University of BRISTOL

SCHOOL OF COMPUTER SCIENCE

Smartwatch Holder Modeling Software Created with openFrameworks

Watch Crafter

Yangfan Duan

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Bachelor of Science in the Faculty of Engineering.

Thursday 9th May, 2024

Abstract

This study aims to develop an easy-to-use 3D modeling software for parents and researchers, capable of exporting STL files specifically designed to accommodate the smartwatch holders for children with ADHD. Developed using openFrameworks, the software includes features such as magnetic attachments and symmetry operations, simplifying the model creation process. Additionally, it allows for the addition of various small decorations like cat ears and wings. It also includes a loop that enables children to connect the holder to a lanyard and hang it around their neck or on their clothing or bags.

The software was tested using the System Usability Scale (SUS) questionnaire with a demographic of users aged 20 to 25, and acceptability testing was conducted by two ADHD children using the 3D printed holders. The results showed an average SUS score of 70, indicating a high level of user satisfaction. However, feedback pointed out issues such as the software being overly simplistic and lacking in challenge, and limited model variety. Child users responded positively to the product, expressing great fondness for the holders, but noted that the material of the holder was slippery and prone to sweat. Future research will focus on improving these issues.

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Taught Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, this work is my own work. Work done in collaboration with, or with the assistance of others including AI methods, is indicated as such. I have identified all material in this dissertation which is not my own work through appropriate referencing and acknowledgement. Where I have quoted or otherwise incorporated material which is the work of others, I have included the source in the references. Any views expressed in the dissertation, other than referenced material, are those of the author.

Yangfan Duan, Thursday 9$^{\text{th}}$ May, 2024

Contents

List of Figures

Ethics Statement

- This project fits within the scope of ethics application 0026, as reviewed by my supervisor, Jon Bird.

Chapter 1

Introduction

Advancements in smartwatch design and other wearable technologies have found widespread application among various groups, including children diagnosed with Attention Deficit Hyperactivity Disorder (ADHD).[7] While these devices offer potential benefits in managing the condition through reminders and tracking functions, studies suggest that standard wearable device designs may not suit the sensitivities and preferences of children with ADHD. Many children find the wristbands uncomfortable and the overall design unappealing to their aesthetic, which may reduce the frequency of use and, consequently, diminish the intended benefits.

In this study, a modelling software that can be used by a parent or guardian with a child was developed through the use of openFramework, aiming to address the discomfort and reluctance in using smartwatches. The software is capable of exporting a customisable model of the brace as an STL file, and then producing an actual usable model through 3D printing. Unlike traditional wristbands, this design specifically incorporates a loop that allows the user to hang the watch anywhere via a lanyard. This innovation aims to alleviate the discomfort associated with wearing a typical wristband, while providing a more versatile wearing method for children with ADHD. Additionally, the software tool developed allows for a variety of decorative elements to be added to the holder, such as ears, wings, and other decorations, which can be easily added and adjusted by the user. By providing customisable features, the tool enables children and their caregivers to transform technological products into personalised and adorable objects.

The main contributions of the project include:

- Development of a flexible and feature-rich modeling software using openFrameworks.
- Development of magnetic and symmetrical features.
- Adaptive design of the smartwatch holder and its iterative testing process.
- Conducting preliminary user testing to collect direct feedback from the target audience, providing information for iterative design improvements.

Chapter 2

Background

This section will look at the challenges faced by children with Attention Deficit Hyperactivity Disorder (ADHD) and how smartwatches can help them as well as cause them distress. The motivation for the development of this software will be discussed, including how technology can help this particular group, and a description of the technical background involved and the gaps that exist in the market. This will provide comprehensive background information to understand the potential value and practical applications of the software.

2.1 User Background

2.1.1 Challenges faced by ADHD children

Impact on Daily Functioning

Children with Attention Deficit Hyperactivity Disorder (ADHD) face a range of challenges in their daily lives, stemming from core symptoms: inattention, hyperactivity, and impulsivity.[7] These symptoms can severely impact their learning abilities, social interactions, and family life. For example, inattention may cause them to fall behind in schoolwork, while hyperactivity can lead to excessive activity or restlessness in situations that require calmness. Impulsive behavior might result in poorly considered decisions, leading to conflicts with peers and affecting their personal development as well as their relationships with family and society, ultimately reducing their quality of life.

Self-regulation,[4] an individual‘s ability to manage emotions, attention, behaviour and cognition, is critical at every stage of a child‘s development, and children with ADHD face challenges in regulating emotional responses, maintaining attention, inhibiting impulsive behaviours, and planning executive tasks. Physiologically, the areas of their brains responsible for executive functions, such as the prefrontal cortex,[8] are structurally and functionally different, which can lead to difficulties in handling complex tasks and social interactions.

And assistive technology can help children with ADHD apply these skills more effectively, such as behaviour management apps and learning tools, including customised reminder systems, task managers and interactive games that help children understand and manage behaviour and emotions.

Parents and teachers play a key role in supporting the development of children‘s self-regulation skills.[6] They help children develop self-regulation skills over time by establishing consistent daily structures, clarifying expectations and consequences, and providing positive feedback and appropriate supervision to help them succeed at specific tasks and learn to adapt to a variety of social and academic settings.

The help of smartwatches

As an advanced wearable technology, smartwatches provide a unique tool for daily management of children with ADHD, particularly excelling in time management and behavior monitoring. For instance, the CoolTaco app [5] allows children to set specific reminders on their smartwatches for academic, social, and health activities. These reminders help them better adhere to their daily schedules and reduce anxiety caused by forgetting or missing important events. Further, the app motivates children by rewarding them with points upon task completion, gradually cultivating their ability to pursue long-term goals. It also allows parents to adjust or update tasks in real-time via their smartphones, ensuring that the activity schedule always meets the children‘s latest needs.

On the other hand, the CoolCraig app[3] tracks and records children's emotional states in an interactive manner. The app regularly asks through the smartwatch, "How are you feeling now" and allows children to express their emotions by selecting corresponding colors and emoticons. This method not only enhances children's interaction with technology but also enables parents to understand their children‘s immediate emotional states through their selections. Moreover, this real-time emotional logging helps parents monitor the patterns of their children's emotional fluctuations, thereby providing more targeted support and intervention and strengthening parental understanding and support of their children's daily behavior and emotional management.

Through these features of smartwatches, children with ADHD can receive more personalized and timely support, which may improve their daily functioning and overall quality of life. However, to realize these benefits, it is necessary to consider that the device‘s design must meet the physical and aesthetic needs of children, ensuring that they can wear the smartwatch comfortably and are willing to use it.

2.1.2 Problems with existing smartwatches

Discomfort and reluctant use

Although smartwatches are designed to help users manage daily tasks and monitor health conditions, standard smartwatch designs often cause discomfort and resistance in children with ADHD, particularly those models equipped with silicone sports bands. While silicone is durable and easy to clean, it can be too rigid or tight for some children, especially after prolonged wear. Additionally, this material may cause skin irritation or allergies during physical activity or in hot weather, increasing the child‘s discomfort.

Specific user feedback

User feedback reveals specific issues encountered by children with ADHD when using smartwatches. For example, in a study [3] that involved collecting health data from children with ADHD using smartwatches and having them interpret this data, one child participating in the field study commented, "I kept [the smartwatch] loose a lot when I sleep. Because it still bothers me, and I don‘t sleep if something just bothers me." This indicates that the presence of the watch during sleep can significantly impact the child's sleep quality, potentially affecting their attention and energy levels the following day.

Similarly, during physical activities, the accumulation of sweat under the watch may cause skin irritation, or the need for frequent readjustments due to active movements can be particularly troubling for children with ADHD. Another child in the same study described their reduced interaction with the device: "Well, lately I haven‘t been putting [the watch] on much, but the first few days I‘ve been keeping it on, and also for the night. Then it‘s kind of got really sweaty on my hands since, also because of this heat. And I also go to the gym, so it also makes my wrist sweat more and it‘s really uncomfortable because they always move it like every 30 seconds or so." This description not only reflects physical discomfort but also highlights a decrease in usage frequency due to the discomfort.

Changes in frequency of use

Based on the comparison of smartwatch usage between weekdays and weekends, it is evident that children tend to choose not to wear the watch on weekends. This change reflects that in the absence of external pressures (such as school mandates), children prefer comfort over the functionality of the watch: "Weekends are my days off, and I don‘t really... its just more comfortable to not wear [the watch]."[3] This indicates that smartwatch designs need to consider children‘s comfort and personal preferences more to enhance the acceptability and sustainability of daily use.

These feedback and observations clearly point out that smartwatches designed for children with ADHD need to consider their unique needs, including the use of materials that are more comfortable and suitable for sensitive skin, as well as providing more flexible wearing options to ensure these devices are used more widely and continuously.

2.1.3 Innovative design for smartwatch holders

The need for customisability and comfort

Traditional silicone bands of smartwatches often cause discomfort for children with ADHD, which is not only a matter of material but also a lack of flexibility in design. To address the discomfort experienced by

ADHD children when using smartwatches, a decision was made to develop a comprehensive and specifically comfortable and aesthetically pleasing watch holder for ADHD children. This holder accommodates the smartwatch well, and users can hang it, for example, around the neck or on a backpack or clothing, thus avoiding direct contact with the skin. This can effectively reduce the discomfort ADHD children experience when wearing smartwatches.

Software design

To meet the aforementioned needs, this project has developed a modeling software specifically for personalized smartwatch holders, which supports exporting models as STL files for realization through 3D printing technology. The initial software design was targeted for use by neurotypical adults. This means that the user interface and functionality of the software need to cater to the daily usage habits and cognitive abilities of general adults, ensuring it is intuitive and user-friendly. Although the final product will aid ADHD children through the assistance of parents or caregivers to manage daily tasks and activities, the preliminary design focus of the software is to ensure that adults without special needs can also easily use this tool, thereby benefiting the entire family with the convenience and functionality of the technology.

The design of the holder includes a unique loop feature, allowing the watch to be worn in various ways. For instance, it can be simply hung around the neck with a lanyard, or attached to backpacks and clothing, providing children with flexible wearing options while reducing the discomfort caused by direct contact with the skin.

Furthermore, the holder can be personalized through the software, where users can easily add or change decorations according to personal preferences. We have also specially designed the holder to ensure that all added decorations and the smartwatch remain securely attached to the holder during vigorous activities.

The introduction of these innovative designs allows the project to meet the diverse needs of ADHD children in terms of comfort, aesthetics, and functionality, thereby significantly improving their daily life experience and frequency of device use.

2.2 Technology Background

User-Centred Design

Human-computer interaction (HCI)[9] principles play a crucial role in the development of technology products for children with ADHD. This means that an in-depth understanding of the specific challenges that children with ADHD face when using technology products, such as attention maintenance, sensory sensitivity, etc., is needed.

2.2.1 Applications of openFrameworks

OpenFrameworks,[1] an open-source C++ toolkit designed for the creative coding community, plays a central role in this project, particularly in supporting rapid prototyping. It provides a powerful and flexible programming environment that allows for quick construction and testing of new design concepts.

Rapid Prototyping

One of the core strengths of openFrameworks is its ability to rapidly iterate on prototypes. openFrameworks offers a wide range of libraries and pre-built modules, including processing capabilities for graphics, which allow developers to quickly assemble complex application functionality. For example, user interface prototypes can be created quickly by using openFrameworks‘ graphics libraries.

Project Support and Community Resources

As an active open source community, openFrameworks provides a wealth of documentation, tutorials, and forum discussions that are invaluable for solving programming challenges or implementing new feature ideas. Not only can you use these resources to solve technical problems, but you can also gain creative inspiration and development advice through the power of the community.

