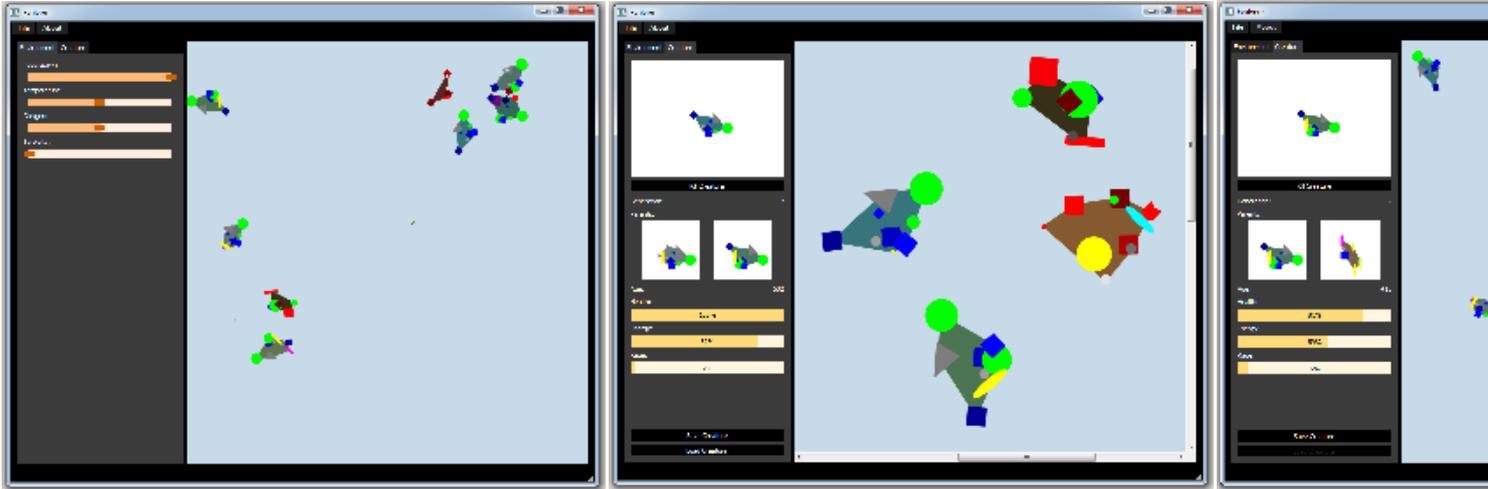


Evolvery

Evolvery is a simulation game from Laura Gohl and Marcus Soll about observing abstract creatures evolving.

Example screenshots of the software



Game mechanics

The following chapter describes the core game mechanics.

Environment

All actions happen in a liquid environment (like a lake or the ocean). The environment itself has different properties which effect the creatures living inside the environment. These are also the variables which the player can change and play around with.

These are:

Property	Description
Food spawn	Sets the rate at which food is spawned in the environment.
Temperature	Temperature of the environment.
Oxygen	Oxygen contained in the water of the environment.
Turbidity	Describes how clear the water is. If the turbidity is high, the creatures living in the environment can see worse.

An environment can hold up to 10 creatures and up to 5 food objects. As long as there is still space for creatures / food, they will spawn automatically.

Creatures

Creatures (also called fishes) populate an environment. The player can not directly interfere with them and has no control over the creatures actions.

A creature has different properties as following (properties in braces are hidden from the player):

Property	Description
health	Health of the creature. Once this reaches zero, the creature dies.
energy	Current energy of the creature. If this reaches zero, the creature loses health quickly.
(energy loss)	Describes how quickly the creature loses energy.
(food refill)	Describes how much energy (and health) a creature gets if it eats food.
(preferred temperature)	Preferred temperature. If the temperature of the environment is too far off, the creature loses health.
(preferred oxygen)	Preferred oxygen. If the oxygen of the environment is too far off, the creature loses health.
(temperature resistance)	Describes how much the temperature might be off before the creature loses health.
(oxygen resistance)	Describes how much the oxygen might be off before the creature loses health.
(horniness)	Describes how much the creature wants to reproduce.
(horniness gain)	Describes how strong the horniness of a creature increases.
(maximum speed)	Maximum movement speed of the fish.
(attention)	Radius in which the creature can detect things (e.g. food, other fish).
rage	Describes how likely a creature will attack another creature.

