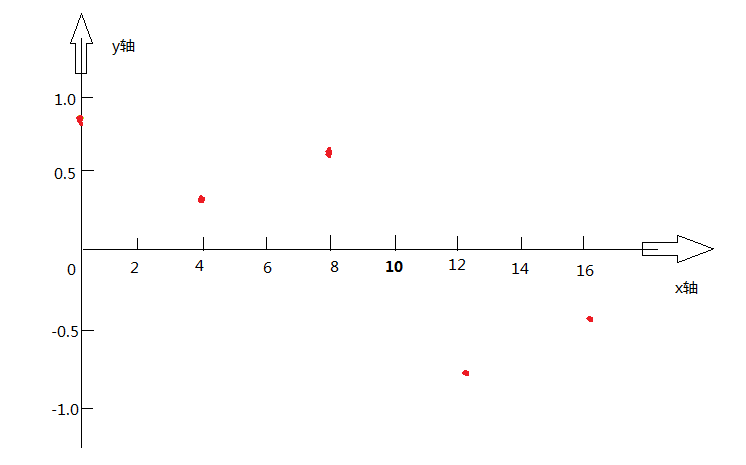


所以五次样条曲线插值是**二阶连续**的。

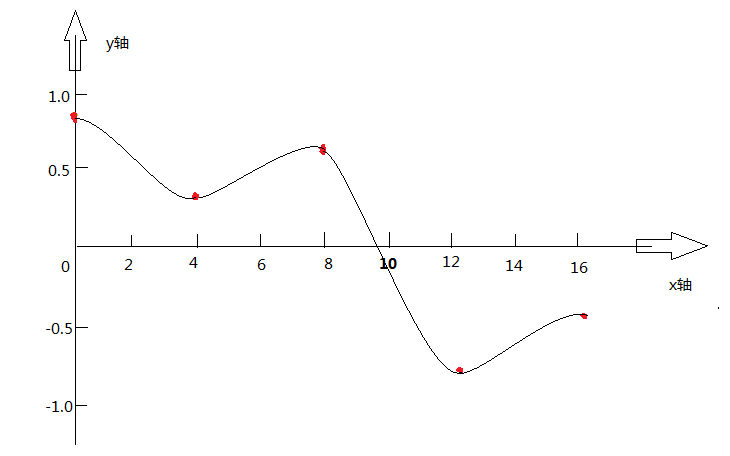
**三、1D Perlin Noise**

比如生成17x1的Perilin noise。

1. 用随机函数生成一组随机数，这里间隔是4（间隔越小，噪音的频率就越高）



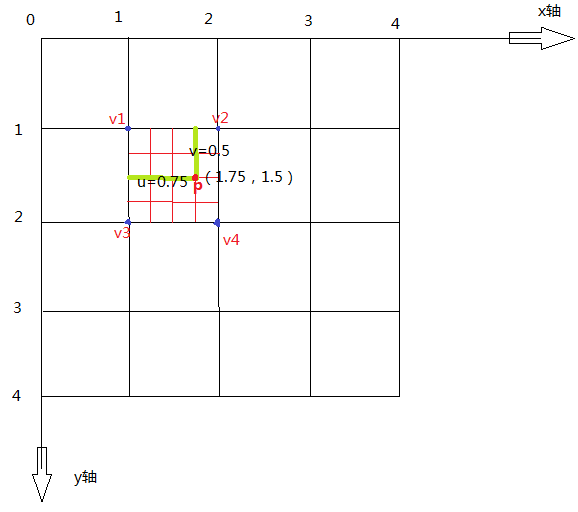
1. 在上述生成的随机数（红点）之间的 数值通过插值函数（例如余弦插值）得到。



3、分形叠加，上述通过1和2步骤得到的Noise。

1. **2D Perlin Noise**

与一维的类似，例如要生成256x256的Perlin noise，如下：



1、将256x256的纹理，用5x5的网格将其划分（网格越密，生成的Perlin Noise频率就越高）。

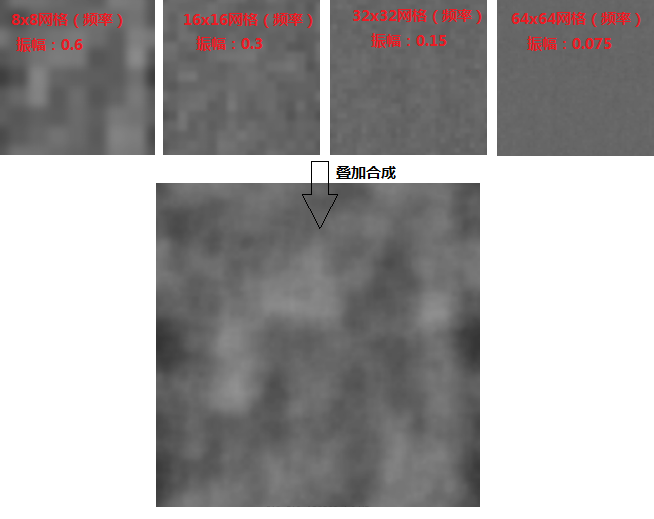
2、通过网格顶点的二维坐标，生成在该顶点处的随机噪音值，例如v1,v2,v3和v4。

3、网格内的像素点的噪音值通过插值得到。

P点的噪音值可以通过x方向两次插值和y 方向一次插值得到



4、分形叠加。经过前面3步生成的Noise还不是很好，如果叠加若干不同频率（frequency）和不同振幅（amplitude）的Noise，效果将会大大地得到改善。