2.2.2 Integration of 3D Printing Technology

3D printing technology also played a crucial role in this project. By combining openFrameworks' STL file export with 3D printing technology, it was possible to convert the virtual file into a physical model, allowing for better user experience testing and product optimisation. The STL file format, known as Stereolithography, is one of the most commonly used file types for 3D printing. It holds the 3D geometry of a model, usually a series of triangles that describe the surface of an object. Each triangle consists of three vertices and a normal vector pointing outwards, which are essential for determining the orientation of the model in space.STL files exist in two formats: ASCII and binary, with binary format files being smaller and faster to read and write.

Implementing personalised designs

A core benefit of 3D printing technology is the ability to produce highly personalised products. In the context of this project, this means that a fully customised smartwatch holder can be designed and manufactured according to the specific needs and preferences of an ADHD child.For example, the size and shape of the holder can be adjusted, as well as the inclusion of specific decorative elements, such as cat ears or small butterfly wings, which are configured according to the user's choices directly in the design software, and then exported as an STL file for printing.

Rapid prototyping iterations

Using the Ultimaker 2+ Connect 3D printer in the Hackspace at the University of Bristol's MVB building, it was possible to quickly transform these designs into physical objects within reach. Printing a customised holder takes around 4 hours, which allows multiple versions to be iterated on in a single day, responding quickly to test feedback and making adjustments as necessary.

Hands-on challenges

Due to the popularity of Hackspace's 3D printers, printing events had to be highly planned. I needed to be at Hackspace by 9am every morning to ensure access to the printers. This high level of demand usually means that if you miss your morning opportunity, you may not be able to print for the entire day. This situation requires a high level of organisation and prioritisation of print jobs to ensure that each print maximises resource efficiency and that design iterations are not delayed by equipment availability.

Through this efficient use of 3D printing technology, the project was able to not only deliver a personalised product that meets the needs of children with ADHD, but also enable rapid iteration and continuous improvement during the development process. This ability for rapid prototyping is an integral part of modern product design, especially in innovative projects for special needs groups.

2.2.3 Market demand

While wearable technology has rapidly gained global popularity for a wide range of applications such as health monitoring, fitness tracking, and daily activity management, the market has fallen short in addressing the specific needs of children with ADHD, who use such devices in ways that conventional smartwatch products often fail to adequately support due to their unique sensory and behavioural needs, particularly in terms of comfort and design of the user interface. These children may be particularly sensitive to physical touch and need devices that are not only functional, but also made of materials and designed to avoid causing discomfort. As for the modelling software, there are very many modelling software on the market, the advanced ones are similar to maya but have a high threshold, and the low threshold ones are such as tinkercad. but there is no modelling software specifically for smartwatch holders. This project identifies this gap and develops it.

2.2.4 Summary

By exploring the principles of Human-Computer Interaction (HCI) and the application of 3D printing technology, this project aims to address the neglect of the specific needs of ADHD children in the existing wearables market. Through a detailed study of the daily challenges and technology usage habits of this particular group, this project identifies a significant market gap and proposes an innovative solution. The solution includes a full-featured software that can export smartwatchholder model files, and the holder design takes into account the sensory and emotional needs of ADHD children, greatly enhancing the comfort and interactivity of the product through a variety of decorations and wearing style options. The core of the project is to use the open source tool openFramework to develop and produce a modelling software that can be easily used by both children and adults. Through these innovative designs, this project not only fills a gap in the market by providing ADHD children with a product that meets their functional needs as well as their emotional and sensory needs, but also has the potential to change the acceptance and frequency of use of wearable technology for this group, thereby truly enhancing their quality of life.

Chapter 3

Project Execution

In the implementation phase of this project, the development of the software, originally designed for the parents of children with ADHD and researchers, will be detailed. The software is intended to provide an intuitive, user-friendly interface that allows users to efficiently and quickly create a model of a smartwatch holder. Special attention was given to the simplicity of the user interface and the fluidity of interaction to ensure that even users without a technical background can easily get started. The project uses GitHub for development recording and version control. The first milestone was implementing the import and export of STL files, ensuring the software could handle key STL files, which is essential for the core functionality of the software. Following this, a basic user interface was designed to be straightforward and intuitive, facilitating use by parents and researchers. Additionally, models of smartwatch holders and various decorations were created using tools like Tinkercad and Maya. Furthermore, features such as magnetic attachment and symmetric operations were developed to enhance the software‘s practicality and interactivity.

3.0.1 Overview of the software system

Within this software system, several key classes were designed to support the processing of 3D models and interface interactions. The core classes include:

- Model: Responsible for loading and manipulating basic 3D models.
- AccessoryModel: Manages accessories related to the models.
- EarModel: Specifically handles symmetrical models, such as ears and other attachments, ensuring symmetrical adjustments are automatically applied during modifications.
- KeyHandler: Manages keyboard inputs to streamline user interface interactions.
- ofApp: Acts as the centerpiece of the entire application, coordinating the instantiation and overall logic flow of various classes.

This structural design facilitates future expansion and maintenance, allowing for straightforward updates and enhancements to the software's capabilities.

3.1 Beginning of the project: challenges and transformations

3.1.1 Importing and Exporting STL Files

Challenges of using Processing

At the beginning of the project, Processing[2] was chosen as the development tool due to its good graphical and interactive programming support, ideal for rapid prototyping and visual art projects. In order to implement the export function of STL files, various methods and tools were tried, including using ModelBuilderMk2 library, netfabb, HEMesh, and even writing custom code to generate STL files.

Unfortunately, none of these attempts were successful. The problems encountered varied from using outdated and incompatible libraries to exporting STL files with serious flaws. For example, ModelBuilderMk2, while reliable when working with simple models, could not effectively support the export of complex or highly customised 3D models. Self-written export codes, while providing greater flexibility, encountered technical obstacles in terms of accuracy and file integrity. These technical challenges were rooted in the fact that Processing was built primarily for visual art and simpler interaction design, and was not fully suited to complex 3D modelling needs. At the suggestion of my supervisor Jon, it was decided to turn to openFrameworks in order to find a more stable and full-featured solution. openFrameworks provides a much more robust C++ environment suitable for dealing with complex user interface and 3D model processing needs, and in particular its support for modern 3D printing file formats such as STL is significantly better than that of Processing.

Initial attempts to use the ofxSTL library

Switching to openFrameworks, I started experimenting with the ofxSTL library to import and export STL files. Although this library performed well with simple models, it encountered issues when dealing with more complex models, a challenge I had previously experienced with Processing. To address this, I turned to the ofxAssimpModelLoader library, which supports multiple model formats and successfully displays most complex models, although some specific models still had display issues. 3.1

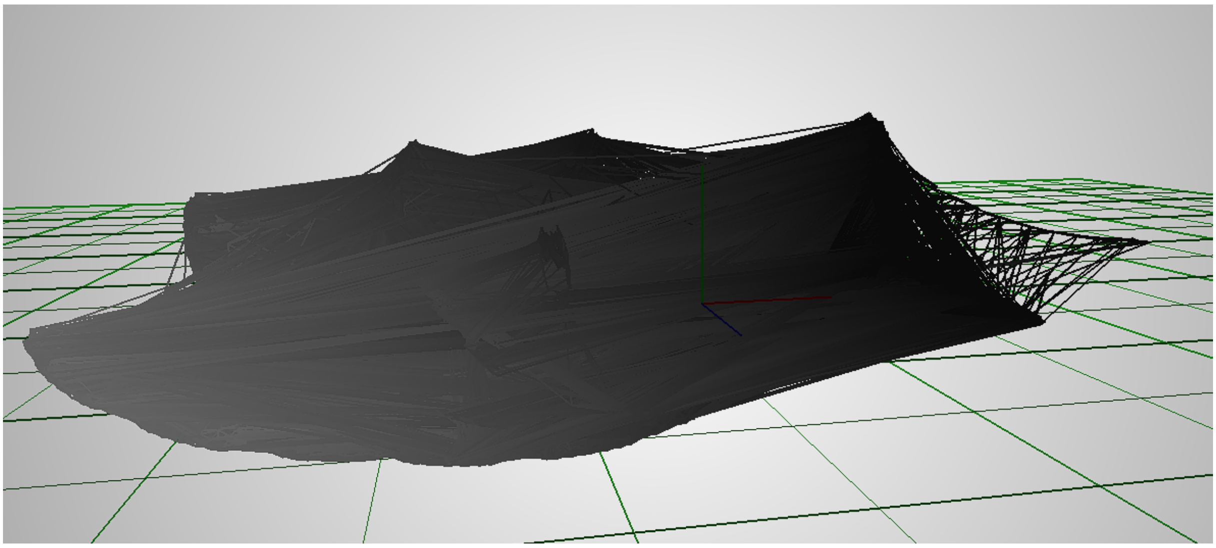

Figure 3.1: Error-displayed wing model

Development of custom export functionality

While the 'ofxAssimpModelLoader' excelled in loading 3D models, it lacked support for exporting STL files. Thus, I decided to develop a module to handle the export of STL files.

Initially, I created a Model class that encapsulated the data processing of 3D models. I then implemented a 'loadModel' method using 'ofxAssimpModelLoader' to import model data.

Following this, I developed the 'addMesh' method3.1. The main responsibility of this method is to handle the mesh data imported from 'ofxAssimpModelLoader'. Since the model data imported by ofxAssimpModelLoader is immutable, 'addMesh' allows for passing the data to a mutable Mesh variable.

The most critical part was the development of the 'exportModelToSTL' method3.3. This method is responsible for converting the processed mesh data into the STL file format. During this process, I wrote an algorithm to traverse all the vertices and indices, generating the triangular data required by the STL format and calculating the corresponding normal vectors. Finally, I used the 'saveModel' method from the 'ofxSTL' library to export the STL file.

This approach allowed for the integration of advanced model loading capabilities with the necessary export functionality, providing a comprehensive solution for managing 3D models within the project's software framework.

Although the initial implementation of the export function was achieved, testing revealed that the exported STL files were still flawed, often being empty. After a series of troubleshooting, I discovered issues with the 'addMesh' method and the 'loadModel' method from 'ofxAssimpModelLoader'. The ‘exportModelToSTL‘ method was designed to ensure that each processed mesh could correctly form triangles, filtering out indices that are not multiples of three. Since a triangle is composed of three vertices, the index count for each mesh must be a multiple of three. If the index count is not a multiple of three, the mesh may not correctly represent a complete triangle, which could cause issues during rendering and exporting. Therefore, I modified the ‘addMesh‘ method to include a check for the index count if ‘(tempMesh.getNumIndices()% 3 == 0)‘ skipping any that do not meet this criterion.

During testing, I found that there might be issues with the ‘loadModel‘ method3.2 of ‘ofxAssimpModelLoader‘. Since ‘addMesh‘ is called immediately after model data is imported3.4, an empty print result indicates that there might be errors within ‘loadModel‘ or it may not perfectly parse the imported STL file. However, since this is a library provided by openFrameworks, it was difficult to modify, and I did not pursue further investigation since the export function was correctly implemented after adding the index value filter.

