Physica: Interactive Tangible Physics Simulation based on Tabletop Mobile Robots Towards Explorable Physics Education

Jiatong Li
University of Chicago
Chicago, IL, USA
jtlee@uchicago.edu

Ryo Suzuki
University of Calgary
Calgary, Canada
ryo.suzuki@ucalgary.ca

Ken Nakagaki
University of Chicago
Chicago, IL, USA
knakagaki@uchicago.edu

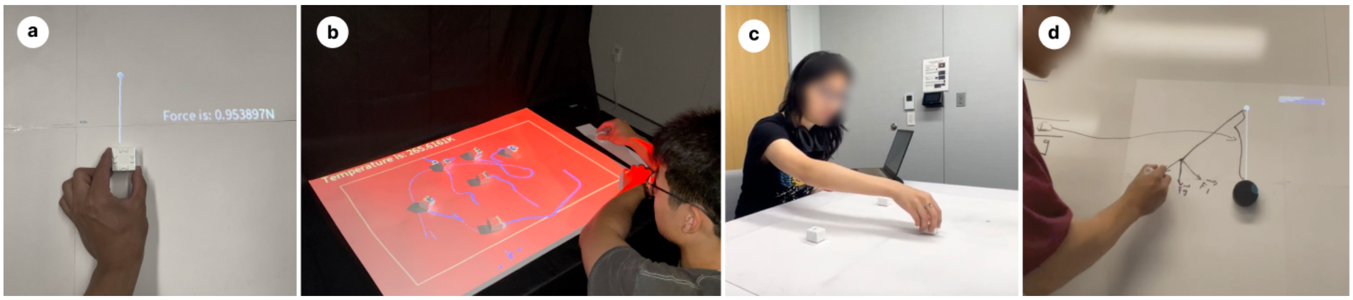


Figure 1: Overview of Physica (a) Physica enabling a user to feel the haptic force created by a virtual spring (b) Using a toio robot as a slider to heat up a gas chamber and observing how the molecules behave (c) Example of a user study participant dropping the toio on the interactive surface (d) Potential user scenario with Physica for teaching physics.

ABSTRACT

In this paper, we introduce Physica, a tangible physics simulation system and approach based on tabletop mobile robots. In Physica, each tabletop robot can physically represent distinct simulated objects that are controlled through an underlying physics simulation, such as gravitational force, molecular movement, and spring force. It aims to bring the benefits of tangible and haptic interaction into explorable physics learning, which was traditionally only available on screen-based interfaces. The system utilizes off-the-shelf mobile robots (Sony Toio) and an open-source physics simulation tool (Teilchen). Built on top of them, we implement the interaction software pipeline that consists of 1) an event detector to reflect tangible interaction by users, and 2) target speed control to minimize the gap between the robot motion and simulated moving objects. To present the potential for physics education, we demonstrate various application scenarios that illustrate different forms of learning using Physica. In our user study, we investigate the effect and the potential of our approach through a perception study and interviews with physics educators.

CCS CONCEPTS

• **Human-centered computing → Systems and tools for interaction design**.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
DIS '23, July 10–14, 2023, Pittsburgh, PA, USA
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9893-0/23/07…$15.00
https://doi.org/10.1145/3563657.3596037

KEYWORDS

Physics Simulation; Actuated Tangible UIs; Swarm UIs

ACM Reference Format:
Jiatong Li, Ryo Suzuki, and Ken Nakagaki. 2023. Physica: Interactive Tangible Physics Simulation based on Tabletop Mobile Robots Towards Explorable Physics Education. In *Designing Interactive Systems Conference (DIS '23), July 10–14, 2023, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3563657.3596037

1 INTRODUCTION

"Whence has it [the mind] all the materials of reason and knowledge? To this I answer, in one word, from experience" — John Locke [55]

In science education, *physical exploration* plays an important role in facilitating learning and understanding abstract concepts [47, 71, 72]. For example, children can easily learn abstract physics concepts, such as gravity and friction, by exploring and interacting with physical artifacts [9]. Such physical exploration allows users to do an interactive experiment in a collaborative space [67], which provides more engaging and memorable experiences than learning through textbooks or videos [9].

On the other hand, digital physics simulation tools have been widely studied and deployed using screen-based GUIs [3, 35, 98] to allow learners and teachers to understand physics concepts exploratively. Such a virtual 2D physics simulation environment allows users to manipulate different physics parameters in real-time, such as gravity, mass, spring, or damper, so that they can observe how these parameters affect the result of the simulation, which facilitates a deeper understanding of abstract concepts.

While these tools remain on screen, the HCI community has actively explored making these intangible and pixelated contents into a tangible medium [39]. In particular, recent HCI research has demonstrated the potential of tabletop swarm user interfaces as a way to embody digital objects [49, 63, 89], which have been gradually deployed as children toys, (e.g. toio [84]). Such a form of hardware (tabletop, hence moving in 2D plane) provides a unique opportunity to create interactive and tangible 2D digital physics simulations.

In this paper, we introduce Physica, a system that allows people to *tangibly* interact with digital physics simulation by leveraging tabletop swarm robots, or self-propelled Actuated Tangible UIs (A-TUIs.) By synchronously coupling the movement of each robot (Sony Toio [5]) with generic 2D physics simulation software (Teilchen [20]), the robots can tangibly embody a wide range of physics simulation behaviors, including gravity, spring, damper, and molecular movement. Since the system can control the robots' motion independently based on physics simulation, the users can perceive digital physics properties via haptic perception, such as gravity or virtual spring force (Figure 1A). With a swarm of tabletop robots, our approach enables to simulate rather complex models, such as particle movements to simulate a relationship between temperature and molecular behavior (Figure 1B). In our proposed system, users can not only observe the physics behaviors by watching them but also actively interact and engage with them tangibly (e.g. grab, hit, pull) to affect the digital simulation, while perceiving them.

While the idea of using A-TUIs for physics or science education is not new [43, 69], we contribute the following three key aspects beyond these prior works: 1) a system implementation to support for interaction with general-purpose physics simulation, 2) a design space for defining tangible interaction with tangible physics simulation and its application space in education, and 3) a user evaluation to understand how people perceive the system as well as interview for physics education expert.

Specifically, we first integrate a general-purpose physics simulation with a *target speed control* algorithm for the tabletop robots, which minimizes the gap between the robot's movement and the simulated one. We also develop an *event detection* method that can distinguish between various tangible interactions (e.g. hit, slide) and collision with physical obstacles, which is essential for our projection mapping setting, as opposed to the existing non-collaborative mobile AR setting (e.g., Sketched Reality [43]). We also contribute to the community by releasing this underlying simulation and controlling system as an open source software [1].

Second, we introduce a *design space* that consists of interaction design, graphical components, interaction framework, interactive surface configurations, and different forms of learning to convey concrete potential use scenarios of such tangible physics simulation tools broadly.

Finally, a *user evaluation* investigates how users perceive physics parameters (e.g. gravity, spring constant) conveyed via our approach. Through a perception study as well as an expert interview with physics instructors, we discuss the limitation and challenges of our current system towards deploying our system for educational applications.

In summary, this paper's contributions include:

- A general approach and design space for Physica to allow people to interact with digital physics simulation via tabletop mobile robots tangibly.
- An implementation of a proof-of-concept prototype, based on off-the-shelf tabletop robots (i.e. toio [5]) integrated with open source generic 2D physics simulation software that leverages an event detection algorithm, target speed control, and the existing screen-based physics simulation library [20].
- A demonstration of potential applications of Physica for physics education (learning, teaching, and tangible gaming).
- A user study to evaluate the capabilities of Physica, which illustrates and refines potential use case scenarios in physics education with physics educators.

2 RELATED WORK

Physica is built on prior studies, including interactive physics simulation, tabletop actuated TUIs, and interactive physics education systems.

2.1 Interactive Physics Simulation

Computational physics simulation has been extensively researched in the field of computer graphics (CG). Computer graphics researchers have used a variety of approaches to embody a range of physics behaviors. For example, they have developed algorithms to predict how virtual force affects different geometric shapes, such as mesh-based surfaces [16, 60], rigid/soft body [14, 32], and Finite Element Method (FEM) [18, 93]. On the other hand, particle-based physics simulation has been developed to simulate physics behaviors based on free-moving particles and their interactions. This has been applied to simulating fluids [21, 25, 29, 76], deformable bodies [19, 22, 54, 56], and granular materials [13]. Another approach is to use numerical methods to simulate the volume of the fluid [15, 34] and phase-field modeling [40].

While many of these works are originally developed for the realistic 3D rendering for movies and games, researchers have also repurposed these available physics simulation tools to physics education software [61, 83, 97]. These educational physics simulation tools often leverage 2D physics simulation, as opposed to complex 3D graphics, as it can simplify the illustration of the physics concepts. The simulations serve as complementary material to traditional instruction for students, allowing them to obtain better grades compared to students who learn physics through traditional methods only [59].

Beyond screen-based simulation, many HCI and haptics researchers have also explored ways to convey digital physics simulations in haptic and tangible ways [26, 39]. For example, Hapkit uses a 3D printed structure as a proxy to let users experience the *stiffness property*, and it is widely used in MOOC and college classrooms [58]. By using a pin-based display, Materiable is able to render three properties (*flexibility*, *elasticity*, *viscousity*) via users' direct touch interaction. Similarly, inFORCE [62] also leveraged pin-based displays to create bi-directional force-based interactions with the help of 3D volumetric information. Haptic feedback is also widely used in software settings. For example, HapticTouch [50] can enhance the interaction experience by letting users directly feel the stiffness and friction from the software by implementing a DIY haptic device.

[1] https://github.com/AxLab-UofC/Physica_DIS_2023

Our approach contributes to this line of research by introducing a novel approach to tangibly and haptically represent underlying physics properties by leveraging wheeled A-TUIs. In contrast to other approaches mentioned above, where the hardware systems are either grounded or constrained with their locomotion capability, our system can provide unique interactions and affordances with free-moving tabletop robots, which is not explored in the literature.

2.2 Actuated TUIs and Swarm UIs

In recent years, the HCI research community has shown a growing interest in self-propelled modular robots that can collectively produce programmable behaviors. This line of research was initiated by early exploration of A-TUIs, which use electromagnetic arrays to control passive magnetic objects on a tabletop surface [70]. This approach allows bi-directional user interaction between digital simulation and actuated objects [74]. Preliminary development of a wheeled-based system for actuated tangibles was also explored with Curlybot [30].

