Carleton University, Computer Science, Winter 2011

**Comp 4002 Real-time 3D Game Engines**

Course Outline

**Introduction**:

The build process; builder versus engine, the software provided including modeling tools, transformations and their inverses, the SRT order, the model/view/perspective matrix, transformation pipeline, game images.

**Cameras**

Objects versus cameras (the role of inverses), movements via pre- and post-transformations, the dreaded navigation spiral, cinematic cameras.

**Environmental effects**:

Skyboxes, moving clouds, cloud shadows, rain, underwater effects, sunbeams, lens flares.

**Sprites**

Applications, large sprites need special rotational techniques (an object look-at versus a camera look-at), non-rotational techniques (drawing in camera coordinates).

**Terrain**

Height map textures, large worlds, the need for vertex buffers, frustum culling, transforming planes, fast bounding box frustum culling.

**Collision Detection (can see?/can hit?) and Visibility (should draw?/drawing order?)**

Applications, partitioning worlds via binary space partitioning (BSP) trees or bounding box (BB) trees. Point collision detection routines for static and dynamic objects. Drawing back to front with BSP trees.

**Particles and Triggers**

The need for a particle drawing phase distinct from other phases, the need for particle sorting, emitters, triggers, particle managers.

**Portals**

Indoor region based visibility.

**Waypoints**

Navigation and materialization points, guards, create traffic flow of vehicles and AI creatures, waypoint managers.

**Mirrors**

The reflection transformation, stenciling inside the mirror, winding order flipping, recursive mirrors, remote cameras, interdimensional doors..

**Shadows**

Shadows maps (requiring shaders and the ability to generate z-buffer textures) versus shadow volumes (requiring a stencil buffer).