GROUND COVER AND VEGETATION IN LEVEL EDITORS: LEARNING FROM ECOLOGICAL MODELLING

Eugene Ch'ng
School of Computing and Information Technology
The University of Wolverhampton, Wulfruna Street, WV1 1SB, Wolverhampton
email: e.chng@wlv.ac.uk

ABSTRACT

Ground cover and vegetation are important elements in large outdoor gaming environments. These include game scenes such as forest terrains, rural and urban environments. While the graphical representations of plants may look unbelievably real, a crucial factor related to their placements often lacks a natural association with those seen in the physical world. This article explores and pilots a study for simulating vegetation dispersal patterns and object positioning for game scene creation. The simple algorithms simulate growth patterns similar to nature and demonstrate the potentials of application in game level editors.

Keywords: Vegetation, ground cover, Scene Design

1. Introduction

Natural landscapes and large scale rural and urban environments have become a common scene in modern 3D computer game productions. While the look and feel of natural objects such as trees and rocks may create in players a sense of reality or presence (Minsky 1980) – the sense of being there, their functional cause are of much more importance to the survivability of the player. Trees and natural barriers could act as protection, camouflage, resources and even as weapons.

The impressive natural landscapes seen in successful First Person Shooters (FPS) such as Ubisoft's FarCry series and Electronic Art's Crysis are seminal in outdoor game scene creation. Such high quality environments warrant high requirements from the artists and designers that created the games. However, not all artists or designers are equal and it may be difficult to find designers with the botanical knowledge of 'what grows where'. There may be advanced developments in the behaviours of Non-Player Characters (NPC) and Artificial Life algorithms for bird flocks (Reynolds 1987; Lecky-Thompson 2008) inspired from nature, but gaps exist where such developments could be benefited by the inclusion of simple models inspired from the sciences such as ecological modelling (Gillman and Hails 1997). If algorithms can be adopted from this particular research area, creating vegetation covers for outdoor scenes would be made simpler. Artists need only 'block-in' a selection of vegetation onto the landscape without having to reference pictures of nature nor know the niche of plant types in differing global and local environments (Grime, Hodgson et al. 1988) such as tropics and seasonal plants, and their preferences for soil types, water logging, and altitudinal limits among the few. Scene designers need only create an environment conducive for plant growth using heightfield-based soil types and user interface sliders for setting climates and plant preferences. Current game level editors as we shall soon see, lacks these features. The integration of such features will certainly enrich the pre-production process of games.

This article describes the techniques used for piloting a study related to 'growing' vegetation inspired from a field of ecological modelling for game scene creation. Section two briefly explores the background related to this research. Section three describes the techniques and algorithms used for simulating simple vegetation growth. Section four demonstrates the simulation processes in different game scenarios and section five concludes the paper with plans for future research.

2. Background

WYSIWYG 3D FPS level editors first became useful when UnrealEd (UEd) accompanied Epic's Unreal game. The inclusion of a level editor greatly extends the lifespan of the game, allowing players to bridge the gap between inexpensive content generation tools such Autodesk's 3D Studio Max and the now outdated Gmax and a game MOD. Furthermore, the increasing requirements for realistic outdoor scenes initiated the integration of vegetation and object 'painting' tools within these editors (E.g., CryEngine Sandbox Editor I & II) so much that it has now become a mandatory feature in games engines (E.g., Torque 3D Game Engine, Unity3D Engine).

Vegetation and object placement tools differ among games engines. Some, like CryEngine's sandbox editor allows parameter tuning for density, random scale, swaying, plants and object models selection before they are 'painted' onto the landscape. Others like Torque 3D Game Engine's Foliage Replicator and Shape Replicator allows bitmap images of plants and 3D models to appear within a shape boundary (an ellipse) with settings for viewing distance, density, and dimensions. So far, although such features will extensively automate plant placements on a landscape, not one level editor incorporates ecological features for defining vegetation niche related to their adaptability on environments of variable conditions. Although proprietary terrain generators such as L3DT, Digital Element's WorldBuilder (McClure 2009) and 3D Nature's World Construction Set and Visual Nature Studio (Huber 2004) include certain features that allows automated placements of vegetation (terrain convexity/convavity and direction facing slope), more could be developed to extend the functionality to include the creation of different plant types and how each species could be associated with environmental factors that allow them to be distributed naturally across a terrain. L3DT possesses partial environmental attributes such as water flooding, water-table, salinity map, and attributes map that affects the texture of the terrain but do not generate plant to terrain positioning. The lack of exportable 3D coordinates of plants and objects in association with terrain size makes it difficult when large terrains are needed.

