

# A Toolkit for Creating Dynamic Tangible Interfaces Using Compliance Illusion

---

A

Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Master of Science

by

Spencer Thomas Keefer

May 2023

# APPROVAL SHEET

This  
Thesis  
is submitted in partial fulfillment of the requirements  
for the degree of  
Master of Science

Author: Spencer Thomas Keefer



This Thesis has been read and approved by the examining committee:

Advisor: Seongkook Heo

Advisor:

Committee Member: Brad Campbell

Committee Member: Gregory J. Gerling

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:



Jennifer L. West, School of Engineering and Applied Science

May 2023

A TOOLKIT FOR CREATING DYNAMIC TANGIBLE INTERFACES USING  
COMPLIANCE ILLUSION

Spencer Thomas Keefer  
Clayton, North Carolina

Bachelor of Science, Virginia Tech, 2017

A Thesis submitted to the Graduate Faculty  
of the University of Virginia in Candidacy for the Degree of  
Master of Science

Department of Computer Science

University of Virginia

May 2023

Brad Campbell, Chair  
Seongkook Heo, Advisor  
Gregory J. Gerling



# A Toolkit for Creating Dynamic Tangible Interfaces Using Compliance Illusion

Spencer Thomas Keefer

(ABSTRACT)

Tangible and haptic interfaces can enable more intuitive and engaging interactions by allowing users to manipulate the physical control that reacts differently based on the virtual context. However, building such interfaces often requires the development of electronic circuits, moving mechanisms, firmware, and a device driver, making it difficult to be built by designers or users without extensive knowledge. We propose PseudoWidgets, a toolkit developed to allow users to prototype dynamic tangible interfaces easily. By using the deformation haptic illusion created using vibrotactile feedback, users can create tangible controls of various types of movement without creating linkages and using motors. Using PseudoWidgets CAD tool, users can easily add virtual controls, such as buttons, levers, joysticks, and knobs, on a 3D model of their own. The CAD tool automatically generates structures to place haptic modules for the user to print using a 3D printer. Once assembled, the haptic modules will communicate with a computer over WiFi to dynamically adjust its movement properties based on the virtual context.

# Acknowledgments

I would first like to thank my advisor and the rest of the Ultimate Interface Lab team for providing me with support and insights on how to improve my research this past year. Specifically I would like to thank Professor Heo for getting me interested in haptic feedback and human-computer interaction research while enrolled in his Engineering Interactive Technologies course last Spring. Next I would like to thank my old high school engineering teacher Mr. Bill "Ralph" Evans who first introduced me to computer-aided design with Autodesk Inventor and years ago; this knowledge is something I never would have imagined needing again for my Computer Science thesis and it really helped me with creating the Designer Tool add-on in Fusion 360. Finally I would like to thank and dedicate this paper to my parents Tom and Terri Keefer, my sister Ragan, my girlfriend Taylor, and my friends from Blacksburg, Clayton, and here through the CS graduate program.

# Contents

<b>List of Figures</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Related Work</b>	<b>4</b>
2.1 Dynamic Reconfiguration of Tangible Interfaces . . . . .	4
2.2 Illusion of Material Compliance . . . . .	5
2.3 Toolkits for Tangible Interfaces . . . . .	7
<b>3 PseudoWidgets: Utilizing Pseudo Movement</b>	<b>10</b>
3.1 Widget Types . . . . .	11
3.2 Software-induced Dynamic Properties . . . . .	13
<b>4 PseudoWidgets Implementation</b>	<b>14</b>
4.1 PseudoWidgets Designer . . . . .	14
4.1.1 Generating a Button Widget . . . . .	16
4.1.2 Generating a Lever, Knob, or Joystick Widget . . . . .	17
4.2 PseudoWidgets Connector . . . . .	17
4.3 Application Programming Interface . . . . .	19

4.4	PseudoWidgets Module . . . . .	21
<b>5</b>	<b>Evaluation</b>	<b>24</b>
5.1	Individual Widgets . . . . .	25
5.2	Application-specific Controllers . . . . .	25
5.2.1	Video Game . . . . .	25
5.2.2	Music Player . . . . .	26
<b>6</b>	<b>Discussion</b>	<b>28</b>
6.1	Expandability of PseudoWidgets . . . . .	28
6.2	Limitations . . . . .	29
<b>7</b>	<b>Future Work</b>	<b>31</b>
<b>8</b>	<b>Conclusion</b>	<b>33</b>



# List of Figures

1.1	PseudoWidgets allow designers and developers to use a standardized module to prototype various control widgets, such as buttons, levers, knobs, and joysticks. PseudoWidgets also offers a PseudoWidgets Designer, where the user can modify existing 3D models to have structures to accommodate the PseudoWidgets modules. PseudoWidgets API allows developers to easily use the devices developed using PseudoWidgets for their applications. . . . .	2
3.1	The Four PseudoWidgets: Button (top left), Lever (top right), Knob (bottom right), and Joystick (bottom left). Transparent green region represents PseudoWidgets module placement. . . . .	12
4.1	(a) Creating a widget outline (b) Selecting widget outline for generation (c) Generated widget output (d) Widget pieces separated and sent to AnkerMake slicing software (e) Final printed product . . . . .	15
4.2	(a) PseudoWidgets Connector interface with one sensor module assigned to a Button widget. (b) Diagram of force visualization via gradual color change. . . . .	18
4.3	Example Python code instantiating a PWManager object and invoking some supported commands. . . . .	20
4.4	PseudoWidget Module. (a) Top and bottom views of the device, (b) electronic components in the module. . . . .	21

4.5	Sensor measurement by force, and the estimated force after calibration using quadratic curve fitting. . . . .	22
5.1	Haptic Space game and PseudoWidget controller . . . . .	26
5.2	Music player app and PseudoWidget controller using Button on the front side and a lever and a knob on the top . . . . .	27

# List of Abbreviations

API Application Programming Interface

CAD Computer-aided Design

FSR Force Sensing Resistor

Li-Po Lithium Polymer Battery

LRA Linear Resonant Actuator

PWC PseudoWidgets Connector

UDP User Datagram Protocol

VR/AR Virtual and/or Augmented Reality

# Chapter 1

## Introduction

Tangible and haptic user interfaces allow users to physically manipulate the user interface to interact with the system. These interfaces offer several benefits, such as the physical affordance (Fitzmaurice, Ishii, and Buxton 1995), eyes-free interaction (Taylor et al. 2009; Kirk et al. 2009), and more engaging experiences. Some user interfaces use actuators and control movements or deformation of the user interface based on the virtual context, such as the material of the virtual object (Nakagaki et al. 2019; Benko et al. 2016) and movement constraints (Shim et al. 2020). While useful, these physical interfaces are not accessible to many users since many devices are tailored to specific applications, such as video games (Shim et al. 2020), or bulky and heavy to build (Nakagaki et al. 2019) to be used in everyday life for various uses. Fortunately, 3D printers and physical interface prototyping toolkits (Savage, Chang, and Hartmann 2013; Lin et al. 2022) can help users build interactive physical interfaces that can sense user input (Savage, Chang, and Hartmann 2013) and produce passive (Lin et al. 2022; W. Y. Yang et al. 2022) and active haptic feedback (Cruz et al. 2021).

However, prototyping a tangible user interface that can dynamically vary its haptic properties is still challenging for designers and developers who do not have expertise in hardware prototyping. Even with the help of the toolkits, users often need to prepare electric components and assemble them (Ramakers et al. 2016) in addition to

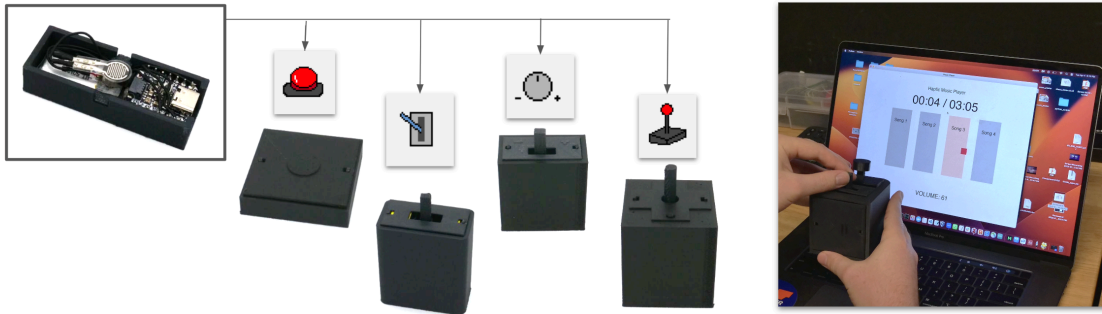


Figure 1.1: PseudoWidgets allow designers and developers to use a standardized module to prototype various control widgets, such as buttons, levers, knobs, and joysticks. PseudoWidgets also offers a PseudoWidgets Designer, where the user can modify existing 3D models to have structures to accommodate the PseudoWidgets modules. PseudoWidgets API allows developers to easily use the devices developed using PseudoWidgets for their applications.

3D printing modules, unless they use special equipment that can fabricate functional devices (Nisser, Liao, et al. 2021). This is inevitable as changing physical properties requires mechanical or pneumatic actuators that move the structure.

