

Implementation of a Push Button -Based Tangible User Interface for Virtual Object Control in Unity

**Rizki Dwi Irianti ^{1*}, Ashafidz Fauzan Dianta ¹, Zakha Maisat Eka Darmawan ¹,
Firnanda Pristiana Nurmaida ¹, Armany Rizqullah Saputra ¹**

Politeknik Elektronika Negeri Surabaya

Article Info

Article history:

Received 7 January 2026

Revised 10 January 2026

Accepted 13 January 2026

Keywords:

Tangible User Interface, Push Button, Unity, Physical-Digital Interaction, Interactive Multimedia

ABSTRACT

This study discusses the implementation of a push-button based Tangible User Interface (TUI) as a physical interaction medium to control virtual objects in the Unity environment. The system is designed by integrating physical input devices in the form of push buttons connected to a microcontroller with a Unity application via serial communication. This study uses a system design and implementation method that includes the stages of requirements analysis, system architecture design, hardware and software implementation, and functional testing. User input is processed by the system to produce a visual response in the form of movement of virtual objects to the left or right with a toggle mechanism. The test results show that the system is able to respond to physical input in real-time with stable and consistent object movement according to user commands. This TUI implementation provides a more concrete interaction experience compared to conventional mouse-or keyboard-based interfaces, and has the potential to be developed as a basis for interactive media, control simulations, and multimedia applications based on physical-digital interaction.

This is an open access article under the CC BY-SA license.



Corresponding Author:

Rizki Dwi Irianti | Politeknik Elektronika Negeri Surabaya

Email: irianti@pens.ac.id

1. Introduction

The development of interactive multimedia technology has encouraged the emergence of new forms of interaction between humans and digital systems that no longer rely on conventional devices such as mice and keyboards. One emerging approach is the Tangible User Interface (TUI), an interface that allows users to interact with digital systems through physical objects. This approach provides a more concrete, intuitive, and aligned interaction experience with the user's physical activities, so it is widely applied to interactive systems, learning media, simulations, and digital installations (Preece, Rogers, and Sharp (2015)).

In the context of interactive multimedia, the integration of physical input devices and digital visual environments is crucial for building real-time interaction feedback. Several studies

have shown that the use of TUIs can increase user engagement and strengthen the causal relationship between physical actions and digital responses. A recent study reported that physical interfaces combined with digital visualizations can create a more effective embodied interaction experience than graphical interfaces alone (Uhrík, Koniar, and Novák, 2022).

Unity, as a game engine, provides an interactive visual environment that supports the integration of external devices through serial communication. The combination of Unity with microcontroller devices enables the development of responsive and flexible physical-digital interaction systems. Previous research has shown that integrating TUIs with digital systems can enhance user engagement and experience, particularly in interactive game- and simulation-based applications (Salazar-Cardona, Muñoz-Sáenz, and Roldán-Gómez, 2023). Furthermore, Unity is also widely used as a prototyping tool for interactive systems due to its support for real-time processing and dynamic visualization (Unity Technologies, 2023).

However, most TUI implementations still focus on the use of complex sensors or specialized devices, while studies on the application of simple physical inputs such as push buttons in Unity-based TUI systems are relatively limited. Previous research generally emphasizes technical implementation aspects without systematically and structuredly describing interaction mechanisms (Antle and Wise, 2019). This suggests an opportunity to examine the design and implementation of simple TUI systems that still provide real-time interactive responses and are easy to implement.

Based on this background, the problem in this research is formulated as follows: how to design and implement a push button-based Tangible User Interface system integrated with Unity to control virtual objects in real-time. This research aims to design and implement a physical-digital interaction system using push buttons as input and evaluate the system's response through functional testing of virtual object movements. The results of this research are expected to contribute to the development of a simple TUI-based interactive multimedia system and become the basis for the development of more complex interactive applications in the future.

2. Research Methods

This research uses a system design and implementation method aimed at designing and realizing a push-button-based Tangible User Interface (TUI) system integrated with Unity. This method includes the stages of requirements analysis, system design, hardware and software implementation, and functional testing of the system.

