Johannes Przybilla

+81 80 7044 6183 prjoh.dev@gmail.com prjoh.xyz linkedIn bitbucket

WORK EXPERIENCE

NAGOYA UNIVERSITY | Software Engineer — Nagoya, Japan | Feb 2022 - Today

- Implementation of VR scenario for mobile robotics platform in cooperation with Nagoya University Kawaguchi Lab, allowing interspace communication between virtual and real spaces using **Unity**.
- Presented our work as poster at SIGGRAPH 2022: MetaPo: A Robotic Meta Portal for Interspace Communication.

株式会社 **MOLCURE** | Lead Software Engineer — Tokyo, Japan | Feb 2022 - Today

- Leading small team to realize novel robotics module to enable fully-automated PCR experiments.
- Design and implementation of digital twin of robotics platform using **Unity**.

株式会社 **MOLCURE** | Software Engineer — Tokyo, Japan | May 2020 - Jan 2022

- Created VR demonstration of robotic system using **Unity** for **Occulus Quest 2**.
- Designed and implemented calibration system using **Python** and **C++** to enable precise control of dispensing robot.
- Designed and implemented desktop application using **Python**, **Qt** and **ROS** decreasing time to enable automation of biological experiments.
- Developed low level modules on embedded systems using **C++**.

ITEC RWTH AACHEN | Research Assistant — Aachen, Germany | Jan 2019 - Dec 2019

- Daimler AG coop: Manufacture of MAKI robot and GUI implementation
- Implementation of Experiment Builder interface for Cozmo

EDUCATION

Computer Science M.Sc. 2 Semester — Aachen, Germany | Oct 2018 - Oct 2019
RWTH Aachen University
Coursework: Embedded Systems, Media Computing Project (Manufacture of pneumatic soft robotic gripper)

Computer Science B.Sc. — Aachen, Germany | Oct 2014 - Oct 2018
RWTH Aachen University
Thesis: Automated Testing of Software Containing Externally Triggered Event Handlers

- Realized algorithm for C++ Symbolic Execution Engine KLEE; GNU gzip bug report

Coursework: Computer Graphics, HPC, Webtechnologies, Compilerconstruction

PROJECTS

PARALLAX VIEW — WebGL, dlib

A small side project, where I implemented off-axis projection correction using eye-tracking. It uses a facial landmark detection algorithm provided by the dlib library. The graphics side was implemented with WebGL. As a sensor, only a MacBook Air webcam was used.

SKILLS

Languages: Python, C#, C/C++, JavaScript
Tools: Unity, Qt, OpenGL, WebGL, ROS, Git, Docker, AWS, Perforce
Hardware: Arduino, ESP, Pololu A-Star, Raspberry Pi, Arbotix-M Robocontroller