ALGAE
Programmierpraktikum II SS 2000

Clemens Bauer
cbauer@cosy.sbg.ac.at
Peter Eisenmann
peisen@cosy.sbg.ac.at
Thomas Frommwald
tfromm@cosy.sbg.ac.at
Daniel Kadir
dkadir@cosy.sbg.ac.at
Michael Liedlgruber
mliedl@cosy.sbg.ac.at

Project manager: Roland Schwaiger

June 21, 2000

Contents

Chapter 1

Introduction

We will give an overview of all relevant aspects of the project `Algae` without going into too much detail. Take a look at the following chapter for further and more detailed information.

1.1 What is ALGAE

ALGAE is a followup system of elc (evolution, learning and communication), which is an implementation of a multi agent svstem (MAS) conceptualized and implemented by Roland Schwaiger and Konrad Lang. They investigate the evolution of wandering behavior of simple, reactive agents in a highly dynamic environment. Those simple agents wander based on the outcomes of feed forward ANNs (multi layer perceptrons). This system is implemented basing on the algae *Logopodium.*

1.1.1 Parameters

As a basic rule, each single parameter used for the simulation, the world or the agents, can be set by the user himself.

1.2 World of Algae

Our world where the simulation takes place is an two-dimensional aquarium. On the top resides a uniform light source and at the bottom there is some nutrition substrate. Algae (agents) live in the world between.

1.2.1 Light source

The light source is completely uniform, rays are sent from top to bottom in a straight line. The light source can be set to simulate a day/night cycle, but may also be set to a specific value during the simulation, or even turned off completely by the user. Light has a higher intensity at the top of the aquarium than at the bottom. Per default,the intensity is modelled by the function 1/y, where y represents the vertical world axis, but an other function may be defined by the user.

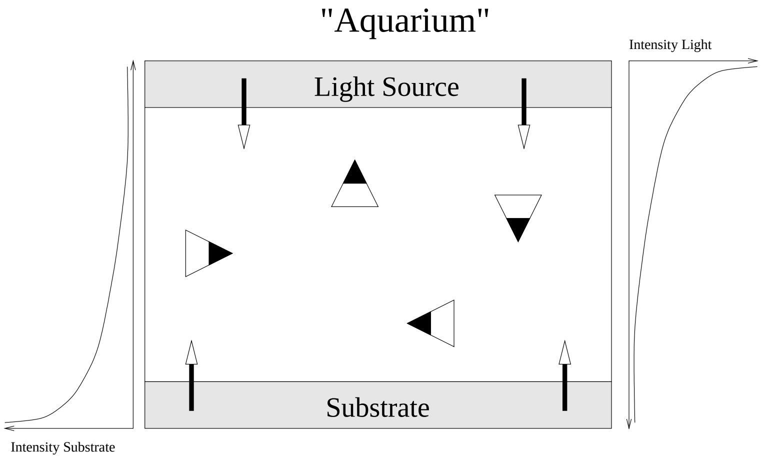


Figure 1.1: The world where our simulation takes place.

Agents reduce the light intensity for every spot below them, but they do not block the light completely due to a transparency value.

1.2.2 Substrate

Similarly conditions apply for the substrate which diffuses into the water. The distribution of substrate is based on a mathematical function. If a substrate particle is consumated, it is rebuilt after a specific amount of time. At bottom, substrate is rebuilt faster than above, because the source resides there.

1.3 Agents

Our agents, the algae, do not have a fixed life span. They live as long as they got an amount of energy greater than zero.

1.3.1 Energy

Every agent is “born” with the same amount of energy. Energy is used for every action the agent may perform. If the energy level reaches zero, the algae dies. Energy may be regained through photosynthesis or substrate accomondation.

1.3.2 Sensors

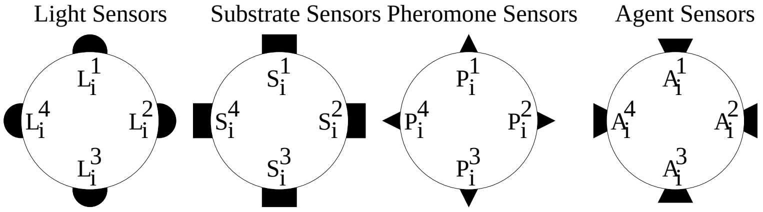


Figure 1.2: The various sensors of agents a_i.

To enable the agents to get information about their environment, we equip them with several sensors. An agent got four different sensors, each divided into four parts which point to the main directions forward, right, left and backward. Therefore, the agent gets 16 different sensor inputs every timestep.
All sensor parts have a certain field of view (FOV) and a specific range. Due to the fact the strength of the input values from the various positions of the sensor parts differ, the agent should be able to detect where the source of signal is.

Light sensor

This sensor gets the summed light intensity of its FOV within its range.

Substrate sensor

This sensor gets the value of the substrate particles in its FOV. The value each particle adds to the sensor input depends on its distance.

Pheromon sensor

This sensor delivers the pheromon density in its FOV. The value each particle adds to the sensor input depends on its distance. Pheromons are released by the agents in order to find reproduction partners.

Bumper sensor

This sensor detects any obstacle (other agents or the world's border) around the algae.

1.3.3 Actions

An agent is able to perform an arbitrary amount of *different* actions every timestep of the simulation. All of these actions consume a different amount of energy. Of course, the agent won't perform actions that would use a higher amount of energy than it currently has.

Movement

Moving is probably the most important task an agent has to perform, in order to navigate to reproduction partners, towards substrate or higher light intensity. The agent may move once per timestep, with abritrary speed (up to its maximum speed). Movement with higher speed consumes much more energy than slow movement.
The agent always turns towards the direction it is moving - it cannot make some kind of sidestep.

Photosynthesis

One way for an agent to get energy is photosynthesis. The agent directly converts the light intensity at its current location into life energy. Higher light intensity produces a higher amount of energy.

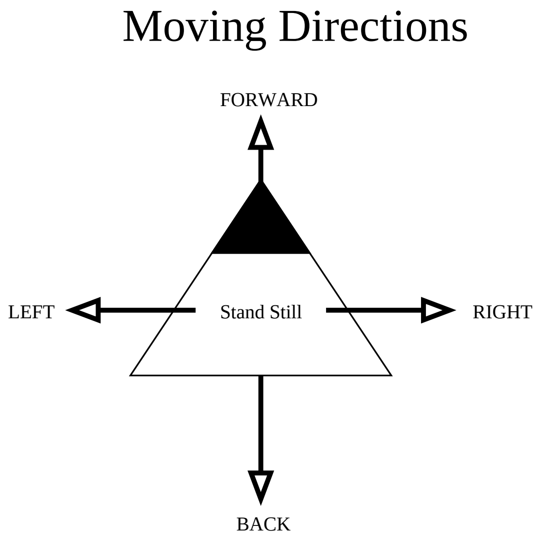


Figure 1.3: The possible moving directions of agent a_i.

Substrate acommomdation

If an agent is at the same location as a substrate particle, it consumes the particle and regains a specific amount of energy. The substrate is rebuilt after a certain number of timestep, depending on the y position.

Reproduction

If the energy level of an algae reaches a certain amount the algae may reproduce either sexually (meiosis) or asexually (mitosis).

- Meiosis, Pheromones
  Once an agent has collected enough energy it may reach a critical limit for sexual reproduction. From that moment on the agent may produce pheromone in order to signal other agents its ability for sexual reproduction. The agent will release the pheromone in its actual location. The pheromone will be apparent in the spot for given amount of timesteps.
  If one agent releases pheromone and meets another agent it will force the other one to reproduce, without taking care for the other's current energy level.

- Mitosis
  If an agent reaches a certain energy level before it finds a partner for meiosis, it reproduces itself asexually through the way of mitosis. This means the agent parts itself so we get two nearly identical individuals (they do not have exactly the same genotype because there is a certain mutation rate).

