## WebGL Water Tutorial - Rendering Pipeline State Diagram Initialize WebGL - Create WebGL context - Load VAO/Depth extensions Setup viewport Success Load Assets - Load shaders (vertex/fragment) - Compile shader programs - Load textures (dudv, normal, stone) Load meshes (water\_plane, terrain) Create GPU buffers Assets Loaded Setup Framebuffers - Create reflection framebuffer - Create refraction framebuffer Setup depth textures Configure texture parameters FB Ready Render Loop Render Water Water Shader Pipeline: - Use water shader program - Bind water mesh 1. Transform vertices to clip space 2. Sample dudy texture for distortion - Set perspective/view matrices - Bind reflection texture (TEXTURE1) 3. Sample reflection/refraction textures - Bind refraction texture (TEXTURE0) 4. Apply Fresnel effect - Bind dudv texture (TEXTURE2) 5. Calculate specular highlights - Bind normal map (TEXTURE3) 6. Mix reflection/refraction based on angle - Set water uniforms (reflectivity, fresnel, etc.) 7. Blend with water base color - Calculate dudy offset for animation Draw water plane triangles Error/Exit Next Frame Real-time Updates: Update State - 60 FPS render loop - Update clock time - WebSocket state sync - Update camera matrix - Mouse/keyboard input Update water properties - Camera orbit controls Process user input - Water property changes Render Refraction - Bind refraction framebuffer - Set clipping plane (below water) - Render terrain mesh Apply stone texture Render Reflection - Bind reflection framebuffer - Set clipping plane (above water) - Mirror view matrix Render terrain mesh

Render Main Scene

Bind main framebuffer
Clear depth/color buffers
Render water surface
Render terrain (no clipping)
Apply multi-texturing