

Discussion 8: From Geometry to Pixels

Discussion Topic:

Provide an example of rasterization in action. Choose a simple geometric shape, such as a triangle, and explain how the vertices of the triangle are transformed into pixels using the rasterization process. Discuss how attributes like color and depth are assigned to the resulting pixels.

My Post:

Hello Class,

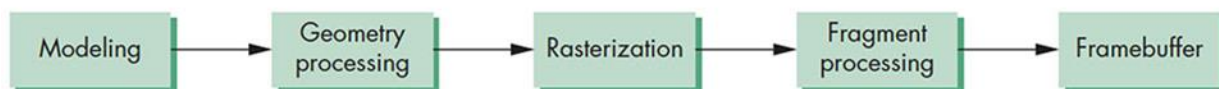
In computer graphics, understanding the rasterization process is essential to efficiently create applications that render 2D and 3D images. Rasterization transforms geometric data into pixels that are displayed usually on a screen. For example, it takes a simple geometric shape (e.g. a triangle) and transforms it into pixels with the position, color, and depth attributes.

The process of rendering a geometric shape involves four essential tasks, modeling, geometry processing, rasterization, and fragment processing. These tasks are arranged in a pipeline with rasterization positioned at the center of this process, making it a critical stage that converts geometric data into pixels or fragment shaders, see Figure 1.

Note that the process of rasterization includes rasterization itself or scan conversion and fragment processing, the processes where the rasterizer assigns the color and depth attribute to the pixels.

Figure 1

The Five Tasks



Note: From “Chapter 12: From Geometry to Pixels. Interactive Computer Graphics. 8th edition” by Angel & Shreiner (2020, p.299)

The process of rationalizing usually involves three steps. Two pre-rasterization steps include defining the triangle(s) (modeling) and performing transformations and clipping (Geometry Processing), and the third step is rasterization itself.

Step-1 Modeling. In computer graphics, polygons are primitive shapes that are used to model other objects. WebGL uses triangles, being the simplest polygon it can be used to model other polygons. A triangle can be defined in 2D space by three vertexes $v_1(x_1, y_1)$, $v_2(x_2, y_2)$, and $v_3(x_3, y_3)$ representing the corner of the triangle.

Step-2 Geometry Processing. This step often includes projection (to move from 3D to 2D in perspective or orthographic views) and clipping (to remove parts of the triangle that are outside the viewing

volume). Note that the x and y values are used for realization and the z values are stored for hidden surface removal.

Step 3: Rasterization. Rasterization, also called scan conversion, is the process that determines which pixels (also called fragments) on the screen correspond to the triangle's interior. After identifying the triangle's pixels, fragment processing is performed, where the rasterizer applies z-buffering (depth testing) to handle overlapping objects in the 3D scene. Finally, each pixel is assigned a color and sent to the framebuffer, which stores the final image before being displayed on a screen.

To summarize, understanding the rasterization process is essential to efficiently create applications that render 2D and 3D images. Rasterization transforms pre-processed geometric data into pixels also called fragments by computing the positions, color, and depth attributes for each pixel before sending it to the framebuffer to be displayed.

-Alex

References:

Angel, E., & Shreiner, D. (2020). Chapter 12: From geometry to pixels. Interactive computer graphics. 8th edition. Pearson Education, Inc. ISBN: 9780135258262