



Descensor Technical Data Sheet

This is a brief introduction to technical details about Descensor, the real-time procedural world generation engine.

Deliverables

Descensor is provided as a library (static or dynamic) and has to be shipped together with your program code.

API

The engine provides a simple and clean ANSI-C API. This API, accessible through the provided header (.h) files is used to initialise the library, configure it, notify the camera position, and render a single frame.

Descensor detects which objects are visible from that camera position and computes them at the required level of detail.

Integration with other engines

Descensor integrates easily with other engines because it does not control the rendering. It just uses the graphics API (DirectX, OpenGL) to create static geometry.

Typically for each frame in your application you would:

1. Start a new frame using the render API (DirectX/OpenGL).
2. Call the Descensor API render function to generate the static geometry.
3. Call your own or 3d party engine code to do the rendering of other objects, static or dynamic (for instance animated objects).
4. Finish the frame notifying the render driver to present the scene in the graphics device.

Note: steps 2 and 3 might be swapped.

Resource limits

The API allows limiting the maximum amount of RAM used by the engine.

It is possible also to specify the time the engine might use to render a single frame; the generation level of detail is adapted accordingly.

Descensor Editor

The editor is a Windows GUI application that allows visualizing the Descensor world while you customize it to your application needs.

With the editor you can import meshes, materials, and define generation constraints.

Your world specification is stored in a small text file

(it does not contain 3D modelling data). You can share this file to reproduce the scenario in several computers.

Mesh import

The user can import meshes from 3D studio files (.3ds). These meshes are loaded as Descensor objects when they become visible.

The engine uses 3D Studio meshes internally for basic objects like cars and streetlamps.

Material import

Materials (including textures) are loaded from 3D studio files (.3ds). The user can add its own materials and textures.

Terrain

Descensor provides fractal generation terrain functions. You can also provide your own terrain based on a data elevation map file (new feature being developed).

Generation constraints

Descensor can generate entire scenarios with very little configuration. Nevertheless it is possible to define constraints on the allowed object typologies (for instance building styles) and the generation area limits (this allows to create cities on specified locations).

Portability

The engine is written in portable standard ISO-C++. The code compiles with MSVC and GCC compilers.

The engine has been successfully ported to RedHat Linux. Ports to Xbox or Playstation are also possible. Descensor engine currently uses DirectX render API. An OpenGL driver can be provided if required.

Specifications

- CPU: 800 MHz P3 or AMD Athlon (2 GHz P4 or AMD Athlon XP 1800+ recommended)
- OS: Win98, Win98 SE, Win ME, Win 2000 or Win XP (Home and Professional)
- 128 MB RAM or higher
- 2 MB free hard disk space. Engine library footprint: 1 MB. Basic mesh and material files: 1 MB.
- DirectX 8.1 compatible graphics card with 16MB RAM or higher.
- Frame rate: 20-60 FPS depending on detail and hardware.