Research Data

The research data used consists of system response data to user physical input, specifically changes in direction and motion status of virtual objects in the Unity environment. The data was obtained through direct observation of the system's behavior when receiving input from push buttons.

Research Tools and Materials

The tools and materials used include an Arduino microcontroller, two push buttons as physical input devices, a breadboard and connecting cables, as well as the Arduino IDE and Unity software as an interactive application development environment.

System Approach and Design

This research approach is carried out through system engineering by integrating physical input devices and digital visual systems to build a Tangible User Interface (TUI) interaction mechanism. The system is designed as a unit that connects the user's physical actions with direct visual responses, thus enabling responsive and real-time physical-digital interactions. The system architecture consists of push buttons as input devices, Arduino as input signal processors, serial communication as data transmission media, and Unity as a virtual object visualization environment.

The system's conceptual workflow is shown in Figure 1, which illustrates the stages of the interaction process from input to output. The process begins with the Arduino reading the push button status as a low or high logic state. Each change in the button's status is interpreted by the Arduino as a numeric signal that represents the virtual object's movement direction command. In this system, signal 1 is used to represent movement to the left, while signal 2 represents movement to the right.

The resulting numeric signal is then sent to Unity via serial communication for further processing. Unity acts as a data receiver and virtual object controller, where each received signal triggers a change in the object's movement conditions in the visual environment. The interaction mechanism is designed using the toggle principle, where a single button press activates the movement of the virtual object in a specific direction, while a subsequent button press stops the movement and maintains the object's last position on the screen.

This interaction mechanism design allows the system to work simply yet effectively, by providing clear visual feedback to every physical action of the user. Thus, the cause-and-effect relationship between user actions and system responses can be directly observed, which is a key characteristic of implementing a Tangible User Interface in an interactive multimedia system.

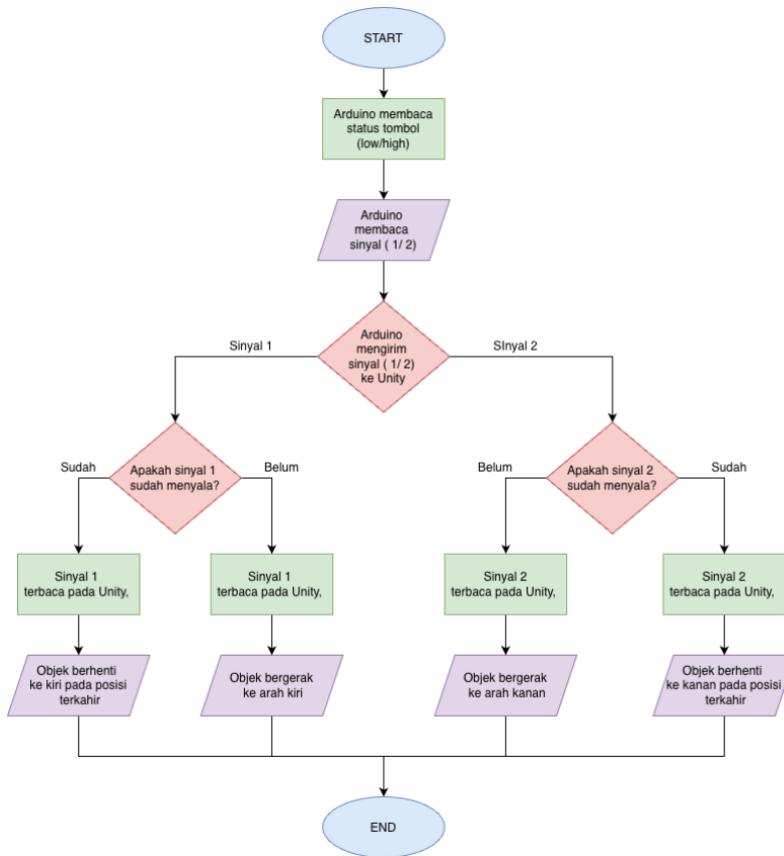


Figure 1. Flowchart of the Tangible User Interface system interaction mechanism

Hardware Implementation

The hardware implementation of the Tangible User Interface (TUI) system is shown in Figure 2. The system consists of an Arduino microcontroller that functions as an input processing center, two push buttons as a physical user interface, and a breadboard and jumper cables as a medium for assembling the circuit. The push buttons are connected to Arduino digital pins and configured using INPUT_PULLUP mode, so that the logic state of the button will be low when pressed and high when released. This configuration allows stable input readings without the need for additional external resistor components.

