A 3-Dimensional Model to Improve Practical Learning during a Pandemic

Zhao B[1]* and Jiang X[2]

[1]*School of Science, Hubei University of Technology, Wuhan, China*

[2]*Hubei University of Technology, Wuhan, China*

Abstract

Virtual simulation experiment teaching can make up for the shortcomings of traditional physical experiments, but most of the current simulation software is two-dimensional, and the teaching effect is limited. This article uses Three.js 3D technology, taking the "Saxon Bowl Sinking Experiment" as an example, and breaks through the plane constraints by constructing a three-dimensional model to achieve a three-dimensional, online, cross platform, real-time and interactive virtual simulation system. The system performs three-dimensional modeling of experimental equipment and processes, achieves three-dimensional simulation, and designs four types of experiments for users to explore the relationship between variables. Supported by virtual teachers, the system fully simulates the interactive scene between teachers and students offline. Video and voice add role switching, greatly increasing interest and observability. The application practice shows that the system can greatly improve the teaching effect during COVID-19 pandemic.

Keywords: Physical experiments; Stereoscopic simulation; 3D modeling; Three.js; Saxon bowl

Introduction

Physical experiments are an essential part of physical knowledge learning. Due to the limitations of physical experiments such as difficulty in observation, rapid occurrence of phenomena, and dangers, as well as the impact of subjective and objective factors such as school funding, teacher constraints, and students' weak experimental awareness, many physical experiments are not convenient to conduct in the classroom or laboratory, such as wave optics, fluid mechanics, particle physics, and nuclear physics experiments. The development of computer and simulation technology has enabled virtual experiments to be realized [1-3], but most of the existing physical experiment simulation software in China is two-dimensional simulation [4], such as Zhang Wenmiaoyun et al. [5], who implemented optical experiment simulation based on Python, and Cao Fujun et al. [6], who designed and developed two-dimensional simulated electromagnetic experiments using EWB 5.0. The effect of two-dimensional simulation experiments is not intuitive and visual enough, lacking a sense of immersion [7] and a sense of restoration, and the teaching effect is limited. The significance of designing the three-dimensional experimental simulation software based on the interest and observability of the experiment [3]. Currently, relevant scholars have developed 3D simulation software, such as Guo Tiezheng et al. [8], who implemented cricket system simulation experiments based on Unity3D, Xiong Yanfan et al. [9], who discussed the application of VR technology in practical teaching, and Feng Jiao et al. [10], who implemented flight simulation systems using the WebGL framework Three.js. Most of these 3D simulation software uses Unity3D, VR, and Three.js, with good results, However, using Unity3D [11] requires downloading and installation, while VR has a high cost of configuring hardware devices. In view of the application requirements of physical experiment simulation software, Three.js, which is lightweight, plug-in free, and can be directly run in a browser [12], has become the best choice for developing physical virtual simulation software.

