

Application of Virtual Simulation in Traditional Chinese Medicine Processing Education

Qi Zhang, Zixi Song, and Huaying Zhou*

Abstract

Traditional Chinese medicine (TCM) processing is a highly practical discipline, and experimental teaching is an important component of the curriculum. However, the experimental teaching methods often face the limitations of seasonal time, experimental materials, and safety factors, resulting in fewer opportunities for students to practice, insufficient learning interest, and poor teaching effectiveness. Therefore, integrating virtual simulation into experimental teaching is an important means of solving the experimental teaching problem. In this paper, a virtual laboratory platform for TCM processing is built based on the principles of TCM processing, providing learners with a real experience in a TCM processing environment. Aiming at the problem of realism and immersion in virtual scenes, we present a series of key technologies, including using Maya for 3D model construction, physically based rendering technology for texture mapping, and Unity3D for complete interactive functions. The result shows that our developed platform has good performance, high immersion, and accurate interaction. Learners can roam through the scene, visualize the herb model, and complete experiments according to the prompted steps. Moreover, the virtual laboratory can deepen students' understanding of the basic theories and enhance their learning outcomes, including knowledge, skills and enjoyment.

Keywords

Realism and Immersion, Traditional Chinese Medicine Processing Experiment, Virtual Simulation

1. Introduction

Traditional Chinese medicine (TCM) processing is a fundamental course in TCM, which mainly studies the theory, technology, methods, historical evolution, and development direction, containing rich historical and cultural connotations. Since the record in Huangdi Neijing (Inner Canon of the Yellow Emperor), till now, the processing of TCM has experienced more than 2000 years of inheritance, innovation, and development, which is a combination of TCM theory and clinical practice, and plays an extremely important position in the field of TCM [1]. The processing technology has accumulated many valuable ancient methods and experienced inheritance after thousands of years of inheritance and development. Therefore, experimental teaching is very important in learning the TCM processing curriculum, it is a critical way to connect theory with practice. However, due to the limitations of specific production areas and harvesting seasons, especially the high prices and scarce resources of rare medicinal

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript received June 11, 2024; first revision July 17, 2025; accepted August 2, 2025.

*Corresponding Author: Huaying Zhou (huayingzhou@gdpu.edu.cn)

School of Medical Information and Engineering, Guangdong Pharmaceutical University, Guangzhou, China (jackie.zhang@gdpu.edu.cn, 2252860899@qq.com, huayingzhou@gdpu.edu.cn)

Qi Zhang and Zixi Song contributed equally to this work.

materials, it is difficult to conduct experimental teaching in this curriculum. This makes it difficult for learners to comprehensively understand of the all-experimental steps and outcomes, and restrict the cultivation of practical and innovative abilities.

Virtual simulation is one of the most popular techniques used in simulation-based education to structure and guide learning with experiences that approximate reality [2]. The trend in the use of simulation in education continues to expand rapidly [3]. It plays an important role in TCM education. Especially for the experimental teaching of TCM processing, there are potential safety hazards in the teaching process, large consumption of experimental materials, and waste of resources [4]. With the application of virtual simulation technology into TCM experimental teaching, it not only provides a secure and low-cost teaching platform, but also recreates the scene of the Chinese herbals processing environment.

2. Related Work

In recent years, virtual simulation experiments have been widely applied in the field of medical education, such as surgical [5] and anatomy [6] trainings, healthcare [7], nursing [8], radiology [9], and clinical rehabilitation [10]. It allows learners to apply textbook knowledge prior to treating patients, and provides a safe environment to make mistakes [11]. The application of virtual simulation technology in the education of TCM is still in its beginning period. Zhu et al. [12] developed an augmented reality (AR) Chinese herbal medicine teaching system, which helps the user to enhance their vision and interest in the herbs. These 3D models can vividly display the characteristics of TCM to learners, replacing text with images, helping students efficiently distinguish the surface details of TCM. On a deeper level of consideration, we can use virtual simulation technology to construct a virtual experimental environment, allowing learners to become familiar with the experimental operation process and deepen their understanding of its processing principles and methods through real-time interaction. This article discusses how to create and develop an immersive virtual experimental platform for TCM processing.

3. Key Technologies and Methods

This paper takes Maya as the modeling tool, uses Substance Painter as the mapping tool, and utilizes Unity3D as the development software. We built a virtual 3D TCM processing laboratory, where learners can interact with the 3D herb model through rotation, scaling, and transforming of their views. The core functionality of the platform lies in the virtual simulation of four classic TCM processing experiments, including cutting *Polygonum multiflorum*, stir-frying gardenia, steaming red ginseng, and salt-making *Morinda officinalis*.

3.1 Platform Overall Design

According to the functional requirements, the platform is divided into three modules, including the experimental module, the three-dimensional roaming module, and the user management module. The functional structure of the platform is illustrated in Fig. 1.

