Comparison of Wiimote-based interaction techniques within a 3D environment

Written by
Moritz Resl (moritz@moritzresl.net)
Markus Petschnigg (markus.petschnigg@gmx.at)
Anders Spælling (spaelling@gmail.com)
Paul Sommersguter (paul.sommersguter@gmx.net)

June 15, 2010

1 Abstract

This paper compares three new approaches to Wiimote-based 3D interaction techniques for basic spatial object manipulation in a data-independent domain based on user testing. For this purpose we built a three-dimensional environment.

In contrast to previous work on Wiimote-based interaction which was mostly about gesture recognition and machine-learning, we pursued a different approach by providing an application with out-of-the box functionality and evaluated the different techniques using interviews, questionnaires, monitoring, video captures and logfiles.

Even though Wiimote accelerometers are not satisfyingly precise and sensitive enough, we are convinced that our findings can be considered as inspiration for future 3D interaction, where Wiimotes could easily be replaced by similar devices.

Overall, two of three techniques received positive feedback in the majority of aspects like precision, efficiency, physical & mental effort and satisfaction.

Keywords: interaction technique, Wiimote, Wii, 3D environment, comparison, user testing, evaluation

Contents

1 Abstract **1**

2 Introduction **4**
2.1 Related Work 4
2.1.1 General 3D Interaction Techniques 4
2.1.2 The Wiimote 5
2.1.3 Interaction with the Wiimote 6
2.2 Summary 9

3 Interaction **11**
3.1 Technique 1: Single Wiimote 11
3.2 Technique 2: Dual Wiimote / Spinning Top 11
3.3 Technique 3: Dual Wiimote / Distributed Control 14
3.4 Problematic or Abandoned Additional Features 15
3.5 Problematic or Abandoned Interaction Techniques 16

4 Application **18**
4.1 Used Libraries 18
4.2 Design Decisions & Encountered Problems 19
4.3 Visualization 20

5 Evaluation **23**
5.1 Evaluation Goals 23
5.2 Participants 24
5.3 Proposed Tasks 24
5.4 Schedule 26
5.5 Questionnaire 28
5.6 Monitoring 29
5.7 Logged data 29
5.8 Interview Questions 30

6 Results **32**
6.1 Outcome of Interviews & Monitoring 32
6.2 Statistical Analysis 34
6.2.1 Interpreting questionnaire statistics 34
6.2.2 Interpreting log statistics 34
6.3 Conclusion 35

6.4 Future Work 35

A Data Statistics **40**

B ANOVA **50**

C T-tests **53**

D Example Questionnaire **54**

E Participant Timetable **55**

2 Introduction

Since the introduction of the Nintendo Wii in late 2006, a lot of research deals with Wiimote interaction. Before that, only a few papers gave a general overview of what has been done in the area of interaction in a 3-dimensional space.

2.1 Related Work

After a short introduction to general 3D interaction, we dive directly into the newest Wiimote-based interaction design techniques.

2.1.1 General 3D Interaction Techniques

In [25, 33] alternatives to the mouse for navigating in 3D environments are developed and evaluated. Both studies deal with the replacement of the mouse with other interaction methods in 3D environments. The authors of [33] point out the necessity to both develop alternative general 3D interaction metaphors and devices to the rather limited standard mouse while retaining its simplicity. The study made use of a device called *Desktop Bat*, which provided 5 degrees of freedom. The researchers identified two basic tasks of 3D interaction - which are navigation and object placement - and showed a more efficient way to navigate (by a velocity control metaphor) and to rotate/translate objects (applied relative to the eyepoint coordinate system).

A paper by Chris Hand [11] on general 3D interaction techniques identifies three different tasks as being common for 3D interaction: As stated by [33], object manipulation (moving an object) and viewpoint manipulation (changing the users point-of-view) are among these tasks. Furthermore, Hand counts application control (*"...communication between user and system which is not part of the virtual environment"*) as being basic 3D interaction as well.
Another technique is *Action At A Distance*, abbreviated *AAAD*, a method for manipulating an object far away. An example is a virtual laser pointer that origins at the users point-of-view, emitting a straight line. The first object or surface that is intersected by that line is available for manipulation.
AAAD is useful when interacting with menus in 3D, in which the cursor must intersect the appropriate menu item the user wishes to interact with.

2.1.2 The Wiimote

Since its release at the end of 2006, Nintendo Wii's popularity has grown ever since. Recent numbers show that it is one of the most successful gaming consoles so far (67.45 million sold consoles by December 31st, 2009[1]).
The success of the Wii console is *"largely attributable to the innovative interactive technology and game-play by (...) the Wii remote."* [18]. 3D objects can be moved around freely using the Wiimote [6]: *"one of the most obvious benefits of the use of a 3D device is the freedom from the axis constraints."*
Other main features of this cutting-edge controller are gesture recognition combined with pointing. This is made possible by the use of three accelerometers and optical infrared sensors built into the Wiimote, in conjunction with the Wii sensorbar.

The Wiimote also includes a storage capacity of 16 kilobytes, realized with an EEPROM (Electrically Erasable Programmable Read-Only-Memory), a speaker and 8 buttons. Two AA batteries are used as power supply.
Yang-Wai Chow [4] comments that the Wiimote's *"...infrared sensor is convenient to use and has a higher update rate compared to most low-cost web cameras"*. According to [36] the wireless technology in the Wiimote, which is based on tiny wireless transceivers, will play a very important role in the near future.

The Wiimote has been proposed in [36] as a useful tool for educators and researchers not only because of its great variety of analog and digital components, but also for the very low price due to mass production and its detailed documentation that can be found on the internet easily.
The international hacking community successfully reverse-engineered the Wii remote to a large extent; early works of the Wii programming community included mouse cursor and robotic control by the Wiimote, as well as synthesized music performance [18].

In contrast to the set of Wiimote features, the Wall Street Journal published [37] in 2006, which points out that working or playing with the Wiimote could be very unfamiliar for novice users. Also a lot of people complained about aches and pains as a result of intensive use of the Wii controller.

[1] http://www.joystiq.com/2010/01/28/ds-sells-125-million-worldwide-wii-up-to-67-million

Extensive use of mouse and keyboard is known to lead to injuries, and the complaints in [37] could just as well be the result of doing something ones body is not yet used to.

We will now have a closer look on both fairly recent Wii-based interaction projects and, if available, their respective evaluation.

2.1.3 Interaction with the Wiimote

To evaluate the Wiimote, D. Cochard and K. Pham [6] used experienced Wii players, as well as people that had never played the Wii before as participants in a user study in order to *"test the learnability of the system for different levels of Wiimote users"*. In their evaluation they found that natural mapping of the users movement to the on-screen movement using a Wiimote worked very well. They point out, that *"the experienced Wiimote users had no problems with navigating around the viewport, moving in all dimensions. The inexperienced Wiimote users found the 3rd dimension movement and rotation non-intuitive."* Although the accuracy of the Wiimote is limited, gesture recognition was not a problem for any participant.

In further investigations, the Wiimote was used for a project called Unigest [3] by Castellucci and MacKenzie, where the Wiimote's motion capture capability was used for character input without the use of a keyboard and occupation of the display. A user enters a character by holding the B-button and performing its corresponding gesture. Releasing the B-button terminates the gesture, which consisted of one or two moves.
The evaluation was based on a web-based empirical study, where the authors gathered movement time values for each primitive motion.
In contrast to gesture-to-letter mapping, we took advantage of the different gestures by mapping them to rotation and zooming.

The complex maneuvering of a Sony Aibo robot has been simplified and become very intuitive using a Wiimote with a Nunchuk attached in [17] and in [10] also dual Wiimotes. Previous (less successful) attempts using dual gamepads had a steep learning curve, as opposed to the Wiimote, where people of all ages gained full control of the robot in a few minutes after a 15-30 seconds of instruction. The Wiimotes 6 Degrees-of-Freedom was mapped onto the Aibo robots 26 DOF in [17] in which the robot played soccer. An obstacle course was completed with the Aibo robot in [10] followed by a posture imitation. Both papers results support that it is possible to make 3D navigation easy and intuitive using a Wiimote.

Gallo, De Pietro and Marra [9] based their investigations on clinicians feedback, where a 3D user interface to interact with volumetric data should be wireless, ergonomic and suitable for a near-real-time interactivity. The observed object should be able to be manipulated with 6 DOF. The Wiimote device fitted those needs very well.

Schlömer et al. [29] wrote about how to recognize gestures to interact with an application using a Wiimote. The built-in accelerometer of a Wiimote perfectly fit their needs as the received data can be used for gesture recognition. Furthermore, they chose it for its ease of use, hardware price and design. The authors used a somewhat traditional recognition pipeline, which consisted of three components: quantizer, model and classifier. According to the researchers, *"Based on the Wiimote we developed a gesture recognition that employs state of the art recognition methodology such as HMM* (Hidden Markov Models ed.)*, filters and classifiers, and aim to optimize hand gesture recognition for the Wiimote."* The recognition results varied between 85 to 95 percent, which shows that the Wiimote is capable of recognizing basic gestures accurately.

Real-time gesture recognition is discussed in [24]. The Wiimote sends data via Bluetooth at 100 Hz, and Hidden Markov Models are used to process the large number of vector data. A filter is applied removing redundant information to let complex algorithms work on the data in real-time. The implemented system is trained by the user: After 10 repetitions of the same gesture, the program has a high success rate of recognizing gestures.
An accelerometer based gesture recognition interface for 3D environments received good response in [32]. A user study with a Wiimote showed that users preferred to mimic avatar actions using gestures, and buttons for navigation in the game Second Life.
The user study also showed an average gesture recognition rate of 92.9%, [35] had a 96% recognition rate, [15] a 95% recognition rate, and [16] showed a recognition rate of 80.0%. These rates are based on training sets (for machine learning algorithms), and are much lower when gesture recognition is performed

out-of-the-box; The recognition rate dropped from 96% to 50% in [15].
The training period is in [15] 10-20 samples, where each sample took up to 25 seconds to process, an expected average training period of 2.5 minutes per gesture.