1.3.4 Neural Network

The various actions of an agent are partially determined by its mapping from input of the environment to output, i.e. reactions. This type of agent may be called reactive agent or stimulus/response agent. In our multi agent system we model this stimulus/response

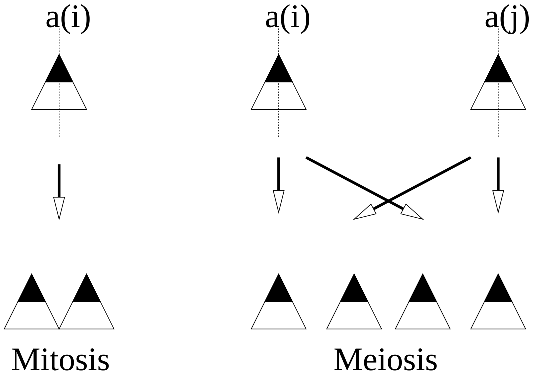


Figure 1.4: Reproduction of the agents a_i and a_j .

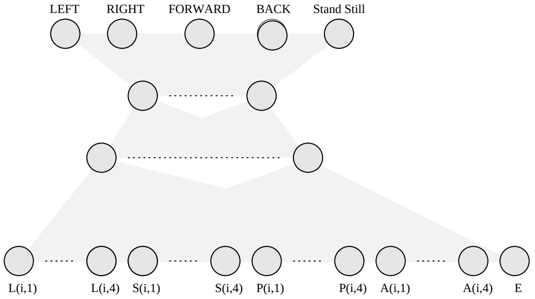


Figure 1.5: The ANN of agent a_i .

relationship by means of an artificial neural network – multi layer perceptron – (ANN). The input to the ANN consists of the sensor inputs and the energy level of the agent. Various hidden layers are used to perform transformations of the signals. Finally at the output level the calculated moving directions are shown.

Chapter 2

Deeper view to the problem

In this chapter, we want to deepen the understanding of the given task and concentrate on relevant parameters and notations of the problem under discussion.

2.1 The Simulation

Our task is to provide a simulation environment that sets the input to the ANN and performs the actions, shown by the output. This also should take care of reproduction methods and the death of an agent when out of energy. At last the simulation should be visualized.
At the beginning a certain amount of agents is randomly initialized. This includes the position and the heading as well as the neural network associated with each agent. This is the point, where the aquarium (the concrete matrix) gets initialized too. This includes the substrate distribution and the light, which are both set to a starting value. The pheromone level is set to zero at each point in the matrix.
After this global initialization, the main simulation loop is started. In that loop each agent performs some actions, which result from influences of its neighbourhood.

2.2 World

Basically, the world is responsible for controlling the experiment, some of its tasks are:

1. Set the output of the experiment in the GUI, i.e. show the agent, light distribution, substrate distribution, show details of the agents, s.a. energy level, genotype, phenotype.

2. Perform all calculations from one timestep t_i to timestep t_{i+1}, new agent positions[1], set sensor values of the agents[2]

---

[1] The agents perform their steps and request from the world their new positions, since only the world knows about the positions of the other agents. Furthermore, just the world knows about the concentrations (light, substrate, pheromone).

[2] This has to happen in the first phase of intervall $[t_i, t_{i+1}]$ since all further actions of the agents rely on the sensors values.

3. Insert new agents in the world once reproduction has taken place.
4. Remove agents with an energy level of zero (or below).

2.3 Aquarium

The aquarium has a two dimensional extension, specified by $(worldx, worldy)$. The borders on the left and right have a transparency factor of zero, i.e. no light may shine through the borders.

2.4 Light and Substrate

Let us denote the light distribution by $d_L(x)$ and the substrate distribution by $d_S(x)$. In the discussion above we have given an example for the distributions, namely $d(x) = \frac{1}{x}$. This is just one example and may be set by the user out of a given set of functions.

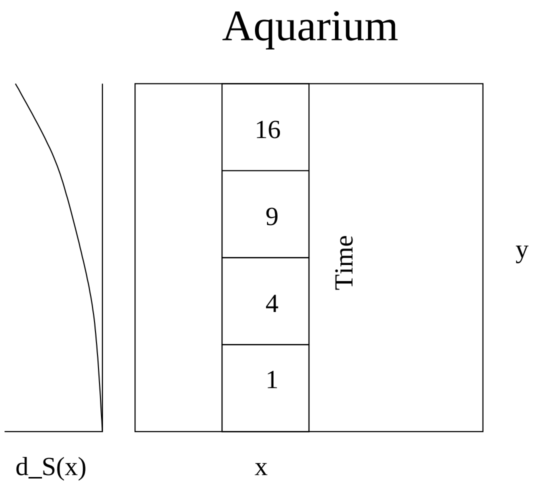


Figure 2.1: The concentration and distribution of the substrate, according to $d_S(x) = x^2$.

In figure 2.1 the distribution of the substrate is given. In each location (x, y) the substrate has concentration 1 or 0. Once a substrate is consumed by an agent the substrate will be rebuilt according to the given $d_S(x)$. In the given figure we have chosen $d_S(x) = x^2$, for $y = 3$ we get $d_S(3) = 9$ which are the timesteps the system needs to rebuild the substrate concentration[3].

Next let's consider the light distribution $d_L(x)$. Consider that an agent a_i in spot (x, y) blocks the light for all the agents a_j in locations $(x, y'), y' < y$. The light intensity will degrade according to a certain transparency factor τ_i of agent a_i.

In order to model day and night in the simulation $d_L(x)$ may depend on the time. As an example consider the sin function, as seen in figure 2.2. Further the time dependend modeling of the substrate concentration may be considered. The light intensity can also be changed manually by the user during the simulation.

---

[3] In a future release of the simulation the substrate concentration in (x, y) only may be rebuilt if there is a substrate concentration in locations $\{(x, y')|\forall y' : y' < y\}$

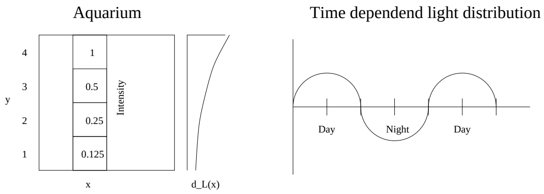


Figure 2.2: The concentration and distribution of the light, according to $d_L(x) = 1/x$.

2.5 Pheromone and Meiosis

How stated before, once an agent a_i has collected enough energy points it may reach a critical limit for sexual reproduction E_s. From that moment on the agent may produce pheromone in order to signal other agents its ability for sexual reproduction. Of course, producing pheromomes will reduce the energy level of the agent, let's denote the reduction ΔE_s. The agent will release the pheromone in its actual location[4]. The pheromone will be apparent in the spot for a given amount of timesteps, let's denote the amount of time by π_t.

For the case an agent has an energy level greater then E_s and it meets another agent we discussed various ways to consider meiosis.

1. To distinguish between the pheromones of different agents we may introduce two genders, male and female. The two genders release different kinds of pheromone. As can be seen easily, we reduce the probability of reproduction significantly.
2. Another way to consider pheromone of different agents we may intro two agent types. The first type just releases pheromone and the second agent type just senses pheromone. Apply the argument above.
3. If one agent releases pheromone and meets another agent it will force the agent to reproduce, no matter if the other agent has an energy level greater than E_s.
4. Finally, we introduce a flag for pheromone for each agent. With this method we get the possibility to differentiate between the pheromone of the various agetns and just agents with energy level greater than E_s may reproduce. Unfortunately not a biological plausible proposal.

In our simulation we will focus on the third possibility.

---

[4] In future releases of the simulation we may consider some distribution function for pheromone release in the neighborhood of the agent

2.6 Artificial Neural Network

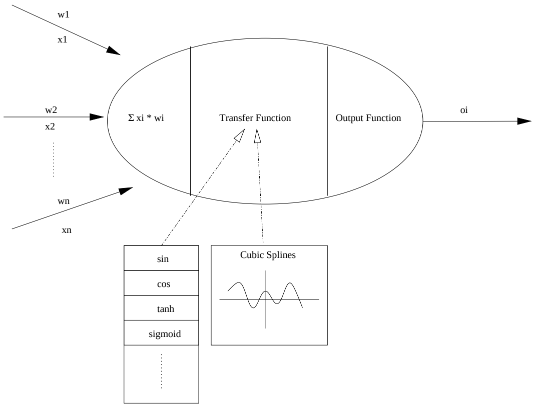


Figure 2.3: The detail view of one neuron in the ANN.

