Project Task Description

For the course of Nature Inspired Computing, a project work is mandatory. The following tasks require both, programming work and short, written answers to several questions.

Since the programming tasks require a robot, you should get one of the available ePuck robots from Matthias (Office 1212). He can also help you with questions about accessing the robot or programming related to the given framework. Since the number of available robots is limited, make sure that you find a way that all students can test their programs.

The project allows for team work. The team's size is limited to three students. Name all team members on the written documents as well as in your program code.

1. Make yourself familiar with EPuck robot. Most of you may already know the robot from the course C++/GUI. A good starting point is the EPuck manual from the website: http://www.amd.e-technik.uni-rostock.de/ma/rs/lv/lvgui.html
2. Download the EPuck Class Files and unzip them to a separate folder so that you can write your own EPuck application. Your EPuck application should be running on the PC while using the provided communication functions to get access to the actual robot.
3. Write* a short (up to 1 page) answer to the question "What is a Braitenberg vehicle?" Make sure that you name all the relevant elements.
4. Implement a Braitenberg vehicle for your EPuck on the PC! Use the communication functions available from the EPuck Class Files (see 2.).
5. Write* a short (up to 2 pages) *concept* of how to use Evolution Strategy to optimize the behavior of the Braitenberg vehicle. Make sure that you address all important elements of your Evolution Strategy. Discuss your ideas with the supervisor.
6. Implement an Evolution Strategy inside of the program code to optimize the EPuck's Braitenberg vehicle in such that it is able to avoid collision with the walls inside of our test labyrinth.
7. Write* a short discussion (up to 1 page) about the question "How adaptive is the evolutionary approach when targeting autonomous/intelligent robots?"
8. Write* a concept of how to achieve an adaptive behavior of our robots.
9. Implement an adaptive behavior according to your concept (from 8.) inside the program code that controls the robot.

All programming work should take part in the provided ePuck framework, e.g., in test.cpp. In addition, you may use the provided framework for the Evolutionary Algorithms[1]. In so doing, you have to add at least ea.c, ea.h, rand.c, rand.h, tools.c, tools.h, vec.c, and vec.h to the folder containing the ePuck framework. Add #include "ea.h" and #include "rand.h" to the CPP-file where want to use the functionality of the EA-framework. After adding the files and include commands, simply call make from the command line. If it's not working ask Matthias or Ralf.

The program code running on the robots has to remain unchanged. Instead, the robot is used only as mobile actuator/sensor with a communication interface. The communication functionality is activated at position three of the selector knob on the top side of the ePuck. A black marking right next to the selector knob indicates selector positon zero. Positions are counted clockwise!

---

[1] Use the updated version of the EA-Framework (uploaded on the class' webpage mid-june 2016)!
*You can write your documents in English or German.