Copyright
by
Heather Petrow
2014
Haptic Interaction
in 3D Stereoscopic User Interfaces

Supervisor: Orit Shaer
Haptic Interaction in 3D Stereoscopic User Interfaces

by

Heather Petrow, MAS 2014

Thesis

Presented to the Program in
Media Arts and Sciences
in Partial Fulfillment
of the Requirements
for Honors in the degree of

Media Arts and Sciences

Wellesley College
May 2014
Acknowledgements

This work is the culmination of my academic study at Wellesley College and my experience in the Human-Computer Interaction Lab. In many ways it is also a summation and milestone of my varied and tangential journey. I find it uncanny and amazing that my path has led me to here.

I would like to especially thank and acknowledge my advisor, Orit Shaer, for introducing me to the HCI Lab and being a truly dedicated mentor. Your guidance, knowledge, and inspiration have been invaluable to my academic experience.

Thank you to my committee members, Sohie Lee and Dave Olsen, and also to my readers, Joy Renjilian-Burgy and Nicholas Knouf, for your assistance and insight. Thank you Sohie for being a constant source of warmth and instruction. Thank you Dave for the crash course in 3D animation, but more importantly for inspiring me to think critically and creatively. A special thank you to Joy for consistently taking me under your wing.

I thank my lab mates and friends, especially Cassie Hoef, for all of your help, advice, and encouragement.

I cannot express enough my sentiment and gratitude to the Davis Scholar Program and my friends in the community, for the academic experience I never imagined and the companionship of such amazing women.

I wish to especially thank my class dean and advisor Susan Cohen, without whom I may never have discovered Computer Science or the Media Arts and Sciences program.

Thank you to my family and dearest friends for your love and support through my best, my worst, and everything in-between.

Paul, thank you for always believing in me, supporting me, reminding me to laugh, and for being my inexhaustible partner and devoted other half throughout it all.
Abstract

Haptic Interaction in 3D Stereoscopic User Interfaces

Heather Petrow, Media Arts and Sciences
Wellesley College, 2014

Advisor: Orit Shaer

Haptic feedback is an area of technology that utilizes the sense of touch, by providing tactile interaction. It has been integrated into gaming consoles and mobile devices, and has been researched for its potential in programs that range from medical training simulations to collaborative workspaces. 3D stereo display is another growing facet of technology that is reexamining the possibilities of the user experience. The zSpace system is a computing hardware platform that simulates realistic, holographic, 3D stereoscopic vision. Using this system, this research project aimed to study how haptic feedback can enhance the user interface and understanding of 3D virtual space, by applying and exploring the effects of different types of haptic interaction in two zSpace applications. User experience in haptic and non-haptic versions of these programs was evaluated through a comparative analysis of various measures including observation, performance, presence, and workload.
# Table of Contents

List of Tables ........................................................................................................................................... ix

List of Figures ............................................................................................................................................ x

**INTRODUCTION** ................................................................................................................................... 1

**BACKGROUND** .................................................................................................................................. 3

Haptic and Stereo ....................................................................................................................................... 3
  Haptic Interaction: What is haptic feedback? ......................................................................................... 3
  3D Stereoscopic Displays ...................................................................................................................... 4

Hardware .................................................................................................................................................. 5
  Overview and Technical Specifications ................................................................................................. 5
  Glasses and Motion Parallax .................................................................................................................. 6
  6DOF Stylus ......................................................................................................................................... 6

Related Work .......................................................................................................................................... 7
  Training Task Simulation .................................................................................................................... 8
  Object Selection ................................................................................................................................. 8
  Collaboration ...................................................................................................................................... 9
  Data Representation .......................................................................................................................... 10

**FIRST STAGE** .................................................................................................................................... 11

Developing for zSpace in Unity .............................................................................................................. 11
  Using Unity for zSpace ......................................................................................................................... 11
  The Unity Workflow ........................................................................................................................... 12

Demo - Development and Learning the Haptic Capabilities of the System ........................................ 14
  Implementation ................................................................................................................................... 14
  Limitations and Future Work .............................................................................................................. 17
  Analysis and Conclusion ..................................................................................................................... 17
Appendix B ......................................................................................................................... 65
    B1. Demo Source Code: HapticObject.cs ................................................................. 65
Appendix C .......................................................................................................................... 68
    C1. zPuzzle Source Code: HapticObject.cs .............................................................. 68
Appendix D .......................................................................................................................... 71
    D1. Creative Project Source Code: HapticObject.cs ................................................. 71
    D2. Creative Project Source Code: LevelManager.cs .................................................. 72
    D3. Creative Project Source Code: ObjectManager.cs ............................................. 73
    D4. Creative Project Source Code: PlaybackControls.cs ........................................... 75
    D5. Creative Project Source Code: PowerUp.cs ......................................................... 77
    D6. Creative Project Source Code: Rods.cs ............................................................. 79
    D7. Creative Project Source Code: StylusRumbler.cs ............................................... 80
References ................................................................................................................................ 84
List of Tables

Table 1: Time per level – Independent t-test and effect results ......................... 25
Table 2: Time per level – Mann-Whitney U test and effect results ....................... 25
Table 3: Workload – Independent t-test and effect results .................................. 26
Table 4: Workload – Mann-Whitney U test and effect results ............................. 26
Table 5: Presence – Independent t-test and effect results ..................................... 27
Table 6: Presence – Independent t-test and effect results ..................................... 28
List of Figures

Figure 1: Unity Editor Interface – Main components (from left to right): Scene View, Game View, Project Panel, Hierarchy Panel, Inspector Panel ........ 13

Figure 2: Demo Application – scene with ball and box screenshot .................. 15

Figure 3: Demo Application – object collision screenshot .......................... 16

Figure 4: zPuzzle Application – level 3 screenshot ..................................... 21

Figure 5: Automaton Project – scene 1 interface screenshot ......................... 34

Figure 6: Automaton Project – scene 2 interface screenshot ......................... 34

Figure 7: Automaton Project – Unity editor view ............................................ 35

Figure 8: Automaton Project – first iteration with stock character models ......... 37

Figure 9: Automaton Project – final version animation screenshot ................. 37

Figure 10: Automaton Project – stylus moving pieces screenshot ................. 38
INTRODUCTION

Haptic feedback is an area of technology that utilizes the sense of touch by providing tactile interaction. It has been integrated into gaming consoles and mobile devices, and has been researched for its potential in programs that range from medical training simulations to collaborative workspaces. Since touch is a significant component of interaction in the physical world, research concerning haptic interaction in technology can be valuable for better understanding how to potentially utilize a broader scope of human sense and perception in the user experience. For example, tactile feedback in various applications could be used as an immersive cue to assist in the representation of realistic proprioception and in the creation of more intuitive interaction.

3D stereoscopic display is another growing facet of technology that is reexamining the possibilities of the user experience. It is already becoming increasingly prevalent in the entertainment industry, as evidenced by the advance of 3D film, television, and gaming consoles, with 3D graphics also evolving to higher resolution and sophistication. While stereo display is not necessary for 3D image representation, stereoscopic vision can provide a heightened level of spatial awareness and could further enhance 3D applications by providing a more immersive virtual environment. The zSpace[19] platform is the first-generation of computing hardware that simulates realistic, holographic, 3D stereoscopic vision. Investigation of its potential could help in understanding how humans understand 3D space in virtual reality, and what the potential applications are.

This research project aimed to study how haptic feedback can enhance the user interface and understanding of 3D virtual space, by applying and exploring the effects of different types of haptic interaction in two zSpace applications: zPuzzle, a 3D puzzle
application; and an original, creative project based on the idea of a virtual automaton. A comparative analysis of haptic and non-haptic versions of these programs, using a between-subjects user test, provide a basis of evaluation for various measures of the user experience, such as problem-solving, spatial reasoning, immersion, and recognition of physics and materiality in the virtual environment.

On a primary level, this research aims to better understand whether and how the inclusion of tactile information as an added immersion cue can help in the comprehension of 3D virtual space. However, on a deeper level, it also seeks to better understand how integrating a greater range of human capacity in technological interfaces could enhance the sense of meaningfulness and intuitive connection for the user, especially as technology continues to grow in sophistication, and become a more ubiquitous feature of human life.
BACKGROUND

Haptic and Stereo

**Haptic Interaction: What is Haptic Feedback?**

Haptic interaction refers to the ability to use the sense of touch in perception and manipulation. In the physical world, our reception of tactile information from the environment is an important channel through which we communicate and interface with our surroundings. It helps us to understand various external qualities, such as space, material, and texture, as well as motion and physics. Haptic feedback is the simulation of this sense of touch in a user interface environment by providing tactile, immersion cues as sensory input. This could take the form of subtle vibrations, such as one might receive on a mobile phone, or as force-feedback in virtual reality games.

As the sense of touch is an important component of human processing in the physical world, haptic feedback can be critical to understanding particular types of information in the virtual world [3]. For instance, it may be used to help understand spatial relationships between objects, or physical constraints and materiality. This type of information processing in UI system designs may be increasingly essential with the advance of technological innovation, for as our machines become more sophisticated and pervasive, our interaction with them may also become increasingly complex. It stands to reason that such complexity could require more intuitive interface techniques, utilizing the full range of human sensory perception.

Haptic research is applicable to many areas of human cognition and interaction, and so it draws upon a range of disciplines from neuroscience to engineering. In technology, it has an extensive and varied history, from pen-based tools to novel displays.
and immersive installations [7]. However, there have also been limitations to haptic interaction design due to lack of spatial continuity in current interfaces, unnatural interaction, and difficulty of combining haptic cues with spatial imagery in present displays.

So, although ability to touch is a prominent feature of human interaction, haptic interaction development and integration with current technology is still relatively limited and underdeveloped. Haptic feedback has yet to be widely integrated into existing user interfaces, or developed to the same level of complexity and sophistication as visual representation. Ben Challis of The Interaction Design Foundation postulates that “tactile interaction is perhaps being undervalued in terms of the potential source of feedback that it might offer,” and that the focus has been primarily too “fixed on the ‘doing’ rather than the ‘receiving’” [3].

3D Stereoscopic Displays

Stereoscopic displays work by providing different, offset images to the left and right eyes, which are combined in the brain to create the 3D illusion [19]. As humans experience the physical world in 3 dimensions, technological advances in 3D stereo displays are a movement towards visual representation with greater fidelity to reality within virtual environments. Therefore, advanced simulation of 3D imagery in technological applications can have the ability provide the user with more natural forms of interaction, with greater reliance on human experience, naïve physics, and intuition.

Research has shown, for instance, that 3D stereoscopic displays can help in virtual game playing. In particular, research employing stereo and haptic media in a networked game [5], showed that the stereo viewing can help with accurate object selection and
depth perception. Furthermore, other research has shown that even if a benefit in gaming performance is not observed, there is still a user preference for 3D stereo technology [8]. Absence of quantifiable benefit may point to the lack of current user interfaces developed to take advantage of stereo depth perception. This points to the possibility of future work to study 3D stereoscopic interfaces with the inclusion of additional immersion cues, such as head-tracking and haptic feedback, as well as software designed specifically to take advantage of 3D-stereo display.

