WARSAW UNIVERSITY OF TECHNOLOGY

From HTML to PostGIS

Faculty of Mathematics and Information Science

GIS of Life

- Final Documentation -

By **Elie SAAD & Tomasz Karwowski**

January 22, 2021

Contents

1 Project Description

The project will work towards creating a simulation of an ecosystem. Imagine a population of computational creatures swimming around a digital pond, interacting with each other according to various rules.

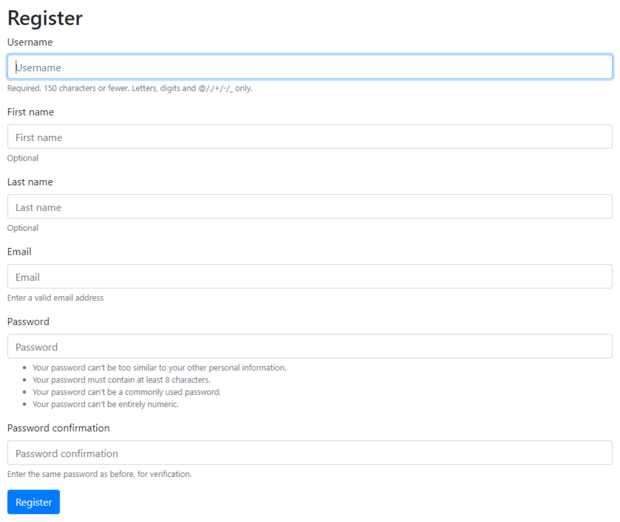
The project is an evolution simulation web application that uses Python as the backend and HTML, JavaScript, and CSS as the frontend to serve as visuals. It uses neural networks and evolutionary algorithms in order to simulate creatures evolving over time in a certain user specified ecosystem.

The heavy computations are handled by the backend and only the current state of the map and it's creatures is presented on the frontend.

2 User Manual

The game is realised in a form of a web application, where users can

- create an account,



Register

Username

Username

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name

First name

Optional

Last name

Last name

Optional

Email

Email

Enter a valid email address

Password

Password

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation

Password confirmation

Enter the same password as before, for verification.

Figure 1: User registration form.

- with all obligatory functionality of login/logout,
- password recovery,
- and create unlimited number of games.

In game creation view, user specifies game name, map size and map terrain by choosing an area of real word map in an Open Layer widget.

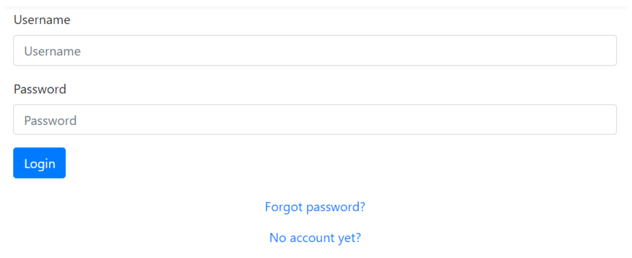


Figure 2: User login form.

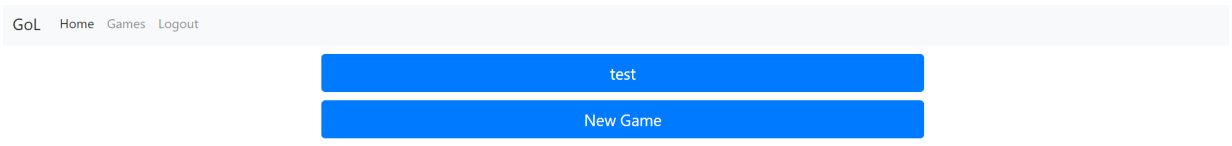


Figure 3: Games list form.