To demonstrate the potential of using the illusion movement in prototyping tangible interface, we developed PseudoWidgets, a new toolkit that simplifies the process of adding illusion compliance to 3D printed controllers. PseudoWidgets uses PseudoWidgets module, a small, self-contained module equipped with a force sensor, vibrotactile actuator, microcontroller, and a battery, to simulate a moving physical control and programmatically change the physical properties of the control. PseudoWidgets has two main components – PseudoWidgets Designer and Connector tools. PseudoWidgets Designer is a Fusion 360 add-in, and users can bring their own 3D model and add a haptic interface to the model by simply specifying the type and the location of the interface. The system will then modify the loaded model and create spaces for the PseudoWidgets module to be inserted into. Once the design

is complete, the users can 3D-print the model to create a hardware controller. The fabricated object will have slots for the user to insert PseudoWidgets module, which is a device with a force sensor and an LRA that creates the compliance illusion. This process does not require the user to prepare any electric or mechanical components. Different types of control, e.g., a button or a lever, may use different number of PseudoWidgets modules. For example, a push button can be created with a single module and a joystick may need four modules to support the 2-Dimensional movement. PseudoWidgets Connector is designed for users to spatially assign modules to their widgets. Once the control configuration is made in the PseudoWidgets Connector, the application can access the PseudoWidget in an abstract form (e.g., 2D joystick). We conducted a preliminary study to understand the effectiveness of the dynamic tangible interfaces created with PseudoWidgets by exposing users to a set of sample applications that come with a custom controller and use PseudoWidgets's API.

In summary, this work contributes 1) the concept and mechanisms of using vibration-based deformation illusion for creating different types of controls; 2) the development of PseudoWidgets, a toolkit that consists of a Designer and a Connector, and physical modules that generates the illusion.

# Chapter 2

## Related Work

### 2.1 Dynamic Reconfiguration of Tangible Interfaces

Many haptic interfaces tailor to one specific use, making them difficult for use in other applications. Because of this limitation, researchers have experimented with concepts that offer multiple modes of interaction. One concept for enriching tangible interactions is the use of several actuated pins that emerge from flat surfaces (Jung, Youn, and G. Lee [2017](#); Robinson et al. [2016](#)). Such devices offer physical widgets on demand that users control based on the current context of the device. For example, if a user wanted to adjust the volume on a device, multiple pixels would emerge to form a small dial that a user can control. Several virtual and augmented reality works also rely on actuator mechanisms of similar or lower resolution to provide a sense of touch to complex virtual objects and surfaces (Follmer et al. [2013](#); Suzuki, Ofek, et al. [2021](#); Suzuki, Hedayati, et al. [2020](#); Siu et al. [2018](#); Suzuki, Nakayama, et al. [2020](#)). All of these ideas take up a lot of space and require assembling several moving parts which is not ideal for small devices like smartphones and tablets. This prompted researchers to consider mobility when designing devices for tangible interfaces. Haptic Tabletop Puck is one such example that utilizes a small, hand-held device with a single moving rod that updates its behavior based on where it is positioned on a touchscreen surface (Marquardt et al. [2009](#)). This offers greater mobility than using

pins to cover an entire interactable surface. Haptic Touch Toolkit provides an API to implement haptic buttons which are composed of graphical 2D button images with different states, where the moving rod of the HTP updates its software properties when hovering over the button when disabled, enabled, or clicked on (Ledo et al. 2012). However, one major drawback of the toolkit is the absence of other haptic widgets such as levers, switches, or rotating dials where forces applied are naturally perpendicular to the surfaces they are on. FS-Pad offers a dynamic force feedback analog replacement for a video game controller (Shim et al. 2020). However, its reliance on motors and moving parts prevent it from being scaled down. ReCompFig provides multimodality to a tangible interface through a set of tensioning cables. The tightening and loosening of different cables in a set allow users to specify degrees of freedom for the interface, allowing it to emulate different widgets (H. Yang et al. 2022). This solution also suffers from using many moving parts as it relies on both tensioning cables and motors, which makes it difficult to scale down the interface.

## 2.2 Illusion of Material Compliance

Vibrations are a common source for supplying tactile feedback to users from a standalone device or wearable, on-body applications. Tactile cues aid in providing information to the skin such as spatial indicators, material compliance, or material softness.

One such technique for vibrotactile feedback, the technique we want to employ in this work, is the simulation of material compliance on a rigid body using grain-based vibrations. The use of grain-based vibrations was first proposed by Kildal when creating an abstract model for material compliance (Kildal 2010). The model



described how a rigid body could behave elastically when subject to an external force. Pressing on the surface would act as if it were attached to a spring that could displace on a single axis. Kinesthetic cues that naturally occur when displacing a spring are replaced with cutaneous vibration grains, where a certain amount of force was needed to reach the next vibration grain or step. Kildal then applied the model in a scenario where a user interacts with a force-sensitive tablet using a stylus rigged with a linear vibrotactile actuator. When a user exerts force on the tablet, force information is transmitted to the actuator on the stylus to render grain-based vibrations. The vibrations permeate across the entire pen and the holder perceives the material compliance from his hand holding the pen. A qualitative study was performed where the goal was to see if participants could perceive material compliance using the stylus without knowledge of its capabilities. When pressing in a designated area, participants could feel the surface's simulated elasticity and distinguish it from a regular rigid surface. Heo, J. Lee, and Wigdor (2019) later employs the same technique to create the illusion of bending, twisting, or stretching a rigid device in multiple degrees of freedom. The design choice was a cylindrical rod that a user would hold with both hands, one on each end of the device. The device housed a 6-DOF force sensor and vibrotactile actuator that would produce grain-based vibrations in response to the amount of force or torque applied by a user holding the device. Software parameters were introduced to adjust the step size of vibration grains and max force allowed in the handle bar, allowing for varied material compliances to render. A psychophysical study was conducted where participants placed their hands inside an opaque box and held the device without seeing it. Different illusions were simulated and participants were evaluated based on their perception of the material compliance. The results showed that participants could perceive different material stiffnesses when interacting with the rod. Grain-vibrations have also been adopted

to gain a sense of physical interaction with virtual buttons on a touchpad, grains are mapped on a force-displacement curve to mimic the interactions before and after the click occurs on a physical button. The technique is also adopted for rendering cues in a person's foot. Strohmeier et al. (2020) designs shoes for VR and AR applications where vibration grains are used to render different materials being pressed on with shoes. Visell, Law, and Cooperstock (2009) shows how floor panels can respond to applied force to simulate different ground materials. Marquardt et al. (2009) and Ledo et al. (2012) simulates breakiness in a touchpad surface using a vertical rod with a servo motor, the breakiness value was adjustable and specified number of breakpoints (similar to grains) placed to emulate surfaces such as sand. Instead of grain-vibrations, the breakpoints represent areas where a vertical rod doesn't displace.

## 2.3 Toolkits for Tangible Interfaces

Ramakers et al. (2016) provides a toolkit, RetroFab, that modifies the existing layouts for legacy devices such as TV remotes and toasters. The process scans a legacy device and then, using a CAD automation tool, creates a 3D printed enclosure with new mechanical widgets which a user can define widget properties using a GUI. Groeger et al. (2019) uses a toolkit for designers to rapidly print electro-tactile controls called Tactlets that can attach to tangible objects. Multiple tools present design methods for applying passive force feedback mechanisms to physical interfaces using planar geometries configured in a GUI for haptic customization and fabricated by 3D printing or laser cutting (Zheng et al. 2022; Lin et al. 2022). Other toolkits help rapidly sketch and convert passive objects into functioning controllers by sensing user interactions through an overhead camera and registering widgets with a computer appli-

cation (Savage, Chang, and Hartmann 2013; Holman and Benko 2011). Shibasaki, Tuchiya, and Minamizawa (2017) democratizes haptic device creation for children with TECHTILE bits toolkit, enabling children to add haptic effects to their toys and other tangible objects. OmniSoft facilitates the printing of 3D printed products with specified regions of softness where the softness value is in reference to real world objects (Kim et al. 2021). The design space of haptic toolkits also extends to shape-memory alloys for skin stretching feedback (Messerschmidt et al. 2022; Nakao et al. 2019), magnets (Yasu 2020; Nisser, Makaram, et al. 2022), audio feedback (Israr et al. 2019), and pneumatics for shape-changing surfaces (Zhang and Sra 2021; Ghosal et al. 2019).

We also want to make note of other existing toolkits that employ vibrotactile feedback. Haptic Enchanters supplies small-form boxes containing actuators that can attach to smartphones or other small IoT devices to create spatial tactile feedback (Park, Cha, and Choi 2019). Google Research offers their own toolkit for prototyping wearable vibrotactile devices using a driver board that can support up to twelve different vibrotactile actuators with sufficient voltage. The implementation was applied to a sleeve, wrist band, and phone case (Dementyev et al. 2021). Cruz et al. (2021) offers five different self-contained modules, one being a vibrotactile module that uses an eccentric rotating mass (ERM) motor to simulate touch and a skin-safe adhesive for easy attachment on a user’s body. A user can customize the intensity and duration of the vibration played. All are great examples for mobilizing vibrotactile feedback but all focus on passive touch techniques, where a user does not control the sense of touch.

Our work builds on the insights and findings of studies on haptic toolkits that are capable of simplifying the design process and lowering the skill prerequisites needed for

crafting effective haptic and tangible interfaces. Different from others, our method utilizes illusory movement caused by grain vibrations and uses a standardized PseudoWidgets module, making tangible interface prototyping soldering- and wiring-free while allowing dynamic control of physical properties, such as stiffness and maximum movement. To the best of our knowledge, this is the first toolkit for creating tangible interfaces using the deformation illusion.

