2010-2011 IP Lectures

1. Virtual Intelligent Environments

Prof. Dr.-Ing. habil. Wolfgang Oertel

Lecture 01: Virtual Intelligent Environments: Computer graphics and multimedia

Contents: The course consists of lectures, exercises, and a complex laboratory. It permits knowledge and experience in the field of computer graphics and multimedia on the basis of the standardised language VRML. Within a complex laboratory the spatiotemporal structure and behaviour of an example virtual intelligent environment is theoretically designed and practically implemented by the students on the computer.
Keywords: computer graphics, multimedia, VRML, virtual reality, spatial structure, temporal behaviour
Schedule: 4 theoretical lessons, 2 practical lessons, and 1 complex project for a group of students for several days
Soft&Hard: VRML, graphics and audio output, standard PC, web browser

Lecture 02: Virtual Intelligent Environments: Knowledge representation and reasoning

Contents: The course consists of lectures, exercises, and a complex laboratory. It permits knowledge and experience in the field of knowledge representation and problem solving on the basis of the standardised language LISP. Within a complex laboratory the logical structure and behaviour of an example virtual intelligent environment is theoretically designed and practically implemented by the students on the computer
Keywords: knowledge representation, problem solving, LISP, logical structure, logical behaviour
Schedule: 4 theoretical lessons, 2 practical lessons, and 1 complex project for a group of students for several days
Soft&Hard: LISP, text output, standard PC, language interpreter

Project Work 01: Intelligent Virtual Object Design

Goal: A software system automating the design of virtual objects
Way: The first way to prepare objects for the use under different conditions is to adapt their physical structure to the new conditions already during the design process. This concerns for example the number, localisation, orientation, size, and features of elements, aggregates, or subsystems. Within the Artificial Intelligence programming language LISP a system is to be developed that automates the design process. The system contains, on the one hand, a knowledge base describing general regularities to be fulfilled by the object. On the other hand, it contains an inference mechanism that

uses the stored knowledge to control the design of the object as problem solving process. The user should have only the task to specify certain global parameters. The design result will be stored in the form of a VRML text file that can be used to visualise the concrete object and to navigate through it, interactively. The final object may contain static, but also dynamic features.
Implementation language: CLisp, VRML
Application field: Intelligent Buildings, Vehicles, User Interfaces, ...

Project Work 02: Intelligent Virtual Object Operation

Goal: A software system automating the operation of virtual objects
Way: The second way to prepare objects for the use under different conditions is to use an existing object and equip it with intelligent technical facilities. In this way the object can react quickly and correctly to changing exterior or interior parameters. This concerns elements, aggregates, or subsystems. Within the Virtual Reality modelling language VRML an example object is to be designed with different technical facilities, manually. Each installed technical component can possess several states or processes. On this basis the dynamics of the system is implemented by formulating generic rules linking the states and processes of the facilities with each other. The rules and their interpretation procedure are implemented in JavaScript. The script itself is divided into a flexible knowledge base and a fixed inference mechanism. The result of the work is a concrete object model that can be visualised and animated. The navigation through the model and interaction with it starts chains of automatic actions leading to an overall intelligent behaviour of the entire object.
Implementation language: JavaScript, VRML
Application field: Intelligent Buildings, Vehicles, User Interfaces, ...

2. Visualization and Scripting in Second Life

Prof. Sariselia Sore

Contents:

Keywords: 3D Virtual World, Second Life, Interaction, Visualization, Scripting, LSL

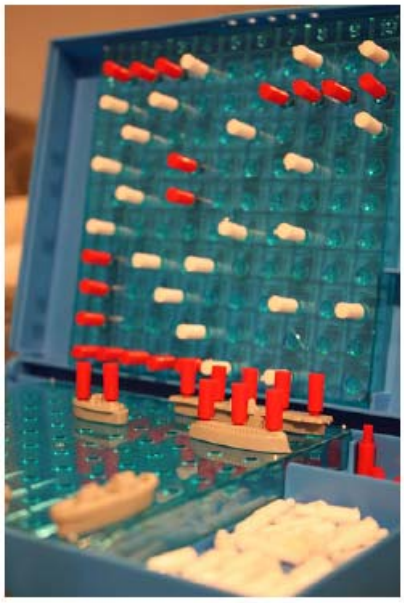
Project Work 1

Create a Mastermind game to be played in Second Life. Mastermind is a board game played by two people. The aim is to solve the opponent's colour-code in fewer turns than it takes your opponent to solve your code.

