TECHNISCHE UNIVERSITÄT MÜNCHEN

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

Master's Thesis in Informatics

Teaching Optical Principles in XR

Tobias Zengerle

TECHNISCHE UNIVERSITÄT MÜNCHEN

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

Master's Thesis in Informatics

Teaching Optical Principles in XR

Vermittlung optischer Grundsätze in XR

Author:	Tobias Zengerle
Supervisor:	Prof. Gudrun Klinker, Ph.D.
Advisor:	Dr. rer. nat. David Plecher
Submission Date:	15.06.2023

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15.06.2023 Tobias Zengerle

Acknowledgments

I want to express my gratitude to the research group Augmented Reality and Professor Gudrun Klinker for the chance to work on this thesis. A very special thanks goes out to my advisor Dr. David Plecher for his continued support and his invaluable input on the design of XR applications and on the writing process of this thesis. His supportive, unagitated way of dealing with students always helped me stay grounded, when doubts or pre-submission panic started to creep in.

Another round of thanks goes to the chair of physics education at LMU and all people involved in the project. Professor Jochen Kuhn, Dr. Salome Flegr, Dr. Christoph Hoyer, Bernhard Emmer and Vanessa Weber, all provided me with their vital expertise and together we crafted a resource that will hopefully serve them and their students well in the future. Our collaboration was highly fruitful and, more importantly, extremely pleasant throughout its duration.

I want to also thank the RPTU department of sport science for providing me with access to the project codebase that served as the foundation for the learning application that accompanies this thesis. Specifically, I want to thank Sergey Mukhametov and Steffen Menne for their support and for sharing their technical know-how.

Last I want to thank my family, my friends and anyone else who helped keep me sane through the last months. I couldn't have done it without you.

Abstract

Extended Reality (XR) is seeing a massive rise in interest for the use in educational contexts, with a number of studies hinting at beneficial effects on learning outcome when using immersive XR platforms, compared to classic educational approaches. However, since the necessary hardware has only recently become more accessible and consumer level XR is just now beginning to push towards more widespread use, these technologies have yet to fully arrive in everyday university teaching. In this thesis, we will explore how modern XR technologies can be utilized to create new types of access to knowledge and investigate how they can enhance the learning experience in actual day-to-day universitary practice. To this end, we partnered with the LMU chair of physics education and developed an immersive Virtual Reality (VR) application for their optical physics lab course for students of human medicine.

The resulting application will be evaluated in two stages. Stage one is contained in this thesis and will see us conduct a usability study. Here, we explore the level of user acceptance for XR applications as a replacement for traditional study setups, while also aiming to identify any existing usability hurdles. Future work will entail a second evaluation stage, that investigates potential learning benefits of the application during field use in real universitary teaching.

The stage one evaluation of the application yielded a System Usability Score of 82.53, putting it in the 90-95 percentile range of applications in terms of usability and earning it the second highest possible ranking. With this, a solid groundwork has been laid for an accessible learning experience in future usage of the application.

Kurzfassung

Extended Reality (XR) erfährt einen massiven Zuwachs an Interesse für den Einsatz in der Lehre. Zahlreiche Studien deuten auf einen lernfördernden Effekt hin, wenn statt klassischen Lehrmitteln, immersive XR Platformen zum Einsatz kommen. Nachdem die benötigte Hardware erst jüngst breitere Verfügbarkeit erlangt hat und sich XR erst nach und nach als Endnutzer-Technologie durchsetzt, ist die entsprechende Technik allerdings noch nicht im alltäglichen universitären Lehrbetrieb ankommen. In dieser Arbeit werden wir erkunden, wie moderne XR Technologien eingesetzt werden können um neue Zugänge zu Wissen zu schaffen und wie sie die Lernerfahrung von Studenten im universitären Tagesgeschäft bereichern können. Zu diesem Zweck haben wir uns mit dem Lehrstuhl für Didaktik der Physik der LMU München zusammengeschlossen und eine immersive Virtual Reality (VR) Anwendung für ihr Optik-Praktikum für Studenten der Humanmedizin entwickelt.
Die entstandene Anwendung wird in zwei Stufen bewertet, wovon die erste Stufe in dieser Arbeit enthalten ist. Hier führen wir eine Studie zur Benutzerfreundlichkeit durch, um das Akzeptanzlevel für den Austausch von traditionellen Lernsetups durch XR Anwendungen auszuloten und um etwaige Nutzungshürden zu identifizieren. In einer zweiten, zukünftigen Stufe sollen potentielle Lerngewinne durch den Einsatz der Anwendung im realen Feldeinsatz der universitären Lehre untersucht werden.
Die Bewertung der Anwendung in Stufe 1 ergab einen System Usability Score von 82.53, womit sie eine höhere Usability als 90-95% aller getesteten Anwendungen aufweist und sich das zweithöchst mögliche Ranking verdient. Damit wurde ein solider Grundstock gelegt für eine zugängliche Lernerfahrung im späteren Einsatz der Anwendung.

Contents

1. Introduction

In recent years, Extended Reality (XR) has seen a massive rise of interest in the domain of science education. This is hardly surprising, as both of it's sub-branches, Augmented Reality (AR) and Virtual Reality (VR) are exceptionally well suited for this scenario. These technologies provide near limitless possibilities for presenting scientific learning content to their users. This, for example, includes the visualization of concepts that are normally invisible in the real world (e.g. magnetic field lines). They also open up the possibility for a user of these technologies to engage with objects that otherwise are too big (e.g. stars), too small (e.g. molecules) or too dangerous (e.g. hydrogen atoms during nuclear fusion) to be engaged with. At the same time, they allow for natural ways of interaction with these objects, with users being able to manipulate them with their own hands in many cases. This can take the shape of moving around physical markers or using tracked hand gestures in an AR setting, or utilizing a handheld controller to grab virtual objects in a VR scene.

This combination of vast illustrative capabilities and hands on interaction makes XR a perfect fit for the use in lab course scenarios for the natural sciences. Integrating XR technology into such lab courses can be approached in a number of ways. In the realm of AR, this might entail augmenting a real world experiment and overlaying additional visualizations to supplement the observations in the experiment. For the case of VR there are two main possibilities. One can either faithfully re-create an existing experiment inside the virtual world, or they can create novel experimental setups from scratch.

This thesis will be dedicated to this second approach and the potential benefits it can yield for university students in science lab course scenarios. More specifically, we will explore how established experimental setups can be translated into an immersive VR environment in a way that exploits the technology's strengths to provide new ways of access to knowledge to students in higher education.

To this end, we partnered with the LMU chair of physics education to create a VR application for the use in their lab course module on optical physics for students of human medicine. The current module features a classic optical rail experiment, through which the students learn about the underlying optical principles of the human visual system. For the new application, our aim is to take the same underlying abstract principles and re-wrap them with a visual representation that's closer to the actual medical context. To this end, we will introduce an interactive cross-sectional model of the human eye with simulated behavior for the basic optical image formation process, that the students can engage with in the virtual environment. For the resulting VR application, the two aspects most important to us are a high level of usability of the system and it's interactions, and clear, descriptive visualizations of the optical principles at work in the human eye.

To assess these factors and to identify any existing usage hurdles, we will conduct a usability study with a small batch of participants. A future second evaluation stage, will then see the application being used in real teaching practice of the physics lab course. The accompanying study will investigate differences in learning outcome for the use of our application, versus using the analogous optical rail setup.

In the beginning of this thesis, we will provide a basic overview of the physical and medical concepts that will be conveyed in our learning environment and give some more context on the optical physics lab course module. We will then explain the features of the "LIN-VR" application in detail and share some insights into the development process, before moving on to the evaluation of the conducted usability study. Last, we will give an outlook of potential avenues for future work and provide a conclusion based on the findings of the study.

2. Related Work

The possible application of VR and AR to various fields of education and training has been of great interest in recent years. Propelled by the near universal availability of smartphones or other handheld devices in the case of AR and the increasing affordability of consumer level VR Headsets, schools, universities and private companies have been looking for ways to enhance traditional learning approaches with these new technologies.

The idea to harness the immersive nature of VR in the context of physics education has a longstanding history. In 1993 Loftin et. al. [1] presented a "Virtual Physics Laboratory" that allowed students to alter physical variables in a virtual environment (e.g. magnitude and direction of gravity, atmospheric drag, etc.) and to observe and measure the resulting effects on a simple pendulum.

In a more recent example Georgiou et. al. [2] developed a VR application for teaching general relativity to high school students. Using this, a study was conducted in a practical classroom scenario, that had students rotate between learning stations where they, in turn, engaged with the VR application, textual sources on general relativity and video based learning content. The quantitative evaluation of pre- and post-tests showed significant learning gains for the participating students.

In a 2020 work Sidanin et. al. [3] recreated a nuclear physics laboratory experiment in a virtual environment. This work is designed as a pilot project to show the capabilites of VR to be used as a replacement tool for the highly specialized equipment usually required for such an experiment, which only few universities have access to. Based on real spectroscopy data they have created a VR simulation for an experiment to determine the mass of the deuteron. In the resulting application students can swap between interacting with the virtually recreated equipment in the way they would for the real experimental setup, and "leaving" their virtual body to take a look inside the experiment apparatus, allowing them to zoom all the way down to see what happens on the atomic scale. After engagement the students were tasked with analyzing the gathered data, like they would for the regular experiment.

A similar "proof-of-concept" approach was applied to the realm of optical physics by Engeln and Gómez Puente [4]. Here, they set out to develop a virtual lab for university students to safely experiment with lasers and optical measurement techniques. Students were taught concepts like emission and absorption spectroscopy or laser induced fluorescence, all without the need for expensive and potentially dangerous equipment.

In a 2020 work Al Amri et. al. [5] performed a more qualitative investigation in which they examined the effects on students academic achievements in the domain of geometrical optics after using VR learning content, compared to a regular classroom learning setup, including textbook work and presentations of the topics by a teacher. In the associated study they used a commercially available VR learning application that contains some lessons on the refraction of light in convex lenses. The results of this study indicated a higher scoring in a post-test for students that learned the concepts through the use of the VR environment, as opposed to the students in the control group that took the traditional approach. Additionally, the results of a learning motivation questionnaire indicated that using the VR application increased the students overall learning motivation in the field of physics.

Looking into the domain of AR, Strzys et. al. [6] worked on augmenting a standard experiment on heat conduction in metals in the context of a introductory level thermodynamics lab course. The resulting AR system utilizes the data from the infrared camera of the employed HoloLens to estimate the surface temperature of a metal rod that is being heated from one side. Based on this information a color gradient is overlayed over the rod, through which the user of the system can observe the heat slowly being transferred to the colder side of the rod. In a pilot study, they observed a small positive effect on the conceptual understanding of the conveyed topic for engagement with the augmented experiment, as opposed to the unaltered standard experiment.

The work of Tarng et. al. [7] concerned itself with the development of a mobile AR app for elementary school astronomy teaching. The app can track and record the path of the sun, by estimating it's position based on the user's GPS location and the local time of the respective time zone. The recorded path can later be reviewed for a set of astronomical visualizations to learn more about the day and night cycle and the change of the seasons, based on the relative position and orientation between sun and earth. A teaching experiment conducted with fifth graders, revealed a better results on a post-test for the students that used the mobile app, over those that solely relied on traditional learning aid on the topic.

Back on the topic of lab course experiments, Altmeyer et. al. [8] augmented an experimental setup for electric circuits, through the use of a tablet PC. For multiple components of the circuit, measuring scales for electrical quantities are overlaid into the scene near the respective element, showing data transmitted to the tablet via Bluetooth. A second app for the tablet featured these same measurement scales on a simple 2D UI, without any AR usage. A study with university students was performed to compare the two forms of visualizations. For one group of participant the tablet showing the 2D scales was simply placed next to the experimental setup, while the second group used the AR version of the application during experimentation. The results of the study suggested higher gains in conceptual knowledge for the AR group. A second evaluation for the cognitive load of participant in both groups, didn't show any differences between the two groups, indicating that none of the two experimental setups was more mentally taxing than the other.

Medicine is another field with an ever growing interest in enhancing students learning experience through XR technology. Maresky et.al [9] investigated the learning impact of students inspecting a non-interactive, anatomically correct model of the human heart within a VR environment, compared to students that studied cardiac anatomy in a non-immersive fashion. The evaluation of an anatomy post-test yielded higher scores in all tested categories for the students that engaged with the VR content.

Diving more specifically into the field of ophthalmology, as part of a 2022 work, Williams [10] commissioned the creation of a VR application that allows medical students to practice pupil examinations, designed as an auxiliary training tool during their studies. As the main benefit of such a simulation, he cites that it provides students the possibility to study a great range of rare pupil abnormalities, as opposed to real clinical practice where the chance of encountering these abnormalities is extremely low.

In a similar scenario, Wilson et. al. [11] created a VR training tool for ophthalmoscopy examinations. The resulting application has the user perform a series of steps, involved in a full eye examination, on a simulated patient. Afterwards it featured a knowledge retrieval quiz for the covered concepts. The conducted user study indicated high levels of usability associated with the application and a high perceived usefulness reported by the participating medical students.

Chan et. al. [12] conducted another usability study comparing VR and AR simulations of ophthalmoscopy procedures regarding the available interaction methods in both contexts. This study indicated a higher level of usability for the employed VR implementation and it's controller based interaction technique, compared to the used AR application and it's hand motion capture system.

We have observed a manifold of educational XR applications emerging in both the domains of medical training and physics lab course setups. However, the intersection of these two domains for the teaching of optical physics in the context of the human eye, appears to be largely unexplored so far. This thesis aims to push into that gap by combining physics simulations of refraction processes with the context relevant presentation form of a human eye, to create an immersive learning experience on the basic optical principles of the human visual system.

3. Application Domain

Before we can dive into the VR application itself, we first need to build an understanding of the underlying physical and medical processes of the human eye, that will be on display in the application. Additionally we will explore the current analogous experimental setup of the LMU medical students' optics lab course, which we aim to improve upon as part of this thesis.

3.1. Optical Physics

We start by looking at the physical principles that will be visualized and taught in the VR environment. Most importantly the principle of refraction in optical lenses. The explanation of optical phenomena and formulas will be largely based on the undergraduate physics textbook "Experimentalphysik 2" written by Wolfgang Demtröder [13], specifically on chapter 8 "Elektromagnetische Wellen in Materie" (electromagnetic waves in matter) and chapter 9 "Geometrische Optik" (geometrical optics).

In much of the following discussions, we will utilize the central simplification of geometrical optics of lightwaves into rays. A light ray represents a straight line of radius zero in the propagation direction of a lightwave. The aggregate of all light rays within a bound area (e.g. light falling through an aperture, all the light falling onto the area of a lens or mirror) is termed light beam. These simplifications will yield a sufficient accuracy, as long as the area of a light beam is large compared to the lights wavelength. Otherwise diffraction effects become too large to be ignored. As a rule of thumb, at a wavelength of 0.5µm, a beam needs to have a diameter greater than 10µm [13, p. 250]. This relation is easily fulfilled in the context of the human eye, with the spectrum of light visible to humans ranging from 400nm to 750nm [14, p. 602] and the optical elements of the eye being in the milli- or centimeter range.

3.1.1. Refraction

To explore how light is influenced by optical lenses, we first need to build an understanding of refraction, the basic underlying principle of their functioning. Refraction is one of two effects that occur when a lightwave hits the boundary between two media, the other being the principle of reflection.

In a vacuum light will always travel at the speed of light c. However, when it is moving through matter, it turns out that it propagates at lower speeds. The propagation speed

v_{Ph} (also called phase velocity) is smaller than c by a medium dependent factor $n > 1$ [13, eq. (8.1)]:

$$v_{Ph}(n) = \frac{c}{n}$$

This factor n is also called the index of refraction. The larger n, the slower lightwaves will propagate through the material. As an example, air has a relatively low refractive index of $n = 1.0003$ [13, p. 212], meaning light will move through air only barely slower than through a vacuum.

The aforementioned principles of refraction and reflection now come into play, when the light moves from one medium into another. For a more in depth explanation, let's shortly return to the non-simplified representation of light as electromagnetic wave phenomena. The incoming wave will put the electrons of atoms on both sides of the boundary in a state of excitement. This in turn will prompt them to release secondary light waves that interfere with the primary incoming wave. The result are two new light waves, one reflected back into the first medium and the other entering the second medium, usually at a different angle to the primary wave [13, p. 220]. To return to the ray representation: a ray of light that hits a boundary to a different medium, will split off into a reflection ray and a refracted ray continuing through the second medium. A graphical depiction of this split can be seen in Figure 3.1.

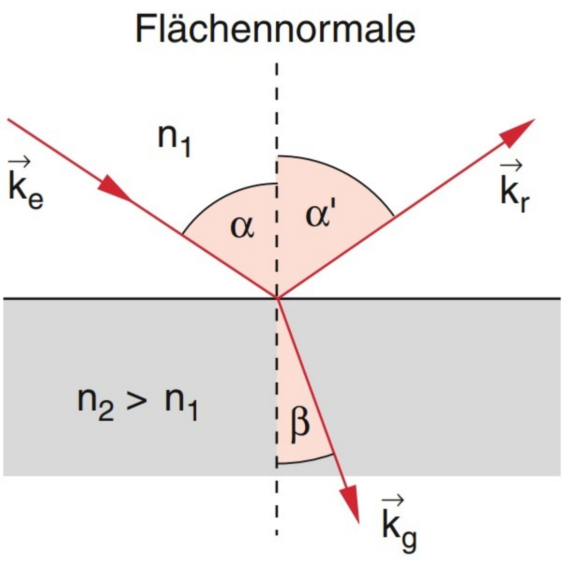


Figure 3.1.: Reflection ($\vec{k}_r$) and refraction ($\vec{k}_g$) of incoming ray vector $\vec{k}_e$ at a flat boundary between two media. Taken from Demtröder [13, p. 220]

The direction of the reflected ray follows the law of reflection as given in [13, eq. (8.57)]:

$$\sin \alpha = \sin \alpha' \implies \alpha = \alpha'$$

Thus the reflection angle α' is equal to the angle of incidence α (both mearured with regard to the surface normal of the boundary).

As mentioned above, the refracted ray goes on into the second medium behind the boundary in a direction generally different from that of the incoming wave. The exact refraction angle β is based on the angle of incidence α as well as the indices of refraction of both materials. This angle can be determined using Snell's Law [13, eq. (8.58)]:

$$\frac{\sin\alpha}{\sin\beta} = \frac{c_1'}{c_2'} = \frac{n_2}{n_1}$$

Note that c_1' and c_2' are simply another notation for the phase velocities v_{Ph} in the respective media [13, p. 222]. This law presents a way to determine the direction a light ray will travel after moving through the boundary of two media, like for example air and glass.

3.1.2. Spherical Lenses

An interesting special case with many practical implications is presented when such a boundary exhibits a perfectly spherical curvature. Figure 3.2 shows an illustration of such a spherical boundary with middle point M. The horizontal line drawn trough this point is called the optical axis. One of the characteristics of this type of optical boundary is, that for all incoming rays parallel to this axis, the corresponding refraction rays cross the optical axis at the same exact point. This is referred to as the focal point [13, p. 258]. The described effect can be observed when orienting a magnifying glass towards the sun. Based on the large distances, the sun rays reaching the surface of the earth are virtually parallel to each other. Therefore all the light passing through a magnifying glass (and the energy it carries) concentrates at a single point on the other side, the focal point.

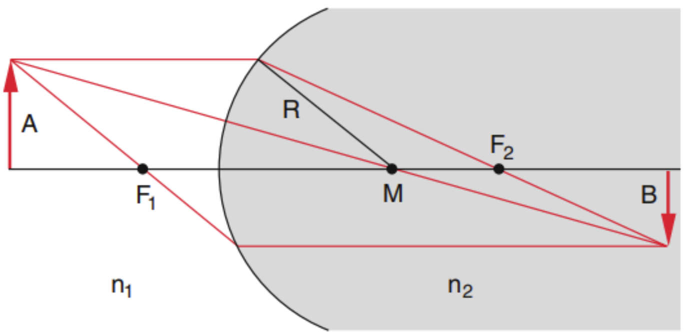


Figure 3.2.: Refraction of light at a spherical boundary with a projection from object A to image B. Taken from Demtröder [13, p. 258]

The distance measured from the medium boundary (at the level of the optical axis) to the focal point, is known as focal length [13, p. 258]. The focal length gives an indication for the strength of refraction at the associated surface. A smaller focal length means that parallel rays are refracted at a steeper angle and therefore converge closer behind the boundary. For a spherical boundary with radius R, between two media with refractive indices n_1 and n_2, the focal length can be calculated using the formula [13, eq. (9.21a)]:

$$f = \left(\frac{n_2}{n_2 - n_1}\right) \cdot R$$

An alternative quantity that is often used to describe the refractive capabilities of an optical element is the optical power with the accompanying unit [dpt]. The optical power of a lens is given by the inverse of it's focal length [13, p. 263]:

$$D^* = \frac{1}{f}$$

Image Projection

We now go back to Figure 3.2 to look more closely at what happens to the light that originates from a surface point of an object in the first medium. We assume that point A belongs to a diffusive surface. This means that light shining on the object from an outside light source is not reflected back into the environment as a single, perfectly bundled ray, but instead spreads out in all directions at a (nearly) equal distribution. In Figure 3.2 we can see three of these rays visualized, propagating away from point A in different directions.

Further, we can see that all these rays, after being refracted at the medium boundary, converge back together in a single point B. Here, B is called the "image point". It is the result of the projection of the "object point" A through the refractive system into the second medium. If we were to put an optical screen at this location we could see a projected replica of the entire object from the first medium, since the same process happens not only for the individual point A, but for all surrounding surface points a well.

It is worth noting, that the three rays in Figure 3.2 are not chosen at random. They are part of a "ray diagram" used to determine the image point B through geometrical construction [13, p. 258]. For the middle ray, a straight line is drawn from object point A, extending through the center point M of the circular surface. This ray meets the boundary at a right angle and therefore no refraction occurs. The topmost ray runs parallel to the optical axis in the first medium and is therefore, as described before, refracted towards the focal point in the second medium (F_2 in the Figure). The image point B is now located where these two lines meet. The third ray at the bottom is not required for this construction, but it is added for illustration purposes. It comes from an imagined reversal of the projection. Point A can also be re-interpreted as the image point in a projection from point B through the boundary into medium 1. Therefore the bottom ray denotes an axis parallel ray originating from point B, then being refracted at the boundary through the focal point for the reverse direction F_1, until it meets the other rays back in point A.

Thin Lenses

A "lens" is what we call an optical element that has two spherically curved surfaces "back to back". Light that propagates through a primary medium (usually air), enters a secondary medium (usually glass) through the first boundary and exits back out through the second. At both surfaces the light rays are subject to refraction. Based on the distance between the two boundary surfaces (i.e. the thickness of the material of the secondary medium), we distinguish "thin lenses" from "thick lenses". In the context of the optical elements involved in the human visual system we purely deal with thin lenses, an idealization of lenses for which this distance is very small in comparison to the focal lengths of the two boundaries [13, p. 259]. In this case we can calculate the focal length for a thin lens as [13, eq. (9.25a)]:

$$f = \frac{1}{n-1}\left(\frac{R_1 \cdot R_2}{R_2 - R_1}\right)$$

This formula is derived, based on the assumption that the medium surrounding the lens is air, for which the refractive index was rounded to 1. Therefore, the only remaining refractive index n in the formula belongs to the material the lens is made of. Next off, R_1 denotes the radius of the first spherical surface where the light enters the lens, and R_2 is the radius of the second surface where it exits back out. Usually the direction of light is depicted from left to right in illustrations on geometrical optics, which means R_1 corresponds to the left side of the lens, while R_2 is associated with the right side. The radii of the lens surfaces are defined in a way, where they can have a negative sign as well. If the center point of the spherical surface is located to the right of the boundary (i.e. towards the image), the radius has a positive sign. However, if the center point is located to the left of the boundary (i.e. towards the object), the distance from center point M to the boundary is multiplied by -1 to receive the signed radius value [13, p. 259]. This connection is depicted in Figure 3.3.

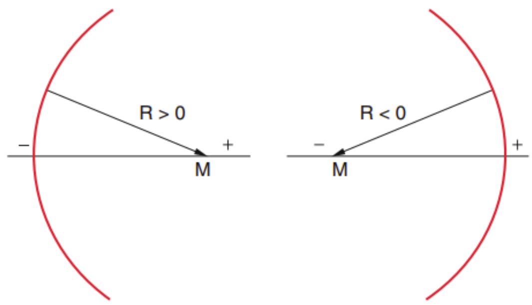


Figure 3.3.: Sign of the radii for a convex and a concave lens surface. Taken from Demtröder [13, p. 259]

Based on this, the given formula for the focal length can yield both positive and negative results. If the focal length is positive, incoming axis parallel rays will converge, as was the case for all deliberations up until now. Here, we speak of a "converging lens". If the focal point is negative however, axis parallel rays will diverge apart instead [15, p. 344]. The associated type of lens is called a "diverging lens". A collection of lenses from both categories is shown in Figure 3.4.

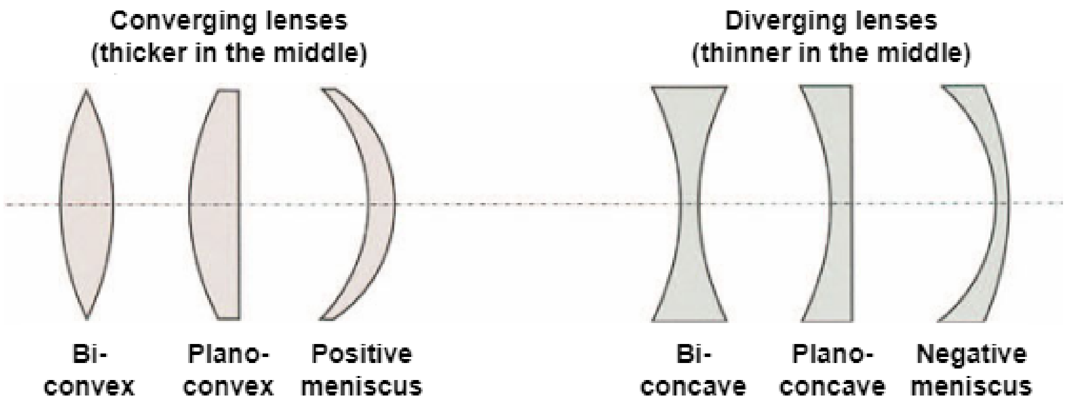


Figure 3.4.: Examples for different forms of lenses. Illustration taken from Eichler et. al. [15, p. 344]; annotations translated from German

Figure 3.5 shows the ray diagram for a bi-convex thin lens. Here, the refraction process is drawn in a simplified representation. The rays are not refracted twice, once for entering the lens and once for exiting it, as would be the case for a real lens. Instead the change in ray direction is only depicted as one combined refraction at an imagined surface called the principal plane [16, p. 6], at the center of the lens.

