Multi Agent Based Approach to Assist the Design Process of 3D Game Environments

Ramesh Maddegoda
Prof. Asoka S. Karunananda

Faculty of Information Technology,
University of Moratuwa,
Sri Lanka
Introduction

- A new report from DFC Intelligence forecasts that the global market for video games is expected to grow from $66 billion in 2010 to $70.1 billion in 2015.

- Video game developers put a significant amount of effort in designing 3D game environments.

- Designing 3D game environments is a time-consuming and very expensive process that requires high-end proprietary software.

- Hundreds of developers work for several years to deliver some modern 3D video games.
introduction (cont…)

- Obviously a potential solution for the above problem would be to automatically arrange 3D objects in a given 3D environment

- However traditional procedural techniques for 3D environment generation are only focused on generating a specific type of 3D environments
State of the Art in Automated Design of 3D Game Environments

- Our literature review was categorized in the following areas.
  - Traditional Procedural Techniques for 3D Environment Generation
  - Applications of Multi Agent Systems and Multi Agent Based Engineering Design
  - Multi Agent System based Approaches for 3D Environment Design
  - Major Issues Identified
### Traditional Procedural Techniques for 3D Environment Generation

<table>
<thead>
<tr>
<th>Research</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Citygen: An Interactive System for Procedural City Generation</strong> - by George Kelly and Hugh McCabe, 2007</td>
<td>An interactive procedural city generation system, Top down approach</td>
</tr>
<tr>
<td><strong>Procedural Modelling of Buildings</strong> - by Pascal Müller and his colleagues, 2001</td>
<td>Using shape grammars, Specific to buildings</td>
</tr>
<tr>
<td><strong>Visualization of Forest Landscapes by VRML</strong> – by Tsuyoshi Honjo and En-Mi Lim, 2002</td>
<td>Forest landscapes with VRML</td>
</tr>
</tbody>
</table>
### Traditional Procedural Techniques for 3D Environment Generation (cont…)

<table>
<thead>
<tr>
<th>Research</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A Survey of Procedural Methods for Terrain Modelling</strong> – by Ruben M. Smelik and colleagues, 2009</td>
<td>A survey to identify limitations in current terrain modeling techniques</td>
</tr>
<tr>
<td><strong>A Survey of Procedural Techniques for City Generation</strong> – by George Kelly and Hugh McCabe, 2006</td>
<td>A framework for generation of virtual worlds using procedural generation</td>
</tr>
</tbody>
</table>
## Applications of Multi Agent Systems and Multi Agent Based Engineering Design

<table>
<thead>
<tr>
<th>Research</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Go to the Ant: Engineering Principles from Natural Multi-Agent Systems</strong> - by H. Van Dyke Parunak, 1997</td>
<td>Summarizes the studies of multi agent systems in natural world.</td>
</tr>
<tr>
<td><strong>Applications of Self-Organizing Multi-Agent Systems: An Initial Framework for Comparison</strong> - by Carole Bernon and colleagues, 2006</td>
<td>Provides several examples self organization of multi agent systems</td>
</tr>
<tr>
<td><strong>Multi Agent System for Intelligent Game Cinematography</strong> – by Erick Baptista Passos and Esteban W. Gonzales Clua, 2007</td>
<td>A multi agent system to efficiently distribute the tasks needed by intelligent camera control for game spectators</td>
</tr>
<tr>
<td><strong>Multi-agent Modelling in Comparison to Standard Modelling</strong> – by Franziska Klügl and colleagues, 2002</td>
<td>Comparison of multi agent modelling with standard modelling</td>
</tr>
</tbody>
</table>
### Applications of Multi Agent Systems and Multi Agent Based Engineering Design (cont…)

