Computer Graphics Final Project Writeup

Daniel Kiepfer, 12/13/07

User Manual

When the program first starts, the user will be presented with a mostly blank screen, and instructions on how to start a game. The user can click one of two buttons to start a new game. One button represents a normal game difficulty level, and the other represents a hard difficulty level. Once a game has begun, the user can pause it by hitting the pause button.

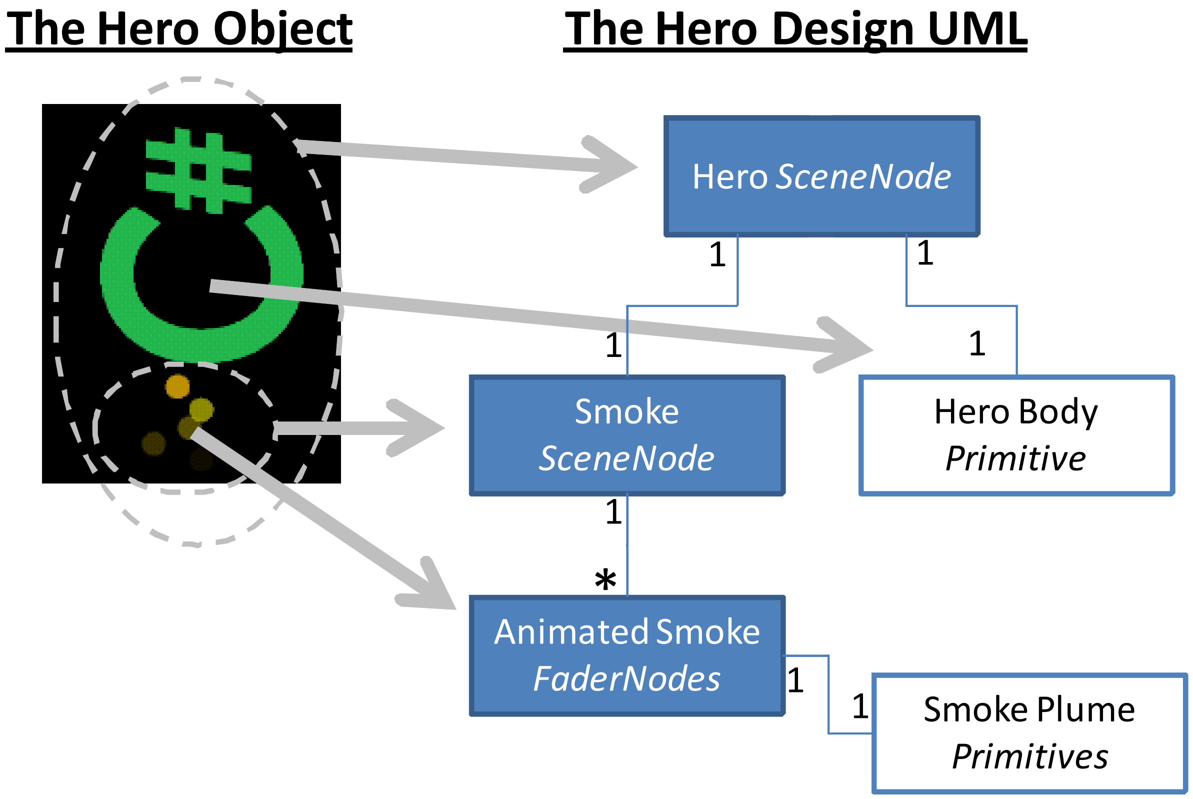
This game is a simple remake of the old Asteroids video game. The user controls the C# logo using the arrow keys: left and right rotate the logo, up accelerates it in the direction it's facing. Hitting the spacebar fires a "bullet" from the front of the C# logo.

Representing a danger to the user are a lot of unruly, unmanageable C++ code objects flying around, which bounce off each other. Hitting a C++ code object once with a bullet causes it to split into four memory leaks, which do not bounce off each other. It is up to the user to hit those memory leaks with bullets so that they can be garbage-collected (destroyed). The game is finished once all memory leaks have been garbage collected.

The user has two ways to view the world in the game. On the left side of the window is the large main viewing area displaying a subset of the whole game world. On the lower right of the window is a smaller, complete view of the world. This view will always show the entire world. The grey box in the complete world view indicates which part of the world the main view is looking at. The grey box can be manipulated with the mouse: clicking and dragging the left mouse button will translate the box around (with the main view updating itself at the same time), and clicking and dragging the right mouse button will cause the grey box to grow or shrink, such that the main view zooms out and in. The grey box will also move automatically such that the C# logo never disappears off the main view.

Hero Design

The design of the hero is very simple. It consists of a SceneNode with a single primitive that has the C# texture. A child SceneNode holds the exhaust FaderNodes (a subclass of SceneNode that fades away after a specified length of time). Each FaderNode contains a primitive circle representing a plume of "smoke". The smoke is animated both when the hero is stationery and when it's accelerating. My SceneNode classes correctly implement cascading Matrix transforms, and are able to contain both other SceneNodes and geometric primitives. Collision detection is done at the SceneNode level, and each SceneNode knows its own cumulative transformation matrix.



System Evaluation

Known Issues

Currently, all objects (including the C# logo) bounce off the edge of the world at their center rather than their edges.

Program Limitations

There are no strict limitations to the program, although I have noticed it to be more choppy than some of my colleagues' programs. This is likely due to the managed code that it was written in.

Possible Future Features

This list could be long. I could add levels to the game, such that finishing the game automatically starts a new game at a higher difficulty level. I could add other types of objects that could either a) shoot at the C# logo, or b) influence its movement (e.g. a gravity well). I could add some kind of scoring system. I could add a different type of bullet, perhaps one that explodes into many pieces of shrapnel a specified time after leaving the C# logo.