SIMEARTH: A GREAT TOY

Fred Haslam
Vancouver, Washington, USA

Keywords: SimEarth, simulation, cellular automata, game, Wil Wright, James Lovelock, Gaia Hypothesis, continental drift, Jet Stream, trade winds, evolution, mutation, greenhouse effect

Contents

Summary

SimEarth is a game that simulates the major environmental factors on a global scale. It uses a multilayered cellular automata. The simulation covers geologic activity, weather systems, plant life, animal life, and the effects of civilization on planet Earth.

1. Introduction

SimEarth was first conceived by Wil Wright while he was reading James Lovelock's *Gaia Hypothesis*. This theory proposes that the living matter of a world modifies the planetary environment to suit its own needs. Wil began working on a planet simulator trying to incorporate all major aspects of environmental science into a single model. I came on board for the last year of the project helping out with the geologic, biologic, and sapient models.

Our target machine for the game was the Macintosh 512k. At the time, this was the pinnacle of personal computers. We were constantly running into memory limitations. Between our large maps, window displays, and code segments we had to make choices that frequently limited our representation. Our final map had a total of ten bytes of information per tile. At the time this seemed lavish. Most of our values were a subsection of a byte and so have a binary range such as 0 to 7 or 0 to 31.

Another limitation on the simulation was our desire to make the resulting application into a game. We had to consider what would be interesting for the player, and we had to give him the power to change the environment. Ironically, we sort of failed in our initial attempt to make SimEarth into a game. Players could frequently win without touching a key. Our attempt to mold the simulation into a game worked best with the models of Mars and Venus. Without player intervention, life would never form in these environments and thus player intervention was vital to success.

We had fun creating SimEarth. Wil and I had the opportunity to apply our theoretical knowledge of the physical sciences into a fascinating toy. I garnered quite a few insights into the ways that environmental systems in our world interact. It gave me the feeling that good old Gaia can survive a lot more punishment than Man has yet dished out. I hope that feeling is correct.

2. Overview

The model is broken down into five general systems areas over four developmental eras. The five general model systems are Lithosphere, Aquasphere, Atmosphere, Biosphere, and Civilization. Lithosphere is the "solid" model—mantle, core, and plate tectonics. Aquasphere is the "liquid" model—ocean levels, temperature, and motion. Atmosphere is the "gaseous" model—air composition, temperature, motion, cloud formation. The Biosphere is the "life" model—biomes, creatures, evolution. Civilization is the "sapient" model—development of technology, philosophy, and other intelligent endeavors. Player intervention is a sixth model that is beyond the scope of this article.

The game eras are Geologic, Evolutionary, Civilized, and Technological. The geologic era covers the formation of oceans and continents and the evolution of early biology to the point that creatures crawl onto land. The evolutionary era concentrates on the alteration and evolution of species to the point that tool-using creatures invent fire. The civilized era concentrates on the development of an intelligent species up to the point that they figure out how to utilize nonrenewable sources of energy. The technological era is identical to the civilization era except that the prime species is using fossil fuels

and atomic fuels. The game can advance forward through the eras or go backward depending on random events and player interaction.

There is a time limit built into the simulation. Over the course of time solar temperature increases and planetary core temperature decreases. After 10 billion years game time, the sun has expanded to the point that it engulfs your world, ending the simulation.

2.1. Modeling Methods

The basic model in this game is a state-based cellular automata. Cells maintain information on all five systems mentioned above. Our cells are organized into a number of two-dimensional arrays collectively called "the map." Generally speaking, cells are only affected by themselves and the eight adjacent cells—although there are exceptions. There are also a number of global values. These values record systemic state changes (such as the current era), summarized values (such as biomass or zoomass), and cumulative values (such as fossil fuels or nitrogen levels). Finally there are the model variables. These values adjust the behavior of the diverse models. Because I will not be discussing player interaction, I will be ignoring the model variables.

Each cell has 10 bytes of information. Here is a list of the values each tile contains: terrain altitude, magma drift direction, magma drift speed, ocean existence bit, ocean temperature, ocean motion direction, ocean motion speed, air temperature, air motion direction, air motion speed, air cloud density, random events, biomes, creatures, sapient objects, and a city preclusion bit. We think of these values as existing in "layers." When I refer to the "air temperature layer," I mean the cell value for air temperature across the entire map.