Within the past decade, Zooids [49] introduced the concept of Swarm User Interfaces (Swarm UIs), where a swarm of self-propelled robots can physically render digital information, providing rich tangible interactions and affordances. Accordingly, other researchers explored the self-assembly property of Swarm UIs [80, 81], gesture interaction [45], interactions with Virtual Reality (VR) [86, 90, 99], modular mechanical attachments [63], remote demonstration/teaching [53, 92], and education [33, 69]. Using each individual robot in a swarm, researchers can create enriching bi-directional haptic interactions between the physical world and the virtual world.

Most recently, Sketched Reality introduced a method of using sketching interactions in a tablet-based Augmented Reality (AR) system to control and interact with Swarm UIs [43]. Motivated by the recent advances in AR and A-TUIs [87], the authors focused their work on developing bi-directional interactions between virtual sketches and A-TUIs via AR. Within these demos, they have employed preliminary physics simulations to define the actuating relationship between virtual sketches and actuated objects, which partially overlaps with our approach. However, Physica, in contrast, focuses on developing an interactive system that embodies generic 2D physics simulation. To approach this specific goal, we introduce a control algorithm for the digital simulation to identify and reflect rich tangible and physical interaction by users (e.g. hit, grab) as real-time feedback. With this focus, our work further evaluated how users interpret and interact with the simulation via the robots. Additionally, our system applied projection as a geometrical component to convey virtual physics properties (e.g. force, speed, etc) directly on the interactive surface, which provides a more natural interaction when compared to tablet-based AR, demanding users to hold the device and switch their attention between the robots and the tablet.

2.3 Physics Education with TUI/AR

Since the TUI concept has been proposed [39], many fellow researchers have found its value in educational purposes, especially with young users. While modern and commercial GUI-based tools are dominant in computer-supported education [2–4, 98], cognitive psychology theories emphasize the importance of directly manipulating task-appropriated physical objects on children's cognition [38]. Lev Vygotsky also discusses how tangibility has a positive effect on children's psychological function [96].

Due to these advantages of TUI in education, different educational applications using TUIs have been developed for chemistry [27], drawing [82], music [42, 57], circuit design [1], optics [31], and general exploratory learning [91]. With the recent emergence of actuated TUIs, such educational applications were further advanced for kinetic robotic motion design [78], programmable drawing [30], and educational AR [94, 95]. Brown et al. mentioned in their survey [17] that actuated modular robots can be used to represent the configuration of molecules [73]. Actuated pin-tables are used to teach terrain analysis (e.g. Relief [52], XenoVision Mark III [24], Recompose [51], etc). HapKit [58] is an actuated haptic tool for students to program physical haptic properties that allow people to learn how programmed physics equations affect the haptic feedback generated on the device.

As reviewed in [41], there have been broad sets of research developments and studies in the educational applications of swarm robots and tabletop robots [12, 33, 66, 69]. The studies explored the education topics in drawing shapes and letters [66], storytelling [69], gaming [33], and robot theater [12]. These studies verified the effect of tabletop mobile robots in education topics, and Physica seeks to develop a system that shall contribute to the line of above research for developing interactive physics education tools.

As for research focused on physics-based learning, AR is a heavily-employed and studied approach [77]. For example, Suzuki et al. demonstrated the use of AR to augment physics experiments by connecting the physical world with AR real-time visualization [88]. Paper Trail designed an immersive paper to create an educational AR experience [79].

In contrast, by developing bi-directional tangible interaction between users and a digital simulation via tabletop robots, Physica aims to help students explore physics concepts. While we have not yet evaluated our work in an actual classroom setup, in our paper, the potential of this approach is preliminarily studied through user interviews with physics educators and university students.

3 PHYSICA

This section introduces the basic setup and the design space of Physica as a generalizable physics simulation tool. To illustrate the potential use case for educational purposes, we also broadly present different forms of learning to discuss how students and physics educators can benefit from using Physica.

3.1 General Design Space of Physica

As a generalizable design space of Physica, we introduce 1) the basic interaction setup, 2) the interaction framework, 3) graphical components, 4) interaction surface configuration, and 5) interaction design (see Figure 2).

1) Interaction Setup: The overall setup utilizes tabletop wheeled robots, an interactive surface, graphical components, and a computer (managing physics simulation and interaction events).

2) Interaction Framework: We aim to render physics simulation with tabletop self-propelled actuated TUIs. Users interact with

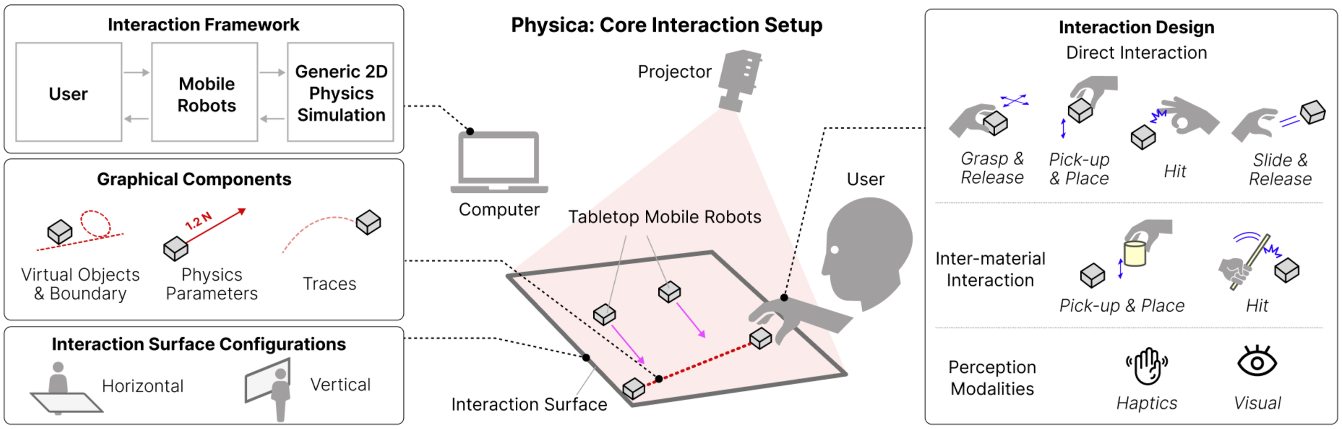


Figure 2: Design Space of Physica

generic 2D physics simulation algorithms (Figure 2) via swarm robots. This framework allows for bi-directional interaction where users can affect the simulation, and the simulation can give feedback to users via swarm robots. This framework was inspired by Materiable's framework [65], but we apply it to tabletop swarm robots.

3) Graphical Components: To augment the user experience and provide a multimodal interaction, we use a projector (Figure 2) to map graphical components (e.g. tabletop robot icon, object position, velocity direction, etc) onto the interactive surface as visual guidance. Users can choose the graphical components they want to see on the mat by selecting the corresponding GUI element.

4) Interactive Surface Configurations: Physica can be deployed to two types of interaction surfaces: *horizontal* and *vertical*. The horizontal surface can be easily set up on the table, and the vertical surface can be attached to a wall or a whiteboard to provide more diverse spacial configurations. Magnets-embedding robots can be employed for on a vertical ferromagnetic surface [46].

5) Interaction Design: We also explore three possible interaction design modalities to illustrate how users can interact with tabletop robots and the underlying physics simulation. These modalities include 1) direct interaction, 2) inter-material interaction, and 3) perception modalities. Such interactivity becomes a guide for developing event detection modules in our software.

5-1) Direct Interaction: First, the user can directly interact with these robots through various tangible interactions. *Grab & Release* typically occurs when a user wants to grab the robot in order to change its spatial location and release it in order to observe how the tabletop robot will behave. For example, a user can grab the robot, which represents the attached object on the spring, and feel the actuated force created by the motor. Meanwhile, users can also change the spring constant and observe how it affects the actuated force. *Pick Up & Place* allows the user to pick up the robot and place it anywhere on the interactive surface, changing its location and observing how the motion changes based on the physics simulation the tabletop robot is in. *Hit* can be defined as a quick and strong input exerted by the user on the robots. For example, users can use their fingers to flick the tabletop robot, imparting an initial velocity to the virtual object. Compared to a *Hit*, the time it takes before a user releases the object is much longer for *Slide*. A user can hold and then release the robots to observe their trajectory on the interaction surface.

5-2) Inter-Material Interaction: Second, the system can also leverage inter-material interaction, instead of using users' hands as a direct input source. In this context, *Pick Up & Place* allows a user to use another real-world object (e.g., passive blocks, pencils) to in the field, and *Hit* lets the tabletop robots to collide with these objects. This interaction opens up the possibility of introducing props to facilitate the exploration of concepts in physics. Prior researchers in A-TUIs explored inter-material interaction [28], a method where passive objects interact with active objects.

5-3) Perception Modalities: Lastly in interaction design, we can also categorize the way users can *Interpret and Perceive* using an individual tabletop robot. We can create *Haptic Sensations* in our physics simulation. For example, a user can grab an object connected to a virtual spring and feel the haptic force generated by the spring. Force is the primary effect that can be provided through the Physica system. Another perception we offer is *Visual Perception*, which comes from two sources: 1) the movement of the tabletop robots on the tracking mat providing a visual cue explaining how the physical system will evolve over time, 2) we overlay the graphical components created by the projector (Figure 2) on the interaction surface so that users can choose the information they want to inspect using the GUI. The additional visualizations further enrich the user experience.

3.2 Supported Physics Simulation Types

Physica also enable the following five supported physics simulation: 1) Gravity, 2) Collision/Bounce on Virtual Boundary and Objects, 3) Projectile Motion, 4) Pendulum, and 5) Spring-Mass System. Figure 3 illustrates how the robots can behave based on these physics simulation types. To decide on the supported physics simulation, we first broadly explored possible simulations that are available in the existing simulation software, then narrowed it down to these five simulation types based on the following two considerations: 1) simulations that best utilize the wheeled robots' capability of being freely able to move on a 2D plane, and 2) simulations that relate

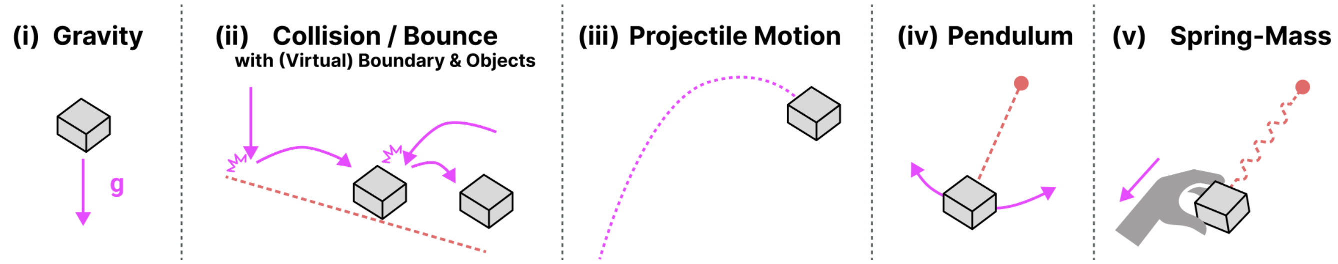


Figure 3: Design space of supported physics simulation in Physica: (i) Gravity: an object is dropped under the influence of a gravitational force on the interactive surface (ii) Bouncing on a Virtual Boundary: an object bounces on a virtual plane defined by two robots (iii) Projectile Motion: an object is experiencing projectile motion (iv) Pendulum: a simple pendulum simulation (v) Spring-Mass System: an oscillating object is connected to a virtual spring.