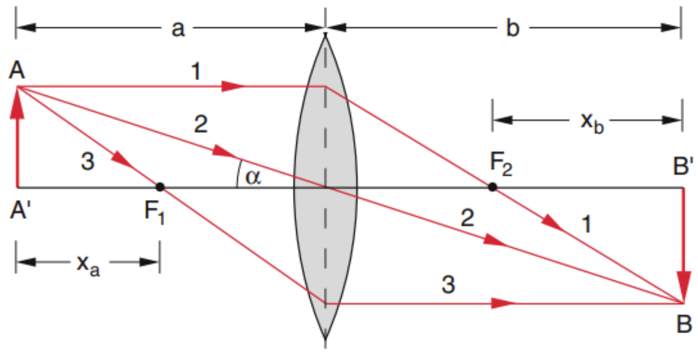


Figure 3.5.: Projection of object A onto image B via a thin lens. Taken from Demtröder [13, p. 260]

The image projection for thin lenses yields some specialized behavior. For one, we can determine the correlation between the object distance a, the image distance b (both measured from the principal plane) and the lenses focal length via the "thin lens equation" [13, eq. (9.26)]

$$\frac{1}{a} + \frac{1}{b} = \frac{1}{f}$$

Another topic of interest when examining the projection through a thin lens is the size of the resulting image. In Figure 3.5, we again deal with an object point A that is projected onto the image point B. The ratio of the respective distances from these points to the optical axis is called the lateral magnification of the lens. It is given by the formula [13, p. 261]:

$$M = \frac{\overline{BB'}}{\overline{AA'}} = -\frac{b}{a}$$

For a magnification $M < 0$, the image is projected upside down, as is the case in the given figure.

3.1.3. Systems of Lenses

Up to now, we have only looked at the refractive behavior of a single lens. In reality, multiple lenses can be combined to form lens system that reroute incoming light rays in a huge variety of ways. This enables the construction of all sorts of complex optical devices, like telescopes, microscopes and cameras.

However, for the sake of this thesis and the affiliated application, we are not concerned with any of the more advanced interactions for complex systems of lenses. With regards to the optical elements of the human eye or additional lenses for sight correction, we are mostly interested in how stacking multiple lenses back to back affects the focal length of the resulting lens system. As to that, we can calculate the combined focal length of two lenses with individual focal lengths f_1 and f_2, according to [13, eq. (9.32c)]:

$$\frac{1}{f} = \frac{1}{f_1} + \frac{1}{f_2} - \frac{D}{f_1 f_2}$$

Here, D denotes the distance between the the two lenses, measured from their principal planes. When working with a distance that is far smaller than the individual focal lengths of the lenses, the last expression grows small enough to essentially be ignored. This leads to a widely used simplification, that the optical power (inverse of focal length) of two closely adjacent lenses behaves additively with respect to the combined system [13, p. 263].

3.2. Medicine

With an established foundation of the underlying physical principles, we now want to give an introduction how they are practically applied in the human visual system. We will first go over the basic anatomy of the eye, before moving on to the physiological processes involved in (sharp) vision. The information presented in this section will mostly be based on Walter and Plange's "Basiswissen Augenheilkunde" [17], as well as the chapter "Der Sehsinn" (the visual sense) from the book "Tier- und Humanphysiologie" by Müller et. al. [14].

3.2.1. Basic Anatomy

First, we will briefly go over the anatomical structures of the eye that are most important for the intersection between human medicine and optical physics. A full cross section of the human eye with labeled components is provided in Figure 3.6.

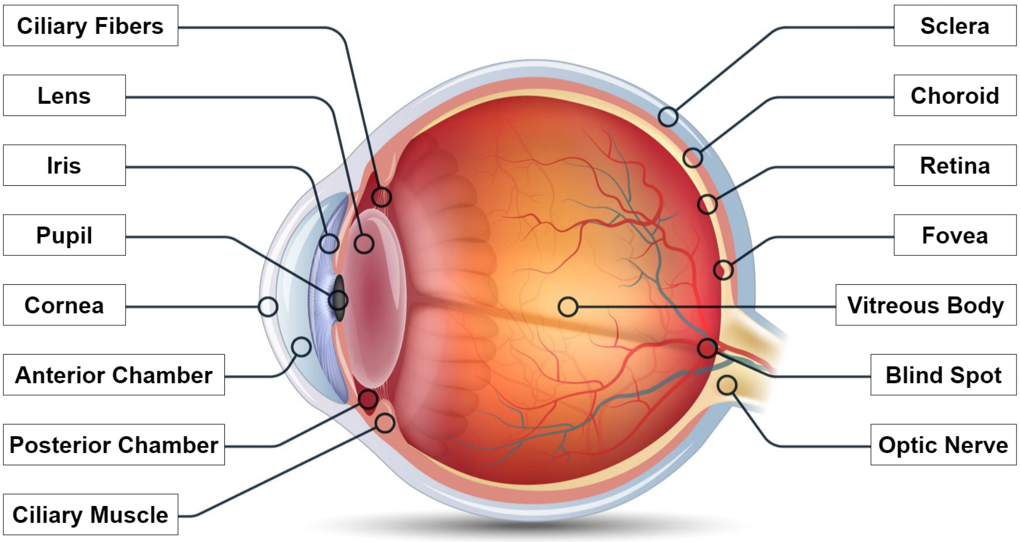


Figure 3.6.: Overview of the anatomical structure of the human eye. Illustration taken from [18]; labels translated from German

The actual visual sensing happens on the surface at the back of the eye, called the retina. Here, two types of photoreceptors, one for colored vision and one for seeing under dark conditions, detect photons that fall onto them [14, p. 605]. The aggregated signals from these receptors are then being pre-processed to a small degree by neurons in the retina, before they are sent to the brain via the optic nerve [14, p. 610].

On it's way to the retina, the light passes through a number of refractive elements that are collectively called dioptric apparatus [14, p. 617]. It contains the cornea, the lens and the vitreous body, with the cornea providing the highest contribution to the overall refraction. All these elements combined achieve an optical power of around 59 dpt, which corresponds to a focal length of 17 mm [14, p. 618].

The eye's lens is attached to a circular arrangement of ciliary fibers, which in turn are attached to the ring shaped ciliary muscle. The lens has a certain amount of elasticity and can change it's shape to adjust it's focal length, if needed [19, p. 17].

In front of the lens there is the iris, which acts like an optical aperture to limit the amount of light falling into the eye and avoid blinding in high illumination environments [14, p. 611]. The opening of the iris is known as the pupil and the reflexive change of it's diameter to adapt to bright or dark environments is called adaptation [20, p. 358].

3.2.2. Accommodation

In the relaxed default state of the eye, it's dioptric apparatus is focused for clear vision in the far distance. Light rays that run parallel to the optical axis are converging exactly on the retina [17, p. 24]. In this, the light rays from a point on a very distant object are close enough to being parallel, to still be projected as a sharp image at the back of the eye. The distance focused state is characterized by a full relaxation of the ciliary muscle (i.e. the largest possible opening of it's ring), which puts the ciliary fibers under full tension. This in turn causes the eye lens to be stretched from it's natural heavily curved shape into a flattened state with decreased curvature that yields maximum focal length [14, p. 618].

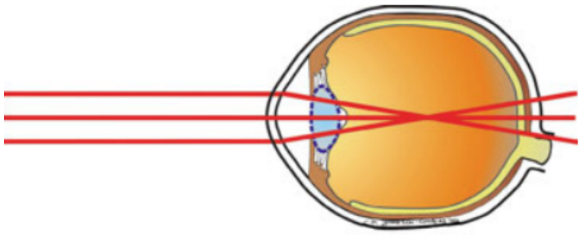
When now looking at closer objects, the overall focal length of the system needs to be decreased, in order to keep the point of projection located on the retina. In the human eye, this is achieved by a contraction of the ciliary muscle. This takes off some of the tension from the ciliary fibers and the inherent elasticity of the lens causes it to regain it's natural, more curved state [14, pp. 618-619]. This process is called accommodation. With age, the elasticity of the lens diminishes and the lens can no longer achieve the degree of curvature needed for clear vision at close distances, which results in age-related farsightedness [14, p. 618].

3.2.3. Near- and Farsightedness

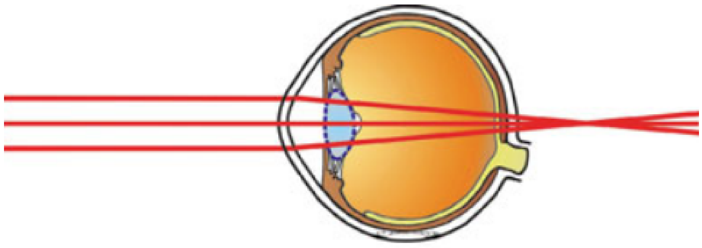
To understand what that means exactly and to explain other non age-related vision defects, we need to look into the interplay between the dioptric apparatus and the length of the eye. In this case we are talking about the "axis length" [17, p. 24], that is given by the distance measured along the optical axis from the cornea to the retina. For sharp vision, the refractive capabilities of the dioptric aparatus need to fit together with the axis length. As an example, in an eye with axis length 24 mm, the Cornea needs to have an optical power of 43 dpt and the lens one of 19 dpt, to ensure full visual acuity [17, p. 24]. If there is a mismatch between axis length and the combined optical power of the refractive elements, then the incoming light won't converge at the correct distance and the image on the retina becomes blurred.

A depiction of the two basic forms of sight defects, that result from such a mismatch, can be seen in Figure 3.7. In the case of nearsightedness (myopia), the eyeball is too long for the given dioptric apparatus [17, p. 25]. The point of sharp projection lies somewhere in front of the retina. Meanwhile, in the case of farsightedness (hyperopia), the refracted rays converge at a point behind the retina. Here the eyeball is too long, relatively speaking [17, p. 26].

However, these refractive deficits can easily be corrected utilizing the properties of a lens system. Additional lenses can be set in front of the eye, in the form of glasses or contact



(a) Nearsighted eye



(b) Farsighted eye

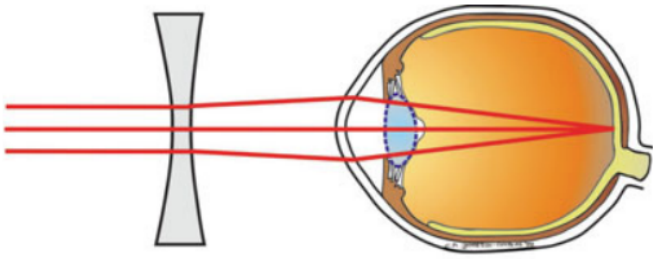
Figure 3.7.: Illustration of refractive deficits. Taken from Walter and Plange [17, pp. 26-27]; original source: Krieglstein et. al. [21]

lenses, to increase or decrease the optical power of the system, so that it again fits the axis length. This process is shown in Figure 3.8.

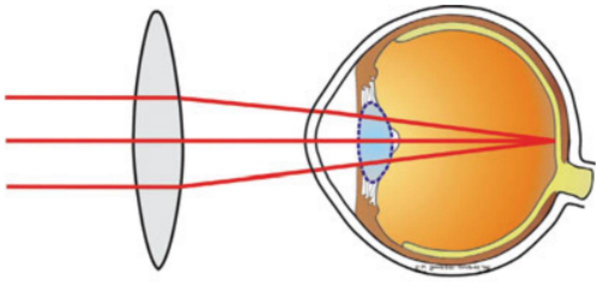
3.2.4. Cataract

The cataract is a condition in which the lens of an affected eye gradually turns cloudy, impairing vision up to the point of blindness, if allowed to progress untreated. Gerste [19, pp. 11-14] cites old age as the most common cause. Through biochemical changes during the natural aging process, the affected lens initially receives a slight yellowish tint, that generally doesn't negatively impact vision. However, over the span of several years the lens turns more and more cloudy, while vision becomes foggy as well. In the cataract's final stage, the lens has an opaque, milky white appearance and the patient becomes fully blind.

Fortunately, advances in modern medicine have opened up avenues for highly effective treatment, long before this stage is reached. If the progression of the cataract reaches a debilitating stage for the patient, an ophthalmologist can perform a cataract operation to restore their visual acuity. Walter and Plange provide an overview of the underlying procedure [17, p. 50]. First a small circular opening is created in the anterior lens capsule. Through this opening a small apparatus is introduced, that uses ultrasonic waves to break down the eye lens in a process called "phacoemulsification". The lens fragments can then be sucked off via the same device. Once the lens is fully removed, a rolled up intraocular lens is inserted through the same opening, where it unfolds and is kept in place by the lens capsule. The artificial lens has a fixed focal length and thus the eye loses it's accommodation abilities.



(a) Correction of nearsightedness with a diverging lens



(b) Correction of farsightedness with a converging lens

Figure 3.8.: Use of additional lenses to correct refractive deficits. Taken from Walter and Plange [17, pp. 26-27]; original source: Krieglstein et. al. [21]

Commonly the focal length is choosen to allow sharp vision in the distance, while additional glasses will be needed for reading.

3.3. Physics Lab Course for Human Medicine at LMU

Students of human medicine at the LMU undergo a physics lab course [22], in order to deepen their knowledge of some of the underlying physical principles of human physiology. The lab course includes a number of hands on experiments, each dedicated to a different sub-field of physics. One of these experiments, that goes by the module name "LIN - Abbildung durch Linsen" (LIN - image formation through lenses), deals with geometrical optics for simple lens setups.

Currently in this module, the students work with an optical rail. This is a classic experiment where different optical elements, including a light source, a projection slide, a screen and various lenses, can be slid along a metal rail to change their distances from each other. The specific rail setup used can be seen in Figure 3.9. Throughout four sub-experiments with different lens setups, the students are asked to move the elements on the rail to a number of given positions. After every step they take measurements of the object distance and the image distance, through which they can calculate the focal length of the optical system.

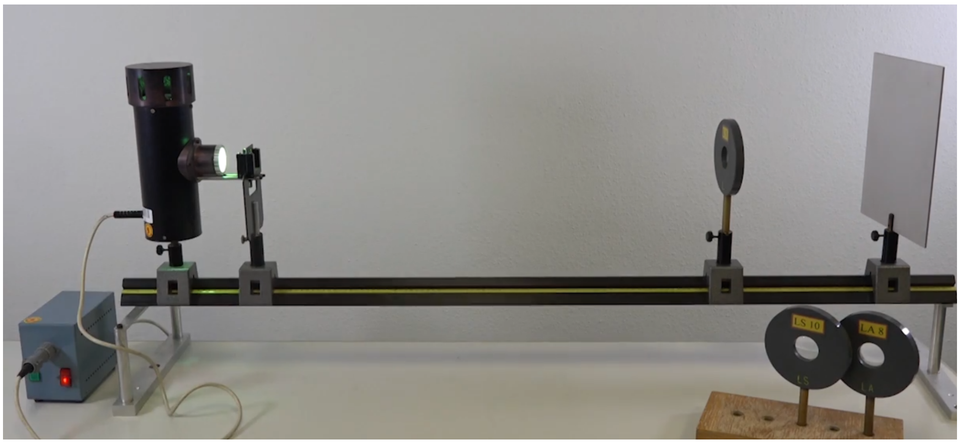


Figure 3.9.: Old experimental setup of the LIN module with an optical rail. Taken from the website of the LMU chair of physics education [23]

4. LIN-VR Application

As an alternative to the current analogous lab course procedure, we want to propose an immersive VR learning application for the teaching of optical physics. The shift into the virtual worlds allows us a number of benefits compared to the traditional optical rail experiment. While this experiment is well suited to cement the students knowledge of optical physics through a hands on approach, it is also highly abstract and far removed from the actual subject matter of the human eye. We however, intend to present the same content in a way that is much closer to the medical domain and thus, hopefully, easier accessible to students of human medicine. Therefore, in our application the students will perform the optical experiments on an enlarged cross-sectional model of the eye, which can be seen in Figure 4.2. This model covers all the basic optical principles that can be investigated through the optical rail and provides additional visualizations for them, that are only possible through the use of XR technologies. Going beyond the optical basics, the eye model also incorporates simulated physiological behavior to convey how these principles are practically applied in the human visual organ.

Figure 4.1 shows the full application environment, the students will work in. The presented application was created using the game engine Unity [24] and during development we worked with the HP Reverb G2 VR Headset [25]. The same device will also be used during evaluation and, ultimately, in the actual lab course practice. Throughout the following chapter, we will take a closer look at each of the individual elements of the simulated eye and the greater application environment. First however, we will touch on the larger context this project is located in, before giving a rundown of the covered learning content and the individual exercises the students will be going through.

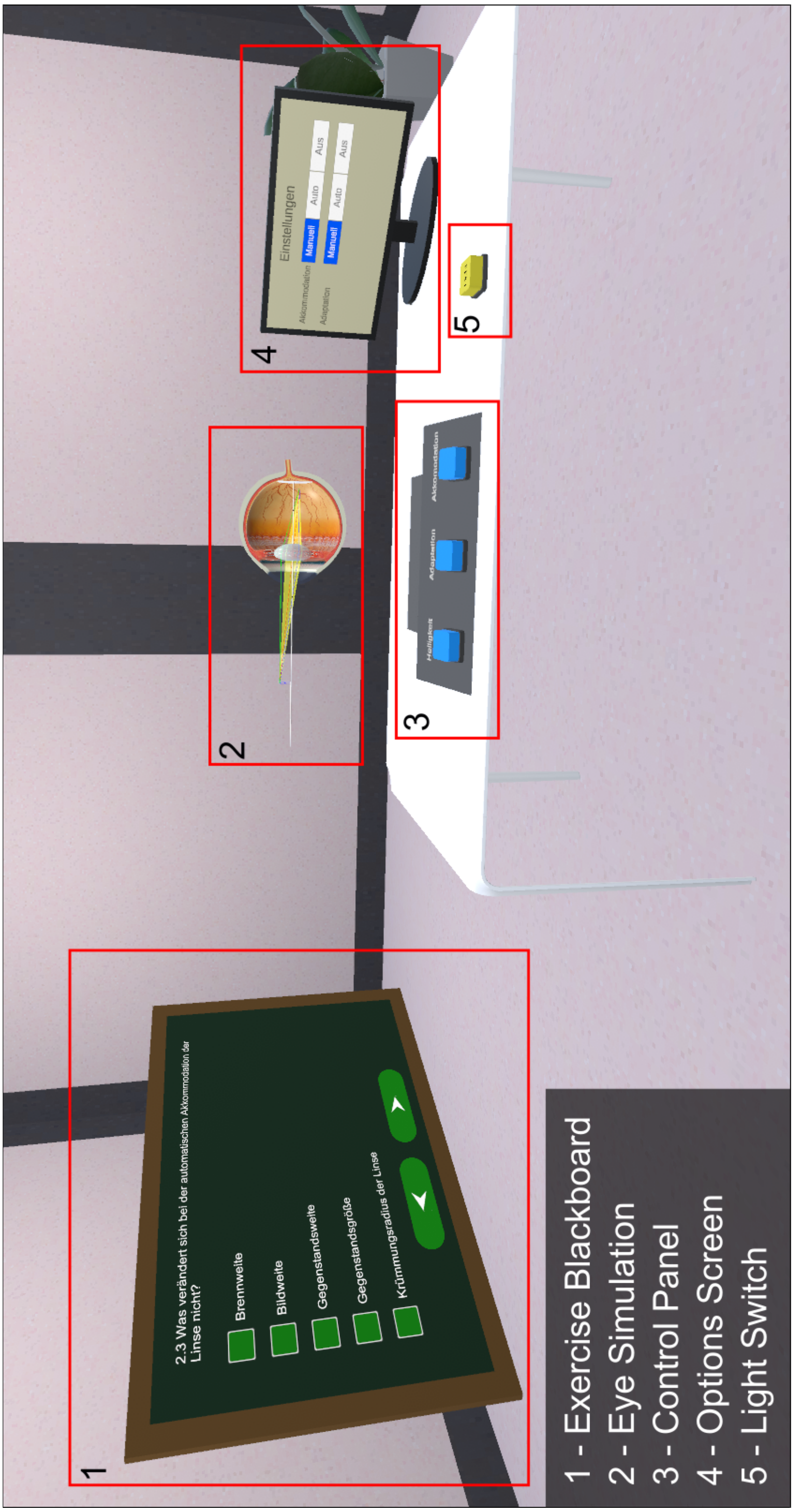


Figure 4.1.: Overview of the application's working space and it's elements

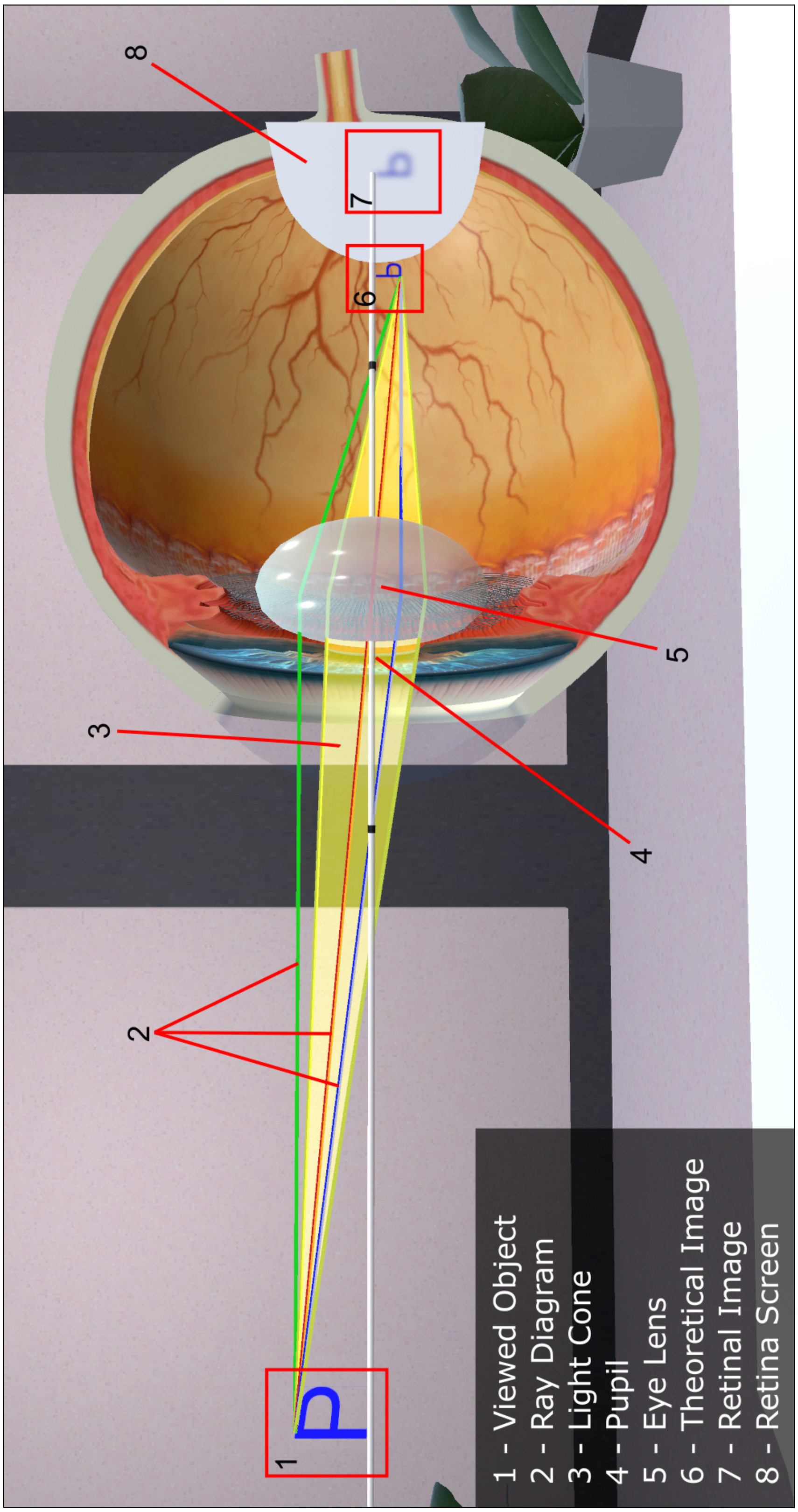


Figure 4.2.: Closer view of the eye simulation and it's individual components

4.1. Au.Ge Project

The efforts of the LMU to create a VR application for the use in teaching of optical physics is not an isolated affair. It stands in the broader context of a government subsidized project of the name "Au.Ge", carried by the RPTU Kaiserslautern-Landau [26]. The goal of this project is to examine how the learning process in health sciences is influenced by the use of AR and VR technologies, with a special focus on the optics of the human eye.

Within the framework of this project, a close cooperation between the RPTU and the LMU has formed to bundle up resources and work on common projects. So far, this cooperation yielded a number of different VR and AR applications, many of which are meant for the use in student labs for school classes. One of these applications is the "EyeLearnRoom", a sandbox environment where users can explore various topics related to lens refraction and the human eye, at different learning stations. For example, these stations include a virtual version of the optical rail experiment and a 3D model of the human eye, where the user can grab and remove some of the anatomic structures to inspect them more closely.

