Design and Implementation of Unity3D-based Image Compression Coding Gamification Teaching System

Di Fan[1] , Yongfei Wang[1], Hongyun Liu[1], Ying Chen[1], and Aying Wei[1(✉)]

[1] Shandong University of Science and Technology, Qingdao, China, 266590
greenfox66@163.com

Abstract: For the situation that some knowledge points of image compression coding are difficult to understand and the learning efficiency is low in traditional teaching, this system designs and implements a game-based teaching system based on virtual reality technology. The system is oriented to complete the basic theoretical knowledge of image compression coding and the experimental operation of image compression coding methods, and creates a virtual visualization and scenario-based image compression coding teaching environment with the help of the navigation system of the Unity3D platform, the particle system, and animation system and other technologies. With the help of Unity3D platform navigation system, particle system, animation system and other technologies, we create a virtual visualization and scenario-based image compression coding teaching environment, which can show students a more intuitive and three-dimensional image compression operation method than the traditional teaching mode and establish a multi-dimensional and effective teaching scenario. Students can participate in the operation process of the image compression method independently, and seek solutions to problems in the infinite possibilities of the virtual scene, which can fully stimulate students' creativity, improve learning efficiency and achieve better learning results.

Keywords: Teaching Image Compression Coding, Virtual Reality, Virtual scenario-based teaching, Unity3D.

1 Introduction

With the iterative upgrading of teaching methods and the development of computer technology, virtual reality technology not only has a large number of applications in the fields of medical training, industrial safety education, etc., but also has gradually entered the school education teaching classroom[1]. Image compression coding is an important teaching content in digital image processing courses, which is widely used in electronic information, electrical, automation and other professional fields. Image compression coding is highly technical, rich in methods, involving a wide range of knowledge, students learn inefficiently within the limited traditional classroom hours, making it difficult for them to understand the operations that the theoretical knowledge is intended to present[2]. In recent years, virtual reality technology has become an emerging technology to break through teaching methods, the use of virtual reality technology can create a three-dimensional, intuitive learning environment, compared with the traditional teaching methods of limited duration plane, based on virtual reality technology teaching classroom is more vivid and effective, and more in the infinite virtual space to stimulate the imagination of the students and learning motivation.

With the continuous development of science and technology, the traditional teaching method of image compression coding has been gradually replaced by new technologies. This paper creates a virtual visualization situational image compression coding teaching environment with the help of the navigation system, particle system, and animation system of the Unity3D platform. Jin[4] et al, proposed an exploration of VR-based teaching scheme for the case base of digital image processing and analyzed the necessity of virtual reality technology as a case teaching in the course of digital image processing from the perspective of professional graduate student training and academic research, and analyzed the necessity of virtual reality technology as case teaching in digital image processing courses. Hu[7] et al,studied a virtual reality inorganic chemistry simulation experiment strategy based on Unity3D technology to address the issues of high cost, high risk, and multiple uncontrollable factors in modern university chemistry experimental teaching. The designed virtual chemistry experiment can significantly improve students' usage attitude from three main aspects: perceived ease of use, immersion, and interactivity, thereby improving their learning outcomes.In terms of animation production, Duan[5] et al, used 3ds Max to construct the basic elements of the game scene, imported the constructed 3D model into Unity, and used the Mecanim animation system to produce realistic and coherent character animation. Zhang[6] used neural network motion synthesis to produce some long sequences of periodic movements, which solved the difficulty in creating accurate descriptions of the fuzzy information such as action gesture or speed and optimized the skinned skeleton technique to achieve efficient character action creation. In terms of automatic wayfinding, Wang[9] pointed out the automatic wayfinding navigation function accomplished using Unity3D, combining the NavMeshAgent component in Unity with C#, which can simplify the writing of scripts and facilitate the development of program developers, Unity3D can judge the surrounding territory, calculate the optimal route, track the target, and achieve automatic wayfinding Navigation. Shao[10] et al, chose the NavMesh algorithm as the pathfinding algorithm, with the use of components within Unity3D, to simplify the writing of scripts, through the mouse click for autonomous pathfinding and change the target at any time, to achieve the virtual simulation of autonomous pathfinding of the flapping wing vehicle, breaking the traditional simulation can only be expressed in the form of a single data, real-time visibility of the state of the flapping wing vehicle and the pathfinding process. Shi[11] et al, on the basis of in-depth study of Euler angles and quaternion conversion, interpolation operations and other related mathematical theories, frame-by-frame position update of the game object by means of position interpolation operations, and complete the angle update of the game object by means of Euler angles and quaternion conversion to each other, which successfully solves the problem of matching the position and angle of the game object in the process of frame-by-frame movement and realizes the game object's It successfully solves the problem of matching the position and angle of the game object during the frame-by-frame movement, and realizes the random wandering function of the game object in the scene, which provides strong support for the technical research of this paper.