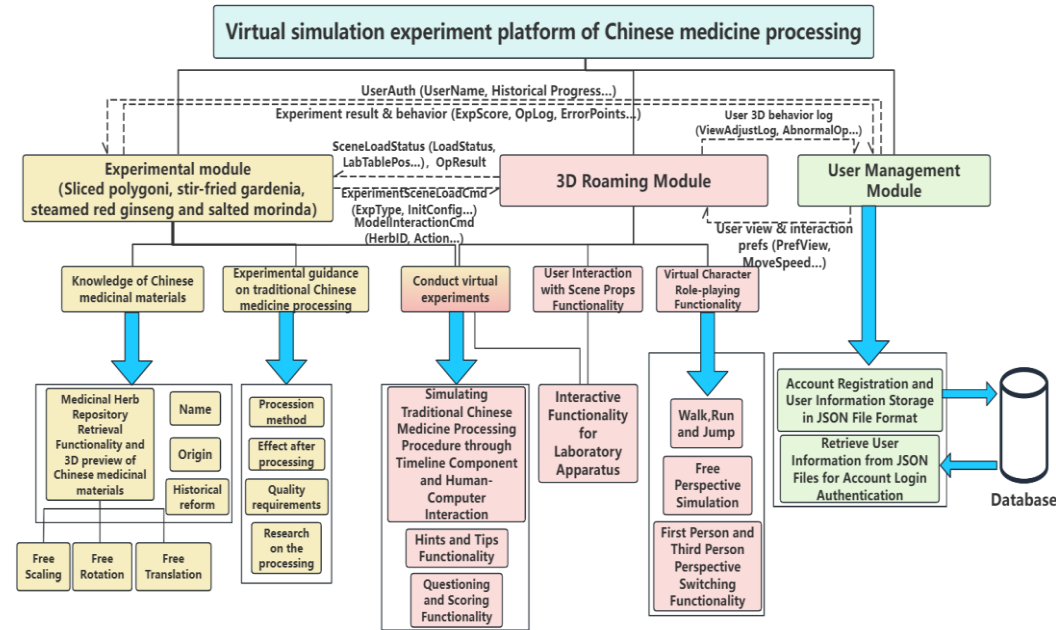


Fig. 1. Functional structure of the platform.

3.1.1 Experimental module

This module comprehensively covers four experimental procedures, namely cutting *Polygonum multiflorum*, stir-frying gardenia, steaming red ginseng, and salt-making *Morinda officinalis*. Moreover, each of these experiments is subdivided into three distinct sub-modules. The first sub-module, the TCM information retrieval module, offers functionalities such as searching for prescription names, sources, historical evolution, TCM repository searches, and even a 3D preview of medicinal herbs. The second sub-module, the TCM processing knowledge learning module, aims to educate users on TCM processing methods, quality requirements, processing purposes, and related research. The third sub-module, the virtual simulation module of TCM processing, enables students to manipulate laboratory equipment under guidance to successfully complete herbal processing experiments. Throughout this process, pertinent questions, valuable procedural tips, and supplementary knowledge points are provided to enhance their learning experience.

3.1.2 Three-dimensional roaming module

In this module, users can immerse themselves in a virtual laboratory environment, assuming the role of students and performing various actions like walking, running, and jumping. They can seamlessly switch between first-person and third-person perspectives. This module incorporates timeline components and human-computer interaction to simulate the herbal processing experiments chronologically. Step-by-step animations and sound effects guide users through the process, ensuring correct participation. It also features a question-and-answer system with a scoring mechanism based on users' progress and correctness, including timekeeping functionality.

3.1.3 User management module

This module provides login and registration functionalities, storing user information efficiently. JSON, known for its lightweight nature, serves as an ideal format for representing structured data concisely [13]. Leveraging the LitJSON component, registered user data is serialized into a database, facilitating efficient account management and storage.

3.2 3D Modeling

The Autodesk Maya software is the influential 3D animation software, which is more perfect than ordinary 3D visual effect [14]. In this work, all modeling of objects, including virtual TCM, experimental supplies and equipment have been achieved using Maya 2020 on the Windows 11 platform. The traditional method of building 3D models is primarily employed to add lines to the surface of the model, shaping the surface characteristics of virtual TCM. However, this method results in an increase in the total number of faces of the model, thereby leading to excessive consumption of resources during computer operations [15]. Therefore, we have used the polygon modeling method to establish low poly models as shown in Fig. 2. Besides, we also completed experimental display animations, such as cleaning, cutting, drying, stir-frying, steaming, and other animation effects.

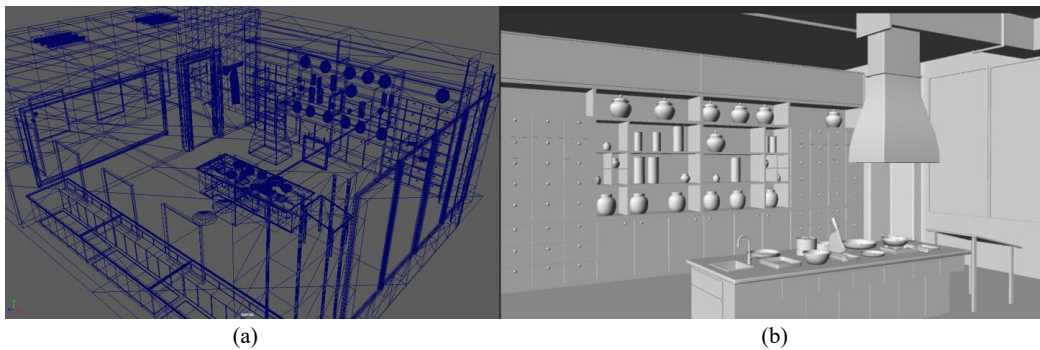


Fig. 2. The 3D scene of TCM processing laboratory: (a) wireframe model and (b) solid model.