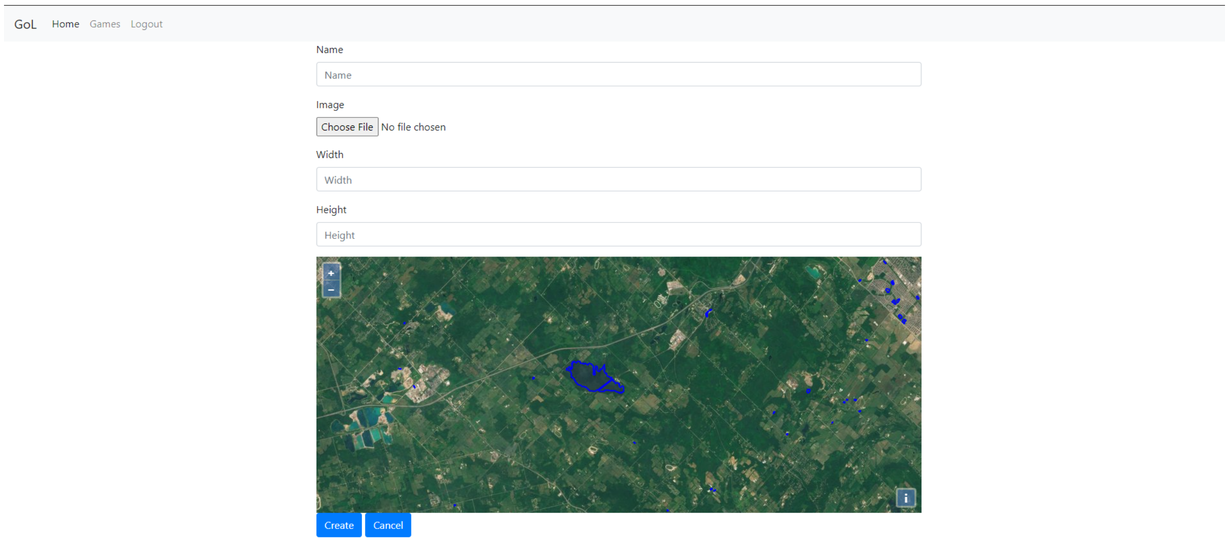
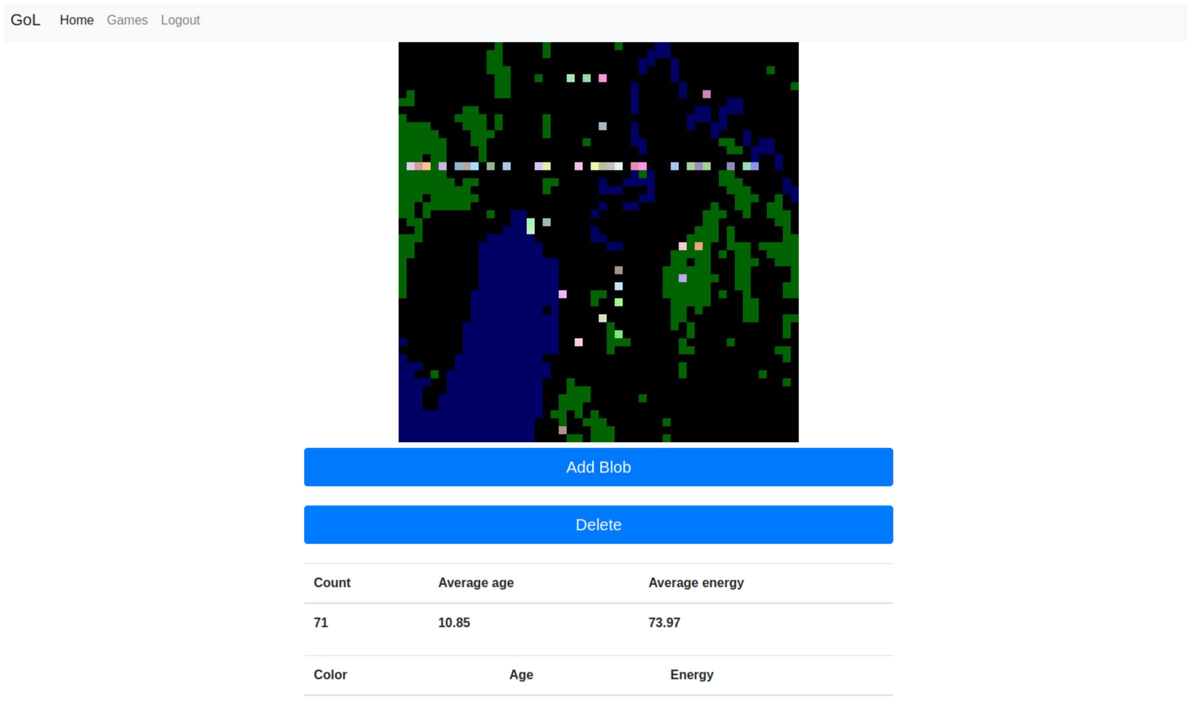


Figure 4: Game creation form.

Once that's done they can enjoy the view of their simulation and potentially in later version also see general stats about the simulation as well as information about specific blob organisms by clicking on them. User can list all his games in the Game List View delete the game at any time by clicking a button on the specific game page.

During the game, the user is able to add new Blobs to the already existing ones that are living on the created user map.



Count	Average age	Average energy
71	10.85	73.97

Color	Age	Energy

Figure 5: Game form.

