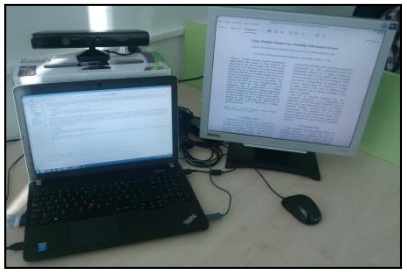
Modern approaches to human-computer interaction based on observation of a software developer

Student: Lukáš Galko
Supervisor: doc. Ing. Jaroslav Porubän, PhD.

1. Motivation: Multi-monitor work environment



- Multiple monitors in workplace result in performance benefit of **20-30%**
- Working with multiple monitors requires executing actions on **both of these** monitors
- When **switching focus** between the monitors, the developer uses **computer tools** to switch the program windows and **rotates the head** for viewing the second monitor
- The **developer's behavior** determines his intentions

2. Observation of the developer

Tools for observation:

- Camera
- Kinect device

Observation methods:

- Scanning the developer's eyes and calculation of the developer's point of regard on the monitor
- Identifying the developer's nasal tip and its position
- Estimating the developer's point of regard on the monitor with the help of his face map

3. Midas Touch problem - Uses of gaze

Tools for observation of workplace

The developer **identifies the objects** in his workplace by gaze and thus **gathers information** he needs for working with them

Tools for execution of commands

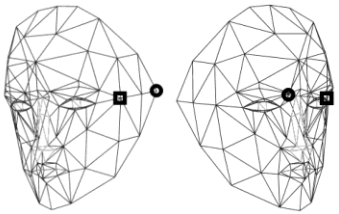
The gaze initializes **execution of reactions** in a system that **executes reactions on the observed actions** of the developer

4. Addressing Midas Touch problem

The system that executes reaction on the developer's action needs to differentiate between the **look used to observe workplace** and the **look for execution of actions**. Three main ways to address the Midas Touch problem are to:

1. Use **dwell time of focused look** to execute reaction
2. Execute reaction on a **specific amount of action occurrence**
3. Execute reaction when **the developer performs a specific gesture**

5. Identification of developer's actions with Kinect device



Kinect detects the **face map** of a developer. This map indicates the **direction of the developer's look** and can be used to decide **towards which monitor** the developer is currently looking.
The change of look from one monitor to another can be used to **indicate the need** to switch the active window, in which the developer is currently working. The computer system can use information about the currently used monitor to **show alerts** on this monitor and to **open new windows** on it, since the developer is looking at it. **The rotation** of the developer's head is **calculated** with **points** from the side **of his face map**. The **difference** between these points **indicates** whether the user is looking towards the kinect device or sideways from it.

6. Solution - Face Of useR Command Executioner (FORCE)

- Works in **two-monitor environments**
- Uses a **kinect device** for observation of the developer
- **Estimates the point of regard** of the developer on the main monitor
- **Identifies the monitor** the developer is currently looking at
- Needs **one calibration** in a stable environment
- **The developer defines the commands** for the human-computer interaction
- Definition of commands:

1. **Key sequence** that defines the behavior of commands
2. **Developer action** on which the command is executed
3. **Numeric indicator** for alteration of command execution
4. Assignment of the **program window** for the execution of a command

7. Results

- Creating a system that **performs a human-computer interaction** based on observation of the developer
- This system can perform:
  1. Execution of key shortcuts
  2. Activation of window with look
  3. Edition of text
  4. Document manipulation
- The system is **fully adjustable** to match the developer's needs.
- The system runs in **real time**
- Reuse of the system is improved through **saving created commands**