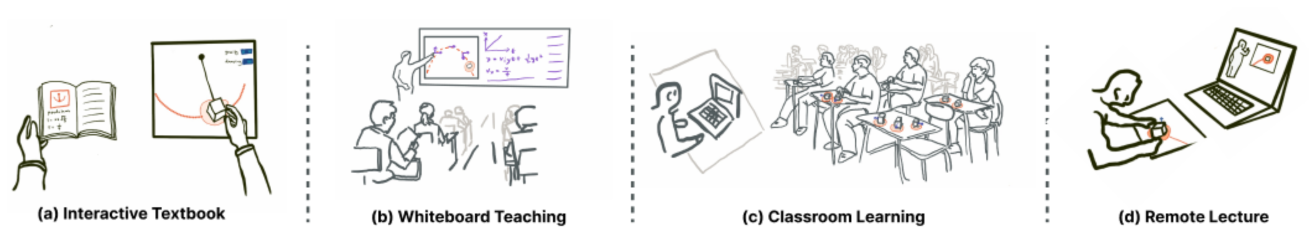


Figure 4: Skeches of proposed learning scenarios for Physica

to physics concepts taught in standard curriculum [3, 98]. These primitive simulations can be combined to produce and represent complex models, such as a swarm of robots that can represent particle behaviors based on the collision and bounce (e.g., the robot collides with the other robots and bounding box each other to simulate molecular movement in a chamber, as seen in Figure 1B). These five simulations preliminarily help us to explore, demonstrate and evaluate Physica, while further simulations could be developed, as later discussed in the future work section.

3.3 Forms of Learning and User Scenarios

We present and illustrate potential future scenarios of ways in which Physica could be beneficial while teaching and learning physics. By introducing different types of scenarios that employ Physica, we share the broad potential of our prototype for various purposes (Figure 4). These illustrated scenarios are also discussed in the evaluation through user interviews in our paper.

1) Interactive Textbook that Supports Students' Self-Learning: As a self-learning tool, Physica could work with textbooks so that students can use the tabletop robots as a proxy to activate the corresponding tangible physics simulation and learn and explore the concept on demand (Figure 4A).

2) Whiteboard Teaching that Supports Demonstrations on a Whiteboard: Our second scenario illustrates, in turn, how teachers could use Physica as a teaching aid in a classroom lecture setting. As the robots can also move on a vertical surface with the attached magnets, the system can be operated on a whiteboard-like surface so that instructors can directly annotate on the whiteboard and provide rich demonstrations while teaching by placing the robots directly on the sketch (Figure 4B).

3) Classroom Learning that Supports Interactive Learning During Lectures: In the third scenario, students use Physica during a lecture in a classroom (Figure 4C). Physica's interactive and tangible features allow students to explore physics concepts on their personal desktops. The tool also facilitates collaborative learning, where students can work together to deepen their understanding using these robots in a group.

4) Remote Lecture that Supports Long Distance Education: Our fourth scenario illustrates how Physica can be used in remote teaching/learning for students and educators (Figure 4D). Especially with the increased demand for remote lectures after the COVID-19 pandemic, we believe that Physica could engage remote students with tangible experiences even if they are not in the classroom. This use case could encourage remote collaboration as both students and instructors (or with other students) can interact with the same simulation from the devices in their possession.

4 IMPLEMENTATION

4.1 Overall System Design Based on Toio Robots

The overall system consists of toio robots, a computer running software, a toio mat, and a projector. Toio robots are off-the-shelf robotic toys developed by *Sony Interactive Entertainment* [84]. The robots are equipped with two motorized wheels and are controlled wirelessly via Bluetooth Low Energy (BLE), often employed in Swarm UI / A-TUI research in HCI [43, 63, 64, 89]. As depicted on the left of Figure 5, we employed Rust-based toio I/O server code [11] to communicate with the toio robots from a MacOS laptop computer. Our custom software, built with Processing, handles the core interactive functionality that sends and receives toio's position and orientation information from the Rust toio I/O server software.

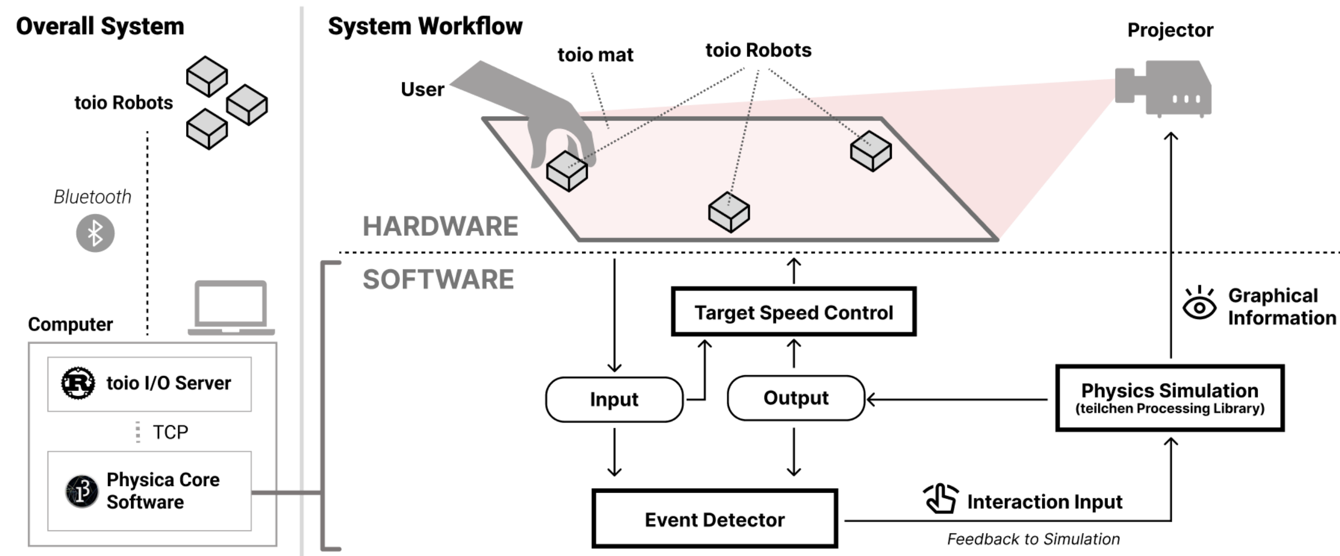


Figure 5: (Left) Overall System that describes the implemented architecture (Right) System Workflow that presents how the robots interactivity is supported by physics simulation, event detector, and target speed control.

The toio robots are equipped with a bottom-facing camera to localize them in 2D coordinates on the mat [5]. This 2D tracking capability allowed us to build a closed-loop control system for autonomously navigating the robots and detecting a variety of interactions by users. Toio has its coordinate system [5], where 1 toio pixel on the mat equates to 1.4mm. We refer to this pixel unit in the digital simulation tool, and the event detection threshold described in later sections. We also modified the toio robots by attaching two disc-shaped neodymium magnets (6 mm diameter, 1.5mm thickness) underneath the body to increase the frictional force with the metal sheet, a known technique employed in prior research [46, 63]. The dimension of each toio robot is 3.2cm x 3.2cm x 2.5cm. With this setup, we were able to stably control up to eight toio robots. The maximum speed of a toio is 35cm/s and can generate a torque of up to 1.1N on a horizontal toio mat surface, thanks to the added magnets.

4.2 Control Workflow

The software of our system can mainly be divided into three sections: a physics simulation, a target speed control, and an event detector. The relationship of these three components are depicted in Figure 5 right. The physics simulation is core to simulate the virtual physics environment for Physica. The output from the physics simulation is passed to the target speed control method, which dynamically controls the toio motion to match the simulated target moving objects in closed-loop control. The event detector was developed to detect interaction events by users by identifying the gap between the output and input for toio's 2D position — and giving feedback to the digital physics simulation for users to affect the digital simulation. This overall control framework is built on [65], but for self-propelled tabletop robots. Below, we briefly describe each of the methods, which are detailed in the appendix.

4.2.1 Physics Simulation. We employ an existing 2D physics simulation library in Processing, named teilchen [20]. teilchen provides a collection of physics concepts, such as force, constraints, and behaviors, for modeling interactive virtual physics systems. Within teilchen's library, different properties, including gravity, spring, and bouncing coefficient, are adjustable/tunable dynamically. This was utilized in our study. This simulation, combined with target speed control and event detector, was the key to developing a tangible interactive physics simulation that can replicate the speed and force of virtual simulation via the toio robots.

4.2.2 Target Speed Control. To render physics simulations using toio robots, we implemented a speed control function that navigates toio robots to follow the virtual object in the simulation. By doing so, the system takes both the target position and the target velocity of the simulated object together with toio's actual position into account in order to minimize the gap between the dynamic moving target (simulated objects) and toio robots. Such methods for dynamic target coordinate and velocity control for wheeled robots are well-explored topics in the robotics research domain [37, 68], but have not been developed in the context of building interactive haptic systems based on self-propelled A-TUIs. While Sketched Reality [43] is an A-TUI system that preliminary incorporated digital physics simulations, such a velocity-based robot control was not applied to minimize the gap, as the purpose of the system was less about conveying physics properties to users. The implementation details of our target-speed control can be found in A.1.

