Otto-von-Guericke University Magdeburg

Faculty of Computer Science

Master Thesis

Supporting Anatomy Education with a 3D Puzzle in a Virtual Reality Environment

Daniel Pohlandt, B.Sc

First Examiner:	Prof. Dr.-Ing. Bernhard Preim
Second Examiner:	Jun.-Prof. Dr. Christian Hansen
Supervisor:	Patrick Saalfeld, M.Sc

Filing Date: 19.06.2017

Abstract

This thesis introduces an approach for an educational scenario in a fully immersive virtual reality environment. By solving 3D puzzles of anatomical models, medical students shall learn names and spacial relations of different anatomical structures. The approach is proposed as a supplement to the work with traditional human body donors. To meet the needs of medical students and satisfy the different stages of learning, the approach contains different modes, that help the student to gather and review medical knowledge. Different feedback features, that support the student with the puzzle task in the virtual reality world, are proposed as well. The concept is realized as a prototype application using the *HTC Vive* and the *Unity* game engine. This prototype is evaluated in a user study, which mainly produces qualitative results. The study suggests that a 3D puzzle in a fully immersive virtual reality environment can potentially be an interesting and valuable extension to traditional teaching methods in the medical curriculum. A more thorough long-term evaluation is needed to support this impression.

Contents

List of Figures

Abbreviations

CAL Computer-Aided Learning

DOF Degrees of Freedom

FPS Frames per Seconds

HMD Head Mounted Display

HWD Head Worn Display

POI Proband of Interest

UI User Interface

VR Virtual Reality

1 Introduction

For many environments, learning involves the understanding of complex spatial relations. Engineers need to know where parts of a machine are situated even if they are not visible from the outside. This especially applies for the most complex "machine" known to mankind; the human body. In both fields of applications it is essential to learn and understand the position of even the tiniest part of a system.
Traditionally, learning is based on abstract visualizations in books or real life models. These methods are usually limited by the lack of interactivity. Nowadays, various computer applications are a promising alternative to traditional learning formats. Especially learning the human anatomy becomes significantly easier with the use of precise anatomical data and interactive feedback. For this purpose virtual 3D puzzle games have already been introduced 15 years ago [63, 65]. With the rise of high-quality virtual reality headsets such as Oculus Rift and HTC Vive, immersive virtual reality applications are a reasonable extension to provide a proper learning environment [75]. In this context, a 3D puzzle is a good application example since it takes simple real-world mechanics and extends them with powerful information visualization tools [46].

1.1 Objectives

This thesis proposes a prototype solution for an educational scenario in form of a *Virtual Reality* 3D puzzle. The main objective of the scenario is to support anatomical education for medical students. By using the 3D puzzle, a student shall receive an easy and intuitive learning experience that is almost as educational as a real body donor. It shall explicitly be discussed how this education process can be supported in general and how a 3D puzzle can be of use. Further, the thesis shall deal with the question if and how virtual reality can provide a useful environment for such an application. Here, the topic of sound will not be covered in any way.

From a technical point of view the prototype shall generate a legit and clear solution space for any given puzzle surface mesh data. To comply with the puzzle functionality the given data shall only provide enough information about the position of the puzzle pieces in the correct solution. The data is then processed at runtime. Further, the concept provides solu-

tions to assist the user with solving the puzzle.
The developed prototype is then realized in the game engine *Unity* using the head-mounted display *HTC Vive*. Finally, the prototype is evaluated with a user study. Particular attention is hereby paid to the feedback features that intend to assist the user.

1.2 Structure

The thesis is structured into the following chapters,:

Chapter 2 - Related Work introduces the necessary background of *Virtual Reality* and existing interaction concepts are shown. It further describes the basics of anatomical teaching and presents existing solutions.

Chapter 3 - Approach describes the requirements that need to be met by the proposed educational scenario. It further describes and debates every detail of a concept that meets these requirements.

Chapter 4 - Implementation briefly states relevant aspects of the process of implementing a working prototype of the proposed concept. The used hardware is more precisely examined and the used game engine *Unity* is briefly described.

Chapter 5 - Evaluation deals with the evaluation of the implemented prototype. This comprises the selection of viable evaluation methods, the evaluation setup, its execution and the gained results. Beforehand, some feedback is presented that has already been received while the prototype was still under development.

Chapter 6 - Conclusion summarizes the thesis and formulates a closing appraisal for the results of the thesis. The concluding section discusses possible approaches for future work.

2 Related Work

The following chapter presents necessary terms and concepts that are relevant for the comprehension of this thesis. Relevant scientific papers are referenced to present a general outline of existing and contemporary developments. Initially the term of *Virtual Reality* is specified. In this context the basic principles of visual perception are presented as well as a historical overview and relevant interaction concepts. Following this the chapter describes the basics of anatomical teaching and the part that computer technologies take in it.

2.1 Virtual Reality

The Oxford Dictionary defines *virtual reality* or *VR* as:

> "The computer-generated simulation of a three-dimensional image or environment that can be interacted with in a seemingly real or physical way by a person using special electronic equipment, such as a helmet with a screen inside or gloves fitted with sensors."[74]

In a perfect virtual environment, every last detail, generated by the computer would be perceived as real by the human observer. Every sense would be triggered and could be fooled by a digital replication of the real world. As early as 1969, the father of the concept of virtual reality, Ivan E. Sutherland, described this experience even further:

> "Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal."[77]

Even though the current technology is still far from close to Sutherland's vision, there have already been real-time VR applications introduced, that are under immediate control of the user [8]. Four key elements are of fundamental importance to define the quality of the virtual reality [70]. Foremost, the *virtual world* itself. This collection of objects in a space and the rules and relationships governing those objects are often manifested through a medium and then experienced as virtual reality. This experience is further enhanced through the feeling of *immersion*. This term can be distinguished in mental and physical immersion. Mental immersion is often described as being in a state of *deep engagement* or *involvement* with an environment. Additionally, physical immersion implies the feeling of *physical presence* in an environment by stimulating the bodies senses. This does not necessarily imply that all senses need to be stimulated of that the entire body is immersed. Another essential ingredient to virtual reality is the *sensory feedback*. This describes the real-time feedback the user receives as a result of his position in the virtual world or his interaction with it. This *interactivity* represents the final key aspect. Not only is the user present in a virtual reality environment, he has also the ability to manipulate the environment to a certain degree. These aspects are realised by the use of different input and output devices, further described in section 2.1.4. Before looking at the actual hardware, it is relevant to understand functionality of the human visual perception, which is the foundation for today's virtual reality technologies.

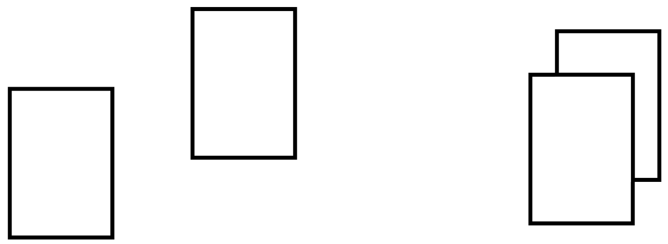
2.1.1 Visual Perception

Nowadays, the realization of virtual reality is often linked to the concept of head mounted displays or *HMDs*. Such displays take advantage of the stereopsis of the human eye and other different depth cues, to create the impression of three-dimensionality from two-dimensional images. In the current scientific context, these displays are also referred to as head worn displays, or *HWDs*.

Depth Cues

The human eyes are recording two different images at the same time, but the person does only perceive one single picture. This can be attributed to the human brain, which processes those two pictures into a single three-dimensional representation. This provides the possibility to properly estimate distances and have objects appear three-dimensional. Not all objects are perceived sharply that appear in the human field of view. This can be accounted to the area of *sharp vision*, where seen objects can be perceived most clearly. The area of *sharp vision* is created by the *fovea centralis*, where the density of cone cells is highest. This effectively results in a higher resolution of the picture that is perceived in this area. To fixate smaller or faraway areas, the human eyes make use of *convergence* movements, where the viewing rays are no longer parallel but instead cross in some point. Other key factors are the *accomodation*, the adaption of the eye's optical power, and *miosis*, the constriction of the pupil to widen the depth of field [13]. These three factors are utilized in current 3D-technologies, to give depth to images that appear on flat screens. This makes stereopsis one of the most crucial aspects for the development of novel display technologies and especially head-mounted displays that are necessary for virtual reality experiences.
Next to physical factors that are solely determined by the functionality of the human eye, other aspects affect the depth perception. These factors are usually learned through daily experience. More precisely, how an object is perceived, in terms of depth, depends on the information that is present about the object and its environment [55]. One of the most interesting depth cues is the so called *motion parallax*. Even though it is hardly applicable

to still images, it is most important to create the impression of depth in moving images. The effect can usually be seen by looking out of the window of a moving vehicle. Objects that are further away, appear to be moving significantly slower than closer objects.
By experience, the observer knows that, if objects block each other out of sight, the object that blocks the other one is usually closer. It can be said that an object whose outline pattern looks more continuous is felt to lie closer to the observer (see figure 2.1b). This cue is commonly referred to as *overlapping*.



(a) (b)

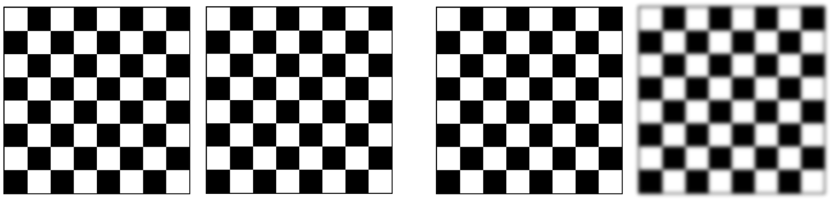
Figure 2.1: *Overlapping*: The left figure shows two rectangles without any depth information. In the right figure, one of the rectangle appears to be behind the other one.

Another relevant cue is the *retinal image size*. When the real world size of an object is known, the human brain compares the sensed size of an object to this known size, thus, acquires information about the distance of the object and its placement to other objects. In figure 2.2a two people appear to be standing next to each other whereas in figure 2.2b one of them appears to be further behind.

(a) (b)

Figure 2.2: *Retinal Image Size*: The left figure shows two people next to each other. In the right figure, one of them appears to be far behind the other.

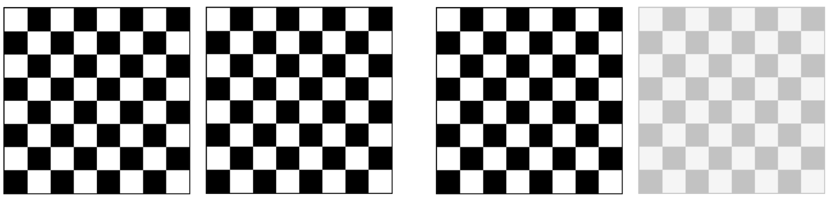
If more information about an object is known, the *texture gradient* cue is also important. The closer the observer is to an object the more detail he can see of its surface texture. Objects with smooth textures are often interpreted as being farther away. This effect is especially visible if the surface texture spans all the distance from near to far. In figure 2.3b the blurry and less detailed chess board appears to be further in the back.



(a) (b)

Figure 2.3: *Texture Gradient*: The left figure shows two chess boards without depth information. In the right figure, one of them is less clearly visible which lets it appear to be further away.

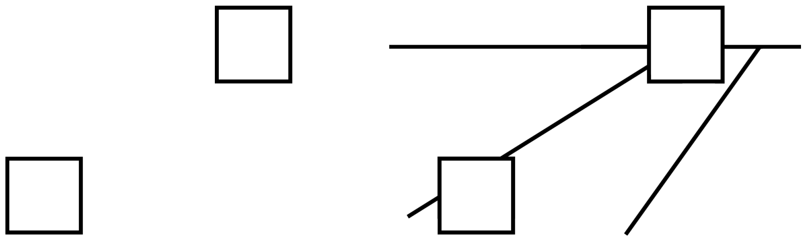
In this context, the *aerial perspective* cue is also relevant. Mountains in the horizon look always slightly hazy. Sometimes it even appears as if they take on the bluish or yellowish color tone of their surroundings. To the observer it looks as if the object is blending into the background (see figure 2.4b). This is due to particles (e.g. dust or water) in the air between the eye and the mountains. For this reason, it can be said that the farther an object, the hazier it looks.



(a) (b)

Figure 2.4: *Aerial Perspective*: The left figure shows two chess boards without depth information. In the right figure, one of them appears to be blending into the background, thus, making it appear to be further away.

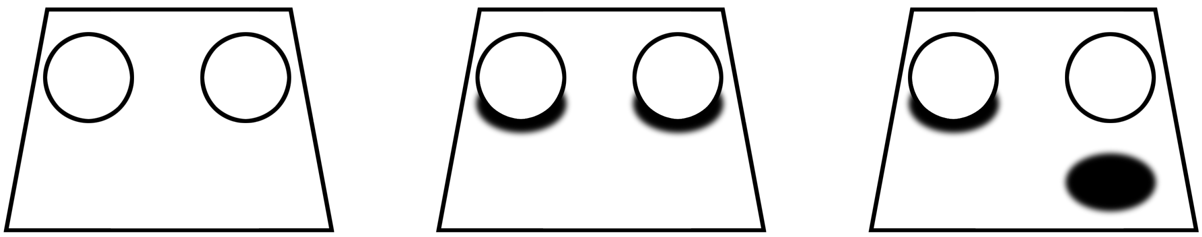
The *linear perspective* cue also depends on the horizon. When looking down a straight level road it can be seen, that the parallel sides of the road meet in the horizon. In two-dimensional pictures, the part where the road sides are more narrow appear to be further away. This can be said about all straight lines that are pointing towards a mutual vanishing point. In figure 2.4a, both squares have apparently the same distance from the observer. By adding a horizon and two diagonal lines, one of the squares in figure 2.4b appears to be further away. If the observer is familiar with the environment, i.e., knows the location of one more multiple light sources, he can gather valuable information from the *shades and shadows* cue. It can be recognized that an object casting shadows on other objects must be closer to the light



(a) (b)

Figure 2.5: *Linear Perspective*: The left figure shows two squares without depth information. In the right figure, the square in the more narrow part of the two diagonal lines appears to be further away.

source. Whereas the distance between an object can give further information about the distance between the object and the object its casting a shadow on. Additionally, bright objects often to be closer to the observer than darker ones. Figure 2.6a shows two circles and a quadrangle without distinct depth information. By adding shadows, figure 2.6b creates the impression that the two circles are resting on the quadrangle. The difference of the shadows' positions in 2.6c invokes the effect of one circle hovering over the quadrangle.



(a) (b) (c)

Figure 2.6: *Shades and Shadows*: The left figure shows two circles and a quadrangle without clear depth information. In the centre figure, both circles are apparently resting on the quadrangle. Finally, in the right figure, one of the circles is apparently hovering.

Stereopsis and all of the presented cues enable humans to generate a three-dimensional perception of their surroundings. They already are used in computer applications but for virtual reality systems, they are more important than ever before [58].

2.1.2 Genesis of Virtual Reality Devices

In 1987, the highly acclaimed TV show *Star Trek - The Next Generation* introduced a wide audience to the concept of a room, where people can freely interact with artificially cre-

ated objects in a fully accessible environment [66]. Even though it's fictitious, the so called *Holodeck* is probably the most widely known representation of the concept of virtual reality. Many people all over the world strive to make this futuristic dream become reality. However, the scientific foundation has already been laid 30 years earlier.
As already mentioned in section 2.1 the actual idea was already presented in 1965 by Ivan E. Sutherland [77]. It took only three years for Sutherland to realize a working prototype of his *ultimate display* [78]. This head-mounted display can be deemed the first working output device which makes use of the stereopsis of the human vision (see 2.1.1) and simultaneously changes the field of view based on the head orientation of the user. Despite its primitive depictions of simple wire frame models, the device is still be regarded as a revolutionary invention.
In 1969 the computer artist Myron Krueger developed a series of computer experiences which he termed *artificial reality* [41]. More precisely, he developed computer generated environments that responded to the people in it. His early projects *GLOWFLOW*, *METAPLAY* or *PSYCHIC SPACE* are simple progressions in his research which finally led to the development of his *VIDEOPLACE* technology. With this installation users were able to communicate with each other in a responsive computer generated environment, even though they were in separate rooms.
The actual term *virtual reality* or *VR* has only been coined in 1984 by Jaron Lanier, when he founded his company *Visual Programming Lab* or *VPL* for short [42]. His company's goal was to make VR experiences accessible for the general public. Over the years his company made major developments on the field of VR, most noticeable the *Dataglove*. Using six degrees of freedom, or *DOF* for short, this input device made it possible to interact with virtual objects in a digital environment using natural hand gestures such as gripping or pointing. Additionally, the device provided tactile feedback in the form of vibrations. The company also distributed an advanced HWD, called *EyePhone*. Even though the device was a huge improvement to Sutherland's prototype, its bulkiness, high price and incapability to generate more than five or six frames per second were too much of a deterrent to spark public interest in the VR technology.
The release of the *Nintendo Power Glove* for the *Nintendo Entertainment System* in 1989 was also of little avail. Although the gaming community was usually open to new gimmicks and the glove itself proved to be working without major restrictions, the lack of actual applications that justified the device, made the glove a commercial failure for the *Nintendo Corporation* [43]. Existing input controllers were not only cheaper, but also superior for playing existing games. Nevertheless, the device is still regarded as one of the most relevant milestones in VR and gaming history.
In the late 20th century, introducing HWDs to the gaming community also proved to be of no success. While Nintendo had to discontinue the production and sale of its portable 3D gaming console *Virtual Boy* after only a couple of month on the market, the *Sega Corpora-*

tions VR glasses could not even spark enough interest to come out of its prototype phase [43].
At the same time, new unique output and input devices had been presented. One of the most noteworthy input devices being *PHANTOM* [47], a personal haptic interface device that measures the user's finger tip position and responds with drag on the same finger tip. Introduced in 1994, this pen-like device, which was attached to a static mounting, was able to simulate resistance of virtual objects. While the device could be moved freely, the movement could be hampered or even blocked by appearing virtual objects. Equally important is the publication of the unique output system *CAVE* [19] in 1993. The *CAVE Automatic Virtual Environment* was able to track its user in a room that was further equipped with at least three projection screens. Based on the user's position and orientation the system provided a view on a virtual environment which could then be explored without the weight or restrictions of HWDs. Stereoscopic glasses even provided a three-dimensional perception of the images. Due to their high prices and their limited area of application, those systems were not appealing for a wider audience. Without commercial value, public interest in the field of VR began to fade.
More than a decade after the latest major milestones in VR history, a humble *Kickstarter* campaign succeeded in reigniting the public interest in everyday VR applications. *Occulus Rift* promised an affordable HWD that could provide cutting-edge graphics for an extensive area of application [21]. Many companies followed in providing solutions for affordable HWDs and thereby making it reasonable to refocus on developing and understanding the possibilities of virtual realities. Even a more advanced concept for a *CAVE* system has been developed. The proof-of-concept system called *RoomAlive*, developed by *Microsoft Research*, shows that the current technology is on a good way to someday achieve the dream of a real life *Star Trek Holodeck* [39]. With this current development, it stands to reason to research VR applications as a possible supplement for education purposes.
Relevant devices that have been mentioned in this chapter and further hardware will be more closely discussed in terms of suitability for a 3D puzzle in the following section 2.1.4.

2.1.3 Immersion in Virtual Reality Environments

On of the keywords that is often used to promote virtual reality systems is *immersion*. As described in section 2.1 there are different aspects that determine the degree of immersion. Accordingly, different systems that present a virtual environment can be classified [31]. If the users perception of reality is not altered, the system is described as *non-immersive*. This applies mostly to traditional displays that present the virtual environment as a two-dimensional depiction. This hardly allows mental immersion, as the user always stays fully aware of his surroundings. By using traditional input devices such as a keyboard, a mouse or a gaming controller a physical immersion into the virtual world is not possible. Contrary to

this, *fully immersive* systems completely permit the user to experience physical and mental immersion. The result is usually achieved by using HWDs and provide head-tracking to completely alter the users visual experience. This sensation is further enhanced by tracking the users position or using motion-based controllers that permit the use of hand gestures. At this point, one can already talk about a *deep immersive* system. As the user has to wear specific gear such as HWDs and controllers, these systems tend to be very exhausting at long term use. Additionally, many people experience motion sickness from the discrepancy between their actual physical movements and the movements that are conveyed by the HWD [30]. Both experiences strongly conflict with the immersion into the virtual world. Flawless solutions for these problems have yet to be found.
Some systems use a middle course between those two extremes. These systems are referred to as *semi-immersive*. 3D displays might provide mental immersion into a virtual world even if the input is performed by non-immersive devices. Such devices are further discussed in section 2.1.4.

2.1.4 Hardware

This subsection briefly discusses various VR input and output devices in terms of their suitability for the realization of a 3D VR puzzle. To reasonably narrow the selection only contemporary devices that are currently affordable and available are considered. If the presented application shall be applicable as an actual supplement for anatomy education, universities and schools must be able to provide the necessary setup without extensive efforts.

Output Devices

The most reasonable choice for an output device that presents the virtual environment of a three dimensional puzzle is a HWD. Not only does a HWD provide a more immersive experience, compared to stereoscopic displays such as the *zSpace*, it also is much more affordable in terms of costs and needed space, compared to projector based setups like the *CAVE*. Performance-wise, HWDs that are controlled through an actual PC are obviously superior to smartphone-based HWDs such as *Samsung Gear*, *Google Cardboard* or *Google Daydream View*. Smartphone-based HWDs, on the other hand, are very affordable, which would make it easily possible to provide each student with a VR device for his personal use, especially since most students nowadays already own a smartphone that is is capable of presenting simple VR experiences. Unfortunately, those devices provide only very limited options for user input, which is evidently a driving factor for the proposed application. The remaining devices of interest are the *HTC Vive* and the *Oculus Rift*, previously mentioned in section 2.1.2. While both devices are almost identical regarding their hardware specifications, the *HTC Vive* has a clear advantage, as of December 2016. The *Oculus Rift* provides a *seated* experience [21]. As the name suggests, this means that when wearing the HWD the user sits inside of the virtual reality environment and can look around himself in a 360° angle. It is possible to move the head closer to objects and even slightly peek around their corners. The *HTC Vive*, however, even provides a *room scale* experience. To be more precise, the user is actually able to walk around virtual objects in a given area, by being tracked by the provided hardware. If one imagines to be working with a body donor in real life that is situated on a table, it is reasonable to say that one might be able to walk around said table to get a better view at relevant details. This metaphor can only be fully realized in a *room scale* setup, as provided by the *HTC Vive*. This makes it the preferred output device for realization of the presented application. A more detailed explanation of the *HTC Vive* and its components is later given in section 4.1.

