

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 an imported door asset.

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. Therefore watching through the walls can be used to cheat in the maze as it is also possible to teleport through them, but as this is no competitive game I did not restrict those actions.

Interaction and manipulation:

I have added several interaction methods in the scene. Firstly I implemented gaze interaction while watching the cat-portrait in the starting room and hidden in the labyrinth to open the doors and enable your way into the maze, after the second meeting with the cat two cubes spawn inside of the labyrinth.

Additionally I included a sound clip, which is played when you watch the cat to make the user notice that something special happened in the game. The spawned cubes 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 rotate the cubes correctly you can juggle them or rotate them slowly but nevertheless it always worked for me. 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.

Future Improvements:

Revising the teleportation experience I should have chosen a curved line instead of a straight one to improve the felt experience of it and make aiming slightly more accurate.

Also more realistic interaction with the doors like pressing a handle or rotating a knob would create a slightly more immersive feeling.

Resources:

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