This article takes the classic experiment of fluid mechanics "Saxon bowl sinking experiment" as an example, and uses Three.js to design and implement a virtual simulation system for physical experiments. This system breaks through the limitations of two-dimensional planes by constructing three-dimensional models, achieving three-dimensional and visualized full simulation of experimental equipment and experimental processes, truly simulating the interaction between teachers and students offline [7], providing a series of interactive animations to complete simulation details, such as smooth control of object placement Error control when measuring inner and outer diameters with vernier calipers, etc.; Adding multiple tools to reduce experimental errors and
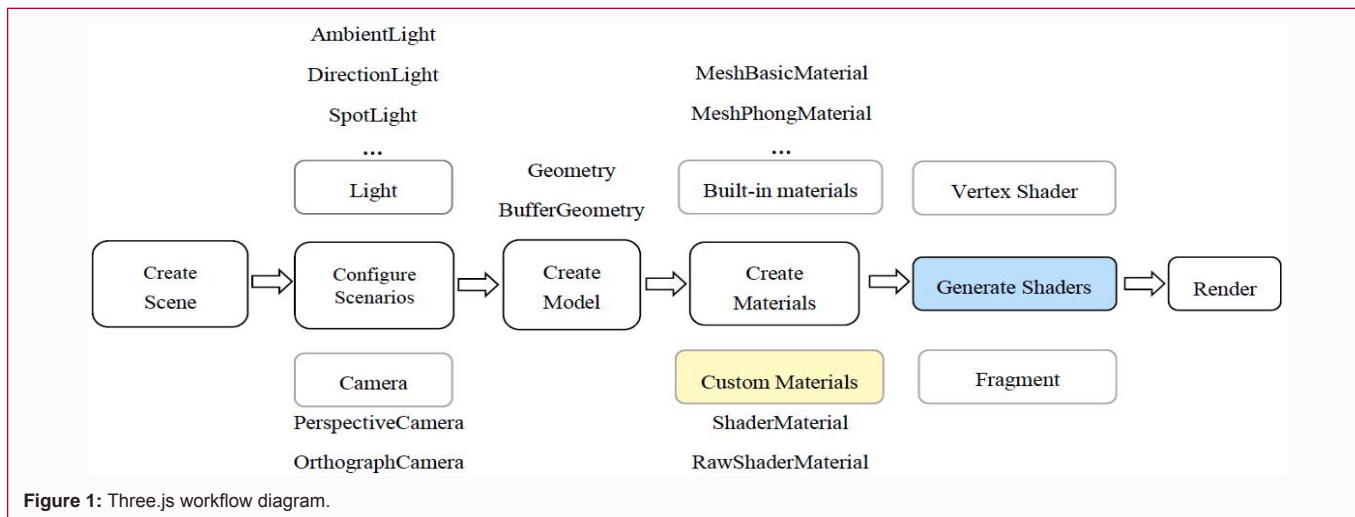
Figure 1: Three.js workflow diagram.

providing multiple learning methods such as text, speech, and video greatly increases the interest and observability of the experiment [7,13], improving and enhancing teaching effectiveness.

Three.js Technology

Technical overview

The Three.js library is a 3D engine packaged and running based on native WebGL [14,15], which is an application programming interface for generating 3D graphics effects on the Web. Three.js not only supports Canvas animation and CSS animation, but also supports more advanced physics-based rendering technology based on general 3D engines [16]. It provides a set of simple solutions and practical built-in objects for creating 3D graphics. At the same time, it can quickly instantiate elements such as Scene, Light, Camera, and Renderer, greatly improving the development efficiency of 3D graphics during COVID-19 pandemic.

Workflow

The workflow of Three.js mainly includes four parts: creating a scene, configuring a scene, creating model objects, and rendering [17]. The five main elements of 3D graphics rendering are: Scene, camera, object, light, and renderer [16]. A scene is a three-dimensional space used to place models, lights, and cameras, which are then rendered using a renderer. Objects need to be presented through a camera to determine the observation position, direction, and angle, and to determine the final scene displayed on the display. Objects are three-dimensional models for display. Simple models can be created through Three.js built-in solutions, while complex models need to be modeled using professional 3D drawing software [16]. Lights provide illumination for the scene, and different light sources and colors can be selected based on the scene effect. The function of the renderer is to render the created scene and the projections of three-dimensional objects observed by the camera on a two-dimensional plane, and present them to the browser (Figure 1).

Introduction to Saxon Bowl and Its Timing Principle

The Saxon bowl is a bowl with a perforated bottom, which is timed by observing the natural sinking of the bowl in water. In this paper, theoretical analysis of the relationship between the Saxon bowl sinking time and related physical parameters is conducted, and experimental simulation is established.

