Computerized Mind-Reading Magic Trick with Remote Transmission

Pu-Sheng Tsai[1], Ter-Feng Wu[2], En-Jia Hsu[1]*

[1] Department of Electronic Engineering, Ming Chuan University, Taiwan
[2] Department of Electrical Engineering, National Ilan University, Taiwan
pusheng@mail.mcu.edu.tw, tfwu@niu.edu.tw, hsuangel920130@gmail.com

Abstract

Magic is typically presented live, and computer presentation is relatively rare. This article is based on mind-reading magic and presents the magic process on a computer. Twenty-one cards are given to the spectator to make choices. Through the processes of dealing, collecting and questioning, the playing cards considered are transformed through mathematical derivation. This article uses the tree table lookup method to replace the mathematical algorithm used in the original magic. The program execution results can indeed achieve good results. Finally, we considered only displaying the card that the person was thinking about on the screen, but found this too monotonous and lacking in interactivity. Thus, the magic result will be presented on a touch panel or mobile webpage, so as to set off the final climax at the end of the magic. The playing card suits and numbers generated by the computer are transmitted to the touch panel via Bluetooth. Another method is to use Raspberry Pi to set up a server and create a database form, sending the card suits and numbers to the database form through Wi-Fi. Players can then connect to the IP address through their mobile phones to display the picture of the playing cards.

Keywords: Tree table lookup method, Raspberry, Touch panel, Mind-reading magic

1 Introduction

Many studies have shown that incorporating magic into teaching can significantly enhance students' curiosity and willingness to learn by combining education with entertainment [1-2]. For example, when teachers use playing cards to perform magic tricks in class, students are not only amazed by the tricks but also observe the mathematical techniques behind them and actively ask questions. By sparking curiosity, creating surprises, and promoting collaboration, educators can attract learners and stimulate their interest in science and mathematics. Mathematics teaching should be student-centered, with teachers and students collaboratively constructing and exploring mathematics through interaction and dialogue. This allows students to gain knowledge and cognitive skills through processes of thinking, transforming, and reflecting [3-4]. Magic-based teaching not only helps to convey scientific, mathematical, and computational concepts but also fosters creative thinking. Additionally, applying magic to computer programming instruction can attract students in an innovative and engaging way, sparking their interest and curiosity in learning programming while developing their skills. This approach not only captures students' attention, making them eager to understand and replicate the magic programs to show others, but also enhances their learning experience [5-6]. The studies by Beck *et al*. [7-8] analyzes how the application of mathematical magic in teaching influences students' learning attitudes and performance. Through the method of action research, it helps teachers reflect on and improve their teaching methods, thereby further enhancing their professional teaching skills.

This article utilizes Python programming language to develop a computerized version of mind-reading magic. Python's robust support for extensive libraries and powerful import package makes it an ideal choice. Using a graphical user interface (GUI) toolbox for developing the window interface is highly recommended [9]. To align the computerized card magic performance with real-life scenarios, the program incorporates various ingenious elements for dealing and collecting, such as card stacking effects and fan-shaped card arrangements. The magic trick starts with suspenseful music to create a mysterious atmosphere, complementing a window that appears onscreen, inviting the audience into the world of magic. Furthermore, this article employs a Raspberry Pi4 to set up a server and create database forms [10]. The Raspberry Pi4 is a 64-bit embedded microprocessor developed based on the Linux operating system. Its powerful functions and computing capabilities are almost equivalent to a complete computer, capable of connecting peripherals such as keyboards, mice, and monitors, and it includes I/O pins and communication interfaces like GPIO, I2C, and UART [11-12]. A small and low-cost embedded microprocessor, Raspberry Pi, in this system not only replaces the personal computer but also establishes an IoT server. In addition, this paper uses Bluetooth and Wi-Fi for remote transmission, sending the magic results to a remote touch panel or webpage. Relevant studies can be referred to in references [13-14]. One study uses IoT to integrate the GPS tracking data of cyclists into Google Maps. Another study uses a touch panel to control a six-axis robotic arm, completing tasks assigned by a teaching robot.

*Corresponding Author: Ter-Feng Wu; E-mail: tfwu@niu.edu.tw
DOI: https://doi.org/10.70003/160792642025032602012

2 System Description

This study presents the process of a mind-reading magic performance on a computer screen using a Python program. The program includes the design of the magic trick, the principles behind it, and the employed algorithm. The magician hands 21 cards to the audience, and then asks them to pick one and remember it. The first round of dealing begins, with the cards divided into three groups of seven cards each. The magician asks the person which group contains the card they have in mind. After receiving an answer, the cards are collected, ensuring that the group with the selected card is always placed in the middle, with the other two groups placed above and below it. This procedure is repeated three times. After three rounds of dealing, asking, and collecting, the card thought of will inevitably appear as the eleventh card in the set of 21. We can prove that "for any N cards, if we repeatedly deal, ask, and collect according to the mentioned rules for r times, the card in mind will automatically move to the position $int(N/2) + 1$ among the N cards".

To enhance the interactivity of the card display, we aim to transmit the suit and number of the card the audience member thought of to a touch panel (MT6071iE) via Bluetooth, creating a suspenseful effect. Another way to reveal the card is to set up a private cloud using Raspberry Pi. The suit and number of the card are uploaded to a cloud database via Wi-Fi, allowing the person to view the card's suit through a webpage on their mobile phones. The entire system architecture is shown in Figure 1.