The Arduino is connected to a computer via a USB cable, which serves as both a power source and a serial communication medium between the hardware and the Unity application. Through this serial communication, the pushbutton signal is sent as numeric data to the Unity system for further processing. The hardware circuit is designed to be simple and modular for ease of implementation and allows for further system development by adding additional input devices if needed.

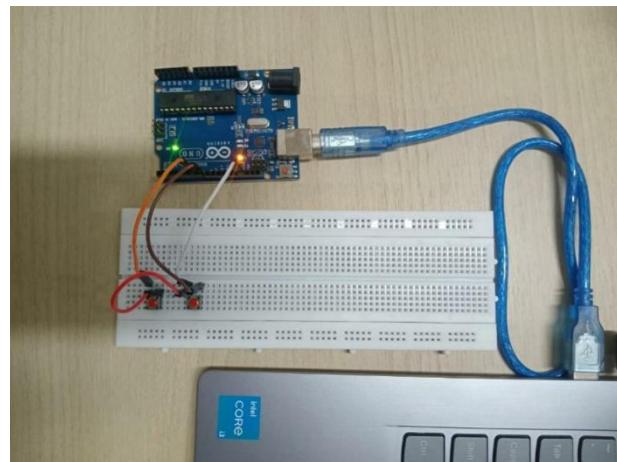


Figure 2. Tangible User Interface Hardware Circuit Based on Push Buttons

Software Implementation

On the software side, Unity is used as a game engine to build interactive visual environments and manage the system's response to physical user input. Unity was chosen for its capabilities in handling three-dimensional visualization, real-time data processing, and ease of integration with external devices via serial communication. The Unity environment acts as a data receiving system from the Arduino and controls the behavior of virtual objects based on the received signals.

The virtual object used in this study is a cube placed in a three-dimensional scene in Unity. The object is configured with a mesh renderer component for visualization, a collider to support object interaction, and a control script written in the C# programming language. This script functions to read serial data sent by the Arduino, interpret the received numeric signals, and control the direction and movement status of the virtual object according to the designed interaction mechanism.

The Unity scene display and virtual object configuration are shown in Figure 3. In this implementation, each signal received from the Arduino will trigger a change in the object's condition, namely moving left or right, or stopping at the last position based on the toggle principle. Thus, the user's physical actions via push buttons can directly produce a consistent visual response and can be observed in real-time in the Unity environment.

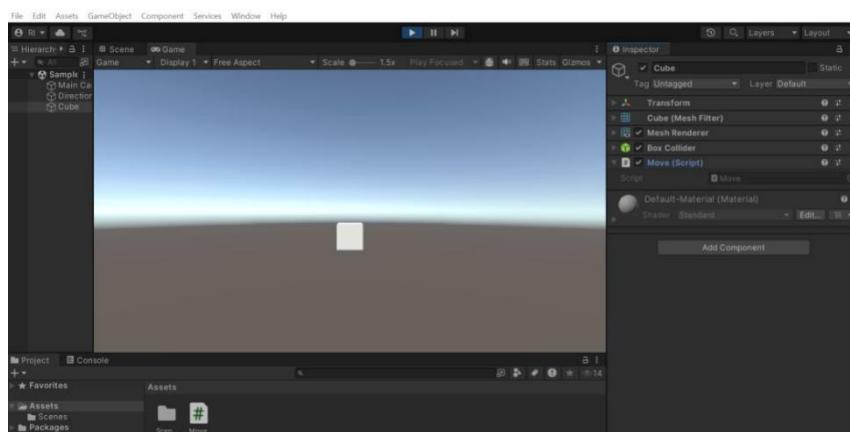


Figure 3. Implementation of virtual objects in the Unity environment

Scope and Object of Research

The scope of this research is limited to the development and testing of a push-button-based Tangible User Interface (TUI) system to control the movement of virtual objects in the Unity environment. The control system studied only covers two-way object movement, namely left and right, using two push buttons as physical input devices. The interaction mechanism is designed using the toggle principle, where one press of a button will activate the movement of the virtual object, while the next press will stop the movement and maintain the object's last position.