3.3 PBR Technology

With the development of information technology, there still exist some differences between the real objects and the constructed ones. Physically based rendering (PBR) is an approach in computer graphics that seeks to render graphics in a way that more accurately models the flow of light in or over the real object [16]. In order to make the shape and surface texture of the constructed virtual TCM very similar to the real ones, we use a low-poly model with a high level of detail by PBR texturing to get better performance results. We take the precious Chinese medicinal herb *Morinda officinalis* as an example to illustrate the production process. Firstly, its 3D model is constructed and UV unwrapped in Maya, and then exported in an FBX file format. Secondly, PBR texture maps, including base color, height, and normal, are created in Substance Painter 2021. Finally, import them into Unity3D to view the rendering effect, as shown in Fig. 3. It can be seen that the 3D rendered image of the virtual *Morinda officinalis* is very close to the real image in terms of shape, color, and texture when compared to the real photograph.

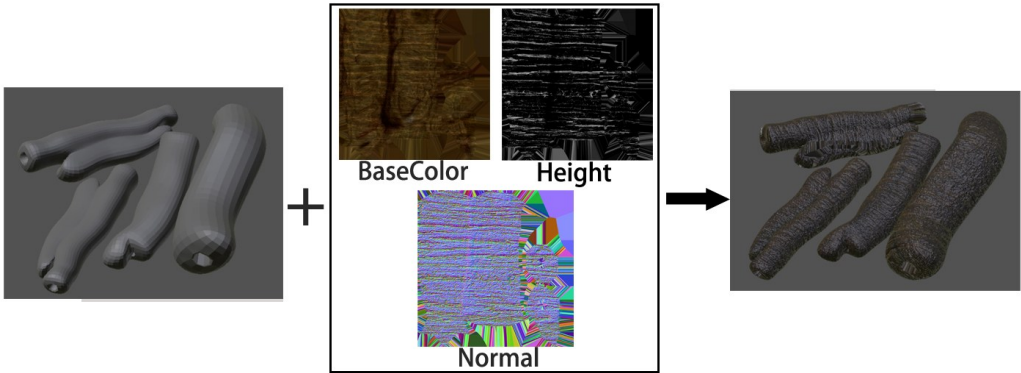


Fig. 3. Virtual *Morinda officinalis* production process.

3.4 Interaction Functions

3.4.1 Accurate character movement

Character movement is controlled by input from devices such as keyboards or controllers. The two-dimensional vector information from these peripheral inputs is applied as a force to the character's rigid body component, thereby achieving movement or jumping effects. In our platform, the control function for character movement is carefully designed and implemented to ensure that the two-dimensional vector information inputted through the keyboard can accurately map to the physical laws that control character movement in the real world. In Unity3D, adding a rigid body component to the character game object is the primary step to enable movement. Rigid body simulates object movement in the physics engine, considering factors like mass, gravity, and applied forces.

When the user holds down the "W" or "S" keys, the character moves along the Y-axis of the two-dimensional plane, with speeds ranging from -1 to 1. Similarly, holding down the "A" or "D" keys moves the character along the X-axis of the plane, with speeds also ranging from -1 to 1. These inputs are continuously updated, and through vector addition, the direction and magnitude of the force controlling the character's movement are determined. However, when the user simultaneously holds down keys from different axes, such as "W" and "D", the resulting vector forms a 45° angle with the positive X-axis, but its length is $\sqrt{2}$ instead of 1. This leads to a faster movement speed along the diagonal in the first quadrant than along the X or Y axes, which is not the expected outcome. To more realistically simulate character movement in the real world, it's necessary to dynamically transform the square region formed by all two-dimensional vectors input by the user into a circular region. To achieve this, each point (x, y) within the square region needs to be projected onto the corresponding point (u, v) within the circular disk [17]. The transformation formula is as follows:

$$u = x \sqrt{1 - \frac{y^2}{2}} \tag{1}$$

$$v = y \sqrt{1 - \frac{x^2}{2}} \tag{2}$$

where u represents the calculated length of the vector on the new X-axis, and v represents the calculated length of the vector on the new Y-axis.

It's worth noting that this mapping is smooth and reversible. Fig. 4 illustrates the process of converting vectors read from the keyboard into the final vectors assigned to the virtual character actions. By preprocessing the data received from peripherals, the absolute value of the magnitude of the new vector always remains between 0 and 1. This prevents inconsistencies between the expected and final outcomes caused by different input values. It makes character actions more easily controlled by users, facilitating the subsequent development of experiments.

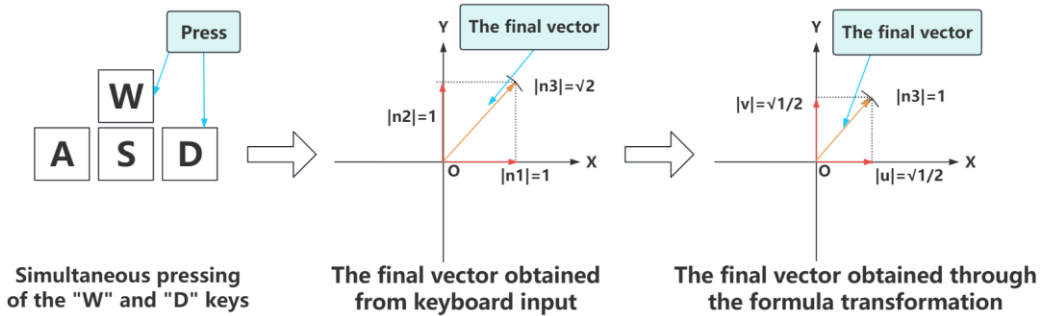


Fig. 4. Vector value correction process.

