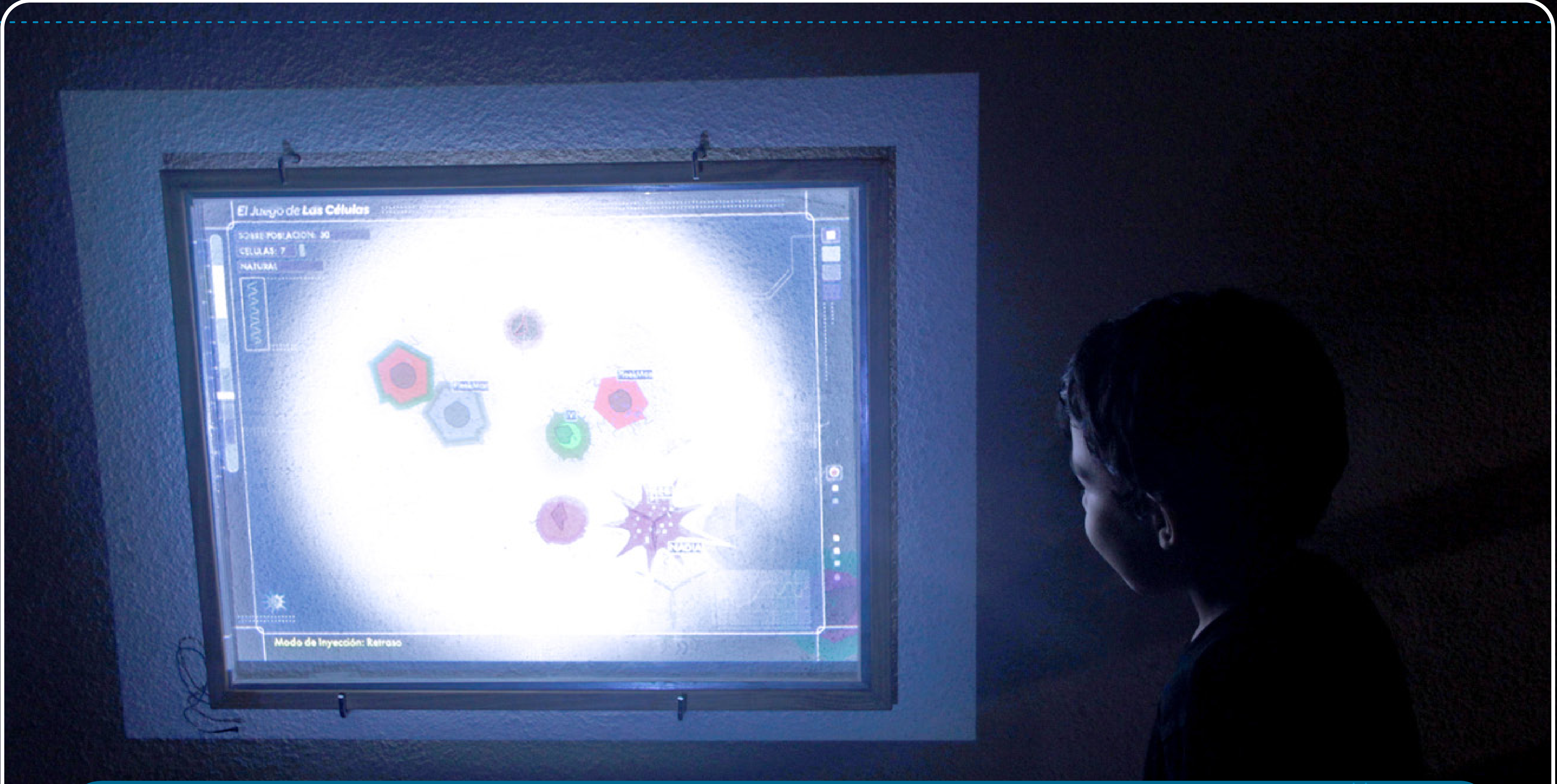


The Game
of Cells

An Interactive Installation that explores a Digital Genetic Model and invites the User to become a Scientist:

The Creator and the Observer in a Simulation.

The Game
of Cells

:00

The Game of Cells

Guanine
1 The Game
Adenine
2 The Installation
Photographs
Stations
Schematics
Cytosine
3 The Cells
Digital Biology
Digital Genesis
Digital Genetics
Digital Simulation
Thymine
4 The Experience
The Exhibition
Press
Thanks to:

:01

The Game

↓↓
The Discovery of Life.
| |
The Understanding of Life.
| |
The Engineering of Life.
| |
The Observation of Life.
| |
The Discovery of Life.
↓↓

The Magical Feeling of Creation.
Or the powerful feeling of being able to model a symbolic representation of an agent that can interact and react to other agents.

The Finger of God.
Or the possession of a fleshy finger that can place, at will, a graphical Instance of a complex Class over a digital Petri dish.

The Mad Scientist's Lab
Or an installation that makes Genetic Biology accessible to those foreign to the Natural Sciences.

The Decoding of a Genome.
Or the Decoding of a Digital Genome.

Welcome to
The Game of Cells.

:02

The Installation

The Game of Cells is conceived to be an installation that merges various scientific aspects, in order to give the user an accessible experience to Cellular Biology and Genetics.

It is strongly inspired by John Conway's famous “The Game of Life”, and the abilities frequently use in Role-Playing Games.

The aim is to, over the surface, bring a friendly user interface to operate on, and below the surface, carry out digital virtual analogies of biological processes.

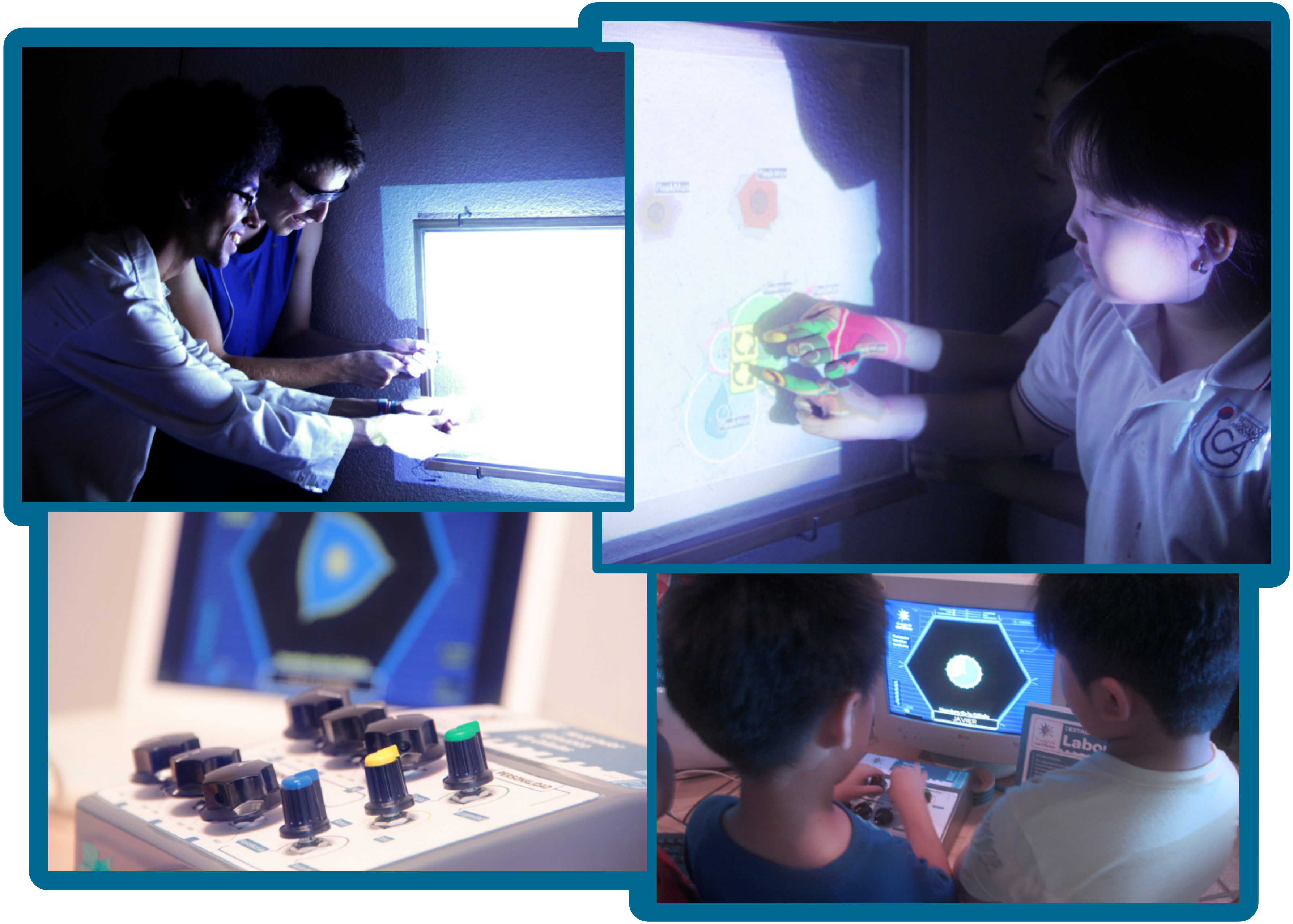
At the same time, merge them in a way that would not only become a fun experience, but also spark an interest on what is actually behind the scenes, what is creating such an attractive visualization.

