ICAT '99

THE VIRTUAL REALITY SOCIETY OF JAPAN

LIFE SPACIES II: from text to form on the Internet using language as genetic code

Christa SOMMERER, Laurent MIGNONNEAU, Roberto LOPEZ-GULLIVER

ATR Media Integration and Communications Research Lab,
2-2 Hikaridai, Seika-cho, Soraku-gun, 61902 Kyoto, JAPAN
christa@mic.atr.co.jp

Abstract

We are artists working on the creation of interactive computer systems that use the audiences' participation as essential input for the creation of image structures within the systems [1]. In 1999 we created an interactive web sites for the Internet called "Life Spacies II". This web sites allow users on the Internet to write text messages that are instantly translated into visual three-dimensional forms. Our custom designed text-to-form editor takes the letters, syntax and sequencing of a text message as genetic code and translates them into three-dimensional forms. As the text messages of the different users on the Internet are usually unique and diverse, unique and personal three-dimensional forms can be created by the users.

Key words:

1. Introduction

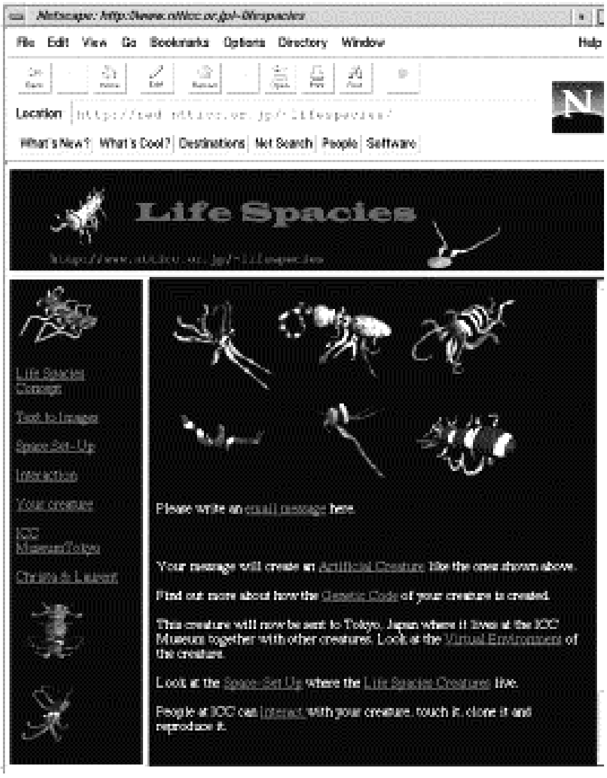
"colorless green ideas sleep furiously"
(Noam Chomsky)

According to Noam Chomsky, human language acquisition is based on a universal grammar that is genetically embedded within the human mind of all normal children, allowing them to learn their native languages naturally and seemingly effortlessly. [2] It was also Chomsky who coined the above phrase of "colorless green ideas sleep furiously," an expression that might not make much logical sense to a more scientifically oriented person, but does have quite a lot of meaning for a more visual or art minded person. Though this sentence, as Chomsky has shown, is grammatically correct, its meaning cannot be grasped through logic alone. When we hear this sentence for the first time we see pictures or forms or shapes appearing in our minds. These forms are vague, yet they are defined to a certain degree and can certainly create visual sensations and emotions. Inspired by Chomsky's sentence and based on the idea of translating words or sentences into visual forms, we created an interactive systems for the Internet, called "Life Spacies II".

This system consists of an Internet web page, that allows users to write text messages into a text window.

Through our specifically designed text-to-form editor we are able to translate these text messages into three-dimensional visual forms and shapes. The following chapters will describe the text-to-form editor in more detail.

2. LIFE SPACIES - from text to form on the Internet



Please write an email message here.

Your message will create an Artificial Creature like the ones shown above.

Find out more about how the Genetic Code of your creature is created.

This creature will now be sent to Tokyo, Japan where it lives at the ICC Museum together with other creatures. Look at the Virtual Environment of the creature.

Look at the Space-Set Up where the Life Spacies Creatures live.

People at ICC can interact with your creature, touch it, clone it and reproduce it.

Fig. 1 "Life Spacies" web site

The "Life Spacies II" web page (Fig. 1) allows people throughout the world to interact with the system. By simply typing and sending an email message to the "Life SpaciesII" web site (http://www.ntticc.or.jp/~lifespacies), one can create an artificial three-dimensional creature. As soon as the message is sent it creates a creature that starts to live in the interaction environment at the ICC Museum in Tokyo [3].