**Hardware**

**Overview and Technical Specifications**

zSpace is “an integrated hardware and software platform for developing 3D applications” [19]. It is a leading-edge technology that provides for the display of virtual-holographic images with stereoscopic vision, while also allowing for direct control and manipulation through stylus input. One feature that sets the system apart from other similar technologies, such as head-mounted gaming platforms, is the desktop environment. Also unique to the system is the addition of motion parallax, which is the ability for the 3D image to change perspective as the head moves, effectively allowing the user to look around the virtual object as one might do in actual space.

There are four main components to the zSpace system: the display, the glasses, the stylus, and the software development platform. The display component is a 23.6inch (diagonal), stereoscopic, LCD screen with 1920X1080 full HD resolution. The stereo system is a quad buffer stereo system, which works by alternatively presenting the full resolution image to the left eye, followed by the right eye [19]. It is also equipped with infrared cameras for head tracking and stylus tracking. Due to this high graphic demand,
the system requires connection to a computer equipped with a powerful graphics card. The graphics system used for this research was the NVIDIA Quatro 600.

**GLASSES AND MOTION PARALLAX**

The glasses component of the system are polarized 3D eyewear, with reflective tracking points, so that the system can track the head’s position and orientation, using infrared sensors. This information is sent to the rendering system, so that the image on the screen is projected accurately from the viewer’s perspective. When the head moves, the image on the screen will update accordingly. In effect, this head-tracking component allows the user to experience the simulation of “looking around” a virtual object, as one would in real space. This contributes to the simulation and also reduces the motion-sickness sensation sometimes experienced when using stereoscopic technology without motion-parallax.

**6DOF STYLUS**

Interaction within the system is managed by the stylus component. The stylus input tool is a pen-shaped design, for intuitive manipulation. It is designed for full six-degrees-of-freedom (6DOF), which refers to the complete range of movement: forward/back, up/down, left/right, pitch, yaw, and roll. Additionally, it is equipped with 3 buttons for input and a motor for vibration feedback. zSpace can track the stylus position and orientation with respect to its different coordinate systems. This information can then be used to program stylus behavior, depending on the particular type of interaction desired. zSpace also provides a set of utilities and prefabricated stylus objects for use in the Unity3D game engine. This growing library of utilities provides high-level support for stylus development. Some of the customizations for the stylus include the effectual length of influence, the mode of selectivity, and graphical representation in the display,
which provides a visual indication of the stylus’ position and orientation in virtual space. Specifically for the purpose of programming haptic feedback, there are methods in the zSpace core API for setting and triggering the vibration of the stylus motor. Vibration is measured in seconds, for the on and off periods, and by the integer number of vibration cycle repetitions.

Related Work

There has been a range of related research concerning both 3D display and haptic interaction, and some at the intersection of both, but research concerning the combination of haptic feedback with 3D technology is still comparatively underexplored. Some of the applications that are being developed and investigated for researching haptic feedback in technology are task-training simulations [7][12], object selection both in 2D and 3D interfaces [6][11], devices for mental-health therapy [16], collaborative-virtual workspaces [13], and data representation [2][18].

In addition, there has also been a body of research specifically concerning haptic interaction design. Concepts investigated in haptic design include the use of a sketching tool to digitally conceptualize and construct haptic feedback [4] and workshops to create tangible, haptic mechanism for better understanding haptic design through the process of doing [9]. There is an implication in this body of research that the creation of sensible, haptic feedback may require creative approaches that deviate from convention.
**TRAINING TASK SIMULATION**

In simulations of specialized training tasks, particularly when fine motor control is required, haptic feedback could be necessary for realistic reproduction of actual task conditions. For example, a prototype developed for virtual dentistry training showed that there are potential benefits of haptic interaction for accurate simulation of task execution [12]. The research prototype included haptic feedback as part of a multimodal learning system with multisensory feedback to reproduce task conditions and physical tools. It was implemented using a 3D stereoscopic display with motion and head tracking as well as a tangible user-interface for physical tools. Preliminary studies demonstrated the importance and potential of haptic interaction as part of a multimodal system for virtual learning and specialized tutorials.

**OBJECT SELECTION**

Object selection is another task that has been researched in the context of both haptic and 3D technology. As a fundamental aspect of interaction, it is an exercise that has relevance to a wide range of virtual applications. A study of whether and how pointing tasks could be enhanced by providing haptic cues, using a remote pointer device, showed haptic feedback contributed to user ability in object selection in 2D [6]. Haptic feedback was employed to indicate various actions, such as confirmation on completion, warning on error, and also continuous analog feedback on a certain value. The experimental task in this study used a haptic wand pointer device and a 2D display of circular targets. Results showed that haptic feedback improved performance and completion of tasks, but not necessarily of the user’s perceived performance and experience. This points to the potential benefit of haptic feedback for object selection, but
also the possibility that the design of the haptic feedback has an impact on the user’s perception of haptic interaction.

Another study analyzing the performance of 3D selection tasks used a large-scale immersive interface, equipped with haptic force-feedback and natural hand movement interaction [11]. The experiment investigated user performance in a task between two target objects, when varying haptic cues were provided to inform the trajectory from the first target to the second target. In particular, it studies whether soft feedback, hard feedback, or no feedback affected the user’s ability to select parts from the first object and then move to the second object. Results showed that there were differences in performance between the tasks of selecting one target and two targets. Force-feedback improved completion time in the task of selecting two targets, but target object size was also a significant influence in all conditions.

**Collaboration**

Collaboration in technology and virtual reality is another area of research where haptic interaction and 3D visualization could be relevant and informative, especially as remote interaction becomes an increasingly prevalent method of communication. Using a 3D, virtual desktop environment, one research project studied the effects of haptic feedback in a collaborative setting to improve performance on-task, as well as increase perceived virtual presence, social presence, and task performance [13]. Testing was executed using collaborative pairs who were tested on a series of five tasks, such as lifting and moving cubes together. Tasks were done across conditions, with haptic feedback and without. Results showed that haptic feedback had a significant impact on
the improvement of task performance, as well as perceived task performance, and perceived virtual presence, but did not significantly affect perceived social presence.

**DATA REPRESENTATION**

Haptic interaction for 3D spatial comprehension has also been applied to the study of understanding data representation. Audio-haptic interaction used to create and explore 3D graph structures has been shown to provide cues for recognizing the presence of graph elements, specifically for visually impaired and blind persons [2]. The conducted research was implemented using the Phantom Omni haptic device and was paired with a speech-synthesizer for the addition of audio feedback. This prototype environment guides the user in the creation and exploration of graph structures, using vibration cues to indicate the location of edges and nodes. Preliminary testing showed that graph nodes were successfully interpreted using haptic points.

Another research project that demonstrated the use of haptic feedback for data representation is Slurp [18], an interface designed as a physical mode of storing, transferring, and interacting with abstract digital media data. Using a tangible user interface, this research demonstrated the development and design of novel haptic interaction for processing abstract information, such as digital data. The implemented interface included an eyedropper shaped object, symbolizing a liquid metaphor, to extract digital information from physical objects and then transfer the data to a digital display. Haptic feedback in the device provided vibrational signals to indicate the detection of extractable data, with short and long vibration periods used to differentiate between discrete and continuous information. This feedback was designed to resolve the potential issues of cognitive mapping between abstract and physical objects.
FIRST STAGE

Developing for zSpace in Unity

The development phases of this research required gaining knowledge and familiarity with the zSpace system SDK and API, as well as the Unity 3D development platform and API. Unity [15] is an integrated, multi-platform game engine for the creation of interactive 3D and 2D content. As a popular gaming engine, designed for the generation of interactive, 3D content with relative speed and ease, it is the environment zSpace uses to facilitate the development of zSpace-enabled applications. The zSpace SDK documentation also provides instructions for porting pre-existent Unity 3D applications to zSpace, as well as user-interface guidelines to maximize the potential of the system’s stereoscopic and proprioceptive properties. At a high level, using Unity as the intermediary communicator between the system and the graphics hardware abstracts away much of the underlying complexity for the creation and display of stereo 3D content.

Using Unity for zSpace

zSpace application development is ported from Unity through a plug-in and editor patch that can be downloaded from the zSpace website [19]. The plug-in is a Unity package that contains the prefabricated assets for making Unity applications work with zSpace, including the core libraries and stereoscopic rendering capability. To create the stereoscopic image, the zSpace core object creates two extra cameras, one for each eye. It takes the main camera information and supplements it with information to provide the
stereo separation for the two cameras, left and right. This stereo capability can also be
turned off and on, by accessing the core methods. There are also methods to set the inter-
pupillary distance, set the plane clipping, and check the coordinate-system scales.

zSpace utilities are an optional, extra layer of control over the core class
functions, which can be imported into Unity. The utility packages contain additional
plug-ins and pre-fabricated objects that can be used for creating more sophisticated and
varied user-interface behavior. Some of these utilities include additional stylus shapes
and manipulation tools, buttons, and a tool bar.

THE UNITY WORKFLOW

The main components of the Unity editor workspace contain a scene view, a game
view, hierarchy panel, project panel, and inspector, (see Figure 1). The scene view is a
panel for interactive creation of the environment. Elements such as objects, characters,
cameras, and lights can be arranged and transformed, by either editing the values directly
in the inspector, or by using the tools in the tool bar for moving, rotating, and scaling.
This is similar to the type of interface provided by many graphics-editing software tools,
such as Adobe Photoshop or Autodesk Maya [1]. The game view provides a scene
preview, as it would appear within the rendered project, from the perspective of the
camera. Game objects are instantiated in the project from the hierarchy panel, which also
maintains relationships and dependencies between objects, as well as the attachment of
object scripts and components. The inspector displays all of the information and
attributes in a listed format, for management, editing, adding components, and assigning
variables.
Figure 1: Unity Editor Interface – Main components (from left to right): Scene View, Game View, Project Panel, Hierarchy Panel, Inspector Panel
Demo - Development and Learning the Haptic Capabilities of the System

IMPLEMENTATION

The first stage of development began with the creation of a simple demo or “Hello World” application as a self-tutorial to assist in learning the basics of the zSpace system and Unity development. It also provided a basis for better understanding the creation of sensible haptic interaction and the range of haptic feedback possible for development with the zSpace stylus. Learning development for zSpace and Unity required extensive research, reading through both systems’ documentation, SDKs, and APIs, looking through developers’ forums, and following introductory lessons and tutorials. The steps for creating the demo were as follows:

1. Begin a new project in Unity
2. Create and lay out a simple 3-dimensional scene
3. Import the zSpace packages
4. Enable the stereo by assigning the main camera of the scene to the camera variable in the zSpace core script
5. Create the stylus object, using the prefabricated package
6. Attach components and behaviors to the objects
Figure 2: Demo Application – scene with ball and box screenshot

The demo interface consists of a ball and box enhanced with components, such as color, material, physics, and haptic interaction, (see Figure 2). Sphere and cube Unity game objects are the primary players in the scene, and Unity plane and cube game objects act as the floor and walls to compose the environment. Each object was given a specific configuration of attributes depending of their particular type and function. This required learning how to create and apply game object components and also how to configure the physics and material properties. For physics components, objects were also provided with rigid bodies for collision detection and gravity.
Rigid bodies in Unity are components attached to game objects to enable realistic behavior through control from the physics engine. They serve several important purposes in this application. As stated, their primary function is to provide an entity for physical interaction, since game objects are just empty, non-reactive shells. Collision detection is critical for determining the collision event, which is then used to trigger the haptic response. It is also important for development with the zSpace stylus, because one way to program the stylus is by scripting for the selection and movement of objects by detecting collision between the stylus ray and other rigid bodies. So, without the presence of rigid bodies the stylus would not be able to pick up and manipulate objects in the scene.