The export function underwent multiple tests, including 3D printing tests of the generated STL files, to ensure that the files met the standards required for 3D printing. This rigorous testing helped to confirm the functionality and reliability of the export feature, ensuring that users could successfully use the software for practical 3D printing applications.

```
void EarModel::adMeshes() {
    mesh.clear();
    for (int i = 0; i < modelLoader.getNumMeshes(); i++) {
        ofMesh tempMesh = modelLoader.getMesh(i);
        if (tempMesh.getNumIndices() % 3 == 0) {
            for (int j = 0; j < tempMesh.getNumVertices(); j++) {
                mesh.addVertex(tempMesh.getVertex(j));
                if (tempMesh.hasNormals()) {
                    mesh.addNormal(tempMesh.getNormal(j));
                }
            }
            for (int k = 0; k < tempMesh.getNumIndices(); k++) {
                mesh.addIndex(tempMesh.getIndex(k));
            }
        }
        else {
            ofLogNotice() << ‘‘Mesh ‘‘ << i << ‘‘ is not a multiple of three, skipped‘‘;
        }
    }
}
```

Listing 3.1: Processing imported mesh data

```
bool loadModel(const std::string& filepath) {
    modelLoader.setScaleNormalization(false);
    bool loaded = modelLoader.loadModel(filepath);
    return loaded;
}
```

Listing 3.2: Calling the import function of ofxAssimpModelLoader

```
void Model::exportModelToSTL(const std::string& filePath) {
    combineMeshes();
    ofxSTLExporter stlExporter;
    stlExporter.useASCIIFormat(false);
    stlExporter.beginModel(‘‘CombinedModel‘‘);

    auto& vertices = combinedMesh.getVertices();
    auto& indices = combinedMesh.getIndices();

    if (indices.size() % 3 != 0) {
        return;
    }

    for (size_t i = 0; i < indices.size(); i += 3) {
        if (indices[i] >= vertices.size() || indices[i + 1] >= vertices.size() || indices[i + 2] >= vertices.size()) {
            continue;
        }
        ofPoint vert1 = vertices[indices[i]];
        ofPoint vert2 = vertices[indices[i + 1]];
        ofPoint vert3 = vertices[indices[i + 2]];
        ofPoint normal = ((vert2 - vert1).cross(vert3 - vert1)).normalized();
```

```
        stlExporter.addTriangle(vert1, vert2, vert3, normal);
    }
    stlExporter.saveModel(filePath);
    ofLogNotice() << ‘‘Model exported to STL file: ‘‘ << filePath;
    combinedMesh.clear();

}
```

Listing 3.3: Export mesh data to STL file

```
void ofApp::loadLoopButtonPressed() {
    std::shared_ptr<EarModel> newAssimpModel = std::make_shared<EarModel>(‘‘loop‘‘);
    std::string modelPath = ‘‘loop.stl‘‘;
    if (newAssimpModel->loadModel(modelPath)) {
        newAssimpModel->adMeshes();
        models.push_back(newAssimpModel);
        selectedModel = newAssimpModel;
        mainModel->addPart(newAssimpModel);

        ofVec3f newPosition = ofVec3f(26.0f, 0.0f, 0.0f);
        newAssimpModel->setPosition(newPosition);
        newAssimpModel->setPresetPositions({ ofVec3f(-31.5f, -2.0f, 0.0f), ofVec3f(30.0f, -2.0f, 0.0f), ofVec3f(0.0f, 24.0f, -3.0f), ofVec3f(0.0f, -25.5f, 0.0f) });

    }
    else {
        ofLogError(‘‘AssimpModelButtonPressed‘‘) << ‘‘Failed to load model: ‘‘ << modelPath;
    }
}
```

Listing 3.4: loadLoopButonPressed

3.2 User Interface and Interaction Design

Design Concept

The user interface design of this project aims to provide an intuitive and user-friendly experience, enabling parents of children and researchers to easily and simply create the desired model. The core design philosophy is to create a clear, distraction-free 3D space that allows users to focus on the creation and customization of the model. To enhance usability and accessibility, the interface employs a design that includes simple visual elements and easy-to-understand instructional prompts.

Interface Layout

The user interface includes a centrally located 3D space equipped with a 2D grid to help users accurately place and align models, making the process of creating and modifying models intuitive and straightforward.

In the upper left corner of the 3D space, there is a scalable and movable menu bar, which users can expand or collapse as needed. The menu bar is divided into three main sections: basic holder, decorations, and control options. Dynamic operational tips on the right and bottom of the interface guide users on how to move the viewpoint and the model.

In the project, the ofxGUI library is used to create a scalable basic menu bar, and ofxButton is used to implement operable buttons. By adding listeners, we have implemented the functionality of generating models in the center of the 3D space upon button clicks. For example, clicking the “loop” button generates a loop model at the center. Additionally, users can adjust the orientation of the model selected, other than the basic holder, via sliders (such as rotating along the y-axis). The slider functionality is implemented using ofxGUI‘s ofParameter, which allows setting the range of rotation angles (e.g., -360 to 360 degrees).

However, there is an issue with ‘ofParameter‘: the movement step size is fixed at 1 and cannot be changed. Given that the slider is quite sensitive, using the mouse to achieve the desired angle can be challenging. Therefore, the method of rotation has been modified.3.5 In the process of adjusting the rotation step size, a new step variable ‘step‘ is first defined, set to 2 in this example. Next, the new rotation angle ‘newRotation‘ is calculated. This calculation uses rounding to ensure the rotation value

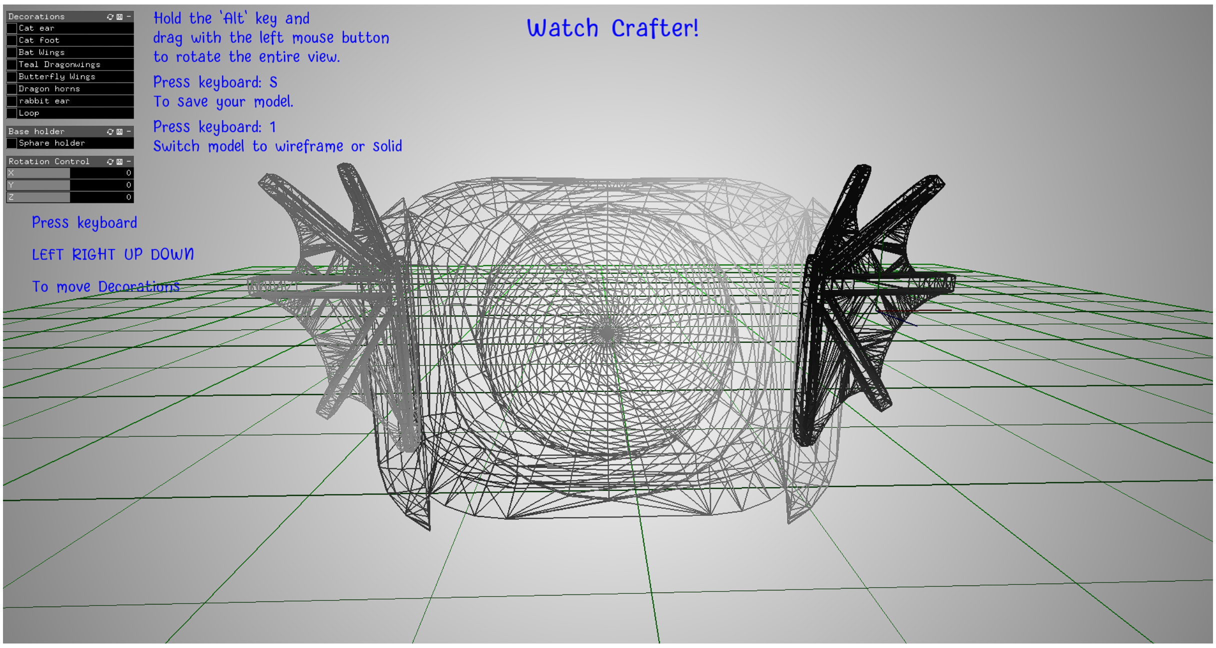

Figure 3.2: interface

is a multiple of the step size. Thus, any given rotation value will be adjusted to the nearest value that conforms to the step setting. If the calculated ‘newRotation‘ is different from the current rotation value ‘rotation‘, the program first temporarily removes the listener to avoid triggering additional events when setting the new rotation value. Afterward, the value of ‘rotateY‘ is updated to ‘newRotation‘, and then the listener is re-added to ensure future changes are properly handled. Additionally, if there is a selected model, the ‘setRotation‘ method is called to update the model‘s rotation on the Y-axis to the new rotation angle. This way, each move of the slider will now increment in steps of 2.

```

void ofApp::modelYrotationChanged(int& rotation) {
    int step = 2;
    int newRotation = ((rotation + step / 2) / step) * step;
    if (newRotation != rotation) {
        rotateY.removeListener(this, &ofApp::modelYrotationChanged);
        rotateY.set(newRotation);
        rotateY.addListener(this, &ofApp::modelYrotationChanged);
    }
    if (selectedModel) {
        selectedModel->setRotation(newRotation, ofVec3f(0, 1, 0));
    }
}
```

Listing 3.5: Specific implementation with a step of 2

3.2.1 Interaction Design

To ensure that the interface is both intuitive and efficient, the project uses keyboard and mouse controls designed for precision and ease of use. Key interactions include:

Model selection and movement

Users can select and move any decorative model using the left mouse button, providing an intuitive way to reposition the model. When a model is selected, three axes (X, Y, Z) automatically appear at the model‘s center, and the originally gray tooltip turns blue, providing immediate visual feedback that helps users understand the model‘s positioning in three-dimensional space.

Users simply need to hold down the Alt key to activate the camera view. In this view, users can rotate the scene by holding and dragging the left mouse button, or pan the entire scene by holding and moving the middle mouse button.

This camera view is implemented using openFrameworks' 'ofEasyCam'. However, because moving the model also uses the left mouse button, which conflicts with moving the view, the function to activate the camera view only when holding down the Alt key was created. This arrangement prevents any conflict between the two actions.

Magnetic function

The project has implemented a "magnetic" feature, which is central to enhancing user experience by simplifying the model placement process. When users move a model using the mouse and release the left button at any location, the system automatically aligns the model to the preset optimal position. This design significantly reduces the precision required for placement, enabling users to easily and accurately position decorations at the desired locations. For example, if a user wants to place a pair of cat ears on top of a basic holder, they simply drag the cat ears upwards and release the mouse, and the system will automatically position the cat ears at the optimal location above the holder.3.6 This smart alignment feature is implemented by pre-setting four key positions for the model (top, bottom, left, right). After the user's operation is complete, the system calculates the distance between the model's current position and these four preset locations, automatically moving the model to the nearest position to optimize placement accuracy and user experience.

Additionally, the keyboard arrow keys can be used for fine adjustments of the selected model, a method particularly suitable for making small and precise positional changes. The step size for the arrow keys is set at 0.5, with options for up, down, left, and right directions. After roughly positioning the model using the mouse, users can use these four arrow keys for more detailed adjustments. This feature allows for greater control over the positioning of models.

```
if (!earModel->presetPositions.empty()) {
    ofVec3f currentPos = earModel->getPosition();
    float minDistance = std::numeric_limits<float>::max();
    int closestIndex = -1;
    for (int i = 0; i < earModel->presetPositions.size(); i++) {
        float distance = currentPos.distance(earModel->presetPositions[i]);
        if (distance < minDistance) {
            minDistance = distance;
            closestIndex = i;
        }
    }
    if (closestIndex != -1 && altPressed == false) {

        earModel->updatePosition(closestIndex, altPressed);
```

Listing 3.6: Part of the mouseReleased method that detects the nearest position.