(aggression)	Describes how quickly a creature builds up rage.
(attack)	Describes how much damage a creature does if it attacks.
(defense)	Describes how much damage a creature takes if attacked.
(lifetime)	Describes how much additional energy loss a creature has for high ages.
age	Describes how long a fish lives.

The base values of the properties as well as the appearance are defined through the gene of the creature. One gene consists of 4-12 alleles. Each allele can have one of the following traits:

Allele type	Effect
health	Increases the maximum health.
energy	Increases the maximum energy.
food refill	Increases the energy a creature gets from eating.
preferred temperature	Highers / Lowers the preferred temperature.
preferred oxygen	Highers / Lowers the preferred oxygen.
temperature resistance	Increases the temperature resistance.
oxygen resistance	Increases the oxygen resistance.
horniness gain	Increases the rate at which a creature builds up horniness.
maximum speed	Increases maximum speed.
attention	Increases attention radius.
attack	Increases damage to other creature and aggression.
defense	Reduces incoming damage and aggression.

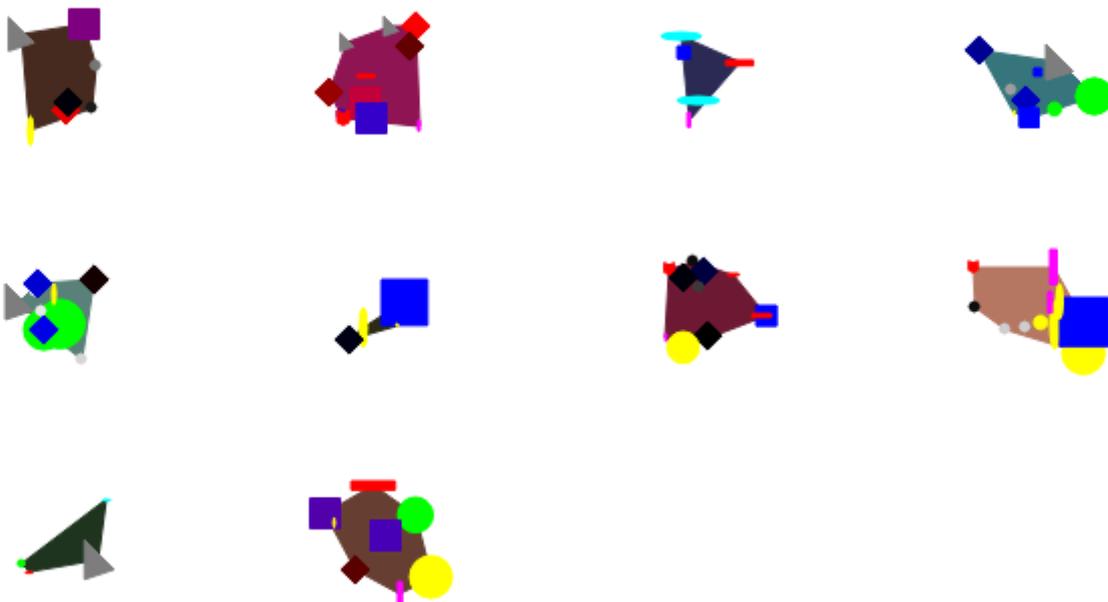
lifetime	Reduces the penalty for high ages.
----------	------------------------------------

A creature has different behaviours between it can choose based on its current properties. These are:

- Moving around
- Eating
- Reproducing
- Attacking other creatures

When a creature reproduces, the gene of the new children consists of a combination of the genes of the two parents. In addition the children gene might have some small mutations.

Example creatures (this is how creatures could look like)



Implementation

The game is implemented in c++ using Qt. This allows the game to be portable across different operating systems.

QGraphicsScene

The foundation of the Implementation is the QGraphicsScene, on which the environment and the creatures get drawn on. The whole simulation runs in real-time. A creature can interact with other creatures or food if their bodies collide on the QGraphicsScene.

Qt Style Sheet

The UI elements (like buttons and sliders) are customised using a Qt Style Sheet. The design itself is based on the design of the windows phone.

Qt Style Sheets are a CSS like description file. To change for example the look of a button (QPushButton class in Qt) you can write the following lines:

```
QPushButton {  
    border: 0px solid #8f8f91;  
    border-radius: 0px;  
    background-color: black;  
    min-width: 80px;  
    min-height: 20px;  
    color: white;  
}
```

```
QPushButton:pressed, QPushButton:hover {  
    background-color: #212121;  
    color: white;  
}
```

```
QPushButton:!enabled {  
    color: #212121;  
    background-color: black;  
}
```