Opleiding Informatica

Usability of VR in the construction of category theoretical diagrams

Mattias Tuk

Supervisors:
Henning Basold & Marcello Bonsangue

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

15/03/2023

Abstract

In this thesis, the question “Is it possible to provide a VR environment for category theoretical diagrams, which enhances the collaborative work of scientists that work with such diagrams according to a set of use cases?” is answered. This is done by creating such a VR environment, and designing a usability study that can be used to review its usability.

Contents

1 Introduction

1.1 Using tikz to draw category theory diagrams

In category theory, diagrams like these are used:

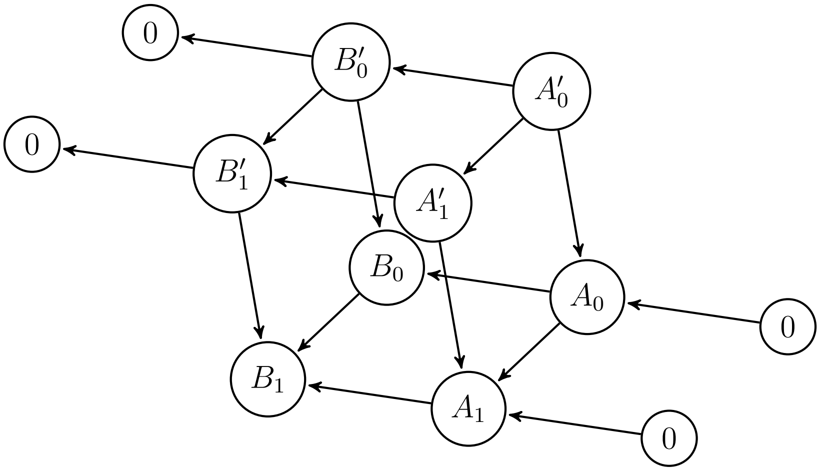


Figure 1: An example of a diagram

Scientists often want to draw these diagrams in LaTeX. To do this, they can use commands from the "tikz" package, and the tikz library "arrows". Below is the example code of a diagram drawn with tikz (the diagram in figure 1):

```
\begin{tikzpicture}[->,>=stealth',shorten >=1pt,auto,node distance=3cm
   ,thick,main node/.style={circle,draw}]

\node[main node] at (10,1.475782) (0) {0};
\node[main node] at (7.346128,1.867354) (1) {$A_{0}$};
\node[main node] at (8.429331,0) (2) {0};
\node[main node] at (5.775269,0.3900079) (3) {$A_{1}$};
\node[main node] at (4.693215,2.25727) (4) {$B_{0}$};
\node[main node] at (3.121275,0.7803407) (5) {$B_{1}$};
\node[main node] at (6.874884,4.588971) (6) {$A'_{0}$};
\node[main node] at (5.303257,3.1107) (7) {$A'_{1}$};
\node[main node] at (4.219888,4.977129) (8) {$B'_{0}$};
\node[main node] at (2.649264,3.500662) (9) {$B'_{1}$};
\node[main node] at (1.566133,5.367168) (10) {0};
\node[main node] at (0,3.88991) (11) {0};

\path[every node/.style={font=\sffamily\small}]
(0)
edge node {} (1)
(1)
```

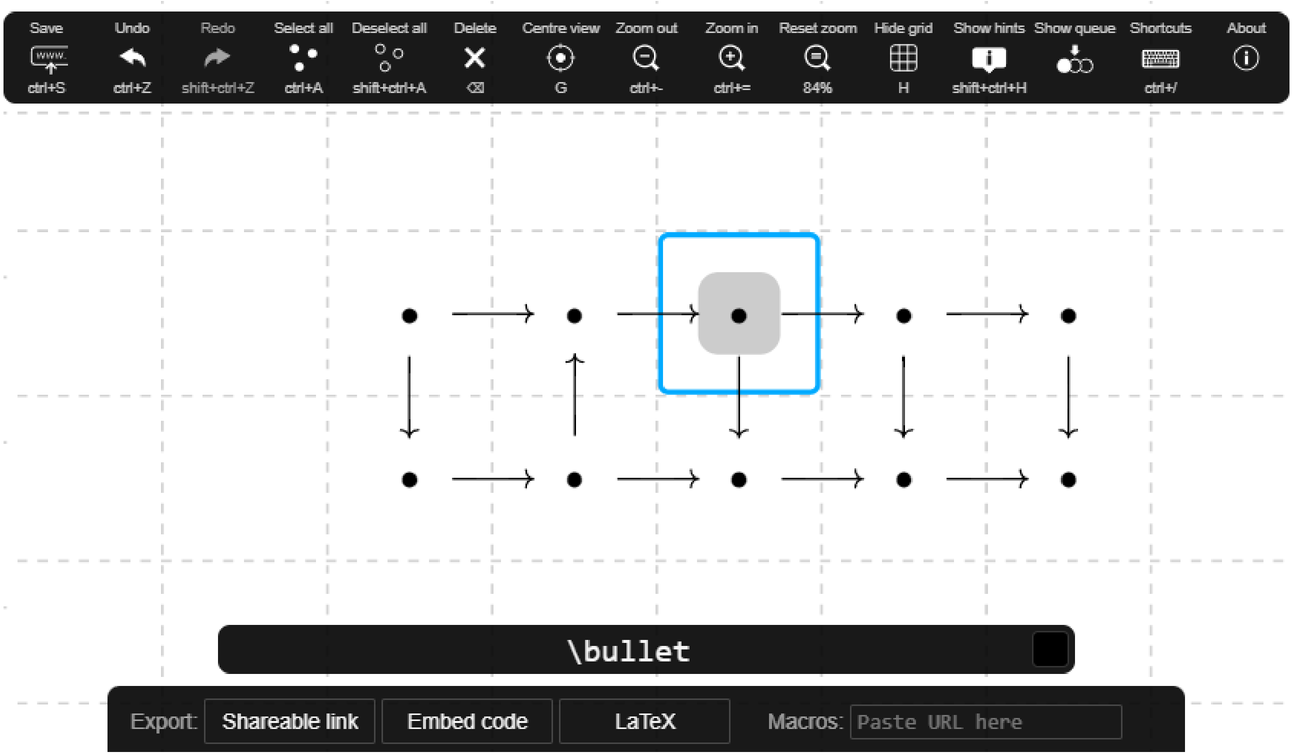
```
edge node {} (4)
edge node {} (3)
(2)
edge node {} (3)
(3)
edge node {} (5)
(4)
edge node {} (5)
(5)
(6)
edge node {} (1)
edge node {} (7)
edge node {} (8)
(7)
edge node {} (3)
edge node {} (9)
(8)
edge node {} (4)
edge node {} (9)
edge node {} (10)
(9)
edge node {} (5)
edge node {} (11)
(10)
(11);
\end{tikzpicture}
```

As shown above, the code starts with `\begin{tikzpicture}`, and ends with `\end{tikzpicture}`. Within those lines, nodes are made using `\node[main node]`. Positions are defined explicitly and the nodes get a number and a label to be referred by in the future. After the nodes are declared, edges are made. This starts with the line `\path[every node/.style={font=\sffamily\small}]`. Nodes are mentioned using the number from before. Between the nodes, `edge node {}` is written. Inside the {}, a label can be written.

It is difficult to write this code by hand, without exactly knowing what the end result will look like. It is important that this problem is solved, because it would make the work of scientists a lot easier if they can draw these diagrams instead of having to use code.

1.2 Quiver

A solution to the problem described in subsection 1.1 already exists: q.uiver.app. [Varkor et al., nd] On Quiver, users can draw graphs, and export them to LaTeX code. The problem with Quiver, is that it is only in 2D. Drawing a picture like figure 1 would be difficult in a 2D application. It is possible, but arrows would cross above each other, and it would be difficult to place the nodes in a way that is easy to interpret.



\bullet

Figure 2: A screenshot from q.uiver.app[Varkor et al., nd]

1.3 Using VR to draw category theory diagrams

In this thesis, a solution to this problem is suggested. The objective is to find a way to draw diagrams in 3D. This can be done by making a digital 3D environment in which a user could draw diagrams. There are multiple ways in which a user could control such an environment: using a mouse, using a tablet etc. The problem with these methods, is that they operate on a 2D plane. Of course depth could still be added. For example by enabling the user to move forward and backward using the keyboard. However, there is a better solution: Virtual Reality (VR) controls.

Most modern VR headsets are controlled using two controllers, which are held by and moved with the user's hands. The advantage of controlling the environment this way, is that the user has control over the environment from two different positions, which they can change intuitively in any direction in the 3D space. The control is highly intuitive, because it feels like using real-life hands. Some headsets even allow using only hands to control VR applications, without the need for controllers.

Another advantage of using VR, is that the user intuitively and easily controls the camera with their head. Besides that, they see the environment in 3D.

For this thesis, a VR application is designed in which the user can draw diagrams, and export them to LaTeX code. Afterwards, I answer my research question: Is it possible to provide a VR environment for category theoretical diagrams, which enhances the collaborative work of scientists that work with such diagrams according to a set of use cases?

1.4 Overview

In this bachelor thesis, the application's visuals are shown, and its controls and design are explained. The rest of the thesis consists of research on usability studies, and the design of a usability study that could be conducted to review and improve the usability of the application.

2 The application

2.1 Instructions

2.1.1 Headset and controllers

The application should be usable on most VR headsets, but it has only been tested on the Oculus Quest 2 (Figure 3). When describing the controls, the layout of the Oculus Touch controllers (Figure 4) is used.

Figure 3: The Oculus Quest 2 with controllers[bol.com, nd]

2.1.2 Moving through the environment

The virtual environment of this application looks like Figure 5.
The user can move around using the stick on the left controller. The stick on the right controller rotates the camera.

2.1.3 Drawing nodes and arrows

A node can be placed by pressing the trigger button. Both controllers have such a button, and they do exactly the same. Instead of immediately releasing the trigger, the user can choose to hold the button, move their hand to a new position, and then release the trigger. In this case, a new node will appear at this new position, with an arrow between the two nodes. If the trigger is released in

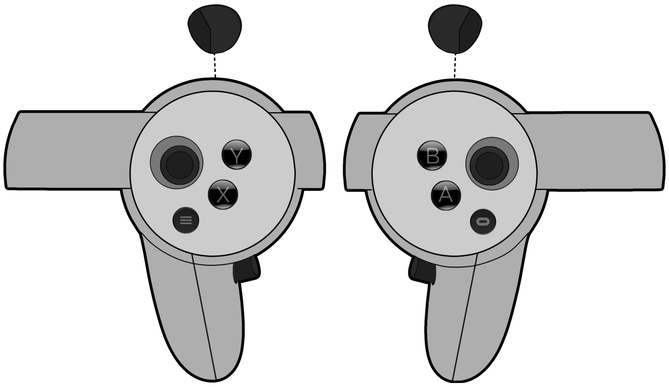


Figure 4: The Oculus Touch controllers. The trigger buttons are on the top in this image. They are pressed with the user's index fingers. The grip buttons are under the user's middle fingers. Picture taken from [u/Arc8ngel, 2018], from reddit.com.