Thus operating in different levels: There's a beauty that can be somewhat controlled by the creator, us. But deep down, many levels below, very complex processes are being executed with incredible precision. Concurrently, many levels above, clear behaviors are recognize, and patterns emerge.

Just like in Life. Just like in Computer Programming. Just like in Genetics.
Just like in The Game of Cells.

Although the Simulation aspect is full of rules, the objective of “The Game of Cells” is to make it as playful as possible. This means that a lot of randomness is put into the simulation, somehow blurring possible emerging patterns, to clear the way to the animation of frisky cells, predominant reproduction, interactive magic, and an attractive visual canvas.

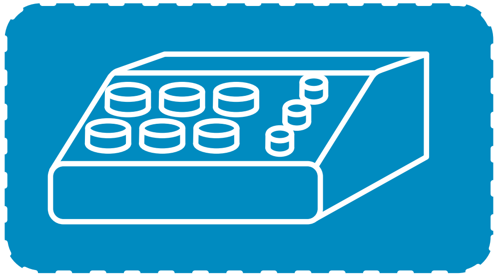
Photographs



Photographs

Stations

The installation operates over 2 Stations.

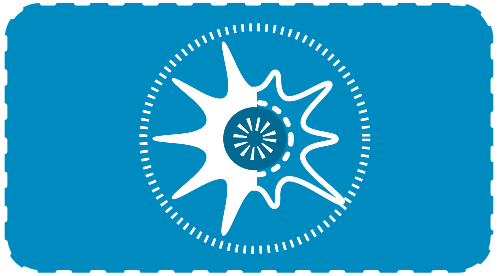


Station 1
The Laboratory

The Player, via a Custom Modeling Control Panel specially built for the Genetic Modeling of the Virtual Cell, is able to engineer his own cell, drawing it's shape parametrically and assigning personality attributes such as Love or Hate. At the same time, the player can name his cell.

When ready, the pressing of a giant button will create the Genetic Seed and shoot it towards the Second Station.

The electronics for the control panel are built with an Arduino Mega.



Station 2
The Microscope

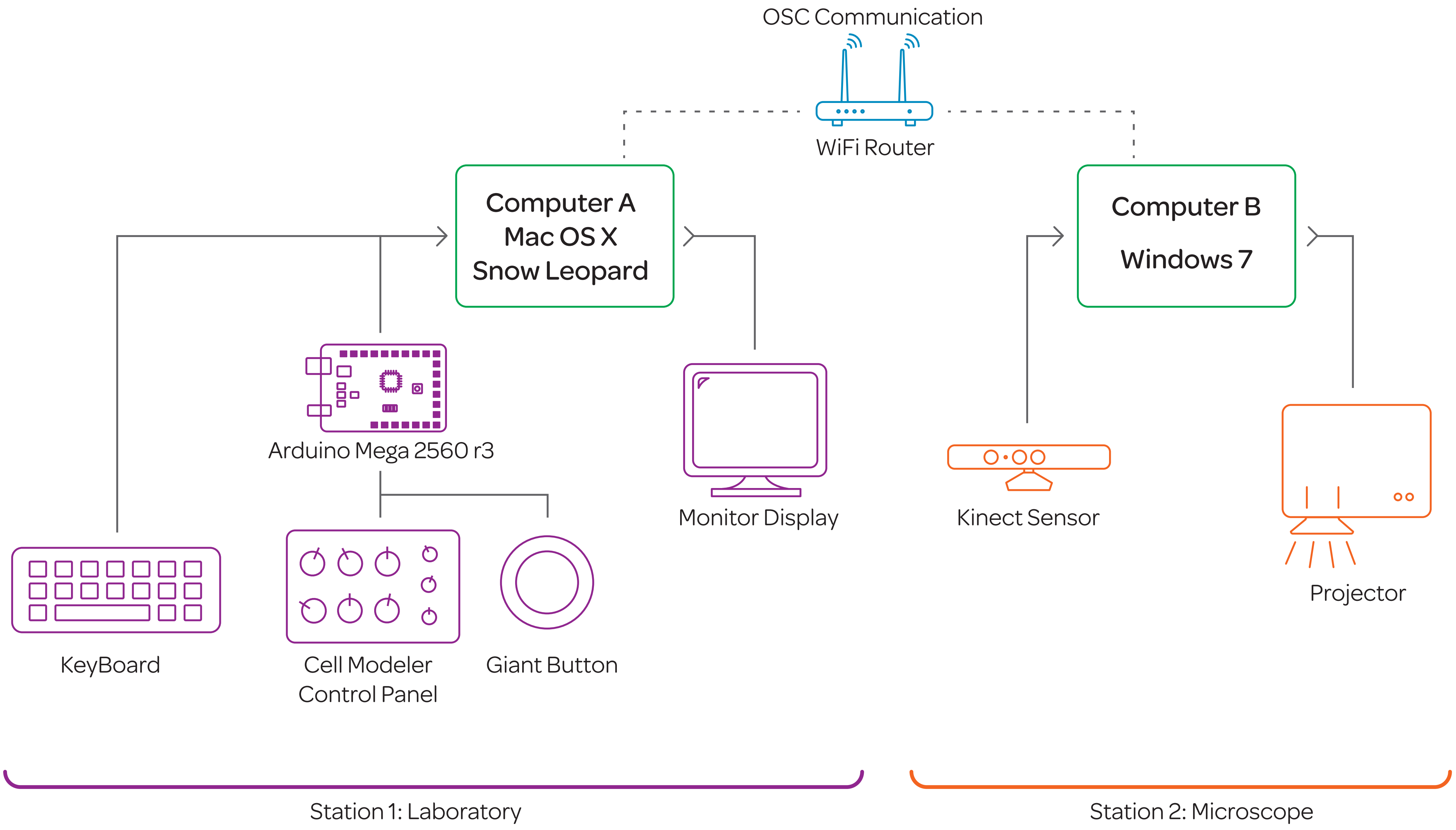
It allows for the observation of the interaction between the cells of all players, inside a simulation with visual cues that permit the understanding of the interactions taking place.

By means of a projection that simulates the view over a microscope eye-piece, the player is able to insert his cell in any location just by pointing and touching with his finger over the projection.

A Player Cell displays it’s name-tag so that each player can follow his own cell throughout his lifeSpan.

The Microsoft Kinect Sensor is used to sense user spatial data.

