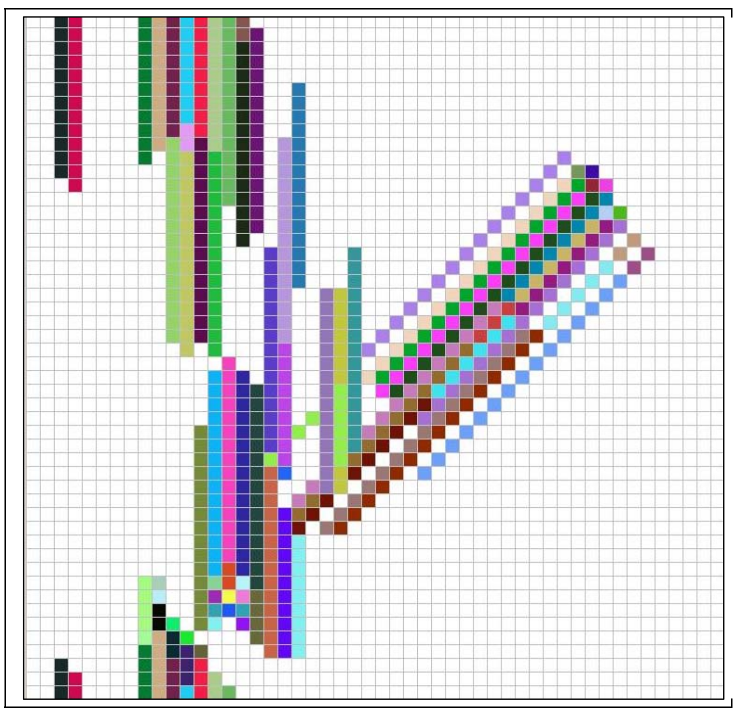
Design of Autonomous Systems

LifeForm

project leider: Peter van Lith



Team:
Liang Wang
Nicolaas Heyning
Sjoerd Kerkstra
Wouter Suren
Coen Pieterse

Table of contents

1. Introduction

The goal of this project is to create a simple environment which holds a number of simulated animats. The animats react to each other through a set of rules so it can extract, repel or ignore an animat. By influencing these rules the experimenter can alter the collective behavior of the world and complex stable states may emerge. The rules can be influenced by a number of sensors thus allowing this program to become an interactive work of moving art. This environment allows us to experiment with a simple agent world and creates a framework for artists to create cool interactive art.

2. Background

The theory used in the lifeForm project is most closely related to a field of research often referred to as 'emergent systems' or more generally 'Artificial life' or 'A-Life'. This field has become more prominent in the past few years. While there is no consensus on the name, or the exact boundaries of this field, it is possible to describe some common traits. Research generally focuses on the complex interactions of large numbers of relatively simple agents. Such systems are often called emergent because of the structures and patterns that often emerge from this interaction without having been explicitly defined. The most well known example of such a system is undoubtedly ‘Conway's Game of Life’[1], named after the American mathematician John H. Conway. The Game of Life consists of an infinite two-dimensional grid of cells (these cells are frequently referred to as 'cellular automa') in which each cell can be ‘alive’ or 'dead'. The game runs in discrete time steps in which the state of each cell is evaluated according to the state of the eight cells surrounding it. Since the original conception of Game of Life, many similar systems have been invented. Among the most well known are the one-dimensional Wolfram cellular automata[2] and the boids flocking algorithm by Craig Reynolds[3].

2.1 Applications

A few applications of emergent systems include the testing of circuits in hardware[4] and the modeling of large flocks of creatures or people. One of the earliest examples of this are the penguins in Tim Burton's Batman Returns (1992), which were modeled using a modified boids algorithm.

3. Environment Setup

3.1 Software

Our environment is a simple testing ground for experimenting with emerging behavior from animated agents. The environment consists of a raster of squares. Each square can hold a piece of an agent and can be any color. The edges of the grid can be linked to each other, creating a toroidal environment (figure 1), or the environment can be set to flat. The agents look like worms and are displayed as a series of connected squares in the same color. The agents can vary in length and the head of an agent can move in any of the 8 adjacent squares. The direction of choice is dependent on a combination three behavioral factors; persistence, sociability and ignorance.

Persistence is a measure of eagerness of an agent to continue the direction it went in the last step. It can be scaled from -50 to 50. A persistence of -50 will force the agent to always turn around. A persistence of 0 will always make the agent go in a random direction and a persistence of 50 will make the agent continue its current direction. Any value between these extremes represents the probability that the agent will choose between these behaviors.