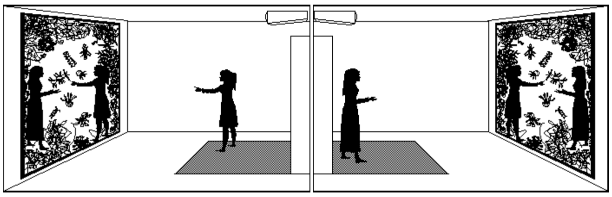


Fig. 2 “Life Spacies” Interaction Setup at the ICC Tokyo

The interaction setup consists of two independent interaction sites (Fig. 2) that are linked together via a data line, allowing visitors at remote locations to be displayed and interact in the same virtual three-dimensional space.
On-site visitors can directly interact with the creatures through touching and catching them. Once a creature is caught by the visitor, it will clone itself. However, if two remotely located people are in the same virtual space, they can each catch a creature with their hands, which causes these two creatures to mate and to create an offspring by genetically exchanging the parents’ codes.

2.1. "Life Spacies’s" text-to-form editor

To create the creatures, we developed a special text-to-form editor that enables us to translate the written text of the text messages into the genetic code of a creature. The text-to-form editor is based on the idea of linking the characters and syntax of a text to specific parameters in the creature’s design.

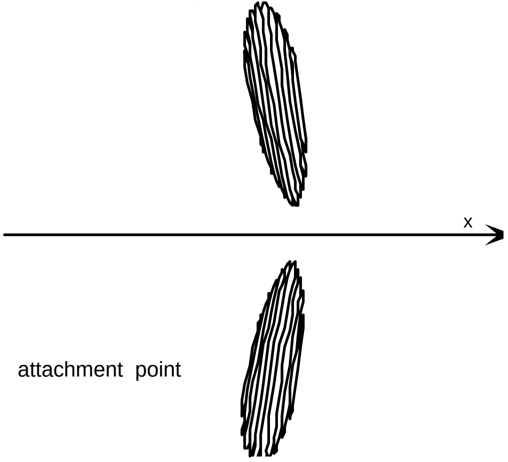


Fig. 3 Creature with two bodies and one pair of limbs

The default form of a creature is a body composed of a sphere consisting with 100 vertices, that is, 10 rings with 10 vertices each. All vertices can be modified in x, y and z axes to stretch the sphere and create new body forms.
Several bodies can be attached to each other provided that their attachment point is located on the x-axis. If the attachment point is not on the x axis, a limb is created instead of a body; this limb is copied and the copy is attached at a position symmetric to the original position. Figure 3 show a creature with two spheres as bodies and one pair of limbs.
According to the sequencing of the characters in the text, the parameters of x, y and z for each of the 100 vertices can be stretched and scaled, the color values and texture values for each body and limb can be modified, the number of bodies and limbs can be changed and new locations for attachment points of bodies and limbs can be created. Since each of the vertex parameters is changeable and all of the bodies and limbs can be changed as well, about 50 different functions for the creature’s design parameters are available. The design functions are subsumed in a design function table (Fig. 4).

function1 stretch default body/limbs in x
function2 stretch default body/limbs in y
function3 stretch default body/limbs in z
function4 set the next stretch function to global
function5 set the next stretch function to a vertex point
function6 set the next stretch function to a ring
function7 create a new location for an attachment point
function8 copy a new location for an attachment point
function9 compose a new texture for body/limbs
function10 copy texture of body/limbs
function11 change parameters of RED in body/limbs texture
function12 change parameters of GREEN in body/limbs texture
function13 change parameters of BLUE in body/limbs texture
function14 change patterns of body/limbs texture
function15 exchange positions of bodies/limbs
function16 copy body/limbs
function17 create a new body/limbs
function18 add or replace some of the above functions
function19 randomize the next parameters
function20 copy parts of the previous operation
function21 modify life span (default is 24 hours)
function22 add the new parameter to previous parameter
function23 ignore the current parameter
function24 ignore the next parameter
function25 replace the previous parameter by new parameter
....
function50

Fig. 4 “Life Spacies” design function table

Next, in translating the characters of the text message into these design function values, we first assign an ASCII value to each character. This is done according to the standard ASCII table shown in Figure 5.