Input Devices

As the relevant output device in this thesis, the *HTC Vive* is shipped with a pair of fully tracked wireless motion controllers. These *HTC Vive Controllers* are the obvious choice for input devices for the presented application. As these controllers can be tracked by the accompanying hardware per default, they can also be represented in the virtual reality and are therefore visible for the user, even when he is wearing the HWD. Alternative direct input devices, such as the *PHANTOM* can not easily be carried around by the user, which would make the use of a *room scale* system invalid. In case of the *Leap Motion Controller*, it is possible to carry the device around by attaching it to the users HWD. This enables freehand interactions in a *room scale* system. Unfortunately, this entirely restricts the users interaction area to the area in front of his face. As soon as his hands, leave the tracking area of the device, they can no longer be tracked and interaction would not always be possible. As the *HTC Vive Controllers* are tracked entirely independent of the HWD, this would not be an issue. Traditional input devices, such as a mouse, a keyboard or a gaming controller, might be portable but provide almost no feasible possibilities to interact with a virtual environment that justify the use of a HWD. All those input devices could not be tracked by a *HTC Vive* setup without further ado and, hence, would not be visible for the user when wearing the HWD, which ultimately hampers the interaction with the device and the virtual environment. As a consequence, using the default *HTC Vive* controllers is the most logical choice for the presented application. A more thorough explanation of the controllers is given in section 4.1.

2.1.5 Interaction in VR

To enable an effective and pleasant human-computer-communication appropriate interaction techniques are inevitably necessary. The current state of the art allows very complex user interfaces that can be operated with input devices with many degrees of freedom. Depending on the area of application this leads to a variety of interaction possibilities, that each have their advantages and disadvantages. Initially, the following section gives a brief overview about traditional interaction concepts that are proven to be still relevant. It is further discussed which concepts are of interest for interaction in a virtual reality environment. Moreover, relevant guidelines for VR interfaces and their usability are presented.

Human-Computer Interaction

According to Doug A. Bowman and colleagues, four *universal tasks* need to be considered when talking about the problem of interaction with a virtual environment [7][9].

Navigation: This task can be split into actions that are required for pathfinding and the actual action that is required to move from one specific point to another.

Selection: As the name implies, this task consists of selecting or more specifically specifying, one or multiple virtual objects.

Manipulation: This task contains all actions that change properties of an object. Properties could for example be position, orientation or shape.

System Control: The final task describes commands or actions that affect variables or states of the system.

Navigation can be distinguished in *direct* and *indirect* variations. Direct navigation describes the complete control over the camera. The user's input is directly transferred into camera translation and rotation. His position is obviously represented through the camera. This is often referred to as first-person or egocentric view [27]. Contrary to this, the user's position can be represented through a 3D object which is influenced by the user's input. In this so-called third-person or exocentric view [27] the camera follows the translation of the controlled 3D object. If those basic approaches do not follow any constraints, the user can move freely along all three axes and rotate as he wishes, often called *Flyer navigation* [24]. Without restricted navigation possibilities, the user will easily run into certain problems. Reoccurring problems are, e.g., too great distances between the user and relevant relation points or a camera view that is upside down in relation to the surroundings. In some cases, in an infinite environment, the user may travel too far away from the virtual world, also called *end-of-world problem*, or enter big meshes by accident [24]. To solve these problems, some constrains are made. Often the user's movement possibilities are restricted along one axis. The user either cannot accelerate in a specific direction or not rotate around a defined axis. Navigations based on this constraint are often referred to as *Pedestrian navigations* [24]. In virtual 3D environments, this restricted axis is commonly the y-axis or the upward-vector, relative to the object representing the user's position. In a fully immersive VR environment it is obvious that the user experiences an egocentric view of his environment. In case of the application described in this thesis, the view follows the tracking information of the user's

HWD. As a consequence, his movement is only constrained by real-world factors, such as gravity or solid obstacles.

Selection is a key task in 3D environments where the interaction with objects is relevant. It can be realized through many approaches. One of the most natural and intuitive approaches is the so called *grabbing*. Just like in real life, objects can be directly grabbed and in this way be selected [23]. Especially applications where objects are presented in a close distance to the user benefit from this selection approach. An obvious downside of this approach is the selection of objects that are further away from the user. The *Go-Go Arm extension*-technique, famously known from cartoon detective *Inspector Gadget*. With help of this technique the users reach is artificially lengthened which can be envisioned like a stretch of the arm. With either linear or exponentially increasing speed the arm stretches faster, as soon as a certain threshold distance from the user is exceeded [10]. A more abstract approach is the so called *Raycasting*. This *Action-At-A-Distance* technique works by shooting a virtual ray, going from a tracked object that represents the input position of the user straight into the environment [49]. The ray's hit point then decides the selected object. To make the selection of smaller objects that are getting even smaller if they are further away, as explained in section 2.1.1, the ray is often extended into a cone [57]. To more clearly identify the selection, objects that are hit with a ray are sometimes temporarily scaled bigger or temporarily tinted with a color. The most relevant issue with raycasting approaches is the selection of objects that are partially or completely hidden by others. A feasible extension that tackles this problem is the so called *Bending Ray* [9]. By using Bézier curves a bent ray is created that makes it possible to reach around masking objects, to a certain degree. From an egocentric point of view, *Grabbing* seems to be the most intuitive selection approach. However, the exocentric approach of *Raycasting* often turns out to be more precise [23]. A combination of both approaches appears to be the logical conclusion. The *HOMER* technique represents such a combination [10]. The idea of *Hand-centered Object Manipulation Extending Ray-casting* is already used in many applications. By using a regular *Raycasting* technique a selected objected is temporarily brought into the users near range, independent of its actual distance. The object can then be interacted with by using egocentric techniques, such as *Grabbing*. This interaction usually leads to the manipulation of the object. As described in section 2.1.4 the presented VR application uses motion controllers that are being fully tracked in a *room scale* environment. This makes the selection method of *Grabbing* an obvious choice to interact with the virtual objects. Unfortunately, the problem of having objects that are out of the user's reach stands. Accordingly, an additional *Raycasting* functionality should be considered. To provide a comfortable experience for the user it seems logical to provide functionality where *Grabbing* and *Raycasting* can be harmoniously used.

Manipulation can be split into three more specific tasks, *translation*, *rotation* and *scaling*

[54].
Translation describes the displacement of a 3D object. *Direct* or *indirect* methods of translation can be distinguished. With *direct* translation methods the movements of an actual 3D input device are immediately translated into alterations of the position of the virtual object. The interaction often feels intuitive and natural. A big disadvantage is the often limited space that is available for interaction and a noticeable lack of precision, especially with untrained users. The latter is often diminished by using a so-called *Snapping* approaches [88]. Objects are oriented according to a given grid or other objects. A threshold clearly determines which exact position the object is assigned to. Admittedly this limits the user's capability of positioning objects with nearly unlimited precision, but on the other hand provides results that are foreseeable and understandable for the user. *Indirect* translation methods manipulate the orientation of an object through widgets or control elements. Clickable arrows are often used to move objects into a shown direction. Numerical input is also often applied to provide an immensely precise positioning of virtual objects. Such methods are primarily designed to work with 2D input devices and do hardly use specific features of 3D input devices. They are further rated as less immersive. It stands to reason that a virtual reality application focuses mainly on
Rotation is an indispensable trait 3D object manipulation. To fully explore a virtual object, it is necessary to view it from every angle. As intuitive as this idea might seem, approaches for rotation are often not intuitive. As with translation methods, *indirect* rotation methods rely on widgets or other control elements. Usually the user first selects a rotation axis and then indicates the amount of rotation. Again, such methods are less useful for virtual reality applications, which moves the focus *direct* rotation methods. Methods where the orientation directly follows the gyration of the input device are commonly more easy to learn and more naturally to apply [40]. One such method is the so called *Mesh-Grab*. Using a raycasting, this approach captures a point on the surface of a mesh and then creates a connection similar to a ball-point joint. The distance between the the origin of the ray and its hit-point then keeps steady. This causes the object to behave like being skewered on a stick. Another method is the *Arcball-3D*. Here, a preferably small sphere is generated around the object of interest, which servers as the actual interaction element. This sphere is rotated around its center, which is directly applied to the enclosed object. The *Scaled-HOMER* approach is an extended version of the already mentioned *HOMER* concept [86]. By additionally considering velocity-scaling of the input device, the manipulation of the object is accordingly affected.
The *Scaling* of an object refers to the change of its dimensions. As there are no input-devices that can actually be manipulated with affect to their dimension, scaling is usually realized trough *indirect* methods. Often scaling operations are enabled by widgets [9]. After selecting an object, interactive elements appear that allow to manipulate the different dimensions of said object. Another approach is to select a point on the surface of the object by *grabbing*

and translating the following position change of the input device into a change ofthe objects dimensions. Some applications make use of implicit scaling. The dimensions of the virtual object are not actually affected, only the distance to the user is changed, using *translation* concepts. This leads to a perceived change of the objects size. In a virtual reality environment it is plausible to utilize methods that are implicit as well as concepts that actually chance the size of an object.

Interfaces

This interface where interactions between humans and the computer application occur is also referred to as a user interface, or UI for short. Traditionally, human-computer interaction tasks have been realized by using *Windows*, *Icons*, *Menus* and *Pointers*, for short *WIMP* [43]. Originally developed for 2D applications, it is still a vital component of many applications that solely rely on a 3D environment. By primarily using mouse and keyboard the concept makes most applications accessible and comprehensible for a wide audience. Unfortunately, the widespread concept does not stand up to the demands that come from continually advancing 3D applications [83]. For instance, the manipulation of a virtual object with six DOFs, can hardly be represented by a customary computer mouse, that can only be moved on a two dimensional plane. It is further impossible to perform rotations of a virtual object, with an input device that usually does not come with any rotation sensors at all. Complex combinations of mouse and keyboard inputs are necessary to solve such apparently simple tasks. In the past, this led to a disregard of alternative input methods for a lot of applications, even though alternatives like speech input or eye tracking are potentially promising for specific areas of interest. Alternative input methods like that are usually referred to as *Post-WIMP* interfaces [83]. Despite some interfaces still being not fully developed or less intuitive, most of them prove to be less cognitively straining after an initially steep learning curve. Following this, many alternative input devices have been spread into all different areas of application. With regards to VR the *HTC Vive* provides such methods, which will be further discussed in section 4.1.

Many desktop applications present their UI in a *non-diegetic* way [34]. This means that the interface is not part of the virtual environment, or better said, it is rendered over it. Interface elements that are part of the virtual environment, more specifically, are spacially present in it, are called *diegetic*. UIs for 3D applications often consist of both *diegetic* and *non-diegetic* components. For example, widgets that are placed around a virtual object are considered *diegetic* while simple UI buttons are usually *non-diegetic*. For non-immersive systems, *non-diegetic* user interface components are usually considered to be easier to interact with while *diegetic* components are said to heighten the degree of immersion [34].

In a fully immersive VR environment, it is basically impossible to interact with *non-diegetic*

interface elements because the input devices are spacially present in the virtual world [69]. Elements that the user actively needs to interact with must therefore be present in the virtual world as well. In such a scenario *non-diegetic* interface elements can only be used to present text or pictures. *Diegetic* user interfaces are often simply realized by putting the interface components in the virtual world as virtual objects.
For applications that are utilised healthcare environments it is common to use bright colors for interface elements. Since whitish tones are often linked with feelings of peace and cleanliness, they are especially common [85]. Blueish tones are often referred to as calming, so they are also frequently applied. Following this, it stands to reason to use whitish and blueish tones in an application that focuses on medical education

Usability

When developing human-computer interactions *Usability* is an indispensable factor. According to the Oxford dictionary the term describes *the quality of being easy to use* [74]. What appears trivial on first glance has already been proven to be quite a challenge for many developers [50]. Rolf Molich and Jakob Nielsen elaborated that any system that is designed for people to use always needs to deal with three issues. It has to be easy to learn and remember, effective and pleasant to use [50]. This has been adopted into the three required keywords that need to be fulfilled by every human-computer-interface – *Effectiveness*, *Efficiency* and *Satisfaction* [35]. In 1995, Nielsen compiled ten heuristics that serve those requirements [52]. Briefly worded, the ten general principles that need to be considered to create a user interface are *Visibility of system status*, *Match between system and the real world*, *User control and freedom*, *Consistency and standards*, *Error prevention*, *Recognition rather than recall*, *Flexibility and efficiency of use*, *Aesthetic and minimalist design*, *Help users recognize, diagnose, and recover from errors* and *Help and documentation*.

2.2 Anatomical Teaching

Before discussing the specific traits of anatomical learning, a more fundamental understanding the term of learning is necessary. The Oxford Dictionary defines it as

> "the process of acquisition of knowledge or skills through study, experience, or being taught."[74]

Dave Meier divides this process into four indispensable phases that all need to be covered until an information or skill can be considered *learned* [48]. These phases are:

Preparation: The first phase consists of arousing interest in a new skill or new knowledge. This interest can either be sparked by external stimuli or the realization that something new can be learned as current knowledge is not sufficient.

Presentation: This phase covers the encountering of the new knowledge or skill. All necessary and relevant details need to be provided and thoroughly examined. It is crucial to understand that this phase takes a different amount of time for each individual and might need different strategies for different learners.

Practice: After encountering the new knowledge or skill it is necessary to integrate it. This is usually realized by repeating and rehearsing it without the aids that were used in the *presentation* phase. The previous phase and this are often alternated multiple times.

Performance: The final phase is the application of the new knowledge or skill into a new environment. Even though the situation might be different from the *presentation* or *practice* phase, the learner is able to apply his knowledge.

In every educational environment these phases are covered and each student needs to go through them over time. As for anatomy education, it usually takes years for some skills or knowledge to reach the final phase.
Anatomical knowledge is the basic prerequisite for understanding clinical problems and therefore a fundamental component of medical education. Medical students need to learn many facts to establish a comprehensive understanding for the human body and its functionality [76]. The main fields that need to be covered during the study of anatomy are [11]:

Gross anatomy: The focus is on the structure and positioning of organs.
Histology: This field describes the microscopic study of cells and tissues.
Embryology: The formation and early development of the foetus are the main focus.
Neuroanatomy: This field covers the brain, spinal cord and peripheral nervous system.

As the most fundamental field, gross anatomy will be the main focus of this thesis.
The information that need to be taught have been classified by Cornelius Rosse into the *Spacial Domain* and the *Symbolic Domain* [67]. The *Spacial Domain* covers the shape, size, texture and subdivisions of anatomic structures. It further describes internal and external morphological features. Contrary to this, the *Symbolic Domain* contains the names and verbal descriptions of anatomic entities.
The *symbolic* knowledge is traditionally acquired trough lectures, discussions and text books. These methods have been proven successful on a long term basis. Books that cover such information can be classified into two categories. On the one hand, there are text books that describe names and functions of anatomical structures, generally only in text. On the other hand, there are anatomy atlases that clearly depict names and spacial relations of medical structures. These atlases rely either on actual photographs or more commonly on simplified depictions that are easier to understand [22]. Pictures are labelled, in order to link the spacial and symbolic domain. Unfortunately, 2D graphics can not flawlessly represent 3D structures. This requires a lot of visual thinking from the student, who then needs to view the same structures from as many angles as possible. This often demands multiple books

and tedious research. This effort and the monotonous atmosphere that is created by solely working with books often lessens the effectiveness of valuable learning sessions [2].
Traditionally, most *spacial* knowledge is acquired trough cadaver dissections [61]. This method is received as one of the most powerful teaching approach to communicate all the necessary details and information [22]. A variation of this method is the so called *prosection*, where already dissected specimen are examined. These last longer through plastination, a way of preserving organic tissue. These methods are very resource costly. Not only is the process of a dissection or prosection very expensive and takes a lot of time, it further needs appropriately trained staff. New body donors are always needed, as the different processes can only be repeated a very limited amount of time. Quite often, this raises ethical concerns, which are further strengthened by the fact that medical students are exposed to potentially harmful substances such as formalin fumes [22]. During a section, some areas are not accessible, so that other representations are required. Physical models are often used to represent said areas and most structures in general. These models are often very simplified or idealized and do not always represent the actual scale of the structure.
In the real world, text books and physical models used to be the most commonly used tools to gain the necessary *symbolic* and *spacial* knowledge, especially in self-directed study sessions [17]. All methods come with their individual advantages and disadvantages. A single teaching tool that fully meets all requirements of anatomy education and can be applied in an anatomy curriculum remains to be found [22, 38]. Since the entry of computer technology into everyday life, the development of such a computer based tool appears promising.

2.2.1 Computer-Aided Learning

Over the years a vast amount of computer approaches has been introduced with the goal to improve and supplement traditional learning environments. *Computer-aided Learning* Applications, or *CAL* for short, range from widespread medical databases [72] to interactive 3D anatomy atlases [82]. With the possibility to generate virtual 3D meshes from 2D scan data [44] or even scan 3D data directly from actual human bodies makes software applications that present such data very valuable for educational purposes. Contrary to many text books or physical models, such software does not rely on simplification, abstraction or the imaginative power of the user. The data can be presented very lifelike. It must further be noted that digital data removes most of the disadvantages that are present in cadaver dissections. Contrary to well preserved human bodies, the availability of digital models is neither limited nor decreasing [51]. Furthermore, there is no risk of accidentally destroying anatomical structures that are relevant for the student. Besides, dissections are usually performed in groups of students in a restricted time-frame [18]. This does not only devalue the process for each individual student, but also carries the risk of errors in the process. As the human body structures are not labelled in real life, students need to stay fully focused

during the whole dissection to avoid confusions. Some steps are irreproducible which makes it impossible to correctly review every step of the dissection [67]. Already 50 years ago it has been proposed to realize computer based applications that emulate real life dissections and thereby removing said drawbacks through the potential computers [73].
The pioneer in three-dimensional anatomy teaching is the highly acclaimed *VOXEL-MAN* [32], initially presented in 1986. Based on the *Visible Human* [1] dataset the application presents the human anatomy in an interactive three-dimensional environment. *Spacial* information can be gathered by inspecting every detail of the model from different angles. It is further possible to remove parts of the model which is similar to a real life section. *Symbolic* information is provided by labels and detailed descriptions of each structure. Today *VOXEL-MAN* is still further developed and regarded as one of the most comprehensive virtual anatomy atlases of our time.
Approaches like *VOXEL-MAN* are oriented towards an atlas metaphor. A similar example is the *Zoom Illustrator*, which presents an extension to fisheye views to explore three dimensional anatomy models [59]. The software provides a close relationship between images and associated text by influencing the appearance of each other with respective interactions.
Despite *VOXEL-MAN* or *Zoom Illustrator*, many web-based tools, like *OsteoScope*[1] have already been successfully introduced to serve as supplements in anatomy education [16, 26, 37][16]. Applications like that can be used by potentially large numbers of learners worldwide, at any time, either in private or in classrooms. Especially tools that are based on WebGL, like the *LiverAnatomyExplorer* introduced by Bernhard Preim and colleagues, are suitable for a wide audience as they even present high-quality 3D renderings on portable devices such as smartphones or tablets [6]. This provides students with the possibility to easily encounter anatomical data in their every day life without being restricted to any educational facilities or the need of additional devices such as books or laptops. Students and teachers are generally open-minded towards *CAL* approaches. Unfortunately the experience is often diminished by poor interface design that distracts from the actual learning process [33].
In 1999 it has already been shown that students would further appreciate to have more freedom in interacting with 3D anatomy models [56]. Especially assembling and disassembling of anatomical models by themselves is regarded as desired [64]. In 2000, this led to the introduction of a 3D puzzle metaphor for learning spacial relations of anatomical structures [65]. Adding this gaming component into an educational environment does indeed improve the user's understanding of spacial relations from 3D illustrations [63]. This work by Felix Ritter and colleagues lays the foundation for the thesis at hand, as presented in Chapter 3. Most of the medical applications use 2D input- and output devices, which is significant obstacle when examining 3D structures. This makes it necessary to look into more immersive systems that provide more natural and intuitive interfaces to the virtual environment. In

[1]http://http://taxonstudios.com/labs/osteoscope/ (Accessed: 29.05.2017)

this context, as described in section 2.1.2, it is promising to look further into CAL applications that make use of VR technologies to further remove the disadvantages that came with former applications.

2.2.2 Learning in Virtual Reality

Already in 1990 William Bricken describes the potential that VR applications bear for education [12]. Teachers want to provide an environment that transfer information trough a variety of stimuli [16]. They further want to control the sequence of information and the environment these information are presented in. This replaces the commonly used desktop metaphor with a world metaphor [60]. The higher degree of immersion, which ultimately leads to a closer encounter with relevant information, is an interesting aspect for many students and educators alike [28]. It is additionally proven that students are generally more excited and motivated to learn with contemporary technologies [5, 20, 60]. The excitement that arises from working with a fun new VR interfaces makes learning itself more interesting for many students [33].

Positive results that come from learning in VR environments have already been presented in 1993 [62]. It is stated that the knowledge gained in virtual reality environments can successfully be transferred into real world scenarios. Since then, VR technologies have already been integrated in educational applications with great success [33]. Fully Applications like *Cyber Science 3D*[2], *Anatomyou*[3] or *SpectoVR*[4] are positively recognized as being naturally and intuitively accessible . Compared to non-immersive systems, the cognitive effort of interacting with such system is significantly reduced which lets the user fully lay focus on the learning scenario [31].

Most commonly, VR systems that are used for education use free-choice learning and discovery [81]. Based on a museum metaphor the user is encouraged to explore the virtual environment on his own. On the one hand this enables the user to learn at his own speed and put focus on details that are of special interest to him, but on the other hand there is the possibility that the user misses crucial information or wastes his time by unproductively strolling through the virtual world. The learning outcome is often unpredictable which is not desirable for education methods. A potential solution for this, is to present the user with certain task [68]. This puts the user's focus on the subject of interest without reducing his involvement into the VR experience [36].