Adding the haptic interaction to the scene required listening for different types of physical interaction between objects in the scene, including interaction with the stylus. The assigned vibration is played as the appropriate response to a triggered action. Within the zSpace core API there are methods for enabling the stylus vibration, setting the on
and off periods, and number of vibration cycle repetitions, (see Appendix B). These are passed as parameters. On and off periods are in seconds passed as floats, and the repeat parameter is the integer number of cycles to play the on and off periods [19].

After a period of experimentation, five distinct types of possible haptic interaction were identified and tested in this demo. These included object hover, object selection, collision between objects, combination of hover and collision, and a haptic indicator for the ball’s rolling motion. In addition to these interaction types, the type of haptic cue was assessed, such as the strength and repetition of the vibration. Each of these haptic types was applied to the ball and the cube in varying permutations.

LIMITATIONS AND FUTURE WORK

Since the stylus hardware only provides a limited range of vibrational feedback, the possible complexity of haptic interaction was somewhat restricted. To potentially mitigate this limitation, different durations of vibration were assigned to simulate varying strengths of feedback. Distinct vibration strengths were also implemented for each interaction type to create a separation between different actions. Future work could include exploring more haptic interaction types, as well as hacking the system to integrate another input device, such as the Phantom Omni, with a wider range of haptic feedback possibilities.

ANALYSIS AND CONCLUSION

Informal testing was conducted by comparing different implementations of haptic feedback in the program. Three individuals separately tested the program and provided verbal responses, and the design was then reiterated upon based on feedback and
observations. Testing showed that the most sensible haptic interaction between these objects was the vibration on collision. The on hover haptic interaction was also assistive for showing object selection by the stylus; however, combined with collision cues, the feedback became confusing and overloaded. This seemed to be true even if the strength of vibration was different for each cue, but could perhaps be more made coherent, with more experimentation.
ZPUZZLE

INTRODUCTION

zPuzzle is an application developed in the Wellesley Human-Computer Interaction Lab for viewing and manipulating 3D interlocked puzzles using the zSpace stereoscopic display. Modeled after the computer game “Interlocked,” the application is structured to cycle through a series of 11 levels of varying complexity, which are solved by taking the pieces of each puzzle apart.

The program was initially designed to examine spatial problem solving across 3 test conditions of varying immersive cues and interaction methods: interactive 3D stereoscopic display (i3DS); interactive 3D monoscopic display (i3DM); and 3D graphical user interface (3DGUI). The i3DS condition includes the use of 3D stereoscopic-enabled display and the motion-parallax feature, with bimanual interaction. Bimanual interaction uses the mouse for rotation of the scene, and the stylus for direct manipulation of the puzzle pieces. The i3DM condition also supports bimanual interaction, but provides zero immersion cues, with stereo and motion-parallax disabled. Lastly, the 3DGUI condition was created as a control, without either immersive cues or bimanual interaction, using mouse operation through an axis selection gimbal instead.

This research aimed to extend this study by including the haptic immersive cue with 3D stereo-enabled display and motion-parallax feature (i3DS), as well as bimanual interaction. It was compared against a control group, using i3DS without haptic feedback. The objective of the study was to better understand whether and how haptic feedback paired with interactive 3D stereoscopic display supports spatial reasoning compared to 3D stereoscopic display without the haptic cue.
IMPLEMENTATION

Implementing haptic feedback in the zPuzzle application first necessitated studying and understanding the underlying structure and composition of the program. Overall, each level of the puzzle is a separate scene in the zPuzzle Unity project. These are added to the project build settings hierarchy in level order. To create the puzzle objects within the scenes, each was designed as a composition of separate pieces. Puzzle pieces were constructed using groups of Unity game object primitive shapes, primarily cubes, parented under an empty game object to represent the respective part. Finally, each part was provided with collider and rigid body physics components to detect forces and collisions, and allow for interaction with the stylus. The logic of the program is controlled through C# scripts, for tracking block positions, level management, stylus and mouse behavior, and also to log interaction data.

To add the haptic feedback to the puzzle, a haptic object script was applied to each puzzle piece, (see Appendix C). This haptic object script was a slightly modified version of the script created for the demo application, developed during the first phase of research. Modification included a revision for application of the script to generic shapes, whereas the demo haptic object script was written specifically for the ball and box game objects. The script contains variables for assigning particular vibration strengths to different interaction types, such as on hover, on selection, and on collision. Methods were then included in the script to detect these interactions, and trigger the appropriate response by calling a method to play the assigned stylus vibration setting. The setting is passed as a numerical parameter to a switch statement and contains an on period (in seconds), off period, and cycle repetition. For this application, the collision setting was 0.34f for both the on and off period, and 0 for repeat.
To implement the on hover and on selection methods for hover and selection detection, the zSpace stylus class methods, included in the core API library, are accessed to report the hovered or selected object to the haptic script. The OnCollisionEnter(), detection method is part of the Unity API, and executes the function’s code upon initial collision between the rigid body of the game object it is applied to and another game object’s rigid body. Finally, the stylus vibration is executed by accessing the zSpace core API methods for setting and playing the stylus vibration.

Figure 4: zPuzzle Application – level 3 screenshot

**Evaluation Methodology**

To analyze the effects of haptic feedback in the zPuzzle application, a between-subjects study was conducted across 2 conditions: haptic and non-haptic, both using the immersive, 3D-stereoscopic display (i3DS) and bimanual interaction. The sample
consisted of 26 adults, aged 18-37, randomly chosen on a volunteer basis, and residing within the Wellesley-Boston area. The participants included 3 males and 23 females. Most participants were Wellesley College undergraduate students, concentrating in the sciences. Volunteers were equally split between conditions, 13 units per group: 1 male and 12 females in the haptic group, 2 males and 11 females in the non-haptic group.

The study design consisted of a pre-task training and debriefing, the puzzle-solving task, a set of oral, observational questions, and a post-task, online questionnaire. Training included a short, zSpace system-check to allow to user to acclimate to the system and its capabilities, such as stereo, head-tracking, and stylus control. This also provided an opportunity to test whether the user could perceive 3D without discomfort.

For the task, each participant was tasked with solving a series of 11 3D puzzles of varying complexity, either with the inclusion of haptic feedback or without. The non-haptic sample served a control group, against which to compare the potential effects of the haptic condition. Both conditions included use of the enabled stereoscopic display and bimanual interaction, using the mouse for scene rotation and the stylus tool with 6-DOF for manipulation and selection of the puzzle parts.

Lastly, participants were asked a series of questions to provide both quantitative and qualitative data to assess the user’s ability to solve the puzzles. Observational questions were asked in a verbal, interview context, directly following the task. The qualitative feedback gathered related to the user-interface and the user’s reactions to its features: the program and puzzles, stereo, stylus, and haptic feedback, if applicable. This was followed by the submission of an online questionnaire, which was designed to quantitatively assess the task, using different categorical indices: presence, workload, engagement, sickness, and prior experience.
Puzzle Task

The task entailed solving a series of 11 3D puzzles, of varying complexity. Each puzzle is presented as a 3D shape, composed of separate, interlocking blocks. To solve the puzzle, the components must be shifted to liberate individual pieces, until the puzzle is fully dismantled. Once a puzzle is fully dismantled, the screen automatically goes blank and displays the next level. Within the 11 levels, there are 5 distinct puzzle configurations, 3 of which are repeated twice with alterations in color. The level order is as follows: L1, L2, L3, L4, L3-rep, L11, L4-rep, L11-rep, L3-rep2, L4-rep2, L11-rep2, where rep indicates puzzle repetition. The task is complete when the program has cycled through all of the levels and each of the puzzles has been solved.

Analysis

Statistical analysis was executed using IBM SPSS Statistics software. Measures assessed included total time to complete all puzzles, total time per puzzle level, mental demand, presence, and prior experience. In all cases, the Kolmogorov-Smirnov test was applied to assess normality. When the distribution was normal, an independent t-test was used to evaluate significance at the 0.05 level. All normally distributed data categories were also tested for homogeneity of variance, using Levene’s test, and were found non-significant (p > .05), so equal variances were assumed. In cases, where the distribution deviated from normal, the Mann-Whitney test was applied, also testing for significance at the 0.05 level. In all cases the effect size was calculated as a standardized measure of the magnitude of the effect. No significance was found in the differences between haptic and non-haptic conditions for any of the testing categories, using independent t-tests and Mann-Whitney tests respectively. To further test whether the effect was substantive, t-
statistics were converted to Pearson’s Coefficient R values. The effects per category are discussed below.

**Total Time On-Task**

Total time is measured as the number of seconds taken to complete all 11 levels of the puzzle task. Using an independent 2-sided t-test at the 95% confidence interval, results showed that on average, participants in the haptic condition had greater total time on-task (M = 662.15, SE = 45.65) than in the non-haptic condition (M = 642.08, SE = 57.60). However, this difference was not significant t(24) = .273, p > .05, and it did not represent any substantial effect r = 0.056.

**Total Time Per Puzzle**

Total completion time was also evaluated per puzzle. On average, participants in the haptic group had lower or equal completion times per puzzle in the following levels: L1, L2, L3, L4, L3rep, L3rep2, and L11rep2. This difference was not significant, but the effect size varied between levels, (see Tables 1&2). Completion times per puzzle were higher in the haptic group in L11, L4rep, L11rep, and L4rep2. Again, this difference was not significant, and effect size varied. L1, L11, and the first iteration of L4 showed no substantial effect between the haptic and non-haptic conditions, (r < 0.1). L2, L3, L3rep, L4, L3rep2, L11rep, and L11rep2 represented a small effect, (0.1 ≤ r < 0.3). Level 4rep2 represented a medium effect (0.3 ≤ 0.5).
<table>
<thead>
<tr>
<th>Level</th>
<th>Haptic</th>
<th>Non-Haptic</th>
<th>$n(24)$</th>
<th>$r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>L4</td>
<td>$M = 79.15, SE = 10.49$</td>
<td>$M = 93.85, SE = 16.57$</td>
<td>-.75</td>
<td>0.15 *</td>
</tr>
<tr>
<td>L11</td>
<td>$M = 106.08, SE = 20.85$</td>
<td>$M = 100.46, SE = 12.39$</td>
<td>.23</td>
<td>0.05</td>
</tr>
<tr>
<td>L4-rep</td>
<td>$M = 53.31, SE = 6.39$</td>
<td>$M = 51.38, SE = 8.58$</td>
<td>.18</td>
<td>0.04</td>
</tr>
<tr>
<td>L11-rep</td>
<td>$M = 103.15, SE = 12.82$</td>
<td>$M = 81.85, SE = 12.97$</td>
<td>1.17</td>
<td>0.23 *</td>
</tr>
<tr>
<td>L11-rep2</td>
<td>$M = 93.08, SE = 12.94$</td>
<td>$M = 91.77, SE = 13.15$</td>
<td>.07</td>
<td>0.14 *</td>
</tr>
</tbody>
</table>