Symmetry operation

Considering the visual importance of symmetry, this project has implemented a symmetry adjustment feature for paired decorative models, such as a pair of cat ears. When a user selects and moves one side (e.g., the left ear), the system automatically adjusts the position of the other side (right ear) to ensure both maintain visual symmetry. This feature not only simplifies the operation but also ensures the aesthetic quality of the design. To achieve symmetry, the system uses different calculation methods based on the relative positions of the models. For example, if a pair of ears is positioned above the holder, moving the left ear in the x-direction will cause the right ear to move in the opposite direction by the same x value, i.e., -x; movements in the y-direction will be applied in the same direction.3.7 This method successfully implements symmetrical adjustment of the models.

```
    if (selectedModel->getDirection() == ``left``) {
        maxPosX = -23;
        ear = -23;
        pos2.x = calculateNewPosition(pos2.x, moveStep, ear);
    }
    else if (selectedModel->getDirection() == ``right``) {
        maxPosX = 27;
        ear = 27;
        pos2.x = calculateNewPosition(pos2.x, moveStep, ear);
    }
    else if (selectedModel->getDirection() == ``up``) {
        maxPosX = -12;
        ear = 10.5;
```

```
        pos2.x = calculateNewPositionplus(pos2.x, moveStep, ear);
    }
    else if (selectedModel->getDirection() == ``down``) {
        maxPosX = 21;
        ear = -21;
        pos2.x = calculateNewPositionplus(pos2.x, moveStep, ear);
    }
}
pos.x = calculateNewPosition(pos.x, moveStep, maxPosX);
break;
```

Listing 3.7: Symmetry calculation in KeyHandler

To ensure that the models remain properly connected and suitable for printing after moving, necessary restrictions are placed on user manipulation. Taking the cat ears as an example, the system sets boundaries to prevent users from moving the ears outside the base model or too close to the interior, which could interfere with the functionality of the watch storage compartment. These restrictions ensure that the decorations not only visually comply with the design intent but also do not affect the functionality of the watch holder. This is achieved by using the 'KeyHandler' class to precisely control key input, ensuring that decorative models seamlessly integrate with the base holder without disconnections. Specifically, the initial position and category of the model are read using the 'pos' position variable and 'modelType' type variable within the 'handleKeyPress' method. Depending on the category of the model, specific movement restriction rules are applied to calculate the allowable amount of movement.

Model Deletion

Users can quickly delete selected models by pressing the Backspace key, providing an intuitive and efficient way to manage elements within the scene and avoiding cumbersome menu operations. This functionality is implemented based on mainModel, a primary container that manages all model components. Whenever a user adds a new model, the system invokes the mainModel-¿addPart() method to integrate it into the scene. Conversely, to delete a model, the mainModel-¿removePart() method3.8 is called, which directly removes the currently selected part from the main model group.

```
 case OF_KEY_BACKSPACE:
     if (selectedModel) {
         auto it = std::find_if(models.begin(), models.end(), [&](const auto& model) {
             return model == selectedModel;
             });
         if (it != models.end()) {
             models.erase(it);
         }
         mainModel->removePart(selectedModel);
         selectedModel.reset();

         ofLogNotice() << ``Selected model deleted.``;
     }
     break;
```

Listing 3.8: Delete in KeyHandler

Toggle view

Additionally, by pressing the number 1 key, users can toggle between wireframe model view and filled model view. This feature allows users to choose the view that best suits their current work needs, thereby better controlling the display of details during the design process.

Export Model to STL file

Pressing the S key directly invokes the previously mentioned exportModelToSTL method to save all models in the scene as STL files, facilitating the preservation and further utilization of the designs created.

```

case `s`:
    if (selectedModel) {
        ofLogNotice() << ``Exporting selected model to STL file.``;
        mainModel->exportModelToSTL(``yourModel.stl``);
```

```
        ofLogNotice() << ``Selected model exported to STL file.``;
    }
    else {
        ofLogError() << ``No model selected for export.``;
    }
    break;
```

Listing 3.9: Save Model into STL file in KeyHandler

3.3 Smartwatch holder modelling and testing

Design and preliminary modelling

In order to design a base stand for the Google Pixel 2 Watch, the watch body was first roughly modelled using Maya and Tinkercad. This process was informed by the physical watch provided by our mentor Jon, as well as specification data from the official website. The initial design idea was to create a rectangular box that the watch would fit straight into. However, after the first printout it was found that the entrance to the box was too small for the watch to fit in completely. The holder was then redesigned and eventually divided into two parts: a base and a cover. The cover was designed as a trapezoidal structure that could be inserted into the trapezoidal openings on the side of the base, with a circular opening in the centre of the cover allowing the user to view and operate the watch even when the cover was closed. Finally, by splitting the cover along the centre line, the user is able to insert it from both sides of the base to complete the closure.

Printing and Testing

The watch model was scaled up appropriately to ensure that the printed stand would fully accommodate the watch. Initial testing showed that the cover and base could hold the watch and close well, but the cover was on the thin side, making the closing process somewhat difficult. Due to differences in printers, some models opened easily while others required more force. To solve these problems, the entire cover was thickened to half the thickness of the watch. And in testing, it was found that the cover near the crown part of the watch was looser, while the other one was tighter. So the length of the cover near the crown was increased by 0.5mm to solve the problem of looseness and tightness when closed. With these adjustments, the final test results show that the watch fits nicely into the holder and the cover closes with the proper amount of force, neither too loose nor too tight. When closed, the stand holds the watch securely and steadily.

3.4 Decoration design and function realisation

Ornament Design

In the initial stage of ornament design, I participated in a creative workshop with my supervisor Jon and his PhD student Sydney. We worked freely with materials such as cardboard and tape, and the process greatly stimulated my creativity. During the workshop, we created a variety of interesting decorations, including images of flowers, suns and kittens, and these ideas provided a wealth of inspiration for subsequent model designs. So began modelling the decorations using Maya and Tinkercad. Initially, I modelled cat ears and rabbit ears and tested them for 3D printing. The printing results were satisfactory, which verified the feasibility and attractiveness of the design. Then continued to expand the range of ornaments and designed a series of animal wings, including butterfly wings, dragon wings and bat wings. To make it even more vivid, animal leg decorations were also added to add more interest. In addition to this, a loop was added so that users could link the holder with a string and hang it around their neck or elsewhere.

Functional implementation and user optimisation

In order to improve the efficiency and accuracy of ADHD children‘s use of the design software, the project implemented a ‘magnetic attraction’ feature. The core of this feature is that when the user moves the ornament model near the base model with the mouse, the ornament will automatically be attached to the nearest side of the base model. This design greatly reduces the need for placement accuracy and allows the user to easily and accurately position the ornament in the desired location. To further refine the

Figure 3.3: Unclosed holder with cover and bat wings

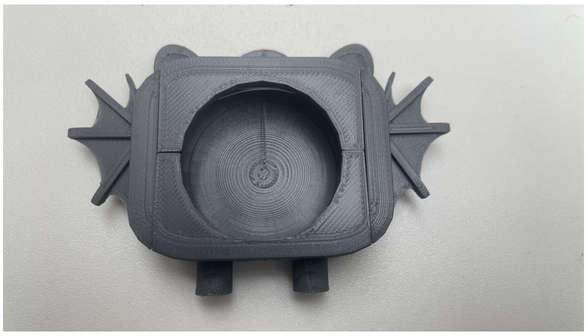

Figure 3.4: Closed holder with cover and bat wings

Figure 3.5: Closed holder with SmartWatch and bat wings

operation and enhance the user experience, the system allows the user to use the keyboard arrow keys to make even finer positional adjustments to the adhered ornament. This flexibility is particularly suitable for ADHD children, helping them to stay focused during the creative process while reducing frustration caused by the complexity of the operation. In addition, considering the visual importance of symmetry, this project implements a symmetry adjustment function for ornament models that appear in pairs (e.g., a pair of cat ears). When the user selects and moves one side (e.g., the left ear), the system automatically adjusts the position of the other side (the right ear) to ensure that both remain visually symmetrical. This feature not only simplifies the operation, but also ensures that the design is aesthetically pleasing. In order to ensure the integrity of the model in the final saved STL file, necessary limitations are imposed on the user‘s manipulation. In the case of cat ears, for example, the system sets boundaries to prevent the user from moving the cat ears outside the base model or too close to the interior, which could interfere with the function of the watch‘s storage compartment. This restriction ensures that the decorations not only visually fit the design intent, but also do not interfere with the functionality of the watch holder. Through these design features, this project aims to provide ADHD children with a modelling environment that is both intuitive and easy to manipulate, supporting their freedom of expression in creative activities while ensuring a sense of success and satisfaction in the process of manipulation.

Figure 3.6: Dragon wings and horns

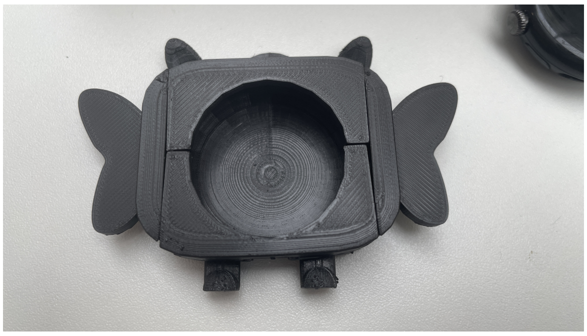

Figure 3.7: Butterfly wings

3.5 Summary

At the beginning of the project, I tried to use Processing as the development tool for software development and production, but due to its limitations in 3D modelling and STL file export, it failed to achieve the expected results. Faced with various technical difficulties such as incompatible libraries and defective export files, I switched to the more powerful openFrameworks for development at the suggestion of supervisor Jon.

In order to provide an intuitive and friendly user experience, the project was designed with a clear, non-intrusive 3D spatial interface. The usability and ease of use of the interface was optimised through clean visual elements and easy-to-understand operation prompts. In addition, keyboard and mouse interaction was designed to simplify operations such as model selection and movement, view switching and model saving, thus enabling users to carry out design work more easily.

Several key features have been implemented in this project, including the 'magnetic attraction' auto-attachment function and the symmetrical operation function. In particular, the symmetry adjustment function for pairs of decorations ensures the aesthetics and consistency of the design while reducing the need for manual adjustment by the user.

In addition to this, the iterative modelling and testing of the smartwatchholder has refined the holder as a whole and many interesting decorations have been constructed.

Through a series of innovative technical solutions and user-centred optimisation, the project successfully overcame initial technical challenges and greatly enhanced the user‘s design experience.

Chapter 4

Critical Evaluation

This section will explore the comprehensive evaluation of the software and the holder, including the user feedback after applying the System Usability Scale (SUS) for interaction, and the functional evaluation of the software functionality as well as the feedback from the ADHD children after the actual use of the smartwatch holder. In addition, this chapter will detail the challenges encountered during the development of the project and suggest future directions for further research and improvement.

4.1 User satisfaction and product usability

System Usability Survey

In this project, the System Usability Scale (SUS) was employed to evaluate the user experience of the software and its 3D printed outputs. The participants, whose ages ranged from 16 to 55, with a majority being males aged 20 to 25, varied in their experience with modeling software like Maya. Some were familiar with and had used such software, others had some knowledge but no practical experience, and a portion were complete novices. The project was showcased on April 17th at the University of Bristol, attended by students, teachers, as well as my classmates, roommates, and family. In addition to live testing, a GitHub link was provided in an electronic SUS questionnaire for users to download the software and conduct remote testing. The test task involved designing and saving an STL file of a preferred smartwatch holder, followed by completing the SUS questionnaire and providing suggestions for improvements.