Even though VR systems provide distinct benefits for educational purposes, they are still hard to integrate into everyday classroom settings. The necessary hardware is still expensive and requires a considerable amount of space, especially if more than one setup needs to be

[2]http://cyberscience3d.com/ (Accessed: 29.05.2017)
[3]http://http://anatomyou.com (Accessed: 29.05.2017)
[4]http://http://http://www.diffuse.ch/#spectovive (Accessed: 29.05.2017)

provided for an educational institution [33, 81]. As describes in section 2.1.3, the technology might not yet be suitable for every student, as it might induce nausea and exhaustion while using, especially when a lot of actual physical movement is demanded. It is obvious that these factors can negatively impact learning experiences and therefore need to be considered when developing educational VR applications.

3 A Fully Immersive VR Approach for Interactive Learning

In 2000, Felix Riter and colleagues introduced a 3D puzzle metaphor for learning spacial relations of anatomical structures [65]. They showed that adding this gaming component into educational environment makes spacial relations more easy to understand and even improves the user's understanding of spacial relations from 3D illustrations [63]. Their work serves as the foundation for this thesis. As described in section 2.1.2, applying VR technologies into educational environments is now promising than ever as the necessary hardware became affordable and interesting for the public.

The proposed concept for an educational scenario will be build on Ritter and colleagues' 3D puzzle metaphor and transfer their idea into a fully immersive *Virtual Reality* environment with focus on anatomical structures. A semi-immersive realization of his concept has already been done in 2008, but unfortunately an informative evaluation was not given [68]. In engineering environments, similar concepts have already been introduced [36] and its been proven that such approaches are at no disadvantage to traditional methods and it is promising to pursue further research [75].

Basically, in such an environment the user shall be presented with a scattered 3D anatomy model and then solve the task of correctly putting all pieces together, equivalent to a real world jigsaw puzzle. While doing so, learning the names and spacial relations of each structure and thereby extending anatomy knowledge play the most important part.

This chapter initially describes the most important stakeholders for such an educational scenario and the thereby arising requirements for the virtual environment and interaction approaches. Following this, the adaptation of these requirements into a thorough concept is presented. Particular attention is paid to feedback features that support the user in solving the puzzle.

3.1 Goal of the Approach

This thesis addresses the concept of an educational scenario in form of a 3D puzzle in a fully immersive VR environment. The most obvious area of application is the education of medical students that need to learn names and spacial relations of anatomical structures.

It will be almost exclusively applied in teaching facilities. The target audience therefore consists primarily of medical students that do not have advanced anatomical knowledge. As the application will provide proper possibilities to review information, it is also certainly interesting for advanced students or even medical practitioners that want to refresh their knowledge. Another relevant stakeholder for the application are the educational facilities. As the concept includes a testing scenario, teachers could apply the application in exams, which could possibly create more valuable examination procedures. The application and its required hardware will also most definitely have a prestigious effect for the facility. As *Virtual Reality* systems are still considered novel by many people, the educational facility might earn a positive reputation for using contemporary technologies. If presented at open house events at the facility, the application and the VR system might arouse interest in the subject of anatomy or medicine in general, which might lead to an increasing number of matriculations.
The necessary requirements for the educational scenario and the application prototype will be presented in the following section.

3.1.1 Requirements Analysis

Before a thorough concept can be elaborated it is necessary to determine relevant requirements that need to be considered. First of all general prerequisites to the virtual environment can be considered.
The most positive user experience is induced when a software application is running in *real time*. Desktop applications that run with 15 frames per second, or 15 FPS for short, are generally considered as such [3]. For VR applications this FPS value rises to 50 FPS that are needed to ensure a *real time* experience and prevents potential nausea to a certain degree [46]. It is further required that the system regularly provides *feedback about its activity* [52]. This improves the reception of the real time functionality and also assures the user that the system is running and responding to his input. As the scenario shall take place in a *room scale* setup, it is essential that environment gives the user a feeling of *assurance*. He shall not be afraid of accidentally walking into real world walls or obstacles while wearing the HWD. The interaction with the virtual world shall be as *natural* and *intuitive* as possible [7, 9]. Applications that are more easy to learn and not demanding are generally adopted more positively by students and teachers alike [51]. In this context it must still be considered to use the possibilities of a computer based application to a meaningful extend.
A higher degree of *immersion* leads to a more pleasant user experience and often leads to an improvement of his performance. The virtual world shall be presented as a closed system, that allows the user to fully focus on his task and not be distracted by outside influences or distractions.
It is especially relevant to *consider the target audience* and their needs. As primarily medical

students are using the system, the application needs to consider their terminology and meet their visualization expectations. Teachers need to be able to add their own 3D models to the application in an uncomplicated manner. When talking about the target audience it is also necessary to discuss a certain degree of *accessibility*. The application shall be unrestrictedly usable for both right-handed and left-handed people [4]. For the regarded target audience it is valid to expect users with two functioning upper limps, as medical procedures mostly need to be done with both hands. Furthermore, students with color vision deficiency need to be taken into consideration. Even though the *room scale* setup allows the user to walk around, is must still be possible to operate the application in a seated position to minimize potential exhaustion effects.

The interaction techniques need to consider the *Principles for the Design of Performance-oriented Interaction Techniques* [7]. Virtual objects need to be *globally selectable*. More precisely, the user must be able to select virtual object within his reach as well as objects that are far away from him. The selection, the manipulation and the connecting of the virtual 3D puzzle pieces needs to be especially *precise*. It is further necessary to ensure that the 3D puzzle always has a *correct* solution that is attainable by the user.

To support the user in fully experiencing the scenario, it is relevant to help him with the most essential task of the application – the solving of the 3D puzzle. As he will initially not know how to perfectly put all the pieces together, the application must *provide features that help him* with this task. In the course of this prototype a number of feature solutions shall be presented and evaluated to determine which approach is most suitable to support the student.

Ultimately, it is required that the conceived educational scenario covers all of Dave Meier's *Phases of Learning*, as described in section 2.2. The application must somehow arouse interest in anatomical structures and then present all the necessary information to the user. It shall further be possible to repeat and rehearse the acquired knowledge and even apply it into a new context.

Taken all together the following requirements must be met by the educational scenario and the created concept:

Real Time: The application must run at least 50 FPS.

Activity Feedback: Every action shall give an according response or feedback to the user.

Assurance: The system must inform the user about his orientation in his real world environment.

Intuitive and Natural: Every interaction must feel natural and be easy to learn. It must further be straightforward to operate the application.

Immersion: The virtual world is credible and consistent. No unnecessary stimuli shall distract the user from the experience.

Consider Target Audience: The concept must adapt to the needs of the target audience.

Accessibility: To a meaningful degree the concept must consider potential limitations of the user.

Global Selection: It must be easily possible to select and interact with virtual objects independently from their distance to the user.

Precision: A preferably high precision is provided for selecting and manipulating virtual objects. The connecting of virtual puzzle pieces is as precise as possible as well.

Correctness: It is ensured that the presented puzzle task is solvable by the user and that the solved state of the puzzle is identical to the actual intended solution.

Help Features: The application provides multiple features that help the user with putting virtual puzzle pieces together.

Phases of Learning: The educational scenario covers Dave Meier's four *Phases of Learning* in a meaningful way.

3.2 Creating a Solution Space

Analogous to a real-life jigsaw puzzle, a three-dimensional virtual reality puzzle has a distinct given solution. Usually referred to as *solution space* [29, 84], a complete set of system states is needed, that describe all possible final states that the user can achieve to properly finish the task given in this education scenario. In case of a jigsaw puzzle, this solution space simply consists of all pieces being in correct alignment with each other, according to the completely solved puzzle. For the prototype, it is therefore necessary to provide the model, that shall function as a puzzle, in its solved state. If a users puts his own 3D models into a designated folder while using the filename extensions *.obj*, *.fbx* or *.prefab* the system can load these models at runtime. The extension *.prefab* is unique to the *Unity* framework which is further described in Section 4.2. The application will then prepare the given model accordingly to make it interactable and solvable. This implies that the user can only put the pieces together as it is defined by the given model. This complies with the jigsaw-puzzle metaphor, in which each piece usually needs to be placed according to the picture on the box.

Since the three-dimensional space can be fully used in this prototype, it is not sufficient to just put the pieces into the position that was defined by the given model, it is further possible to solve the puzzle anywhere in the environment. Hence, the solution space for this three-dimensional virtual reality puzzle consists of the correct alignment of each piece relatively to all the other pieces.

An unambiguous approach is needed, to permanently check the correctness of the solution. Since the prototype is running as a real-time application, where all changes to the system happen at runtime, not only correctness is a crucial factor but also performance. This is not only necessary for checking the progress of the whole puzzle, but also for checking if two pieces can be joined when they are aligned correctly, further referred to as *being snappable*. During the implementation of the prototype, two different approaches for this problem have been tested; one using calculations for relative distances and angles and the other one using a hierarchy parenting system. This chapter describes these approaches more thoroughly.

3.2.1 Relative Alignment

The initial attempt to check if two pieces are snappable as well as if the whole puzzle has been completely solved, made use of the relative distance and the relative rotation between the pieces. Since the model is given to the system in a solved state, i.e., all pieces are correctly positioned and oriented, these values can be easily computed for all individual pieces. Each piece needs to store the relative position and rotation it initially has with all other pieces. When all pieces are aligned, so that their relative position and relative rotation is identical to the respectively stored values, the puzzle can be considered as solved. While the user is trying to solve the puzzle, it can be checked if the current relative position and rotation of the selected pieces correspond to the values that are saved as being correct. In terms of computation, the absolute length of the vector between the current position and the correct position needs to be zero, while the absolute angle between the the current rotation and the correct rotation needs to be zero as well . Since it would be impossible for a human user to align pieces with infinite precision, a threshold has been defined, that makes it possible to combine pieces, even if they are not perfectly aligned according to their computed values. Initially, this threshold was defined as 50 mm for the length of the vector between the current piece position and the correct piece position and 5° between current piece rotation and the correct piece rotation. These threshold values should have been empirical improved to adapt to the needs of actual users. Figure (see Figure 3.1a) illustrates how the positions (marked red) of the pieces need to be perfectly aligned to be regarded as *snappable*. Figure 3.1b illustrates the threshold area (marked blue). As long as the position (marked red) and the rotation of the right piece is within this area it is regarded *snappable*. The two slightly transparent pieces illustrate exemplary orientations that are also regarded as *snappable* due to their orientation within the threshold area.

While testing this approach, a significant error source became clear. Even if their position and rotation is correct according to the relative values, the pieces might still be aligned wrong. If one piece is put in the exact opposite position of it's correct relative position, it is still regarded as correct, even though it is clearly not. This can be attributed to the fact that it is necessary to work with absolute values when checking the distance, which is again required to make it possible to solve the puzzle without regards to the world space.

To solve this issue, not only the relative position and rotation between the pieces need to be checked but also their respective orientation in the world space. During the implementation process a simpler solution appeared, which made a further implementation of this approach

(a) (b)

Figure 3.1: An illustration of a perfect alignment and valid alignments within the threshold area.

obsolete.

3.2.2 Child Helper

To precisely describe the alignment of two pieces, a separate helper object that makes use of hierarchical structures can be introduced. When the model is loaded in its initially solved state, the simple helper object is created in the global position of the model. For each piece of the model, a copy of said helper object is then generated. This needs to happen without changing any of the helper objects orientation values, so that the copies are identical in every aspect. Afterwards, each copy is assigned to one puzzle piece respectively as a child. This hierarchical structure ensures that child helper object always follows its puzzle piece and still keeps its correct orientation to it.

When trying to fit pieces together it is no longer necessary to check the puzzle pieces itself, but their respective child helper objects. If the global orientation of those child helper objects is identical, then the pieces must obviously be in the alignment that has initially be considered as correct. To be more exact, if the absolute length of the distance vector between two child helper objects is zero and the absolute angle between their rotations is zero as well, their respective pieces are guaranteed to fit together. When two pieces are fit together, one child helper object can be removed to simplify further computations. The joined pieces are then basically considered as a single piece. The puzzle is solved as soon as all child helper objects orientations are identical and only one child helper object remains. Compared to the *relative alignment* approach, this solution potentially requires more memory to be realized. Nowadays, memory usage of this magnitude is usually not an issue. In return, this approach certainly requires less computing power, which makes it more performance efficient.

Identical to the *relative alignment* approach, a threshold is needed to make it possible for actual human users to fit the pieces together with a certain degree of precision. After the initial implementation of the approach, this threshold was defined as 50 mm for the length of the vector between the child helpers positions and 5° between their rotations (see Figure

3.1b). These threshold values were later empirical improved to adapt to the needs of actual users, as described in section 5.1.

3.3 Concept for an Educational Scenario

This section describes the proposed educational scenario in detail. As previously mentioned the main task of the scenario is to fit 3D puzzle pieces of anatomy models together, hereby learning the names of anatomical structures and their spacial relations. The educational scenario is proposed as a single application that contains different modes that again all cover the relevant phases of learning, as described in section 2.2. Furthermore, a virtual world needs to be designed that is appropriate for medical education. In addition, it is necessary to create features that improve the user's experience of the scenario.
The proposed application uses a *room scale* VR setup. This means that the user can navigate the environment simply by actually walking. The system is operated via two fully tracked motion controllers. The controllers provide buttons that are necessary for certain interactions. The relevant buttons are referred to as grab button, menu button, touch pad and grip button. An exemplary setup is presented more detailed in Section 4.1.
After selecting a model, the application prepares it to be usable as a puzzle. It further ensures that the puzzle is neither to small or to big for the user. This means that the model is scaled to fit into a 2 m by 2 m cube that is positioned so that the user does not stand in it when the application starts.

3.3.1 Selection

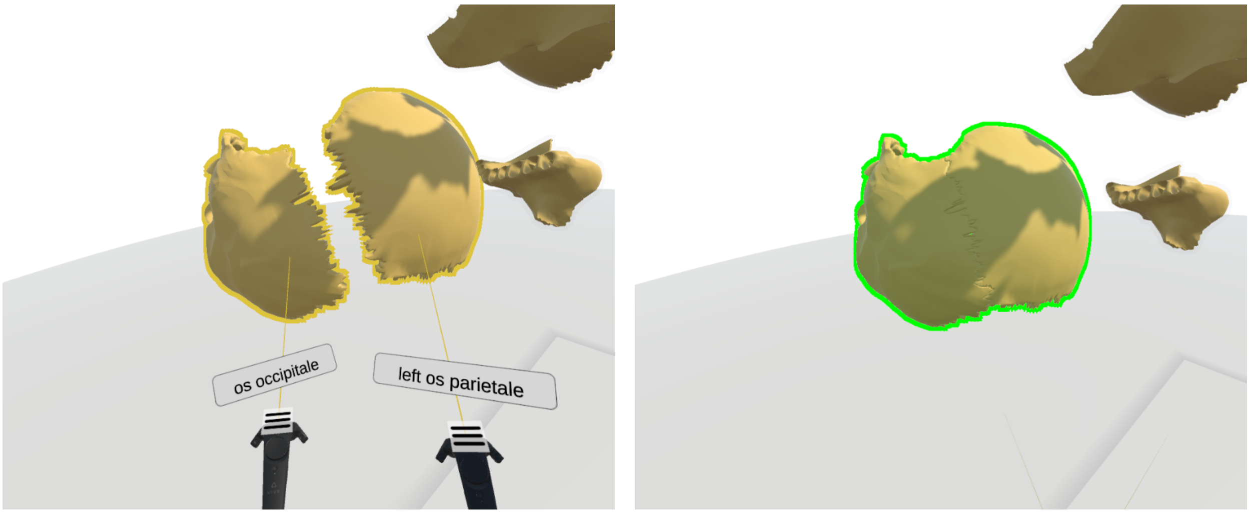
Within the virtual world the user can select puzzle pieces by either touching them with a controller or by pointing at them with a virtual laser that comes out of the front of the controllers. This virtual laser follows the basic principles of *Raycasting*. The laser has a semi-transparent orange color. This choice of color as been empirical determined to be most suitable as it is always clearly distinguishable from the environment and the puzzle pieces. A selected object is clearly highlighted to give a significant feedback about the selection (see Figure 3.2b). A semi-transparent outline with the same color as the virtual laser is rendered around a selected object. Additionally, the object's texture is slightly tinted with the same orange tone. As soon as the laser or the controller no longer touches the selected object, the highlighting is removed. If an object is selected with the laser, a label appears on the laser that shows the according name as defined in the loaded 3D model file. That way a student can easily learn and associate the names of different anatomical structures.

(a) (b)

Figure 3.2: Selecting and *grabbing* pieces either by touching or with the laser.

3.3.2 Manipulation

While being selected, pieces can be *grabbed* by holding down the grab button on the controller. As soon as a piece is *grabbed* the highlighting is removed (see Figure 3.2a). To maintain the association between a pieces shape and its name, the label with the name is also displayed when the piece is *grabbed*. The piece follows the respective controllers movements as long as its *grabbed*. This makes it easy to closely examine each individual piece and understand its significant traits. To avoid contradictions, a piece can only be *grabbed* by one controller at the same time and each controller can respectively only grab one piece at a time. It is a deliberate design choice to require a constant down holding of the grab button. On first glance it appears to be more easy to activate a *grab* with a click on the grab button and deactivate the grab with a second click. This would allow to more freely examine the piece as the controller can be more freely rotated in the users hand. Unfortunately, this behaviour could lead to the user accidentally dropping the controller and thereby damaging the equipment. In the virtual world it could also easily lead to confusions. The user might unintentionally grab a piece without realizing that he now has a piece appointed to his controller which makes selecting another piece impossible.
If the user can not reach a piece that is far away from him, he can get it closer to himself, either by double clicking the grab button while the piece is selected or by using the touch pad while the piece is already *grabbed*. *Grabbing* two pieces at the same time indicates an assembly attempt, or to be more precise, it indicates that the user wants to put these two pieces together. An assembly attempt only works in this situation. This choice was made to detect that the user wants to consciously assemble the pieces. In a real world scenario it is not uncommon to use both hands when assembling something, which makes this choice reasonable.



(a) (b)

Figure 3.3: The assembling of two pieces resulting in a connected group.

As described in Section 3.2.2 the two pieces need to be aligned within a certain threshold to each other. This threshold is further referred to as *snapping distance*. When two pieces are within this *snapping distance* they will be connected, as soon as one controller lets go of its piece. This process is further referred to as *snapping*. The *snapping* is based on a magnet metaphor. If one holds two real world magnets, A and B, close together and lets go of magnet B, it will attach itself to magnet A by force. The concept strives to recreate this effect.

After the *snapping* the regarding group of pieces is treated like a single piece. Structures that were correctly assembled like this are highlighted with a light green outline (see Figure 3.3). This outline is only visible as long as the group is neither *selected* or *grabbed*. It is possible to detach individual pieces from an assembled group by *grabbing* the whole group with one controller and then *grabbing* one of the connected pieces with the other controller. To give the user the possibility to more thoroughly examine the model, he can scale and rotate the model at will by holding down the grip buttons on both controllers. Moving the controllers closer to each other or further away from each other then results in an increase or decrease of the size of the whole model. It is not possible to rescale each piece individually, as this would certainly cause trouble with the assembling of the pieces. Additionally, the whole model can be freely rotate by rotating the controllers around each other. As of now, it is not possible to move the whole model like this. Early tests with the prototype suggested that this might significantly intensify feelings of nausea. The arising transformation can easily be reverted by clicking a button in the user menu, as described in section 3.3.9. These features clearly show the advantages of a virtual environment over a real world one. In a real world it is not easily possible to freely rotate a cadaver during a dissection and utterly impossible to actually enlarge it to get a better view at tiny structures.

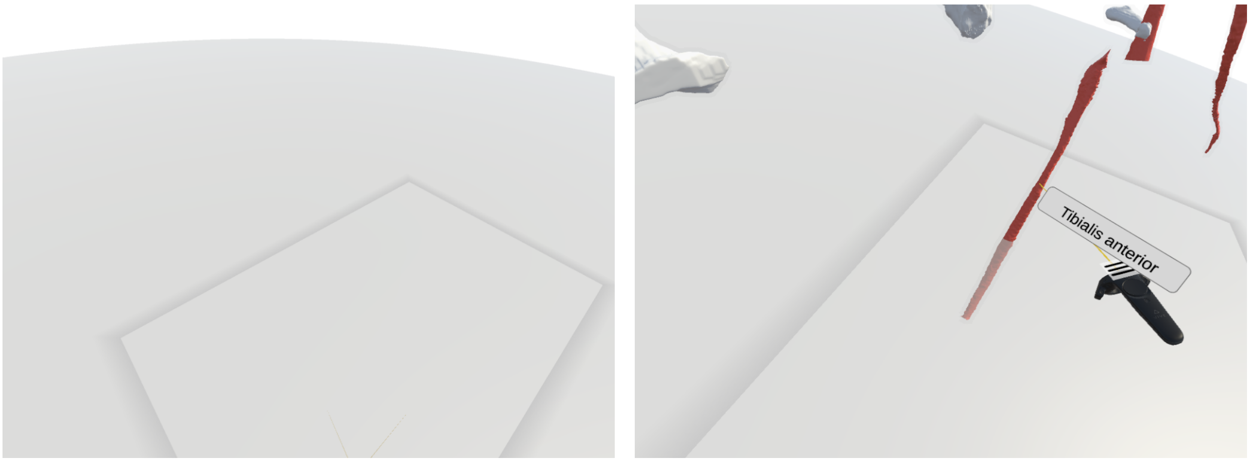
3.3.3 The Virtual Environment

The virtual environment needs to be designed to be in line with the requirements of the application. To achieve a better and more comfortable user performance, the environment should be somehow based on real world rooms [71]. This does not only improve the effect of immersion but also prevents a certain degree of discomfort and uneasiness. The user is mainly surrounded by a sky dome of a very light whitish color that can be perceived as white. As described in section 2.1.5, such bright colors are commonly found in healthcare environments. Another positive point is the clear contrast the bright color provides to the actual objects of interest in the environment – the anatomy model (see Figure 3.5b). Pieces are easily distinguishable from the environment. If the surroundings had textures, shading or animations applied, this might not always be the case. One of the most relevant aspects in many virtual environments is the ground floor. Not only does it serve as a crucial landmark for orientation [14], it can also illustrate how much space is available for the user to operate in. The ground floor is designed in a circular shape that is significantly bigger than the actual area the user can navigate in, which is defined by the setup of the VR system. This shall prevent that the user feels cramped in the virtual world. The ground floor features a slightly darker shade of grey than the sky to be clearly distinguishable. As presented in Section 2.1.1 the hereby created horizon line benefits how the user perceives his surroundings. By moving pieces around in the environment it can unintentionally happen that a piece is moved below ground level. To avoid that such a piece can no longer be found, the respective piece is presented semi-opaque instead of completely hidden. Obviously the ground floor object does not interfere with the selection of pieces in any way (see Figure 3.4b).

