Otto-von-Guericke University Magdeburg

Faculty of Computer Science
Department of Simulation and Graphics

Bachelor Thesis

Enhancing 3D Interaction for Broad Audiences: Exploring Slider-based Rotation Affordance Techniques

Author:
Felix Schwarz

March 7, 2024

Advisers:

PD Dr.-Ing. habil.
Monique Meuschke

Department of Simulation and Graphics
Faculty of Computer Science
Otto-von-Guericke University
Universitätsplatz 2
39106 Magdeburg, Germany

Prof. Dr.-Ing. habil.
Bernhard Preim

Department of Simulation and Graphics
Faculty of Computer Science
Otto-von-Guericke University
Universitätsplatz 2
39106 Magdeburg, Germany

Schwarz, Felix:
Enhancing 3D Interaction for Broad Audiences:
Exploring Slider-based Rotation Affordance Techniques
Bachelor Thesis, Otto-von-Guericke University
Magdeburg, 2024.

Contents

Abstract

The goal of this work was to enhance 3D interaction for broad audiences by exploring slider-based rotation affordance techniques.

Users are generally not familiar with virtual 3D interaction because a screen is often understood as a 2D plane with no depth. Reading and scrolling through text, images, and videos is the main use of a screen. This makes for a challenge if frontend developers want to get the user to understand that virtual 3D interaction is possible.

A meaningful use of virtual 3D interaction is to see an object from more sides than just the front. Virtual 3D rotation around all three axes is a complex task for a non-expert user. Therefore the focus of this research was only on rotation around the vertical axis.

The chosen base affordance to facilitate interaction was a slider. A slider is a rather familiar UI element and has room for adaptation when designing variants.

Of all chosen variants, a curved slider was best understood by the users to enable rotation of the shown object and animating the object with the slider by turning the object and dragging the slider was rated most engaging by the users.

The implementation of the user study also serves as a proof of concept for creating an online user study with 3D interaction using Unity UI Toolkit.

1 Introduction and Motivation

1.1 Motivation

With current hardware 3D interaction is used more widely than ever. Online shops, museums, and professional tools reach more users. Many of them do not have experience in such an environment. One example is the use of actual research data in visualizations for museums. There users can explore the data like a scientist would. Generally speaking, digital 3D interaction is highly unusual for a general audience because if people interact with digital screens they expect 2D interaction as the screen is just a surface. But even then the expectation is not to move around 2D objects. Most interaction is viewing websites with texts, pictures, and videos. The interaction is scrolling through the website, clicking links, and filling out forms. The user usually does not interact directly with the text or picture they are viewing.

In a prior user study participants had a particular task to estimate the growth rate of aneurysms and tumors (MITTENENTZWEI et al., 2023). In more abstract terms this means they had to estimate the growth of 3D virtual objects. To support the participants with the task it was explained that they could rotate the objects for better understanding. However many users did not make use of this or stopped interacting after scene changes. Although it would have made it easier to answer the questions. The authors assumed that rotating a digital 3D object is not intuitive for people without experience in such an environment.

There is a lot of research on professional tools with 3D interaction. But how to bring the best 3D interaction experience to an untrained user is not well-researched.

When the user enters an environment where the 3D interaction is only a part of the whole experience, it is important to make him intuitively understand what can be done in the scene without exposing him to unnecessary cognitive load. A common interaction technique is the rotation of an object. This interaction technique is the focus of this research.

1.2 Goal of this Thesis

This research focuses on 3D interaction with mouse input. Specifically on rotating the object horizontally. The user should understand the possibility of this type of interaction intuitively and independent of the object being rotated. This also translates to the user quickly learning what interaction is possible so the user can use it in the next scene.

To facilitate intuitive understanding it is important to grab the user's attention with cues that are perceived as quickly as possible. These cues fall in the category of preattentive vision.

After the user's attention is on the interactive element the user gets an intuitive understanding of the interaction possible which is called "affordance". The more the understanding of the user matches the actual interaction possible, the better the so-called "affordance".

The focus of this research is slider-based rotation affordance techniques. A horizontal slider is a good base to explore these affordances because it is a familiar interaction element with room for adaptation and modification. A slider also provides useful constraints for the user.

Specific research questions are:

- Which affordance technique for rotation with mouse input is perceived most accurately and with the highest confidence by the user?

- What are the strengths and weaknesses of each affordance technique according to the user?

- How does the user rank the presented affordance techniques for slider-based rotation?

1.3 Contributions of this Thesis

The following list describes the contributions of this thesis:

- A structured approach for exploring slider-based rotation affordance techniques for mouse input. The Feature Integration Theory and Gestalt laws provide this structure as well as statistics of attention cues in data videos.

- The design and results of a user study exploring the created slider-based rotation affordance techniques with a focus on an order of study parts that gains the most insights. The study had four parts. The first part focused on the understanding of the affordances by the user, the second part focused on the usage impact of the affordances, the third part focused on strengths and weaknesses of the affordances according to the user and the last part focused on ranking the affordances.

- The implementation of a complete pipeline from user input to data analysis without the need to manually copy parts of the data. This is done by using Unity, MariaDB, PHP, and Tableau Desktop. Depending on the database setup, it has to be copied once for local use.

1.4 Structure of this Thesis

The following list provides an overview of the chapters:

- Chapter 2 introduces theory on visual perception which is used to create affordances in a structured way. This includes a definition for "affordance", preattentive vision, the Feature Integration Theory and selected Gestalt laws.

- Chapter 3 shows related work. There a focus is on the benefit of interactive visualizations, what is important when designing interaction for a wide audience and what attention cues are most popular in data videos. The top attention cues are also used for the creation of the affordances.

- Chapter 4 describes the processes of designing the affordances, the user study, and the implementation. This includes the application of the theory in Chapter 2 and various figures about the technical architecture of the user study.

- Chapter 5 deals with the evaluation of the data collected by the study and the verbal feedback I received.

- Chapter 6 provides a summary of the work, further research possibilities, and technical improvements to the process of implementing an online user study using Unity.

2 Selected Theory of Visual Perception

In this chapter, I will explain the terms "affordance" and "preattentive vision" followed by the theories that I used to create the different slider-based affordance techniques. The theories stem from the field of psychology and influence the practical fields of visualization and design.

2.1 Affordance

The term "affordance" describes a perceivable possibility for action in the real world. The perception of an affordance is immediate, meaning not built from the bottom up from sensory clues but instead from the top down from experience (WARE, 2021b).

2.1.1 Real World

An example of an affordance is a door handle. A person readily perceives a door handle without the need to recognize specific aspects of the handle: material, color, or exact measurements. In the mind of the perceiving person the door handle is a learned concept that captures varieties of door handles. Depending on the culture a person is primed to recognize different door handles faster than others. Inherent in the perception of the door handle is the function or possibility for action. The door handle is recognized as such not only because of its shape but also because of the association of the shape with the action of opening a door. Then after opening the door using the handle the concept of the door handle gets updated in a person's mind, associating all sensations which are primarily experienced while opening the door: pressure needed to pull down the handle and push the door open. If for any reason the door does not open or maybe opens the other way, this feedback will also be integrated and used in the future for opening similar door handles under similar circumstances.

This leads to what constitutes a good affordance and what constitutes a bad affordance. It is quickly recognized that when the perceived possibility for action fits with the real action possible, a good affordance is observed. However, as presented, affordance is based on the individual experience of a person. This creates a problem because an affordance might be good for one person or a group of people but bad for others.

A good way to improve an affordance for a big group of people is to label the affordance with additional information. For example, writing on the door "push" or "pull". This can help a person but also introduce more confusion if the labeling does not fit. Overall a person then has to use a more active approach to understanding the affordance but might do so with higher accuracy.

So the goal for designing affordances is that they are quickly and accurately understood by the audience.

2.1.2 UI Design

Translating the affordance theory to UI design comes with its challenges (WARE, 2021b).

First and foremost the physicality of an affordance gets lost. Even pressing a button on a touch screen is not the same as using a real-world button. This also raises the question of what a real-world button affordance is. Unlike a door handle a button is already an abstraction from the specific action. Usually, the context identifies the action of the button. A button next to a door is most likely a light switch, on a coffee maker it is the on-switch. The context cues in a digital interface are sparse in comparison to the real world and interaction is more indirect. For example, using a slider to rotate an object is very different from rotating an object with your own hands in the real world. However, the opportunities for digital affordances are much greater. Rotating very small or very large digital representations of real-world objects is much more feasible on a digital device.

Since real-world affordances do not only consist of door handles and digital affordances do not only consist of sliders for rotation, it is unavoidable that not every affordance is understood by everyone the same. There are many variations for each affordance and everyone has their own prior experience with them. So creating affordances for a general audience is quite difficult. But an affordance correctly understood by an inexperienced user will generally be also understood by everyone with more experience. So assessing the baseline of understanding for a particular affordance is a good approach to finding a good affordance. In this research variations of slider-based affordances are investigated to see which one could be a baseline.

2.2 Preattentive Vision

Seeing and understanding are complex human abilities. To see the right thing and see it first was vital for our ancestors. Many things are not consciously processed by our brains. In visualization, we use the term "preattentive" to indicate a process that is not conscious or happens before conscious attention. So preattentive vision is the process that decides what to look at closer after a first glance of a picture, visualization, or interface. For this process to be so fast parallel processing is required in the brain (WARE, 2021a).

Examples to illustrate this process are used in psychology experiments. There screens with a lot of the same shapes are displayed with only one shape differing in one aspect for example color. Then the differing shape has to be identified. This happens very fast if the shapes are not complex. A red dot in a sea of blue dots pops out immediately and the observer does not have to pay much attention to this except for looking at the screen.

So aiming to construct a user interface where the important information pops out is the goal because then the user has to put little effort into the search and can focus better on the task at hand.

2.3 Feature Integration Theory

An integral theory that puts more structure to the idea of preattentive vision is the Feature integration theory (FIT).

Since there are features that are preattentive understood FIT categorizes these features in feature maps. These feature maps are color, orientation, movement, curvature, and depth. For some features, there are concrete specializations in the brain that enable the parallel processing of these feature maps. After processing each feature map the found information is recombined in the mind. If there is more than one feature map required to identify a stimulus this step requires conscious focus (POLLMANN, 2020).

An issue with this theory is that even complex features can be processed preattentive if the observer has a lot of experience with that specific feature (POLLMANN, 2020).

For example, if there is a picture of "X"s and "O"s with the same color and only one element is from a different color the search for this one different element happens through one feature map and thus is processed preattentively. If however "X"s and "O"s are of different colors and the element to search for is an "X" with the color of the "O"s the search cannot be processed through only one feature map. This requires conscious attention and more time because two feature maps have to be recombined to find the element.

Overall the feature maps are a good starting point to create visual cues that pop out to the user.

2.4 Gestalt Psychology

Previously explained were theories about a top-down and a bottom-up approach to visual perception. These theories can be connected to Gestalt psychology. Perceiving just a difference in color or a whole different object is connected by pattern recognition to complete the process of visual perception. Gestalt psychology describes a set of patterns or so-called laws to categorize different types of patterns that we readily perceive (WARE, 2021c).

These laws provide another point of reference for designing good user interfaces since user interfaces consist of multiple objects and their relation to the design space.

I will present a subset of the Gestalt laws which are relevant to this research.

2.4.1 Law of Proximity

Objects that are close to each other are perceived as a group and space with similar object density is also perceived as a group (WARE, 2021c).

For example, figures in the starting position on a digital chess board create two visual groups and data points in a 2D graph are easily identified as distinct clusters.

Generally speaking, members of a group have something in common. In UI design this can be used to group elements of similar semantic meaning or elements for interaction and the object on which the interaction is applied. So the user has a better intuition of how to interact with the UI.

2.4.2 Law of Past Experience

Objects that in the past built a group in the user's mind are often perceived again as such if the overall experience is not completely different (TODOROVIC, 2008a).

Scroll bars have very different designs but the user expects it on the right side of the view and as a functional group with the view for scrolling up and down. So although the design is different the user infers the grouping from experience.

Practically this translates to building UIs with the user's experience in mind. A button on the top right corner of a window typically closes the window. Anything that goes against the user's experience requires effort to learn and slows down the user in his task.

2.4.3 Law of Closure and Common Region

Regions are defined by contours. If contours are incomplete they will be mentally completed to create a visual image that makes the most sense. Additionally, regions defined by complete contours have a greater visual effect than the law of proximity. (WARE, 2021c).