To present a preliminary performance of the target speed control, Figure 6 illustrates example paths of simulated object motion (target paths) and the robot's paths (actual paths), comparing with and without the target speed control. For the non-target speed control, the *aimCube()* function in the publicly available toio targeting control tool was used [11]. For this setup, the simulated

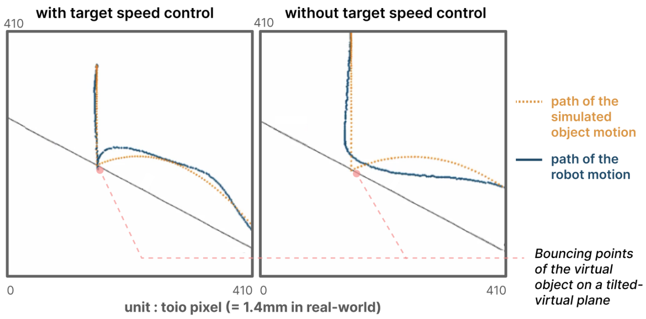


Figure 6: Paths of simulated target object motion, and actual robot motion, comparing two different robot control method, Left: with target speed control (incorporating both target coordinate and velocity to control a robot), and Right: without target speed control (incorporating only target coordinate to control a robot).

virtual environment was set up so that a virtual object fell from above with gravity and bounced on the tilted-virtual plane. The yellow-dotted lines, representing the simulated object motion, show a sharp bouncing curve in both records, but for the blue lines, representing the actual robot paths, only the one with target speed control (left) makes a sharp bouncing curve to better match the simulated path. With this example, we also collected average 2D positional gaps (errors in the distance) between the simulated object and robot during the simulated motion in action. The ones with target speed control resulted in 18.3 pixels/25.62mm average gap, and, in comparison, the ones without the target speed control resulted in 52.8 pixels/73.92mm errors. Therefore, the target speed control contributed to minimizing the gap between the robot and the simulation to better represent the physics simulation compared to the existing targeting control of toio. Given the advanced robotics research in closed-loop robot targeting control methods and theory, future Physica systems could further employ them [36, 48].

4.2.3 Event Detector. The event detector classifies users' direct interactions with the robots, so that they can be reflected in the digital physics simulation. This detector was able to detect different interaction methods that are covered in Figure 2 right. In the event detector algorithm, we have built a general flow to detect and classify input interactions based on *inputs* and custom *thresholds*, and to provide *feedback* to the simulation.

The overall flow of the event detector is as follows; (1) the system receives *inputs*, which are sensing parameters from toio robots (including the 2D position on the mat, and calculated velocity based on a series of positions over time). (2) By comparing the *inputs* with predefined *thresholds*, the event detector passes the classified interaction input as *feedback* to the digital physics simulation.

In appendix A.2, we described the detailed thresholds and setting to detect and classify each event/interaction.

4.3 Projection of Graphical Components

We use the Keystone [100] Processing library to render the projection mapping. By matching the four corners of the projection with the corners of the toio mat, Keystone is able to calibrate the distortion of the projected image (Figure 7) so that the graphical components are properly rendered regardless of the projector's position and orientation.

4.4 Customizable Physics Parameters with a GUI

To customize a variety of physics parameters (e.g. the length of the spring, gravitational acceleration, etc) (Figure 7), we implemented a GUI system on the laptop screen. There are three types of parameters users can control: (1) physics property control (e.g. amount of gravity, friction, spring, etc), (2) checkboxes for overlaying graphical components (e.g. force and speed vectors, object traces), (3) checkboxes for activating interaction inputs from the user (Figure 7). These flexible GUI controllers help users construct and customize to extract and visualize certain abstract physics parameters, as well as to tune physics properties to test different objects' behavior rendered on the robots exploratively.

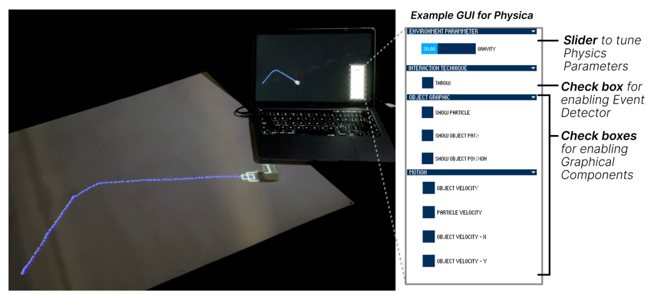


Figure 7: Physica GUI: Users can customize features such as physics parameters, interaction techniques, and graphical components.

5 APPLICATIONS

5.1 Demonstrating Physics Concepts

Firstly, Physica can help students quickly construct physics simulations and inspect how each physics parameter will affect the object. Such experimental aspects cannot be offered in a traditional static textbook. Studies have shown that applying simulation-based learning methods in subjects such as physics and chemistry can greatly increase students' level of engagement and self-satisfaction [10]. In Figure 8 and Figure 9, we demonstrate a number of physics simulations that commonly appear in the high school physics curriculum. By changing the physics parameters using our projected GUI, students can instantly launch simple physics experiments and start exploring. Moreover, with the help of the projector, Physica can augment the tangible simulation with geometrical components to better facilitate the learning experience.

5.1.1 Gravity. This example demonstrates how an object can move under the influence of gravitational acceleration (Figure 8A). A user can *pick-up & place* the object anywhere on the tracking mat to see what the path of the object will be under the influence of acceleration. Moreover, users can also change the magnitude of the gravitational acceleration and inspect what the relationship between acceleration and velocity is.

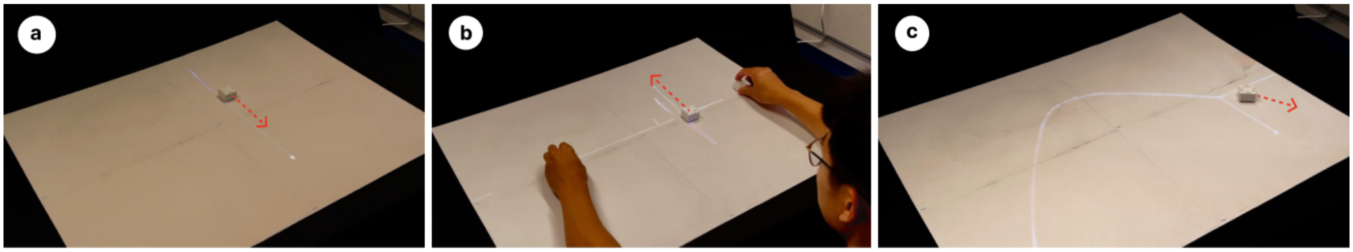


Figure 8: Example Demonstrations of physics simulations: (a) Gravity: an object is dropping under the influence of gravitational force (b) Bouncing on Virtual Boundary: a user is using the adjustable anchor to dynamically change the angle of a tilted plane, while an object is bouncing on such surface (c) Projectile Motion: a projectile is fired and displays a ballistics trajectory.

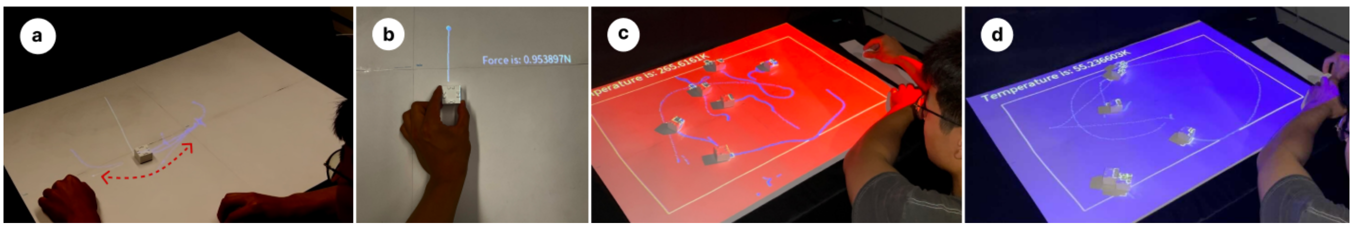


Figure 9: Example Demonstrations of physics simulations: (a) Pendulum: a user is interacting with a pendulum (b) Spring-Mass system: an object is creating a haptic force due to the attached spring (c) Heating Molecules in a Gas Chamber: a user is heating up molecules in a gas chamber and the kinetic energy of each individual molecule is increasing (d) Cooling Molecules in a Gas Chamber: a user is cooling down molecules in a gas chamber and the kinetic energy of each individual molecule is decreasing

5.1.2 Bouncing on Virtual Boundary. This application, shown in Figure 8B, demonstrates the concept of coefficient of restitution (COR) during the collision. In the picture, we can see three toios. Two of them will serve as movable anchors which define the orientation of the virtual boundary. The last toio represents the object which will bounce on the boundary. A user can *pick-up & place* the object anywhere above the virtual boundary. After placing it, the user can *grab & release* the two anchors and change their location in order to create different bouncing angle. The resulting simulation is similar to an object bouncing on a trampoline. By doing so, we create an illusion that a virtual boundary is physically affecting the object's movement.

5.1.3 Projectile Motion. Another physics simulation we built is projectile motion which is also a very common topic in middle school/high school physics textbook. Projectile motion describes a special form of motion experienced by a moving object (i.e. the projectile) under the influence of gravity. The curved path is a parabola (Figure 8C), but it also can be a near straight line if the user *throws* or *hits* vertically upward.

5.1.4 Pendulum. Another example we built is a pendulum simulation (Figure 9A). We suspend a weight under the anchor (or pivot) and connect the weight and the anchor with a virtual rope. Users can *grab & release* the weight and examine how the amplitude changes over time. Moreover, the user also can change the gravitational acceleration and observe how the property will affect the pendulum's period.

5.1.5 Spring-Mass System. We also implemented a spring-mass system as indicated in Figure 9B. Similar to the Pendulum simulation, we defined a fixed anchor which connects with an object. Based on Hooke's law,

$$ForceOutput = -SpringConstant * Displacement$$

We implemented the system such that a user can *grab & release* the attached object and observe how it oscillates under the influence of the spring.

5.1.6 Heating/Cooling Molecules in a Gas Chamber. Lastly, we also implemented a molecular gas chamber simulation. Users can use another toio robot as slider to adjust the temperature of the gas chamber. When the chamber is heating up (Figure 9C) as indicated by the red background color, the gas particles will gain kinetic energy and start bouncing inside the chamber. On the contrary, if the temperature cools down (Figure 9D), each molecule (i.e. toio robot) will lose kinetic energy and move slower. The movement of each individual toio robot provides users with an intuitive understanding of what the temperature means, and the tangible slider presents a more enriching and interactive experience.

