



Doox3D Engine Specification

Contents

1. Character Animation/Avatar System
2. Scene Management
3. Rendering
4. Special Effects
5. Collision Detection
6. 3ds Studio MAX Plug-in

System

1.Character Animation System



Pic. 1



Pic. 2



Pic. 3



Pic. 4

•Each parts of characters can be easily replaced and scaled maintaining seamless skin animations like **Pic. 1** and **Pic. 2**.

•Engine's exporter plug-in for 3ds MAX generates a character structure type file(.PTY), files for each body part meshes(.PCP), files for each body part animations(.PAP), a file for combinations of body parts, a file for combinations of animations so that various customizable characters can be easily produced without further content change.

•By using engine's character animation system, not only various shapes of characters can be easily created but also various combinations of body part animation can be produced right away.



Pic. 1



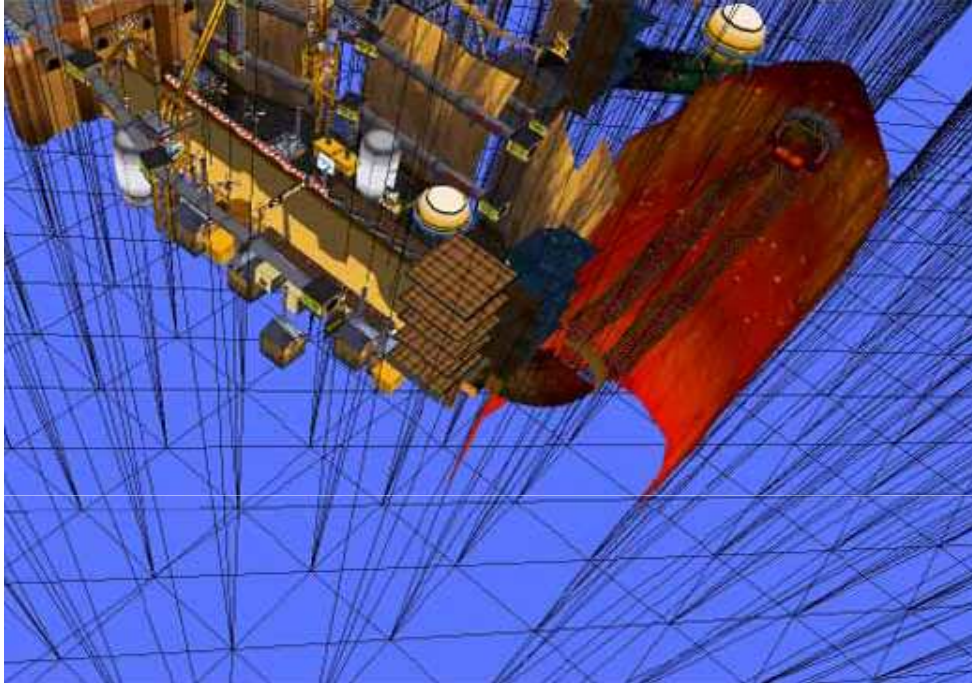
Pic. 2



Pic. 3

- **Character designers can use 3ds MAX Character Studio's Biped and Physique to create various characters and export it into the engine's content files using built-in 3ds MAX exporter plug-in application.**
- **By using 3ds MAX Character Studio's motion captured data, animators can create realistic animations from real life like Pic. 1.**
- **Engine supports skin animation with seamless scalability of each parts like Pic. 2 so that various types of characters can be produced in real-time and also character designers can iterate their design ideas more quickly.**
- **Any items can be attached/detached easily on characters like Pic. 3.**

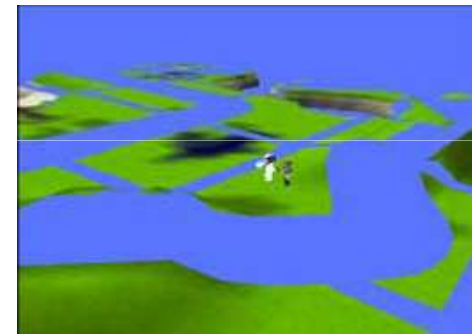
2. Scene Management



Pic. 1



Pic. 2



Pic. 3

•Engine's exporter plug-in for 3ds MAX also generates data for optimized scene management. This exporter divides the 3D game world into Octree based cells like Pic. 1 and creates data for engine's view frustum culling and collision detection.