The basic layer is a rectangular array of 128 horizontal tiles by 64 vertical tiles. Many of the games layers are half this size: 64 horizontal by 32 vertical. These smaller layers have "fat" cells. One fat cell will correspond to four "tiny" cells from the basic layers. For example, an animal in cell (10,10) would look into the fat weather cell (5,5) when looking for air temperature. A tiny cell with coordinates (X,Y) would correspond to a fat cell at (X/2,Y/2). The fat cell with coordinates (H,V) would correspond to the four tiny cells at (H×2,V×2), (H×2+1,V×2), (H×2,V×2+1), and (H×2+1,V×2+1).

Tiles on the right side wrap around to the left. Tiles on the top will "offset" wrap to another top tile halfway around the map. The same applies at the bottom. We did not adjust the representation of the tiles to account for the deformation of a sphere. Our only concession to the need for a spherical representation is to adjust the values of the model so that systems that occur near the "poles" are generally correct. For example, since arctic biomes usually appear at the poles, we adjust their effect on global temperature and water storage.

2.2. Responsive Updates

Most cellular automota update all the values of all their cells simultaneously. For our model this was too much work. What I mean is that updating everything took up so much processing time that the player found the game to be jerky and unresponsive. We

use two techniques to make the game more interactive. The first is layered updating. The second is transfer map updating.

Layered updating simply means that we update the simulation one layer at a time. During the evolutionary era we first update the ocean temperature, then the ocean motion, then the air temperature, then the air currents, then animal life, and so on. Each layer update is performed completely before beginning another layer. This method also allows us to ration the simulation time between the layers. If one era requires more geologic simulation, we simply call the magma and altitude layer simulations more frequently.

Transfer map updating means that we copied our new cell values into a temporary map. Upon completion of a layer simulation we copy the new values back into the original map. This is a standard technique for cellular automota—it is used in the game of Life. When cells depend on adjacent cells for their new value, the mechanism of program looping can alter the results. By placing the new values into a temporary map we more effectively simulate simultaneous updating. For SimEarth this method has another benefit. We can update part of a layer, respond to the player, then continue updating the layer.

2.3. Approximations

Many of the numbers we use to create the simulator can only be described as "proximate." We frequently create value ranges that merely look correct. As commercial game designers it was more important for us to create an environment that felt right than to create a simulation that was accurate. For example, we use an air temperature range of eight bits equal to 0 to 255. When we created the model we were more concerned with how the temperature map looked than with what each number represented. After we finished balancing the model we went back and created functions that interpret the temperature value into something that looks more correct: a value range of from –50f to +150f.

3. Lithosphere Model

The lithosphere simulates the motion of magma and the collision of continental plates. The core model is linked to the planet's age. Magma motion is driven by earthquake events. Plate formation is the result of volcano events. Terrain altitude is the result of magma motion driving plates into collision. The terrain altitude is five bits equal to a range of 0 to 31. Values of more than 3 are considered to be tectonic plate material. The magma drift uses three bits for directions 0 to 7 (north, northeast, east, southeast, and so on), and four bits for speed equal to the range 0 to 15. A speed of 0 indicates that the magma is motionless.

It is possible to examine the planet's core in SimEarth. You will see the various layers—magma, mantle, core. This was included mostly as an educational tool. The thickness of these layers varies over time, but only in direct correlation to the planet's age. We found numbers indicating current magma layering and an estimate on values from several billion years ago. We did a simple linear extrapolation to find mantle

thickness over time. The age of the core is expressed by the global value core heat.

3.1. Driving Events

During the geologic era the drift model is applied every 30 million years. Every time the drift model is called, the collision rules are applied and a "massive upwelling" event occurs. We also called this event the Nemesis Effect. This upwelling is the driving event for the magma model. When the upwelling hits a thick section of crust (terrain > 1), it becomes an earthquake. When it hits a thin section of crust it becomes a volcano. The power of these events is inversely proportional to the core heat of the planet.

The earthquake event alters the magma flow layer. The magma flow is altered over a circular region. All tiles in this area are set to a randomly selected direction and speed value. Magma flows are "smoothed" over time. Each time the magma motion model is called, some number of randomly selected cells will change their values to copy an adjacent cell. Over time this gives the appearance of fluid turbulence and helps to generate interesting terrain maps.