5.2 Tangible Gaming

The importance of gaming in helping young children acquire new knowledge and develop new skills [75] has been studied extensively. So far, we have discussed different ways of rendering physics simulations via A-TUIs. What if we can incorporate physics concepts in a gaming environment? Figure 10C shows our attempts at implementing a pinball game with Physica system. Users can use real-world

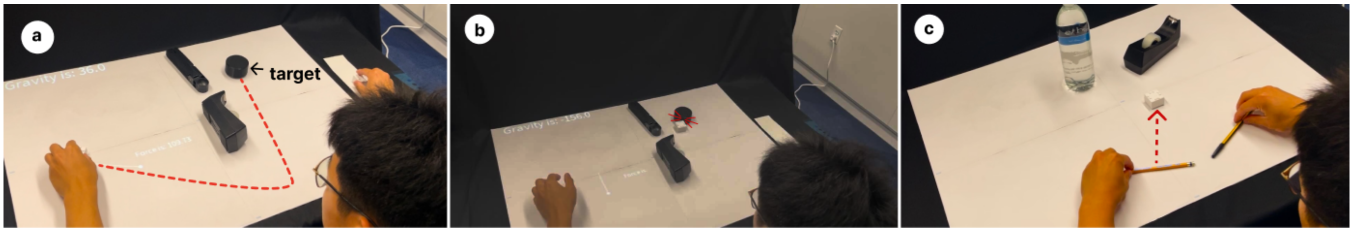


Figure 10: (a) - (b) Tangible Gaming: Dynamic Gravity Slingshot: a user is using a tangible slider (i.e. a toio robot) to dynamically change the gravity of the physical system so that he/she can guide the fired projectile to avoid the obstacles and eventually hit the target (c) Pinball: users can turn any flat real-world objects into flippers and use it to actuate the dropping object (i.e. toio). Similarly, a stapler can serve as an obstacle in the playfield so that users can instantly use daily objects to set up a gaming environment.

object (e.g. pencil) as the flippers to actuate the dropping object. In the pinball machine's playfield, a user can use real-world objects (e.g. a stapler) as obstacles to block the object's path. By doing so, we create an easy-setup gaming environment using daily objects and demonstrate how virtual objects can interact with physical object.

Inspired by the award-winning game Angry Birds [7], Figure 10A shows a preliminary implementation of using a slingshot to shoot target objects. In the simulation, a user can *grab & release* the projectile and feel the haptic feedback created by the virtual spring. We also designed a tangible slider (e.g. a toio robot) which enables the user to change the gravity by sliding the toio up or down. By doing so, users can have total control over the object and guide the projectile towards the final target (Figure 10B). Such interaction is impossible in reality since one cannot change the gravity of the Earth.

6 USER EVALUATION

To examine our developed prototype and approach of Physica, we conducted two user evaluations: **1) a perception study with 15 participants** and **2) an expert interview with three physics instructors**. The **perception study** was conducted to understand how users interact and interpret physics properties conveyed via Physica. The **expert interview** was conducted to learn professional physics instructors' perspectives and thoughts about Physica's demonstration and future potential in teaching physics in real-world practical classrooms and studies. Both studies were approved by our institution's IRB committee.

6.1 Perception Study

6.1.1 Procedure. We conducted a preliminary user study to examine how users interpret physics simulations presented via the Physica system (Figure 11). We recruited a total of 15 participants (9 male, 5 female, and 1 not-identified) within our institution, from the age of 18 to 50 (average of 24.7), with no perception disability. For this study, Physics education backgrounds or ages were not considered in the recruitment process of the participants, because the study's purpose was to understand Physica's general capability in conveying people with basic physics simulation properties. Hence, this preliminary study was not intended to evaluate physics education feasibility and effects, and future evaluation should address them.

In the study, participants were asked to interact with and perceive rendered physics simulations with Physica. Participants interacted with three types of Physica demos: *gravity*, *spring constant*, and *coefficient of restitution (COR)*. For each physics property, we changed five different magnitudes and asked users to score their perceived physics property on a scale from 0 to 10. The set of physics parameters and each magnitude, controlled in the teilchen digital simulation library [20], can be found in Figure 11 right. The range of five magnitudes for each property was decided based on a pilot study among the authors. For each physics property, we first demonstrated users with the minimum and maximum (mentioned they are equivalent to 0 and 10, respectively) to establish a reference. Then we shuffle the values and test each magnitude twice – hence they perceived and rated ten total for each property. After they completed all studies, we interviewed them with questions, including the general experience reflection, and the most realistic property among the three. The magnitude ratings were collected on a laptop digitally, and we made observations with recorded videos and extracted participants' actions and comments, as we asked them to think aloud.

6.1.2 Result.
Perception Rating: As a result, Figure 12 a-c shows the average scores of participants' responses for the five different magnitudes for gravity, spring constant, and COR. Overall, via direct tangible interactions, participants rated the perceived properties according to the set parameters in the digital simulation. Among all the tested physics parameters, COR was the most successful physics parameter in terms of both perception magnitude (consistent trend across the magnitude range as in Figure 12c) and realism (Figure 12d). As for the gravity and spring constant simulation, although the overall perceived magnitudes correspond to the actual trend, the graphs exhibit non-linear relationships between participants' scores and the actual simulation parameters, especially in the larger magnitude where the curve got less steep. We assume that this was due to toio's hardware limitation of maximum speed to convey higher magnitudes in gravity and spring appropriately.

We also asked participants to 'think aloud' to describe the properties they were interacting. In the COR simulation, participants

Evaluated Physics Parameters	Gravity	Spring Constant	Coefficient of Restitution (COR)
Teilchen Function	Gravity.force().set(float[x], float[y]) *only y adjusted*	Spring.strength (float)	PlaneDeflector.coefficientofrestitution(float)
Values for each Variable	10, 40, 70, 100, 130	1, 2.25, 3.5, 4.75, 6	0.3, 0.6, 0.9, 1.2, 1.5
Scaled Value for Each Variable	0, 2.5, 5, 7.5, 10	0, 2.5, 5, 7.5, 10	0, 2.5, 5, 7.5, 10

Figure 11: User Study information: (Left) a participant interacting with the Physica system in a user study. (Right) Tables of physics properties and magnitude used in the user study.

described the experience as P2 *"bouncing ball"* and P6 *"trampoline-like"* when the COR value is high (elastic collision), while P1 mentioned *"feels like dumbbell drops on the floor"* when experiencing a low COR value (inelastic collision). This shows they could associate the robots' motion with real-world material properties and physics-related phenomena.

Observation of User Interaction: Through the observation of participants' interactions, we found unexpected ways in which people interacted with the system. For example, P6 used their fingers to poke the object with the aim of compressing the spring to feel the resistive force (Figure 13a). P5 was attempting to *slide* in counter to virtual spring (Figure 13d). We also observed P5 and P6 used their arm to push the robots. These observations of unexpected interactions indicate the rich affordance of our system, allowing for free interaction and future development opportunities in our event detector algorithms.

Overall Experience: As for the post-interview about their experience, 13 out of 15 participants provided positive remarks that they "enjoyed" and had "fun" or "great feeling" interacting with the system. 14 out of 15 participants saw a strong potential in Physica for education, as one of them mentioned *"It's cool to see how Physica can help students feel abstract physics concept"*. One participant mentioned that the system *"was great for feeling different comparison relationships [physics parameters] (which one is greater/smaller)"*. In terms of the opportunity for future improvement, P4 mentioned *"the robots sometimes glitch, which could lead to unnatural behavior"*. Also, one participant commented critically *"the motion and interaction are far less natural than playing with a normal spring"*. While these two were the only negative comments on our interactive simulation, they indicate future improvements and challenges to improve our interactive system. Participants successfully perceived differences in physics property magnitudes, 'how we can improve the realism' is a key challenge in design and implementation.

6.2 Expert Interview

6.2.1 Procedure. In addition to learning general novices' perceptions and experiences towards our system, to collect initial insights from expert physics education practitioners, we recruited three experts who specialize in either teaching experimental physics or developing physics education tools for academic courses at our institution. They had an average experience of 16 years teaching physics. We structured the interview as follows: we first showed them a short video demonstrating illustrated forms of learning, as shown in section 3.1 (we were not able to give an in-person demo due to the interview site constraints). Then we asked a series of questions about their view of the practicality, advantage, and challenge of employing Physica in physics education through open-ended questions. We also asked questions regarding the level of physics education they would consider Physica to be more effective for and the usefulness of each form of learning.

6.2.2 Results. Overall, all three physics educators were intrigued by our system to learn how physics concepts can be conveyed in a dynamic, tangible, and flexible manner. For example, E1 mentioned *"students can quickly change parameters and get a response"*. E2 expressed *"It would be useful for letting students get some hands-on interactive feel for some of the physics concepts"*, considering such tangible experience can be used as a supplement in addition to traditional physics instruction mode. One physics educator mentioned *"Physica can help students get a sense of how the phenomenon works by playing hands-on and trying different things"*, which implies that our event detector is practical.

E1 and E2 suggested including a graphing feature in the software as a future feature of Physica, commenting that it conveys changes in certain physics properties over time. This suggests that our system shall be improved to enable students to see how each physics parameter changes over time via plotted projection graphs as they interact. As for the actual utility of the system, E3 mentioned that *"It needs time for professors to start adopting the tool."* This highlights the need for simple instruction or introductory guides for the lecturers to learn and adapt Physica to their lectures easily. Lastly, E2 shared that *"the university-level physics lab's hands-on experiment usually is designed for students to confirm the theory/model they learned in the lecture,"* but he stated that *"Physica is offering a reverse effect that, through the tangible experience, it better lets them learn the model."*

Lastly, E2 mentioned an inherent problem existing across all kinds of physics simulation software/hardware, saying *"The weakness I see is it is not in the real system. Everything is based on simulation with this system. At some level, it is important for students to distinguish between real physics phenomena and simulation models. The point of doing physics experiments (especially in university-level*

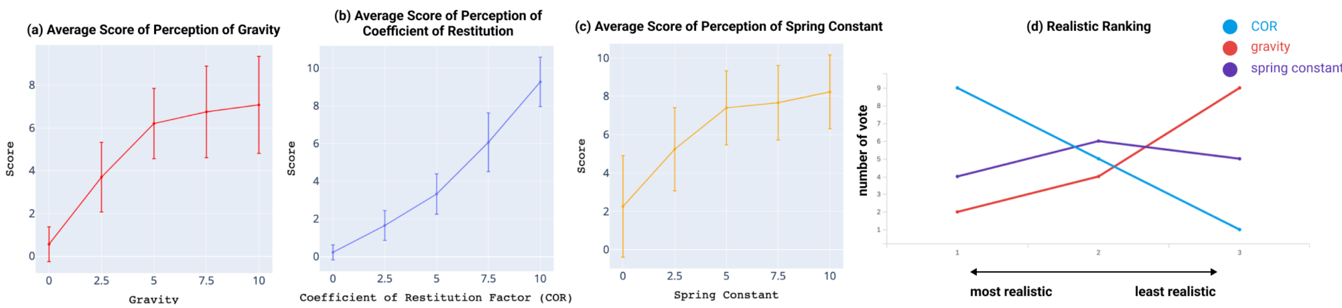


Figure 12: User Study Results - (a)-(c): Results of quantitative user study (vertical bars represent error bars) (d) Average score of "Which demo felt more realistic?" from the general audience participants.