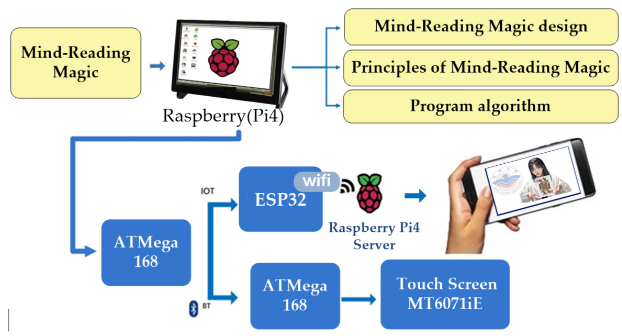
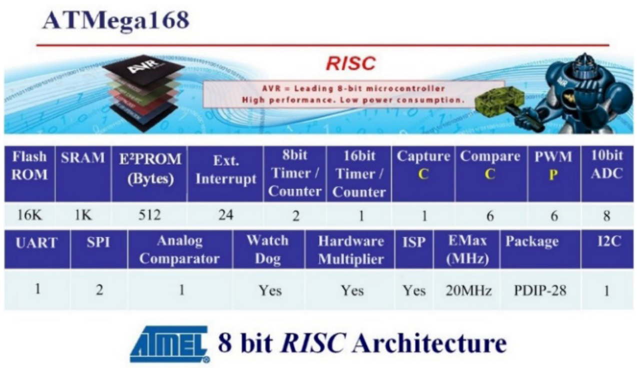


Figure 1. System description

2.1 Embedded Microcontroller

The ATmega168 is an 8-bit RISC microcontroller chip manufactured by Atmel Corporation, serving as the core chip in the system architecture described in Figure 1. The built-in memory includes programmable flash memory (ROM), totaling 16K, EEPROM totaling 512 bytes, and SRAM totaling 1K bytes. It includes features such as a watchdog timer, in-system programming (ISP) capability, a structure with multiple interrupt vectors, and three sets of timers/counters that can be individually configured for output comparison (Compare), capture mode (Capture), and pulse width modulation generator (PWM). Additionally, it has eight channels of 10-bit analog-to-digital converter (ADC), programmable asynchronous serial communication interface (UART), serial peripheral interface (SPI), and inter-integrated circuit (I2C), among others. For detailed specifications, please refer to Figure 2.



Flash ROM	SRAM	E²PROM (Bytes)	Ext. Interrupt	8bit Timer / Counter	16bit Timer / Counter	Capture C	Compare C	PWM P	10bit ADC
16K	1K	512	24	2	1	1	6	6	8

UART	SPI	Analog Comparator	Watch Dog	Hardware Multiplier	ISP	EMax (MHz)	Package	I2C
1	2	1	Yes	Yes	Yes	20MHz	PDIP-28	1

Figure 2. ATmega168 specifications

2.2 Embedded Microprocessor

This article uses the Raspberry Pi 4 as a personal cloud server. The key specifications were as follows: (1) Broadcom BCM2711 chipset; (2) Quad-core 64-bit Cortex-A72 (ARM V8) microprocessor running at 1.5GHz; (3) Dual-core VideoCore IV 3D graphics core; (4) 4GB of LPDDR2 memory; (5) 2.4 GHz/5.0 GHz IEEE 802.11b/g/n/ac wireless LAN and Bluetooth 5.0 BLE; (6) HDMI output supporting rev. 1.3, 1.4, composite video jack supporting NTSC, PAL, and 3.5mm audio jack; (7) 2 USB 3.0 ports and 2 USB 2.0 ports; (8) 40-pin 2.54mm header providing 27 GPIO pins and power pins, including +3.3V, +5V, GND; (9) 15-pin MIPI CSI-2 camera interface; (10) Display Serial Interface (DSI) display connector; (11) micro SD card reader supporting SDIO; (12) Operating System booting from micro SD card with support for Linux and Windows. The dimensions of the device were as follows: 85mm x 56mm x 17mm. Please refer to Figure 3.

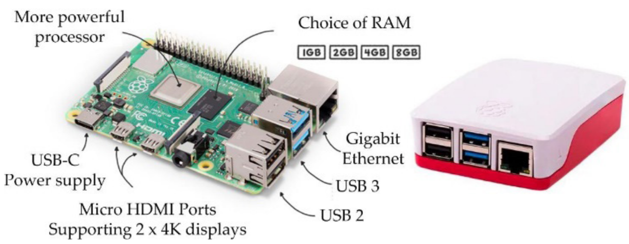


Figure 3. Appearance and specifications of the raspberry Pi4

2.3 Touch Panel

As shown in Figure 4, interactions between the magician and audience are conducted on a human-machine interface utilizing an MT6071iE touch panel (Weintek) with the following specifications: (1) 7" TFT backlit LCD panel; (2) 800 x 480 px resolution; (3) Four-wire resistive touch; (4) Project file download via USB Client' (5) COM1 (supporting RS232), COM2, and COM3 (supporting RS485) serial communication ports; (6) NEMA4/IP65 protection rating and built-in power isolation; (7) Microprocessor using 32-bit RISC Cortex A8 operating at 600MHz; (8) 128MB of internal flash memory and 128MB of RAM; (9) Built-in perpetual calendar. The Power consumption of the device was 350mA at 24Vdc.

Figure 4. Touch panel MT6071iE

3 Software Programming

This article uses a computer to perform a mind-reading card trick, employing mathematical sorting to reveal a person chosen card. The magician hands 21 cards to the spectator, who selects one to memorize. The cards are divided into three piles, and the audience indicates which pile contains their card, as shown in Figure 5. Through the algorithm, after three rounds of dealing, questioning, and gathering the cards, the program will deduce the card that the person has in mind.

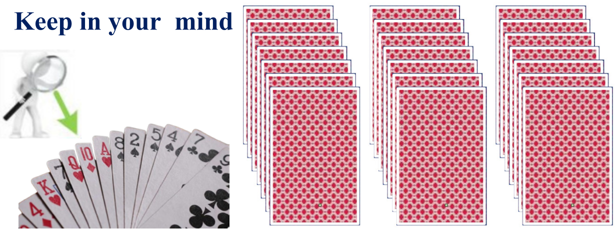


Figure 5. A mind-reading card trick

3.1 Underlying Principles

One deck consists of 54 playing cards. From these, 21 cards are selected to perform a magic trick, as shown in Figure 7. The cards are sequentially named $P[k]$, $k = 1 \ldots 21$. Suppose the card chosen by the person is the fourth card, coded as $P[4]$. The magician sequentially divides the cards into three piles and questions the person. During the first inquiry, the card is located in the second position of the first pile, as shown in Figure 6.

