

Figure Generation NG (Next Generation)

Introduction:

This document is intended to describe the next generation of character rendering engines for Maxis. The original effort was built in SimCopter, as a cheap and fast rendering engine for figures.

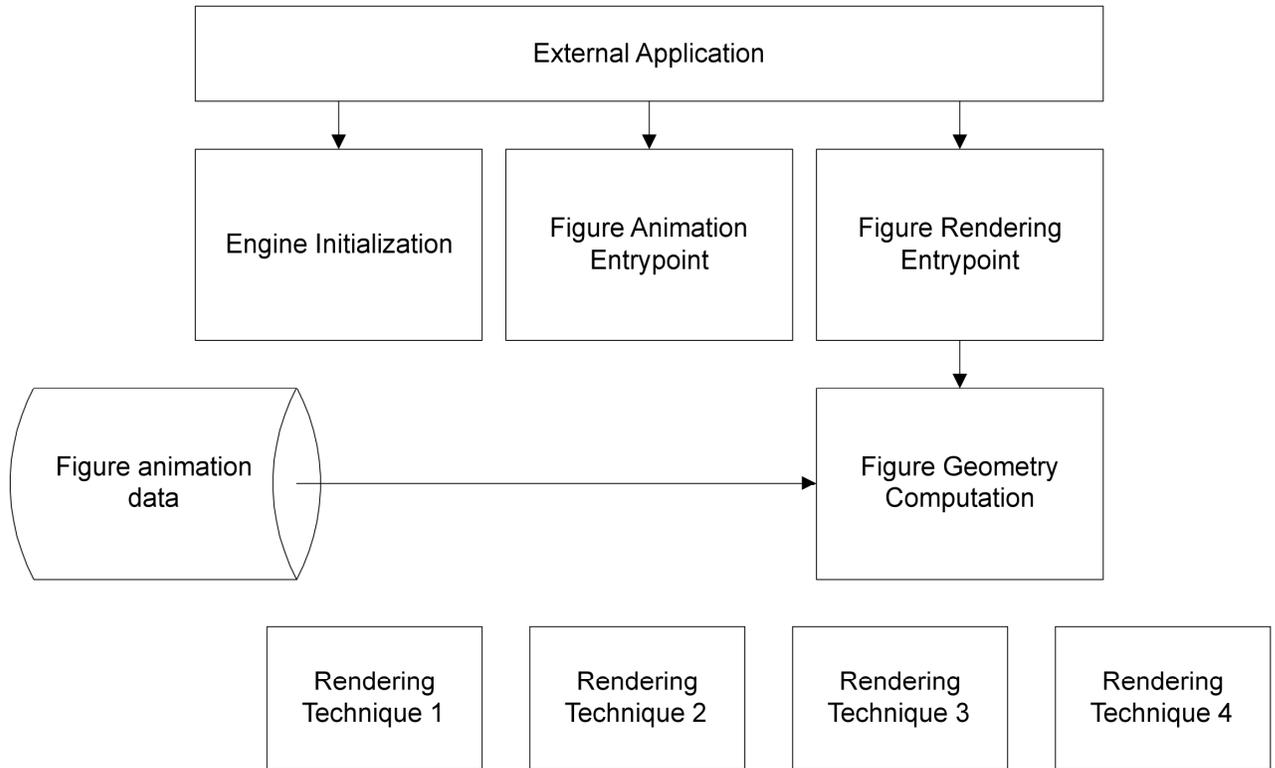
Type of Figure Rendering:

The figure rendering engine will potentially render using several different primitives, or branches of the rendering engine. Each type of primitive, or rendering branch will provide different quality, speed, and technology limitations on the potential uses. Potential rendered figure types are:

- SimCopter (fast) rendered figures. These figures are made up of line draws, spheres, and face decals that is applied to an ellipse. This rendering pipeline is very fast, however the results are very crude.
- SimCopter-like rendered figures, except the figures are made up of true solid primitives, i.e. the arms are rounded tubes, and the torso are spheroids, and the head is a true ellipsoid.
- Polygon built figures. This routine uses animation information in the form of polygon positions per figure movement, and use them to build the figure on the fly in real time. This limits the figure from responding outside of the already scripted animation movements. However, this removes the load from the rendering engine.
- Polygon built figures, with complete freedom. This rendering routine, uses the same primitives to build the figures, however figure movement/motion is directly controlled by the user, i.e. there is no limit to the movement the figure can move in. This requires that joints, or moved polygons be joined at the seams by an algorithm on the fly. Potential solutions to this problem are rendering joining spheroids at joint locations, or an on the fly polygon patching routine.
- Voxel rendered engine. This type of rendering would construct the figures out of volumn pixels instead of polygons. The voxel figure would really be designed as a skin of scalable squares, or cubes that would scale with distance. This allows more detail, as well as a highly scaleable figure representation. The figure animation would likely be built around a completely free motion engine, with full user control.
- MetaBalls?
-

Potential Figure Rendering Layout:

The following is a potential layout of the proposed general purpose figure rendering engine:



Potential Development Path:

A potential milestone laden development path could be:

Milestone 1: Define expected SimCircus figure requirements. Determine animation data stream, as well as figure primitives. Research in figure rendering games, systems etc. to determine what is feasible wrt to SimCircus.

Milestone 2: Build test shell for external application to supply callbacks to figure rendering engine. Build interface to rendering engine.

Milestone 3: Drop in simple SimCopter figure rendering to test figure rendering etc.

Milestone 4: Build figure renderer based on higher-tech approach, a la SimCircus.

Milestone 5: Test new data stream.