We were kindly given access to the codebase containing the Unity project of the EyeLearnRoom by the RPTU, so it could serve us as a starting point for our own application. With the basic VR environment already set up and with most of the fundamental controls already implemented, we could direct our full attention towards developing the more advanced content required of the lab course application.

4.2. Features and Taught Content

Working through the learning content will be divided into a series of exercises. Through these, the students will gradually build an understanding of the physical workings of the diotric apparatus and the associated advanced physiological behavior of the eye. At the start of the thesis project, together with the representatives from the chair of physics education at LMU, we devised a list of six planned exercises, for the students to go through. Each of them covers a different physical or medical concept, which include in that order: Anatomy, Basic Image Formation, Accommodation, Adaptation, Corrective Lenses and Cataract Surgery.

In each exercise the students will engage with the cross-sectional eye model, which is used to simulate and visualize the respective concept (Figure 4.1, annot. 2). Instructions what to do in the simulation and what to look for are displayed on the exercise blackboard to the side of the working table (Figure 4.1, annot. 1). On the same blackboard, the user will be presented a set of multiple choice questions regarding their observations in the simulation or as follow up for other informational content presented on the blackboard. Each question offers five answer options and is rewarded with up to five points (one point for each answer correctly crossed/correctly left uncrossed). At the end of each exercise, the students need to reach a set point threshold in order to advance to the next one. A full list of the questions included in the application (in German language) can be found in Appendix B. Since the application is specifically designed with medical students in mind, a lot of the topics require some pre-existing knowledge of anatomy or physiology.

It has to be noted that the final application, at the end of this thesis working period, ultimately contains three of the six initially planned exercises. The most important exercises 1-3 are fully implemented and were included in the version used in the usability study. Meanwhile exercises 0, 4 and 5 were assigned lesser priority and eventually had to be excluded owned to the limited time frame. Although these exercises remain a topic for future work, for the sake of completeness, we want to include their planned concepts here, alongside the description of the actual implemented exercises.

4.2.1. Exercise 0 - Anatomy (Concept)

Exercise 0 acts as a chance for the students to refresh their existing knowledge of eye anatomy and review the medical terminology of each element. This happens on a plain cross-sectional eye model (without any additional visualizations), where empty labels point at the various parts of the eyes. The students are now presented with an unsorted set of Latin names for the different anatomic structures. They then need to pick up the individual terms with their virtual hand and place them inside the correct label in a Drag&Drop type interaction. The correctly solved anatomy quiz would look roughly like the illustration seen in Figure 3.6, only with Latin names instead.

4.2.2. Exercise 1 - Basic Image Formation

This exercise deals with the basic refraction of light at the eye lens and introduces the associated concepts, like focal length, object distance or image distance. Throughout the exercise, the accommodation of the eye is turned off, meaning the focal length of the lens (Figure 4.2, annot. 5) remains static. The user is instructed to move the viewed object (Figure 4.2, annot. 1) along the optical axis. For the changing object distance, they are asked to observe at what distance the refracted rays converge back together, and how this, in turn, affects the sharpness of the image on the retina (Figure 4.2, annot. 7).

4.2.3. Exercise 2 - Accommodation

Exercise 2 introduces the physiological process of accommodation, in which the eye lens changes shape, to put objects at varying distances into focus. The students are again asked to move the object in front of the eye. However, this time they can observe that the lens will deform to set the correct focal distance and maintain sharp vision. They will also be able to see the anatomical structures that are involved in stretching out the lens and how they behave during accommodation.

In the second part of the exercise, the user gets to manually control the eye's accommodation. They will be presented with a series of preset object distances and are asked to change the lenses focal length accordingly, to receive a sharp image on the retina.

4.2.4. Exercise 3 - Adaptation

Next, the students are confronted with the process of pupil adaptation and with the influence that different lighting condition have on the final image. In the first half of the exercise, the brightness of the self-luminous viewed object is fixated at its brightest level. The user now gets manual control over the diameter of the pupil (Figure 4.2, annot. 4), to see what happens to the image if too much light is allowed to enter the eye.

In the second half, the natural adaptation is turned on. This time the user gets to control the brightness of the viewed object instead and will be able to observe how the pupillary light reflex changes the pupil diameter to prevent overexposure.

4.2.5. Exercise 4 - Corrective Lenses (Concept)

This exercise deals with basic vision defects and how additional lenses can be used to correct them. To this end, the eyeball of the model will either be elongated or shortened, for a fixed focal length of the eye lens, to demonstrate a state of near- or farsightedness, respectively. A set of various different corrective lenses, both converging and diverging, will now be available on the table, for the user to pick up and place in front of the eye. The result is a lens system with a new focal length and the image sharpness changes, for better or for worse. Over the course of the exercise, the user will be confronted with a number of varying states of near- or farsightedness and will be asked to try out different corrective lenses, until they find the correct one that restores sharp vision.

4.2.6. Exercise 5 - Cataract Surgery (Concept)

Exercise 5 is dedicated to the cataract of the eye lens. At the start of the exercise, the lens turns a milky grey and the user sees how the incoming light is diffused into a muddled, blurry image. Following this, a stylized animation of a cataract surgery will be be played, and the user can watch as the afflicted lens is liquefied and sucked off, before inserting an artificial lens. Afterwards the simulated behavior of the eye model is turned back on and the user can see how vision is affected by the use of the artificial intraocular lens with fixed focal distance.

4.2.7. Sandbox Mode

After the user has completed all the official exercises, they are invited to linger around in the application and keep exploring the eye simulation. During this "sandbox mode", all functionalities from the individual exercises are active. The user is also given full control of the background state of the simulation (e.g. automatic vs. manual accommodation), which was previously fixed for any given exercise. This way, they can review some of the previously covered concepts, if they wish. The sandbox mode also allows the user to observe some more complex interactional effects between different parts of the simulation. As an example, students can set a non-optimal focal length and then play around with different lighting conditions, to observe the compounding effect poor lighting has on already bad eyesight.

4.3. Eye Simulation

At the heart of the application stands a cross-sectional model of the human eye with simulated behavior of the covered physical and biological principles. The interactive simulation reacts to changes made in the environment by the user and visualizes the various concepts, using overlaid illustrative aids. The 3D base model, textures and the accompanying animations for lens deformation and pupil dilation were created by a professional 3D Artist, commissioned by the RPTU. The basic model can be seen in Figure 4.3.

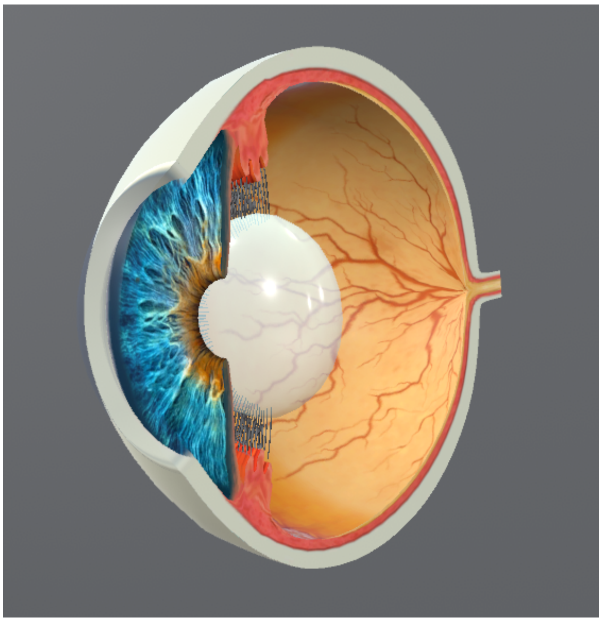


Figure 4.3.: Close up of the eye model without added visuals

The version of the EyeLearnRoom that we inherited at the start of the thesis working period did already contain an instance of this model, outfitted with a basic visualization of the image formation process and a simulated automatic accommodation. However, taking this as a base and adding new features to the existing architecture proved to be difficult. The code behind the simulation turned out to be too monolithic in places, making it hard to add new functionality in a clean way. At the same time it was unnecessarily decentralized in other places, sometimes leading to unwanted interactional effects in unrelated parts of the simulation when adding new functions. Due to these issues, at some point, the code of the eye simulation was completely remade from scratch with a new modular architecture. Each individual subsystem of the simulation (Figure 4.2), like the eye lens, the pupil or the retinal image, received it's own behavioral script, breaking the initial monolith. These modules encapsulate behavior, that concerns only the respective part of the eye (e.g. the eye lens script setting the blend shape of the lens deformation, when the focal length has been changed).

Coordination between these individual parts and the actual simulation of overall eye behavior is split into two areas of responsibility. A lens simulation handles everything that has to do with accommodation and focal length calculation, while the pupil simulation deals with adaptation and the resulting image brightness.

4.3.1. Lens Simulation

The first of two sub-simulations concerns itself with image projection and the process of accommodation. It's central piece, that orchestrates all the involved elements, is the "Lens-Manager" script. This script implements the equations introduced in subsection 3.1.2 to determine the focal length of the lens and to simulate the proper projection from the viewed object (Figure 4.2, annot. 1) to the final image on the retina (Figure 4.2, annot. 7).

It has to be noted that here the focal length is not actually calculated based on the current radii of the lens mesh. The employed animation for the deformation of the lens and the resulting radius changes, purely serve the purpose of illustration. We want to merely visualize the basic concept underlying accommodation and thus an approximation of the refractive process will be sufficient. So instead, the lens script simply manages an internal floating point value for it's focal length with a preset minimum and maximum value. Only in the second step, the deformation state of the model's lens is set, based on this value. Through this we achieve a behavior that roughly matches the physiological processes in the real eye, where the curvature of the lens is increased to put closer objects into focus.

The LensManager can be configured to three different modes. In the mode "Off" the focal length always remains static, while in "Manual" mode the user is given direct control over it. Lastly, for the "Auto" accommodation mode, the focal length is adjusted to make sure the sharpest projection always falls directly onto the retina. To do so in this last mode, we can rearrange the thin lens equation from subsection 3.1.2 to get the required focal length:

$$f = \frac{1}{\left(\frac{1}{d_{ret.}} + \frac{1}{d_{object}}\right)}$$

Here, $d_{ret.}$ is the distance between the retinal image and the principal plane of the lens and d_{object} is the distance between the same plane and the viewed object.

During the other two modes (manual accommodation and no change in focal length at all), the basic purpose of the lens simulation is to visualize any discrepancies between the position of the retina and where the incoming rays actually converge back together into a sharp image. Therefore, we introduce a "theoretical image" into our simulation (Figure 4.2, annot. 6). This is a copy of the image that is placed at the convergence point, to indicate where, in theory, a sharp image would form for the given optical setup. This also allows us to infer the level of blurring we need to introduce to the retinal image, from the size of the gap between the theoretical and real image.

The simulation has to first calculate the position and scale of the theoretical image for the current state of the lens and the viewed object. For the distance of the theoretical image from the lens, we can again rearrange the thin lens equation:

$$d_{theo.} = \frac{1}{\left(\frac{1}{f} - \frac{1}{d_{object}}\right)}$$

Next, we can use the formula for lateral magnification outlined in subsection 3.1.2, to calculate the height of the theoretical image H_{theo}, given the height of the viewed object H_{object} and the distances from both elements to the lens plane, as:

$$H_{theo.} = -\frac{d_{theo.} \cdot H_{object}}{d_{object}}$$

The negative sign of the resulting value denotes that the image is projected up side down. We therefore flip the associated illustration on it's head and scale it according to the absolute value of the height parameter. Using the intercept theorem, we can now easily derive the height of the retinal image as well:

$$H_{ret.} = \frac{(H_{theo.} \cdot d_{ret.})}{d_{theo.}}$$

Lastly, we need to compute a blur strength factor $b \in [0, 1]$ for the retinal image that serves as the input for a blur shader, which we will discuss in more detail later in this thesis. If the theoretical and the retinal image are located at the exact same position, we assign the value $b = 0$, denoting a completely sharp image on the retina. The blur strength then grows linearly for increasing distances between the two images, until $b = 1$ is assigned for their maximum possible distance in the simulation.

4.3.2. Pupil Simulation

The second part of the simulation is all about the eye's response to light and it's influence on the seen image. As before, this sub-simulation is organized in a modular architecture. First of all, a Pupil module manages the current pupil diameter value of the simulation. The diameter is given in the local space of the eye model, with a preset minimum and maximum diameter. Whenever the diameter is changed, the pupil script makes sure to update the blend shape deformation of pupil and iris, so that the opening distance of the model's pupil matches the current diameter.

Additionally, this half of the simulation again involves the viewed object, the retinal image and the theoretical image. All of these three have an assigned brightness value, that is given as a floating point factor in the range $[0, 1]$. A brightness factor of 1.0 equates to the highest possible level of illumination, while value 0.0 denotes extremely weak lighting. With this, we can simulate and visualize the relationship between object brightness and image brightness for the given pupil dilation state. In the final application, the object brightness is directly controlled by the user, while the brightness of the theoretical and the real retinal image is automatically updated, based on this incoming brightness and the current pupil diameter.

The actual simulation code is located in the "PupilManager" script. It is responsible for updating the pupil diameter, based on light exposure, and for the brightness update of the image. The PupilManager has three possible adaptation modes. Mode "Off" deactivates the pupil simulation and sets all the involved brightness values to an unchanging, neutral value of 0.5. In "Manual" adaptation mode, the user has manual control over the pupil diameter, to explore how the diameter affects the resulting image for different object brightness states. Finally, the "Auto" adaptation mode enables the automatic simulation of the pupillary light reflex.

Automatic Adaptation

In the automatic adaptation mode, the pupil diameter has to be dynamically adjusted in real-time, based on light exposure or the lack thereof. In pursuit of a physiologically sound simulation behavior, we turned to the available literature. There turns out to be a myriad of potential mathematical models to choose from. Starting in 1926 with the work of Holladay [27], a stream of research has emerged that aims to abstract a mathematical formula for the pupillary light reflex from empirical data, gathered from study participants. This same pursuit was continued by various other authors, including Crawford et. al in 1936 [28], De Groot and Gebhard in 1952 [29], as well as Stanley and Davies in 1995 [30].

It has to be noted, that all of these models are only concerned with the pupil diameter after the pupil has reached the so called "equilibrium state". This means that the pupil size has been given time to stabilize for a given level of illumination before taking a measurement for the underlying dataset [31]. This ignores any dynamic changes in pupil size when the eye is exposed to a sudden change in lighting. While there exist some approaches for fully dynamic pupil simulations, like the method proposed by Pamplona et. al. [32], that works with a delay differential equation to simulate the pupil diameter of characters in real-time computer graphics, this would be overkill for our particular use case. For the purpose of visualizing the basic adaptation process to students, the more simple formulas that assign a pupil diameter to a given input illumination are more than sufficient.

In the end, the choice fell on a frequently referenced empirical model, that originated from Moon and Spencer in 1944 [33] and was later updated to modern SI units by Watson and Yellott, resulting in the formula for pupil diameter D [34, eq. (7)]:

$$D(L) = 4.9 - 3\tanh(0.4 \log L)$$

The expected input here is a "luminance" value L between 10^{-4} and 10^4. Luminance is a measure of light intensity (more specifically "luminous intensity") over the area of a given light-emitting or -reflecting surface, measured from a specific viewing direction [35]. The accompanying unit is $cd \cdot m^{-2}$. Any more extensive simulation would require us to dynamically determine the luminance reaching the eye, based on the environment's light sources, their strength and the relative viewing position and angle. However we are merely interested in maintaining the rough shape of the curve shown in Figure 4.4a for our own pupil diameter evolution. So instead, we just perform a simple nonlinear mapping of the viewed object's brightness value $B_{object} \in [0, 1]$ into the given luminance interval $[10^{-4}, 10^4]$:

$$L = 10^{-4+8 \cdot B_{object}}$$

Following this simplification, the pupil diameter of the simulation is only dependent on the total brightness of the object in front of the eye, also fully disregarding it's distance. In the last step, a linear mapping is applied from the diameter in millimeters, as output by the Moon-Spencer formula, onto a pupil diameter in local space of the eye model. This procedure results in pupil behavior that follows the shape outlined in Figure 4.4b. As was the goal, there are only small changes in diameter in the very bright or very dark ranges, whereas a higher rate of change occurs for the mid-range brightness values.

Image Brightness

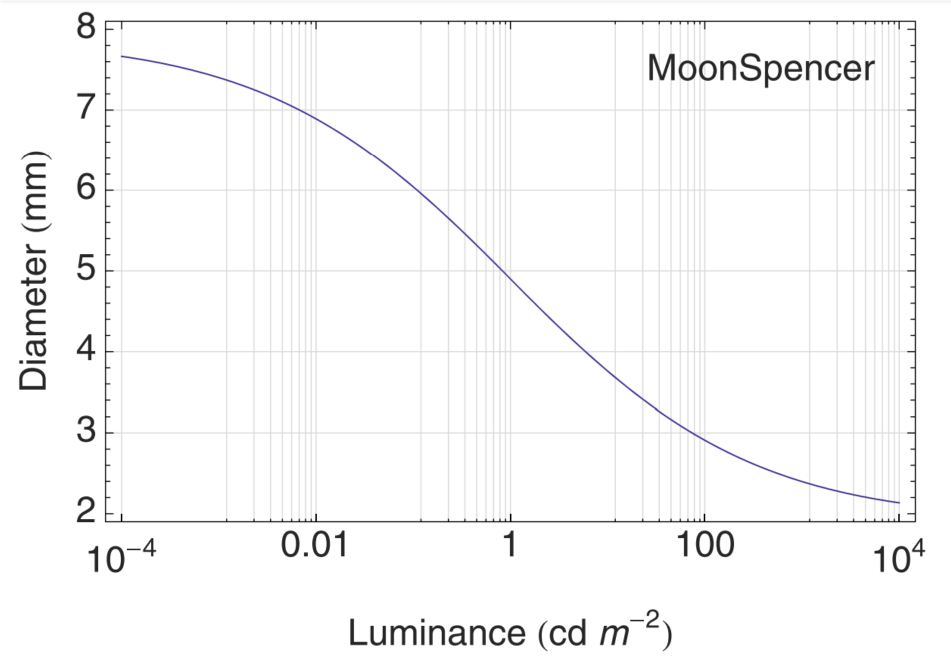
With the pupil diameter properly set, either automatically in the fashion outlined above or manually by the user, the next step is to determine the projected image's brightness for the current simulation state. The goal here is to scale down the incoming brightness according to the current degree of pupil dilation. The smaller the pupil, the less light can pass through and the image becomes darker. To accomplish this, a scaling factor for the input brightness is introduced. The factor is based on the ratio of the two circle areas for the current diameter D and the maximum possible diameter D_{max}, respectively:

$$\frac{A_{current}}{A_{max}} = \frac{(\frac{1}{2} \cdot D)^2 \cdot \pi}{(\frac{1}{2} \cdot D_{max})^2 \cdot \pi}$$

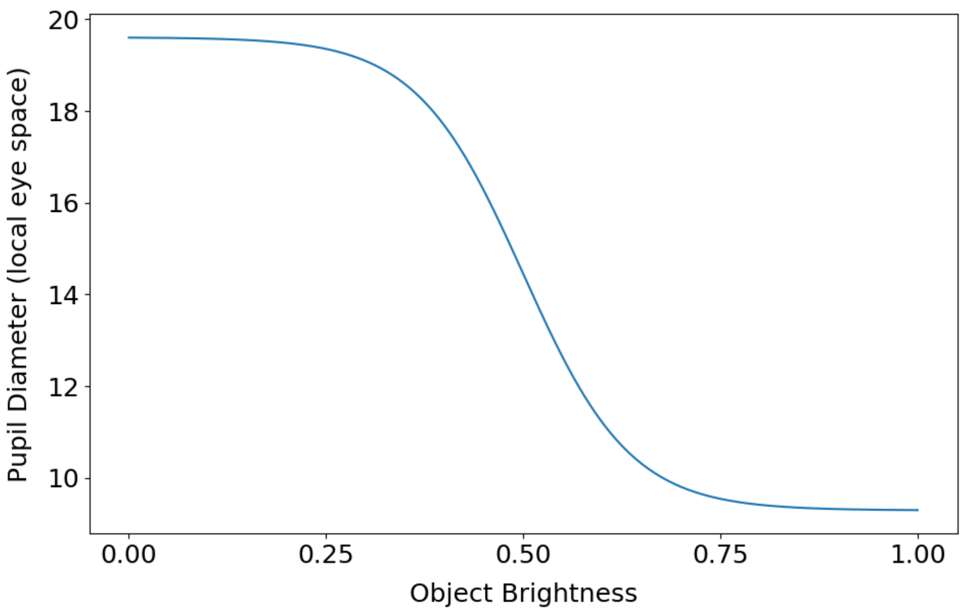
For this expression, the π as well as the $(\frac{1}{2})^2$ can be omitted. By adding a dampening constant $C_{damp.} \geq 0$ to both diameters, we arrive at the final image brightness formula:

$$B_{image}(B_{object}, D) = B_{object} \cdot \frac{(D + C_{damp.})^2}{(D_{max} + C_{damp.})^2}$$

When the pupil is at it's maximum dilation (i.e. $D = D_{max}$), the full, unaltered input brightness is assigned to the image. For diameters below that, the brightness is gradually reduced until it reaches a lower limit ≥ 0, given by the minimum pupil diameter. The addition of the dampening constant $C_{damp.}$ opens up the possibility to optionally tone down the degree of brightness reduction. This helps with fine-tuning the effect to achieve the desired visual results for the simulation.



(a) Adapted Moon-Spencer pupil diameter formula on a logarithmic scale. Taken from Watson and Yellott [34]



(b) Plotted simulation values, showing the pupil diameter in local space of the eye model for a given input brightness

Figure 4.4.: Plotted graphs for the Moon-Spencer formula and our simulated automatic adaptation

4.4. Interaction Methods

Through the course of the exercises, the user will be asked to interact with various parts of the eye simulation, to see firsthand what impact specific changes have on the image formation process. All of these interactions are performed via the two wireless controllers of the VR system. In the application they are represented as two disembodied virtual hands.

The user can directly manipulate the state of the simulation by two means. Some elements of the simulation can be affected by directly grabbing them and pulling them around. For instance, the viewed object in front of the eye can be grabbed and moved along the optical axis to increase or decrease it's distance from the eye. To engage this interaction, the user has to position one of the virtual hands inside the bounding box of the respective element and press the controller button under their index finger to grab it. However, not all simulation parameters are based on positional changes and are therefore not suited for a grab and pull interaction. An example for this is the brightness level of the viewed object. These values can instead be adjusted via a control panel on the working table (Figure 4.1, annot. 3).

A third method of interaction does not revolve around the simulation itself, but is instead used to engage with a Graphical User Interface (GUI). Our application can make great use of a GUI's text output capabilites (e.g. for educational content or exercise instructions) and of their highly flexible forms of interaction, which should be well known to most users. Throughout the rest of this thesis we will use the terms GUI and UI (short for User Interface) interchangeably. While technically every interactive element of the application represents a user interface, everyday technical language rarely makes a distinction between the two terms. An example of this is Unity itself [36].

To manipulate the individual UI elements, a basic point and click interaction is employed. An image of this interaction can bee seen in Figure 4.5. A pointing ray, visualized by a straight white line, will extend from the tip of the controller, whenever it is aimed at an object in the environment that has a graphical interface. The "click" part of the interaction is triggered by a press of the index finger button. There are two elements in the application workspace that support this interaction; the exercise blackboard and the options screen for changing the simulation settings (Figure 4.1, annot. 4).

While all these interactions have been designed to be as intuitive as possible, we want to make extra sure that every user, regardless of their level of technological literacy, knows what to do at all times. We therefore provide the user with an additional written clarification, whenever a new type of interaction is first introduced.

In the following we want to present some of the interactional elements of the application in more depth. We will touch on the underlying design decisions and reflect on our considerations regarding user experience.

Figure 4.5.: Pointer interaction on a world space GUI

4.4.1. Control Panel

At multiple points during the exercises the user has to adapt some parameter of the simulation. This includes changing the focal length of the lens, the diameter of the pupil and the brightness value of the viewed object. All of these three are linear parameters with fixed minimum and maximum values. Therefore they lend themselves perfectly to be set via some type of slider, where the user can gradually move a handle to increase or decrease the underlying value within the preset boundaries.

In the EyeLearnRoom project, that served as the starting point for this application, this function was fulfilled by "physical" levers on the working table. They could be grabbed with one virtual hand and moved along a linear path to change the associated value. Two of these table levers can be seen in Figure 4.6. However, this was ultimately a highly impractical solution, in need of a complete rework. First of all the positioning on the table, right below the user, meant that they had to look downwards to see the levers. This in turn makes it impossible to observe the simulation at the same time, which is located straight ahead. This is already an undesirable state in it's own right, as good practice dictates that the user always receives clear feedback for any interaction. When the cause (lever movement) and effect (simulation change) of their actions are never visible at the same time, this principle is violated.

Even worse, the fact that the levers can't be seen while watching the simulation, actually impairs the proper functioning of the interaction. The reason for this, lies in the way controller motions are tracked in the used VR System. The HP Reverb G2 Controllers feature inertia

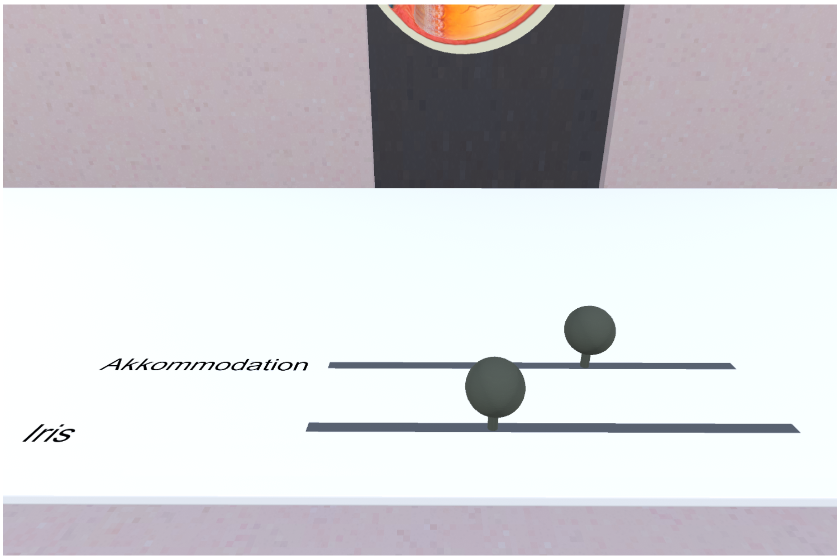


Figure 4.6.: Old table levers for setting linear simulation values

sensors to detect basic acceleration and rotations and a series of LED lights [25]. The latter are used to detect them as part of the HMD's "inside-out tracking" [37]. In this method, the HMD observes it's surroundings via four cameras and uses these video feeds to orient itself in the room. At the same time, it can also determine the relative position and orientation of the controllers, based on the visible LED pattern. The cameras are placed on the front and to the sides of the headset.