The environment is deliberately left empty. Decoration like tables or medical equipment might not only distract the user from the virtual anatomy model he is interested in, but also might cause confusion. It is essential for the user to be able to clearly identify each anatomical object. Following this, the environment refrains from showing object's shadows on the ground, as these might accidentally be mistaken for puzzle pieces. Only objects cast shadows on each other, to retain a sense of realism.

As the application provides a *room scale* experience, it is necessary to somehow communicate the boundaries of the real world surroundings to the user that he is able to move in. A subtle rectangle on the ground shows the approximate boundaries (see Figure 3.4a). In this way, it is essentially prevented that the user accidentally collides with real world obstacles while using the application.

The environment does not simulate gravitational effects on the puzzle pieces. Even though it might improve the sense of immersion, it will most likely be annoying for the user, if the pieces fall to the ground, as soon as he lets go of them. Not only would this mean that the user needs to bend down, every time he wants to pick up a piece, it would also heavily limit the possibility to arrange and assort pieces around him. For this reason, the



(a) (b)

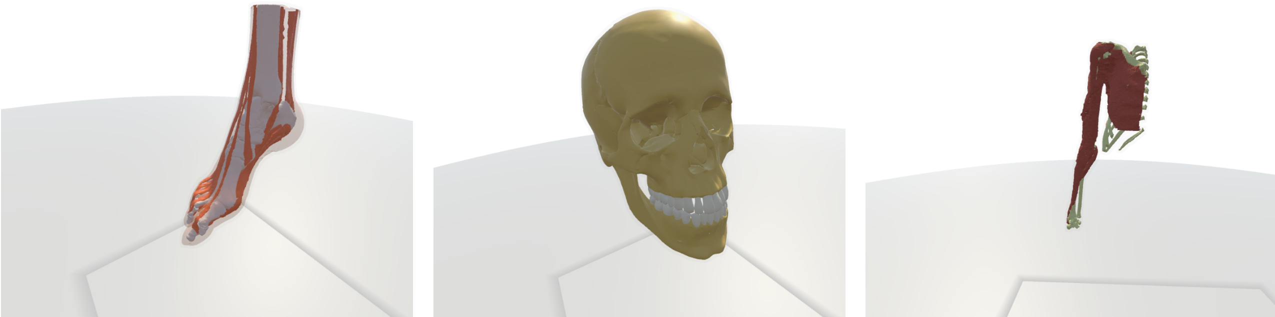
Figure 3.4: The rectangle that communicates the boundaries of the play are and a piece below ground level.

anatomical structures behave without any sense of inertia. If the user lets go of a piece, it stays in the exact same place. For the same reason the pieces do not collide with each other. Unfortunately, this can lead to two or more pieces overlapping each other, which can make it harder to identify individual pieces or even let some pieces be mistaken for others.

3.3.4 3D Models

For the prototype realization useful 3D models are required. As this thesis deals with an application that focuses on anatomical education, it is obvious to use appropriate 3D models of anatomical structures. Actual 3D scan data of a foot, a human skull and a shoulder section has been provided. As this data is basically a copy of the real world structures it is perfect for an educational scenario. The models were prepared so that each structure that is interesting for a student is present as a single piece which can then be used for the puzzle (see Figure 3.5). It is ensured that each structure is named correctly with their English term. The textures are attempted to be looking very lifelike by using high-resolution (1024x1024px) textures that are provided by *DOSCH DESIGN*[1]. Some structures are consciously textured in an abstract way that is more familiar for medical students. Arteries are tinted red, veins are blue and nerves are yellow.
To demonstrate the application to people that are not familiar with anatomic structures and to simply present the concept, more plain models were used (see Figure 3.6). Two models that only consist of four pieces each, a house and a circle, were especially useful for the implementation as it is very easy to spot two pieces that correctly fit together. Additionally, the model of a *Rubik's Cube* was used, as this is widely known by many people and an

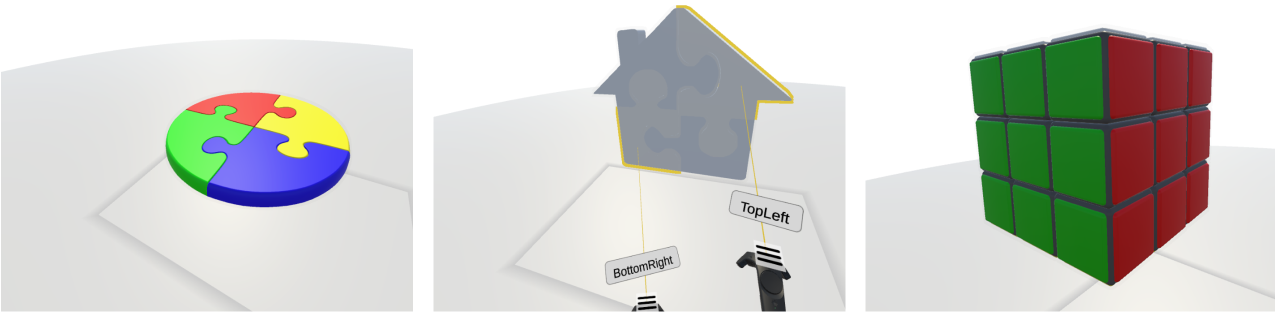
[1]https://www.doschdesign.com/produkte/textures/Medical_Visualization_V3.html (Accessed: 14.06.2017)



(a) (b) (c)

Figure 3.5: Plain models that are interesting for users that are familiar with anatomical structures.

interesting subject for puzzling.



(a) (b) (c)

Figure 3.6: Plain models that are interesting for users that are not familiar with anatomical structures.

More models can easily be added to the application, as long as they represent the object that has to be puzzled in a solved state. It is thereby essential to make sure that each structure is named correctly, as the application will only show the names according to the 3D model file.

3.3.5 Visual Feedback

The most important aspect of a jigsaw puzzle is to fit the pieces together. If a user knows that two pieces match with each other, he needs to be able to put them together. It can be frustrating and even cause confusion if this task turns out to be too cumbersome. Additionally, to the snapping threshold that has been described in Section 3.2.2, some feedback is needed that can help the user, if required. It is further necessary to provide some support if the user has no idea how the pieces fit together. The same feedback thus becomes an essential feature to initially learn the spacial relations of the anatomical structures. This makes it clear that the proposed feedback features need to approach two basic tasks. If the user knows how two pieces need to be aligned, but struggles to put them together due to his inexperience with the application, the system must assist him in a meaningful way that does not result in tedious trial and error attempts. Secondly, the system must provide a way for the user to easily learn the spacial relations of two specific objects, if he does not know them. Some feedback approaches will probably be more fitted for the first task, while other approaches might be more useful for the second task.

The most obvious way to provide the required feedback is as a visual stimuli. As part of the concept each feedback feature can be freely enabled or disabled by the user. It is completely up to him to choose which feedback he prefers. He can combine every feedback feature as he wishes and even use them all at once. To not unnecessarily distract from the virtual world, a feedback is only visible when the user is holding two pieces at the same time to indicate an assembly attempt.

Meaningful visuals feedback features are presented in the following sections. Some feedback features require the use of colors to illustrate information. To indicate that something is *correct* the concept applies a light green color. A dark red color is applied to indicate that something is *wrong*. The light and dark contrast makes it possible to distinguish these colors even for people with color vision deficiency [45]. Some features make use of a transition between those complementary hues. Unfortunately, the association of colors with certain information, especially red and green, is still very controversial. Regarding the fact that for the time being the presented concept only targets a limited audience, this choice of color is justifiable. In western culture and many computer applications a green color is still commonly associated with terms like *on* or *correct*. The color red is often regarded as the opposite and invokes terms like *off* or *incorrect* [87]. This specific color association is kept consistent for the whole concept.

Display the Values

A very straightforward approach for a visual feedback that conveys information about the difference of the orientation of two objects is to directly present this difference as values. For the user it is important to know the distance between the two pieces he is holding as well as the difference between their rotations. As long as the pieces are not in *snapping distance*, the values are tinted dark red (see figure 3.7a). This choice of color additionally highlights that the orientation of the pieces is not yet correct (see Figure 3.8a). For an optimal snapping it is necessary to bring both values to zero. Since the application provides a certain threshold, it is not necessary to be precise. As soon as the two pieces are *snappable*, the color of the values turn light green (see figure 3.7b). This color highlights that both pieces are oriented correctly (see Figure 3.8b).

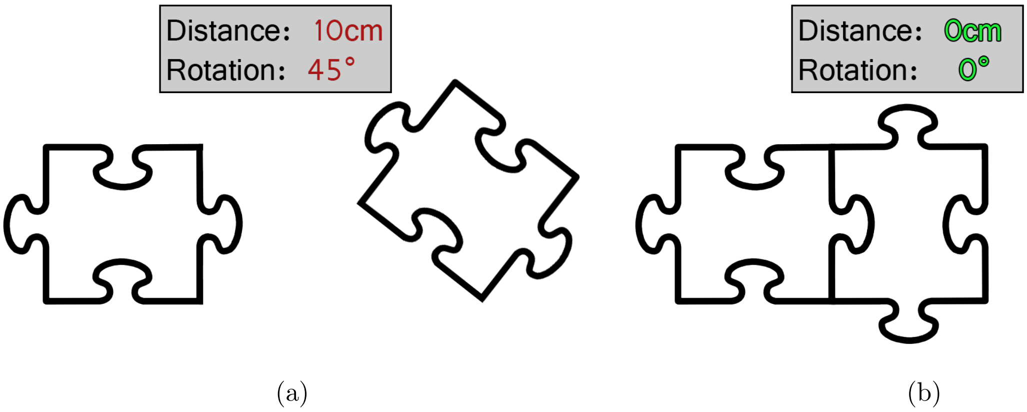
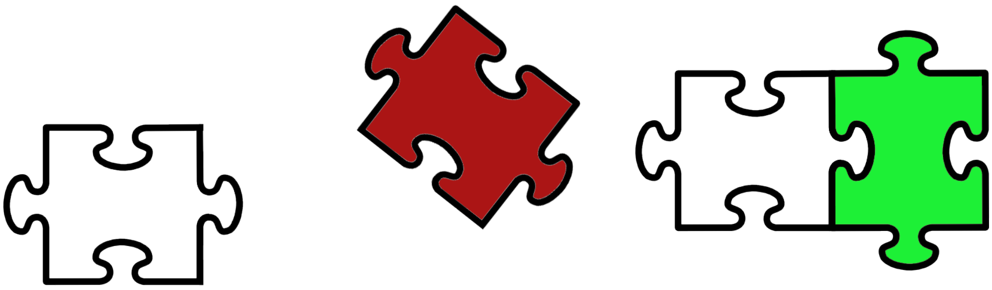


Figure 3.7: A simplified concept for a *Display* showing values.

Figure 3.8: Screenshots of the *Display* while the pieces are not *snappable* and *snappable*.

Tinting

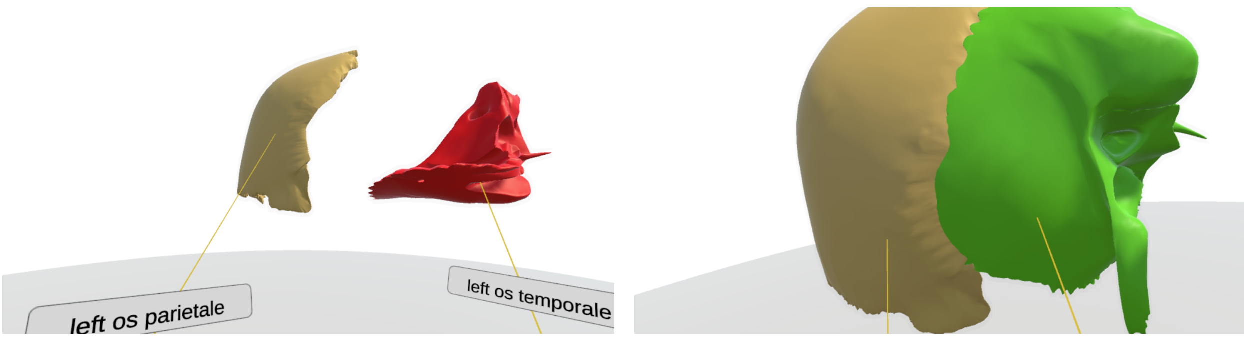
Similar to the *Display the Values* approach, this approach conveys the correctness of the current distance and rotation of the two pieces that the user wants to merge. Instead of directly presenting these values, they are coded into a color. As soon as the user holds two pieces, the piece he selected last is tinted with a color that signals said correctness. The tinting color changes from a dark red to a light green. This transition value is simply calculated by the average of the distance's and the rotation's correctness. As long as the tinting color is not light green, the pieces are not *snappable* (see figure 3.9).
The light green color is assigned to an average value of zero, while the dark red color is assigned to an average of 100. Average values in between are accordingly assigned with



(a) (b)

Figure 3.9: A simplified concept for *Tinting*.

respect to the red-green gradient (see Figure 3.10). If the distance between both piece's helper objects is equal or less than the *snapping threshold*, the distance value is zero. If it is larger than 500 mm, the value is 100. Equally, if the rotation difference between both pieces is less equal or less than the rotational *snapping threshold*, the rotation value is zero. Accordingly, it is 100 when the difference between both rotations in 180°.
Compared to the *Display the Values* approach, this approach has the clear advantage of not blocking the user's view with additional visual elements.

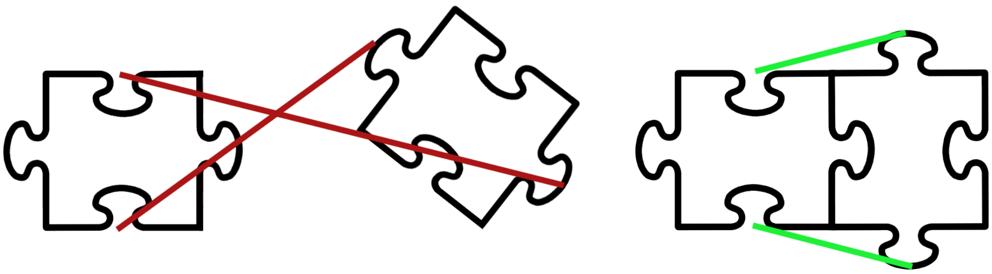


(a) (b)

Figure 3.10: Screenshots of the *Tinting* while the pieces are not *snappable* and *snappable*.

Elastic Strings

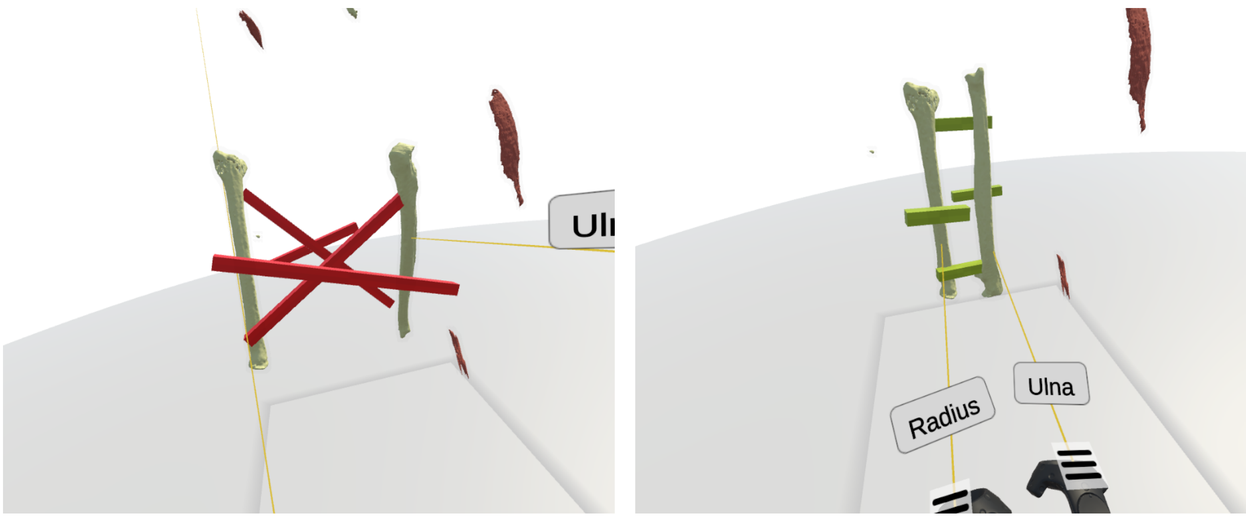
This feedback features separates the presentation distance and rotation differences. The rotation is represented through a rubber string metaphor. To realize that, each puzzle piece is assigned four anchors. These anchors are not actual objects, but rather values that are stored by the piece. Two anchors are assigned along the piece's local x-axis and two more along its y-axis. One of each follows the positive direction of the axis, the other follows the negative direction. Each pair of anchors has the same distance from the piece's visual center. This distance is determined by the object's mesh's most extreme point on the respective axis.



(a) (b)

Figure 3.11: A simplified concept for *elastic strings*.

When the user is holding two pieces, a cuboid is drawn between each anchor and its respective counterpart. Altogether, this creates four cuboids that resemble elastic *strings* that are attached to the pieces. These *strings* continuously adapt to the orientation of the pieces. The difference between the rotation of the pieces is now represented through these *strings*. If two or more *strings* are crossing each other, the rotation is not correct (see figure 3.11a). Using real world experience and imaginative power, the user can then rotate the pieces to *untangle* the *strings*.
The distance difference is represented similarly to the *Tinting* approach, but here the *strings* are tinted. This tinting is identical to the *Tinting* only without regards to the rotation values. A light green color indicates that the pieces are within *snapping* distance and a dark red color shows that they are more than 500 mm away (see Figure 3.11). Depending on the actual distance difference, the hue is transitioned accordingly (see Figure 3.12).



(a) (b)

Figure 3.12: Screenshots of the *strings* while the pieces are not *snappable* and *snappable*.

Ghost Copy

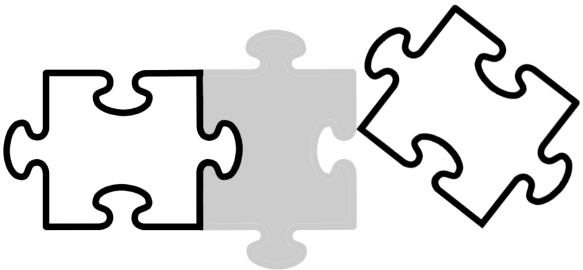


Figure 3.13: A simplified concept for a *Ghost Copy*.

In real-world jigsaw puzzles the user usually has a picture of the solved puzzle at hand. When he tries to figure out the orientation of a single piece, he can easily refer to the picture on the box. Unfortunately, such a 2D depiction is not helpful in a 3D environment. Showing a picture of the finished model will only provide vague information about the alignment of the pieces. As certain depth information is missing and such a depiction will most likely cause confusion as some parts are looking very different in 3D compared to their 2D representation. A lot of pieces would most likely occlude each other in such a depiction, which will not be of much help to the user. One could argue that an additional complete 3D model could serve as a feasible equivalent to the real-world picture on the box. Unfortunately, in a 3D environment a complete 3D model will not always show the structures of interest. The user might try to assemble two internal pieces, e.g. bones, that are covered by external structures, e.g. skin. The obvious solution would be to remove everything from the additional model except the regions of interest. This would lead to an additional rendering of the same two 3D models the user is currently holding, which can easily be regarded as an unnecessary obstruction of the user's view. At this point it is most reasonable to just present the solution to the user. More precisely, directly show a copy of one of the pieces aligned correctly with the other piece (see figure 3.13). The user can than just match the piece he is holding with the copy he is seeing. To make the copy more distinguishable from the actual pieces, it is rendered with a semitransparent dark grey material, which resembles a ghost, hence the name *Ghost Copy* [79].

3.3.6 Vibrotactile Feedback

As described in Section 2.1, sensory feedback can be seen as one of the key elements of virtual reality interaction [70]. An often overlooked part of sensory feedback is the haptic experience. This is not only useful to enhance the user's immersion into the virtual world, but also a possibility to communicate information,therefore, a reasonable addition to the features that help the user solving the puzzle. Vibrotactile systems, that are used to communicate

Figure 3.14: Screenshot of a *Ghost Copy*.

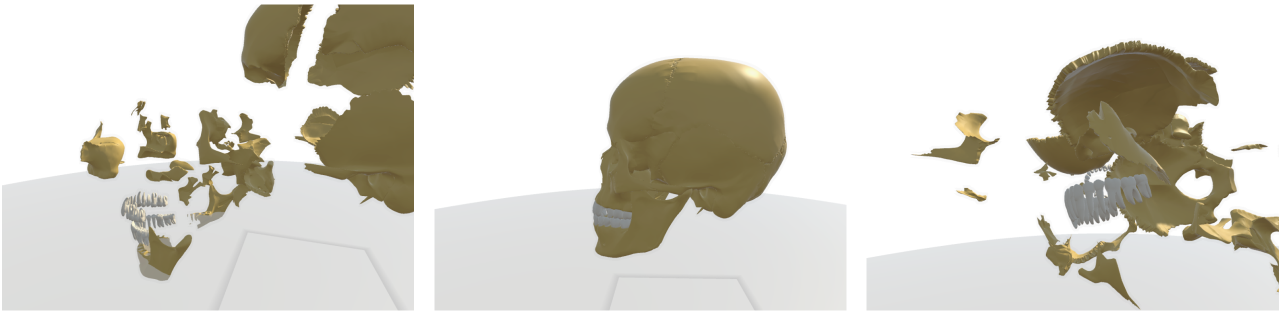
information have already been introduced in 2002 [15] and it has been shown that users are able to comprehend very fine vibrotactile stimuli [80]. Using the vibration feature of the *HTC Vive* controller, the prototype provides two different kinds of feedback to the user. If the user's attempt to unite two pieces is successful and the pieces snap together, the user experiences a distinctive, steady vibration for 0.5 s from both controllers. This helps the user to understand that is attempt was successful. Beforehand, while trying to align the pieces, the controllers vibrate lightly as soon as the pieces are snappable. The actual feedback is a loop of increasing vibrations, with each loop lasting for 0.3 s, starting with an intensity of zero going to an intensity that is determined by the correctness of the solution. More precisely, this pulsating vibration gets more intense, the more correct the user aligns the pieces. The maximal intensity of the vibration is only half of the intensity of the vibration that is used to communicate the successful snapping of pieces.
All values of the vibrotactile feedback have been empirical determined to be distinguishable and comfortable for the user.

