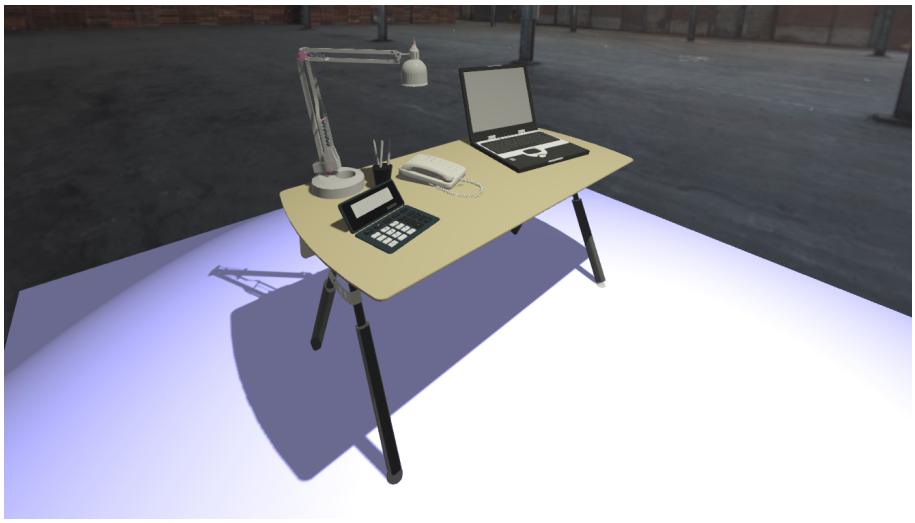
Virtual Reality
Winter Term 2018/2019

Assignment Sheet 6 (preliminary)

In this assignment sheet, you are going to implement selection and manipulation techniques in one of our immersive Virtual Reality setups. Talk to the teaching assistants regarding the choice of a setup you want to work with:

- *Mitsubishi* 3D Television (Hostname: `perseus`)
- *Samsung* 3D Television (Hostname: `orestes`)
- Small Powerwall (Hostname: `athena`)
- Semi-Transparent Mirror Display (Hostname: `andromeda`)
- *HTC Vive* Head-Mounted-Display A (Hostname: `eos`)
- *HTC Vive* Head-Mounted-Display B (Hostname: `arachne`)

In the framework provided for this assignment sheet, an individual class is prepared for each technique to be implemented in the file `Manipulation.py`. The class `VirtualRay` is already fully implemented and can be used as a reference implementation for the other techniques. All techniques are derived from the base class `ManipulationTechnique`, which includes intersection and dragging functionalities. The class `ManipulationManager` initializes all techniques and provides a toggle mechanism to switch between techniques.



For your submission, please pack your exercise folder into a `*.zip` file and upload it to Moodle. Your are required to do so by **17 January 2019, 11:55 PM**. Group work in pairs of two is permitted. You will present and explain the submitted code in the lab class on **18 January 2018**. This assignment sheet contains tasks worth **12 points** and will be weighted by **20%** for your total lab class grade.

Exercise 6.1 (3 points)

Implement the *Depth Ray* technique to disambiguate between multiple intersections of a virtual pointing ray. For this purpose, add a marker to the ray which can be adjusted in depth using the roll angle of the pointing device. A function `get_roll(MATRIX)` to retrieve the roll angle from a matrix is given in the respective class. More information on the depth ray can be found in this publication[1].

Exercise 6.2 (3 points)

Implement the *GoGo* manipulation technique to dynamically adjust the translational control-display ratio of a virtual hand. For this purpose, use a piecewise transfer function with a 1:1 control-display gain for the first half of the arm's reach (35cm) and a non-isomorphic function for the remaining arm reach ($>$ 35cm). Print your combined transfer function with a graph plotter.

Exercise 6.3 (6 points)

Implement the *PRISM* manipulation technique to dynamically adjust the *translational* control-display ratio of a virtual hand. In each frame, determine the velocities of the input device along all three cardinal axes and decide which mappings to apply to the virtual hand. The velocity thresholds `MinS`, `SC` and `MaxS` are given in the respective class. The *PRISM* technique itself is thoroughly described in this publication[2].

---

[1] Tovi Grossman and Ravin Balakrishnan. The design and evaluation of selection techniques for 3D volumetric displays. In Proceedings of the 19th annual ACM symposium on User interface software and technology (UIST '06). https://dl.acm.org/citation.cfm?id=1166257

[2] Scott Frees, G. Drew Kessler, and Edwin Kay. PRISM interaction for enhancing control in immersive virtual environments. ACM Trans. Comput.-Hum. Interact. 14, 1, Article 2 (May 2007). https://dl.acm.org/citation.cfm?id=1229857