The interaction with the table levers below the user, therefore places the engaging controller outside the field of view of the tracking cameras. This leads to a significantly worse quality of motion detection. In the context of manipulating the linear simulation parameters, this manifested as jittery behavior, sudden jumps in values or as the interaction becoming unresponsive altogether.

Taking this into account, we can derive a fundamental design principle for avoiding such issues: Any interactible object that requires some form of motion controls has to remain in vision at all times during the engagement. Following this principle, we have two options for fixing the outlined problem. Either we make sure, that the grabbable handles can always be seen, or we switch to a different type of interaction that doesn't require motion controls. The first option is not really actionable in our case. We want the user to be able to get as close as they want to the simulation to properly observe any changes while they adapt the parameters. As an example, when the viewed object is far away, and the image therefore small, the user should be able to move their head right up to the retinal screen to see any changes in sharpness when they change the lenses focal length. Following this condition, there remains no place where we could put the lever handles, for them to always stay in vision without blocking the actually important content.

This then leaves us with moving away from the hand grabbed levers altogether. As an alternative, we introduce UI sliders on a world space canvas right below the eye model. Manipulation of this slider happens via the controllers' joysticks. Pushing one of the sticks to the left, while the slider is active, decreases the accompanying value, while pushing it to the right increases the value. The resulting interaction always works, whether the virtual hands or the slider itself are in vision or not. This alleviates the problems when observing some part of the simulation from a very close distance. At the same time, for all other (more common) use cases, the placement of the UI canvas right below the simulation, ensures visibility of both at the same time to give the user the proper level of feedback.

Since we have three different linear values that the user can manipulate, we need to provide them with a way to swap out which one of them is currently being controlled. We also want them to be able to disengage the slider interaction altogether and hide the UI canvas if it's not needed. To fulfill these functions, we introduced the control panel, that can be seen in Figure 4.7. When pressing a virtual hand down on one of the buttons on the panel, the UI slider for the associated parameter is activated. Pressing the same button again disables the interaction and hides the world space UI. When a button is pressed while another one is already active, the current UI slider is replaced to fit the new selection.

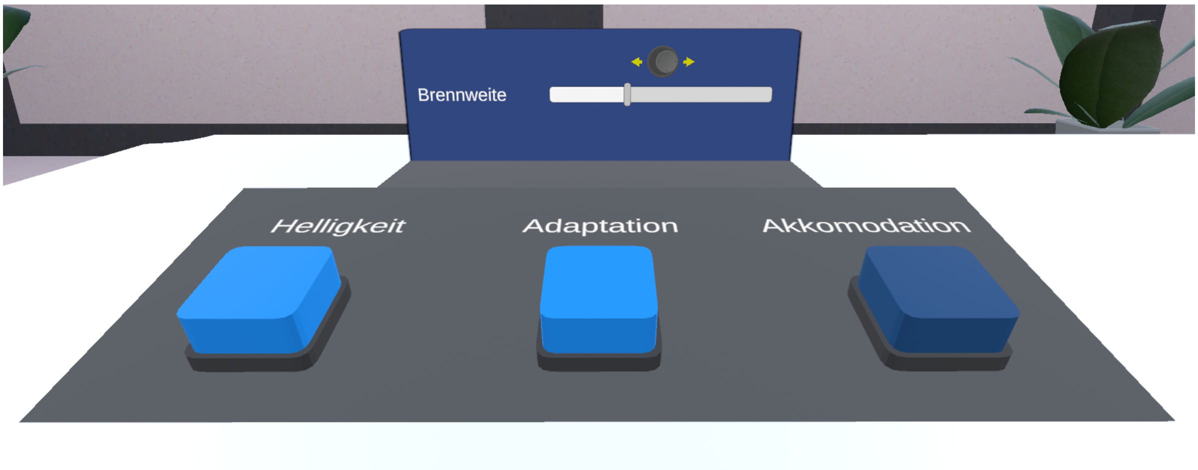


Figure 4.7.: The control panel with an active value slider for focal length

4.4.2. Exercise Blackboard

The educational nature of the application comes with some inherent requirements. For example, there needs to be a way to display informational content to the user. Also, we have to provide them with some interface to engage with knowledge retrieval questions. Previous VR applications of the Au.Ge project featured small clipboards to fulfill these functions. The grabbable clipboards featured an A4 sized sheet with a world space UI Canvas to display information or single choice questions. One of these clipboards can be seen in Figure 4.8.

Beside the obvious need to extend this functionality to support multiple choice questions, as needed in the practical course application, there were a number of additional problems with this design.

Figure 4.8.: Two handed clipboard usage in previous applications

First of all, the small size of the clipboard presented some usability issues. To ensure proper readability, the user necessarily had to pick it up and hold it directly in front of them. This fully occupies one hand, leaving only one controller able to engage with the application content in situations where instructions need to be visible while working. The handheld nature of the clipboard also opens up the possibility to accidentally drop it, forcing the user to stoop over to retrieve it. Depending on the usage environment, this poses a potential risk of harm or injury, as the user's head could collide with a physical object not visible to them with the HMD on. Further, some users have reportedly managed to get the clipboard stuck in places, that were not accessible given the real physical surroundings, leaving a restart of the application, including repetition of all previous working steps, as the only option to move forward past that point.

In addition, there were some ergonomic concerns with the clipboard design. As with the use of real world pen and paper, or small handheld displays, the virtual clipboard promotes a flexed neck posture. The already significant negative effects of such a posture are even further amplified by the added frontal weight of the HMD, leading to an increase in muscular stress and perceived pain or discomfort [38]. Secondly the two handed usage led to a severely unnatural arm position for the side with the pointing controller. The most accurate and arguably the most intuitive way to engage in a pointer interaction, is to point in a direction perpendicular to the UI plane. To achieve this perpendicular angle of incidence with the clipboard held in front of the user with one hand, the second arm is forced to lift at the shoulder, at a near 90° angle, and the wrist to strongly bend sideways. This represents a highly uncomfortable arm position, particularly for extended engagement.

As a solution to all of these issues, we decided to move away from the clipboard and introduce a blackboard in it's stead. The large landscape format board services as the new medium for a world space UI. This alleviates all of the problems, stemming from the clipboard's small size. A blackboard does not need to be carried around and is readable even from across the virtual room, while both hands are free to engage with the simulation content. The ergonomic risk factors are eliminated as well. The floating blackboard is set around the user's eye level and all content is visible from a neutral head position, while the pointer interaction can be performed with a relaxed, hanging arm, using only slight elbow and wrist movements.

The blackboard's UI contains several elements that can be activated or deactivated, depending on the current use case. This includes a board filling text box for all types of informational content or as a smaller text box for multiple choice questions, accompanied by a collection of answer options with checkboxes to cross or uncross. When reviewing the correct solutions of a question, these checkboxes are deactivated to prevent a retrospective change of answers, while the accompanying answer text is colored green or red, according to the statement's correctness. The bottom of the board features two navigation buttons, allowing the user to move back and forth between a series of displayed panels. Figure A.1 of Appendix A shows some more screenshots of the exercise blackboard, displaying various forms of content.

4.4.3. Options Screen

Generally throughout the exercises, the background settings of the eye simulation are set automatically, based on the current exercise context. As an example, Exercise 3 is "scripted" in a way, where the automatic pupillary light reflex is activated half way through the exercise. However, in the sandbox mode, after the official exercises have been concluded, the user should be given full control over the simulation. We therefore need to offer them an interface through which they can switch between the different behavioral modes of the eye simulation as they please. To this end, an options screen appears in the corner of the table, once sandbox mode has been entered. A medium sized computer monitor acts as host for another world space UI canvas, that again supports the controller based point and click interaction.

In the given scenario, we have a small set of mutually exclusive options for each parameter, that the user can choose from. This lends itself perfectly for a form of UI Unity calls "Toggle Groups" [39]. These are sets of individual checkboxes, where only one of them can be selected at a time. In UI design, this pattern is more commonly referred to as "radio buttons" [40, 41]. While radio buttons oftentimes appear as a list of horizontally or vertically stacked checkboxes with their labels on the side, we decided to stylize our toggle elements as larger rectangular buttons that change color when switched on, with the labels drawn on top. Not only does this result in an appealing look that can be seen in Figure 4.9, it actually improves the interface's usability. The buttons provide a bigger target for the pointer interaction than standard checkboxes would. While the same topic is of no particular concern for the much larger exercise blackboard, the mentioned change helps to greatly improve the operability of the comparatively small options screen.

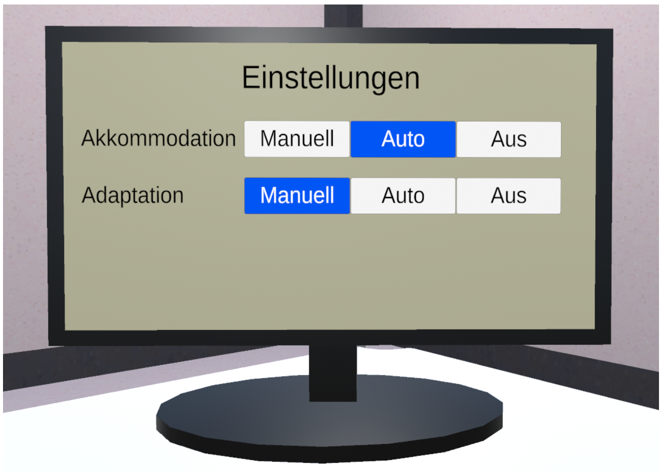


Figure 4.9.: User interface of the options screen

4.5. Exercise Management

When using VR in a serious educational context, we can benefit from many of it's unique capabilities. One of the biggest strengths of VR lies in the possibility to engage with virtual content "hands on", within an immersive environment. Here, the user is provided with natural and intuitive ways to interact with the virtual world (e.g. grabbing objects, throwing things, pulling levers, etc.). At the same time, the objects they interact with are not bound by the limitations of the real world, which opens up near limitless possibilities to present novel concepts to a user of a VR platform.

However, we still don't want to pass on some more traditional forms of learning material, like textbook content, visual learning aids or knowledge retrieval questions. Yet, we want to include these materials in a way that doesn't break the user's immersion (e.g. forcing them to take of the HMD to engage with content on physical paper) and therefore need to neatly integrate them into the virtual environment itself. This provides a bunch of interesting possibilities to interweave interactive VR content with these traditional forms of learning, but it also presents some technical challenges that developers of such an application need to address.

If we want to provide a guided learning experience, we will have to present the individual taught concepts in a specific order, where later concepts build on the knowledge acquired in the earlier ones. All throughout, we need to sync up the interactive content and the auxiliary learning material with the currently covered concept and with each other. If the user goes on to a new learning unit, they might need to be presented with a new virtual object to interact with and will be asked follow up questions after the engagement. All this needs to be coordinated somewhere in the applications code, optimally in a way that's easy to adapt and to extend when new features are introduced to the system.

It is worth noting, that the same principle is in no way exclusive to VR applications alone. All types of educational applications that combine the interaction with illustrative content or simulations of real world concepts with traditional learning materials in a guided learning experience, will benefit from a clean software design to coordinate all this. Some other domains that come to mind, where this is oftentimes the case, are educational AR applications or Serious Games.

In this section, we present a general purpose architecture that deals with managing all the involved sub-systems during the guided engagement with a series of exercises. While the architecture was designed as part of the creation of the LIN-VR application, we made special care that it is sufficiently generic and flexible enough to be adapted to other use cases. With this, we hope to provide other developers, facing the same challenges, with a comprehensive, yet powerful architecture that might serve as a readily usable package, as a starting point to be built upon or as an initial inspiration for their own design.

4.5.1. Requirements

We were given a set of general requirements and demands on our exercise management system. Basis for the engagement with the exercises should be a GUI, drawn on a virtual medium in the application environment. As mentioned in subsection 4.4.2, this function was ultimately assumed by a floating blackboard. Through the GUI, the user can view and interact with exercise content, like multiple choice questions or instructions for interacting with the eye simulation. The individual sub-tasks of the exercise should be shown on separate panels, where only one panel is active and visible on the medium at a time.

Therefore various panel types have to be available, to support different forms of content and interactions. As an example, the application requires one type of panel that can display information text, without any additional interaction, while another panel type should show one isolated multiple choice question to the user and allow them to cross the statements they deem correct. The user should then be able to move through a linear series of these exercise panels via the two navigation buttons on the blackboard, that allow jumping to the next or the previous panel of the chain, respectively.

At the same time, the exercise management architecture should be able to access and make changes to the virtual environment, including the eye simulation. In cases where the exercise context changes, the state of the simulation or other interactional elements may have to be adapted, so that the interactive content fits what is covered in the exercise. To again give an example, the first part of Exercise 3 allows the user to set the pupil diameter themselves, while the second part covers the automatic pupillary light reflex. Here, the mode of the pupil simulation has to be changed from manual to automatic adaptation, once the first sub-task panel of the second phase is shown. This should happen in the background, without the user needing to initiate these changes themselves.

In addition, we had some secondary demands of our own on the architectural design. Not only should it be flexible enough to incorporate all of the aforementioned requirements, it should also be easily expandable to support new types of exercise content in the future (e.g. images as part of info panels, calculation tasks, quiz formats other than multiple choice). Secondly, as mentioned previously, we did not want our architecture to be bound to the specific use case of this particular application only. We wanted to provide a reusable solution that is versatile enough to plug into all kinds of educational applications, VR or otherwise. At the same time, the resulting code should be easy to read and easy to maintain.

4.5.2. Our Architecture

As a result of all these considerations, we designed a multi-layered, fully modular exercise management architecture. An overview of this architecture can be seen in Figure 4.10. At the top layer of the hierarchy sits the "ExerciseManager" that orchestrates what content is shown on the blackboard and what state the eye simulation is in, at any point during the exercise it belongs to. To this end, the ExerciseManager sequentially processes a series of "exercise steps" that define the system's behavior during the associated sub-tasks. In the layer below that are sub-manager scripts that encapsulate some more specialized functionalities. This serves the purpose to clearly separate responsibilities and to avoid forming a monolithic structure. The ExerciseManager still remains in charge of anything that happens, but it delegates some tasks to these specialized modules. In the case of the LIN-VR application, this layer is populated with only one sub-manager that handles the display and scoring of the multiple choice questions. At the very bottom, we have the user interface of the exercise blackboard and a wrapper script that acts as an interface between the UI and the upper layers. The interactive content of the application domain (in our case the eye simulation) exists outside the layered structure and is depicted as a blackbox somewhat separately to the side. Yet still, the ExerciseManager is given full access and control over any part of the simulation. This way, we can properly synchronize the simulation behavior with the current exercise content. In the following we will take a closer look at the individual modules of this architecture and provide some details on their implementation.

UIWrapper

As the name suggests, the UIWrapper script provides a wrapping around the user interface of the exercise medium, which in our case is the GUI of the exercise blackboard. In Figure 4.10

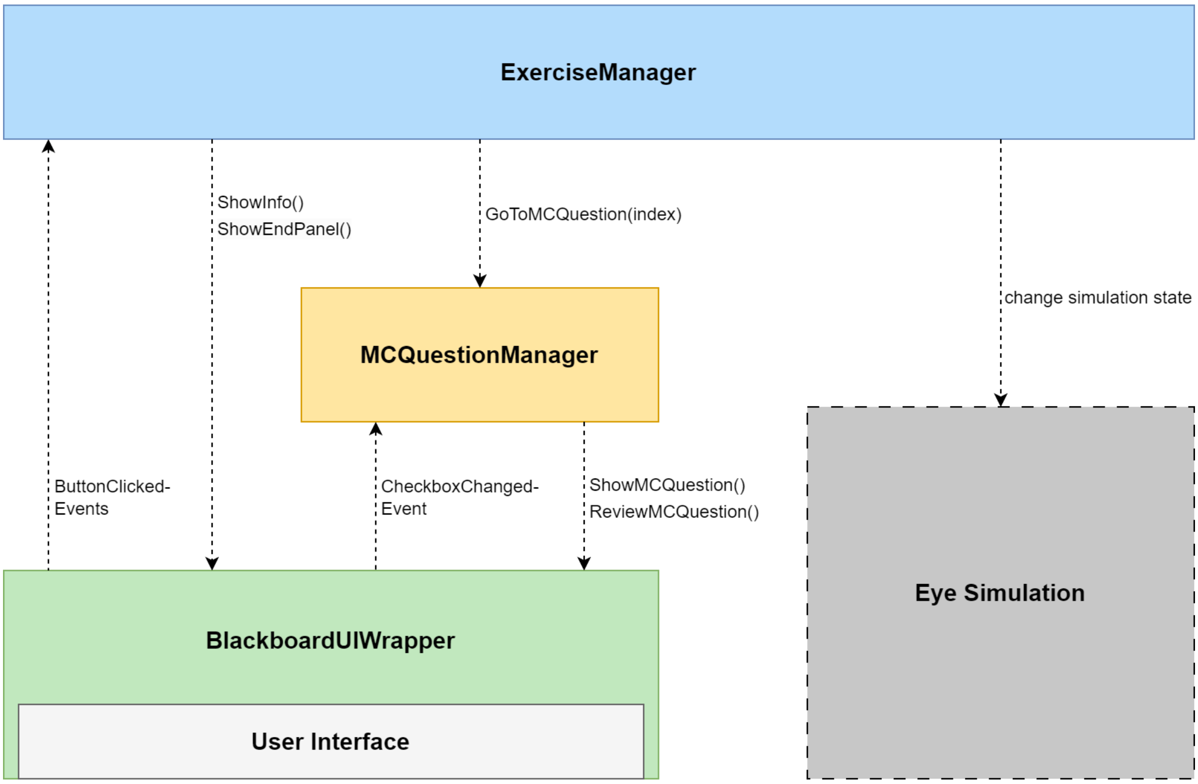


Figure 4.10.: Schematic overview of the exercise management architecture

this corresponds to the second lowest layer of the architecture. The UIWrapper provides a clean separation between the user interface and the modules in the higher layers.

While in this application the specific implementation is called "BlackboardUIWrapper", the module and the overall architecture are in no way limited to this specific medium. An adapted UIWrapper can be assigned to any form of interactional UI (e.g. clipboard, handheld tablet computer, ancient stone tablet, etc.). If a new type of interaction medium or a new UI structure is introduced, a developer only needs to make corresponding changes to the wrapper script, while all other modules and their functionality remain unaffected. This ultimately makes the code much easier to maintain.

The wrapping itself, for one, entails taking in a user's UI inputs and notifying the modules that need to know about them. The UIWrapper listens for UI events and re-packages them to raise it's own interaction events. The other modules of the architecture subscribe to these secondary events and perform their desired functionality once the user performed the assigned input action.

Secondly, we need a clean interface to access the output functionalities of the UI from other modules of our system. The blackboard's individual UI elements are grouped into views for different types of content. These correspond to the panel types for different sub-tasks described in subsection 4.5.1. When a specific view is displayed, it's respective elements are drawn on the UI canvas, while all non-related elements are being deactivated and hidden. For the special use case of the Au.Ge project, three available views are implemented. An information view for displaying text based info content (Figure A.1c), a multiple choice question view (Figure A.1a and A.1b) and a results view for the last panel of the exercise (Figure A.1d). This result panel shows the reached point score for the current pass. Depending on whether the score exceeds a required point limit, this view also displays and enables the finish button for the exercise, allowing the user to move on to the next one. For other use cases this script can be extended by new views with different input/output modalities (e.g. a number pad view for input of numerical values).

The UIWrapper now provides public methods to display the information from other modules as part of these views. As an example, the MCQuestionManager has access to the method "ShowMCQuestion". Via this call, the question manager can pass on the string variables for question and answer texts, alongside meta information about which answer boxes have been checked during the last visit to that same question panel. Following one of the show methods, the UIWrapper first switches view, if neccessary, and then fills the UI elements with the passed information and sets any interactible elements to the desired preset state. The UIWrapper explicitly does not keep any persistent information regarding the displayed content. It simply shows what it's told to show and discards all content of the old panel, when a new one is requested to be displayed.

Question Manager

The script "MCQuestionManager" encapsules everything that directly concerns the multiple choice questions posed to the students. The questions themselves are stored in a JSON file [42], which contains the strings for the question and the five answer options, alongside boolean values indicating the correctness of the individual answers. The question manager first parses the JSON file associated with the current exercise to load these questions into memory. When the ExerciseManager then issues a request to show a certain question, the MCQuestionManager passes all required data of that question to the UIWrapper for display. While the user interacts with a question, the module is informed of any changes to the answer checkboxes, through an event of the wrapper script. The question manager keeps a persistent record of these checkbox states over the course of the entire exercise. This way it can, for example, pre-tick any previously selected boxes when a question is shown for the second time. The information is also used in a public method that compares the current answer choices with the correct boolean values, to return the total point score (1 point for each correctly toggled checkbox). Combined with a second method, that returns the maximum possible score, this enables the ExerciseManager script to determine if the passing percentage threshold has been reached for the current exercise.

Exercise Manager

As mentioned several times before, the ExerciseManager script is the highest controlling authority within our architecture. One instance of the manager is responsible for a single exercise from section 4.2. The functionalities of the module can be divided into two categories: shared behavior between all exercises and features that are specific to one individual exercise. In our design, all the shared parameters and functionalities are encapsulated in an abstract parent class. These include, for example, the pass/fail check at the end of an exercise to see if the required amount of points has been reached. The abstract class also contains a parameter for the current exercise phase, that's updated based on the result of this pass/fail check. Every exercise starts in "working phase", where the user can interact with the questions and select their answer options. If the passing threshold is met for a run-through, then the system switches to the "review phase". Here the user can look back over the correct answers for each question or directly move on to the next exercise, without reviewing. The review phase is also entered after the second failed attempt to gather enough points, as to not "trap" the user for too long if they already seriously engaged with the content for two tries.

To implement the specific individual behavior of an exercise, we now create a dedicated sub-class that inherits from the abstract ExerciseManager class. Each of these sub-classes is required to overwrite a set of abstract methods defined (but not implemented) in the parent class. These include a preparation method, containing all commands that have to be performed at the start of the exercise (e.g. activate automatic simulation of accommodation at the start of Exercise 2), and a cleanup method that returns the application to a neutral state after the particular exercise is finished. With this object oriented paradigm, the classes for the individual exercises are kept short and readable, while the general shared functionality is still easily accessible through the base class implementations.

ExerciseStep Pipeline

Up to this point, we have looked at some exercise-wide meta behavior and at the behavior for beginning and end of an exercise. To now understand what happens during the exercises and their sub-tasks, we introduce the true centerpiece of our design, a vastly flexible pipeline for so called "exercise steps".

Let's first recap some of the requirements. During the exercises we go through a series of informational content and sub-tasks, all presented on different panels on the blackboard. At the same time, to keep the interactive content synchronized with the topic on the current panel, we at times need to change the simulation mode, set the parameters of individual eye elements to a certain value or activate/deactivate certain objects in the environment. Looking at this, we define an "exercise step" as the collection of every command or system behavior that has to occur when we switch from one panel to the next. Figure 4.11 shows an exemplary chain with three such steps from one of the implemented exercises, including the panels displayed and the associated background behavior.

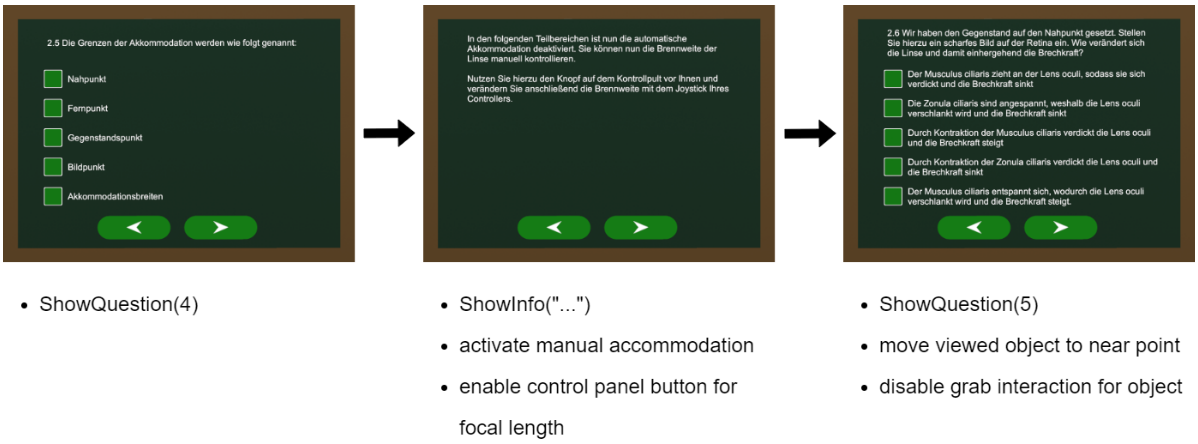


Figure 4.11.: Example of a series of steps with associated behavior during Exercise 2

The following C# codeblock contains the definition of the "ExerciseStep" struct used in the pipeline:

```
public enum StepAllowed { Always, OnlyInWorkingPhase, OnlyInReviewPhase }

public struct ExerciseStep
{
    public Action action;
    public StepAllowed allowed;
```

```
   public ExerciseStep(Action a, StepAllowed sa)
   {
      action = a;
      allowed = sa;
   }
}
```

First of all, we can use the "allowed" parameter to limit during which exercise phase the current step is performed. This enables us to hide some information panels that are no longer needed during review or, for the opposite case, show some "review phase only" info content with additional explanations for the correct results. However, the more important part of the ExerciseStep struct is the "action" parameter. A .Net "Action Delegate" [43] acts as a sort of container for any type of method that has no input or output parameters. This means, that any method that fits this signature can be assigned as the "value" of a variable with type Action. In many contexts, such delegates are used to dynamically change the behavior of a function call at runtime, since an Action can also freely be re-assigned with another method, at any point. We are however more interested in the basic fact, that we can use Actions to create the generic ExerciseStep struct outlined above. This allows us to create a list of these steps, with vastly different behavior, and still process all of them in the same fashion as part of the underlying pipeline. Finally, we can call the assigned method of these steps by simply adding a set of brackets after the responsible parameter's name (i.e. "action();").