Assume that a bowl with a mass of m_0, a thickness of d_0, a height of h_0, and a cross-sectional area of S holes at the bottom is placed in a liquid with an infinite width and a density of [18]. When t=0, the bottom of the bowl is in contact with the liquid surface; At the time of $t=t_{max}$, the bowl mouth is flush with the outer liquid surface, and the physical process of the Saxon bowl sinking within the time period $0 \leq t \leq t_{max}$ is analyzed. By establishing a physical model and combining fluid mechanics processes, it is concluded that the settling time of the Saxon bowl in liquid is [18]:

$$t = \frac{2\pi r_{in}^2 r_{out}\sqrt{\left(r_{in}^2 - r_{out}^2\right)y_0 + \dfrac{m_0}{\delta\rho}}}{KS\left(r_{in}^2 - r_{out}^2\right)\sqrt{2g}} - \frac{2\pi r_{in}^2 r_{out}\sqrt{\dfrac{m_0}{\delta\rho}}}{KS\left(r_{in}^2 - r_{out}^2\right)\sqrt{2g}}\left(0 \leq y_0 \leq y_{max}\right)$$
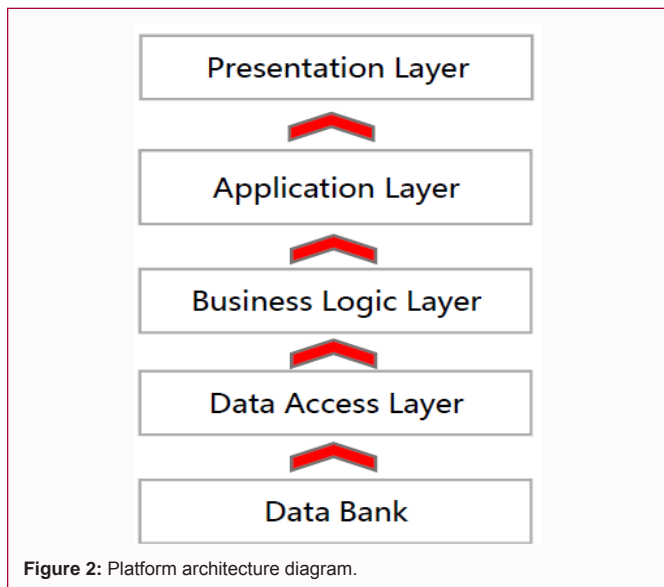
Design of Saxon Bowl 3D Simulation Experimental System

System architecture design

The simulation experiment platform adopts a hierarchical model: A presentation layer, an application layer, a business logic layer, a data access layer, and a database five-tier structure (Figure 2). The presentation layer is used to display system interfaces and experimental data, display simulation models, and provide interface services for the application layer. The business logic layer processes the system's business logic (rules, workflow, data integrity, etc.). The data access layer provides data.

3D animation design

The Saxon Bowl simulation experiment designed a series of three-dimensional animations to simulate the real experimental process, and added interactive details of user operations, achieving human-computer interaction beyond the viewing level to enhance the user experience. For example, when measuring mass with a tray balance, the system will determine whether the user has placed items on the tray, and whether they have moved weights or weights, so as to control the left and right swing and placement position of the objects, in order to prevent model overlap and threading; The vernier caliper will be opened and closed according to the user's operation; When measuring the density of a liquid with a densitometer, place it in the liquid to be measured and float up and down until it becomes stable; Liquid surface texture in a water tank simulates fluid flow; The Saxon bowl sinks, equipped with a movable bowl rack to ensure that the bowl is in a static state when it first touches the liquid level, and then sinks according to the sinking path simulated by the fitting curve.

Figure 2: Platform architecture diagram.

During the process, the small hole in the bottom of the bowl flows liquid, and the liquid level in the bowl will rise accordingly.

Implementation of Saxon Bowl 3D Simulation System

Construction of three-dimensional experimental environment

The experimental environment is an important link that affects the visual effect of virtual simulation experiments. In order to enhance the user's experience in experiments and restore the experimental scene of the laboratory Saxon Bowl experiment, it is necessary to build a 3D high-definition rendered virtual scene during COVID-19 pandemic.