## Chapter 3

# PseudoWidgets: Utilizing Pseudo Movement

Many haptic toolkits offer a variety of tools that enable users to quickly prototype devices with haptic feedback. For an end-to-end toolkit, it is important to provide tools that each specialize in a different stage of the design process. Specifically for tangible controls, we want to facilitate the designs of ubiquitous interfaces that are used to perform daily tasks such as buttons and levers. A toolkit for the material compliance illusion technique requires three distinct tools: design software for creating and fabricating controllers, a management tool for configuring wireless sensor modules, and a programming interface that application developers can invoke in their source code to communicate with a controller's registered modules. For creating tangible input devices with material compliance illusions, there is a large emphasis on the structural design of rigid bodies that allow for effective vibration permeation. Poor design choices would likely result in illusions failing to give a user the perception of moving parts. This convinces us that a design tool is crucial as it can generate rigid widgets that are known to supply sufficient vibrations in response to user input. Using a design tool, a user can simply designate a section of a rigid body to generate a structure for user input. Many CAD suites are accessible to users online for making products and offer extensive libraries for developing add-ons that introduce new features. Plus,

some CAD programs provide built-in functions for exporting CAD designs to slicing programs such as Cura for 3D printing. The other tools are needed if we are allowing third parties to customize and communicate with wireless sensor modules. Controller designers need a simplified way of registering devices to a controller design, allowing them to deploy a data structure for application developers to interact with using API commands. And without a dedicated interface for app developers to use, they would have to handle all the communication protocols from scratch which can be tedious. Providing these three tools can sufficiently cover the entire design process from early product design stages to deployment of the final product.

### 3.1 Widget Types

Rigid widgets should represent the common input devices that people use to perform daily tasks. For PseudoWidgets, we chose four widgets to represent the building blocks for creating our interfaces: a button, a lever, a rotating knob, and a joystick (see Fig. 3.1). The button acts as a pressure plate, where a sensor module can rest under a thin surface that a user will apply force on. Applying force on the plate creates a small displacement at the point of contact. This displacement of the plate then hits the force sensor in the module which responds with vibrations from the LRA. The lever, rotating knob, and joystick widgets share a different approach since users apply forces on these widgets that are parallel to the controller's surface. To capture these interactions, force sensors must be positioned orthogonal to and under the controller's surface allowing them to oppose the incoming forces. This can be accomplished using the stick or shaft feature that comes naturally with real levers, knobs, and joysticks. Rigid shafts can be created with extruded bumps or prongs

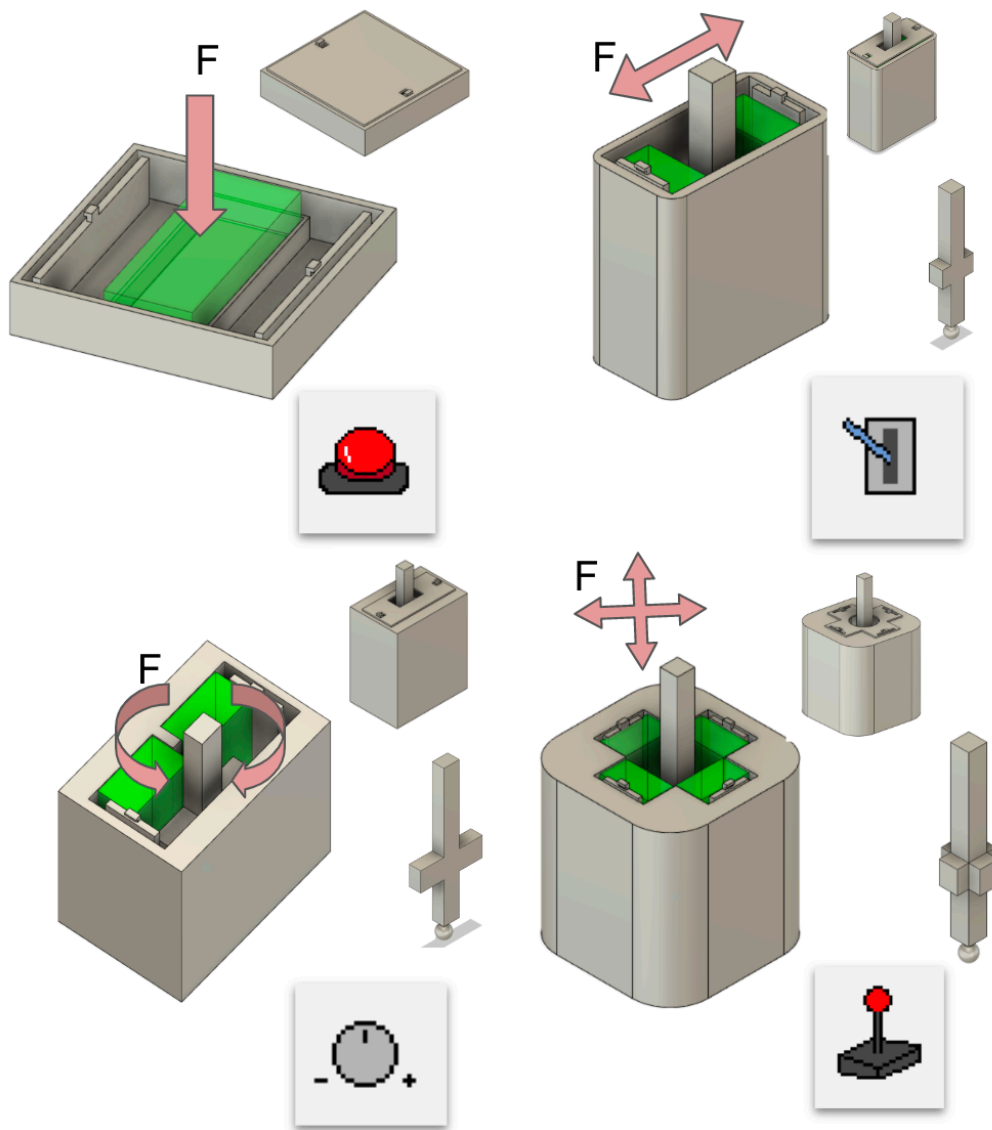


Figure 3.1: The Four PseudoWidgets: Button (top left), Lever (top right), Knob (bottom right), and Joystick (bottom left). Transparent green region represents PseudoWidgets module placement.

that can effectively press on force sensors. When exerting a force on the shaft, the shaft will slightly displace allowing the prong in the direction of the applied force to hit a force sensor.

## 3.2 Software-induced Dynamic Properties

There is a significant lack of haptic interfaces with multi-modal properties. Using the widgets discussed, we can mount sensor modules in designated areas of the widget to simulate moving parts that are present in other multi-modal haptic interfaces. Since all simulation is thanks to software, we can ignore any moving parts and solely use adjustable parameters to offer multi-modality in rigid widgets. For example, a rigid button can be clickable until a certain event occurs in a program which forces it to disable. Software parameters can be sent to disable the button without the hassle of replacing or suspending actual parts. Beyond enabling and disabling widgets, the software parameters can also control the frequency of vibrations which allows rigid bodies to express moving parts of various stiffnesses. To make all of this happen, we need firmware running in the modules that can monitor for force readings and trigger grain vibrations in the LRA. Since the force sensor values are just analog values from a micro-controller and we want to implement a force-displacement model, we use a quadratic formula to map those values to a linear force-displacement curve. The curve we generated has an R-squared coefficient of 0.9787. Processing sensor values into actual force exerted provides the necessary information needed to determine the number of grain vibrations to produce. During application use, the firmware inside each sensor module used will continuously read in force sensor input and convert it into a force value.



## Chapter 4

# PseudoWidgets Implementation

In this section, we provide detailed descriptions for the implementation and design decisions of each tool that make up PseudoWidgets: the designer tool, the WiFi sensor module management tool, and the supplied application programming interface.