In order to create a natural looking terrain, scene designers frequently reference pictures of natural landscape when placing vegetations and terrain objects. This works for manageable sized terrain and outdoor scenes, but when large scale terrains and complex scenes are needed, their job becomes drastically tedious. Although, level editors have features that provide various parameters which designers can tune to set the scale, rotation, and position, tree types and density, but requires careful 'painting', or blocking in a community of forest canopy and undergrowth to get the effects right. If a feature could be included in level editors that allows vegetation to be 'grown' on any type of landscapes by simply tuning parameters associated with the climate and vegetation preferences, a designer's job would be made simpler. Lessons for simulating the natural distribution of vegetation could be obtained from a branch of ecological modelling related to Agent-Based Modelling (ABM) termed Individual Based Ecology (IBE) (Grimm 1999; Grimm and Railsback 2005). There are four criteria in which IBE is differentiated from other less individual-oriented modelling methods: (1) the degree to which the complexity of the individual's life cycle is reflected in the model; (2) whether or not the dynamics of resources used by individuals are explicitly represented; (3) whether real or integer numbers are used to represent the size of a population; and (4) the extent to which variability among individuals of the same age is considered (Uchmanski J and V. 1996; Grimm and Railsback 2005).

The next section explores a vegetation modelling approach (Ch'ng 2009) and seeks to simplify it for the pilot study.

3. Methodology

Game level design and ecological modelling are two very different fields, but lessons could be learned from the latter. The former sought believability and aesthetics while ecological modelling aims to capture a representation of the physical world within a computer model for understanding aspects of it and for proving theories. This section looks at a simplified model for growing vegetation and natural object placements in game level designers.

3.1 *A Simplified Model of Vegetation Growth*

Models in IBE are complex and use relevant environmental factors associated with the habitat of the biological community. These may include sunlight, temperature, altitude, carbon dioxide, and variations of soil types such as soil texture, acidity, depth, etc. Important factors for plant interaction such as competition for space, sunlight, and nutrients are also accounted for. A game design tool does not need such complexities but requires only factors that have high impact on plant growth. Environmental conditions may also be merged into an average. For example, the many different types of soil can be merged into an average condition using a single heightfield as reference.

A survey of plant ecology (Weaver and Clements 1938; Tivy 1993) shows that factors such as sunlight, temperature, moisture, and soil are the most influential factors affecting the growth of a plant. Plant competition on the other hand, is a deciding factor for the survivability of species in the environment and the patterns found on landscapes as suggested by studies in complexity science (Resnick 1994; Camazine, Deneubourg et al. 2001).

3.2 *Modelling the Environment*

In the environment, seasonal changes are not implemented. Time is continuous and is connected with growth, reproduction and competition. The environment has a tropical setting.

Factors	Units
Sunlight	0 < S < 1 (default=0.8), *v=5*
Temperature	0 < T < 1 (default=25 ° C), *v=0.2*
Moisture	0 < M < 1 (default=0.4), *v=0.2*

Table 1. Tropical Settings

The stochastic variations v of each factor is simulated for more interesting results. Soil conditions L are merged into a heightfield of value 0-255 normalised to the range $0.0 > L > 1.0$. Lower values (0.0~0.3) describe better soil conditions, and higher values (0.7~1.0) describe soils that are difficult to grow in.

3.3 *Plant Genotype*

Plant genotypes are defined using the relative measure (Ch'ng 2007). Each allele describes its capacity for adaptation to a certain environmental parameters and its growth and reproduction traits:

Description	**Lower**	**Ideal**	**Upper**
Sunlight	0.3	0.9	1.0
Temperature	-10	38	40
Tolerance to Crowd	0	0.2	0.3
Soil	0	0.4	0.6
Age, Growth and Reproduction			
Energy		1	
Seed Count		5	
Maximum Age		30	
Reproduction Age		3	

Table 2. Plant Genotype

3.4 *Plant Phenotype*

Plant phenotypes are graphical representation of the various stages of growth. Four simple representations are shown in Figure 1. These four stages are seedling, young plant, mature plant, old plant.