<table>
<thead>
<tr>
<th>Research</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A Framework for Distributed Agent-based Engineering Design Support</strong></td>
<td>An approach to design support through agent oriented technology</td>
</tr>
<tr>
<td>– by Brian Lees, Cherif Branki and Iain Aird, 2001</td>
<td></td>
</tr>
<tr>
<td><strong>A Multiagent System for the Support of Concurrent Engineering</strong></td>
<td>A conflict resolution mechanism based on negotiation</td>
</tr>
<tr>
<td>- by Hu Yongtong and colleagues, 1996</td>
<td></td>
</tr>
<tr>
<td><strong>An Agent-Based Conflict Management System for Co-operative Design</strong></td>
<td>An agent based conflict resolution system for Co-operative design</td>
</tr>
<tr>
<td><strong>Environment</strong> - by A. Gayretli and S. Kucukgokoglan, 2001</td>
<td></td>
</tr>
<tr>
<td><strong>Agent Based Negotiation for Collaborative Design Decision Making</strong></td>
<td>Agent based negotiation system based on existing negotiation research</td>
</tr>
<tr>
<td><strong>Multi-Agent Systems in Engineering Design of Fixture Systems</strong></td>
<td>An agent based approach for engineering design of fixture systems</td>
</tr>
<tr>
<td>– by Farhad Ameri and Joshua D. Summers, 2009</td>
<td></td>
</tr>
</tbody>
</table>
Multi Agent System based Approaches for 3D Environment Design

- **Procedural City Modelling** – by Thomas Lechner and colleagues, 2003

- **Simulation-Based Generation of 3D Urban Environments using a Multi Agent System** – by Reza Haddadi and Andrew Jönsson, 2009
Major Issues to be Addressed

- Current 3D environment generation techniques are specific to one or few types of environments such as city environments

- There are no customizable frameworks available which are common to many types of environments
Technology Adapted: Emergent Behaviours through Multi Agent Systems Technology

- The intelligence is emerged in an MAS (Multi Agent System) as a result of interactions between agents
- A complex outcome through simple rules
- Self organizing behaviour in modelling real world problems
- In MAS an agent can operate on incomplete information
Multi Agent Systems based Approach to Assist the Design of 3D Game Environments

- The natural world and man made world has emergent properties

- According to literature, in multi agent systems surprisingly complex and interesting global behaviours can arise from simple rules that are followed by number of simple agents operate in an environment
Multi Agent Systems based Approach to Assist the Design of 3D Game Environments (cont..)

- We hypothesize that this emergent behaviour of multi agent systems can be used to design 3D game environments with emergence properties that were not visible in initial constituents.

- This hypothesis is inspired by emergent behaviours of some natural systems arising from simple rules that are followed by individuals.
Multi Agent Systems based Approach to Assist the Design of 3D Game Environments (cont..)

- It is proposed to associate each 3D model in game world with a simple agent rule set.

- The agents in the multi agent system will use these simple rule set to place 3D models in the most appropriate places in 3D environment.
Multi Agent Systems based Approach to Assist the Design of 3D Game Environments
(cont..)

- **Input**
  - A parameterized description of an imaginary 3D environment
    - Size of the terrain (as a height map)
    - List of 3D models selected from ontology with number of instances required
    - Size and number of towns/ villages/ industries
    - Water level

User Provided Height Map
Multi Agent Systems based Approach to Assist the Design of 3D Game Environments (cont..)

- **Process**
  - Upon the request of desired game environment,
    - Relevant agents will be created
    - Come up with their 3D models and place them by negotiating among the agents
  - When self organizing, the agents will consider aspects such as
    - Type of the terrain which a given agent is located
    - Types of neighbouring agents
    - Sizes of neighbouring agents
    - Locations of neighbouring agents
    - Orientation of neighbouring agents and messages/requests received from other agents
Multi Agent Systems based Approach to Assist the Design of 3D Game Environments (cont..)

- **Output**
  - A 3D game environment with self organized 3D models located and oriented in most suitable places.
  - The final output will be rendered on a computer screen using an open source 3D graphics rendering engine.

- **Primary Users of the System**
  - *Level Designers* of 3D game projects
Design of the Multi Agent System to Assist 3D Game Environments Design