### 4.1 PseudoWidgets Designer

Since we want to maximize the usability of PseudoWidgets, we incorporate existing services that both industry and personal projects commonly use. For the designer tool, instead of using a standalone application we opted for a Fusion 360 add-on. Autodesk's Fusion 360 is a popular CAD design tool for creating products and provides an extensive API for adding new features through add-on programs. Many product designers already have some familiarity with the tool which would significantly decrease the amount of time needed to instruct people on how to use our design tool. After completing a design, Fusion 360 offers a feature to easily export designs to slicing programs such as Cura for 3D printing. All of the designs generated by PseudoWidgets are composed of PLA material which is commonly used in commercial 3D printers. Another key consideration for using Fusion 360 is because of its cloud sharing capabilities that allow groups of designers to work collaboratively online. In an ideal group project scenario, each group member can simply load the

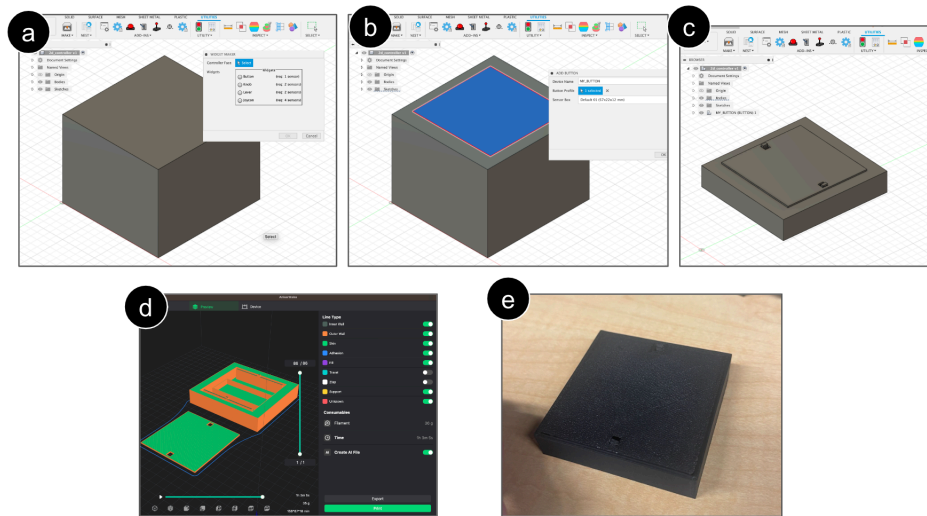


Figure 4.1: (a) Creating a widget outline (b) Selecting widget outline for generation (c) Generated widget output (d) Widget pieces separated and sent to AnkerMake slicing software (e) Final printed product

PseudoWidgets’s design add-on to their local machine and work together to design haptic-enabled devices.

To enable the add-on, the user must insert the design add-on project folder under the designated Fusion 360 add-ons folder. Upon running the add-on through the Fusion 360 interface, a set of buttons appear on the user interface that represent the actions a user can perform to create their controllers. There are five buttons: one for sketch generation and four for each individual widget. The sketch generation button is for designating a region for creating a widget. The user will click this button first and specify what type of widget to use and the plane or face of the controller he/she would like to place the widget at. After answering these two questions through a prompt that appears, a sketch of fixed dimension will appear on the specified surface. Using a mouse, the user can drag the generated sketch around on the selected face to specify the precise location of the new widget. After determining the location of

the sketch region, the user then proceeds with the actual generation of their widget in the sketch region using a widget button associated with their sketch. The user clicks the appropriate widget button and a second prompt pops up asking for the specified sketch region to generate at. After completing the second prompt, the add-on generates the design.

The generation of a rigid widget comprises many known Fusion 360 actions into one compact script and relies on the dimensions of the sensor module to make appropriate cuts into the sketch region of interest. A visualization of this process can be seen in Figure 4.1. We separate the discussion of generating a button and generating the other three widgets that rely on an additional shaft piece.

#### 4.1.1 Generating a Button Widget

The button widget requires one sensor module that is centrally placed. When a user applies force to a plate resting above it, the plate displaces and presses on the force sensor resting under the point of contact (see Fig. 3.1). Generating a button component first takes a sketch region and cuts into the controller by the height of the sensor module (12 mm). At the bottom of the cut, three wall structures are extruded: one thin-walled rectangle outline for easy placement of the sensor module and two walls adorned with snap joints at the top for easy placement of the widget plate. The widget plate generates after the wall structures are finished, this plate is designed to cover the hole that was created. The button's widget plate is 1 mm in height allowing for sufficient flexibility for plate deformation. Two holes are cut through the plate itself allowing the plate to snap on to the controller with the snap joints. The snap joints are shared across all widget types and are used to keep the plates attached.

### 4.1.2 Generating a Lever, Knob, or Joystick Widget

Although the remaining widgets do not share the same fixed sketch region, they each share the same generation procedure. Same as the button, we first cut into the controller shell and create snap joint walls and a widget plate but with two main distinctions. Firstly, since the forces applied to these widgets are parallel to the surface, we have to extrude by the length of the sensor module (57 mm). Secondly, the widget plate is no longer the source of user input so a hole is cut at the center of the plate to make room for a shaft component that users interact with instead. The shaft is designed to be attachable using a ball-and-socket joint and relies on prongs on its side to supply force to sensors. Right after extruding the snap joint walls, a socket hole for the shaft is created at the center. The shaft used for each of the three widgets vary (see Fig. 3.1). The lever needs two prongs for pressing on the two modules positioned on opposite sides for lateral translation. The knob also uses two modules but both share the same side and separated by a small spacer, the shaft has two longer prongs to reach each module's FSRs when being turned by a user. Lastly, the joystick has the same behavior as a lever but uses four modules for two-dimensional translation.

## 4.2 PseudoWidgets Connector

The next step after designing a controller and sending it off to print is configuring WiFi sensor modules provided by PseudoWidgets. The number of sensor modules depends on the number of widgets used and how many each widget needs to function properly. Running PseudoWidget's Connector (PWC) program on the same network allows designers to register sensor modules with specific widgets, configure their vir-

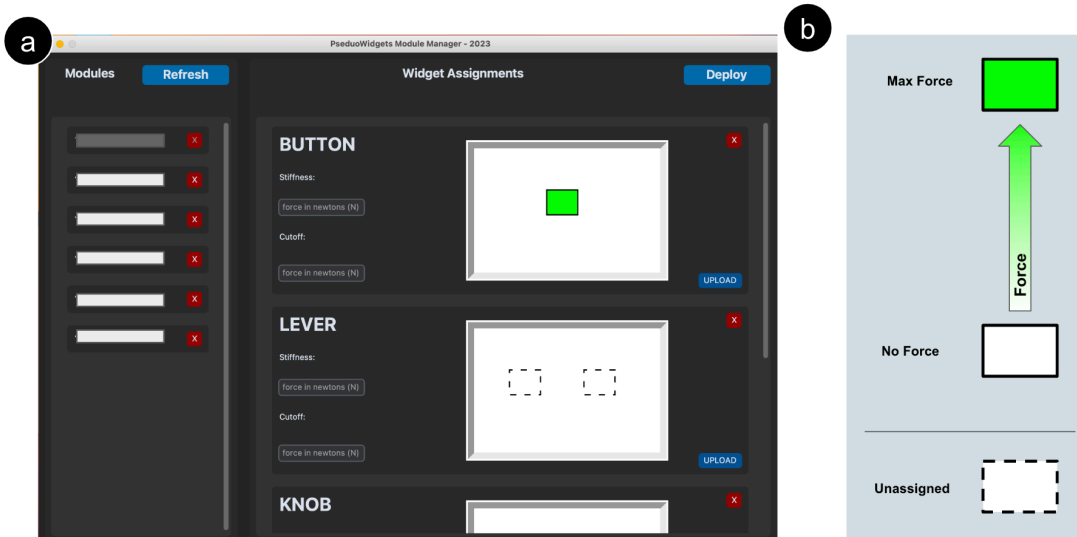


Figure 4.2: (a) PseudoWidgets Connector interface with one sensor module assigned to a Button widget. (b) Diagram of force visualization via gradual color change.

tual parameters, and provide force visualization. PWC establishes a UDP server on a provided WiFi network, allowing sensor modules to connect. Upon module connection, the module persistently informs PWC of its current force, stiffness, and cutoff values. The PseudoWidget’s Connector program is a multi-threaded Python application that utilizes customtkiner, a Tkinter extension that gives the application a modern appearance.

PWC’s interface provides two main panels for interaction. The left panel currently displays a list of sensor modules that are connected to the UDP server. Upon starting PWC, the list first appears empty until the designer hits refresh to check if any sensor modules made contact with the server.

The right panel is for spatially mapping sensor modules to different widgets and visualizing their force readings. Each widget entry contains empty slots (appears as dashed outline boxes) for assigning sensor modules and field inputs that allow

designers to setup virtual parameters for all of the widget's modules. Assigning a sensor module is a two-step process. First the designer selects the sensor module on the left pane. Then, the designer clicks on a empty slot for filling in with the sensor module selection. Filling in an empty slot with a module turns the box white with a solid black outline. If a designer can not recall where a module's assignment is, they can simply press on the module's force sensor and the corresponding slot will respond back by gradually changing color based on force applied. This feature is also important for designers who need a sense of how strong a force is in a widget or controller design.

After a designer finishes configuring sensor modules and widgets, the designer is ready to deploy a database that application developers can interface with over the server. Clicking the "Deploy" button on the top right-hand side fills creates a dictionary data structure where each widget is represented by a key and value. The key is the name of the widget and the values are the modules associated with it.

### **4.3 Application Programming Interface**

For adding dynamic states we provide an Application Programming Interface (API) written in Python, enabling application developers to communicate and change virtual parameters over PWC's server. The UDP server on PseudoWidget's Connector end interprets string queries sent by both modules and application developers. Such string commands are abstracted for app developers keeping requests simple to call. To begin using the package, the app developer would instantiate an object from the API's PWManager class, just like any normal class instance in object-oriented programming. This class handles all requests made by the application developer such

```

# import package line
from pseudo_widgets_api import PWManager

# Manager object
manager = PWManager('Connector IP address', port=3333)

# Use widget name to retrieve assigned sensor module IPs
button_ip = manager.get_device_modules("BUTTON")[0]
lever_ips = manager.get_device_modules("LEVER")

# GET example
manager.get_force(lever_ips[0])

# SET example
manager.set_stiffness(button_ip, 0.8)

# DISABLE/ENABLE WIDGET example
manager.disable_widget("BUTTON")
manager.enable_widget("BUTTON", 1, 30)

```

Figure 4.3: Example Python code instantiating a PWManager object and invoking some supported commands.

as accessing widget information, disabling or enabling a widget, and getting/setting virtual parameters for modules (Figure 4.3). Developers can utilize these commands in their code logic to create dynamic states for their applications. One example is using a PseudoWidgets button with a countdown application, where each press of the button with virtual stiffness on decrements an integer by 1 in Python code. The developer's code will listen for force changes using the appropriate get command with a PWManager object. When the countdown reaches zero, a condition is checked and the developer's program calls to disable the entire widget making the button feel rigid. This is a trivial example but offers some insight on the wide range of applications that could benefit from PseudoWidget's API.