To date, 27 users have completed the SUS questionnaireA, and feedback has also been received from 2 ADHD children on their experience with the smartwatch holder. The average SUS score was 70, with scores above 68 generally considered acceptable in terms of usability. This indicates that the majority of users found the software intuitive and easy to use. However, some user feedback indicated areas for improvement. For example, a friend who tested the software at the exhibition commented: “Overall, it‘s good and easy to understand, but since I haven‘t used modeling software like Maya before, I found some hindrance in navigating the interface, such as needing to hold down alt to move the view.” Most users who had never used modeling software mentioned difficulties with the interface. The need to hold down alt to activate the camera mode, a common feature in software like Maya, can be challenging for those unfamiliar with such operations and requires more time to adapt.

Furthermore, one female participant noted that the product does not support non-round smartwatches, like the Apple Watch: “This is cool, I have a smartwatch, and I really wanted a holder like this, but unfortunately, I have an Apple Watch, which is square.” Her feedback is invaluable as the ADHD children involved in the test were using Google Pixel watches, which are round, and the project‘s holder models were based on this shape. Moving forward, updates will include modeling for more popular smartwatch types to accommodate a broader user base.

In addition to on-site feedback, a friend who conducted a remote test commented, “It‘s very convenient and quick to make, but I think the menu bar isn‘t very attractive; it could be made to look better.” This friend pointed out the aesthetics of the user interface, suggesting that the menu bar appeared simplistic and could use some beautification. In the future, consideration will be given to employing the ofxDatGui library to enhance the menu bar‘s appearance and improve the overall user experience.

4.2 Functionality Evaluation

4.2.1 Magnetic Attachment Function

The magnetic snap feature is a major highlight of this project‘s design, greatly simplifying the user‘s operational workflow by automatically attracting decorations to the nearest base surface. During the exhibition tests, nearly everyone was able to quickly place decorations without needing multiple adjustments. Most participants greatly appreciated this feature, with one male student who tested it extensively commenting, “I just need to move the decoration near the holder, and it snaps right into place—so cool and super fast!” However, another participant felt it was overly simplistic: “It‘s really convenient, but I think it‘s a bit too simple. I prefer manually moving it into place and then fine-tuning with keys for more of a challenge.” Indeed, while the feature reduces complexity, in future updates, an option will be added to toggle the magnetic snap feature on or off, allowing for more customization and challenge.

In the survey, the question about the performance of the magnetic attraction and symmetry adjustment featuresA.3 in the “Watch craft” modeling software (where 1 is strongly disagree and 5 is strongly agree) received an average score of 4.46. This high score indicates that most users strongly agree that the magnetic snap functionality is outstanding. One user mentioned in the survey, “I thought I would need to spend time moving the model to the right position, but it automatically snapped to the best spot when I released the mouse.” This positive feedback not only demonstrates the practical utility of the magnetic snap feature but also highlights its importance in enhancing user confidence and satisfaction with their designs.

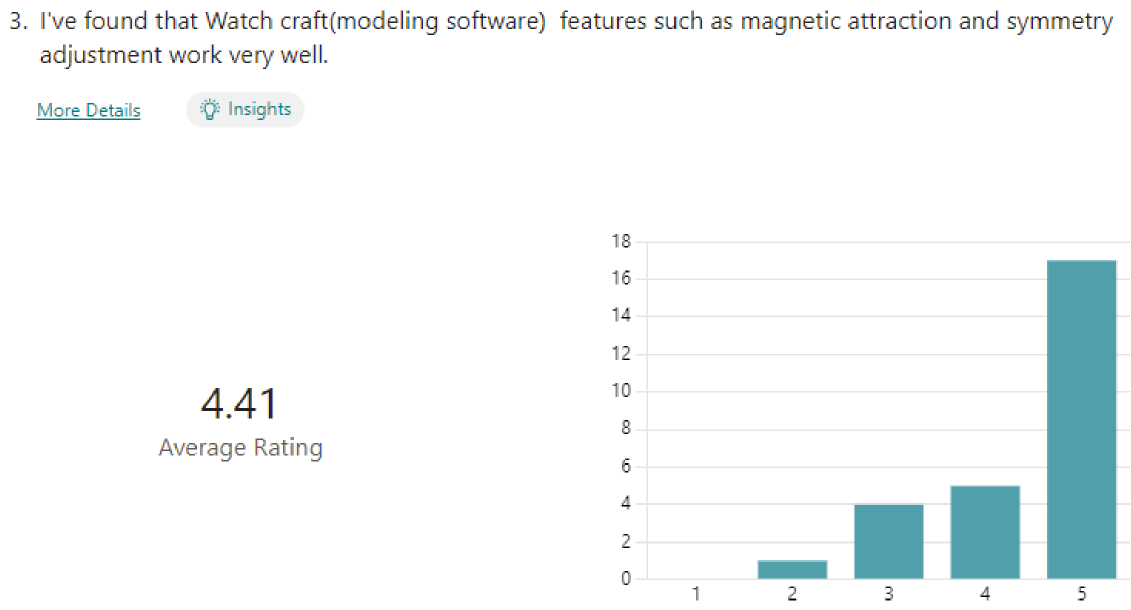

Figure 4.1: Questions about Magnetic and Symmetry functions

4.2.2 Symmetry Manipulation Functions

The model symmetry manipulation functions played a crucial role in also this project, especially when dealing with symmetrical elements (e.g. ears, wings). These features greatly alleviated the need for users to make precise manual adjustments, allowing them to focus more on creative design, thus improving design efficiency and accuracy.

4.2.3 Quantitative Data Analysis

When evaluating the practical benefits of the symmetry operation feature, quantified measurements of time and operation counts clearly supported its advantages. Additionally, qualitative feedback from users provided valuable insights into the utility and user experience of this feature.

The experimental procedure was as follows:

- First Experiment: Participants first clicked the basic holder button to create a holder, then clicked the cat wings button to create a pair of bat wings.
- Second Experiment: Initially, participants clicked the basic holder button to create a holder, followed by clicking the cat ear button to create a pair of cat ears, and then moved the cat ears to the left side of the holder.
- Third Experiment: At the start, participants clicked the basic holder button to create a holder, then clicked the bat wings button to create a pair of bat wings, followed by clicking the cat ears and feet buttons, moving the feet above the holder and the cat ears below.

There were a total of 3 participants. Each participant performed both symmetric and asymmetric operations for the experiments they were involved in, recording the time and number of operations required to complete the tasks. This setup was designed to assess how the symmetry function affects operational efficiency and the time required to complete tasks.

In these experiments, I compared the time and number of operations spent by users using the symmetry function and not using the function when completing the same design task. The specific data are shown in Fig:

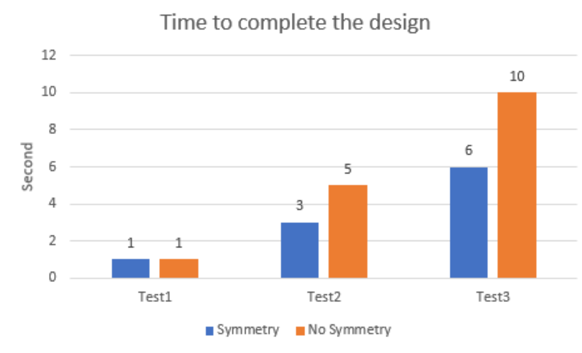

Figure 4.2: Time to complete the design

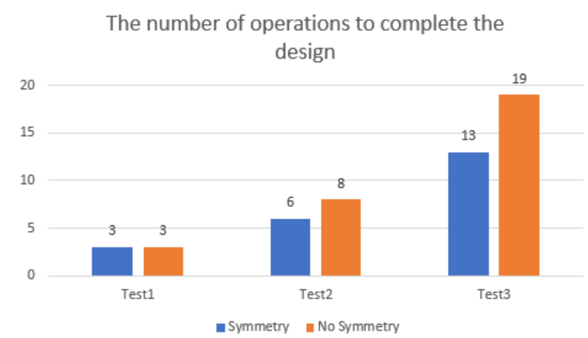

Figure 4.3: Number of operations

Task completion time

As shown in Figure 4.2, experiments 1, 2, and 3 took 1, 3, and 6 seconds, respectively, when using the symmetry function. Without the symmetry function it was 1,5,9 seconds. Since experiment 1 requires model creation and no movement, the time taken is almost the same. The difference between experiments 2 and 3 was almost double.

Number of operations

As shown in Figure 4.3, when using the symmetry feature, the number of operations performed by the three users in the three experiments were 3, 6, and 13 respectively; The steps for each experiment were as follows:

- Experiment One: Click, move, click.
- Experiment Two: Click, move, click, move, hold left mouse button and move, release left mouse button.
- Experiment Three: Click, move, click, move, click, move, click, move, hold left mouse button and move, release left mouse button, move, hold left mouse button and move, release left mouse button.

Without using the symmetry feature, except for Experiment One where the count remained the same as with symmetry at 3 operations, the counts for the other two increased to 8 and 19 respectively.

- Experiment One: Click, move, click.
- Experiment Two: Click, move, click, move, hold left mouse button and move, release left mouse button, hold left mouse button and move, release left mouse button.

- Experiment Three: Click, move, click, move, click, move, click, move, hold left mouse button and move, release left mouse button, move, hold left mouse button and move, release left mouse button, move, hold left mouse button and move, release left mouse button, move, hold left mouse button and move, release left mouse button.

As Experiment One involved only creating two models, the number of operations was consistent whether symmetry was used or not. However, in Experiments Two and Three, it is evident that the number of operations almost doubled when moving symmetrical decorations to different places without the symmetry feature. This further confirms the significant effectiveness of the symmetry function in reducing user workload and increasing efficiency.

Qualitative Feedback Analysis

User feedback reveals the dual impact of the symmetry feature. One user‘s comment is particularly striking: ‘Although I could create the model I wanted more freely without the symmetry feature, the process was too cumbersome. With the symmetry feature enabled, the whole process became much easier and faster.’ This suggests that while the symmetry feature may have limited the creative freedom of individual users to some extent, as a whole it greatly simplified the design process, especially in scenarios where efficiency and precise symmetry were required.

Comprehensive Evaluation

With this data and user feedback, it is clear that the symmetry manipulation feature not only improves design efficiency and precision on a technical level, but also brings a positive impact in terms of user experience. While a few users expressed a desire for more freedom in creation, the majority appreciated the convenience and time savings this feature brought. This feedback suggests that in future iterations of the product, it may be necessary to consider providing a more flexible symmetrical feature set to accommodate the needs and preferences of different users.

4.3 Issues and challenges

In the initial stages of the project, when attempting to use Processing for development, the key technical issue of not being able to export STL files was encountered. Despite spending a week learning and building the basic operator interface and model, basic import and export functionality was not possible due to library incompatibility and file quality issues. This problem seriously threatened the viability of the project, as the handling of STL files was the cornerstone of the project.

At the suggestion of my mentor Jon, I turned to the OpenFrameworks platform, a much more powerful tool that specialises in supporting complex 3D model processing, and the documentation Jon provided on how to set up OpenFrameworks and use the ofxSTL library was extremely helpful. While this transition initially added a learning burden, it ultimately significantly improved development efficiency and output quality, solved the initial problems encountered, and led to greater project flexibility.

4.3.1 Design Challenges and Optimisation for Smartwatch Stand Development

During the development of the smartwatch stand, we encountered a series of design challenges and technical difficulties, experiences that were critical to refining the product design.

Initial Design Challenges

At the beginning of the project, the watch holder was designed with an almost square shape and a very thin cover, resulting in the smartwatch having to be almost completely embedded inside the holderholder. However, because the rounded nature of the watch‘s edges was not taken into account, the initial printout of the holder was unsuccessful in accommodating the watch. The main reason why the watch could not fit completely inside the holder was that the edge used to hold the watch was too narrow. To address this issue, the retention edge was widened to ensure that the watch could fit completely without being too loose.