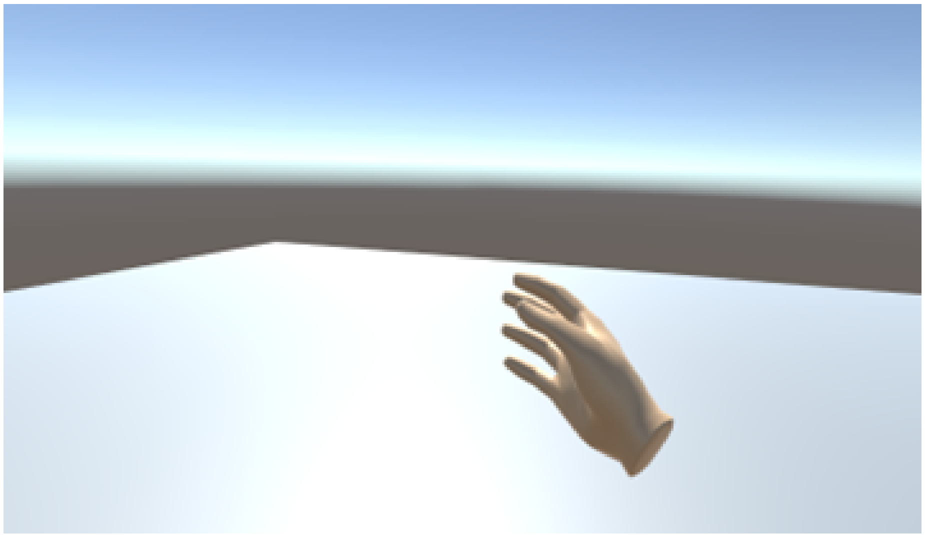


Figure 5: The empty environment.

the position of an existing node, no new node will appear. If the node in which the user releases the trigger is not the same as the node they started in, an arrow does appear. If the user starts drawing near an existing node, that node will be the node that the arrow starts from, and no new starting node will be created. To remove an arrow, an arrow can be drawn over an existing one. No new nodes or arrows will be created, and the existing arrow will disappear.

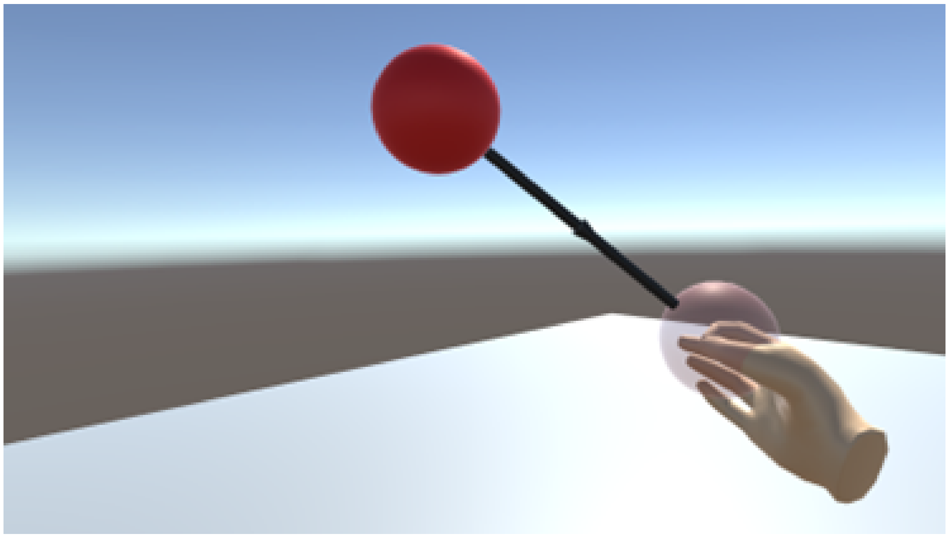


Figure 6: Drawing two nodes and an arrow.

2.1.4 Moving and deleting nodes

The grip button can be used to grab and move an existing node. The grip button is located below the user's middle finger, and it is the button that is usually used for grabbing things (hence the name "grip button"). Using this button closes the virtual hand. Both hands have this button, and they do exactly the same. Once a node is grabbed, a bin symbol will appear. Release the node in this bin to delete the node. If the node is moved away from the bin, the bin symbol will disappear, so the node's position can also be slightly moved, without deleting it.

2.1.5 Drawing more precisely

There are two ways to draw more precisely. One of them is rounding the angles using the B or Y button. The other one is using a snap grid, by pressing A or X.

By pressing B or Y, the angle the arrow is drawn in, can be rounded. This means that if an arrow is drawn while the user holds B or Y, the angle between the arrow and each plane in the 3D space is always rounded to 45 degrees. This may sound complicated, but it is basically the same as drawing a line while holding shift in applications like Paint or Word, but in 3D. Instead of doing this while drawing, it can also be done while moving a node with the grip button, if that node is connected to exactly one other node.

By holding A or X, the node held or drawn moves to the nearest position in an invisible snap grid.

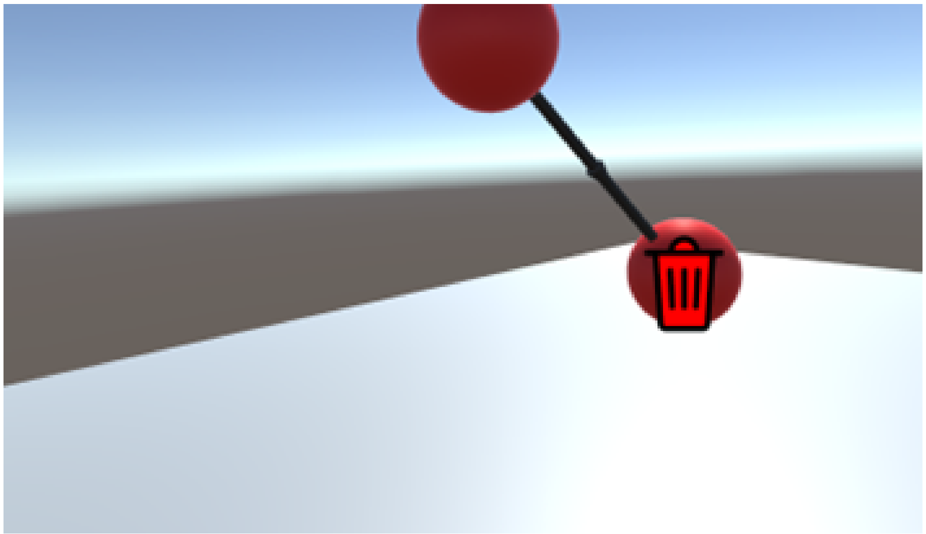


Figure 7: Deleting a node.

2.1.6 Rotating the diagram

By grabbing the air with both hands, the diagram can be rotated and moved. If the user moves their hands at least one meter apart, the diagram's rotation will reset.

2.1.7 Labeling the nodes and arrows

If the user presses A or X while holding nothing, labels will appear. These labels are uppercase for nodes, and lowercase for arrows. A label can be grabbed with the grip button, moved, and released to label a node or arrow. By releasing it in the air, the label will be deleted.

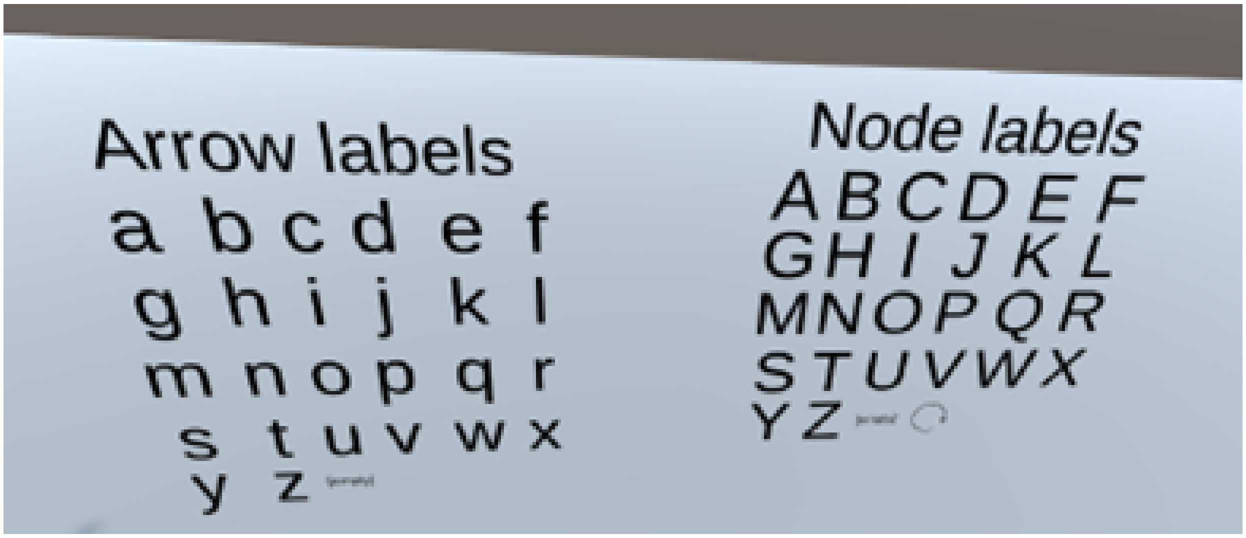


Figure 8: The available labels.

By pressing B or Y while holding the label, an apostrophe will appear or disappear. Pressing A or X while holding it, will increase the number in the lower right, or make it “0” if there is no number yet.

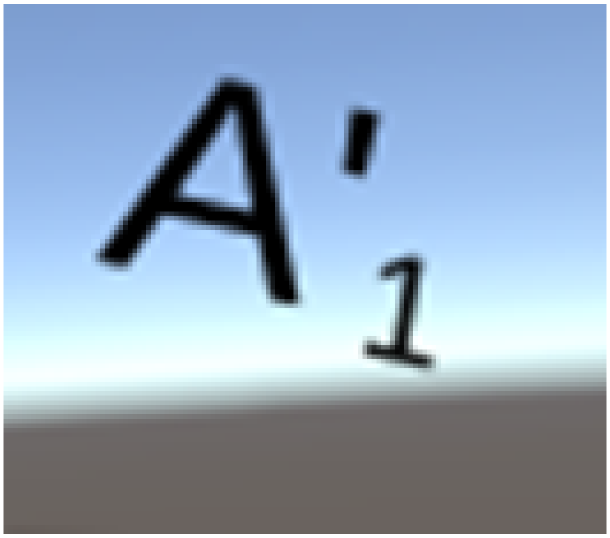


Figure 9: The “A” node label with an apostrophe and a “1”.