The full flexibility of our approach now comes into play, when we combine Actions with another C# language feature called "lambda expressions" [44]. Using these, a developer can initialize an Action parameter with any arbitrary collection of code encased by two curly braces, without the need to explicitly define a method first. This concept can be seen in practice in the code block below. An override method in the individual sub-class scripts, called "InitializeSteps", is used to create and populate the list of steps for the respective exercise:

```
protected override void InitializeSteps()
{
   Steps = new List<ExerciseStep>();

   Steps.Add(new ExerciseStep(() =>
   {
      //As part of a step's Action you can:

      //show a blackboard panel (Info Panel, Multiple Choice Question or
         End Panel)
      blackboard.ShowInfo("This is an info panel");

      //change the state of the eye simulation
```

```
        lensManager.SetAccommodationMode(AccommodationMode.Auto);

        //or do anything else you like
        int sum = 2 + 2;
        Debug.Log(sum);
    },
    StepAllowed.Always));

    //define and add all the steps needed for the exercise, following this
        template
}
```

Here, input value for the "action" parameter of the new ExerciseStep is formulated as a lambda expression, using "() =>" before a block of code in curly braces. Within these braces, we have absolutely free rein to implement the exact step behavior we want. The options what we can do here are near limitless. As a reminder, the ExerciseManager module sits at the very top of the hierarchy and has full access over the lower layers, over the eye simulation and if needed, over any other parts of the application environment. Using the outlined syntax, the code block of a step's action behaves exactly like a class method would, without the need to explicitly declare one. This also means we can freely access and change any class parameters of the parent or child class, which allows us to even circumvent the "no input/no output" limitation of the .NET Actions. The code of a step Action can use references to these class parameters to read values that have been set beforehand (input) or assign new values for the use in the following calls (output). In turn, this opens up the possibility for even more complex behavior inside the steps, including communication and coordination between the individual steps.

As per the requirements, the navigation between the individual steps is tied to the navigation buttons at the bottom of the exercise blackboard. When the forward button (arrow to the right) is clicked, we search for the next entry in the step list that is allowed during the current phase and execute the associated Action. Likewise we search backwards in the list and perform the first valid step, if the backward button (arrow to the left) is pressed. While it is not technically required to include the display of a new blackboard panel in every step, we still highly recommend to do so, because otherwise there would be no immediate visual feedback for the user when clicking one of the buttons. At the start and end of the step list (or if no more currently allowed steps come before or after), we disable and hide the respective navigation button for that direction. Since all of this navigation behavior is the same for every exercise, we again outsource it to the parent class.

At this point it should also be noted that the background behavior in the annotations of Figure 4.11 are composed for the perspective of moving strictly forward through the panels. When taking the backwards navigation into account, there is in truth one more thing happening in the action of the leftmost panel. In transition from the first panel to the middle panel, the accommodation mode is changed to "Manual", since this is the simulation behavior required for the second part of Exercise 2. If we then go back from the middle panel to the left one, we likewise have to change the accommodation mode back to "Auto" to fit the first exercise part. The easiest solution to ensure this consistency is to use the associated command like "brackets" around the set of steps that share the same simulation context. In other words, we set the accommodation mode to "Auto" at the beginning of the first part and then set it to "Auto" again for the last step of that interval. When we move forward through these steps, this second call is redundant, as the state is already set accordingly, but it does no harm either. Yet, when we now move backwards from the second part of the exercise the "closing bracket" call triggers the necessary change and the simulation context remains consistent. This simple approach was sufficient for all requirements of the sort in this particular application. If another use case requires a finer level of control than this, the pipeline could easily be extended to use separate "Action" variables for forward and backward navigation, altogether.

Exercise Master

Each ExerciseManager controls a single exercise of the overall application. In order to switch between these individual exercises, we introduce one last module, we call the "ExerciseMaster". It works off a list of individual ExerciseManagers and handles the transitions from one exercise to the next. It comes into play whenever the finish button of the current exercise has been pressed. In that case, the master script engages the cleanup routine of the exercise that's coming to and end, before initializing the preparation of the upcoming one and launching it's first step. The ExerciseMaster also retracts access to the navigation button events from the old exercise and passes it to the new one. This is to ensure, that no two exercises "occupy" the UI at the same time.

4.6. Visualization and Graphics

To support the learning goals of our application, we need to provide the user with high quality visualizations and illustrative aids for the various topics taught. They should be able to see what's fundamentally going on in the human eye and be provided with clear visual feedback when interacting with the simulation. The main goal for the visualizations is to present the concepts in a way that's both engaging and clearly understandable to the user. So long as the point of a covered topic is getting across, the visualization is a success. We therefore value a high visual appeal and a generally plausible visual behavior over a 100% accurate representation, down to the last detail. Some of the effects are consciously exaggerated to ensure better readability of what is happening in the simulation.

The visuals of the simulation can be divided into three categories. The first one is the cross-sectional eye model itself. Besides acting as housing for all the other elements, it provides a foundational visualization for the anatomical components of the covered phenomena. As briefly mentioned in section 4.3, the 3D model comes with blend shape animations for mesh deformation. One of these modifies the shape of the model's lens, in the same way the actual eye lens changes to achieve different focal lengths. The second one deforms the iris in order to narrow or dilate the pupil. With these animations, the user gets to see the physiological changes the human eye undergoes for various scenarios.

The second category of visual content includes the depictions for the viewed object, the theoretical image and the retinal image. For illustrative purposes, the most important of these three is definitely the retinal image. It is the center piece for visualizing how any changes in the environment or the eye's physiology affect the final seen image. Here, we use a simplification to improve visual clarity for the user. Instead of being projected onto the curved and colored surface of the eye model's actual retina, the image is drawn onto a flat white screen that is positioned near the end of the eyeball (Figure 4.2, annot. 8). The flat monochrome screen allows the user to better make out any changes made to the retinal image. In the application the viewed object is represented by a flat uppercase letter "P" that sits on top of the optical axis. As a result, the two images are illustrated as instances of the same letter, only mirrored horizontally. In Unity all three of these Ps are rendered as a sprite on top of an otherwise transparent plane. Here, back-face culling has been deactivated, so that the respective element is still visible from both sides.

The last category includes additional illustrative aids that supplement the other visualizations. Through artificial overlays we can better illustrate some otherwise poorly visible or entirely invisible phenomena. For one, this entails the ray diagram for the eye lens (Figure 4.2, annot. 2), to help visualize the basic image formation process. Here, we are using the visual simplification for thin lenses mentioned in subsection 3.1.2 and only portray one single ray refraction at the principal plane. The resulting view is shown in Figure 4.12. The second form of overlay, used in the application, is a yellow light cone (Figure 4.2, annot. 3) to show the amount of light entering the pupil.

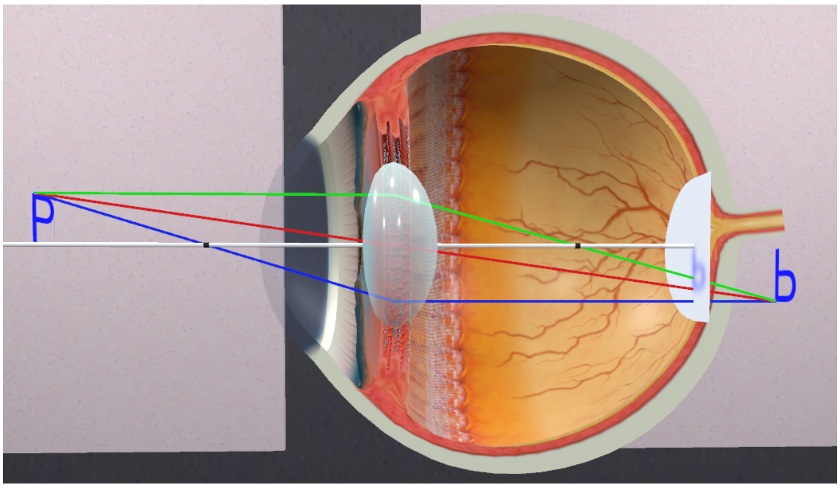


Figure 4.12.: Eye model with ray diagram during Exercise 1

In the following section we will dive deeper into the specific implementation for some of the mentioned visual effects and illustrative aids.

4.6.1. Blur Shader

One crucial part of the application is the depiction of blurred vision as a result of wrongly adjusted lenses. It is needed to visualize the importance of correct accommodation, the mechanisms behind sight defects or the effects of corrective lenses on the final image. All these scenarios call for the ability to adaptively control the sharpness of the image on the screen representing the retina.

Whenever the involved lenses are not optimally adjusted for the specific setup of eye shape and the observed object, the image distance of the lens system will not match the actual distance to the retina. This means that all the individual rays, which originate from a single point of the viewed object, are not converging back together at the correct distance. They either converge somewhere in front or somewhere behind the retina. In both cases, the light diffusing off a single point in the observed scene gets distributed over a localized area on the retina instead. This is what results in a blurred image. The greater the distance between the theoretical point of projection and the actual position of the retina, the greater the area of distribution becomes and vision gets more blurry.

This distribution of light rays over an increasing area has inspired a relatively simple approach to image blurring with Unity shaders [45, 46]. In this method the image to be blurred (in our case a 2D Sprite) is duplicated multiple times. One instance remains at the original position. The others receive a small positional offset, each in a different direction. These sub images are additively stacked back into one, before dividing the RGB and Alpha Channels of the result by the total number of instances. The last step makes sure, that no additional brightness is introduced by adding the sub images. Instead the color and alpha value of the original image are distributed over a larger area, resulting in a blurred appearance. In theory, the blurriness increases for greater offset values. This simple offset approach was the starting point for implementing the blur shader of the eye simulation. While this approach gave solid results for light blurs, it introduced significant visual artifacts for stronger blurs with higher offsets. These artifacts can be seen in Figure 4.13.

The quality of these results were in no way satisfying for the use in the VR application. We therefore came up with an extension of the offset based method. Our approach allows us to achieve renderings with high levels of blur, without introducing obvious visual artifacts. In the new method, the blur shader doesn't only use a simple sprite. Instead it works with a 2D texture array, containing the initial sprite and a set of preprocessed versions of that same sprite, with increasing levels of blur. The pre-blurring of the sprite will not be performed at runtime, but instead the texture array is created outside of Unity beforehand and fed to the shader as a static resource. This relieves computational work from the shader code itself, while also opening up greater possibilities for image editing, using third party software.

As an example, the texture array used for the final version of the eye simulation was created via the image editing program GIMP [47]. Starting with the original sharp image, the following two steps were iteratively repeated to get seven more images of increasing

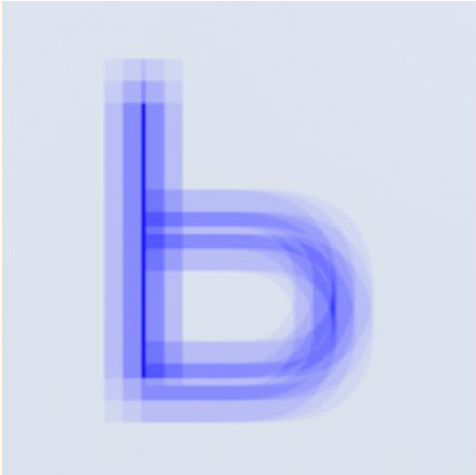


Figure 4.13.: Artifacts for stronger blurs with the simple offset approach

blurriness. First a noise filter was applied to the current image, randomly shifting some pixels around within their local neighbourhood. This step was followed up with a Gaussian blur, to introduce additional blurring, while also making the result more smooth. The resulting images were then stitched back together into a single PNG file, that can be used as a texture array resource within Unity. Figure 4.14 shows the final texture array with 8 levels of blurriness.

Figure 4.14.: The texture array with pre-blurred images

In the first step of the new shader code, this array is used to sample a sprite version with the appropriate amount of blur, for a given input blur strength value $b \in [0, 1]$. The goal here is to get a smooth transition from the first entry of the texture array, all the way to the last, for increasing values b. To achieve this, the process divides the blur strength scale into $n - 1$ disjointed intervals, with n being the number of entries in the texture array. Based on the interval a given value b falls into, the two closest sprites are selected from the array. Then a linear interpolation is performed between them, providing a smooth intermediate image.

This sampled image is then used as an input for the offset based method, meaning copies of it are created and then offset in differing directions. Here, the offset margin is also derived from b, to get greater offsets (meaning more additional blur), for greater blur strength values. In this combined approach, the already blurry image samples will forgive the use of much greater offset margins, without introducing the same undesirable visual artifacts as before. In Figure 4.15 you can see, that the improved method results in renderings of much greater visual quality, even when heavy blur is applied.

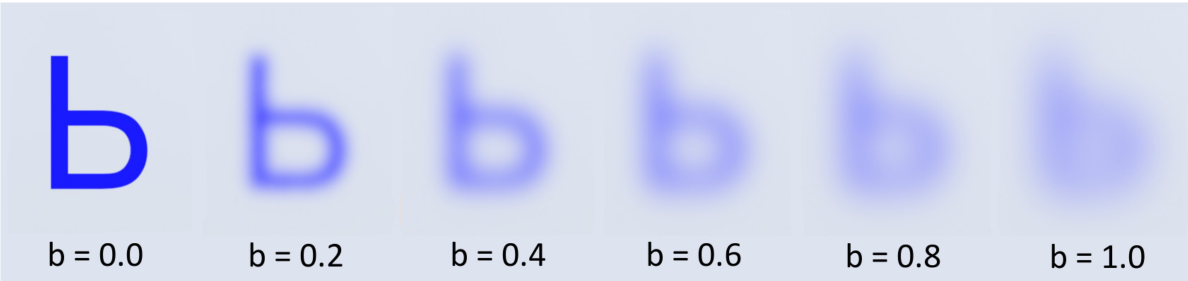


Figure 4.15.: Rendered retinal images for various blur strength values b

It has to be noted, that in principle you could only use the first stage of the new method to achieve similar visual results, using even less computational resources at runtime. This assumes that the texture array is populated with visually appealing versions of the sprite for each given stage of blurring, with the last one having the highest overall level of blurriness desired in the application. In this case, sampling from that array with linear interpolation might prove to be enough to get a smooth progression from unblurred to a highly blurred sprite, while maintaining good visual quality all along the way. We have also created a separate shader file for this sampling-only approach. However, given our limited experience with manual image processing, the combined method outlined above provided the best overall visual results for this particular project.

4.6.2. Brightness Based Rendering Effects

As part of the eye model's pupil simulation, we want the user to see the effects of lighting condition and pupil dilation state on the final retinal image. In subsection 4.3.2, we introduced the underlying brightness parameter $B \in [0, 1]$ and how it's value is determined for the viewed object and for the two images during simulation. Now, we want to take a deeper dive into the accompanying visuals and provide a rundown of the shader code and the visual effects that were employed to achieve them.

The basis for most of the upcoming techniques, forms a rendering concept called High Dynamic Range (HDR). HDR extends the available range between the darkest and the brightest values that can be assigned to a surface, beyond what is possible with the conventional 8 bit color channels [48]. The increased number of distinct color values allows for the display of very bright and very dark regions on the same screen, without losing details in any of them.

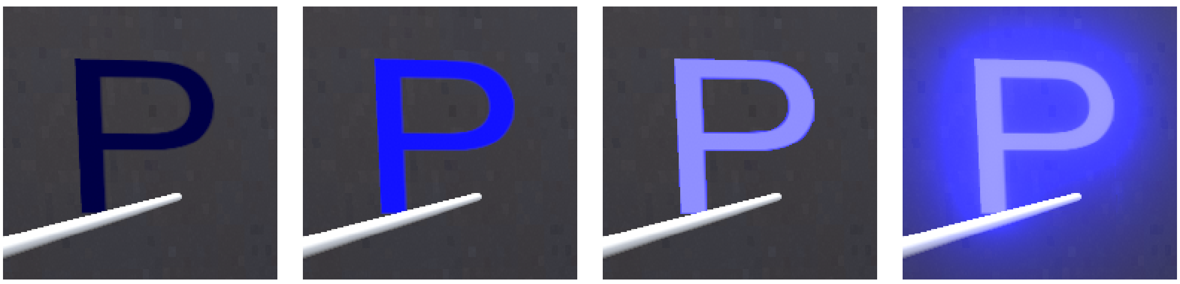
With HDR Rendering set up, we move on to the actual shaders for our sprite elements. Here, we employ so called emission shaders. In rendering, "emission" signifies that a flat, lighting and view independent color value is added on top of the regular color of a pixel. The result are self-luminous objects, that emit a fixed color, independent of outside lighting. For such an emission shader, we need two parameters: an emission color $Color_{emission}$ that stores the basic RGB values added to the rendering and an emission intensity $I_{emission}$ that is used to optionally downscale or amplify these values. The shader code then calculates the final pixel color as:

$$Color_{final} = Color_{base} + Color_{emission} \cdot 2^{I_{emission}}$$

An intensity > 0 increases the values of the individual RGB channels of the emission color, leading to a greater amount of additional luminance introduced into the rendered image. At the same time, an intensity < 0 dampens the impact of the emission on the final color. It is worth repeating that with HDR Rendering, the values for the red, green and blue color channels can exceed the traditional limit of 255 for high emission intensities. We can now incorporate our input brightness value B, by converting it into an emission intensity, thus controlling the amount of added luminance. In prior testing, the most appealing visual results were achieved for the intensity range $[-5.5, 5.5]$. Therefore we calculate the emission intensity as:

$$I_{emission} = -5.5 + 11 \cdot B$$

For the base color we choose a darker shade, so that most of the final output is determined by the emission itself. This gives us a nice progression moving from a dark coloring for the respective element (Figure 4.16a), to the fully saturated color at medium brightness (4.16b), leading up to an overly bright, whitish appearance caused by a state of overexposure (4.16c).



(a) $B = 0.0$ (b) $B = 0.5$ (c) $B = 1.0$ (w.o. Bloom) (d) $B = 1.0$ (with Bloom)

Figure 4.16.: The rendered viewed object for a number of brightness levels

These results, however, do not yet convey an actual "sense of brightness" to the observer. Without any additional effects, an object with full brightness value B appears rather pale, but not actually all that bright. To remedy this, we made use of Unity's post-processing capabilities [49], to add a widely utilized visual effect called "Bloom". The Bloom effect was introduced into computer graphics by Spencer et al. in 1995 [50], as a method to simulate glare in the human visual system. Here, the luminance from very bright light sources appears to "spills over" into the surrounding regions of the seen image. The underlying real world visual phenomenon is caused by a scattering of light in the cornea, lens and retina of the eye. Through this scattering, some of the photons from a bright light source are redirected onto neighboring parts of the retina and "overlayed" onto the image in these regions, introducing additional luminance that does not originate from the object in said region [50]. An example for the appearance of the viewed object with the added Bloom effect can be seen in Figure 4.16d. It is here, that HDR becomes especially relevant. To set up the effect, one needs to define a threshold luminance value. For surfaces below that nothing happens during post-processing, while for pixels above that the Bloom is applied. In regular rendering the highest possible value is 1.0, which is associated with the color white (i.e. all three color channels have full value 255). Therefore it would be impossible to find a threshold, that adds bloom to self-luminous elements, without applying it to every white object in the scene as well. With HDR however we can set this threshold to a value > 1, thus only applying Bloom to surfaces where luminance has been extended beyond the usual range.

Let's now look into the behavior for the lower emission intensity ranges. With the process described up to this point, the rendered elements will grow darker and darker, until they appear nearly black, once we turn down their brightness levels. During all this time they remain fully opaque. For a physical object in the environment this makes absolute sense (i.e. for the viewed object). However, the same behavior doesn't exactly work for the projected image on the retina. Here, a lower brightness value signifies that less light has fallen through the pupil into the eye. With less detected light, the projection grows weaker. In this regard, the white retinal screen might be a bit misleading. A clear white screen in the simulation model would actually correspond to the owner of the eye seeing pitch black. For all these reasons, it's not enough to darken the color of the retinal image at low brightness values. We also need to lower the opacity to accurately represent a faint image in the absence of light. To do so, the retinal image script converts the current brightness value into an alpha value, that is used by the rendering pipeline to represent opacity. For a brightness $B \geq 0.45$, alpha remains at its maximum of 1. For values below that the alpha value slowly scales down, in a linear fashion, to a minimum value of $\alpha = 0.2$.

To increase the visibility of all these effects for the three (self-luminous) "P" sprites, we added the possibility to dim all other lighting in the virtual environment via a light switch on the working table (Figure 4.1, annot. 5). This, at last, leads us to the final visuals for the retinal image, which are presented in Figure 4.17.

4.6.3. Light Cone

The visualization techniques in the previous section are great for showing the impact of available light on the visible image. Yet, even with carefully finetuned visuals, it can at times

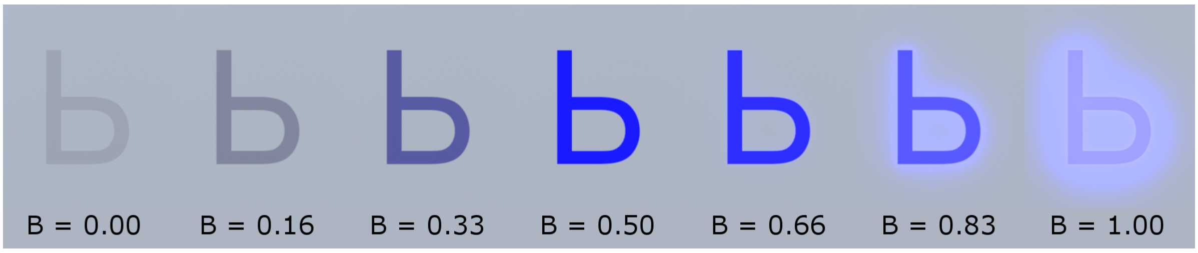


Figure 4.17.: Final retinal images for escalating brightness values B

be hard for the user's own eyes to make out some of the more subtle changes in image brightness. One reason for this, is that human brightness perception is highly subjective and not equally sensitive in all ranges [51]. For this reason, we wanted to support the existing visualization with an additional, clearly discernible indication of the amount of light, that manages to fall through the pupil into the eye.

A great way to do this is via a stylized light cone. For diffusive (or self-luminous) surfaces, the light from a single surface point spreads out in all directions virtually equally. When looking at the light that actually passes through the pupil, we are only interested in the subset of light rays that fall within a given solid angle, bound by the outer rim of the pupil. These rays form a primary light cone, that originates from the point of interest on the object and then falls through the pupil onto the eye's lens. A secondary light cone is then used to visualize the individual rays being refracted at the lens and converging back together into a single point in the sharp theoretical image. Here, a wider pupil area lets through more light, which is in turn signified by an increased volume of the light cones. This leads to a visualization, where brightness changes caused by pupil dilation are more easily noticeable via the size of the cones.

In the application, we work with two cone meshes that have been cut in half vertically, just like the eye model itself. The cones are semi-transparent and emit a flat yellow color, that's independent of lighting conditions or view angle. Since this form of material turns out to not be visible through the boundaries of the normally transparent eye lens, we added yellow line segments that frame the half cones at their top and bottom edges. That allows the user to also follow the course of the light cones at places where they disappear inside the lens. The resulting view can be seen in Figure 4.18.

To achieve the correct shape and placement of the two light cones, two steps need to be performed for the cone meshes. First of all, the position of the two cones' tips need to be set to the original point on the object and the projected point on the image, respectively. This step has an easy solution within the Unity environment. Here, a triangle mesh can be imported with "read/write mode" enabled. This allows developers to modify the meshes vertex and triangle data at runtime [52]. With this, the vertex corresponding to a cone tip, can simply be reassigned to the required position. This deforms the mesh in the desired way, while the vertices that comprise the cone's base remain unaltered.

Secondly, the shared (semi-)circular base surface of the cones needs to be positioned and scaled in a way, so that the edges of the primary light cone run exactly along the pupil's boundaries. In other words, the incoming light cone has to exactly fill out the pupil area and then continue at the same angle until it reaches the eye lens. Following this conception, the needed parameters can be determined by projecting reference points on the pupil, forward onto the plane spanned by the center of the lens. Figure 4.19 shows a sketch of this projection. First, we construct a line that originates from the starting point at the top of the image and extends through the center of the pupil. Now, a line-plane intersection is performed with the lens plane, in order to determine the correct center point for the base of the cone. By repeating this process for a line through the topmost point of the pupil, we get a point on the edge of the circular base. Measuring the distance from this edge point to the previously calculated center point then yields the cone's radius and with that the required scaling factor.

