Chapter 2

State of the Art in Automated Design of 3D Game Environments

2.1 Introduction

This chapter reviews the state of the art of automated 3D game environment designing. Based on the findings of our literature review we have categorized researches in to 3 broad areas: traditional procedural techniques for 3D environment generation, applications of multi agent systems and multi agent system based approaches for 3D environment design. Following sections discuss about those 3 areas.

2.2 Traditional Procedural Techniques for 3D Environment Generation

The paper "Citygen: An Interactive System for Procedural City Generation" [9] is presented by George Kelly and Hugh McCabe to solve the problem of time-consuming and expensive content creation process in 3D games. This process requires the modelling of vast amounts of geometric detail including terrain, roads and buildings. This paper presents a system called CityGen in order to automate most of the 3D content generation tasks by mainly focusing on geometries of a modern city. Their system is based on procedural methods to automatically generate 3D cities and this approach has been used in the field of computer graphics over 20 years. Their work has been strongly influenced by the research carried out by Parish Müller and colleagues on the *CityEngine* [15], a system that is capable of producing realistic and detailed models. *CityGen*'s procedural building generation has been focused on the application of grammars to describe building structures. This paper mentioned about other approaches to solve the problem using intelligent agents, evolution of cities by modelling land use to create a cityscape, real time city generation and template based generation. A unique design goal of *CityGen* is to allow the user to control the city generation process by manipulating parameters of generation algorithm through a visual interface. Therefore they claim that *CityGen* is an interactive procedural city generation system. The city generation problem has been divided in to three stages namely primary road generation, secondary road generation and building generation. They have identified the pattern of a primary road network as an essential characteristic of a city and therefore *CityGen* uses this characteristic as the starting point of city generation process. After the generation of primary roads, the secondary road generation is executed followed by the construction of buildings. However this system mainly focuses on city generation and the top down approach proposed by system can be a limitation in the generation of other 3D environments such as forests and interiors.

Pascal Müller and his colleagues have presented *CGA shape*, a novel shape grammar for the procedural modelling of building shells to obtain large scale city models in there paper "Procedural Modelling of Buildings" [14]. This paper claims that this is the first paper to address the aspect of volumetric mass modelling of buildings including the design of roofs. This paper presents an approach for procedural modelling using shape grammars to solve the problem of modelling large three-dimensional environments, such as cities, which is a very expensive process and can require several man years worth of labour. They have been used a shape grammar with production rules that iteratively evolve a design by creating more details. Their approach combines two main ideas of modelling urban environments using shape grammars. The first idea is to generate large urban environments where each building consists of simple mass models and shaders for facade detail which is presented by Parish and Müller [15] and the second idea is to generate geometric details on facades of individual buildings which is presented by Wonka et al. [21]. This paper explains the basic shape grammar which is used in CGA shape and extensions that allow to model complex shape configurations and shape interactions. Also it gives an overview of how to generate mass models and explains how to create facade and roof details. The proposed technique to solve the transition from mass modelling to facade and roof modelling is the main approach presented in this paper. The mass modelling is described in detail using the techniques for assembling solids, occlusion and snapping. Several examples has been presented using simple building model, a model for office buildings and a model for single family homes to describe modelling with *CGA shape*. An evaluation has been carried out for *CGA shape* considering efficiency, robustness, usability and real-time rendering. However CGA shape inherits a general disadvantage of a procedural approach that it sometimes generates configurations of shapes that are not plausible.

The paper "Undiscovered Worlds – Towards a Framework for Real-Time Procedural World Generation" [6] is presented by Stefan Greuter and his colleagues to provide a framework for generation of virtual worlds using procedural generation to minimize the costs for the creation of complex game worlds. Re-using textures and geometries for objects that occur frequently throughout the game and generate recurring objects or even entire game worlds using procedural methods have been analysed as approached to minimize the cost of 3D environment creation. The work presented by this paper is inspired by the previous research in to generation of plants, trees and cities. The proposed framework consists of three major components: view frustum filling, geometry caching and geometry generation. The procedural generation of geometry to regions visible from the viewpoint is limited by *view frustum filling* component. *Geometry caching* component is used to store previously generated geometry in a temporary database and the *building geometry generator* is a collection of procedures for constructing different types of objects. This framework has been tested using an example of a virtual city consisting of regular street grid and skyscraper type buildings. An iterative function system has been used to generate floor plans of buildings, which are extruded to building facades. This paper claims that the proposed framework is not limited to generation of only virtual cities, but by changing parameters of the algorithm it is possible to generate other types of environments. However this framework is not providing support for collision detection and there are no laws of physics preventing the user from flying through a building. In addition to that the framework does not track building changes such as damage or aging. It seems at the time writing this paper the main focus was on generating 3D content than enforcing secondary features such as collision detection.

