GPU Programming with WebGL

Definition: Polygon model/mesh

Comprises a set of connected polygons to represent an object

1 Why WebGL

- Cross-platform, browser-based
- Hardware-based rendering
- Support programmable rendering pipeline
- Zero setup effort before you start programming

2 GPU Programming

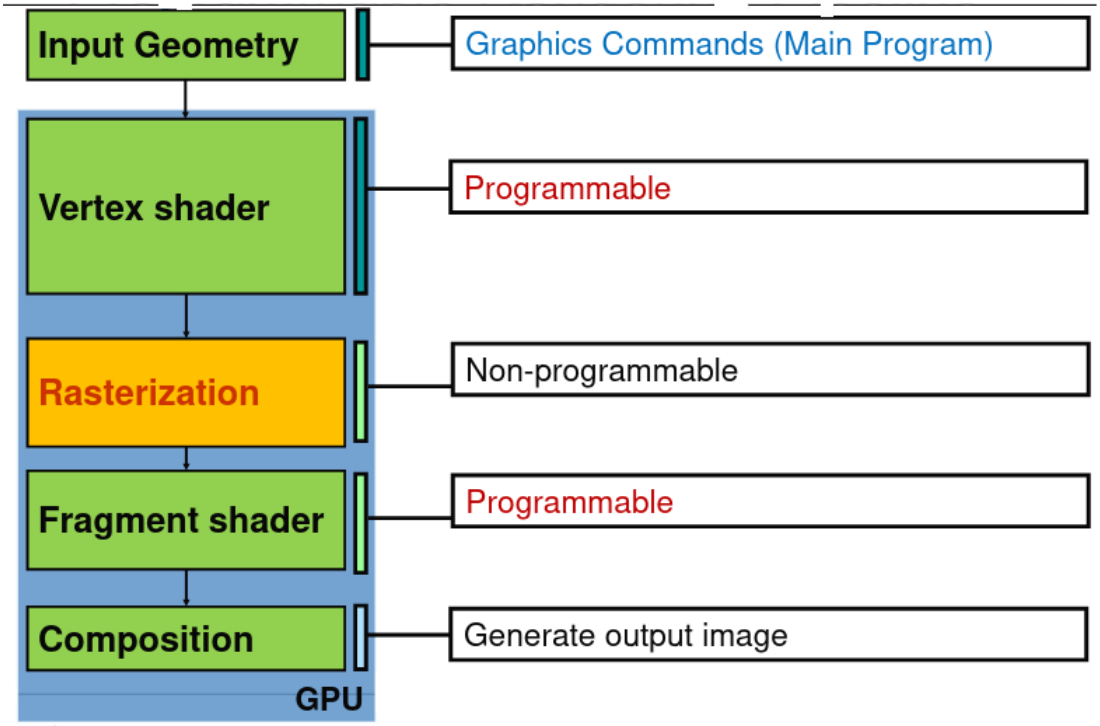
GPU

- Typically comprises of hundreds to thousands of processors
- Process graphics primitives in parallel

Programmable rendering pipeline

- Vertex shader and fragment shader are programmable
- GPU programming is also called shader programming

3 Programmable rendering pipeline



Pixels - Colour info
Fragment - Single point with more information than a pixel, e.g. lighting effects

4 Shaders

Vertex shader:

- Manipulates per-vertex data such as vertex coordinates, normals, colors, and texture coordinates

Fragment shader:

- Deals with surface points for processing
- Main goal: calculate colour for each pixel that will display on the screen

Rasterization process:

- A black box (non programmable) generates fragments from outputs of vertex shader

5 Data Structures

Definition: Vertex Buffer Objects (VBOs)

Contain the data that WebGL requires to describe the geometry that is going to be rendered

Definition: Index Buffer Objects (IBOs)

Contain integers that use as references pointing to data in VBOs, in order to enable the reuse of the same vertex

Attributes, uniforms and varyings are the three types of variables that you will find when programming with shaders

Definition: Attributes

Input variables used in the vertex shader (dynamic)

Definition: Uniforms

Input variables available for the vertex shader and the fragment shader (static)

Definition: Varyings

Used for passing data from the vertex shader to the fragment shader