The volcano event randomly adds to the terrain level terrain over a circular region. This addition to altitude creates the crustal material that forms our tectonic plates. High terrain levels (terrain>3) are considered to be tectonic plates that are subject to collision rules. Each time the model is called, tiles with an appropriate speed will be pushed onto adjacent spaces. Tiles with high magma speed are moved more frequently.

3.2. Collision and Smoothing

When the terrain from a moving cell collides with terrain in a slower cell, collision rules are applied. First, if the target cell altitude is below plate level, then the moving cell value overrides the target cell, or else apply the second rule. Second, if the moving cell altitude is below plate level, then the moving cell is overridden by the target cell, or else apply the third rule. Third, if the speed of the moving cell exceeds the sum of the moving and target cell altitudes, then the altitudes are summed and placed in the target cell, or else no movement occurs. This third rule implies that mountain ranges will appear when tectonic plates are pushed together by fast magma flows.

The drift model will move all the tiles without repeating or skipping. When operating a two-dimensional cellular model, we scan the cells from left to right, top to bottom. In this case we have to be sure that the tile we scan has not already been moved AND that collision occurs on the wave front of a magma flow. We need three programming tricks to ensure this. First, tiles are moved onto a transfer map. Second, when we try to move a tile, we examine the cell it is entering and try to move that cell first. Moving the target cell first leads to recursion which necessitates our third trick: we mark the cells with a bit to indicate whether they have already been processed during this simulation cycle.

The final part of the geologic model is erosion. When eroding a cell, we begin by creating a target value which is the average of all the adjacent cell altitudes. If the current cell is more than one level higher than this average, we lower the current altitude by one. If the current cell is more than one level lower than this average, we raise the

altitude by one. This results in an extremely gentle smoothing algorithm that halts when slopes reached a 1:1 rise/decline.

One tricky question of smoothing is how to handle altitude level zero. Zero level terrain is usually created by magma flows pulling terrain away from a tile. During development, "cracks" would appear in the map and did not erode away fast enough for our tastes. We created a special smoothing algorithm for the case of altitude level zero. If a crack appears in the middle of a tectonic plate, we replace the altitude with half the average of the adjacent tiles. This represents plates stretching. If the crack appears in the middle of an ocean, we assume that the crack is an oceanic trench. For this case we generate a small random addition to the altitude to represent magma upwelling.

4. Aquasphere Model

The water model concentrates on the effect of oceans. Lakes and rivers are ignored. Ocean volume and position are determined by the lithosphere model. Oceanic heat comes from the atmosphere. Oceanic heat releases into the atmosphere and generates atmospheric moisture. We originally intended to move ocean temperature based on ocean currents, but we never managed to balance that part of the simulation, and finally discarded it.
Ocean temperature is a byte value equal to 0 to 255. Sea level is calculated from the terrain and a DeltaSeaLevel constant. Terrain altitude below sea level is marked as ocean. Ocean motion is stored by four bits representing nine values; the eight grid directions and one "not-moving" value.

The timeframe for the ocean simulation is radically unbalanced. During the evolution era, ocean activity is simulated once every half a million years. During the Civilization model it gets simulated every 150 years. These time choices were a balance of how much time we were trying to represent versus our limits of processing power.

4.1. Ocean Regions

Ocean volume is approximated by a single constant called DeltaSeaLevel. This value is added to the average land altitude to create the Sea Level. The size of the ice packs can modify the Sea Level by at most +1 or −1. The Sea Level value is compared to the land altitude in each cell to determine if the cell is covered in water.

During development, we had to carefully select our DeltaSeaLevel constants. A value of zero will generally mean that 50% of the surface is ocean. Our geologic model will generally create a map with an average altitude of from 12 to 16 and a deviation of about 6. We selected a DeltaSeaLevel of 5 for Earth. This gave us the 70% water to 30% land ratio we desired. For random worlds DeltaSeaLevel was a random range of 0 to 6. For the Aquarium scenario, DeltaSeaLevel was an 8.

Two events can modify DeltaSeaLevel. The Boil Off event will occur when the average air temperature exceeds the value of 240 (255 is the maximum). In this case, the DeltaSeaLevel is decremented and atmospheric moisture is set to maximum—creating clouds whose albedo will cool the planet. The Ice Meteor event is an extremely rare

random event that increments our DeltaSeaLevel.