33 !   34 "   35 #   36 $ 37 %   38 &   39 '
40 (41)   42 *   43 +   44 ,   45 -   46 .   47 /
48 0   49 1   50 2   51 3   52 4   53 5   54 6   55 7
56 8   57 9   58 :   59 ;   60 <   61 =   62 >   63 ?64 @ 65
A   66 B   67 C   68 D   69 E   70 F   71 G
72 H   73 I   74 J   75 K   76 L   77 M   78 N   79 O
80 P   81 Q   82 R   83 S   84 T   85 U   86 V   87 W
88 X   89 Y   90 Z   91 [92 \   93]   94 ^   95 _
96 `   97 a   98 b   99 c   100 d   101 e   102 f   103 g
104 h   105 i   106 j   107 k   108 l   109 m 110 n   111 o
112 p   113 q   114 r   115 s   116 t   117 u   118 v   119 w
120 x   121 y   122 z   123 {   124 | 125 }   126 ~

Fig. 5 ASCII table

Each character refers to an integer. We can now proceed by assigning this value to a random seed function *rseed*. In our text example from Figure 4, *T* of *This* has the ASCII value 84, hence the assigned random seed function for *T* becomes *rseed(84).* This random seed function now defines an infinite sequence of linearly distributed random numbers with a floating point precision of 4 bytes (float values are between 0.0 and 1.0). These random numbers for the first character of the word *This* will become the actual values for the modification parameters in the design function table. Note that the random number we use is a so-called "pseudo random," generated by an algorithm with 48-bit precision, meaning that if the same *rseed* is called once more, the same sequence of linearly distributed random numbers will be called. Which of the design functions in the design function table are actually updated is determined by the following characters of the text, i.e., *his*; we then assign their ASCII values (104 for *h*, 105 for *i*, 115 for *s* ...), which again provide us with random seed functions *rseed(104*), *rseed(105*), *rseed(115)*. These random seed functions are then used to update and modify the corresponding design functions in the design function look-up table, between design function1 and function50. For example, by multiplying the first random number of *rseed(104*) by 10, we get the integer that assigns the amount of functions that will be updated. Which of the 50 functions are precisely updated is decided by the following random numbers of *rseed(104)* (as there are 50 different functions available, the following floats are multiplied by 50 to create integers). Figure 5 shows in detail how the entire assignment of random numbers to design functions operates. As mentioned above, the actual float values for the update parameters come from the random seed function of the first character of the word, *rseed(84)*. An example of the entire procedure is given in Figure 6.

Example word: *This*

T => *rseed(84)* => {0.36784, 0.553688, 0.100701,...}
(actual values for the update parameters)

h => *rseed(104*) => {0.52244, 0.67612, 0.90101,...}
#0.52244 * 10 => get integer 5 => 5 different functions are called within design function table

#0.67612 * 50 => get integer 33 => function 33 within design function table will be updated by value 0.36784 from 1. value of *rseed(84)*

#0.90101 * 50 => get integer 45 => function 45 within design function table will be updated by value 0.553688 from 2. value *rseed(84)*
.... until 5. value

Fig. 6 Example of assignment between random functions and design functions

As explained earlier, the basic "module" is a sphere with white default color and no texture. When messages are sent, the incoming text modifies and "sculpts" this default module by changing its form, size, color, texture, number of bodies/limbs, copying parts and so forth. Depending on the complexity of the text, the body and limbs of the creature become increasingly shaped, modulated and varied. As there is usually great variation among the texts sent by different people, the creatures themselves also vary greatly in appearance, thus providing a personal creature for each author. Figure 7 shows an example of a short and simple email message sent to the "Life Spacies" web site.

Date: Sun, 01 Nov 1998 13:14:32 +0900
From: Christa Sommerer <christa@mic.atr.co.jp>
To: life@lc.ntticc.or.jp
CC: christa@mic.atr.co.jp
Subject: test creature1

This is a test creature.

Fig. 7 Example of email message to "Life Spacies"

2.2. Picture of the "Life Spacies" Creature

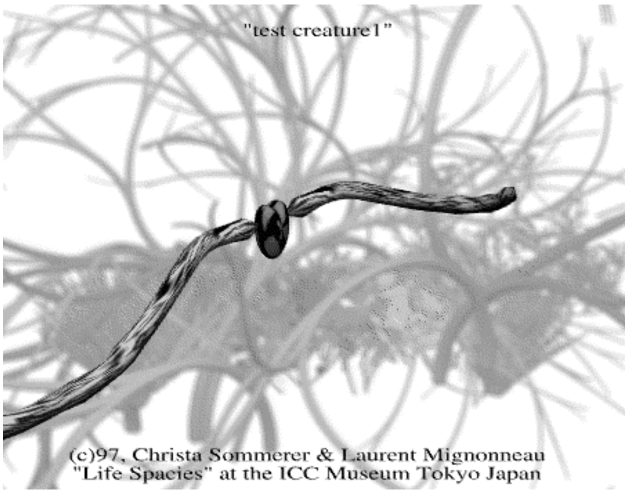

"test creature1"
(c)97, Christa Sommerer & Laurent Mignonneau
"Life Spacies" at the ICC Museum Tokyo Japan

Fig. 8 Creature created by email in Fig. 7

As soon as this message is sent to the server in Tokyo, the creature starts to live in its virtual environment and the author of the text receives a picture of his or her creature in return. Figure 8 shows an image of the creature created by the text message of Fig. 7. Because the text message was rather short, the corresponding creature consists just of one body and one pair of limbs, similar to the default case but with long limbs and a heart-shaped body.

