Artipixoids!
The Concept (draft)

by Andrey Zamaraev
a5kin@github.com

April 9, 2017

> *2D body, 2D brain,*
> *2D pleasure, 2D pain.*
> — Flatlander credo

Artipixoids! is half generative art, half artificial life project with a focus on genetic cellular automata obeying energy conservation principle. Started in the end of 2014, it has undergone several major modifications, and is still in development. But now is the right time to share basic concepts underlying all previous modifications, so others could use it to build their own wonderful worlds.

The document is split up in two parts. In *The Manifesto* we will outline the main areas of the project development, and the main goals per each area. In *The Core* section, we will give the formal definitions to the features we would like to see in the core of our future framework.

Contents

1 The Manifesto

1.1 Artistic

1. We are going to generate the beautiful objects of abstract digital arts, even at the cost of computational resources.

2. The objects of arts are: static images, videos, sounds, 3D models, etc.

3. We will consider the object as Artipixoid if and only if:

   (a) the object is generated by CA fulfilling all the definitions from section 2 (The Core);

   (b) the object's generative algorithm is not using anything but CA states as input;

   (c) the object's generative algorithm is not based on the objects of arts, other than Artipixoids (eg. deep art based on Van Gogh paintings or photos doesn't count, but deep art based on other Artipixoids is OK).

4. Artipixoids that has no artistic value are bad and a subject to genocide.

5. Yet, if at least one person found them beautiful, we let them live.

1.2 Scientific

1. We are going to make experiments to test the hypothesis that a life can self-organize in artificial (digital) environment from a random initial state.

2. ***The properties of life***, as we see it, are following.

   (a) *Integrity:* every lifeform should be distinct from its local environment, and should preserve its individuality during the lifespan.

   (b) *Sustainability:* a lifeform should be able to self-repair in case of damage, and to maintain the physical manifestation (body) for a long enough period of time.

   (c) *Reproductivity:* a lifeform should be able to replicate, creating new individuals with small enough differences to consider them inexact copy.

   (d) *Novelty:* new lifeforms should arise out of the old ones, and they should act more effectively than the old ones.

(e) *Diversity:* there should be enough different lifeforms to put them in groups called 'species'.

(f) *Complementarity:* different species should learn to live in symbiosis and help each other to survive and improve, forming stable ecosystems.

(g) *Adaptability:* species should learn to evolve in a way they are constantly adapting to aggressive environmental changes.

3. ***The properties of artificial environment***, as we see it, are following.

(a) *Discreteness:* the space and time of our environment should be discrete. So, space should consist of indivisible quanta, and their updates should occur in separate timesteps synchronously. In addition, any parameter of spatial quanta should take integer values only. This requirements will allow us to run the simulation of our environment on modern digital computational systems.

(b) *Determinancy:* a global state of the environment at the timestep t should lead to one and only one global state at the timestep $t + 1$. This will give us ensurance of the exactly same outcome, every time we run a particular experiment with the same initial state.

(c) *Locality:* each quantum should have the same number and the same topology of its neighbors, and quantum's state should be updated using nothing more than the states of its neighbors and its own state. This will allow us to exploit the massively parallel computations, thus greatly speeding up the simulation.

(d) *Isolation:* the environment as a whole should behave like an isolated thermodynamic system. So, each quantum should have an energy level, and sum of all quanta's energy levels should be a constant value. Energy conservation will make us sure, that none of our experiments will lead neither to the extinction nor to the explosion of 'matter'.

(e) *Isotropy:* all possible reflections of quantum's neighborhood at the timestep t should lead to the same new quantum's state at timestep $t + 1$. This will allow energy to spread in all possible directions uniformly.

(f) *Polymorhism:* each quantum should have a microprogram, influencing its local update rules to some degree. A microprogram in return could also be changed according to update rules. This will allow us to add genetic informational component to the quantum level. We believe, this component is the key for the emergence of self-organized matter effects at the higher (macro) levels.

(g) *Stochasticity:* optionally, some pseudorandom component would influence local updates too. For that, each quanta should have its own PRNG with long enough period. This will allow energy to spread in chaotic clouds, rather than in direct lines.

(h) *Mutability:* optionally, a quantum's microprogram would randomly mutate over time. This will result in wider genetic search for better microprograms, and also may prevent the whole environment from being conquested by a single genome.

(i) *Magnetism:* optionally, quanta would have the tendency to 'attract' each other with some sort of force (gravitational, electromagnetic, or any other). We believe, magnetism is another key aspect behind the phenomenon of life as we know it.

4. **Random initial state**, could be obtained using following known patterns.