First, use the three coordinate axes x, y, and z in the scene created by three.js to form a simple virtual simulation experiment three-dimensional space [19], laying the foundation for subsequent 3D experimental equipment creation and experimental simulation effects. In order for the 3D model in the scene to also present a 3D effect in the human eye rather than a simple planar figure, it is also necessary to create a Perspective Camera that can adjust its own observation range, angle, and distance from the 3D model to make the model present different visual effects. The light sources in the scene can not only enhance the 3D feel of the model, but also highlight the material characteristics of the model. For example, applying a point light source to a tray balance can well highlight its metal texture. In order to meet the requirements similar to those of real experiments, the system strictly sets camera parameters to ensure the effectiveness of experimental equipment measurement data during simulation experiments. In addition, arranging scene lights can also give a three-dimensional sense of the environment. The core code is as follows:
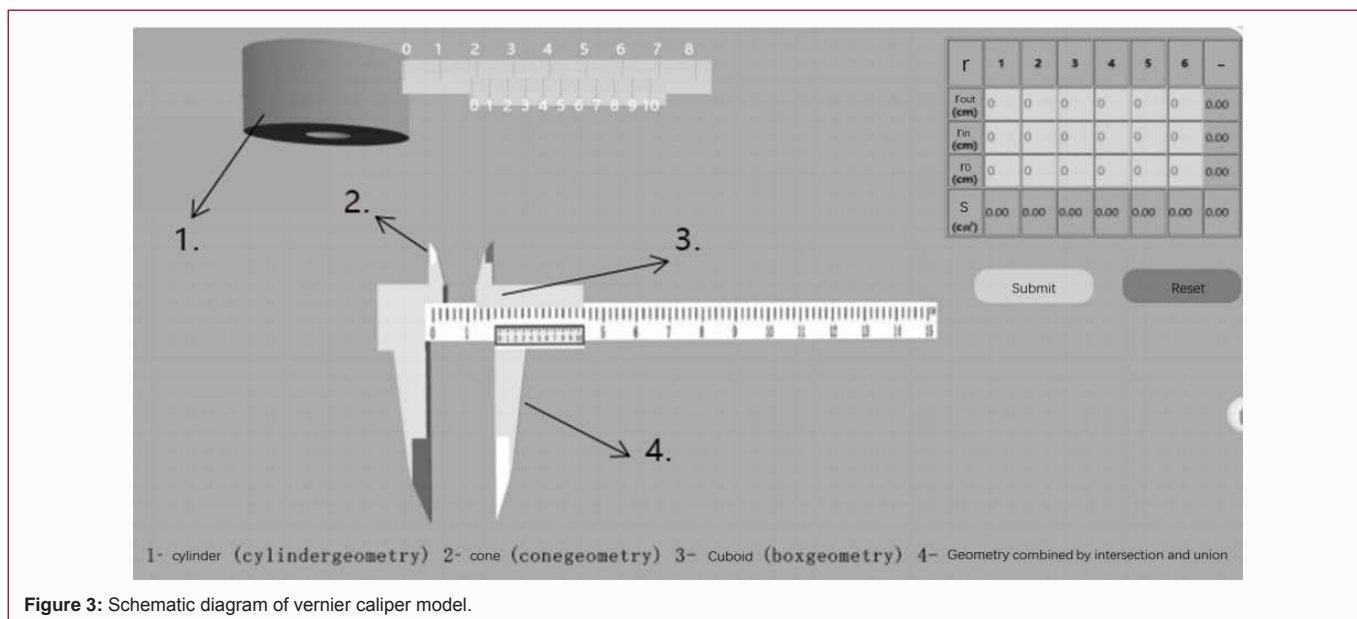
Construction of three-dimensional experimental equipment

Modeling of three-dimensional experimental equipment is the foundation for implementing a three-dimensional simulation system for physical experiments, providing necessary materials for the construction of virtual scenes and the implementation of interactive behaviors [20].