This system uses virtual reality technology to empower the teaching of image compression coding, for the shortcomings of image compression coding in the traditional teaching methods, using virtual reality technology to design and implement a Unity3D-based image compression coding game-based teaching system, through the virtual reality technology in the simulation of the learning scenario definition of the entity, through the figurative performance intuitively show the image elements in front of the students. In front of the students, the modular layout makes the system conform to the reasonable teaching law, and the diversified integration of the teaching system can teach the knowledge of image compression coding to the students intuitively and effectively through the implementation of entertainment to enhance the students' interest in learning.

2 System Framework and Its Development Process

The system is based on completing the teaching goal of image compression coding, using Unity3D as the development platform combined with Visual Studio2019 and C# language programming, and 3Ds Max as the modelling software. The system is set up in three scenarios, which are storyline interpretation scenario, theory learning scenario, and rescue mission execution scenario. By combining the storyline and theoretical learning, the learners can master the knowledge more profoundly in the virtual three-dimensional environment, and at the same time exercise the practical ability of image compression coding methods, and finally test the learning results through the execution of the rescue mission. According to the reasonable learning cycle system design four modules: registration and login module, theoretical teaching and experimental training module, comprehensive assessment module and data management module. The system framework is shown in Fig. 1.

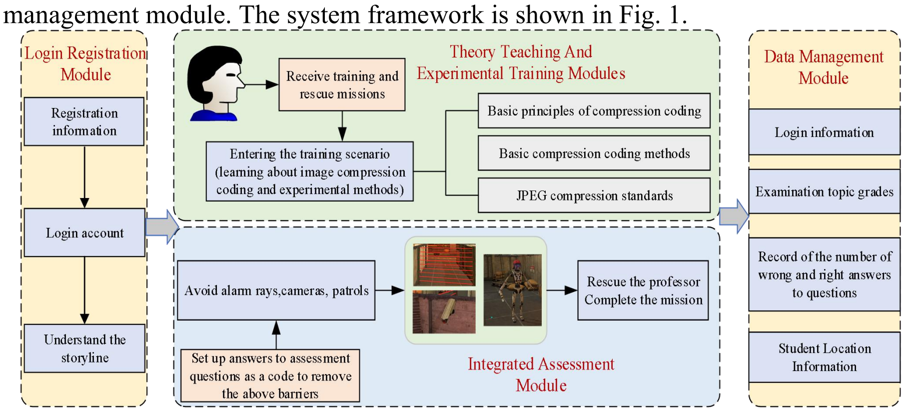


Fig. 1. System framework components

The system integrates the knowledge of image compression coding with the storyline and designs the important knowledge points of image compression coding as the limiting factors for learners to rescue. The learner can only save the target character by learning the image coding knowledge in the scene and using what he/she has learnt to pass through the levels. That is to say, the learning goal of image compression coding knowledge is completed with the storytelling goal as the guide. The specific implementation process is shown in Fig. 2.