2.7 Evolutionary Cycle and Reproduction

As pointed out above we want to build our system based on the steady state approach, i.e. no generations are built and definite end is proposed but reproduction occurs on the fly. Basically all relevant information has to be encoded into various chromosomes (bitstrings). In our simulation we want to use three chromosomes per agent. The first chromosome C_1 encodes the topology (neurons and connections) of the ANN in a Modified Miller Matrix (MMM) manner. Basically the main diagonal of the connectivity matrix where an entry $m_{ii} = 1$ states that the neuron is present. Whereas entry $m_{ii} = 0$ states that the neuron is not present. If the entry is not in the main diagonal m_{ij}, where $i <> j$, then the entry states the existance of the connection from neuron i to neuron j.

The second chromosome C_2 encodes the weights of the connections and the biases of the neurons, two Bytes are used for encoding. The third chromosome C_3 encodes the cubic splines, where each point of the spline is encoded with its x and y coordinate, two Bytes respectively.

In order to reproduce we have to consider two different cases, meiosis and mitosis. In figure 2.4 we give an example for meiosis. In the mitosis case we duplicate the chromosomes and perform the same steps as in the meiosis case.

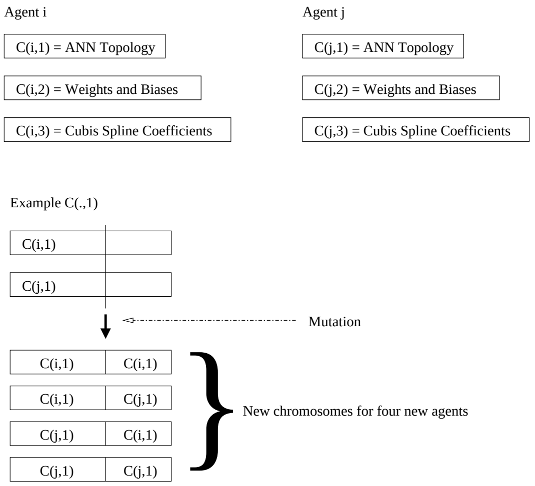


Figure 2.4: The details for crossing over between chromosome C_1 of agent a_i and a_j.

Chapter 3

ALGAE File Format Specification

The format in which configuration information in ALGAE will be saved to disk

3.1 Introduction

The ALGAE-settings for different simulation environment presets are saved using the XML fileformat.
This format was chosen because of the flexibility and the fact, that one doesn't have to read or edit any binary fileformat. Therefore the files can easily be created or modified with any text editor. The restrictions within the creation or modification process of such a settings file are donated in the file *ElementDefinitions.DTD*. This file holds some simple guidelines for creating or modifying the ALGAE-settings and shouldn't be changed unless the format is to be extended or modified.

3.2 Section `Settings`

The settingsblock is the main section and holds all subsections with information about the initial simulation environment. This time only one Settingsblock per file is used, but it is also possible for future development of the fileformat to extend the format in such a way, that more than one settingsblock could be stored in one file.
This time, for each preset (or initial ALGAE-settings) a seperate file is created.

The Settingsblock has the following attributes:

- Version - this attribute stores the fileformat version
- Creator - the program (editor) which created this file (in most cases this would be ALGAE itself)
- Name - the name of the settingsfile
- Date - date and time of filecreation (or filemodification) in the format DD.MM.YY HH:MM:SS

3.3 Section `Simulation`

This section holds some general information about the initial state of a simulation.

- Duration - how many steps should the simulation last
- Width - the width of the aquarium
- Height - the height of the aquarium
- Light_enabled - sets the behavior of the light, can be
    all the time
    never
    function
- Light_intensity - intensity of the lightsource on top of the world
- Light_distribution_func (not implemented yet)
- Light_cycle_func (not implemented yet)
- Substrate_amount - how much substrate should be at the bottom at the beginning of the simulation
- Substrate_renew_rate - how fast consumated substrate will be renewed
- Pheromone_dec_rate - how fast pheromone dissolves

3.4 Section `ALGAES`

This section stores information about the algaes and their initial states.

- Initial_algaes - how many agents should be placed in the simulation at the beginning of the simulation
- Initial_energy - the initial energylevel for an algae

3.5 Section `Sensors`

This section stores information about the algaes' sensors and their initial states.

- Pheromone_fov - the field of view of a pheromone sensor
- Pheromone_range - within that range the pheromone sensor will detect a pheromone particle
- Substrate_fov - the field of view of a substrate sensor
- Substrate_range - within that range the substrate sensor detects substrate

- Light_fov - the field of view of a light sensor
- Light_range - within that range algae is influenced by light
- Bumper_range - within that range the bumper sensor detects an obstacle

3.6 Section `Action`

The attributes of this section store information about the amount of energy which is used respective collected by specific actions.

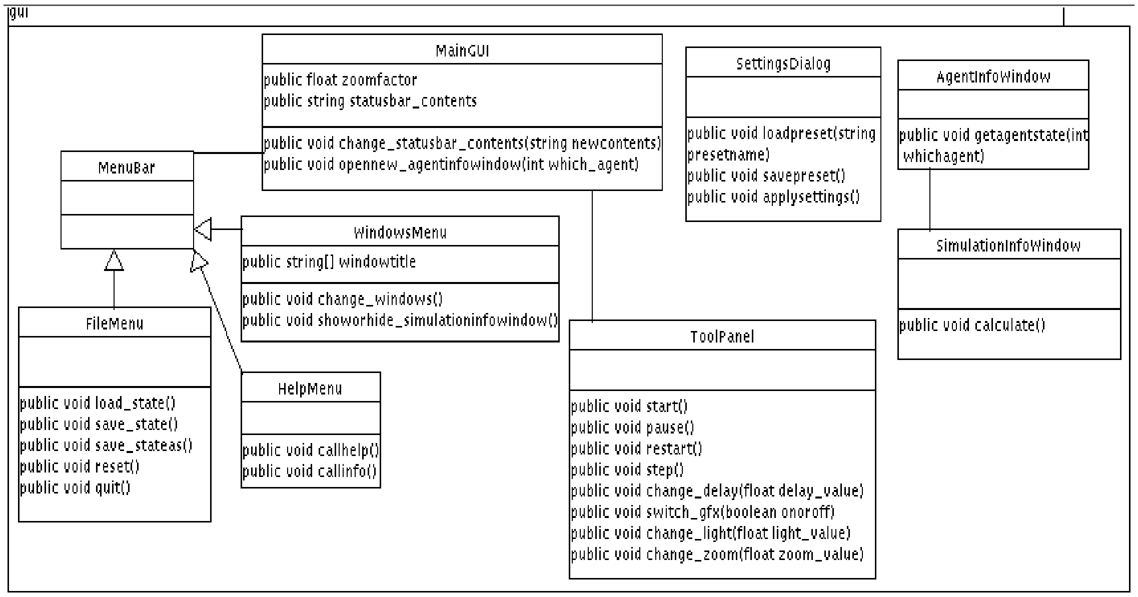
- Energy_movement - amount of energy which is used for movement (slowest speed)
- Energy_mitose - amount of energy used for mitose
- Energy_meiose - amount of energy used for meiose
- Energy_pheromone_output - amount of energy used for releasing pheromone
- Energy_substrate - amount of energy an algae gets by consuming substrate
- Energy_light - amount of energy an algae gets for consuming light

Chapter 4

Class Diagrams and Use Cases

We will discuss the classes and use cases our project will based upon in this chapter.

