

Living Emerging Worlds for Games

Nicolas Jakob^(✉) and Carlos Andrés Peña

School of Business and Engineering Vaud (HEIG-VD), University of Applied Sciences
and Arts Western Switzerland (HES-SO), Yverdon, Switzerland
nicow.jakob@gmail.com

Abstract. In this paper we propose an agent-based, artificial-life inspired, model for living emergent worlds intended to provide realistic behaviors to, already graphically sophisticated, game environments. The ultimate goal of such a proposal is to create the impression to play in a real, living environment. The model is conceived as a physical world where several layers of geographic characteristics and sources of influences define the environment for the living entities. These former are agents that react, based on very simple rules, to the state of the environment and to the influences of other agents. The combined behavior of all these agents produces a dynamic environment that changes gently, but constantly in an unpredictable, although plausible, manner. The proof of the concept is established through the simulation of such a world performed on a fully-functional implementation of the model. We present an example of a simulation and discuss the encouraging results of this and other simulations. These results show that it is possible to create complex and stable ecosystems using simple sets of rules. However, it appears also clear that setting some of the parameters to obtain an acceptable behavior is a delicate task that deserves further exploration and development.

Keywords: Emerging systems · Dynamic game ecology · Artificial life

1 Introduction

In order to provide realistic environments, modern games rely mainly on graphic rendering rather than on nature-like ecosystems. Because of that, players face static virtual worlds that exhibit little, if any, changes as all is predefined by design. An increasingly used alternative is the one proposed by, so-called, sandbox games that provide players the ability to affect permanently the world. The question that arises, and that is addressed by the current work, is: why not simply, or in addition, enabling the world to change by itself? This paper focuses thus on techniques that model living emerging ecosystems as background environments for games aiming at providing players with dynamic environments — i.e., systems that change gently but constantly according to the normal development of their living components such as creatures or plants.

This paper is organized as follows. The remaining of this section discusses some lack of realism, from an ecological behavior point of view, observable in current landscapes or environments. Then, the next Section presents the main ideas of our proposal intended to improve realism through emerging living systems. Afterwards, it describes several features that should allow building living emerging worlds, enabling game developers to create ecosystems closer to reality than pre-generated worlds. The paper continues presenting and analysing the results of some experiments conducted on the prototypical implementation. Finally, the last section draws some conclusions and sketches possible ways to continue the work.

1.1 Static Environments in Games

In order to completely immerse players into a game, it is necessary to present a game environment as close as possible to reality. Usually, the expected realism is brought through carefully conceived graphic enhancements. However, they are rarely accompanied by the corresponding, realistic behavior of the environment (e.g., natural cycles of growing, life, and death). Some recent products start to timidly bring some realistic behaviour to the game environment, as is the case of *Far Cry* [1]. Usually, all the decisions about the entities present in a given environment are taken in advance and defined following some strict rules. During the development process, someone decides where these entities will be placed by manually designing logical clusters like forests, mountains, etc. causing a static distribution of landscape. A first attempt to palliate this limitation has been proposed in some games like *Minecraft* [7], where the designer, or an algorithm, predefines some *biomes*—i.e., logical areas that share some features and/or conditions—in order to automatically generate appropriate flora and fauna. Even if this technique is an important improvement for creating more realistic environments, it still exhibits some limitations as discussed below.

Biome shapes. Most of the time, biomes are defined as static areas, making it difficult to place entities that ensure smooth transitions between different kinds of biome. The consistency of neighbouring areas can also be an issue as it is unrealistic to suddenly jump, for instance, from an ice plain to a jungle.

Frozen ecosystems and persistence of changes. From a player’s perspective, it might be frustrating to walk in a region several times and never see some changes made either by themselves or simply occurring naturally. Furthermore, whenever a change is made, it is natural to expect it to become persistent in time. For instance, the creation of a new city or the destruction of a forest are noticeable events that should persist to reinforce the realism of a game.

Development process. Finally, it is difficult for developers to bring new game components without redesigning significantly, if not completely, the game environment so as to keep it consistent.