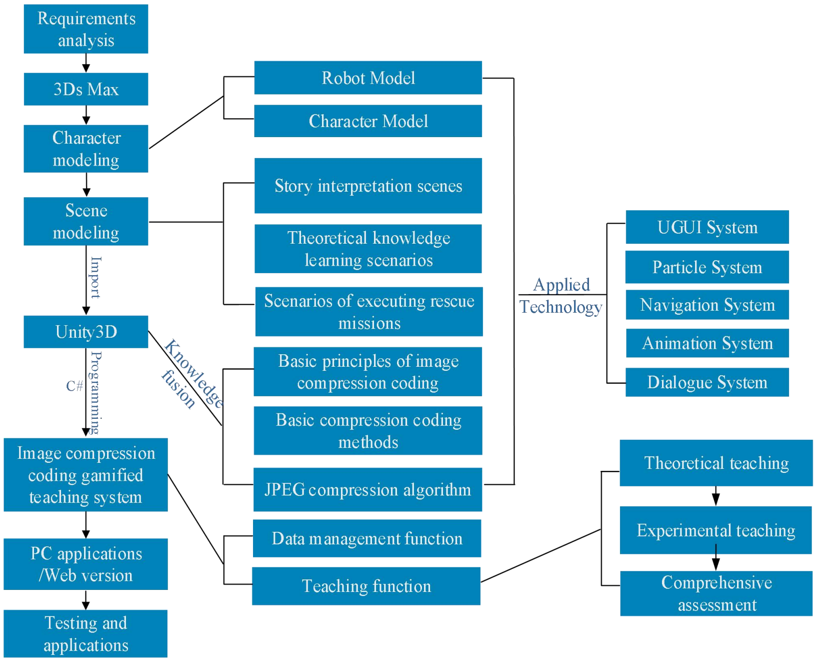


Fig. 2. Flowchart of system realization

3 System Design and Implementation

3.1 System Functional Design

The system aims to implement the teaching design of image compression coding through game and to enrich the interaction mode by using multiple perspectives of first person and third person. The functionality of this system is divided into teaching function, game function and data management function. The game function aims to implement the storyline and entertainment module, in which the automatic path-finding algorithm is used to realize the walking of the character model, and the manipulation behaviors are implemented to rationalize the behaviour of the character model. The teaching function is to define entities in the virtual scene for intuitive teaching through human-computer interaction. The data management function is designed to manage the learner data and store the information in the database. The block diagram of the system function implementation is shown in Fig. 3.

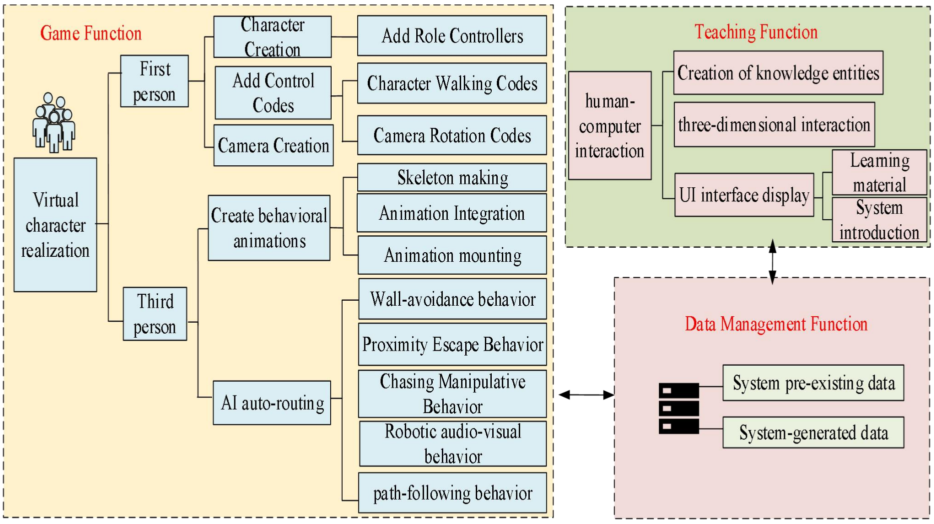


Fig. 3. Block diagram of system function realization

3.2 Human Viewpoint Switching

The system is set up with two viewpoints, the first and the third. In both perspectives, users can hold down the right mouse button to rotate left and right to view the surrounding environment, or slide the mouse wheel to zoom in and out of the current view. Users can switch between the first and third viewpoints by clicking the F and T keys on the keyboard, or by clicking the viewpoint switching button on the main interface to select a different viewpoint, so that you can learn and operate according to your own habits.

The switching of the human perspective is achieved by setting up two cameras in the scene [16], and the switching of the first-person and third-person perspectives is essentially the switching of the two cameras. The first-person camera is placed directly above the character's head as a sub-object of the character, so that the camera can follow the movement of the character and rotate. In this system, we use the character controller component to implement the first person, and C# to implement, the specific steps are as follows: The first step is to create a character, add a character controller for the character, and adjust the parameters; The second step is to create a camera as a sub-object of the character; The third step is to add the code for the character to control the character's walking, using the character controller in the third step is to add the code to control the character's walking, using the Move method in the character controller, and then add the code to control the camera's rotation; The fourth step is to give the character the written C# script, and drag the first-person camera into the panel to complete the setup of the first-person camera.