4.1 Package GUI



This package represents every function that the program's user interface provides. *For further explanation of the functionality of the GUI, see the next chapter.*

4.1.1 Class MainGUI

- Attribute `zoomfactor` (public float) *The zoomfactor of the aquarium output is held in this variable.*

- Attribute `statusbarcontents` (public string) *The string which is currently present at the bottom of the GUI.*
- Operation `changestatusbarcontents` (public void, parameters: string `newcontents`) *The statusbar text is changed via this methode.*
- Operation `opennewagentinfowindow` (public void, parameters: int `whichagent`) *Opens another window with agent-related information. This methode is called whenever a specific agent is selected.*

4.1.2 Class MenuBar

This is an abstract class without any attributes or operations. It consists of several subclasses:

Subclass FileMenu

- Operation `loadstate` (public void) *This Operation loads a previously saved simulation state. A Dialog box opens to determine the filename.*
- Operation `savestate` (public void) *The status is saved, using the actual filename. If the simulation wasn't saved before, a dialog asks for the filename.*
- Operation `savestateas` (public void) *The simulation is saved; the user has to provide a filename first.*
- Operation `reset` (public void) *The currently running simualation is aborted, and the configuration window opens again, so the user may enter new starting values for the sim.*
- Operation `quit` (public void) *This operation stops the program*

Subclass WindowsMenu

- Attribute `windowtitle` (public string) *We identify the different windows currently open using its caption/title (each window gets an unique caption).*
- Operation `changewindows` (public void) *Makes another window the currently active.*
- Operation `showorhidesimulationinfowindow` (public void) *Opens the statistic window if it is not visible and closes it otherwise.*

Subclass HelpMenu

- Operation `callhelp` (public void) *Shows the online-help.*
- Operation `callinfo` (public void) *Shows the program credits.*

4.1.3 Class Tool Panel

- Operation `start` (public void) *Starts/resumes the simulation.*
- Operation `pause` (public void) *Pauses the running simulation.*
- Operation `restart` (public void) *Restarts the simulation using the same parameters as before.*
- Operation `step` (public void) *Advances the simulation one timestep.*
- Operation `changedelay` (public void, parameters: float `delayvalue`) *Changes the speed, respective the delay factor of the sim.*
- Operation `switchgfx` (public void, parameters: boolean `onoroff`) *Turns the complete graphical output on or off.*
- Operation `changelight` (public void, parameters: float `lightvalue`) *This operation is called when the user wishes to change the light intensity during simulation runtime.*
- Operation `changezoom` (public void) *Changes the zoom factor of the aquarium/world output.*

4.1.4 Class SettingsDialog

- Operation `loadpreset` (public void, parameters: string `presetname`) *If a preset is selected from the dropdown list, this methode is called to apply a pre-saved configuration.*
- Operation `savepreset` (public void) *The user specifies a filename where the actual parameters are saved in.*
- Operation `applysettings` (public void) *Closes the configuration window and uses the actual parameters for the simulation*

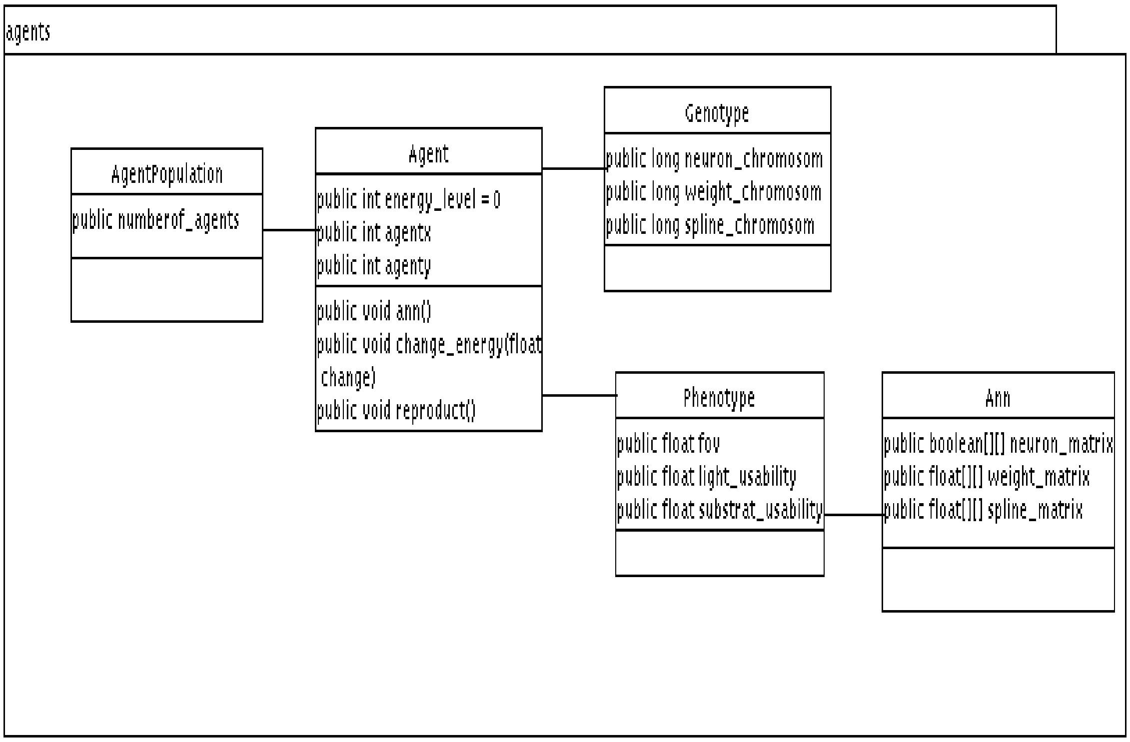
4.1.5 Class AgentInfoWindow

- Operation `getagentenergy` (public float, parameters: int `whichagent`) *Gets the energy level of a specific agent.*
- Operation `getagentx` (public int, parameters: int `whichagent`) *Gets the vertical position of an agent.*
- Operation `getagenty` (public int, parameters: int `whichagent`) *Gets the horizontal position.*

4.1.6 Class SimulationInfoWindow

- Operation `calculate` (public void) *In this function, all statistics given to the user are calculated and output.*

4.2 Package Agents



With this package, the core of the program is built: The agents and their neural networks.

4.2.1 Class AgentPopulation

This is a superclass of the following classes.

- Attribute `numberofagents` (public int) *This variable holds how many agents live in the world at the moment.*

4.2.2 Class Agent

- Attribute `energylevel` (public float, default = 0) *The energy level of the agent.*
- Attribute `agentx` (public int) *The horizontal position of the agent.*
- Attribute `agenty` (public int) *The vertical position.*
- Operation `ann` (public void) *This operation sets the input values for the agent's neural network and takes the output from the ANN too.*
- Operation `changeenergy` (public void, parameters: float `change`) *The methode raises or lowers the energy level of the agent by the given amount.*
- Operation `reproduct` (public void) *Starts reproduction. If there is another algae at the same location then meiosis is started, otherwise mitosis. Calls the* ***`newagent`*** *operation of the class* ***`WorldController`****.*

4.2.3 Class Genotype

- Attribute `neuronchromosom` (public long) *The coded neuron data of the agent's ANN is saved in this variable.*
- Attribute `weightchromsom` (public long) *The weight matrix is coded in this bit-string.*
- Attribute `splinechromosom` (public long) *The spline matrix is coded in this attribute.*