3.3.7 Puzzle Modes

As described in Section 2.2 an education scenario needs to cover different *Phases of Learning*. To do so, the concept proposes three different modes that consider the needs of anatomical education. The first phase of learning, the *Preparation*, is covered by the fact that a medical student must have an interest in learning about human anatomy per se. Additionally, it can be argued that the use of an exciting technology as VR arouses interest in this application and the consequent educational scenario. The following sections cover the modes that are provided as part of the concept and highlights how they fulfill the needs of an anatomy student and the specific *Phases of Learning*. If none of these modes is running, the system is considered to be in its main stage. As described in Section 3.3.9, the different modes can here be accessed via the user menu.

Exploration Mode

Within this mode all information that can be gained in the proposed educational scenario is presented. This corresponds to the *Presentation* learning phase. When starting this mode, the application loads and prepares the selected model and then presents it to the user in a completed state (see Figure 3.15b). The user can then calmly examine and explore the anatomical models and its various structures. By *grabbing* individual pieces he can disassemble the model and reassemble structures at will.
Inspired by the *Explosion* view that is often applied in engineering environments [75], this mode also provides a similar *Scattering* feature. By clicking the according button in the user menu, the user will see an animation of the model *exploding*. To be more exact, all pieces will simultaneously move away from the visual center of the object over a certain amount of time. This is evidently helpful to identify spacial relations. Within the scope of this application a time of 2 s has been used. The *scattering* can be affected by two factors that are adjustable in the user menu. One factor is "Random Position" and the other is "Random Rotation". As the names suggest the first factor determines if the pieces are scattered randomly, or follow the exact path that goes from the models visual center trough each individual piece. The second factor determines if a random rotation is applied to the piece as it is *scattered*. If both factors are deactivated, the *Scattering* is identical to an *Explosion* animation that is commonly found in most computer-aided design applications that are used in engineering environments (see Figure 3.15a). As opposed to this, if both factors are activated, the *Scattering* will be fully random (see Figure 3.15c), which is identical to the starting situation in the *Training* (see 3.3.7) and *Testing* mode (see 3.3.7). This setting is useful to calmly practice the assembling of individual structures or the whole model. The *Scattering* feature is only available to the user if the model is in a solved state. In case the model is *scattered*, the user is provided with the possibility to automatically solve the puzzle. If activated, this feature will reassemble the model into its solved state. The reassembling takes places in form of an animation which can be observed by the user.
Not only is the *Exploration* mode relevant for gaining knowledge, it is also very suited to refine the user's skills in operating the features of the application. Moreover, it is plausible that a teacher applies this mode as part of a lecture to present anatomical information in a novel and interesting way.



(a) (b) (c)

Figure 3.15: The non-random explosion of the skull model, the complete skull model and a random explosion of the skull model.

Training Mode

The learning phase *Performance* is covered by this mode. The user can review his gained knowledge by applying it in a situation he is familiar with, without getting extensive help. In this mode, the actual puzzling takes place. When starting, the application loads and prepares the model. Afterwards it randomly scatters the individual pieces around the user. Only then the user can see the virtual environment. He now has to assemble the model, based on the knowledge he gained in the *Exploration* mode, analogous to Felix Ritter's 3D puzzle metaphor [65]. As described in Section 3.3.5 the user can activate various feedback features that help him to solve the task. As soon as the user has completely assembled the model he is presented with a floating screen that shows him the time and the number of *grabs* it took him to solve the task as well as the minimal number of *grabs* required (see Figure 3.16). For a perfect solution with minimal *grabs* the user would only need to *grab* each piece once. Results like this can be used to keep track of the individual learning progress. It also adds a competitive element to the application, as students can compare their individual results which again might lead to an increased motivation for using the software.
After finishing the *Training* task, the student use the user menu can return to the main stage or stay in this mode. If he chooses to do the latter, he can disassemble and reassemble parts of the model. This has no further effects on his initial result but might be helpful to review and discuss the task.

Testing Mode

This mode can be considered as the final stage of this application. It is basically identical to the *Training* mode, with one minor difference. Instead of solving the puzzle as he likes, the user has a specified order in which he needs to assemble the pieces. This mode satisfies the learning phase *Performance* as the student needs to apply his knowledge into a somehow unfamiliar situation. The mode has two variations: *Assembly* and *Disassembly*. A floating

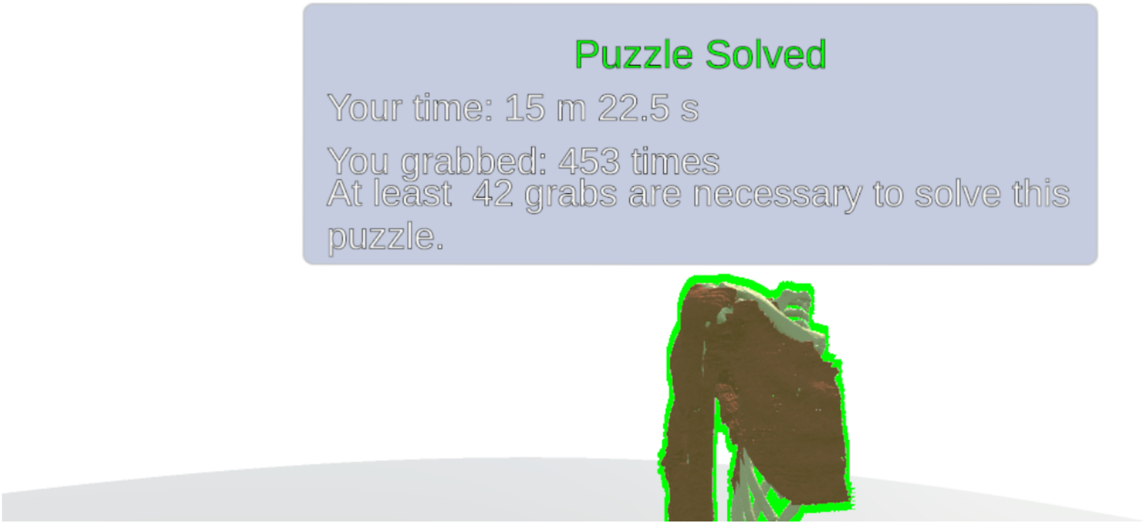


Figure 3.16: The results screen of the *Training* mode.

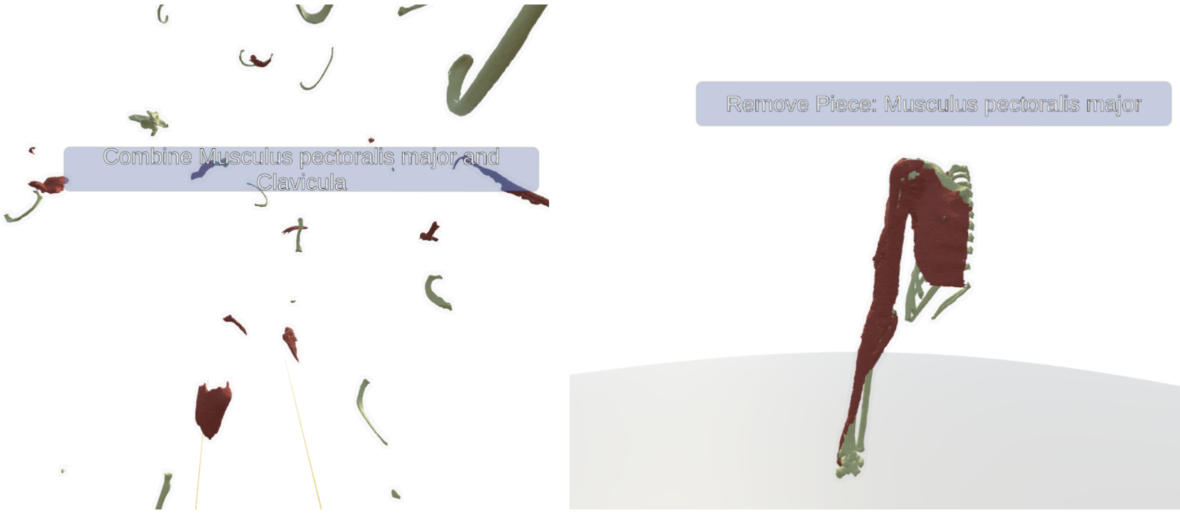
text describes the action the user needs to follow. With this mode not only the knowledge about the structure's names can be tested but, additionally, a more profound knowledge about the spacial alignment of each piece.

The *Assembly* mode follows the same idea as the *Training* mode. The user finds the puzzle in an unsolved state with each piece being scattered randomly around him. He than needs to put the pieces together as described by the displayed text (see Figure 3.17a). Only the pieces that are indicated can be put together. This variation is identical to the 3D puzzle metaphor and should be easy to grasp for the student. The task is considered as finished as soon as all pieces are put together.

The other variation, the *Disassembly* mode, is less familiar to the student. As in the *Exploration* mode he gets the model presented in a solved state. He then needs to remove the correct piece as indicated by the floating text (see Figure 3.17b). Only the indicated piece can be grabbed by the user. The task is considered as finished as soon as all pieces are removed from the model. This variation is similar to a real world dissection, where external structures need to be removed before internal structures can be reached.

After finishing the *Assembly* or *Disassembly* the user is presented with a floating screen that shows him the time it took him to solve the task. This is well suited to keep track of the individual learning progress and serves as meaningful measurement to compare different users' results.

Either students or teachers can easily create sequences for this mode by simply putting a *.csv* file that contains the name of the model, the desired variation and the desired order into a designated folder (see Listing 3.3.7). Students can then easily review their knowledge by



(a) (b)

Figure 3.17: The *Assembly* and *Disassembly* variation of the *Testing* mode.

providing test sequences for each other. Furthermore, the nature of this mode makes it an interesting possibility for actually testing students anatomy knowledge as part of an exam. It is even thinkable that surgeons use this mode as part of a preparation for a surgery.

Listing 3.1: An excerpt from a *Testing* file for the shoulder model.

```
Shoulder //the name of the model, this must correspond to the name of the
    folder the model data is in
1 // 0 = Assembly, 1 = Disassembly
Musculus pectoralis major //the desired of the pieces, the names must
    correspond to the names provided in the model data
Clavicula
Musculus deltoideus
Musculus biceps brachii
Musculus triceps brachii
[...]
```

3.3.8 Tutorial

As with every software application, first time users do not know how to properly operate the application. To minimize the learning curve, a simple tutorial is provided. This tutorial covers the basic interaction concepts of the application and is started automatically, each time the application is run. It is divided into eight tasks that need to be accomplished by the user. The tasks are presented in a consecutive order and each individual task needs to be done before the next one is presented. A floating text describes the action the user has to fulfil (see Figure 3.18). To make tutorial less complicated a simple model of a house that is composed of only four pieces is used. The individual tasks are as follows:

1 - Select a Piece: The user needs to select an object, either by pointing at it with a laser or by actually touching it with the controller.

2 - Grab a Piece: The user needs to grab an object by pressing the respective button on the controller while the object is selected.

3 - Get a faraway Piece: The user needs to get a piece closer to himself. It is stated that he can do this by double clicking the grab button on the controller while pointing at the piece.

4 - Grab two Pieces: The user needs to hold two objects at the same time. The description states that this is necessary to indicate an assembly attempt.

5 - Assemble both Pieces: The user needs to assemble both pieces. As a simple model is used for this tutorial, this task should not be to much of a challenge.

6 - Assemble all Pieces: The user needs to complete the puzzle with the skills he has acquired at this point.

7 - Open the Menu: The user needs to open the Menu using the menu button on the controller. The resulting menu shows only a single button.

8 - Click the Button: The user needs to click the button in the menu using the grab button on the controller. The text on the button clearly states "Finish Tutorial", which implies that clicking the button will finish the tutorial and the main stage will be loaded. In this context, the term "Main Menu" is used as an equivalent to the main stage and not the *main menu* sub-menu, as described in section 3.3.9. The choice of terminology derives from computer games, where the area from which all different game features can be accessed is often referred to as"Main Menu".

After finishing the tutorial the main stage is loaded. From there on the user can again restart the tutorial or use the different modes. To not cause any inconveniences for advanced user the tutorial can easily be skipped. It is already possible to open the menu at the beginning of the tutorial and then skip the tutorial. To make the tutorial mode distinguishable from the regular modes, the environment's sky and ground are tinted with a very dark grey color. This does not only make the puzzle pieces very easy to detect but also creates a sharp contrast to the light environment of the regular modes.

3.3.9 User Interface

As described in section 2.1.5, the user interface in a fully immersive environment needs to be *diegetic* if the user is required to directly interact with some of its components. Many VR applications simply place the UI as a static virtual element in the environment. This

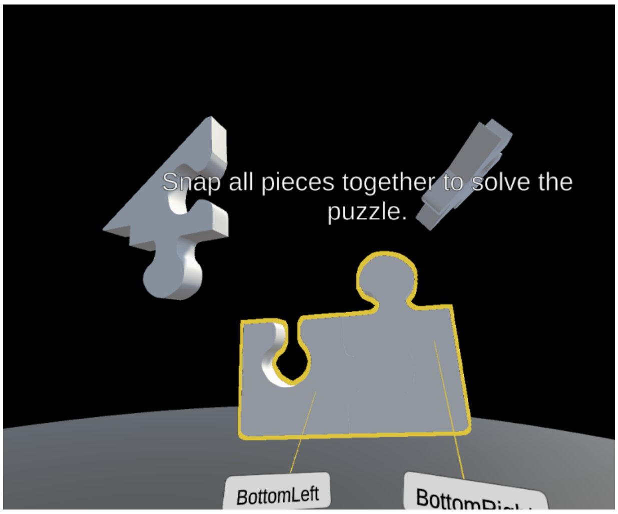
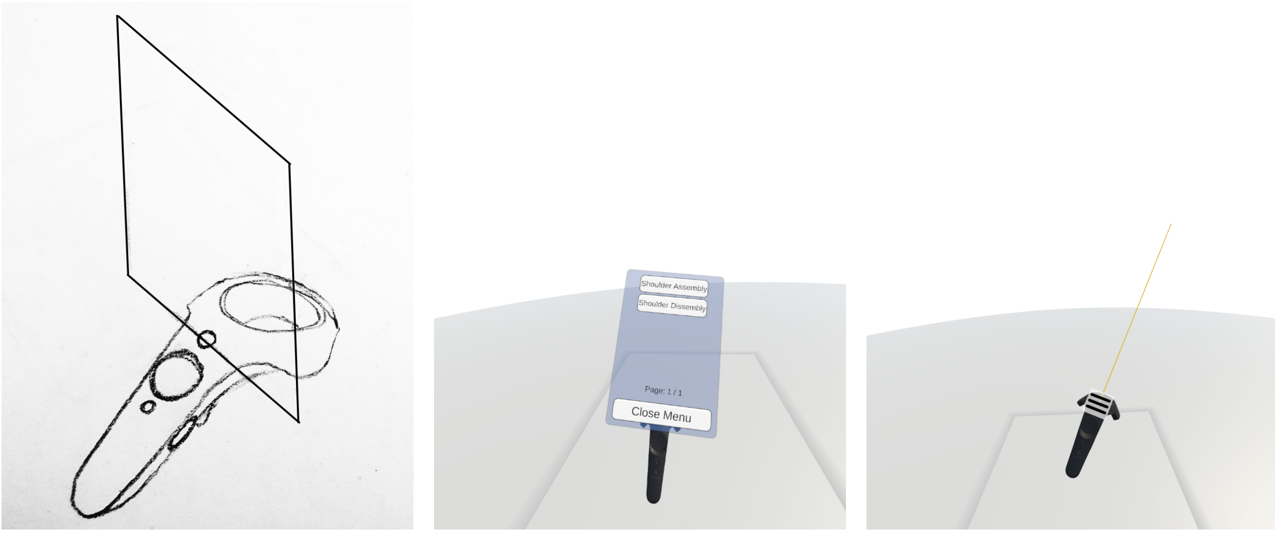


Figure 3.18: A screenshot of the *Tutorial*.

concept, based on a poster or blackboard metaphor, always requires the user to turn in the direction of the UI, if he wants to operate it. For a *room scale* setup, where the user can physically walk around, such an approach would obviously be a nuisance. Further, virtual puzzle pieces that are near the static interface, might sometimes be harder to detect or even fully occluded. A more convenient approach is needed. This leads to the concept of a user interface menu, that the user carries with him. The user is already carrying the real world controllers around, as described before, which suggests that it might be an intuitive solution to place the menu on the controllers (see figure (see Figure 3.19a)). To avoid occlusions or other distractions, the menu is simply not visible when its not needed. With a single click on the menu button, the user menu gets opened. A second click on the menu button closes the menu again. A hovering icon above the virtual menu button can be interacted with by the user like every other other virtual object ((see Figure 3.19c)). Clicking the icon will open the menu, which then replaces the icon. As soon as the menu is closed, the icon reappears. These two different interaction methods are used to comply to different users preferences.
The layout of the menu is kept minimalistic to enable the user to quickly find the interface elements he is looking for. Interface elements are vertically arranged on an upright plane to comfortably correspond to the reading habits of a western audience (see figure 3.19b). All interface elements are given a semi-transparent blueish touch, as this creates a familiar environment for users that are situated in an medical environment (see section 2.1.5). At the bottom of each menu the user can see a button that can be used to close the menu.
The user menu can contain multiple sub-menus at a time, depending on the mode the user is in. Only one sub-menu can be seen at once, to clearly indicate which sub-menu is active (see figure 3.19b). If more than one sub-menus are available, additional triangle icons appear left and right of the user menu (see Figure 3.20b). Similar to the hover icon, the triangle icons can either be operated directly with the controller or by pressing the touch pad. That way

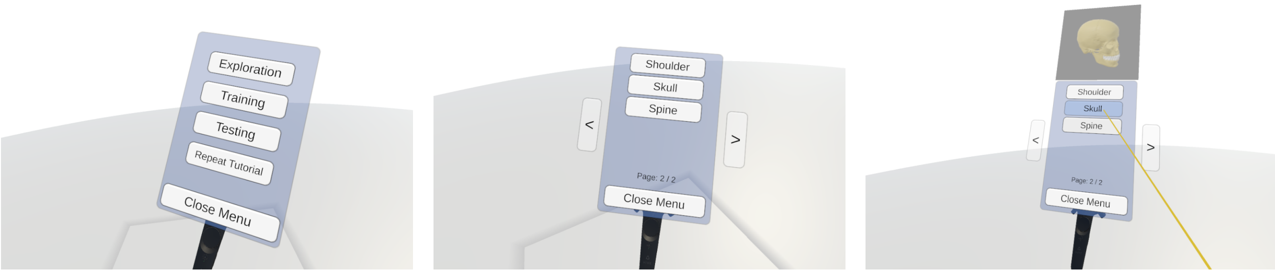
the user can flick through the different sub-menus in the manner that is most comfortable to him.



(a) (b) (c)

Figure 3.19: The concept for the user menu, a single sub-menu and the hover icon on the virtual controller.

Aside from the tutorial, the first environment the user is confronted with is the main stage. Here can only access the sub-menu *main menu*. It gives him the choice to either load one of the three modes (as described in section 3.3.7) or restart the tutorial (see Figure 3.20a). After selecting a mode, the *main menu* will disappear and the sub-menu *model selection menu* will be loaded. As the name suggests the *model selection menu* provides the user's models, that he can use in his chosen mode. It is dynamically created by checking a designated folder and providing information to all existing models. The extent of this sub-menu depends on the number of models present in said folder. This is necessary to regard the fact that all stakeholders can provide as many models as they wish to the application, as described in section 3.2. The text on each respective button matches the name of the model data file. If more models are present, than UI elements fit on a sub-menu page, further pages are created. This is clearly displayed by a text that shows both the total number of pages and the current number of the page the user has activated (see Figure 3.20b). To further indicate, that multiple sub-menu pages are available, the same triangle icons appear that are used to navigate between multiple sub-menus. This has been defined to keep the menu interaction consistent and not unnecessarily introduce further unfamiliar interaction concepts to the user. When selecting a button that belongs to a model, a preview image of the model is loaded and shown at the top of the user menu (see Figure 3.20c). This preview image is either loaded from a provided image file in the same folder as the model data file or directly from the model data file itself. This makes selecting the correct model much easier.
After starting one of the different puzzles modes, multiple sub-menus are available for the



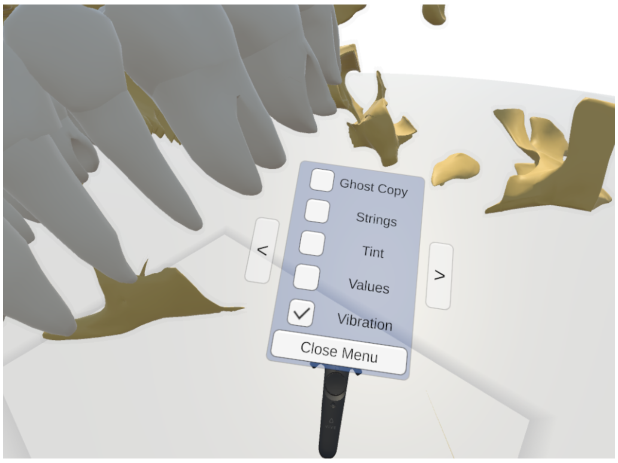
(a) (b) (c)

Figure 3.20: The *main menu* and the *model selection menu* without and with preview image.

