

TJHSST Senior Research Project Proposal: Development of a 3D Graphics Engine 2006-2007

Kevin Kassing

September 13, 2006

1 Introduction: program versions

1.1 1st Quarter

Since I have some background in programming 3D graphics with OpenGL, I have reason to believe that I can make significant progress quickly. By the end of the first quarter, I believe that I can have the following features completed:

- Texture loader for PNG, BMP, and PCX formats
- MD2 animated model loader and animation rendering support
- Methods to generate basic geometry
- C++ objects representing OpenGL concepts, such as lights
- Scenegraph tree node objects for BSP and Octrees
- Movable camera interface
- Flexible performance tuning mechanisms

1.2 2nd Quarter

I have researched following features, but I have not implemented them in code. I believe that it is feasible to include these features into the engine before the end of the second quarter:

- MD3 or MD5 skeletal animated model loader, without seamless skinning
- Collision detection and collision response modelling linear kinematics
- Rendering of Non-Uniform Rational B-Splines (NURBS) surfaces
- Animated skyboxes and skydomes

1.3 3rd Quarter

The following features are things that I have not yet researched or have only skimmed over. These would be the "icing on the cake" that induce suspension of disbelief. I do believe that it would be feasible to incorporate the following features into the engine before the end of the third quarter:

- Support for linear and rotational kinematics
- Fog and reflection effects
- Single texture terrain generation using fractals
- Scripted sequences

1.4 Extra goals

If I manage to finish all the mentioned features, I could expand on the engine with one or more of the following, which I do not expect to finish before the start of the fourth quarter, but would like to implement nonetheless:

- Full scripting capabilities, including an in-engine console for dynamic modification of simulations
- Level of Detail (LOD) meshes for decreased cost of rendering for meshes far from the camera

- Customizable particle effects for fire and similar phenomena
- Bump mapping, normal mapping, and environment mapping
- Support for vertex and pixel shaders
- AI agent management system with flexible message routing
- Support for loading more 3D model formats