If a blue rectangle overlaps a red circle the circle is still perceived as a whole despite no visible evidence. Putting a rectangle around part of a cluster of data points immediately forms a stronger grouping than the cluster itself.

This law is important for UI design because it allows for a very clear grouping of different objects not depending so much on their distance.

2.4.4 Law of Common Fate

Objects that move at the same time build a visual group. This is true even if there is a large distance between the objects (TODOROVIC, 2008b).

An example is a flock of birds in the sky. There are many individual birds but they are all moving at the same time and in the same direction. So a group is recognized.

This law is interesting for UI design for interaction because it allows for an intuitive explanation of interaction controls by moving the control and showing the effect at the same time. Thus the user can quickly understand their connection.

3 Related Work

In this chapter, related work is presented. This includes the benefits and challenges of interaction for broader audiences, principles for good 3D rotation, and attention cues used in data videos.

3.1 Benefit of Interactive Visualization

YNNERMAN et al. (2016) describe the process, concepts, and benefits of bringing large data sets of CT scanned data of mummies to a general audience for interaction via a big touch screen in a museum. Although the interaction works with a touch screen, the concepts and benefits are relevant to this research as the target audience is similar. After installing their interactive visualization an increase of visitor dwell time by 40% was observed. The visitor time observing the original display also increased alongside the digital display. They argue that the interactive experience is more engaging than passively absorbing information in a museum. Objects behind can be accessed virtually and the visitor can even look inside the object which is very similar to what research scientists do. Thus raising the authenticity of the information presented. Some of their concepts are also relevant to this research. One focus was the time to "walk up and use" must be close to zero which is akin to our definition of affordance. The interaction has to be perceived as free but also reasonably constrained so the user is optimally guided. Icons on the screen also should not divert too much attention away from the main object.

3.2 Designing Visualizations for Broad Audiences

MA et al. (2012) explore in their work scientific storytelling using scientific data. They explained some results of discussions from a workshop about storytelling with data. A difficulty for the experts was determining the level of freedom for users compared to proper guidance through the data. A solution was to present data with the most important data points readily visible and allow for further exploration from the initial setup. They acknowledged the difficulty for non-expert users viewing scientific visualizations. This challenge is apparent in museums trying to enhance their visualizations with interaction. They want to solve this by including a narrative and interaction possibilities as a result improving comprehensibility and engagement. They ask for support to better reach the general public with the stories scientists want to tell.

BESANÇON et al. (2021) surveyed state-of-the-art spatial interfaces for 3D visualization and introduced a taxonomy of interaction techniques for 3D visualization. This research can be placed in the taxonomy in "Task Group 1: Volumetric view and object manipulation including clipping". They also acknowledge that research is needed to identify interaction challenges when dealing with a diverse group of users. Interaction techniques can help convey scientific knowledge and improve engagement in learning.

SARIKAYA et al. (2019) investigated dashboard designs and the literature around dashboards. Dashboards are visual displays of data to enhance understanding of the data for different kinds of audiences. There are visual and functional dashboards where functional dashboards enable some level of interaction. They characterized dashboards by different features with one of them being interactivity. They were aware of the challenges of presenting dashboards to casual users but were still surprised that in their terms "simple" features like filters were not useful for a casual audience. That gave them an understanding of why dashboard designers were so focused on reducing complexity for non-experts. They also concluded that fixed designs for all user groups cannot adequately support all users of different abilities.

GARRISON et al. (2021) explored the visual communication of biomedical processes. They used different presentations of these processes. The presentations differed in model and visual abstraction. One of the subtasks that expert and non-expert users had to do was choosing keywords to describe top and bottom choices for all scenarios. These were represented as a word cloud in the paper. In this research, a similar task was done by the users. By choosing out of a set of keywords for each affordance the strengths and weaknesses of each should be revealed. The choice of keywords is inspired by their work but also adapted to slider-based interaction.

3.3 Research on Interaction

BADE et al. (2005) compared different mouse-based interaction techniques for 3D rotation and shared design principles which when applied make the interactive rotation "predictable and pleasing". They investigated rotation along all axes while this research focuses on horizontal rotation only but the principles can still be applied. The first principle is "Similar actions should provoke similar reactions." In this research the slider always creates the same rotation. The second principle is "Direction of rotation should match the direction of 2d pointing device movement". Here moving the slider in one direction matches the rotation direction. The third principle is "3d rotation should be transitive". Since the rotation enabled by the slider is only along one axis, this principle is not violated. The fourth and last principle is "The control-to-display (C/D) ratio should be customizable". The goal of this research is more on understanding than on interaction, so there was no setting for the user to set the C/D ratio. Of the different techniques investigated the two-axis valuator trackball was the best. Their horizontal rotation was mapped to mouse movement about the up-vector of the virtual camera. The same is applied here to the slider.

AALTONEN et al. (1998) looked into how users interact with pull-down menus to reach their desired item. They recorded the mouse cursor and the point of gaze. Analyzing these recordings together they found different types of users. Some used the mouse cursor as a reading stick while others placed the cursor on the most likely candidate or only searched with their eyes. This makes it clear that not all users do not explore everything on the screen with the cursor. There has to be some guidance to make the user interact with something on the screen. So knowing what is an interactive element and what it does without actually trying it out is important to the user.

3.4 Attention Cues

AMINI et al. (2015) systematically explored data videos. Data videos are videos where data is visualized and presented to a broader audience. They identified features and how often they appear in these videos. They also observed experts designing storyboards for these videos and summarized their findings to help a wider audience create data videos. The observed features in the videos are separated into two categories: visualization type and attention cues. Attention cues bring the user's attention to a specific part of the video to then show a visualization type. The three most used attention cues were animation (35%), gradual text (18%), and highlighting (12%) while the usage of the other attention cues was under 6% for each. So the top three are the most prominent and therefore incorporated in this research.

4
Online User Study with 3D Interaction

In this chapter, I will explain the process of creating slider-based rotation affordances, the creation of attention cues, how the study was developed and the implementation of it. The frontend was implemented with Unity UI Toolkit and the backend was designed with phpMyAdmin. PHP scripts made the connection between the frontend and backend. Evaluation was supported by Tableau Desktop.

4.1 Study Design

The study was designed by discussing different approaches with my supervisors. The goal was to improve 3D interaction for inexperienced users as they do not expect interaction to be possible by presenting only a 3D object on a 2D screen. We discussed different ideas for affordances and parts of the study.

4.1.1 Creating Slider-based Rotation Affordances

We had the base idea of slider controls for interaction. Sliders are a rather natural control element as they are often used in other contexts as well. Other contexts include volume and brightness controls. Before we settled on slider-based affordances, I presented different ideas with more varied components.

One idea was to use a moving mouse symbol on the object while the object was moving in the same direction as the symbol. As the user starts interaction by clicking on the object, the mouse symbol disappears and the animation stops.

A very different idea was to use a kind of teaching video to let the user understand the interaction possibilities. In the video, a user interacts with a mouse and rotates an object on a screen. The video shows the hand of the user on the mouse as well as what is happening on the screen. The video could have been shown over the whole screen, and then the user could click to minimize it in some corners of the screen, so the user could still have a look while learning the interaction.

Another idea was to let the object rotate on its own if the user's mouse is over the interactive area. By clicking while in the interactive area, the rotation of the object stops. Then the user can drag the cursor outside the interactive area to keep the desired rotation.

An idea to grab the attention was to create a blinking text above the object which states something like "Rotate the object by dragging on the object".

A more gamified idea was to create a small rectangle where the user can put the cursor and after an animation the cursor is locked in place and any mouse movement now rotates the object. When the user holds a mouse button down for a short time, the cursor is freed from the looked-in place and can be used regularly again.

We concluded to create comparable affordances and get better insights into concrete properties that differentiate the affordance to the user, it is best to use a simple base case and modify it only partially. Then we can better conclude what exact part of the affordance is an improvement or a downgrade to the base case.

The chosen base case is the slider because of its potential to be adapted in different ways. As the abstract object to rotate, we chose a cube. It is simple but also has a clear 3D profile when rotated. The base affordance as well as the variants are created by applying the theories and insights presented in the previous chapters.

So the teaching video, auto rotation on mouse over and the mini game idea did not make it to the study. The other ideas were used in one way or another.

Figure 4.1 shows the first step of constructing the base affordance. There is already a theory applied. First, the law of proximity between the object and the slider. They are close and build a group for the user. Second, the law of experience is also applied. While expecting an inexperienced user

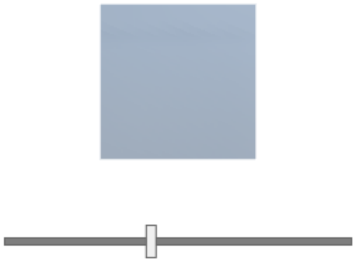


Figure 4.1: Constructing the base slider affordance with rotation object step 1.

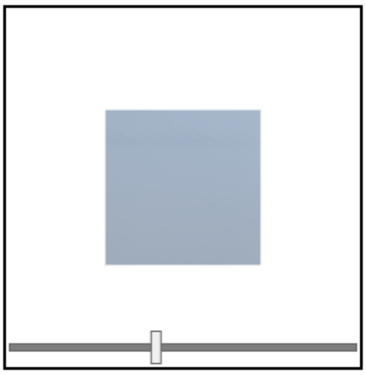


Figure 4.2: Constructing the base slider affordance with rotation object step 2.

to understand that rotation is enabled by the slider is asking for much, the experience still applies to the slider itself. The user should know from experience how a slider functions. The user can drag the vertical bar along the horizontal bar.

Generally speaking, the affordance is part of a screen with other things showing. To separate it more strongly from different objects on the screen, we can apply the law of closure and common region. By enclosing the affordance with the object in a rectangular border, shown in Figure 4.2, the connection of both as well as the separation to other objects on the screen is even more clear. It also creates a more pleasing visual because the user understands where the scene of the object starts and ends. A sense of figure and ground is conveyed.

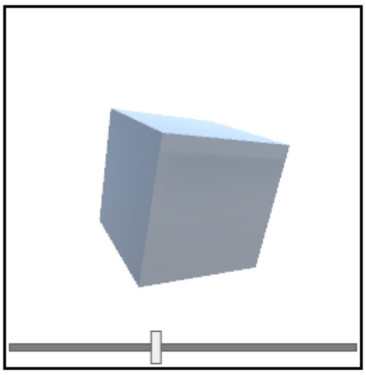


Figure 4.3: Constructing the base slider affordance with rotation object step 3 (nickname: standard).

MA et al. (2012) described the need to present an interactive visualization in a way such that the most important information is already displayed without the user needing to interact. The user is guided through the interaction but also perceives the interaction as free. That is also in line with the recommendations of YNNERMAN et al. (2016). In this case, only a cube is displayed. But even a cube can presented to reveal as much information about itself as possible. For this reason, as well as making all sides visible by using only one rotation axis, I tilted the cube along two axes. That is shown in Figure 4.3. Rotating an object around all three axes is a complicated task for a non-expert user. So rotation is fixed on one axis.

The next step was to create variants of these base affordances to get insights as to what property helps the user best understand the affordance of rotating the object with a slider. I proposed to use the feature maps of the Feature Integration Theory as described in section 2.3 to approach the construction of variants in a structured way. The feature maps are processed preattentively and thus give the user insights which do not need conscious attention. The feature maps are color, orientation, curvature, depth, and movement. Using color does not improve affordance, as the color of a slider does not translate to the user gaining more understanding of a rotation affordance. A color does not give hints for moving an object in a particular way. We also separated movement to the creation of attention cues. The movement has a much bigger impact than shapes on the attention. That leaves orientation, curvature, and depth for static variants of the base affordance. The resulting affordance variants are shown in Fig-

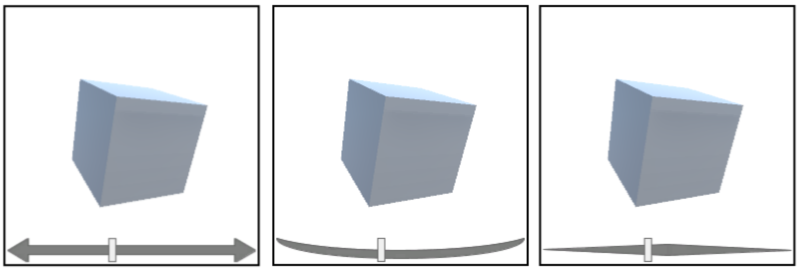


Figure 4.4: Static slider affordances variants with rotation object (nicknames from left to right: arrow, curve, spike).

ure 4.4. The most left one is adapted by using the feature map of direction. The arrows at the end of the slider imply a direction when dragging the vertical bar of the slider to either side. The middle variant is constructed by using the feature map of curvature. The horizontal bar is now curved around the object in the scene. This gives the user a hint as to what kind of action will be triggered by moving the slider. The action will be of elliptic form. The most right affordance variant is based on the feature map of depth. The slider gets thinner with each ending. An object getting smaller usually implies that it is further away, thus raising the awareness of a possible 3D interaction in general when using the slider.