The research object is a TUI system that integrates physical user interaction with digital visual responses. This system consists of hardware in the form of Arduino and push buttons, and software in the form of a Unity application that displays virtual objects and processes input data. This research does not discuss aspects of complex graphical interfaces, user experience evaluation, or testing based on subjective user responses, so the focus of the research remains on the system engineering aspects and physical-digital interaction mechanisms.

Data Collection and Analysis Techniques

Data collection in this study was conducted through direct observation of system responses during functional testing. Testing was conducted by repeatedly pressing the left and right push buttons to observe changes in the behavior of virtual objects in the Unity environment. Data collected included the accuracy of the object's movement direction, the success of the toggle mechanism, and the stability of the system's response to physical user input.

Data analysis was conducted descriptively by comparing the results of the system implementation with the interaction mechanism design established during the design phase. The analysis focused on the system's ability to translate physical input into accurate, consistent, and real-time visual responses. The results were used to evaluate the system's success in meeting the research objectives and to identify potential further developments in the proposed TUI system.

3. Results and Discussion

This section presents the implementation and testing results of a push-button-based Tangible User Interface (TUI) system integrated with Unity, along with a discussion of the findings. The presentation of the results follows the stages of the research methodology, ensuring that each stage of the system design and implementation has interrelated results and discussions.

a. Hasil Implementasi Sistem Tangible User Interface

The implementation results show that the push button -based Tangible User Interface system was successfully realized according to the established design. The system is able to integrate hardware in the form of Arduino and push buttons with Unity software through serial

communication, so that the user's physical input can be translated into visual responses in a digital environment in real-time.

The relationships between user actions, system processes, and visual responses are summarized in Table 1, which describes the system's interaction mechanisms in a structured manner. This table serves as a formal representation of the interaction flow that was previously conceptually depicted in the system flowchart (Figure 1).

Table 1. Tangible User Interface System Interaction Mechanism

Component	Input (User Action)	System Process	Output (Respon pada Unity)
Push Button Kanan (D6)	Press the button once	Arduino reads the logic change on pin D6 and sends a numeric signal of value 2 to Unity via serial communication.	The virtual object starts moving towards the right continuously.
Push Button Kanan (D6)	Press the back button	Arduino detects the logic change and sends a stop signal to Unity according to the toggle mechanism.	The virtual object stops and settles at its last position.
Push Button Kiri (D7)	Press the button once	Arduino reads the logic change on pin D7 and sends a numeric signal of value 1 to Unity via serial communication.	The virtual object starts moving towards the left continuously.
Push Button Kiri (D7)	Press the back button	Arduino detects the logic change and sends a stop signal to Unity according to the toggle mechanism.	The virtual object stops and settles at its last position.

Based on Table 1, the system interaction mechanism is designed using the toggle principle, where one press of a button activates the object's movement, while the next press stops the movement. For the right push button (pin D6), the system sends a numeric signal of 2 to Unity, which is interpreted as a command to move to the right. Conversely, the left push button (pin D7) produces a numeric signal of 1, which is interpreted as a command to move to the left.

This process demonstrates that Arduino acts as a translator of the user's physical actions into meaningful digital signals, while Unity acts as a signal processor and visual response renderer. This separation of roles makes the system more modular and easier to develop. The clarity of the input-process-output relationships in Table 1 demonstrates that the system's interaction mechanisms have been designed systematically and consistently, a key characteristic of systems engineering research.

b. Results of Initial System Condition Testing

Initial testing is conducted to observe the system's behavior in a state without input, or in its initial state. In this state, the virtual object is in a central position and does not experience any movement. The initial state of the system is shown in Figure 4.