Installation Schematics



OSC Communication

WiFi Router

Computer A
Mac OS X
Snow Leopard

Computer B

Windows 7

Arduino Mega 2560 r3

Monitor Display

Kinect Sensor

Projector

KeyBoard

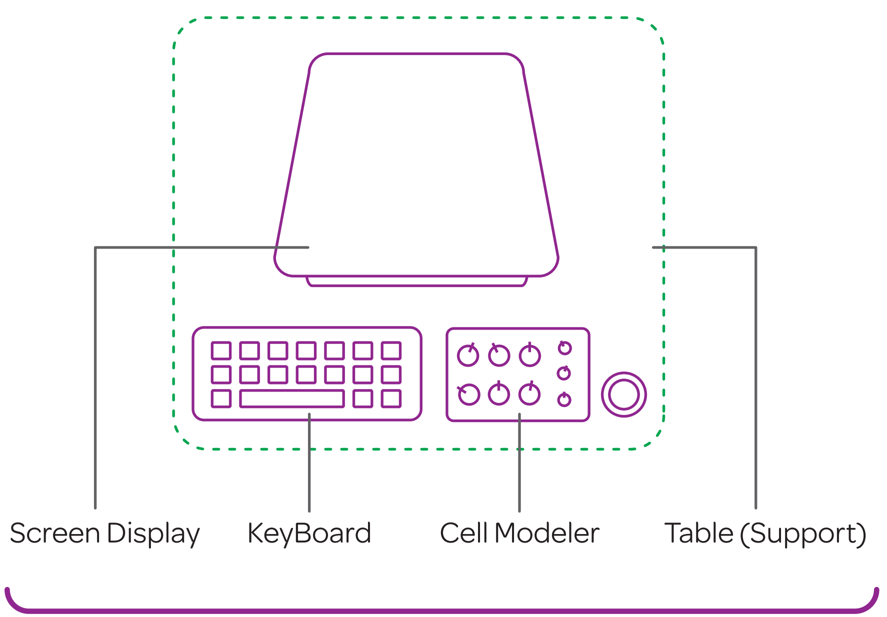
Cell Modeler
Control Panel

Giant Button

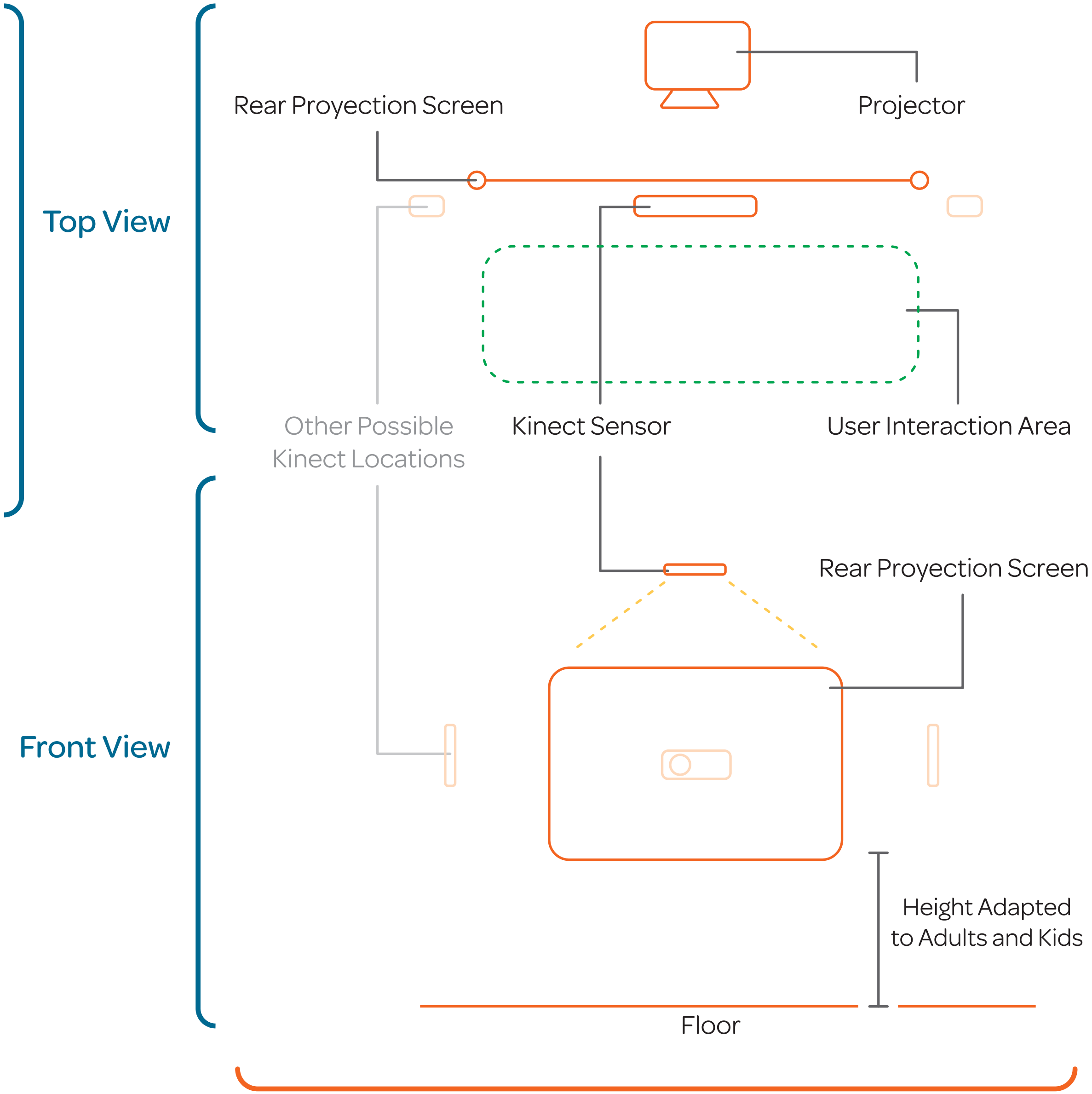
Station 1: Laboratory

Station 2: Microscope

Installation Setup



Station 1: Laboratory



Station 2: Microscope

The Kinect Sensor can be positioned to sense from any side, bearing in mind that the cone should capture all of the projection area.
The software is armed with a calibrator that can flip and rotate axes accordingly.
To avoid self-shadowing, it is better to rear-project.

The Game
of Cells

:03

The Cells

Digital Biology

The Cells are the stars of the show.
Living throughout the 2 stations, they are shaped, engineered, moved, and born following digital analogies of biological processes and mathematical interpretations.

It is important to understand that everything that makes up a cell is encoded in their Digital DNA, and any kind of user-constructed attribute will inevitably have to code it's parameters into the genetic level.
This way, processes like Inheritance, Mutation or CrossOver can be carried out correctly.

This section spreads across 3 sub-Sections:

Digital Genesis

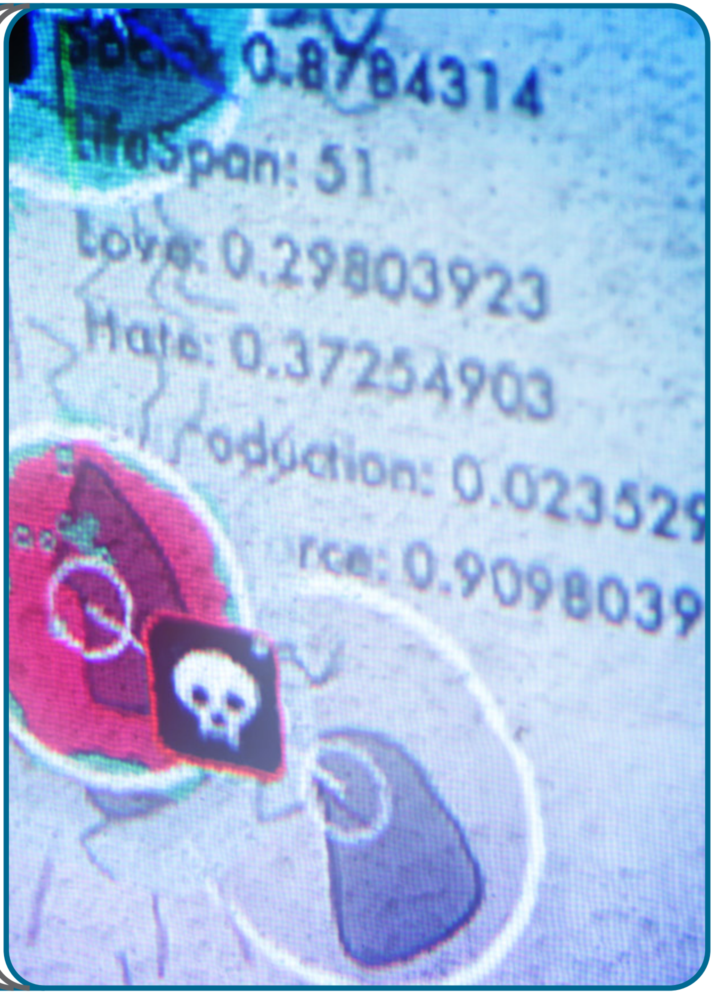
Shape and the Super-Formula
Balanced Attributes.