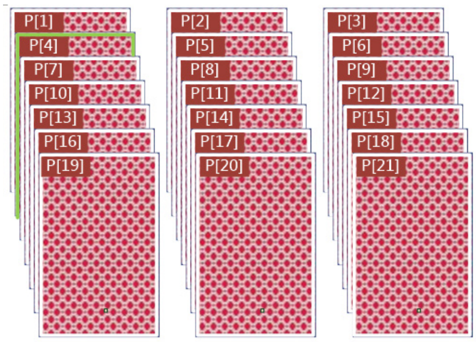


Figure 6. Three card piles in the first inquiry

After confirming with the spectator, the magician retrieves the cards. The card collection process is the most critical step of this trick. Since the card in mind is in the first pile, it is essential to place the first pile in the middle during collection, with the other two piles placed on the top and bottom, as shown in Figure 7.

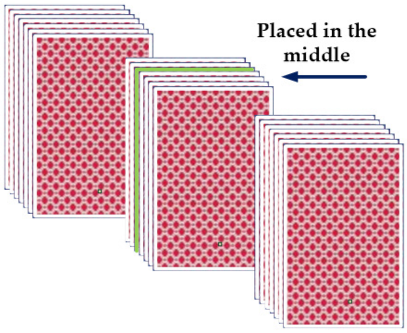


Figure 7. Follow the card collection rules

After re-dealing, the order of the cards becomes P[2], P[5], ..., P[20], P[1], P[4], P[7], ..., P[19], P[3], P[6], ..., P[21]. When divided into three piles again, the arrangement of the cards in each pile follows the sequence. For the second inquiry, the card appears as the third card in the third pile, as shown in Figure 8.

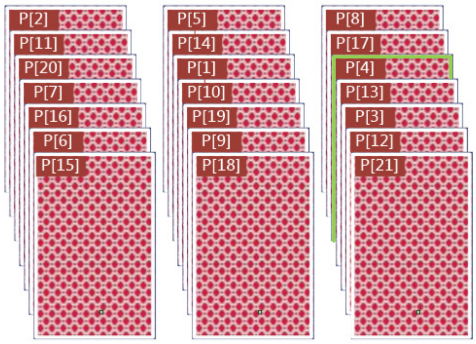


Figure 8. Three card piles in the second inquiry

Place the third pile in the middle, and place the first and second piles above and below the third pile, respectively. Repeat the previous action. This time, during the inquiry, the card is located in the fourth position in the first pile, as shown in Figure 9.

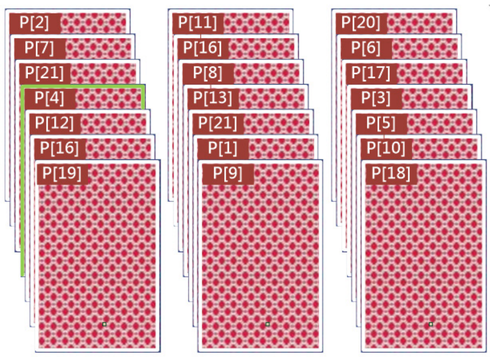


Figure 9. Three card piles in the third inquiry

After repeating the process of dealing, questioning, and gathering the cards three times, the magician performs the final collection, placing the first pile in the middle and the other two piles on the top and bottom. The resulting order of the deck is shown in Figure 10. This example demonstrates that, regardless of which card the spectator is thinking of, after repeating the dealing, questioning, and collecting process three times, the chosen card will automatically move to the eleventh position in the deck.

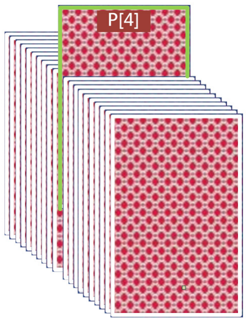


Figure 10. The selected card is in the eleventh position

3.2 Mathematical Derivation

To facilitate the analysis, the total number of cards is denoted as N, which is divided into 3 piles, each containing M cards, such that $M = N/3$. The position of a card is denoted as $P[k]$. For clarification of the card arrangement, we redefine the position as $P[m, n]$, where $k = 3(n - 1) + m$. Here, $n = 1,2,3$ represents the pile number, and $m = 1 \ldots M$ indicates the position within the n-th pile. The position $P[k]$ can then be determined as follows:

[Case I] If k mod $3 \neq 0$, where *mod* represents the remainder, then
the pile in which $P[k]$ is located: $n = k \bmod 3$ **(1)**
and the position of $P[k]$ in the pile: $m = (k \div 3) + 1$. **(2)**

[Case II] If $k \bmod 3 = 0$, then
the pile in which $P[k]$ is located: $n = 3$ **(3)**
and the position of $P[k]$ in the pile: $m = (k \div 3)$ **(4)**

Consider the example of 21 cards in this study, where $N = 21$, $M = N/3 = 7$, $n = 1,2,3$, $m = 1 \ldots 7$, $k = 1 \ldots 21$. During the first dealing, the target card $P[k]$ in mind may appear in any position within the three piles, as shown in Figure 11 and Figure 12 with the red-labeled card. In the first collection of cards, following the previously defined rules, the card the audience is thinking of may appear at $P[k]$, where $k = 8 \ldots 14$, reducing the range from 21 cards to 7 cards, as shown in Figure 11 with the blue-labeled cards. This can be expressed as $P[m, n]$, where $\{n = 1, m = 4{,}5 \; or \; n = 2, m = 3{,}5 \; or \; n = 3, m = 3{,}4\}$ as shown in Figure 12. In the second deal, the cards are collected as previously described, narrowing the audience's card to $P[k]$, $k = 10 \ldots 12$, as shown with gray-labeled cards in Figure 11. This can also be expressed as $P[m, n]$, where $\{m = 1, n = 4 \; or \; m = 2, n = 4 \; or \; m = 3, n = 4\}$ as shown in Figure 12. In the third deal, following the previously described collection method, the first seven cards are guaranteed not to be the chosen card. The selected card will be the fourth card in the second pile, corresponding to the eleventh position among the 21 cards, as indicated by the black-labeled card in Figure 11 and Figure 12. In other words, the card thought of appears at position $P[11]$, which can also be expressed as $P[2,4]$.