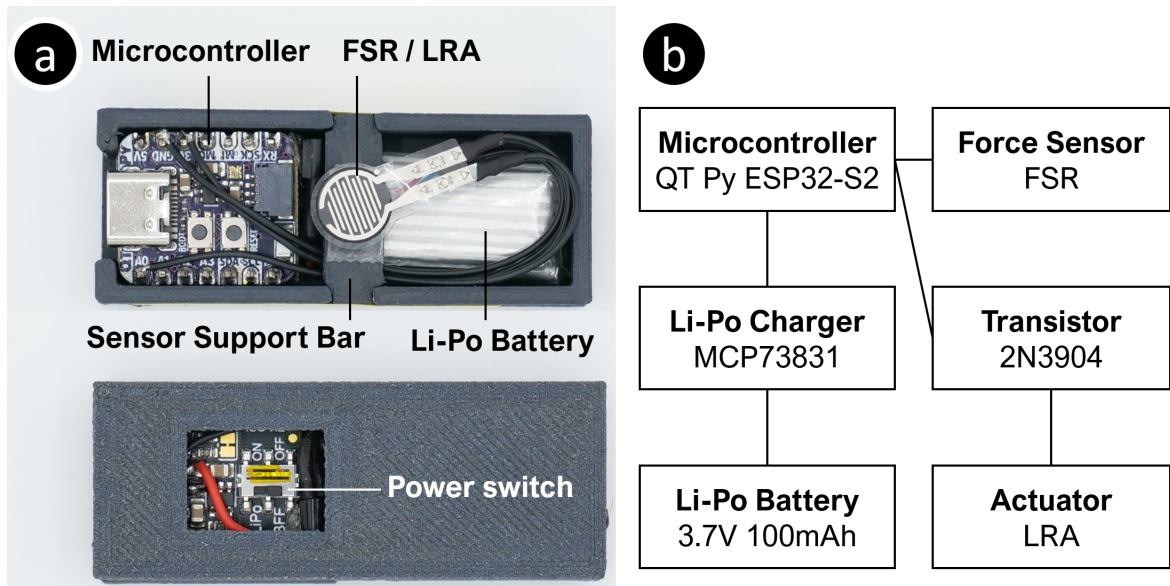


Figure 4.4: PseudoWidget Module. (a) Top and bottom views of the device, (b) electronic components in the module.

## 4.4 PseudoWidgets Module

PseudoWidgets module is designed to be a plug-in component that can be inserted into 3D-printed parts to enable dynamic haptic interactions by creating the compliance illusion. Therefore, we developed PseudoWidgets module to be small, lightweight, and self-contained without the need for wiring to connect to a computer (See Fig. 4.4). The module is  $57 \times 22 \times 12\text{mm}$  ( $W \times H \times D$ ), and weighs 15.4 grams. It uses a QT Py ESP32-S2 module to process the force measurement, render grain vibrations, and communicate with a PC via Wi-Fi. The device is powered by a 3.7V 100mAh lithium polymer (Li-Po) battery, and an Adafruit Li-Po battery charger add-on with MCP73831 Li-Po charge management controller was attached to the QT Py ESP32-S2 module to allow the module to be charged through the USB-C connection. The module has a window on the bottom for easy access to the power switch. Based on our measurement, the device consumes approximately 350 mWh and will last for



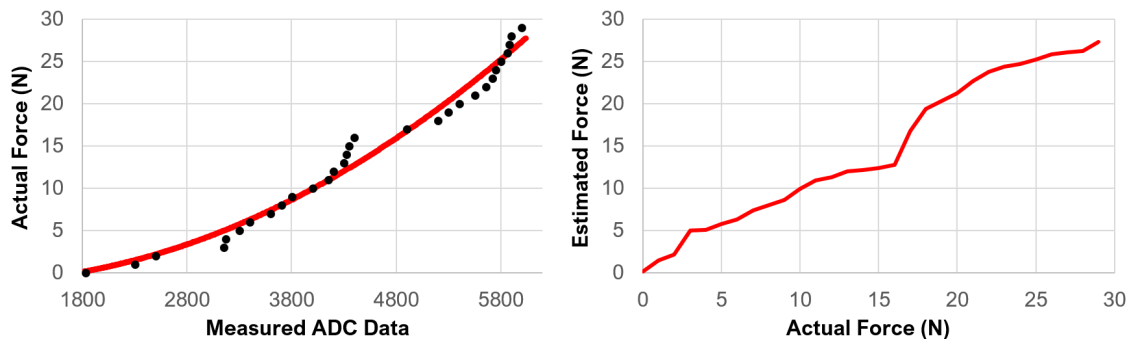


Figure 4.5: Sensor measurement by force, and the estimated force after calibration using quadratic curve fitting.

around an hour on a 3.7V 100mAh battery.

The device uses a Force-Sensing Resistor (FSR) with an 8-mm diameter sensing area and a minimum force detection value of 0.2 N. Using a voltage divider with a 2.2 resistor connected in series and the ESP32 Analog-Digital Converter (ADC), the force is converted to the measurement at 13-bit resolution. To ensure the linear force readings, we calibrated the force sensors by fitting the measurements using a quadratic curve fitting. Fig. 4.5 shows the result of the force calibration. The haptic feedback generation uses a simplified version of the PseudoBend algorithm Heo, J. Lee, and Wigdor 2019. The device measures and processes the force at 1000 Hz, and if the difference between the measured force and the force when the last grain vibration was played is greater than the force threshold (stiffness parameter), the device will render a grain vibration and store the measured force. If the device did not play any grain vibrations, then the last grain vibration force will be 0. To generate the grain vibration, we use a 10ms-long digital pulse. The pulse generated by the digital I/O of the ESP32 device will be sent to a 2N3904 transistor to drive a Linear Resonant Actuator (LRA, Jinlong Machinery G0832012, diameter: 8mm, thickness: 3.2mm). The FSR and the LRA are attached to each other to minimize the footprint, and

our pilot tests found that the vibration from the LRA does not cause significant disturbance to the force measurement.

# Chapter 5

## Evaluation

We conducted a preliminary study to gain initial insights into the user experience of using tangible user interfaces created using PseudoWidgets. Since the toolkit's main goal is to bring new immersive experiences to users, it is important that products made with PseudoWidgets are effective in supplying vibrotactile feedback. The study protocol was approved by the Institutional Review Board of [anonymized institution]. Three participants were recruited via word-of-mouth. The average age of participants was 21.33 years old (20-24, 3 males). The study took around 30 minutes to complete, and the participants received a \$10 gift card for their participation. The participant first interacted with each of the four standard widgets (button, lever, knob, and joystick) and then used tangible user interfaces developed using PseudoWidgets in two applications: a tangible music player and a video game - Haptic Space. The applications were implemented in Python using pygame package, and they used the PseudoWidgets API to use the tangible user interfaces. The participants were asked to fill out a questionnaire about their experience of using the interface after trying each application.

## 5.1 Individual Widgets

Before using the two applications, the participants were asked to try the four basic widgets: a button, a lever, a knob, and a joystick, to obtain feedback from the participants regarding the experience of using each widget. During this process, the researcher changed the stiffness of the widgets and enabled and disabled the illusion, and asked the participants if they could discern behavior changes. For button, lever, and joystick widgets, all participants reported that they could feel when the device starts or stops moving and perceive the differences in stiffness. However, two of the three participants mentioned that the differences were not very clear when using the knob widget.

## 5.2 Application-specific Controllers

After handling each individual widget and filling out the first part of the survey, participants played a video game and music player applications by using their associated controller. To see if participants favored using the haptic technique in controllers, a second version of each application was created for participants to play after using the haptic-enabled version. The second version mutes all haptic properties, making all of the widgets on the controller rigid. After playing with each application for roughly five minutes, the participants filled out a survey to provide feedback.