Sociability is a measure of eagerness of an agent to move towards another nearby agent. If there is a neighbor agent, either the head or the tail, in any of the 8 adjacent squares, then the agent will react to his own sociability measure. If the sociability is set to one then the agent will move toward its neighbor, if sociability is set to zero the agent will ignore other agents. When sociability is set to minus one then the agent will move away from its neighbor.

The last measure that can be influenced by the viewer is ignorance. This measure is scaled from 0 to 100 and controls the chance of which the persistence of an agent is overruled by it's sociability. When set to 100 the agents movement will only be influenced by its sociability, when set to 0 the agents movement will only be influenced by its persistence. The ignorance controls an agent’s ignorance to the other agents in the environment.

There will also be a controllable variable speed. This won't directly influence the agent’s behavior, but controls the frequency of the update of all the agent’s position, therefore controlling the speed with which all the agents move over the screen. A high speed gives the viewer a more lively sensation of the agents and a low speed allows the viewer to study the agent’s movements and influence more closely.

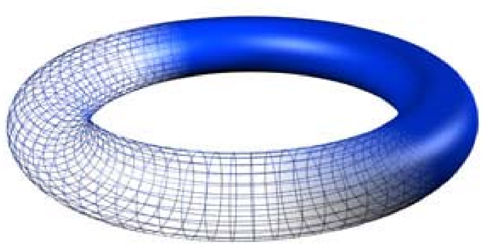
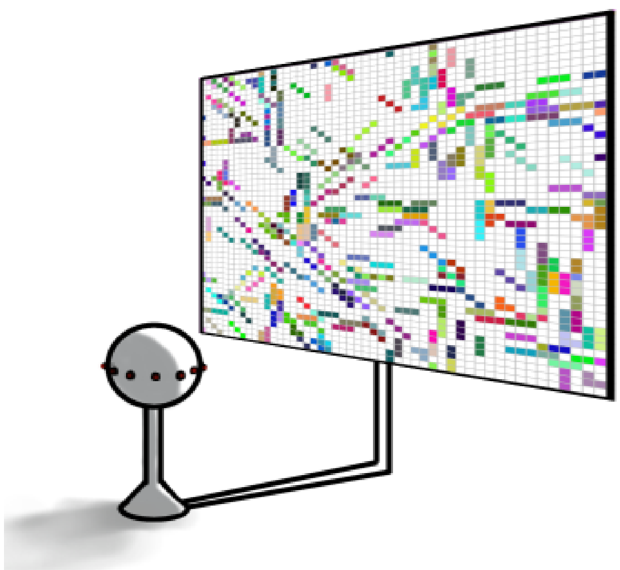


Figure 1. A toroidal environment

3.2 Hardware

By addition of sensors to the environment an interaction between the user and the agents behavior is created. Through this setup the lifeForm experience is greatly enhanced and can be used to entertain viewers. The ultimate goal of the experiment is the creation of an interactive work of moving art in which the viewer can change the look of the picture. Because the continuous user interaction, different kinds of behavior will emerge, stimulating viewers to participate in the art. The current idea of the setup of the artwork is as follows. Using a beamer, the grid and the agents of the lifeForm software are projected onto a wall in the gallery. In front of the screen there will be a carbon fiber ball with a diameter of approximately 30 cm. standing on a base of approximately 1 meter high. Sensors are integrated into the surface of the ball. These sensors are connected to different agent's behavioral properties. For example by touching the ball on a certain spot, the user can change the agent's persistence from low to high, making them behave in a completely different way. By removing the hand the behavior might drop back to its original state.



The physical environment setup of the lifeForm Project showing the projection screen and the sensor ball

4. Sensor input

The goal of the LifeForm project is to create a framework for artists to create cool interactive art. To make the project interactive we want to control the behavior and speed of the agents with sensors. This chapter describes how the sensors are coupled into the project.

4.1 The sensor hardware

To connect the sensors to the PC we use a Muvium micro controller. This is a small device that connects to the COM port of the PC on one side and connects to up to 16 sensors on the other side. There is room for 8 analog and 8 digital sensors. We can choose from a wide range of sensors. The following types of sensors are readily available:

- passive infrared sensors
- active infrared sensors
- on/off switches

The chip can be programmed in JAVA. The JAVA program will run on the chip and reads the sensor values and can do any kind of (light weight) pre processing. The controller is accessible through a static IP address and opens a port for every sensor.

