Geometry y *Tessellation Shaders*

Miguel Angel Astor Romero

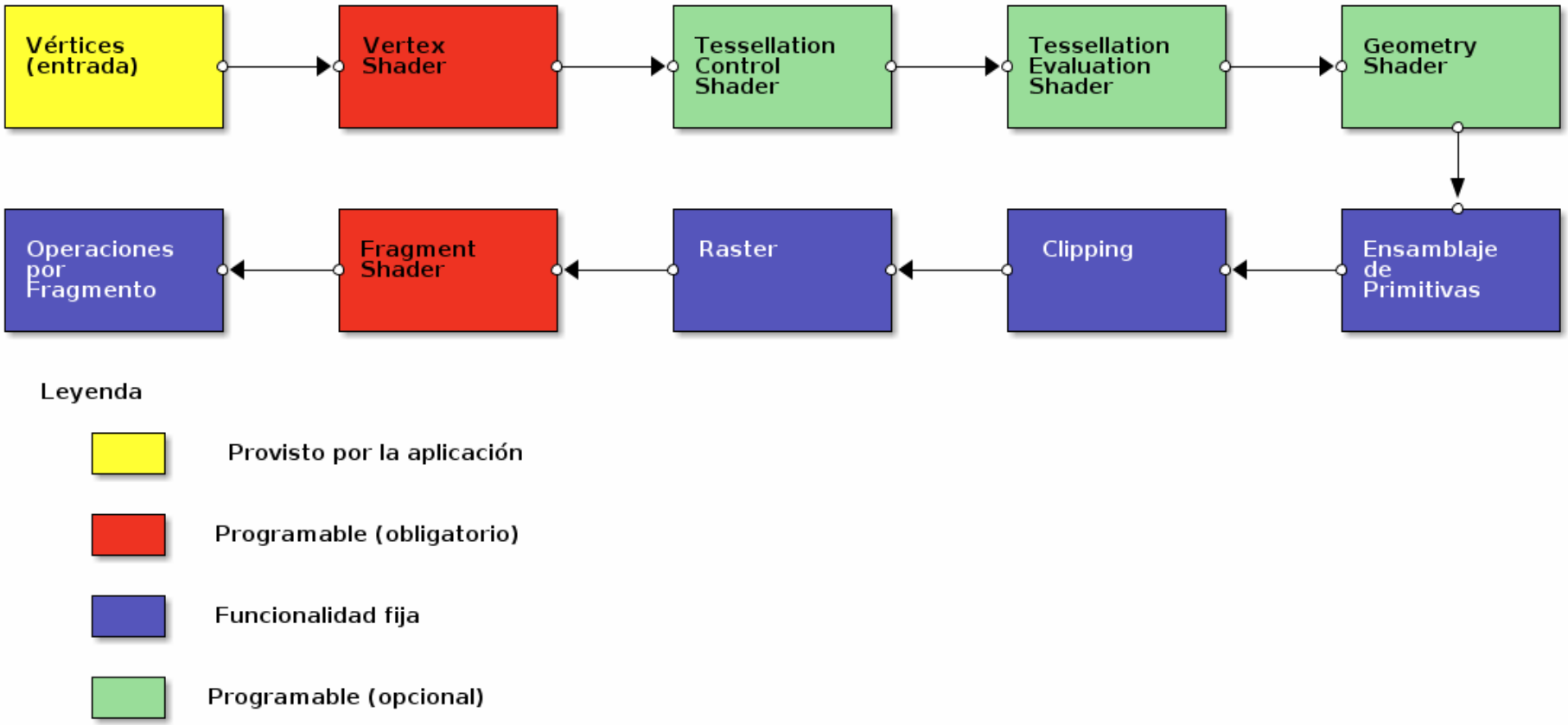
13 de julio de 2016

Agenda

Introducción

- En OpenGL >= 3.2 se introduce el *Geometry shader* como funcionalidad núcleo.
  - Anteriormente era una extensión (>= OpenGL 2.0)
- OpenGL >= 4.0 continúa la tendencia de desfasar el modo inmediato e introduce nuevas etapas en el proceso de renderizado:
  - *Tessellator* automático controlado por dos *shaders*.
  - *Tessellation control shader*.
  - *Tessellation evaluation shader*.