3.4.2 Dynamic adjustment of virtual camera

In case of the existing camera input approach, such as Unity3D, every value of X, Y, and Z has to be input as a figure by each coordinate [18]. In contrast, our platform allows the virtual camera to automatically adjust its angle and position relative to characters based on collisions with objects in the scene. The execution steps are as follows: firstly, a camera object positioned behind the character model serves as the third-person perspective camera, tracking the character's movements. Secondly, to achieve smooth camera tracking, we utilize the interpolation functions (Lerp and Slerp functions) provided by the Unity API. These functions can uniformly control the changes in position and orientation of objects, such as the rotation of a robotic arm [19]. Finally, by acquiring the character model's position and rotation information in the script and modifying the camera's position using interpolation techniques and adjusting rotation values by modifying quaternions. Consequently, regardless of the character's actions, the camera adjusts its position and orientation accordingly, providing a natural viewing experience.

However, when the character moves into narrow areas such as corners or tight spaces, the camera, due to the need to maintain a certain relative distance from the character, may become obstructed by the model itself or blocked by obstacles in front, affecting the field of view. To address this issue, a line casting method is employed to continuously assess whether there are any obstacles between the camera and the character, thereby dynamically adjusting the actual position of the camera. In Fig. 5, the "People" represent virtual characters within the platform. To prevent the camera from penetrating models, a ray needs to be continuously emitted from the character (depicted in red in the diagram) towards the camera. If the ray collides with actual objects within the scene, such as walls, the camera's position will be updated to the collision point. It is important to note that the direction of the ray should not point from the camera

towards the character. Otherwise, the camera's position will update to outside the wall, resulting in the user's view being obstructed by scene objects.

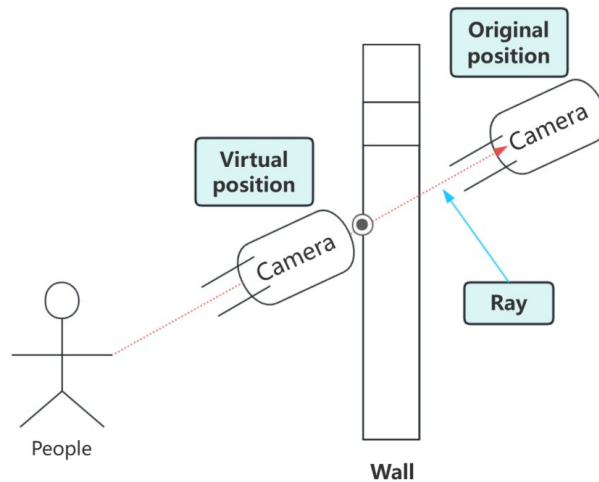


Fig. 5. Display of camera position changes.

```
void UpdateCameraPosition()
{
    // Define ray start (character position) and end (camera target position)
    Vector3 rayStart = character.transform.position;
    Vector3 rayEnd = camera.transform.position;

    RaycastHit hit;

    // Check for collision between ray and "Obstacle" layer
    if (Physics.Linecast(rayStart, rayEnd, out hit, LayerMask.GetMask("Obstacle")))
    {
        // If obstacle is hit, position camera slightly behind the hit point
        camera.transform.position = hit.point - camera.transform.forward * 0.1f;
    }
    else
    {
        // If no collision, smoothly move camera to target position
        camera.transform.position = Vector3.Lerp(
            camera.transform.position,
            targetPosition,
            smoothSpeed
        );
    }
}
```

Fig. 6. The function code of the line casting method.

The line cast method has more precise directionality compared to capsule cast. It effectively solves the occlusion problem of traditional fixed viewing angles in narrow spaces through precise collision detection and position correction mechanisms, providing users with a more natural virtual experimental observation experience. The algorithm is described as follows: first, add Box Collider components to the camera and character objects respectively to build collision detection foundation, then define the direction of the ray from the character's position to the camera's position by obtaining the world coordinates of the character and camera in the script, and then call the Physics.Linecast() function to perform ray casting to detect the collision between the ray and objects in the scene (such as walls, experimental appliances, etc.) in real time. If a collision is detected, immediately adjust the camera position to a point where the collision point is pushed forward 10 cm in the opposite direction of the ray, in order to avoid the camera model penetrating the scene objects, ensure that the user's perspective is always within a reasonable visual range, and improve the realism and smoothness of the interaction process. The function code is shown in Fig. 6.

3.4.3 Interactive ray casting

Object interaction primarily relies on casting a ray from the camera, passing through the mouse cursor on the screen, to detect collisions with target objects within the scene [20]. Using the ray tracing capabilities provided by the Unity engine, simulate the speed and accuracy of character interaction with objects [21]. In our platform, all objects within the scene are equipped with colliders. Using a virtual camera as the source point and the mouse cursor as the pathway, a ray is emitted to perform collision detection with experimental apparatuses within the scene, serving as the primary mode of interaction between users and objects. This ray enables three-dimensional objects in the scene to detect whether the user's mouse cursor enters, exits, or clicks on them. Subsequently, different functions are executed based on these three conditions, enabling features such as highlighting objects when the user moves the cursor over them and triggering specific actions when the user clicks on objects to use them for experiments.