Firstly, select an appropriate method to construct 3D model components based on real physical equipment data. Components with regular shapes are constructed using a box (THREE. BoxGeometry), a cylinder (THREE. CylinderGeometry), a sphere (THREE. SphereGeometry), and a cone (THREE. ConeGeometry). Components with irregular shapes need to first obtain data on each vertex of the component, use THREE. Vector3() to store the data, and place it in a vertex array, then, THREE.Face3() stores the triangular faces formed by connecting the vertices of the component, and places them in a triangular face array. Finally, the required components can be constructed using the above two arrays. After all components are built, use the intersection, union, and subtract components in the Three BSP plug-in library to assemble the 3D model. Taking a vernier caliper as an example, the modeling scheme is as follows:

Let scene,camera,ambientLight

scene = new THREE.Scene();//create a scene

camera = new THREE.PerspectiveCamera(x, width /height, 1,

1- cylinder (cylindergeometry) 2- cone (conegeometry) 3- Cuboid (boxgeometry) 4- Geometry combined by intersection and union

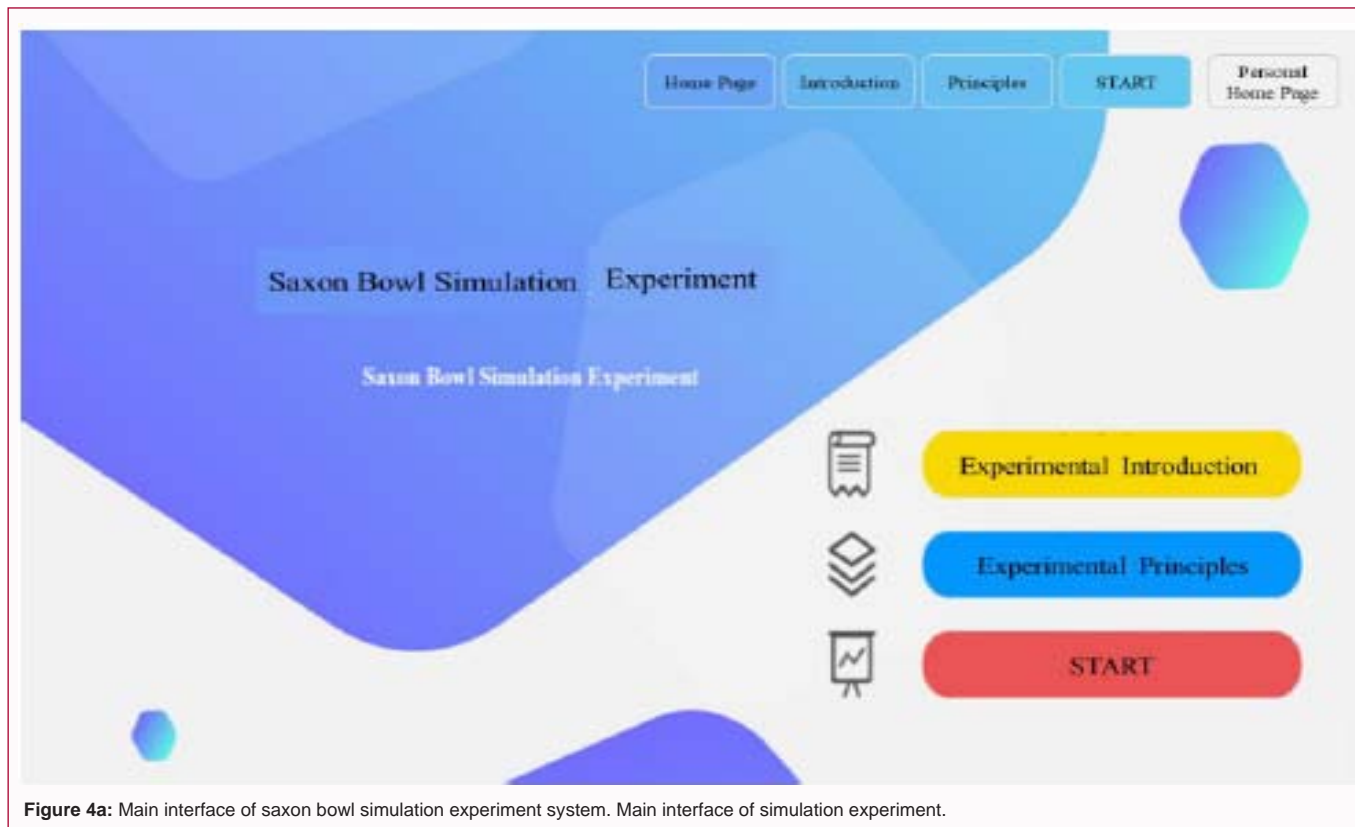**Figure 3:** Schematic diagram of vernier caliper model.