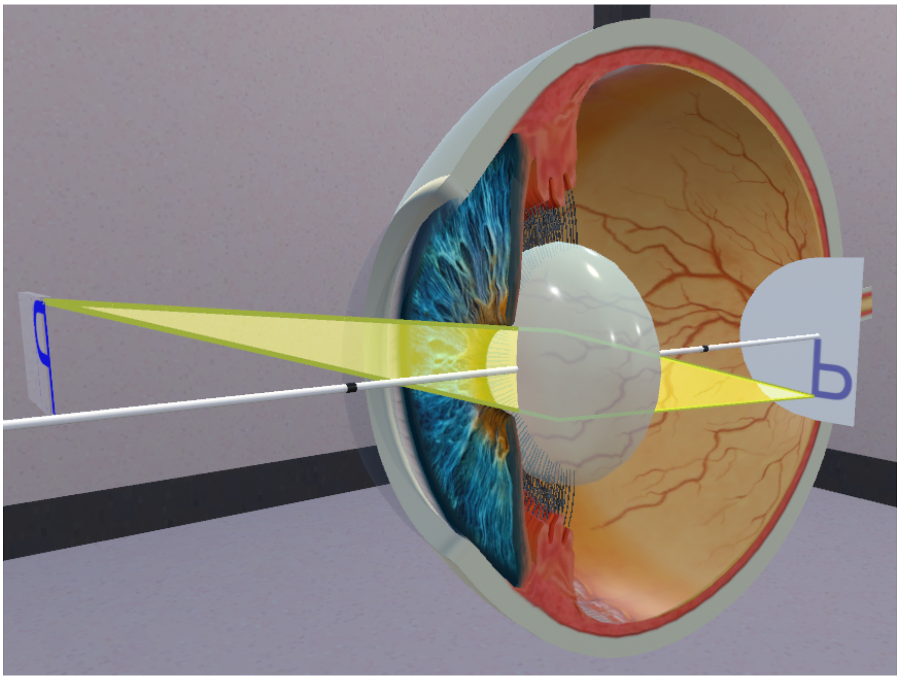


Figure 4.18.: Enabled light cone visualization during Exercise 3

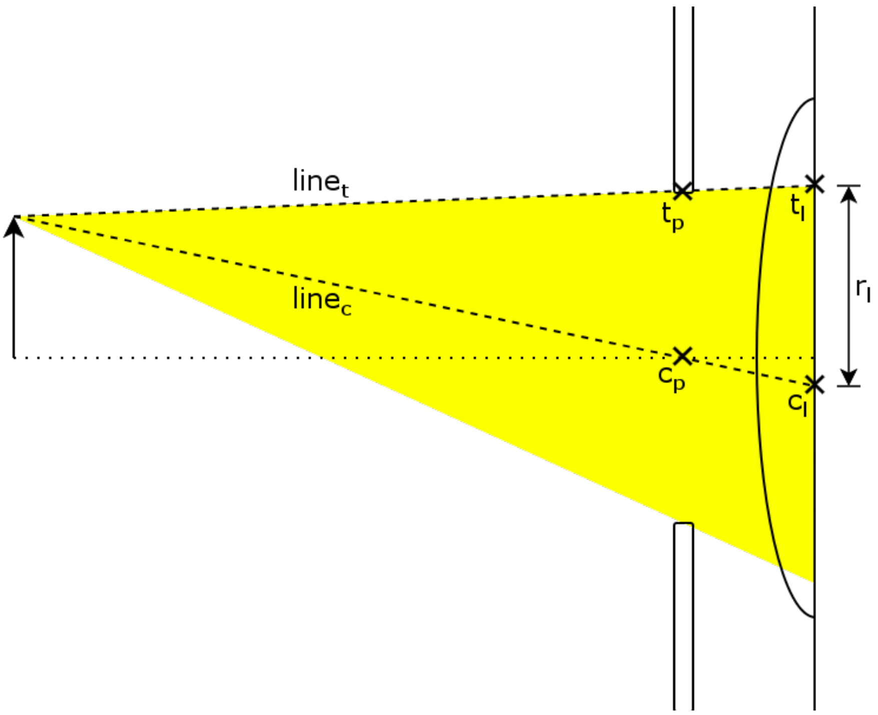


Figure 4.19.: Geometrical sketch for the determination of the light cone radius and base center. Points t_l and c_l are calculated via intersection of $line_t$ and $line_c$ with the lens plane

5. Evaluation

The evaluation of our application will be split into two stages. A preliminary first study, that is included here, deals with the topics of usability and general acceptance for the use of VR in university teaching. The second stage, will be subject to future work. It will see the application being utilized during true use in the field, to evaluate learning benefits associated with the immersive application and the novel ways of visualizing and interacting with learning content, it comes with.

5.1. Usability Study

To prepare for this second stage, we will need to rule out the existence of any glaring issues with the software. With the application being used in actual teaching of a much frequented module, the system needs to be properly polished, if we intend to ask of the students to use it as an official resource. Furthermore, it would be immensely detrimental to the success of the study, if poorly usable interactions or an application breaking bug, were not discovered until the actual study conduct, where hundreds of participants will be exposed to the application. Due to all these factors, special care has to be taken, that anything in need of fixing or fine-tuning is addressed beforehand and that the application is in the best shape it can reasonably be brought to. In order to identify any such points of friction, we conducted the aforementioned usability study with a small number of preliminary testers. In the following we will present the setup and the results of this study.

5.1.1. Participants

Subject to the study were 10 university students, ages 19-26, from a variety of academic backgrounds. Half of the participants are studying for a teaching degree in physics, while other subject areas include computer science, history and sociology. The gender distribution of the study was 8 male and 2 female. 3 of the 10 participants have indicated, that this study was the first time, they ever came in contact with VR Technology of any kind. All 10 have stated German as their first language, which eliminates the factor of any language barrier potentially hindering the participants' understanding of the exercise content or the study questions.

5.1.2. Questionnaire

As a follow up to their trial run through the application, the participants were presented with a fully anonymous four page questionnaire. The empty template of the questionnaire in German language can be found in Appendix C. The first page included the general information and demographics discussed in the previous section. For the judgement of the application itself, we included two popular assessment scales. The first and main one is dedicated to the level of usability of a system and whether it's users subjectively feel able to reach their desired goals using what the system provides them. To do so, the System Usability Scale (SUS) [53] contains ten statements, for which the user can indicate their level of agreement, ranging from "strongly disagree" to "strongly agree" in 5 increments. From these ten items a score $\in [0, 100]$ is formed that acts an an indicator for the usability of the system, with 0 being the worst and 100 being the best possible value.

In a secondary assessment, we want to investigate how "present" the user feels within our application and see if the level of experienced immersion influences how a user rates it's usability. Therefore, we included a german translation of the Igroup Presence Questionnaire (IPQ) [54], [55] for the participants to fill out. An evaluation of the 14 items of the IPQ provides scores from 0-6 for three individual sub-factors of presence. The sub-scale Spatial Presence (SP), describes if a user feels like they are actually spatially located in the virtual room. Involvement (INV) denotes how much attention the user directs towards the virtual world as opposed to their real physical surroundings. And lastly, Experienced Realism (REAL) gives indication how real the virtual world feels to the user of the system. An overall score for the user's experience of presence (PRES) can be obtained by calculating the mean of these three sub-scores [56], [57].

In addition to these two quantitative scales, we also included three qualitative open questions about the participants' experiences during the test and their general acceptance of VR for university teaching. The respective questions from Appendix C can be translated as:

- "What did you like most about the application?"
- "Where do you still see potential for improvement in the application?"
- "Could you imagine using applications of this kind in university lab courses? If yes, what would be the most important factor for you, that has to be accounted for in such an application?"

To gather some more contextual information on the participants and to investigate potential influencing factors on their assessment of the application's usability, we included two additional rating scales with 5 items each. First up, we wanted to get an impression of the user's overall experience in dealing with modern digital technology. To this end, we have adopted the scale for "Students' Perceived ICT (Information- and Communication-Technology) Competence" from the scales manual of the 2015 PISA program [58, p. 124] and integrated it into our questionnaire.

Additionally we are interested, if there is a difference in usability rating from participants that are more involved in the medical domain, compared to ones that don't usually engage with medical topics at all. Therefore we took the scale "Enjoyment of Science" [58, p. 90] from the same manual, and adapted the items to instead assess the user's level of enjoyment in engaging with medical and physiological topics. For all five items we kept the general wording, but swapped out the operand "science" with words related to physiology and medicine. As an example the item "I like to read about science", became "I like to read about how the body works".

5.1.3. Study Conduct

The bulk of the study was conducted at the LMU chair of physics education in Freimann. Here, the participants were assigned a testing station with one VR system each. They then received a basic instruction on how to put on and adjust the HMD and how to use the controllers, accompanied by some safety information regarding the wearing of a headset that blocks vision of their true surroundings. Afterwards they were given a short time to familiarize themselves with the technology within the Microsoft Mixed Reality Hub World [59]. They were also encouraged to try out the teleport functionality to move around the virtual environment while remaining seated in their chair.

As they entered the LIN-VR application, they were reminded of the fact that the learning content they will encounter was designed for medical students and that they will likely not be able to answer every question posed to them. With this in mind, they were instructed to try and engage with the questions as best they could, but to skip any question that requires advanced medical or physical background knowledge above their ability. For the same reason they were also let in on the fact that they would be able to continue to the next exercise, regardless of the point score, after running through all the panels a second time.

At this point the participants began engaging with the application for roughly 20-25 minutes, all in parallel at their individual testing station. During the engagement with the application, no further instructions or explanations were given to the participants other than what is provided by the application itself. Only if a user would have been completely stuck at some point in an exercise, unable to continue at all without assistance, they would have been given a small hint to ensure they could continue their test. This scenario however did not occur throughout the entire study, as every participant was able to finish all three exercises without additional outside help.

To bolster up the sample size a little more, a couple of single person sessions were conducted afterwards with additional study subjects. Other than the number of participants working in parallel, these sessions were following the same procedure outlined above.

5.1.4. Results

In Table 5.1 we compiled the overall results for the SUS scale and the IPQ sub-scales, based on the averaged scores of the individual questionnaires.

Table 5.1.: Averaged final scores for the SUS and IPQ scales of the user questionnaire

	N	SUS	IPQ-PRES	IPQ-SP	IPQ-INV	IPQ-REAL
overall	10	82.53	3.21	4.23	2.85	2.55

SUS

From this table we can read that our application achieved an average SUS score of 82.53. However, this number alone is not really meaningful in it's own right without something to compare it to. To address this fact, Sauro [60], [61] has developed a qualitative benchmarking scale for the SUS score, that assigns the tested system a grade from "F" (lowest) to "A+" (highest). According to this rating scale, which is shown in Figure 5.1, the LIN-VR application achieves the grade "A" with the adjective "Excellent". This is an extremely positive result and it puts our system in the 90-95 percentile range of applications in terms of usability [61]. This highly suggests that we have succeeded in our goal of creating an application that is accessible and sufficiently easy to use for the employment in university teaching.

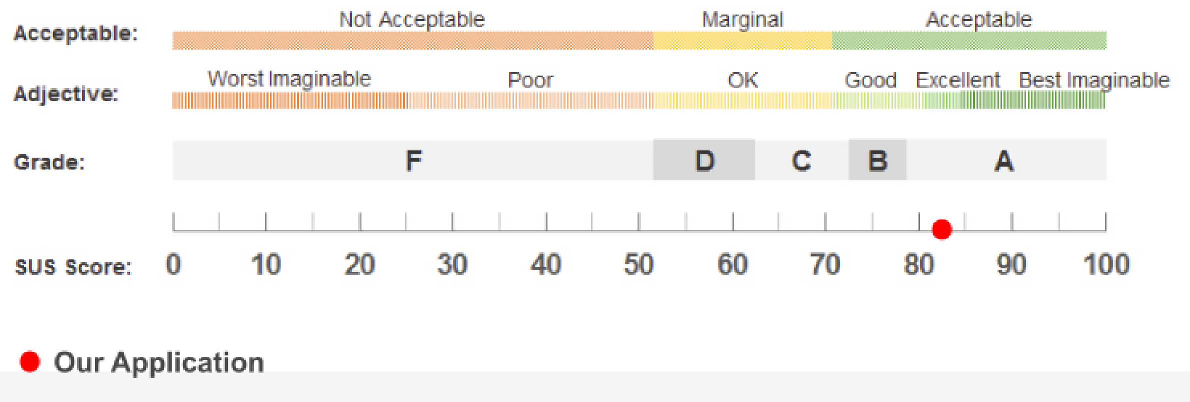


Figure 5.1.: Placement of our application on the grade based ranking scale for raw SUS scores. Base graphic taken from Sauro [61]

IPQ

Besides the SUS score of our application, Table 5.1 also contains the scores for the four IPQ factors, with $PRES = 3.21$, $SP = 4.23$, $INV = 2.85$ and $REAL = 2.55$. The comparative profile of the latter three IPQ sub-scores is visualized in Figure 5.2. Here, we see a clear dominance of the spatial presence factor compared to the other two.

However, once again, these values don't really mean anything, unless they're set into relation with some other system. Therefore, inspired by Sauro's approach, Melo et. al. [62] developed a grade ranking for the IPQ and it's sub-scales, based on the aggregated data from

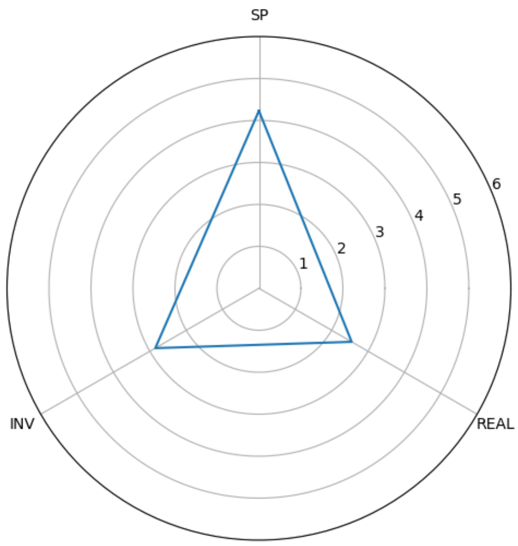


Figure 5.2.: The presence profile of the application, based on the collected IPQ scores

nearly two thousand collected samples. According to their benchmarking scale, which can be seen in Figure 5.3, our application's scores for the PRES, INV and REAL factors actually turn out to be classified as "F", thus being labeled as "Not Acceptable". The spatial presence factor SP is graded one category higher and receives an "E", while missing the next higher category "D" by just a 0.02 difference in score. This puts the spatial presence in the "Marginally acceptable" range. These results are evidently not great, so let's dive into what might be the reason for the poor performance in the three sub-domains of the IPQ.

Presence	Spatial Presence	Involvement	Experienced Realism	Grade	Adjective	Acceptability
≥ 4.41	≥ 5.25	≥ 4.87	≥ 4.50	A	Excellent	Acceptable
≥ 4.07	≥ 4.76	≥ 4.50	≥ 3.75	B	Very Good	
≥ 3.86	≥ 4.50	≥ 4.00	≥ 3.38	C	Satisfactory	
≥ 3.65	≥ 4.25	≥ 3.75	≥ 3.00	D	Marginal	Marginally acceptable
≥ 3.47	≥ 4.01	≥ 3.38	≥ 2.63	E	Unsatisfactory	
< 3.47	< 4.01	< 3.38	< 2.63	F	Unacceptable	Not Acceptable

Figure 5.3.: Classification of IPQ scores into qualitative grade based ranks. Table taken from Melo et. al. [62]

Regarding the experienced realism scale, a low grade is somewhat to be expected for the given application. The highly stylized graphics and elements like the blackboard that floats in mid air stand in direct conflict with any experience of realism. In the end, the design choices made during development put a far greater emphasis on clarity and expressiveness of the displayed visualizations over any concerns of true realism. Therefore, while perhaps not perfect, the low rating is not really a reason for concern.

For the involvement scale, on the other hand, we ultimately do strive for a higher ranking, as the application aims to be immersive enough to capture the user's full attention and not have them constantly focus on their real surroundings. Therefore, the low score clearly indicates that more work has to be put into improving the application in this regard. However, at this point in time, the low grade is not entirely surprising either. So far, not many resources have been directed toward raising the immersive quality of the LIN-VR beyond the level present in the preceding EyeLearnRoom application and there still remains a lot of untapped potential to do so. As an example, there currently is no sound in the application and barely any haptic feedback, both of which could be employed to draw the users attention away from the real world, into the virtual environment.

The biggest cause for surprise to us was the inadequate level of spatial presence. We initially presumed, that this factor was largely dependent on the used VR technology itself, rather than any active design decisions on our part. Therefore, we expected that using a state of the art VR HMD alongside a PC with a powerful graphics unit would suffice to, at the very least, achieve a medium level of spatial presence inside the 3 dimensional environment of the application. However, the given results of the IPQ evaluation indicate that there are, after all, some properties in our application that actively stand in the way of spatial presence. These could potentially be related to the active movement inside the virtual environment, or rather the lack thereof. The participants engaged with the application in a seated position and only navigated around the virtual room via a teleportation functionality. It is not hard to imagine, that the associated instantaneous change in position, might disrupt a user's sense of place within the environment. Once they were positioned correctly in front of the working table with the eye simulation, the only further movement was associated with the user's head or upper body motion. Within the setup of the study, they never freely walked around inside the room, which might have prevented them from building a proper mental map of their environment. The fact that the cubic, white room doesn't present any clear environmental reference points, other than the working table itself, surely also doesn't help in that regard.

SUS and IPQ correlation

To put the results of the SUS and the IPQ scales into relation, we created scatter plots with the corresponding data points of the individual study subjects to see if any patterns can be made out. The resulting plots for the four IPQ factors and the associated SUS score can be seen in Figure 5.4. Interestingly, only one of the IPQ factors seems to potentially have any correlation with the level of usability. The plotted data for the spatial presence (SP), shows faint hints of a linear relationship with the SUS score. Thus it appears that there might be some type of connection between a high level of spatial presence felt in the virtual environment and a positive assessment of the system's usability. Meanwhile, the other IPQ sub-factors of investment and experienced realism don't seem to have any connection with the level of usability whatsoever. Consequently, it also makes sense that there is no clean correlation between the SUS score and the overall presence score, since the latter is calculated as the mean of the other three factors and thus any correlation the spatial presence might have will be dampened by the other two factors.

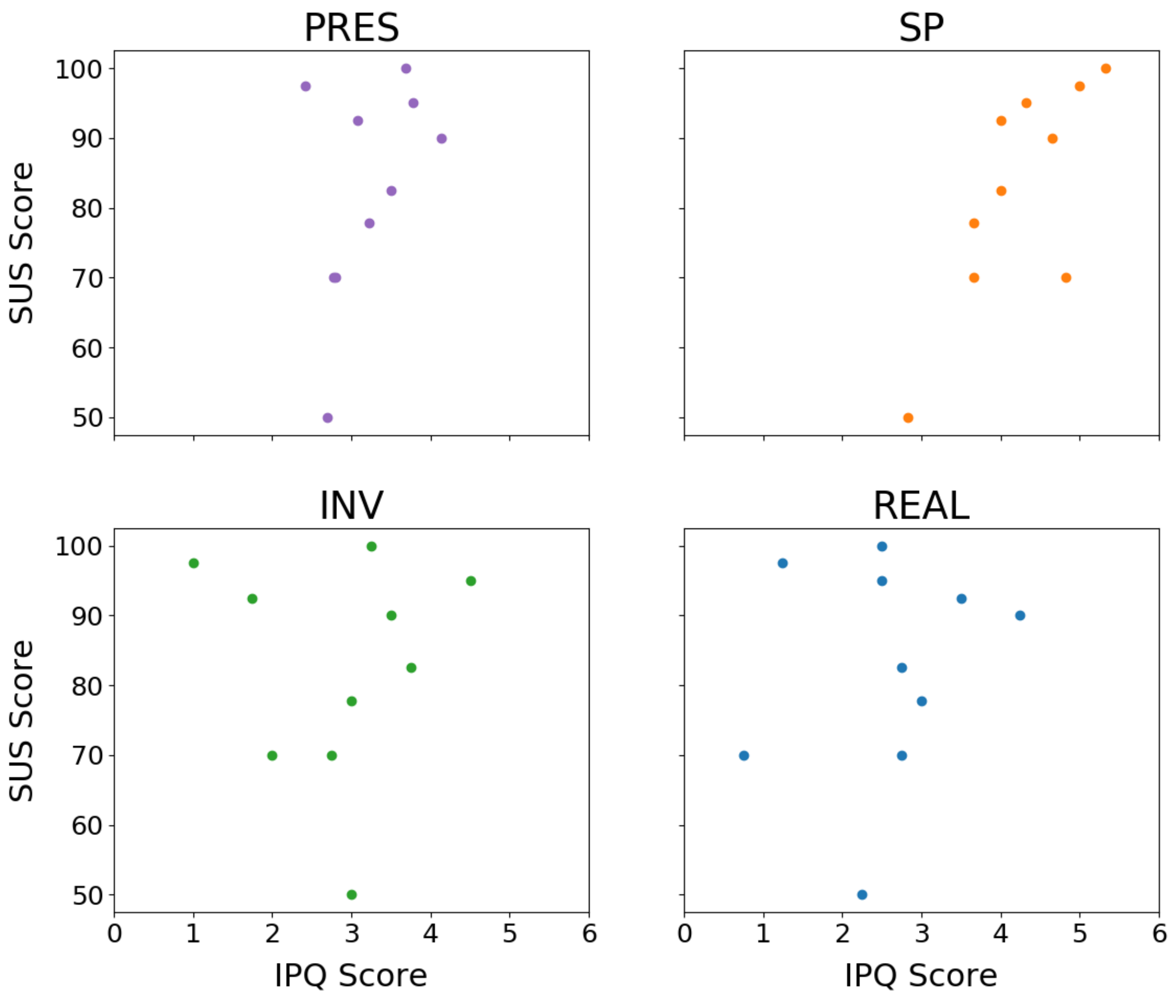


Figure 5.4.: Scatter plot for the SUS score and the individual IPQ sub-scores

First Time Use

In the next step we investigate the relationship between our primary and secondary scores and the collected contextual information of the individual participants. We start by comparing the average scores for participant that have already engaged with VR systems before, to those for participants that had no VR experience, prior to the study. The respective breakdown can be seen in Table 5.2. First time users generally rated the system's usability a little higher than non-first time users. Perhaps this difference can be attributed to the experienced users having a more solid frame of reference and something the compare the system to, in terms of user friendliness. Or perhaps the excitement and the novelty factor played a part in a general positive experience with the system that impacted the first time users' "generosity" when assessing the system. At the same time, the first time users directed significantly more of their attention to their real surroundings, as indicated by the low investment score (IPQ-INV) of that group. The other IPQ sub-factors, on the other hand, show no real variation between the two groups. Thus the differences in overall presence score (IPQ-PRES) can be reduced to the discrepancy in involvement mentioned prior.

Technological Competence

Next up, in Table 5.3 we have broken down the SUS and IPQ scores by the participant's self reported level of competence in dealing with digital technology. Here, we divided the participants into three categories based on the calculated score for the "Students' Perceived ICT Competence" scale. A score of ≥ 3 puts them in the category "high", a score s with $3 > s \geq 2$ results in the category "medium" and a participant with a score < 2 is assigned to the "low" category. In this particular study, no participant reported a "low" level of technological competence, as is somewhat expected since we are dealing exclusively with a very young age group. The other participants are distributed among the "high" and "medium" categories in a fashion that reminds of the "First Time vs. Not First Time" table, with a 7 to 3 split between the "high" and the "medium" level. The values in the four IPQ columns of Table 5.3 do not communicate any significant connection between subjective presence and self reported technological competence. The values for involvement and experienced realism are slightly higher for the group with "high" competence, but these fluctuations could very well be attributed to the small sample size and they are certainly not as pronounced as the previously discussed gap in involvement for first time VR users. Meanwhile, the assembled SUS scores show some signs of an underlying correlation. Subjects who report a high level of technological competence rate the usability slightly lower on average. This is again reminiscent of the situation between first-time VR users and more experienced users. That might suggest, that a higher overall skill level when dealing with technology has a similar effect on the assessment of the application's usability, as the more specialized experience with VR technology in particular.

Interest in Medical Topics

Last, we take a look if there is a connection between the user's ratings and their interest level in medical topics. Here, we again applied the same categorisation into "high", "medium" and "low" based on the score of the scale for "Enjoyment of Medicine and Physiology". The broken down results in Table 5.4 indicate an inverse correlation between the SUS score and the participants level of medical interest. A deeper dive into the responses to the open questions, to search for a potential underlying reason of this connection, came back with no results. The positive and negative feedback the participants shared with us did not display any particular differences based on the sub-category of medical interest they fall into. The IPQ scores for overall presence, involvement and experienced realism do not show a clear trend between the high. medium and low categories. The spatial presence score, however, has the same inverse correlation with medical interest level that the SUS score does, further hinting at a connection between the SUS and IPQ-SP domains. But again the open questions didn't provide us with a satisfying conclusion as to what causes this inverse correlation.

Table 5.2.: Average scores for users with and without prior VR experience

First VR Usage	N	SUS	IPQ-PRES	IPQ-SP	IPQ-INV	IPQ-REAL
no	7	80.75	3.40	4.23	3.39	2.57
yes	3	86.67	2.77	4.22	1.58	2.50

Table 5.3.: Scores broken down based on the participants' self reported level of technology competence

Technology Competence	N	SUS	IPQ-PRES	IPQ-SP	IPQ-INV	IPQ-REAL
high (≥ 3)	7	80.40	3.30	4.19	3.00	2.71
medium (≥ 2)	3	87.50	3.00	4.33	2.50	2.17
low (< 2)	0	-	-	-	-	-

Table 5.4.: Score breakdown for the participants' level of interest in medical topics

Medical Interest	N	SUS	IPQ-PRES	IPQ-SP	IPQ-INV	IPQ-REAL
high (≥ 3)	4	75.70	3.12	3.62	2.88	2.88
medium (≥ 2)	3	85.00	3.57	4.22	3.33	3.17
low (< 2)	3	89.17	2.96	5.05	2.33	1.5

5.1.5. Discussion

The data collected during the study indicates a highly satisfactory level of usability of the LIN-VR application. While users who are more involved with either the medical field or with digital technology, rated the application slightly more critical, the measured usability scores for these groups still remains at a high level, well within our expectations of the system.