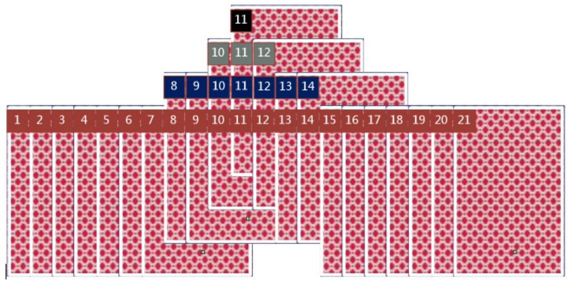


Figure 11. Target's position after collection

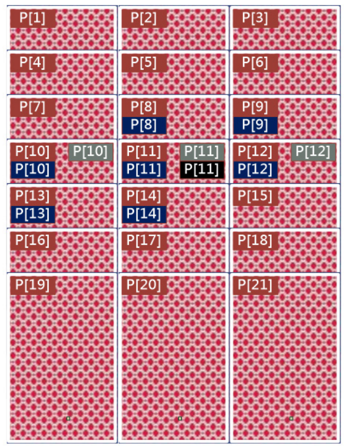


Figure 12. Target's position in three piles

3.3 Extension

In previous discussions, the number of cards was limited to 21. This section will go beyond that restriction for broader applications. When the total number of cards is a multiple of three and an odd number, repeating the processes of dealing, inquiring, and following the previously established card collection procedure r times will automatically move the chosen card to the center, regardless of which card is selected. The number of playing cards is N, which is divided into three equal parts, each containing M cards, where $M = N/3$. During the first deal, the chosen card, $P[k]$ ($k = 1 \ldots N$), could potentially appear in any position within the three piles. By following the previously established collection method, after the second deal, the card silently chosen by the spectator will shift to position $P[M + i]$, where $i = 1 \ldots M$. This can further be expressed as $P[m, n]$, where $n = 1$ to 3, and $m = \min(m)$ to $\max(m)$. Consider the example in Figure 12, where any card in the first deal could be the target, so $n = 1 \ldots 3$ and $m = 1 \ldots 8$, where $\min(m) = 3$, $\max(m) = 5$. During the second reveal, the card silently chosen will appear in $P[k]$, $k = 8 \ldots 14$. Another way to express this is $P[m,n]$, where $\{n = 1, m = 4{,}5 \; or \; n = 2, m = 3{,}5 \; or \; n = 3, m = 3{,}4\}$. In other words, $n = 1 \ldots 3$, and $m = 3 \ldots 5$, where $\min(m) = 3$, $\max(m) = 5$. It is clear that by the next reveal, the magician has narrowed down the card's position to $P[M + \min(m)] \ldots P[M + \max(m)]$. In fact, m represents the position of the card $P[k]$ within each pile. For clarity, by substituting (2) and (4) in the formulas, it can be derived that

[Case I] If $(M + i) \bmod 3 \neq 0$, $i = \min(m) \ldots \max(m)$,
$$\text{next reveal } \min(m) = [(M + \min(m)) \div 3] + 1 \quad \textbf{(5)}$$
$$\text{next reveal } \max(m) = [(M + \max(m)) \div 3] + 1 \quad \textbf{(6)}$$

[Case II] If $(M + i) \bmod 3 = 0$, $i = \min(m) \ldots \max(m)$,
$$\text{next reveal } \min(m) = [(M + \min(m)) \div 3] \quad \textbf{(7)}$$
$$\text{next reveal } \max(m) = [(M + \max(m)) \div 3] \quad \textbf{(8)}$$

Repeat the actions of dealing, inquiring, and collecting the cards. With each deal, the position of the target card will further narrow down to $P[M + i]$, $i = \min(m) \ldots \max(m)$. Suppose after r rounds of inquiry, the condition $\min(m) = \max(m)$ is satisfied. This indicates that the chosen card has converged to $P[(3M + 1)/2]$ and has automatically moved to the middle of the deck, marking the end of the magic performance. Using a deck of 93 cards as an example, let $N = 93$ and $M = 93/3 = 31$.

- First deal: The chosen card may appear at any positions among the 93 cards.
- Second deal: The chosen card may appear at positions $P[M + i]$, $i = 1 \ldots 31$. From formulas (5) … (8) yields $\min(m) = 11$, $\max(m) = 21$.
- Third deal: The range of the chosen card narrows to $P[M + i]$, $i = 11 \ldots 21$. From formulas (5) … (8) yields $\min(m) = 14$, $\max(m) = 18$.
- Fourth deal: The chosen card is further narrowed down to $P[M + i]$, $i = 14 \ldots 18$. From formulas (5) … (8) yields $\min(m) = 15$, $\max(m) = 17$.
- Fifth deal: The chosen card has already moved to the middle position of the deck, $P[M + i]$, $i = 15,16,17$, and satisfies $\min(m) = \max(m) = 16$.

From the above analysis, it can be concluded that the number of times dealing, asking, and collecting cards (r) is related to the number of expanded playing cards (N). This relationship can be expressed by the following equation,

$$N \leq 3^{\overline{r}}, r = \min(\overline{r}) \quad \textbf{(9)}$$

3.4 Algorithm Programming

During the first dealing, the cards are divided into three piles, with seven cards in each pile from left to right. The arrangement of each pile can be represented by $P[m, n]$ with $n = 1 \ldots 3$, $m = 1 \ldots 7$ indicating that each pile has seven cards. The computer sequentially asks the spectator if the chosen card is in each of the three piles. When the person confirms the pile with the card, that pile is placed in the middle, and the other two piles can be placed on either side. Following this card collection rule, the order of the entire deck is rearranged. The computer then deals the cards again into three piles, but the order of the cards in each pile has been significantly altered. The numbers of the cards in the first pile become [2, 11, 20, 7, 16, 6, 15]. These numbers are now difficult to represent using a mathematical formula. The same problem occurs with the second and third piles. The numbers in each pile become more chaotic, making it impossible to solve using a mathematical formula. Therefore, a tree diagram lookup method is used to replace the original mathematical algorithm.