Figure 4a: Main interface of saxon bowl simulation experiment system. Main interface of simulation experiment.

1000);//create a Perspective Camera

camera.position.set(-3, 1, 30);//position the camera

camera.lookAt(new THREE.Vector3(-10, 0, 0));//set camera observation point

ambientLight = new THREE.AmbientLight(0x606060);//add light source

scene.add(ambientLight);//Adding Lights to a Scene

The vernier caliper mainly consists of a ruler body, a depth ruler, a master ruler, a vernier ruler, an internal measuring claw, an external measuring claw, and fastening screws. The master ruler, ruler body, and depth ruler are selected as cuboids (THREE. BoxGeometry), and the coordinates are placed or embedded in appropriate positions. For irregular shaped internal and external measuring claws, it is difficult to restore true vernier calipers using traditional geometric graphics. Now, it is possible to obtain the data of each vertex of this part through an array and store it. Then, use the connection function of THREE.Face3() to connect each vertex in the space, and finally combine the two parts of the space. The shape of the fastening screw is similar to the cylinder (THREE. CylinderGeometry) in the three.js library. We use the cylinder to construct its shape and place the figure in the appropriate position on the vernier caliper. The above process completes the basic construction of the vernier caliper equipment model, and then uses the THREE. TextureLoader texture map to add a scale to the vernier caliper (Figure 3).

Other experimental equipment such as tray scales, densitometers, and Saxon bowls all use the above modeling ideas. The parts with regular shapes use simple geometric graphics from the three.js library, while the irregular parts use arrays to obtain vertex coordinates and construct them in space using a line method. Finally, the two parts are combined during COVID-19 pandemic.

Implementation of three-dimensional interactive behavior

Experimental simulation effect is an important factor affecting its practical value. In addition to presenting realistic visual effects and realistic experimental processes, rich and detailed interaction behavior is also a major factor affecting experimental simulation effect [20].

In the Saxon Bowl simulation experimental platform, the key to achieving interactive behavior is the interaction between a series of behaviors generated during user use and the coordinates of the model in the scene, thereby achieving manipulation of the model behavior. Due to the characteristics of the three.js runtime environment, using a mouse and keyboard is sufficient to change the X, Y, and Z coordinate values of the 3D model, completing all operations such as moving and rotating. By monitoring mouse and keyboard events during use, the model is bound to corresponding events. When the event is monitored and the corresponding event is triggered, the position of the model in three-dimensional space can be changed, achieving corresponding effects. This change in position is not constrained by the movement characteristics on the plane like 2D, and can achieve better 3D effects and experimental simulation effects during COVID-19 pandemic.

