Data storage:

Model 1

* Name/UUID

8 bounding points

Points

Triangles

Model 2

Point: x, y, z floats

Triangle: indexes to points, normal vector, color values for each coordinate, texture coordinates, texture pointer

Pixel Map

* Grid of colored pixels to draw to screen

Z-buffer

* Unsigned integer z values for each pixel

Graphics process:

Iterate through models and check bounds. If any of the bounds lie on the camera side of the xy-plane (could be optimized using AABB?), apply 3D transforms to all of the points and save these to a persistent vector. Also save the z values as fixed-point unsigned integers scaled between near and far bounds and booleans indicating if the point crosses the near plane or the far plane. Start iterating through the triangles (skip triangles with all near or all far crossing vertices) and use back-face culling to remove triangles that face wrong direction. If a vertex is invisible, calculate two new points where the triangle gets cut off (using the 3D coordinates and applying transform) and add a new triangle if necessary (near-plane culling). Do the same for far-plane culling, some triangles may require both too. Next calculate the position of each pixel in the triangle using a rasterization algorithm optimized for out of bounds triangles. If the z value of the pixel is less than/equal to the corresponding z value in the z-buffer then discard the pixel. Otherwise, calculate the color and texture of the pixel. If the alpha of the pixel or the pixel already on screen is less than 255, blend the colors (the alpha should be blended so that it is at least the alpha of the opaquest pixel). If the alpha is zero, discard the pixel instead. If the alpha is 255, overwrite the color and the corresponding z-buffer value. Continue for all models, points, and triangles.

y

z x

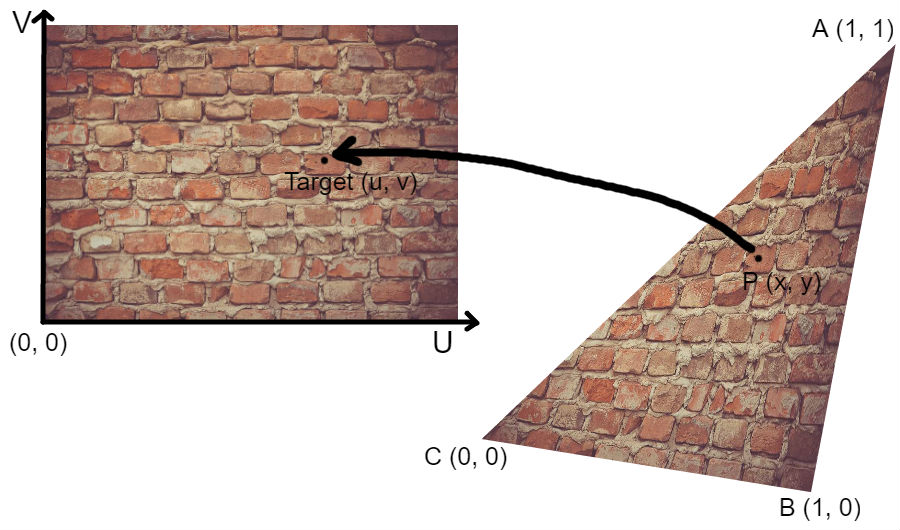
Camera

Far plane

Near plane

UV Mapping:

For flat bottom triangle with points a, b, c (clockwise). Target point is p, s is horizontal (left to right), t is vertical (top to bottom).

A similar calculation will need to be applied for color points.