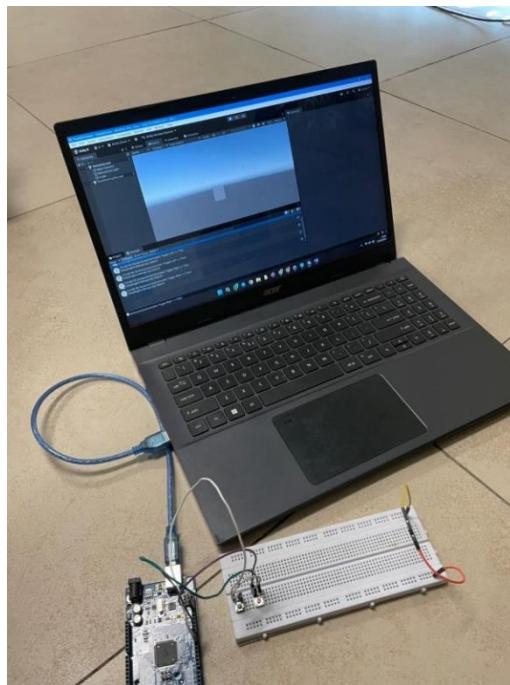


Figure 4. Initial state of virtual objects in the Unity environment

Figure 4 shows that the system is in a stable state before receiving input from the push button. The absence of object movement in the initial state indicates that the system does not produce an unwanted response (false trigger). This condition is important as a basis for system validation, as it ensures that virtual object movement occurs only as a result of the user's physical actions. Thus, Figure 4 serves as a baseline for the system's condition when it receives input.

Next, when the left push button is pressed once, the system sends a numeric signal that triggers the virtual object's continuous movement to the left. The object's movement to the left in response to physical input is shown in Figure 5.

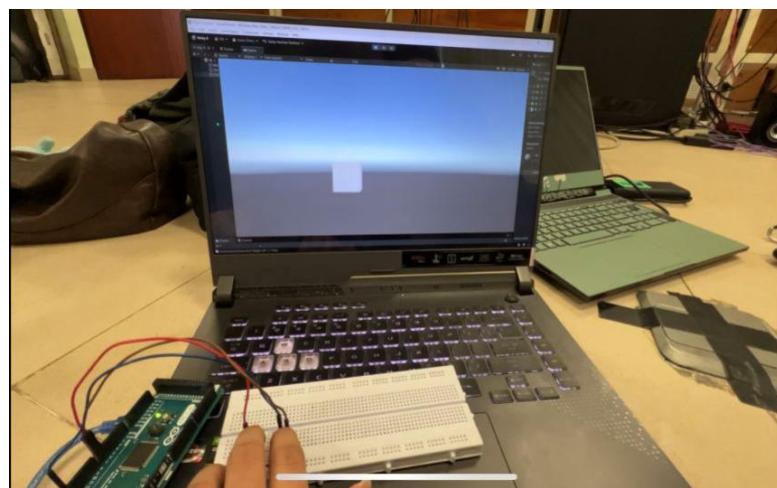


Figure 5. Response of virtual object movement to the left

Figure 5 shows the virtual object's position changing from its initial position to the left in response to physical user input. The object's movement continues until the next input is given. This demonstrates that the toggle mechanism works well, as the system does not require continuous button presses to maintain the object's movement. These results indicate that the system is able to consistently and stably translate left-hand input, while providing clear visual feedback to the user.

The same test was performed for the right push button. When the right push button was pressed once, the system sent a numeric signal of 2 to Unity, and the virtual object moved continuously to the right. The response to the object's movement to the right is shown in Figure 6.