3.5 Applying the Timeline Component

The Unity engine offers a robust design solution akin to the Timeline component, enabling the creation of interactive storylines [22] encompassing animations, visual effects, and more. In simple terms, the Timeline component consists of a "script" and an "actor." The "script" includes multiple tracks, each corresponding to a specific element. For example, the "question track" controls the appearance and disappearance of all questions in the preparation experiment. Each track is divided into transaction blocks, each containing a code file. The code is divided into three main sections: "before the transaction occurs," "during the transaction," and "after the transaction occurs."

The "actor" corresponds to the scanner in the Timeline component, starting from zero and executing each transaction block's code content in three sections until all transaction blocks are completed. This signals the end of the script and the completion of the preparation experiment. Scripts are pre-designed by the software creator for various traditional Chinese medicine preparation experiments. Because different transaction blocks on different tracks can execute different code functions within the same clock cycle. This diverse combination of content enables designers to faithfully replicate the process of traditional Chinese medicine preparation and enhance user interactivity. Below is a schematic representation of the "script" and "actor" architecture within the Timeline component, as depicted in Fig. 7.

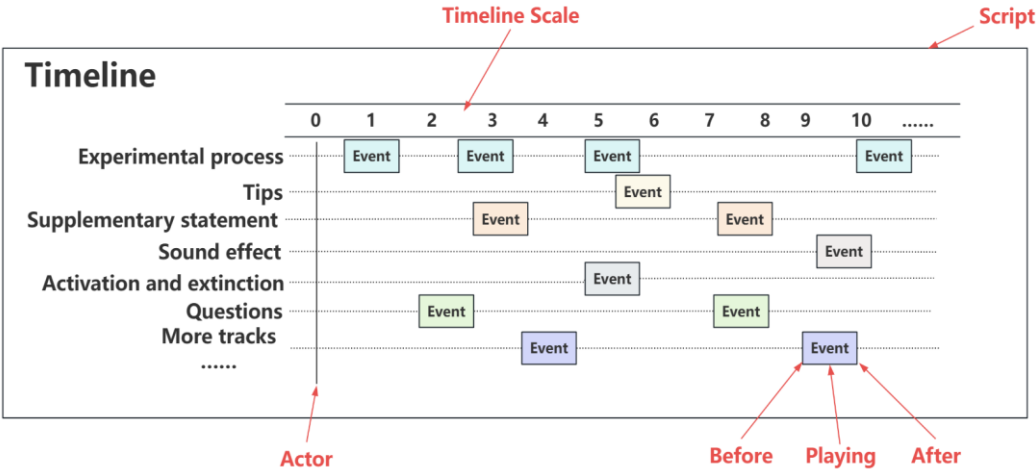


Fig. 7. The overall design of the timeline script.

When the scanner (actor) scans through each transaction block, it executes the code contained within, which is divided into three sections: "pre-transaction," "during transaction," and "post-transaction." For interactive transactions requiring player input, when the timeline moves to the "pre-transaction" state, the timeline's speed is reduced to zero, and it transitions to the "during transaction" state to continuously monitor user interaction inputs. During the execution of experimental steps, the system restricts character movement and viewpoint based on user input. For example, when the timeline moves to a transaction block on the experiment procedure track, it checks whether the correct experimental equipment is used. If correct, the timeline's speed returns to normal, unlocking user movement and viewpoint restrictions to continue the preparation experiment. Otherwise, the timeline speed remains at zero. The partial timeline script diagram of the salt-making *Morinda officinalis* is shown in Fig. 8. Selecting the experimental process track as an example, the corresponding simulation effect triggered by the change of timeline in our platform is shown in Fig. 9.

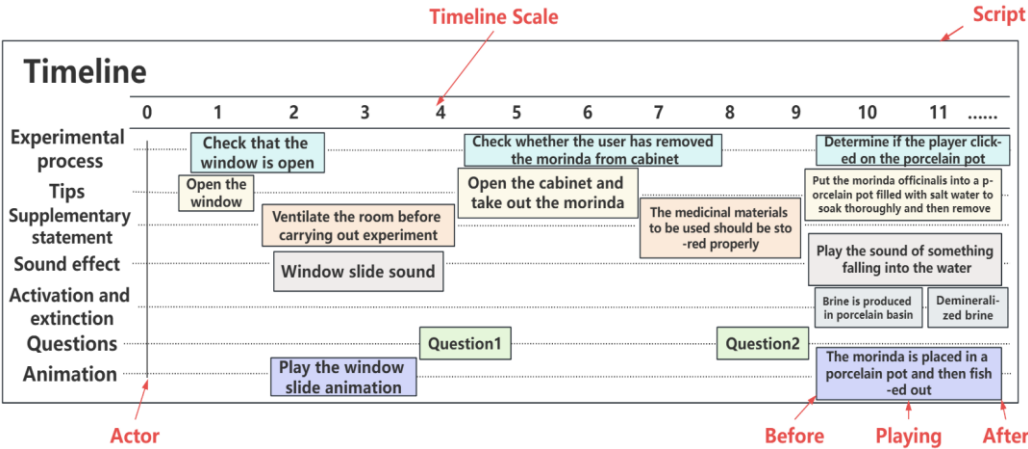


Fig. 8. The partial timeline script diagram of the salt-making *Morinda officinalis*.