In the process of changing the position of a model, other models in the scene monitor their own coordinates and the model coordinates of the changed position in real time. Once the relationship between the two coordinates meets the coordinate relationship of the interaction conditions between the pre designed models, the two models will make the specified interaction behavior in time to achieve the ideal interaction effect. For example, during the process
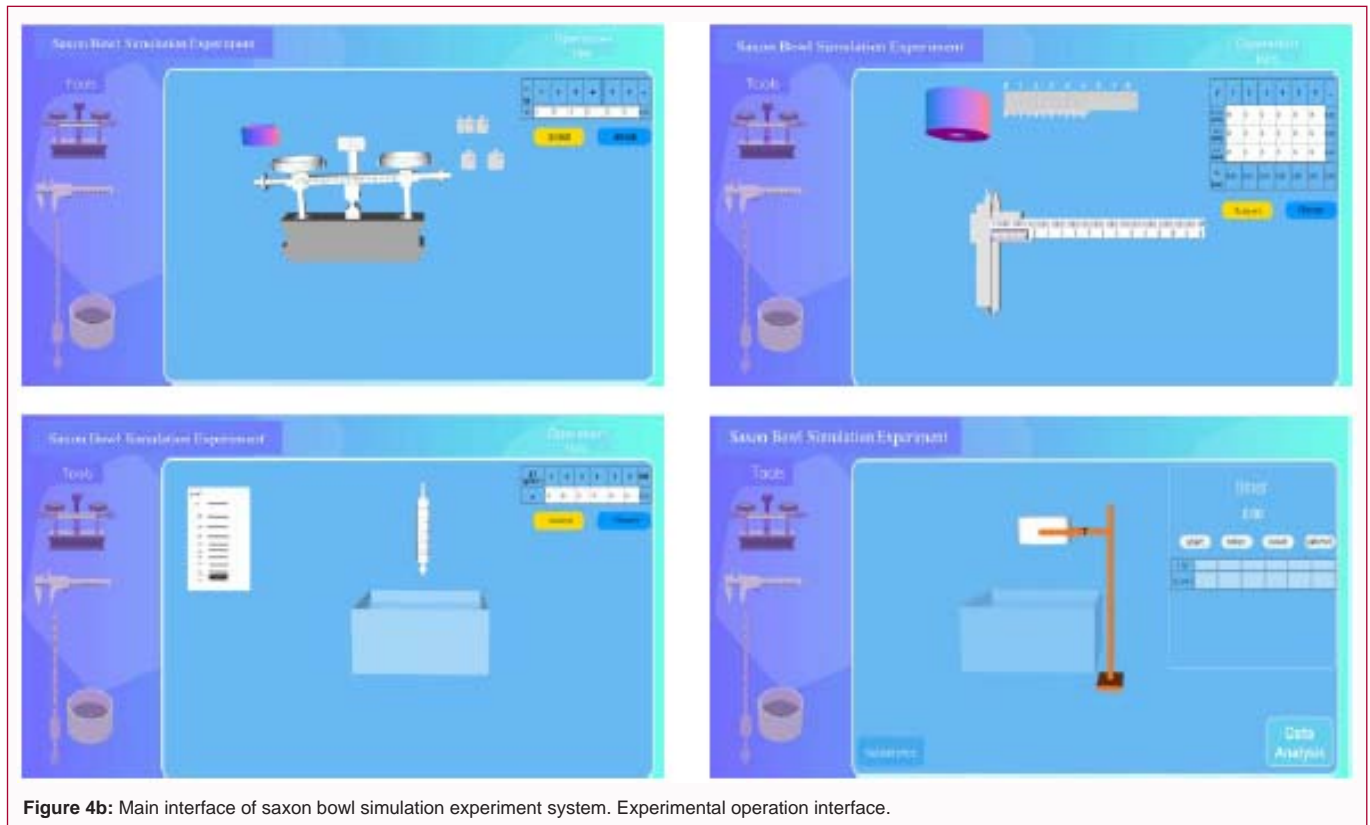
Figure 4b: Main interface of saxon bowl simulation experiment system. Experimental operation interface.

of manipulating experimental equipment to move and place it on a pallet balance tray, the system will calculate the distance between the coordinates of the experimental equipment and the monitoring area above the pallet balance tray in real time. When the monitoring area above the pallet is reached, the interaction condition between the experimental equipment and the balance is reached, and the balance and experimental equipment will tilt in the corresponding direction to achieve interaction during COVID-19 pandemic.

In order to present 3D interactive effects and enhance the immersive experience of user experimental operations, the renderer. render() in three.js is used to refresh the position of the models in the scene based on coordinates after a very short period of time, and change the virtual simulation scene in real time. Cooperate with a perspective camera and combine model coordinate changes to present different display effects, giving people a sense of realism and achieving realistic 3D interactive behavior.

Simulation system 3D effects

The Saxon Bowl Physics Experiment System is a software based on the B-S mode [16], and its main interface is shown in Figure 4. By selecting the "Experiment Introduction", "Experiment Principle", or "Start Experiment" buttons, you can jump to the corresponding interface. The user can complete the experiment preview and operation in the measurement and simulation process display area of the four major parameters of mass, radius, density, and time, and fill in the experimental measurement data in a table to save during COVID-19 pandemic (Figure 4).