On the other hand, the IPQ scores for the application reveal a lot of potential for improvement with regards to the subjective feeling of presence in the virtual environment, especially for the dimensions of involvement and experienced realism.

Here, the experienced realism can likely be left aside, to some degree. The application has no real aspiration to be hyper realistic. On the contrary, much of the visual elements of the application are deliberately stylized and involve some form of simplification to focus on the most important aspects of the taught content. Besides that, the general situation of moving things around in a giant human eye that has been cut in half, likely does not fully lend itself to the promotion of realism.

However, in terms of involvement, the application surely could be improved to be more immersive and capture more of the user's attention to draw it away from their physical surroundings. Although, the available data gives no clear indication that raising this score would improve the overall usability of the system, it is surely still a worthwhile attempt in view of the coming learning benefit study and the field usage beyond. Here, a workable plan should aim to involve more of the user's senses within the virtual environment. This includes adding sound to the application in the form of verbal instructions, auditory feedback for success or fail scenarios or general sound effects associated with the interactions in the environment. The application also should make greater use of the VR controllers' haptic feedback functionality, which currently remains largely unused. The inclusion of these features should lead to a more immersive experience and a greater sense of presence in the virtual world.

Concerning the spatial presence, there might be some potential for improvement associated with the design of the room and the navigation within it. If the physical surroundings allow for it, one available option is to outfit the virtual room with more engaging elements, like informational posters on the walls or other interactive objects, and have the user walk around between them. The natural (continuous) movement within the environment, alongside a larger set of spatial reference points, might help to improve the user's sense of spatial presence. However, here we would need to find the right balance of providing enough points of interest in the environment, without them becoming too distracting from the actual educational content and the exercises the users should be performing.

Open Questions

Based on the user's answers of the three open questions, we were able to gather some more valuable insights and received a number of interesting suggestions.

First up let's look at the answers to the question "What did you like most about the application?". Here, the most commonly commented positive aspect of the application were the provided visualizations, which were praised for being clear, descriptive and not overly abstract. A second aspect that was positively commented on by various participants, was the somewhat explorative hands on approach of the application. Users reportedly enjoyed messing around with the eye simulation to try out all the different possibilities.

Even more valuable in terms of necessary future adaptations of the system came up for the question "Where do you still see potential for improvement in the application?". A number of users reported having had difficulties with two particular interactions, which is consistent with what we observed during the participants' engagement with the application. The first one is the grab based movement of the viewed object on the optical axis. Oftentimes a user would accidentally grab the optical axis instead and rotate the entire eye model when just trying to move the object. Sometimes they needed several tries to actually get a hold of the object, before they were able to move it successfully. This issue should be easily solved through two changes. First of all, we can simply increase the size of the object's collider, so that the hand positioning doesn't have to be as precise. This should be supplemented by a clearer indication of what element will be affected while hovering inside it, before the grab button has been pressed. The current implementation gives the respective element a faint yellow outline to fulfill this role, but the clarity of this connection could be improved upon, as evidenced by the users' struggles.

The second, even greater, area of friction lay with the buttons of the control panel. A large number of users couldn't get a grasp of how the activation of the UI value sliders worked and would struggle to open it or accidentally close it mid interaction. Here the underlying interaction, required from the user, turned out to be way less clear than we assumed. This can largely be attributed to the fact that the activation of the associated UI slider canvas is programmed to be performed "OnButtonUp" of a control panel button. This means that initially nothing will happen when pressing the button down and only when the button is released by moving the hand back up, the canvas opens up. This interaction rule, which is in hindsight not very intuitive at all, led to all sorts of awkward or outright unsuccessful interaction attempts. Users would press down the button, see that nothing happens, and wonder what they did wrong. Most of them would then not pull their hand straight back up, but instead pull it away laterally from the button. Based on this the "OnButtonUp" event would sometimes trigger and sometimes not. This random behavior clearly didn't help the understanding of the interaction and most users never really figured out the true connection, leaving them to somewhat rely on chance whether the slider would appear for their button press attempts. However, again there is a clear path to fixing the interaction. Simply changing the button behavior to "OnButtonDown", meaning that the associated behavior is triggered for the initial press, should help to align the system's behavior with the user's expectation of what happens when pushing a button and alleviate a lot of the pains associated with a mismatch in that regard. This should also take away any randomness factor that appeared for the button release in a lateral direction, that sometimes registered as a "OnButtonUp" event and sometimes not.

For the last question, we first of all observed an overwhelming level of openness to the use of VR technology as part of the users' studies. Every single participant reported that they could imagine using a VR application in a university lab course. Regarding the second part of the question "[...] what would be the most important factor for you, that has to be accounted for in such an application?", we were able to gather some good insight on the participants' expectations of educational VR applications. The largest cluster of answers is centered around the concept of assistance and technical guidance. A number of participants expressed the wish to receive a clear introduction to the system and it's basic functionalities, before engaging with the content. Other participants reported that they value having technical help close at hand in case they need assistance with the VR hardware or the application itself. One participant further specified, that they would like to have the option to call someone for help, yet not be observed the entire time during their engagement and that their video feed should not be visible from the outside unless they specifically ask for assistance with the application. Some of the other provided answers in this category include the avoidance of motion sickness inducing content, scheduled breaks to take off the HMD for a bit or the desire to have a high level of intrinsic motivation within such an application suggesting to introduce some form of gamification.

5.2. Learning Benefit Study

The second evaluation stage will take place in the upcoming winter semester 2023/2024. The associated study will investigate the impact on learning outcome of the LIN-VR application, during real teaching practice. All students of human medicine and dental medicine that attend the physics lab course during this semester, will be taught the optical physics module through the use of the VR application. This would set the estimated number of study participants at roughly 840 people.

The goal of the study is to investigate whether dealing with a more naturalistic representation (anatomically correct eye model) in an immersive environment, provides a better access to the taught content for the medical students. To do so, the new application is contrasted with the classic optical rail experiment. Because the covered topics have been slightly overhauled for the occasion of the VR app's development, the instructions for working on the optical rail have to be adapted compared to previous semesters, to match the instructions in the application. Furthermore, all the additional multiple choice questions, that are presented during an exercise, are also the same for both the real and the virtual setup.

We therefore have two options available for every exercise, performing it in the application, or performing it using the optical rail. Based on this, permutations are formed to let the participants run-through differing sequences of analogous or digital exercise work. These permutations could be set up on the basis of individual exercises or based on a coarser resolution, by grouping some exercises together (e.g. first half vs. second half). As an example for the fine-granular approach, Participant A runs through the six exercises in the order "VR-Rail-Rail-VR-VR-Rail", while Participant B engages with the exercise following "VR-VR-VR-Rail-Rail-Rail". The aim behind this approach is to figure out which exercises are a better fit to be transferred into the virtual world and to also see if there is an optimal order and distribution between working with the real and the simulated experiment.

To evaluate these parameters, the participants are subjected to knowledge retrieval tests after engagement with the exercises. Combined with another quiz concerning prior knowledge before the start of the lab course session, the score on these tests is used to compare what information actually stuck with the students for the various permutations.

Since here we are dealing with a true teaching situation as part of the students actual curriculum, special care has to be taken to provide fair treatment to the individual participants, even during such a study. This includes giving everyone access to the new VR learning resource in some (if possible roughly equal) capacity. Therefore, keeping to a 50:50 split between VR application and classical experiment for every student would be the preferred variant. This would require the exclusion of some of the more one sided permutations. In case the experimental setup were to include groups working purely analogous, the respective participants would need to be provided with the option to try the application themselves after the study session concluded.

6. Future Work

The first step going forward should be to implement the takeaways of the usability study and the test users' feedback into the application. This includes cleaning up the two interactions that some users struggled with and adding sound, as well as haptic feedback, to increase the level of immersion and the user's subjective feeling of presence in the environment.

After that, the three missing exercises on basic anatomy, corrective lenses and the cataract need to be implemented as well. However, with all the basic systems in place, this shouldn't take an excessive amount of work, especially since said systems were specifically designed with the exercises' planned functionality already in mind. Following this, the application should be ready to be used in the upcoming learning benefit study.

Going beyond the scope of the study, it could be interesting to explore the concept of cooperative learning for the lab course application. As a reference, the old analogous experimental setup of the optics lab course had the student working in pairs. The LIN-VR application technically already provides the ability for multiple people to be in the same virtual room, inherited from the EyeLearnRoom it was based upon. However, the respective feature has not seen further pursuit, up to this point, as the learning benefit study is planned to be conducted for a single person working style. When extending the application in that direction, there should be a special emphasis to create a true cooperative experience with all involved students actively working together, instead of one student conducting the experiment, while the others are limited to an observer role.

Another interesting cooperative approach would be some form of tandem work with asymmetric roles. As an example, the current VR setup doesn't really lend itself to working on mathematical problems, yet they were a large part of the old LIN lab course. In the analogous setting, you can easily ask the students to measure some values, write them down on a piece of paper and perform a more or less complex series of calculations for them. This process becomes non-trivial when moving into the VR environment without the (easy) possibility of handwritten notes. A proposed tandem working scenario could therefore have one person use a HMD to enter the application, while providing the other with a tablet computer whose screen is shared into the virtual environment. This way, the student in the VR environment could take measurements of the simulation state, while the second writes them down on the tablet and performs handwritten calculations that the VR user can still see.

Going further, it would be imaginable to perform another subsequent field study to compare how such cooperative approaches affect the learning experience and it's success, compared to the single person working setup.

Another potential avenue for future exploration we would like to propose, is the concept of a virtual assistant. With this, we refer to an entity within the application that acts as a guide to the user and introduces them to the virtual environment and the interactions therein. Using such an approach, we might be able to address the desire expressed by some users, for a strong level of initial guidance and the availability of outside help when required, in a way that doesn't break the immersion of the system. Through the use of the eye-tracking capabilities offered by some VR HMDs, alongside advanced predictive neural networks, the background system of such an assistant could potentially detect when the user is struggling with a task or when they are looking in the wrong direction for an item, and provide situational hints based on that context. With the massive recent advances in the respective area, it would also be imaginable to outfit the assistant with some form of conversational AI, enabling the user to ask for highly specific guidance or allowing them to request further explanations for certain topics.

7. Conclusion

As part of this thesis, a VR learning application has been created that can be used in teaching optical physics to students of human medicine, as a replacement for the classic optical rail experiment. The applications enables students to engage hands on with the basic concepts of geometrical optics, utilizing an interactive eye model with simulated physical and physiological behavior. As a side product to the main implementation we have created a highly versatile, re-usable architecture for the background management and synchronization of learning content during guided exercise work in interactive applications.

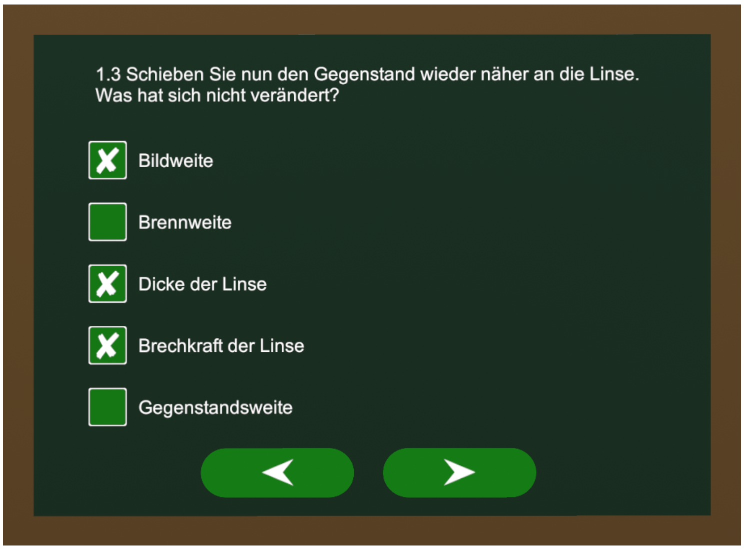
We initially set out to create a VR learning environment that provides students with clear and descriptive visualizations of the optical principles involved in human vision. At the same time, the application should exhibit a strong degree of usability, to ensure that students will not be held up by poorly usable interactions in the upcoming learning study or the lab course use beyond.

The conducted usability study indicated that we were successful in both these regards. The evaluated data shows that our application is highly accessible and easy to use, even for first time VR users. Additionally, the handed out questionnaire revealed a large level of satisfaction with the visualization techniques used to illustrate the various taught concepts and with the hands on nature of the engagement. More generally, the responses provided by the participants also hinted at great deal of openness to the concept of using VR applications in university lab course settings. A secondary evaluation hinted at existing deficits in subjective presence for users of the application, some of which can be ascribed to the general nature of the learning environment, while others should certainly be rectifiable in future versions.

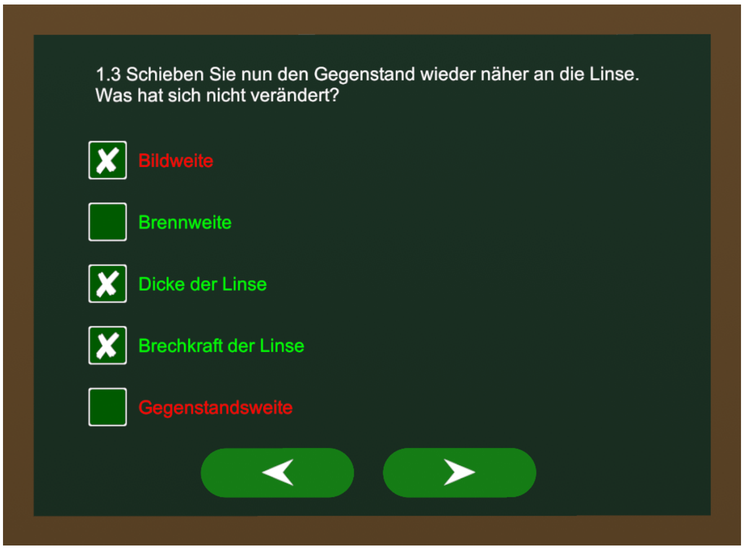
Further, we have analyzed the study results and the provided feedback and have devised a plan to alleviate the last remaining points of friction in terms of usability and to further enhance the immersion of the engagement. Building on the foundation we laid out, future work will aim at implementing these improvements and finalize the LIN-VR application to be ready for the use in the optics lab course.

Following this, a large scale study will be conducted in real universitary field use to investigate potential learning benefits of using our VR application in comparison with the traditional optical rail experiment.

A. Figures

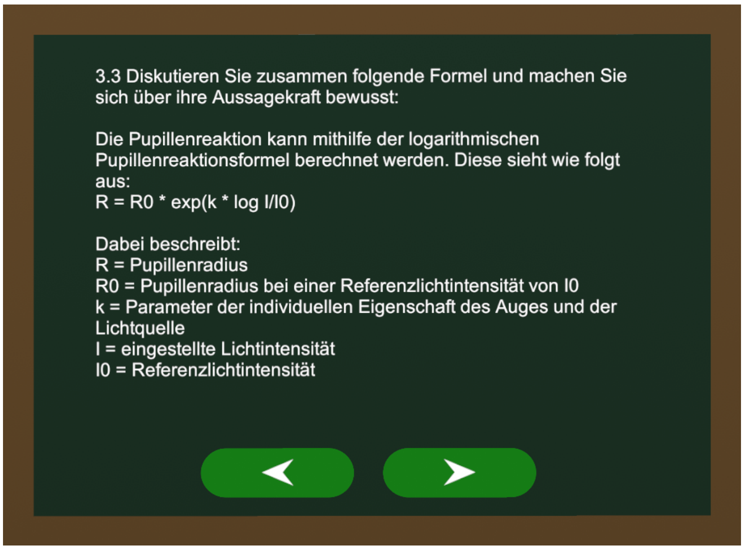


(a) MC-Question view

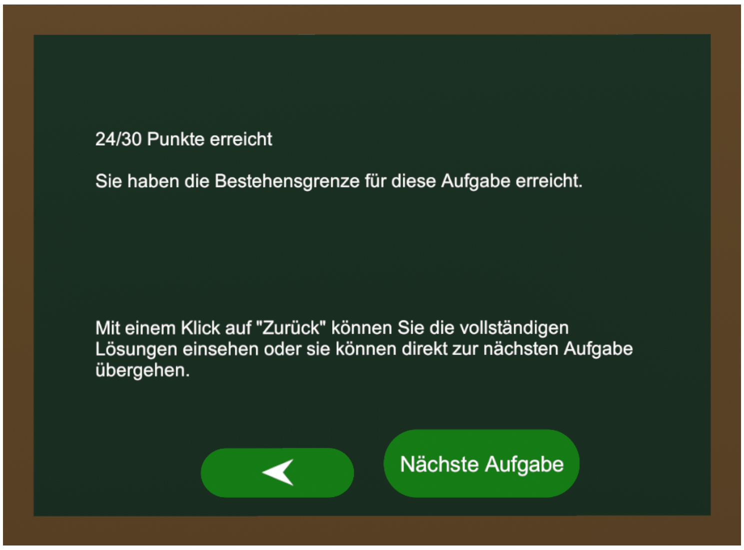


(b) MC-Question view during review

Figure A.1.: The four panel types of the exercise blackboard



(c) Info view



(d) Result view

Figure A.1.: The four panel types of the exercise blackboard (cont.)

B. Exercise Questions

The following pages contain the full catalogue of multiple choice questions and instructions included in the application (Exercises 1-3).

Aufgaben für die VR Anwendung

Zu Beginn wird ein allgemeiner Hinweis für die Studierenden auf der Tafel eingeblendet:

„Im Folgenden werden MC-Fragen und Anwendungsaufgaben gestellt. Während einer Versuchsreihe kann zwischen den Aufgaben hin und her gesprungen werden. Bei den MC-Fragen ist stets mindestens eine Aussage richtig. Jedes richtig gesetzte Kreuz (oder nicht Kreuz) gibt einen Punkt, bis zu einem Maximum von 5 Punkten. Es müssen mindestens 60% der gesamt möglichen Punkte erreicht werden, um die Aufgabe zu bestehen. Wird diese Grenze nicht erreicht müssen die Fragen erneut beantwortet werden.“

1. Aufgabe: Grundlegender Abbildungsvorgang im Auge

1.1 Verschieben Sie den Gegenstand auf der optischen Achse, bis auf der Netzhaut ein scharfes Bild entsteht.
(Studierenden kommen erst zur nächsten Aufgabe, wenn ein scharfes Bild auf der Retina ist, sprich wenn sie die richtigen Gegenstandsweite eingestellt haben)

1.2 Verschieben Sie erneut den Gegenstand auf der optischen Achse. Stellen Sie die größtmögliche Gegenstandsweite ein. Was kann beobachtet werden?

1.3 Schieben Sie nun den Gegenstand wieder näher an die Linse. Was hat sich **nicht** verändert?
a) Bildweite
b) Brennweite
c) Dicke der Linse
d) Brechkraft der Linse
e) Gegenstandsweite

1.4 Patienten, die an Presbyopie (Alterssichtigkeit) leiden, haben eine verkürzte Brennweite, weshalb sie z.B. Zeitungen weiter weg halten, um diese lesen zu können. Welche Weite wird hier aktiv vergrößert?
a) Brennweite
b) Bildweite
c) Doppelte Brennweite
d) Gegenstandsweite
e) Optische Achse

1.5 Um die Presbyopie auszugleichen können die Patienten Lesebrillen nutzen. Bei Lesebrillen handelt es sich um Sammellinsen, dies bedeutet
a) die Parallelstrahlen werden hinter Linse in einem Punkt vereinigt
b) die Parallelstrahlen verlaufen divergent auseinander
c) die Parallelstrahlen werden sind vor der Linse einem Punkt vereinigt
d) die Parallelstrahlen konvergieren
e) die Parallelstrahlen vergrößern die Wellenlänge

1.6 Welche der folgenden Bestandteile gehören zum dioptrischen Apparat?
a) Corpus vitreum
b) Choroidea
c) Lens oculi
d) Musculus ciliaris und zonula ciliaris
e) Cornea

1.7 In welcher Reihenfolge passieren einfallende Lichtstrahlen unser Auge?
   a) Cornea → Camera anterior bulbi → Lens oculi → Corpus vitreum → Retina
   b) Cornea → Lens oculi → Camera anterior bulbi → Corpus vitreum → Retina
   c) Cornea → Camera posterior bulbi → Lens oculi → Corpus vitreum → Retina
   d) Cornea → Lens oculi → Corpus vitreum → Camera anterior bulbi → Retina
   e) Cornea → Corpus vitreum → Lens oculi → Camera posterior bulbi → Retina

1.8 Das Auge kann auf mehr als nur eine Distanz scharf sehen. Welcher Prozess ermöglicht es dem Auge, sich auf Objekte unterschiedlicher Entfernung zu fokussieren?
   a) Refraktion
   b) Konvergenz
   c) Adaptation
   d) Akkommodation
   e) Dilatation

2. Aufgabe: Akkommodation der Linse

Automatische Akkommodation:

2.1 In folgender Versuchsreihe ist die Akkommodation der Linse aktiviert. Verschieben Sie erneut den Gegenstand auf der optischen Achse. Bei welchen anatomischen Strukturen können Veränderungen beobachtet werden?
   a) Pupilla
   b) Musculus ciliaris
   c) Zonula ciliaris
   d) Iris
   e) Lens oculi

2.2 Welche weiteren anatomischen Strukturen müssen zwangsläufig an der Akkommodation beteiligt sein?
   a) Choroidea
   b) Retina
   c) Nervus opticus
   d) Makula lutea
   e) Papilla nervi optici

2.3 Was verändert sich bei der automatischen Akkommodation der Linse nicht?
   a) Brennweite
   b) Bildweite
   c) Gegenstandsweite
   d) Gegenstandsgröße
   e) Krümmungsradius der Linse

2.4 Bei der Anpassung auf einen Gegenstand, der sich in der Nähe befindet wird:
   a) der Musculus ciliaris angespannt
   b) die Zonula ciliaris entspannt
   c) die Krümmungsradien verkleinert
   d) die Brechkraft der Lens oculi vergrößert
   e) die Brennweite der Lens oculi verkleinert

2.5 Die Grenzen der Akkommodation werden wie folgt genannt:
   a) Nahpunkt
   b) Fernpunkt
   c) Gegenstandspunkt
   d) Bildpunkt
   e) Akkommodationsbreiten

Manuelle Akkommodation:

2.6 In den folgenden Teilbereichen ist nun die automatische Akkommodation deaktiviert. Wir haben den Gegenstand auf den Nahpunkt gesetzt. Stellen Sie hierzu ein scharfes Bild auf der Retina ein. Wie verändert sich die Linse und damit einhergehend die Brechkraft?
   a) Der Musculus ciliaris zieht an der Lens oculi, sodass sie sich verdickt und die Brechkraft sinkt.
   b) Die Zonula ciliaris sind angespannt, weshalb die Lens oculi verschlankt wird und die Brechkraft sinkt.
   c) Durch Kontraktion der Musculus ciliaris verdickt die Lens oculi und die Brechkraft steigt.
   d) Durch Kontraktion der Zonula ciliaris verdickt die Lens oculi und die Brechkraft sinkt.
   e) Der Musculus ciliaris entspannt sich, wodurch die Lens oculi verschlankt wird und die Brechkraft steigt.

2.7 Sie sehen sich einen weit entfernten Baum an. Der Gegenstand wurde nun auf den Fernpunkt gesetzt. Was geschieht mit der Linse und damit einhergehend mit der Brennweite, damit ein scharfes Bild entsteht?
   a) Die Lens oculi verschlankt, weshalb die Brechkraft sinkt und die Brennweite steigt
   b) Die Lens oculi verdickt, weshalb die Brechkraft steigt und die Brennweite steigt
   c) Die Lens oculi verschlankt, weshalb die Brechkraft steigt und die Brennweite sinkt
   d) Die Lens oculi verschlankt, weshalb die Brechkraft steig und die Brennweite steigt
   e) Die Lens oculi verdickt, weshalb die Brechkraft sinkt und die Brennweite sinkt

3. Aufgabe: Adaptation der Pupille

Hinweis: In der ersten Hälfte dieser Versuchsreihe ist die Helligkeit des Gegenstandes konstant. Auf dem Tisch befindet sich ein virtueller Lichtschalter, dieser kann, falls nötig, betätigt werden, um Helligkeitsveränderungen in der Augensimulation besser zu erkennen..

Manuelle Adaptation

3.1 Verändern Sie den Radius der Pupille und beobachten Sie wie sich die Pupillengröße auf das Bild auf der Retina auswirkt.

3.2 Wie verändert sich das Bild auf der Retina, wenn sich der Pupillenradius vergrößert?
   a) Das Bild wird schärfer
   b) Bei stark erhöhter Lichtintensität vermindert sich die Farbwahrnehmung
   c) Das Bild wird dunkler
   d) Das Bild wird unschärfer
   e) Das Bild wird heller

3.3 Diskutieren Sie zusammen folgende Formel und machen Sie sich über ihre Aussagekraft bewusst: Die Pupillenreaktion kann mithilfe der logarithmischen Pupillenreaktionsformel berechnet werden. Diese sieht wie folgt aus:

$R = R0 * \exp (k* \log I/I0)$

Dabei beschreibt:

R = Pupillenradius

R0 = Pupillenradius bei einer Referenzlichtintensität von I0

k = Parameter der individuellen Eigenschaft des Auges und der Lichtquelle

I = eingestellte Lichtintensität

I0 = Referenzlichtintensität

3.4 Die logarithmische Pupillenreaktionsformel beschreibt:

a) Einen exponentiellen Anstieg der Pupillengröße bei erhöhter Lichtintensität

b) Einen nichtlinearen Zusammenhang zwischen der Lichtintensität und der Pupillenreaktion

c) Dass die Pupillengröße bei hohen Lichtintensitäten weniger stark auf Veränderung der Lichtintensität reagiert als bei niedrigen

d) Dass das Ergebnis der Pupillengröße Individuen abhängig ist

e) Nichts der obengenannten Aussagen

3.5 Was passiert mit dem einfallenden Licht, wenn sich die Pupille verkleinert?

a) Das Strahlenbündel wird durch die Verkleinerung der Durchtrittfläche konzentrierter
b) Aufgrund der kleiner Durchtrittsfläche gelangt weniger diffuses Licht ins Auge
c) Aufgrund der kleiner Durchtrittsfläche verändert sich die Dispersion der Wellenlänge
d) Der fokussierte Strahl verbessert die Bildschärfe
e) Der konzentrierte Strahl ermöglicht, dass Objekte in unterschiedlichen Entfernungen als scharf wahrgenommen werden können

<u>Automatische Adaptation</u>

3.6 Mithilfe des Reglers kann nun die Lichtintensität des Gegenstandes variiert werden. Welche Reaktionen können im Auge bei erhöhter bzw. verminderter Lichtintensität wahrgenommen werden?