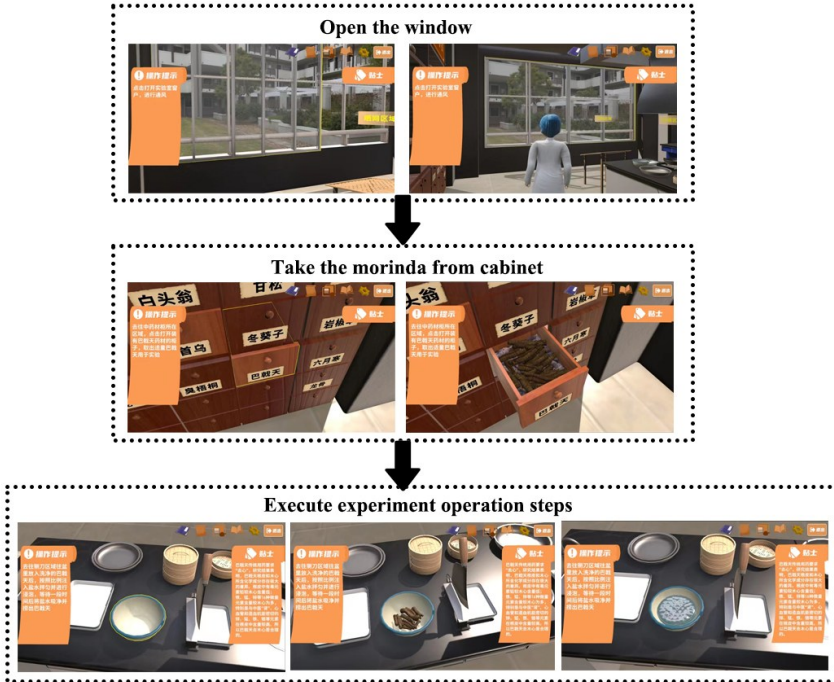


Fig. 9. The platform simulation effect.

4. Result Analysis and Validation

4.1 Performance Quantitative Analysis

Frames per second (FPS) is a key indicator for measuring the real-time performance and interactive fluency of virtual simulation systems, and its numerical stability directly affects learners' cognitive experience and operational feedback quality. In this paper, we quantitatively evaluate the platform performance in typical experimental scenarios by writing an FPS statistical module with high-precision time series sampling capability. This module is developed based on the Unity3D engine and uses a singleton pattern to achieve FPS dynamic monitoring. It accurately records the complete process from the user clicking the "Start" button to clicking the "End" button after completing all operations through the variable (Time.realtimeSinceStartup). It collects rendering time data on a frame-by-frame basis and calculates real-time frame rates.

The following selects the operation process of "salt-making *Morinda officinalis*" as a test case, which involves multiple complex steps such as medicinal herb cleaning, saltwater soaking, and frying, and can comprehensively reflect the performance of the platform. By continuously monitoring the frame rate changes throughout the experimental operation, we obtained the following data results: the average frame rate reached 137 FPS, the standard deviation was $\sigma = 2.3$, and the single frame rendering time was about 0.00730 seconds. The mouse click event processing time ratio measured by the Profiler tool is 2.8%, and the system interaction response delay is 0.2044 ms, as shown in Fig. 10. The data shows that the platform can maintain excellent real-time interaction even at high frame rates.

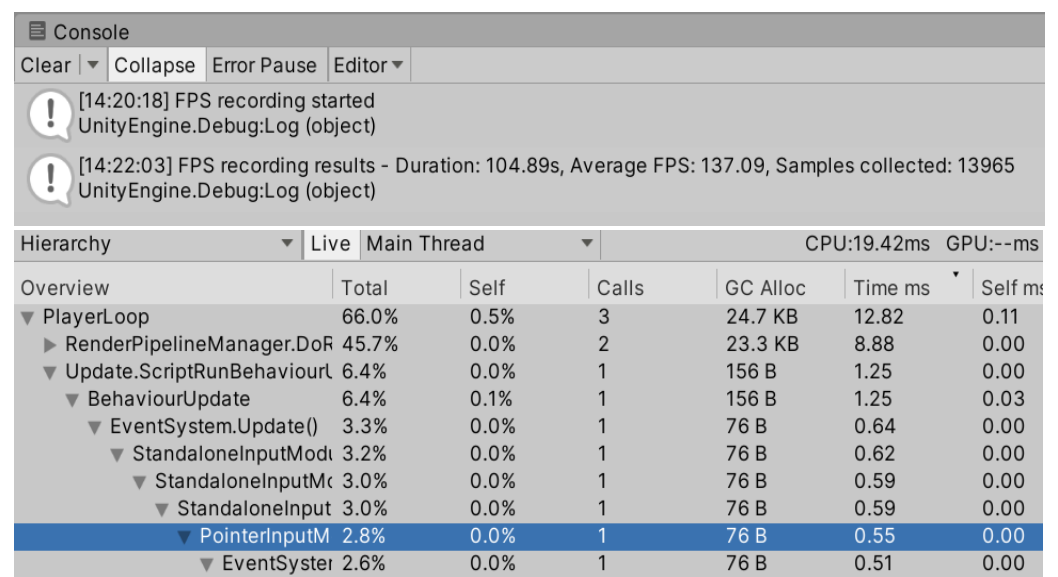


Fig. 10. Average frame rate and percentage of mouse click event processing time.

