UNIVERSITY OF WISCONSIN

Computer Sciences Department EVENTS | SEARCH OUR SITE | CSL

INFORMATION - PEOPLE - GRADS - UNDERGRADS - RESEARCH - RESOURCES

CS559 Fall 2004: Project 3

A (Tiny) Amusement Park

Due Date: 5pm Friday Dec 10 is the latest time for grading.

Project Source Code

The Basic Task

This project will provide you with experience at modeling objects for computer graphics, and introduce you to many more of the features of OpenGL. The starting point is a small virtual space that consists of a grassy square with a roller-coaster track running around it. A carriage (a box for now) runs around the track. Your task is to model and render the carriages and other objects in the environment.

Like project 1, this project defines a set of sub-goals with points awarded for each goal. Unlike project 1, the goals are far more loosely defined, so there is scope to try interesting things to get all the points available.

The Tasks

Each task requires modeling one or more objects using a specific technique from class. The points available for each technique varies according to the difficulty of the task. In all cases, you get a base number of points for implementing one object with a technique, then an extra 5 points for each additional, but distinct, object with the same technique. You can score points for a maximum of three objects with any one technique. For instance, if you create a texture mapped polygonal ticket booth, and a texture-mapped polygonal roller-coaster carriage, and extrude the roller-coaster tracks, then you get 20 + 5 + 25 = 50 points. If an object involves more than one thing, such as a texture mapped, swept surface, then you can score points for both texture mapping and sweep objects.

The maximum number of points is 100. As with Project 1, you can do as much as you like, and we will truncate to 100 as the final step in computing the grade.

The individual tasks, point value, and example objects are:

Technique	Requirements	Points	Suggestions
Texture Mapping	Add texture mapped polygonal objects to the environment. Each "object" for grading purposes consists of at least 5 polygons all texture mapped. Different objects require different maps.	20	Buildings, walls, roadways
	Add a hierarchical, animated model. The model must combine multiple components in a	25	

Hierarchical Animated Model	transformation hierarchy. Different models need different hierarchies.		Ferris Wheel, any number of other fairground rides.
Paramatric Instancing	Add an object described by parameters. You must create multiple instances with different parameters, and each class of model counts for separate points, not each instance.	20	Trees (cones on sticks), buildings, even rides
Sweep Objects	Add an object created as a sweep, either an extrusion or a surface of revolution. The important thing is that it be created by moving some basic shape along a path. The overall object must use at least three different uses of the swept polygon. In other words, something like a cylinder isn't enough, but something like two cylinders joined to form an elbow is.	25	Rails for the roller-coaster, trash bins, trees
Subdivision	An object defined using the modified butterfly scheme. You must include a key press that refines the model, so that we can see the improved quality. The sphere example from class can help, somewhat, with this.	50	The roller-coaster car, organic looking roofs, ...
Change the Navigation System	The navigation system now is not great. Change it to something better. To get all the points, you must have a mode where the viewer rides the roller-coaster.	20	Ride the roller coaster, hotkeys to jump to specific views (good for demos), many possibilities.

Multiple objects modeled using the same technique must be significantly different in order to gain the additional points. Objects that differ only by a sequence of transformations are not different, nor are parameterized objects that only differ by the parameter. Hierarchical objects with the same hierarchy and animation are not different.

Make models that truly display the underlying technique. For instance, make subdivision surfaces that really are curvy.

The Program Skeleton

The program skeleton is an extensive piece of software. Here are some notes on its operation:

- Back-face culling is turned on, so faces with normals that point away from the viewer will not be drawn. Polygon vertex order is used to determine that normal, so you must always order polygon vertices counter-clockwise about the outward-pointing normal.
- One light is defined and turned on. It is a directional light source with no specular component, so do not define specular components for your models. The light is coming in at an angle, so all the faces of an upright box have different shades. With lighting turned on, you must specify normal vectors for all your objects.
- Texture mapping is used for the ground plane. In particular, it uses mipmapping and a targa image. You do not have to use either of these features with your own texture mapping, but for anything other than the simplest textures, it is probably sensible (and you can copy the code).

The Interface

The existing simple interface uses three mouse buttons. Moving the mouse with the left button pushed rotates the world about a fixed point - the look-at point. Moving the mouse up or down with the middle button pushed changes the distance from the viewer to the look-at point. Moving the mouse with the right button pushed translates the world - by moving the look-at point in the ground plane. It's not the best, but it is relatively simple.

Hints and Suggestions

- Have fun and be inventive.
- A key thing to consider is polygon count. Graphics hardware can only display so many polygons in a second, and if you try to display too many the frame rate will collapse. Texture maps also use memory, so too many textures can even more dramatically affect performance.
- The way the train alignment is set up, it is simplest to do just a single carriage, and a short one at that. Doing lots of cars makes it harder to keep then on the track, although it is possible.
- Make use of the OpenGL error checking mechanism. It is described in the OpenGL Programming Guide.
- Start simple - just try to get a polygon to appear in the center of the world.
- The way the current carriage transformations are set up, the origin for the train is assumed to be at the bottom (at track level).
- It is OK to have multiple modeling techniques in one object. For instance, you could have a carriage made up of some texture mapped polygons with some subdivision areas. You get all the points if you do a sufficient amount of each technique.
- It is OK to borrow code from other sources - but not other students. You will probably learn as much trying to figure out how someone else's code works as you would doing it yourself.
- Texture images abound on the web, so feel free to use them. Or you can use a program like Photoshop to create your own. You might even find a use for the first project.

Grading

Grading for this project differs from the others. We will publicize several times in the final week of classes when you can get your project graded. Grading consists of demonstrating your program. **You must have your project graded by Friday, Dec 10 at 5pm.** Once your project is graded, you can have a second chance at grading provided it is before the due date. At most two gradings will be allowed for any one person. Once you are graded, the project is finished for you.

There is no late submission. University rules prevent grading during the exam period.

Back to CS559 Home Page