Note that some games, like *Embers of Caerus* [2, 3], are working on the development of a dynamic game ecology, a subject closely related to the work proposed herein.

2 Living Emerging Worlds in Brief

Keeping in mind the goal of offering realistic game environments from an ecological behavior point of view, the simplest manner to solve (some of) the main difficulties described previously is to use environmental models that are close enough to real living environments. Besides, rather than specifically defining how the environment is supposed to behave in every area, why not giving to every entity in the environment the ability to behave by itself according to its nature and to its interactions? These two considerations call for solutions inspired on Artificial Life. Note, however, that while pure Artificial-Life approaches tend to explore complex models and/or behaviors, our proposal intends to address an important issue in games respecting some constraints in terms of computational cost and maintainability. Our general idea turns, thus, around an agent-based environment — conceived as a background service for games — where the living behavior of each agent is driven by a simple set of independent rules. These rules define, for example, if a given area is suitable for a living component according to the current environmental state (See Fig. 1). All agents are, however, not necessarily living components; some of them may only have the purpose to bring essential resources for others. As a simple example, imagine a system composed by three kinds of agents: water sources, trees, and rabbits. There is a clear dependency between them because rabbits require vegetation to survive and trees require water to grow. It is possible to build a system that behaves according to these simple rules and let the system find how to organize these agents.

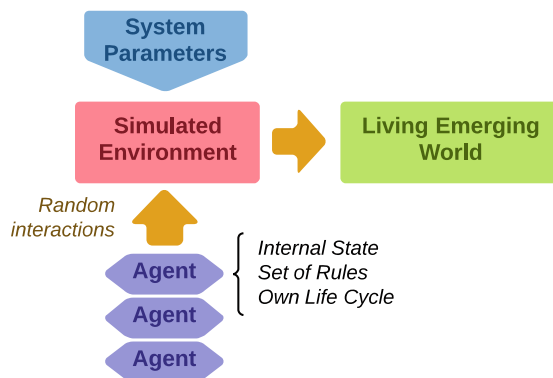


Fig. 1. General idea for emerging living worlds. The global environmental behavior is determined by the cumulated actions and interactions of individual agents. A simple set of rules define how such agents behave according to the influence of their environment.

Describing rule-based behaviors for individual agents is simpler than describing all the possible interactions by hand.

Unpredictability of content. In complex systems, locally-simple behaviours that make sense at an individual level, may bring unexpected results that a human mind might not be able to design manually or to predict precisely. In this context, the term emerging means that complex interactions between agents are brought by the execution of simple behavior blocks.

Introduction of new features. It is possible to easily introduce new rules into a system (i.e., a game) while it is in production without being obliged to take it out of service. It is, however, fundamental to monitor the impact of such changes on the system given that, as mentioned before, the result might be unpredictable.

3 Design Considerations

The proposed concept [5] turns around the interactions between living and environmental agents. So, the design of the sought-after living emerging worlds will depend on the features of both, the environment itself and the agents that populate it.

3.1 Environmental State

The environment, by definition, is a space that must support different kinds of living entities (species). It is expected to be diverse enough so as to contain different biomes occurring naturally without artificial boundaries between them and where living entities are able to live according to their characteristics and behavior rules. To attain this goal, the terrain is defined by means of a set of environmental parameters, determining its geography and its climate. The initial values of these parameters are obtained through procedural content-generation algorithms which are commonly used in games [4] — e.g. Perlin Noise [6]. They may change over time in order to affect the agents living in there. The environmental state is organized in layers that automatically shape areas where agents may appear (See Fig. 2). So, there is no need to define by hand the location of game objects in the simulation system.