Ruben M. Smelik and colleagues have presented the paper "A Survey of Procedural Methods for Terrain Modelling" [17] to review procedural methods applied to terrain modelling by evaluating realism of their output, performance and control users can exert over the procedure. The authors of this paper have identified that even though procedural modelling has been an active research topic for at least thirty years, it is not often applied in mainstream terrain modelling. Also they have identified several factors limiting the applications of automated modelling. At the moment both research papers and commercial tools typically focus on one aspect of terrain modelling and address other aspects to a limited extent. The authors of this paper see this as a one of the main limitations. Also they see the integration and adjustment of existing procedural methods in such a way that they can automatically generate a complete and consistent terrain model is an unsolved problem at the time of writing this paper. And this paper mentioned that the lack of control current procedural content generation techniques offered to users as another known issue. This paper reviews several procedural methods by discussing important properties of the methods, such as the realism of the output, the performance of the algorithm and the facilities it provides users to influence and control the generation process. They have reviewed procedural methods for elevation data, water bodies, vegetation, road networks and urban environments. This paper has identified three future directions of procedural modelling. First, performance and interactivity of procedural methods will continue to improve. Second, road networks and urban areas will continue to improve in variation and level of detail, but the improved realism will likely be given by deploying more and more semantics in both the procedural generation process and the generated models. Third, to a widespread deployment of procedural methods by non-experts will be the integration of procedural methods within a framework.

George Kelly and Hugh McCabe have presented their paper "A Survey of Procedural Techniques for City Generation" [10] to analyze the main body of existing research into procedural city generation. Abstraction, parametric control and flexibility have been identified as important features of procedural techniques. Also they have described a number of fundamental procedural techniques and algorithms that have been successfully employed within the domain of computer graphics. These procedural techniques include fractals, L-systems, Perlin noise, tiling and Voronoi texture basis. To evaluate the procedural city generation systems, the authors have identified a number of key criteria: realism, scale, variation, input, efficiency, control and ability to view the city in real-time. An overview and an insight of the functioning of the techniques and algorithms applied in the systems has been presented of each of the city generation systems. The "Undiscovered City" system based on research of Stefan Greuter and colleagues [5], the L-systems based "*CityEngine*" of Parish and Müller [15], the agent based "*CityBuilder*" of Tom Lechner and his colleagues [12], template based generation and split grammars have been studied and evaluated in this research. The final goal of the authors was to capture the information required to create a city generation system suitable for real-time applications that is capable of creating realistic, varied and large scale cites in an efficient manner while remaining accessible to non-expert users. However in the scope of this paper the research is limited to a survey on existing procedural techniques for city generation.

The paper “A Proposal for a Procedural Terrain Modelling Framework” [18] presented by R. M. Smelik and colleagues to propose high-level framework for automatic generation of virtual worlds that requires intuitive user input and results in a rich 3D terrain model. This paper emphasis the importance of automating terrain construction considering the effort in time and money required for designing terrains manually. The authors of this paper have identified two main drawbacks of existing procedural methods. First, it is not clear how to tune individual procedural algorithms to work well together; there is no tool or integrating framework that combines these various algorithms in a usable way. Second, the parameters of these algorithms and tools often require an in-depth knowledge of the algorithm to predict the effect of a parameter on the outcome. The authors of this paper argue that an ideal terrain modeller should be able to generate a large variety of realistic, natural terrains, controlled by a small number of intuitive user input parameters, while still allowing the user to perform detailed fine-tuning of the generated terrain, or (partial) regeneration. They define this as a paradigm shift from *terrain construction* into *terrain declaration*. To propose this approach, they have identified modelling workflow requirements by analysing how terrain and game level designers currently work. An overview of a terrain modelling framework has been provided by describing the workflow procedural modelling. However at the time of writing this paper the research was limited to the proposal of a Procedural Terrain Modelling framework and the realisation of such a framework has been identified as future research.