Digital Genetics

Binary DNA Strand
Genotype and Phenotype

Digital Simulation

Rules of Interaction
Reproduction



0.8784314
Lifespan: 51
Love: 0.29803923
Hate: 0.37254903
oduction: 0.023529
rce: 0.9098039

Digital Genesis - Description

Cells can be born in 2 ways:
1 – With randomly-generated genetic information.
2 – With partly user-generated genetic information.

Station 1 focuses on letting the user generate some of the genetic information, while other is left random, for practical reasons.

An Arduino controlled Modeling panel was built to deal with Cell Shape and Personality Attributes.
It is much more exciting and engaging for the user to handle non-conventional input devices. Thus, the Control Panel was customized to follow the aesthetics and bring some immersion to the installation.

Cell Shape is parametrically constructed following Johan Gielis formula for the Super-Shapes, and personality attributes are balanced against each other.

The User-Controlled Personality Attributes are the following:

Sociability ⟺ LifeSpan (Life Expectancy)

Love ⟺ Hate
⇓ ⇓
Reproduction ⟺ Brute Force

Other Non User-Controlled Attributes deal with Motion (Velocity, Distance, Timers), which also directly affect cell interaction.

Although other cell shaping methods were tried, the superFormula undoubtedly became the most straightforward and effective way of dealing with complex shapes with few controllable parameters.

From Wikipedia:
The superFormula is a generalization of the superEllipse and was first proposed by Johan Gielis.
Gielis suggested that the formula can be used to describe many complex shapes and curves that are found in nature.

$$r(\Theta) = \left[\left| \frac{\cos\left(\frac{m\Theta}{4}\right)}{a} \right|^{n2} + \left| \frac{\sin\left(\frac{m\Theta}{4}\right)}{b} \right|^{n3} \right]^{-\frac{1}{n1}}$$

Where:

r = radius
Θ = angle (in radians)

a, b, m, n1, n2, n3 = Input for the shape (resulting in an organic output)

The Modeler Panel

The Modeler is a Custom-built Arduino-controlled Panel that let's the user create a cell via an array of knobs.

It presents 3 sections:
- 6 knobs to input parameters into the superFormula,
- 3 other knobs to balance personality attributes,
- 1 Giant Button (which is a hacked Staples' "Easy Button") to send the cell seed.

At the same time, the user can name the cell via a keyboard.

Once the user is satisfied with his cell, the Button triggers a function that encapsulates the data into an OSC message, which is sent to the Simulation (The microscope).