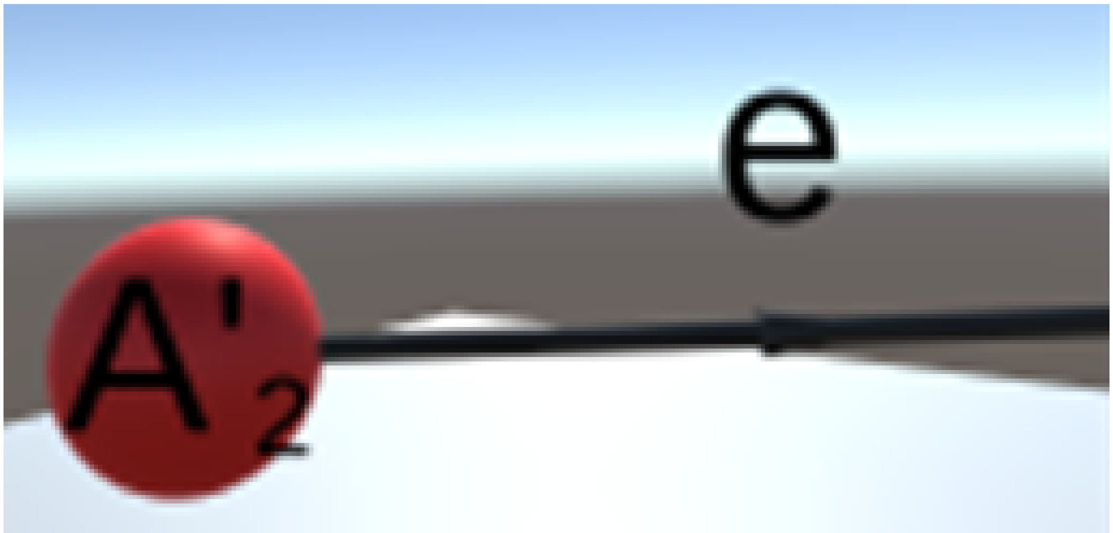


Figure 10: A node and an arrow, both labeled.

Morphisms can be composed by releasing one label while holding another one. The labels will become one new label. That label will contain the letters that were concatenated, with the function composition symbol between them. The released label will be behind the held label.
To create a label that is different from the above options, one can be made from the menu. Pressing Y or B while holding nothing will open the menu. The trigger can be used to click “Make nodelabel

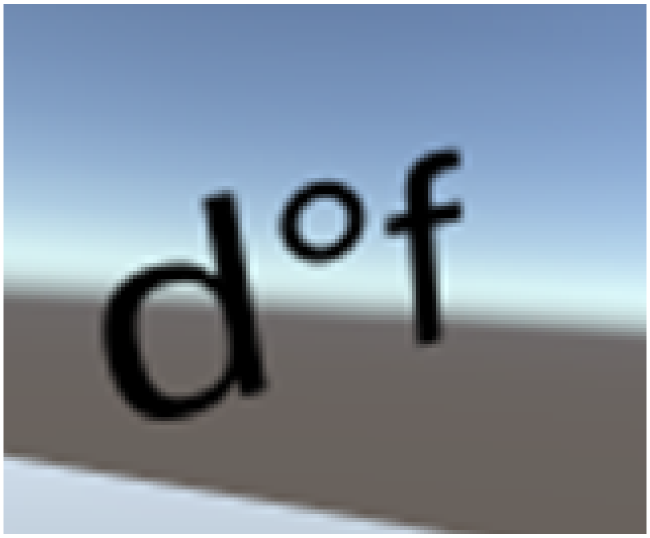
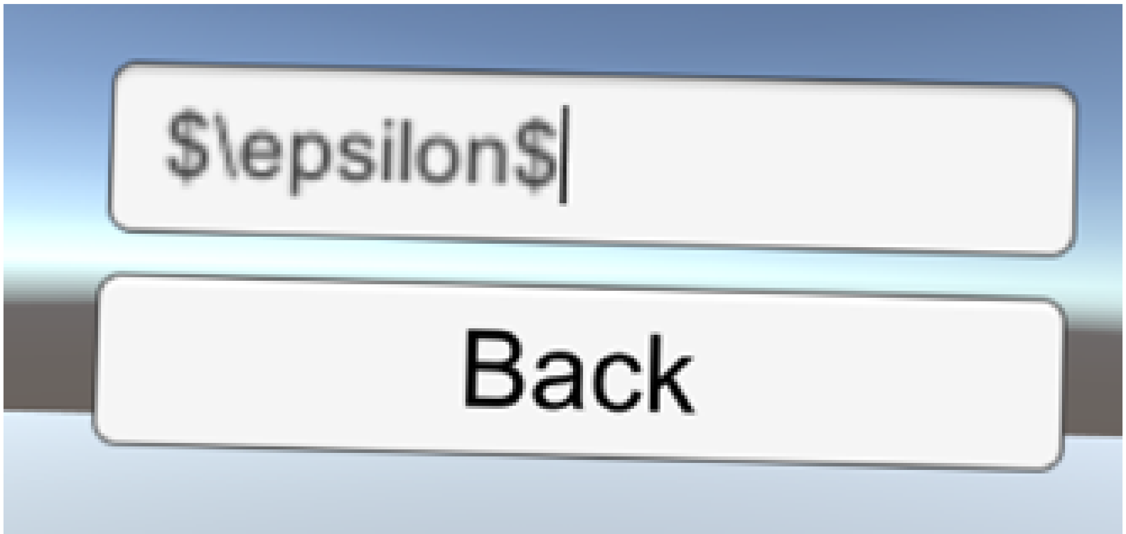


Figure 11: Two concatenated labels.

with keyboard" or "Make linelabel with keyboard". Now the keyboard can be used to type a label with LaTeX code. After pressing enter, the label will be generated as a png image, and shown inside the application. This works by passing a command to the command prompt and using LaTeX, Imagemagick and Ghostscript. If the code is incorrect, this will not work and the label will instead show the original code in red.



```
$\epsilon$
```

Figure 12: Typing an epsilon label.

There are two labels that work differently from the rest. These are the label that has the text "[Empty]" and the label that shows a loop symbol. The "[Empty]" label clears the label when applied to a node or arrow. The loop label gives a node a loop symbol, which will give the node an arrow to itself in the final output.

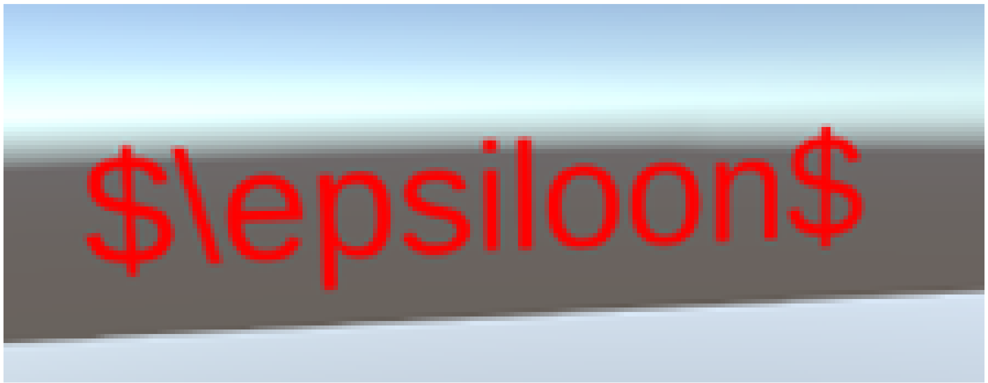


Figure 13: A misspelled label.

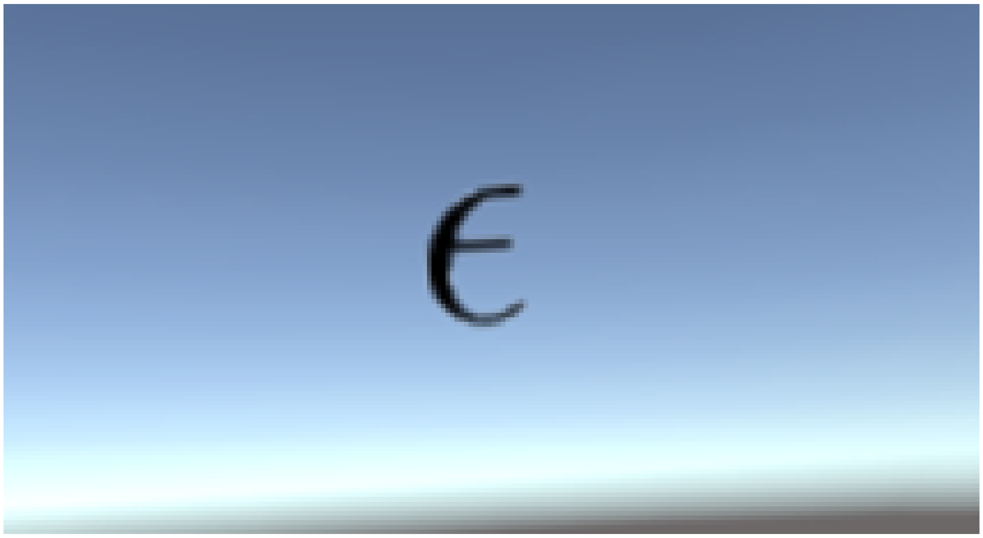


Figure 14: What the epsilon label looks like in the application.

2.1.8 Taking a picture

The user can convert the diagram to LaTeX code, by using the camera. The camera can be grabbed with the grip button. Pressing the trigger while holding the camera, will take a picture from that perspective. This means that a latex png will be generated. This png will be visible on the camera screen. The code with which this image is made, is copied to the user's clipboard, and can be found in the files of the application.