Tsuyoshi Honjo and En-Mi Lim have presented their paper “Visualization of Forest Landscapes by VRML” [8] to introduce a visualization approach for forest landscapes by using GIS and a plant modelling technique. Also the applications of VRML including server side generation of a VRML program are also described. The computer aided drafting system called AMAP (Atelier de Modelisation pour l'Architecture des Plants) [2] has been used for the visualization in this research. The authors of this paper have studied and presented about the applications of VRML (Virtual Reality Modelling Language) which is one of the high performance language for 3D visualization on the World Wide Web. The visualization process of forest landscapes was divided into three main steps. In the first step, the 3D digital data of the terrain are retrieved from a contour map and then data on the attributes and locations of the dominant trees of each stand are retrieved from a forest stand table and stand map. In the second step, the data on the terrain and vegetation in the forest landscape are converted into VRML format using a conversion program. In the third step, a 3D image of the forest landscape is generated. To evaluate the performance and feasibility of this visualization system, they have simulated an actual forest landscape and compared output of this system with photographic images of forests. However at the time of writing this paper, their system has been limited to the generation of forest landscapes based on GIS data of actual forest landscapes and it not extended to generate unseen or imaginary landscapes which are common in video games.

From the literature review of above mentioned approaches we have identified that the traditional procedural techniques for 3D environment generation are only focused on generating a specific type of 3D environments, for example 3D environments of cities. In addition to that these traditional techniques are very algorithmic and do not operate based on natural factors of environment evolution. The standard method in these techniques to control the generated environment is by modifying parameters of algorithms. Therefore there are no facilities to control the arrangement of objects in 3D world by applying real world natural conditions. Also the top down and centralized approaches which are used in these traditional procedural techniques prevent game developers from using simple decentralized rules which are reusable in different environments.

2.3 Using Multi Agent Systems for Problem Solving

The paper “Go to the Ant: Engineering Principles from Natural Multi-Agent Systems” [16] is presented by H. Van Dyke Parunak to summarize the studies of multi agent systems in natural world. Prior to analyzing the natural multi agent systems, the author of this paper has compared the level of encapsulation and localisation used in monolithic programming, structured programming, object-oriented programming and agent-oriented programming considering ”How does a unit behave?”, “What does a unit do when it runs?” and “When does a unit run?”. Also the advantages of using an agent based bottom up approach in software development over the traditional top down approaches have been discussed. The paper has defined and described the theoretical context of a multi agent system as a set of agents, an environment and coupling between them. Homo-dynamic systems and hetero-dynamic systems were mainly considered under coupling of systems. As the main goal of this research, the path planning of ants, brood sorting of ants, nest building of termites, task differentiation of wasps, flocking of birds/ fishes and surrounding prey of wolves have been considered as natural multi agent systems. These natural systems were studied and compared considering system behaviour, responsibilities and integration. A set of general principles have been derived as a result of the study. These principles are described under several topics: considering agents as things rather than functions, keeping agents small, decentralized system control, supporting agent diversity, providing an entropy leak, enabling agents to share information and planning/ executing concurrently. This paper claims that these principles can be used to support multi agent system behaviours which are significantly more complex that the behaviour of individual agents. In addition, the paper "Applications of Self-Organizing Multi-Agent Systems: An Initial Framework for Comparison" [1] which is presented by Carole Bernon and colleagues provides several examples self-organization of multi-agent systems which are used to solve complex problems.

Erick Baptista Passos and Esteban W. Gonzales Clua have presented the paper "Multi Agent System for Intelligent Game Cinematography" [4] to improve the visual experience of spectators in massively multiplayer online games. The movie industry and academia have already used the concepts of cinematography, including methods for controlling cameras and lights in a scene, which can influence the perception of moods and emotions of the audience. However the video games are interactive applications and automatically controlling a camera in a game environment is difficult and different from films, because there is no prior knowledge over the events happening in the scene. This paper proposes a multi-agent system to efficiently distribute the tasks needed by intelligent camera control for game spectators. This system mainly consists of three agents named Director Agent, Editor Agent and Cinematographer agent. This multi agent system creates a real-time cinematographic experience for game spectators through the communication among agents. The architecture and design of the system is completed. However the implementation of the system has not started at the time of writing this paper.

The paper "Multi-agent modelling in comparison to standard modelling" [11] is presented by Franziska Klügl and colleagues to compare multi-agent modelling with standard modelling. The authors have identified that designing a multi-agent simulation is very effortful, because there may be a large amount of parameters in the model. Therefore they argue that a modeller has to be sure about the advantages gains from multi-agent modelling. To address this, they have used an example of a bee recruitment model. This example has been modelled as Queuing Networks, Petri-Net, Cellular Automata and multi-agent systems. As a result of above exercise the authors have identified some properties for a modelled system that recommends the use of a multi-agent simulation. However they have identified that the frameworks and methods available for designing and simulating a multi-agent at the time of writing this paper are not mature enough. Because of that they propose theorists and multi-agent practitioners to learn from established modelling techniques also.