Table 1: Time per level – Independent t-test and effect results

<table>
<thead>
<tr>
<th>Level</th>
<th>Haptic</th>
<th>Non-Haptic</th>
<th>$U$</th>
<th>$r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>$Mdn = 26$</td>
<td>$Mdn = 23$</td>
<td>$ns$</td>
<td>-0.07</td>
</tr>
<tr>
<td>L2</td>
<td>$Mdn = 23$</td>
<td>$Mdn = 30$</td>
<td>$ns$</td>
<td>-0.17 *</td>
</tr>
<tr>
<td>L3</td>
<td>$Mdn = 34$</td>
<td>$Mdn = 41$</td>
<td>$ns$</td>
<td>-0.16 *</td>
</tr>
<tr>
<td>L3-rep</td>
<td>$Mdn = 31$</td>
<td>$Mdn = 32$</td>
<td>$ns$</td>
<td>-0.12 *</td>
</tr>
<tr>
<td>L3-rep2</td>
<td>$Mdn = 31$</td>
<td>$Mdn = 28$</td>
<td>$ns$</td>
<td>-0.10 *</td>
</tr>
<tr>
<td>L4-rep2</td>
<td>$Mdn = 40$</td>
<td>$Mdn = 27$</td>
<td>$ns$</td>
<td>-0.33 *</td>
</tr>
</tbody>
</table>

Table 2: Time per level – Mann-Whitney U test and effect results

**Workload**

Workload was assessed using the NASA-TLX index [10], which divides workload between 6 subscales: mental demand, physical demand, temporal demand, performance, effort, and frustration. These were assessed on a scale 1-5, with 1 representing lowest demand, and 5 representing highest demand. On average, haptic
condition participants had lower perceived workload contributions in Mental Demand, Physical Demand, and Performance. They perceived higher average workload in the Temporal Demand, Effort, and Frustration subscale categories. These differences were not significant. There was a small effect for the Haptic group in the Mental, Performance and Effort categories, reporting lower contributions to workload from these categories, \((0.1 \leq r < 0.3)\). Temporal Demand and Frustration had a medium sized effect, with higher frustration and temporal demand in the haptic group, \((0.3 \leq 0.5)\). There was no substantial effect on physical demand, (see Tables 3&4).

<table>
<thead>
<tr>
<th>Category</th>
<th>Haptic</th>
<th>Non-Haptic</th>
<th>t(24)</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal Demand</td>
<td>(M = 3.31, SE = .429)</td>
<td>(M = 2.38, SE = .331)</td>
<td>1.70</td>
<td>0.328</td>
</tr>
</tbody>
</table>

Table 3: Workload – Independent t-test and effect results

<table>
<thead>
<tr>
<th>Category</th>
<th>Haptic</th>
<th>Non-Haptic</th>
<th>U</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mental Demand</td>
<td>(Mdn = 4.00)</td>
<td>(Mdn = 5.00)</td>
<td>ns</td>
<td>-0.26 *</td>
</tr>
<tr>
<td>Physical Demand</td>
<td>(Mdn = 2.00)</td>
<td>(Mdn = 2.00)</td>
<td>ns</td>
<td>-0.06</td>
</tr>
<tr>
<td>Performance</td>
<td>(Mdn = 6.00)</td>
<td>(Mdn = 6.00)</td>
<td>ns</td>
<td>-0.17 *</td>
</tr>
<tr>
<td>Effort</td>
<td>(Mdn = 5.00)</td>
<td>(Mdn = 4.00)</td>
<td>ns</td>
<td>-0.12 *</td>
</tr>
<tr>
<td>Frustration</td>
<td>(Mdn = 3.00)</td>
<td>(Mdn = 2.00)</td>
<td>ns</td>
<td>-0.36 *</td>
</tr>
</tbody>
</table>

Table 4: Workload – Mann-Whitney U test and effect results
**Presence**

Presence was assessed using the MEC Spatial Presence Questionnaire, (MEC-SPQ) [17], which is divided into the following factors: Attention Allocation, Spatial Situation Model (SSM), Spatial Presence: Self Location (SPSL), Spatial Presence: Possible Actions (SPPA), Higher Cognitive Involvement, Suspension of Disbelief (SOD), Domain Specific Interest (DOI), Visual Spatial Imagery (VSI). These were evaluated using a 5-point Likert scale, 1-5, with 1 representing least agreement and 5 complete agreement. On average, the haptic condition users had higher average presence in all categories. There was a small to medium effect on user presence in the categories SSM, and SPSL, (0.1 ≤ r < 0.3), and medium effect size in the categories SPPA, Attention Allocation, and VSI, (0.3 ≤ 0.5). The condition had no substantial effect on SOD, DOI, or Higher Cognitive Involvement, (see Tables 5&6).

<table>
<thead>
<tr>
<th>Category</th>
<th>Haptic</th>
<th>Non-Haptic</th>
<th>t(24)</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSM</td>
<td>$M = 3.83, SE = .14$</td>
<td>$M = 3.71, SE = .16$</td>
<td>.53</td>
<td>.11 *</td>
</tr>
<tr>
<td>SPPA</td>
<td>$M = 3.87, SE = .20$</td>
<td>$M = 3.46, SE = .12$</td>
<td>1.71</td>
<td>.33 *</td>
</tr>
<tr>
<td>SOD</td>
<td>$M = 3.00, SE = .17$</td>
<td>$M = 2.94, SE = .19$</td>
<td>.23</td>
<td>.05</td>
</tr>
<tr>
<td>DSI</td>
<td>$M = 3.65, SE = .24$</td>
<td>$M = 3.52, SE = .21$</td>
<td>.42</td>
<td>.09</td>
</tr>
<tr>
<td>VSI</td>
<td>$M = 2.96, SE = .23$</td>
<td>$M = 3.52, SE = .21$</td>
<td>-1.80</td>
<td>.35 *</td>
</tr>
</tbody>
</table>

Table 5: Presence – Independent t-test and effect results
<table>
<thead>
<tr>
<th>Category</th>
<th>Haptic</th>
<th>Non-Haptic</th>
<th>U</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attn. Allocation</td>
<td>$Mdn = 4.5$</td>
<td>$Mdn = 4.25$</td>
<td>$ns$</td>
<td>-.34 *</td>
</tr>
<tr>
<td>SPSL</td>
<td>$Mdn = 3.5$</td>
<td>$Mdn = 3.0$</td>
<td>$ns$</td>
<td>-.22 *</td>
</tr>
<tr>
<td>Cog. Inv.</td>
<td>$Mdn = 4.0$</td>
<td>$Mdn = 4.25$</td>
<td>$ns$</td>
<td>-.07</td>
</tr>
</tbody>
</table>

Table 6: Presence – Independent t-test and effect results

**Observations**

Qualitative observational data was recorded to assess the user’s reactions to the system, interface, and haptic feedback. Results showed that all users, regardless of condition found the system engaging and intuitive. Puzzle solving strategies were consistent, employing a mostly trial-and-error approach, with some examination of the spatial relationships between pieces. All users also reported the influence of color as an assistive visual aide, to distinguish different puzzle pieces from each other. Users within the haptic condition mostly reported a positive reaction to the haptic feedback. 11 out of 15 of the participants said that it definitely helped their interaction with the puzzle, by providing an indication when two pieces were locked together or couldn’t be moved any further. For instance, one user said, “It was really necessary to indicate that I had picked something up and when I kept running into walls.” Another user said, “It was good. There was more sensory information for me to use to help me solve the puzzle.” The remaining 2 reported that they were unsure if it helped. None of the haptic users reported feeling confusion or discomfort from the feedback.
LIMITATIONS AND FUTURE WORK

One limitation of this study was the narrow gender and demographic range of participants. The majority of participants were Wellesley College students, majoring in the sciences. Another limitation of this study was the size of the sample group. Due to time and resource constraints, the study was limited to 13 users per condition. This small sample size may not have been large enough to provide an accurate representation of the population. It would be important to continue the study on a larger, and more diverse sample.

Additionally, the haptic condition could be retested using different versions of the haptic feedback. Two of the thirteen haptic condition users reported that the haptic feedback was useful in moderation, but occasionally became irritating, in particular when the puzzled was especially difficult or highly interlocked. One possible explanation for this is that the vibration may have been too strong and caused a stimulus overload, which negated any potential benefit. This points to possible future work to test the haptic immersive cue with less strong or more moderate applications of the vibrational response.

CONCLUSION

This research presents a study of whether haptic feedback contributes to spatial reasoning and problem solving using 3D stereoscopic display. While results show no significant difference between the haptic group and the non-haptic group, it can be concluded that the haptic feedback is also not having a negative effect on performance. Qualitative data shows that the users in the haptic condition group liked the haptic feedback overall and felt that it helped them to solve the puzzles and understand spatial relationships. They thought that the feedback contributed to their interaction with the puzzle by providing cues when pieces were stuck together or not able to be moved any
further. This provides a basis for future work, potentially indicating that haptic feedback could be assistive, but with modifications to the interaction design.
CREATIVE PROJECT

Initial Ideas

Considerations for the design of the original, creative project application included creativity, practical application, and feasibility. Most importantly, the project’s main objective was to explore how haptic interaction could enhance the user experience in a novel program using the 3D user-interface. 3D displays inherently facilitate the exploration of 3D models, such as machinery, architecture, anatomy, or biology, because they provide a visual representation of 3-dimensional spatial relationships. This was the departure point for much of the brainstorm process. The decided direction for the application of the program was divided between implementing a tool for practical or educational use, and creating a more explorative, recreational experience. Feasibility was also a consideration, considering the limited scope of time, experience, and resources. For instance, the idea to develop a tool for anatomical study was rejected, because it would require significant, additional expertise to create a simulation realistic enough to be a viable product. Finally, the main consideration was what type of interaction could benefit from the additional sensory cue of tactile feedback. It was also necessary to take account of the haptic feedback available with the technology for representing the desired range of input.

An initial idea was to use this project to create an application for learning in the sciences, such as biology or chemistry. Such an application could show the 3-dimensional composition of complex models, such as cells, molecules, or anatomy. Moreover it could allow for immersive exploration, through ability to interact with components and simulate bio-chemical processes. Haptic feedback could be assistive in such an application by providing a physical cue for indicating the presence and nature of
reactions and by providing spatial guidelines, such in in navigation and assembly of modeled parts.

Another initial idea was the creation of a game that could benefit from haptic feedback or interaction, such as target selection, racing and movement, or spatial-logic types of game. Video game systems and virtual-reality devices are a reasonable domain for the application of haptic feedback, as they are interactive and already have a precedent with haptic integration. Historically, haptic devices both experimental and commercial have been introduced, such as the Nintendo haptic glove, and arcade-style simulators. Now, many current games have utilized haptic feedback, from the addition of rumble effects in handheld console controllers to the incorporation of mobile-phone applications with vibration. However, while 3D game systems with haptic feedback have already been implemented to an extent, the range of available feedback still limited, and the addition of stereoscopic display would be a novel element. Games could also provide a metric to study spatial reasoning.

