

Computer Graphics

Example Topics for the Final Project

Prof. Marcelo Kallmann
mkallmann@ucmerced.edu

M. Kallmann - UCM

Instructions - Summary

2

- The final project has 3 main required parts:
 - 1) Checkpoint, 2) Presentation, and 3) Submission
 - Submission needs: code, and representative image.
- A topic has to be chosen!
 - A custom topic is possible but:
 - You will have to implement a topic related to the class,
 - It has to be in C++ and SIG (exceptions possible)
 - You have to identify "interesting features"
 - Each member has to develop one interesting feature of the project
- Presentations
 - Come prepared to run your project: you will have to run it !
 - I will bring my laptop computer, you can use it
 - You may of course bring your own laptop (check projector connection in advance)
- Grading and Additional Information
 - [Read Proj2-Description.pdf](#)

M. Kallmann - UCM

Topic 1 - Ray Tracer

3

- 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



(these two images are from a previous CSE170 final project!)

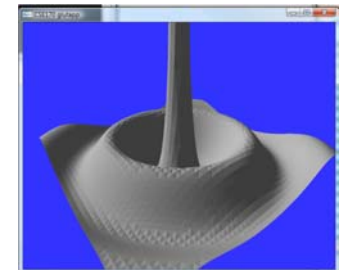
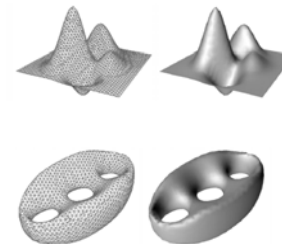


M. Kallmann - UCM

Topic 2 - Marching Cubes/Tetrahedra

4

- 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
 - Be sure to include interesting surfaces with holes (see below)!
 - Present an interesting solution to include texture mapping and show how it works for surfaces with holes (see bottom-left example below)

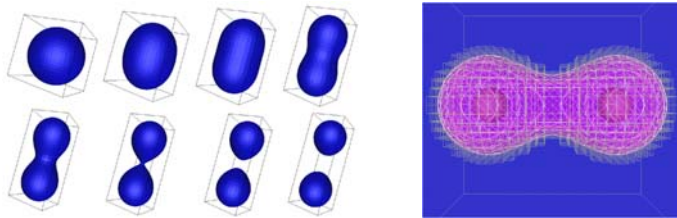


M. Kallmann - UCM

Topic 3 - Metaballs

5

- 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

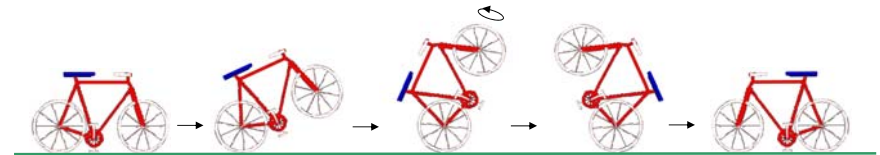


M. Kallmann - UCM

Topic 4 - Keyframe Animation

6

- Goal: use your hierarchical object of Project 1 and build your own library of keyframe animations for it
- Features
 - Define at least 5 non-trivial keyframe animations: each animation is defined as a sequence of poses that are interpolated when activated, then:
 - Have different animations to concatenate as you press some keys in order to achieve complex movements with meaningful global motion
 - Include some special keys to trigger interesting long sequences of animations
 - Make your object to follow a terrain defined by several Bezier patches in a floor grid, and have the control points to be randomly generated. Show different random terrains that you can generate.



M. Kallmann - UCM

Topic 5 – Camera Fly-Through

7

- Goal:
 - visualize a large environment with smooth camera motions along trajectories (at least 3) inside the environment
 - 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.
- Features
 - The environment will have several buildings and also buildings with “at least 2-floors with windows”:
 - At least one camera trajectory will enter inside a 2-floor building and get out through a window without collisions with the building.
 - Add animated objects to your scene
 - Like cars moving around



M. Kallmann - UCM

Topic 6 – A Videogame or Simulation

8

- Goal: build your own video game
 - It must be in 3D!
 - It can also be a “scene simulation”
 - It has to 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



(Note: Support code has a “visibility graph” path planner which can be used for generating paths for things to move around)

M. Kallmann - UCM

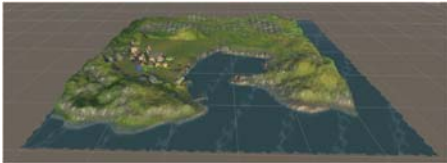
Topic 7 - 3D Mesh Modeler

9

- Goal: build your own mesh modeling application
- Examples:
 - Head sculpting example: start with any closed smooth surface (sphere, ellipsoid, etc). The surface can then be edited (locally deformed) by selecting a location in 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!



M. Kallmann - UCM