4.2.4 Class Phenotype

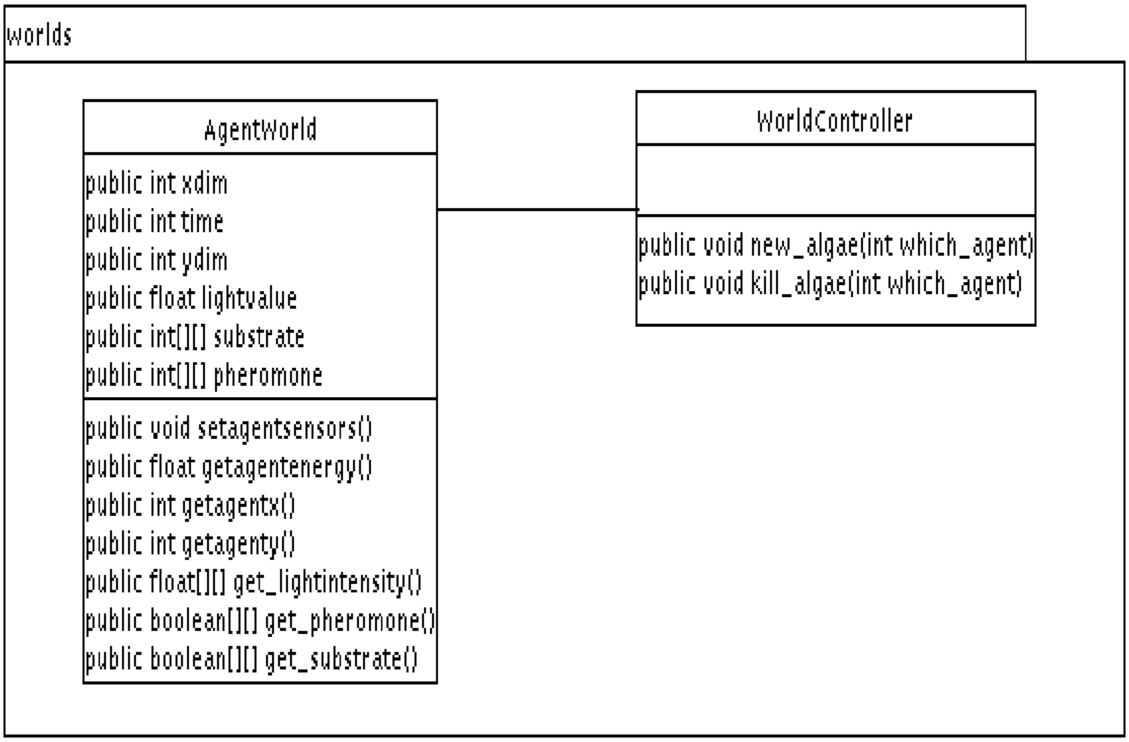
- Attribute `fov` (public float) *The maximum field of view of the agent's sensors.*
- Attribute `lightusability` (public float) *The amount of energy that can be gained through photosynthesis.*
- Attribute `substrateusability` (public float) *How much energy can be gained through substrate accomondation.*

4.2.5 Class Ann

- Attribute `neuronmatrix` (public boolean matrix) *The neuron values are held in this matrix.*
- Attribute `weightmatrix` (public float matrix) *The weights of the connections between neurons are saved in this matrix.*
- Attribute `splinematrix` (public floatm atrix) *The anchor points of the cubic splines are saved here.*

4.3 Package Worlds

The world where our simulation takes place is built with these classes.



4.3.1 Class AgentWorld

- Attribute `xdim` (public int) *The width of the aquarium.*
- Attribute `ydim` (public int) *The height of the aquarium.*
- Attribute `lightvalue` (public float) *The current light intensity at the top of our world.*
- Attribute `substrate` (public int matrix) *The number of timesteps since a substrate particle was consumated. We have to keep track of this number to know when the particle will be rebuilt. The operation* ***`substrate`*** *makes use of this matrix.*
- Attribute `pheromone` (public int matrix) *The number of timesteps since a phermone particle has been discharged. Used by the operation* ***`pheromone`****.*
- Operation `setagentsensors` (public void) *This methode calculates all sensor values for every agent and therefore sets new input values for its ANN.*
- Operation `getagenenergy` (public float) *Reads the current energy level of an agent.*
- Operation `getagentx` (public int) *Gets the vertical position of an agent.*
- Operation `getagenty` (public int) *Gets the horizontal position.*
- Operation `getlightintensity` (public float matrix) *Calculates the light value for every position of our world matrix, based on the variable* ***`lightvalue`*** *and all agents (which cast shadows).*

- Operation `getpheromone` (public boolean matrix) *Checks every position in the aquarium for present pheromone parts. Uses the* ***`pheromone`*** *matrix to check if a pheromone is still active.*

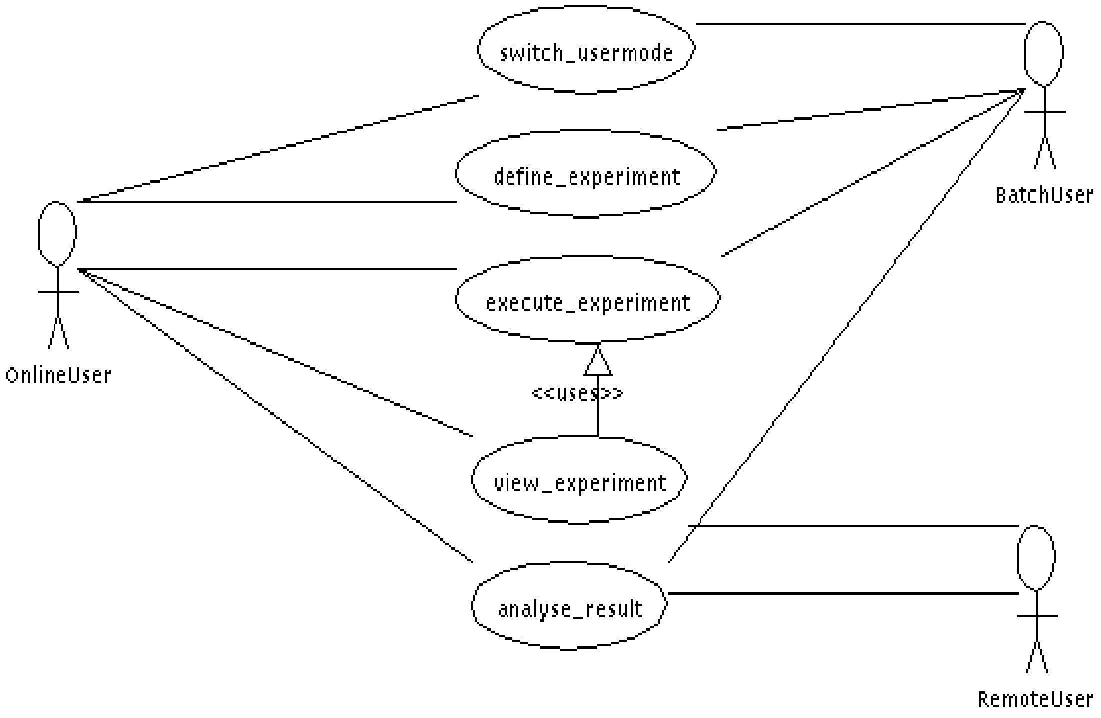
- Operation `getsubstrate` (public boolean matrix) *Checks every position for present substrate particles. Uses the* ***`substrate`*** *matrix to check if a particle should be renewed.*

4.3.2 Class WorldController

- Operation `newalgae` (public void, parameters: int `x`, int `y`) *Sets a new agent in the world after reproduction has taken place.*

- Operation `killalgae` (public void) *Eliminates an agent which is out of energy.*

4.3.3 UseCases Experiment



- Actor `OnlineUser` *This actor represents the "normal" user who defines an experiment, afterwards starts it and is present most of the time to watch and manipulate the running simulation.*

- Actor `BatchUser` *This user defines one or more experiments and starts them. (S)he doesn't want to watch the whole simulation, so no graphical output is needed. After all experiments ended, the BatchUser analyses the results.*

- Actor `RemoteUser` *The remote user cannot define or start experiments by himself. Normally, (s)he just watches the simulations's output (or a non-graphical variation* *of it) and gets the results after the program ended. Normally, the remote user's connection to the simulation will be via the WWW.*

- UseCase `defineexperiment` (accessible for online- and batch-users) *The user sets the configuration for one or a number of experiments.*

- UseCase `executeexperiment` (accessible for online- and batch-users) *Starts the defined experiment(s). If more than one experiment has been specified, they are processed one after another.*

- UseCase `viewexperiment` (accessible for online- and remote-users) *This use case represents the output of the graphical simulation or eventually a textual variant (for the remote-user).*

- UseCase `analyseresults` (accessible for all users) *All actors may analyse the results of the simulation(s) once they are finished. These results have the form of several statistics calculated during/at the end of each simulation.*

- UseCase `switchusermode` (accessible for online- and batch-users) *The online user may leave the simulation and therefore he wishes to turn off the output of the sim to optimize run speed. This action action is represented by this use case.*
  Otherwise, a batch user can switch mode to become an online user, so he is able to watch the experiment in real-time.