Figure 1. Representation of a species of plant in various stages of growth (he representation is a herb)

3.5 *Principles of Plant Growth*

Growth, reproduction, and plant interaction are described using algorithms. Here we define some simple rules for vegetation interaction.

```
FOR EACH PLANT:
  Fitness (f=1.0)
  Energy (e=1.0)
  Increment age every t seconds
  Compute fitness f
    Compute sunlight, temperature, soil
    Interact with surrounding plants
      If f=0, reduce energy loss l
      If e=0, remove plant
  If probability (p<0.5), reproduce offspring
    Reproduction based on fitness:
      Scatter number of seeds (seedCount*f)
        at distance (d) around this plant
END FOR EACH
```

Table 3. Algorithm for growth, reproduction, adaptation, and interaction

The fitness is the deciding factor for plant survivability. The fitness also decides the reproduction capacity of the plant; it contributes to the number of seeds that are reproduced defined in the Seed Count allele (Table 2) and is computed by multiplying it with the seed count: `seedCount * f`. The energy loss l is added for plant resilience to death, this by default is $l=0.1$. Each new seed carries the same genotype as the parent plant and is scattered at a distance surrounding it. Age is incremented by the seconds.

The rules of interaction between a plant and its environment is measured by Ch'ng's Adaptability Measure (*Am*) (Ch'ng 2007). The *Am* measures each environmental factor and generates the simple fitness (Eq. 1) in each system cycle:

$$f = cLUT \qquad (1)$$

Where c is the competition, L is the condition of the soil, U is the sunlight, and T is the temperature. Competition in particular occurs if the plant diameter intersects:

$$\sqrt{(O_x - u_x) + (O_y - u_y)} - [O_{size} + u_{size}] < 0 \qquad (2)$$

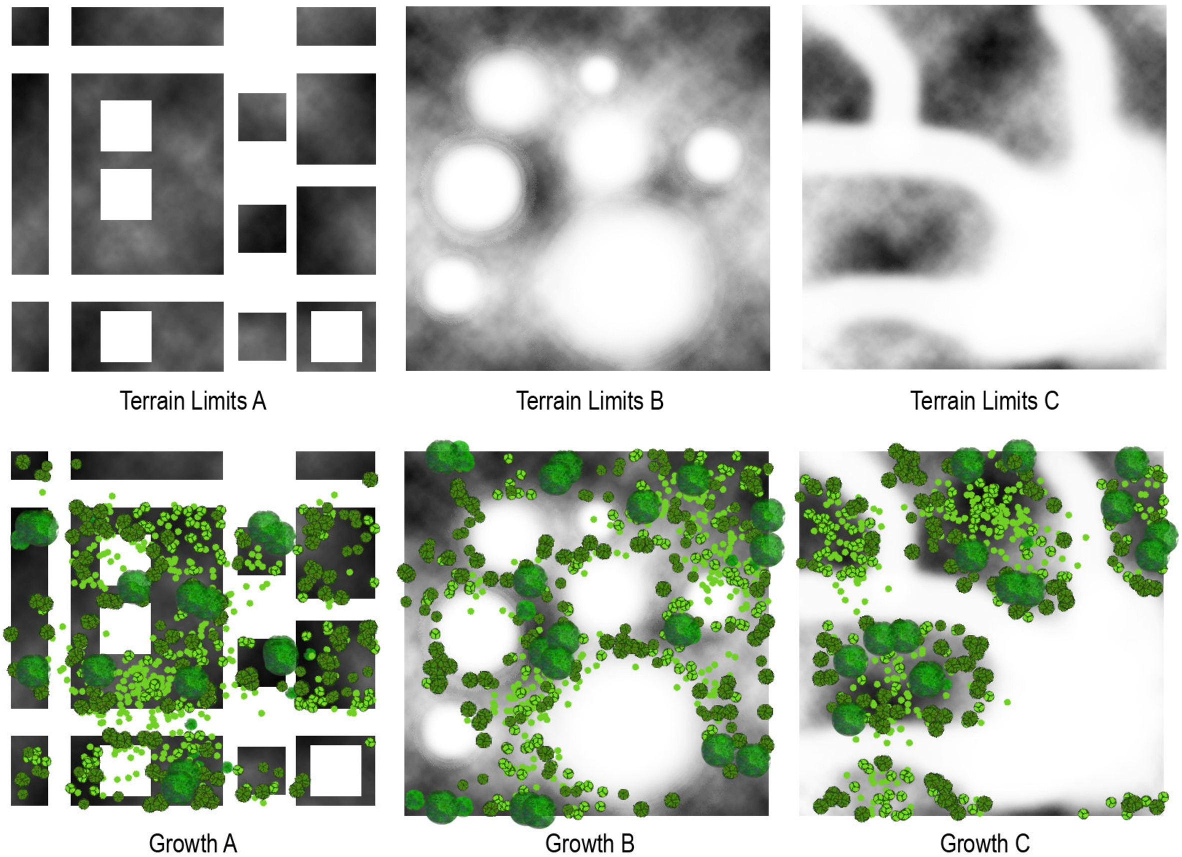

Figure 2. Terrain limits and growth scenarios

Where $O_{x,y}$ refers to the position of the opponent plant and $u_{x,y}$ refers to the position of the plant. O_{size} and u_{size} are respectively the diameter of the competing plants.