First, defining new identifiers facilitates the use of the tree table lookup method. The poke_1_1 represents the array variable of the first pile after the first deal. A "1" before denotes the first deal, and a "1" after denotes the first pile. Similarly, poke_1_2 and poke_1_3 represent the array variables of the second and third piles respectively after the first deal. For the second deal, an additional pointer is required in the array variable. For example, poke_2_1_1 indicates the second deal, with "1" in the middle representing the pile where the audience's selected card appeared during the first deal, and "1" at the end representing the array variable of the first pile in the second deal. Similarly, for the third deal, an additional pointer in the array variable is needed. For instance, poke_3_2_3_2 signifies that during the first deal, the selected card appeared in the second pile; in the second deal, it appeared in the third pile; and in the third deal, it represents the array variable of the second pile.

Based on the provided information, the array variables are defined:

(1) After the first deal, the array variables for the first, second, and third piles are as follows:

$$poke\_1\_i,\ i = 1 \ldots 3, \quad \textbf{(10)}$$

(2) After the second deal, the array variables for the first, second, and third piles are as follows:

$$poke\_2\_j\_i,\ i = 1 \ldots 3,\ j = 1 \ldots 3, \quad \textbf{(11)}$$

The array variable notation for the second deal depends on which pile the silently chosen card appeared during the previous deal. Index j indicates the pile where the card appeared in the first deal.

(3) After the third deal, the array variables for the first, second, and third piles are as follows:

$$poke\_3\_k\_j\_i,\ i = 1 \ldots 3,\ j = 1 \ldots 3,\ k = 1 \ldots 3, \quad \textbf{(12)}$$

The array variable used for the third deal requires an additional index, indicating the pile k in which the chosen card appeared during the second deal.

3.5 Tree Table Lookup Method

After defining the array variables, create a program flowchart to be used for the structure diagram of the tree table lookup method, as shown in Figure 13. All possible branches encountered during the magic process are presented in the tree diagram. The branches are defined as "YES" on the top and "NO" on the bottom. Let's start by explaining from the leftmost node, poke_1_1. Previously, we defined poke_1_1 as the array variable for the first pile after the first deal.

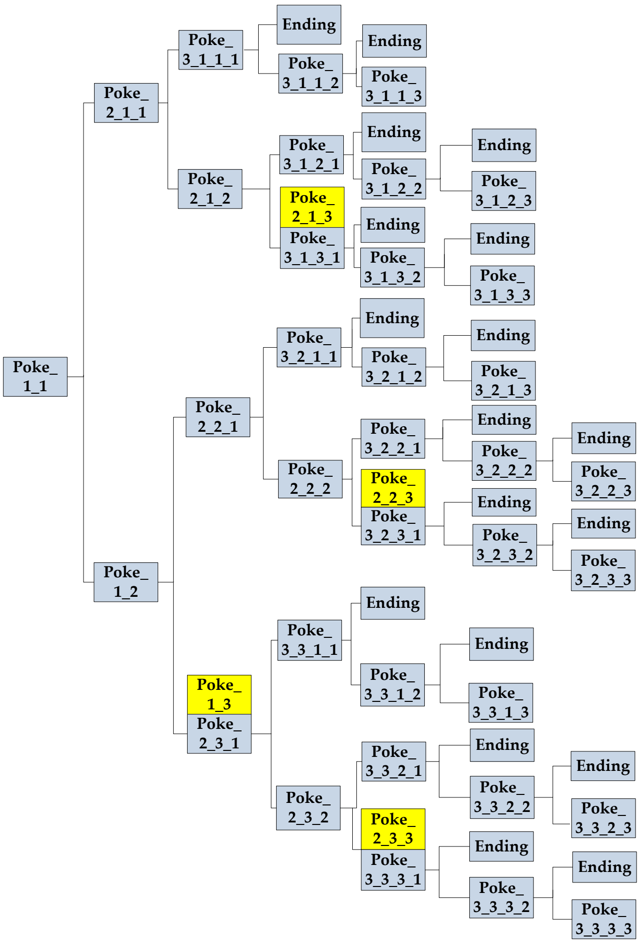


Figure 13. Tree diagram for table lookup algorithm

$$poke\_1\_1 = [1,4,7,10,13,16,19], \quad \textbf{(13)}$$

The magician will ask the spectator if the chosen card is in this pile. There are only two possible answers, "YES" or "NO" which will be discussed separately.

Case I: If "YES" is chosen, it means the card the person is thinking of is in this stack. There is no need to inquire further about the other two stacks. Proceed directly to collect the cards and conduct the second dealing. The tree diagram branches upward, with the array variable being poke_2_1_1, indicating that in the first dealing, the target is found in the first stack of cards, where

$$poke\_2\_1\_1 = [2,11,20,7,16,6,15], \quad \textbf{(14)}$$

Case II: If "NO" is chosen, it means the card the person is thinking of is not in the first stack. The program continues by revealing the second stack and querying the spectator. Therefore, the tree diagram branches downwards, with the array variable defined as poke_1_2, where

$$poke\_1\_2 = [2,5,8,11,14,17,20], \quad \textbf{(15)}$$

In summary, the entire tree diagram covers all possible scenarios during the magic performance. Depending on the location of the card the selected card, different branches lead to different outcomes. In Figure 14, "ending" indicates the magician has queried three times, determining the target card as the eleventh card among 21. Users determine each branch's path by pressing "YES" or "NO", with details of each branch's array variable content shown in Figure 14.