•The exporter also divides a big mesh like height map plane into pieces of meshes that fits in Octree cells for optimized rendering performance like Pic. 2. and Pic. 3.

•This scene management data exported from the plug-in application can be used various types of optimization such as rendering, collision detection and AI by narrowing down the game instances to the ones only inside the cells that matter.

3. Rendering



Pic. 1



Pic. 2



Pic. 3

- Engine also supports various Vertex/Pixel shaders.
- Pic. 1 shows a game object with Fixed Function rendering.
- Pic. 2 shows the one with cartoon shader applied.
- Pic. 3 shows the one with cartoon shade and cartoon outline applied at the same time

4. Special Effects



Pic. 1 Particle System



Pic. 2 Motion Blur

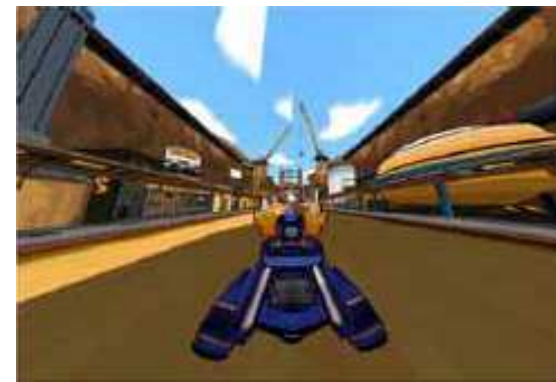


Pic. 3 Motion Blur

•Engine's particle system like Pic.1 manages many instances of particles in a memory pool to generate, transform, render and recycle them fast in an optimized way.

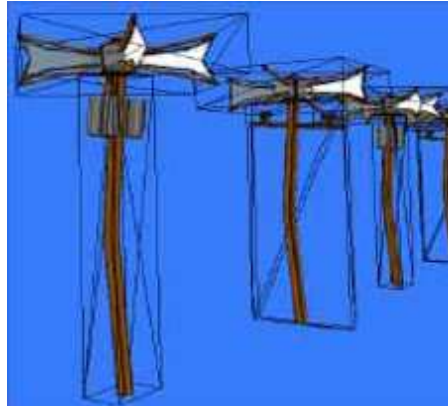
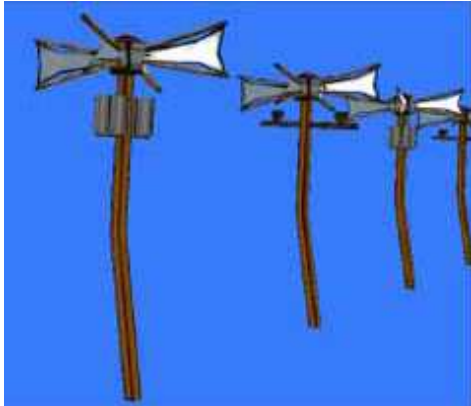
Pic. 4

•Motion blur effect like Pic. 2 and Pic. 3 can be used to visualize the feeling of speed effectively.



•Depth of field animation can be used to visualize extreme speed or dreamy atmosphere.