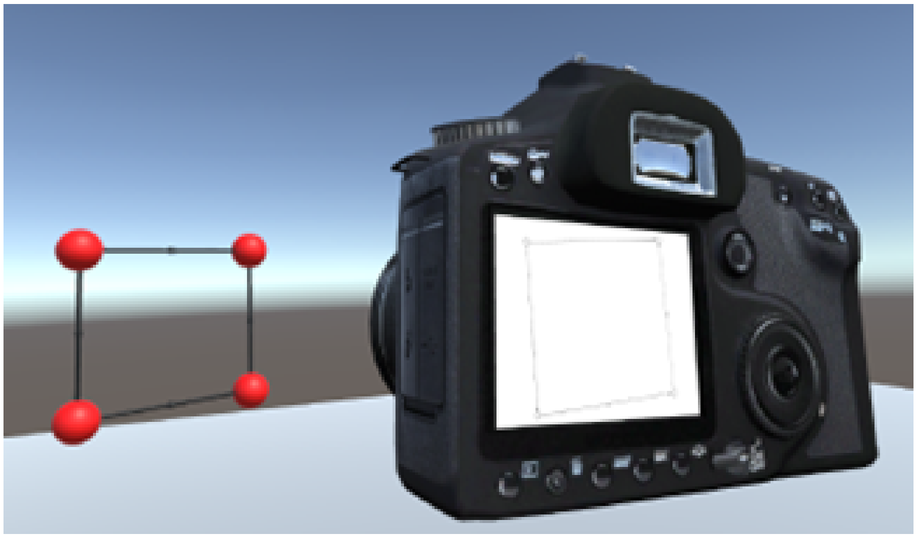


Figure 15: Taking a picture.

2.1.9 Using the menu

In the menu, the user can look at the controls, save, load, clear the diagram and exit. The user can also create labels like described above, and access the settings. In the settings, the user can change the size of the labels, nodes and arrows. The user can also tick a box to show the labels as LaTeX code instead of showing it as it will look like in the pdf. There is also an option called "LaTeX node visibility". Here the user can choose "Always visible", "Only empty" or "Never visible". "Always visible" means a circle will always be visible in the pdf when there is a node. "Only empty" means a circle will only be visible when the node has no label. "Never visible" means there will never be a circle. The final option in the settings is "Snap grid size" this increases or decreases the distance between points in the snap grid.

2.2 Software design

2.2.1 Engine

The application is made in Unity. This is a game engine, in which VR games can be made. To implement VR, Unity's own XR SDK is used.

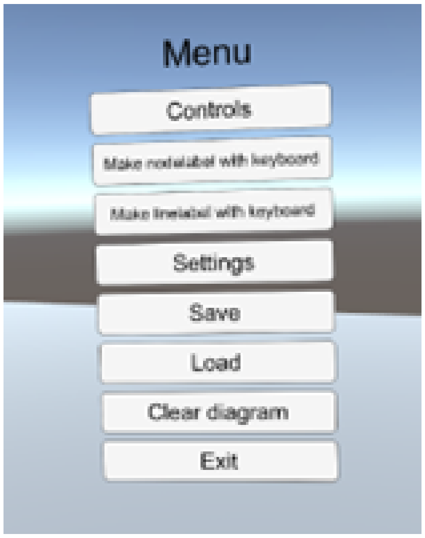


Figure 16: The menu.

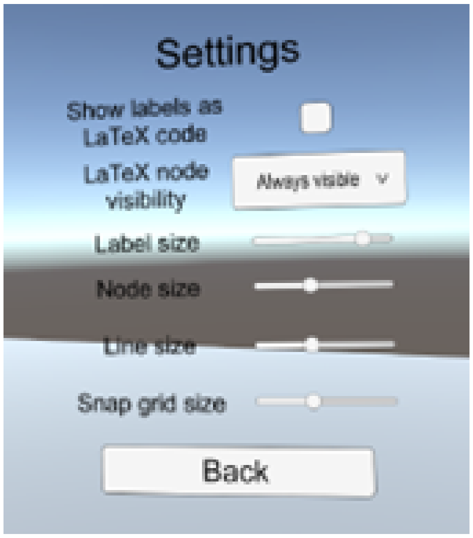


Figure 17: The settings.

2.2.2 Movement and camera control

Movement is done as in most VR games or applications: by using the stick on the left controller. It is done this way, because it is the most common way, so it is the most intuitive way. A teleportation system could also have been implemented for people who get motion sick easily, but this seemed unnecessary, because for those people it is easier to only move the diagram instead of themselves.

Moving the camera is done in the most common way: using the right stick. It works in small jumps instead of continuous movement, to prevent users from becoming motion sick.

2.2.3 Placing nodes and arrows

The chosen button for node and arrow placement, is the trigger button. This button is chosen, because it is the easiest to use. The user will have their thumb on the thumbstick, their index finger on the trigger, and their middle finger on the grip button. Their ring finger and their little finger are not on any buttons, but only used to hold the controller. The thumbstick is used for movement and camera rotation, the grip button for moving nodes (see below), so the trigger is the only button left. Of course there are also the A, B, Y and X buttons, but these are usually meant for less used actions, because the user needs to move their thumb to them to use them. Pressing the thumbstick was also an option, but the trigger is an easier button to press. In conclusion, the trigger is chosen for this action, because it is a common action, and the trigger is the easiest button for it. The reason both trigger buttons can be used, is that users can choose which hand they want to use, and can use both if desired.

2.2.4 Moving and deleting nodes

The grip button is, in most VR games and applications, the button that closes the user's virtual hand, and with which the user can grab objects. This is why the grip button is used for grabbing nodes. Both hands can do this, because in real life, most users can use both hands to grab objects.

For deleting nodes, multiple ideas were tried. The first idea, was to use a button while holding the node to remove it. This works fine, but eventually, there were too few buttons to keep this method in. The second idea, was to use a bin symbol, in which the user can drop the node. In a 2D application, such an icon would probably be placed in the corner or at the edge of the screen. In VR, this is a bad idea, because the corner of the screen moves when the user moves their head. This means another position had to be chosen. It is hard to find a good spot relative to the user's head, because whichever spot is chosen, it can always be the position where the user want to move the node to. Eventually, the best position turned out to be the position of the node. The only problem is that a user may want to only move the node slightly, and then the user would drop the node in the bin. The solution, is to make the bin disappear when moving the node away from it. This way, the user can grab the node, move it away, move it back and place it in their chosen position.

2.2.5 Rounding the angles

When testing the application, it turned out to be difficult to draw the diagrams in a good-looking way. This is because it is difficult to draw in an angle of exactly 90 or 45 degrees. The solution, is to do what some 2D drawing applications already do: use a button to round the angle to 45 degrees. Usually, this is the shift button. For this application, the B and Y buttons were chosen for this. The reason for these buttons, is that the trigger and grip buttons were already taken, and the sticks are harder to press. There are also two menu buttons, but this is not a menu, so it makes no sense to use it for this purpose.

After trying this new feature, it turned out not to be good enough for users who want to be really precise. This is because it is still difficult to draw lines in exactly the same length. For this purpose, another button was introduced: one that moves the nodes to the closest position in a snap grid. For this action, the A and X buttons were chosen.

2.2.6 Moving and rotating the diagram

Interaction in VR feels best when it is as intuitive as possible. This is why, instead of something like rotation buttons, it would be best to be able to grab and rotate the diagram using hands. Implementing grabbing the whole diagram is difficult, because there is no logical place to grab it. The nodes move apart from each other, and grabbing arrows would be difficult if nodes are close, because the user will often accidentally grab a node. An implementation that was briefly considered, is creating two handles around the diagram which the user would grab. This idea was quickly dismissed, because this would be impossible to use when the diagram gets too big. In the end, the idea occurred that there is no need for handles, because the user can just grab the air. The way rotation now works, is that the user grabs the air, and moves their hands as if the diagram is between their hands. If the user moves their hands forward or backward, the diagram moves the same way. The diagram's rotation will reset if the user moves their hands apart.

2.2.7 Adding labels

For labelling nodes and labels, dragging and dropping seemed the most intuitive way. It would be annoying to always have the label menu in view, so the user can make them appear or disappear with A or X. To add and apostrophe or a number, Y and B and A and X can be used, because those buttons are available in that moment. Concatenating labels also works by grabbing and dropping. It is very difficult to put all possible labels in the application this way, so for other labels, LaTeX code can be used.

2.2.8 Converting the diagram to LaTeX code

To convert the diagram to Latex code, a perspective must be chosen from which the diagram is viewed. In an earlier version of the application, this was the perspective of the user. This implementation worked, but was later replaced by a better solution. Choosing the position and converting the diagram, felt like making a photo. For this reason, the mechanic was replaced by a virtual photo camera, with which the user makes the "photo". This implementation feels intuitive, and the user can see the diagram in the screen on the camera, so they can retake the photo if they are unsatisfied.

2.2.9 Settings

Some settings were made available to the user, to improve their experience. One of these, is to resize the labels, nodes or arrows. This was added, to account for users having different preferences. The same goes for the snap grid size setting. Another setting, is a toggle for showing the labels as LaTeX code or showing them as they will look like in the diagram. This option was added so users can easily see what can be changed in the labels if something is not to their liking. Finally, there is a setting for LaTeX node visibility. This is not a setting that changes how the application is used, but one that changes what the diagram looks like when it is exported to LaTeX code. This setting is added to account for the different preferences users have.

3 Usability studies

3.1 Reasons for doing usability studies

To see if the application is a good tool for drawing graphs and converting them to LaTeX, a usability study could be conducted. But what would be the advantages for doing this over not doing it? According to Usersense [UserSense, 2022a], there are five main reasons for using a usability study:

A lot of the users' issues are noticed. Even when an issue is not so big as to ruin the whole program, it can be frustrating enough for a user to find an alternative. These issues need to be found, and the best way to find them, is a usability study.

By using usability testing while developing the product, feedback can be used to make the program exactly how users want it.

See the product through the lens of a user. Developers understand their applications more than anyone, so it is hard to see how a user would interact with it. Using usability testing, can help increase a developer's understanding of how new users would use their application.

Usability testing exposes the issues themselves. Even if a developer has data of how many users stop using their product, they do not know why, until they do a usability test.

Usability testing can be used to find improvements. Even if there are no issues, users can still find improvements developers would not think of themselves, through a usability test.