4.1.2 Creating Attention Cues for Slider-based Rotation Affordances

We separated the feature map of movement as dynamic hints are very strong compared to static changes to the base affordance. Also, the dynamic movement falls in the category of attention cue instead of affordance as a whole. However, getting the user's attention to the slider and object is a prerequisite so that the user can interact with affordance as designed. As previously explained there are often more things happening on the screen. To guide the attention of the user attention cues can be applied.

The choice of attention cues is based on the research of AMINI et al. (2015) as explained in section 3.4. The top three choices of attention-grabbing cues they found while examining data videos were animation, gradual text, and highlighting.

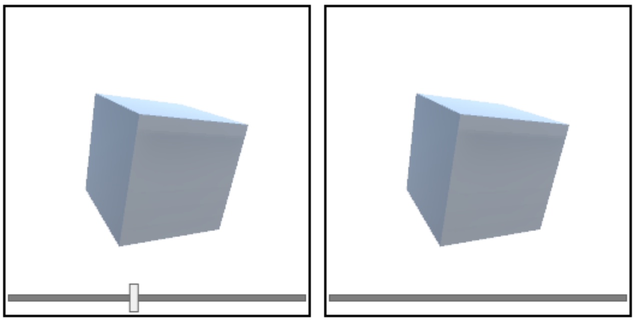


Figure 4.5: Different states of dynamic slider affordance variant "blink" with rotation object, left changes to right with 0.4s delay and then back with 0.4s delay.

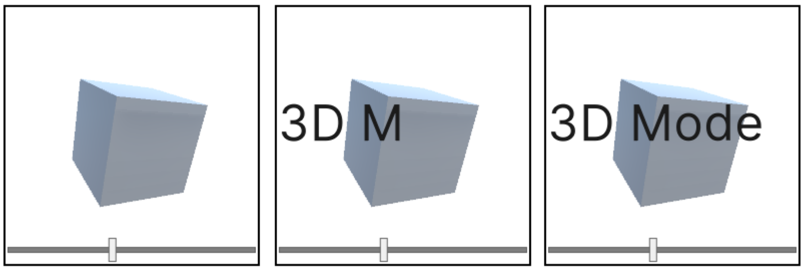


Figure 4.6: Different states of the dynamic slider affordance variant "text" with rotation object.

The third most used cue was highlighting. For this use case, I applied it to the slider by letting the vertical bar of the slider blink until the user interacts with it, shown in Figure 4.5. This focuses the user's attention directly on the interactive part of the slider.

The second most used cue was gradual text. To apply this cue I placed a text "3D Mode" over the scene. The text is gradually shown on a loop with a delay of 0.2s for each letter and after the whole text is shown a delay of 0.6s takes place. Then the text is removed and shown starting from the first letter again. This is shown in Figure 4.6.

The most used attention cue was animation. By animating the slider and the object together I applied this cue. The slider moves about the middle section of the horizontal bar and the object rotates with it. This simulates the user moving the vertical bar. If the user interacts, the animation stops. This is shown in Figure 4.7.

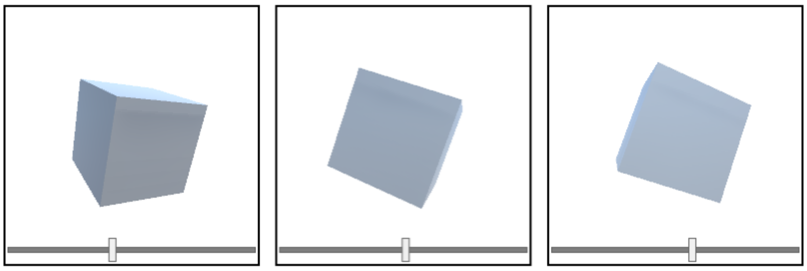


Figure 4.7: Different states of the dynamic slider affordance variant "animation" with rotation object.

4.1.3 Creating the Parts of the User Study

To understand the user's experience with each affordance I created an online study with four parts with an additional part at the start asking for gender, education level, age group, and virtual 3D experience. The user could choose to not answer the first three. The study was made available in English and German. The user's name was not asked for. Every user's input data was connected by a unique random identifier stored in the database.

Before settling on the final order and contents of the study my supervisors and I discussed different designs for the study parts.

How well the user understands the different static affordance variants was of main interest. So as the first part of the study, we created a task where the user has to rate what to expect of the shown static slider affordance variant and then rate the sureness of the answer. The possible interaction techniques to choose from were rotating the object, zooming into the object, and moving the object horizontally. At first, I put every interaction technique on one screen with one affordance showing. We concluded that to not overcrowd the scene and have a better opportunity to mix up the affordances with the interaction techniques, every affordance should be shown only with one interaction technique each. The problem is that the user learns more about the study and its contents with each screen. This creates a bias for later parts. So mixing up is a way to counteract that. However, it cannot be avoided without having many more users for the study.

The second part focuses on the usage of the affordance by the user. The user had to solve tasks where using the slider supports reaching a solution.

Initially, I focused on creating tasks of different complexity where the user can use the rotation affordance in a variety of settings to solve a problem presented. There the focus is the interaction.

A task with a low difficulty I came up with was selecting the correct object from three to choose which was the same as the main object displayed. There, rotating the displayed object can help decide on the right pick. The objects were rectangular cuboids of differing sizes and different rotations.

Another task my supervisors came up with was deciding how many dots are on a displayed cube. There, rotation is necessary to see all sides of the cube.

Inspired by the user study of MITTENENTZWEI et al. (2023) I came up with tasks focusing on volume differences and direction in 3D space. The first one was a task where the user had to decide which of the three objects shown had the biggest volume. There, rotation is necessary to see how far away the objects are from the observer which gives a cue to their individual size. E. g. a sphere in the background could be the size of the sun. But there are no depth cues in the scene, so rotation would give a deciding hint.

A similar task was to decide which object was closest to the user. The rotation of the scene would show which object would travel the most distance on the monitor. The slowest traveling would be the most far away.

A complex task I came up with regarding direction in the 3D space was showing a cube and a sphere where the user had to choose which of the three solutions contained a line segment that connected the cube with a sphere. Here perspective is of high importance to see if the line segment is in front of the objects or if the endpoint has the right direction in the 3D space.

We concluded that to understand which affordance variants were best suited for rotation, there are not many complex tasks necessary. Much of what makes an affordance good is happening before the user interacts with it. I argued measuring the time usage with different slider variants would not give great insights because after the user understands what the slider does, usage would be the same regardless of the graphic design. We kept the task of counting the dots to get an idea of the proficiency of the users in 3D space. This task also tests the willingness of the user to user to interact with the scene. Therefore only the dynamic slider variants are used in this part. The first task had the "standard", the second task had the "text", the third task had the "blink" and the fourth task had the "animation" affordance. So they were shown in order of how big their visual impact was.

GARRISON et al. (2021) used a selection of keywords for the user to choose from to describe visualizations. This gives insights into the user's preferences and associations with an affordance. We used a similar approach with our adapted set of keywords.

In the last part, the user had to rank the slider affordances. We decided on a separate ranking for static and dynamic variants as they are very different. They also mainly serve a different purpose as explained before.

Each part has an introduction screen and its own background color. This helps the user understand when a part starts and when it ends. On the bottom left it is also shown how many screens the user has already completed compared to how many screens there are overall. The user can only go forward in the study by pressing the button on the bottom. The user can learn after part one that the main interaction is rotation. To prevent this from affecting the first part the user cannot go back in the study.

There is feedback provided for the user because pressing the bottom button to go to the next screen only works if all necessary inputs are made on a screen. The feedback consists of red rectangular highlights on the input elements which are not yet selected. The highlights are shown after the user clicks on the bottom button to continue. If the input is made the red highlights vanish. This feedback is very clear since generally red is the color to indicate that something is wrong and should be understood by every user independent of their personal background. On the last input screen the text on the bottom button changes from "Next" to "Send", indicating that now the input data will be sent and saved. After clicking the button, the text changes from "Send" to "Sent" and the button is disabled. So the user knows that the upload has been done. The last screens thank the user for participating.

The four following sections describe the parts of the user study in more detail.

4.1.4 Part 1 – Understanding

The first part focuses on the intuitive understanding of the affordance technique. This part is presented first because the user did not have a chance to learn yet about what the actual function of each affordance technique is. The introduction shows each interaction technique animated so the user knows what is meant by "Zoom", "Rotation" and "Movement". The user is shown a scene with interaction disabled and should rate how likely a specific 3D interaction technique in the presented scene is and how sure he is with his answer. Only the static slider affordances are presented in this part. Each of the static variants is paired with one interaction technique. This creates 12 screens for this part. The following interaction techniques are included: “Zoom”, “Rotation”, “Movement”.

The possible evaluations to choose from are: “highly likely”, “likely”, “neutral”, “unlikely” or “highly unlikely”.

Sureness is evaluated with: “very sure”, “sure”, “neutral”, “unsure” or “very unsure”.

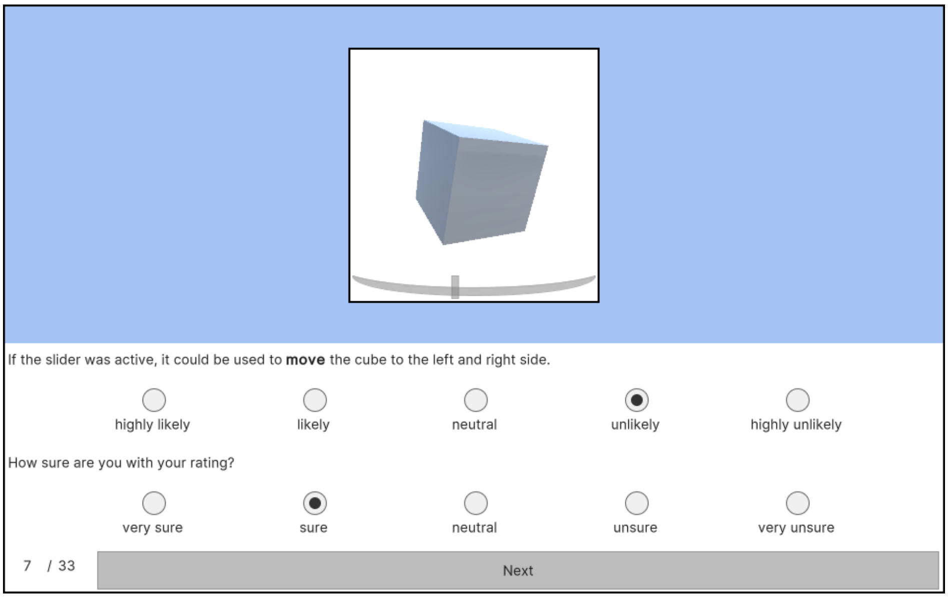
Interaction is disabled to minimize the user’s learning of what interaction is really possible before completing this part. Additionally to the evaluations, the time to answer each questionnaire will be measured. One screen of this part is shown in Figure 4.8.

4.1.5 Part 2 – 3D Proficiency

This part tests the 3D proficiency of the user. This part is in second place because it was not yet clear what interaction was possible with the scene. Interaction is enabled. The picked answers as well as the time needed to pick them are measured. As described before here the task to pick the right number of dots which are on the cube is presented. The task is repeated four times with a different number of dots on the cube, each for one dynamic slider affordance. One screen of this part is shown in Figure 4.9.