(a) *'Primordial Soup':* each and every quantum initially has an equally small amount of energy, other parameters are random. This is a good test for the ability of energy to self-organize in clusters from the completely uniform distribution. The bigger clusters are, the better. Fractal clusters are even better.

(b) *'Big Bang':* a small area of space is initialized with a high amount of energy and random parameters per each quantum. Outside the area, quanta has either zero or minimum possible amount of energy. This is a good test for the ability of energy to spread in empty space. We could also detect the fastest types of energy using this pattern.

(c) *'Super Cluster':* same as Big Bang, but several areas with a bit lower energies are randomly initialized. The positions of areas are random across the space. This is a good test for gravitational effects, and also for collisions of different energy beams, produced in micro-bangs.

5. To obtain the better results, we reserve the rights to intervene the natural process of evolution. This will ruin the idea of self-emergence, but still may prove the possibility of artificial life in our environment.

The presumable methods of intervention are following.

(a) *Genetic selection:* during the simulation, we are collecting the most interesting and promising areas of space, then combine them all in a single initial state and repeat the process recursively while novel phenomena of interest are showing up.

(b) *Directed mutation (a.k.a. 'Hand of Fate')*: during the simulation, we are occasionaly pointing our finger to a small area of interest, and all quanta in this area are randomly changing parameters, excluding energy level. Then, evolution continues.

(c) *Pre-designed components:* we are designing patterns of matter manually quantum by quantum, then using them either as a parts of more complex designs or to impact on 'raw' matter in some way.

1.3 Divine

1. We, as young goddesses and gods, are willing to create worlds with one click of our fingers. Actually, it doesn't have to be a click, a tap of one finger is even better.

2. After the tap, we are willing to say the Word, which contains the essence of the future world.

3. Then, the world should begin to bloom, and it should be the same every time we said the same Word.

4. Some of us will just meditate on the beauty of their worlds, wandering around and changing the views.

5. Others will actively play demiurges using scientific wizardry from previous section, or even more powerful stuff.

6. And some will get bored and destroy the creation with a single tap of the finger. Thus, letting its habitants live in some parallel universe, free of divine will and happy.

1.4 Social

1. We are going to build the community over those who interested in any of previously mentioned ideas.

2. For that, we need a framework, implementing basic CA engine described below, along with a set of modular tools to easily build engine modifications.

3. Then, we will be able to exchange new modifications in a form of compact modules.

4. We will also be able to exchange new modular tools and extend our framework with them for future re-using.

5. Finally, for each modification, we will be able to make a collection of the experiment files with initial state and hyperparameters included, and run them in uniform interactive environment on massively parallel hardware like GPU clusters.

1.5   Alien

Nothing there yet. This section is reserved just to fit the law of fives.

2 The Core

There is a special subclass of cellular automata that allows us to implement all the properties of environment described in section 1.2. We call it *Buffered State Cellular Automata* (BSCA). In this section, we will give the formal definition of all BSCA features, to be integrated in the core of our future framework.

2.1 Grid and neighborhood topologies

As in any CA, grid in BSCA is D-dimensional lattice of cells, each having N neighbors. $(D, N) \in [1..\infty)$.

The cells are stored in a sequence of length M:

$$C = \langle c_0, ..., c_{M-1} \rangle, \quad M \in [1..\infty), \tag{1}$$

where elements are nested sequences with uniform structure called *cell state*. See the exact definition of the cell state in Eq. 17 below. We may refer to C as to a *board state* later.

The cartesian coordinates of the cell with index i could be obtained as

$$[x_0, ..., x_{D-1}] = \chi(i), \quad i \in [0..M), \tag{2}$$

where χ is *lattice topology function*.

Let also define a sequence of all cells' coordinates as

$$X = \langle \chi(0), ..., \chi(M-1) \rangle. \tag{3}$$

We may refer to X as to just a *board* later.

The cartesian coordinates of cell's j-th neighbor could be obtained as

$$[x_0, ..., x_{D-1}] = \nu(X_i, j), \quad i \in [0..M), \quad j \in [1..N], \tag{4}$$

where ν is *neighborhood topology function*.

Thus, we will assume the whole grid topology is homogeneous if and only if the following equation holds for each value of j:

$$X_i - \nu(X_i, j) = d_j, \quad \forall i \in [0..M), j \in [1..N], \tag{5}$$

where d_j is a constant vector for each j. So, the vector difference between the positions of cell and its j-th neighbor should be a constant for each cell in a grid.