2.3. Variations in the Creature's Design

By sending more complex messages with more characters, words and varied syntax, it becomes possible to create more elaborate creatures with more bodies, limbs and variation in body form, texture, size and color. Figure 10 shows an example and Fig. 9 is the corresponding text message.

Date: Sun, 01 Nov 1998 13:20:32 +0900
From: Christa Sommerer <christa@mic.atr.co.jp>
To: life@lc.ntticc.or.jp
CC: christa@mic.atr.co.jp
Subject: example #4

this is not a sentence, it is a creature, it is now in Tokyo, where it lives.
it is a creature, this is not a sentence, where it lives, it is now in Tokyo.
it is now in Tokyo, this is not a sentence, it is a creature, where it lives.
where it lives, it is a creature, it is now in Tokyo, this is not a sentence.

Fig. 9 Complex email messa**ge**

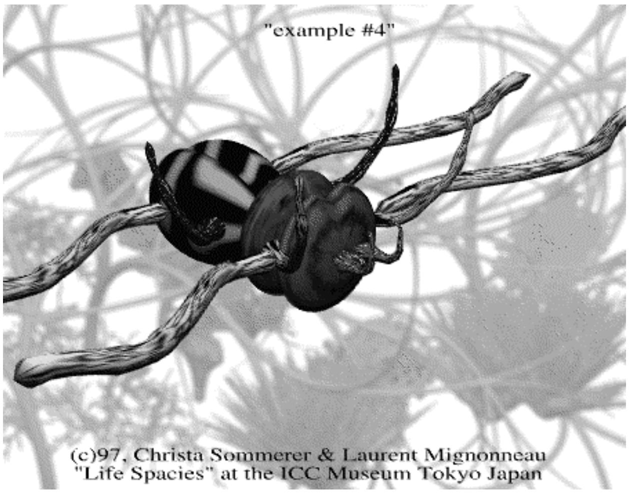

"example #4"
(c)97, Christa Sommerer & Laurent Mignonneau
"Life Spacies" at the ICC Museum Tokyo Japan

Fig. 10 Creature created by complex email message

3. Behaviour of the Creatures

3.1. Energy and Speed

A creatures behaviour is basically dependent on two paramters:
a) its Energy level (E)
b) its Speed (S) or ability to move
While the Energy level (E) is a value that constantly changes as the creatures moves in its environment and decreases by increased movement, the Speed (S) value is designed by the creatures body physics. A creature with a large body and small limbs will typically move slower than a creature with a small body and long limbs. Additionally the shape of the creatures body and limbs have an influence on the ability to move. On the other hand the Speed (S) value is decided at creation through the text characters arrangement in the creatures genetic code that is interpreted and translated by the design function table as shown in Figure 4.

Speed (S): depends on creatures body and limb size
decides how fast the creature can move
Energy (E): E = 1 at birth
Speed (S) of movement reduces E
E < 0 ..... creature becomes hungry
E > 0 ..... creature can mate

Fig. 11 Creatures behaviour decision parameters

4. Interaction among Creatures

Based on how much Energy (E) a creature has at a given moment and how fast it can move in the environment (Speed (S)) the creature's interaction with other creatures will be determined. If, for example, the creatures Energy level (E) reaches a certain value of E < 0, the creature becomes hungry and wants to eat. On the other hand, if the Energy level has risen to E > 0 the creature wants to mate with other creatures. Figure 12 shows this relationship between energy levels, feeding and mating behaviour.

Feeding: if E < 0 .... creature wants to eat text characters
it eats the same characters as its genetic code
("John" creature eats: "J', "o", "h", "n")
Mating: E > 0 .... creature wants to mate, if successful,
parents will exchange their genetic code
-> a child creature can be born
Evolution: Selection on faster creatures, as they can eat and
mate more frequently

Fig. 12 Creatures interaction parameters

4.1. Feeding

A creatures who's Energy level has risen to E < 0 becomes virtually hungry and aims to eat text characters provided by the user through the text input editor of the system. What kind of text characters are being released depends solely on the users decision.
Creatures also have preferences for certain types of food, the only eat text characters that are contained in their genetic message. As an example, a creatures who's genetic code is "John" will only eat "J', "o", "h", "n" characters. By eating text characters the creature will manage to add up a certain amount of energy and eventually its Energy level can rise above E > 0 again.
However it might be necessary for the creature to eat several text characters as the creatures often moves while looking for text characters.