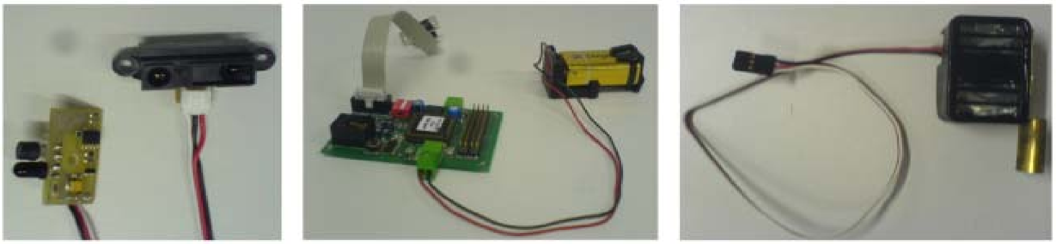


Illustration 1: sensor hardware. From left to right:

2 passive infrared sensors, a microcontroller, a passive infrared sensor

4.2 Reading sensor values

To read the values from the sensor we use a Remote Procedure Call server. This server connects to the controller and listens to a local IP address. On the server we install the services that can be called on the controller (for example: getSensorValue, getMotionValue).

We can send web service requests to the server which will send the request to the controller and return the response of a controller. The values returned by the sensors are 16 bit (from 0 to 1023).

4.3 Connecting the sensors

The sensor values cannot be used directly in the program. The value range of the sensors is different from the value range of the agent properties. It is desirable to limit the range of the sensor output to keep the agent behavior in a certain state.
To do this we have created the Sensor setup. This simple screen allows you to change the properties for each sensor.

We can change the range of the:

- sensor value
- speed
- persistence
- sociability
- ignorance

Once the values are set, the ignorance values are calculated with this equation:

$$\hat{V} = P_{FROM} + \frac{V - S_{FROM}}{S_{TO} - S_{FROM}}\left(P_{TO} - P_{FROM}\right),$$

where
PFROM and PTO are the start and end value of the value range of the ignorance
SFROM and STO are the start and end value of the value range of the sensor
V is the read value from the sensor

The result is the corrected ignorance value

All changes to the properties will take effect in the next cycle of the program.
A screenshot of the setup is shown in Figure 2 below.

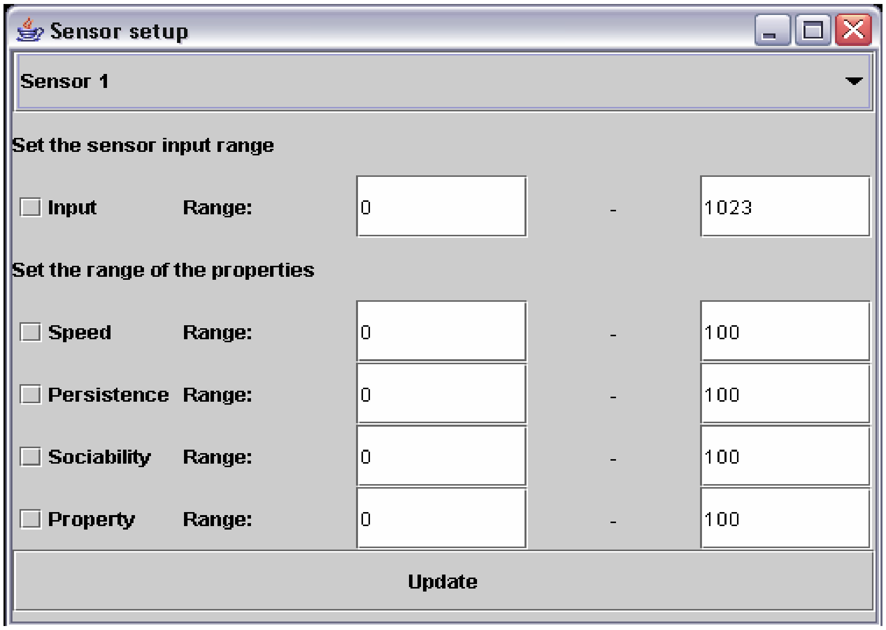


Figure 2. Screenshot of Sensor setup screen

5. Experiment

5.1 Measuring project success

The goal of the lifeForm project is to create an interactive work of art. Because of this rather broadly formulated goal, it is a challenge to find a way to objectively measure the success of the project. One somehow needs to measure the quality of the resulting artwork objectively. Literature on this subject is extensive[5] but either too philosophical or too complex to implement without hooking people up to an fMri scanner. In short, existing literature never fully meets our needs for a simple implementable rule which can be used to express the amount of appreciation for a work of art. We realized that a number based on a simple rule could only be a very rough indicator for an objective- or commonly held 'quality'.
Whilst keeping in mind the fundamental difficulties in using clinical methods to measure something as subjective as 'quality', test was defined by us that roughly calculates the quality or ‘interestingness’ of our project.