There is a major tracking issue described in [18] - *"the tracking data is relatively insensitive to orientation"*, which Lee points out in his article. His work includes finger and object tracking, interactive whiteboards and tablet displays, head tracking for desktop VR displays and spatial augmented reality. All of these projects make use of remote interaction techniques (in particular the Wiimote and sometimes the Wii sensor bar) without the console itself. According to Lee, gesture recognition with the Wii remote is still an open research problem.

The rate of success in some implementations is limited by *Gesture Force Peaks* [8]: *"When a force gesture is made, such as a swinging the Wiimote like a tennis racket there is a peak of force in a direction intended by the user to trigger an action, we call this an intended peak of force. However when this gesture comes to a stop there is peak of force in the opposite direction caused by the deceleration of the Wiimote, we call this a rebound peak."*
Instead of pointing with the Wiimote, which is somewhat inaccurate, [8] suggests using hand flicks as: *"pointing with the Wiimote can become tiresome when long input is required, such as the entering of web addresses."* We find that this could also be an interesting way of interacting with radial menus.

In [5] an interactive, three-dimensional design program for 3D objects has been developed. The creation and design of 3D objects must often be orchestrated using a two-dimensional perspective, which is a notable flaw of current CAD (Computer-aided design) systems. The author, whose work was largely influenced by Johnny Lee's efforts in this field (see for example [18]), took advantage of the Wiimote's sensors to provide a 3D interface to aid object design and implemented a 3-dimensional drawing tool that features the creation, selection and modification of graphic primitives. For instance, rotation of the objects could be performed by processing the data from the Wiimote's tilt sensors. Translation of the graphic primitives was accomplished by pointing the Wiimote to the desired location in space, however, this interfered with the camera control of the drawing tool. The author did not evaluate his software, therefore it is not clear how well its interaction methods work for users.

Musicians interest in the Wiimote has been growing as many demo videos on the Internet arose and audio environments became available, among them for instance *GlovePIE* and *DarwiinRemote OSC* [13].
According to [22] critics of computer music performed on a laptop argue that *"performances of this nature fail to engage audiences as many performers use the mouse and keyboard (...) leaving no visual cues"*. These interfaces do not communicate their task: there is no clear correlation between visual gestures and musical outcomes. A Wiimote in combination with a Nunchuck was used for a realtime electroacoustic musical performance to make this correlation clear. However, the authors did not reach any formal conclusion about the controllers qualities in realtime performance.

A case study on implementing HCI techniques for the evaluation of musical controllers has been done in [14]. For this purpose, a Wiimote was evaluated by a think-aloud-test and compared to a *Roland HandSonic*, a controller which features drum pads for triggering, and knobs for continuous control.
Among the results of the study was comments that the Wiimote lacked physical feedback in the Triggering task, and that it was hard to determine the triggering point. A comment on the ergonomics of the device during the pitch task: *"going past certain points of rotation felt unnatural"*. The fun aspect of the Wiimote was experienced, but also a general lack of precision, even when participants preferred the *Roland HandSonic*.

In [30] a multiple Sensor Bar system has been developed. The Wiimote and the Nunchuck controllers were used in an Immersive Projection Theatre, that renders correctly across its two angled walls. A fast-paced, first-person-shooter game, whose engine was modified to work appropiately with the Wiimote, was tested on this system and gained *"favourable response from experienced games enthusiasts"*. Users enjoyed this setup after extended amounts of time.

2.2 Summary

During our literature study we have found surprisingly few papers dealing with 3D-interaction devices in a conventional 3D application. Most studies including the Wiimote are based on gesture recognition, often for the sake of recognizing the gesture. The majority of papers also suggest the Wiimote as being an intuitive tool for most people, even for complex tasks with little to no training in advance.

These papers were a great inspiration to us, and we especially found it interesting to map gestures to aid interaction in a 3D application.

3 Interaction

The main purpose of this project was to compare our proposed Wiimote-based interaction techniques. In the first technique, only a single Wiimote is needed, whereas the second and third use both Wiimotes.

In this section, the term *cube* refers to the one and only available 3D object the user can manipulate. Depending on the task (see section 5) up to 8 cubes can be manipulated. For evaluating these techniques, we wrote our own test application (see section 4). In order to compare the techniques, the design of these was intended to always support the following kinds of manipulation: movement, zooming and rotation.

We now describe all three techniques in detail, then we talk about problematic and abandoned features and techniques.

3.1 Technique 1: Single Wiimote

- **Pin cube**: After pointing at the cube and clicking the A-button, the cube is selected. It is then pinned at its dead center. This is needed to rotate the cube.
- **Rotate**: The rotation is performed by spinning (hold down A-button and move the cursor) the pinned cube. It is possible to rotate the cube freely.
- **Grab cube**: After pointing at the cube and clicking the B-button (notice the difference to pinning), the cube is ready to be moved.
- **Move**: The movement itself is performed by moving the Wiimote corresponding in the desired direction. If one wants to zoom, one have to press the A and B buttons at the same time while moving the Wiimote closer to or further away from the sensor bar.

3.2 Technique 2: Dual Wiimote / Spinning Top

The main difference to the first interaction configuration lies in the use of dual Wiimotes and distinct ways of zooming and rotation.
For this technique we came up with a spinning top metaphor: we used the way one would normally start the rotation of a spinning top (fast motion with one's

Task	**Technique**
Pin cube center	Point at cube and hold A button
Grab cube	Point at cube and hold B button
Move	Grab and move Wiimote
Rotate	Pin cube and move Wiimote
Upscale (Movement on Z-axis)	Hold A and B button and move Wiimote away from the sensor bar
Downscale (Movement on Z-axis)	Hold A and B button and move Wiimote towards the sensor bar

Table 1: Single Wiimote

thumb and index finger in opposite directions) as an inspiration.

For zooming, we thought about the well-known pinch-to-zoom multitouch-gesture (as seen on most touchphones and PDAs) and implemented it for two Wiimotes as well. This combination creates a higher level of immersion, as one is making physical efforts while using this technique. For the tasks **"Pin cube"** and **"Grab cube"** please see description of interaction technique 1.

Task	**Technique**
Pin cube center	Point at cube and click A button
Grab cube	Point at cube and hold B button
Move in X/Y-plane	Grab and move righthand Wiimote (with use of the sensor bar)
Upscale (Movement on Z-axis)	Grab and move both Wiimotes away from each other (in X/Y-plane) (figure 3.1)
Downscale (Movement on Z-axis)	Grab and move both Wiimotes towards each other (in X/Y-plane) (figure 3.1)
Rotate cw (around X-axis)	Pin cube, hold both B buttons, then move left Wiimote up to the ceiling, and right Wiimote down to the floor (figure 3.2)
Rotate ccw (around X-axis)	Pin cube, hold both B buttons, then move left Wiimote down to the floor, and right Wiimote up to the ceiling (figure 3.2)
Rotate cw (around Y-axis)	Pin cube, hold both B buttons, then move left Wiimote towards and right Wiimote away from screen (figure 3.3)
Rotate ccw (around Y-axis)	Pin cube, hold both B buttons, then move left Wiimote away from and right Wiimote towards screen (figure 3.3)

Table 2: Dual Wiimote / Spinning Top

- **Move**: Moving a cube on the X/Y-plane is performed by grabbing a cube and moving the right Wiimote, just like it is done in technique 1. For zooming (movement on Z-axis) the well-known multi-touch zooming gesture is applied. Grab and move both Wiimotes away from each other (in X/Y-plane) for upscaling and vice-versa for downscaling.
- **Rotate**: Rotation around either axis is implemented by moving both Wiimotes in opposite directions like starting the rotation of a spinning top with your fingers.

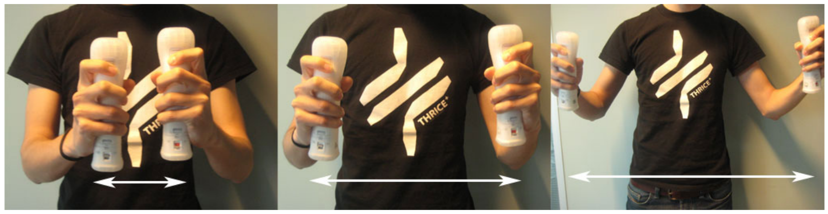


Figure 3.1: Spinning Top: Scale/Zoom

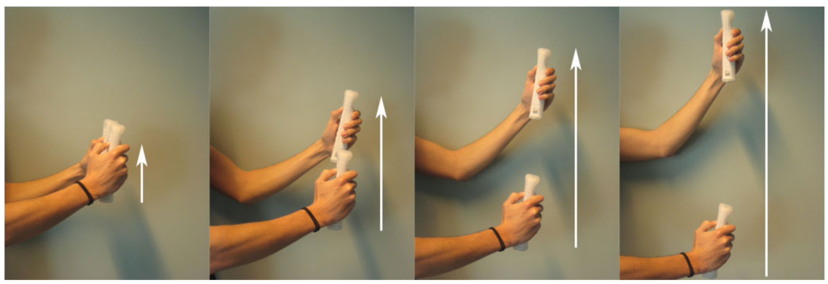


Figure 3.2: Spinning Top: Rotate X-Axis

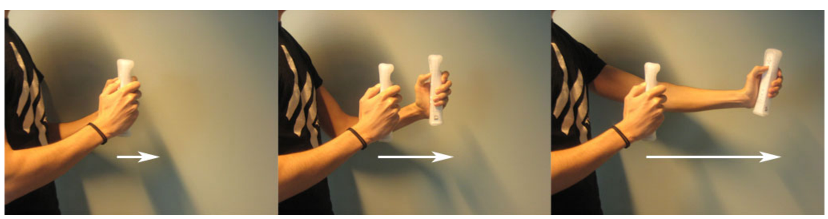


Figure 3.3: Spinning Top: Rotate Y-Axis

3.3 Technique 3: Dual Wiimote / Distributed Control

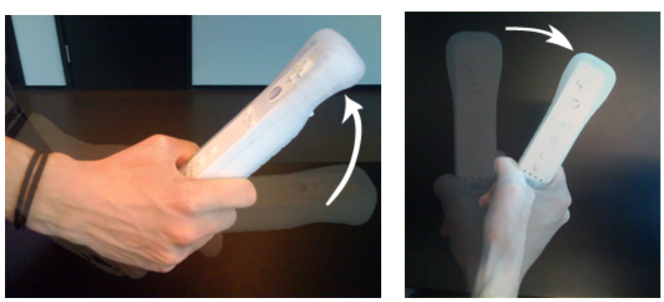
This technique is a distribution of object manipulation control.