Art and design was another possible category for the project application. A drawing or painting platform could allow for the creation of 3-dimensional pictures, with haptic feedback to represent the feel of contact, when drawing with a physical tool. Additionally, a program for the design and construction of architectural models, such as in a building design or urban planning could potentially benefit from 3D visualization and tactile feedback. Such an interface could allow for the placement and layout of different structures, providing different haptic representations of material, environment, or contact with other structures. There were foreseen technical limitations to the execution of this idea, due to the limited range of haptic feedback provided by the zSpace hardware. It would be difficult to represent such detail of tactile representation using only the vibrations available.
The final idea was the creation of a virtual automaton. Utilizing the 3D representation of models, this project could show the construction of a mechanical device and its interacting parts, as well as allow for interactive building and creation. Differing from the zPuzzle task, this application would be developed to further understand whether and how haptic feedback could provide an additional perceptual cue for 3D visualization in a building and creation environment, in which the task was more based on exploration and experimentation than problem solving.

**Project – Automaton**

The application interface consists of an assembled automaton model, in the style of a rudimentary, DIY machine. The machine consists of a base with a rod and lever, and 3 interchangeable cams (gears) and characters, (see Figure 5). The cams can be placed onto the rod in 3 separate positions to initiate different actions for the characters, which are simple animated dance moves. Similarly, the characters can be rearranged to fit into 3 different slots on the machine base. This allows for various permutations of characters and cams, for the creation of different action sequences in the characters.

There is also a second scene in the application to playback the animation sequences created for each character, (see Figure 6). So, as the characters are animated on the automaton, the actions of each character are stored in a queue to be played back when the user enters the next scene. In this scene the characters are in a room, and there is a play and stop button to start and stop the animation playback. Music plays in the background when the play button is pressed, so that the characters appear to be dancing.
Figure 5: Automaton Project – scene 1 interface screenshot

Figure 6: Automaton Project – scene 2 interface screenshot
IMPLEMENTATION

The project entailed zSpace development using Unity. Building upon the introductory knowledge gained during the previous phases of this research: 1) development of the demo, and 2) the integration of haptic feedback with zPuzzle, the implementation of this project required additional tools and skills to create a more graphically complex scene and for more extensive interaction. Graphically, the Unity prefabricated game objects are limited to simple polygons. To integrate more advanced graphics, it is necessary to import models from other sources, either bought commercially, downloaded from open source platforms, or created and imported from 3D modeling applications.

Figure 7: Automaton Project – Unity editor view
Models

The base object models for this application were developed using the 3D modeling and animation software, Autodesk Maya [1]. These included the base box shape, the shaft, the crank, and the cams. They were created using polygonal meshes, and exported as .obj files. The first iteration of character models were downloaded from Unity project tutorial demos and from the online digital media supply website, TurboSquid [14]. These models were pre-animated with several different animation states, and imported to Unity as packaged .fbx files. These pre-fabricated characters were used as interim placeholders to test the program until they could be replaced through the development of new character models, (see Figure 8 below). The final iteration of models consists of a turtle, pig, and chick. They were created and rigged in Maya, then exported as .fbx files with baked key frame animations, (see Figure 9 below).

Animation

Animation of the crank and cams is enabled through a method within the C# script ObjectManager, (see Appendix D). This method listens for a boolean when the stylus hover object equals the “Crank” game object. It then rotates the crank and cams around the appropriate axis of rotation, by transforming the objects’ rotation vector for as long as the crank is hovered. The character animation is also triggered by this event. When the body of the code is executed, the character’s function, BeginAnimation(), is called at the initiation of the hover event, and StopAnimation() is called to end the animation at the hover termination, (see Figures 8 & 9).
Figure 8: Automaton Project – first iteration with stock character models

Figure 9: Automaton Project – final version animation screenshot
Stylus

The zSpace stylus with 6-DOF is the main input tool. It is graphically represented in this application using a blue stylus ray with a pointer tip icon to aide in the visual representation of the stylus’ position and location. Using the default zSpace stylus class and utilities, the stylus behavior script reports object intersection events, such as hover, selection, and object collision so that it can select and move game objects with rigid body components. It is configured to select objects by pressing the center circular button of the pen. This used to dissemble and reassemble the machine, or move components around (see Figure 10).

Figure 10: Automaton Project – stylus moving pieces screenshot
Logic

The logic for the program was scripted in C#, (see Appendix D). The ObjectManager script is the primary means of control and organization of the different game objects’ states and locations. It maintains 4 arrays of the objects, separated by type: base pieces, player pieces, cam pieces, and rod pieces. It is also responsible for triggering the animation when the crank is hovered over with the stylus tip, by listening for the event in its update method.

The script PowerUp controls character animation, and is applied to each character’s game object. This script contains an array to store the character’s animation clips, and functions to start and stop the playing of the animation. It also maintains a queue of animation clips that have been initiated when the crank is hovered over. This queue is used to store the sequence of animations for playback in Scene 2.

Haptic feedback in the project is created using a combination of a modified version of the zSpace utility StylusRumbler script and the haptic scripts written for the demo project and zPuzzle. StylusRumbler maintains variables for vibration settings, and functions to play the stylus vibration. Within the script, it listens for the stylus hover event, to play the hover vibration. Other actions, such as snapping, and collision are detected elsewhere, the Snap class and HapticObject class respectively. These classes use instances of the stylus game object to access the StylusRumbler.

Limitations and Troubleshooting

Some of the problems encountered during the implementation of this program were learning to correctly import models and animations, and instantiate them in the project. Unity and zSpace version control was another issue. This application was developed using Unity version 3.5.6 and the corresponding zSpace SDK. These platforms
have since been updated to newer versions. However, issues were encountered with the newest zSpace patch for Unity. Additionally, Unity has since created a new animation system called Mecanim, which was previously Legacy. Therefore, problems were encountered in the process of following protocols that were applicable to the updated versions, but not the previous versions. Research was required to find the appropriate, archived documentation.

**Evaluation Methodology and Analysis**

Informal studies on 5 volunteer participants were conducted to assess the base features of the program and the potential benefits of haptic feedback for interaction, compared to a non-haptic version. Subjects were asked to perform basic tasks in the program with haptic feedback enabled and without. Tasks included hovering, moving objects, and snapping objects into place. They were subsequently asked several qualitative follow-up questions regarding their reactions to the program, the haptic feedback and perceived differences between the haptic-enabled version and the non-haptic version. All of the users were observed to have more difficulty completing the tasks in the non-haptic condition than in the haptic condition. They also reported liking the haptic version better, having greater ease performing the tasks and manipulating objects with haptic feedback enabled. Overall, participants said the haptic feedback helped to know what to do and whether things were working properly. One user said, “I definitely liked the haptic version better. It was much easier to know when things were working properly, because I felt more connected to the application.” Other users also said that the feedback helped to “connect” them to the program and 3D environment. According to one user, “The haptic feedback was definitely helpful. I think being
introduced to an entire 3D world can be very disorienting, but the feedback helped to connect me to it and ground me in it by using more of my senses.”

**Future Work**

Future work for this project includes stylistic and design improvements, including the creation and addition of more 3D animated models and greater detail in the machine components. Additionally, the program can be further developed to include additional interface features, such as 3D buttons and pop-up indicators to provide instructions and verbal feedback. Other future work includes performing formal user testing and analysis to quantitatively analyze any potential benefits of the haptic feedback in this application.
CONCLUSION

This research presents a study of haptic interaction paired with 3D stereoscopic technology and its potential impact on the user experience. It investigates whether and how haptic feedback can provide the user with assistive sensory information, as a cue for spatial comprehension, navigation, task execution, and immersion.

Two zSpace applications were used to analyze the potential effects of haptic feedback for spatial reasoning and problem solving. Haptic feedback was applied to the zPuzzle application to statistically analyze performance in users executing a spatial reasoning task with haptic feedback and without. Although, significance was not observed, there was no negative consequence to haptic feedback, and all participants reported a perceived benefit. A novel application was developed to better understand haptic feedback in a 3D stereoscopic environment designed for immersive interaction and exploration. Users reported feeling more connected to the program and 3D environment when haptic feedback was enabled, and had greater ease in task execution. Both studies showed that there is a user preference for haptic interaction.

This research is also a discovery of haptic interaction design, including its strengths, limitations, and possibilities. Research has shown that the making and design of haptic interfaces can help designers achieve greater sensitivity and understanding of haptic interaction [9]. This was definitely true throughout the course of this work.

This research was conducted in the spirit of discovery, and in the hope that future developers, researchers, and designers will add to the body of research concerning haptic interaction, and work towards better understanding haptic feedback implementation and design for the next-generation of human-computer interfaces.
Appendix A

A.1 zPuzzle User Study Materials

Introduction and Summary

Link to Protocol - Must be completed per session.

Hi, ___________. My name is ___________, and I’m going to be walking you through this session today. Before we begin, I have some information for you, and I’m going to read it to make sure that I cover everything. You probably already have a good idea of why we asked you here, but let me go over it again briefly.

zSpace Spatial Rotation Test Explanation
We’re asking people to try using a 3D holographic display that we’re working on, so we can see what are the benefits of using it. The session should take about 30-40 minutes. The first thing I want to make clear right away is that we’re testing the system, not you. You can’t do anything wrong here. As you use the system, I’m going to ask you as much as possible to try to think out loud: to say what you’re looking at, what you’re trying to do, and what you’re thinking. This will be a big help to us. Also, please don’t worry that you’re going to hurt our feelings. We’re doing this to study the system, so we need to hear your honest reactions.

If you have any questions as we go along, just ask them. I may not be able to answer them right away, since we’re interested in how people do when they don’t have someone sitting next to them to help. But if you still have any questions when we’re done I’ll try to answer them then. And if you need to take a break at any point, just let me know.

You may have noticed the camera. With your permission, we’re going to record what happens on the screen and our conversation. The recording will only be used to help us figure out what are the benefits of using this system, and it won’t be seen by anyone except the people working on this project. And it helps me, because I don’t have to take as many notes.

If you would, I’m going to ask you to sign a simple permission form for us: https://docs.google.com/spreadsheet/viewform?formkey=dERQWjhQejJyTm9LN3lidERHYUhNcG6MA#gid=0. It summarizes everything I mentioned so far and asks for your permission to videotape the session, it also clarifies that the recording will be kept confidential and will only be seen by the researchers working on the project.
**Stereo and Stylus test**
First, we’re going to test whether you can perceive 3D using this system. (Show stereo test)

Now, you’re going to do a little training exercise to help you acclimate to the system and it’s capabilities. The glasses allow the system to track your head so that the 3D images can change perspective for you, the stylus is how you manipulate objects in the system. (Show glasses test and stylus test)

**Interlocked**
Now you’ll get to solve some 3D puzzles. Please enter your name in the text field and select (haptic).

This is not a puzzle yet, it’s a training task to introduce you to the task of solving these puzzles by taking these puzzle pieces apart.

After 5 minutes, you have the option of resetting the interface and/or asking for a hint.

**Condition 1: 3D - Pen**
Take the stylus in one hand. The pink line you see is the stylus ray. When the ray hits a puzzle piece you can grab the piece by clicking the circular button on the stylus and move the piece by moving the stylus.

Try looking around the puzzle, when your head moves, the display will update according to your perspective.

If you want to rotate the object you can move a mouse around. Try doing so now.

Take your time manipulating the pieces here. Once you’re ready, press the right arrow key to start.

Once you completely take apart a puzzle, the screen will stop responding, go black, and display the next level.