A : 1.7442764

B : 1.1095952

:: 3.8047752

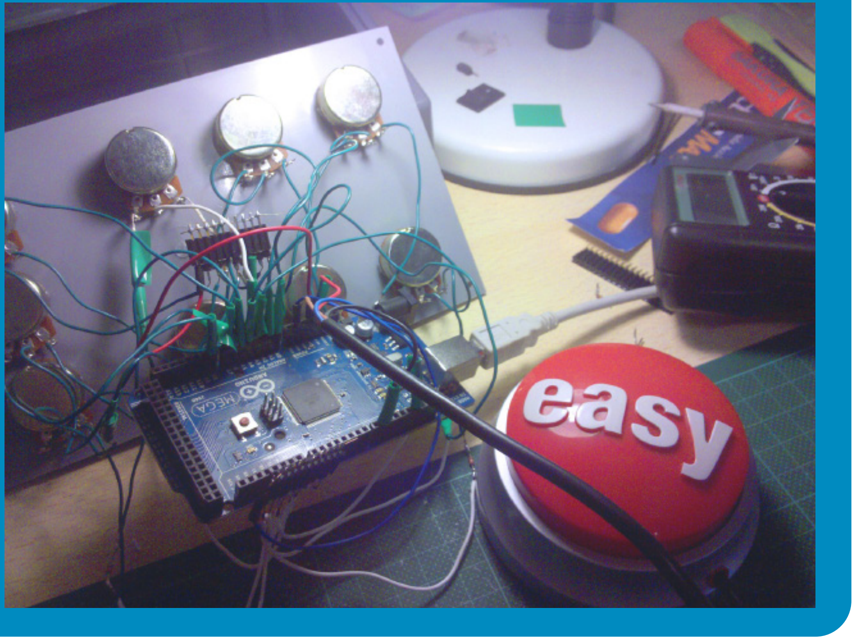
/\ 4.797297

& 0.8384998

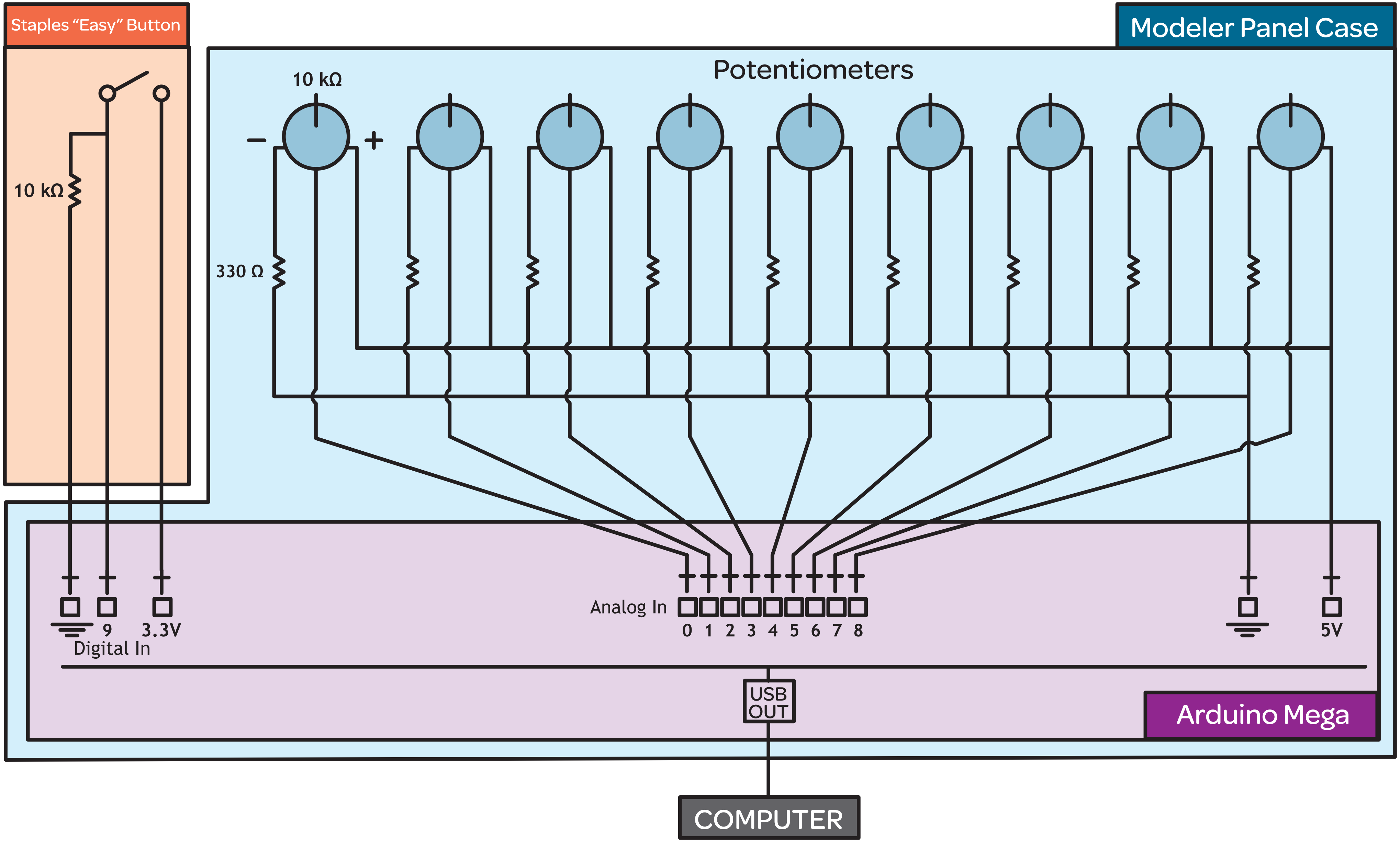
6.000

Nombre de la Célula

INVADER ZIM



The Modeler Panel - Electronics Schematics



```
0101101011010100000100010
11101111000100001011010110
1011010101010101101011101111
0001000010101101010101011
01011101111000100001011010
11010110101010101101011101
11100010000101110101101011
10111100010000101101011010
11010101010101101001101110
```

This is a Cell

One of the objectives of The Game of Cells is to create a simple digital analogy of DNA Structures, and present an abridged reproduction system.

This is hidden from the user and his interactions, but plays a crucial part during the simulation and cell interaction.

Genotype

The correspondence between the Natural DNA structure and it's digital translation is quite direct and straightforward.

Biologically speaking, there are 4 molecules (called **MacroMolecules**) that constitute the most basic building blocks of every living organism: DNA, Proteins, CarboHydrates and Lipids.
DNA is special, because it contains all the necesary information, the **Genotype**, that gives the organism it's identity. The expression of the genotype into physical attributes is called the **Phenotype**, and this will determine how the organism deals with his environment.

A **DNA Strand** is formed out of **4 Bases** (Adenine, Thymine, Guanine and Cytosine) that are coiled and linked together. These 4 bases pair with one another in a very strict way: Adenine only pairs with Thymine, and Guanine only pairs with Cytosine. These 2 pairs are called **nucleoBases**.

In order to be read (or transcribed) correctly, these nucleoBases are grouped together in units called **Genes**, which actually make sense to the reader/transcriber. *(Note that not all portions of the DNA Strand are Genes, only the nucleoBases that can be coded into proteins.)*
Although the structure and workings of genes is far more complex, this simplification provides enough room to play with.

In Computer Software Programming, the simplest, most basic building block is the **Bit**. Everything that is programmed will eventually be down-mix to bits for the computer to process and carry out procedures.

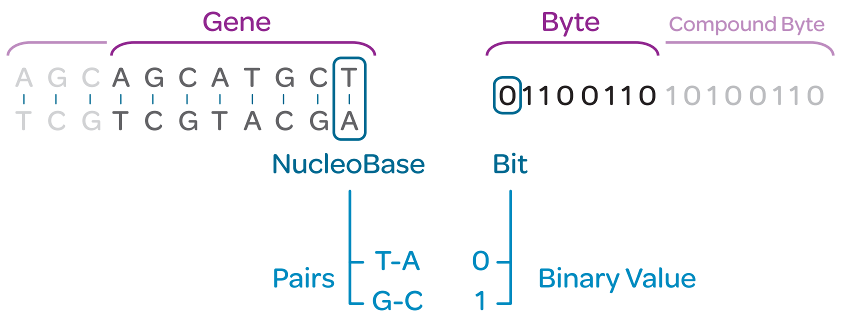
But the tiniest block of data that the computer can process is a **Byte**, which is group of 8 Bits. As the C++ and C Language Standard defines it, it is the *“addressable unit of data storage large enough to hold any member of the basic character set of the execution environment".*

Just like a Gene, a Byte makes sense to a computer.

While a single base can be 1 out of 4 types (A, T, G, C), there only 2 ways in which they can be found, as nucleoBases *(ignoring inverted bases)*.
Thus, they could be represented as Pair 0, or Pair 1.

A single bit is restricted to 2 “types” (Zero, One).Therefore, a digital DNA Strand can be formed out of Gene-Bytes that contain 8 nucleoBases-Bits, with each one representing 1 of the 2 pairs of base bonding.

There's a difference between the Biological Genes and Digital Bytes: Genes do not have a fixed length (number of nucleoBases), while a Byte does have a fixed length of 8 bits. This is called an **Octet**.
Strictly speaking, it is not the only type of Byte (there exists the **“Nybble”**, or half-Octet, or HexDigit, which is a 4-bit Byte) but it is the Standard Unit Type.
To make up for this length issue, Bytes are abstractly grouped into **Compound Bytes** that address certain attributes of the cell.

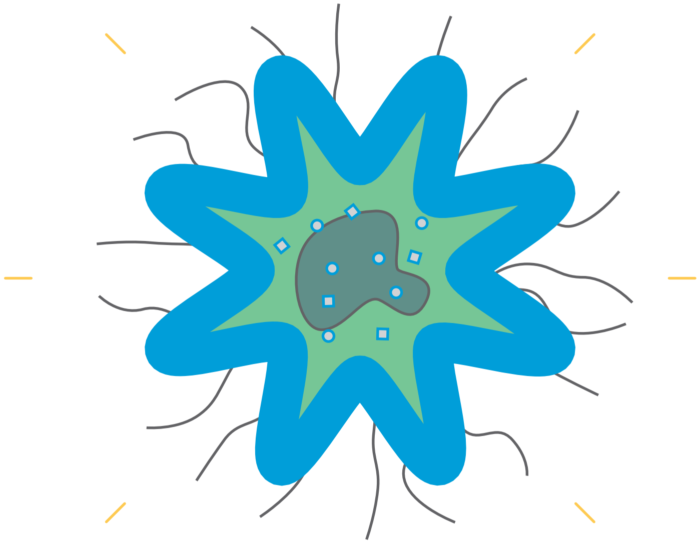


In The Game of Cells, the genotype is encapsulated in a Class simply called “Genes”, which also contains decoding information and useful methods for Bit manipulation (done via the Java BitSet Class).

Gene Decoder			Attributes (Genes)	
Slot	Key		☐ Physical ☐ Personality	
0	0		Cell Size	
1	1.0		Cell Shape (Johan Gielis SuperFormula parameters)	A
	1.1			B
	1.2			N1
	1.3			N2
	1.4			N3
	1.5			M
2	7.0		CytoPlasm Color	R
	7.1			G
	7.2			B
3	10		Wall Width	
4	11.0		Wall Color	R
	11.1			G
	11.2			B
5	14		Wall Width Variation	
6	15		Cellular Apparatus Count	
7	16		Cilia Count	
8	17		Cilia Shakiness	
9	18		Velocity	
10	19		LifeSpan / Life Expectancy	
11	20.0		Movement	Max Distance
	20.1			Move Timer
	20.2			Wait Timer
12	23.0		Attributes	Sociability
	23.1			Love
	23.2			Reproductive Power
	23.3			Hate
	23.4			Brute Force
Total Genes: bytes	28	x8	224 bases bits	