### 5.2.1 Video Game

In the Haptic Space video game, participants controlled a spaceship and shoot down asteroids in a maze, using a tangible game controller that has Joystick and Button

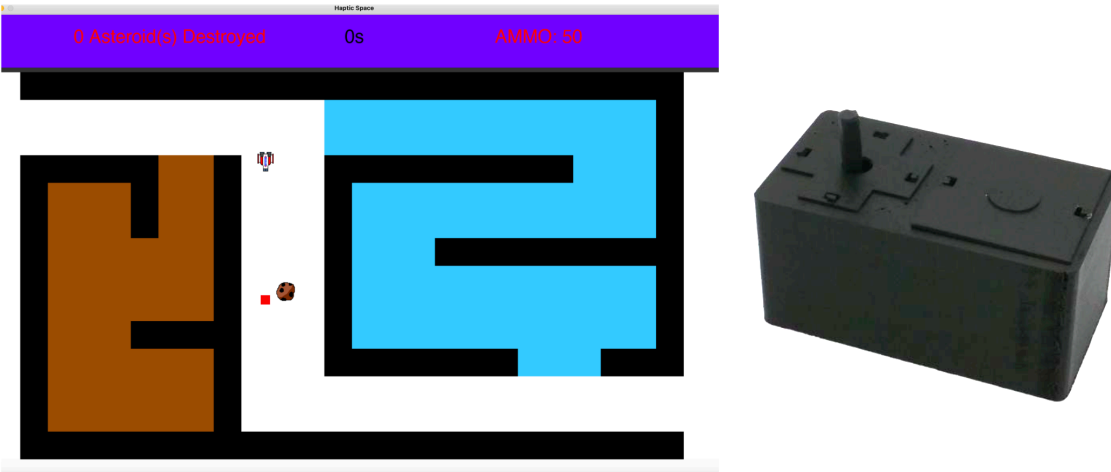


Figure 5.1: Haptic Space game and PseudoWidget controller

widgets developed using PsuedoWidgets. The video game dynamically changed the stiffness of the joystick based on the game context. In the game space (see Figure 5.1), the air density differed by different areas. The brown-colored area had a more dense air and causes more friction, and the joystick became stiffer to manipulate. On the other hand, the blue-colored area had less friction and the joystick had less stiffness. The button behavior was also dynamically controlled; the button became disabled when the spaceship ran out of ammo. Participants found that the Button on the controller made combat more realistic when playing the game. Participant C said "the Button haptic feedback felt useful". Participant B appreciated the haptic-enabled version of the video game compared to the other, saying "the haptic feedback definitely made the user experience more realistic feeling and more interesting".

### 5.2.2 Music Player

The music player application allows users to interact with a small music playlist in a more tangible way by using a controller containing a button, lever, and rotating knob.



Figure 5.2: Music player app and PseudoWidget controller using Button on the front side and a lever and a knob on the top

The music player user interface shows four song blocks. The songs could be selected using a cursor controlled by the lever, and pressing the front button started playing the music. The button was deactivated when the cursor was not on the song block. The knob was used to control the music volume level, which stopped rotating when the volume reached the maximum or minimum level. After using the application, all participants reported that they liked the haptic feedback, and navigating the music and controlling the music volume felt easier with the illusive movement. One participant mentioned: "the feedback from the knob was weaker than the others, but it was also the most noticeable missing when the haptics were turned off", and another participant mentioned that "the music player controller without haptic feedback felt limiting".

# Chapter 6

## Discussion

PseudoWidgets can effectively simplify the tangible interface prototyping process. Using the design tool, the researchers could design and implement dynamic tangible interfaces in a short amount of time. The tangible user interfaces created using PseudoWidgets could also effectively manipulate the perceived movement of the widgets based on the application context that is distinguishable by the participants. We believe that this toolkit will allow users without extensive hardware prototyping experience to build functional prototypes using 3D printers. The design of the PseudoWidgets module will become opensource so that it can be produced by 3rd parties and shared or sold to others.

### 6.1 Expandability of PseudoWidgets

The sample applications created with pygame are just a fraction of where PseudoWidgets can be employed. Mobile games use flat touchscreen controls that lack tangible interactions and eyes-free control. PseudoWidgets such as joysticks and buttons can replace these and could be attached to a smartphone or tablet device. Virtual and augmented reality environments are now commonly used in workspaces and personal use that allow users to immerse themselves in a digital environment to perform operations. However, virtual reality and augmented reality are intolerant to tangible

interactions and are only capable of supplying visual and audio information. Actions such as clicking a button or a switch in the virtual menu offer no haptic feedback. PseudoWidgets can be used as attachments to existing VR controllers or can represent virtual objects in a physical setting. One specific example where PseudoWidgets has a purpose is improving the tangible interactions with virtual menus such as a settings menu. The virtual buttons and switches on a menu can be represented with real PseudoWidgets that are spatially arranged or attached to the body of a user. Another area where PseudoWidgets could benefit is in remote-controlled devices. Flying a remote-controlled drone, for example, sometimes comes with the challenge of flying through windy conditions, making it difficult to fully control the device. To aid balancing in weather conditions, using PseudoWidgets with dynamic stiffness changing could be useful for providing a user an idea of the weather conditions occurring in the air from the ground. Toys are also another area where PseudoWidgets can be incorporated. One popular toy such as the Bop-it is adorned with different physical widgets such as a twisting knob, button, and lever. A substitute controller can be created with PseudoWidgets where we can program the stiffness, cutoff and dynamic behaviors such as how far the knob can be twisted.

## 6.2 Limitations

PseudoWidgets does a great job at simplifying the design of tangible interfaces but there still remain some drawbacks. The first drawback concerns the current size of the widgets that PseudoWidgets support. Although we can create hand-held controllers, the controllers are slightly bigger compared to other physical controllers used today. This limitation is present as the size of the widgets and their internal structures are



dependent on the dimensions of the sensor modules we use. To reduce the size of widgets, we would need to figure out how to scale down the sensor module which has hardware limitations. Controllers made by PseudoWidgets are also limited to short play times due to the sensor modules short battery life. When a sensor module is not charging and in active use, it can stay on for 1 hour which is not favorable. One workaround is just keeping the sensor module wired and making holes in the controllers. Other issues come from the limitations of using an LRA for vibration grains. The vibrations in a 3D-printed device are noisy and tend to permeate across entire controllers. Both of these limitations may deter the immersive experience when interacting with tangible interfaces. One idea used by Haptic Enchanters (Park, Cha, and Choi 2019) is using vibration attenuation layers or dampening mechanisms to contain vibrations in one specific area. Also, the vibrations from the LRA can be noisy, which may deter the immersive experiences of interacting with the widgets. This is one consideration but we also want to preserve PseudoWidgets minimal assembly advantages.

# Chapter 7

## Future Work

We want to extend PseudoWidgets to other applications outside of simple Python games. One specific area for exploration is implementing dynamic tangible interfaces for virtual reality applications. Common tools such as Unity are utilized for creating such applications and require the use of C-Sharp programming. We plan to expand the current API to support the C-Sharp language, allowing Unity developers to easily include dynamic tangible interfaces into their virtual or augmented reality applications. We also intend to conduct a more extensive study on both end-user applications and the usability of the toolkit. Although our findings from three participants show positive end-user experiences surrounding the different controllers, more participants are needed to validate our findings. For usability, we seek to recruit other participants in the future who have knowledge of programming and 3D Cad design to see how easy it is to create PseudoWidget controllers. We also intend to address some of the constructive feedback from the preliminary study to make improvements to the widgets supported by the Designer tool. The knob widget performed the lowest out of the four widgets where some of the participants noted that the widget seemed weaker in use. However, they also mentioned it was the most useful widget in the haptic-enabled version of the music player versus the non-haptic-enabled version when adjusting the volume of the music. This information convinces us that improving the knob widget will improve the tangible experiences of twisting a rigid device. We also

need to quantify how well vibrations travel in a controller when a user presses on each of the widgets.

# Chapter 8

## Conclusion

Designing tangible and haptic interfaces brings several challenges for designers who lack the interdisciplinary experience needed to craft mechanisms, electrical components, and firmware on embedded devices. Many haptic interfaces are restricted to one form of interaction, making them difficult to use for multiple contexts. PseudoWidgets toolkit tackles both of these problems by allowing rapid creation and dynamic programming of tangible interfaces that simulate moving parts. This simulation of moving parts is performed by changing virtual stiffness parameters, known as material compliance illusion. PseudoWidgets toolkit offers a Designer tool for creating and slicing 3D widgets, a Connector tool for configuring sensor modules over a UDP network and assigning to widgets, and an application programming interface for developers to interface with registered sensors. In a user study, we used PseudoWidgets to create two different applications and controllers to use in an end user study to gain qualitative feedback. Two versions of each application are created for comparing haptic and non-haptic scenarios. Participants found all of the widgets to be effective in supplying the illusion of movement and all but the rotating knob for discerning behavioral or dynamic changes. All participants also recognize the usefulness of the haptics used in the controllers compared to controller interactions without haptic feedback. Finally, we intend to expand our toolkit's capabilities to virtual and augmented reality applications but this first requires extending the API support. We

hope that this work promotes further research into creating dynamic haptic interfaces and toolkits for democratizing haptic prototyping.

# Bibliography

- Fitzmaurice, George W., Hiroshi Ishii, and William A. S. Buxton (1995). “Bricks: laying the foundations for graspable user interfaces”. en. In: *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95*. Denver, Colorado, United States: ACM Press, pp. 442–449. ISBN: 978-0-201-84705-5. DOI: [10.1145/223904.223964](https://doi.org/10.1145/223904.223964). URL: <http://portal.acm.org/citation.cfm?doid=223904.223964> (visited on 04/05/2023).
- Kirk, David et al. (Sept. 2009). “Putting the physical into the digital: Issues in designing hybrid interactive surfaces”. en-US. In: URL: <https://www.microsoft.com/en-us/research/publication/putting-the-physical-into-the-digitalissues-in-designing-hybrid-interactive-surfaces/> (visited on 04/05/2023).
- Marquardt, Nicolai et al. (Nov. 2009). “The Haptic Tabletop Puck: tactile feedback for interactive tabletops”. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ITS '09. New York, NY, USA: Association for Computing Machinery, pp. 85–92. ISBN: 978-1-60558-733-2. DOI: [10.1145/1731903.1731922](https://doi.org/10.1145/1731903.1731922). URL: <https://dl.acm.org/doi/10.1145/1731903.1731922> (visited on 04/05/2023).
- Taylor, Stuart et al. (Apr. 2009). “Turning the tables: an interactive surface for vjing”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. New York, NY, USA: Association for Computing Machinery, pp. 1251–1254. ISBN: 978-1-60558-246-7. DOI: [10.1145/1518701.1518888](https://doi.org/10.1145/1518701.1518888). URL: <https://dl.acm.org/doi/10.1145/1518701.1518888> (visited on 04/04/2023).