You can find more information about the game e.g. on Wikipedia:
http://en.wikipedia.org/wiki/Mastermind_%28board_game%29

One online implementation may be found on Maths Is Fun:
http://www.mathsisfun.com/games/mastermind-game.html

Project Work 2



Create a Battleship game to be played in Second Life. Battleship is a board game played by two people. The aim is to sink the fleet of the opponent. Both players have two play areas, one for their own fleet and the other for marking the hits and misses against the opponent. The game is over when the whole fleet of the other player has been sunk.

You can find more information about the game e.g. on Wikipedia:
http://en.wikipedia.org/wiki/Battleship_%28game%29

One online implementation may be found on Learn4Good:
http://www.learn4good.com/games/board/battleship.htm

3. Image processing for gesture recognition: from theory to practise

Prof. Michela Goffredo

Contents:

Keywords: image processing, tracking, gesture recognition, C++, Opencv

Project Work 1: Real time gesture recognition based on skin detection algorithms

The project aims at developing a program for recognising a set of gestures executed in front of a webcam. The system will work in real time and will be based on a smart hands' skin detection algorithm. A training phase will allow to set thresholds and parameters for increasing robustness and accuracy.

Language: C++, OpenCV

Project Work 2: Real time gesture recognition based on hand tracking

The project aims at developing a program for recognising a set of gestures executed in front of a webcam. The system will work in real time and will be based on hand tracking algorithms.

Language: C++, OpenCV

4. Speech interfaces and dialogue systems

Prof. María José Castro & Eduardo Lleida

Contents:

The aim of the lecture is to introduce to the students on spoken dialogue systems suitable for human to computer interfaces. The students will be introduced to the human speech generation and perception system and to the basic technologies used to build speech interfaces (speech recognition and synthesis). Some examples of those systems for computer-aided language learning and Human computer interaction for accessibility will be discussed. Finally the students will be introduced to the spoken dialog systems as a natural way to human-computer interaction.

Project Work : Speech enables Web sites:

1. Development of a web-based language learning activity for the training of pronunciation capabilities.
2. Development of a web-based oral dialog game with accessibility for visually impaired individuals.

This project work aims to develop a series of speech-enabled websites for different purposes. The requirements for the project are the UZ distributed recognition and synthesis applet (provided by the teachers) and a certain knowledge in HTML, Javascript and in creating JSGF grammar files. Students can apply any other knowledge in web programming they have (PHP,…).

Idea: try to integrate a front face detection system to initiate the dialog between the web site and the user.

Language: OpenCV, php, html, JavaScript, C++

5. Visualization

Prof. Tobias Schiebeck

Contents:

Lecture 1. Introduction to Scientific Visualization and SVG

This course gives a general introduction to visualization techniques and visualization tools for as well as a good range of examples of Visualization applictions. The examples include techniques for information visualization (visualization of business data) as well as scientific visualization (visualization of physically located data). The students will become familiar with the visualization pipeline and will be presented tools that use the concept. Part of the course is a hands-on session using AVS/Express - a high performance Visualization tool.

Lecture 2. Introduction to SVG

Scalable Vector graphics is an emerging standard (for ~10 years) that has the potential to revolutionize graphics on the web. SVG is an XML based description language for vector graphics objects that can be rendered directly by the web browser (at least that will be soon). Vector graphics is can be scaled and rendered on the fly without loosing qulaity. This course will provide an introduction of the basic elements and some advanced animation schemes provided by SVG. The course will have hands-on demonstrations and interactions where students will write there own SVG scripts.

Lecture 3. Introduction to GWT

The Google Web Toolkit is an advanced Open Source Library to create sophisticated Web user interfaces and applications. It provides a way of compiling Java code into JavaScript and provides tools to create AJAX based interactive and scalable Web Applications using well predefined widgets and coding patterns. Aim of the course is to provide the essentials to create a GWT web-application ready to deploy on a tomcat web server providing basic server client communication and AJAX based interactive user interfaces. During the course we will craeate a simple visualization of stock market data that updates on a regular basis.

Project Work 1: Interactive Parallel Coordinates using SVG

The aim of the project is to produce a fully interactive SVG script visualize the car dataset in parallel coordinates. The interface should provide a way of highlighting a group and the selection of an individual dataset and output the the car details of the selected car.

Skills: JavaScript, SVG, asking questions

Project Work 2: Interactive Parallel coordinates using GWT

The aim is to provide an intercative Web Application that reads a multidimensional dataset and visualizes the individual components in a description box. the user interface should provide a way of highlighting a
group of items and the selection of an individual item showing the details of the dataset.

The two projects are welcome to work together and produce a combined result using both technologies together. Creative input technologies are encouraged.

Skills: Java, asking questions