The basic idea is to use one Wiimote for movement only (right hand for movement and zooming) and the other one for rotation, instead of using both hands at the same time for a single form of object manipulation (opposed to the rotation in technique 2). For the tasks **"Grab cube"**, **"Move"**, **"Upscale"** and **"Downscale"** please refer to description of technique 1, for **"Pin cube"** refer to technique 2.

- **Rotate**: Rotation around the X- and Y-axis is accomplished by twisting ones wrist (pitching and yawing), moving the Wiimote in a pendular-like motion.

Task	**Technique**
Pin cube center	Point at cube and hold A button of the left Wiimote
Grab cube	Point at cube and hold B button of the right-hand Wiimote
Move in X/Y-axis	Grab and move right-hand Wiimote (with use of the sensor bar)
Upscale (Movement on Z-axis)	Hold A and B button and move right Wiimote away from the sensor bar
Downscale (Movement on Z-axis)	Hold A and B button and move right Wiimote towards the sensor bar
Rotate cw (around X-axis)	See figure 2.4 (a)
Rotate ccw (around X-axis)	See figure 2.4 (a)
Rotate cw (around Y-axis)	See figure 2.4 (b)
Rotate ccw (around Y-axis)	See figure 2.4 (b)

Table 3: Dual Wiimote / Version 2



(a) X-axis (b) Y-axis

Figure 3.4: Rotation

3.4 Problematic or Abandoned Additional Features

Z-rotation was not possible in this application since the library we were using for rotation was simply not supporting a rotation around the global Z-axis. Only for technique 3 we found a suitable metaphor for Z-rotation and this is what it would have looked like:

- **Z-rotation in technique 3**

Task	Technique
Rotate cw (around Z-axis)	Turn left-hand wrist clockwise
Rotate ccw (around Z-axis)	Turn left-hand wrist counter-clockwise

Table 4: Z-rotation for technique 3

- **Swiping**: Swiping with the Wiimote in this context means a short rapid movement of the Wii, like holding a pencil and swiping over the screen. Compare the short rapid movement to the one performed on a trackpad with one's fingers. Instead of pointing directly at the cubes, the selection jumps to the nearest cube relative to the cursor and the direction of the swipe. Although we actually implemented swiping, we did not use it in our test-application. The primary reason for doing so was the surprisingly high accuracy of cursor pointing, hence it was not needed after all. Initially, we wanted to enable swiping for all three interaction techniques.

3.5 Problematic or Abandoned Interaction Techniques

Wiimote / Blowgun

The major difference to our actual interaction techniques lies in the use of an enhanced pointing device: a so-called Blowgun. According to its configuration it will either blow or suck air in a direction originating from the pointer. The blowing and suction power is affected by the distance to the target. We did not pursue this particular idea because translating the positioning of the Wiimote to the positioning of the blowgun onscreen was deemed too difficult within the given timeframe.

- **Pin cube**: The idea is basically to point at a cube while having a mode where the blowgun is off. Then the pinning is realized by clicking the A-button. Thus, a cursor is needed, that has potential of navigating within the three-dimensional space. This can be realized by dividing the depth (i.e.: Z-axis) into different layers.

- **Jump to cube**: Realized by swiping.
- **Rotate**: Once the cube is pinned to its dead center, it can be rotated by pointing at it with the blowgun and blow/suck air. It should then be rotated according to where the air hits the cube.
- **Move**: Movement is realized by blowing/sucking air on the cube.

Task	**Technique**
Pin cube	Blowgun off, hover over the cube and click button A
Jump to cube	Swipe Wiimote
Rotate	Pin cube and blow at cube
Move	Blowing at a cube moves it away from blowing direction/towards suction direction
Blow/Suck	Press button B
Change blowing/suction direction	Hold button A and rotate wiimote - direct mapping between Wiimote and blowgun
Move blowgun	Move Wiimote
Increase blowing power	Directional pad up
Decrease blowing power	Directional pad down
Increase suction power	Directional pad down
Decrease suction power	Directional pad up

Table 5: Wiimote / Blowgun

Rolling Cubes through 3D space

Initially we thought about rolling the cube through a 3-dimensional space by giving it an inital push on one of its faces. A big problem was the rather constrained movement of the cube because it is not possible to move the cube along more than one axis. Another issue was the placement of the cursor in the third dimension. Sometimes the cubes might have overlapped which would have complicated matters of selection.

4 Application

For user testing we built our own application in order to be able to compare our proposed interaction techniques, since there was no suitable applictaion available.
The application was entirely built in Java 1.5 & Processing 1.0. The main reason for using Processing was the lightweight graphics pipeline, which enabled us to make a both appealing and powerful visualization. Since Processing is tightly bound to Java we had the possibility to exploit the set of available libraries for both languages. Furthermore, the bluetooth protocol made it possible to connect Wiimotes to our computers.

Our application consists of a three-dimensional space with cubes that can be moved and rotated. In order to maintain a neutral data domain and take advantage of the third dimension, our implemented objects are cubes. On each of their faces distinct textures are placed. Thus, an obvious reason for object manipulation (i.e. rotation) was provided.

In this section, we present all used libraries, some crucial design decisions, encountered problems and furthermore the visualization itself, consisting of the three different graphical environments (i.e. numbers, colors, photos) as well as menus.
Different enviroments helped users to remember techniques visually and to easily recognize the switches between these.

For anyone interested in the source code of this project it can be found at google code[2].
Additionlly there are two demo videos available[3] [4].

4.1 Used Libraries

To provide an adequate set of functionality, we made use of various additional visualization, data transfer and debug libraries.

[2]https://wiimote-based-interaction.googlecode.com/
[3]http://www.vimeo.com/11298844
[4]http://www.vimeo.com/12080216

- Processing: Processing[5] has been used for implementing the graphics pipeline. We did not use it as a stand-alone application, but as an IDE-integrated Java-extension.
- Wiigee Wiimote Plugin: Wiigee[6] is a Java-based open-source library for accelerometer-based gesture recognition for the Wiimote. Furthermore it featured the identification of Wiimote events.
- BlueCove Bluetooth Library: Bluecove[7] provided us with essential bluetooth funcionality for a proper Wiimote connection and data transfer.
- shapes3d Visualization Library: shapes3d[8] is a re-written version of Apache Commons Math for Processing. Vector and matrix operations are supported, as well as a suitable picking-algorithm (used for selection of cubes). Moreover, texturing the cube faces was possible.
- controlP5: controlP5[9] provided us with a basic set of common User-Interface components (i.e.: buttons, sliders, bangs etc.). Main purpose was the visualization of a graphical user interface for changing various parameters during execution.

4.2 Design Decisions & Encountered Problems

While implementing the test enviroment we encountered several problems and had to make crucial design decisions, listed below.

- Since the shapes3d library does not support free rotation with the cursor (only camera movement around an object), we had to implement our own method of rotation. Internally, the shapes3d library calculates rotations with quaternions, instead of Euler-angles. An object's orientation directly correlates to the movement of the cursor, which makes rotation convenient enough to use it with a Wii controller. Unfortunately, it was not possible to rotate objects around a global z-axis due to some library related restrictions.
- In the beginning, we used rectangles instead of cubes. We then decided to change the object's shape to a cube in order to properly exploit the

[5] http://processing.org/
[6] http://www.wiigee.org/
[7] http://bluecove.org
[8] http://www.lagers.org.uk/s3d4p/index.html
[9] http://www.sojamo.de/libraries/controlP5

third dimension. To have indiviual textures on each face provided us with a larger variety within our test-environment.

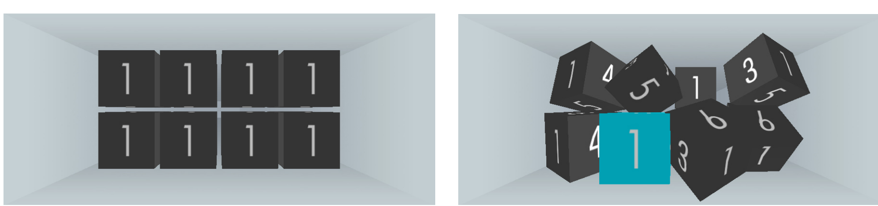
- We made several decisions concerning the visualization: We used proper lightning and color gradients on the canvas' textures to provide a more in-depth three-dimensional look-and-feel. We also made sure to stick to best-practices for graphical user interface design by giving visual feedback (i.e.: selection of a cube changes its color)

- Wiimotion Plus did not provide the angle as expected, instead it gave the angular velocity. Because of that it was not possible to compute the angle between the initial position (Wiimote lying on a table) and current position and we substracted gravity only on the y-axis.
  This resulted in a less precise recognition of gestures as they would never be performed perfectly (gravity also affects other axes). Had we been able to retrieve the exact angle of the Wiimote, we could have split gravity into force-vectors.

