Virtual Reality
Winter Term 2018/2019

Assignment Sheet 7

In this assignment sheet, you are going to implement navigation techniques in one of our immersive Virtual Reality setups. For this purpose, talk to the instructors regarding the choice of a setup you want to work with:

- *Mitsubishi* 3D Television (Hostname: `perseus`)
- 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, an individual class is prepared for each technique to be implemented in the file `Navigation.py`. The class `SteeringNavigation` is already fully implemented and can be used as a reference implementation for the other techniques. All techniques are derived from the base class `NavigationTechnique`. The class `NavigationManager` initializes all techniques and provides a mechanism (projection/monitor setup: keys 1,2,3 for direct selection; Vive: grip button for toggle) to switch between techniques.

Figure 1: The scene to be traveled in this assignment includes narrow pathways and different height levels.

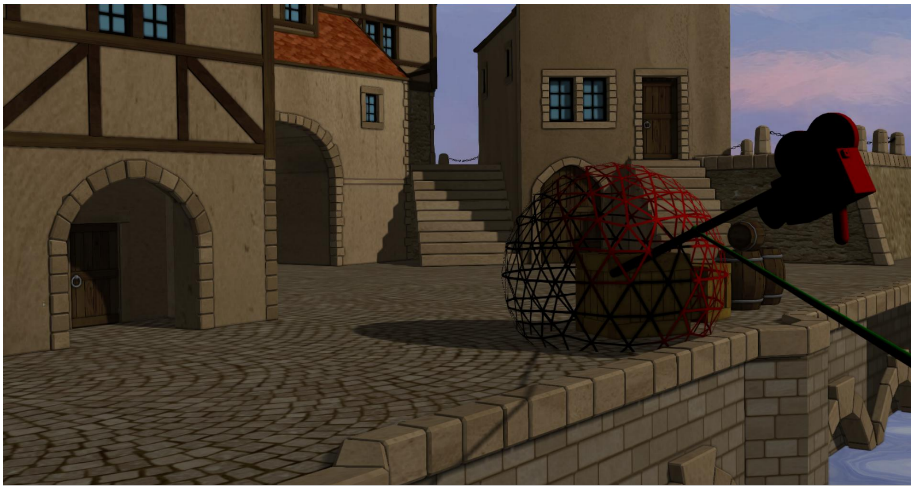

Figure 2: In the *Navidget* technique, a camera widget can be positioned with respect to a pre-defined reference point.

Figure 3: After the path animation in *Navidget*, the user is located and oriented according to the camera widget defined before.

For your submission, please pack your exercise folder into a `*.zip` file and upload it to Moodle. You are required to do so by **31 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 **01 February 2019**. This assignment sheet contains tasks worth **6 points** and will be weighted by **15%** for your total lab class grade.

Exercise 7.1 (2 points)

Implement navigation by *Teleportation*. An intersection ray, which returns the intersection point of the tracked pointing device with the scene geometry, is already implemented in the class `NavigationManager` and can be used for this task. Retrieve this intersection and teleport the user (i.e. not the center of the navigation coordinate system) to this point. The viewing orientation should be kept unchanged during this process.

Exercise 7.2 (4 points)

Implement the *Navidget* technique. The functional principles of this target-based navigation technique are explained in the lecture notes on 3D navigation and in these two papers[1 2]. An intersection ray, which returns the intersection point of the tracked pointing device with the scene geometry, is already implemented in the class `NavigationManager` and can be used for this task. After the reference point is set, the position and orientation of a virtual camera avatar (`data/objects/camera.obj`) should be defined by moving the ray on the surface of a transparent sphere (`data/objects/sphere.obj`) surrounding the defined reference position. A function `get_rotation_matrix_between_vectors(vec1, vec2)` is given to compute the rotation matrix necessary to rotate `vec1` to match `vec2`. Releasing the button triggers a transition sequence animating the camera position and orientation towards the defined camera pose.

---

[1] Martin Hachet, Fabrice Decle, Sebastian Knödel, and Pascal Guitton. Navidget for 3D interaction: Camera positioning and further uses. Int. J. Hum.-Comput. Stud. 67, 3 (March 2009), 225-236. http://www.sciencedirect.com/science/article/pii/S107158190800133X

[2] Sebastian Knödel, Martin Hachet, and Pascal Guitton. Navidget for immersive virtual environments. In Proceedings of the 2008 ACM symposium on Virtual reality software and technology (VRST '08). https://dl.acm.org/citation.cfm?id=1450589

Glossary

`start_vec.lerp_to(target_vec, fraction)`	linearly interpolates between `start_vec` and `target_vec` by the given fraction (from `0.0` to `1.0`) and returns the resulting `Vec3`
`start_quat.slerp_to(target_quat, fraction)`	spherically linearly interpolates between `start_quat` and `target_quat` by the given fraction (from `0.0` to `1.0`) and returns the resulting `Quat`
`mat4.get_rotate_scale_corrected()`	Extracts the rotation quaternion (instance of `Quat`) from a given transformation matrix. The matrix is transformed to have scale `1.0` on all three axes first, which resolves potential ambiguities since some of the rotation components in a transformation matrix are also used to encode scaling.

Workstations

Each of the workstations is equipped with a tracked pointer and stereo hardware. Please do **not** move equipment between workstations and power off all your systems after work.

Mitsubishi 3D Television

The Mitsubishi Stereo-TV is usually set to stereo mode, and this setting is kept between sessions. If, however, the 3D mode needs to be re-enabled manually, open the on-screen menu, select `3D Mode` and switch to `ON Standard`. For tracking, use the ART tracking system controller located in the corner of the lab.

Samsung 3D Television

The Samsung Stereo-TV has to be set to stereo mode every time using the remote control since this setting is not kept between sessions. Open the on-screen menu, select `Setup` and switch the `DLP 3D/Dual-View` menu item to `ON-STD GLS` or `ON-INV GLS` (find out which of the options produces a proper stereo impression). For tracking, use the ART tracking system controller next to the display and the `DTrack` application located in `/opt/DTrack2_v2.13.0/DTrack2`.

Small Powerwall

Turn on the black power distributor right next to the projectors. Afterwards, turn on the projectors by pressing the power switches on the upper side. Furthermore, turn on the `athena` computer, which is located next to the projectors. For the shutdown procedure, double press the power switches on the projectors and turn off the computer. Then, wait one to two minutes for the projectors to cool down before also switching off the power distributor. For tracking, use the ART tracking system controller located in the corner of the lab.

Semi-Transparent Mirror Display

The semi-transparent mirror display requires you to select the `Ubuntu` session instead of `Gnome` on the login screen. Once logged in, make sure the output of the upper monitor is reflected along X using the tool `nvidia-settings`. For tracking, use the ART tracking system controller located in the corner of the lab.

HTC Vive Head-Mounted Displays

Switch on a computer with a HTC Vive and press `Esc` during the startup process to boot into Windows. Once logged in, open `SteamVR` pinned to the task bar. Plug the cables of the headset and the two lighthouse trackers in power sockets. When done working, remove these cables again and power off the machine.