3.2 Meaning of usability

3.2.1 The problem with defining usability

Multiple researchers consider the lack of definition for usability an obstacle:

"One of the most important issues is that there is, as yet, no generally agreed definition of usability and its measurement." [Shackel, 1990]

“Attempts to derive a clear and crisp definition of usability can be aptly compared to attempts to nail a blob of Jell-O to the wall.”[Gray and Salzman, 1998]

“A major obstacle to the implantation of User-Centered Design in the real world is the fact that no precise definition of the concept of usability exists that is widely accepted and applied in practice.” [Alonso-Ríos et al., 2009]

In conclusion, there is still no agreed definition of usability. The reason for this, is probably that usability is something that emerges from multiple interactions between users, products, tasks and environments. This makes it difficult to measure with a simple number, or to give it one definition that fits all situations.[Lewis, 2014]

3.2.2 A sufficient definition of usability

According to Nigel Bevan, Jurek Kirakowski and Jonathan Maissel [Bevan et al., 1991], there are four views of what usability is. The first one is a product-oriented view. This means that the usability is measured by the quality of the ergonomic attributes of the product. The second view is a user-oriented view. This is measuring usability in terms of the user’s mental effort and their attitude. The third view is one of performance. This is a view measuring usability by examining the user’s interaction with the product. Within this view, two different aspects can be emphasised: ease-of-use and acceptability. Ease-of-use means how easy the product is to use. Acceptability means how useful the product would be in the real world. Finally, there is the contextually-oriented view. This means the product’s usability depends on three aspects at the same time: the user, the task, and the environment. The best definition of usability, would be one that fits all of these views. In this section, the goal is to find one that does.

The ISO standard for software qualities [International Organization for Standardization, 1991] defines usability as follows:

“a set of attributes of software which bear on the effort needed for use and on the individual assessment of such use ...”

This is a product- and user-oriented view of usability.

The definition of the ISO ergonomics, proposed by Brooke at al [Brooke et al., 1990], is user-oriented and contextually oriented:

“the effectiveness, efficiency and satisfaction with which specified users can achieve specified goals in a particular environment”

Eason [Eason, 1998] defines usability as follows:

“the degree to which users are able to use the system with the skills, knowledge, stereotypes and experience they can bring to bear”

This definition is ease-of-use-oriented.

All these definitions fit one or more of the previously mentioned views from Nigel Bevan, Jurek Kirakowski and Jonathan Maissel [Bevan et al., 1991], those views being a product-oriented, user-oriented, performance-oriented or contextually-oriented view. However, none of them fit all of them. So how do we find a definition that does fit all of the views? ESPRIT MUSiC [Bevan et al., 1991] defines usability as follows:

“the ease of use and acceptability of a system or product for a particular class of users carrying out specific tasks in a specific environment; where ‘ease of use’ affects user performance and satisfaction, and ‘acceptability’ affects whether or not the product is used.”

This definition fits all views, therefore it is a sufficient definition of usability.

The figure 18 shows the relationships between factors in this definition.

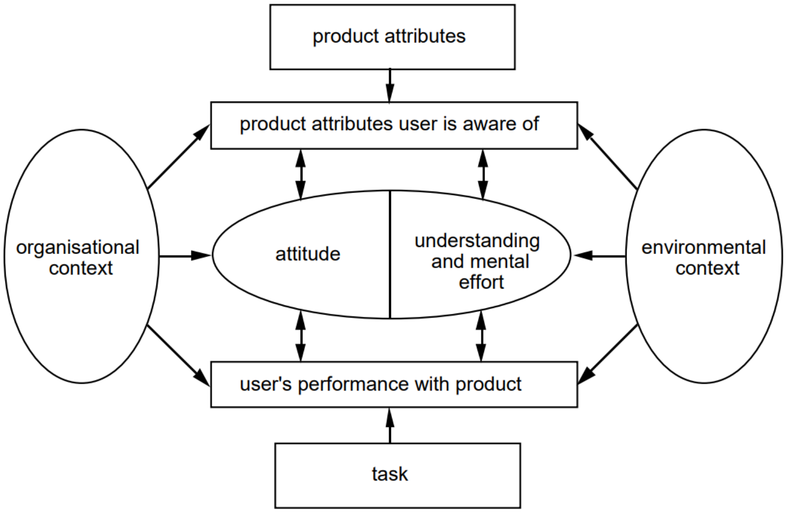

product attributes
product attributes user is aware of
organisational context
attitude
understanding and mental effort
environmental context
user's performance with product
task

Figure 18: The relationships between factors of usability, from [Bevan et al., 1991]

3.2.3 Defining usability problems

Now that usability is defined, a new question arises: What is a usability problem? To find the answer to that question, it is first necessary to consider from which views of usability problems arise, and which views of usability are instead impacted by them.

The first view of usability, is the product-oriented view: the quality of the ergonomic attributes of the product. Usability problems arise from those attributes, so this view can be used to find usability problems. The second view is the user-oriented view: measuring usability in terms of the user’s mental effort and their attitude. This is a view that is impacted by usability problems, and not one from which problems arise. The third view is the view of performance: measuring usability by examining the user's interaction with the product. The two aspects of this view are acceptability, meaning how useful the product would be in the real world, and ease-of-use, meaning how easy the product is to use. Both of these aspects are impacted by the usability problems. The final view of usability, is the contextually-oriented view: the combination of the user, the task and the environment. Usability problems arise from the interaction of these aspects and the product, so this view can also be used to find usability problems. The context does not change when usability problems are present. The user, task and environment stay the same, no matter the usability problems. Therefore, the contextually-oriented view of usability is not impacted by usability problems.

Now, a definition of usability problems can be formed, using all these views as either something that causes usability problems, or something that is impacted by them: "Usability problems are elements of a product that arise from the quality of the ergonomic attributes of the product and the context it is used in, and interfere with the user's mental effort and attitude and the product's ease-of-use and acceptability."

3.3 Measuring usability

Sandeep Napa, Michael Moore and Tania Bardyn did a study called "Advancing Cardiac Surgery Case Planning and Case Review Conferences Using Virtual Reality in Medical Libraries: Evaluation of the Usability of Two Virtual Reality Apps"[Napa et al., 2019]
In this study, two VR applications were evaluated. The applications were meant to reduce surgical errors. A usability study was conducted, to "evaluate the ease of use, physician attitudes toward VR technology, and viability for presurgical case planning". The method they used was a concurrent think-aloud protocol. They used both the System Usability Scale (SUS)[Brooke, 1995] and the National Aeronautics and Space Administration-Task Load Index (NASA TLX)[NASA, 1986] to evaluate user satisfaction. They also evaluated the VR experience with a post-study questionnaire. Before entering the VR environment, participants were asked a few questions regarding their job and VR to find out about their past experiences. Before starting the applications, participants got a few minutes to get familiar with VR itself. The three before mentioned elements of usability "effectiveness", "efficiency" and "satisfaction" (subsubsection 3.2.2) were all evaluated in this study. Effectiveness was measured by completion rate, efficiency was measured by the time it took to complete a task, and satisfaction was measured by the System Usability Scale and the National Aeronautics and Space Administration-Task Load Index.

3.3.1 System Usability scale

The System Usability Scale is a tool for measuring usability, consisting of 10 questions, each with 5 possible answers. The answers range from "strongly agree" to "strongly disagree". It was originally created by John Brooke in 1986, and it is widely used. [usability.gov, nd]
The benefits of using SUS, are that it is a scale that is easy to administer for participants, can give reliable results for small sample sizes, and it differentiates between usable and unusable effectively. One of the drawbacks of using SUS, is that it is a complex scoring system. The scores are easily confused with percentages, and the scores have to be normalized to percentages to be used in that way. It also does not show diagnostics, so one does not always know what to do to improve the

score.

SUS always has the following 10 questions:
1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

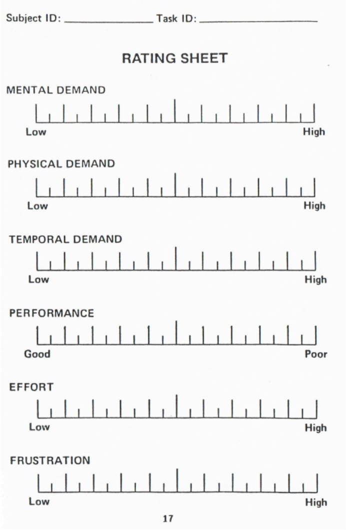
The responses range from “strongly disagree” to “strongly agree”, with five options in total. The options are given a number of points from 1 to 5. For the odd questions, one point is subtracted (score = answer - 1). For the even questions, the score is subtracted from 5, and the result is the final score for the question (score = 5 - answer). These points are added, and multiplied by 2.5 (final score = sum of all scores * 2.5). Now there is a score of 0-100. A score of 68 is considered average, so one has to be careful not to interpret the result as a percentage.[UserSense, 2021b]

3.3.2 NASA task load index

Another tool, is the NASA Task Load Index
[Agency for Healthcare Research and Quality, nd]. This is a tool to examine the mental workload of a task. These are the six dimensions that are measured:
1. Mental demand: how much thinking, deciding, or calculating was required to perform the task.
2. Physical demand: the amount and intensity of physical activity required to complete the task.
3. Temporal demand: the amount of time pressure involved in completing the task.
4. Effort: how hard does the participant have to work to maintain their level of performance?
5. Performance: the level of success in completing the task.
6. Frustration level: how insecure, discouraged, or secure or content the participant felt during the task.

The participant answers these questions on a scale from 0 to 100, where 0 is “low” and 100 is “high” for most questions. The only difference is “performance”. For “performance”, 0 is “good” and 100 is “poor”. The way the user gives their rating, is by indicating the score on a horizontal line divided in 20 intervals. At the end of every interval and in the beginning, is a tick mark. This means there are 21 tick marks. The user selects one of these marks to give their score. If the user marks between two ticks, the score will be rounded up to the next tick. This way, the user gives a rating between 0 and 100, with intervals of 5.