4.1.6 Part 3 – Strengths and Weaknesses

This part focuses on the strengths and weaknesses of the affordance techniques. This is the third part because the understanding part has to come



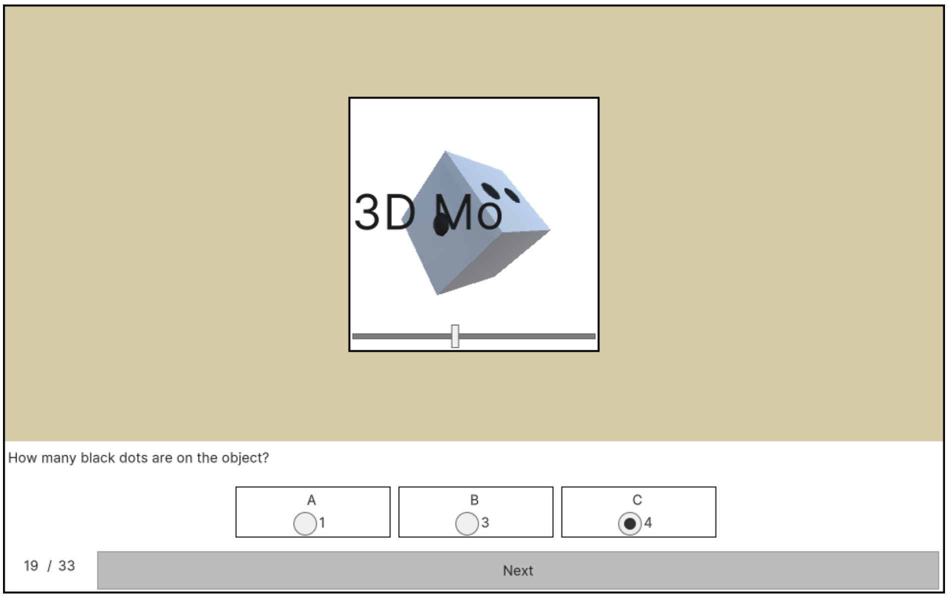
If the slider was active, it could be used to **move** the cube to the left and right side.

○ highly likely ○ likely ○ neutral ● unlikely ○ highly unlikely

How sure are you with your rating?

○ very sure ● sure ○ neutral ○ unsure ○ very unsure

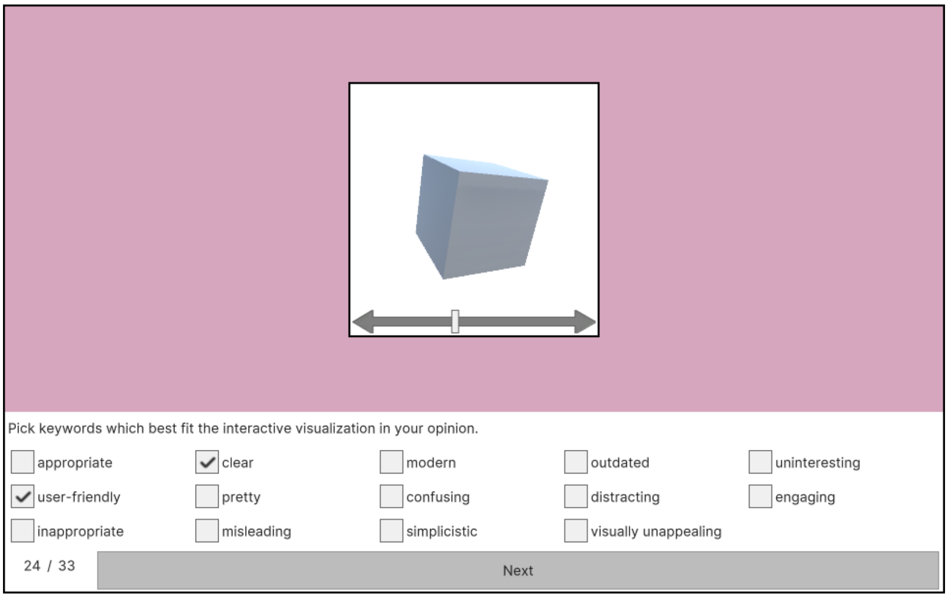
Figure 4.8: One screen of part 1 of the user study with the "curve" affordance and the movement interaction for rating by the user.



How many black dots are on the object?

A ○1 B ○3 C ●4

Figure 4.9: One screen of part 2 of the user study with the "text" affordance and the task to choose how many dots are on the object.



Pick keywords which best fit the interactive visualization in your opinion.

☐ appropriate ☑ clear ☐ modern ☐ outdated ☐ uninteresting

☑ user-friendly ☐ pretty ☐ confusing ☐ distracting ☐ engaging

☐ inappropriate ☐ misleading ☐ simplicistic ☐ visually unappealing

Figure 4.10: One screen of part 3 of the user study with the "arrow" affordance and the task to choose keywords that fit the affordance.

before, the attention cues are shown and this prepares the user for ranking each affordance technique as the last part. The strengths and weaknesses are analyzed with a set of given keywords. The user picks the ones that fit his impression the best. The user can interact with affordance. One screen of this part is shown in Figure 4.10.

The set of keywords is: "appropriate", "clear", "modern", "outdated", "uninteresting", "user-friendly", "pretty", "confusing", "distracting", "engaging", "inappropriate", "misleading", "simplicistic" and "visually unappealing".

4.1.7 Part 4 – Ranking

This part focuses on the appeal of each affordance technique. This part is shown last because the other parts have to be shown before and after using each affordance the user is prepared best to rank the affordances as his last task. Ranking of the static and dynamic variants are separate. The user is shown the standard variant as well in both cases. The user can interact with the affordances. One screen of this part is shown in Figure 4.11.

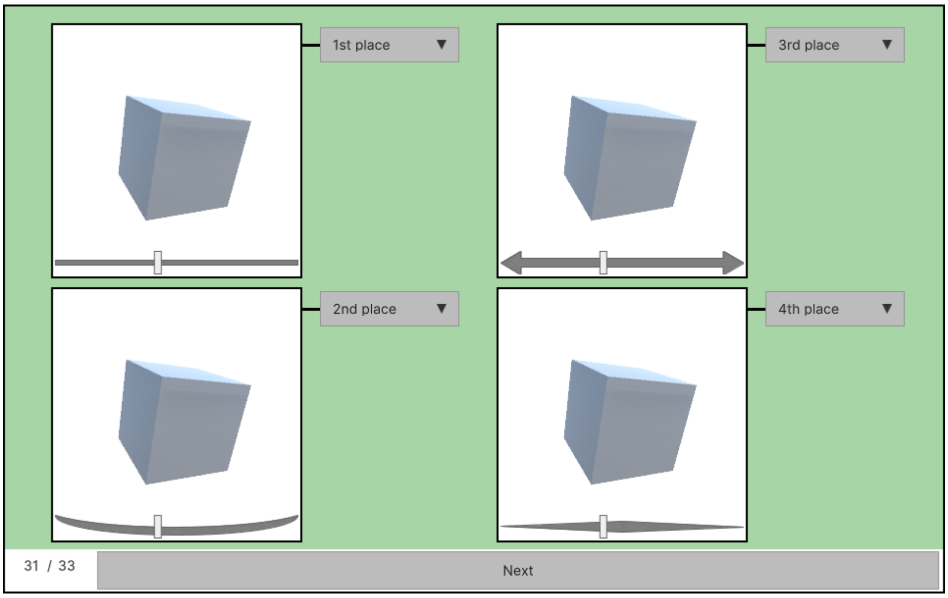


Figure 4.11: One screen of part 4 of the user study with the static variants of the slider affordance and the task to rank the affordances.

4.2 Implementation

In this section, I will explain the implementation of the affordances, the study parts, and how the data gets saved to the database. This includes challenges during development. I started the development of prototypes before my supervisors and I finalized the requirements. I was advised to keep in mind that requirements will change as we further discuss them. I accommodated this by creating a flexible architecture for my implementation. I wanted to reduce copy-and-paste programming and to make rearranging the screens as simple as possible. The development environment, which was mainly Unity, was new to me but I managed to create an appropriate architecture.

4.2.1 Frontend

The frontend is everything shown to the user. As the study is supposed to be online a web page as frontend makes sense.

Software

Unity is a game engine that supports a multitude of platforms (DEALESSANDRI, 2020). It can run 2D and 3D games. It is also used for virtual reality and can render movies. The target platform I used Unity for in this research was WebGL. This produces files that can be served via a web server to a web browser and run without the need for additional programs by the end user. For this to work the browser has to support the following (UNITY, 2024f):

- WebGL 2.0
- HTML 5
- 64bit and WebAssembly

I chose Unity because of the ability to produce Web builds and rendering is already implemented. So I can focus on building the application. Unity was also used in the work of MITTENENTZWEI et al. (2023). I estimated that Unity had all the capabilities I would need to create the user study.

I used Unity Editor 2022.3.7f1 on Windows 11.

Unity offers three UI systems to choose from for projects: UI Toolkit, Unity UI (uGUI), and Immediate Mode GUI (IMGUI) (UNITY, 2024a). UI Toolkit is intended to become the main system for UI design and programming. It does not offer all the features of the other systems but has advantages over the others which are of importance to this implementation. The UI Toolkit is recommended for use if there are many complex user interfaces needed. Individual parts can be better decoupled and reused. This is good because an online user study has many screens which are very similar in design. Overall the UI Toolkit also has better performance than the other two systems.

To test the deployment of the WebGL build produced by Unity locally I used XAMPP (APACHEFRIENDS, 2024). XAMPP stands for "XAMPP Apache + MariaDB + PHP + Perl". This software stack is the same as most websites use. Apache serves the frontend files to the browser of the user. MariaDB is a relational database that saves the input data of the user and PHP is used to transmit the data from the frontend to the backend. The application in the browser sends a request to the PHP scripts which then process the input and send it securely to the database. The backend will be explained in more detail later.

I used XAMPP 8.2.4 on Windows 11. Later I switched to XAMPP 8.2.12 because the database stopped working in the older version on my machine.

Prototypes

I started working on prototypes while I was discussing requirements with my supervisors to ensure functionality and a solid foundation for the final architecture. Challenging was that I had not worked with Unity before. I was familiar with the C# programming language which is used by Unity.

The first prototype was a proof of concept that I could create a working build of a simple program on my machine using Unity. I created a scene with some cuboids and a UI with only a slider. I programmed the slider to rotate the scene when dragged.

In the previous prototype, the interface was layered on top of the view like a HUD in a video game. The second prototype was designed to create a UI first and then place a 3D scene as part of the UI. So the 3D scene does not fill up the whole screen. The UI can be created in the UI Builder which is a WYSIWYG (What You See Is What You Get) editor. To create I window into the scene, I added another camera to the scene. The camera captures the object in this case a cube and renders the view into a render texture. This texture can then be applied as a background of a `VisualElement` in the UI Builder. `VisualElement` is the base class for all UI Elements in the UI Toolkit. The elements are placed in a tree hierarchy and usually saved to `.uxml` file (UNITY, 2024d). In this prototype, I placed three radio buttons as a child of a visual element. This automatically groups the buttons and the user can only select one of them at a time.

The last prototype was intended to create an appropriate architecture to arrange multiple screens after one another and decouple the screens as well as possible from each other. Unity offers an example of a 2D application using a UI Toolkit called "UI Toolkit Sample – Dragon Crashers" (UNITY, 2024e). I investigated this example to see how multiple screens are controlled. The example is a mock game with a main menu consisting of a menu bar and five screens to choose from. There is a "Home" screen, a "Heroes" screen, a "Resources" screen, a "Shop" screen, and a

"Mail" screen. The menu bar with the buttons to access these screens is always visible while switching between these screens. I understood the UI management to be organized in three layers:

- The C# programming layer where the decisions are made what happens if a button is clicked and what screen should currently be shown. These programs are stored in the `.cs` files. There is a control class and a base screen class which every screen derives from. The control class selects the appropriate screen class to show to the user. The screen class then enables showing the respective `.uxml` file. On a fundamental level, all `VisualElements` have to be queried from the C# layer before they can be accessed in the code. This happens either through exact names given in the design layer or by querying all elements of a specific type or only first and other variants are also possible. The set of methods for this is called UQuery (UNITY, 2024b).

- The design layer, where all elements of each screen are arranged and styled. These designs are stored in the `.uxml` files. There is one main `.uxml` file from where every other screen is referenced. As previously stated the `.uxml` files are edited with a WYSIWYG editor, the UI Builder in Unity.

- And the scene layer with all the game objects. There the screen classes are attached to game objects and instantiated with references to the management class and game objects which should be accessible via the interface. So the connection from the screens to the contents of the 3D scene is made.

Interesting to note is that every screen has its own C# class and `.uxml` file which is named the same. This enables a form of decoupling between the screens.

For my work, I adapted the architecture for more screens and the screens being ordered after one another.