Chapter 5

GUI Description

In this chapter, we explain how the program is controlled and how it outputs the information. We use a **G**raphical **U**ser **I**nterface for input and output.

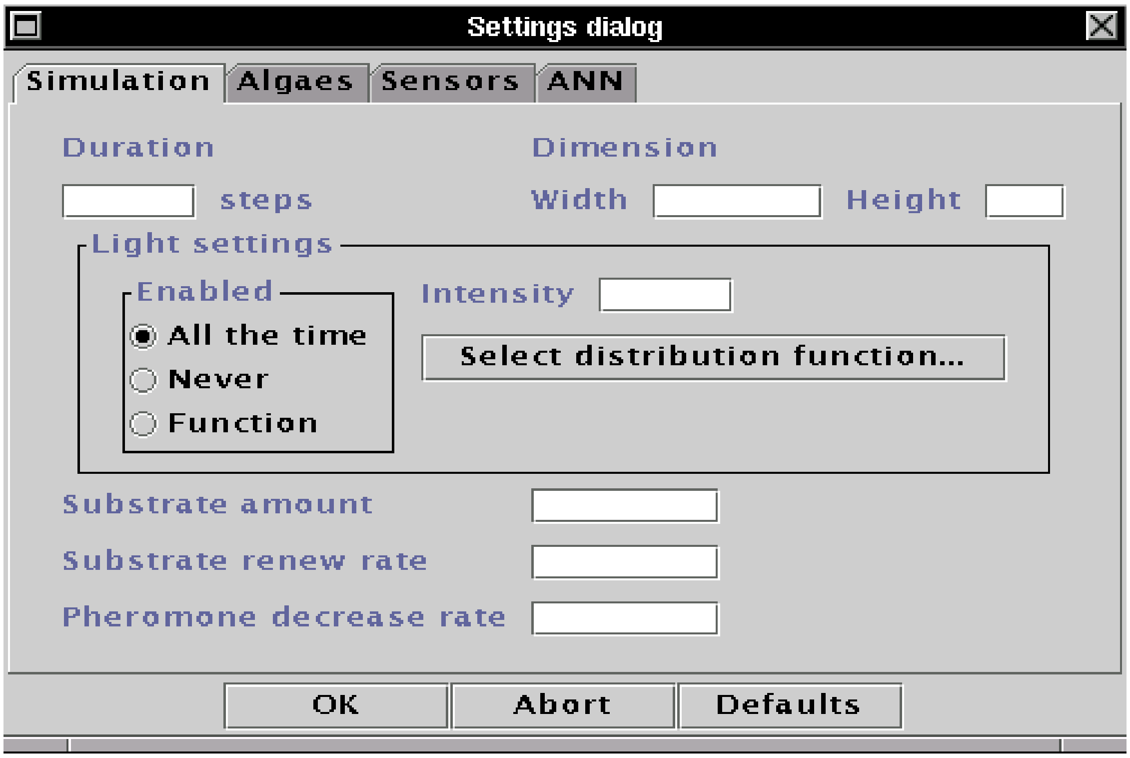
5.1 Configuration Window

The program starts with the configuration window, where all parameters relevant for the simulation are adjusted.
With the `OK` button the user accepts the adjustment; the configuration window is closed and the Simulation window opens.
With the `Abort` button the user may abort the program.
The user can choose between several pre-defined configurations using the `Default` drop-down list. Every adjustment can be saved in an own preset.

There are four tabstrips:

1. Simulation
2. Algaes
3. Sensors
4. Actions

5.1.1 Simulation



- The text field labeled `Duration` determines the duration of the simulation.
- The text fields labeled `Dimension Width` and `Dimension Height` determine the size of the aquarium.
- The three text fields at the end determine:

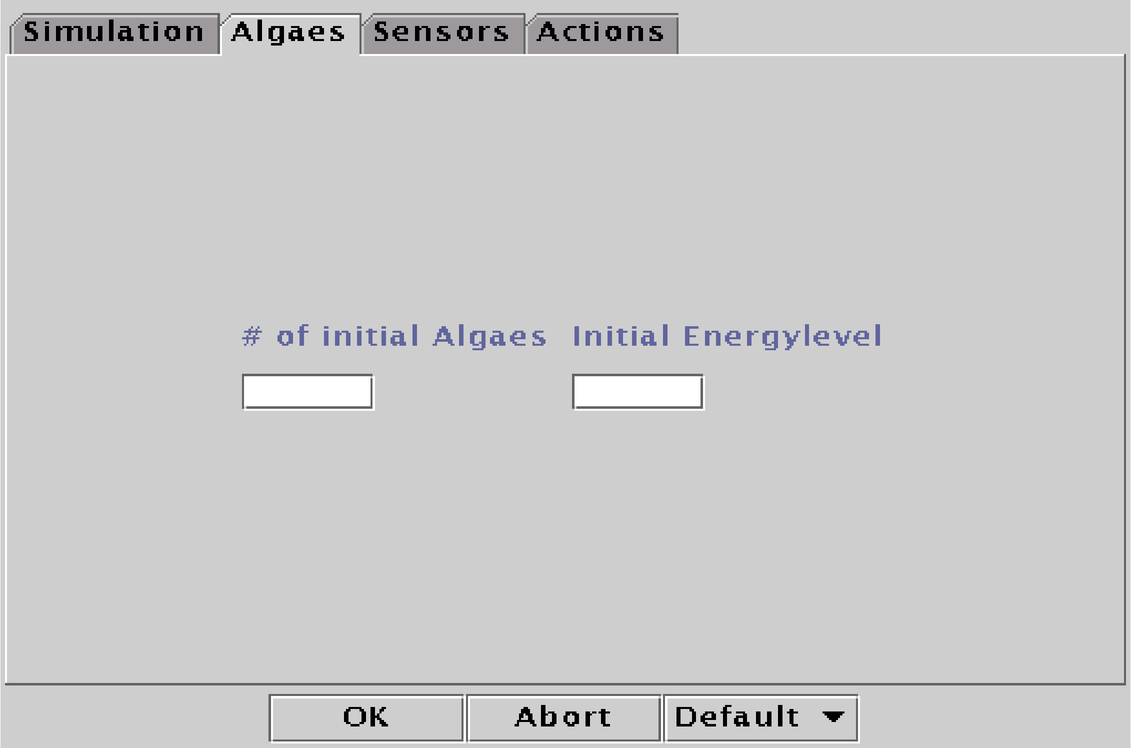
  the amount of substrate. `(Substrate amount)`

  after how many timesteps the substrate particles are renewed `(Substrate renewal rate)`

  after how many steps a pheromone disapears `(Pheromone decrease rate)`
- In the frame `Light settings` there are
  - radio buttons labeled `Enabled` with the following possibilities:

    `All the time`, which means the light shines the whole simulation
    `Never`, which means the light doesn't shine at any time of the simulation
    `Function`, the light intensity is determined by a function.
  - a text field labeled `Intensity`, which determines the light intensity on the top of the aquarium.
  - and a button labeled `Select distribution function...`, where the user can select one of several light distribution functions