Figure 13: Participants' various interactions with the Physica system: (a) a participant (P1) is using his finger to poke the toio robot connected by the virtual spring (b) a participant (P8) is *place & drop* an object on the toio mat (c) an example of a user (P5) grabbing an object connected by a virtual spring and feeling the haptic feedbacks (d) a user is *releasing* an object connected by a virtual spring.

experimental physics) is to evaluate the model, not the reversed process.". This feedback helped us understand the specific aspect of the learning experience we can contribute to both students and teachers. Instead of viewing Physica as a substitute for traditional physics lab, E2 mentioned that *"it is very useful in evaluating the math relationship behind the equations"*.

Forms of Learning and Suitable Level of Education: Figure 14 shows the result for the scenarios of Forms of Learning rating, and the *remote lecture* scenario (Figure 4 d) gained the highest score. E3 mentioned this scenario is useful because *"the students can see and play in their home. Makes it more like a classroom setting"*. We assume that due to the past few years of experience in remote teaching during the pandemic, this scenario attracted the lectures the most. As for the suitable level of education rated by the experts, the high school received the highest score of 8, followed by middle school = 7.3, university = 6, primary school = 5.7, and kindergarten = 5.3. This indicates the experts' intuition for Physica to be most effective for the physics education level that is not too abstract and complex as the university level, but a level that benefits from conveying variable properties to be explored and represented.

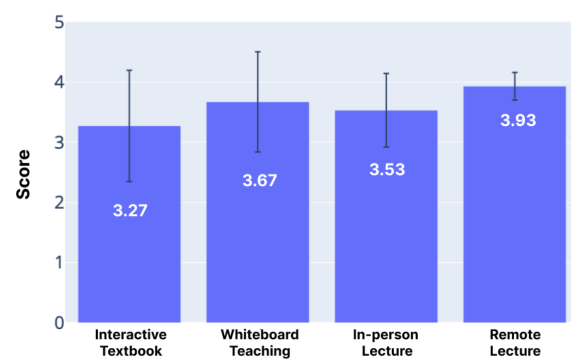


Figure 14: Average score of Forms of Learning scenarios from physics educators.

7 LIMITATIONS AND FUTURE WORK

7.1 Control Systems and Simulation Improvements

In this paper, we have demonstrated Physica to support a variety of interaction techniques (e.g. slide, hit, etc) by leveraging the event detector. A future direction could incorporate more types of interaction such as *rotating* to allow for richer input. Another direction is incorporating a sandbox-like customizable approach and features so that users can have more freedom to explore physics concepts. A potential future can be a sketching function similiar to what is proposed in Sketched Reality [43]. For example, a user can use one toio robot as pen to paint the gravitational field, mechanical gears, or solid surface. By utilizing the sketching function, users can create a unique physical system setup that cannot be replicated in the real world. Additionally, the sketching functionality can encourage children to collaboratively learn physics concepts as multiple children can cooperatively draw on the same mat. Lastly, another future research direction is to enable physics educators to quickly design their own physics simulation to suit their needs.

7.2 Additional Simulations

As for simulation, other types of simulation properties could be explored and demonstrated beyond what we supported in our paper (Figure 3), including simulation in linkage mechanisms, wire mesh, or fluid. Furthermore, in the computer graphics field, as a large number of simulated particles can achieve high-resolution material models [76], we could achieve making such massive particle simulation tangible by increasing the number of robots in Physica, allowing for simulating fluids, winds, or sand. We could employ open source control architecture [6] to control more than 200 toio robots simultaneously. In such a space, we expect it could advance physics concepts to be explored and learned by students.

7.3 Hardware Improvements

While the toio hardware is constrained with its non-holonomic drive with two-wheeled implementations, future implementation could incorporate an omnidirectional drive [85] that allows the robots to freely move in any direction on the mat to best represent digital simulations. Another hardware improvement we can make is to increase the maximum motor velocity. As the current toio robot can have a maximum linear speed of 35cm/s [8], if the speed of the virtual particle is greater than this speed limit, toio can no longer catch up to the digital simulation. Future robots with much higher maximum speed, thus, will make the simulation more accurate and minimize the gap between the virtual simulated objects and the robots. On the other hand, unlike pixels on screens, any physical robots have the inherent limitation of maximum speed. Hence, handling and representing physics simulation that is beyond the limit of the hardware is one of the UI and interaction design challenges, especially in the educational contexts.

7.4 Extended User Studies and Long-Term Deployment

In the future, we should evaluate how Physica can help students learn physics. Such a direction will possibly include long-term studies in an actual learning setting and evaluate how exploratory behavior is associated with neurophysiological learning output by conducting empirical experiments [12, 23, 44]. Specifically, how the simulation represented with Physica could appropriately relate to real-world physics models for students to understand physics concepts effectively shall be investigated. We could also observe how students can better collaborate to learn physics through Physica through such a study, as that is one of the core advantages of TUIs. Such a study shall also investigate the suitable range of ages or levels of physics education to better engage children or students in the learning topics, to further confirm the interview results from our study. Additionally, future work should explore how teachers would employ Physica in real-classroom environments to teach custom topics in physics or even other subjects.

The insights gained from such an experiment will further help us understand the usability and effect of Physica. It should also inform us of the design and engineering improvements towards bringing the Physica system into practice. As toios are already employed and deployed for STEM education applications (e.g. programming, music making, and robotics) in Japan [84], integrating Physica-like advanced interactive applications would expand its potential in education and can be easily deployed into their product.

8 CONCLUSION

In this study, we proposed Physica, a system that enables users to feel, experience, and touch digital physics simulations via tabletop A-TUIs. We implemented a proof-of-concept system that supports a variety of interaction techniques (e.g. grab, pick-up, throw, hit). Based on our physics simulation algorithm, event detector, and customizable GUI, we demonstrated various applications such as tangible gaming and physics concept demonstrations. Our system was evaluated through a perception study to see how users interact with Physica and how they interpret the physics parameters. We also interviewed professional physics educators to get their perspectives on Physica and investigate whether this could be potentially used in future physics classrooms/learning. For future work, we also discussed how our approach could be further improved toward the goal to facilitate the learning experience for physics education.

ACKNOWLEDGMENTS

We thank Prof. Marshini Chetty and Prof. Diana Franklin for their advice during our user study planning. We also thank Anup Sathya, David Wu, and Ramarko Bhattacharya for their proofreading and feedback on the paper.

REFERENCES

[1] 2004. CircuitTUI. https://tangible.media.mit.edu/project/circuittui/
[2] 2011. Khan Academy Physics Simulation. https://www.khanacademy.org/computing/ap-computer-science-principles/x2d2f703b37b450a3:simulations/x2d2f703b37b450a3:exploring-simulations/a/physics-simulations
[3] 2018. Ophysics. https://ophysics.com/
[4] 2020. National Science and Technology Medals Foundation. https://lab.nationalmedals.org/gravity.php
[5] 2020. Sony Interactive Entertainment. toio^{TM} Mat for Developers. https://toio.io/news/2020/20200423-1.html
[6] 2020. Zorozoro System. https://github.com/Whatever-Inc/zorozoro-experiments
[7] 2022. Angry Birds. https://www.angrybirds.com/
[8] 2022. toio mobility performance by toio official documentation. https://toio.github.io/toio-spec/en/docs/hardware_other
[9] Sue Allen. 2004. Designs for learning: Studying science museum exhibits that do more than entertain. *Science education* 88, S1 (2004), S17–S33.
[10] Firas Almasri. 2022. Simulations to teach science subjects: Connections among students' engagement, self-confidence, satisfaction, and learning styles. *Education and Information Technologies* (2022), 1–21.
[11] Alexis Andre. 2021. toio rust server. https://github.com/MacTuitui/toio-osc/
[12] Jaclyn Barnes, S Maryam FakhrHosseini, Eric Vasey, Chung Hyuk Park, and Myounghoon Jeon. 2020. Child-robot theater: Engaging elementary students in informal STEAM education using robots. *IEEE Pervasive Computing* 19, 1 (2020), 22–31.
[13] Nathan Bell, Yizhou Yu, and Peter J Mucha. 2005. Particle-based simulation of granular materials. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 77–86.
[14] Jan Bender, Kenny Erleben, and Jeff Trinkle. 2014. Interactive simulation of rigid body dynamics in computer graphics. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 246–270.
[15] Jeremiah U Brackbill, Douglas B Kothe, and Charles Zemach. 1992. A continuum method for modeling surface tension. *Journal of computational physics* 100, 2 (1992), 335–354.
[16] R. Bridson, S. Marino, and R. Fedkiw. 2005. Simulation of Clothing with Folds and Wrinkles. In *ACM SIGGRAPH 2005 Courses* (Los Angeles, California) *(SIGGRAPH '05)*. Association for Computing Machinery, New York, NY, USA, 3–es. https://doi.org/10.1145/1198555.1198573

[17] Ashlie Brown and Hayes Raffle. 2009. Opportunities for Actuated Tangible Interfaces to Improve Protein Study. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI EA '09)*. Association for Computing Machinery, New York, NY, USA, 2417–2426. https://doi.org/10.1145/1520340.1520344

[18] JH Coggon. 1971. Electromagnetic and electrical modeling by the finite element method. *Geophysics* 36, 1 (1971), 132–155.

[19] Gabor Czilli and Ulrich Rüde. 2010. Particle-based simulation of deformable bodies. (2010).

[20] dennisppaul. 2020. Teilchen: A simple physics library based on particles, forces, constraints and behaviors. https://github.com/dennisppaul/teilchen

[21] Mathieu Desbrun and Marie-Paule Cani. 1999. *Space-time adaptive simulation of highly deformable substances*. Ph. D. Dissertation. INRIA.

[22] Mathieu Desbrun and Marie-Paule Gascuel. 1996. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Computer Animation and Simulation'96*. Springer, 61–76.

[23] Son Do-Lenh, Patrick Jermann, Sébastien Cuendet, Guillaume Zufferey, and Pierre Dillenbourg. 2010. Task performance vs. learning outcomes: a study of a tangible user interface in the classroom. In *Sustaining TEL: From Innovation to Learning and Practice: 5th European Conference on Technology Enhanced Learning, EC-TEL 2010, Barcelona, Spain, September 28-October 1, 2010. Proceedings 5*. Springer, 78–92.

[24] Pierre Dragicevic and Yvonne Jansen. 2012. List of Physical Visualizations. www.dataphys.org/list. Last accessed 2022-07-28.

[25] Douglas Enright, Stephen Marschner, and Ronald Fedkiw. 2002. Animation and rendering of complex water surfaces. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 736–744.

