Multiple Tactile Feedbacks System for Platform Jumping Game

1st Leyi Yu
Department of Electrical, Computer, and Systems Engineering
Case Western Reserve University
Cleveland, USA
lxy443@case.edu

2nd Jiayi Chen
Department of Electrical, Computer, and Systems Engineering
Case Western Reserve University
Cleveland, USA
jxc2077@case.edu

***Abstract*—Platform jumping is a very classic type of game, but due to the game interaction method, the player's feeling of jumping only stays at the visual level. We design a hapkit-based control system and a corresponding Unity-based platform jumping game. The hardware part of the system simulates clearly distinguishable different damping, friction and elasticity, and the haptic feedback is able to interact with the game in both directions. The visual interaction within the game and the haptic feedback of the system can be corresponded in real time.**

I. Introduction

Platform jumping games have long been a classic in the gaming industry, celebrated for their simplicity and engaging mechanics. However, despite advancements in graphics and gameplay complexity, the sensory interaction in such games predominantly remains visual and auditory, often neglecting the tactile dimension of user experience. This sensory gap can diminish the immersion and realism that modern gamers seek.

To address this shortfall, our study introduces a novel haptic feedback system designed to enhance player interaction through tactile responses.The core innovation lies in the dual hapkit setup which simulates various physical properties such as damping, friction, and elasticity through tactile cues. These cues correspond to in-game actions and scenarios, providing real-time feedback that is contextually relevant to the player's actions.

In the following sections, we detail the device setup, the software and control mechanisms employed, and the algorithmic strategies developed to synchronize the tactile feedback with the game dynamics. We further evaluate the effectiveness of this integrated system through a user study, analyzing the impact of enhanced tactile feedback on the overall gaming experience.

II. DEVICE AND METHODS

A. Device

Our device contains two hapkits. The equipment is shown in Fig. 1. One of the hapkits is placed with its narrow side facing down. Users can interact with the hapkit handle in the vertical direction. This haptic controls the strength of the jump and gives the user force feedback. Another hapkit is placed with its back facing down, and the user interacts with the hapkit handle in a horizontal direction. hapkit controls the direction of jumping and simulates the texture of the block surface. At the same time, a vibrator is installed on the hapkit handle for direction control. This vibrator provides users with prompts for the start and end of the game. In order to ensure the stability of the device during interaction, the two hapkits are supported by 3D printed brackets. Two hapkit brackets are screwed to a wooden board. Suction cups are installed under the wooden board to facilitate fixing equipment

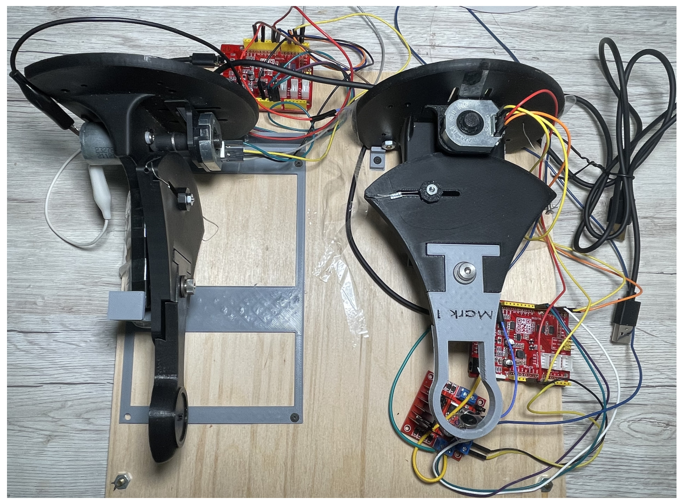

Fig. 1. An overhead shot of the device

B. Software and Control

We use the Unity engine to make games and Arduino IDE to write hapkit control programs. The game part is shown in Fig. 2, and the yellow object is the game character. The game contains blocks of three materials: rough and hard blocks (brown), ordinary blocks (blue), and soft and smooth blocks (pink). Three types of blocks are randomly generated, and for the fun of the game, the blocks may slowly move back and forth along the x-axis or z-axis of the world coordinate system. The user's current results are recorded in the upper left corner of the game. The red arrow indicates the direction of the user's jump, and the end position of the red arrow shows the landing point of the game character according to the current user's operation. The progress bar on the right side of the screen