After giving these scores, the participants get 15 cards, shown in random order, with on each card 2 dimensions. The participants have to circle the one that contributed most to their workload. For each dimension, the total number of selections is divided by 15. The final score is calculated by multiplying these weights by their ratings. [NASA, 1986]



Subject ID: ____________ Task ID: ________________

RATING SHEET

MENTAL DEMAND

Low High

PHYSICAL DEMAND

Low High

TEMPORAL DEMAND

Low High

PERFORMANCE

Good Poor

EFFORT

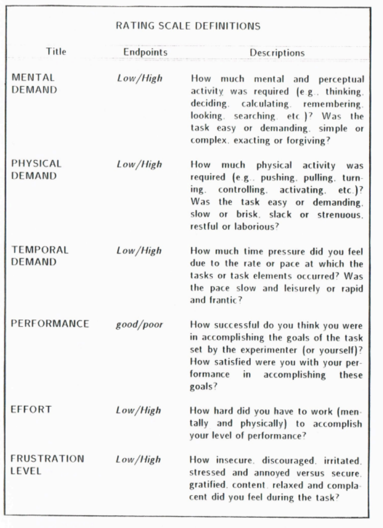
Low High

FRUSTRATION

Low High

17

Figure 19: TLX rating sheet, from [NASA, 1986]



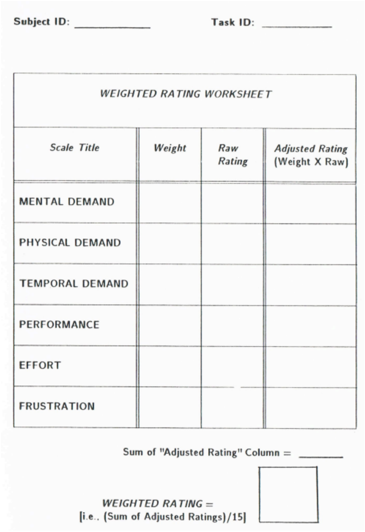
RATING SCALE DEFINITIONS

Title	Endpoints	Descriptions
MENTAL DEMAND	*Low/High*	How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?
PHYSICAL DEMAND	*Low/High*	How much physical activity was required (e.g., pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
TEMPORAL DEMAND	*Low/High*	How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?
PERFORMANCE	*good/poor*	How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?
EFFORT	*Low/High*	How hard did you have to work (mentally and physically) to accomplish your level of performance?
FRUSTRATION LEVEL	*Low/High*	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Figure 20: Definitions of TLX dimensions, from [NASA, 1986]

3.3.3 Net Promoter Score

The Net Promoter Score (NPS) is a tool to measure customer loyalty. It only contains one question: “How likely would you be to recommend [company/product name]?” The answer is a number from a scale from 0 to 10. This answer can categorise the respondent in one of the following three categories:



Subject ID: __________ Task ID: __________

WEIGHTED RATING WORKSHEET

Scale Title	*Weight*	*Raw Rating*	*Adjusted Rating* (Weight X Raw)
MENTAL DEMAND			
PHYSICAL DEMAND			
TEMPORAL DEMAND			
PERFORMANCE			
EFFORT			
FRUSTRATION			

Sum of "Adjusted Rating" Column = ______

WEIGHTED RATING =
[i.e., (Sum of Adjusted Ratings)/15] □

Figure 21: TLX counting sheet, from [NASA, 1986]

Detractors: people who answer 0-6. These people may damage the company or product's reputation by telling people about their bad experiences.
Passives: people who answer 7-8. These people are content, but will probably not damage or improve the company's or product's reputation.
Promoters: people who answer 9-10. These people are likely to improve the company's or product's reputation, by recommending it to others.

To calculate the Net Promoter Score, the following formula is used: (number of promoters-number of detractors)/total respondents. The result will be a number between -100 and 100, where higher is better.

Using the NPS has advantages. The first of these, is that collecting data is easy. Respondents only have to answer one question, and can do it wherever and whenever they want. The second advanatage, is that there is a strong correlation between NPS and revenue growth and customer loyalty. Finally, NPS has a strong correlation with the System Usability Scale.

There are some points to look out for when using NPS. The first one, is that one should always try to find out what the average NPS is for similar companies or products. This is because some sectors always have low NP scores, meaning the customer is just as unsatisfied at the competitor, and is unlikely to leave. Another thing to keep in mind, is that NPS gives no insight in the reason for a customer's disappointment. This can partly be solved by allowing the respondent to explain their score, but it is best to combine NPS with another type of usability testing. Finally, a disadvantage of NPS is that sometimes progress is not measured. This is because the range for detractors is so big, that a shift from 0 to 6 is not measured. [UserSense, 2020a]

3.3.4 Single Ease Question

Like the NPS, the Single Ease Question (SEQ) is only one question: "Overall, how difficult or easy was this task?" The answer is a scale from 1 to 7, with 1 being the most difficult and 7 the easiest. This question is asked after every task. This data is not processed to find some other score.

There are three main advantages to SEQ. The first advantage, is that it is very clear which components need improvement. The second, is that the answer is accurate, because it is asked directly after completing the task. The third advantage, is that it is a short and simple question.

A disadvantage to using SEQ, is that the user may give higher scores to later tasks, because of the experience gathered in the previous tasks.

The SEQ correlates with the task-completion rate. This means tasks that take much time, are often viewed as difficult.

Like with NPS, the SEQ does not show any reason for why a score is given. The solution is the same as for NPS: ask the user to motivate their answer, and combine SEQ with other metrics, like SUS.

The average SEQ score is 5.5.[Sauro, 2018] Knowing this, can give a better view of how easy of hard the task was.
[UserSense, 2021a]

3.3.5 Task completion rate

Usersense [UserSense, 2020b] suggested using the task completion rate to show how well people could use the application. This score is calculated by dividing the successfully executed tasks by the total number of tasks.

3.3.6 Time to task completion

Another metric Usersense [UserSense, 2020b] suggested, is the time to task completion. It is the average time it took a respondent to execute the task, so the total time divided by the number of respondents.

3.3.7 Error rate

The final metric suggested by Usersense [UserSense, 2020b], is the error rate. This is the total number of errors made divided by the product of the total number of possible errors and the number of respondents. A problem with this metric, is that it is hard to define what a possible error is, and how many there are.

3.4 Priority scores

Usersense [UserSense, 2020b] suggested that usability issues should be stored in a table, writing down what the issue is, who had the issue, what task it was and where in the application it happened. Issues can then be given a score to define importance. There are three categories in which a score can be given:
The critical score: the importance of the functionality in question. Rate this on a scale from 1 to 5.
The frequency ratio: the amount of people that had this issue. Rate this by dividing the amount of people that experienced the issue by the total number of respondents.
The impact score: the extent to which the problem impacted the respondent. Rate this on a scale from 1 to 5, using the following guidelines: a suggested improvement is a 1, a small problem is a 2, a big problem is a 3 and a blocker is a 5.
After calculating all this scores, a priority score can be calculated. This is done by multiplying the three scores.

3.5 Think-aloud studies

A common type of usability study is a think-aloud study, abbreviated to a TA study. In these studies, participants think aloud for the entire study, so the researches can know all their relevant thoughts. Ercisson and Simson found that reliable data can be produced with certain kinds of verbal reports [Ericsson and Simon, 1980]. This is a common justification for the use of TA studies. From this data, the most reliable are the verbalizations that only require cognitive processing for the performance of the task and its verbalization.[Lewis, 2014]

The main reason researchers use TA studies, is that they are very productive for finding usability problems [Van den Haak and de Jong, 2003]; [Virzi et al., 1993]. There is evidence that thinking aloud should not affect a user's performance [Bowers and Snyder, 1990]; [Ohnemus and Biers, 1993]; [Olmsted-Hawala et al., 2010]. However, there is also evidence indicating the opposite. According to Berry and Broadbent [Berry and Broadbent, 1990], thinking aloud can invoke cognitive processes that improve task performance. TA study and silent usability study were compared by Wright and Converse [Wright and Converse, 1992]. They found that the TA group performed better. The difference between the groups increased with task difficulty. According to MacDonald, McGarry and Willis [MacDonald et al., 2013], differences in TA protocols impact task complexity.[Lewis, 2014]

If we compare TA to silent participation, we can see that an important difference is that TA can affect task performance and reported satisfaction, while silent participation cannot. This can be a negative for TA, because the scores it would produce for task performance and reported satisfaction would be too high. This would be a problem if measuring task performance or satisfaction would be the goal of the study. However, if problem discovery is the primary goal, TA can be advantageous over silent task completion. So which method one should use, depends on the goal of the study.[Lewis, 2014]

3.6 Conduct study at home versus in lab

A usability study can either take place in a lab or in the user's natural environment. The advantages of doing the study in a lab are that the researcher has full control over the situation, and can intervene when necessary. The advantage of doing it in the user's natural environment, is that the results are truer to how they would be in the real world, because the user uses the product exactly like they would in the real world. [UserSense, 2022a]

3.7 Moderated versus unmoderated study

Another thing to consider is moderation. A study can be done with or without a moderator. A moderator can help and ask questions. The advantage of having a moderator is that they can ask questions based on the user's actions, to get more information about certain aspects that are otherwise missed. The disadvantage is that the user's experience with a moderator is different from the experience the users are going to have with the end product. [UserSense, 2022a]

3.8 Sample sizes

When doing usability tests, there is always the question what sample size is best. There are some common rules of thumb for this. 5 is the best known magic number [Nielsen, 2000] [Nielsen and Landauer, 1993]. 8 [Perfetti and Landesman, 2001] [Spool and Schroeder, 2001] and 10 [Hwang and Salvendy, 2010] are also common.

3.8.1 Sample sizes for finding usability problems

First, the right sample size for finding usability problems is discussed. To find the right sample size, Lewis [Lewis, 1982] came up with the idea to use the cumulative binomial probability formula: $P(at\ least\ one\ occurrence) = 1 - (1-p)^n$ In this formula, p is the likelihood that a usability problem occurs, and n is the sample size. Nielsen and Landauer [Nielsen and Landauer, 1993] found that on average p=.31. They decided that the best value of n is 5, because then the probability for each error to occur at least once is 85%. This means that 5 people are needed to find 85% of all problems.

However, this only works if p=.31, and that is not true for all studies. To find the right sample size, we need to find the value of n in $.85 = 1 - (1-p)^n$. Unless we want another percentage than 85% of course. So that changes the formula to $discoveryGoal = 1 - (1-p)^n$. To find n, we rewrite the formula to $n = ln(1 - discoveryGoal)/ln(1-p)$.[Lewis, 2014]

When using this formula, one should keep in mind that this only works for similar users doing similar tasks. For example, when testing an application that is meant for two types of users, more than n people are needed to find dicoveryGoal percent of problems. What is needed, is n people from group one, and n people from group two. So for example, when testing an application meant for selling and buying used items, one needs a group of buyers and a group of sellers, because the use the application differently, and will find different problems. [UserSense, 2022b]

3.8.2 For measuring usability

For measuring usability, another sample size is better. According to [Lewis, 2014], a sample size of 30 is a common rule of thumb for measuring usability. The theory behind using this sample size, is the central limit theorem. This theorem states that if the sample size increases, the distribution of the mean gets more and more normal. This distribution is close to normal when the sample size is at least 30.

4 The usability study

4.1 The chosen definition of usability

In subsection 3.2, multiple possible definitions of usability were discussed. Here, a definition is chosen that will be used for this study.

As was mentioned in subsection 3.2, there are four views of usability. The four views are product-oriented, user-oriented, performance-oriented and contextually-oriented. The performance-oriented view contains two different aspects: ease-of-use and acceptability.

The only definition that fits all these views, is the one from ESPRIT MUSiC [Bevan et al., 1991]: "the ease of use and acceptability of a system or product for a particular class of users carrying out specific tasks in a specific environment; where 'ease of use' affects user performance and satisfaction, and 'acceptability' affects whether or not the product is used."