Iterative Testing and Efficiency Issues

In order to achieve the desired design criteria, we conducted lengthy iterative testing. Considering that each print took approximately 4 hours, the limited number of tests per day significantly reduced the efficiency of the iterations. However, a first version of the model was completed that was nearly perfect, increasing the thickness of the cover to approximately half the thickness of the smartwatch, and adding 0.5mm to the width of the cover near the crown to better hold the watch in place. This version of the model is functionally mature, and the bracket closes well, neither too tightly nor too loosely.

Safety considerations in the design

While the first version of the model performed well functionally, my supervisor pointed out that the edges of the design were too sharp and might not be suitable for children. Based on this feedback, exploration of the second version of the design began with the consideration of a round or oval holder.However, the round design encountered new problems during the 3D printing process. Printers need to add support structures when dealing with round structures, and these supports are not only difficult to remove, but also often leave marks when removed, which affects the aesthetics. Without support, the printed structure would be incomplete due to the lack of support.

Final design optimisation

Ultimately, the choice was made to optimise the design by chamfering each edge with rounded corners, which not only eliminated sharp edges, but also avoided the need to use support structures. In addition, we added small rounded balls at each end of the cover, four additional grooves within the support to fit the balls and a circular track. This design makes it easier for the user to close the stand while maintaining the safety and aesthetics of the design.

Through this series of design iterations and challenge solving, the project not only improved the utility and safety of the product, but also deepened my understanding of 3D printing technology in handling complex shapes.

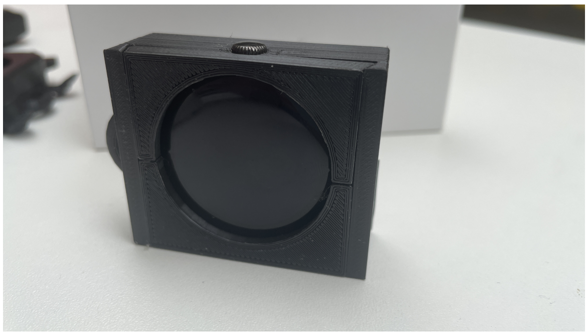

Figure 4.4: First edition perfect holder

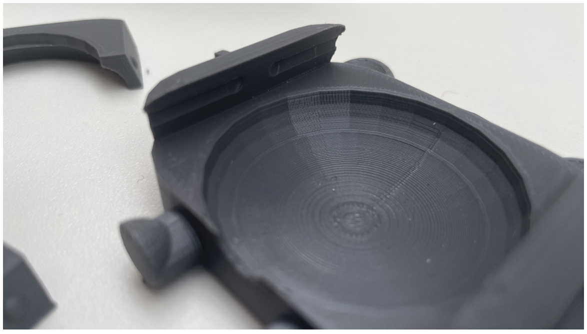

Figure 4.5: The inner wall of the holder

4.3.2 Evaluation of the smartwatch Holder

In the Smartwatch Holder project evaluation, I received significant assistance from Sydney, a PhD student of my supervisor Jon. Midway through the project, Sydney delivered three basic white holders with cat ear decorations to three children diagnosed with ADHD. These children provided valuable feedback through daily journals.

For instance, one child noted in the positive feedback4.6, “Easy to look at, and looks good,” indicating their appreciation for the holder‘s appearance and how it facilitated checking the watch. In the negative feedback, the same child mentioned, “Gets sweaty sometimes!” suggesting that the holder could feel slippery during exercise or in humid conditions. Another child‘s positive feedback stated,4.7 “It looks cool, helps me remember to think about how I feel and what that makes me do. Easy to use. Lanyard makes me feel like a teacher,” indicating that the holder not only reminded him to pay attention to his watch but also gave him a sense of being like a teacher. However, his negative feedback, “It gets in the way sometimes. Can be distracting. Could get lost, stuck a thing,” possibly indicates that the holder sometimes obstructed him, attracted his attention unduly, and was prone to get lost or stuck.

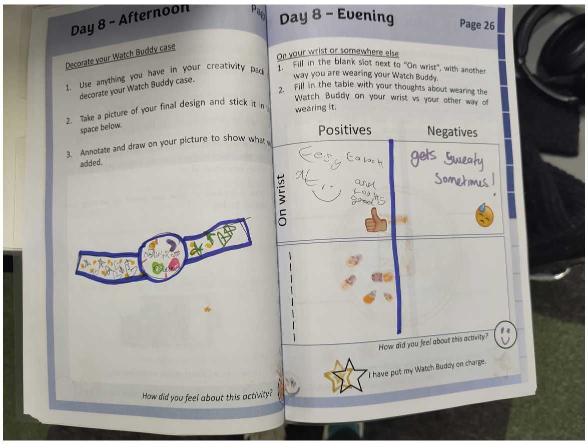

Figure 4.6: Written feedback from a child with ADHD

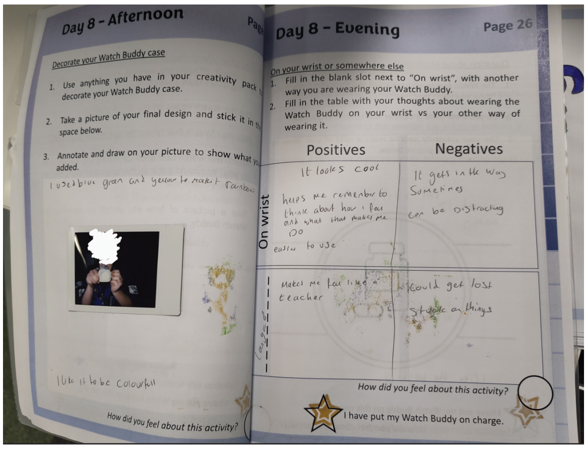

Figure 4.7: Written feedback from a child with ADHD

After discussing this feedback with Jon and Sydney, I learned that the children were very fond of the holder. They particularly enjoyed being able to paint it in their favorite colors, adding a personal touch. However, as this test only provided three standard holders with cat ears, some children might not have fully appreciated this style. Ideally, the children should have the opportunity to use software with their parents to create their own preferred holder designs. Regrettably, I did not have the opportunity to let the children use this software during this test, but I plan to explore the possibility of allowing ADHD children to use this software in future work.

Figure 4.8: The colouring holder Model by ADHD child

Figure 4.9: The colouring holder Model by ADHD child

Regarding the issue of the holder becoming slippery, I encountered a similar situation during testing. I personally sweat a lot on my hands, and handling freshly printed holders often felt slick, which is related to the PLA material used. PLA tends to become sticky in humid environments. To address this issue, theoretically, more water-resistant materials such as nylon or PETG could be used for manufacturing. However, due to limitations with the school‘s printers, I could only test with PLA material.

Regarding the issue of fit, extensive iterative testing was conducted. Feedback from the children indicated that one holder model, which I had not altered in thickness, was likely too tight, while the other two had normal tightness.

4.3.3 Optimisation and future outlook

Implemented and planned optimisation measures

Based on the project‘s evaluation to date, we have implemented a number of product optimisation measures and plan further improvements in the future. Current optimisations include increasing the user safety of the bracket, eliminating sharp edges through rounded corners, and improving support structure issues during the 3D printing process, resulting in a significant improvement in the appearance and usability of the final product. In addition, in response to user feedback, the user-friendliness of the interface will continue to be optimised to ensure that all functions can be operated intuitively, especially for ADHD child users.

Future Directions

In the future, the project will implement changes based on the feedback and issues that have surfaced, such as beautifying the menu and adding a toggle button for enabling/disabling the magnetic snap feature. Efforts will be made to optimize camera movement to allow users to move the camera without the need to hold down the Alt key, thereby separating camera mode from model movement mode more seamlessly.

A key focus of future product development will be the introduction of visual control, implemented through the capture of user gestures with a camera. This advanced feature will allow users to move through the interface with simple hand gestures, offering a more seamless and interactive user experience. Additionally, there will be a heightened focus on research and assessments tailored to children with ADHD. The software's functionalities and user interface will be optimized specifically to meet their needs, enhancing their interaction with the tool and accommodating their unique requirements more effectively.

Product Line Expansion:

As the project progresses, we plan to add more diverse ornament and holder designs, such as covers in various styles and themes, as well as holders that adapt to different smartwatches.While the current design focuses on the Google Pixel 2 watch, it will be expanded in the future to support other mainstream devices, including the Apple Watch.

4.4 Summary

In the evaluation phase of this project, using the System Usability Scale (SUS), valuable data was obtained regarding user satisfaction and product usability concerning the software and its 3D printed outputs, along with invaluable feedback from two ADHD children on their actual use of the holder. The SUS score averaged 70, indicating general satisfaction with the software's interface and functionalities, and suggesting that the product is well-received by users. This score highlights the software's intuitiveness and ease of use but also points to areas for improvement, particularly in enhancing the intuitiveness of the user interface and optimizing the user experience. The children generally liked the holder they created; however, they also provided feedback that they did not like the material of the holder as it tended to become sweaty.

From a functionality assessment perspective, the magnetic snap feature performed excellently, speeding up the placement of decorations. Additionally, the automatic symmetry operation feature significantly enhanced design efficiency and accuracy, reducing the need for manual adjustments and allowing users to focus more on creative design.

Through this quantitative and qualitative feedback, it became apparent that while most users appreciated the convenience and time savings provided by these features, a minority of users desired more freedom during the creative process. Therefore, future iterations of the product need to consider offering a more flexible set of functionalities to meet diverse user needs and preferences.

Overall, this evaluation not only highlighted the current design‘s strengths but also clearly defined directions for improvement. To further enhance user satisfaction and product usability, ongoing optimization of the product will continue based on user feedback. This will focus on enhancing the intuitiveness of features and the overall user experience to ensure that our design meets and exceeds user expectations.

Chapter 5

Conclusion

This paper details the process of developing a smartwatch mount for children with ADHD, a process that covered all stages from initial concept to final product realisation. The project not only focuses on the functional utility of the product, but also takes into account the special needs of children with ADHD, such as sensitivity to the tactile sensation of objects and adaptation to traditional watch designs, by adding a loop for the user to use the lanyard to hang around their necks or on their clothes and bags. Through this customised design, it is hoped that the acceptance of smart wearables by this group will be increased and their experience enhanced.

Project Review and Technical Innovation

On the technical side, software was developed to allow users to customize their designs and export STL files, which can then be directly realized through 3D printing. This software utilizes openFrameworks, providing the necessary flexibility and responsiveness to support complex design requirements and rapid prototyping. Additionally, the project plans to incorporate camera-based gesture control features in the future, which will further simplify the user interface—particularly beneficial for children with ADHD who may have limited motion control. The inclusion of this feature is expected to significantly improve the product‘s usability and appeal.

User Evaluation and Feedback

The success of this project owes greatly to user feedback and active participation. Through the application of the System Usability Scale (SUS), feedback was obtained regarding the user experience of the software and the 3D printed products. This feedback yielded an average SUS score of 70, indicating that most users are satisfied with our design and find the interface intuitive and easy to use. Nevertheless, the feedback also highlighted areas needing further improvement, especially in enhancing the intuitiveness of the interface and reducing the difficulty of use.

Moreover, feedback from the ADHD children about the holder is even more critical. They all greatly enjoyed using the holder, but due to the material used, the children commonly experienced it as being very sweaty. This is an aspect that needs to be addressed in future developments.

Functional Testing and Evaluation

In terms of functionality testing, the introduction of magnetic snap and symmetry operation features significantly enhanced the efficiency and accuracy of the design process. For example, in functional tests, users employing the symmetry feature significantly reduced the time and number of operations needed to complete designs, directly improving design efficiency.