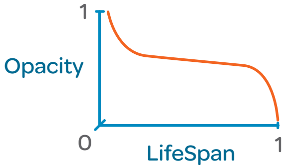
Phenotype

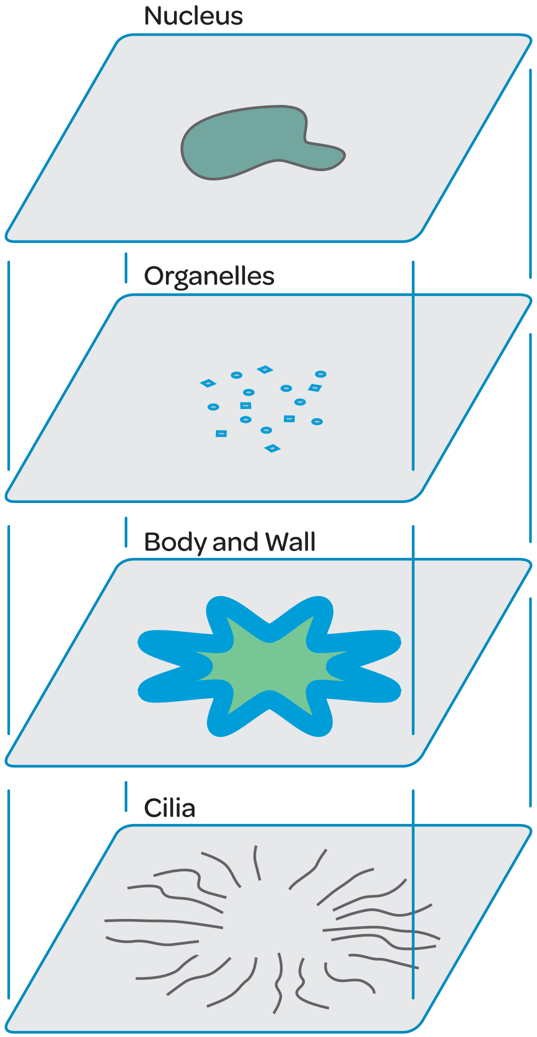
The Genotype is decoded and translate to visual and animation attributes, it's **Digital Phenotype**.
And because not all attributes can be user-controlled and some randomness is added, no 2 cells will be identical.
These are organized in diferent layers, which, in code, are subClasses of a Layer class.



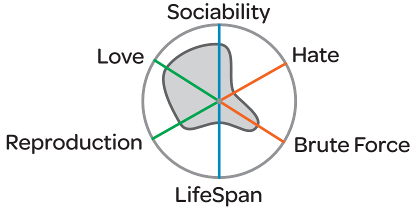
Some attributes translate to the motion of the Cell. For instance, the various Timers in the Gene Table control the translation phases. This, in turn, affects the chances the cell has to come into contact with other cells.

The LifeSpan attribute is in charge of visually decaying the cell: It's colors will gradually lose opacity according to the following curve:





The Nucleus shape is drawn by placing a shape's nodes over a radial line graph.



Miscellaneus shapes floating about the cytoplasm.
Contributes to visually bring the cell to life.

The size roughly determines the active area in which cells come into contact.

Bezier lines.
The nodes oscillate by alternating sine and cosine functions.
Contributes to visually bring the cell to life.

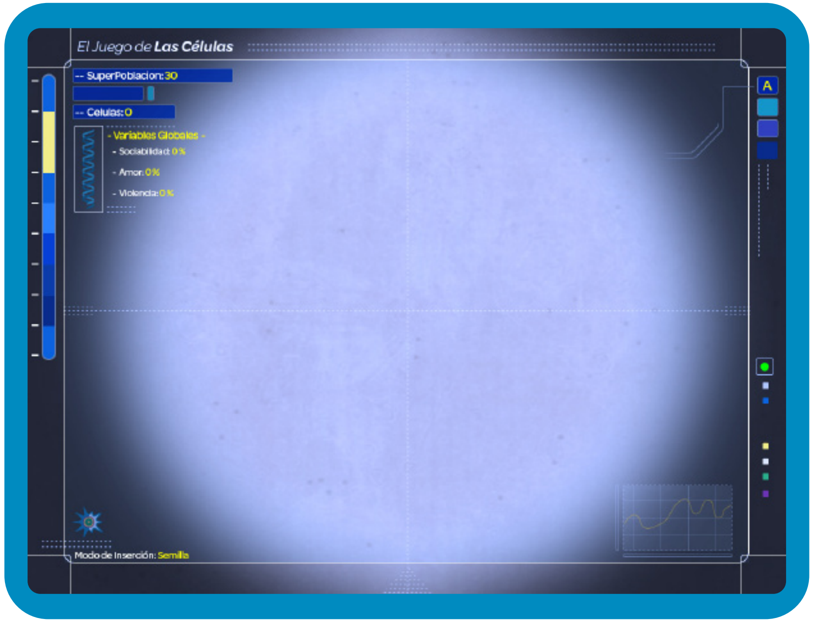
Digital Simulation - Canvas and Rules

Canvas

Station 2 is the simulation canvas.
It's a virtual microscope eye-piece, with the ability of giving birth to new cells via a user interaction.

It also shows a friendly HUD where global simulation variables are display, such as OverPopulation Warning, Cell Count, Cell Insertion Mode, among others.

The simulation starts with no cells. The users are in charge of filling the world with cells but, in case the input is too little over a certain time window, an automatic generation process is triggered, creating cells with no nameTag.



Rules

Cells go about, moving, waiting, shifting, veering, resting, drifting, gliding.
It looks like they are totally free in their medium, but in reality, the simulation follows various rules. These are some of them:

- Cells will only engage in interaction if they are going through their WaitTime, and not when they are drifting.
  This means that if a cell is genetically predisposed to drift a lot, with a short wait time, it is more likely to miss interaction events, which might also mean not reproducing, leaving fewer offspring.
- Interaction is not permitted with a direct member of the genetic family: Father, Mother and Brothers.
  Every cell is assigned an ID. At the moment of birth, the cell stores it's parent's IDs, and this information is used to check direct inBreeding.
- Cells will not engage with more than 1 cell at a time.
  Bullying and orgies are not permitted.
- OverPopulation will idle all interactions.
  An overPopulated society usually means that there are not enough resources to assign to all members, therefore decreasing quality of life, giving free way to death. This rule is an overSimplified abstraction of population control, thus serves well to avoid uncontrolled reproduction.
- Cells will tend to drift inside the microscope active area.
  We want them to be free, but at the same time be able to watch them. Just like a Zoo.