- While implementing technique 2, we encountered the problem of a so-called *rebound peak*, described in section 2.1.3.
  We solved the problem by ignoring acceleration data from the Wiimote for 500 ms whenever a *rebound peak* was detected, essentially ignoring it.

4.3 Visualization

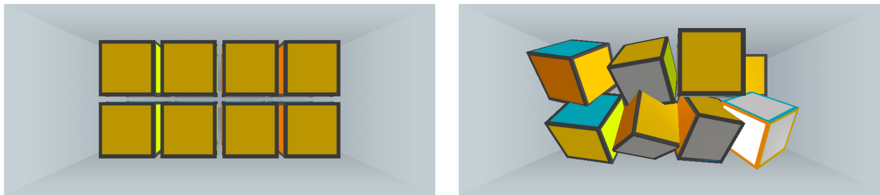
We made sure that the visualization is appealing yet not too much distracting or confusing. We designed a fairly simple but elegant environment.
The following figures show the three different environments, followed by the main menu - which was used to switch between techniques and environments - and the settings menu.



(a) Unmodified

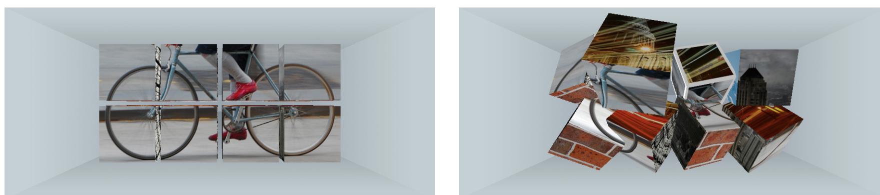
(b) Modified

Figure 4.1: Number Environment



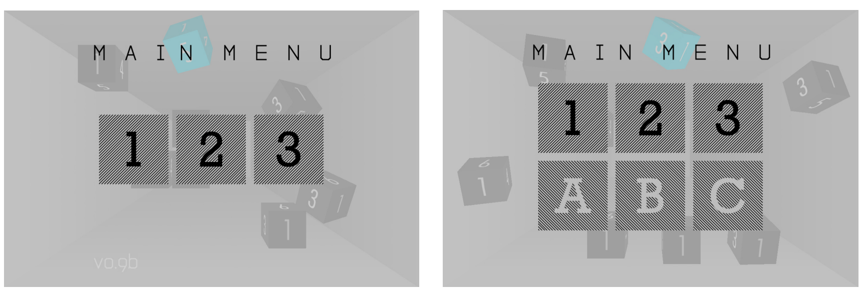
(a) Unmodified (b) Modified

Figure 4.2: Color Environment



(a) Unmodified (b) Modified

Figure 4.3: Photo Environment



(a) Early (b) Updated

Figure 4.4: Main Menu

Figure 4.5: Settings menu

The settings menu only serves internal needs and is not supposed to be operated by the participants. Despite the name, its primary use is the configuration of individual tasks. One can change the interaction method, choose the mode (ie. numbers, colors, photos), start settings (configuration files), turn logging on/off, re-initialize the Wiimotes, show the default configuration and start the task with the applied settings.

5 Evaluation

For our evaluation we pursued a mixed approach. We had 9 participants testing our application and performed 2 training tasks followed by 3 test tasks, for each interaction method.
The first 2 training tasks in which the participant was given the possibility to get familiar with the given technique and try out movement and rotation. The first training task was a single cube, whereas the second task was increased to four cubes, with 1 out of place.

Every test task included both translating and rotating two cubes starting from a specified preset. Depending on a predefined preset, participants were then asked to re-organize the cubes into a more sorted view (see examples in section 5.3). We used semi-structured interviews for the evaluation. We decided to use this method because we wanted to have the possibility of gaining more information than we would have from only using a questionnaire.

In addition, we logged application data like performed rotations or elapsed time (see section 5.7 for further details) during each task in an effort to gain further insights.

5.1 Evaluation Goals

- Find out which technique is preferred by the majority of participants, based on interviews.
- Find out which technique is the most efficient in terms of elapsed time used for fulfilling the tasks, based on logged data.
- Find out which technique requires the least physical effort.
- Find out which technique requires the least mental effort.
- Compare all three techniques in terms of precision and draw conclusions from our observations, based on monitoring, questionnaires and interviews, supported by logged data.
- Get an idea of the learning curve of each technique.
- Obtain additional (technical) results based on logged data (i.e.: number of movement events, number of rotation events, etc.)

5.2 Participants

Most of the participants of our evaluations were international students. In order to get a more diverse spectrum of testers, we succeeded in trying to find both experienced Wii-users and participants who were unfamiliar with the Wii. The participants ages ranged between 20 and 35 years. We had 5 female and 4 male participants.

5.3 Proposed Tasks

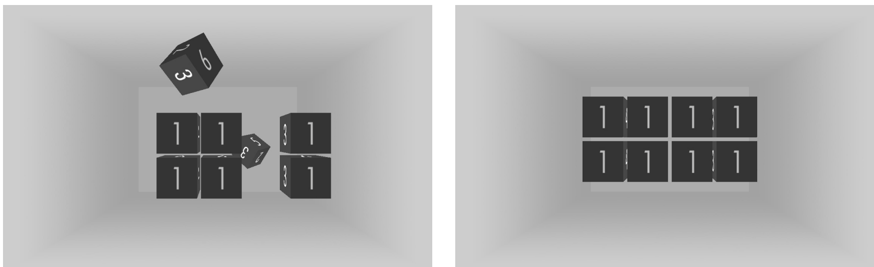
Our user study primarily focused on a comparison of all three interaction techniques.
Within our given environments (numbers, colors and photos) we had five different presets for cube placement and rotation in the canvas. The term *preset* refers to a set of cubes with a distinct starting position and orientation. The first two presets were used for training purposes while the last three were for actual testing (only data gained from the actual test tasks was taken into account for the final results).
We preferred presets over random start settings in order to ensure the same difficulty of each task among the participants.
Depending on each preset, several modifications had to be performed on the objects to accomplish the goal, shown on screenshots projected on a wall during user testing. In every testing task two cubes were modified in terms of rotation and movement.
For the learning tasks we told the participants to take as much time as they wanted for learning the interaction technique.
Please refer to section 3 for a more detailed description interaction techniques. For additional information about the application please refer to section 4.

List of tasks:

- Learning 1 (1 cube):
  One cube just for learning movement, rotation, zooming.
- Learning 2 (3 cubes):
  Perform a single operation on each of the three cubes (movement, rotation, zooming). Arrange them correctly in order to fit the other cubes order.
- Test 1, 2, 3 (8 cubes):
  You see eight cubes. Two of them are not in the right order. Arrange

them correctly in order to fit the other cubes alignment (as shown on the screenshot).

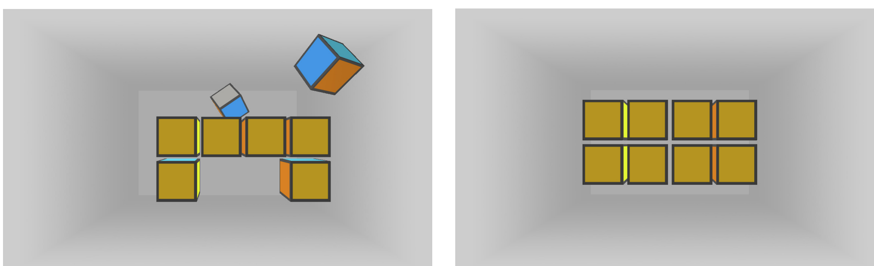
Number Environment



(a) start preset (b) solved

Figure 5.1: Number Environment

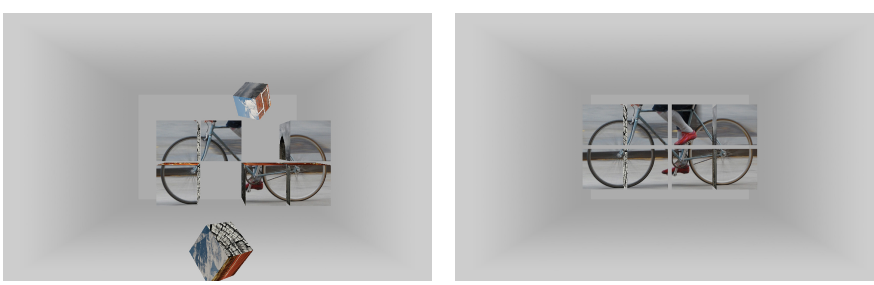
Color Environment



(a) start preset (b) solved

Figure 5.2: Color Environment

Photo Environment



(a) start preset (b) solved

Figure 5.3: Photo Environment

5.4 Schedule

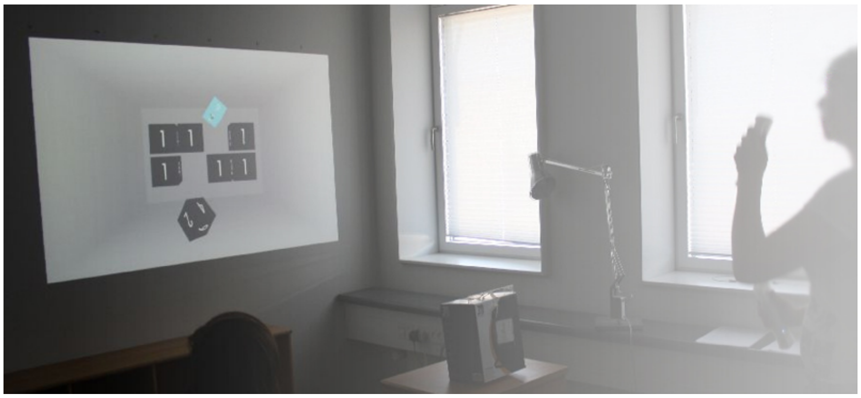


Figure 5.4: participant performing a task

The evaluation took place on three consecutive days, which were divided into several timeslots lasting one hour each. Every participant was assigned to a single timeslot. The test procedure for each participant was like this: At the very beginning we introduced our application in order to give a general overview and present the purpose of our project.