**Post-zspace Surveys**

3D Experience and Engagement NASA Survey:
https://docs.google.com/spreadsheet/viewform?formkey=dHNtcEpGczhuV2dpLWhXelFvaXZmemc6MA#gid=0
zSpace Consent Form

Please consider this information carefully before deciding whether to participate in this research.

Purpose of the research: To understand the strengths and weaknesses of 3D stereo interfaces.

What you will do in this study: If you decide to participate, work to solve 6 puzzles using the zSpace system then you will be asked to complete a post-task test and fill in questionnaires about your Experience. You will be videotaped.

Time required: Approximately, 30-40 minutes.

Benefits: There are no direct benefits, but you may find it interesting to use next-generation technology. Upon study completion you will be paid a $10 gift certificate.

Risks: There are no risks associated with this study. During the study there is a small chance that you will fill minor visual discomfort due to the use of a stereo display (eyestrain, blurred vision, difficulty in focusing). Such discomfort is short term and will cease when you stop using the 3D stereo display.

Participation and withdrawal: Your participation is completely voluntary. You may quit at any time without penalty.

Confidentiality: The session will be videotaped. This information will be kept strictly confidential. Your videotape will only be used for research purposes. It will be kept in a secure server and will be destroyed after the study is complete. When research results are reported, responses will be aggregated (added together) and described in summary.

To Contact the Researcher: If you have questions or concerns about this research, please contact: Orit Shaer, Phone: (781) (283 3093); 106 Central Street, Wellesley, Massachusetts 02481; email: oshaer@wellesley.edu.

If you have questions about your rights in this research, concerns, suggestions, or complaints that are not being addressed by the researcher or research-related harm contact: Nancy L Marshall, Chair, Wellesley College IRB, Phone: (781)(2832551), 106 Central Street, Wellesley, Massachusetts 02481 (email: nmarshall@wellesley.edu).

Please sign below if you agree with the statement below. "I have read the description of the study and voluntarily consent to participate. I understand that I may discontinue participation in this study at any time without any penalty and that
there are no risks associated with this study. I can request a copy of this consent form."

zSpace Haptic Study Protocol

This is the zSpace Interlocked User Study protocol. Condition instructions are listed below.

3d with Stereo and Haptic Feedback.

The haptic version can be accessed by selecting the haptic option at the start menu, or by initializing the "level tutorial - haptic" level.

**Before the study**
- Check the cameras
- Set up the screen in 3D
- Pull up consent form
- zSpace system check (from zSpace menu)

**Once the user gets in**
- Consent form
- zSpace system check “tutorial”
- Manipulation tutorial

**Participant Name**

**Sex**
- Male
- Female

**Condition**
- 3D Non-Haptic
- 3D Haptic

**During the study...OBSERVATIONS**
Elaborate on items checked below here as well as anything that stands out.

For each item checked describe in more detail in "Other" field. What level(s)? Anything that stuck out? etc.
- Sighed when they completed a level or several
Used parallax: Moved their head around to try to look around the puzzle.
Held pen close to screen: Note if this is for all levels or some in particular. Or after they’ve had a hard time with the puzzle.
Held pen far from screen: Note if this is for all levels or some in particular. Or after they’ve had a hard time with the puzzle.
Other:

**After activity before questionnaire**
After the study...Ask the user what they thought of the system
Get user quotes on experience.

**Puzzle Specific**
What strategies did you use to solve these puzzles?
How do you find using this system compared with solving a traditional puzzle?
Was it easier to solve the puzzles that were repeated after having done it once? If not, why?
Which puzzle was your favorite? Why?
If the puzzles were colorless or only one color, would that change the difficulty of the puzzle?

**Pen Specific**
How does it feel to use a stylus pen to move objects?
How did you like using the stylus?
Have you used a stylus as input before?

**3D Specific**
What other apps can you see for this type of 3D stylus interaction?
Did you find the head-tilting/leaning (parallax) feature helpful?
Does the stereo popping out effect help or hinder how fast you can interpret the puzzle?

**Haptic Specific**
How did the haptic feedback feel?
Did the haptic feedback help or hinder your interaction with the puzzle?
Was the haptic feedback uncomfortable or confusing?
zSpace Post-Task Questionnaire

MEC Spatial Presence (MEC-SPQ)

Please rate your experience with the virtual environment and the task.

I feel I am generally interested in solving spatial puzzles. *

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strongly Disagree   □ □ □ □ □    Strongly Agree

When I read a text about spatial solving puzzles, I can usually easily imagine the arrangement of the puzzle pieces described. *

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strongly Disagree   □ □ □ □ □    Strongly Agree

I feel that when someone shows me a blueprint I am able to imagine the space easily. *

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strongly Disagree   □ □ □ □ □    Strongly Agree

Spatial puzzles have often attracted my attention in the past. *

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strongly Disagree   □ □ □ □ □    Strongly Agree

In general, I love to think of solving puzzles. *

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strongly Disagree   □ □ □ □ □    Strongly Agree

In the past, I have spent a lot of time solving spatial puzzles. *

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strongly Disagree   □ □ □ □ □    Strongly Agree

During the session I felt as though I was physically present in the virtual environment of the application. *

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
During the session it seemed to me that I could do whatever I wanted in the virtual environment of the application.*

It's easy for me to negotiate a space in my mind without actually being there.*

When someone describes a space to me, it's usually very easy for me to imagine it clearly.*

I felt like I could move around among the puzzle pieces in the virtual environment.*

Even now, I can still draw a plan of the virtual environment.*

During the session I devoted my whole attention to solving the puzzles.*

During the session I thought intensely about the solution of the puzzle.*
I felt like I was a part of the virtual environment in the zSpace application. *

1 2 3 4 5

It seemed to me that I could have some effect on the puzzle pieces in the application as I do in real life. *

1 2 3 4 5

I felt the virtual environment captured my senses. *

1 2 3 4 5

I had the impression that I could reach for the puzzle pieces in the virtual environment. *

1 2 3 4 5

Even now, I still have a concrete mental image of the virtual environment. *

1 2 3 4 5

During the session I thought about whether the configuration of the the puzzles was really plausible. *

1 2 3 4 5

I thoroughly considered the relationship of the puzzle pieces to one another. *

1 2 3 4 5
I felt that I was able to make a good estimate of the size of the puzzle pieces in relationship to each other in the presented space. *

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly Disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly Agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Using the zSpace application activated my thinking. *

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly Disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly Agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I felt like I was actually in the virtual environment of the application. *

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly Disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly Agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

During the session I thought mostly about things having to do with solving the puzzle. *

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly Disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly Agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

During the session I concentrated on solving the puzzles. *

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly Disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly Agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I was able to make a good estimate of how far the puzzle pieces were from each other. *

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly Disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly Agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

During the session it was important for me to check whether inconsistencies were present in the virtual environment. *

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly Disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly Agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

My perception focused on the virtual environment almost automatically. *
During the session I concentrated on whether there were any inconsistencies in the virtual environment. *

It was not important for me whether the virtual environment contained errors or contradictions. *

I felt like the puzzle pieces in the virtual environment surrounded me. *

NASA-TLX Workload

Please select the item that in your opinion contributes the most to Workload *
- Effort
- Performance

Please select the item that in your opinion contributes the most to Workload *
- Temporal Demand
- Frustration

Please select the item that in your opinion contributes the most to Workload *
- Temporal Demand
- Effort
Please select the item that in your opinion contributes the most to Workload *
- Physical Demand
- Frustration

Please select the item that in your opinion contributes the most to Workload *
- Performance
- Frustration

Please select the item that in your opinion contributes the most to Workload *
- Physical Demand
- Temporal Demand

Please select the item that in your opinion contributes the most to Workload *
- Physical Demand
- Performance

Please select the item that in your opinion contributes the most to Workload *
- Temporal Demand
- Mental Demand

Please select the item that in your opinion contributes the most to Workload *
- Frustration
- Effort

Please select the item that in your opinion contributes the most to Workload *
- Performance
- Mental Demand

Please select the item that in your opinion contributes the most to Workload *
- Performance
- Temporal Demand

Please select the item that in your opinion contributes the most to Workload *
- Mental Demand
Please select the item that in your opinion contributes the most to Workload *

- Mental Demand
- Physical Demand

Please select the item that in your opinion contributes the most to Workload *

- Effort
- Physical Demand

Please select the item that in your opinion contributes the most to Workload *

- Frustration
- Mental Demand

Mental demand- How mentally demanding was the task? *

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Very High</td>
</tr>
</tbody>
</table>

Physical demand- How physically demanding was the task? *

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Very High</td>
</tr>
</tbody>
</table>

Temporal demand- How hurried or rushed was the pace of the task? *

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Very High</td>
</tr>
</tbody>
</table>

Performance- How successful were you in accomplishing what you were asked to do? *

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Very High</td>
</tr>
</tbody>
</table>

Effort- How hard did you have to work to accomplish your level of performance? *

<p>| 1 | 2 | 3 | 4 | 5 | 6 | 7 |</p>
<table>
<thead>
<tr>
<th>Frustration- How insecure, discouraged, irritated, stressed, and annoyed were you during the task? *</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6 7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task Engagement</th>
</tr>
</thead>
<tbody>
<tr>
<td>I lost myself in this activity. *</td>
</tr>
<tr>
<td>1 2 3 4 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I was so involved in this activity that I lost track of time. *</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I blocked out things around me when I was doing this activity. *</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>When I was doing this activity, I lost track of the world around me. *</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The time I spent doing this activity just slipped away. *</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0</td>
<td></td>
</tr>
</tbody>
</table>
I was absorbed in this activity. *

1 2 3 4 5

Strongly Disagree □ □ □ □ □ Strongly Agree

During this activity I let myself go. *

1 2 3 4 5

Strongly Disagree □ □ □ □ □ Strongly Agree

I was really drawn into this activity. *

1 2 3 4 5

Strongly Disagree □ □ □ □ □ Strongly Agree

I felt involved in this activity. *

1 2 3 4 5

Strongly Disagree □ □ □ □ □ Strongly Agree

This experience was fun. *

1 2 3 4 5

Strongly Disagree □ □ □ □ □ Strongly Agree

The content of this activity incited my curiosity. *

1 2 3 4 5

Strongly Disagree □ □ □ □ □ Strongly Agree

I felt interested in this task. *

1 2 3 4 5

Strongly Disagree □ □ □ □ □ Strongly Agree

Doing this activity was worthwhile. *

1 2 3 4 5

Strongly Disagree □ □ □ □ □ Strongly Agree
I consider this experience a success. *

1 2 3 4 5

Strongly Disagree ○ ○ ○ ○ Strongly Agree

My experience was rewarding. *

1 2 3 4 5

Strongly Disagree ○ ○ ○ ○ Strongly Agree

I would recommend solving puzzles with this environment to other students. *

1 2 3 4 5

Strongly Disagree ○ ○ ○ ○ Strongly Agree

This experience did not work out the way I had planned. *

1 2 3 4 5

Strongly Disagree ○ ○ ○ ○ Strongly Agree

This activity is attractive! *

1 2 3 4 5

Strongly Disagree ○ ○ ○ ○ Strongly Agree

This activity was aesthetically appealing. *

1 2 3 4 5

Strongly Disagree ○ ○ ○ ○ Strongly Agree

I liked the graphics and images used on this activity. *

1 2 3 4 5

Strongly Disagree ○ ○ ○ ○ Strongly Agree

This activity appealed to my visual senses. *

1 2 3 4 5
The design of this activity was visually pleasing. *

1  2  3  4  5

I felt frustrated while doing this activity. *

1  2  3  4  5

I found this activity confusing. *

1  2  3  4  5

I felt annoyed while doing this activity. *

1  2  3  4  5

I felt discouraged while doing this activity. *

1  2  3  4  5

Doing this activity was mentally taxing. *

1  2  3  4  5

This activity was demanding. *

1  2  3  4  5
Simulator Sickness
Did you experience any of the following? If so, to what degree?