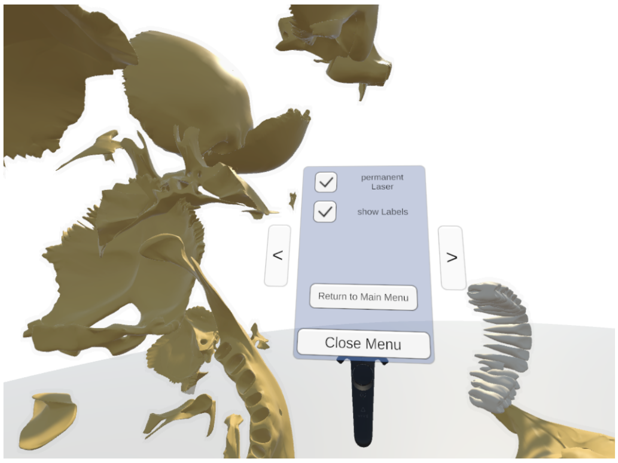
user, depending on the mode he has started. Here, one of the most significant sub-menus is the *feedback menu*. This menu provides an overview of all available feedback features, as described in section 3.3.5 (see Figure 3.21a). To activate or deactivate a feature, the user can click the corresponding checkbox next to the feedback's name. If a feature is active, the corresponding checkbox is checked and if it's deactivated, the checkbox is empty (see Figure 3.21a). This concept of a checkbox is widely known from other computer applications and should not be a problem for medical students, that are used to regularly use computer applications. This sub-menu is available in all three modes.

Another sub-menu that is available in every mode is the *system menu*. As of now, it contains three buttons (see Figure 3.21b). The most important one reads "Return to Main Menu". As the text suggests, clicking this button will abort the current mode and the user will be brought back to the main stage. Just as in the section 3.3.8, the term "Main Menu" is used as an equivalent to the main stage and not the *main menu* sub-menu. The other two buttons allow the user to enable or disable the laser that is permanently drawn or the label that is always shown when a pieces is *selected* or *grabbed*. If the permanent laser is deactivated by this setting, the user can always use a temporary laser that appears as long as he keeps the center of the touch pad pressed. Both options are only provided in case a user feels distracted by those features. This sub-menu additionally servers as a placeholder for settings that might be adjustable in future extensions of the concept, like changing the language of the application or changing settings related to sound.

The only mode that has an exclusive sub-menu is the *Exploration* mode. It provides the user with an *exploration menu*. This menu operates the features that are exclusive to the *Exploration* mode. The most prominent feature is the "Scatter Model"-button. As described in section 3.3.7, this button will trigger the *Scattering*. The randomness of the rotation and position can be adjusted by two checkboxes that are placed just above the button. After a model has been *scattered*, this button will disappear and a "Reassemble Model"-button appears in its place. Clicking this button will reassemble the model and then again change places with the "Scatter Model"-button. This sub-menu further provides a button that



(a)



(b)

Figure 3.21: The *feedback menu* and the *system menu*.

resets all of the transformations that the user has applied to the model (see section 3.3). This feature is not present in the other modes because scaling or rotating the whole model does not have any irreversible effects on the puzzle task.

4 Implementation

The following chapter describes relevant information about the developed prototype. At first, the used hardware is described. Following this, the *Unity* game engine is presented. This chapter concludes with briefly presenting specific implementations of some aspects of the prototype that was realized over the course of this thesis.

4.1 Hardware

To develop the software prototype useful hardware is required. This section briefly described the hardware that serves as input and output interface for the user. Further, the computer setup that was used for development is presented.

4.1.1 HTC Vive

At the time the prototype for this thesis was developed, two VR systems are available that could potentially be used for the application – *Oculus Rift* and the *HTC Vive*. While the *Oculus Rift* can only provide a fully immersive experience, let alone on a *room scale* experience, with additional hard- and software, the *HTC Vive* provides these features per default. Everything that is needed to use a fully immersive *room scale* VR application comes with the basic *HTC Vive* setup. Because of this, using the *HTC Vive* for a prototype is the obvious choice.

The *HTC Vive* system basically comes with three components – the HWD, the controllers and the infrared-cameras (see Figure 4.1[1]). The most important part, the HWD, contains two OLED displays, one per eye. Each provides a resolution of 1080 x 1200 pixels with a refreshing rate of 90 Hertz. The displays are positioned so that the user has a field of view of about 110°. The device provides a small adjusting wheel to precisely adjust the field of view to the users actual pupillary distance. Additionally, the HWD contains 16 sets of respectively a gyro sensor, an accelerometer and a photosensor. This enables a precise 360° experience with 6 degrees of freedom. These sensor are also used by two so called *Lighthouse* cameras, also known as *Steam-VR Base Stations*. The systems transmit infrared lasers

[1]http://www.independent.co.uk/life-style/gadgets-ansQA (Accessed: 03.06.2017)

Figure 4.1: All components of the *HTC Vive* system.

that can then be received by the photosensors on the HWD. Based on the time difference between transmitting the laser signal and the receiving on the HWD's sensors, the system can calculate the exact position and orientation of the HWD. By placing both *Lighthouse* cameras in two opposite corners of the room, a play area of roughly 5 m x 5 m can be created. Of course, the area is much smaller if both cameras are positioned closer to each other. As long as the sensors are not occluded by any obstacles, it can be fully tracked in this play area.

As the chosen output device *HTC Vive* is shipped with a pair of fully tracked wireless controllers, these *HTC Vive Controllers* are the obvious choice for input devices for the presented application. By each using 24 sets of the same sensors as the HWD, they are as well fully tracked in the play area by the *Lighthouse* cameras. Multiple input interfaces on each controller enable the user to almost naturally interact with virtual objects (see Figure 4.2a[2]). For the prototype, these interfaces have been assigned to serve as the buttons that are described in section 3.3. The input layout for the developed application is as follows. The numbers correspond to the ones labelled in Figure 4.2b[3].

1 - Menu Button: This button serves as the menu button of the approach. It's exclusively used to open and close the user menu.

2 - Trackpad: The multi-function trackpad serves as the touch pad . While an object is *grabbed*, vertically swiping over the pad will get the object closer to the user. If a menu

[2] http://www.tomshardware.com/reviews/vive-rift-playstation-vr-comparison,4513-6.html (Accessed: 03.06.2017)

[3] https://www.vive.com/de/support/category_howto/about-the-controllers.html (Accessed: 03.06.2017)

with multiple sub-menus is active, clickin on the right and left side of the pad will switch through the available sub-menus.

3 - System button: This button is not used for the prototype. It is generally used to open the external application *Steam-VR Home*.

4 - Status LED: These LEDs represent the current state of the batteries.

5 - Micro-USB port: This port is used to connect a micro-USB cable to the controller, which is necessary to load the internal battery.

6 - Sensor set: Set of a gyro sensor, an accelerometer and a photo sensor.

The haptic feedback that is provided by the controller is used to realize the vibrotactile feedback feature.

(a) (b)

Figure 4.2: The *HTC Vive* controllers and a simplified illustration that shows all relevant elements.

The only major downside of the *HTC Vive* system is the fact that the HWD weighs 470g. This could potentially end up as a physical nuisance for long term users and needs to be further observed.

4.1.2 Developer PC

To develop a VR application for the *HTC Vive*, a PC with powerful hardware specifications is required. The specifications of the PC that was used for development are as follows.

CPU: Intel® Core™ i7-6700 @ 3.40 GHz

GPU: NVIDIA GeForce® GTX 980

RAM: 16 GB

Operating System Windows 7 Professional SP1 64 Bit

With this setup, the developed prototype runs with approximately 55 FPS and without running into significant performance issues. Loading highly detailed models at runtime sometimes results in a short noticeable performance drop, which however does not disturb the work flow of the user.

4.2 Unity

To implement the proposed prototype the *Unity*[4] game engine by *Unity Technologies* has been used. With the main focus on digital games, the framework can also be easily used for applications that do not serve an exclusively entertaining purpose. By using the *Unity* game engine, applications for different operating systems can easily be created. The inclusion of advanced graphics concepts such as *Real-time Global Illumination*, *High Dynamic Range Rendering* and *Physically Based Shading* make it possible to create even very visually advanced virtual worlds. The included *Asset Store* provides numerous, partially free, assets (3D models, textures, scripts, etc.) so that prototype solutions can be created very fast. Apart from all these factors, one of the most meaningful reasons to use *Unity* for VR development is the fact that all concurrent VR systems' interfaces are natively supported by the engine. To use the *HTC Vive*, the *SteamVR* plugin[5] from the asset store needs to be imported into the respective project.

Some terms and concepts need to be clarified before talking about the actual implementation with the *Unity* framework.

Scene: A *Scene* serves as a container for objects and everything else that is needed for a *Unity* application. Usually, applications consist of multiple *Scenes*. In many game applications each level is realized as a separate *Scene*. However, it also possible to create applications entirely with only one *Scene*.

GameObject: Every object in a scene can be considered as a*GameObject*. This does not only include visible assets like 3D models or UI elements, but also objects that contain logic, like managing settings or game states. It is possible to hierarchical interlink many *GameObjects.GameObjects* that are hierarchical lower than another *GameObject* are called children while the respectively higher *GameObject* is called parent. A child will always keep its relative orientation to its parent when the parent is moved.

Component: Every *GameObject* contains *Components*. They further describe the object and its purpose. Some *GameObject* can for instance contain 3D mesh *Components* which give them a visual representation in a *Scene*, while another *GameObject* simply contains a camera *Component*, which is needed to render the *Scene* itself.

[4]https://unity3d.com/ (Accessed: 03.06.2017)
[5]https://www.assetstore.unity3d.com/en/#!/content/32647 (Accessed: 03.06.2017)

Script: Every *GameObject* can be extended through *Scripts* which are basically classes that are treated like *Components*. They provide additional functionality or operate existing *Components*. By using *C#* or *JavaScript*, *Scripts* are used to actually program a *Unity* application. *Scripts* that are actively used in *Scenes* usually are inherited from the class *MonoBehaviour*. Of course it is also possible to utilize independent classes and *Scripts*.

Prefab: *Prefabs* are special *GameObjects* that are saved as files outside of a *Scene*. Through *Scripts* they can be easily instantiated into any *Scene*. They are primarily used to handle *GameObjects* that are needed very often but are not a crucial element of a *Scene*.

LifeCycle: As most *Scripts* are usually inherited from the *MonoBehaviour* class, they follow the so called *LifeCycle* of a *Unity Script*. This involves functions that are automatically called at certain times during the execution of the application. The most relevant are *Awake* (called while a *Scene* is loaded) , *Start* (called after a *Scene* is loaded and before the first frame is drawn) and *Update* (called for each frame).

Further information is provided in an extensive online documentation[6] that serves as a great base to learn more about *Unity*.

4.3 Developing the Prototype

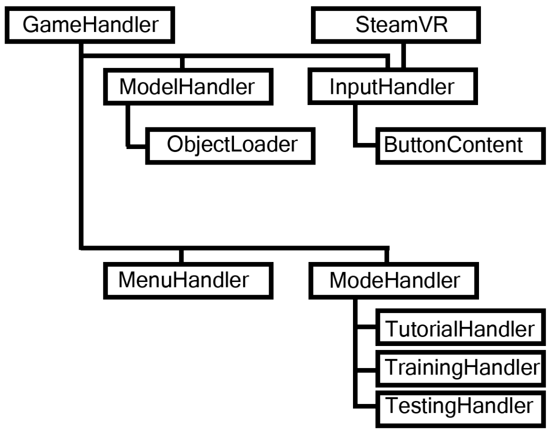


Figure 4.3: A simplified diagram for the relevant classes of the prototype.

Apart from the *Tutorial*, the whole application runs within a single *Scene*. The most important element is the *GameObject [SteamVR]*. It is completely provided by the *SteamVR* plugin and handles all basic features that are necessary to develop a *HTC Vive* application, such as tracking the user and the controllers, rendering the controllers and creating

[6]http://docs.unity3d.com/manual/index.html (Accessed: 03.06.2017)

the stereoscopic camera image. The whole puzzle experience is handled by a *GameObject* named *GameHandler*. It contains a *Script [*GameHandler] that manages the whole *Scene*, its *GameObjects* and all relevant *Scripts*. It further handles the communication between all relevant *Scripts* that are attached to this *GameObject* as well. The most relevant of these *Scripts* are the *ModelHandler*, the *InputHandler*, the *MenuHandler* and the *ModeHandler*. The *ModelHandler*, as the name suggests, takes care of the model that is used for puzzling. Using another class *ObjectLoader*, it loads the selected model into the *Scene* and prepares everything that is necessary to turn any model into a working 3D puzzle for this application (see Pseudocode 4.3). Furthermore, the *ModelHandler* takes care of automatically assembling and disassembling the model, which is necessary at the start of the *Training* and *Testing* mode and for the *Explosion* in the *Exploration* mode. This class also runs the *Snapping* process that needs to happen when two puzzle pieces are successfully combined (see Pseudocode 4.3).

```
find visualCenter c of loadedModel;
create HelperObject h in position of c;
for all Piece p in loadedModel do
  if p does not contain mesh information then
    remove p;
  end if
  add copy of h to p as child;
end for
remove h;
scale loadedModel to fit into defined dimensions;
move loadedModel away from Player;
```

Algorithm 1: Pseudocode for the necessary preparation of a model (see Section 3.3).

```
%% Piece1Helper is child of Piece1
%% Piece2Helper is child of Piece2
if Piece1Helper and Piece2Helper are within snapping distance then
  make Piece1 child of Piece1Helper;
  move Piece1Helper to position of Piece2Helper;
  make Piece1Helper child of Piece1;
  remove Piece1Helper;
  tell Piece1 and Piece2 that they are now connected;
  tell Piece1 that it has to use Piece2Helper;
end if
```

Algorithm 2: Pseudocode for the *Snapping* of pieces (see Section 3.2.2).

As the name implies, the *InputHandler* handles everything related to the input that comes from the *SteamVR* plugin. To be more precise, it tells the other *Scripts* what to do, if certain buttons are pressed. By using the *Unity* standard function *Physics.Raycast*, this class realizes the *Selection* of pieces with the laser. It also manages the collision detection between the puzzle pieces and the controllers to realize the *Grabbing*. The *MenuHandler* instantiates *prefabs* of all sub-menus(see Section 3.3.9) that are needed in the *Scene* by attaching them to the controller models of the *[SteamVR] GameObject*. The functionality of each sub-menu is handled by this class, including the *feedback menu*. By using the class *ButtonContent* it handles all actions that occur when the different buttons are pressed. Which sub-menus are needed at which time is managed by the *ModeHandler*. By communicating with the *MenuHandler*, this *Script* also calls functions from classes that handle the unique behaviour of each mode. These classes are the *TutorialHandler*, the *TrainingHandler* and the *TestingHandler*. There is no specific class that handles the behaviour of the *Exploration* mode, as there is a unique sub-menu that provides all the necessary functionality as described in Section 3.3.9. The *TutorialHandler* class manages each task that needs to be fulfilled by the user to advance trough the tutorial (see Section 3.3.8). It waits for the specific user action by using *Unitys* coroutines. For the statistics shown after successfully solving the puzzle in the *Training* mode the *TrainingHandler* keeps track of the time, and number of the user's grabs (see Section 3.3.7. It also keeps track of the user's general puzzle progress to determine when exactly a puzzle is solved. The *TestingHandler* class inherits this functionality and further monitors that the user can only progress according to the specified piece order in the *.csv* file that was also loaded by this class.

5 Evaluation

Existing software can be evaluated in various ways. Principally qualitative and quantitative evaluation is distinguished. Often, a combination of both evaluation approaches is advantageous. A combination like this, which is referred to as *Mixed-Method Evaluation* [25], is used for this thesis. Qualitative insights are gained through the so called *Thinking Aloud* method. This method, while being easily applicable, produces robust results, which is why it is one of the most used methods in evaluation processes [53]. To gather quantitative data a questionnaire is used. In this context, the focus is mainly on questions regarding usability and the usefulness of the application's features. Additionally, while operating the application, each proband's progress and various times are tracked. While the prototype was still under development it was presented to various experts in an informal setting. This resulted in first feedback and suggestions for improvement, as described in Section 5.1. The actual informal evaluation, using the questionnaire and the data tracking happened after the completion of the prototype. Sections 5.2 and 5.3 describe the setup of a user study and its results. Furthermore, possible improvements for the application are discussed. At last, the results are interpreted to show how the requirements of the application have been met, as discussed in Section 3.1.1.

5.1 Work-In-Progress Feedback

In early 2017, while the prototype was still under development, a workshop for *Visual Computing for Biology and Medicine* at the *Otto-von-Guericke University* in Magdeburg provided the great opportunity to obtain initial feedback. Experts in the fields of computer graphics, visualization, computer vision, visual analytics and human computer interfaces with a clear focus on biology and medicine applications, briefly tested the prototype. Among them, Felix Ritter, who is one of the main contributors to the scientific foundation of this thesis (see chapter 3). The people that actually tested the prototype, where generally confident that the basic idea of such a scenario might be interesting for an application in medical education in the future. However, they doubted that, at this time, the application is superior enough to traditional learning methods that it justifies the required resources for such a VR setup, like money and space. It was also remarked that the physical interaction, that is required from the user, might be exhausting in the long term, which might be counterproductive

to the learning experience. People agreed that this effect could be minimized by using the application as a *seated* experience or might even be diminished by the fun the user is experiencing while using it. The latter comment was based on an analogy that can be easily drawn between the presented prototype and computer games. It is often said that computer gamers forget about feelings of exhaustion when they are experiencing fun or a strongly immersed into a game.
Further, it was pointed out that the *snapping* threshold was to small for inexperienced users. Changing the threshold to the length of the vector between the child helpers positions to 100 mm and the angle between the rotations to 10° was considered as more comfortable.
While various people where testing the prototype, a fundamental problem became apparent. Most users where generally more interested and excited about the VR technology than the application itself. Some even had a very hard time, to grasp that they can physically walk around in the virtual environment and can use the controllers like a virtual representation of their hands. Both observations can be attributed to the fact that the VR technology is not yet widely prevalent even though it is publicly available. Many people have just not yet experienced a fully immersive virtual environment. This turned out to be a valuable observation for the actual evaluation process. It it utmost necessary to thoroughly explain the functionality of the *HTC Vive* hardware and give the user time to marvel at the virtual reality he experiences.

5.2 User Study Setup

To keep the results of the user study consistent and comparable, each proband was tested according to the same fixed pattern. Before actually experiencing the virtual reality, the VR hardware was explained. It was especially emphasized, that the proband needs to be careful to not accidentally damage the hardware or hurt himself by running into real world obstacles. Following this, the prototype and its purpose was briefly explained. It was also declared, that the task of the study, would be to learn the names and spacial relations of structures from a human shoulder and that during the whole process, the examiner would be present and could be talked to. Afterwards, the probands completed the first page of the questionnaire (see Appendix 6.1). Before actually doing this, they where assured that their collected data would only be used in the scientific scope of this thesis. This first part focused on the demographics details of the probands, as well as their experience in relevant topics. The handedness and potential color vision deficiencies were also inquired.
As described in Section 5.1, many people are excited to use VR technology for the first time. This can easily lead to distractions from a given task. To counter this effect, the probands are presented another VR application before actually using the prototype. For this purpose,

Tilt Brush[1] by *Google* has been chosen. This 3D painting application is well suited to present the functionality of the *HTC Vive* and VR in general. For five minutes, the probands could explore this application as desired. Afterwards, the application was closed and the prototype was started. It was again pointed out that the probands can ask questions at all times, as well as present their thoughts and give feedback or offer criticism. It was expected that this segment would take five minutes altogether.

At first, the probands were presented the *Tutorial* (see Section 3.3.8). The task was to complete it, with as little help from the examiner as possible. After solving the *Tutorial*, the probands had to complete the according questionnaire section as shown in Appendix 6.1. As this would be impossible while wearing the HWD, the probands were allowed to fully remove it. This served the secondary purpose that every proband could relax from the weight of the HWD. The questions mainly focused on the quality of the *Tutorial*. It was expected that this segment would take six minutes altogether.

After putting the HWD back on, the probands had to start the *Exploration Mode* (see Section 3.3.7). To convey the idea of this mode, the probands first started it using the simple model of a *Rubik's Cube*. This model is regarded as already being familiar for many people. For five minutes, they could freely explore the model using the different features of this mode. Every feature was explicitly introduced by the examiner. Afterwards, they had to restart the mode, but this time use the shoulder model. It was again stated, that the task was to learn names and spacial relations of the structures that are part of this model. During ten minutes, the probands had to gain as much knowledge as possible. After the time ran out, the probands could again remove the HWD to complete the section of the questionnaire that regards the *Exploration Mode*, as shown in Appendix 6.1. The questions focused primarily on the quality of the *Exploration Mode*. The expected duration for this segment was sixteen minutes.