5. Collision Detection

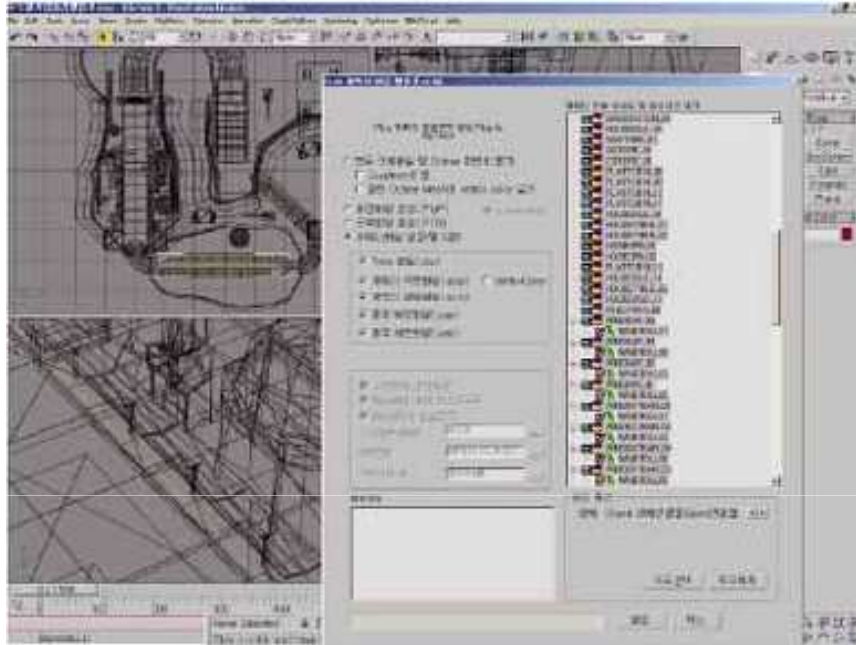


• Every game object has its own bounding volume (Sphere, AABB, OBB) automatically calculated by the engine. Engine's collision detection system supports various functions to calculate collisions between combinations of these bounding volumes with minimum calculation.

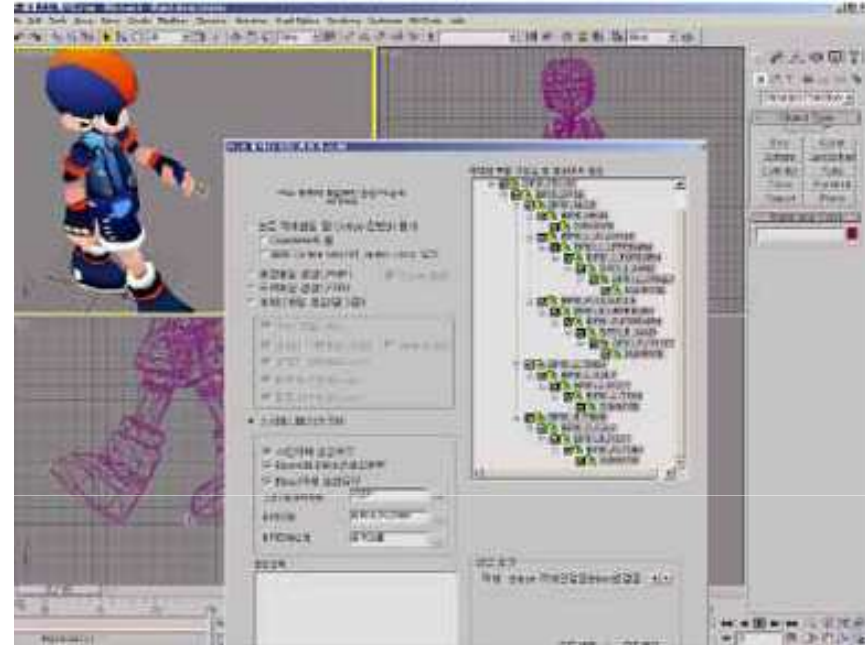


• Also, polygon based collision detection is implemented for more detailed collision cases. Pictures above show dummy polygons placed to detect collisions between the car and the background. These dummy polygons are also divided into Octree based cell for optimization.

6. 3ds MAX Exporter plug-in



Pic. 1



Pic. 2

•**Pic. 1 shows constructing map, locating game objects and generating scene management data out of engine's 3ds MAX exporter plug-in application.**

•**Pic. 2 shows exporting character meshes and animations into engine's content files using the exporter.**

•**Exported file formats:**

character type(.PTY), character part meshes(.PCP), character part animations(.PAP), character mesh combination(.PCS), character animation combination(.PAS)

Ocree data file format for static game objects(.POC)