5.2 Test setup

The only practical way to approximate any kind of quality from a human perspective is to use statistics.
As the final project will be on display at an exhibition there will be plenty of potential viewers to collect these statistics. A good example of using statistics is to use questionnaires asking viewers to mark their appreciation of a piece of art. While simple and easy to implement, questionnaires do not meet our demands for a quality measure test. People visiting an exhibition will usually not enjoy filling out paperwork. Bugging people with forms to fill out will disrupt the experience of the artwork and scare away potential viewers. It is much better to use less intrusive methods of gathering statistics, preferably in such a way that it is invisible to viewers. Our solution uses the artwork’s own sensors to measure the quality of the artwork. The simple method proposed hinges on the following assumption:

'Interactive art needs to cause it's viewers to interact with it. If a viewer looses interest quickly after engaging the sensors, the interaction is not successful'

This assumption can be used to formulate a statistic method for calculating a measure of quality; the quality of a piece of interactive art can be expressed as the average length of interaction. The length of interaction is defined as the time between the first activation of the sensors by a person, and the time he or she stops activating the sensors. This measure is very much dependent upon particular configuration of the sensors. An artwork with very broad room filling sensors could get activated much more easily. A normalizing factor could be obtained by deactivating artwork's sensors. This would yield a base interaction length. Restoring the sensors would then show the added interaction length and with it the quality of the interaction.

The quality of the interaction component Q of an interactive artwork can be expressed as:

$$Q = Ia/Id,$$

where

Ia = Mean interaction length with activated sensors

Id = Mean interaction length with deactivated sensors

5.3 Observations

Because of the short term nature of this project and the limited resources at our disposal, it is not possible to actually test the artwork using the method described above. It is our hope that this will be possible during the exhibition which will take place after the end of the course.

6. Future work

6.1 Groups of Agents

The project has been extended with groups of agents. The idea behind this is that if there are more groups of agents in the grid, and each group has a certain behavior, then it will be easier for the user to find out what the different behavioral properties are because you can see the differences easier.

It will also be easier to recognize behavior changes invoked by the sensors, because if a certain sensor is activated (through, for example, movement), then it can have an increase in a certain behavior property in one group, but a decrease of the same property in another group. Because of the bigger difference of the behavioral change it will be easier to analyze how the sensors influence the agents. This, of course, makes playing with the art more fun.

6.2 Interaction between agents

With more than one group it is also possible to let the groups interact with each other, which makes it possible to create complex behavior. A high sociability of one group to other groups and a low sociability of another group will result in one group following the other group (if the inner sociability is high enough).

6.3 Conway's Game of Life

Conway's Game of Life became known in the 70's. It was invented by John Conway, who was a mathematician in Cambridge. The Game of Life can be seen as a cellular automaton. Life is represented as a collection of cells in a grid. The cells are able to replicate, live or die. These behavioral properties are based on simple mathematical rules. For example, if a cell has less than two neighbors it will die. An extension to the project that is based on Conway's Game of Life could be desirable. It would give the agents a more organic feel, which gives the viewers a more engaging interactive experience.

6.4 Birth and death of Agents

An interesting extra feature is to simulate the natural predator and prey equilibrium of an environment. In a normal environment the number of predators and the number of preys are closely coupled. If there are too many preys, the number of predators will grow, because there is enough food. If there are too few preys, predators will die of starvation.

To simulate this, the framework is extended to support more intelligent types of agents, the Predator Prey mode. An agent group in this mode will be divided into predators and preys. The preys and predators choose optimal actions, learned with a reinforcement learning algorithm. The preys will learn to 'mate', which can create a new (prey) agent and they will learn to stay away from predators. The predators will learn to hunt the agents. Catching a agent can create a new predator.