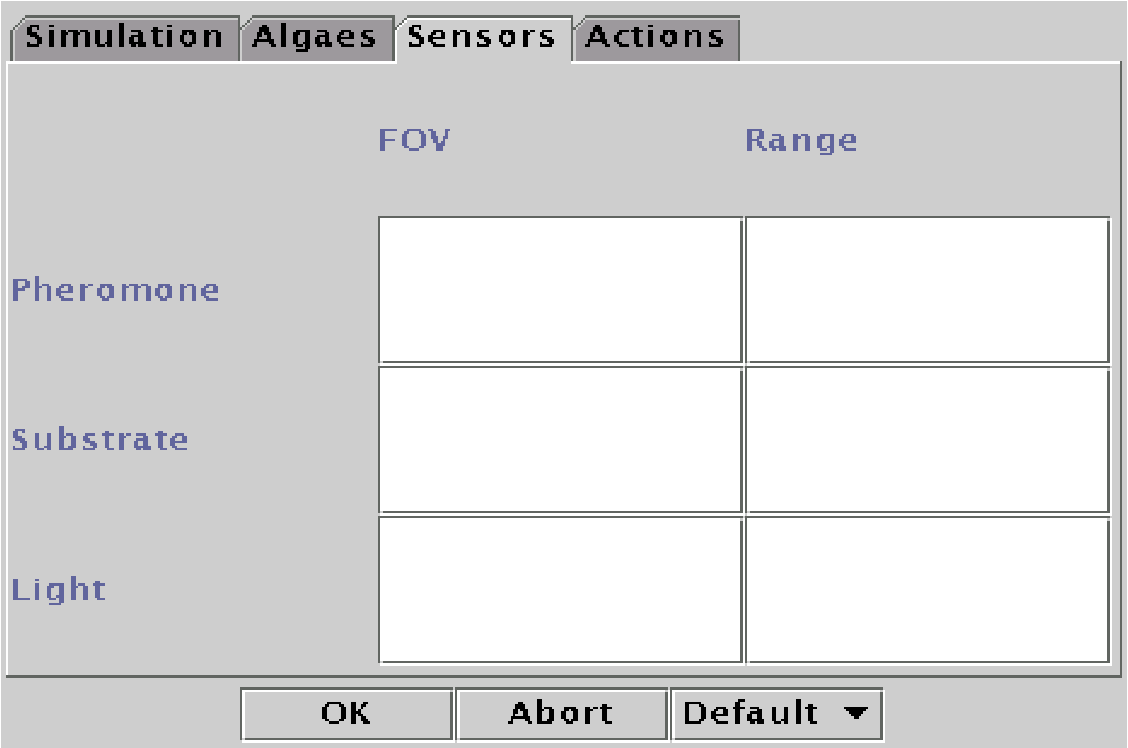
5.1.2 Algaes



In this tabstrip there are two textfieldes.

- The first field labeled `# of initial Algaes` determines the number of algae the simulation should begin with..
- The second textfield labeled `Initial Energylevel` determines the energylevel which the algae have at the beginning of the simulation.

5.1.3 Sensors



In this tabstrip the user determines the adjustment of the three sensors (`Peromone`, `Substrate and Light`).

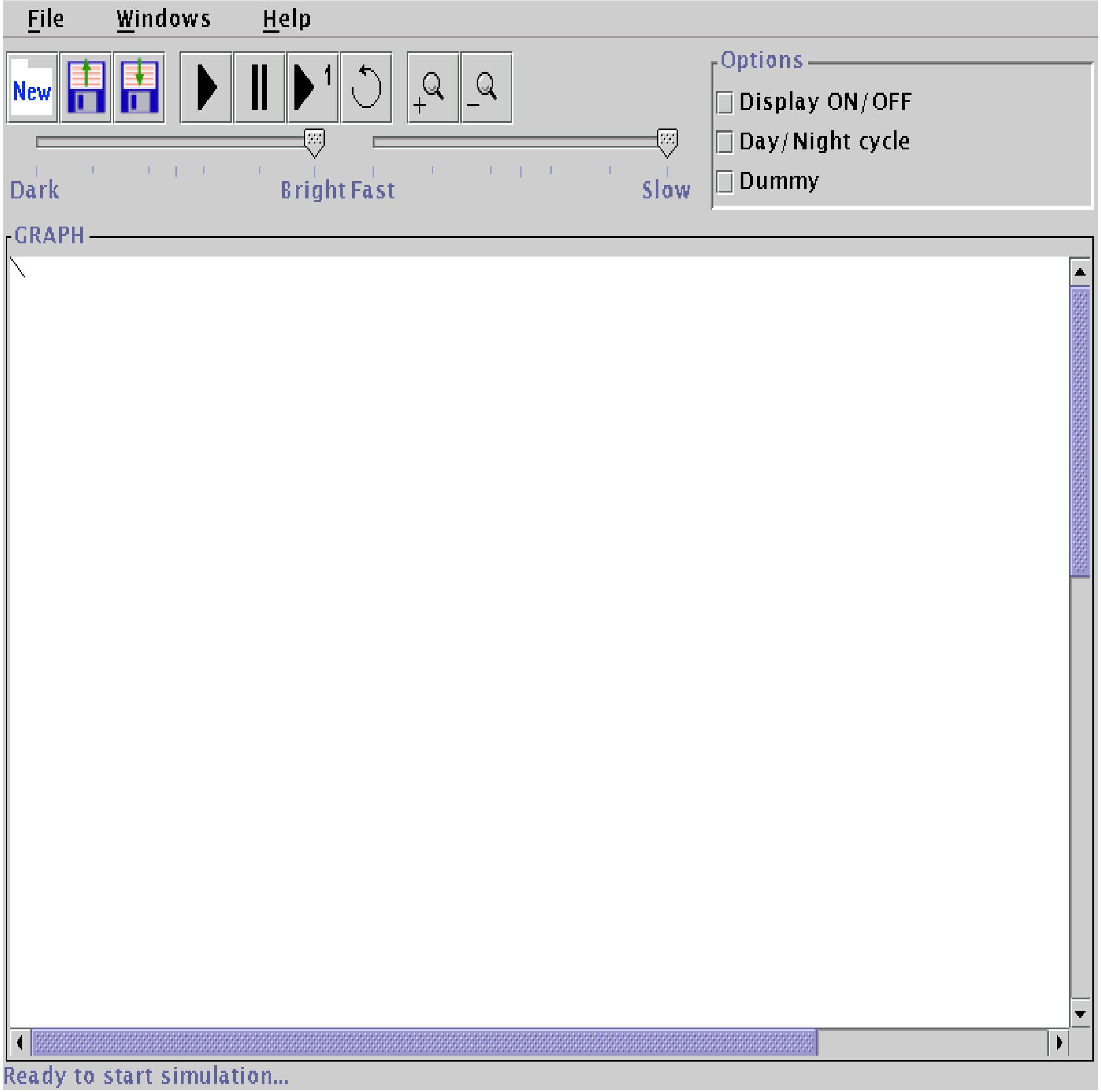
- `FOV` Field of View, this is the angle that the sensor field covers
- `Range` the range of the sensor

5.1.4 Actions

In this tabstrip are two frames:

- In the first frame labeled `Energy consumption` the user can determine
  How much energy an agent needs to move around `(Movement)`.
  How much energy an agent needs to reproduce by Mitose `(Mitose)`.
  How much energy an agent uses to reproduce by Meiose `(Meiose)`.
  How much energy an agent uses to discharge pheromone `(Peromone output)`.
- and in the second frame labeled `(Energy collecting)` the user determines
  how much energy an agent gets from `(Substrate)`
  and how much energy can an agent can take from `(Light)`

5.2 Main GUI



☐ Display ON/OFF
☐ Day/Night cycle
☐ Dummy

Ready to start simulation...

5.2.1 Aquarium

This is the world where our simulation takes place. It shows the agents, pheromon tracks and substrate parts.

When the sim is paused, it is easy to select a specific agent; a new agent information window is opened then (see below)

5.2.2 MenuBar

File

1. `New` (starts a new simulation and resets the parameters)
2. `Load configuration` (loads another configuration)
3. `Save configuration` (saves the current configuration)
4. `Save sim status` (saves everything concerning the world and its agents)
5. `Load sim status` (opens a previously saved status)
6. `Quit` Ends the application after a safety query..

Windows

1. `A list of all open windows - when one label is clicked, that window becomes the active one`
2. `Show/hide simulation info window` (see below)

Help

1. `Help content` (shows the online help)
2. `About` (displays the program's credits)

5.2.3 ToolBar

The important (e.g. the most used) parts of the user interface can be handled with the toolbar.