I encountered a problem when referencing a design template (`.uxml`) multiple times. The concrete case was about reusing a five-point Likert scale which is used in the first part of the study. The scale consists of five radio buttons grouped. It turns out that reusing the template works but input data is not properly encapsulated. So the references affect each other. Already referencing the file twice in another `.uxml` file creates problems. The purpose of the scale is that only one option can be picked by the user. But by having the same template twice on the same screen only one of the resulting ten options can be picked by the user instead of one for each of the five options. I decided to reference the template only once and after each screen save the inputs in the screen class before resetting the scale for the next screen.

Architecture

As presented before the architecture has multiple layers and sub-layers. In this section, I will give a condensed overview of my final architecture for the frontend.

I begin with the C# class design shown in Figure 4.12.

The main adaptation from the Unity example is the organization of the screens as a linked list. Before the screens could be accessed independently from each other and they had no inherent order. For the user study, I established a linked list of screens, so every screen has information about what screens come next. This creates an order of screens. This makes reordering, removing, and creating screens relatively easy because changes only have to be made to the screen before and after the new or removed screen. So even the `ScreenManager` class does not need to be touched for changes if the changed screen is not the first one. An example of the advantages of this design is the counter of screens on the bottom left of the user study (can be seen in Figure 4.11). The user is shown how many screens there are and how many he still has to do. These numbers are calculated by going through the linked list of screens and counting how many there are. If there are changes made to the number of screens. The code for the display of screen numbers does not have to be touched.

I also introduced classes between the base class `ScreenLinked` and the classes that are connected to the `.uxml` files. As stated before every screen has its own class and design file. These I excluded in the class diagram and indicated them with dots. So similar functionality across each part of the

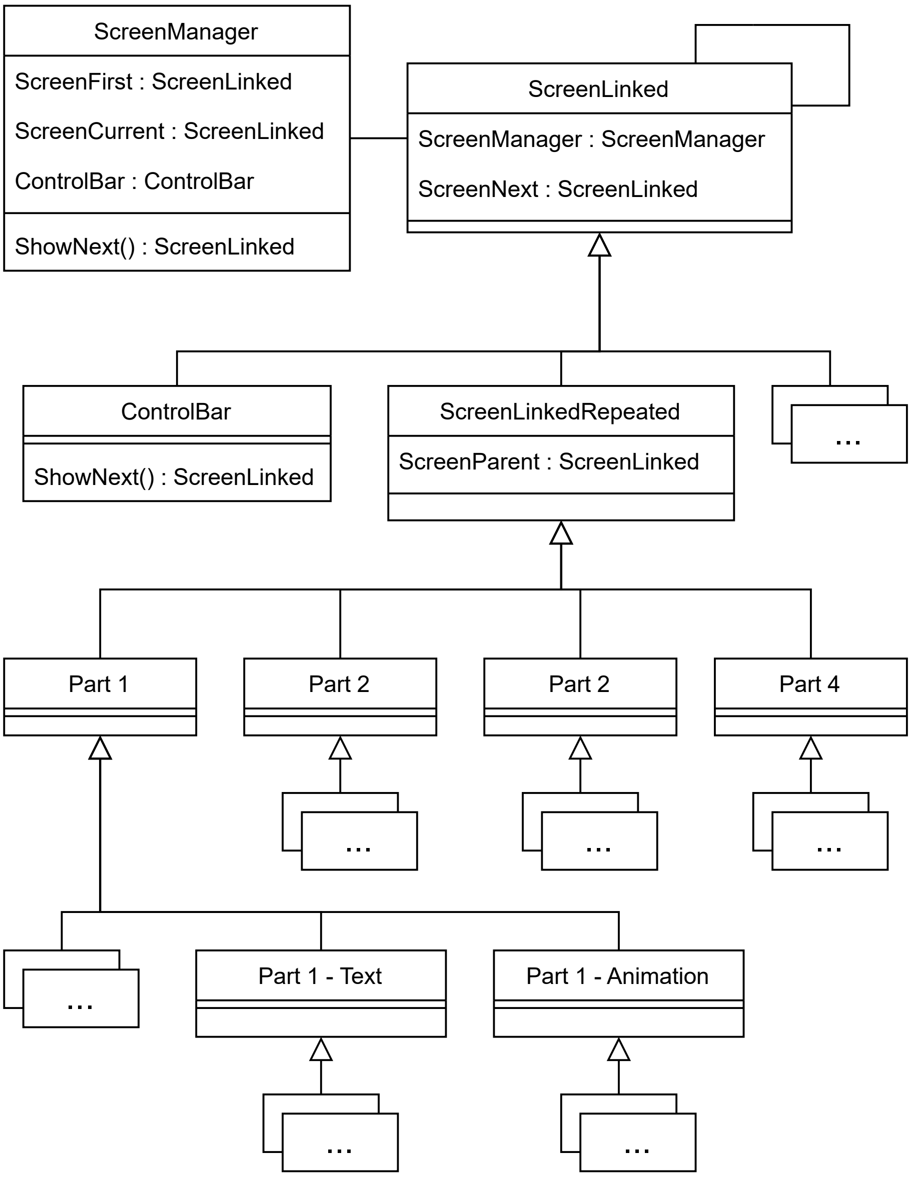

ScreenManager
ScreenFirst : ScreenLinked
ScreenCurrent : ScreenLinked
ControlBar : ControlBar
ShowNext() : ScreenLinked
ScreenLinked
ScreenManager : ScreenManager
ScreenNext : ScreenLinked
ControlBar
ShowNext() : ScreenLinked
ScreenLinkedRepeated
ScreenParent : ScreenLinked
...
Part 1
Part 2
Part 2
Part 4
...
...
...
...
Part 1 - Text
Part 1 - Animation
...
...

Figure 4.12: Very condensed class diagram, the dotted parts are derived classes that are not derived further, most attributes and methods are not displayed, and the original classes are also named differently.

user study is abstracted away in parent classes. Examples of functionality that is placed in a parent class for a study part are:

- saving input data before resetting the input UI
- connecting the slider to the rotation of the cube in the 3D scene
- feedback which is given if a user does not make all inputs necessary

For the first part, I also managed to create extra classes for displaying gradual text and animating both the object and the slider. This makes changing parts of the study simpler if it affects one whole part.

The next layer I will explain is the design layer with the `.uxml` files and their hierarchy.

The UXML format is similar to HTML, XAML, and XML (Unity, 2024c). In those files, the visual tree is contained and they can be referenced from each other. As I mentioned before there are data encapsulation problems but separating the design over multiple files and referencing them is still very useful. For example, I put the slider variants in different files. Then referenced them everywhere where they are needed. So changes can be made in one place. I combined the slider variants with different cubes in the scene. There are the basic cubes and the cubes with dots on them which are used for part 3 where the user is asked to count the dots. So the combinations can be referenced together for example in part one multiple times. But as stated before in the end every screen still has its own file. These files have to be arranged accordingly.

This happens in the main `.uxml` file. There all screens are referenced and the hierarchy is defined. So screens are put in their place in the hierarchy. For example, each screen for part one is placed on the same level. And only the first one is shown via the property `display` which is either set to `flex` or `none`. All screens of part one are contained in a parent screen which is on the same level as the parent screens of the other parts. And again only the current one is shown. There are introduction screens of each part which are not contained in the parent screens as they are different and do not have user input. A visual depiction of this system is shown in Figure 4.13. The concrete screens are shown in Figures 4.8, 4.9 and 4.10. They can be compared to Figure 4.13.

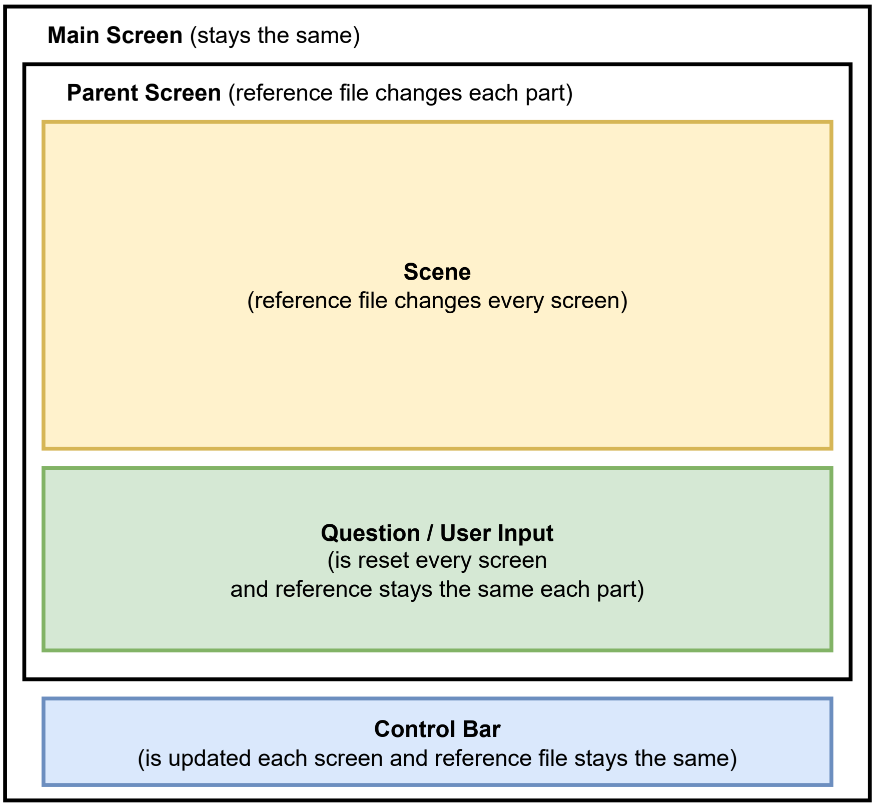

Main Screen (stays the same)
Parent Screen (reference file changes each part)
Scene
(reference file changes every screen)
Question / User Input
(is reset every screen
and reference stays the same each part)
Control Bar
(is updated each screen and reference file stays the same)

Figure 4.13: Overview of screen arrangement across the application.

The last layer I will explain is the 3D scene layer where game objects can be placed and arranged.

In Figure 4.14 the Unity editor is displayed. On the left, there are the game objects in a hierarchy, on the right there is the 3D scene, and in the middle there is the inspector which shows details for the currently picked game object.

Every game object has a position in the 3D scene and additionally other components like geometry on in this case a C# script of a specific screen. Interesting to note is that the scripts for screens are attached to game objects despite the position in the 3D scene not being relevant for the screen. But this is generally the way how classes are instantiated in Unity. The inspector then shows the attributes of the instantiated class. Here is also the point where the order of the screens is determined. By referencing other game objects which are containers for other screens the linked list of screens is constructed. This is a specific use of the game objects but the usual use case is arranging objects in the scene.

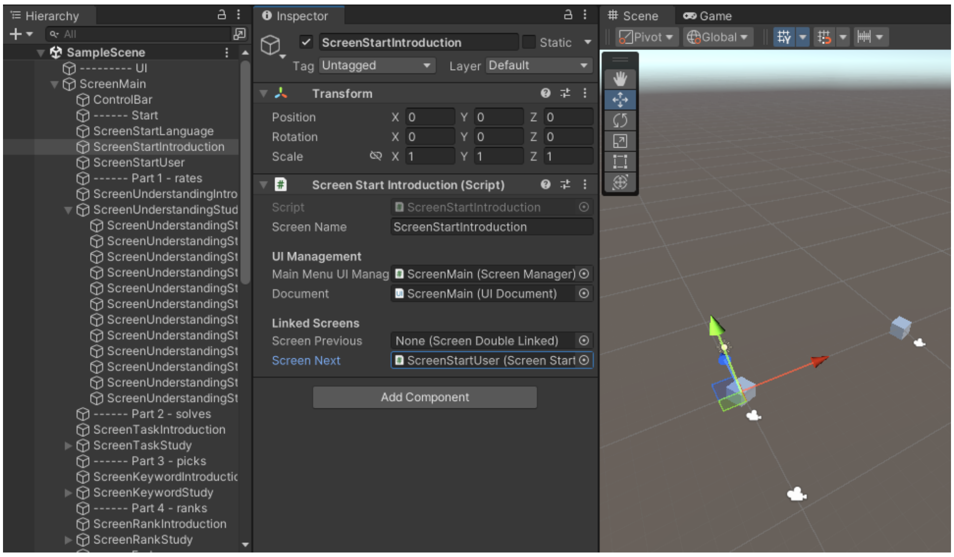
I created multiple cubes and cameras in the scene so every instance where a 3D cube is needed in the UI is separated. Each camera renders its image to render texture and the render texture is then placed as the background of a `VisualElement` in the design layer. This makes it easier to debug and program each instance. For the cubes with the dots in part 2 of the study I used spheres, gave them a black material, and placed them inside the cubes with only a small portion of the surface showing outside the cube. This is how I created the dots on the cubes.