4.2 Student Learning Effect

To test the teaching effectiveness of the simulation experiment platform, a questionnaire survey was conducted to analyze and verify students' satisfaction with their experimental experience. The results are shown in Table 1. Twenty-eight male and 24 female students were selected from second-year students majoring in TCM and randomly divided into an experimental group and a control group, with 14 male and 12 female students in each group. The experimental group used a simulation platform for teaching, while the control group used traditional teaching methods. According to the questionnaire results, the satisfaction rates of the experimental group students in improving their skill operation ability, stimulating learning interest, and enhancing self-learning ability were 96.15%, 96.15%, and 92.31%, respectively, all significantly higher than the control group. Therefore, the virtual simulation experiment platform has significant advantages over traditional teaching due to its interactivity and visualization advantages.

Table 1. Two groups of student questionnaires survey results

	Experimental group (n=26)				Control group (n=26)			
	Very satisfied	Satisfied	Dissatisfied	Satisfaction (%)	Very satisfied	Satisfied	Dissatisfied	Satisfaction (%)
Improve skills operation ability	18	7	1	96.15	10	9	7	73.08
Stimulate learning interest	17	8	1	96.15	8	10	8	69.23
Enhance self-learning ability	16	8	2	92.31	7	9	10	61.54

5. Conclusion

TCM processing experiments have drawbacks such as limited experimental resources, safety hazards and time-consuming issues, limitations in experimental operation skills, and limitations in experimental content. Virtual simulation experiments can allow students to immerse themselves in a virtual experimental environment, simulate the real process of preparing herbal medicine through their own personal operation, and comprehensively master traditional processing experimental techniques. However, there are also some shortcomings, such as the lack of realism in the implementation of herbs through computer simulation, which affects their experimental experience and learning effectiveness, as well as insufficient interactivity. This paper focuses on the issues of realism and immersion, and develops this platform using 3D modeling technology, PBR technology, and Unity3D technology. The system provides an excellent immersive experience, allowing learners to fully immerse themselves in virtual experimental operations. In the future, virtual simulation technology will be integrated with artificial intelligence and big data technology to make systems more intelligent and adaptive. For example, the system can automatically adjust teaching content, difficulty, and progress based on user behavior and feedback, achieving a personalized learning experience.

Conflict of Interest

The authors declare that they have no competing interests.

Funding

The College Student Innovation and Entrepreneurship Training Program of Guangdong Province, China (No. 202310573014) and the Guangdong Medical Research Fund Project, China (No. B2025449).

References

- [1] Y. Tian, Y. Shi, Y. Zhu, H. Li, J. Shen, X. Gao, B. Cai, W. Li, and K. Qin, "The modern scientific mystery of traditional Chinese medicine processing: take some common traditional Chinese medicine as examples," *Heliyon*, vol. 10, no. 2, article no. e25091, 2024. <https://doi.org/10.1016/j.heliyon.2024.e25091>
- [2] H. Y. Chang and H. L. Chang, "A virtual simulation-based educational application about complementary and alternative medicine: a pilot study of nurses' attitudes and communication competency," *Nurse Education Today*, vol. 97, article no. 104713, 2021. <https://doi.org/10.1016/j.nedt.2020.104713>
- [3] C. Phanudulkitti, S. Puengrung, R. Meepong, K. Vanderboll, K. B. Farris, and S. E. Vordenberg, "A systematic review on the use of virtual patient and computer-based simulation for experiential pharmacy education," *Exploratory Research in Clinical and Social Pharmacy*, vol. 11, article no. 100316, 2023. <https://doi.org/10.1016/j.rcsop.2023.100316>
- [4] G. Cao, H. Du, X. Liu, G. Wang, and Y. Liu, "Discussion on the application of virtual simulation technology in the experimental teaching of Chinese medicine processing, taking the processing of *Atractylodes lancea*