For usability problems, the definition from subsubsection 3.2.3 is used: "Usability problems are elements of a product that arise from the quality of the ergonomic attributes of the product and the context it is used in, and interfere with the user's mental effort and attitude and the product's ease-of-use and acceptability."

4.2 How the study will be conducted

The study will be a think-aloud study, because that is the best way to find usability problems. The downside is that performance can be a bit higher this way than it would be in the real world (where the users do not think aloud), but that is a acceptable compromise for better feedback on the issues. The screen and microphone will be recorded, so the context of their thoughts will be visible.

To be able to see the impact of using this application while solving problems, there will also be a group that tries to solve the same issues, but without using the application. They will be allowed to use Quiver, so a comparison can be made.

For the location (lab versus at home), a lab study is chosen. The reason for this, is that the lab's environment does not negatively influence the user's response, as long as no-one intervenes, while the equipment surely works correctly.

On the subject of moderators, the best option for my study is to do it without a moderator, because it is important that the user will figure everything out themselves, so they cannot get input from a moderator.

A lesson from the surgery study, was that it is a good idea to let the users get familiar with VR before starting the actual study, so the time they need to finish a task is not influenced by being unfamiliar with VR. For this reason, the users will get an introduction in VR before starting the study.

This research does not cost much, so we can have a big sample size. Using the formula mentioned before: $n = ln(1 - discoveryGoal)/ln(1 - p)$, filling in 0.31 for p and 0.99 for dicoveryGoal, we need a sample size of n=12. Finding 99% of usability issues is probably enough, so this is the goal for the sample size. The target audience of the application, is people who need to solve problems with category theory diagrams. This is only one group, so there is no need to increase the amount of testers further.

Keep in mind that those 12 people are only meant for finding usability issues. To actually measure usability, and to be able to compare the group to a control group that is not using the application, we need a lot more people. As mentioned before, 30 is a good rule of thumb for this. So we need 30 people to measure usability, from which 12 are also used for finding problems. Another 30 people are used as a control group.

Even though 30 people would be ideal, it may be hard to find enough test subjects with that goal. If this goal cannot be achieved, testing with less people is also a possibility. It is important to then keep in mind the results are less reliable. Using the formula, the amount of usability problems found can still be estimated.

To measure usability, the System Usability Scale (SUS), the NASA task load index, the Net Promoter Score (NPS), the Single Ease Question (SEQ), the task completion rate, the time to task completion, and the error rate will be used. For SUS, the scores should not be confused with percentages, but otherwise the before mentioned downsides will not be an issue, because the metric is combined with others. The standard questions will be used, so the score can be compared to other studies. For NPS, the scores cannot be compared to the scores of competitors, because there are not many of them, and that data is not available for this study. This means NPS is not very useful, but because it is only one question, it does not hurt to use it anyway. The user will also be asked to explain their answer to this question. The same goes for SEQ.

Priority scores will be used to make a prioritized list of all found issues.

4.3 The procedure

The first thing to do, is to find people to study. It is probably best to ask students and lecturers or research staff from Leiden University. First, they will get an introduction in VR. This introduction is done by making them play the “First Steps” game, which is made by Oculus as an introduction to an Oculus headset. After getting an introduction in VR, they get instructions and some problems to solve. They use the application to do this, and think aloud. The microphone and screen are recorded. There will also be people selected to solve the same problems, without the application, so the results can be compared.

When the results are in, usability will be measured and analysed using SUS, NASA TLX, NPS, SEQ, priority scores, task completion rate, time to task completion and error rate. The recordings are also watched to find additional points to improve upon.

4.4 Instructions

4.4.1 Test group

The 12 people from the test group that are used for finding issues, will get the following instructions:

"In this study, you will use a VR application that could help solving category theory problems. Using your feedback, the application will be reviewed and improved.

Before you start using the application, you will get an introduction to VR in general. This introduction is the application "First Steps". You can start this up yourself, and exit it when you are done.

When you are done with the introduction, start the application. Using this application, solve the problems written below. Please think aloud, so we can get as much feedback as possible. Good luck!"

The users will solve these problems:
[problems to solve]

In the appendices, the questions for SUS and NASA TLX the users will get can be found. For TLX, they will also receive the cards on which they have to indicate the dimensions that contributed most to their workload. Appendix B is from [NASA, 1986].

The users will also get the following questions:

"How likely would you be to recommend this application? Answer on a scale from 0 to 10.
Please explain your answer."

"Overall, how difficult or easy was this task? Answer on a scale from 1 to 7, 7 being the easiest and 1 being the hardest.
Please explain your answer."

The people who are only used for measuring usability, do not need to be told to think aloud, and in the feedback section, they will not be asked to explain answers.

4.4.2 Control group

Like mentioned before, there will also be a control group, so the scores can be compared and the effect of the VR environment can be seen. They will be allowed to use Quiver. They will get the following instructions:

“In this study, you will solve category theory problems. You are part of a control group. The test group solves the same problems, using a VR application.

Solve the problems written below. Good luck!”

[problems to solve]

5 Conclusion

The research question is: “Is it possible to provide a VR environment for category theoretical diagrams, which enhances the collaborative work of scientists that work with such diagrams according to a set of use cases?” This question cannot fully be answered, because the usability study could not be executed, because of a lack of facilities at the university. It is hard to say how usable it is for users that are unfamiliar with it, but that does not mean nothing can be said about it. It can be concluded that it is possible to provide the environment, and from personal experience, it can be said that 3D and intuitive controls help drawing diagrams more easily. The application also has potential to be used outside of the field of category theory, because it can be used for any type of graphs.

In the future, the application could be improved in multiple ways. The first, is hand tracking. With hand tracking, users can control the application without using controllers. The benefit of this, is that users can more easily switch between using the VR environment, and doing things in real life, like typing on a keyboard or writing on paper. The downsides, are that not every headset supports hand tracking, that the users have no buttons, and that the tracking is less accurate.
Secondly, the “collaborative” part of the research question was not explored. This means this is also something that could be done as future work. “Collaborative work” means that users could work together in the same environment, at the same time.
Finally, the application could be improved using AR technology. AR means that the user can see the real-world environment through the headset, and draw diagrams in this environment. This could especially be useful combined with hand tracking, because it would make it easier to use a keyboard while using the application.

References

[Agency for Healthcare Research and Quality, nd] Agency for Healthcare Research and Quality (n.d.). Nasa task load index. https://digital.ahrq.gov/health-it-tools-and-resources/evaluation-resources/workflow-assessment-health-it-toolkit/all-workflow-tools/nasa-task-load-index.

[Alonso-Ríos et al., 2009] Alonso-Ríos, D., Vázquez-García, A., Mosqueira-Rey, E., and Moret-Bonillo, V. (2009). Usability: a critical analysis and a taxonomy. *International journal of human-computer interaction*, 26.

[Berry and Broadbent, 1990] Berry, D. C. and Broadbent, D. E. (1990). The role of instruction and verbalization in improving performance on complex search tasks. *Behaviour & Information Technology*, 9.

[Bevan et al., 1991] Bevan, N., Kirakowski, J., and Maissel, J. (1991). What is usability?

[bol.com, nd] bol.com (n.d.). Oculus quest 2. https://www.bol.com/nl/nl/p/oculus-quest-2-vr-bril-standalone-256gb/9300000020591744/.

[Bowers and Snyder, 1990] Bowers, V. and Snyder, H. (1990). Concurrent versus retrospective verbal protocols for comparing window usability.

[Brooke, 1995] Brooke, J. (1995). Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189.

[Brooke et al., 1990] Brooke, J., Bevan, N., Brigham, F., Harker, S., and Youmans, D. (1990). Usability statements and standardisation - work in progress in iso.

[Eason, 1998] Eason (1998). Information technology and organisational change.

[Ericsson and Simon, 1980] Ericsson, K. A. and Simon, H. A. (1980). Verbal reports as data. *Psychological Review*, 87.

[Gray and Salzman, 1998] Gray, W. D. and Salzman, M. C. (1998). Damaged merchandise? a review of experiments that compare usability evaluation methods. *Human–computer interaction*, 13.

[Hwang and Salvendy, 2010] Hwang, W. and Salvendy, G. (2010). Number of people required for usability evaluation: The 10±2 rule. *Communications of the ACM*, 53.

[International Organization for Standardization, 1991] International Organization for Standardization (1991). Ergonomic requirements for office work with visual display terminals, iso 9241.

[Lewis, 1982] Lewis, J. R. (1982). Testing small system customer set-up.

[Lewis, 2014] Lewis, J. R. (2014). Usability: Lessons learned . . . and yet to be learned. *International Journal of Human–Computer Interaction*, 30.

[MacDonald et al., 2013] MacDonald, S., McGarry, K., and Willis, L. M. (2013). Thinking-aloud about web navigation: The relationship between think-aloud instructions, task difficulty and performance.

[Napa et al., 2019] Napa, S., Moore, M., and Bardyn, T. (2019). Advancing cardiac surgery case planning and case review conferences using virtual reality in medical libraries: Evaluation of the usability of two virtual reality apps. *JMIR Hum Factors*, 6.

[NASA, 1986] NASA (1986). Nasa task load index (tlx) v. 1.0 manual. https://humansystems.arc.nasa.gov/groups/tlx/downloads/TLX_pappen_manual.pdf.

[Nielsen, 2000] Nielsen, J. (2000). Why you only need to test with 5 users. *Alertbox*.

[Nielsen and Landauer, 1993] Nielsen, J. and Landauer, T. K. (1993). A mathematical model of the finding of usability problems.

[Ohnemus and Biers, 1993] Ohnemus, K. R. and Biers, D. W. (1993). Retrospective versus thinking aloud in usability testing.

[Olmsted-Hawala et al., 2010] Olmsted-Hawala, E. L., Murphy, E., Hawala, S., and Ashenfelter, K. T. (2010). Think-aloud protocols: A comparison of three think-aloud protocols for use in testing data-dissemination web sites for usability.

[Perfetti and Landesman, 2001] Perfetti, C. and Landesman, L. (2001). Eight is not enough.

[Sauro, 2018] Sauro, J. (2018). Using task ease (seq) to predict completion rates and times. https://measuringu.com/seq-prediction/.

[Shackel, 1990] Shackel, B. (1990). Human factors and usability. *Human-computer interaction*.

[Spool and Schroeder, 2001] Spool, J. and Schroeder, W. (2001). Testing websites: Five users is nowhere near enough.

[u/Arc8ngel, 2018] u/Arc8ngel (2018). Oculus touch controller diagrams. https://www.reddit.com/r/oculus/comments/8ycgov/oculus_touch_controller_diagrams/.