3 Game Rules

The game consists of a map, on top of which a life-like simulation takes place. The user begins by selecting the map that they desire. The map selection is done through the world map presented to the user during pregame options selection. The map that the user selects shows the rivers, lakes and seas which will be converted to water squares.

After the map's selection, it gets processed, simplified to 3 tiles types and pixelated, and the processed version is the final version that is shown to the user where the life simulation takes place. The map gets processed and divided into squares, where each square adopts one of the three colors:

1. Black square: is a generic square where nothing interesting can happen on it, meaning that it holds no resources and is not fatal.

2. Green square: is a square which holds food.
3. Blue square: is a water square which penalizes the movement cost.

4 Blob Rules

A single Blob can make the decision (on its own) to do one of the following:

1. move upwards,
2. move downwards,
3. move to the left,
4. move to the right,
5. give birth to a new Blob,
6. eat,
7. or do nothing.

A Blob's lifespan is dictated by its energy level. Different actions done by the Blob requires different amounts of energy to be spent. The maximum amount of energy that a Blob can have is 100. If the Blob reaches an energy level of 0, the it dies.

Movement takes 10 energy on land and 20 energy on water, as swimming is harder.

Blobs are parthenogenetic, and therefore do not require another Blob to give birth to an offspring. The birthing act reduces the parent Blob's energy by half. The offspring Blob is birthed with the maximum amount of energy and can immediately decides to give birth if it so desires.

The only way for a Blob to restore energy is by eating. Eating grants the said Blob with 50 energy. Though for a Blob to eat, it must be on top of a green square, otherwise the Blob cannot eat and the energy level remains unchanged.

The do nothing action is the only action that does not cost any amount of energy for the Blob, and therefore executing it leaves the Blob with the same amount of energy that it had prior to the action's execution.

5 During the Game

When the game starts, the user will get the selected map from the previous game creation menu with a select number of Blobs populating its surface.

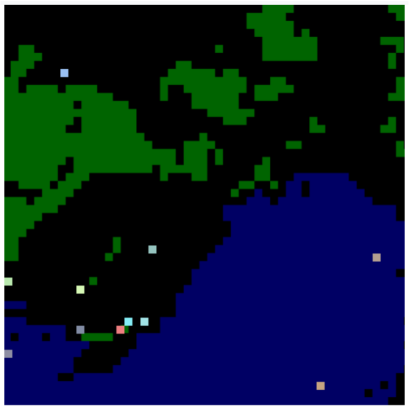


Figure 6: Game running where Blobs live and populate a selected map by the user.

Each one of the brightly colored pixels in the figure 6 is either a single or multiple Blobs occupying the area pixel underneath.

Each pixel from the map itself is represented by three distinct dark ranged colors: green which represents a food food populated tile, black which represents an empty tile, and blue which represents a water populated tile.

When a Blob decides to eat and it is occupying a food tile, the Blob replenishes its energy to full, which is one hundred energy points. When a Blob decides to eat and it is not occupying a food tile, the Blob does not get affected and its energy level remains the same as it was before making the decision.

When a Blob decides to move to an area that is considered as a water tile - meaning a blue area - the Blob loses twenty energy points. And if the Blob goes under the minimum energy limit - which is zero - the Blob immediately dies. Otherwise, the Blob goes on to make its next decision.

When a Blob decides to move to an area that is considered as a land tile - meaning a black area - the Blob loses ten energy points. And if the Blob goes under the minimum energy limit - which is zero - the Blob immediately dies. Otherwise, the Blob goes on to make its next decision.

The Blob can decide to give birth in any tile, but by doing so the Blob gets split into two, and the parent Blob uses half of its energy level to give birth to a child Blob that is born with the maximum energy level of one hundred.

6 Model Architecture

The model is made out of a deep neural network. There are 9 input neurons, the features that were selected are as follows: x position, y position, top square type, bottom square type, left square type, right square type, energy level, and a memory neuron. The memory neuron will be used in the evolutionary algorithm later on. The square types are as follows: -1 is given to a square of a blue color, 0 for the square with a black color, 0.5 for the square with a green color, and 1 for a square with other Blobs present on it.

A sigmoid function is applied on the nine inputs which are connected to nine neurons in the hidden layer. The sigmoid function was chosen because it forces the inputs to all be of positive values between zero and one exclusive.

And finally the nine neurons in the hidden layer are connected to eight neurons in the output layer with no additional functions applied to them. The eight output neurons describe the decision of the Blob. They have the following meanings in the exact following order: move to top, move to bottom, move to left, move to right, give birth, eat, and remember (which is basically a do nothing action that relates to the memory neuron in the input layer).