The Predator Prey mode follows the following rules:

- Catching a prey can create a new predator. The predators have enough food and are able to support another predator.
- Colliding predators can get killed.
- Every cycle predators and preys can get killed. Agents grow older and there might not be enough food to support the population.
- The chance of dieing is correlated to the size of the population. The agents age and don't get enough to eat.
- The chance of a new agent is negatively correlated to the size of the population. This prevents the population growing extremely fast and extremely large.

7. Conclusion

The goal of the project was to create an environment which can hold a number of simulated animats. The animats were to interact with each according to a simple set of rules which can be altered by the viewer. This environment was then supposed to be connected to a set if sensors allowing the viewer an intuitive interaction. This whole setup would be on display as an interactive digital artwork.

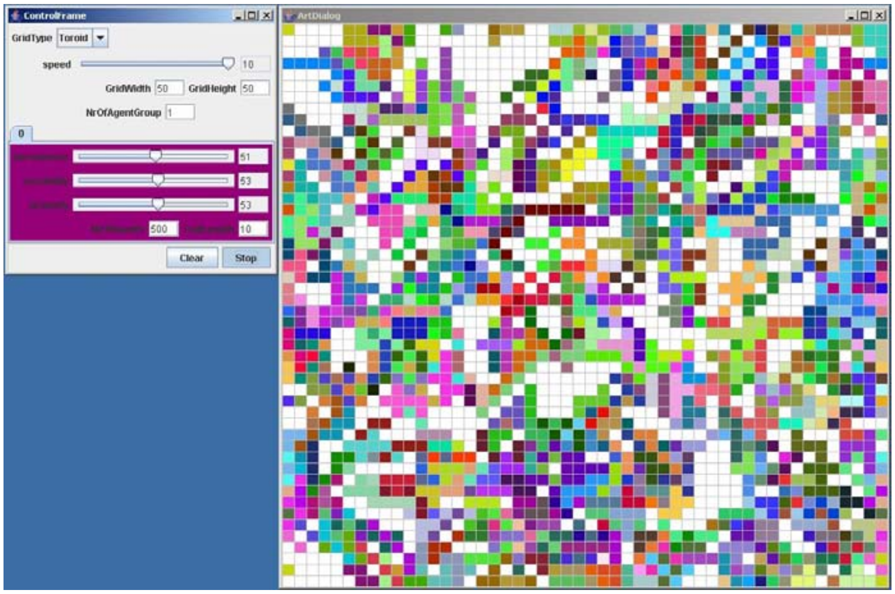
Research into the field of artificial life originated in the early seventies and has produced some impressive experiments like Conway's game of life. Artificial life focuses on the complex interactions of large number of relatively simple agents. The agents in our project take the form of wormlike animats which can move over a grid. The agents can move in any of the 8 adjacent squares and their direction is determined by a combination of three behavioral factors; persistence determines the eagerness to continue in the same direction. Sociability determines the agent's eagerness to move towards a nearby agent. And the ignorance factor determines the probability that one agent's persistence is overruled by its sociability.

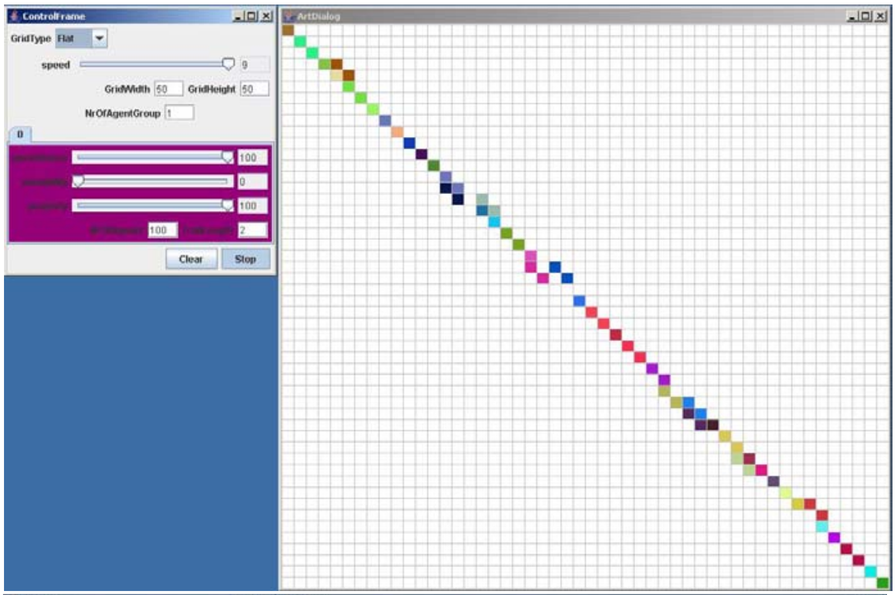
The sensors will be installed on the surface of a ball which will be setup in front of the display. With these sensors the viewer can manipulate the collective behavior of the agents. The emerging behavior has to be interesting enough to keep the viewer engaged with the artwork.