A neighbouring plant is a threat if $O_{size} > u_{size}$. The space used by an opponent plant is accumulated only if it is smaller (a herb) than the opponent (E.g., a tree). Each opponent plant contributes 0.1 to the accumulated space *c* and is limited within the range $0.0 < c < 1.0$.

3.6 *Plant Position Conversion: From 2D to 3D*

The position of the plants is within the screen coordinate systems. To convert it to the 3D coordinate system in DirectX, a formula is given below. Conversion to the OpenGL 3D coordinate system is accomplished by flipping the z axis:

$$x_{3d} = \frac{W_{terrain}}{W_{heightmap}} x_{screen} - W_{terrain} \qquad (3)$$

$$z_{3d} = \frac{-H_{terrain}}{H_{heightmap}} y_{screen} + H_{terrain} \qquad (4)$$

Where $W_{terrain}$ and $H_{terrain}$ are respectively the width and height of the 3D terrain, $W_{heightmap}$ and $H_{heightmap}$ are the width and height of the height map, x_{screen} and y_{screen} are the screen coordinate position of the plant, and x_{3d} and z_{3d} are the 3D coordinate position of the converted plant position.

For ease of export/import to game packages, the information associated with the generated plants in the landscape (plant type, position, and age) are output as XML file (see Table 4). For clarity, the structure shows only five plants. The 'Position' tag shows the converted axis.

```xml
</Plants>
  <Plant>
    <Type>Tree</Type>
    <Age>9</Age>
    <Position>34.3472819 52.2839478</Position>
  </Plant>
  <Plant>
    <Type>Tree</Type>
    <Age>15</Age>
    <Position>14.2039487 22.1829918</Position>
  </Plant>
  <Plant>
    <Type>Herb</Type>
    <Age>6</Age>
    <Position>54.2930029 32.1290039</Position>
  </Plant>
  <Plant>
    <Type>Herb</Type>
    <Age>4</Age>
    <Position>61.1920394 27.9938490</Position>
  </Plant>
  <Plant>
    <Type>Herb</Type>
    <Age>1</Age>
    <Position>71.7758392 67.6685930</Position>
  </Plant>
</Plants>
```

Table 4. Plant type, age, and 3D coordinate system position in the XML format

4. Simulation Scenarios

This section demonstrates the potential of the algorithms in three pilot scenarios. The scenarios show that the genotype of vegetation enables them to grow in different niche, mimicking the patterns in natural environments. Figure 2 shows the scenarios. The upper section of the image shows the terrain limits, an average condition of the soil type. The lower section shows the growth patterns that are based on the genotype defined in Table 4 and 5. There are two plant types – Tree and Herb.

Figure 2A is a ‘city block’ scenario, the white areas are limit regions and may represent hard ground (roads, pavements, buildings). The algorithm of the plants naturally prevents them from growing in these regions. The upper part of the terrain shows little plant growth because they have not yet spread to that region.

Figure 2B shows a village scenario. The circular white regions represent inhabitable grounds (from shallow grounds to hard surfaces). It is a noticeable phenomenon that herbs and bushes grow near to the hard surfaces. This was the functioning of the Adaptability Measure. The observed trees grow in deeper soils.

Figure 3B is a scenario where rivers lead into a pond. In this scenario, the gradient (white to black) represents soil humidity. Plants that are adaptable to the such condition grows well.