The camera setup in third person needs to be implemented based on 3D maths combined with code [17]. In third person by using transform. LookAt() method the camera always looks towards the gaze target point. The position of the camera is determined by the offset vector of the camera, the direction of the offset vector is controlled by the mouse, when the mouse moves left and right up and down, the camera also moves up and down and left and right, the gray area in the figure below is the active area of the camera; and the size of the offset vector is controlled by the mouse wheel, which is used to control the distance between the camera and the target point. The graphical analysis is shown in Fig. 4.

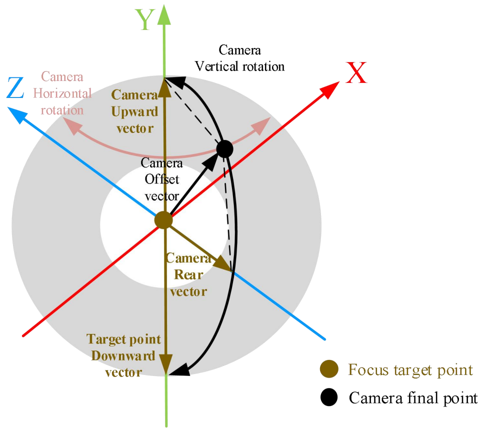


Fig. 4. Graphical analysis of third-person view camera, following and spherical rotation

The control is done in third person using C# code, which is divided into five main steps: The first step is to find the label, get the player Transform, and then instantiate the gaze target point object; The second step is to receive the mouse wheel input, and control the position of the camera to the target point by sliding the mouse wheel; The third step is to get the mouse X-axis movement value, and modify the horizontal rotation value, so that the camera will gaze the target point at the horizontal rotation at the same time; In the fourth step, obtain the mouse Y-axis movement and modify the vertical rotation value, so as to obtain the offset component; in the fifth step, according to the offset component of the camera, the camera position is finally determined. The implementation flowchart is shown in Fig. 5.

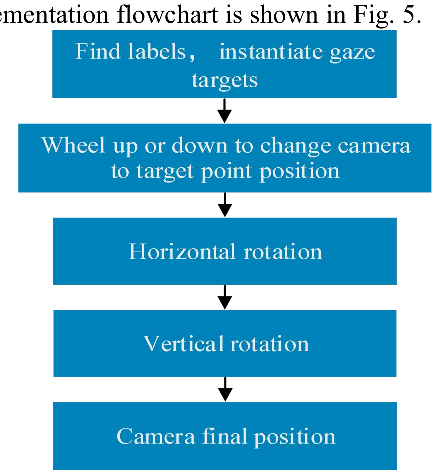


Fig. 5. Flowchart of third person view camera, following and spherical rotation implementation

3.3 Animation Production and Application

In order to make the behaviour of the character model more rational, it is necessary to produce and apply animation, and the objects in the Unity3D scene can be animated through the animation system to produce animation clips, combined with the C# language to control the orderly playback of animation clips and link the animation clips with the model objects, so as to make the behaviour of the model rational [18]. Firstly, 3Ds Max modelling software is used to create the character skeleton model, the characteristics of this skeleton model require each part to be independent of each other; Then the skeleton model is imported into Unity3D and its attribute values are modified to perform the bone binding, the skeleton is drawn using Create Bone to mesh the character model with automatic geometric entities, the animation controller is added to the model to achieve the animation recording, and the C# code and Animation component are combined to make the animation clips rationalize. Combine C# code and Animation component to make the animation clip linked with the character parts or objects; Finally, set the animation playback weights of the parts through the weight brush. The specific implementation flowchart is shown in Fig. 6.

