**Shader programs.**

Начиная с некоторого объёма данных для обработки и вывода на экран становится очевидно, что ресурсов центрального процессора недостаточно для быстрой обработки данных. Решение проблемы скорости работы программы заключается в переносе алгоритмов ввода, обработки и вывода моделей на ресурсы видеокарты. Для алгоритмизации процессов видеокарты используется система шейдеров. Шейдер – программа, принимаемая видеокартой и выполняющая сложную вычислительную работу.

Реализация шейдеров разделена на несколько этапов для полноценного управления процессом и модификации вывода. Шаги следующие:

* Вершинный шейдер – первый шаг, основа которого – изменение параметров переданных вершин и их освещения. Принимает в себя – данные об одной вершине геометрической модели, которая обрабатывается в данный момент. Обычно передается позиция, нормаль, цвет и текстурная координата объекта. Пример их использования – создание реалистичных волн в динамике, в том числе, и воды
* Тесселяционные шейдера (контрольный и вычислительный) – основа преобразования формы объекта или создания модели из нескольких опорных точек. Контрольный теселяционный шейдер определяет сетку опорных точек, а также её частоту. Вычислительный создает на основе созданной сетки готовый объект. К примеру, создание реалистичных тканей на основе бикубических поверхностей легко реализуется на тесселяционном шейдере.
* Геометрический шейдер становится между вершинным и фрагментным шейдером и позволяет преобразовывать переданные ему вершины, а также создавать новые вершины и изменять формат вывода вершин. При использовании необходимо помнить, что геометрический шейдер, в отличие от других шейдеров, принимает не одну вершину, а все созданные на предыдущих этапах, и выдает так же все вершины, так ччто при прямом взаимодействии с передаваемым геометрическим шейдером объектами (кроме вершин) требуется работать с массивом объектов, что не всегда удобно. Классический пример использования геометрического шейдера – генерация объектов малой детализации (таким объектом может быть линия, сфера или другой простой геометрический объемный объект), а также изменение формы объекта при его ударах обо что-либо.
* Если предыдущие шаги касались изменения параметров примитива, в том числе его создания, то фрагментный шейдер отвечает за цвет каждого отдельно взятого пикселя

.