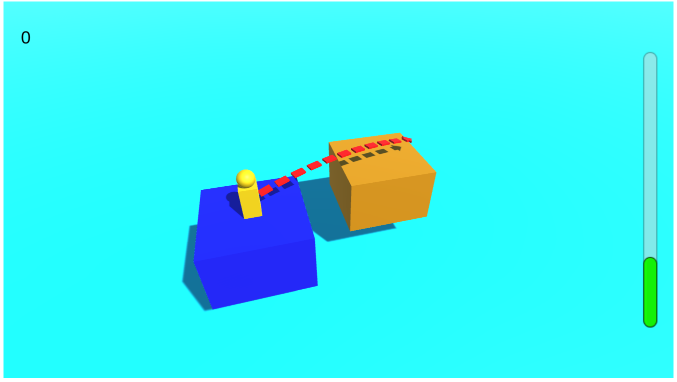

Fig. 2. In-game interface

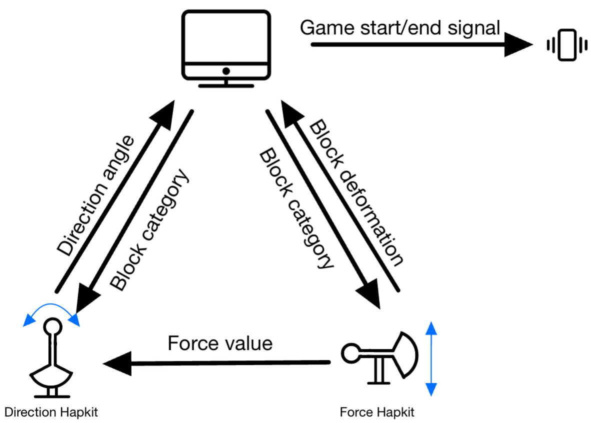

Fig. 3. Schematic diagram of communication between various devices

gives the user a reference to the relative magnitude of the applied force.

Hapkits communicates serially with the game through ports "COM4" and "COM5". The meaning of data flow is shown in Fig.3. When the game starts, the vibrator vibrates. The game program randomly generates a block and passes the type data to hapkits. Hapkits selects a corresponding set of parameters. When the player presses the force hapkit downward, the difficulty of pressing the handle is different and the block in the game will deform to varying degrees according to the material. At the same time, due to the increased pressure, the sliding friction will also change when the direction hapkit is manipulated left and right. When the acceleration direction of the hapkit handle is upward and greater than the threshold, the program determines that the user lets go and the game character takes off. If the character fails to jump to the next block, the game fails and the vibrator vibrates. The control delay in the entire process is very low, and the human eye cannot detect the delay.

C. Algorithm and Parameters

Whether it's the direction controller or the force controller, it's extremely important to the game.

- Direction Controller

For direction controller, which control direction, finding the current direction is crucial to the game as a whole. For the controller that controls the direction, the number of turns the motor turns can be a good reflection of this. Its formula is as follows:

$$\theta_{Dsector} = 0.75 \cdot \frac{r_{\mathrm{p}}}{r_{\mathrm{s}}} \cdot updatedPos \tag{1}$$

Where $\theta_{\mathrm{Dsector}}$ represents the angle of the current direction controller, rp represents the radius of the pulley, and rs represents the radius of the sector. `UpdatedPos` keeps track of the encoder position.