This summarizes the three layers of frontend development. Unity provides in-depth resources for more details on each of the layers.

Flexibility during Development

Here I want to quickly mention how my architecture helped me during development.

In the C# layer I also programmed the text of each screen which is available in English and German. The text is most of the time contained in a parent class of a part of the study. There it was easy to format individual parts of the texts so the important words are highlighted for the user. I also pro-



```
SampleScene
  --------- UI
  ScreenMain
    ControlBar
    ------ Start
    ScreenStartLanguage
    ScreenStartIntroduction
    ScreenStartUser
    ------ Part 1 - rates
    ScreenUnderstandingIntro
    ScreenUnderstandingStud
      ScreenUnderstandingSt
      ScreenUnderstandingSt
      ScreenUnderstandingSt
      ScreenUnderstandingSt
      ScreenUnderstandingSt
      ScreenUnderstandingSt
      ScreenUnderstandingSt
      ScreenUnderstandingSt
      ScreenUnderstandingSt
      ScreenUnderstandingSt
      ScreenUnderstandingSt
      ScreenUnderstandingSt
    ------ Part 2 - solves
    ScreenTaskIntroduction
    ScreenTaskStudy
    ------ Part 3 - picks
    ScreenKeywordIntroductic
    ScreenKeywordStudy
    ------ Part 4 - ranks
    ScreenRankIntroduction
    ScreenRankStudy
```

☑ ScreenStartIntroduction □ Static

Tag Untagged Layer Default

Transform

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

Screen Start Introduction (Script)

Script ScreenStartIntroduction

Screen Name ScreenStartIntroduction

UI Management

Main Menu UI Manag ScreenMain (Screen Manager)

Document ScreenMain (UI Document)

Linked Screens

Screen Previous None (Screen Double Linked)

Screen Next ScreenStartUser (Screen Start

Figure 4.14: Unity Editor with a hierarchy of game objects, inspector, and 3D scene.

grammed the better part of the upload functionality to the database inside the parent classes of the parts. Then in each screen, only the information specific to the screen had to be connected to the upload.

As mentioned before the feedback for the user was also not planned at the beginning and only after a few screens were working we decided it was best if the user had feedback if there were missing inputs. So the user does not have to question why he cannot continue to the next screen. This was also implemented in the parent classes of the parts.

Lastly, a linked list of screens is very helpful to focus development on a specific screen or a specific part of the study because the first screen can simply be connected to the desired screen and then the developer does not have to click through every screen before the one which is intended to be debugged.

Overall the architecture worked out well.

4.2.2 Backend

The backend has different definitions. Here the backend is the database where the user data and the user inputs are saved.

Here I describe how I developed the database and what software I used for the development and the evaluation.

Software

For designing the database I used the software I already mentioned: XAMPP 8.2.4 on Windows 11. It provides a software stack for local web development. XAMPP includes phpmyadmin. There, databases can administered with a GUI. It is possible to create tables and columns via a GUI. Important is also that the references for each table can be designed by a so-called designer where the connections can be made by interacting with the tables in a diagram and the tables can then be arranged for a better overview. The database can then be exported as an `.sql` file for further use.

Database Design

But before I used XAMPP I created an entity-relationship (ER) model and discussed it with my supervisors. Designing ER models is less technical than designing a database in the designer. This makes it better for discussing requirements.

In Figure 4.15 the ER model is depicted. It is not the same as the final design in the database. I added more dedicated tables for the attributes of the user, sureness and likelihood. The attributes of the user are selected by my supervisors and myself. The user can only choose from a particular set. So I created a table for each. This makes changes also easier. Sureness and likelihood have five levels each, a name and a value. So I separated them out into extra tables. But from a conceptual standpoint, it is better to discuss these as attributes not as entities.

Now I want to explain the four relations in the diagram. From top to bottom, the relations are in the same order as the parts of the study.

The first relation, "rates", connects user to interaction and affordance. Each user rates all combinations of affordance and interaction with each rating having a likelihood, sureness and the time it took to complete the screen.

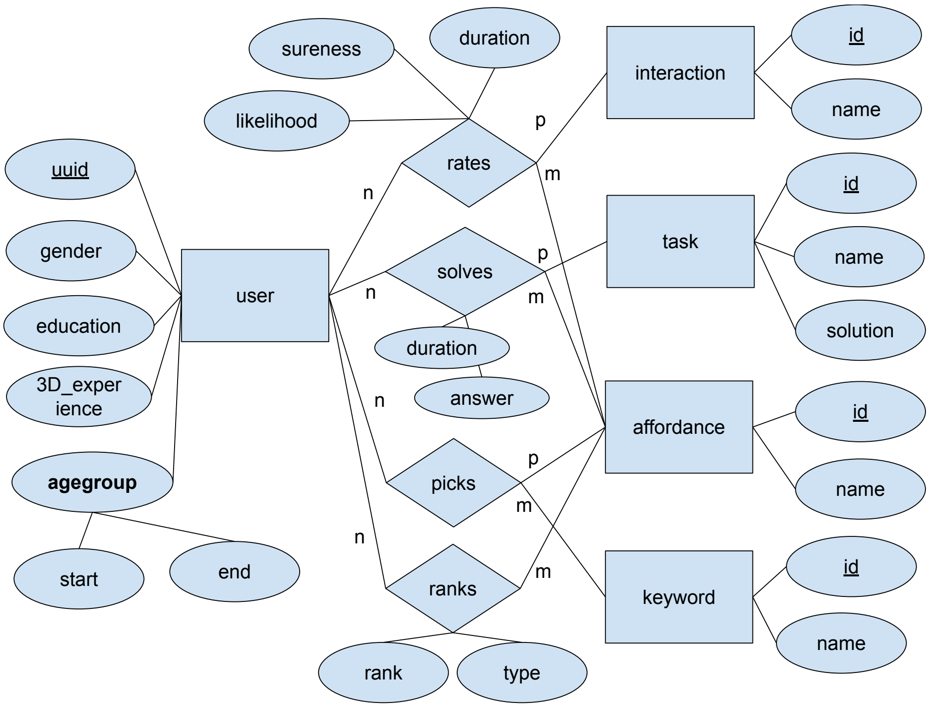


Figure 4.15: Entity-relationship model of the database.

The second relation, "solves", connects user, task and affordance. Every user has to solve each task and this is always in conjunction with an affordance. Only the dynamic affordances are involved but this is a detail for the frontend. The answer picked by the user and the time for completion is saved. The answer can then be compared to the actual solution of the task. The tasks were about choosing the right number of dots on a cube.

The third relation, "picks", connects the user to the affordances and the keywords. Each user can pick for each affordance all keywords possible.

The fourth relation, "ranks", connects the user with the affordances. The user picks a rank for each affordance. There are two rankings, one for the static and one for the dynamic ranking. This is achieved by adding the attribute type which is either "static" or "dynamic".

The design connects the same user to all relevant aspects of the study. So analyzing the data can be done by filtering other parts of the study on the time of completion of the second part. Data can also be filtered by the user attributes.

Flexibility during Development

Here I will explain how I implemented changes to the requirements and how responsive the database design was to the changes.

I put the user attributes in extra tables because it would make changes easier and the design overall cleaner. Over the time of development, we made changes to the user data. The user should also have the possibility to choose no answer. This was relatively easy to change. I made new rows in the attribute tables of the user and in the frontend, I added entries in the drop-downs in English and German as well.

We split the ranking into dynamic and static affordances. This change I made by adding the attribute type to the relation. So all the rankings are saved in the same table and the upload functionality does not have to accommodate another table.

The "solves" relation was added later as well. It required another entity. I thought about only saving if the answer was correct or not. But for the purpose of analyzing I thought it be better to save the answer picked and later compare the answer to the solution of the task.

I managed to implement the changes without too many complications in the database.

Evaluation

The evaluation is done by analyzing the data in the backend. For this, I used SQL and Tableau Desktop 2023.1.10 (TABLEAU, 2024).

To prepare the data I exported the database from the running web server back to my local database in XAMPP. There I was able to evaluate the data with no need for an internet connection.

SQL can be written directly in phpmyadmin. Connecting the tables with "joins" and filtering with "where" clauses allow for complex queries for analysis.

Since visualization of results is important I sought a solution that does not involve copying and pasting the results of database queries into an Excel sheet or similar. GARRISON et al. (2021) used a web version of the Tableau software to display the data in their work as an additional resource on a website. Tableau offers a variety of database connections and I estimated that it would be appropriate for analyzing and displaying the data.

I chose Tableau Desktop and was able to connect it to my local database in XAMPP.

The software automatically recognizes all tables in the database but the relations between the tables have to be modeled again.

An interesting feature of the software is the ability to dig down into the data. After creating a table of data with rows and columns. For example for part one of the study with the affordances as columns and the interaction as rows, the software automatically creates sub-columns or rows if you drag the gender attribute at the end of the data definitions for the rows or columns. So the visualization can be quickly adjusted to investigate different partitions of the data.

There is also the possibility to create filters for the data in a visualization. Complex filters can be programmed with a specific language.

The software can produce `.pdf` files of the visualizations.

The tool fulfilled its job and I did not have to copy data by hand for a visualization.

4.2.3 Web Development

In this section, I will explain the software and the architecture necessary to bring the frontend and the backend to a prepared server, so the user can access the study from the web.

Software

The most important software running on the server was:

- Apache 2.4.58
- PHP 7.4.3-4ubuntu2.19
- MariaDB 10.3.39

Apache serves the files from the server to the client. The client can use a browser to access the study. Then all the files for the frontend get downloaded by the browser.

PHP enables scripts to be run on the server side. These are important to save the data to the database.

The database running on the server was MariaDB. MariaDB is relational relational database management system (RDBMS).

I used Visual Studio Code with an SSH plugin to connect to the web server and upload the necessary files as well as importing and exporting the database.

Architecture

Here I want to briefly explain the different parts of web development.

There are three parts I want to differentiate:

- client / user
- server
- database

A key aspect is to ensure that the database is safe from unwanted access. Since all files accessed by the user can be modified locally, the security can not depend on the frontend. So by design, the database can only be accessed from the web server. And the web server with its PHP scripts which are used to upload the data ensures a secure upload. The contents of the PHP scripts are not visible to the user and with prepared statements to upload the data SQL-injections are prevented.

Connecting Frontend and Backend

The application calls the web addresses of the PHP scripts and sends the data through the `POST` variable. Then the script executes a SQL command for upload.

The upload of data is designed by sending the appropriate ID of each input with a unique identifier for the user to the server. I made a PHP script for each part of the study as well as the user data in the beginning. After the user clicks "Send" on the second to last screen, all scripts are executed paired with a unique ID. The scripts are called per screen. So the first part of the study calls the upload script for that part twelve times.

Deploying to Production

After connecting via SSH to the web server the frontend files and upload scripts can be uploaded. And the database can be updated.

Unity produces a web build with an `index.html` file. The build can be placed at the root of the web page. I also needed to place a specific `.htaccess` provided by Unity inside the `Build` folder of the Unity web build so the user can download the contents of the page. The PHP scripts are also placed at the root of the page. Then the PHP scripts have to be updated with the correct login for the database. Lastly, the database has to be updated by removing all existing tables first and then importing the database schema from a `.sql` file via the command `source filename.sql`. For this to work the `filename.sql` has to be uploaded beforehand.

The download for the user was approximately 5.5MB.

5 Evaluation

In this chapter, I will evaluate the data produced by the study. Besides the quantitative results, I also received verbal feedback from some users. After presenting this I will answer the research questions.

5.1 Quantitative Results

In this section, I will describe the quantitative results for each part of the user study. The database received 39 user entries.

However, it turns out that not all data per user was successfully transmitted. I used the data as is because deleting data in an already small data set has a high influence. You can see this by adding all numbers in each cell of Figure 5.1. Not all cells add up to 39. Data should be complete because upload was only possible after the user had finished all screens of the study. And the user could only finish a screen if all inputs were made. The data upload should be investigated for future use of the architecture.

5.1.1 Part 1 – Understanding

The results for part one of the study are displayed in Figures 5.1 and 5.2.