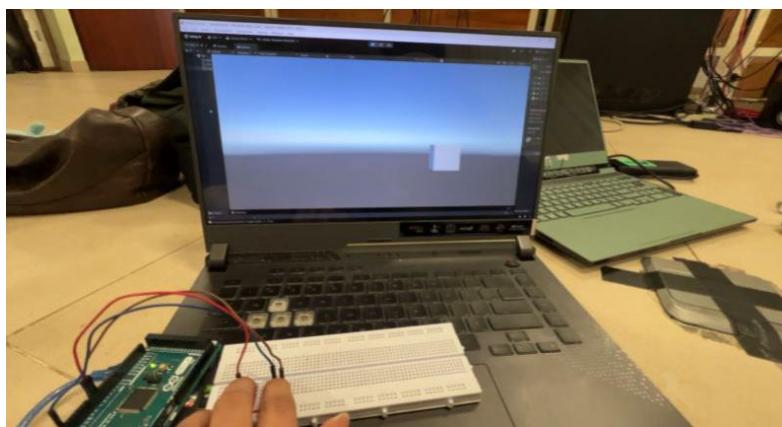


Figure 6. Response of virtual object movement to the right

Figure 6 shows that the system is able to distinguish right-hand input from left-hand input and generate an appropriate visual response. The virtual object moves to the right until the next button press stops the movement and maintains the object's final position. This consistent response demonstrates that the system is not only capable of reading physical input but also of accurately managing the object's movement status based on the designed interaction mechanism.

Overall, the test results indicate that the developed push-button-based Tangible User Interface system functioned according to the research objectives. The integration between hardware and software worked well, demonstrated by the system's real-time, stable, and consistent response to each user's physical input.

Compared to other studies that use complex sensors such as cameras or motion sensors, this study has the advantage of simplicity of design and ease of implementation. Despite using simple physical input, the system is still able to present a clear and easy-to-understand concept of physical-digital interaction. The uniqueness of this study lies in the use of a push-button-based toggle mechanism as a Tangible User Interface, which allows control of virtual objects without requiring continuous input and can be the basis for the development of more complex interactive systems in the future.

4. Conclusion

This research successfully designed and implemented a push button-based Tangible User Interface system integrated with Unity to control the movement of virtual objects in real-time. The developed system is able to connect the user's physical interaction with digital visual responses through the integration of Arduino hardware and Unity software using serial communication. The implemented interaction mechanism, namely toggle, allows users to control the movement of virtual objects to the left and right simply and consistently without requiring continuous input.

The test results show that the system can respond to every physical input stably and in accordance with the interaction mechanism design. The virtual object only moves when receiving a command from the push button and stops at the last position when the toggle mechanism is deactivated, so that the cause-and-effect relationship between user actions and system responses can be clearly observed. With a simple yet functional design, this system proves that simple physical input such as a push button can be utilized as an effective Tangible User Interface. This research is expected to be the basis for further development of interactive multimedia systems, simulations, and more complex physical-digital interaction-based applications.

References

Antle, A. N., & Wise, A. F. (2019). Getting down to details: Using theories of cognition and learning to inform tangible user interface design. *Interacting with Computers*, 31(4), 377–390. <https://doi.org/10.1093/iwc/iwz019>

Lindeman, R. W., Sibert, J. L., & Hahn, J. K. (2019). Towards usable VR: An empirical study of user interfaces for immersive virtual environments. In *Proceedings of the IEEE Virtual Reality Conference* (pp. 64–71). IEEE.

Preece, J., Rogers, Y., & Sharp, H. (2015). *Interaction design: Beyond human–computer interaction* (4th ed.). Wiley.

Salazar-Cardona, A. G., Muñoz-Sáenz, P. A., & Roldán-Gómez, J. J. (2023). Designing a tangible user interface (TUI) for the elderly based on their motivations and game elements. *Sensors*, 23(23), 9513. <https://doi.org/10.3390/s23239513>

Sheer, O., & Hornecker, E. (2010). Tangible user interfaces: Past, present, and future directions. *Foundations and Trends in Human–Computer Interaction*, 3(1–2), 1–137. <https://doi.org/10.1561/1100000026>

Uhrík, M., Koniar, D., & Novák, P. (2022). Augmented reality and tangible user interfaces as an extension of computational design tools. *Architecture Papers of the Faculty of Architecture and Design STU*, 27(2), 41–52. <https://doi.org/10.2478/alfa-2022-0021>

Unity Technologies. (2023). Unity user manual. <https://docs.unity3d.com/Manual/index.html>

Yildirim, B., Yildirim, S., & Celik, E. (2024). Learning circuits and coding with Arduino board in higher education using tangible and graphical user interfaces. *Information*, 15(5), 245. <https://doi.org/10.3390/info15050245>