3.7 Wie wird die Anpassung des Auges an unterschiedliche Lichteinfälle beschrieben?

a) Adaption
b) Dilatation
c) Dilation
d) Akkommodation
e) Adaptation

3.8 Was ist ein möglicher Auslöser für eine Pupillendilatation?

a) Hohe Lichtintensitäten
b) Niedrige Lichtintensitäten
c) Entspannung des Musculus ciliaris
d) Überreizung der Zonula ciliaris
e) Kurze Wellenlängen

3.9 Wird die Lichtintensität des Gegenstandes erhöht, so kann folgendes beobachtet werden:

a) Bei erhöhter Lichtintensität erweitert sich die Pupilla
b) Der Musculus ciliaris entspannt sich um die Lens oculi zu entlasten
c) Bei erhöhter Lichtintensität zieht sich die Pupilla zusammen
d) Die Retina schützt die Photorezeptoren vor der erhöhten Lichtintensität
e) Die Akkommodation der Lens oculi reguliert den Lichteinfall

C. Usability Questionnaire

The following pages contain the questionnaire used during the usability study.

Codewort:______________

Demografische Angaben

Alter: _________

Geschlecht: _____________

Studienfach: _____________

Ist Deutsch Ihre Muttersprache: ja( ) nein ( )

Benutzerfreundlichkeit

		Stimme überhaupt nicht zu				Stimme voll zu
1	Ich kann mir sehr gut vorstellen, das System regelmäßig zu nutzen.	☐	☐	☐	☐	☐
2	Ich empfand das System als unnötig komplex.	☐	☐	☐	☐	☐
3	Ich denke, das System ist einfach zu nutzen.	☐	☐	☐	☐	☐
4	Ich denke, ich bräuchte die Unterstützung einer technisch versierten Person, um das System nutzen zu können.	☐	☐	☐	☐	☐
5	Ich denke, die verschiedenen Funktionen des Systems waren gut integriert.	☐	☐	☐	☐	☐
6	Ich denke, es gibt zu viele Widersprüchlichkeiten im System.	☐	☐	☐	☐	☐
7	Ich kann mir vorstellen, dass die meisten Personen sehr schnell lernen das System zu nutzen.	☐	☐	☐	☐	☐
8	Ich empfand das System als sehr umständlich zu bedienen.	☐	☐	☐	☐	☐
9	Ich fühlte mich in der Bedienung des Systems sehr sicher.	☐	☐	☐	☐	☐
10	Ich musste viele Dinge lernen, bevor ich mit dem System arbeiten konnte.	☐	☐	☐	☐	☐

Präsenz in der virtuellen Umgebung

1. In der computererzeugten Welt hatte ich den Eindruck, dort gewesen zu sein...

überhaupt nicht	☐	☐	☐	☐	☐	☐	☐	sehr stark

2. Ich hatte das Gefühl, dass die virtuelle Umgebung hinter mir weitergeht.

trifft überhaupt nicht zu	☐	☐	☐	☐	☐	☐	☐	trifft voll und ganz zu

3. Ich hatte das Gefühl, nur Bilder zu sehen.

trifft überhaupt nicht zu	☐	☐	☐	☐	☐	☐	☐	trifft voll und ganz zu

4. Ich hatte nicht das Gefühl, in dem virtuellen Raum zu sein.

hatte nicht das Gefühl	☐	☐	☐	☐	☐	☐	☐	hatte das Gefühl

5. Ich hatte das Gefühl, in dem virtuellen Raum zu handeln statt etwas von außen zu bedienen.

trifft überhaupt nicht zu	☐	☐	☐	☐	☐	☐	☐	trifft voll und ganz zu

6. Ich fühlte mich im virtuellen Raum anwesend.

trifft überhaupt nicht zu	☐	☐	☐	☐	☐	☐	☐	trifft voll und ganz zu

7. Wie bewusst war Ihnen die reale Welt, während Sie sich durch die virtuelle Welt bewegten (z.B. Geräusche, Raumtemperatur, andere Personen etc.)?

extrem bewusst	☐	☐	☐	☐	☐	☐	☐	vollkommen unbewusst

8. Meine reale Umgebung war mir nicht mehr bewusst.

trifft überhaupt nicht zu	☐	☐	☐	☐	☐	☐	☐	trifft voll und ganz zu

9. Ich achtete noch auf die reale Umgebung.

trifft überhaupt nicht zu	☐	☐	☐	☐	☐	☐	☐	trifft voll und ganz zu

10. Meine Aufmerksamkeit war von der virtuellen Welt völlig in Bann gezogen.

trifft überhaupt nicht zu	☐	☐	☐	☐	☐	☐	☐	trifft voll und ganz zu

11. Wie real erschien Ihnen die virtuelle Umgebung?

vollkommen real	☐	☐	☐	☐	☐	☐	☐	gar nicht real

12. Wie sehr glich Ihr Erleben der virtuellen Umgebung dem Erleben einer realen Umgebung?

überhaupt nicht	☐	☐	☐	☐	☐	☐	☐	vollständig

13. Wie real erschien Ihnen die virtuelle Welt?

wie eine vorgestellte Welt	☐	☐	☐	☐	☐	☐	☐	nicht zu unterscheiden von der realen Welt

14. Die virtuelle Welt erschien mir wirklicher als die reale Welt.

trifft überhaupt nicht zu	☐	☐	☐	☐	☐	☐	☐	trifft voll und ganz zu

Offene Fragen (nutzen Sie falls nötig gerne die Rückseite)

Was hat Ihnen an der Anwendung besonders gefallen?

Wo sehen Sie noch Potential für Verbesserung bei der Anwendung?

Könnten Sie sich vorstellen, Anwendungen dieser Art in Uni-Praktika zu verwenden? Falls ja, was wäre für Sie der wichtigste Faktor, der in einer solchen Anwendung berücksichtigt sein muss?

Erfahrung mit digitalen Medien und Geräten

		Stimme überhaupt nicht zu			Stimme voll zu
1	Ich fühle mich auch bei der Nutzung digitaler Geräte gut, mit denen ich weniger vertraut bin.	☐	☐	☐	☐
2	Wenn Freundinnen, Freunde oder Verwandte neue digitale Geräte oder Anwendungen kaufen wollen, kann ich Ihnen Ratschläge geben.	☐	☐	☐	☐
3	Ich fühle mich gut, wenn ich zu Hause meine digitalen Geräte nutze.	☐	☐	☐	☐
4	Wenn sich ein Problem mit einem digitalen Gerät ergibt, denke ich, dass ich es lösen kann.	☐	☐	☐	☐
5	Wenn meine Freundinnen, Freunde oder Verwandte ein Problem mit einem digitalen Gerät haben, kann ich ihnen helfen.	☐	☐	☐	☐

6	Dies stellt meine erste Berührung mit Virtual Reality dar.	ja	☐	nein	☐

Interesse an Medizin und Physiologie

		Stimme überhaupt nicht zu			Stimme voll zu
1	Im Allgemeinen macht es mir Spaß, mich mit medizinischen Themen zu befassen.	☐	☐	☐	☐
2	Ich lese gerne etwas über die Funktionsweise des Körpers.	☐	☐	☐	☐
3	Ich beschäftige mich gerne mit physiologischen Zusammenhängen.	☐	☐	☐	☐
4	Ich eigne mir gerne neues Wissen über den menschlichen Körper an.	☐	☐	☐	☐
5	Ich bin interessiert, Neues aus der Medizin zu lernen.	☐	☐	☐	☐

List of Figures

List of Tables

Abbreviations and Acronyms

AR Augmented Reality. 1

GUI Graphical User Interface. 1, 30

HDR High Dynamic Range. 1, 49

HMD Head Mounted Display. 1

INV Involvement. 1, 56

IPQ Igroup Presence Questionnaire. 1, 56

PRES Presence. 1, 56

REAL Experienced Realism. 1, 56

SP Spatial Presence. 1, 56

SUS System Usability Scale. 1, 56

UI User Interface. 1, 30

VR Virtual Reality. 1

XR Extended Reality. 1

Bibliography

[1] R. B. Loftin, M. Engleberg, and R. Benedetti. "Applying virtual reality in education: A prototypical virtual physics laboratory". In: *Proceedings of 1993 IEEE Research Properties in Virtual Reality Symposium*. Journal Abbreviation: Proceedings of 1993 IEEE Research Properties in Virtual Reality Symposium. Oct. 1993, pp. 67–74. DOI: 10.1109/VRAIS.1993.378261.

[2] Y. Georgiou, O. Tsivitanidou, and A. Ioannou. "Learning experience design with immersive virtual reality in physics education". In: *Educational Technology Research and Development* 69.6 (Dec. 2021), pp. 3051–3080. ISSN: 1556-6501. DOI: 10.1007/s11423-021-10055-y. URL: https://doi.org/10.1007/s11423-021-10055-y.

[3] P. Šiđanin, J. Plavšić, I. Arsenić, and M. Krmar. "Virtual reality (VR) simulation of a nuclear physics laboratory exercise". In: *European Journal of Physics* 41.6 (Oct. 2020). Publisher: IOP Publishing, p. 065802. ISSN: 0143-0807. DOI: 10.1088/1361-6404/ab9c90. URL: https://dx.doi.org/10.1088/1361-6404/ab9c90.

[4] R. A. Engeln and S. M. Gómez Puente. "Does virtual reality help students learn to use optical measurement techniques?" In: *10th International Conference on Physics Teaching in Engineering Education PTEE 2019* (May 2019). (Visited on 05/23/2019).

[5] A. Y. Al Amri, M. E. Osman, and A. S. Al Musawi. "The Effectiveness of a 3D-Virtual Reality Learning Environment (3D-VRLE) on the Omani Eighth Grade Students' Achievement and Motivation towards Physics Learning". In: *International Journal of Emerging Technologies in Learning (iJET)* 15.05 (Mar. 2020). Section: Papers, pp. 4–16. DOI: 10.3991/ijet.v15i05.11890. URL: https://online-journals.org/index.php/i-jet/article/view/11890 (visited on 01/17/2023).

[6] M. P. Strzys, S. Kapp, M. Thees, P. Klein, P. Lukowicz, P. Knierim, A. Schmidt, and J. Kuhn. "Physics holo.lab learning experience: using smartglasses for augmented reality labwork to foster the concepts of heat conduction". In: *European Journal of Physics* 39.3 (Mar. 2018). Publisher: IOP Publishing, p. 035703. ISSN: 0143-0807. DOI: 10.1088/1361-6404/aaa8fb. URL: https://dx.doi.org/10.1088/1361-6404/aaa8fb.

[7] W. Tarng, K.-L. Ou, Y.-C. Lu, Y.-S. Shih, and H.-H. Liou. "A Sun Path Observation System Based on Augment Reality and Mobile Learning". In: *Mobile Information Systems* 2018 (Mar. 2018). Ed. by J. Fontecha. Publisher: Hindawi, p. 5950732. ISSN: 1574-017X. DOI: 10.1155/2018/5950732. URL: https://doi.org/10.1155/2018/5950732.

[8] K. Altmeyer, S. Kapp, M. Thees, S. Malone, J. Kuhn, and R. Brünken. "The use of augmented reality to foster conceptual knowledge acquisition in STEM laboratory courses—Theoretical background and empirical results". In: *British Journal of Educational Technology* 51.3 (May 2020). Publisher: John Wiley & Sons, Ltd, pp. 611–628. ISSN: 0007-1013. DOI: 10.1111/bjet.12900. URL: https://doi.org/10.1111/bjet.12900 (visited on 06/13/2023).

[9] H. S. Maresky, A. Oikonomou, I. Ali, N. Ditkofsky, M. Pakkal, and B. Ballyk. "Virtual reality and cardiac anatomy: Exploring immersive three-dimensional cardiac imaging, a pilot study in undergraduate medical anatomy education". In: *Clinical Anatomy* 32.2 (Mar. 2019). Publisher: John Wiley & Sons, Ltd, pp. 238–243. ISSN: 0897-3806. DOI: 10.1002/ca.23292. URL: https://doi.org/10.1002/ca.23292 (visited on 01/15/2023).

[10] M. Williams. "Virtual reality in ophthalmology education: simulating pupil examination". In: *Eye* 36.11 (Nov. 2022), pp. 2084–2085. ISSN: 1476-5454. DOI: 10.1038/s41433-022-02078-3. URL: https://doi.org/10.1038/s41433-022-02078-3.

[11] A. S. Wilson, J. O'Connor, L. Taylor, and D. Carruthers. "A 3D virtual reality ophthalmoscopy trainer". In: *The Clinical Teacher* 14.6 (Dec. 2017). Publisher: John Wiley & Sons, Ltd, pp. 427–431. ISSN: 1743-4971. DOI: 10.1111/tct.12646. URL: https://doi.org/10.1111/tct.12646 (visited on 01/09/2023).

[12] M. Chan, A. Uribe-Quevedo, B. Kapralos, M. Jenkin, N. Jaimes, and K. Kanev. "Virtual and Augmented Reality Direct Ophthalmoscopy Tool: A Comparison between Interactions Methods". In: *Multimodal Technologies and Interaction* 5.11 (2021). ISSN: 2414-4088. DOI: 10.3390/mti5110066.

[13] W. Demtröder. *Experimentalphysik 2: Elektrizität und Optik*. 7th ed. Springer-Lehrbuch. Berlin, Heidelberg: Springer Spektrum, 2017. ISBN: 978-3-662-55790-7. DOI: 10.1007/978-3-662-55790-7. URL: https://doi.org/10.1007/978-3-662-55790-7.

[14] W. A. Müller, S. Frings, and F. Möhrlen. "Der Sehsinn". In: *Tier- und Humanphysiologie: Eine Einführung*. Ed. by W. A. Müller, S. Frings, and F. Möhrlen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 601–665. ISBN: 978-3-662-58462-0. DOI: 10.1007/978-3-662-58462-0_19. URL: https://doi.org/10.1007/978-3-662-58462-0_19.

[15] H. J. Eichler, H.-D. Kronfeldt, and J. Sahm. "Linsen". In: *Das Neue Physikalische Grundpraktikum*. Berlin, Heidelberg: Springer, 2001, pp. 341–354. ISBN: 978-3-662-21753-5. DOI: 10.1007/978-3-662-21753-5_33. URL: https://doi.org/10.1007/978-3-662-21753-5_33.

[16] B. Lachenmayr, D. Friedburg, E. Hartmann, and A. Buser. *Auge - Brille - Refraktion: Schober-Kurs: verstehen - lernen - anwenden*. 4th ed. Stuttgart: Thieme, 2006. ISBN: 3-13-139554-0.

[17] P. Walter and N. Plange. *Basiswissen Augenheilkunde*. 1st ed. Springer-Lehrbuch. Berlin, Heidelberg: Springer, 2017. ISBN: 978-3-662-52800-6. DOI: 10.1007/978-3-662-52801-3. URL: https://doi.org/10.1007/978-3-662-52801-3.

[18] *Das Auge: Aufbau & Funktion - Kuratorium Gutes Sehen e.V.* URL: https://www.sehen.de/sehen/rund-ums-auge/das-auge-aufbau-und-funktion/ (visited on 04/22/2023).

[19] R. D. Gerste. *Der Graue Star: Etablierte Operationsverfahren und neue Lasertechnik*. 1st ed. Berlin, Heidelberg: Springer, 2016. ISBN: 978-3-662-47281-1. DOI: 10.1007/978-3-662-47282-8. URL: https://doi.org/10.1007/978-3-662-47282-8.

[20] S. Silbernagl and A. Despopoulos. *Taschenatlas Physiologie*. 7th ed. Georg Thieme Verlag, 2007. ISBN: 978-3-13-567707-1.

[21] G. K. Krieglstein, C. P. Jonescu-Cuypers, M. Severin, and M. A. Vobig. *Atlas of Ophthalmology*. 1st ed. Berlin, Heidelberg: Springer, 2000. DOI: 10.1007/978-3-642-57132-9. URL: https://doi.org/10.1007/978-3-642-57132-9.

[22] *Physikpraktikum für Humanmediziner - LMU München*. URL: https://www.praktikum.physik.uni-muenchen.de/humanmed/index.html (visited on 05/23/2023).

[23] *PMed - LIN - Brennweite Brillenglas*. URL: https://www.didaktik.physik.uni-muenchen.de/pmed/lin/lin_v1.php (visited on 05/23/2023).

[24] *Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine*. URL: https://unity.com/ (visited on 05/13/2023).

[25] *HP Customer Support - Knowledge Base*. URL: https://support.hp.com/rs-en/product/hp-reverb-g2-virtual-reality-headset/33835976/document/c06938191#AbT3 (visited on 04/21/2023).

[26] *Auge - RPTU Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau*. URL: https://rptu.de/uedu/arbeitsfelder/ausbildungskonzepte-af2/auge (visited on 05/04/2023).

[27] L. L. Holladay. "The Fundamentals of Glare and Visibility". In: *Journal of the Optical Society of America* 12.4 (Apr. 1926). Publisher: Optica Publishing Group, pp. 271–319. DOI: 10.1364/JOSA.12.000271. URL: https://opg.optica.org/abstract.cfm?URI=josa-12-4-271.

[28] B. H. Crawford and J. H. Parsons. "The dependence of pupil size upon external light stimulus under static and variable conditions". In: *Proceedings of the Royal Society of London. Series B - Biological Sciences* 121.823 (1936). _eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rspb.1936.0072, pp. 376–395. DOI: 10.1098/rspb.1936.0072. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rspb.1936.0072.

[29] S. G. de Groot and J. W. Gebhard. "Pupil Size as Determined by Adapting Luminance". In: *Journal of the Optical Society of America* 42.7 (July 1952). Publisher: Optica Publishing Group, pp. 492–495. DOI: 10.1364/JOSA.42.000492. URL: https://opg.optica.org/abstract.cfm?URI=josa-42-7-492.

[30] P. A. Stanley and A. Davies. "The effect of field of view size on steady-state pupil diameter". In: *Ophthalmic and Physiological Optics* 15.6 (Nov. 1995), pp. 601–603. ISSN: 0275-5408. DOI: 10.1016/0275-5408(94)00019-V. URL: https://www.sciencedirect.com/science/article/pii/027554089400019V.

[31] P. Napieralski and F. Rynkiewicz. "Modeling Human Pupil Dilation to Decouple the Pupillary Light Reflex". In: Open Physics 17.1 (2019), pp. 458–467. DOI: 10.1515/phys-2019-0047. URL: https://doi.org/10.1515/phys-2019-0047 (visited on 05/13/2023).

[32] V. F. Pamplona, M. M. Oliveira, and G. V. G. Baranoski. "Photorealistic Models for Pupil Light Reflex and Iridal Pattern Deformation". In: *ACM Trans. Graph.* 28.4 (Sept. 2009). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0730-0301. DOI: 10.1145/1559755.1559763. URL: https://doi.org/10.1145/1559755.1559763.

[33] P. Moon and D. E. Spencer. "On the Stiles-Crawford Effect". In: *Journal of the Optical Society of America* 34.6 (June 1944). Publisher: Optica Publishing Group, pp. 319–329. DOI: 10.1364/JOSA.34.000319. URL: https://opg.optica.org/abstract.cfm?URI=josa-34-6-319.

[34] A. Watson and J. Yellott. "A unified formula for light-adapted pupil size". In: *Journal of vision* 12 (Sept. 2012). DOI: 10.1167/12.10.12.

[35] *17-21-050 | CIE*. URL: https://cie.co.at/eilvterm/17-21-050 (visited on 05/15/2023).

[36] *Unity UI: Unity User Interface | Unity UI | 1.0.0*. URL: https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/index.html (visited on 05/27/2023).

[37] *How inside-out tracking works - Enthusiast Guide | Microsoft Learn*. URL: https://learn.microsoft.com/en-us/windows/mixed-reality/enthusiast-guide/tracking-system (visited on 06/02/2023).

[38] J. Knight and C. Baber. "Neck Muscle Activity and Perceived Pain and Discomfort Due to Variations of Head Load and Posture". In: *Aviation, space, and environmental medicine* 75 (Mar. 2004), pp. 123–31.

[39] *Toggle Group | Unity UI | 1.0.0*. URL: https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/script-ToggleGroup.html (visited on 05/26/2023).

[40] *Radio Buttons - Win32 apps | Microsoft Learn*. URL: https://learn.microsoft.com/en-us/windows/win32/uxguide/ctrl-radio-buttons (visited on 05/26/2023).

[41] *Toggles | Apple Developer Documentation*. URL: https://developer.apple.com/design/human-interface-guidelines/toggles/#Radio-buttons (visited on 05/26/2023).

[42] *JSON*. URL: https://www.json.org/json-en.html (visited on 06/04/2023).

[43] *Action Delegate (System) | Microsoft Learn*. URL: https://learn.microsoft.com/en-us/dotnet/api/system.action?view=netstandard-2.1 (visited on 04/27/2023).

[44] *Lambda expressions - Lambda expressions and anonymous functions | Microsoft Learn*. URL: https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/operators/lambda-expressions (visited on 06/04/2023).

[45] Zii. *UI BLUR in Unity ShaderGraph for 2D & 3D games*. June 2020. URL: https://www.youtube.com/watch?v=cjfNgra1uWk (visited on 03/31/2023).

[46] Code Monkey. *Sprite Blur - 2D Shader Graph Tutorial*. Dec. 2019. URL: https://www.youtube.com/watch?v=dKsqupXLuSU (visited on 03/31/2023).

[47] *GIMP - GNU Image Manipulation Program*. URL: https://www.gimp.org/ (visited on 05/13/2023).

[48] S. Green and C. Cebenoyan. "High dynamic range rendering on the geforce 6800". In: (Jan. 2004). URL: https://www.researchgate.net/publication/228787225_High_dynamic_range_rendering_on_the_geforce_6800.

[49] *Post Processing Stack v2 overview | Post Processing | 3.3.0*. URL: https://docs.unity3d.com/Packages/com.unity.postprocessing@3.3/manual/index.html (visited on 05/29/2023).

[50] G. Spencer, P. Shirley, K. Zimmerman, and D. P. Greenberg. "Physically-Based Glare Effects for Digital Images". In: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '95. New York, NY, USA: Association for Computing Machinery, 1995, pp. 325–334. ISBN: 0-89791-701-4. DOI: 10.1145/218380.218466. URL: https://doi.org/10.1145/218380.218466.

[51] S. S. Stevens. "On the psychophysical law." In: *Psychological Review* 64.3 (1957). ISBN: 1939-1471 Publisher: American Psychological Association, pp. 153–181. DOI: https://doi.org/10.1037/h0046162.

[52] *Unity - Scripting API: Mesh*. URL: https://docs.unity3d.com/ScriptReference/Mesh.html (visited on 05/16/2023).

[53] J. Brooke. "SUS: A quick and dirty usability scale". In: *Usability Evaluation In Industry* 189 (Nov. 1995).

[54] T. Schubert, F. Friedmann, and H. Regenbrecht. "The Experience of Presence: Factor Analytic Insights". In: *Presence: Teleoperators and Virtual Environments* 10.3 (June 2001), pp. 266–281. DOI: 10.1162/105474601300343603. URL: https://doi.org/10.1162/105474601300343603 (visited on 10/05/2023).

[55] *igroup presence questionnaire (IPQ) overview | igroup.org – project consortium*. URL: http://www.igroup.org/pq/ipq/index.php (visited on 06/10/2023).

[56] J. Vasconcelos-Raposo, M. Bessa, M. Melo, L. Barbosa, R. Rodrigues, C. M. Teixeira, L. Cabral, and A. A. Sousa. "Adaptation and Validation of the Igroup Presence Questionnaire (IPQ) in a Portuguese Sample". In: *Presence: Teleoperators and Virtual Environments* 25.3 (Dec. 2016), pp. 191–203. DOI: 10.1162/PRES_a_00261. URL: https://doi.org/10.1162/PRES_a_00261 (visited on 10/05/2023).

[57] *IPQ Calculo 1*. URL: https://massive.inesctec.pt/wp-content/uploads/2016/11/IPQ-Calculo-1.pdf (visited on 06/10/2023).

[58] J. Mang, N. Ustjanzew, I. Leßke, A. Schiepe-Tiska, and K. Reiss. *PISA 2015 Skalenhandbuch: Dokumentation der Erhebungsinstrumente*. Waxmann Verlag, 2019. ISBN: 3-8309-9032-4. URL: https://www.pisa.tum.de/fileadmin/w00bgi/www/Berichtsbaende_und_Zusammenfassungungen/Skalenhandbuch_PISA2015_openaccess.pdf.

[59] *Mixed Reality for developers - Microsoft Developer | Microsoft Developer*. URL: https://developer.microsoft.com/en-us/mixed-reality/ (visited on 06/07/2023).

[60] J. Sauro. *A practical guide to the system usability scale: Background, benchmarks & best practices*. Denver: Measuring Usability LLC, 2011. ISBN: 1-4610-6270-5.

[61] *5 Ways to Interpret a SUS Score – MeasuringU*. URL: https://measuringu.com/interpret-sus-score/ (visited on 06/10/2023).

[62] M. Melo, G. Gonçalves, j. Vasconcelos-Raposo, and M. Bessa. “How Much Presence is Enough? Qualitative Scales for Interpreting the Igroup Presence Questionnaire Score”. In: *IEEE Access* 11 (2023), pp. 24675–24685. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3254892.