Buttons

- The first button restarts the simulation and reuses the actual configuration
- The second button loads a previously saved simulation state
  (this corresponds with `Menu, File, Load sim status`
- The third button saves the actual state of the simulation
  (this corresponds with `Menu, File, Save sim status`
- The fourth button starts the simulation or resumes if the simulation is paused.
- The fifth button pauses the simulation.
- The sixth button is used to run the simulation step by step..

- With the seventh button it is possible to raise the zoom factor of the aquarium
- and with the ninth button is used to lower the zoom factor

  (The possible zoom factor should be, from 1 pixel for each algae to a 10*10-pixel-per-agent view.)

Sliders

There are two sliders in the main GUI:

- The first slider controls the intensity of the light source
  (from `Dark`, no light to `Bright`, much light).
- The speed of the simulation is controlled by the second delay slider
  (from `Fast` to `Slow`).

Check buttons

With the check buttons labeled `Options`:

- the graphic output can completely be turned of with the `Display ON/OFF` switch.
- To simulate the day/night cycle, the delay slider `Dark Bright` can be disabled with the `Day/Night cycle` switch. However, the actual light intensity can still be seen by the value of this slider.
- The `dummy` switch is only a substitute symbol at the moment.

5.2.4 StatusBar

By default, it shows the current number of agents and the elapsed time. When a specific point in our world is selected, we get the following information:

- Is there substrat at this place ?
- Are any phermomon tracks here ?
- Light intensity at this place
- The XY Position

5.3 Agent Information Window

- Current sensor values
- The neural network
- The agent's genotype (bitstring)
- The speed, direction and action (i.e. eating) of the algae
- The energy level, age and current position
- Other information ?

5.4 Simulation Information Window

Here we get several statistics about the status of our population and the simulation itself.

Appendix A

Session logs

A.1 March 15, 1999

Time: 17:00 - 17:30
Place: Room 202.2, Institute for Computer Sciences, University of Salzburg
Present persons: R. Schwaiger, C. Bauer, P. Eisenmann, T. Frommwald, M. Liedlgruber and D. Kadir

Treated points:

- Formalities
- General introduction to the problem

Tasks to perform until March 22,2000

- Get photos for the project-homepage
- Have a careful look at the document
- Have a look on JAVA

A.2 March 22, 2000

Time: 17:00 - 17:30
Place: Room 202.2, Institute for Computer Science, University of Salzburg
Present persons: R. Schwaiger, C. Bauer, P. Eisenmann, T. Frommwald, M. Liedlgruber and D. Kadir

Treated points:

- We have a concrete matrix where the simulation should take place
- Substrate

  Distribution of substrate is based on a mathematical function

  There is a specific amount of the substrate at every point in our matrix

The regeneration of the substrate depends on the distribution function and a basic time period (determined by some parameters)

- Sensors

  The sensors of our agents have a specific range and an angle of 'sight'. These things are variabel.

  We got a new sensor that can detect contact (with the wall or other agents)

- Pheromons disintegrate after a while
- Possible agent actions: Power consumption, movement, pheromon output, reproduction

  All actions consume different amounts of energy

- Energy consumption through photosynthesis or substrat accomondation
- Movement

  Speed ?

  Agents may move one unit in one time unit

- Reproduction

  Meiose / Mitose

  If the agent wants to reproduce through meiose he has to have a certain amount of energy. The agent follows the pheromon tracks of other agents and puts out pheromons himself.

- Pheromon output

  During movement

  The agent should not follow his own pheromon track

- Simulation steps

  The current status of the sim is displayed after each simulation step

  During one step the agents may perform an arbitrary amount of actions (exception: movement !)

Tasks to fullfill until March 29, 2000

- Prepare further questions

A.3 March 29, 2000

Time: 17:00 - 17:30
Place: Room 202.2, Institute for Computer Science, University of Salzburg
Present persons: R. Schwaiger, C. Bauer, P. Eisenmann, T. Frommwald and D. Kadir

Treated points:

- Agent input

  We need another input for our neural network: the current energy level of the agent

  It is probably necessary that agents know whether the detected pheromon is their own or not

- The neural network

  There are units in the ANN which are connected through weighted lines

  The more important the connection of two unit becomes, the thicker the connecting line.

  Output consists of the movement direction and everything concerning reproduction

- The light

  Probably there will be some sort of 'fight' for the positions that are near the light

  Agents that are not on top level will get less light (shadows)

  For this reason they may try to consume more of the substrate - the according connecting line of the ANN gets thicker

- How could the agent learn that sexual reproduction is better than the mitose ?

Tasks to fullfill until April 12, 2000:

- Revise the paper:

  Exact problem definition

  Session logs

  Discussions

A.4 April 12, 2000

Time: 17:00 - 17:30
Location: Room 202.2, Institute for Computer Sciences, University of Salzburg
Present persons: R. Schwaiger, P. Eisenmann, T. Frommwald, M. Liedlgruber and D. Kadir

Treated points:
(document revised:)

- Light/Substrate:

  Diffusion: Substrate regenerates faster at the bottom than above.

  Agent has transparency factor.

- Pheromon, Mitose, Meiose:

  Pheromon is placed only at the point, were agent is situated;

  time-delayed pheromon reduction.

  4 various ways to consider meiosis - we choose the 3rd one: forced reproduction.

- World(object):

  Output over all partial aspects.

- ANN:

  Processing of signals with 'Cubic Splines-Methode'(set points to calculate a function)

  In the ANN 'the winner takes all' - the maximum evaluated response is selected

Tasks until Mai 2, 2000

- first GUI (/World)-prototype
- create UML-model
- ev. other questions

A.5 May 2, 2000

Time: 17:00 - 17:30
Location: Room 202.2, Institute for Computer Science, University of Salzburg
Present persons: R. Schwaiger, C. Bauer, P. Einsenmann, T. Frommwald and D. Kadir

Treated points:

GUI Interface

- Saving

  Possibility of saving from intermediate stages

  To draft on a particular step of the simulation

- Buttons, menu bar and tabstrip

  "Settings" in the menu bar is problematic

  Button for gradual running

  Rename "Main" by "Aquarium"

New tabstrip which shows a list of all algae and the possibility to select one of them

Move the "ANN Sensors" tabstrip into the algae tab writer

A new menu point "Windows" which shows the details of the agents.

- others

  Show light and substrate in the aquarium.

  Possibility to hold back indication for a certain amount of steps

Tasks until May 24, 2000

- Description of the GUI Interface
- Some agents should move in the aquarium
- Think about in which information we would be interested
- Make treated changes of the GUI

A.6 May 24, 2000

Time: 17:00 - 17:30
Location: Room 202.2, Institute for Computer Science, University of Salzburg
Present persons: R. Schwaiger, C. Bauer, P. Einsenmann, T. Frommwald and D. Kadir

Treated points:

- Document

  The document has to contain the class diagrams and use cases

  The source code should not be included

- GUI

  We will include the agent selection list discussed in a former session

- ANN

  The ANN is initialized randomly

  Visualization of the ANN is done by another project group, so we can omit that part

- We should use the same random seed for each start
- The maximum number of neurons may be entered
- Our agents "learn through reproduction"
- We save information using the XML format

Tasks until June 6, 2000

- Revise the paper, insert case diagrams and use cases
- Finish the GUI
- Make multiple agents move around in the world

A.7 June 6, 2000

Time: 17:00 - 17:30
Location: Room 0.09, Institute for Computer Science, University of Salzburg
Present persons: R. Schwaiger, C. Bauer, P. Einsenmann, T. Frommwald and D. Kadir

Treated points:

- GUI - Settingsdialog

  Simulation: change substrate-settings to be similar to light-settings (with function)

  Sensors: add bumper-sensor

  add button to save settings

- Simulation:

  output the sim state only every x-steps (definable)

  make the duration of simulation (number of timesteps) changeable during runtime

  Statistics: protocol the simulation state on harddisk and make sure functionality can be added later easily

- There should also be a slider to change the amount of substrate (similar to light)
- Colored agents: color depends on their amount of energy

Next meeting: June 21, 2000