It is evident from above literature that Multi Agent Systems technology can introduce very high level of autonomy and the self organizing behaviour in modelling real world problems. Therefore this paper postulate that self organizing feature of multi agent technology can be used to automate the design of 3D game environments.

2.4 Multi Agent System based Approaches for 3D Environment Design

Thomas Lechner and his colleagues have presented the paper "Procedural City Modelling" [12], to propose a method to procedurally generate cities. The main intention of this research has not been to reproduce existing cities, but to generate artificial cities that are convincing and plausible by capturing development behaviour. Their approach was mainly focused on land usage and building distribution for creating realistic city environments based on a multi agent system. In their paper the authors have discussed about the importance of procedural modelling of cities in gaming industry, filming and architecture. They have identified that previous attempts [15, 21] of graphical community to model cities have been primarily focused on road networks and the L-system based procedural content generation has been mainly considered and studied. Their proposed system allows users to specify input parameters and control the appearance of the generated city. Two agents have been used in this system: commercial developers and residential developers. Each developer roams the world within a certain distance of the road network. The developer agents place building on land by evaluating plots near the road.

The paper “Simulation-Based Generation of 3D Urban Environments using a Multi Agent System” [7] presented by Reza Haddadi and Andrew Jönsson to determine how suited Multi Agent System is for generating urban environments which are usable in computer simulations. This research has been conducted to propose a system which supports fully automatic procedural generation, avoid producing anomalies, uses parameters as input, scalable and usable with a modern personal computer. A study on Multi Agent Systems technology has been conducted followed by a study on aspects of using Multi Agent Systems. Also challenges related with mimicking real world feedback effort has been discussed. A critical review on existing procedural content generation methods has been done considering grid layouts [6, 10], Lindenmayor systems [15], template-based generation, data oriented methods and semi-procedural tools. In addition to that the authors of this paper have identified the research done by Thomas Lechner and his colleagues [12] as a previous attempt to use a multi agent system to model cities. In their proposed multi agent system, a commercial agent, an industrial agent and a landmark agent has been used as main agents. Also research has been conducted to generate terrains such as forests, mountains and water also. However since the main focus of this research is to generate cities, the quality of terrain generation has been considered as a lower priority. A prototype system called “Urban” has been implemented to evaluate the proposed approach considering extendibility, adaptability, performance, stability and realism.

The multi agent based approaches proposed by both of above papers are very natural and practical, however these systems are focused only on city generation and outcome is limited to urban environments. And the paper “Procedural City Modelling” [12] is limited to generating a 2D view using NetLogo tool at the time of writing this paper, their system was only limited to generating tertiary road networks. The agents such as developer agents and land mark agents defined by above approaches may suitable for city environments, since in real world also these activities are carryout by similar human actors. However these agents are not natural in environments such as forests where the trees and other entities are self organising without the supervision of any top down control of human actors or any other entity.

2.5 Major Issues Identified

We have identified several major issues in current approaches for automated 3D game environment design. Current 3D environment generation techniques are specific to one type of environment such as city environment and there are no customizable frameworks available which are common to many types of environments. Also in traditional top down methods, there is no natural correlation between real world and 3D world; therefore it is impossible to control 3D world generation using natural parameters. In addition to that the data representation with minimum memory consumption, time to generate a 3D environment and realism of generated 3D environments has been identified as major issues.

Among above mentioned issues we have identified that current 3D environment generation techniques being specific to one type of environment such as city environment and the lack of customizable frameworks which are common to many types of environments as the main issue to be addressed. Also the solution to this problem would at least partially address some of the other problems as well. For example the solution to above mention problem will solve the problem of keeping a natural correlation between real world and 3D world and controlling 3D world generation using natural parameters.

2.6 Summary

This chapter reviewed the state of the art of automated 3D game environment designing. The review has been conducted for traditional procedural techniques for 3D environment generation, applications of multi agent systems and multi agent system based approaches for 3D environment design. Major issues identified have been stated at the end of this chapter. The next chapter discusses about the applicability of multi agent technology to solve the major issues identified in this chapter.