Poke_1_1={1,4,7,10,13,16,19}	Poke_1_2={2,5,8,11,14,17,20}	Poke_1_3={3,6,9,12,15,18,21}
Poke_2_1_1={2,11,20,7,16,6,15}	Poke_2_1_2={5,14,1,10,19,9,18}	Poke_2_1_3={8,17,4,13,3,12,21}
Poke_2_2_1={1,10,19,8,17,6,15}	Poke_2_2_2={4,13,2,11,20,9,18}	Poke_2_2_3={7,16,5,14,3,12,21}
Poke_2_3_1={1,10,19,9,18,5,14}	Poke_2_3_2={4,13,3,12,21,8,17}	Poke_2_3_3={7,16,6,15,2,11,20}
Poke_3_1_1_1={5,10,18,20,6,17,3}	Poke_3_1_1_2={14,19,2,7,15,4,12}	Poke_3_1_1_3={1,9,11,16,8,13,21}
Poke_3_1_2_1={2,7,15,1,9,17,3}	Poke_3_1_2_2={11,16,5,10,18,4,12}	Poke_3_1_2_3={20,6,14,19,8,13,21}
Poke_3_1_3_1={2,7,15,4,12,14,19}	Poke_3_1_3_2={11,16,8,13,21,1,9}	Poke_3_1_3_3={20,6,17,3,5,10,18}
Poke_3_2_1_1={4,11,18,19,6,16,3}	Poke_3_2_1_2={13,20,1,8,15,5,12}	Poke_3_2_1_3={2,9,10,17,7,14,21}
Poke_3_2_2_1={1,8,15,2,9,16,3}	Poke_3_2_2_2={10,17,4,11,18,5,12}	Poke_3_2_2_3={19,6,13,20,7,14,21}
Poke_3_2_3_1={1,8,15,5,12,13,20}	Poke_3_2_3_2={10,17,7,14,21,2,9}	Poke_3_2_3_3={19,6,16,3,4,11,18}
Poke_3_3_1_1={4,12,17,19,5,16,2}	Poke_3_3_1_2={13,21,1,9,14,6,11}	Poke_3_3_1_3={3,8,10,18,7,15,20}
Poke_3_3_2_1={1,9,14,3,8,16,2}	Poke_3_3_2_2={10,18,4,12,17,6,11}	Poke_3_3_2_3={19,5,13,21,7,15,20}
Poke_3_3_3_1={1,9,14,6,11,13,21}	Poke_3_3_3_2={10,18,7,15,20,3,8}	Poke_3_3_3_3={19,5,16,2,4,12,17}

Figure 14. Content of each branch

4 Remote Transmission

After the target card has been identified, the virtual magician performs a "reveal". However, a simple presentation of the target card would be somewhat anti-climactic. In this study, we emulated the remote transmission trick used by real-world magicians to create suspense and leave a lasting impact. This illusion could be performed using a Bluetooth transmission or WiFi.

4.1 Bluetooth Wireless Transmission

The embedded microcontroller connects to the Bluetooth module via serial communication (UART). It transmits the suit and number of the card to the remote MT6071iE touch panel, thereby displaying the card the audience is thinking of to create a sense of suspense. The hardware architecture is shown in Figure 15. This article utilizes Atmel's microcontroller ATMega168 to send or receive signals from the Bluetooth wireless transmission module. After the magic program concludes, the embedded microcontroller Raspberry Pi (Pi4) transmits the card suit and rank codes to the embedded microcontroller via the USB communication interface. The suit and rank codes are transmitted to a remote microcontroller via the Bluetooth module and then connected to the touchscreen panel MT6071iE using an RS232 communication interface, as seen in Figure 15. The touchscreen panel receives the suit and rank codes, and the screen displays the card revealed by the magician. The touchscreen panel is placed inside a remote box. When the person opens the box, the card they have silently thought of appears inside, creating a shocking effect.

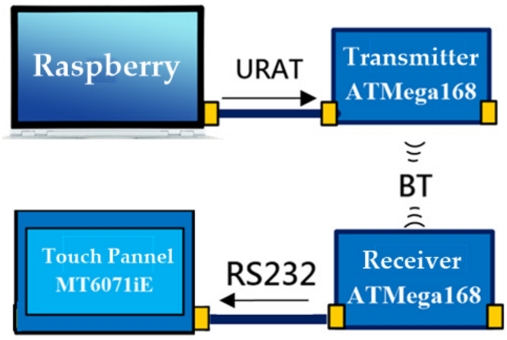


Figure 15. Bluetooth transmission framework

Bluetooth transmission is a short-range wireless radio transmission technology. Note that the Bluetooth module was pre-paired and configured to allow automatic connections. After each device is powered on, a red LED blinks to indicate pairing. Both LEDs blinking simultaneously indicates that the devices have been paired and connected.

4.2 Wi-Fi Transmission

Another option involves the use of Wi-Fi transmission to send the suits and numbers of playing cards to a remote server's database. Through a mobile phone, tablet, or computer connected to the internet, users can access and retrieve the data, displaying the card the person is thinking of on a webpage. To realize this creative concept, this paper designs an IoT sensor system architecture, as shown in Figure 16.

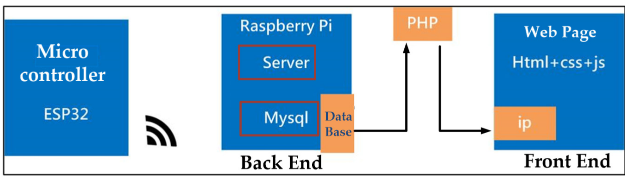


Figure 16. Structure of Wi-Fi transmission

This study utilizes the embedded microcontroller ESP32 to receive the suits and numbers of playing cards transmitted from the Raspberry Pi (Pi4) master control platform. These data are then sent via Wi-Fi to the server's database. An Apache web server is set up on the Raspberry Pi, with PHP and MySQL database systems also installed on this server. On one hand, the PHP program processes the received data and stores it in the MySQL database. On the other hand, the PHP program retrieves data from the database, updates the suits and numbers of the playing cards, and generates webpage content for users to browse online. In Figure 17, the client side on the left wants to browse a webpage. It must send an HTTP request to the server side on the right to request a webpage. Apache is a server software specifically designed for handling HTML services. Once the webpage service request is received, the corresponding HTML document is sent back to the client's browser. The PHP interpreter on the server side can directly embed PHP code into HTML pages, which are then processed by Apache. This study involves setting up a server on Raspberry Pi, creating a MySQL database, and designing dynamic PHP webpages. The MySQL database includes tables for playing card suits and numbers. Thus, a dynamic PHP-based app package, titled "Remote Connection Mind-Reading Magic" is developed. This app connects to the MySQL database and allows users to browse in it on tablets or smartphones. The process of setting up an IoT system on a Raspberry Pi and installing web server software such as Apache, PHP, and MySQL is shown in Figure 18.