The next step was to actually solve a puzzle, using the *Training Mode* (see Section 3.3.7). Again, this mode was first introduced with the *Rubik's Cube* model. Using the actual jigsaw puzzle metaphor, the task was thoroughly described by the examiner. During a two minute timeslot, the probands could understand the task using the example of the *Rubik's Cube*. Afterwards, they restarted the mode, this time using the shoulder model. This evaluation segment was further divided. At first, all feedback features had to be disabled by the probands. Now they had to try to solve the task without any feedback, for one minute. During the following five minutes, the five feedback features (see Section 3.3.5 and 3.3.6) where iterated, so that the proband could only use one feature at a time for one minute each. The order, the feedback features had to be used, was randomly decided for each proband. After having used each feedback feature once, the probands where given another four minutes to freely solve the puzzle as they like. It was explicitly explained that they

[1] https://www.tiltbrush.com/ (Accessed: 03.06.2017)

can use any feedback feature in any combinations. If the proband was able to completely reassemble the shoulder model, this segment was prematurely finished. Again, the probands had to complete the section of the questionnaire that regards the *Training Mode*, as shown in Appendix 6.1. These questions focused in the quality of the *Training Mode*. It was expected for this segment to take thirteen minutes.
Finally, the probands encountered the *Testing Mode* (see Section 3.3.7). For this segment, the *Assembly* variation of the mode was used. As with the previous segments, this mode was first introduced using the *Rubik's Cube* model for two minutes. Afterwards, the probands had ten minutes to solve the task using the shoulder model. Afterwards, they could remove the HWD for good. Now they first completed the part of the questionnaire that regards the *Testing Mode*, as shown in Appendix 6.1. The probands then completed the final part of the questionnaire that contained questions about their experience in VR, the usage of the prototype and the usefulness of the feedback features. This whole segment was expected to take seventeen minutes. To conclude the evaluation process, the examiner and each proband had an informal discussion about the probands experience while using the prototype.
Altogether, it was expected that the evaluation process would take at least one hour per proband. This gave each proband enough time to thoroughly try out each feature of the prototype and would also tell if the probands would show signs of exhaustion, after using the application for a longer period.
While creating the questionnaire it was considered to phrase each questions so that they would not already imply an answer. For example, instead of asking "How easy was it to operate the User Interface?" the question was phrased as "Rate the User Interface in terms of usefulness.". The whole questionnaire uses a consistent 5-point scale, whereby the odd number of possibilities enables the user to give a neutral answer.

5.3 User Study Results

Mainly people that belong to the target audience of the application would have been relevant for a formal evaluation process. Due to the medical routine, that leaves not much time for activities outside of healthcare facilities, it was not possible to gather a significant number of probands that could be regarded as potential stakeholders for this application. Therefore, the group of probands mainly consists of people that were eager to participate in an informal evaluation process for a VR application. For this reason the presentation of results from the user study will mostly focus on qualitative information.
Altogether, ten people participated in the study. All of them were between the age of 24 and 32. Out of the ten probands, two were female. Three probands were left-handed. By his own account, one proband suffers from color vision deficiency, mildly affecting his red-green hue discrimination. Half of the probands, were majoring in Computer Science. Two participants

were Engineering students. Of the remaining probands, two were psychologist and one was a paramedic. The two psychologists had to learn the anatomy of the human skull and the human brain during their study. The paramedic had to learn comprehensive basic knowledge about human anatomy during his apprenticeship and even took part in an actual dissection as preparation for his future career. These three probands and their impressions were especially valuable for the evaluation of the educational application. They are further referenced as the *Probands of Interest*, or *POIs* for short. Due to their educational background, all probands were familiar with computer systems and 3D applications. All of them were also familiar with using 3D input devices, mainly contributed to the popular gaming console *Nintendo Wii*. None of the probands had any previous experience with fully immersive virtual reality applications. This was the most driving factor for most of the probands to participate in the study, which suggests that the technology can spark interest in the subject. All of the tests took between 70 and 90 minutes. This can be attributed to the fact that the participant were excited to talk about the experience between the test segments. Afterwards, none of the probands were bothered by this time, as they were enjoying the virtual reality. During the whole user study, only one proband accidentally walked into a chair, that was located too close to the playing area. From a technical point, no problems occurred.
The results of the user study will be presented in order of the segments of the study, as presented in 5.2. A more thorough discussion is given subsequently.

Tutorial All probands were able to successfully complete the *Tutorial*. Everyone described it as comprehensible and easy to solve. Two probands remarked that the whitish text, that describes the individual tasks, is sometimes hard to read if it is in front of one of the light-grey pieces. All probands agreed that the extent and the difficulty of the *Tutorial* was appropriate for the presented functionality. One proband proposed that it might be helpful to dictate the order in which the *Tutorial* puzzle must be solved, to remove the initial trial-and-error moment. The probands rated the *Tutorial* with an average grade of 1.2, corresponding to a western school grading system.

Exploration All probands enjoyed exploring the *Rubik's Cube* model, as they were familiar with it. Everyone commented that it would be interesting if the pieces could actually be moved like in the real world counterpart. Even though this remark seemed not to relevant at first, similar comments came for the shoulder model. Here, the *POIs* noted that anatomical structures are not always attached to their adjacent neighbours,as implied by the nature of the 3D puzzle. As an example the jawbone was given, which is only attached to the skull at specific fixation points. Understanding such functional relations, can be very helpful for understanding spacial relations. In general, all probands understood the purpose of this mode and liked the general concept of the 3D puzzle. Only two probands commented that they do not like the *Explosion* feature as they would prefer to detach pieces by themselves to gain a more thorough insight into individual structures. As the *Explosion* feature does not interfere with their preferred behaviour, they did not consider this as something negative.

On the contrary, three other probands really liked the possibility to see how the model is reassembled automatically. After randomly scattering the pieces, they tried to assemble different structures and would then watch where this assembled structure would be located in the completed model. Furthermore, the *POIs* noted that the possibility to freely scale, rotate and move the whole model and individual pieces in this VR environment, was a great feature which gave the application a huge advantage over the learning methods they used when they gained their anatomical knowledge. Altogether, the participants rated this mode with an average grade of 1.3. This might be attributed to the fact that some probands felt a little bored, after having used all the features of this mode. Four probands explicitly wished for additional “entertaining” features. As the main purpose of this application is educational and not entertaining, this remark must not be too heavily considered.

Training The task and purpose of this mode was easily understood by all probands. While being confronted with the *Rubik's Cube model*, most of the participants already underestimated the difficulty of the puzzle task. This mindset changed immediately when the participants encountered the scattered shoulder model. For the first minute, where not feedback feature was active, none of the probands managed to assemble any pieces. The same occurred during the following five minutes where the probands had to try out each feedback feature individually. Afterwards, the situation improved. Every participant immediately activated the vibrotactile feedback and the *Ghost Copy*. Four participants also activated the *Tinting* while two choose the *Strings*. The *Display* was used by no proband at all. With the help of this feedback features all participants were able to assemble some parts of the structure. Most of them focused on parts that were somehow familiar, like the rib cage or the arm bones. Interestingly enough, only two people tried to assemble muscle structures, while the others considered these as too complicated. It was also easy to observe that the probands did not acknowledge the names of the individual pieces, as presented on the labels, but rather chose the pieces by looking at their shape. One proband even deactivated the labels to be less distracted. When the time ultimately ran out, no proband was able to finish the puzzle. The most progress was gained by the paramedic, who could've probably finished the puzzle within a reasonable amount of time. This proband attributed his performance to the limited amount of time and the fact that the smaller pieces are very hard to select with the laser, especially if one is not used to the controllers. All other participants explained their performance with the absence of relevant anatomical knowledge. For an average user without prior anatomical knowledge, this task was rated as too difficult. However, all probands remarked that they would most likely be able to solve the puzzle of a model they are familiar with. As all participants enjoyed the basic idea behind this mode and remarked that an actual medical student would most likely perform better, they rated this mode very well with an average grade of 1.2. The presence of the feedback features was very much appreciated, even though some were clearly more useful than others.

Testing Just as the *Training* mode, no proband was able to solve the task of this mode.

Apart from the paramedic, all participants first relied on an trial-and-error search for the stated pieces and then used the vibrotactile feedback and the *Ghost Copy* to assemble them. Due to his experience from the prior mode, the paramedic was able to easily identify most of the pieces and assemble them using the same feedback features as before. Again, due to the small pieces being to difficulty to select, he was not able to finish the task. This mode was surprisingly well received by all probands. Despite their bad performance, all participant claimed that this mode was really "fun" and could see the relevance in an educational scenario. One proband jokingly compared this mode to "stitching together a person after a car accident". The probands enthusiasm for this mode resulted in an average grade of 1.1

Feedback Features As all feedback features were initially presented to the probands, it was up to them to later choose which feature they prefer. According to the questionnaire, the *Display* feature was rated as useless and incomprehensible. All probands agreed that they were not able to translate the shown values into the movements they had to make to assemble pieces. This corresponds to the tracked data, that clearly shows that all probands used the *Display* feature between 70 s and 90 s. Taking the time into account that the probands were forced to use this feedback and the time it might take to deactivate a feedback feature, it is clear that the participants only used this feature for as long as necessary. Contrary to this, the probands agreed that the *Ghost Copy* was very useful and clear. According to all probands it was the only feature that easily helped in finding the correct orientation of the pieces. Seven probands put this further into perspective by saying that even if the *Ghost Copy* suggests the correct orientation, it is still not easy to fit the piece into the ghost shape. This is not attributed to inaccurate interaction with the virtual world, but to the fact that many structures do not have very significant features that are required to compare a piece to its *Ghost Copy*. Especially for muscular structures it is hard to tell which orientation is the correct one. For this reason the vibrotactile feedback was very appreciated as well. While the vibrotactile feedback was not immensely helpful with orienting the pieces, it was the most significant indicator for a successful *snapping attempt*. Without it, probands performed significantly worse and felt more "helpless", according to their own statements. The tracked data indicates, that all probands used the vibrotactile feedback and the *Ghost Copy* for as much time as possible. The *Strings* and the *Tinting* were met with mixed feelings. Half of the probands did use neither of these features after initally trying them out. Three participants used the *Strings* for the whole *Training*, but not for the *Testing* and three probands used the *Tinting* for the whole *Training* a well. This includes one proband who had both feedback features activated at the same time. According to the probands, the idea behind the *Strings* was clear but it was not always easy to comprehend which movements are necessary to align the pieces correctly. The people that kept using the *Tinting* said, that this feature was a great addition to the vibrotactile feedback and the *Ghost Copy*. The results of the questionnaire clearly show that the participants preferred the combination of vibrotactile feedback and *Ghost Copy*. Some additionally named the *Tinting* as a preferred addition. No one included the *Display* feedback or the *Strings* feedback in their list of favorites.

General Remarks and Suggestions for Improvement According to the probands own statements, neither of them experienced any effects of nausea while using the application. Some remarked that they indeed experienced light exhaustion after using the application for more than an hour. They attributed this to the fact that they had to hold up their arms for a long time. In this context they agreed that sitting on a chair might help to reduce such effects. Nevertheless, this would probably also reduce the fun they all experienced while using the fully immersive VR system, which made the exhaustion in the end tolerable. The user interface was met with positive feedback. The participants rated it as easy to operate the different menus and liked the idea of having a "holographic menu" to carry around. In this context, the left-handed probands did at no time feel restricted while using the different interfaces. The choice of the dark red - light green hue for some of the feedback features was considered acceptable and the proband with color vision deficiency said he was clearly able to discriminate the hues. All participants agreed that is was intuitive and to a certain degree natural to *select* and *grab* the pieces. Unfortunately, the *Selection* of small object that were far away, required too much accuracy with the laser from the participants. This was mellowed by the possibility to easily get objects to come closer once they were successfully *grabbed*. It is still required to introduce a feature that makes it easier to select small far away objects. A possibility would be an adjustment of the size of the laser itself to make the potential selection are bigger. Some participants remarked that the names of the different feedback features were not always intuitive and that they did not always know exactly which feature is which. To solve this issue, it was proposed to add a simplified figure that illustrates the feedback to the menu, similar to the preview image in the *model selection menu*. Some participants noted that a highscore feature for the *Training* and the *Testing* mode would be interesting. Not only would this be a great opportunity for the user to keep track of his study progress, but would also give him the chance to compare and discuss his results with fellow students. Both engineering students agreed that working with an application like this would be very welcome in their curriculum. This suggests that the concept of a 3D VR puzzle is potentially interesting for students that are looking for non-traditional ways of gaining knowledge. The *POIs* noted that it would be an interesting addition to not only connect the pieces like a puzzle, but to consider how they are actually attached to each other. Adjacent parts are not always conjoined. This would require a whole new approach for the functionality of a 3D puzzle. they further noted that it would be interesting to provide models that are as lifelike as possible. Not only the shape and the texture of the anatomic structures should be photo realistic, but maybe even their consistency could be conveyed. Real life muscle tissue is, for example, not as rigid as the application presents it. For many students it could be interesting to somehow get an impression of how soft or even fragile some anatomical structures actually are. The only participant that has actually attended a real life dissection, the paramedic, really enjoyed the prototype. He agreed that it is imaginable to supplement anatomy education with an application like this.Even though he is well aware of the problems that come with real life dissections, he declared that these procedures would be irreplaceable in the context of anatomy education. According to him, many medical students realize only through the dissection courses that real arteries are not red and veins are not blue. The process itself is also a great indicator to tell which students are physically and psychologically fit to be working as medical doctors. He anecdotally explained that the visual, acoustic and olfactory impressions that students get from the work with a real body donor, are nowhere near as bad as the things many people that work in healthcare facilities often encounter. He conclusively remarked that working with a real body donor teaches things that can not easily be taught such as the ethical and moral competency a student gets while working with an actual human being. According to him, the respect and veneration for the human life, that most people that work in healthcare facilities have, can not be taught virtually.

5.4 Interpretation of the Results

Before the concept of an educational scenario was introduced, various requirements were determined that had to be met by the concept and the prototype. This section discusses in which extend each requirement has been met.

Real Time: As described in Section 4.1, the application runs stable with approximately 55 FPS. For VR applications, this qualifies as *real time*.

Activity Feedback: As described in Chapter 3, the application only contains features that immediately react to the user's input. At no point the user needs to wait for the system to react. The only exception might be the process of loading a large 3D model as a puzzle. There is not yet any feature that indicates this loading process, as the amount time it takes for the model to load is considerably small.

Assurance: A subtle grey rectangle on the ground indicates the physical area the user can freely walk in while wearing the HWD, as described in Section 3.3.3. The user study showed that users feel generally safe walking around the virtual world and relied on illustrated area.

Intuitive and Natural: According to the participants of the user study, the general concept of the application is easy to understand. The use of the *HTC Vive* hardware, the interaction with the virtual world and the operation of the user menu have all been rated as natural and intuitive as described in Section 5.3.

Immersion: Even though the design of the virtual environment is kept very minimalistic, it is still based on real world features. As described in Section 3.3.3, the world presents the user with a ground floor that matches the physical floor he is moving on. The 3D models are generated from actual scan data of human bodies and are textured to look lifelike. The application contains no unnecessary elements that distract the user from the virtual world or break his immersion. Even the *diegetic* user menu supports the immersion.

Consider Target Audience: The concept considered the needs of the target audience and the phases of learning that need to be fulfilled by an educational scenario. In Chapter 3 it has further been discussed how the application can be interesting for different stakeholders. If the concept is indeed valuable for the main target audience, the medical students, could not yet be evaluated as no respective people participated in the user study.

Accessibility: According to some participants, the application neither poses a problem for left-handed users nor users that suffer from a certain degree color vision deficiency. Even though the application is designed as a *room scale* experience, it is still possible to use it as a *seated* experience without any limitations.

Global Selection: The application provides the possibilities to select and grab object either with a laser or actual collision with the controller. As described in Section 5.3, the selection of small objects that are far away from the user still poses a problem for inexperienced users.

Precision: The use of a laser and the possibility to directly interact with virtual objects provides the user with a significant amount of precision. The assembly of different pieces uses a threshold, so that two pieces can be assembled, even if they are not perfectly aligned. This still gives the user an impression of precision.

Correctness: As described in Section 3.2.2, the approach that describes the solution space of each puzzle only delivers valid results.

Help Features: The application provides different feedback features that help the user to solve the puzzle task. The vibrotactile feedback appears to be the most relevant feature. For users that do not have existing anatomical knowledge, the *Ghost Copy* feedback poses a valid opportunity to learn the spacial relations between different structures. It could not yet be quantified if the other features are useful for users that already have the required anatomical knowledge and only need help with bringing the pieces precisely into *snapping distance.*

Phases of Learning: By including three different modes, the application satisfies all *Phases of Learning.* A user that studies anatomy can be considered as already interested in the subject of anatomy education. This covers the *Preparation* phase. This phase is further covered by the use of concurrent VR hardware. The following phases are each covered respectively by one the modes.

6 Conclusion

This thesis introduced an educational scenario that teaches spacial relations of anatomical structures to medical students. The scenario considered the use of a 3D puzzle in a virtual reality environment. Initially, the basic functionality of virtual reality and its history has been presented. Following this, the topic of anatomical education has been introduced. Thorough attention was paid to existing computer aided learning approaches that have been applied to the field of anatomical education. This included existing VR applications. Based on this, a concept for an educational scenario has been created that covers the *Phases of Learning* and relies on a virtual reality 3D puzzle to teach anatomical knowledge. Different interaction concepts for such an application have been discussed, as well as potential solutions for a user menu. Special focus has also been put on feedback features that help the user with solving the puzzle. This concept was then realized in a prototype application using the *Unity* game engine and the *HTC Vive*. Relevant aspects of this implementation have been briefly presented. The developed prototype has then been evaluated in a small user study. Due to the absence of members of the target audience, this study only resulted in a qualitative evaluation of the prototype and the underlying concept. The results and possible improvements have been presented and discussed.

The prototype that was developed as part of this thesis presents the user with the task of assembling virtual 3D models of anatomical structures. This task is based on the concept of real world jigsaw puzzles. To create a comprehensive educational scenario, the application features three modes that either provide knowledge to the user, let the user review his knowledge or let him apply his knowledge to an unknown scenario. The application additionally provides feedback features that help the user with successfully assembling the anatomical structures. As shown by the user study, the concept is generally met with positive feelings. The interaction with the virtual objects is precise and intuitive and operating the user menu poses no challenge for inexperienced users. The selection of small, far away objects is still a subject for improvement. As the prototype has not been tested with members of the target audience, another more thorough user study is required that specifically includes medical students that do not yet have extensive anatomical knowledge. The study must also run on a long-term to determine if potential educational benefits can actually be attributed to the proposed educational scenario.

6.1 Future Work

As of now, the concept of an educational scenario contains a basic amount of features that are required to teach anatomical knowledge with a 3D VR puzzle. One addition that the prototype might benefit from could be the inclusion of complete, comprehensive data of the human anatomy. To be more precise, instead of providing multiple 3D models, the system could provide a single model, where the user can then choose his area of interest from. The user could also be able to only puzzle certain structures, such as bones or internal organs. This could give the student the possibility to get a more comprehensive understanding of the spacial relations of certain parts and lets him additionally decide which information is relevant for him. The existing prototype could also be connected to an online medical databse to provide more information about the structures he is interacting with. This could help him further understand the purpose and functionality of each structure. If the topic of audio design is considered for the application, such information could be delivered as an acoustic stimuli. This might result in a deeper immersion into the system and have positive effects on the students learning behaviour. The students could also potentially be more motivated to use the application if more competitive elements, such as the results displays at the end of the *Training* and *Testing* modes, are added. In the same context, could it be interesting to provide the user with the possibility to share his experience on social media platform such as *Facebook*, *YouTube* and *Twitch*. This would enable the students to easily share and review their learning progress with each other. At this point it is also necessary to look into the possibilities of collaborative work in virtual reality. Two students could solve the same puzzle at the same time and while doing so present explanations and questions to each other. To make the concept more affordable and accessible for as many medical students as possible, it might be interesting to determine if it is possible to realize the concept with a smartphone-based HWD. This would also reduce the costs of such a system for the educational facilities. Finally, it can be valuable to determine if people that already work in healthcare environments are interested in such a system. Doctors could use the system to easily refresh their anatomical knowledge. Surgeons might even be interested in using the *Testing* mode for preparing surgical procedures or discuss necessary interventions with their patients. Not only the field of healthcare might be interested in the concept of a 3D VR puzzle. The application could be equally valuable for engineers or engineering students. In this case,it would be necessary to determine which requirements need to be considered for an application in the field of engineering.

Bibliography

[1] M. Ackerman, V. Spitzer, A. Scherzinger, and D. Whitlock. The visible human data set: an image resource for anatomical visualization. *Medinfo. MEDINFO*, 8:1195–1198, 1994.

[2] C. Adamczyk, M. Holzer, R. Putz, and M. R. Fischer. Student learning preferences and the impact of a multimedia learning tool in the dissection course at the university of munich. *Annals of Anatomy-Anatomischer Anzeiger*, 191(4):339–348, 2009.

[3] T. Akenine-Möller, E. Haines, and N. Hoffman. *Real-time rendering*. CRC Press, 2008.

[4] S. Alnassar, A. N. Alrashoudi, M. Alaqeel, H. Alotaibi, A. Alkahel, W. Hajjar, G. Al-Shaikh, A. Alsaif, S. Haque, and S. A. Meo. Clinical psychomotor skills among left and right handed medical students: are the left-handed medical students left out? *BMC medical education*, 16(1):97, 2016.

[5] D. S. Barry, F. Marzouk, K. Chulak-Oglu, D. Bennett, P. Tierney, and G. W. O'Keeffe. Anatomy education for the youtube generation. *Anatomical sciences education*, 9(1):90–96, 2016.

[6] S. Birr, J. Mönch, D. Sommerfeld, U. Preim, and B. Preim. The liveranatomyexplorer: a webgl-based surgical teaching tool. *IEEE computer graphics and applications*, 33(5):48–58, 2013.

[7] D. Bowman. Principles for the design of performance-oriented interaction techniques. *Handbook of Virtual Environments*, page 277, 2002.

[8] D. Bowman, E. Kruijff, J. J. LaViola Jr, and I. Poupyrev. *3D User Interfaces: Theory and Practice, CourseSmart eTextbook*. Addison-Wesley, 2004.

[9] D. A. Bowman, J. Chen, C. A. Wingrave, J. F. Lucas, A. Ray, N. F. Polys, Q. Li, Y. Haciahmetoglu, J.-S. Kim, S. Kim, et al. New directions in 3d user interfaces. *IJVR*, 5(2):3–14, 2006.

[10] D. A. Bowman and L. F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the symposium on Interactive 3D graphics*, pages 35–ff. ACM, 1997.

[11] H. Brenton, J. Hernandez, F. Bello, P. Strutton, S. Purkayastha, T. Firth, and A. Darzi. Using multimedia and web3d to enhance anatomy teaching. *Computers & Education*, 49(1):32–53, 2007.

[12] W. Bricken. Learning in virtual reality. 1990.

[13] V. Bruce, P. R. Green, and M. A. Georgeson. *Visual perception: Physiology, psychology, & ecology*. Psychology Press, 2003.

[14] H. Buchholz, J. Bohnet, and J. Dollner. Smart and physically-based navigation in 3d geovirtual environments. In *Information Visualisation, Proceedings. Ninth International Conference on*, pages 629–635. IEEE, 2005.