Figure 5.1 shows in absolute numbers what the users picked when they were asked which interaction was likely when they were shown a specific static affordance. The "curve" affordance was picked overwhelmingly often for the interaction of rotation while also being rated highly unlikely for zoom and movement. The ratings for the "spike" affordance were the most ambiguous. No interaction was recognized by the majority as fitting. The "arrow" affordance was picked the most for moving an object horizon-

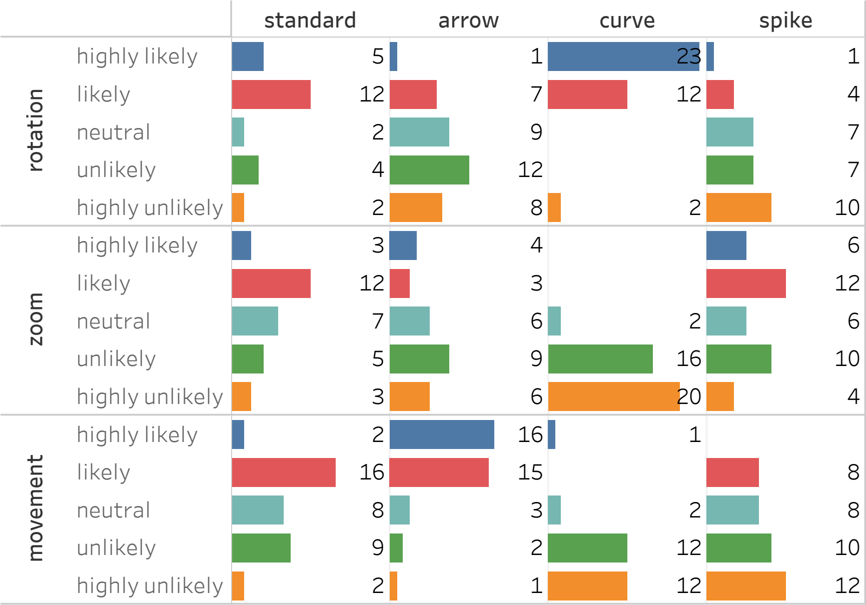
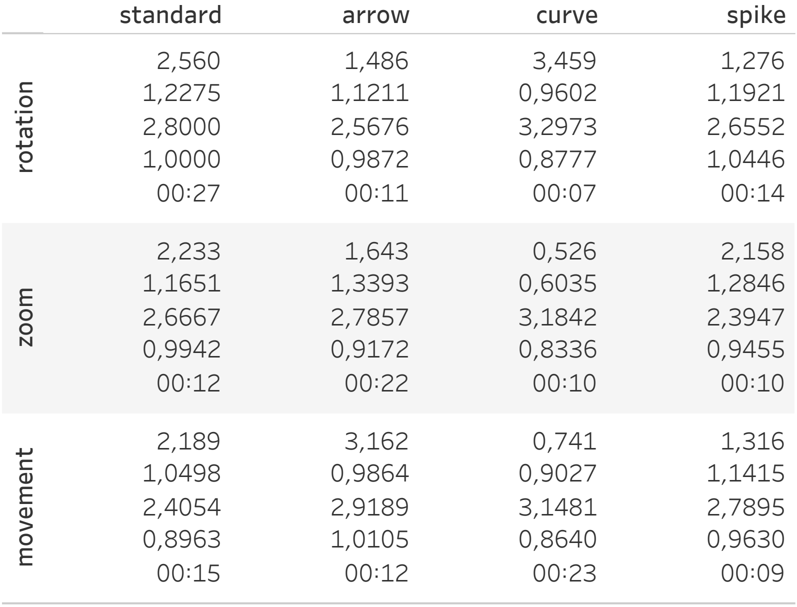


Figure 5.1: Absolute counts of likelihood ratings for each static affordance and interaction combination.

tally on the screen. For zoom and rotation, the vote was ambiguous. The "standard" affordance was rated likely for all interactions.

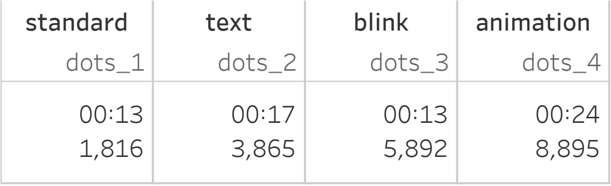
Figure 5.2 shows the averages of the likelihood and sureness as well as their standard deviation and the average completion time for each affordance and interaction combination. It is to be noted that each combination was shown in the same order for each user. The order with the average completion time was: (standard, rotation, 00:27), (zoom, arrow, 00:22), (movement, curve, 00:23), (rotation, spike, 00:14), (zoom, standard, 00:12), (movement, arrow, 00:12), (rotation, curve, 00:07), (zoom, spike, 00:10), (movement, standard, 00:15), (rotation, arrow, 00:11), (zoom, curve, 00:10), (movement, spike, 00:09). This shows that the user became faster with each screen but the "curve" and rotation combination still sticks out because it had the fastest average completion time while being at position seven. The last position was twelve.

Filtering the data by gender, education, 3D experience and age group did not reveal significant differences. Also binning the data by average completion time of the second part of the study did not reveal significant differences.



	standard	arrow	curve	spike
rotation	2,560 1,2275 2,8000 1,0000 00:27	1,486 1,1211 2,5676 0,9872 00:11	3,459 0,9602 3,2973 0,8777 00:07	1,276 1,1921 2,6552 1,0446 00:14
zoom	2,233 1,1651 2,6667 0,9942 00:12	1,643 1,3393 2,7857 0,9172 00:22	0,526 0,6035 3,1842 0,8336 00:10	2,158 1,2846 2,3947 0,9455 00:10
movement	2,189 1,0498 2,4054 0,8963 00:15	3,162 0,9864 2,9189 1,0105 00:12	0,741 0,9027 3,1481 0,8640 00:23	1,316 1,1415 2,7895 0,9630 00:09

Figure 5.2: Computed data for each affordance and interaction combination: from top to bottom in each cell: average likelihood (5 - highly likely, 4 - likely, 3 - neutral, 2 - unlikely, 1 - highly unlikely), standard deviation of likelihood, average sureness (5 - very sure, 4 - sure, 3 - neutral, 2 - unsure, 1 - very unsure), standard deviation of sureness, average duration.



standard	text	blink	animation
dots_1	dots_2	dots_3	dots_4
00:13	00:17	00:13	00:24
1,816	3,865	5,892	8,895

Figure 5.3: Data from all given answers: average duration, average selected answer.

5.1.2 Part 2 – 3D Proficiency

The results for part two of the study are displayed in Figures 5.3 and 5.4.

The users had to solve four tasks where they were asked to count the black dots on a cube. They could choose from three answers. The counting became more difficult because the number of dots increased with each task. In this part, the dynamic affordances were used. For the first task the "standard" affordance, for the second task the "text" affordance, for the third task the "blink" affordance and for the last task the "animation" affordance.

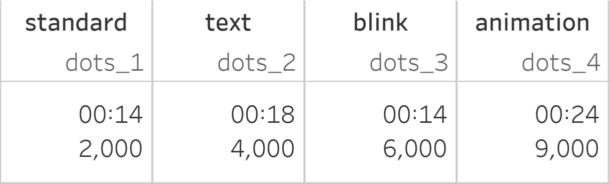
Figure 5.3 shows the average completion time and answer value for all users. Figure 5.4 shows the same but with only the correct answers. The average times for "text" and "animation" are probably higher because the users waited for the whole text to show or the animation to complete one time.

Most users answered correctly but seven users had at least one wrong answer. Only one user answered all four tasks wrong. And only one user answered the first three tasks wrong. The other five users either answered the first or the first and second tasks wrong but then finished with only correct answers. So users who initially did not think about rotating the object were more inclined to use rotation to discover all black dots on the cube the further they were in this part.

5.1.3 Part 3 – Strengths and Weaknesses

The results of the third part are displayed in Figure 5.5.

Here the users had to pick keywords for all affordance variants that they thought would fit them the best. They could interact with affordance



standard	text	blink	animation
dots_1	dots_2	dots_3	dots_4
00:14 2,000	00:18 4,000	00:14 6,000	00:24 9,000

Figure 5.4: Data from only correct answers: average duration, average selected answer.

which resulted in the rotation of the object shown. The user had to choose at least one keyword per affordance.

The affordances were rated differently. The "standard" affordance had the highest votes for "simplistic", "user-friendly", and "outdated" compared to the other affordances. The "arrow" affordance had the most votes for "misleading" and the third most votes for "user-friendly". The "curve" affordance had the most votes for "appropriate", "clear, "modern", and "pretty". It also had the third most votes for "confusing". The "spike" affordance had the most votes for "confusing", "inappropriate", and "visually unappealing". The "blink" affordance was voted most often "distracting". The "text" affordance had the second highest votes for "simplistic" and "distracting". The "animation" affordance had the most votes for "engaging" and the second most for "clear", "modern", and "pretty". It had the least votes for "confusing".

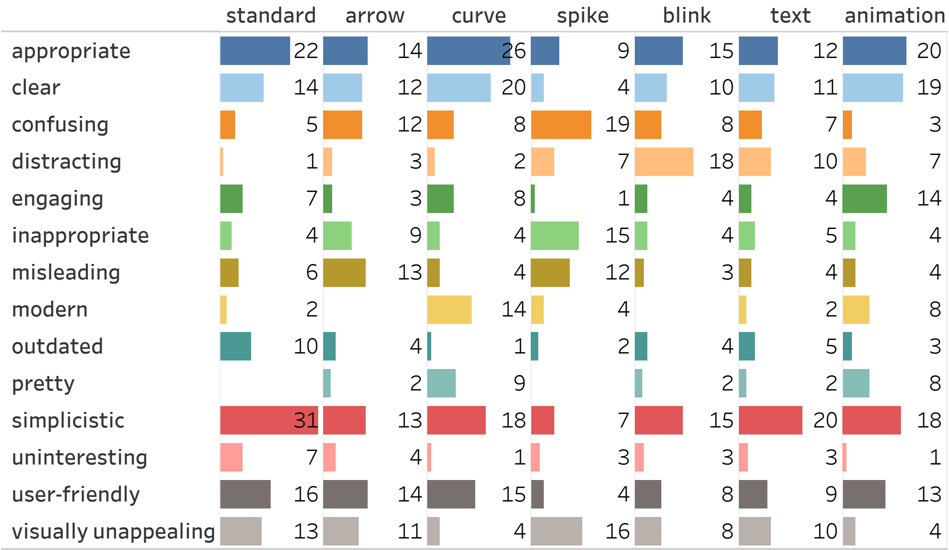
Filtering the data by gender, education, 3D experience, and age group did not reveal significant differences. Also binning the data by average completion time of the second part of the study did not reveal significant differences.

5.1.4 Part 4 – Ranking

The results of the last part of the study are displayed in Figures 5.6 and 5.7.

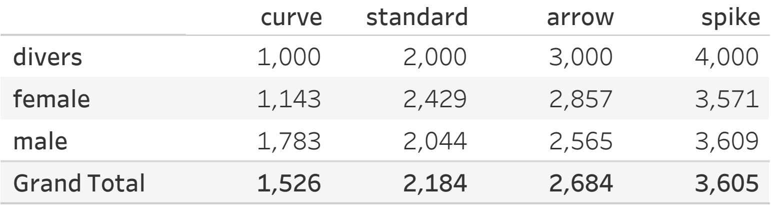
The user had to make two rankings. First, the user had to rank the static affordances and on the next screen, the user had to rank the dynamic affordances with respect to the interaction of rotation.

The average ranking for the static affordances from first to last was: "curve", "standard", "arrow", and "spike".



	standard	arrow	curve	spike	blink	text	animation
appropriate	22	14	26	9	15	12	20
clear	14	12	20	4	10	11	19
confusing	5	12	8	19	8	7	3
distracting	1	3	2	7	18	10	7
engaging	7	3	8	1	4	4	14
inappropriate	4	9	4	15	4	5	4
misleading	6	13	4	12	3	4	4
modern	2		14	4		2	8
outdated	10	4	1	2	4	5	3
pretty		2	9		2	2	8
simplicistic	31	13	18	7	15	20	18
uninteresting	7	4	1	3	3	3	1
user-friendly	16	14	15	4	8	9	13
visually unappealing	13	11	4	16	8	10	4

Figure 5.5: Absolute counts of keywords picked for each affordance.



	curve	standard	arrow	spike
divers	1,000	2,000	3,000	4,000
female	1,143	2,429	2,857	3,571
male	1,783	2,044	2,565	3,609
Grand Total	**1,526**	**2,184**	**2,684**	**3,605**

Figure 5.6: Average rank value for static affordance variants (1 - first place, 2 - second place, 3 - third place, 4 - fourth place).