[usability.gov, nd] usability.gov (n.d.). System usability scale (sus). https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html.

[UserSense, 2020a] UserSense (2020a). Net promoter score (nps) voor meten usability. https://www.usersense.nl/usability-testing/analyseren/net-promoter-score-nps.

[UserSense, 2020b] UserSense (2020b). Usability testen analyseren. https://www.usersense.nl/usability-testing/analyseren.

[UserSense, 2021a] UserSense (2021a). Single ease question als onderdeel van usability-onderzoek. https://www.usersense.nl/usability-testing/analyseren/single-ease-question-seq.

[UserSense, 2021b] UserSense (2021b). System usability scale voor meten gebruiksvriendelijkheid. https://www.usersense.nl/usability-testing/system-usability-scale-sus.

[UserSense, 2022a] UserSense (2022a). Usability testing - wat is het en waarom zet je het in? https://www.usersense.nl/usability-testing.

[UserSense, 2022b] UserSense (2022b). Waarom je voor usability testing vaak meer dan vijf gebruikers nodig hebt. https://www.usersense.nl/usability-testing/hoeveel-gebruikers-nodig.

[Van den Haak and de Jong, 2003] Van den Haak, M. J. and de Jong, D. T. M. (2003). Exploring two methods of usability testing: Concurrent versus retrospective think-aloud protocols.

[Varkor et al., nd] Varkor, Steenkamp, S. C., AndéC, and Corbyn, N. (n.d.). Quiver. https://q.uiver.app/.

[Virzi et al., 1993] Virzi, R. A., Sorce, J. F., and Herbert, L. B. (1993). A comparison of three usability evaluation methods: Heuristic, think-aloud, and performance testing.

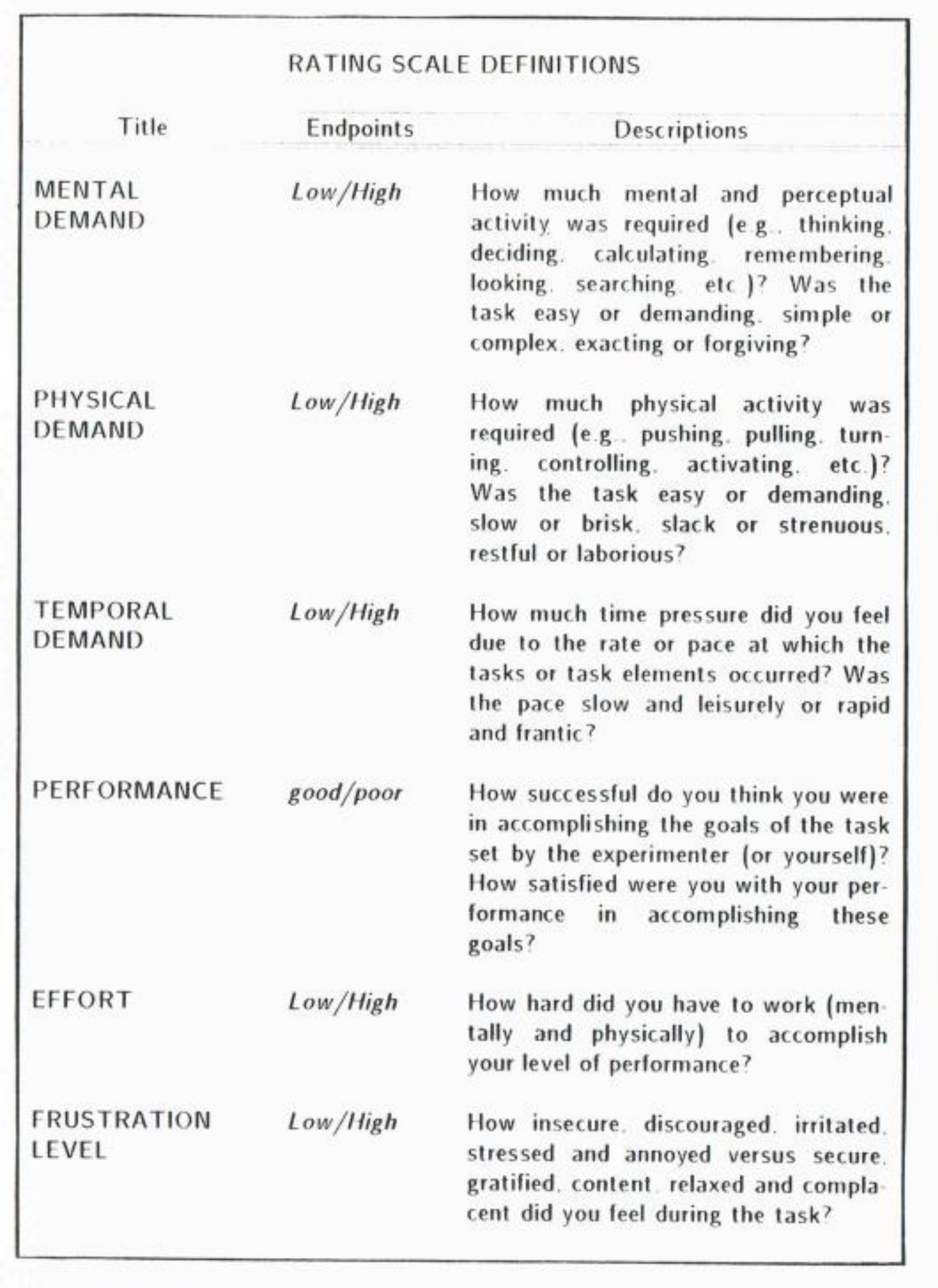
[Wright and Converse, 1992] Wright, R. B. and Converse, S. A. (1992). Method bias and concurrent verbal protocol in software usability testing.

Appendix A

Please answer the following questions:	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently.	○	○	○	○	○
2. I found the system unnecessarily complex.	○	○	○	○	○
3. I thought the system was easy to use.	○	○	○	○	○
4. I think that I would need the support of a technical person to be able to use this system.	○	○	○	○	○
5. I found the various functions in this system were well integrated.	○	○	○	○	○
6. I thought there was too much inconsistency in this system.	○	○	○	○	○
7. I would imagine that most people would learn to use this system very quickly.	○	○	○	○	○
8. I found the system very cumbersome to use.	○	○	○	○	○
9. I felt very confident using the system.	○	○	○	○	○
10. I needed to learn a lot of things before I could get going with this system	○	○	○	○	○

If you want to explain any of your answers, please do below:

Appendix B

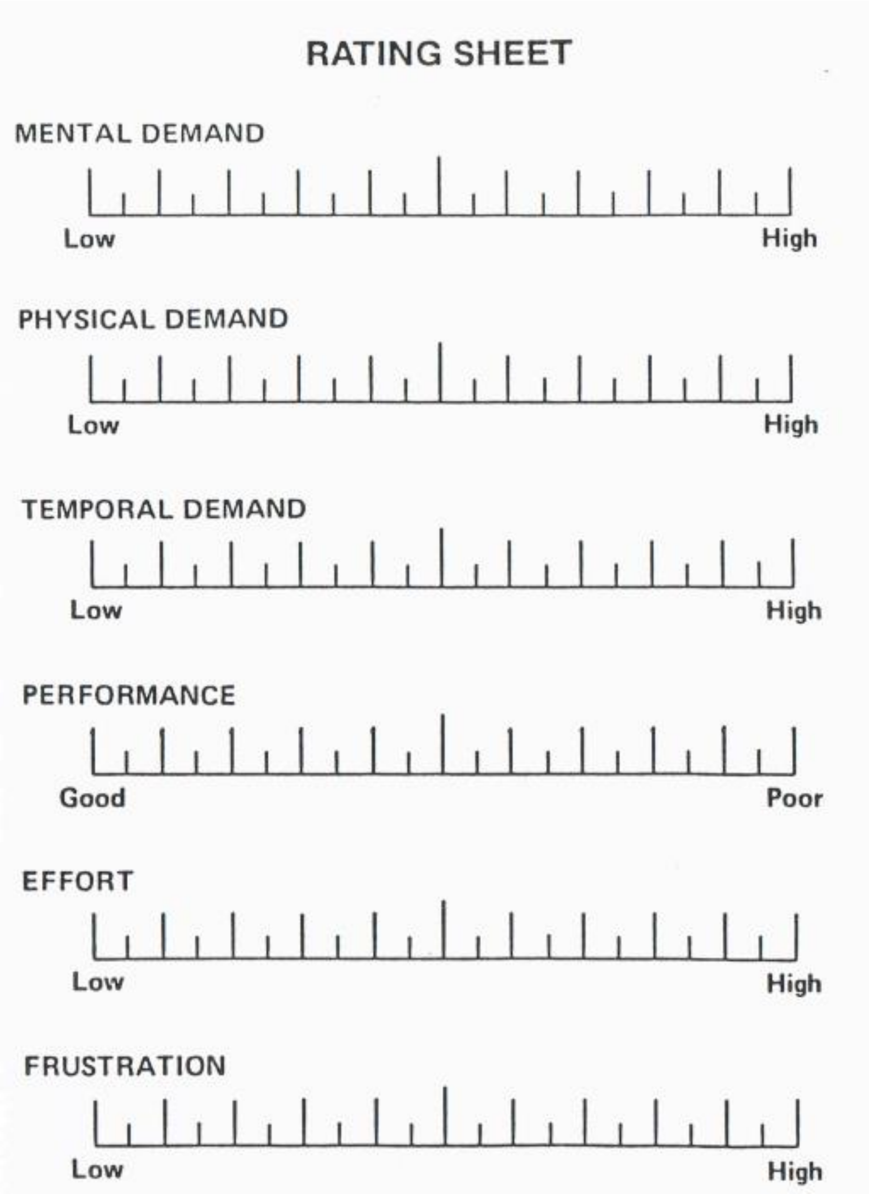


RATING SCALE DEFINITIONS

Title	Endpoints	Descriptions
MENTAL DEMAND	*Low/High*	How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?
PHYSICAL DEMAND	*Low/High*	How much physical activity was required (e.g., pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
TEMPORAL DEMAND	*Low/High*	How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?
PERFORMANCE	*good/poor*	How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?
EFFORT	*Low/High*	How hard did you have to work (mentally and physically) to accomplish your level of performance?
FRUSTRATION LEVEL	*Low/High*	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Appendix B

Please rate your experience by marking a vertical line for each subject. A description of each subject is given on the previous page.



RATING SHEET

MENTAL DEMAND

Low High

PHYSICAL DEMAND

Low High

TEMPORAL DEMAND

Low High

PERFORMANCE

Good Poor

EFFORT

Low High

FRUSTRATION

Low High