- slices as an example,” *Chinese Medicine Modern Distance Education of China*, vol. 18, no. 18, pp. 21-23, 2020. <https://doi.org/10.3969/j.issn.1672-2779.2020.18.009>
- [5] A. Luca and R. Giorgino, “Augmented and virtual reality in spine surgery,” *Journal of Orthopaedics*, vol. 43, pp. 30-35, 2023. <https://doi.org/10.1016/j.jor.2023.07.018>
 - [6] M. Aebbersold, T. Voepel-Lewis, L. Cherara, M. Weber, C. Khouri, R. Levine, and A. R. Tait, “Interactive anatomy-augmented virtual simulation training,” *Clinical Simulation in Nursing*, vol. 15, pp. 34-41, 2018. <https://doi.org/10.1016/j.ecns.2017.09.008>
 - [7] O. G. Shet, V. K. Ponduri, P. Gupta, H. Rohan, and M. S. Roopa, “Augmented reality in healthcare: a systematic review,” *International Journal for Research in Applied Science & Engineering Technology*, vol. 11, no. 2, pp. 1226-1232, 2023. <https://doi.org/10.22214/ijraset.2023.49212>
 - [8] J. Ma, Y. Wang, S. Joshi, H. Wang, C. Young, A. Pervez, Y. Qu, and S. Washburn, “Using immersive virtual reality technology to enhance nursing education: a comparative pilot study to understand efficacy and effectiveness,” *Applied Ergonomics*, vol. 115, article no. 104159, 2024. <https://doi.org/10.1016/j.apergo.2023.104159>
 - [9] K. Means, K. Kleiman, D. Ogdon, and S. Woodard, “A review of virtual reality in radiology,” *Current Problems in Diagnostic Radiology*, vol. 53, no. 1, pp. 17-21, 2024. <https://doi.org/10.1067/j.cpradiol.2023.10.006>
 - [10] V. Herrera, D. Vallejo, J. J. Castro-Schez, D. N. Monekosso, A. de los Reyes, C. Glez-Morcillo, and J. Albusac, “Rehab-immersive: a framework to support the development of virtual reality applications in upper limb rehabilitation,” *SoftwareX*, vol. 23, article no. 101412, 2023. <https://doi.org/10.1016/j.softx.2023.101412>
 - [11] S. Azher, A. Cervantes, C. Marchionni, K. Grewal, H. Marchand, and J. M. Harley, “Virtual simulation in nursing education: headset virtual reality and screen-based virtual simulation offer a comparable experience,” *Clinical Simulation in Nursing*, vol. 79, pp. 61-74, 2023. <https://doi.org/10.1016/j.ecns.2023.02.009>
 - [12] Q. Zhu, Y. Xie, F. Ye, Z. Gao, B. Che, Z. Chen, and D. Yu, “Chinese herb medicine in augmented reality,” 2023 [Online]. Available: <https://arxiv.org/abs/2309.13909>.
 - [13] T. H. Kim and B. P. Kyung, “Performance comparison of JSON libraries for game development using unity engine,” *Journal of Digital Contents Society*, vol. 25, no. 3, pp. 771-779, 2024. <https://doi.org/10.9728/dcs.2024.25.3.771>
 - [14] H. Chen, “Research on the application of digital media art in animation control based on Maya MEL language,” *Acta Technica*, vol. 62, no. 1B, pp. 499-507, 2017.
 - [15] Q. Zhang, Z. Q. Lin, X. T. Liang, and H. Y. Zhou, “Realistic 3D modeling method of virtual traditional Chinese medicine,” in *Proceedings of the 2023 International Conference on Frontiers of Artificial Intelligence and Machine Learning*, Beijing, China, 2023, pp. 195-198. <https://doi.org/10.1145/3616901.3616944>
 - [16] M. Pharr, W. Jakob, and G. Humphreys, *Physically-based Rendering: From Theory to Implementation*, 3rd ed. San Francisco, CA: Morgan Kaufmann Publishers, 2016.
 - [17] C. Fong, “Analytical methods for squaring the disc,” 2015 [Online]. Available: <https://arxiv.org/abs/1509.06344>
 - [18] H. Jeon, E. Chae, and H. Pak, “Application of camera motion frame editor tool for Unity 3D game engine,” *Advanced Science and Technology Letters*, vol. 87, pp. 147-150, 2015. <http://dx.doi.org/10.14257/astl.2015.87.30>
 - [19] G. Xiang, “Real-time follow-up tracking fast moving object with an active camera,” in *Proceedings of 2009 2nd International Congress on Image and Signal Processing*, Tianjin, China, 2009, pp. 1-4. <https://doi.org/10.1109/CISP.2009.5303457>
 - [20] J. S. Seo and M. J. Kang, “Comparative analysis of projectile collision detection methods in Unity3D,”

Proceedings of the Korean Society of Computer Information Conference, vol. 25, no. 1, pp. 181-182, 2017.

- [21] A. M. Ahmed Baraka, N. Grozmani, J. Eickelmann, D. Wolfschlager, and R. H. Schmitt, "Examination of ray casting in Unity 3D as a fast pose prediction tool for industrial CT scans of multi-material specimens," *e-Journal of Nondestructive Testing*, vol. 29, no. 3, 2024. <https://doi.org/10.58286/29263>
- [22] A. Parab, N. Rathod, T. Patil, K. Deshpande, and N. Deshmukh, "A 3D storyline using unity game engine," in *Proceedings of 2022 2nd International Conference on Intelligent Technologies (CONIT)*, Hubli, India, 2022, pp. 1-5. <https://doi.org/10.1109/CONIT55038.2022.9848260>



Qi Zhang <https://orcid.org/0009-0002-9347-1217>

She received an M.S. degree from Hong Kong Polytechnic University in 2009. She is a lecturer in the Department of Digital Media Technology, Guangdong Pharmaceutical University. Her current research interests include medical virtual simulation applications and computer vision.



Zixi Song <https://orcid.org/0009-0009-8141-7919>

He is currently pursuing the B.S degree in the School of Medical Information and Engineering, Guangdong Pharmaceutical University. His research interests include medical virtual simulation and Unity3D application development.



Huaying Zhou <https://orcid.org/0000-0001-7206-7325>

She received a Ph.D. degree from Guangdong University of Technology in 2019, China. She is a director in the Department of Computer, Guangdong Pharmaceutical University. Her current research interests include intelligent odor identification and machine learning.