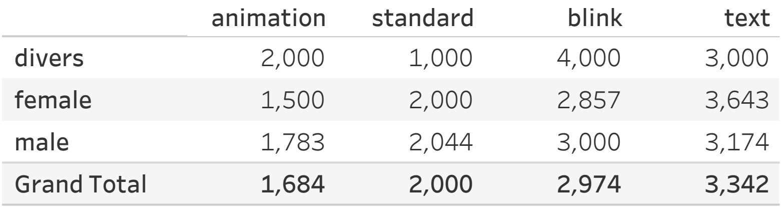
The average ranking for the dynamic affordances from first to last was: "animation", "standard", "blink", and "text".

Filtering the data by gender, education, 3D experience, and age group did not reveal significant differences. Also binning the data by average completion time of the second part of the study did not reveal significant differences.

5.2 Qualitative Feedback

Although I did not ask for qualitative feedback I still received some.

A user with higher education and age 60+ reported that the word "slider" in the German language was not familiar to the user. A colleague of the user



	animation	standard	blink	text
divers	2,000	1,000	4,000	3,000
female	1,500	2,000	2,857	3,643
male	1,783	2,044	3,000	3,174
Grand Total	**1,684**	**2,000**	**2,974**	**3,342**

Figure 5.7: Average rank value for dynamic affordance variants (1 - first place, 2 - second place, 3 - third place, 4 - fourth place).

did not finish the study because it was not understood what the tasks were about. The suggestion was made that the user should be able to review the task while completing screens of a part of the study. Another colleague of the user did not have a computer to do the study.

Another user with age 60+ could not take part in the study because of technical difficulties. A younger colleague of his did the study took part but expressed the wish to be able to change the inputs of the completed screens because the tasks were not understood from the start. An element of fun was also expressed.

Two users had difficulties running the study because the rendering in the browser was not set to a WebGL 2.0 compatible backend. After changing it one of them reported that the application ran with too few frames to complete it.

5.3 Conclusion

In conclusion, the "curve" affordance was perceived most accurately and with the highest confidence by the user with respect to the interaction of rotation. The decision was also the fastest by the user.

The "curve" and "animation" affordance were perceived most positively by the user. The "curve" affordance being the most "appropriate" and the "animation" affordance being the most "engaging". The worst impressions made the "spike" and "arrow" affordance, both being most "misleading" and "confusing". Details can be seen in Figure 5.5.

The average ranking for the static affordances from first to last was: "curve", "standard", "arrow", and "spike".

The average ranking for the dynamic affordances from first to last was: "animation", "standard", "blink", and "text".

The "standard" affordance is shown in Figure 4.3. The static affordance variants are shown in Figure 4.4. The dynamic variants are shown in Figures 4.5, 4.6 and 4.7.

6 Conclusion

In this chapter, I will summarize the work and present recommendations for future work in the area of 3D interaction for broad audiences.

6.1 Summary

The goal of this work was to enhance 3D interaction for broad audiences by exploring slider-based rotation affordance techniques.

Users are generally not familiar with virtual 3D interaction because a screen is often understood as a 2D plane with no depth. Reading and scrolling through text, images, and videos is the main use of a screen. This makes for a challenge if frontend developers want to get the user to understand that virtual 3D interaction is possible.

A meaningful use of virtual 3D interaction is to see an object from more sides than just the front. Virtual 3D rotation around all three axes is a complex task for a non-expert user. Therefore the focus of this research was only on rotation around the vertical axis.

The chosen base affordance to facilitate interaction was a slider. A slider is a rather familiar UI element and has room for adaptation when designing variants.

The evaluated variants of the slider affordance are based on theory of visual perception and statistics on the use of attention cues.

A curved slider was best understood by the users to enable rotation of the shown object and animating the object with the slider by turning the object and dragging the slider was rated most engaging by the users.

Since the focus of the study was the understanding of the user, it was designed to keep in mind that the user gains knowledge the more parts of the study are completed.

The frontend implementation was done with Unity and the Unity UI Toolkit. Unity has three systems for designing UIs. The UI Toolkit is best designed for reuse of UI screens and performance. But it is not yet as feature-rich as the other systems.

The backend was implemented with the database MariaDB and PHP was used to upload the data to the database. The transmission of data was most of the time successful but some data was not transmitted. This should be investigated before starting a study with the same architecture.

Overall the study was a success and the architecture for implementation worked out. Based on this design similar studies can be conducted. Implementing single screens works similarly to designing a webpage and is rather intuitive but implementing a whole study is complex. It involves in-depth knowledge of the Unity UI Toolkit. The database design requires knowledge of relational databases. And since the application is deployed to the web, general knowledge about web pages and database interaction is needed.

6.2 Future Work

Future work is possible in research and technical preparation of the research.

This work focused on rotation using a slider. It is possible to investigate affordances for zooming in on an object or moving an object horizontally as well.

My impression from the qualitative feedback and anecdotal feedback from a colleague was that users were using mobile devices with touch input more often than computers with mouse input in online studies. This would mean that future studies should be designed to be run on mobile devices as well. Affordances could be updated with a focus on mobile touch devices.

I came up with a question if it is possible to create affordances that adapt to the user while the user is using the affordance. A general example would be mouse acceleration. The cursor speed on the screen depends on the speed with which the mouse is dragged. The goal would be to fit an affordance for as many users as possible without the user needing to change a specific setting in the application.

On a technical note, it is possible to improve the implementation of the study. I managed to put similar functionality of screens in parent classes. This was not possible with all functionality. The functionality of the affordances itself was often copy-pasted between the study parts because the focus of my inheritance structure was the data input and upload. It would be good to improve the architecture so that the functionality for each affordance was managed in a central place and could be added in a simple way to a screen.

It is also possible to program editor tools in Unity. The implementation of a study in Unity UI Toolkit is complex and could be partially abstracted away by programming appropriate editor tools.

A
List of Figures

B
Bibliography

[AALTONEN et al. 1998] A. Aaltonen, A. Hyrskykari und K.-J. Räihä. **101 spots, or how do users read menus?** In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '98, p. 132–139. 1998, ACM Press/Addison-Wesley Publishing Co., USA.

[AMINI et al. 2015] F. Amini, N. Henry Riche, B. Lee, C. Hurter und P. Irani. **Understanding Data Videos: Looking at Narrative Visualization through the Cinematography Lens**. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15, p. 1459–1468. 2015, Association for Computing Machinery, New York, NY, USA.

[APACHEFRIENDS] **What is XAMPP?** `https://www.apachefriends.org/index.html`. Accessed: 2024-03-04.

[BADE et al. 2005] R. Bade, F. Ritter und B. Preim. **Usability Comparison of Mouse-Based Interaction Techniques for Predictable 3d Rotation**. In: A. Butz, B. Fisher, A. Krüger und P. Olivier, Eds., Smart Graphics, pp. 138–150. 2005, Springer Berlin Heidelberg, Berlin, Heidelberg.

[BESANÇON et al. 2021] L. Besançon, A. Ynnerman, D. F. Keefe, L. Yu und T. Isenberg. **The State of the Art of Spatial Interfaces for 3D Visualization**. Computer Graphics Forum, Vol. 40(1):293–326, 2021.

[DEALESSANDRI] **What is the best game engine: is Unity right for you?** https://www.gamesindustry.biz/what-is-the-best-game-engine-is-unity-the-right-game-engine-for-you. Accessed: 2024-03-04.

[GARRISON et al. 2021] L. Garrison, M. Meuschke, J. Fairman, N. Smit, B. Preim und S. Bruckner. **An Exploration of Practice and Preferences for the Visual Communication of Biomedical Processes**. 2021, In: Proceedings of VCBM. Accepted for publication, to be presented at VCBM 2021.

[MA et al. 2012] K.-L. Ma, I. Liao, J. Frazier, H. Hauser und H.-N. Kostis. **Scientific Storytelling Using Visualization**. IEEE Computer Graphics and Applications, Vol. 32(1):12–19, 2012.

[MITTENENTZWEI et al. 2023] S. Mittenentzwei, S. Mlitzke, K. Lawonn, B. Preim und M. Meuschke. **Communicating Pathologies and Growth to a General Audience**. In: C. Hansen, J. Procter, R. G. Raidou, D. Jönsson und T. Höllt, Eds., Eurographics Workshop on Visual Computing for Biology and Medicine. 2023, The Eurographics Association.

[POLLMANN 2020] S. Pollmann. **Allgemeine Psychologie: mit 6 Tabellen und 280 Übungsfragen**. UTB 8391 Psychologie. Ernst Reinhardt Verlag, München, 2., durchgesehene auflage Edn., 2020. Stefan Pollmann.

[SARIKAYA et al. 2019] A. Sarikaya, M. Correll, L. Bartram, M. Tory und D. Fisher. **What Do We Talk About When We Talk About Dashboards?** IEEE Transactions on Visualization and Computer Graphics, Vol. 25(1):682–692, 2019.

[TABLEAU] **Tableau Desktop - Insights at the speed of business**. `https://www.tableau.com/products/desktop`. Accessed: 2024-03-04.

[TODOROVIC] **Gestalt principles**. `http://www.scholarpedia.org/article/Gestalt_principles#Past_experience_principle`. Accessed: 2024-02-05.

[TODOROVIC] **Gestalt principles**. `http://www.scholarpedia.org/article/Gestalt_principles#Common_fate_principle`. Accessed: 2024-02-05.

[UNITY] **Comparison of UI systems in Unity**. `https://docs.unity3d.com/Manual/UI-system-compare.html`. Accessed: 2024-03-04.

[UNITY] **Find visual elements with UQuery**. `https://docs.unity3d.com/Manual/UIE-UQuery.html`. Accessed: 2024-03-04.

[UNITY] **Introduction to UXML**. https://docs.unity3d.com/Manual/UIE-WritingUXMLTemplate.html. Accessed: 2024-03-04.

[UNITY] **Introduction to visual elements and the visual tree**. https://docs.unity3d.com/Manual/UIE-VisualTree.html. Accessed: 2024-03-04.

[UNITY] **UI Toolkit Sample – Dragon Crashers**. https://assetstore.unity.com/packages/essentials/tutorial-projects/ui-toolkit-sample-dragon-crashers-231178. Accessed: 2024-03-04.

[UNITY] **WebGL browser compatibility**. https://docs.unity3d.com/Manual/webgl-browsercompatibility.html. Accessed: 2024-03-04.

[WARE 2021a] C. Ware. **Chapter Five - Visual Salience: Finding and Reading Data Glyphs**. In: C. Ware, Ed., Information Visualization (Fourth Edition), Interactive Technologies, pp. 143–181. Morgan Kaufmann, 2021. fourth edition Edn.

[WARE 2021b] C. Ware. **Chapter One - Foundations for an Applied Science of Data Visualization**. In: C. Ware, Ed., Information Visualization (Fourth Edition), Interactive Technologies, pp. 1–29. Morgan Kaufmann, 2021. fourth edition Edn.

[WARE 2021c] C. Ware. **Chapter Six - Static and Moving Patterns**. In: C. Ware, Ed., Information Visualization (Fourth Edition), Interactive Technologies, pp. 183–185. Morgan Kaufmann, 2021. fourth edition Edn.

[YNNERMAN et al. 2016] A. Ynnerman, T. Rydell, D. Antoine, D. Hughes, A. Persson und P. Ljung. **Interactive visualization of 3d scanned mummies at public venues**. Commun. ACM, Vol. 59(12):72–81, 2016.

Statement of Authorship

Thesis:
Enhancing 3D Interaction for Broad Audiences:
Exploring Slider-based Rotation Affordance Techniques

Surname, Name:
Schwarz, Felix

Date of birth:
[redacted]

Matriculation no.:
[redacted]

I herewith assure that I wrote the present thesis independently, that the thesis has not been partially or fully submitted as graded academic work and that I have used no other means than the ones indicated. I have indicated all parts of the work in which sources are used according to their wording or to their meaning.

I am aware of the fact that violations of copyright can lead to injunctive relief and claims for damages of the author as well as a penalty by the law enforcement agency.

Magdeburg, March 7, 2024

....
(Signature)