El pipeline gráfico

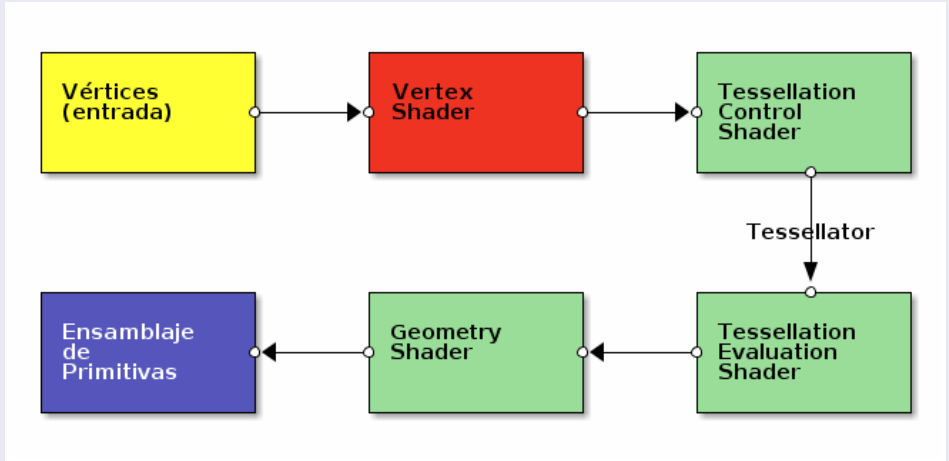


La etapa de procesamiento de vértices

Consiste en la creación de primitivas para desplegar:

- Aplicar operaciones por vértice.
- Determinar el nivel de *tessellation*.
- Transformar vértices creados por el *tessellator*.
- Aplicar transformaciones por primitiva.
- Generar primitivas.

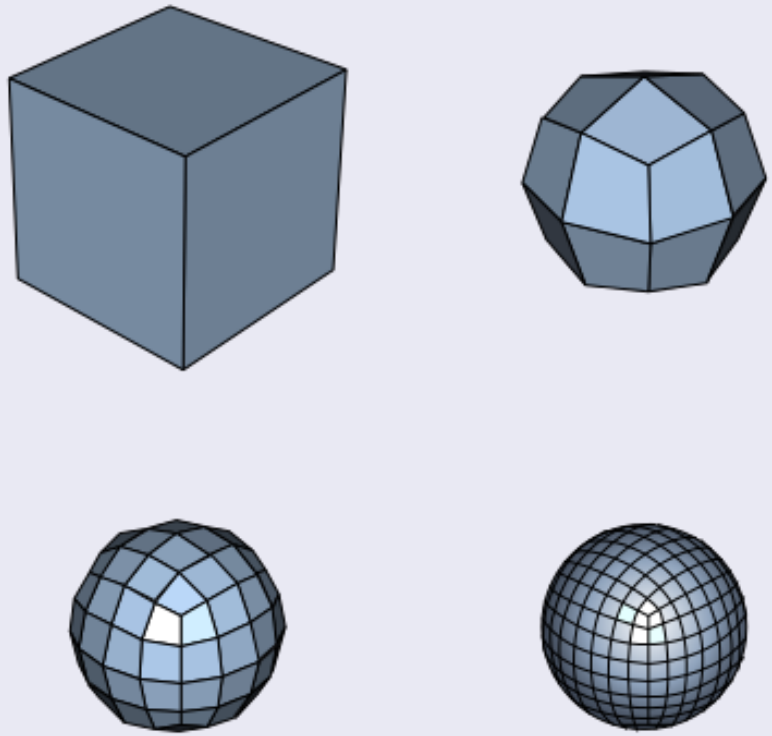
Pipeline



Tessellation en el procesamiento de vértices

- Objetivos del proceso de *tessellation*:
  - Subdivisión de una primitiva geométrica.
  - Funciona sobre parches en lugar de sobre triángulos.
  - Genera triángulos o *triangle strips*.
  - Puede eliminar parches antes de pasar por el *clipping*.

Catmull-Clark Subdivision Surfaces

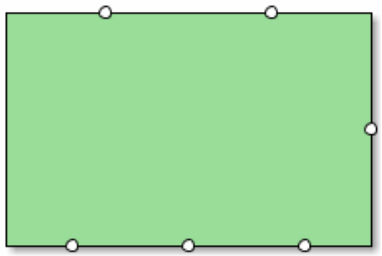


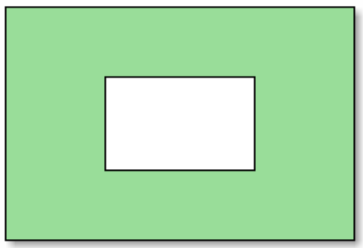
El Tessellation Control Shader (TCS)

Define cómo se debe realizar el proceso de *tessellation*.

- Es controlado por dos parámetros:
  - *Outer tessellation level*
  - *Inner tessellation level*
- También puede aplicar transformaciones por parche.
- Se invoca por lo menos una vez por parche.
- Es opcional.

Niveles en el proceso de tessellation





El Tessellation Evaluation Shader (TES)

Los vertices generados por el *tessellator* son ideales (parches abstractos). El TES se encarga de transformarlos para ubicarlos en sus posiciones respectivas.