[26] George W. Fitzmaurice, Hiroshi Ishii, and William A. S. Buxton. 1995. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '95)*. ACM Press/Addison-Wesley Publishing Co., USA, 442–449. https://doi.org/10.1145/223904.223964

[27] Morten Fjeld, Jonas Fredriksson, Martin Ejdestig, Florin Duca, Kristina Bötschi, Benedikt Voegtli, and Patrick Juchli. 2007. Tangible user interface for chemistry education: comparative evaluation and re-design. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 805–808.

[28] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: dynamic physical affordances and constraints through shape and object actuation. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, St. Andrews Scotland, United Kingdom, 417–426. https://doi.org/10.1145/2501988.2502032

[29] Nick Foster and Ronald Fedkiw. 2001. Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 23–30.

[30] Phil Frei, Victor Su, Bakhtiar Mikhak, and Hiroshi Ishii. 2000. Curlybot: designing a new class of computational toys. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 129–136.

[31] David Furió, Stéphanie Fleck, Bruno Bousquet, Jean-Paul Guillet, Lionel Canioni, and Martin Hachet. 2017. Hobit: Hybrid optical bench for innovative teaching. In *Proceedings of the 2017 chi conference on human factors in computing systems*. 949–959.

[32] Sarah F. F. Gibson and Brian Mirtich. 1997. *A Survey of Deformable Modeling in Computer Graphics*. Technical Report. Mitsubishi Electric Research Laboratories.

[33] Pauline Gourlet, Mathieu Le Goc, and Sean Follmer. 2017. Revisiting Turtles and Termites: an Open-ended Interactive Physical Game with Multiple Robots. https://doi.org/10.1145/3078072.3091979

[34] Cyril W Hirt and Billy D Nichols. 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of computational physics* 39, 1 (1981), 201–225.

[35] M. Hohenwarter, M. Borcherds, G. Ancsin, B. Bencze, M. Blossier, J. Éliás, K. Frank, L. Gál, A. Hofstätter, F. Jordan, Z. Konečný, Z. Kovács, E. Lettner, S. Lizelfelner, B. Parisse, C. Solyom-Gecse, C. Stadlbauer, and M. Tomaschko. 2018. *GeoGebra 5.0.507.0*. http://www.geogebra.org.

[36] Loulin Huang. 2009. Control approach for tracking a moving target by a wheeled mobile robot with limited velocities. *IET control theory & applications* 3, 12 (2009), 1565–1577.

[37] Loulin Huang and Liqiong Tang. 2008. Dynamic target tracking control for a wheeled mobile robots constrained by limited inputs. *IFAC Proceedings Volumes* 41, 2 (2008), 3087–3091.

[38] Brbel Inhelder and Jean Piaget. 2013. *The early growth of logic in the child: Classification and seriation*. Routledge.

[39] Hiroshi Ishii and Brygg Ullmer. 1997. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '97)*. Association for Computing Machinery, New York, NY, USA, 234–241. https://doi.org/10.1145/258549.258715

[40] David Jacqmin. 1999. Calculation of two-phase Navier–Stokes flows using phase-field modeling. *Journal of computational physics* 155, 1 (1999), 96–127.

[41] Wafa Johal, Yu Peng, and Haipeng Mi. 2020. Swarm Robots in Education: A Review of Challenges and Opportunities. In *Proceedings of the 8th International Conference on Human-Agent Interaction*. 272–274.

[42] Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. 2007. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*. 139–146.

[43] Hiroki Kaimoto, Kyzyl Monteiro, Mehrad Faridan, Jiatong Li, Samin Farajian, Yasuaki Kakehi, Ken Nakagaki, and Ryo Suzuki. 2022. Sketched Reality: Sketching Bi-Directional Interactions Between Virtual and Physical Worlds with AR and Actuated Tangible UI. *arXiv preprint arXiv:2208.06341* (2022).

[44] Raphael Kaplan and Karl J Friston. 2018. Planning and navigation as active inference. *Biological cybernetics* 112, 4 (2018), 323–343.

[45] Lawrence Kim, Daniel Drew, Veronika Domova, and Sean Follmer. 2020. User-defined Swarm Robot Control. 1–13. https://doi.org/10.1145/3313831.3376814

[46] Lawrence H. Kim and Sean Follmer. 2017. UbiSwarm: Ubiquitous Robotic Interfaces and Investigation of Abstract Motion as a Display. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (Sept. 2017), 1–20. https://doi.org/10.1145/3130931

[47] David Kirsh. 2010. Thinking with external representations. *AI & society* 25, 4 (2010), 441–454.

[48] Lionel Lapierre, Rene Zapata, and Pascal Lepinay. 2007. Combined path-following and obstacle avoidance control of a wheeled robot. *The International Journal of Robotics Research* 26, 4 (2007), 361–375.

[49] Mathieu Le Goc, Lawrence H. Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, and Sean Follmer. 2016. Zooids: Building Blocks for Swarm User Interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, Tokyo Japan, 97–109. https://doi.org/10.1145/2984511.2984547

[50] David Ledo, Miguel A. Nacenta, Nicolai Marquardt, Sebastian Boring, and Saul Greenberg. 2012. The HapticTouch toolkit: enabling exploration of haptic interactions. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*. ACM, Kingston Ontario Canada, 115–122. https://doi.org/10.1145/2148131.2148157

[51] Daniel Leithinger, Dávid Lakatos, Anthony DeVincenzi, and Matthew Blackshaw. 2011. Recompose-Direct and gestural interaction with an actuated surface. (2011).

[52] Daniel Leithinger, David Lakatos, Anthony DeVincenzi, Matthew Blackshaw, and Hiroshi Ishii. 2011. Direct and gestural interaction with relief: a 2.5 D shape display. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 541–548.

[53] Jiannan Li, Maurício Sousa, Chu Li, Jessie Liu, Yan Chen, Ravin Balakrishnan, and Tovi Grossman. 2022. ASTEROIDS: Exploring Swarms of Mini-Telepresence Robots for Physical Skill Demonstration. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–14.

[54] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B Tenenbaum, and Antonio Torralba. 2018. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. *arXiv preprint arXiv:1810.01566* (2018).

[55] John Locke. 2010. An essay concerning human understanding book II: Ideas. *Early Modern Texts* (2010).

[56] Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. 2014. Unified particle physics for real-time applications. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–12.

[57] Joseph Malloch and Marcelo M Wanderley. 2007. The T-Stick: From musical interface to musical instrument. In *Proceedings of the 7th international conference on New interfaces for musical expression*. 66–70.

[58] Melisa Orta Martinez, Cara M Nunez, Ting Liao, Tania K Morimoto, and Allison M Okamura. 2019. Evolution and analysis of hapkit: An open-source haptic device for educational applications. *IEEE transactions on haptics* 13, 2 (2019), 354–367.

[59] P Martínez-Jiménez, E Casado, JM Martínez-Jiménez, M Cuevas-Rubiño, D González-Caballero, F Zafra-López, and Denis Donnelly. 1997. Interactive physics simulations appeal to first-year students. *Computers in Physics* 11, 1 (1997), 31–35.

[60] Matthias Muller, Matthias Teschner, and Markus Gross. 2004. Physically-based simulation of objects represented by surface meshes. In *Proceedings Computer Graphics International, 2004*. IEEE, 26–33.

[61] Lakshman S Myneni, N Hari Narayanan, Sanjay Rebello, Amy Rouinfar, and Sadhana Pumtambekar. 2013. An interactive and intelligent learning system for physics education. *IEEE Transactions on learning technologies* 6, 3 (2013), 228–239.

[62] Ken Nakagaki, Daniel Fitzgerald, Zhiyao Ma, Luke Vink, Daniel Levine, and Hiroshi Ishii. 2019. inFORCE: Bi-directionalForce'Shape Display for Haptic Interaction. In *Proceedings of the thirteenth international conference on tangible, embedded, and embodied interaction*. 615–623.

[63] Ken Nakagaki, Joanne Leong, Jordan L. Tappa, João Wilbert, and Hiroshi Ishii. 2020. HERMITS: Dynamically Reconfiguring the Interactivity of Self-propelled TUIs with Mechanical Shell Add-ons. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual Event USA, 882–896. https://doi.org/10.1145/3379337.3415831

[64] Ken Nakagaki, Jordan L Tappa, Yi Zheng, Jack Forman, Joanne Leong, Sven Koenig, and Hiroshi Ishii. 2022. (Dis) Appearables: A Concept and Method for Actuated Tangible UIs to Appear and Disappear based on Stages. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–13.

[65] Ken Nakagaki, Luke Vink, Jared Counts, Daniel Windham, Daniel Leithinger, Sean Follmer, and Hiroshi Ishii. 2016. Materiable: Rendering Dynamic Material Properties in Response to Direct Physical Touch with Shape Changing Interfaces. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, 2764–2772. https://doi.org/10.1145/2858036.2858104

[66] Isabel Neto, Wafa Johal, Marta Couto, Hugo Nicolau, Ana Paiva, and Arzu Guneysu. 2020. Using tabletop robots to promote inclusive classroom experiences. In *Proceedings of the interaction design and children conference*. 281–292.

[67] Frank Oppenheimer. 1972. The Exploratorium: A playful museum combines perception and art in science education. *American Journal of Physics* 40, 7 (1972), 978–984.

[68] Giuseppe Oriolo, Alessandro De Luca, and Marilena Vendittelli. 2002. WMR control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Transactions on control systems technology* 10, 6 (2002), 835–852.

[69] Ayberk Özgür, Séverin Lemaignan, Wafa Johal, Maria Beltran, Manon Briod, Léa Pereyre, Francesco Mondada, and Pierre Dillenbourg. 2017. Cellulo: Versatile handheld robots for education. In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*. IEEE, 119–127.

[70] Gian Pangaro, Dan Maynes-Aminzade, and Hiroshi Ishii. [n. d.]. The Actuated Workbench: Computer-Controlled Actuation in Tabletop Tangible Interfaces. 4, 2 ([n. d.]), 10.

[71] Seymour Papert and Idit Harel. 1991. Situating constructionism. *Constructionism* 36, 2 (1991), 1–11.

[72] Seymour A Papert. 2020. *Mindstorms: Children, computers, and powerful ideas*. Basic books.

[73] Michael Park, Sachin Chitta, Alex Teichman, and Mark Yim. 2008. Automatic configuration recognition methods in modular robots. *The International Journal of Robotics Research* 27, 3-4 (2008), 403–421.