General Discomfort *

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fatigue *

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Headache *

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Eyestrain *

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Difficulty focusing *

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Increased salivation *

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sweating *

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Nausea*  
0 1 2 3  
None  ○  ○  ○  Severe  

Difficulty concentrating*  
0 1 2 3  
None  ○  ○  ○  Severe  

Fullness of Head*  
0 1 2 3  
None  ○  ○  ○  Severe  

Blurred Vision*  
0 1 2 3  
None  ○  ○  ○  Severe  

Dizzy (eyes open)*  
0 1 2 3  
None  ○  ○  ○  Severe  

Dizzy (eyes closed)*  
0 1 2 3  
None  ○  ○  ○  Severe  

Vertigo*  
0 1 2 3  
None  ○  ○  ○  Severe  

Stomach awareness*  
0 1 2 3
### Prior Experiences with Gaming

How often do you play the following types of computer/video games?

<table>
<thead>
<tr>
<th>Type</th>
<th>None</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Severe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jump’n’Run*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strategy*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Puzzles*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adventures*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Racing*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sports*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fighting*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Never

Simulation *
Never

Education *
Never

Have you ever played the computer game "Interlocked"? *
Yes

If yes, how often do/did you play this game? *
Never

**Prior experience with stereoscopic entertainment**

Have you previously experienced....

<table>
<thead>
<tr>
<th></th>
<th>Never</th>
<th>1-3 times</th>
<th>4-9 times</th>
<th>10 or more times</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D cinema movie</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3D TV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stereoscopic 3D games</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Prior experience with 3D Software**

How often do you use the following types of 3D software?

CAD (e.g. SolidWorks, AutoCad) *
Never
Animation (e.g. Maya) *
Never

Molecular Visualization System (e.g. PyMol, jMol) *
Never

Geographic Information System *
Never

Other *
Never

What is your name? *
Please type the name you used to sign up for the study.

What is your age? *
Please type your age in numerical form.

Submit
Never submit passwords through Google Forms.

Powered by Google Docs
Report Abuse · Terms of Service · Additional Terms
A.2 zPuzzle Level Figures

Screenshot Images (from left to right):
Level 1, Level 2, Level 3, Level 4, Level 11
Appendix B

B1. Demo Source Code: HapticObject.cs

```csharp
using UnityEngine;
using System.Collections;

public class HapticObject : MonoBehaviour {

    public GameObject stylus;
    public GameObject core;
    public GameObject _oldHoverObject;
    public GameObject sphere;
    public GameObject cube;
    public bool isRolling;
    
    // Use this for initialization
    void Start () {
        stylus = GameObject.Find ("ZSStylus");
        core = GameObject.Find ("ZSCore");
        sphere = GameObject.Find ("Sphere");
        cube = GameObject.Find ("Cube");
        isRolling = false;
        _oldHoverObject = null;
    }

    // Update is called once per frame
    void Update () {
        roll ();
        hover ();
    }

    // 5 haptic settings with on, off, and repeat assignments
    void hapticSetting(int n) {
        int num = n;
        float on = 0;
        float off = 0;
        int repeat = 0;
        switch (num) {
            case 0:
                break;
            case 1:  // hover
                break;
        }
    }

} // end class
```

65
on = 0.1f;
off = 0.1f;
repeat = 0;
break;
case 2:                      // roll
  on = 0.1f;
  off = 0.1f;
  repeat = -1;
  break;

case 3:                  // sphere
  on = 0.15f;
  off = 0.15f;
  repeat = 0;
  break;

case 4:                  // cube
  on = 0.2f;
  off = 0.2f;
  repeat = 0;
  break;

case 5:
  on = 0.3f;
  off = 0.3f;
  repeat = 0;
  break;

}  
playVibration (on,off,repeat);

}  

// plays the stylus vibration, using core object
void playVibration(float on, float off, int repeat) {
  core.GetComponent<ZSCore>().SetStylusVibrationOnPeriod(on);
  core.GetComponent<ZSCore>().SetStylusVibrationOffPeriod(off);
  core.GetComponent<ZSCore>().SetStylusVibrationRepeatCount(repeat);
  core.GetComponent<ZSCore>().SetStylusVibrationEnabled(true);
  core.GetComponent<ZSCore>().StartStylusVibration();
}

// sphere object roll event
void roll() {
  Vector3 velocity = sphere.rigidbody.velocity;
  if((velocity[0] != 0 || velocity[2] != 0) && velocity[1] == 0){
    //update rolling boolean when rolling is initiated
if (isRolling == false) {
    isRolling = true;
    hapticSetting(2);
}
} else {
    if (isRolling == true)
        core.GetComponent<ZSCore>().StopStylusVibration();
    isRolling = false;
}

// stylus hovered over object
void hover() {
    GameObject hoverObject =stylus.GetComponent<ZSStylusSelector>().hoveredObject;
    if ((hoverObject != _oldHoverObject) && (hoverObject != null)){
        hapticSetting (0);
        _oldHoverObject = hoverObject;
    }
}

// object collision
void OnCollisionEnter(Collision collision) {
    if (this.gameObject.name.Equals("Sphere")) hapticSetting(3);
    if (this.gameObject.name.Equals("Cube")){
        // increase vibration strength with increased velocity when the cube
        // hits the floor object
        if (collision.gameObject.name.Equals("Floor")) {
            print (this.gameObject.rigidbody.velocity.y);
            if (this.gameObject.rigidbody.velocity.y <= -3 ||
            this.gameObject.rigidbody.velocity.y >= 1)
                hapticSetting(5);
            else if (this.gameObject.rigidbody.velocity.y <= -2 ||
            this.gameObject.rigidbody.velocity.y >= 0)
                hapticSetting(4);
            else if (this.gameObject.rigidbody.velocity.y <= 0)
                hapticSetting(3);
        }
    }
Appendix C

C1. zPuzzle Source Code: HapticObject.cs

using System.Collections;

public class HapticObject : MonoBehaviour {

    /* haptic settings are 0:short, 1:long, 2:hover, 3:post-selection */

    // setting var manages different haptic interaction configurations: 0-1: on collision only; 2: collision after selection;
    // 3: on hover and collision after selection; 4: on selected and collision after selection
    public int setting;
    public GameObject core;
    public GameObject pen;
    public GameObject _oldHoverObj;
    public bool isSliding;

    // Use this for initialization
    void Start () {
        core = GameObject.Find("ZSCore");
        pen = GameObject.Find("Pen");
        _oldHoverObj = null;
        isSliding = false;
    }

    // Update is called once per frame
    void Update () {
        if (setting == 3)
            hover ();
        if (setting == 4)
            selected ();
    }

    // Sets pen vibration on, off, and repeat settings
    public void playHapticSetting (int n) {
        int num = n;
        float on = 0;
        float off = 0;
    }
}
int repeat = 0;
switch(num){
    case 0:
        on = 0.05f;
        off = 0.1f;
        repeat = 0;
        break;
    case 1:
        on = 0.1f;
        off = 0.1f;
        repeat = 0;
        break;
    case 2:
        on = 0.035f;
        off = 0.035f;
        repeat = 0;
        break;
    case 3:
        on = 0.025f;
        off = 0.025f;
        repeat = 0;
        break;
    case 4:
        on = 0.025f;
        off = 0.1f;
        repeat = -1;
        break;
}
    playVibration(on, off, repeat);
}

// Enable stylus vibration
void playVibration(float on, float off, int repeat) {
    core.GetComponent<ZSCore>().SetStylusVibrationOnPeriod(on);
    core.GetComponent<ZSCore>().SetStylusVibrationOffPeriod(off);
    core.GetComponent<ZSCore>().SetStylusVibrationRepeatCount(repeat);
    core.GetComponent<ZSCore>().SetStylusVibrationEnabled(true);
    core.GetComponent<ZSCore>().StartStylusVibration();
}

// stylus hovered over object
void hover() {

}
GameObject hoverObj = pen.GetComponent<PenGrab>().hoverObject;
if ((hoverObj != _oldHoverObj) && (hoverObj != null)){
    playHapticSetting (3);
    _oldHoverObj = hoverObj;
}

// stylus selected piece
void selected() {
    bool justPressed = pen.GetComponent<PenGrab>().buttonJustPressed;
    GameObject selected = pen.GetComponent<PenGrab>().selectedObject;
    if (justPressed && selected != null)
        playHapticSetting (2);
}

// object collision
void OnCollisionEnter(Collision collision) {
    // play shorter vibration on collision
    if (setting == 0) {
        playHapticSetting (0);
    }
    // play longer vibration on collision
    if (setting == 1) {
        playHapticSetting (1);
    }
    // activates vibration only after the object has been selected
    // to reduce unnecessary vibrations
    if (setting == 2 || setting == 3 || setting == 4) {
        bool pressed = pen.GetComponent<PenGrab>().buttonPressed;
        //Debug.Log (buttonPressed);
        Vector3 velocity = this.gameObject.rigidbody.velocity;
        if (pressed && (velocity[0] > 0 || velocity[1] > 0 || velocity[2] > 0))
            playHapticSetting (0);
    }
}
Appendix D

D1. CREATIVE PROJECT SOURCE CODE: HAPTICOBJECT.cs

using UnityEngine;
using System.Collections;

public class HapticObject : MonoBehaviour {

    protected GameObject _stylusSelector;

    // Use this for initialization
    void Start () {
        _stylusSelector = GameObject.Find("ZSStylusSelector");
    }

    // Update is called once per frame
    void Update () {
    }

    // object collision
    void OnCollisionEnter(Collision collision) {
        if (collision.gameObject.name != "Pointer Proxy")
            _stylusSelector.GetComponent<StylusRumbler>().OnCollideBegin(this.gameObject);
    }

}
using UnityEngine;
using System.Collections;

public class LevelManager : MonoBehaviour {

    public GameObject[] playerPieces;

    void Awake() {
        DontDestroyOnLoad(this);
    }

    // Update is called once per frame
    void Update () {
    }

    protected void transition() {
        foreach (GameObject player in playerPieces) {
            player.GetComponent<PowerUp>().saveClips();
            player.GetComponent<PowerUp>().transitionPosition();
            player.GetComponent<PowerUp>().clipIndex = 0;
        }
    }

    protected void loadPlaybackControls() {
        GameObject _playbackControls = GameObject.Find("PlaybackControls");
        _playbackControls.GetComponent<PlaybackControls>().playerPieces = playerPieces;
    }

    void OnLevelWasLoaded(int level) {
        if (level == 1) {
            transition();
            loadPlaybackControls();
        }
    }
}
public class ObjectManager : MonoBehaviour {

    public GameObject[] basePieces;
    public GameObject[] playerPieces;
    public GameObject[] camPieces;
    public GameObject[] rodPieces;

    protected GameObject _stylusSelector;
    protected GameObject _oldHoverObject;