Ice Packs modify the Sea Level by from +1 to –1. In the biology model, arctic biomes develop near the poles. Ice Pack is a count of the number of arctic biomes. High Ice Pack levels (more than 12% of the surface) will decrease the Sea Level. This represents water from the oceans being bound into the Ice Packs. Low Ice Pack levels (less than 3%) will do the opposite. Increased oceans mean decreased land, which means lower CO_2 production. The decrease in a hothouse gas will lower global temperature. This means that, barring other considerations, Ice Pack growth is a self-regulating system.

4.2. Ocean Temperature

During the first cycle of the geologic model we start with no visible water. As oceans develop, uncovering or covering new cells, we take their initial cell temperature directly from the atmosphere model. After this initial value, ocean heat slowly follows changes in atmospheric temperature. Ocean temperature acts as a brake to the mercurial changes in air temperature.

In every cycle a small amount of heat is added to the ocean cell from the air cell. We add four if the air is warmer, or subtract four if the air is cooler. Please note that we DON'T remove any temperature from the air cell. By ignoring conservation of heat we managed to decrease our processing overhead. If we modified atmospheric heat at the same time as ocean heat we would be updating two layers simultaneously. This way, we only update one layer at a time.

Temperature is returned to the air during the atmosphere model. When we add heat back to the air, we do so by adding back half the difference: air temp = [air temp + ocean temp] / 2. This means that water-to-air transfer can be as large as 127, although the transfer is generally fairly low, 20 or less. This discrepancy in transfer rates, water-to-air being larger than air-to-water, was deliberate. We did this to simulate the ocean's greater capacity for storing heat.

4.3. Ocean Motion

Ocean currents are determined by sea depth, planetary spin, and the doldrums. We designed the currents to follow the shoreline. The doldrums are a region near the equator that has no movement. We create the doldrums by pretending that the ocean depths near the equator are actually shoreline.

Ocean cells look for the deepest adjacent cell. If the current cell is the deepest tile, then no motion occurs. Direction of motion is tangential to the direction of the deepest tile. Planetary spin determines the tangent. In the southern hemisphere the direction is rotated ninety degrees in the clockwise direction. In the northern hemisphere the opposite occurs. Between planetary spin and the doldrums, we see circular motion around the ocean basins, rotating clockwise in the northern hemisphere.

As I said at the start we do not use the ocean currents to move the ocean temperatures. In fact, ocean currents are not really used anywhere in the simulation. The sole

exception is in the geologic era, where we use the ocean currents as an approximation for air currents. This saves on processing time and gives reasonable results.

5. Atmosphere Model

The atmosphere model calculates the effects of weather and atmospheric composition. Air composition derives from the effects of land, ocean, and biology. Air temperature comes from the sun and the ocean. Clouds and arctic biomes reduce solar input, cooling the planet. Air currents derive from air temperature and move both heat and clouds. Air temperature is an eight-bit value equal to from 0 to 255. Air moisture is an eight-bit value equal to from 0 to 255. Air direction is four bits representing nine values: the eight grid directions and one "not-moving value." The atmosphere components are long integers with a range of zero to four billion.

I would like to note that air temperature is probably the most important model in this application. James Lovelock's Daisy World Model used the albedo of daisies covering a world to alter the surface temperature. This model clearly demonstrated the power of life to regulate a planet's environment. In our model, arctic biomes and dense clouds reflect sunlight with their high albedo. Air temperature drives air motion, feeds into oceanic temperature, and limits the survival of land-based biomes, animals, and sapients. The air model, and in particular the air temperature model, received a great deal of consideration from Wil Wright.

-
-
-

Bibliography

Bremer M. (1990). *SimEarth the Living Planet: User Manual*. Maxis.

Lovelock J. (1988). *The Ages of Gaia*. New York: Norton.

Biographical Sketch

Fred Haslam, after ten years of study at three universities, during a period of time when computer science majors were still being formulated, gave up on his dream of collegiate degrees to become a professional programmer. His two most prominent works to date are *SimEarth: The Living Planet*, and *SimCity 2000*. Mr. Haslam now runs an online game site which supports his experiments in artificial intelligence.