3.2 Areas of Effect

The environmental parameters presented above do not permit to model completely all the possible sources of interaction. All the living agents may be able to affect and be affected by every other agent that is close enough. For instance, as illustrated by Fig. 3, a tree will provide vegetation, which can be used by other surrounding agents. Every agent may emit around it a number of influences that permit to create indirect dependency between agents. Indeed, agents react to the emitted influences according to their individual rules — e.g., needs,

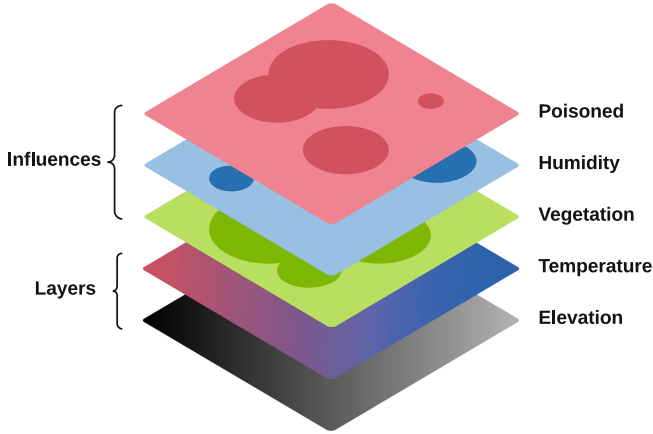


Fig. 2. The state of the world is organized in layers of environmental parameters and influences.

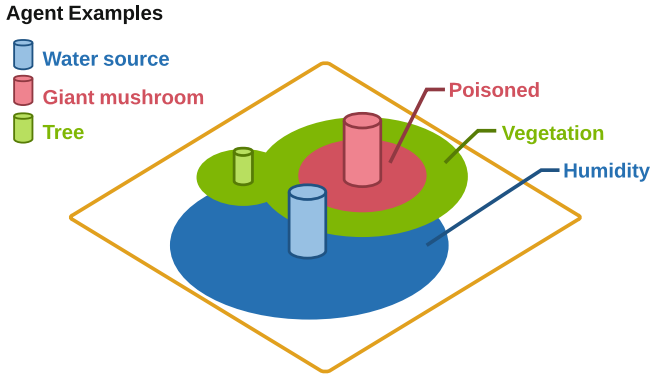


Fig. 3. Agents emit one or more types of influences around them, that may affect other nearby agents. This creates a network of interactions and dependencies among agents.

susceptibilities, weaknesses, or strengths. Moreover, as influences are attached to and managed by each agent, it is possible to use its internal state (age, health, etc.) to change over time the emitted effect. While a plant grows, its vegetation influence will expand too, for instance.

3.3 Live and Death of Agents

As explained before, each individual agent manages its own internal state by sensing its environment. Depending on the environmental state and the surrounding influences, an agent is able to define how suitable is the current area for it to live in. According to the agent’s rules, all these influences affect its internal status and may also trigger actions. As agents must not be eternal,

an age-related death mechanism is introduced: the probability of dying increases with time.

The question that arises at this point is: How are new agents generated to replace those who die? i.e., how do they born? Instead of implementing complex and costly reproduction strategies, we decided to implement a strategy that periodically checks the conditions of an area in order to generate new agents that are adapted to live there. The simplest solution is to randomly select an area, compute its environmental state, compare this state with the profile of every available species and, finally, select one or more new creatures.

3.4 Life Rhythm of Agents

Keeping in mind that the proposed world is intended to constitute a background environment for games, its simulation should be done in real time as players would be able to influence the environment. Continuously computing the internal state of a large number of living agents in the world is computationally costly and may prevent the expected real-time simulation. To palliate to this, we decided to rationalize the amount of status updates by defining an intrinsic rhythm of life for agents. According to the agent species, it is not necessary to update its internal state (e.g. size, color) at the same frequency because they do not have the same life rhythm. For example, an animal-like creature changes its state much more often than a tree.

4 Implementation and Experimental Results

In order to test the validity of the conceived living emerging world, we developed and implemented a simple, but fully-functional, prototype that allows simulating a relatively large area with many inhabitant agents [5]. The simulator allows the designer to define different species of agents and sources of influences, together with their corresponding set of rules. Note that the simulator is oriented towards managing the world rather than to its graphic rendering. Because of that it provides a simple command line interface to interact with it.

To illustrate the functioning of the prototype simulation environment, let it have a simple ecosystem inhabited by four different kinds of species, as described by Table 1.