After a short explanation about both how to use the Wiimote(s) within the given technique and the task itself, the test participant had a few minutes in the two training tasks to get used to the setting. When he/she felt comfortable, three more test tasks were performed.
After every technique the participants were asked to fill out a questionnaire. At the end of the testing phase the participants were interviewed.

We permuted the order of environments and techniques for every test participant. For example, participant 7 started with technique 3 in the color environment while participant 2 started with technique 1 in the photo environment (see table below for detailed view).

Permuting techniques and environments eliminated the possibility of one enviroment favoring a particular technique.
Furthermore, this eliminated similar learning-curves. For instance, if technique 3 had always been the last technique tested, we would have gained inconclusive

Participant	Technique - Environment		
p0	1 - Number	2 - Color	3 - Photo
p1	1 - Color	2 - Photo	3 - Number
p2	1 - Photo	2 - Number	3 - Color
p3	2 - Number	3 - Color	1 - Photo
p4	2 - Color	3 - Photo	1 - Number
p5	2 - Photo	3 - Number	1 - Color
p6	3 - Number	1 - Color	2 - Photo
p7	3 - Color	1 - Photo	2 - Number
p8	3 - Photo	1 - Number	2 - Color

Table 6: Evaluation order

data because participants would have gotten too much training during the two previous techniques.

Detailed time schedule for a test participant:

- General introduction to the application and the Wiimote (2 min)

  First technique

- Train the first technique by using the first two presets (5 min)
- Solve three tasks: presets three, four and five (10 min for all three)
- Answer short questionnaire (2 min)
- Short switch of interaction technique (0.5 min)

  Second technique

- Train the second technique by using the first two presets (5 min)
- Solve three tasks: presets three, four and five (10 min for all three)
- Answer short questionnaire (2 min)
- Short switch of interaction technique (0.5 min)

  Third technique

- Train the third technique by using the first two presets (5 min)

- Solve three tasks: presets three, four and five (10 min for all three)
- Answer short questionnaire (2 min)

- Interview (15 min)

Handing out a questionnaire immediately after completing all tasks for a single technique ensures that the participants can still recall first impressions. For details see sections 5.5 and 5.8.

5.5 Questionnaire

Right after performing the three test tasks every participant was asked to fill out a short questionnaire (see Appendix D). All questions had to be filled out on a scale ranging from 1 to 9. We designed the questionnaire with inspiration from the *QUIS*-template proposed by Ben Shneiderman[10].

- Overall, the technique was:
  - 1 (wonderful) to 9 (terrible)
  - 1 (satisfying) to 9 (frustrating)
  - 1 (flexible) to 9 (rigid)
- General difficulty of the tasks was:
  1 (very easy) to 9 (very difficult).
- Mental effort required for operation was:
  1 (none) to 9 (big).
- Physical effort required for operation was:
  1 (none) to 9 (big).
- How precise was the technique:
  1 (precise) to 9 (imprecise).

[10]http://lap.umd.edu/quis/

5.6 Monitoring

In order not to base our whole evaluation on interviews, logged data and questionnaires, we were monitoring the participants while they were working with our application. We did this to get some impressions and to be able to adapt the interview after the test accordingly. In addition, we screen-captured every user test, and took screenshots of their respective solutions.

We paid attention to the following aspects:

- Learning-speed:
  - How long did it take the participants to be able to work with the Wiimote properly?
  - Was there a method that caused more troubles than others?
- Working-speed:
  - With which method could the tasks be solved the fastest?
- Precision:
  - Were there a lot of problems due to lack of precision?
- Comments:
  - How often did the participant ask for help?
  - What did the participant say while testing?

5.7 Logged data

The following events were logged for each completed task:

- Elapsed time: The time it took the participant to perform a specific task (eg. arranging the cubes in the photo environment until a coherent image appears)
- Rotation events: The sum of all rotation events
- Zoom events: The sum of all zoom events
- Move events: The sum of all translation events

- Acceleration events: The sum of all acceleration events
- Button click events: The number of times a button was clicked by the user
- Zoom distance: The sum of the distance the user moved cubes in the Z-plane (measured in pixels)
- Mouse distance: The distance the mouse cursor moved in the X/Y-plane (measured in pixels)

An event is understood as something created by an external entity, that the application "listens" and reacts to. For instance clicking a button on the Wiimote would create a button click event and the application would react according to which button was pushed, while shaking the Wiimote would create multiple acceleration events (tens of events per second), with each event carrying data on how much force was applied to the Wiimote for that event.
By logging this data, it is possible to compare the different interaction methods based not only on interviews. The data can show the differences and similarities between techniques that are not inherently obvious.

5.8 Interview Questions

Each participant was interviewed after the user test, in order to gain additional information besides that which was gathered from the questionnaires. The participants were asked a sub-set of the following questions, always depending on how much was already answered:

- Which technique did you prefer? Why?
- Which one was the most or the least precise? What were the differences?
- Which one was the most or the least fun to use?
  What were the differences?
- If unexperienced: Did you find it difficult to learn how to use the Wiimote?
- If experienced: Did you benefit from previous knowledge?
- Was there a technique that was more difficult to learn than the others?
- What do you think, which method was the most efficient?
  With which method could you solve the tasks the fastest way?

- Which one of those techniques didn't you like at all? Why?
- Any additional comments?

6 Results

6.1 Outcome of Interviews & Monitoring

Technique 1:
This technique was regarded by most participants as the easiest form of interaction. The reasons for this lie in the perceived high degree of both precision (sometimes even too sensitive, since it was hard to rotate and move the cubes accurately). The single-handed Wiimote was preferred by most for professional work, since it appeared to be very intuitive (a reasonable way of moving and shifting the cubes), responsive and fast.
Even though most participants lacked previous experience with the Wii, they did not have difficulties learning the technique.
Despite the fact that two participants faced physical exhaustion, it was still regarded as the smoothest and best pointing device.
Furthermore, one participant stated this technique to be the most fun. In contrast to that, another one mentioned that the single Wiimote gets boring after some time.

Technique 2:
This technique was a dual Wiimote technique and it was by far the worst perceived one according to the feedback we received from our participants. Most participants had serious problems remembering the controls and the fact that one had to point at the cube in order to pin it caused severe troubles. The majority of the participants did not point at the cubes before trying to pin them.
Since all of our techniques lacked controls for left handed participants, one participant had serious problems performing the rotation properly. In order to fulfill the given task the rotation then had to be performed in reverse.
A participant suggested the boxing metaphor, which was used for rotation should better be used for zooming instead, because *"it would feel more natural"*, according to him. Most participants recognized the missing z-rotation, but could handle the work around. The learning phase was highly appreciated and *"there was a good learning curve between the first and second task"*.

Furthermore, one participant mentioned technique two as being the most difficult to learn and the most exhausting, and another even had problems with selecting cubes. The movement was perceived as being okay and did not get

commented that much. Only two participants specifically mentioned it, whereas zooming was perceived as being either imprecise or the person could not get it to work at all. One participant stated that *"it zooms out when I don't want it to"*.
Generally speaking, only few participants said it was the most fun to use, because of the high degree of integration of one's body, one stated that *"although the technique doesn't work that well, I liked the general idea of it. It almost felt like being in Minority Report."* For most, it was the hardest technique to learn, since one constantly had to switch between modes of the Wiimotes (when a cube is pinned in which rotation could be performed but not moved and vice versa) and the fact that it took them the longest to figure out how to achieve the task specifik transformations. One participant mentioned that it was the most precise technique out of the given three, because one could move and shift the cubes quite slowly.

Technique 3:
As opposed to the first dual Wiimote technique, the second dual Wiimote technique in general received positive feedback. Four out of nine participants clearly mentioned this technique as their favorite for the following reasons: one participant liked the intuitiveness, the others highlighted its low level of difficulty, its high level of entertainment and one pointed out that 'it makes sense' to interact like this. The clearly understandable way of allocating a single possible manipulation to either the left hand (rotation) or the right (movement and zooming) was also among the reasons for the popularity of technique three, stated by two participants. One participant mentioned the level of convenience and that it was always clear how to attain a certain state, another described this technique as fun. Out of the remaining five, three participants stated that this technique (together with technique one) was their preferred way of interacting with the test application. While they mentioned that they may prefer technique one for professional work, they would choose technique three for entertainment reasons.

As in the other techniques, some participants clearly missed a possibility of z-rotation, which made rotation in general more difficult and time consuming, as they needed to figure out a work-around. Two participants handled that very well. Although one participant had troubles figuring out the right timing to release the A-button while rotating upwards, the participant liked the way of interaction. While rotating, two participants were confused by the fact that the mouse cursor is still visible although it actually is not needed for rotation. Concerning precision, three participants mentioned something: One participant found technique 3 as being a little bit too precise, another stated it feels more "natural", than the other techniques, and the third said it worked well and enabled very precise rotations and zooming.