At the same time, the player also has different friction in different materials of the cube. The formula to calculate the friction force is as follows:

$$F_{\mathrm{Friction}} = -0.1 \cdot b_{\mathrm{non}} \cdot v_{\mathrm{h\_threshold}} + \left(-\frac{b_{\mathrm{non}}}{b_{\mathrm{scaling}}}\right) \cdot (v_{\mathrm{h\_filt}} - v_{\mathrm{h\_threshold}}) \tag{2}$$

Where b_{non} is the linear damping, $v_{\mathrm{h\_filt}}$ is the filtered velocity of the handle using an infinite impulse response filter. And $v_{\mathrm{h\_threshold}}$ is the velocity threshold for nonlinear damping.

If vh_f is less than the vh_THR value, and at the same time greater than the -vh_THR value. The friction force is calculated as follows:

$$F_{\mathrm{Friction}} = -b_{\mathrm{nonlinear}} \cdot v_{\mathrm{h\_filt}} \tag{3}$$

- Force Controller

For the force controller, the maximum distance a player can press down on the handle corresponds to the maximum distance a player can jump in the game. The formula for calculating how far the player presses down on the handle is as follow:

$$x_h = r_h \cdot \theta_{Fsector} \cdot \frac{\pi}{180} \tag{4}$$

$\theta_{\mathrm{Fsector}}$ indicates the Angle at which the handle of the current force controller is rotated, and r_h indicates the radius of the handle.

After we get the distance x_h, we scale this distance to the game in equal proportions. The motor output a certain amount of force against the player's force can make the game more realistic. Essentially, this force acts in a very similar way to a virtual wall.The force application formula of the Virtual wall is:

$$F = -k_{\mathrm{wall}} \cdot (x_h - x_{\mathrm{wall}}) \tag{5}$$

Where stiffness in the wall is k_{wall} and stiffness in the wall is x_{wall}. For our experiment, the location of the wall is the starting point. So the value of x_{wall} is 0. Therefore, the formula for calculating force of force controller is:

$$F_{fcon} = -k_{\mathrm{cube}} \cdot x_h \tag{6}$$

- Entire System

In order to simulate a mechanical model in the real world, we must simultaneously gather motor information from both handles and exchange messages between them.

Usually, it is relatively difficult for us to change direction while we are crouching. This is due to the extra force in the squat. The same is true for the direction controller. The process of pressing the force controller is equivalent to the process of squatting, so the direction controller needs to provide extra torque to simulate this friction.The calculation formula of sliding friction force is:

$$F_{\text{Friction\_s}} = \mu \cdot (M_{\text{player}} \cdot g + F_{\text{fcon}}) \tag{7}$$

Where M is the character controlled by the player in the game, and μ is the friction coefficient of different blocks of different materials.

Therefore, for the whole system, the force that the direction controller motor needs to provide is:

$$F_{\text{Friction\_dcon}} = F_{\text{Friction\_s}} + F_{\text{Friction}} \tag{8}$$

- Parameters Setting In order to achieve better discrimination and physical effects, we designed the values of three material parameters multiple times in experiments, as shown in Table 1.

TABLE I
PARAMETERS OF DIFFERENT MATERIALS

Parameter	Pink	Blue	Brown
μ	0.1	1	10
k	200	5	2
b	0.1	4	12

[a] μ: Friction coefficient. k: Spring constant. b: Damping factor.

III. USER STUDY

A. Background

The study gathered user feedback on various haptic feedback devices to evaluate their performance across several dimensions, including ease of use, production values, noticeability, stability, and realism of the haptic feedback.

B. Methodology

Participants were provided with a Qualtrics survey where they could rate the devices on the aforementioned dimensions using a 1-5 scale, where 1 represented 'Non-existent' and 5 represented 'Excellent!'. Additionally, they were asked to list two things that were done well and to make suggestions for improvements.