[15] A. Chang, S. O'Modhrain, R. Jacob, E. Gunther, and H. Ishii. Comtouch: design of a vibrotactile communication device. In *Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 312–320. ACM, 2002.

[16] L. Chittaro and R. Ranon. Web3d technologies in learning, education and training: Motivations, issues, opportunities. *Computers & Education*, 49(1):3–18, 2007.

[17] D. L. Choi-Lundberg, T. F. Low, P. Patman, P. Turner, and S. N. Sinha. Medical student preferences for self-directed study resources in gross anatomy. *Anatomical sciences education*, 2015.

[18] S. Craig, N. Tait, D. Boers, and D. McAndrew. Review of anatomy education in australian and new zealand medical schools. *ANZ journal of surgery*, 80(4):212–216, 2010.

[19] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the cave. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 135–142. ACM, 1993.

[20] C. R. Davis, A. S. Bates, H. Ellis, and A. M. Roberts. Human anatomy: Let the students tell us how to teach. *Anatomical sciences education*, 7(4):262–272, 2014.

[21] P. R. Desai, P. N. Desai, K. D. Ajmera, and K. Mehta. A review paper on oculus rift-a virtual reality headset. *arXiv preprint arXiv:1408.1173*, 2014.

[22] M. Estai and S. Bunt. Best teaching practices in anatomy education: A critical review. *Annals of Anatomy-Anatomischer Anzeiger*, 208:151–157, 2016.

[23] A. O. S. Feiner. The flexible pointer: An interaction technique for selection in augmented and virtual reality. In *Proceedings UIST*, pages 81–82, 2003.

[24] S. Fuhrmann and A. M. MacEachren. Navigation in desktop geovirtual environments: Usability assessment. In *Proceedings 20th ICA/ACI International Cartographic Conference*, pages 2444–2453, 2001.

[25] J. C. Greene, V. J. Caracelli, and W. F. Graham. Toward a conceptual framework for mixed-method evaluation designs. *Educational evaluation and policy analysis*, 11(3):255–274, 1989.

[26] F. G. Hamza-Lup, L. Davis, and O. A. Zeidan. Web-based 3d planning tool for radiation therapy treatment. In *Proceedings of the eleventh international conference on 3D web technology*, pages 159–162. ACM, 2006.

[27] C. Hand. A survey of 3d interaction techniques. In *Computer graphics forum*, volume 16, pages 269–281. Wiley Online Library, 1997.

[28] K. Hanson and B. E. Shelton. Design and development of virtual reality: Analysis of challenges faced by educators. *Educational Technology & Society*, 11(1):118–131, 2008.

[29] E. Harpstead, C. J. MacLellan, K. R. Koedinger, V. Aleven, S. P. Dow, and B. Myers. Investigating the solution space of an open-ended educational game using conceptual feature extraction. 2013.

[30] L. J. Hettinger and G. E. Riccio. Visually induced motion sickness in virtual environments. *Presence: Teleoperators & Virtual Environments*, 1(3):306–310, 1992.

[31] H. Hoffman and D. Vu. Virtual reality: teaching tool of the twenty-first century?. *Academic Medicine*, 72(12):1076–81, 1997.

[32] K.-H. Hohne, B. Pflesser, A. Pommert, M. Riemer, T. Schiemann, R. Schubert, and U. Tiede. A'virtual body'model for surgical education and rehearsal. *Computer*, 29(1):25–31, 1996.

[33] H.-M. Huang, U. Rauch, and S.-S. Liaw. Investigating learners' attitudes toward virtual reality learning environments: Based on a constructivist approach. *Computers & Education*, 55(3):1171–1182, 2010.

[34] I. Iacovides, A. Cox, R. Kennedy, P. Cairns, and C. Jennett. Removing the hud: the impact of non-diegetic game elements and expertise on player involvement. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, pages 13–22. ACM, 2015.

[35] Ergonomic Requirements for Office Work with Visual Display Terminals. Standard, International Organization for Standardization, Geneva, CH, 1998.

[36] S. Jayaram, H. I. Connacher, and K. W. Lyons. Virtual assembly using virtual reality techniques. *Computer-aided design*, 29(8):575–584, 1997.

[37] N. W. John. The impact of web3d technologies on medical education and training. *Computers & Education*, 49(1):19–31, 2007.

[38] E. O. Johnson, A. V. Charchanti, and T. G. Troupis. Modernization of an anatomy class: From conceptualization to implementation. a case for integrated multimodal–multidisciplinary teaching. *Anatomical sciences education*, 5(6):354–366, 2012.

[39] B. Jones, R. Sodhi, M. Murdock, R. Mehra, H. Benko, A. Wilson, E. Ofek, B. MacIntyre, N. Raghuvanshi, and L. Shapira. Roomalive: magical experiences enabled by scalable, adaptive projector-camera units. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 637–644. ACM, 2014.

[40] N. Katzakis, K. Seki, K. Kiyokawa, and H. Takemura. Mesh-grab and arcball-3d: Ray-based 6-dof object manipulation. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*, pages 129–136. ACM, 2013.

[41] W. Krueger Myron. Artificial reality, 1983.

[42] J. Lanier. Virtual reality: The promise of the future. *Interactive Learning International*, 8(4):275–79, 1992.

[43] J. J. LaViola Jr. Bringing vr and spatial 3d interaction to the masses through video games. *IEEE Computer Graphics and Applications*, 28(5), 2008.

[44] M. Li. System and method for 3-d medical imaging using 2-d scan data, 1996. US Patent 5,582,173.

[45] L. W. MacDonald. Using color effectively in computer graphics. *IEEE Computer Graphics and Applications*, 19(4):20–35, 1999.

[46] S. Marks, J. E. Estevez, and A. M. Connor. Towards the holodeck: fully immersive virtual reality visualisation of scientific and engineering data. In *Proceedings of the 29th International Conference on Image and Vision Computing New Zealand*, pages 42–47. ACM, 2014.

[47] T. H. Massie, J. K. Salisbury, et al. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*, volume 55, pages 295–300. Citeseer, 1994.

[48] D. Meier. *The accelerated learning handbook: A creative guide to designing and delivering faster, more effective training programs*. McGraw Hill Professional, 2000.

[49] M. Mine. Working in a virtual world: Interaction techniques used in the chapel hill immersive modeling program. *University of North Carolina*, 1996.

[50] R. Molich and J. Nielsen. Improving a human-computer dialogue. *Communications of the ACM*, 33(3):338–348, 1990.

[51] D. T. Nicholson, C. Chalk, W. R. J. Funnell, and S. J. Daniel. Can virtual reality improve anatomy education? a randomised controlled study of a computer-generated three-dimensional anatomical ear model. *Medical education*, 40(11):1081–1087, 2006.

[52] J. Nielsen. 10 usability heuristics for user interface design. *Nielsen Norman Group*, 1(1), 1995.

[53] J. Nielsen. Thinking aloud: The# 1 usability tool. *Nielsen Norman Group [online]. January*, 16, 2012.

[54] G. M. Nielson and D. R. Olsen Jr. Direct manipulation techniques for 3d objects using 2d locator devices. In *Proceedings of the 1986 workshop on Interactive 3D graphics*, pages 175–182. ACM, 1987.

[55] T. Okoshi. *Three-dimensional imaging techniques*. Elsevier, 2012.

[56] I. Pitt, B. Preim, and D.-I. S. Schlechtweg. An evaluation of interaction techniques for the exploration of 3d-illustrations. In *Software-Ergonomie'99*, pages 275–286. Springer, 1999.

[57] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79–80. ACM, 1996.

[58] B. Preim, A. Baer, D. Cunningham, T. Isenberg, and T. Ropinski. A survey of perceptually motivated 3d visualization of medical image data. In *Computer Graphics Forum*, volume 35, pages 501–525. Wiley Online Library, 2016.

[59] B. Preim, A. Raab, and T. Strothotte. Coherent zooming of illustrations with 3d-graphics and text. In *Graphics Interface*, volume 97, pages 105–113, 1997.

[60] J. Psotka. Immersive training systems: Virtual reality and education and training. *Instructional science*, 23(5-6):405–431, 1995.

[61] G. Ramsey-Stewart, A. W. Burgess, and D. A. Hill. Back to the future: teaching anatomy by whole-body dissection. *Medical Journal of Australia*, 193(11):668, 2010.

[62] J. Regian Jr, W. L. Shebilske, and J. M. Monk. A preliminary empirical evaluation of virtual reality as a training tool for visual-spatial tasks. Technical report, DTIC Document, 1993.

[63] F. Ritter, B. Berendt, B. Fischer, R. Richter, and B. Preim. Virtual 3d jigsaw puzzles: Studying the effect of exploring spatial relations with implicit guidance. In *Mensch & Computer 2002*, pages 363–372. Springer, 2002.

[64] F. Ritter, B. Preim, O. Deussen, and T. Strothotte. Using a 3d puzzle as a metaphor for learning spatial relations. In *Graphics Interface*, pages 171–178, 2000.

[65] F. Ritter, T. Strothotte, O. Dresden, and B. Preim. Virtual 3d puzzles: A new method for exploring geometric models in vr. *IEEE Computer Graphics and Applications*, 21(5):11–13, 2001.

[66] G. Roddenberry. Star trek - the next generation. *American Science Fiction TV Series, Episode*, 1x01 - Encounter at Farpoint, 1987.

[67] C. Rosse. The potential of computerized representations of anatomy in the training of health care providers. *Academic Medicine*, 70(6):499–505, 1995.

[68] G. Ruthenbeck, C. Carati, I. Gibbins, and K. Reynolds. A virtual reality 3d jigsaw for teaching anatomy. *interfaces*, 6:7, 2008.

[69] J. Schild, L. Bölicke, J. J. LaViola Jr, and M. Masuch. Creating and analyzing stereoscopic 3d graphical user interfaces in digital games. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 169–178. ACM, 2013.

[70] W. R. Sherman and A. B. Craig. *Understanding virtual reality: Interface, application, and design*. Elsevier, 2002.

[71] M. Slater, V. Linakis, M. Usoh, R. Kooper, and G. Street. Immersion, presence, and performance in virtual environments: An experiment with tri-dimensional chess. In *ACM virtual reality software and technology (VRST)*, volume 163, page 72. ACM Press New York, NY, 1996.

[72] V. M. Spitzer and D. G. Whitlock. The visible human dataset: the anatomical platform for human simulation. *The anatomical record*, 253(2):49–57, 1998.

[73] J. Starkweather. Computer-assisted learning in medical education. *Canadian Medical Association Journal*, 97(12):733, 1967.

[74] A. Stevenson. *Oxford Dictionary of English (3rd ed.)*. Oxford University Press, 2010.

[75] J. Strobel and G. W. Zimmerman. Effectiveness of paper, vr and stereo-vr in the delivery of instructions for assembly tasks. 2010.

[76] K. Sugand, P. Abrahams, and A. Khurana. The anatomy of anatomy: a review for its modernization. *Anatomical sciences education*, 3(2):83–93, 2010.

[77] I. E. Sutherland. The ultimate display. In *Proceedings of the IFIP Congress*, pages 506–508, 1965.

[78] I. E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM, 1968.

[79] D. S. Tan, G. G. Robertson, and M. Czerwinski. Exploring 3d navigation: combining speed-coupled flying with orbiting. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 418–425. ACM, 2001.

[80] H. Z. Tan, N. I. Durlach, W. M. Rabinowitz, C. M. Reed, and J. R. Santos. Reception of morse code through motional, vibrotactile, and auditory stimulation. *Attention, Perception, & Psychophysics*, 59(7):1004–1017, 1997.

[81] G. Taxén and A. Naeve. A system for exploring open issues in vr-based education. *Computers & Graphics*, 26(4):593–598, 2002.

[82] U. Tiede, M. Bomans, K. H. Höhne, A. Pommert, M. Riemer, T. Schiemann, R. Schubert, and W. Lierse. A computerized three-dimensional atlas of the human skull and brain. In *Neuroimaging I*, pages 185–197. Springer, 1996.

[83] A. Van Dam. Post-wimp user interfaces. *Communications of the ACM*, 40(2):63–67, 1997.

[84] K. Weissgerber, G. B. Lamont, B. J. Borghetti, and G. L. Peterson. Determining solution space characteristics for real-time strategy games and characterizing winning strategies. *International Journal of Computer Games Technology*, page 3, 2011.

[85] L. B. Wexner. The degree to which colors (hues) are associated with mood-tones. *Journal of applied psychology*, 38(6):432, 1954.

[86] C. Wilkes and D. A. Bowman. Advantages of velocity-based scaling for distant 3d manipulation. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 23–29. ACM, 2008.

[87] P. Wright, D. Mosser-Wooley, and B. Wooley. Techniques & tools for using color in computer interface design. *Crossroads*, 3(3):3–6, 1997.

[88] S.-T. Wu, M. Abrantes, D. Tost, and H. C. Batagelo. Picking and snapping for 3d input devices. In *Computer Graphics and Image Processing*, pages 140–147. IEEE, 2003.

Appendix

The appendix contains the questionnaire used for the user study, as described in Section 5.2. Each text is presented in German and English.

3D VR Puzzle

Teilnehmer (*Participant*): _______________

Alter (*Age*): ___ Geschlecht (*Sex*): ____ Händigkeit (*Handedness*): _____________

Tätigkeit (Occupation): __________ Sehschwächen (Color Vision Deficiency): ______________

Ihre erhobenen Daten dienen ausschließlich wissenschaftlichen Zwecken und werden vollkommen anonym behandelt.
Your collected data is solely used for scientific purposes and is treated utterly anonymous.

Schätzen Sie Ihre Erfahrung im Umgang mit Computern ein.
(*Rate your experience with using computers.*)

keine				weit fortgeschritten
☐	☐	☐	☐	☐
none				*highly advanced*

Schätzen Sie Ihre Erfahrung im Umgang mit VR-Anwendungen ein.
(*Rate your experience with using VR-applications.*)

keine				weit fortgeschritten
☐	☐	☐	☐	☐
none				*highly advanced*

Schätzen Sie Ihre Erfahrung im Umgang mit 3D-Eingabegeräten ein.
(*Rate your experience with using 3D input devices.*)

keine				weit fortgeschritten
☐	☐	☐	☐	☐
none				*highly advanced*

Schätzen Sie Ihre Erfahrung im Umgang mit medizinischen 3D-Daten ein.
(*Rate your experience with using 3D medical data.*)

keine				weit fortgeschritten
☐	☐	☐	☐	☐
none				*highly advanced*

Teilnehmer (*Participant*): _______________

Tutorial

Wie haben Sie die grundlegenden Funktionen der Applikation verstanden?
(*How did you understand the basic functions of the application?*)

gar nicht				sehr gut
☐	☐	☐	☐	☐
not at all				*very well*

Schätzen Sie die Schwierigkeit des Tutorials ein.
(*Rate the difficulty of the tutorial*)

zu leicht		angemessen		zu schwer
☐	☐	☐	☐	☐
to easy		*appropriate*		*to difficult*

Schätzen Sie die Nützlichkeit des Tutorials ein.
(*Rate the usefulness of the tutorial.*)

unnütz				sehr nützlich
☐	☐	☐	☐	☐
useless				*very useful*

Bewerten Sie das Tutorial (von 1 = sehr gut bis 6 = sehr schlecht)
(*Rate the tutorial (from 1= very good to 6 = very bad)*)

____

Teilnehmer (*Participant*): _______________

Exploration

Schätzen Sie ein, wie Sie den Nutzen des Modus „Exploration“ verstanden haben.
(*Rate how you understood the use of the mode “Exploration”.*)

gar nicht | sehr gut

☐ ☐ ☐ ☐ ☐

not at all | *very well*

Schätzen Sie ein, wie Sie sich mit dem medizinischen 3D Model vertraut machen konnten.
(*Rate how you could familiarize yourself with the medical 3D model.*)

gar nicht | sehr umfangreich

☐ ☐ ☐ ☐ ☐

not at all | *very extensive*

Schätzen Sie ein, wie der Modus es Ihnen ermöglicht hat räumliche Beziehungen der einzelnen Teile zu verstehen.
(*Rate how the mode made it possible for you to comprehend the spacial relations of the pieces.*)

gar nicht | sehr umfangreich

☐ ☐ ☐ ☐ ☐

not at all | *very extensive*

Schätzen Sie ein, wie der Modus es Ihnen ermöglicht hat die Namen der einzelnen Teile zu erlernen.
(*Rate how the mode made it possible for you to learn the names of the individual pieces.*)

gar nicht | sehr gut

☐ ☐ ☐ ☐ ☐

not at all | *very well*

Teilnehmer (*Participant*): _______________

Training

Schätzen Sie ein, wie Sie den Nutzen des Modus „Training“ verstanden haben.
(*Rate how understood the use of the mode “Training”.*)

gar nicht				sehr gut
☐	☐	☐	☐	☐
not at all				*very well*

Schätzen Sie ein, wie Sie die Aufgabe bewältigen konnten.
(*Rate how you could solve the task.*)

gar nicht				sehr gut
☐	☐	☐	☐	☐
not at all				*very well*

Schätzen Sie ein, wie die Applikation Sie unterstützt hat die Aufgabe zu bewältigen.
(*Rate how the application supported you in solving the task.*)

gar nicht				sehr gut
☐	☐	☐	☐	☐
not at all				*very well*

Schätzen Sie die Schwierigkeit der Aufgabe ein.
(*Rate the difficulty of the task*)

zu leicht		angemessen		zu schwer
☐	☐	☐	☐	☐
to easy		*appropriate*		*to difficult*

Teilnehmer (*Participant*): _______________

Testing

Schätzen Sie ein, wie Sie den Nutzen des Modus „Testing“ verstanden haben.
(*Rate how you understood the use of the mode “Testing”.*)

gar nicht				sehr gut
☐	☐	☐	☐	☐
not at all				*very well*

Schätzen Sie ein, wie Sie die Aufgabe bewältigen konnten.
(*Rate how you could solve the task.*)

gar nicht				sehr gut
☐	☐	☐	☐	☐
not at all				*very well*

Schätzen Sie ein, wie die Applikation Sie unterstützt hat die Aufgabe zu bewältigen.
(*Rate how the application supported you in solving the task.*)

gar nicht				sehr gut
☐	☐	☐	☐	☐
not at all				*very well*

Schätzen Sie die Schwierigkeit der Aufgabe ein.
(*Rate the difficulty of the task*)

zu leicht		angemessen		zu schwer
☐	☐	☐	☐	☐
to easy		*appropriate*		*to difficult*

Teilnehmer (*Participant*): _______________

Sonstiges (*Miscellaneous*)

Schätzen Sie den Grad der Übelkeit ein, den Sie beim Nutzen der Applikation erlebten.
(*Rate the grade of nausea you experienced while using the application.*)

keine		leichte Übelkeit		unerträgliche Übelkeit
☐	☐	☐	☐	☐
none		*light nausea*		*unbearable nausea*

Schätzen Sie den Grad der Ermüdung ein, den Sie beim Nutzen der Applikation erlebten.
(*Rate the grade of exhaustion you experienced while using the application.*)

keine		Leichte Erschöpfung		vollkommene Erschöpfung
☐	☐	☐	☐	☐
none		*light exhaustion*		*complete exhaustion*

Schätzen Sie Grad der Freude ein, den Sie bei der Nutzung der Applikation hatten.
(*Rate the grade of joy you experienced while using the application.*)

keine		Spaß		große Freude
☐	☐	☐	☐	☐
none		*fun*		*intense joy*

Schätzen Sie ein, wie es Ihnen gelang die Puzzleteile in der virtuellen Umgebung auszuwählen.
(*Rate how you were able to select the puzzle pieces in the virtual environment.*)

unmöglich				sehr leicht
☐	☐	☐	☐	☐
impossible				*very easy*

Schätzen Sie ein, wie es Ihnen gelang die Teile in der virtuellen Umgebung zusammen zu setzen.
(*Rate how you were able to connect the pieces in the virtual environment.*)

unmöglich				sehr leicht
☐	☐	☐	☐	☐
impossible				*very easy*

Schätzen Sie ein, wie es Ihnen gelang den Fortschritt Ihres Puzzles zu erkennen.
(*Rate how you were able to tell the progress of your puzzle.*)

unmöglich				sehr leicht
☐	☐	☐	☐	☐
impossible				*very easy*

Wie nützlich fanden Sie die Art der Benutzeroberfläche?
(*How usefull do you think this type of user interface is?*)

unnütz				sehr nützlich
☐	☐	☐	☐	☐
useless				*very usefull*

Schätzen Sie ein, wie gelang es Ihnen fiel die Benutzeroberfläche zu bedienen.
(*Rate how you were able to use the user interface.*)

unmöglich				sehr leicht
☐	☐	☐	☐	☐
impossible				*very easy*

Teilnehmer (*Participant*): _______________

Bewerten Sie die Feedback Features bezüglich ihrer Nützlichkeit.
(*Rate the feedback features in terms of usefulness.*)

	unnütz				sehr nützlich
Display Values	☐	☐	☐	☐	☐
Tinting	☐	☐	☐	☐	☐
Bar Strings	☐	☐	☐	☐	☐
Ghost Copy	☐	☐	☐	☐	☐
Vibration	☐	☐	☐	☐	☐
	useless				*very usefull*

Bewerten Sie die Feedback features bezüglich ihrer Verständlichkeit.
(*Rate the feedback features in terms of comprehensibility.*)

	unverständ lich				eindeutig
Display Values	☐	☐	☐	☐	☐
Tinting	☐	☐	☐	☐	☐
Bar Strings	☐	☐	☐	☐	☐
Ghost Copy	☐	☐	☐	☐	☐
Vibration	☐	☐	☐	☐	☐
	incompreh ensible				*clear*

Welches Feedback bzw. welche Feedbackkombination half Ihnen am Meisten?
(*Which feedback or which feedback combination was most helpful to you?*)

____

Bewerten Sie die Applikation (von 1 = sehr gut bis 6 = sehr schlecht)
(*Rate the application (from 1= very good to 6 = very bad)*)

____

Eigenständigkeitserklärung

Hiermit bestätige ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken (dazu zählen auch Internetquellen) entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht.

Pohlandt, Daniel	Ort	Datum