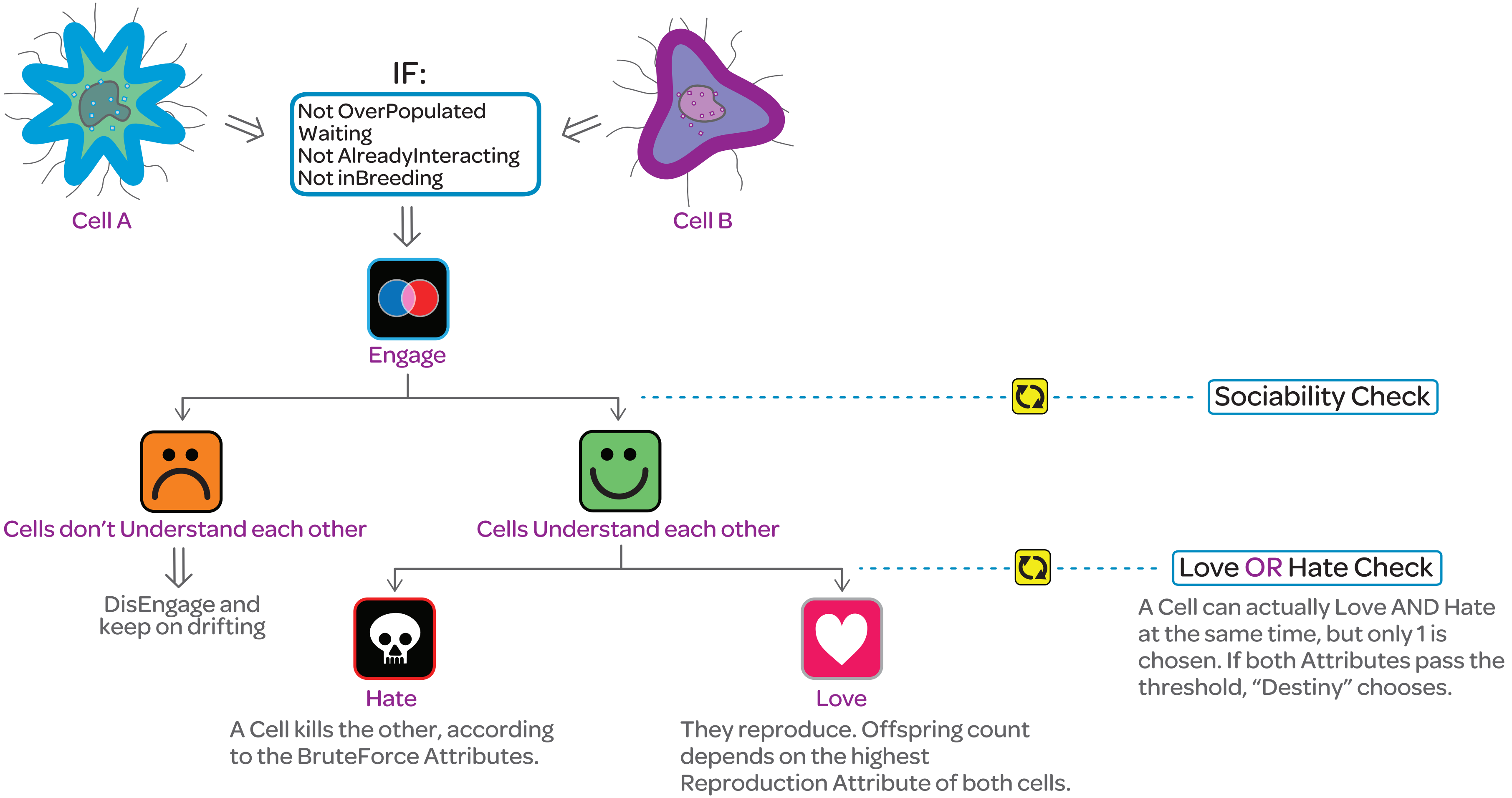
2-Cell Engagement

Once two cells effectively engage in an interaction, there's a simple tree of decisions they follow to end up either reproducing, or killing themselves.

The decisions are based on the Personality Attributes of each cell, and are compared with simple probability, to decide events or come to a conclusion.

Therefore if two cells have a Sociability value of 95%, it is very likely they will both want to engage.
If, by a mere chance (the odds of life), they don't, the cells just disEngage and continue drifting about.

This decision process is repeated along the tree, with Love-Hate and Reproduction-BruteForce, respectively.



Cell A

IF:

Not OverPopulated
Waiting
Not AlreadyInteracting
Not inBreeding

Cell B

Engage

Sociability Check

Cells don't Understand each other

DisEngage and
keep on drifting

Cells Understand each other

Love OR Hate Check

A Cell can actually Love AND Hate at the same time, but only 1 is chosen. If both Attributes pass the threshold, "Destiny" chooses.

Hate

A Cell kills the other, according to the BruteForce Attributes.

Love

They reproduce. Offspring count depends on the highest Reproduction Attribute of both cells.

Reproduction

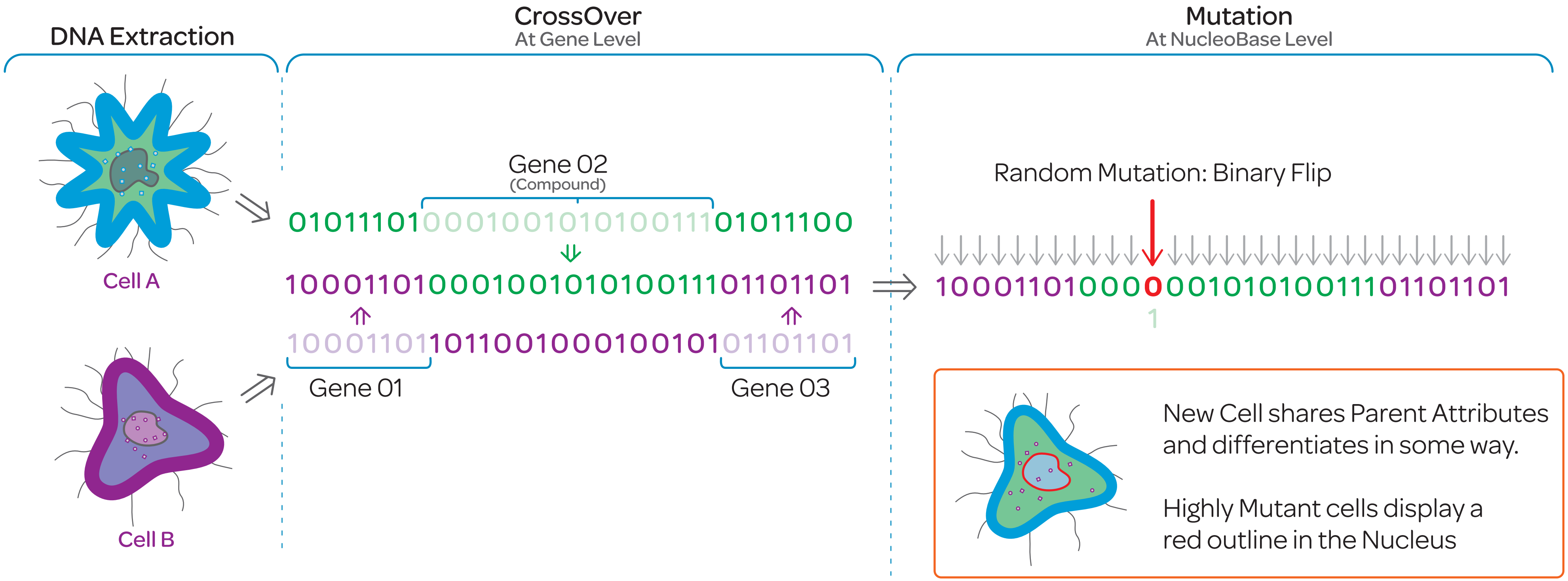
The Reproduction Stage is one of the most important stages of the simulation.
It is where the genetics are more present, and where the previous modeling of a cell starts to express itself in terms of the simulation.

Offspring possess a shared amount of genes coming from the parents, and this inheritance is visually clear.
For example, a child might inherit the shape of the father and the colors of the mother. This results in a cell which is different, but very similar to it's parents, and can graphically be traced back.

The following genetic processes take place, which are an analogue simplification of natural processes:

After DNA Extraction, Gene CrossOver takes place. Apart from the genes themselves, the Gene Package with the decoder, which knows how to properly select the meaningful portions of the Strand. Although Compound Genes can be stripped and subSelected, this might lead to very different variations that did not originate from a mutation.

The Mutation Stage doesn't really care about genes at all. And that's the fun of it: it inserts Bit-flippings so randomly across the strand, that a single mutation can cause a huge variation in the offspring, or have almost no discernable effect. It all depends where that mutation lands. For example, if we were to insert 1 mutation on 1 Gene (1 Byte), flipping the last binary digit will cause a miniature value change, while flipping the first binary digit will mean an enormous change.