We believe that, although difficult to measure, the results of our project are quite satisfying. We completely succeeded in the implementation of the lifeForm software which will serve as a platform for the acting agents. The emerging behavior is very interesting, we can create total chaos with the whole grid filled with colorful agents. But we also created structured groups and a situation where all the agents would follow each other in a single file. It is fascinating to see how small changes to such simple rules can result in completely different states of behavior. We also experimented with some extensions to the rules like creating different independent groups of agents and situation where agents could pursue each other. Each of the extensions resulted in neat behaviors.

We also thought of a quantitative measure for testing how intriguing the artwork is. By recording the time the viewer is engaged with the project one can calculate an average measure of quality. But considering the short amount of time and the scope of our assignment we were not able to actually setup the artwork for display, therefore giving us no empirical testing ground for our project.

We found this was a very interesting project showing us a new side of artificial intelligence. The combination technology and art is a thrilling field which can bring out the best of both sides. We find it very stimulating the artwork will actually be on display next February and look forward to its presentation.


ControlFrame
GridType Toroid
speed 10
GridWidth 50 GridHeight 50
NrOfAgentGroup 1
0
51
53
53
500
10
Clear
Stop
ArtDialog


ControlFrame
GridType Flat
speed 9
GridWidth 50 GridHeight 50
NrOfAgentGroup 1
0
100
0
100
100
2
Clear
Stop
ArtDialog

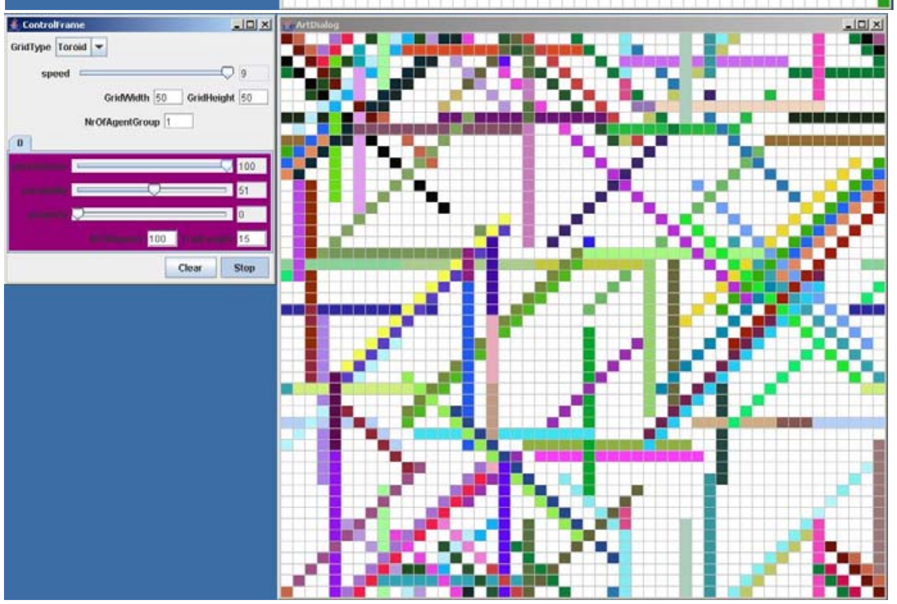

ControlFrame
GridType Toroid
speed 9
GridWidth 50 GridHeight 50
NrOfAgentGroup 1
0
100
51
0
100
15
Clear
Stop
ArtDialog

Figure 3. Screen shots of some of the different patterns that emerge due to simple adjustment in the behavioral properties.

8. Reference

[1] Conway’s Game of Life, http://en.wikipedia.org/wiki/Conway's_Game_of_Life

[2] Cellular Automata (1983) http://www.stephenwolfram.com/publications/articles/ca/83-cellular/

[3] Boids, http://www.red3d.com/cwr/boids/

[4] F. Corno, M. Sonza Reorda, G. Squillero, "Evolving Cellular Automata for Self-Testing Hardware”

[5] The International Association of Empirical Aesthetics, http://www.ume.maine.edu/~iaea/perm.html