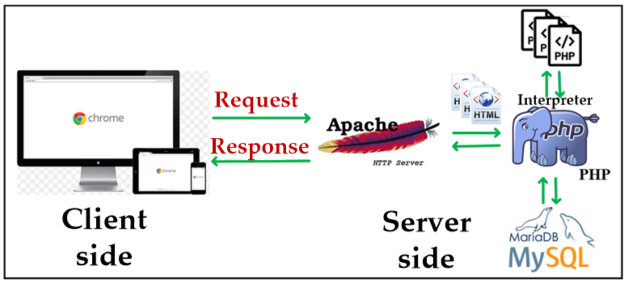


Figure 17. Proposed web server setup

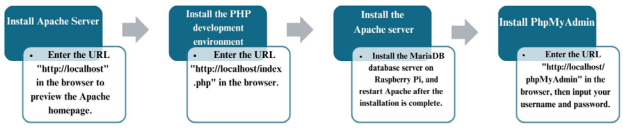


Figure 18. Web server setup procedure

5 Experimental Results

The mind-reading magic software begins with suspenseful music. To create a mysterious atmosphere, the screen displays prompts such as "Enter the magical world and witness miracles," as shown in Figure 19. Clicking the "Next" button leads to another screen, illustrated in Figure 20, where 21 playing cards are displayed, and the person is instructed to mentally choose and remember one card. Using Python's Tkinter's Canvas package and the ImageTk function from the PIL package, a GUI is constructed. Based on the principles of magic card tricks and magic algorithms, a complete program is developed to present the interactive process of the classic 21-card mind-reading magic trick. Combining modern popular AI-generated portraits, real photos are used to create AI virtual avatars for designing cover and back cover images. Figure 19 is used as the cover image.

Figure 19. Magic show homepage

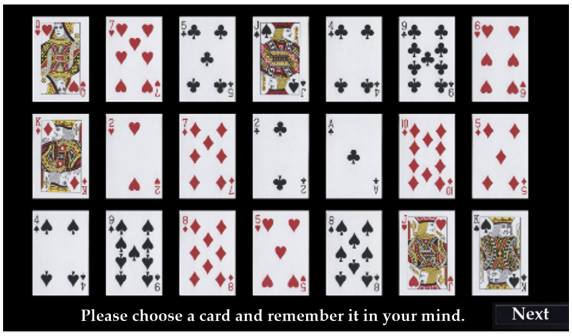
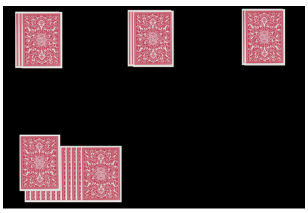


Figure 20. Pick a card and remember it

To ensure that computer magic aligns seamlessly with a real-life performance, the program incorporates various clever techniques when handling card dealing and collection. For example, as shown in Figure 21, the bottom-left corner displays a stack of playing cards. During the dealing process, the cards in the bottom-left stack gradually decrease, while the three piles increase sequentially. The program deliberately creates a stacking effect for the playing cards, achieving a flawless sense of realism. In the scene shown in Figure 22, the program asks the audience whether the card they have in mind is present in a particular pile. To present the cards for selection, the program arranges them in a fan-shaped layout. To display all 21 cards in this manner, it first rotates 21 image files at different angles sequentially, based on a reference coordinate. As shown in Figure 23, the program focuses on the movement effects of playing cards during dealing or collecting. The animation involves not only horizontal and vertical movements but also movements across different positions. The program must move each card to its correct location one by one while maintaining a stacking effect, significantly increasing the complexity of the programming. Next, the process of three rounds of dealing cards, questioning, and collecting cards is completed. According to the magic algorithm discussed in Section 3.4, the card in mind should appear in the middle of the deck. In the program, an AI-generated virtual character serves as the background, and the revealed card will slowly move to the hand of the magical girl, as shown in Figure 24.



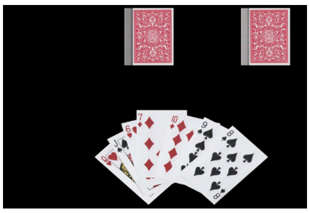
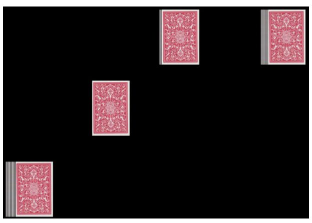


Figure 21. Card-dealing screen **Figure 22.** Audience prompt screen



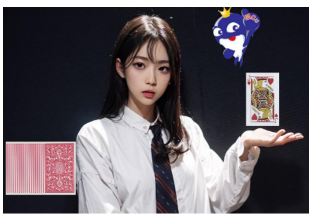


Figure 23. Card collection screen **Figure 24.** End of magic screen

Simply displaying the final revealed card of the mind-reading magic trick on the screen appears lackluster. Using the remote transmission technology discussed in Section 4, we transmit the suit and number of the card via Bluetooth to a hidden small box. The person is then invited to open the box, where the card dramatically appears on the box's touchscreen, enhancing the suspenseful effect of the magic trick, as shown in Figure 25. Another method to reveal the final card is to transmit it to a remote server using the Wi-Fi module built into the ESP32 microcontroller. Using IoT technology, the suit and number of the final card are transmitted to the MySQL database on the server. The participant is invited to take out a phone, connect to the server's IP address, and watch as the card in mind astonishingly appears on their screen, shown in Figure 26. By combining IoT technology and Bluetooth wireless transmission, a mind-reading magic performance is realized. This innovative approach offers the audience a refreshing and exciting experience.