The resulting population dynamics is shown by Fig. 4. It illustrates the final distribution of the populations in the world as well as the amount of time, in arbitrary units, required by the ecosystem to begin stabilizing after having started from an uninhabited world. However, in order to attain such stability, it is important that the underlying environmental state does not change too much during this transient phase. Note that, although four species are available (i.e., defined by the designer) only three of them populate the proposed world. This fact derives from the environmental conditions of the world which are not adequate for the cactus species to grow.

This example shows how a simple set of rules allows generating a world populated by different, but interdependent, species adapted to the reigning environmental conditions. As illustrated in Figs. 4 and 5 the agents are not distributed

Table 1. List of species and their characteristics used in the exemplar experimental simulation.

Name	Rhythm	Emissions	Requirements
Tree	Slow	High vegetation	Medium elevation High humidity High vegetation
Shrub	Medium	Medium vegetation	Medium humidity
Cactus	Medium	Medium vegetation	High temperature Low humidity
Rabbit	Fast	—	Medium elevation High vegetation

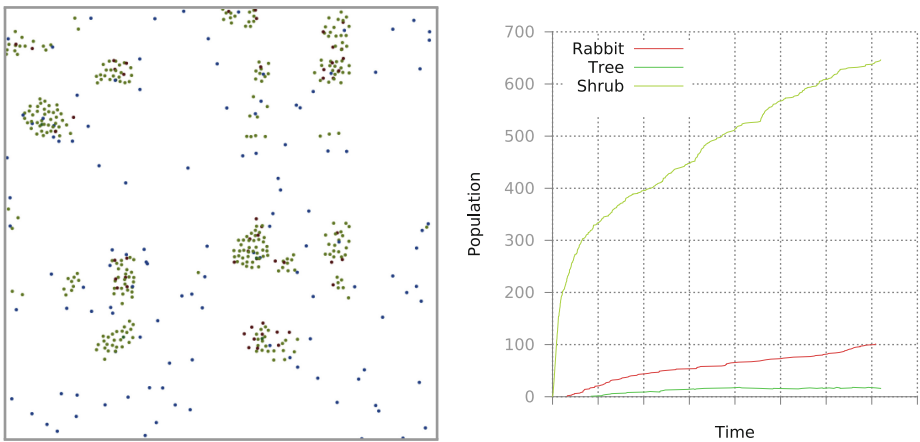


Fig. 4. Example simulation: Three species populate the world during the transient initial phase shown in the figure at the right. At the left, the distribution of the three populations at the end of this simulation period.

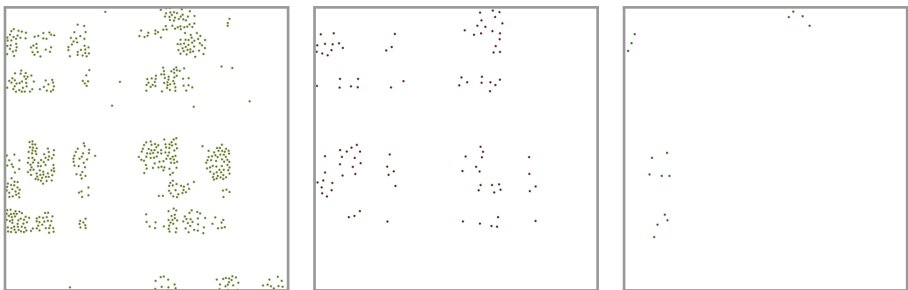


Fig. 5. Example simulation: Final distribution of the three populations across the world. From the left: shrubs, rabbits and trees.

arbitrarily through the world, but form small ecological communities. From the figures one can observe four of such niches formed mainly by shrubs and rabbits. Trees are much rarer and they are even absent in one of them.

On the base of this and several other simulations performed under different conditions, we have made the following observations:

Dynamics of the emerging ecosystems. Being a certain stability one of the pursued goals, it is normal that the emerging patterns of agents will generally remain the same until the local environment changes. Therefore, the rapidity at which an ecosystem changes is mainly defined by the rate of change of the environmental parameters. It appears also clear that the final effect of such changes — i.e., the new point of stability — depends greatly on the sensibility of the agents to these parameters.