Input 1: Parameters Describing 3D Environment

3D Environment Definition Agent

3D Model Definitions

Environment Definitions

Common Message Space and Common Game Map Space

Environment updates / Agent messages

Environment states / Agent messages

Terrain Explorer Agents

3D Model Placing Agents

Road Network Development Agent

3D Model Definitions / Agent Rule Sets

Ontology

3D Model Definitions

Location, Scale and Rotation of 3D Models

3D Rendering Module

Output: 3D Environment on Computer Screen

Input 2: 3D Models, Agent Rule Sets

3D Rendering Module

Legion

Data Flow Module

Data Representation of Generated 3D Environment
## Implementation

<table>
<thead>
<tr>
<th>Module</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Environment Definition Agent</td>
<td>Java 1.6, Java Swing</td>
</tr>
<tr>
<td>Agent Definition Module</td>
<td>Java 1.6, Java Swing</td>
</tr>
<tr>
<td>Ontology</td>
<td>H2 Embedded Database, Java Classes, 3D Models</td>
</tr>
<tr>
<td>Common Message Space</td>
<td>Java 1.6 ArrayList</td>
</tr>
<tr>
<td>Multi Agent Systems</td>
<td>MASON Agent Development Software</td>
</tr>
<tr>
<td>Game Engine</td>
<td>Java Monkey Engine</td>
</tr>
<tr>
<td>User Feedback Capturing Module</td>
<td>Java 1.6, Java Swing</td>
</tr>
</tbody>
</table>
Graphical User Interface of Environment Definition Module
Implementation (cont...)

Graphical User Interface of 3D Model Selection Module
Implementation (cont…)

Introducing New a 3D Model
Implementation (cont...)

3D View 1
Implementation (cont...)
Evaluation

- The system has been evaluated by creating a prototype to test the proposed approach.

- Following aspects of the system were considered during the evaluation process.
  - Time taken to generate 3D environments
  - Customizability and Extendibility of the system
  - Adherence to industry standards
  - Portability
  - Cost Effectiveness
Evaluation (cont…)

Control Experiments
- It was decided to compare the prototype based on the approach proposed in this thesis with following approaches.
  - Using a **coding approach** to design the 3D game environment
  - Using a **world editor** to **manually design** the 3D game environment

Selection of Participants

<table>
<thead>
<tr>
<th>User</th>
<th>Experience with Java</th>
<th>Experience with 3D game environments</th>
</tr>
</thead>
<tbody>
<tr>
<td>User1</td>
<td>3+ years</td>
<td>4+ years</td>
</tr>
<tr>
<td>User2</td>
<td>2+ years</td>
<td>4+ years</td>
</tr>
<tr>
<td>User3</td>
<td>2+ years</td>
<td>3+ years</td>
</tr>
</tbody>
</table>
Evaluation (cont…)

Preferred 3D Game Environment

<table>
<thead>
<tr>
<th>3D Model</th>
<th>Number of Instances Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coconut tree</td>
<td>15</td>
</tr>
<tr>
<td>Lotus plant</td>
<td>10</td>
</tr>
<tr>
<td>Small house</td>
<td>15</td>
</tr>
<tr>
<td>2 Story house</td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td>50</td>
</tr>
</tbody>
</table>

Results

Average time taken to design preferred 3D game environment with coding approach = 215 min

Average time taken to design preferred 3D game environment with world editor approach = 28 min

Average time taken to design preferred 3D game environment with proposed approach = 5 min
Evaluation (cont…)

- Customizability and Extendibility of the System
- Adherence to Industry Standards
- Portability
- Cost Effectiveness

<table>
<thead>
<tr>
<th>Product</th>
<th>Licence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>GNU General Public License (GPL)</td>
</tr>
<tr>
<td>jMonkeyEngine game engine</td>
<td>BSD</td>
</tr>
</tbody>
</table>
| MASON agent development tool     | Mostly Academic Free License [Partial] Artistic License [Partial]  
|                                  | Sun Open Source License [Partial] BSD License                  |
| H2 embedded database             | Mozilla Public License Eclipse Public License                |
Conclusion

- Among the main issues in automated 3D environment design, following main issue were addressed.
  - Current 3D environment generation techniques being specific to one type of environment such as city environment
  - The lack of customizable frameworks which are common to many types of environments

- We postulate that this emergent behaviour of multi agent systems can be used to design 3D game environments with emergence properties that were not visible in initial elements.
Based on that hypothesis we have proposed an approach to develop a common framework which is common to many types of environments.

