**Reinforcement learning. Example of the Q algorithm.**

To study the reinforcement learning and its Q Algorithm in particular the following application was developed. The application shows the learning process of agent in continuously changing environment.

**Stack of technologies**

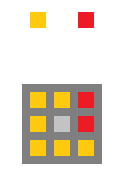
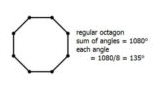
It is possible to write this application using C and Win32API and use software graphics, because it is very primitive.

**Agent:**

Agent is a robot who have sensors (eyes) and can perform simple actions – move in the environment. There are 8 sensors (around a robot) and each sensor can see the food located on the field. Each sensor can hold a value between 0.0 – 1.0 with discrete steps 0.1. Let’s called this sensitivity. The sum of sensitivities of all sensors should be equal to 1.0. Closer this value to 1.0 more food is in the direction the sensor looks.

Agent can do simple actions – move in 8 different direction pixel by pixel. The directions are the following – E (East), NE (North-East), N (North), NW (North-West), W (West), SW (South-West), S (South), SE (South-East). When agent is in intercept with food it is considered that the food is eaten. After that agent gets a lot of points for finding a food. Every single move, the hanger of the agent increases so certain amount of points subtracted from total amount of points.

The preliminary Image of the agent (robot) is the following. The sensors change in colors so that yellow color corresponds to 0.1 sensitivity and red color corresponds to 1.0 sensitivity. Values between have colors between yellow and red. Ares of sensors vision is correspond to octagon.

**Field:**

Field is a continuously changing environment. It is changeable because every time the food spawns on a new position. There is limited amount of food available on the field. Each time agent eating a food it spawns somewhere again. The preliminary image of the food is the following:



Write here the equation how food is affecting the sensitivity of the sensors. There is exact equation for that.

**Learning:**

To make learning possible the agent contains a so-called Q Table. (it is possible to find a picture in Wikipedia) The rows are states of the sensors (80 different states possible). Columns are agent actions – 8 possible actions corresponding to the move.

Effectiveness of learning can be shown as a parameter E – number of points gained per some number of moves (for example per 100 moves). Larger this amount is better. It is also possible to show this value as a simple bar plot of the application window. The values of E can be stored in the txt file. This file can be processed by python script for building a plot with matplotlib library.

How to estimate the effectiveness of a move made by the agent? we can calculate the parameter R - sum of delta distances to all available food. So, if a move is good and we are closer to larger amount of food, the R is positive then. If the move is now good than R should be negative.