Population management. We observed that managing the total amount of agents based only on the species' rules, without explicitly defining a limit, is challenging. In many cases the population of a species will simply explode (or implode) as there is no mechanism to reduce the attractiveness of an area when it is already saturated or close to be it. To deal with this problem, we considered two main possible solutions: (1) introducing a producer/consumer system to manage the available resources and (2) setting up a feedback mechanism into the process of agent creation, based on local population densities. Both of these alternatives would imply an additional source of complexity and were, thus, abandoned. Instead, we implemented much lighter variants of them: On the one side, attenuation of the influences as they are “consumed” by affected agents and, on the other side, reduction of the probability of creating new agents based on the distance of the closer existing agents.

Simulation benchmarking. The configuration of a new emerging world as well as the definition of rules for the different species imply conducting a large number of simulations to observe and tune the general behavior of the world. Several resulting aspects should be monitored, and for long simulation periods, in order to assess the adequacy of a given configuration — e.g., resulting populations, geographic distribution, computational cost related with the updating frequency. All this asks for the development of automatic test protocols with quality measurements as an outcome.

5 Concluding Remarks

The results of the performed simulations are encouraging enough as they tend to show that obtaining realistic game environments based on simple rule-based agents is feasible. These simulations have also allowed us identifying some concerns and issues that should be addressed in future developments of our work. The most relevant of them are discussed below.

Emergence versus design. While the approach of using emerging behaviors reinforces the expected non-deterministic realism of the world, it also makes it hard to force specific results. If one wants the system to move towards a precise type of results, a possibility is to introduce dependency between agents, for example in the form of influences.

Setting and adjusting environmental parameters. When configuring a model, many parameters must be set up and/or adjusted simultaneously. However, due to the strong interdependency of the agents among them and with the environment it is difficult to foresee the effects of such adjustments on the whole system behavior. The only manner available to analyze these effects and prevent unacceptable solutions, is to run multiple simulations under controlled conditions. This implies a high computational cost.

System stability and stagnation. Thanks to the live/death cycle of the entities, the whole population is periodically replaced, causing the renewal of the emitted influences, that is beneficial to escape from stagnating conditions. However, such system destabilization should also come from changes in the environmental parameters — e.g., temperature, humidity, elevation. Indeed, the fixity of these sources of influence have a strong effect on the tendency of the simulated world to converge towards the same results.

Integration in games. Knowing that the ultimate goal of our work is to integrate the model in games, it is crucial to define how the players will interact with the living emerging world. On the one side, the world is intended to be dynamic and reactive to create the impression of a living environment. On the other side, the players, according to game requirements, should not be allowed to destabilize excessively the system by (massively) killing creatures or destroying plants, affecting strongly the available influences in a given area. Such event may render entire areas unsuitable for one or more species. This issue should be carefully addressed to ensure that the delicate equilibrium between realism and security is kept.

References

1. Far Cry 3 Team. Open World, Biomes, Vegetation, and Fauna (2012). http://far-cry.ubi.com/fc-portal/en-au/community/detail_news.aspx?c=tcm:22--62197&ct=tcm:6--231-32
2. Forsaken Studios. Embers of Caerus Dev Blog: Dynamic Ecology (2012). <http://www.indiedb.com/games/embers-of-caerus/features/dev-blog-dynamic-ecology>
3. Forsaken Studios. Embers of Caerus: Dynamic Ecology (2012). <https://www.kickstarter.com/projects/forsakenstudios/embers-of-caerus-investor-prototype/posts/248961>
4. Hendriks, M., et al.: Procedural content generation for games. ACM Trans. Multimedia Comput. Commun. Appl. 9(1), 1–22 (2013). <http://dl.acm.org/citation.cfm?doid=2422956.2422957>, ISSN: 15516857

5. Jakob, N.: Living Emerging Worlds for Games. MA thesis. School of Management and Engineering Vaud, Yverdon-les-Bains, Switzerland, Feb 2014
6. Perlin, K.: Improved Noise Reference Implementation (2002). <http://mrl.nyu.edu/~perlin/noise/>
7. Persson, M.: Minecraft (2009). <https://minecraft.net/game>