

## Вопросы после лекции 8

1. Три вершины треугольника
2. Кол-во точек, на которые нужно разбить ребра треугольника, кол-во треугольников внутри (на сколько треугольников бить) и коэффициенты  $b(ijk)$
3. Множество троек  $(u,v,w)$ , и использование коэффициентов  $b(ijk)$ , полученные ранее
4. Вершины  $b(u,v)$  по формуле
5. В геометрическом шейдере делать тесселяцию очень долго, на линиях где стыкуются треугольники с разной детализацией, во время рендеринга там будут появляться черные дырки и связано это с тем что растеризатор для разных треугольников будет работать исходя из разных данных и из-за этого не всегда будут получаться одинаковые значения для иксов когда растеризуется линия (на границах нужно оставлять общее ребро)
6. Во первых чем больше треугольник, тем на большее кол-во треугольников его надо разбить, дальность камеры, угол наклона треугольника к камере (Если угол почти 90 градусов, их разбивать не надо, а треугольники на гранце разбивают мелко)
7. Pixel shader может заполнять буфер экрана и несколько матриц (в z буфер может записать что хочет) этой функцией как раз воспользуемся для отрисовки горы. Через один цвет пишется в буфер экран и тем самым отрисовывается гора, а в другой цвет (вспомогательная матрица) будет писаться значение  $z$  (для каждого пикселя горы)
8. Bump mapping — простой способ создания эффекта рельефной поверхности с детализацией большей, чем позволяет полигональная поверхность. Эффект главным образом достигается за счёт освещения поверхности источником света и чёрно-белой (одноканальной) карты высот, путём виртуального смещения пикселя (как при методе Displace mapping) как если бы там была вершина (только без физического и визуального сдвига), за счёт чего таким же образом изменяется ориентация нормалей использующихся для расчёта освещённости пикселя (затенение по Фонгу), в результате получаются по-разному освещённые и затенённые участки. Как правило, bump mapping позволяет создать не очень сложные бугристые поверхности, плоские выступы или впадины, на этом его использование заканчивается

9. Вектор направляется по вектору  $V$  текстуры и  $T$  это тангенс вектор, перпендикулярен  $B$  (Ось  $U$  направлена как разность между координатами  $U$ ,  $V$  аналогично).

10. Для того, чтобы можно было пользоваться скалярным произведением при вычислении освещенности: нормаль берем из текстуры, и она в с. к., связанной с пикселем, а направление на источник – в глобальной. Поэтому вычисляем по 3-м векторам матрицу для перевода источника в локальную систему координат и вычисляем освещение