C. Feedback Summary

TABLE II
USER FEEDBACK ON HAPTIC DEVICE CHARACTERISTICS

User	Ease of Use	Prod. Val.	Notic. of Hap.	Stab. of Hap.	Real. of Hap.
user_0	4	3	4	4	3
user_1	5	3	4	4	2
user_2	4	5	5	3	3
user_3	4	3	4	4	3
user_4	4	3	4	4	4
user_5	4	4	4	4	4

The users' feedback is showed in Table 2.

- **Ease of Use:** The average score is about 4.17, and the variance is low at 0.17, indicating that users generally find the product easy to use and there is consistency in this perception.
- **Production Values (Aesthetics and Build Quality):** The average score is about 3.67, with a median deviation of 0.67, suggesting that while users generally rate the production values positively, there is more variation in opinion compared to ease of use.
- **Noticeability of Haptics:** Users gave an average score of 3.83 with a variance of 0.57. This suggests that the haptic functions are quite noticeable, though there is some variability in user responses.
- **Stability of Haptics:** This category also received a high mean score of 4.17 with a low variance of 0.17, indicating that users consistently perceive the haptics as stable.
- **Realism of Haptics:** The lowest average score at 3.17 with a variance of 0.57 indicates a slightly lower satisfaction with the realism and some differences in perception.

IV. CONCLUSION

Overall, the user feedback was positive, with users appreciating the functionality and performance of the haptic devices. In summary, our work demonstrated a simulation of the physical effects of three materials, a complete system with low latency and multiple devices interacting with each other.

Based on the feedback from the demo site and the analysis of the user study results, Some areas for future work include improving the brackets that support hapkits. The upper structure of the current system is not strong enough. Although the system can survive many experiments, the presence of tape may affect the user's confidence in the firmness of the device, thereby affecting the gaming experience. Secondly, remaking the hapkit's handle to reduce its weight. We tested the effect of simulating uneven surfaces during the design process. The effect can be felt when the hapkit is placed normally, but when the hapkit is placed with its back facing down, the effect is not obvious. Our analysis suggests that the handle's own weight has an impact on the axis of rotation, so we will re-design the hapkit handle in the future.

ACKNOWLEDGMENT

We are very grateful to the professor for his support and guidance throughout the course. His depth of knowledge and commitment to academic excellence are critical to enhancing our learning experience.

We would also like to express our heartfelt thanks to all the students in the class. The collaborative efforts, insightful contributions and shared learning environments we create together are both inspiring and enriching. The unity and encouragement found in this team is invaluable and we are truly grateful for the collective growth we have experienced.

We would like to thank everyone who has played a role in this educational journey for their contribution. Your support not only promotes our academic efforts, but also makes a significant contribution to our personal development.

REFERENCES

[1] G. Chursin and M. Semenov, "Learning game development with Unity3D engine and Arduino microcontroller," *Journal of Physics: Conference Series*, vol. 1488, no. 1, Art. no. 012023, 2020.

[2] J.-L. Tseng, "A Study on 3D Interaction System using Arduino Glove Design and Glasses-Free 3D Display Technology," *International Journal of Advanced Studies in Computers, Science and Engineering*, vol. 6, no. 8, p. 1, 2017.

[3] N. Kusumoto, T. Sohmura, S. Yamada, K. Wakabayashi, T. Nakamura, and H. Yatani, "Application of virtual reality force feedback haptic device for oral implant surgery," *Clinical Oral Implants Research*, vol. 17, no. 6, pp. 708–713, 2006.

[4] D. Tsetserukou, S. Hosokawa, and K. Terashima, "LinkTouch: A wearable haptic device with five-bar linkage mechanism for presentation of two-DOF force feedback at the fingerpad," in *Proc. 2014 IEEE Haptics Symposium (HAPTICS)*, 2014, pp. 307–312.

[5] A. Rodriguez, I. Boada, and M. Sbert, "An Arduino-based device for visually impaired people to play videogames," *Multimedia Tools and Applications*, vol. 77, pp. 19591–19613, 2018.