Report VR&AR Assignment 3 - Lukas FRITZ - 686330

Link to the Github repository: https://github.com/Vamus26/vrass3

As a starting point I used my previous assignment 2, the "Ray Wenderlich" tutorial [1] and imported VR samples & SteamVR assets.
After the basic functionalities I created a maze myself with cubes and added puzzle cubes with numbers on each side inside of the labyrinth.

Locomotion:
I have implemented teleportation to enable the movement in the scene, with instantaneous teleportation to avoid nausea.

As a side note, I did not add colliders to the walls, as I read the real movement should never interfere with the perceived movement of the user and it leads to additional nausea.

Interaction and manipulation:

I have added several interaction methods in the scene. Firstly you can turn the knob of the doors with your controller to open them by rotation and open your way into the maze.
Additionally I implemented gaze interaction while watching the cat-portrait hidden in the labyrinth. After you made eye-contact with the cat, a sound clip is played to make the user notice that something special happened in the game and two cubes spawn inside of the labyrinth. These can be picked up with your controllers and brought to the starting area.
The puzzle cubes can be rotated with your controllers and if they are rotated correctly (Spoiler: 42), and placed on the marked area they trigger the win condition, which leads to the disappearing of the podium and provides the user with facts about the number 42.

Both the gaze control and the interaction with the objects in the labyrinth always worked without any problems. To turn the door knobs you needed to be quite precise but nevertheless it always worked. To be honest the functionality of the interactions made quite little problems, compared to the programmatically triggering of the "win condition".
Sometimes it does not trigger at all and sometimes too early. I have learned that checking position in Unity is fairly easy, but checking "right" rotation is quite hard as it can be rotated differently and comparing quaternions is a difficult task.

Resources:

[1] https://www.raywenderlich.com/149239/htc-vive-tutorial-unity