The proposed system has been designed with a multi-agent systems based approach and a prototype is implemented to evaluate the proposed approach.
Conclusion (cont…)

- The prototype has been evaluated by
  - Measuring the time taken to generate 3D environments
  - Extendibility of the system
  - Adherence to industry standards
  - Portability
  - Cost effectiveness
Future Work

- Integrating a software module to generate height maps based on user requirements

- Extending the prototype system to support more powerful and scalable open source graphics engine named OGRE (Object Oriented Graphics Rendering Engine) which is used by many game industry projects
Acknowledgment

We would like to acknowledge,

- The development teams of following software for the free distribution of software to use as development tools in this project
  - jMonkeyEngine
  - MASON
  - H2 database
  - Java
  - Eclipse IDE

- Authors of all the literature reviewed during the period of this project

- ICTer2012 conference organizing committee, reviewers and University of Colombo School of Computing (UCSC)
Appendix
3D Environment Definition Agent

- This agent is used to capture a brief user description of the 3D environment.

- The input of the system would be a parameterized description of an imaginary 3D environment and a height map.

A Sample Height Map
In addition, the user will specify the following details as input:

- Size of the terrain
- List of 3D models selected from ontology with the number of instances required
- Size and number of towns/villages/industries
- Water level

After capturing these inputs, the 3D Environment Definition Agent sends a request to Common Message Space to start the environment generation.
3D Model Definition Module

- *3D Model Definition Module* is used to introduce new 3D models (such as trees, rocks and houses) to the system and associate agent classes with newly introduced 3D models (for example tree agent class associated with a 3D model of a tree).

- When defining a 3D model, following parameters should be specified.
  - 3D model name
  - Location of 3D model
  - Category of 3D model (tree, rock etc)
  - Associated agent class (selected from ontology)
  - Default size
Ontology

- The ontology will act as the main medium to store the knowledge of agents.

- The ontology contains following:
  - 3D model binaries
  - 3D model definitions
  - Agent type definitions
  - Data representations of generated 3D environments
Ontology - Agent rule set definitions

- Generic Agent Rules Set
  - Generic Plant Agent Rules Set
    - Coconut Tree Agent Rules Set
  - Generic Rock Agent Rules Set
    - Lotus Plant Agent Rules Set
  - Generic Building Agent Rules Set
    - House Agent Rules Set
    - Shop Agent Rules Set
Common Message Space

- The Common Message Space acts as a bulletin board or a common white board for agents.

- Agents can publish common messages and requests on common message space.

- These messages or requests will be shown by all agents and relevant agents will respond to them.
Common Game Map Space

- The *Common Game Map Space* can be also considered as a common message space.

- The *Common Game Map Space* represents the current state of game environment.

- The state of *Common Game Map Space* changes as a result of the actions of agents.
Terrain Explorer Agents

- Explorer Agents seek for suitable locations based on the type of Terrain Explorer Agent.

- When evaluating the suitability of land, A Terrain Explorer Agent will consider following attributes.
  - Type of surface (flat / hills / water / sand / rock)
  - Size of the surface
  - Distribution of barriers on surface such as trees and rocks
  - Cost of cleaning up barriers
  - Available areas that can be cleaned up within a predefined cost
Example of a Hierarchy of Terrain Explorer Agents
3D Model Placing Agents

- 3D Model Placing Agents are responsible for placing 3D models in 3D game environment.

- Depending on the rules, 3D *Model Placing Agent* may consider all or some of the following things before marking a place for a 3D model.
  - The type of the terrain
  - Types of neighbouring 3D models
  - Sizes of neighbouring 3D models
  - Distance to roads
  - Distance to cities
  - Locations of neighbouring 3D models
  - Orientation of neighbouring 3D models and
  - Messages/ requests received from other agents
3D Model Placing Agents (cont...)

Example of a Hierarchy of 3D Model Placing Agents
Road Network Development Agent

- *Road Network Development Agent* is responsible for the development of road networks within the game environment.

- To generate roads, this agent searches for the city areas which are not connected to any road.

- Then this agent searches for the nearest city or road to the given city.

- After that it creates a new road from the given city to nearest city or road.
3D Rendering Module

- The Java Monkey Engine - free and open source game engine is used to render 3D output of the system

- The data representation of 3D environment generated will be retrieved from the ontology and processed using the game engine

- The final output is rendered on a computer screen with the ability to flythrough the created 3D environment
Thank You