    // Use this for initialization
    void Start () {
        _stylusSelector = GameObject.Find("ZSStylusSelector");
    }

    // Update is called once per frame
    void Update () {
        hover ();
    }

    public bool originalPositions () {
        // Check to see if pieces are in original position.
        int numPiecesInOrigPos = 0;
        foreach (GameObject piece in basePieces) {
            if (piece.GetComponent<BlockPositions>().IsNearOriginalPos())
                numPiecesInOrigPos++;
        }

        if (numPiecesInOrigPos == basePieces.Length) {
            //print ("origPositions true");
            return true;
        } else {
            //print ("origPositions false");
            return false;
        }
    }
}
void hover() {
    // Initiate animation by checking if the hovered object is the crank and pieces are in original positions
    GameObject hoverObject = _stylusSelector.GetComponent<ZSStylusSelector>().HoverObject;
    if (hoverObject != null && hoverObject.name.Equals("Crank") && originalPositions() == true) {
        hoverObject.transform.Rotate(Vector3.right * Time.deltaTime * -25);
        GameObject.Find("Cams").transform.Rotate(Vector3.right * Time.deltaTime * -25);
        _stylusSelector.GetComponent<StylusRumbler>().OnHoverBegin(hoverObject);
        if (hoverObject != _oldHoverObject) {
            initAnimation();
        }
    } else if (_oldHoverObject != null && _oldHoverObject.name.Equals("Crank")) {
        foreach (GameObject player in playerPieces) {
            player.GetComponent<PowerUp>().stopAnimation();
        }
    }
    _oldHoverObject = hoverObject;
}

void initAnimation() {
    foreach (GameObject rod in rodPieces) {
        GameObject p = rod.GetComponent<Rods>().playerObj;
        int c = rod.GetComponent<Rods>().camPos;
        if (p != null && c != -1) {
            p.GetComponent<PowerUp>().beginAnimation(c);
        }
    }
}
using UnityEngine;
using System.Collections;
using System.Diagnostics;

public class PlaybackControls : MonoBehaviour {

    public GameObject[] playerPieces;
    private Stopwatch sw;

    // Use this for initialization
    void Start () {
        sw = new Stopwatch();
    }

    // Update is called once per frame
    void Update () {
        if (sw.IsRunning) {
            if (sw.ElapsedMilliseconds > 4000) {
                print("next");
                incrementSequence();
                sw.Reset();
                sw.Start();
            }
        }
    }

    void OnGUI () {
        // Make a background box
        GUI.Box(new Rect(10,10,100,90), "Player Menu");

        // Make the first button.
        if(GUILayout.Button(new Rect(20,40,80,20), "Start")) {
            sw.Start();
            this.GetComponent<AudioSource>().Play();
            print("start");
            foreach (GameObject p in playerPieces) {
                if (p.GetComponent<PowerUp>().clips != null)
                    p.GetComponent<PowerUp>().startAnimation(p.GetComponent<PowerUp>().clipIndex);
            }
        }
    }

}
if(GUI.Button(new Rect(20,70,80,20), "Stop")) {
    this.GetComponent<AudioSource>().Stop();
    foreach (GameObject p in playerPieces) {
        if (p.GetComponent<PowerUp>().clips != null)
            p.GetComponent<PowerUp>().stopAnimation();
        }
    sw.Stop();
}

void incrementSequence() {
    foreach (GameObject p in playerPieces) {
        p.GetComponent<PowerUp>().stopAnimation();
    }
    foreach (GameObject p in playerPieces) {
        p.GetComponent<PowerUp>().incrementIndex();
        if (p.GetComponent<PowerUp>().clipPlaying == null)
            p.GetComponent<PowerUp>().beginAnimation(p.GetComponent<PowerUp>().clipIndex);
    }
}
using UnityEngine;
using System.Collections;
using System.Collections.Generic;

global class PowerUp : MonoBehaviour {
    public AnimationClip[] clips;
    public string clipPlaying;
    protected Queue<AnimationClip> queue;
    private Vector3 originalPos;
    public int clipIndex;

    void Awake() {
        DontDestroyOnLoad(this);
    }

    // Use this for initialization
    void Start () {
        queue = new Queue<AnimationClip>();
        originalPos = transform.position;
        clipIndex = 0;
    }

    // Update is called once per frame
    void Update () {
    }

    public void beginAnimation(int index) {
        startAnimation(index);
        queue.Enqueue(clips[index]);
    }

    public void startAnimation(int index) {
        animation.wrapMode = WrapMode.Loop;
        animation.Play(clips[index].name);
        clipPlaying = clips[index].name;
    }

    public void stopAnimation() {
}
animation.Stop (clipPlaying);
clipPlaying = null;
}

public void saveClips() {
    if (queue.Count != 0)
        clips = queue.ToArray();
    else
        clips = null;
}

public void originalPosition() {
    Vector3 position = originalPos;
    this.gameObject.transform.position = position;
}

public void transitionPosition() {
    Vector3 position = new Vector3(originalPos.x, originalPos.y-0.1f, originalPos.z+0.05f);
    this.gameObject.transform.position = position;
}

public void incrementIndex() {
    print (clips.Length);
    print (clipIndex);
    if (clipIndex == clips.Length-1)
        clipIndex = 0;
    else
        clipIndex++;
}
using UnityEngine;
using System.Collections;

public class Rods : MonoBehaviour {

    public GameObject _objectmanager;
    public GameObject playerObj;
    public int camPos;

    protected GameObject[] camPcs;

    // Use this for initialization
    void Start () {
        _objectmanager = GameObject.Find("ObjectManager");
        camPcs = _objectmanager.GetComponent<ObjectManager>().camPieces;
    }

    // Update is called once per frame
    void Update () {
    }

    void OnTriggerEnter (Collider other) {
        foreach (GameObject go in _objectmanager.GetComponent<ObjectManager>().playerPieces) {
            if (other == go.collider)
                playerObj = go;
        }

        for (int i = 0; i < camPcs.Length; i++) {
            if (other == camPcs[i].collider)
                camPos = i;
        }
    }

    void OnTriggerExit (Collider other) {
        if (other == playerObj.collider)
            playerObj = null;
        if (other == camPcs[camPos].collider)
            camPos = -1;
    }
}
D7. CREATIVE PROJECT SOURCE CODE: STYLUSRUMBLER.cs

Modified by Heather Petrow, March 2014

using System;
using System.Collections.Generic;
using System.Linq;
using UnityEngine;
using zSpace.Common;

/// <summary>
/// Rumbles the stylus when objects are hovered, selected, or acted upon with a tool.
/// </summary>
public class StylusRumbler : ZSUMonoBehavior
{
    public int HoverIntensity = 0;
    public int UnhoverIntensity = 0;
    public int ToolOnIntensity = 0;
    public int ToolOffIntensity = 0;
    public int SnapOnIntensity = 0;
    public int SnapOffIntensity = 0;
    public int CollideIntensity = 0;

    protected ZSCore _core;
    protected ZSStylusSelector _stylusSelector;

    protected GameObject _oldHoverObject;
    protected HashSet<GameObject> _oldSelectedObjects = new HashSet<GameObject>();
    protected bool _oldIsToolActive = false;
    protected bool _wasSnapped = false;
    protected bool _isVibrating = false;

    protected override void OnScriptAwake()
    {
        base.OnScriptAwake();
    }
_core = GameObject.Find("ZSCore").GetComponent<ZSCore>();
_stylusSelector = GameObject.Find("ZSStylusSelector").GetComponent<ZSStylusSelector>();

protected override void OnScriptUpdate()
{
    base.OnScriptUpdate();

    GameObject hoverObject = _stylusSelector.HoverObject;
    if (hoverObject != _oldHoverObject)
    {
        if (hoverObject != null)
            OnHoverBegin(hoverObject);
        else
            OnHoverEnd(_oldHoverObject);
    }
    _oldHoverObject = hoverObject;

    HashSet<GameObject> selectedObjects = _stylusSelector.selectedObjects;

    if (_oldSelectedObjects.Except(selectedObjects).Count() != 0)
        OnSelectEnd(null);

    if (selectedObjects.Except(_oldSelectedObjects).Count() != 0)
        OnSelectBegin(null);

    _oldSelectedObjects = selectedObjects;

    bool isToolActive = _stylusSelector.ActiveStylus.Tool.IsOperating;

    if (isToolActive && !_oldIsToolActive)
        OnToolBegin(null);

    if (!isToolActive && _oldIsToolActive)
        OnToolEnd(null);

    _oldIsToolActive = isToolActive;
}
public void OnHoverBegin(GameObject go) { Shake(HoverIntensity); }

public void OnHoverEnd(GameObject go) { Shake(UnhoverIntensity); }

public void OnToolBegin(GameObject go) { Shake(ToolOnIntensity); }

public void OnToolEnd(GameObject go) { Shake(ToolOffIntensity); }

public void OnSnapBegin(GameObject go) { Shake(SnapOnIntensity); }

public void OnSnapEnd(GameObject go) { Shake(SnapOffIntensity); }

public void OnCollideBegin(GameObject go) { Shake(CollideIntensity); }

public void Shake(int intensity)
{
    if (!_isVibrating)
        return;

    float onPeriod = 0f;
    float offPeriod = 0f;
    int repeatCount = 0;

    switch (intensity)
    {
        case 0:
            break;
        case 1:
            onPeriod = 0.032f;
            offPeriod = 0.128f;
            repeatCount = 0;
            break;
        case 2:
            onPeriod = 0.032f;
            offPeriod = 0.064f;
            repeatCount = 0;
            break;
        case 3:
            onPeriod = 0.032f;
            offPeriod = 0.064f;
            repeatCount = 1;
            break;
        case 4:
            onPeriod = 0.064f;
            offPeriod = 0.128f;
            repeatCount = 0;
            break;
    }
case 5:
  onPeriod = 0.64f;
  offPeriod = 0.064f;
  repeatCount = 0;
  break;
case 6:
  onPeriod = 0.64f;
  offPeriod = 0.064f;
  repeatCount = 1;
  break;
case 7:
  onPeriod = 0.128f;
  offPeriod = 0.256f;
  repeatCount = 0;
  break;
case 8:
  onPeriod = 0.128f;
  offPeriod = 0.128f;
  repeatCount = 0;
  break;
case 9:
  onPeriod = 0.128f;
  offPeriod = 0.128f;
  repeatCount = 1;
  break;
default:
  onPeriod = 0.1f * (float)intensity;
  break;
}

_core.SetStylusVibrationOnPeriod(onPeriod);
_core.SetStylusVibrationOffPeriod(offPeriod);
_core.SetStylusVibrationRepeatCount(repeatCount);
_core.SetStylusVibrationEnabled(true);
_core.StartStylusVibration();
_isVibrating = true;

float waitTime = (onPeriod + offPeriod) * (repeatCount + 1);
StartCoroutine(Utility.Delay(waitTime, () => { _isVibrating = false; }));
}
References

http://www.autodesk.com/products/autodesk-maya/overview


   http://humansystems.arc.nasa.gov/groups/tlx/