Future Outlook

Looking ahead, we plan to continue expanding the product‘s functionality and customization options. Planned improvements include expanding the range of holder models to accommodate more types of smartwatches, such as the Apple Watch, and adding more personalized decoration options to enhance the product‘s appeal and market competitiveness. Additionally, ongoing optimization and simplification of the user interface will remain a focus to ensure that all users can easily master and enjoy the design and usage process.

In conclusion, this project not only provides innovative solutions for a user group with special needs but also, through practical product development and user feedback, offers valuable experience and insights for the future development of wearable technology. As technology advances and user needs evolve, we look forward to applying these experiences to broader scenarios, continuing to drive innovation and popularization of wearable technology.

Bibliography

[1] Openframeworks. https://openframeworks.cc. Accessed: date.

[2] Processing. https://processing.org. Accessed: date.

[3] Elizabeth A. Ankrah, Franceli L. Cibrian, Lucas M. Silva, Arya Tavakoulnia, Jesus A. Beltran, Sabrina E. B. Schuck, Kimberley D. Lakes, and Gillian R. Hayes. Me, my health, and my watch: How children with adhd understand smartwatch health data. *ACM Transactions on Computer-Human Interaction*, 30(4):Article 59 (25 pages), August 2023. doi:10.1145/3577008.

[4] Megan McClelland, John Geldhof, Fred Morrison, Steinunn Gestsdóttir, Claire Cameron, Ed Bowers, Angela Duckworth, Todd Little, and Jennie Grammer. Self-regulation. *Handbook of life course health development*, pages 275–298, 2018.

[5] Lucas M. Silva, Franceli L. Cibrian, Elissa Monteiro, Arpita Bhattacharya, Jesus A. Beltran, Clarisse Bonang, Daniel A. Epstein, Sabrina E. B. Schuck, Kimberley D. Lakes, and Gillian R. Hayes. Unpacking the lived experiences of smartwatch mediated self and co-regulation with adhd children. New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3544548.3581316.

[6] Michael J Sulik, M Paula Daneri, Alyssa I Pintar-Breen, and Clancy Blair. Self-regulation in early childhood: Theory and measurement. In *Child Psychology*, pages 123–140. Psychology Press, 2016.

[7] T. E. Wilens and T. J. Spencer. Understanding attention-deficit/hyperactivity disorder from childhood to adulthood. *Postgraduate Medicine*, 122(5):97–109, 2010. doi:10.3810/pgm.2010.09.2206.

[8] Faith Wilkins. How is the adhd brain different? *Child Mind Institute*, 2024. Last accessed on April 10, 2024. URL: https://childmind.org/article/how-is-the-adhd-brain-different.

[9] Yuanyuan Zheng, Rongyang Li, Shan Li, Yudong Zhang, Shunkun Yang, and Huansheng Ning. A review on serious games for adhd. *ArXiv*, abs/2105.02970, 2021. URL: https://api.semanticscholar.org/CorpusID:234093291.

[3]