Of all participants, only one had problems with the usage of it; this person did not have much prior Wii-experience. The participants mainly chose technique 3 as their respective favorite for its entertainment value, but some of them even liked it better than technique 1 in all aspects.

6.2 Statistical Analysis

Only some of our data is normal distributed with 90% confidence, resulting in a mix of ANOVA and t-tests, and some data was simply not possible to test. The null-hypothesis for all ANOVA and t-tests is that the mean value of all/both samples are equal. All ANOVA and t-tests are done with a 95% confidence.

6.2.1 Interpreting questionnaire statistics

In appendix B table 42 and 43 shows that the F-value is less than $F_{0.05}$ for 2 and 26 DoF and we cannot reject the null hypothesis, and therefore conclude that all techniques are perceived as being equally wonderful/terrible aswell as requiring the same amount of mental effort to use.
In appendix C the t-tests show that we can reject the null hypothesis: that both technique 1 and 2 was equally satisfying/frustrating and equally flexible/rigid. We conclude that technique 1 was both more satisfying to use aswell as more flexible based on the fact that it has a lower median value.
On the other hand we cannot reject the null-hypothesis: that technique 2 and 3 requires the same amount of physical effort.

6.2.2 Interpreting log statistics

Both ANOVA tests for the data logs (appendix B) and t-tests (appendix C), show that only mouse distance and mouse events was the same for all techniques. We therefore conclude that there is a significant difference in the way users perform the different techniques.

6.3 Conclusion

In contrast to our expectations which consisted of having the single Wiimote configuration (technique 1) as the favorite, it was surprising that the second dual Wiimote configuration (technique 3) received even more positive feedback. According to the interviews, most participants preferred technique 3 over technique 1 & technique 2, but the data analysis showed that all three techniques were equally on a scale ranging from 1 (wonderful) to 9 (terrible) and required the same amount of mental effort. Technique 1 turned out to be the most flexible and most satisfying and would be the choice to get actual work done. In terms of fun, technique 2 and technique 3 outrun technique 1. Participants chose those two for entertainment reasons, which shows that they could be used for gaming purposes. As expected, both dual Wiimote techniques required higher physical effort than the single Wiimote pendant. Technique 1 required less than half the time of technique 2 for fulfilling all three tasks. While the mean for the first one was around 80 seconds, the mean for the second one was at around 181 seconds. Technique 3 required the participants around 125 seconds. Concerning the learning curve, it turned out that technique 1 was the easiest to learn, but all three techniques improved equally task by task.

We got the impression that most participants were positively biased and used to give a higher score than expected. Our user investigation would have benefited from a bigger sample of participants in order to draw more precise conclusions. We would have gained more expressive data about the learning curve by using more tasks, because all three techniques converge.

6.4 Future Work

Concerning the implementation of the techniques, further work could include the realization of Z-rotation (see section 3), which would mean using a different 3D library other than shapes3d for drawing objects.
The user-study showed that our proposed techniques would benefit from some fine-tuning. For example, an easier to remember mode-switch between rotation and movement could make technique 2 a lot more convenient. We are convinced that our presented interaction techniques could serve as an inspiration for future work in the field of accelerometer-based 3D interaction.

References

[1] Benjamin B. Bederson. PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. *In Proceeding of UIST*, pages 71–80, 2001.

[2] João Bernardes. Integrating the Wii Controller with enJine: 3D Interfaces Extending the Frontiers of a Didactic Game Engine. *Best of SBGames 2007*, 7(12), 2009.

[3] Steven J. Castellucci and I. Scott MacKenzie. UniGest: Text Entry Using Three Degrees of Motion. *CHI '08 extended abstracts on Human factors in computing systems*, pages 3549 – 3554, 2008.

[4] Y.W. Chow. The Wii Remote as an Input Device for 3D Interaction in Immersive Head-mounted Display Virtual Reality. *Proceedings of IADIS International Conference Gaming 2008*, pages 85–92, 2008.

[5] C. Clingerman. Interactive 3D Design Using Nintendo Wii Remote. 2009.

[6] D. Cochard and K. Pham. Navigation in a 3D environment: Using the Wiimote to Map Natural Gestures.

[7] Steven M. Drucker et al. MediaBrowser: Reclaiming the Shoebox. *Proceedings of the working conference on Advanced visual interfaces*, pages 433–436, 2004.

[8] Zachary Fitz-Walter, Samuel Jones, and Dian Tjondronegoro. Detecting Gesture Force Peaks for Intuitive Interaction. *Proceedings of the 5th Australasian Conference on Interactive Entertainment*, (2), 2008.

[9] Luigi Gallo, Giuseppe De Pietro, and Ivana Marra. 3D Interaction with Volumetric Medical Data: Experiencing the Wiimote. *Proceedings of the 1st international conference on Ambient media and systems*, 1(14), 2008.

[10] Cheng Guo and Ehud Sharlin. Exploring the Use of Tangible User Interfaces for Human-robot Interaction: a Comparative Study. *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 121–130, 2008.

[11] Chris Hand. A Survey of 3D Interaction Techniques. *Computer Graphics Forum*, 16(5):269–281, 1997.

[12] Christian Hansen et al. Intraoperative Modification of Resection Plans for Liver Surgery. *International Journal of Computer Assisted Radiology and Surgery*, 3:291–297, 2008.

[13] Chris Kiefer, Nick Collins, and Geraldine Fitzpatrick. Evaluating the Wiimote as a Musical Controller. *Proceedings of the 2008 International Computer Music Conference (ICMC)*, 2008.

[14] Chris Kiefer, Nick Collins, and Geraldine Fitzpatrick. HCI Methodology For Evaluating Musical Controllers: A Case Study. *Proceedings of Int Conf of New Interfaces for Musical Expression, NIME08, Genoa, Italy)*, 2008.

[15] Louis Kratz, Matthew Smith, and Frank J. Lee. Wiizards: 3D Gesture Recognition for Game Play Input. *Proceedings of the 2007 conference on Future Play*, pages 209–212, 2007.

[16] Sven Kratz and Michael Rohs. A $3 Gesture Recognizer - Simple Gesture Recognition for Devices Equipped with 3D Acceleration Sensors. *Proceeding of the 14th international conference on Intelligent user interfaces*, pages 341–344, 2010.

[17] Micah Lapping-Carr et al. Wiimote Interfaces for Lifelong Robot Learning. *Proceedings of AAAI Symposium on using AI to Motivate Greater Participation in Computer Science*, 2008.

[18] Johnny Chung Lee. Hacking the Nintendo Wii Remote. *IEEE Pervasive Computing*, 7:39–45, 2008.

[19] Andreas Lorenz, Ferry Pramudianto, and Andreas Zimmermann. Lessons Learned from an Interaction-kiosk for Open Selection of Input Devices of a Gaming Application. *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, (62), 2009.

[20] Miller and Freund. *Probability and Statistics for Engineers*. Prentice Hall.

[21] Jozef Mlích. Wiimote Gesture Recognition. pages 344–349, 2009.

[22] Dr. Garth Paine. Interfacing for Dynamic Morphology in Computer Music Performance. *Proceedings of International Conference on Music Communication Science*, (12):115–118, 2007.

[23] John C. Platt, Mary Czerwinski, and Brent A. Field. Phototoc: Automatic Clustering for Browsing Personal Photographs. *Technical Report MSR-TR-2002-17*, 2003.

[24] Benjamin Poppinga. Beschleunigungsbasierte 3D-Gestenerkennung mit dem Wii-Controller. 2007.

[25] G. Robertson, M. Czeminski, and M. van Dantzich. Immersion in Desktop Virtual Reality. *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 11–19, 1997.

[26] G. Robertson et al. The Task Gallery: A 3D Window Manager. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 494–501, 2000.

[27] K. Rodden and K.R. Wood. How Do People Manage Their Digital Photographs? *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 409 – 416, 2003.

[28] Kerry Rodden, Wojciech Basalaj, David Sinclair, and Kenneth Wood. Does Organisation by Similarity Assist Image Browsing. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 190 – 197, 2001.

[29] T. Schlömer, B. Poppinga, et al. Gesture Recognition with a Wii Controller. *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 11–14, 2008.

[30] Torben Schou and Henry J. Gardner. A Surround Interface Using the Wii Controller with Multiple Sensor Bars. *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 287–288, 2008.

[31] Takaaki Shiratori and Jessica K. Hodgins. Accelerometer-based User Interfaces for the Control of a Physically Simulated Character. *ACM Transactions on Graphics (TOG)*, 27(123), 2008.

[32] Sreeram Sreedharan, Edmund S. Zurita, and Beryl Plimmer. 3D Input for 3D Worlds. *Proceedings of the 19th Australasian conference on Computer-Human Interaction: Entertaining User Interfaces*, pages 227–230, 2007.

[33] A. Steed and M. Slater. 3D Interaction with the Desktop Bat. *Computer Graphics Forum*, 14(2):97–104, 1995.

[34] Robyn Taylor, Pierre Boulanger, and Patrick Olivier. dream.Medusa: A Participatory Performance. pages 200–206, 2008.

[35] Paul D. Varcholik and James L. Merlo. Gestural Communication with Accelerometer-Based Input Devices and Tactile Displays. 2008.

[36] J.A. Vicaria, J.M. Maestre, and E.F. Camacho. Academical and Research Wiimote Applications. *IADIS08*, 2008.

[37] Jamin Warren. A Wii Workout: When Videogames Hurt. (11), 2006.