DNA Extraction

CrossOver

At Gene Level

01011101000100101010011101011100

10001101000100101010011101101101

10001101101100100010010101101101

Mutation

At NucleoBase Level

Random Mutation: Binary Flip

10001101000000010101001110110 1101

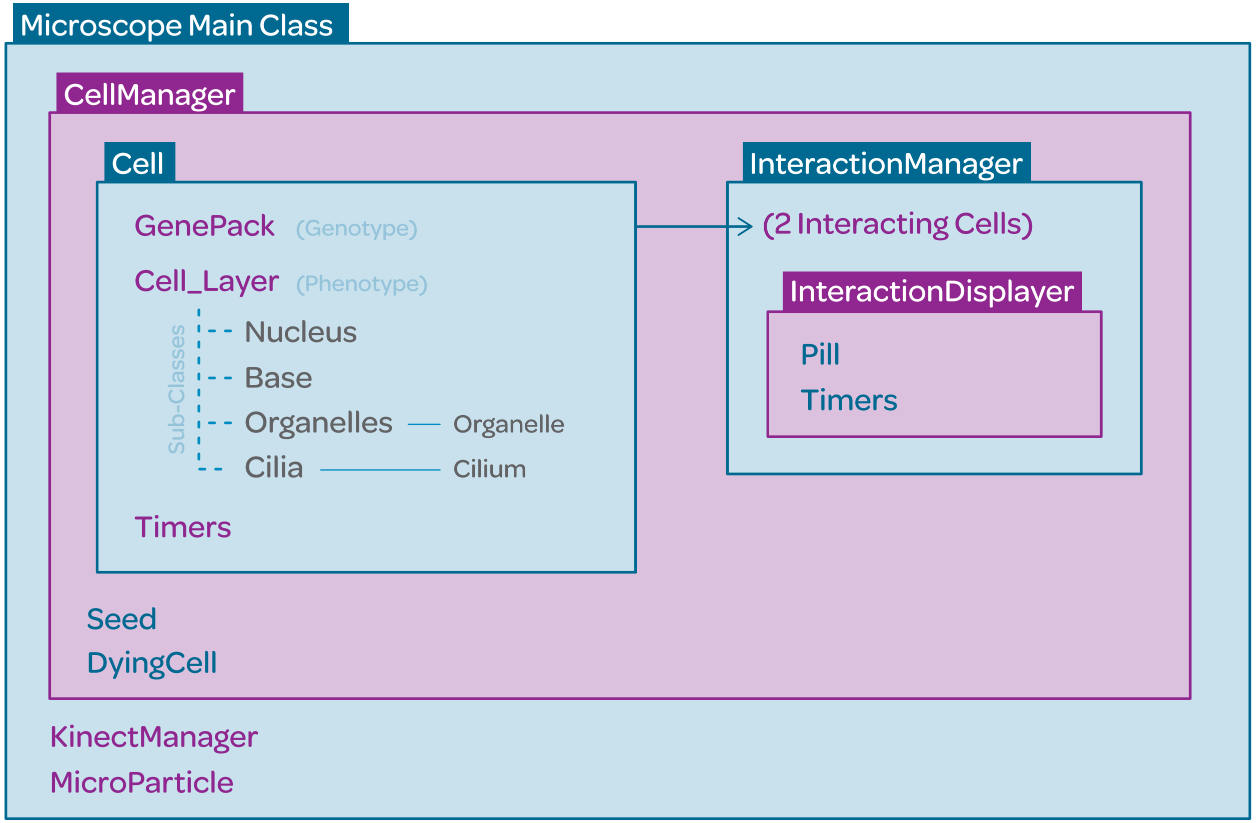
New Cell shares Parent Attributes and differentiates in some way.

Highly Mutant cells display a red outline in the Nucleus

Code - Java Classes PseudoStructure

Articulating the simulation meant a clear hierarchy and class communication. The following graph shows a hierarchical structure of the Application's classes. All names are classes, except what is written between parenthesis.

Code was written in Java, using the Processing Library for graphics, and other libraries written for Processing.



Microscope Main Class

CellManager

Cell

GenePack (Genotype)

Cell_Layer (Phenotype)

Sub-Classes

- Nucleus
- Base
- Organelles — Organelle
- Cilia — Cilium

Timers

InteractionManager

(2 Interacting Cells)

InteractionDisplayer

Pill

Timers

Seed

DyingCell

KinectManager

MicroParticle

Bear in mind that there is a lot of class exchange:

For example, a Cell can live on it's own inside the CellManager, but when it is interacting, both cells get passed over to the InteractionManager, which takes control of them until the interaction is finished. If there are no 2 cells interacting, an instance of this class would not be created.

The same thing happens with the GenePack. Genes are read everywhere from Seed, to the CellManager, onto the Cell itself, and scrambled around in the InteractionManager.

The Seed contains the incoming User-Generated Attributes from the Modeler Station, and does not update until a new Seed is received.

The MicroParticles are tiny elements that simulate the dusty organic granules floating about when observing down a microscope.

The KinectManager is a class that handles all Kinect Sensor input, and delivers a Calibration GUI to properly map and swap 2D and Depth Data.

:04

The Experience

The Game of Cells was first exhibited at Calos Lab, on December 14, 2012, as part of a Tech Expo, where Pleek presented with other visual and audio exhibitors.

Calos is a Social Welfare Foundation of the Korean Community in Argentina, that helps in the treatment of children with mental disabilities. Calos Lab has been merging New Technologies inside their activities for 3 years now, and puts on an Exhibition every end-of-year.

The installation did not only composed of The Game itself, but framed a complete "Scientific" section of the Expo, where various lab toys were available to children, along with versions of the Rubik's Cube, including a digital one.

The Experience & Press

At the start of the evening, people were somehow shy over the different ways of interaction that The Game provided.

Like it usually happens, the first ones to freely interact were the children, who discovered that it was very easy, and even found new ways to engage The Game, ways that were not thought of, or tested, before the exhibition.

Adults slowly approached the first Station, and after naming and finishing their cell, they were not able to insert it into the simulation themselves, as children were packed in front of the Microscope, absorbed in a sort of Cell-Insertion frenzy. This dynamic continued throughout the whole evening.

A very important event occurred during the evening, at repeated times. It was important because it represented a milestone for one of the objectives of the installation: to ignite the scientific-learning spark on the user/child.

I was dressed as the Mad Scientist behind the Lab, and guided the show. At some point or another, a child would approach me, with awe-filled eyes, and a thirst of curiosity running through the veins. The first time it happened, it made me so happy. She was a little girl, of about 7 years of age. She came to me, grabbed my hand, and asked me, “What are the cells? How do you make them?”.
I then realized the installation was working, reaching and creating an effect on the children.
Of course, it was very difficult to answer the question, and even more difficult was to correctly interpret it. But what left me calm and content was the fact the child's parents were there, and they improvised an appropriate answer that only made the child more curious.
The adult's concerns revolved around the workings of the installation and simulation.

사회복지재단 깔로스 세 번째 전시회

사회복지재단 깔로스(CALOS, 대표 사회복지사 이주희)의 3회 전시회가 14일(금) 개최됐다.

Press release of The Korean Network in Argentina

http://www.kornet24.com/kornetnews/180772

Thanx to:

Idea, Graphic Design, Electronics, Software Programming and Implementation:

Agustín Ramos Anzorena

Coded in Processing

Thanx To:

Calos Lab, Perla Lee and everyone at Calos.
The Venue and it's Director/Organizer.

Juan Pablo MacDougall (http://ofmac.blogspot.com.ar), Lautaro Castillo, Diego Paramo, Daniel Hong and all the exhibitors.
The Exhibitors.

Guido Corallo (http://www.gcrll.net/) and his Students.
The Processing Workshop where this Installation took off.

Patricio Gonzalez Vivo (http://www.patriciogonzalezvivo.com)
For igniting our bond with Calos Lab, and for his nerdy programming workshops.

My Family.
Sometimes they don't know what the heck I'm doing, but they still have faith in me.

Guillermina Inés Ortega.
For loving me and letting me populate our small house with all sort of strange gadgets.

Manuel Lopez Lecube.
For sharing his Biology knowledge.

Miguel Ángel Morkin.
For his friendship and valuable support.

Continente 7 (http://www.continentesiete.com).
For “Hydra” and support.

Sabrina Mayra Dubovsky.
For photography.

The Libraries Contributers:
Casey Reas, Ben Fry and the Processing Development Team
Max Rheiner, for Simple-OpenNI.
Benedikt Groß, for ANI.
Andreas Schlegel, for ControlP5 and OscP5.
David A. Mellis, for Arduino Library.

pleek

www.pleek.net
www.pleek.net/blog
play@pleek.net