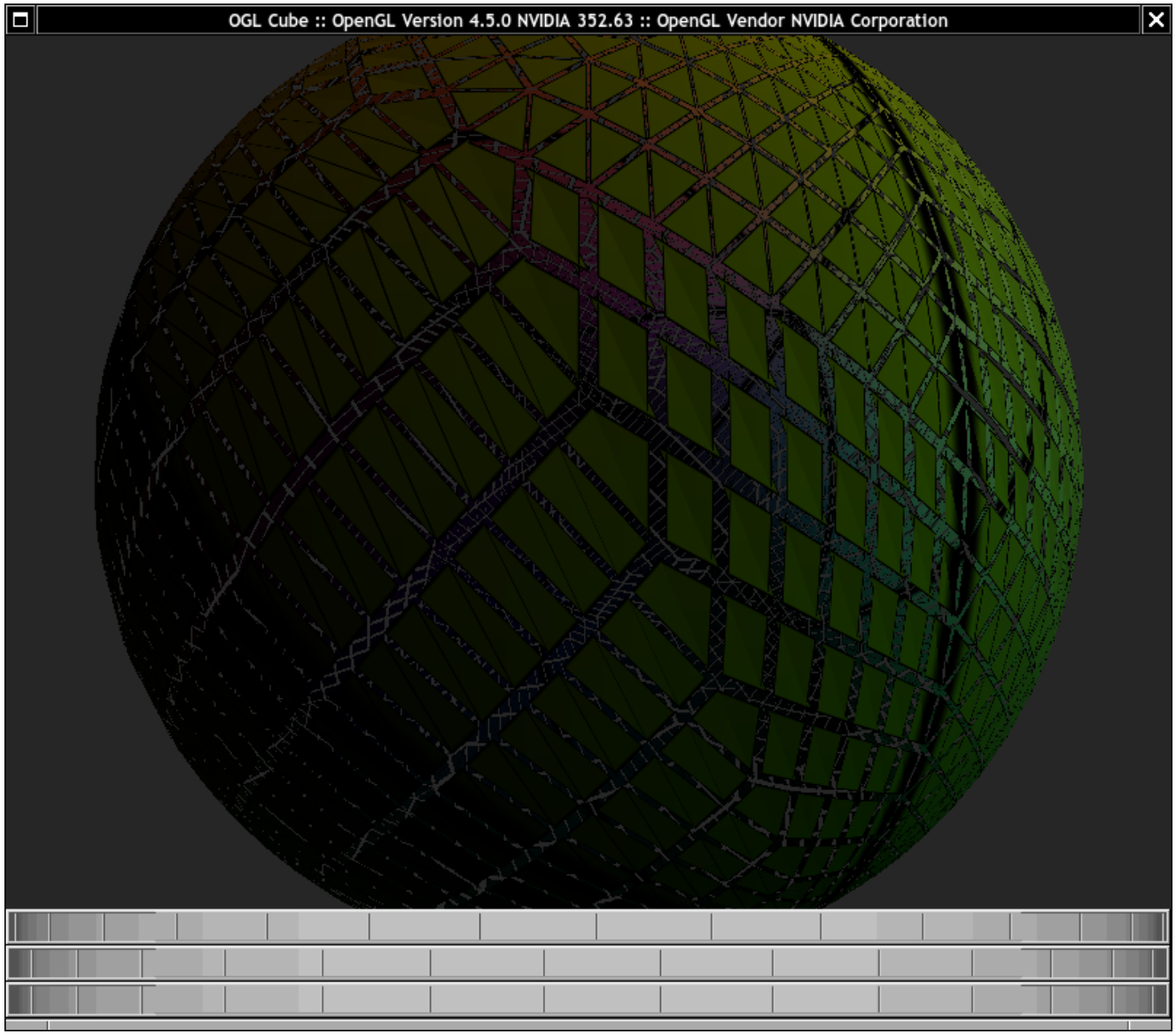
- El TES define el tipo de primitiva a generar.
- Los vértices generados vienen en la variable `gl_TessCoord`:
  - Coordenadas baricéntricas para triángulos.
  - Coordenadas UV para *quads*.
- Se invoca una vez por cada vértice nuevo.

Casos de uso del Geometry shader

El *Geometry shader* permite realizar operaciones sobre todos los vértices de una primitiva geométrica en una sola invocación.
Casos de uso:

- Transformación de primitivas.
- Amplificación de geometría:
  - *Tessellation*.
- Despliegue por capas.
- Retroalimentación de transformaciones.

Demostración



Referencias

1. E. Ramírez, *Despliegue Básico en OpenGL Moderno*, Notas de Docencia ND 2014-01, Esc. de Computación, UCV, 2014.
2. `https://www.opengl.org/wiki`.

¿Preguntas?