另外要值得注意的是，要想生成四方连续的Perlin Noise，那么网格边缘顶点的Noise值（随机噪音函数生成的值），就要进行循环寻址，例如Noise(4,1) = Noise(0,1)，即4行1列顶点的Noise 等于 0行1列顶点的Noise。

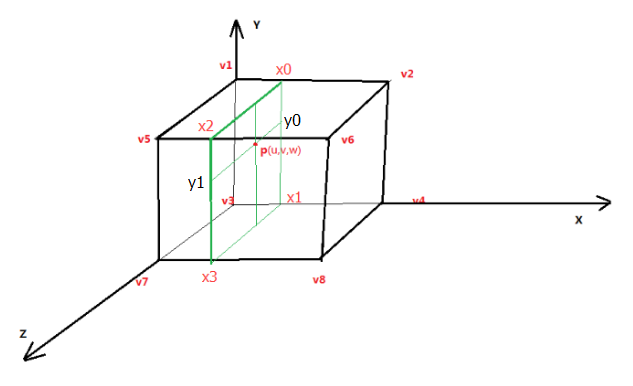
**五、3D PerlinNoise**

与二维的类似，例如要生成256x256x256的Perlin noise，如下：

1. 将256x256x256的纹理，用5x5x5的小单元空间将其划分。小单元空间越密集，生成的Perlin Noise频率就越高。
2. 通过顶点的三维坐标，生成在该顶点处的随机噪音值，例如v1、v2、v3、v4、v5、v6、v7和v8。

3、单元空间内的像素点的噪音值通过插值得到。

P点的噪音值可以通过x方向四次插值、y 方向两次插值和z方向的一次插值得到，如下图所示：







其中Interpolate是插值函数（u,v,w）是p点在单元空间内的坐标。

4、分形叠加。叠加若干不同频率（frequency）和不同振幅（amplitude）的Noise，跟2D PerlinNoise一样。

**六、总结**

Perlin Noise非常重要，在图形学里有很多应用，例如可以用来模拟微风下的水面，云、火焰、大理石、木纹理等等自然现象。

Perlin noise可以用CPU或是GPU生成，本文用的是GPU来生成，需要Shader mode 3.0。

源码链接：

http://pan.baidu.com/s/1bpBLDdt