Description	Lower	Ideal	Upper
Sunlight	0.3	1.0	0.9
Temperature	-10	38	40
Tolerance to Crowd	0.0	0.4	0.6
Soil	0.0	0.2	0.3
Age, Growth and Reproduction			
Energy		1	
Seed Count		5	
Maximum Age		30	
Reproduction Age		3	

Table 4. Tree alleles for four environmental conditions

Description	Lower	Ideal	Upper
Sunlight	0.3	1.0	0.8
Temperature	-10	38	40
Tolerance to Crowd	0.0	0.7	0.9
Soil	0.0	0.6	0.8
Age, Growth and Reproduction			
Energy		1	
Seed Count		5	
Maximum Age		10	
Reproduction Age		2	

Table 5. Tree alleles for four environmental conditions

A culling factor was introduced into each scenario for controlling the population. This is useful as certain game scenarios require dense vegetation whereas other game types may require minimal ground cover. Although restriction applies, it is not strict; plant population may at times grow up to 30% of culling limit. Table 6 shows the culling limits:

Scenario	Population
A	400
B	300
C	500

Table 6. Culling limits for growth scenarios

5. Conclusion and Future Work

This article proposed a novel method for game level designers to efficiently grow natural looking vegetation cover on terrains by simply specifying plant genotypes and environmental limits. The approach, based on ecological modelling, allows designers to specify terrain limits and manipulate the genes of simple vegetation in order to create ground cover for game levels. Three pilot scenarios demonstrated that the algorithms could be integrated into game editors. It could also be developed as an external software package with the exportable XML output of plant types, age, and position. The article also provided a formula for converting from screen coordinates to 3D coordinate system. At present, the software framework is being developed for that purpose. Research is also being conducted to grow vegetation as part of the gameplay for real-time 3D games.

References

Camazine, S., J. L. Deneubourg, et al. (2001). Self-Organization in Biological Systems. Princeton, NJ., Princeton University Press.

Ch'ng, E. (2007). "Modelling the Adaptability of Biological Systems." The Open Cybernetics and Systemics Journal **1**: 13-20.

Ch'ng, E. (2009). An Artificial Life-Based Vegetation Modelling Approach for Biodiversity Research. Nature-Inspired informatics for Intelligent Applications and Knowledge Discovery: Implications in Business, Science and Engineering. . R. Chiong. Hershey, PA., IGI Global.

Gillman, M. and R. Hails (1997). Introduction to Ecological Modelling: Putting Theory into Practice (Ecological Methods and Concepts). Oxford, WileyBlackwell.

Grime, J. P., J. G. Hodgson, et al. (1988). Comparative Plant Ecology: A functional approach to common British species. London, Unwin Hyman Ltd.

Grimm, V. (1999). "Ten years of individual-based modeling in ecology: what have we learned and what could we learn in the future?" Ecological Modelling **56**(1999): 221-224.

Grimm, V. and S. F. Railsback (2005). Individual-based Modeling and Ecology. Princeton, New Jersey, Princeton University Press.

Huber, G. (2004). World Construction Set. Morrison, CO, 3D Nature. **2009**.

Lecky-Thompson, G. W. (2008). AI and Artificial Life in Video Games. Boston: MA, Charles River Media, Course Technology CENGAGE Learning.

McClure, D. (2009). Digital Element World Builder. Oakland, CA, Digital Element, Inc.

Minsky, M. (1980). "Telepresence." Omni **2**(9): 45-51.

Resnick, M. (1994). Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds. Cambridge, Massachusetts, MIT Press.

Reynolds, C. W. (1987). Flocks, Herds, and Schools: A Distributed Behavioral Model. Computer Graphics, Siggraph '87 Conference Proceedings. Anaheim, CA, USA. **21:** 25-34.

Tivy, J. (1993). Biogeography: A Study of Plants in the Ecosphere. Essex, UK, Longman Group Ltd.

Uchmanski J and G. V. (1996). "Individual-based modelling in ecology: what makes the difference?" Trends in Ecology & Evolution **11**: 437-441.

Weaver, J. E. and F. E. Clements (1938). Plant Ecology. New York, NY, McGraw-Hill Book Company, Inc.

BIOGRAPHY

Dr Eugene Ch'ng is currently a Senior Lecturer at the School of Computing and Information Technology, The University of Wolverhampton, UK. He holds a PhD in Electronic, Electrical and Computer Engineering at the University of Birmingham and was a Post-Doctoral Research Fellow at the same department. He has in the past worked on a number of interactive 3D simulation projects related to the reconstruction of terrestrial landscapes (Individual-Based Modelling of biotic and abiotic components), palaeoenvironments and heritage artefacts, the UK's defence arena related to anti-terrorists simulators and training systems, as well as various professional consultations in software, interactive digital media and web applications development. His research areas include Artificial Life, Complex Adaptive Systems, enhanced Virtual Environments (eVE) for Biodiversity Informatics related to marine and terrestrial environments, and Mixed-Reality Interfaces for eVE. The unique strength of Eugene's interdisciplinary field is the fusion of modelling and simulation with interactive 3D visualisation.