Procedural Generation of Urban Environments through Space and Time

Jeremy Noghani[1] and Fotis Liarokapis[1] and Eike Falk Anderson[1]

[1]Interactive Worlds Applied Research Group, Coventry University, United Kingdom

Abstract

Although various methods of procedurally generating cities have been proposed in recent years, the problem remains that a user desiring a city from a particular time period or country would have to make time-consuming modifications to an already existing engine to achieve acceptable results. This paper proposes a set of programmable elements that can be adjusted to accommodate for buildings from a broad range of architectural styles, which can then be incorporated into a larger engine.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Simulation and Modeling

1. Introduction

The creation of virtual cities and environments is an important issue in the computer animation, computer games, digital film effects and simulation industries. Procedural generation is often employed by these industries in order to reduce the time taken by animators and programmers for repetitive or tedious tasks. This is primarily done through the use of iterative rules.

The concept of applying rules to architecture and city layouts has been frequently used within the architecture industry in the form of shape grammars [MWH*06], and these have been adapted and applied to computer graphics to create basic virtual cities. The use of L-systems and divided city segments has also been successfully used to create realistic and editable virtual environments [PM01]. More recently, the concept of altering an urban environment through time (a "4 dimensional" city) has been proposed and implemented into city generation engines [WMWG09].

However, in both academic and commercial applications of city generation programs, the resulting virtual urban environments appear to be specific to a single location and time period, often comparable to a modern New York or San Francisco. If a user wishes for the final product to resemble a different time period or country, then they would be forced to make modifications to the engine itself, or to craft their own models and textures. This process can be time-consuming, which often defeats the purpose of turning to procedural generation to begin with, especially in industries that focus on rapid prototyping and fast turn-around. This paper proposes a set of methods that are currently being implemented with the purpose of creating a single application capable of creating urban environments that resemble a country, city, and time period of the user's choosing.

2. Building Elements

During the planning stage, a set of "key" architectural styles were decided upon that can be used to generalise a large number of the world's cities and towns. Then, a list of elements that distinguish the different architectural styles was created, which included elements like the average window dimensions and style, the average building height, the style and angle of the roof, and the dimensions and texture of visible any beams or columns. For the building shape itself, a set of "complexity" values were implemented that define the number of sides and the overall shape of the building. Finally, "deviation" values were implemented that described how much elements of buildings differed within styles. As an example, classic Tudor architecture would have a very low deviation value for its wall colour and beam style, as the vast majority of Tudor houses are painted white with external wooden beams. By using this method, it is possible to render buildings through recursively creating polygons that are textured and translated according to the element values. However, the method has limitations. Buildings that are based upon complex shapes, such as a set of splines, cannot be replicated. Additionally, external staircases, archways or bridges between building structures would be somewhat difficult to implement. Nonetheless, this method would be suitable for the vast majority of the world's architecture, as the problem only becomes apparent in obscure and isolated instances.

Figure 1: *Examples of potential building outputs. Note that this is work-in-progress.*

For the city layout itself, the use of pre-existing methods of creating road layouts through L-systems or recursive grammars would be appropriate. However, in order to keep the roads appropriate to the area and time period, a set of road and city elements would have to be included. These include the road width, the road type (tarmac, gravel etc.), and the distribution and density of the buildings. Once this is complete, the world's major cities can be described through only a few key values. By noting the city size, the location (e.g. is the surrounding terrain mountainous, or is the area near the sea), and the percentage of each architectural style's presence within a city, we are able to replicate the city's structure. If we define a city's values at "key" dates, such as every 500 years, and interpolate the values for the years in between, then a user would be capable of requesting a virtual city from any year or date. The number of key dates required to maintain accuracy is a matter that can be refined during the user testing stage.

3. Other Factors

Aside from the buildings themselves, there are other aspects of environments that need to be considered in order to ensure that the virtual environment is perceived as appropriate to the desired place and time period. It was deemed early on in the project that accurate architecture took priority over these factors (primarily due to time limitations), but implementation of the listed features could greatly enhance the user's feeling of immersion. The in-simulation weather needs to be relevant to the climate of the desired location. This can be approached by either linking the weather to the location's altitude and distance from the equator, or by manually linking each location within the system to a set of weather types and probabilities. The former method would be faster to implement, but the latter would achieve more accurate results, as it accounts for areas with unusual weather. Plant life could be included, the species of which would match the world location. The simplest method of deciding what species is appropriate would be to match the plant type to the climate, as determined by the weather system. A hot and humid environment would probably support tropical or rainforest vegetation, for example. It would be desirable to add citizens to the environment in the form of AI agents, with clothing relevant to the time period and location. However, this issue is complex enough to warrant its own study, and so it was decided that the matter ought to be left until time restraints were no longer such an issue.

4. Testing and Evaluation

Upon completion, the system will be tested thoroughly through user trials. A selection of potential users will be shown example scenarios from the finished system, and will be asked whether they can identify the location and time period of the urban environment. The program can be considered a success if the majority of users manage to answer correctly, within a degree of reason. Additional testing can then be carried out to determine what elements or features make the environment distinguishable from environments in other locations or time periods. This can be done by repeating the trial and questions, only with certain elements (such as AI citizens and vehicles) removed. If users consistently and correctly identify a city as modern London, for example, even with citizens and vehicles removed, then it would be reasonable to conclude that the citizens and vehicles are not an essential element for distinguishing the environment.

From the results of this test, unnecessary elements can be removed, and existing elements can enhanced, creating an application that fully achieves the original aim of being capable of creating a virtual city specific to a geographic area and time period.

References

[MWH*06] MÜLLER P., WONKA P., HAEGLER S., ULMER A., VAN GOOL L.: Procedural modeling of buildings. *ACM Trans. Graph. 25*, 3 (2006), 614–623. 1

[PM01] PARISH Y. I. H., MÜLLER P.: Procedural modeling of cities. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), pp. 301–308. 1

[WMWG09] WEBER B., MÜLLER P., WONKA P., GROSS M.: Interactive geometric simulation of 4d cities. *Computer Graphics Forum 28*, 2 (2009), 481–492. 1