2.2   Border Effects

A *border cell* is a cell, having at least one of its neighbours off the board X. If A_i is a set of all i-th cell's neighbors, that are inside X:

$$A_i = \{C_i \mid \nu(X_i, j) \in X, j \in [1..N]\}, \quad i \in [0..M), \tag{6}$$

then the following inequality holds for all border cells:

$$|A_i| \neq N, \quad i \in [0..M). \tag{7}$$

In order to correctly process all cells, including border ones, we have to define a conditional function η for neighbors obtainment. Let call it a *neighbour function*:

$$\eta(i, j) = \begin{cases} C_{\chi^{-1}(\nu(X_i, j))} & \text{if } \nu(X_i, j) \in X, \\ \beta(\nu(X_i, j)) & \text{if } \nu(X_i, j) \notin X, \end{cases} \tag{8}$$

where β is a *border function* returning a state for a hypothetic cells outside X, and χ^{-1} is a reverse lattice topology function, satisfying the following equation:

$$\chi^{-1}(\chi(i)) = i. \tag{9}$$

Border function β could take a variety of forms. It could just be a constant pre-defined state (static borders). Or wrap borders into higher dimensional manifold topology, like torus, Möebius strip or Klein bottle. It could even yield a random state each time.

In case of wrapping, the border function will take a form of

$$\beta(x) = C_{\beta^*(x)}, \quad \beta^*(x) \in [0..M), \tag{10}$$

where β^* is a *wrap function*, mapping off-board coordinates to other cells on the board, and on-board coordinates strictly to the cells with those coordinates:

$$\beta^*(x) = \begin{cases} \chi^{-1}(x) & \text{if } x \in X, \\ i \in [0..M) & \text{if } x \notin X. \end{cases} \tag{11}$$

We will assume that the whole grid neighborhood is homogeneous if and only if the border function takes a form as in Eq. 10, and the following condition holds:

$$C_i \in \{\eta(i', j) \mid j \in [1..N]\}, \quad i' \in \{\beta^*(\nu(X_i, j)) \mid j \in [1..N]\}, \quad i \in [0..M), \tag{12}$$

so, each cell should be present in a set of neighbours for all of its neighbours.

Be warned, any border function that breaks the neighborhood homogeneity, will also break an energy conservation, unless the buffered interactions (see below) with off-board cells are explicitly restricted in update rules.

2.3   Single Cell Design

Let recursively define a parameter as either an integer or a sequence of parameters. Then, the infinite set P of all possible parameters is

$$P_0 = \mathbb{Z},$$
$$P_{n+1} = \bigcup_{i=1}^{\infty} P_n^i, \tag{13}$$

thus we can say that p is a *parameter* if and only if $p \in P$.

Let also define a recursive function φ, that will replace all integers with zeros in a parameter:

$$\varphi(p) = \begin{cases} 0 & \text{if } p \in \mathbb{Z}, \\ \langle \varphi(p_0), ..., \varphi(p_{|p|-1}) \rangle & \text{if } p \notin \mathbb{Z}. \end{cases} \tag{14}$$

We will call φ *parameter topology function*, and will assume two parameters p_1 and p_2 have the same topology if and only if the following equation holds:

$$\varphi(p_1) = \varphi(p_2). \tag{15}$$

The *generic state* then is a parameter with the constant topology within a particular BSCA. Let define S_s as a set of all possible generic states:

$$S_s = \{s \mid s \in P, \varphi(s) = p_c\}, \tag{16}$$

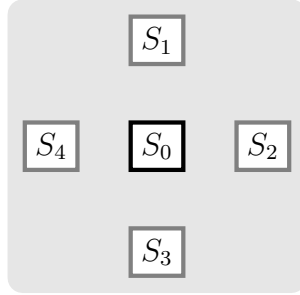where p_c is a constant topology value.

Figure 1: Single cell design in BSCA.

The *cell state* c (see Eq. 1) is a sequence of $N + 1$ generic states:

$$c = \langle S_0, ..., S_N \rangle, \quad S_i \in S_s \forall i \in [0..N], \tag{17}$$

10

where S_0 is the *main state*, and S_1, ..., S_N are the *buffered states*, one per each neighbor. At the Fig. 1, you can see a cell design for 2-dimensional BSCA with Von Neumann neighborhood. Buffered neighbor states is the cruicial aspect of BSCA, it is the basic feature allowing us to implement lossless models with energy exchanges and genome crossbreeding across all neighbor cells. See more on this topic in the sections below.

Let also define two more sets for future use:

$$S_c = S_s^{N+1}, \tag{18}$$

$$S_b = S_c^M, \tag{19}$$

where S_c is a set of all possible cell states, and S_b is a set of all possible board states.