- Visell, Yon, Alvin Law, and Jeremy R. Cooperstock (July 2009). “Touch Is Everywhere: Floor Surfaces as Ambient Haptic Interfaces”. In: *IEEE Transactions on Haptics* 2.3. Conference Name: IEEE Transactions on Haptics, pp. 148–159. ISSN: 2329-4051. DOI: [10.1109/TOH.2009.31](https://doi.org/10.1109/TOH.2009.31).
- Kildal, Johan (Nov. 2010). “3D-press: haptic illusion of compliance when pressing on a rigid surface”. en. In: *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*. Beijing China: ACM, pp. 1–8. ISBN: 978-1-4503-0414-6. DOI: [10.1145/1891903.1891931](https://doi.org/10.1145/1891903.1891931). URL: <https://dl.acm.org/doi/10.1145/1891903.1891931> (visited on 01/23/2023).
- Holman, David and Hrvoje Benko (2011). “SketchSpace: designing interactive behaviors with passive materials”. en. In: *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA '11*. Vancouver, BC, Canada: ACM Press, p. 1987. ISBN: 978-1-4503-0268-5. DOI: [10.1145/1979742.1979867](https://doi.org/10.1145/1979742.1979867). URL: <http://portal.acm.org/citation.cfm?doid=1979742.1979867> (visited on 10/26/2022).
- Ledo, David et al. (Feb. 2012). “The HapticTouch toolkit: enabling exploration of haptic interactions”. In: *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*. TEI '12. New York, NY, USA: Association for Computing Machinery, pp. 115–122. ISBN: 978-1-4503-1174-8. DOI: [10.1145/2148131.2148157](https://doi.org/10.1145/2148131.2148157). URL: <https://doi.org/10.1145/2148131.2148157> (visited on 02/20/2023).
- Follmer, Sean et al. (Oct. 2013). “inFORM: dynamic physical affordances and constraints through shape and object actuation”. In: *Proceedings of the 26th annual ACM symposium on User interface software and technology*. UIST '13. New York, NY, USA: Association for Computing Machinery, pp. 417–426. ISBN: 978-1-4503-

- 2268-3. DOI: [10.1145/2501988.2502032](https://doi.org/10.1145/2501988.2502032). URL: <https://doi.org/10.1145/2501988.2502032> (visited on 03/18/2023).
- Savage, Valkyrie, Colin Chang, and Björn Hartmann (Oct. 2013). “Sauron: embedded single-camera sensing of printed physical user interfaces”. In: *Proceedings of the 26th annual ACM symposium on User interface software and technology*. UIST '13. New York, NY, USA: Association for Computing Machinery, pp. 447–456. ISBN: 978-1-4503-2268-3. DOI: [10.1145/2501988.2501992](https://doi.org/10.1145/2501988.2501992). URL: <https://doi.org/10.1145/2501988.2501992> (visited on 10/20/2022).
- Benko, Hrvoje et al. (Oct. 2016). “NormalTouch and TextureTouch: High-fidelity 3D Haptic Shape Rendering on Handheld Virtual Reality Controllers”. In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST '16. New York, NY, USA: Association for Computing Machinery, pp. 717–728. ISBN: 978-1-4503-4189-9. DOI: [10.1145/2984511.2984526](https://dl.acm.org/doi/10.1145/2984511.2984526). URL: <https://dl.acm.org/doi/10.1145/2984511.2984526> (visited on 04/04/2023).
- Ramakers, Raf et al. (May 2016). “RetroFab: A Design Tool for Retrofitting Physical Interfaces using Actuators, Sensors and 3D Printing”. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. New York, NY, USA: Association for Computing Machinery, pp. 409–419. ISBN: 978-1-4503-3362-7. DOI: [10.1145/2858036.2858485](https://doi.org/10.1145/2858036.2858485). URL: <https://doi.org/10.1145/2858036.2858485> (visited on 09/21/2022).
- Robinson, Simon et al. (May 2016). “Emergeables: Deformable Displays for Continuous Eyes-Free Mobile Interaction”. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. New York, NY, USA: Association for Computing Machinery, pp. 3793–3805. ISBN: 978-1-4503-3362-7. DOI: [10.1145/2858036.2858097](https://doi.org/10.1145/2858036.2858097). URL: <https://doi.org/10.1145/2858036.2858097> (visited on 10/26/2022).



- Jung, Jingun, Eunhye Youn, and Geehyuk Lee (May 2017). “PinPad: Touchpad Interaction with Fast and High-Resolution Tactile Output”. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. New York, NY, USA: Association for Computing Machinery, pp. 2416–2425. ISBN: 978-1-4503-4655-9. DOI: [10.1145/3025453.3025971](https://doi.org/10.1145/3025453.3025971). URL: <https://doi.org/10.1145/3025453.3025971> (visited on 10/26/2022).
- Shibasaki, Mina, Keitaro Tuchiya, and Kouta Minamizawa (Mar. 2017). “TECHTILE Bits: Designing a Tool for Children to Create Haptic Devices”. In: *Proceedings of the Virtual Reality International Conference - Laval Virtual 2017*. VRIC '17. New York, NY, USA: Association for Computing Machinery, pp. 1–4. ISBN: 978-1-4503-4858-4. DOI: [10.1145/3110292.3110308](https://doi.org/10.1145/3110292.3110308). URL: <https://doi.org/10.1145/3110292.3110308> (visited on 02/24/2023).
- Siu, Alexa F. et al. (Apr. 2018). “shapeShift: 2D Spatial Manipulation and Self-Actuation of Tabletop Shape Displays for Tangible and Haptic Interaction”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. New York, NY, USA: Association for Computing Machinery, pp. 1–13. ISBN: 978-1-4503-5620-6. DOI: [10.1145/3173574.3173865](https://doi.org/10.1145/3173574.3173865). URL: <https://dl.acm.org/doi/10.1145/3173574.3173865> (visited on 04/05/2023).
- Ghosal, Radhika et al. (Oct. 2019). “Rapid Prototyping of Pneumatically Actuated Inflatable Structures”. In: *Adjunct Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19 Adjunct. New York, NY, USA: Association for Computing Machinery, pp. 78–80. ISBN: 978-1-4503-6817-9. DOI: [10.1145/3332167.3357121](https://doi.org/10.1145/3332167.3357121). URL: <https://doi.org/10.1145/3332167.3357121> (visited on 03/20/2023).
- Groeger, Daniel et al. (Oct. 2019). “Tactlets: Adding Tactile Feedback to 3D Objects Using Custom Printed Controls”. In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19 Adjunct. New York, NY, USA: Association for Computing Machinery, pp. 1–4. ISBN: 978-1-4503-6817-9. DOI: [10.1145/3332167.3357121](https://doi.org/10.1145/3332167.3357121). URL: <https://doi.org/10.1145/3332167.3357121> (visited on 03/20/2023).

- sium on User Interface Software and Technology*. UIST '19. New York, NY, USA: Association for Computing Machinery, pp. 923–936. ISBN: 978-1-4503-6816-2. DOI: [10.1145/3332165.3347937](https://doi.org/10.1145/3332165.3347937). URL: <https://doi.org/10.1145/3332165.3347937> (visited on 10/20/2022).
- Heo, Seongkook, Jaeyeon Lee, and Daniel Wigdor (Oct. 2019). “PseudoBend: Producing Haptic Illusions of Stretching, Bending, and Twisting Using Grain Vibrations”. In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19. New York, NY, USA: Association for Computing Machinery, pp. 803–813. ISBN: 978-1-4503-6816-2. DOI: [10.1145/3332165.3347941](https://doi.org/10.1145/3332165.3347941). URL: <https://doi.org/10.1145/3332165.3347941> (visited on 09/21/2022).
- Israr, Ali et al. (2019). “Stereohaptics Toolkit for Dynamic Tactile Experiences”. en. In: *HCI International 2019 – Late Breaking Papers*. Ed. by Constantine Stephanidis. Vol. 11786. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 217–232. ISBN: 978-3-030-30032-6 978-3-030-30033-3. DOI: [10.1007/978-3-030-30033-3\\_17](https://doi.org/10.1007/978-3-030-30033-3_17). URL: [http://link.springer.com/10.1007/978-3-030-30033-3\\_17](http://link.springer.com/10.1007/978-3-030-30033-3_17) (visited on 02/22/2023).
- Nakagaki, Ken et al. (Mar. 2019). “inFORCE: Bi-directional ‘Force’ Shape Display for Haptic Interaction”. In: *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI '19. New York, NY, USA: Association for Computing Machinery, pp. 615–623. ISBN: 978-1-4503-6196-5. DOI: [10.1145/3294109.3295621](https://doi.org/10.1145/3294109.3295621). URL: <https://dl.acm.org/doi/10.1145/3294109.3295621> (visited on 04/04/2023).
- Nakao, Takuro et al. (July 2019). “ShareHaptics: a modular haptic feedback system using shape memory alloy for mixed reality shared space applications”. In: *ACM SIGGRAPH 2019 Posters*. SIGGRAPH '19. New York, NY, USA: Association for Computing Machinery, pp. 1–2. ISBN: 978-1-4503-6314-3. DOI: [10.1145/3294109.3295621](https://doi.org/10.1145/3294109.3295621)

3306214.3338597. URL: <https://doi.org/10.1145/3306214.3338597> (visited on 02/24/2023).

Park, Gunhyuk, Hojun Cha, and Seungmoon Choi (Jan. 2019). “Haptic Enchanters: Attachable and Detachable Vibrotactile Modules and Their Advantages”. In: *IEEE Transactions on Haptics* 12.1. Conference Name: IEEE Transactions on Haptics, pp. 43–55. ISSN: 2329-4051. DOI: [10.1109/TOH.2018.2859955](https://doi.org/10.1109/TOH.2018.2859955).

