

Swank-Rats documentation

Thomas Gaida (thomas.gaida@students.fhv.at)

Stefan Lässer (stefan.laesser@students.fhv.at)

Johannes Schwendinger (johannes.schwendinger@students.fhv.at)

Johannes Wachter (johannes.wachter@students.fhv.at)

Michael Zangerle (michael.zangerle@students.fhv.at)

Abstract

Semester Project for course “S1 - Kopplung und Integration von heterogenen Systemen”.

Inhaltsverzeichnis

| | |
|-----------------------------|------------|
| Abstract | iii |
| 1 Introduction | 1 |
| 1.1 Game Idea | 1 |
| 1.2 Architecture | 1 |
| 1.3 Communication | 2 |

1 Introduction

1.1 Game Idea

Swank Rat is a rat fighter game. Two rats are trying to shoot each other with cheese. The rats are represented by robots which are controlled by two players. With a Camera over the Game-World can the software “see” where the rats are. In addition, the obstacles are detected over this camera. This obstacles are straight walls (e.g. wood slates with a red with a red mark). The Rats are able to throw pieces of cheese after the opponent. The walls serve as a limitation for the cheese-bullets.

To control the robots the live video of the world (overlaid with video of the cheese-bullets) will be displayed in a HTML UI in the browser. With buttons (and keyboard shortcuts) can the player control the real robot.

If a robot is hit (one or more) the game is over.

1.2 Architecture

TODO overview image

1.2.1 Hardware

- 2 x “Rat-Robot” with WLAN Dongles to communicate with the server
- 1 x Webcamera (for the detection of position and world)
- 1 x Server (Notebook or PC for image processing and game logic)
- 2 x Clients (Notebooks with modern Browsers)

1.2.2 Server-Software

- Server Application (Java)
 - Image processing
 - Position detection
 - Overlay webcam video with cheese-bullets
 - Stream video for client
- NodeJS Server
 - Robot control
 - Server UI (HTML)
 - User management

1.2.3 Client

- Browser Application
 - HTML5
 - Presentation of game stream
 - Javascript with Websockets
 - Buttons to control robot
 - Login
 - ...

1.3 Communication

TODO