

Procedural Modelling of Cities implemented as a Blender Plug-In

Supervisors: Kevin Glass and Shaun Bangay

19/04/2007

1 Previous Short Term Objectives

1.1 Road Generation Test Platform

The objective set for this week was to further extend the Blender script which attempted to implement some of the basic branching systems used in [?]. After some consideration during the previous VRSIG meeting however I have decided to focus more directly on the abstract generation of the road network and leave the specific implementation of the geometry in Blender as a later task.

1.2 Java Instantiation of a Road Network

As a result of the shift in focus for this weeks objectives, I have decided to implement an abstract representation of the various components of a road network in java. The reasoning behind this is to allow me to efficiently organise and control such objects as road segments and intersections. By implementing this in an object orientated language such as Java or C++ I can effectively encapsulate some of the complexity involved with each individual road element within its own class. This allows me to define for instance a road segment not only as a line which traverses from point A to point B, but also within that road segment class the type of road, the angle of the section of road and other attributes can be easily stored and accessed.

1.3 Java - Blender integration

As a result of the change in approach to the road generation problem, I need to find a way to effectively display the java road network in Blender. One possible solution to this is to package the relevant geometry in an xml format and then use a blender script to implement the xml within the Blender scene.

2 Progress

2.1 Java Classes

Over the past few weeks I have begun setting up the various classes and the underlying mechanics of the road network. At present a fairly skeletal framework for the system has been implemented, specifically the basics of a road which consists of a combination of road segments and possibly intersections has been implemented. The progress on these classes to date is roughly up to where the road generation in Blender had been implemented. The advantage however is that now to extend and advance the system is far easier and less tedious.

2.1.1 Road Network

This class is a simple overview class which will contain a collection of various roads and highways as well as a list of the major intersections between these two types of roadway. Simple data such as the position of the city as a whole and the footprint size of the city can be stored in this class. Further extensions in the future would include storing information pertaining to existing landscape elements which occur in within the boundaries of the road network, information on any specific bodies of water or mountains can be stored within this class.

2.1.2 Highway Class

This class is used to store the information pertaining to a highway type of road system. Included within this class will be such attributes as length and size as well as the overall layout of the highway. Also implemented within this class will be mechanisms for generating and connecting highway segments. This will allow for easy implementation and modification of the various L-System and branching techniques which can be used to generate a highway system. Further extensions to this class will include the ability to define the type of highway and such attributes as the width of the highway in lanes. Finally the class can be used to procedurally generate the geometry required to display the highway in Blender.

2.1.3 Road Class

This class is very similar to the highway class, it however will be focused more on the subdivision methods for creating roads. This class may in future have to be coupled with some form of suburb or area class which can be used to determine the type of road layout within the area, i.e. Circular, Raster etc.

2.1.4 Intersection

This class is used to record the intersection of two road segments. Specifically this will be useful for generating the complex geometry required for an intersection. Also this class can be used to record whether the intersection is between two highways and thus generate an interchange type of intersection, or between a highway and a road in which case an offramp must be generated etc.

2.2 Blender Integration

Progress with the integration of the java data into a Blender script has been much slower. At this stage the simple idea is to generate some form of an XML document which can be read into Blender. This solution however is not in keeping with the requirements of the Text-to-Scene system. Specifically I am attempting to find a way to simply call the java implementation from within a blender script and avoid having to run it separately before importing it into Blender.

3 Problems

The only major problem this week is the integration of the Java system into a python script and thus directly into Blender. Further than this all that is required is the extension of the existing java classes and possibly the creation of further classes and objects to better manage the creation of a road network.

4 Objectives for Next Week

4.1 Visible Results in Blender

The objective for next week is to fix the integration problem so that the system can finally focus on testing and implementing various road generation algorithms and methods. As such the objective for next week is to be able to generate images through Blender which accurately reflect the underlying Java implementation.