[74] James Patten and Hiroshi Ishii. 2007. Mechanical constraints as computational constraints in tabletop tangible interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, 809–818. https://doi.org/10.1145/1240624.1240746

[75] Sonja Petrovska, Despina Sivevska, and Oliver Cackov. 2013. Role of the game in the development of Preschool Child. https://www.sciencedirect.com/science/article/pii/S1877042813029017

[76] Simon Premžoe, Tolga Tasdizen, James Bigler, Aaron Lefohn, and Ross T Whitaker. 2003. Particle-based simulation of fluids. In *Computer Graphics Forum*, Vol. 22. Wiley Online Library, 401–410.

[77] Iulian Radu and Bertrand Schneider. 2019. What can we learn from augmented reality (AR)? Benefits and drawbacks of AR for inquiry-based learning of physics. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–12.

[78] Hayes Solos Raffle, Amanda J Parkes, and Hiroshi Ishii. 2004. Topobo: a constructive assembly system with kinetic memory. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 647–654.

[79] Shwetha Rajaram and Michael Nebeling. 2022. Paper Trail: An Immersive Authoring System for Augmented Reality Instructional Experiences. In *CHI Conference on Human Factors in Computing Systems*. 1–16.

[80] John W. Romanishin, Kyle Gilpin, and Daniela Rus. 2013. M-blocks: Momentum-driven, magnetic modular robots. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Tokyo, 4288–4295. https://doi.org/10.1109/IROS.2013.6696971

[81] Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. 2014. Programmable self-assembly in a thousand-robot swarm. *Science* 345, 6198 (Aug. 2014), 795–799. https://doi.org/10.1126/science.1254295

[82] Kimiko Ryokai, Stefan Marti, and Hiroshi Ishii. 2004. I/O brush: drawing with everyday objects as ink. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 303–310.

[83] Nathan E Sanders, Chris Faesi, and Alyssa A Goodman. 2014. A new approach to developing interactive software modules through graduate education. *Journal of Science Education and Technology* 23, 3 (2014), 431–440.

[84] Sony. [n. d.]. Toy platform toio™ / Stories / Sony Design / Sony. https://www.sony.com/en/SonyInfo/design/stories/toio/

[85] James Patten Studio. [n. d.]. *Thumbles - Robotic tabletop user interface platform*. https://www.youtube.com/watch?v=SsD3uTUtPks - accessed July-10-2020.

[86] Ryo Suzuki, Hooman Hedayati, Clement Zheng, James L. Bohn, Daniel Szafir, Ellen Yi-Luen Do, Mark D. Gross, and Daniel Leithinger. 2020. RoomShift: Room-scale Dynamic Haptics for VR with Furniture-moving Swarm Robots. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM. https://doi.org/10.1145/3313831.3376523

[87] Ryo Suzuki, Adnan Karim, Tian Xia, Hooman Hedayati, and Nicolai Marquardt. 2022. Augmented reality and robotics: A survey and taxonomy for ar-enhanced human-robot interaction and robotic interfaces. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–33.

[88] Ryo Suzuki, Rubaiat Habib Kazi, Li-Yi Wei, Stephen DiVerdi, Wilmot Li, and Daniel Leithinger. 2020. Realitysketch: Embedding responsive graphics and visualizations in AR through dynamic sketching. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 166–181.

[89] Ryo Suzuki, Eyal Ofek, Mike Sinclair, Daniel Leithinger, and Mar Gonzalez-Franco. 2021. HapticBots: Distributed Encountered-type Haptics for VR with Multiple Shape-changing Mobile Robots. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual Event USA, 1269–1281. https://doi.org/10.1145/3472749.3474821

[90] Ryo Suzuki, Clement Zheng, Yasuaki Kakehi, Tom Yeh, Ellen Yi-Luen Do, Mark D. Gross, and Daniel Leithinger. 2019. ShapeBots: Shape-changing Swarm Robots. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM. https://doi.org/10.1145/3332165.3347911

[91] Lucia Terrenghi, Matthias Kranz, Paul Holleis, and Albrecht Schmidt. 2006. A cube to learn: a tangible user interface for the design of a learning appliance. *Personal and Ubiquitous Computing* 10, 2 (2006), 153–158.

[92] Ana M Villanueva, Ziyi Liu, Zhengzhe Zhu, Xin Du, Joey Huang, Kylie A Peppler, and Karthik Ramani. 2021. Robotar: An augmented reality compatible teleconsulting robotics toolkit for augmented makerspace experiences. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.

[93] Florian Vogt. 2009. *Towards an Interactive Framework for Upper Airway Modeling*. Ph. D. Dissertation. PhD thesis, University of British Columbia.

[94] Emanuel Vonach, Georg Gerstweiler, and Hannes Kaufmann. 2014. Acto: A modular actuated tangible user interface object. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*. 259–268.

[95] Emanuel Vonach, Christoph Schindler, and Hannes Kaufmann. 2021. StARboard & TrACTOr: Actuated Tangibles in an Educational TAR Application. *Multimodal Technologies and Interaction* 5, 2 (2021), 6.

[96] Lev Semenovich Vygotsky and Michael Cole. 1978. *Mind in society: Development of higher psychological processes*. Harvard university press.

[97] Tom Walsh. 2017. Creating interactive physics simulations using the power of GeoGebra. *The physics teacher* 55, 5 (2017), 316–317.

[98] Carl E Wieman, Wendy K Adams, and Katherine K Perkins. 2008. PhET: Simulations that enhance learning. *Science* 322, 5902 (2008), 682–683.

[99] Yiwei Zhao, Lawrence H Kim, Ye Wang, Mathieu Le Goc, and Sean Follmer. 2017. Robotic assembly of haptic proxy objects for tangible interaction and virtual reality. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. 82–91.

[100] Fabian "fabiantheblind" Morón Zirfas. 2016. Processing and keystone - doing projection mapping. https://fh-potsdam.github.io/doing-projection-mapping/processing-keystone/

A APPENDIX: DETAIL TECHNICAL IMPLEMENTATION

This appendix describes the implementation details of the target speed control and touch detector to support the information in the body of the paper.

A.1 Target Speed Control Implementation

To control robots with regard to target velocity and position, we first get both the target simulated object's speed and the actual toio robot's speed. We then add a correction factor (Algorithm 1 - line 10) on top of the motor speed from the previous frame to decide the new motor speed. Our speed control function, *aimCubePosVel*, works in the following way:

Algorithm 1: Target Speed Control

input : target x and y positions (tx, tx), x and y components of the target velocity (vx, vy)
Result: the robot will follow a dynamic target

1 $diffAngle \leftarrow$ $angle\ between\ toio's\ orientation\ and\ target's\ velocity\ vector$
2 **if** *the direction of the target velocity is approximately perpendicular to the direction of the wheel* **then**
3   rotate the toio to minimize $diffAngle$
4 **else**
5   **if** *target is in front of the robot* **then**
6   go forward
7   **else**
8   face back
9   **end**
10     $correction\_factor \leftarrow \frac{\sqrt{vx^2+vy^2}}{2}$
11     $left\_wheel\_speed \leftarrow previous\_left\_wheel\_speed + correction\_factor$
12     $right\_wheel\_speed \leftarrow previous\_right\_wheel\_speed + correction\_factor$
13 **end**

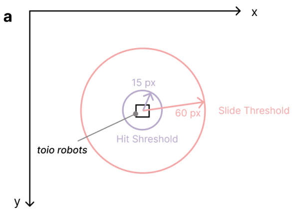
Since toio does not have omnidirectional wheels, a self-correction step (Algorithm 1 - line 2), is required in which toio will rotate toward the velocity direction to avoid first before steering towards the target. After the aforementioned algorithm, the toios can adapt to follow a dynamic target, ensuring the quality of our simulation. This algorithm also supports following curved paths (e.g. for following gravitational projectile motion as in Figure 3 (iii)) by controlling two wheels in a different ratio, based on target velocity, which also includes vector elements.

A.2 Event Detector Classification Implementation

For the event detector's classification method, below describes how each interaction/event was classified based on the general detection flow (described in Section) of *inputs*, *threshold*, and *feedback*.

<u>Pick-Up & Place:</u> Once the object is placed, toio's coordinate point is detected thanks to the toio's localization capability so that the simulation can be interpreted as an event if the object was placed or picked up.

a



b

	Position (unit: pixel)	Velocity (unit: pixel/second)
Pick & Place		
Grab & Release	✓	
Slide & Release	✓	✓
Hit	✓	✓
Collision and Hit with Objects	✓	✓

Figure 15: Illustrated description of event detector algorithm. (Left) Thresholds used in *slide* and *hit* (Right) Input variables used in Event Detector for each classifying interaction.

<u>Grab & Release:</u> After the user releases the pendulum object, the object will follow the bob based on the target speed control function we discussed in Section 4.2.2. Before such release, the algorithm will calculate the distance between the actual location of the robot and the virtual object. If the distance is larger than 40 pixels, the algorithm would detect the user input while grabbing the bob and assign the virtual bob to the object so that the amplitude is reset. For this interaction, the input is the position of the toio robot, the threshold is 40 pixels from the simulated object position, and the feedback, to be reflected on the simulation, is the updated position of the toio robot (Figure 15 right).

<u>Slide & Release:</u> To predict the *slide* behavior, the algorithm calculates the distance between the original position of the toio (i.e. the position where the user starts the sliding behavior) and the current position of the toio. When the distance passes a certain threshold, the system classifies it as a *slide*. Since users need to accelerate the object before releasing it, the distance the toio is scanned during the acceleration phase is longer than a short flick. Therefore, we set the threshold to 60 pixels (Figure 15 left). When the scanned distance exceeds 60 pixels, teilchen will launch a virtual particle based on the release velocity right at the 60 pixels threshold and our program can ask toio to follow the virtual particle, creating a trajectory on the mat. For this interaction, the input is the position and the velocity of the toio robot, the threshold is 60 pixels, and the output is the velocity of the toio robot (Figure 15 right).

<u>Hit:</u> Similar to the *slide* event, we also used the threshold method to implement the *hit* interaction. Compared to *slide*, *hit* can be defined as the toio suddenly receiving a sharp increase in velocity. For this interaction, the input is the position and the velocity of the toio robot, the positional threshold is 15 pixels as the hit has to be detected as a short burst, and the feedback is the velocity and position of the toio robot (Figure 15 right).

<u>Collision with static objects:</u> Here, we use a combination of position and velocity to classify when the toio collides with an obstacle. A collision can be defined as the distance between the toio's position and the virtual particle passing a certain threshold and the magnitude of the toio's speed vector equalling zero. As for the implementation of the distance threshold, we chose 20 pixels after iterations, and the magnitude of the velocity vector can be calculated based on the Euclidean norm. After the event detection detects a collision, it will output the reversed y-axis velocity component to create the bouncing effect.