Appendix A

Appendix A: SUS Questionnaires

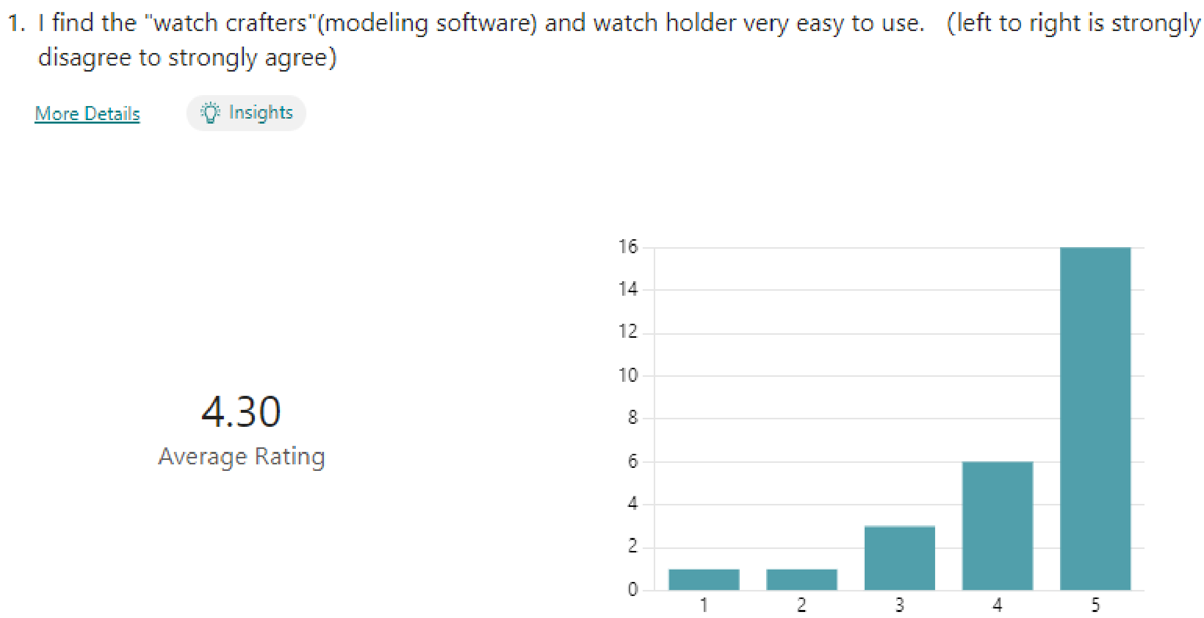

Figure A.1: SUS

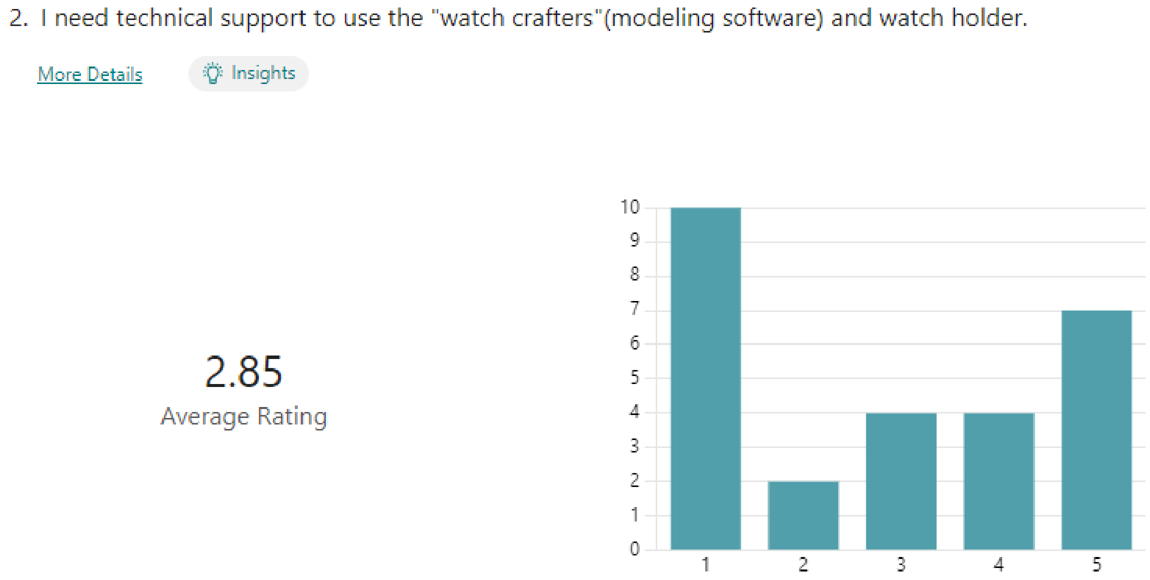

Figure A.2: SUS

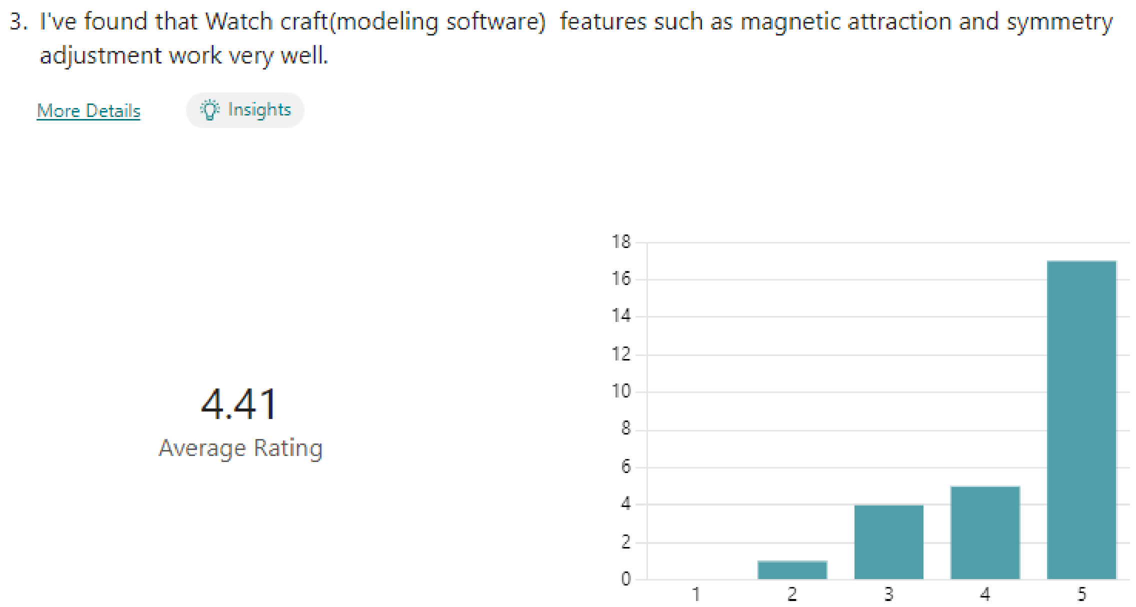

Figure A.3: SUS

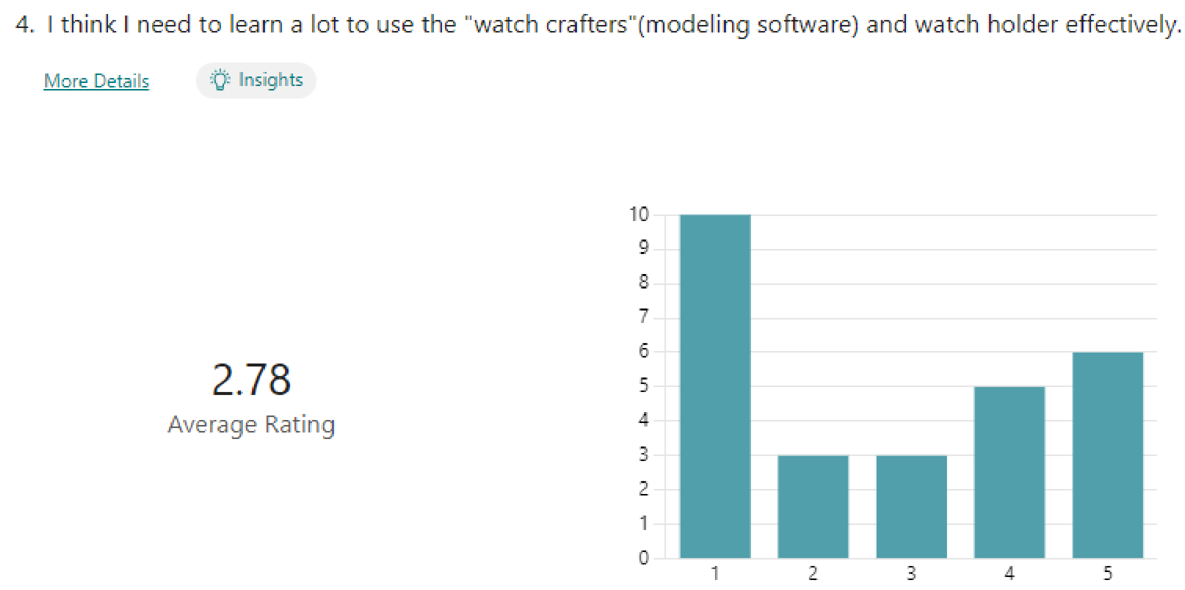

Figure A.4: SUS

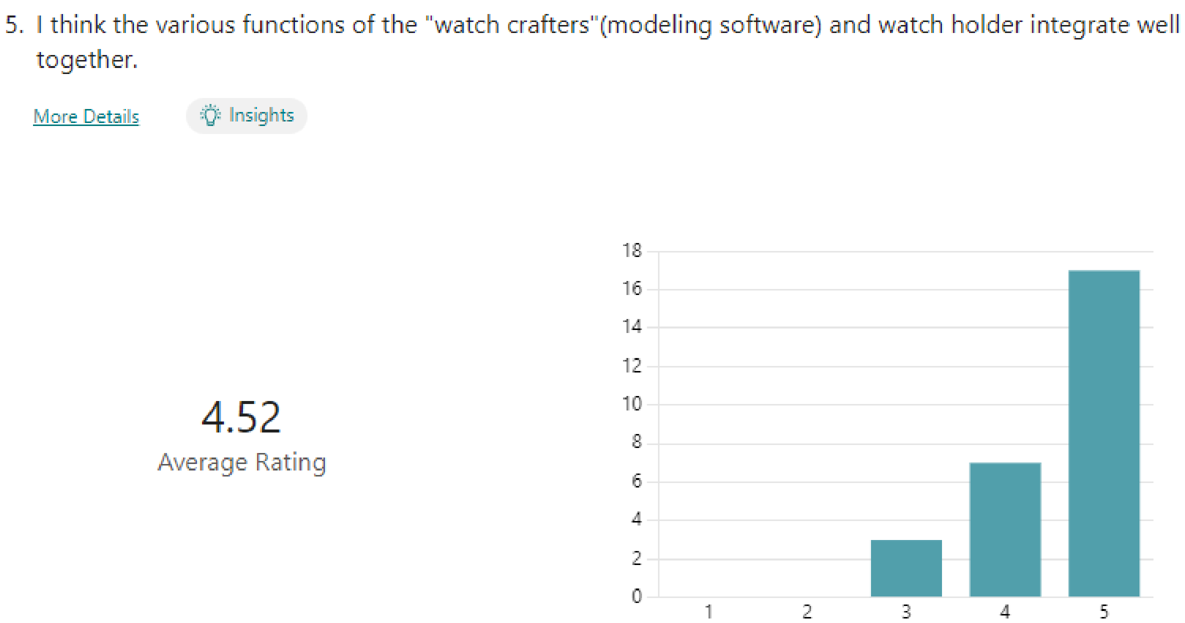

Figure A.5: SUS

6. I think the "watch crafters"(modeling software) and watch holder is too complicated.

More Details   Insights

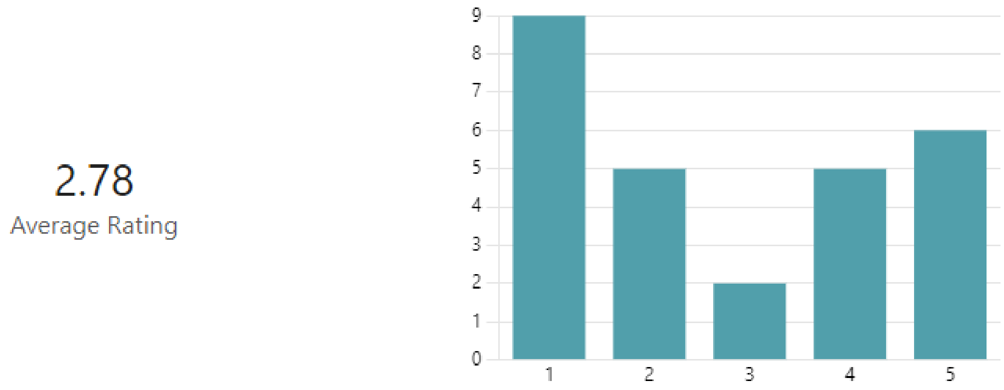

Figure A.6: SUS

7. I think most people would be able to learn how to use the "watch crafters"(modeling software) and watch holder quickly.

More Details   Insights

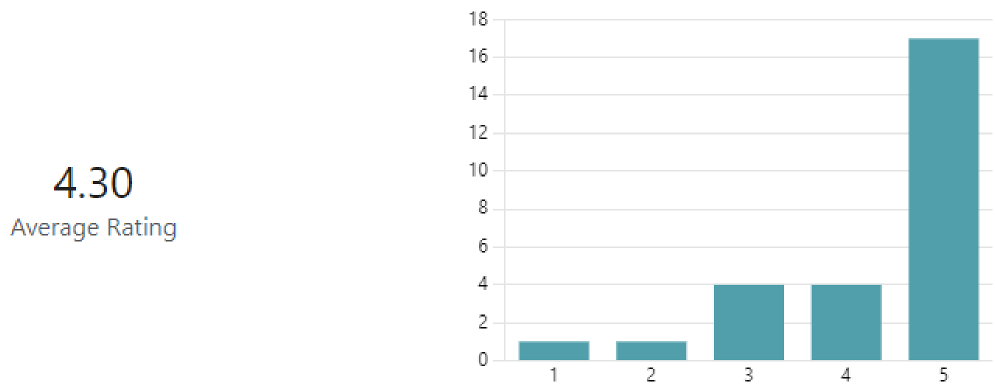

Figure A.7: SUS

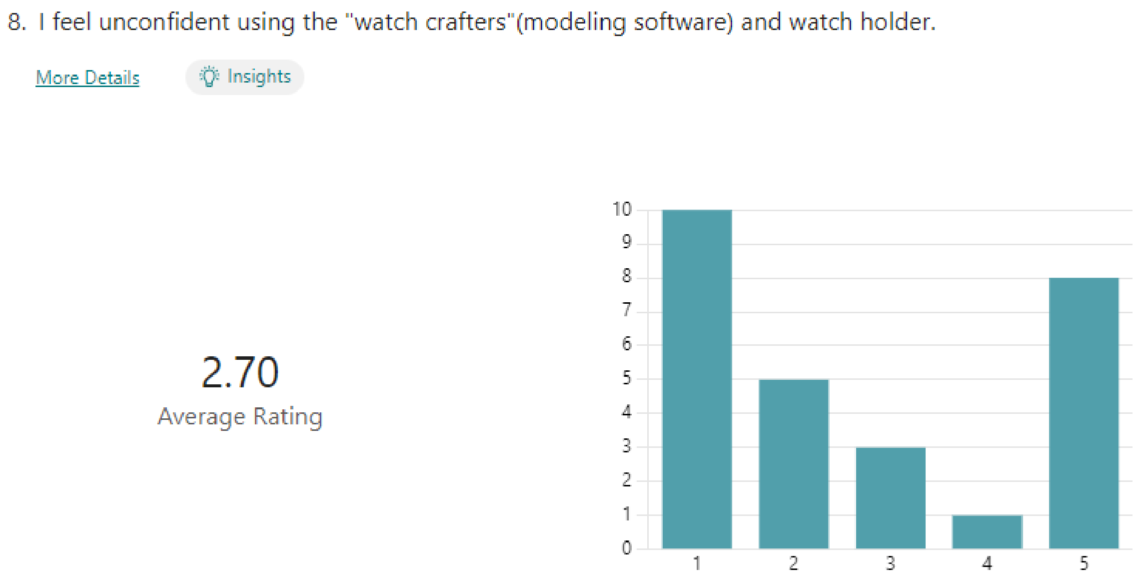

Figure A.8: SUS

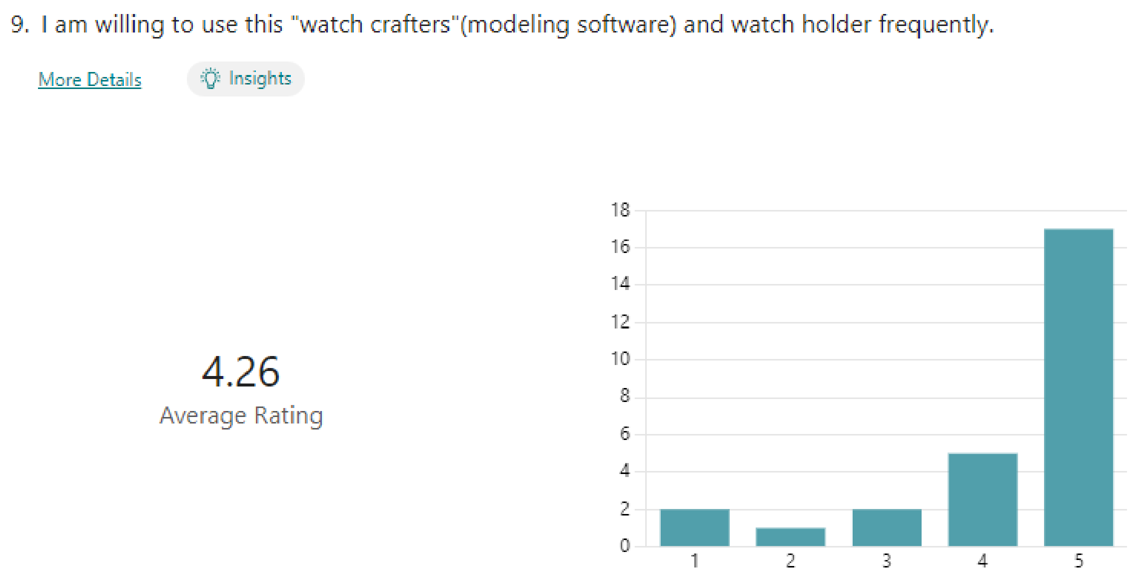

Figure A.9: SUS

10. I need to study repeatedly to master the "watch crafters"(modeling software) and watch holder.

More Details   Insights

3.00
Average Rating

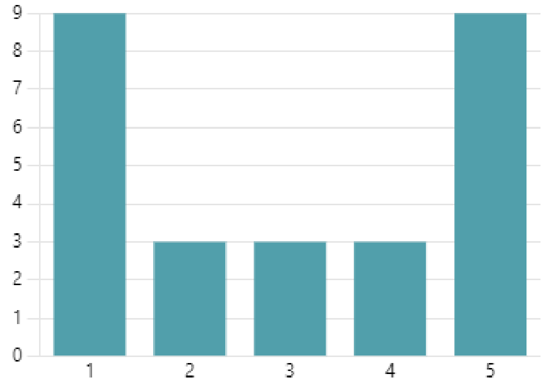

Figure A.10: SUS

11. If you have any questions or suggestions, please let us know.

3 Responses

ID ↑	Name	Responses
1	anonymous	ok
2	anonymous	I just need to move the decoration near the base and it sucks itself over, it's really cool and super fast!
3	anonymous	I thought I needed to spend time to move the model to the HoM position, but I didn't realise that it would automatically attach itself to the most suitable position when I let go of the mouse.

Figure A.11: SUS

Appendix B

AI

https://www.deepl.com/translator