Shim, Youngbo Aram et al. (Oct. 2020). “FS-Pad: Video Game Interactions Using Force Feedback Gamepad”. In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST ’20. New York, NY, USA: Association for Computing Machinery, pp. 938–950. ISBN: 978-1-4503-7514-6. DOI: [10.1145/3379337.3415850](https://doi.org/10.1145/3379337.3415850). URL: <https://doi.org/10.1145/3379337.3415850> (visited on 10/20/2022).

Strohmeier, Paul et al. (Oct. 2020). “bARefoot: Generating Virtual Materials using Motion Coupled Vibration in Shoes”. en. In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. Virtual Event USA: ACM, pp. 579–593. ISBN: 978-1-4503-7514-6. DOI: [10.1145/3379337.3415828](https://doi.org/10.1145/3379337.3415828). URL: <https://dl.acm.org/doi/10.1145/3379337.3415828> (visited on 03/11/2023).

Suzuki, Ryo, Hooman Hedayati, et al. (Apr. 2020). “RoomShift: Room-scale Dynamic Haptics for VR with Furniture-moving Swarm Robots”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI ’20. New York, NY, USA: Association for Computing Machinery, pp. 1–11. ISBN: 978-1-4503-6708-0. DOI: [10.1145/3313831.3376523](https://doi.org/10.1145/3313831.3376523). URL: <https://dl.acm.org/doi/10.1145/3313831.3376523> (visited on 04/05/2023).

Suzuki, Ryo, Ryosuke Nakayama, et al. (Feb. 2020). “LiftTiles: Constructive Building Blocks for Prototyping Room-scale Shape-changing Interfaces”. In: *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied*

- Interaction*. TEI '20. New York, NY, USA: Association for Computing Machinery, pp. 143–151. ISBN: 978-1-4503-6107-1. DOI: [10.1145/3374920.3374941](https://doi.org/10.1145/3374920.3374941). URL: <https://doi.org/10.1145/3374920.3374941> (visited on 03/20/2023).
- Yasu, Kentaro (Apr. 2020). “MagneLayer: Force Field Fabrication by Layered Magnetic Sheets”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. New York, NY, USA: Association for Computing Machinery, pp. 1–9. ISBN: 978-1-4503-6708-0. DOI: [10.1145/3313831.3376552](https://doi.org/10.1145/3313831.3376552). URL: <https://doi.org/10.1145/3313831.3376552> (visited on 02/24/2023).
- Cruz, Ramón E. Sánchez et al. (July 2021). “Modular Haptic Feedback for Rapid Prototyping of Tactile Displays”. In: *2021 IEEE World Haptics Conference (WHC)*, pp. 703–708. DOI: [10.1109/WHC49131.2021.9517256](https://doi.org/10.1109/WHC49131.2021.9517256).
- Dementyev, Artem et al. (Oct. 2021). “VHP: Vibrotactile Haptics Platform for On-body Applications”. In: *The 34th Annual ACM Symposium on User Interface Software and Technology*. UIST '21. New York, NY, USA: Association for Computing Machinery, pp. 598–612. ISBN: 978-1-4503-8635-7. DOI: [10.1145/3472749.3474772](https://doi.org/10.1145/3472749.3474772). URL: <https://doi.org/10.1145/3472749.3474772> (visited on 02/27/2023).
- Kim, Jeeun et al. (Feb. 2021). “OmniSoft: A Design Tool for Soft Objects by Example”. In: *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI '21. New York, NY, USA: Association for Computing Machinery, pp. 1–13. ISBN: 978-1-4503-8213-7. DOI: [10.1145/3430524.3440634](https://doi.org/10.1145/3430524.3440634). URL: <https://doi.org/10.1145/3430524.3440634> (visited on 03/17/2023).
- Nisser, Martin, Christina Chen Liao, et al. (May 2021). “LaserFactory: A Laser Cutter-based Electromechanical Assembly and Fabrication Platform to Make Functional Devices & Robots”. In: *Proceedings of the 2021 CHI Conference on Human*

*Factors in Computing Systems*. CHI '21. New York, NY, USA: Association for Computing Machinery, pp. 1–15. ISBN: 978-1-4503-8096-6. DOI: [10.1145/3411764.3445692](https://doi.org/10.1145/3411764.3445692). URL: <https://dl.acm.org/doi/10.1145/3411764.3445692> (visited on 04/05/2023).

Suzuki, Ryo, Eyal Ofek, et al. (Oct. 2021). “HapticBots: Distributed Encountered-type Haptics for VR with Multiple Shape-changing Mobile Robots”. In: *The 34th Annual ACM Symposium on User Interface Software and Technology*. UIST '21. New York, NY, USA: Association for Computing Machinery, pp. 1269–1281. ISBN: 978-1-4503-8635-7. DOI: [10.1145/3472749.3474821](https://doi.org/10.1145/3472749.3474821). URL: <https://doi.org/10.1145/3472749.3474821> (visited on 09/25/2022).

Zhang, Bowen and Misha Sra (Dec. 2021). “PneuMod: A Modular Haptic Device with Localized Pressure and Thermal Feedback”. In: *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*. VRST '21. New York, NY, USA: Association for Computing Machinery, pp. 1–7. ISBN: 978-1-4503-9092-7. DOI: [10.1145/3489849.3489857](https://doi.org/10.1145/3489849.3489857). URL: <https://doi.org/10.1145/3489849.3489857> (visited on 02/24/2023).

Lin, Hongnan et al. (Apr. 2022). “FlexHaptics: A Design Method for Passive Haptic Inputs Using Planar Compliant Structures”. In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New York, NY, USA: Association for Computing Machinery, pp. 1–13. ISBN: 978-1-4503-9157-3. DOI: [10.1145/3491102.3502113](https://doi.org/10.1145/3491102.3502113). URL: <https://doi.org/10.1145/3491102.3502113> (visited on 02/24/2023).

Messerschmidt, Moritz Alexander et al. (Jan. 2022). “ANISMA: A Prototyping Toolkit to Explore Haptic Skin Deformation Applications Using Shape-Memory Alloys”. In: *ACM Transactions on Computer-Human Interaction* 29.3, 19:1–19:34. ISSN:

- 1073-0516. DOI: [10.1145/3490497](https://doi.org/10.1145/3490497). URL: <https://doi.org/10.1145/3490497> (visited on 02/24/2023).
- Nisser, Martin, Yashaswini Makaram, et al. (Dec. 2022). “Demonstration of Mixels: Fabricating Interfaces using Programmable Magnetic Pixels”. In: *Adjunct Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. UIST '22 Adjunct. New York, NY, USA: Association for Computing Machinery, pp. 1–3. ISBN: 978-1-4503-9321-8. DOI: [10.1145/3526114.3558654](https://doi.org/10.1145/3526114.3558654). URL: <https://doi.org/10.1145/3526114.3558654> (visited on 02/24/2023).
- Yang, Humphrey et al. (Apr. 2022). “ReCompFig: Designing Dynamically Reconfigurable Kinematic Devices Using Compliant Mechanisms and Tensioning Cables”. In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New York, NY, USA: Association for Computing Machinery, pp. 1–14. ISBN: 978-1-4503-9157-3. DOI: [10.1145/3491102.3502065](https://doi.org/10.1145/3491102.3502065). URL: <https://doi.org/10.1145/3491102.3502065> (visited on 09/21/2022).
- Yang, Willa Yunqi et al. (Oct. 2022). “Reconfigurable Elastic Metamaterials”. In: *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. UIST '22. New York, NY, USA: Association for Computing Machinery, pp. 1–13. ISBN: 978-1-4503-9320-1. DOI: [10.1145/3526113.3545649](https://doi.org/10.1145/3526113.3545649). URL: <https://dl.acm.org/doi/10.1145/3526113.3545649> (visited on 03/22/2023).
- Zheng, Clement et al. (Apr. 2022). “Shape-Haptics: Planar & Passive Force Feedback Mechanisms for Physical Interfaces”. In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New York, NY, USA: Association for Computing Machinery, pp. 1–15. ISBN: 978-1-4503-9157-3. DOI: [10.1145/3491102.3501829](https://doi.org/10.1145/3491102.3501829). URL: <https://doi.org/10.1145/3491102.3501829> (visited on 09/21/2022).