GP1 – Rasterization Pipeline (High-Level) Overview (W6)

INPUT

Vertices (in World Space)

Each Vertex has at least a POSITION attribute, but can be decorated with extra attributes such as COLOR, UV, ...

Every Vertex

PROJECTION STAGE

[All vertices]

World(space) → View(space)

1. Multiply all vertex positions with **ViewMatrix**.
   ViewMatrix → Inverse of Camera ONB

View(space) → NDC(space)

(Also referred to as Projection- or Clipping Space)

1. Apply **Perspective Divide (see slides)**
2. Apply **Camera Settings (see slides)**

OUTPUT

Vertices (in 'NDC' space)

Only the positions are changed after this stage, other attributes are passed through.

Position X & Y → NDC (Range [-1,1])
Position Z → ViewSpace (Distance from camera)

INPUT

Vertices (in 'NDC' space)

Each Vertex has at least a POSITION attribute, but can be decorated with extra attributes such as COLOR, UV, ...

Every Triangle

Every Pixel

RASTERIZATION STAGE

[Each Triangle → Iterate Pixels]

NDC (space) → Raster/Screen (space)

See slides for formula. After this transformation your Position X & Y should be in Screenspace (2D raster coordinates), Position Z still contains ViewSpace 'depth'

Optimization → Calculate Triangle BoundingBox (BB) in Rasterspace → Only iterate the pixels covered by this BB.

PIXEL inside TRIANGLE?

Cross product between each Triangle edge and Vertex-2-Pixel vectors (2D vectors) → IF signs of all cross-product results are the same THEN pixel is inside the Triangle.

(Viewspace) Depth Interpolation + Depth Test

Correct 'Depth Interpolation' in LAB 2

Vertex Attribute Interpolation

Correct 'Attribute Interpolation' in LAB2

Barycentric Coordinates

Cross Product Results can be used to calculate interpolation weights for Depth & Attribute Interpolation.

+ See LAB2 for 'correct' Depth & Vertex Interpolation

OUTPUT

Rasterized triangle on back-buffer