

Computer Graphics

Example Topics for the Final Project

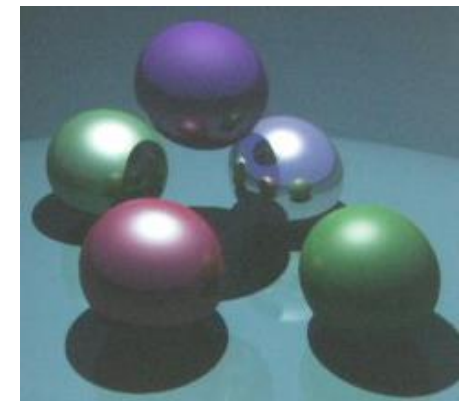
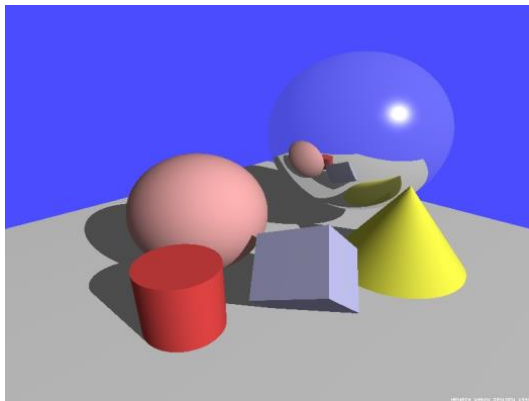
Dr. Renato Farias
rfarias2@ucmerced.edu

Instructions - Summary

- The final project has 3 main required parts:
 - 1) Checkpoint, 2) Presentation, and 3) Submission
 - Submission needs: code & representative image.
- Choose a topic!
 - A custom topic is possible but:
 - You will have to implement a topic related to the class
 - It must be in C++ and use OpenGL (no immediate mode)
 - You must identify "interesting features"
 - Each member must develop one interesting feature of the project
- Presentations
 - As with PAs, demo in lab
 - You have until **YOUR LAST LAB OF THE SEMESTER** (no late demo's or submissions!)
 - You may use your own laptop or a computer in lab
- Grading and Additional Information
 - [Read Final Project Instructions.pdf](#)

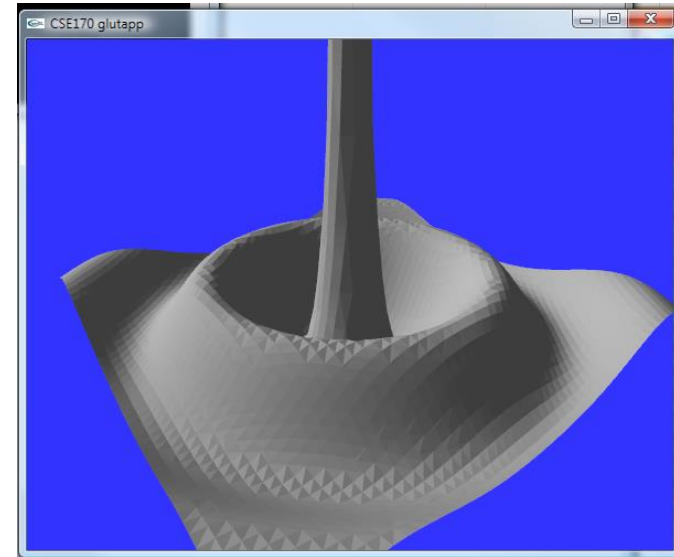
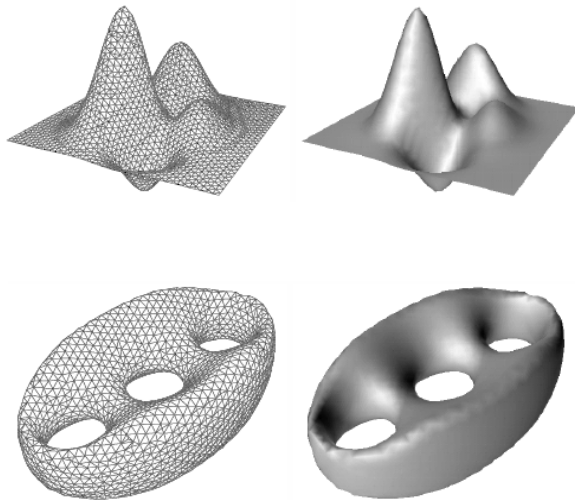
Topic 1 – Ray Tracer

- Goal
 - Render 3D scene descriptions with simple primitives
- Features
 - Produce multiple scenes with light interactions among multiple primitives such as: inter-reflections and shadows
 - Produce videos of animated scenes