2.4 Time and Updates

The *universe* is a board state evolving in discrete timesteps. Let define the universe U as a potentially infinite recurrent sequence:

$$U_0 = C_s,$$
$$U_{t+1} = \omega(U_t), \tag{20}$$

where C_s is an *initial board state* (also called *seed*), t is a timestep number and $\omega : S_b \to S_B$ is an *update function*.

In BSCA, the evolution at a single timestep is going in two phases: *emit phase* and *absorb phase*, so update function will take a form of

$$\omega(u) = \alpha(\varepsilon(u)), \tag{21}$$

where α is *absorb function* and ε is *emit function*. Let describe how both of them works in the next sections.
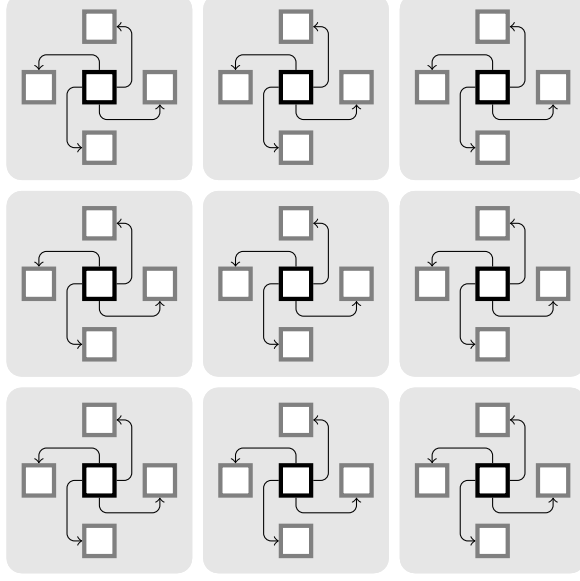
2.5 Emit Phase

Figure 2: Emit Phase in BSCA.

The process of emit is like sending the messages from a cell to its neighbors. Each cell performs it as follows.

1. The cell is taking a sequence of all its neighbors' main states, as well as its own main state:

$$\eta_e(C, i) = \langle \gamma(C_i, 0), \gamma(\eta(i, 1), 0), ..., \gamma(\eta(i, N), 0) \rangle, \qquad (22)$$

where i is the cell's index and γ is a function for obtaining an item from the sequence:

$$\gamma(s, i) = s_i. \qquad (23)$$

2. The cell is modifying its buffers and main state as a result of *decision function* $\sigma : S_c \to S_c$:

$$C_i := \sigma(\eta_e(C, i)), \qquad (24)$$

where i is the cell's index and C is a board state at the current timestep t: $C = U_t$. The decision function σ could be as complex as you need: from naive sum of neighbor values to sophisticated microprogram execution.

3. At the end of emit phase, each buffer should contain a generic state intended for a specific neigbor.

The whole emit function $\varepsilon : S_b \to S_b$ thus takes a form of

$$\varepsilon(C) = \langle \sigma(\eta_e(C, 0)), ..., \sigma(\eta_e(C, M-1)) \rangle, \tag{25}$$

Fig. 2 shows the emit phase for 2-dimensional BSCA with Von Neumann neighborhood. The arrows are indicating the updates. So, at this phase, the main states are read-only, and the buffers are write-only within every given cell, which gives us the ability to use massively parallel architectures.
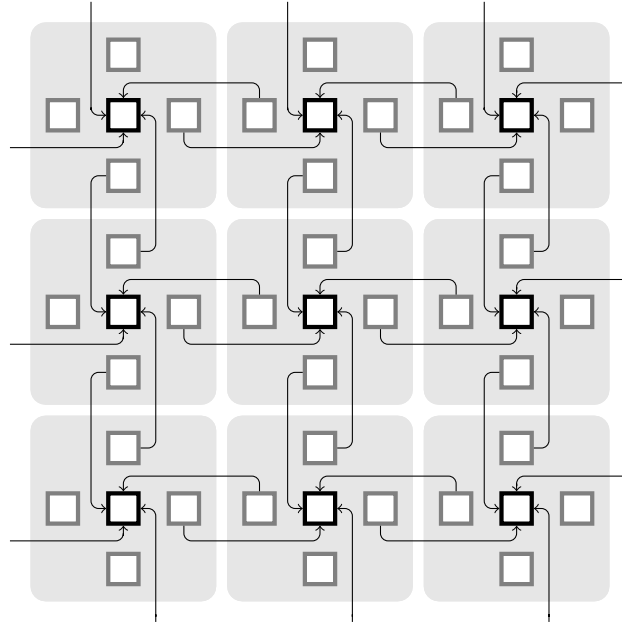
2.6 Absorb Phase

Figure 3: Absorb Phase in BSCA.

(TODO)