Figure 25. Touchscreen display **Figure 26.** Mobile screen

6 Conclusions

This thesis is based on mathematical magic, utilizing algorithms and magic principles to create a mind-reading magic using Python GUI as the primary programming method. In the future, it will be demonstrated with Raspberry Pi and the MT6071iE touch panel, making it more portable and interactive for magic education. The program results will be displayed on the touch panel via Bluetooth for audience interaction, and transmitted to smartphones through IoT, showcasing the complete design and operational process of the mind-reading magic. Students often feel intimidated by mathematics. Promoting mathematical magic widely could significantly reduce this fear by using magic to engage people's interest in mathematics and help them understand its fun and practical applications. Moreover, mathematics enhances memory and concentration, boosting individuals' confidence and communication skills.

Through the experiment, we successfully validated the functionality and stability of the designed system. The suits and numbers of the playing cards were accurately transmitted to the remote server and displayed on the user's webpage. Moreover, the system's response time met the real-time requirements, achieving the expected performance in both data processing and webpage updates. Feedback from the audience indicated that the interactive experience was highly innovative and entertaining, demonstrating the system's effectiveness in performing the mind-reading magic show.

References

[1] P. Schott, The use of magic in mathematics : from primary school to higher education, *International Conference of Education, Research and Innovation(ICERI)*, Madrid, Spain, 2009, pp. 58-70.

[2] N. M. Velez, Mathematical Magic: A Study of Number Puzzles, *Rose-Hulman Undergraduate Mathematics Journal*, Vol. 21, No. 2, Article 1, 2020.

[3] A. Krapp, Interest, motivation and learning: An educational-psychological perspective, *European Journal of Psychology of Education*, Vol. 14, No. 1, pp. 23-40, March, 1999.

[4] C. Liang, Y.-H. Chiang, H.-Y. Chang, Integrating factors of online game stickiness and e-learning continuity to develop the game-based learning materials, *Instructional Technology and Media*, Vol. 83, pp. 61-76, March, 2008.

[5] R. Wiseman, C. Watt, Conjuring cognition: a review of educational magic-based interventions, *PeerJ*, Vol. 8, No. 4, Article No. e8747, March, 2020.

[6] D. F. Reed, R. I. Greenberg, Using magic to teach computer Programming, *14th International Conference on Education and New Learning Technologies*, Palma, Spain, 2022, pp. 3240-3248.

[7] M. Beck, M. Cohen, J. Cuomo, P. Gribelyuk, The number of "magic" squares, cubes, and hypercubes, *American Mathematical Monthly*, Vol. 110, No. 8, pp. 707-717, October, 2003.

[8] H. P. Koirala, The effect of mathmagic on the algebraic knowledge and skills of low-performing high school students, *Proceedings of the 29th Conference of the International Group for the Psychology of Mathematics Education*, Melbourne, Australia, 2005, pp. 209-216.

[9] R. Kumari, C. Fancy, Analyzing the PyGameGUI modules available in python, *International Conference on IoT and Application* (*ICIOT*), Nagapattinam, India, 2017, pp. 1-6.

[10] Hardono, I. Surjandari, A. Rachman, Y. A. B. Panjaitan, A. Rosyidah, Development of theses categorization system search engine using PHP and MySQL, *International Conference on Information Technology Systems and Innovation* (*ICITSI*), Bandung, Indonesia, 2017, pp. 194-199.

[11] H. Han, K. Kim, S. Jung, Y. Chang, Performance Evaluation on the Parallel Processing System with the Raspberry Pi 4, *International Conference on Computational Science and Computational Intelligence* (*CSCI*), Las Vegas, NV, USA, 2022, pp. 1372-1374.

[12] P. D. P. Adi, A. Mappadang, N. Armi, A. B. Santiko, T. Adiprabowo, Suprapto, R. Zulkarnain, A. Wirawan, Optimization and Development of Raspberry Pi 4 Model B for the Internet of Things, *IEEE 9th Information Technology International Seminar* (*ITIS*), Batu Malang, Indonesia, 2023, pp. 1-6.

[13] P.-S. Tsai, N.-T. Hu, T.-F. Wu, J.-Y. Chen, T.-H. Chao, Multifunctional Bicycle Helmet Using Internet of Things Technology, *Sensors and Materials*, Vol. 33, No. 5, pp. 1589-1602, May, 2021.

[14] S.-S. Perng1, P.-S. Tsai, N.-T. Hu, T.-F. Wu, J.-Y. Chen, Control of a Six-axis Robotic Arm Using the Touch Panel, *Journal of Internet Technology*, Vol. 21, No. 2, pp. 407-416, March, 2020.

Biographies

Pu-Sheng Tsai was born in Taiwan, R.O.C., in 1962. He received the M.S. degree in automatic control from the Feng Chia University, Taichung, Taiwan, R.O.C., in 1985 and the Ph.D. degree in electrical engineering from the National Taiwan University, in 1998. He is currently an Associate Professor in the Department of Electronic Engineering at Ming Chuan University, Taoyuan, Taiwan. His main research interests are artificial intelligence, virtual reality somatosensory interaction, internet of things, Embedded Micro-controller application and design.

Ter-Feng Wu was born in Taiwan in 1962. He received the B.S. degree in Department of Industrial Education from National Taiwan Normal University, Taipei, Taiwan, in 1986. He received the M.S. degree in Department of Control Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1993. He received the Ph.D. degree in Department Electrical Engineering from National Taiwan University, Taipei, Taiwan, in 2006. He is currently a professor and also serves as the Dean of College of Electrical Engineering & Computer Science, National Ilan University, Yilan, Taiwan. His research interests include intelligent control, neural network, fuzzy CMAC, unmanned aerial vehicles (UAVs), green energy and mobile robot, etc.

En-Jia Hsu was born in Taiwan, R.O.C., in 2003. She is currently a student in the Department of Electronic Engineering at Ming Chuan University, Taoyuan, Taiwan. Her main research interests are Embedded Microcontroller applications, Internet of Things and wireless communication technologies.