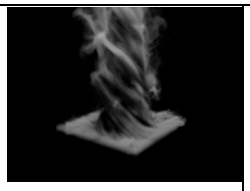
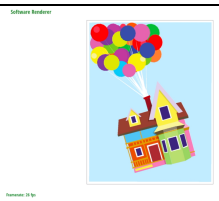
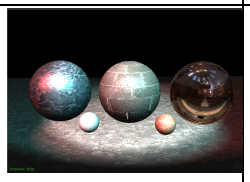
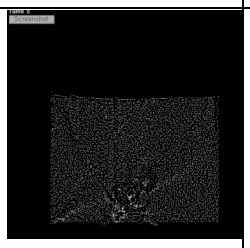
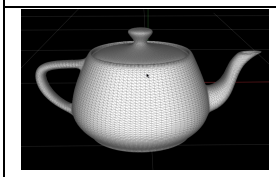
Computer Graphics Projects: juliechavando.com/projects

	**Cat with Coffee: PBRT Image** [C++, Blender] • produced a realistic image using a hair scattering model in the Physically Based Rendering Toolkit • modified and texturized models and leveraged fluid and hair systems in Blender • used an environment map and scaling for the window background		**Four Elements: Special Effects** [Houdini] • leveraged and expanded on simulations in Houdini • wind: pyro smoke sim; upward spiral velocity forces (VEX code) • fire: pyro sim • earth: fractal growth and point replication • water: FLIP fluid solver
	**2D SVG Key Frame Interpolator** [C++] • expanded an SVG rasterizer to perform 2D animations through transformations and interpolations of keyframes • generalized functionality to adapt to any series of SVGs and fill in the blanks to display a smooth animation		**APIC Fluid Simulation** [C++] • implemented the Affine Particle in Cell method based on Jiang et. al [2015] to create 2D water
	**Spaghetti Factory: Particle-Based Collision Processor** [Java] • implemented particle-based collision detection, impulses, and penalty forces		**Real-Time Shader** [GLSL, OpenGL] • implemented Phong reflectance, environment lighting, shadow and normal mapping, and spotlights to improve the quality of a real-time renderer
	**Dancing Groot** [Houdini] • rigged using Mixamo • used copy to points for the modeled leaves and implemented physics based nodes for the leaves to flow as Groot is dancing		**Sand Simulation** [C++] • implemented sliding and rigid friction based on Zhu et. al [2005]
	**Mesh Edit** [C++] • implemented the interactive mesh editing capability of a 3D software		**Tony Stark's Lab: Ray-Traced Image** [Maya, OpenGL] • modeled and texturized objects in Maya • used OpenGL to compose the scene's lighting (including spotlights), reflectivity, transmission, and specular highlights