A Data Statistics

Question\Participant	#1	#2	#3	#4	#5	#6	#7	#8	#9
wonderful-terrible	5	4	3	1	3	3	2	4	3
satisfying-frustrating	6	4	2	2	3	3	3	4	3
flexible-rigid	6	3	1	1	3	3	4	2	3
difficulty	3	6	4	1	2	2	4	7	3
mental effort	3	5	4	2	2	2	3	8	3
phys effort	6	5	2	1	7	2	3	6	5
precision	6	3	2	2	2	3	2	3	3

Table 7: questionnaire data for technique #1

Question\Participant	#1	#2	#3	#4	#5	#6	#7	#8	#9
wonderful-terrible	4	4	2	2	3	7	3	6	3
satisfying-frustrating	6	5	2	3	7	8	4	6	8
flexible-rigid	4	3	2	3	9	7	4	5	5
difficulty	6	7	5	3	3	6	4	7	7
mental effort	3	6	5	2	5	4	4	7	5
phys effort	4	6	2	3	7	8	3	7	5
precision	3	5	2	5	7	8	3	8	6

Table 8: questionnaire data for technique #2

Question\Participant	#1	#2	#3	#4	#5	#6	#7	#8	#9
wonderful-terrible	3	2	4	1	3	4	2	3	3
satisfying-frustrating	3	1	4	1	3	4	6	4	6
flexible-rigid	3	2	4	2	3	5	1	3	5
difficulty	2	2	6	1	3	5	4	8	4
mental effort	2	3	4	1	3	4	4	7	5
phys effort	3	3	2	2	7	4	3	8	5
precision	3	2	5	1	2	5	2	6	6

Table 9: questionnaire data for technique #3

Question	median	StdDev	Relative StdDev	Variance	Normal Distributed
wonderful-terrible	3	1.17	37.50%	1.4	X
satisfying-frustrating	3	1.22	36.74%	1.5	X
flexible-rigid	2	1.54	53.19%	2.4	X
difficulty	3	1.94	54.67%	3.8	
mental effort	3	1.94	54.67%	3.8	X
phys effort	4	2.15	52.23%	4.6	
precision	2	1.27	43.94%	1.6	X

Table 10: questionnaire data for technique #1

Question	median	StdDev	Relative StdDev	Variance	Normal Distributed
wonderful-terrible	3	1.72	45.42%	2.9	X
satisfying-frustrating	5	2.13	39.08%	4.5	X
flexible-rigid	4	2.18	46.70%	4.8	X
difficulty	5	1.66	31.09%	2.8	X
mental effort	4	1.51	33.13%	2.3	X
phys effort	5	2.12	42.43%	4.5	
precision	5	2.22	42.58%	4.9	X

Table 11: questionnaire data for technique #2

Question	median	StdDev	Relative StdDev	Variance	Normal Distributed
wonderful-terrible	2	0.97	34.99%	0.9	X
satisfying-frustrating	3	1.81	50.92%	3.3	
flexible-rigid	3	1.36	43.85%	1.9	
difficulty	3	2.20	56.69%	4.9	
mental effort	3	1.73	47.24%	3.0	X
phys effort	4	2.15	52.23%	4.6	X
precision	3	1.94	54.67%	3.8	

Table 12: questionnaire data for technique #3

Participant	median	StdDev	Relative StdDev	Variance
1	129	65	50.34%	4230.9
2	147	99	67.01%	9811.4
3	156	60	38.68%	3681.2
4	103	75	73.00%	5683.4
5	107	31	29.15%	978.6
6	98	73	74.95%	5467.0
7	164	88	53.59%	7797.9
8	167	76	45.71%	5875.8
9	94	37	39.76%	1399.0

Table 13: Elapsed time for each participant

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	92	47	51.63%	2280.4	
2	81	29	35.63%	843.9	X
3	73	28	39.22%	822.9	X

Table 14: Elapsed time for Technique #1

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	204	93	45.98%	8817.5	
2	187	76	40.93%	5901.0	X
3	152	60	39.83%	3711.2	X

Table 15: Elapsed time for Technique #2

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	156	66	42.16%	4378.9	X
2	122	56	46.07%	3164.8	X
3	98	51	52.50%	2675.1	X

Table 16: Elapsed time for Technique #3

Participant	median	StdDev	Relative StdDev	Variance
1	1371	1215	88.62%	1478112.6
2	1423	1173	82.43%	1376262.4
3	2485	2472	99.46%	6112524.3
4	1317	1574	119.48%	2479256.9
5	1690	2236	132.32%	5002274.4
6	1158	850	73.42%	722911.9
7	2396	2200	91.82%	4841016.9
8	1909	1763	92.37%	3110797.3
9	1160	920	79.35%	847464.9

Table 17: Rotation events for each participant

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	3454	1965	56.89%	3861654.0	X
2	3390	1963	57.91%	3855131.8	
3	3222	1989	61.74%	3957051.9	X

Table 18: Rotation events for Technique #1

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	658	480	72.89%	230600.2	X
2	859	601	69.91%	361227.8	X
3	786	506	64.37%	256276.5	

Table 19: Rotation events for Technique #2

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	1096	815	74.42%	665429.9	X
2	777	422	54.38%	178739.3	
3	668	350	52.42%	122599.8	

Table 20: Rotation events for Technique #3

Participant	median	StdDev	Relative StdDev	Variance
1	190582	120679	63.32%	14563509192.5
2	171590	76587	44.63%	5865576870.9
3	232333	140506	60.48%	19742036360.9
4	92969	40017	43.04%	1601425736.4
5	108880	42558	39.09%	1811230183.6
6	121257	54899	45.28%	3014000518.0
7	185710	80325	43.25%	6452128272.3
8	183500	122215	66.60%	14936611846.1
9	132160	51009	38.60%	2601961591.6

Table 21: Mouse distance for each participant

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	170607	128362	75.24%	16477004816.2	X
2	123683	48417	39.15%	2344221599.9	X
3	109355	46708	42.71%	2181672059.7	X

Table 22: Mouse distance for Technique #1

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	189709	103916	54.78%	10798681681.3	
2	146611	60717	41.41%	3686592009.1	X
3	106098	17079	16.10%	291696799.8	X

Table 23: Mouse distance for Technique #2

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	228810	101905	44.54%	10384826258.5	X
2	187183	103260	55.17%	10662735276.5	
3	156925	132827	84.64%	17643131170.9	X

Table 24: Mouse distance for Technique #3

Participant	median	StdDev	Relative StdDev	Variance
1	2514	1473	58.61%	2172601.1
2	3183	1454	45.69%	2115564.5
3	3268	1989	60.88%	3959022.3
4	2618	1927	73.63%	3717159.8
5	3567	2228	62.48%	4968089.5
6	2499	1083	43.34%	1173125.2
7	3798	2981	78.51%	8890580.4
8	5391	3139	58.23%	9856898.0
9	2659	816	30.70%	666500.3

Table 25: Move events for each participant

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	4047	2376	58.72%	5647718.0	
2	3392	1327	39.12%	1761485.1	X
3	2959	1054	35.63%	1111445.3	X

Table 26: Move events for Technique #1

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	2257	1527	67.65%	2331925.3	X
2	2067	682	33.01%	465599.7	X
3	1525	639	41.93%	408914.0	X

Table 27: Move events for Technique #2

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	5291	2460	46.50%	6054489.9	X
2	4160	3155	75.84%	9955205.4	X
3	3800	2158	56.79%	4657118.1	X

Table 28: Move events for Technique #3

Participant	median	StdDev	Relative StdDev	Variance
1	206	81	39.49%	6669.8
2	231	86	37.47%	7514.7
3	240	109	45.61%	12015.9
4	258	207	80.51%	43253.1
5	298	171	57.33%	29255.9
6	166	71	43.19%	5160.7
7	416	404	97.29%	163799.2
8	440	250	56.96%	62951.1
9	256	43	16.79%	1860.2

Table 29: Zoom events for each participant

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	284	164	57.90%	27037.0	
2	235	82	35.10%	6815.4	X
3	221	92	41.75%	8522.9	X

Table 30: Zoom events for Technique #1

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	398	339	85.28%	115597.3	X
2	267	218	81.45%	47607.1	X
3	269	175	65.06%	30778.8	X

Table 31: Zoom events for Technique #3

Participant	median	StdDev	Relative StdDev	Variance
1	1658	1049	63.28%	1101396.8
2	2137	1464	68.54%	2145847.4
3	2391	1697	70.99%	2881084.8
4	2014	2000	99.32%	4002000.9
5	2517	2111	83.86%	4457636.5
6	1998	1109	55.50%	1230328.8
7	2533	2850	112.51%	8127322.4
8	4355	2567	58.95%	6591919.1
9	2162	679	31.41%	461324.2

Table 32: Zoom distance sum for each participant

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	3187	1917	60.16%	3677372.8	
2	2515	960	38.19%	923178.0	X
3	2253	1004	44.60%	1009907.0	

Table 33: Zoom distance sum for Technique #1

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	1210	1511	124.93%	2285854.4	X
2	1152	755	65.58%	570825.9	
3	942	648	68.73%	419913.7	X

Table 34: Zoom distance sum for Technique #2

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	4243	2342	55.20%	5487220.7	X
2	3186	2576	80.86%	6637939.6	X
3	3077	1981	64.39%	3925185.4	X

Table 35: Zoom distance sum for Technique #3

Participant	median	StdDev	Relative StdDev	Variance
1	1880	1147	61.04%	1316829.0
2	2818	427	15.17%	182910.3
3	3466	2402	69.29%	5770070.3
4	3239	2129	65.75%	4536566.3
5	1647	188	11.45%	35589.3
6	2938	1592	54.22%	2537551.0
7	2966	1164	39.27%	1356706.3
8	2793	1773	63.49%	3144432.0
9	1370	178	13.05%	32002.3

Table 36: Acceleration events for each participant

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	2457	1254	51.07%	1574779.0	
2	2845	1668	58.62%	2782956.4	
3	2403	1377	57.31%	1897217.5	X