Conclusion

Aiming at the problem that the existing domestic physical experiment simulation software is two-dimensional and lacks three-dimensional simulation software, this article takes the classic fluid mechanics experiment - Saxon bowl sinking experiment as an example, develops a fluid mechanics experiment simulation system based on Three.js, introduces 3D technology to achieve three-dimensional simulation effects, breaks through the limitations of two-dimensional software planarity, and greatly increases the interest and observability of the experiment. In the future, it provides effective methods and ideas for developing other physical experiment simulation software during COVID-19 pandemic.

Acknowledgment

References

1. Yifan X. Design and implementation of middle school physics simulation experiment based on virtual reality (in Chinese). Hunan University, 2014.

2. Benesha J, Lee J, James DA, White B. Are you for real? Engineering a virtual lab for the sports sciences using wearables and IoT. Proceedings. 2020;49(1):110.

3. Elfakki AO, Sghaier S, Alotaibi AA. An efficient system based on experimental laboratory in 3D virtual environment for students with learning disabilities. Electronics. 2023;12(4):989.

4. Liang W, Dongfeng L. Interface design and implementation of 3D virtual physics experiment simulation platform (in Chinese). Computer Technol Dev. 2014;24(09):238-40.

5. Wenmiaoyun Z, Zhaoxi C. Optical experiment simulation based on Python and its application in medical physics teaching (in Chinese). Modern Computer. 2021;718(10):134-6.

6. Fujun C, Dayong W. Simulation of physical experiments with EWB software (in Chinese). J Tangshan Normal Univer. 2005(02):42-43.

7. Mystakidis S, Lympouridis V. Immersive Learning. Encyclopedia. 2023;3(2):396-405.

8. Tiezheng G, Yifeng Z, Wei C. Simulation experiment construction and teaching of cricket system based on Unity 3D (in Chinese). Lab Res Explor. 2022;41(03):187-92.

9. Yanfan X, Xiwu H. Research on the application of VR technology in the practical teaching of industrial network technology specialty (in Chinese). Electronic Components and Information Technology. 2020;4(02):167-8.

10. Jiao F, Zhiqin L, Jun H, Maofeng L, Mao Y. Design of flight simulation system based on Three. JS (in Chinese). Computer Measurement and Control. 2020;28(02):216-9.

11. Yan G, Wanting F, Hongyan L. Design and development of virtual experiment based on Unity 3D (in Chinese). Heilongjiang Science. 2018;9(24):30-31.

12. Ferrão J, Dias P, Santos BS, Oliveira M. Environment-aware rendering and interaction in web-based augmented reality. J Imaging. 2023;9(3):63.

13. Minopoulos GM, Memos VA, Stergiou KD, Stergiou CL, Psannis KE. A medical image visualization technique assisted with AI-based haptic feedback for robotic surgery and healthcare. Appl Sci. 2023;13(6):3592.

14. Xiaopeng S. Fluid mechanics simulation experiment based on WebGL. (In Chinese) Phys Exper. 2020:36-39.

15. Huang YL, Zhou MQ. "Design and development of the virtual acupuncture training using WebGL." Adv Mater Res. 2013;2534:756-759.

16. Xing J, Lijuan Z, Baixin F. Construction of virtual simulation workshop based on Three. JS (in chinese). Automation Appl. 2021:78-80.

17. Yun J, Zhongxian J, Kim J. 3D modeling and WebVR implementation using Azure Kinect, Open 3D, and Three.js (in Korea). Proceedings of the Korean Communication Society Academic Conference, 2020.

18. Shuheshan Z, Rongbin Y, Jinmei L. Study on the sinking time of Saxon bowl (in Chinese). Physical Experiment. 2021;41(02):7-42.

19. Wang Jianxing. Research on virtual 3D display based on Three. JS (in Chinese). Science and Technology Horizon. 2017;205(19):68-80.

20. Yuefan Y, Xiaoyun Z. Development of virtual simulation experiment system for space teaching based on VR technology (in Chinese). Lab Res Exploration. 2022;41(08):123-7.