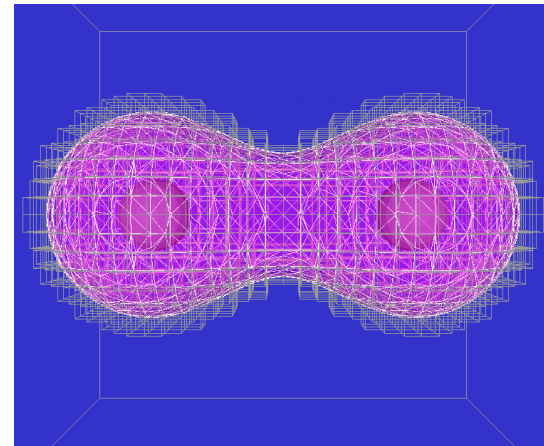
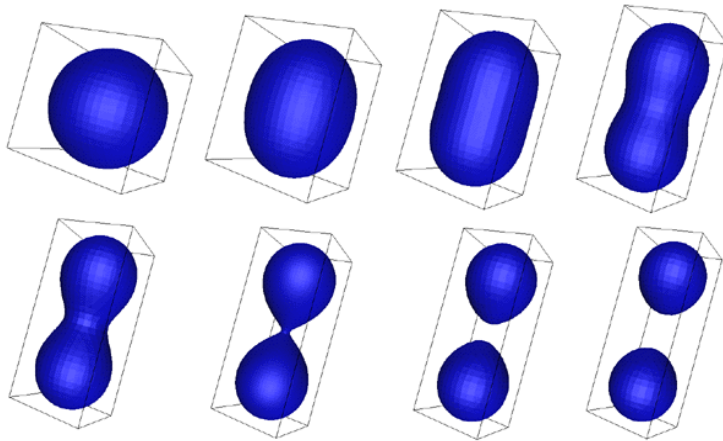
Topic 2 – Marching Cubes/Tetrahedra

- Goal: 3D visualization of several interesting implicit surfaces
- Features:
 - Change the resolution interactively with hierarchical subdivision while showing the results in smooth or flat shading with interesting lighting
 - Include interesting surfaces with holes (see below)!
 - Present an interesting solution to include texture mapping and show how it works for surfaces with holes



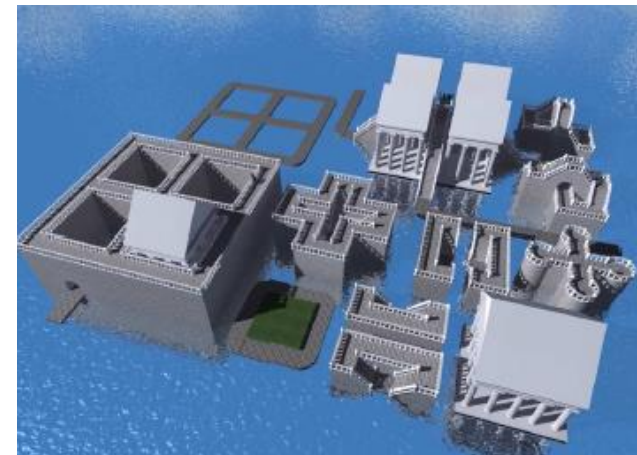
Topic 3 – Metaballs

- Goal
 - Visualize metaball objects with marching cubes
- Features
 - Animate and control multiple 3D points to be used as metaball centers moving around your scene space, and apply your marching cubes algorithm to determine the resulting boundaries in real time
 - Visualize the result in smooth shading with interesting lighting or textures



Topic 4 – Camera Fly-Through

- Goal:
 - Visualize a large environment with smooth camera motions along trajectories (at least 3) inside the environment
- Features
 - The trajectories are based on control points defining a parametric curve that will control both the camera position and the target point of view position. Basically, you will be applying a parametric curve to interpolate/approximate the camera parameters, including the camera orientation.
 - The environment should have some complexity to it, not be too simple
 - You do not have to write the code to load the scene, you may use a library or third-party code to do it. You may find scene files (for ex., .objs) from the internet.
 - Add animated objects to your scene
 - Like cars moving around



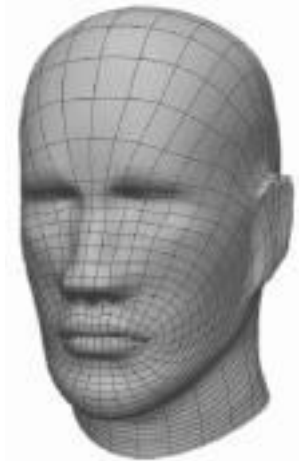
Topic 5 – A Videogame or Simulation

- Goal: build your own video game
 - It must be in 3D!
 - It can also be a "scene simulation"
 - It must include "moving things"
- Features:
 - It includes interesting lighting and textures
 - It implements interesting user control/interaction
 - It includes interesting and non-trivial animations with hierarchical objects



Topic 6 – 3D Mesh Modeler

- Goal: build your own mesh modeling application
- Examples:
 - Head sculpting example: start with any closed smooth surface (sphere, ellipsoid, etc.), then allow the surface to be edited (locally deformed) by selecting a location on the surface and then:
 - One operation will apply local depressions
 - Another operation will apply local bumps
 - Terrain modeler: start with a grid on the floor where for each grid vertex there is an elevation associated with it. The terrain can be generated by a surface passing by the elevated grid points. Then apply operations:
 - One operation will apply local depressions or valleys
 - Another operation will apply local bumps or mountains



Many features possible, for ex.:
add textures!