Table 37: Acceleration events for Technique #2

Participant	median	StdDev	Relative StdDev	Variance
1	111	83	74.72%	6905.7
2	123	116	95.07%	13674.5
3	155	130	84.01%	17005.7
4	90	104	115.80%	10942.5
5	107	77	72.66%	6045.0
6	96	114	118.03%	13047.9
7	154	149	97.04%	22429.0
8	135	132	98.00%	17590.0
9	115	59	51.64%	3560.3

Table 38: Button click events for each participant

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	35	13	38.96%	189.5	X
2	34	10	31.09%	113.2	X
3	33	6	18.61%	38.5	X

Table 39: Button click events for Technique #1

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	216	89	41.18%	7936.5	X
2	274	126	46.14%	16046.0	X
3	226	79	35.02%	6264.3	X

Table 40: Button click events for Technique #2

Task	median	StdDev	Relative StdDev	Variance	Normal Distributed
1	104	57	54.95%	3287.3	X
2	91	46	50.70%	2144.5	
3	73	40	55.83%	1676.5	X

Table 41: Button click events for Technique #3

B ANOVA

$F_{0.05}(v_1 = 2, v_2 = 24) = 3.40$

Source of variation	Degrees of freedom	Sum of squares	Mean square	F
wonderful-terrible	2	4.6667	2.3333	1.33
Error	24	42.0000	1.7500	
Total	26	46.6667		

Table 42: ANOVA for wonderful-terrible

Source of variation	Degrees of freedom	Sum of squares	Mean square	F
mental effort	2	5.4074	2.7037	0.90
Error	24	72.4444	3.0185	
Total	26	77.8519		

Table 43: ANOVA for mental effort

Source of variation	Degrees of freedom	Sum of squares	Mean square	F
Elapsed time, task 2	2	51663.54	25831.77	7.82
Error	24	79277.58	3303.23	
Total	26	130941.12		

Table 44: ANOVA for Elapsed time, task 2

Source of variation	Degrees of freedom	Sum of squares	Mean square	F
Elapsed time, task 3	2	29937.66	14968.83	6.23
Error	24	57673.24	2403.05	
Total	26	87610.90		

Table 45: ANOVA for Elapsed time, task 3

Source of variation	Degrees of freedom	Sum of squares	Mean square	F
Rotation events, task 1	2	40692503	20346252	12.83
Error	24	38061472	1585895	
Total	26	78753976		

Table 46: ANOVA for Rotation events, task 1

Source of variation	Degrees of freedom	Sum of squares	Mean square	F
Mouse distance, task 1	2	14570546411	7285273206	1.09
Error	24	160932000243	6705500010	
Total	26	175502546654		

Table 47: ANOVA for Mouse distance, task 1

Source of variation	Degrees of freedom	Sum of squares	Mean square	F
Move events, task 2	2	20178246	10089123	2.48
Error	24	97458322	4060763	
Total	26	117636568		

Table 48: ANOVA for Move events, task 2

Source of variation	Degrees of freedom	Sum of squares	Mean square	F
Move events, task 3	2	23819863	11909932	5.78
Error	24	49419819	2059159	
Total	26	73239682		

Table 49: ANOVA for Move events, task 3

Source of variation	Degrees of freedom	Sum of squares	Mean square	F
Button click events, task 1	2	150198.00	75099.00	19.74
Error	24	91306.00	3804.42	
Total	26	241504.00		

Table 50: ANOVA for Button click events, task 1

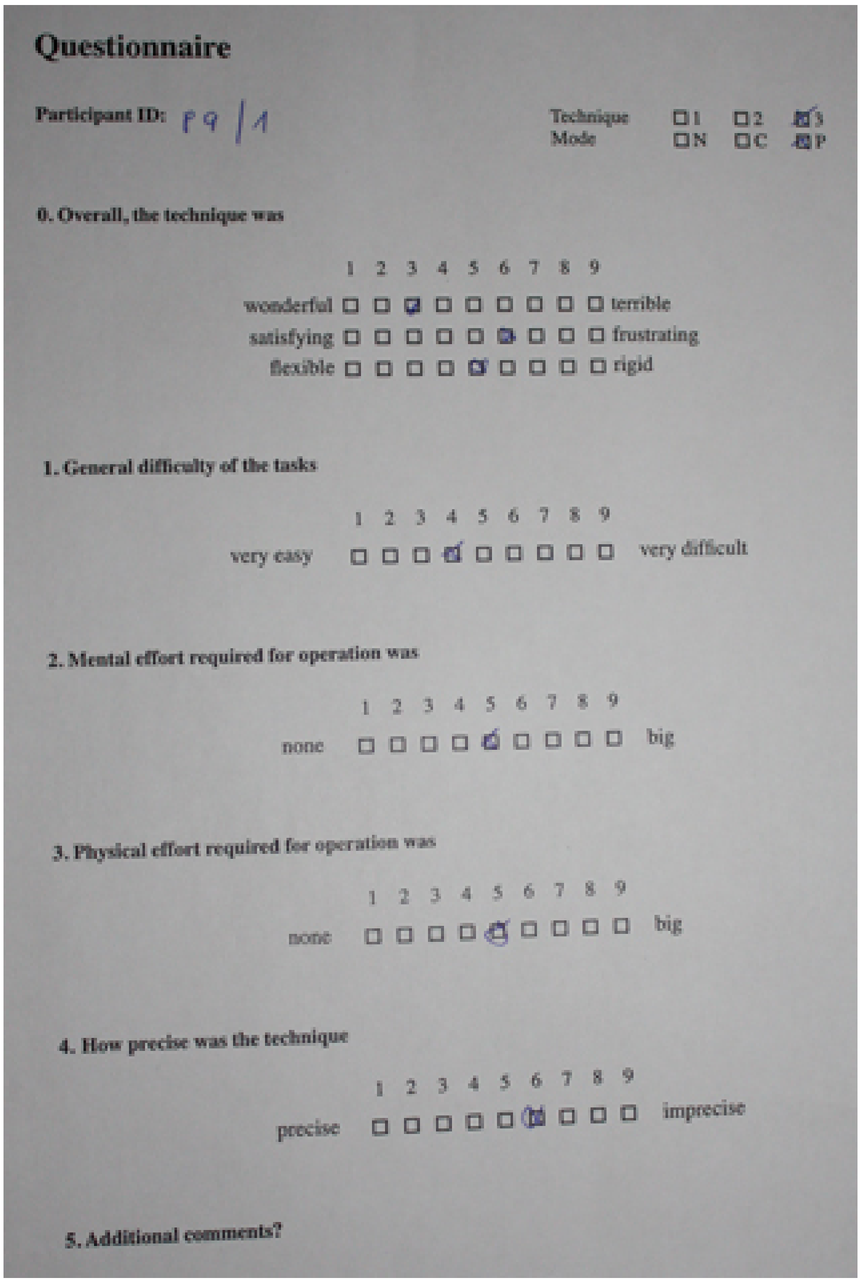
Source of variation	Degrees of freedom	Sum of squares	Mean square	F
Button click events, task 3	2	186082.67	93041.33	34.98
Error	24	63834.00	2659.75	
Total	26	249916.67		

Table 51: ANOVA for Button click events, task 3

C T-tests

$z_{0.05} = 1.645$
$z_{0.0025} = 1.960$
Reject null-hypothesis if $Z < -z_\alpha$ or $Z > z_\alpha$
T-test for satisfying-frustrating, technique (0 & 1): Z = -3.46
T-test for flexible-rigid, technique (0 & 1): Z = -2.77
T-test for phys effort, technique (0 & 1): Z = -1.29
task #1
T-test for Mouse distance, technique (0 & 2): Z = -363.87
task #2
T-test for Mouse distance, technique (0 & 1): Z = -208.21
task #1
T-test for Move events, technique (1 & 2): Z = -144.16
task #2
T-test for Zoom events, technique (0 & 2): Z = -5.65
task #3
T-test for Zoom events, technique (0 & 2): Z = -8.90
task #2
T-test for Button click events, technique (0 & 1): Z = -61.53

D Example Questionnaire



Questionnaire

Participant ID: P9 / 1

Technique ☐1 ☐2 ☑3
Mode ☐N ☐C ☑P

0. Overall, the technique was

	1	2	3	4	5	6	7	8	9	
wonderful	☐	☐	☑	☐	☐	☐	☐	☐	☐	terrible
satisfying	☐	☐	☐	☐	☐	☑	☐	☐	☐	frustrating
flexible	☐	☐	☐	☐	☑	☐	☐	☐	☐	rigid

1. General difficulty of the tasks

	1	2	3	4	5	6	7	8	9	
very easy	☐	☐	☐	☑	☐	☐	☐	☐	☐	very difficult

2. Mental effort required for operation was

	1	2	3	4	5	6	7	8	9	
none	☐	☐	☐	☐	☑	☐	☐	☐	☐	big

3. Physical effort required for operation was

	1	2	3	4	5	6	7	8	9	
none	☐	☐	☐	☐	☑	☐	☐	☐	☐	big

4. How precise was the technique

	1	2	3	4	5	6	7	8	9	
precise	☐	☐	☐	☐	☐	☑	☐	☐	☐	imprecise

5. Additional comments?

Figure D.1: Example questionnaire

E Participant Timetable

Participant	Time	Day
P1	21.00 to 22.00	1
P2	14.00 to 15.00	2
P3	15.00 to 16.00	2
P4	16.00 to 17.00	2
P5	19.00 to 20.00	2
P6	20.00 to 21.00	2
P7	12.00 to 13.00	3
P8	13.00 to 14.00	3
P9	14.00 to 15.00	3

Table 52: Evaluation Schedule