4.2. Mating

Given that a creature succeeded to add energy E > 0 it will become a potential mating partner. It will now look for a suitable mate, who's energy level also lays above 0. The two potential parent creatures now will move towards each other and try to collide. If successful, the two parents exchange their genetic code through a cross-over operation and, as a result, a child creature can be born. This offspring creatures carries the genetic code of its parents with some mutations. Figure 13 shows an example of a mating process.

Example of Mating Process

Parent creature (P1) and (P2), child (C)
| .... area of cross-over
^ ... location of mutation

Parent Creatures P1 and P2

```
(P1) This is a creature, it lives in Tokyo.
              ^    |       |
(P2)  This creature is now living in Tokyo.
                      |        |
```

Child Creature C

(**C**) This is ancrea is now l lives in Toky.

Fig. 13 Mating process and birth of child creature

4.3. Growing and Death

A creatures life time is not pre-determined but influenced by how much a creature eats. Through eating the creature increases its body size until a maximum size of about four times the original body size. On the other hand, a creature will starve when it does not eat enough text characters and ultimately it will die and sink to the ground.

5. Complex Interaction and Evolution

The constant movements, feeding mating and reproduction activities of the creatures result in a complex system of interactions that can display features of artificial evolution with selection upon faster creatures. Additionally, the users input decisions on how to write the text messages and on how to feed the creatures also adds constant change to the system. As a result, a complex system is created that features complex interactions between creatures and creatures and well as users and creatures. Figure 14 shows a screen shot of different creatures as they mate and feed on text characters.

6. Conclusions

We have created an interactive systems that enables Internet users to create artificial creatures from written language. By using our text-to-form editors, we can treat text messages as genetic code and translate the text's parameters, such as letters, syntax and sequencing of text, into three-dimensional structures.
While various artists [5, 6, 7, 8] have been using the Internet to create web-based interactive art works, "Life SpaciesII" is the first systems to translate written text into visual three-dimensional forms that can become alive. Possible applications of this system outside the artistic arena could include the encrypting of text messages into images, or the creation of games for the entertainment industry.

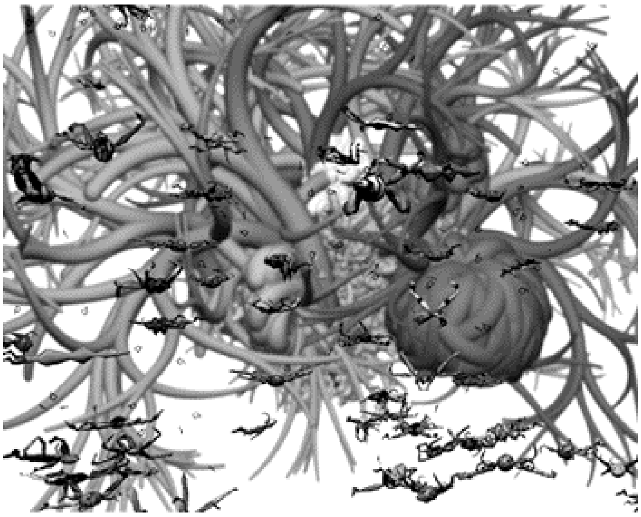


Fig. 14 Complex behaviour among creatures

7. References

[1] C. Sommerer, and L. Mignonneau, “The application of artificial life to interactive computer installations,” In: *Artificial Life and Robotics Journal*, Springer Verlag Tokyo, Vol.2, No.4, 1998, pp. 151-156.

[2] N. Chomsky, *Language and Mind*, New York: Hbj College & School Div, 1972.

[3] C. Sommerer, and L. Mignonneau, “Life Spacies,” In: *ICC Concept Book*, NTT-ICC Tokyo, 1997, pp. 96 -101.

[4] C. Sommerer, and L. Mignonneau, “Life Spacies: a genetic text-to-form editor on the Internet,” In*: Proceedings AROB 4th’99*, Beppu, Oita, 1999, pp. 73-77.

[5] T. Anzai, and R. Nakamura, 1994, "RENGA," http://www.renga.com/

[6] M. Fujihata, 1996, "Light on the Net Project," http://www.flab.mag.keio.ac.jp/light/

[7] M. Amerika, 1997, "Gammatron," http://www.grammatron.com/

[8] K. Goldberg, 1998, "Memento Mori: an Interface to the Earth," http://memento.ieor.berkeley.edu/