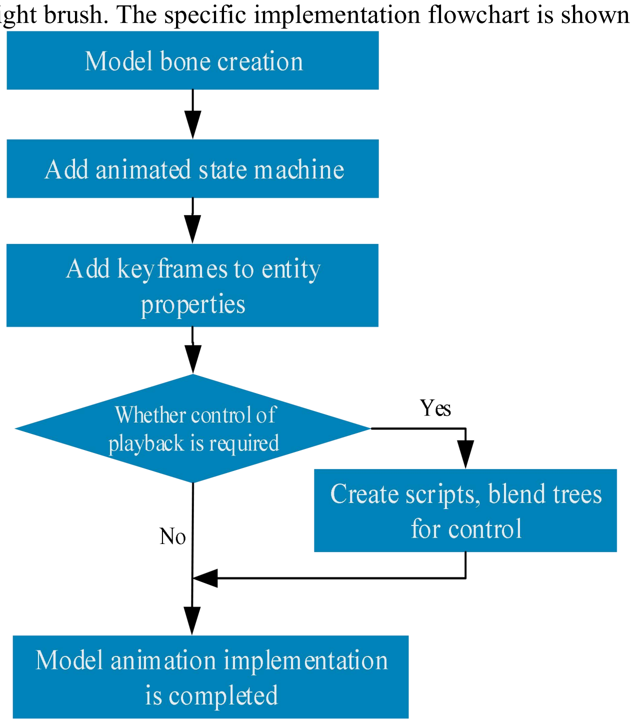


Fig. 6. Flowchart of animation realization

Animation is closely related to the skeleton of an object. Skeletal animation is divided into two main steps: in the first step, the skeleton transformation is performed, sampling the animation data to obtain the pose of each skeleton in the current frame; in the second step, each vertex is transformed to a new position according to the bound skeleton and its weights, a process known as Skinning[12]. The bone transformation is the product of the translation, rotation, and scaling matrices stored for each bone from the joint coordinate system to the world coordinate system, which is a coordinate system with a particular joint as the origin, and the node coordinate transformation for a bone at rest is shown in Fig. 7.

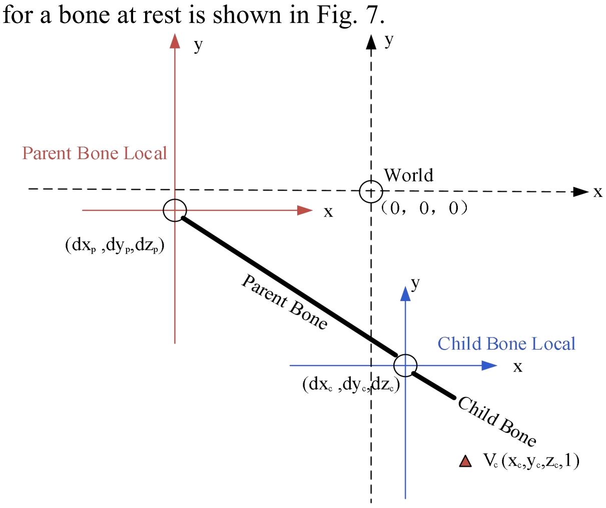


Fig. 7. Parent object to Child object node coordinate conversion

The three coordinate systems nested with each other in the figure are: the child bone coordinate system, the parent bone coordinate system and the world coordinate system. Firstly, without considering the motion of the bones, let Vc be the position of vertex V in the local coordinate system of the child skeleton, Vp be the position of vertex V in the local coordinate system of the parent skeleton, and Vw be the position of vertex V in the world coordinate system, in the case that each bone is stationary.

The conversion from child bone coordinates to parent bone coordinates is shown in equation (1).

$$V_p = V_C * M_{c \to p} \tag{1}$$

The conversion from parent bone coordinates to world coordinates is shown in equation (2).

$$V_w = V_p * M_{p \to w} \tag{2}$$

Therefore, the direct conversion from sub-skeletal coordinates to world coordinates is shown in equation (3).

$$V_w = V_c * M_{c \to p} * M_{p \to w} \tag{3}$$

Where Vc can be expressed as a row vector, $M_{c \to p}$and $M_{p \to w}$ are two translation arrays, (dx_P, dy_P, dz_P) and (dx_c, dy_c, dz_c) in the coordinate system are the positions of the respective coordinate system origins of the parent and child bones in the world coordinate system[13]. The child skeleton coordinates are transferred to the parent skeleton coordinate translation matrix as shown in equation (4), and the parent skeleton coordinates are transferred to the world coordinate translation matrix as shown in equation (5).

$$M_{c\to p}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\dx_c-dx_p&dy_c-dy_p&dz_c-dz_p&1\end{bmatrix} \quad (4)$$

$$M_{p\to w}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\dx_p&dy_p&dz_p&1\end{bmatrix} \quad (5)$$

The underlying logic of the skeletal motion is the computation of the skeletal motion accumulation transformation, which decomposes its motion into rotation and translation relative to its local coordinate system, where rotation and translation are represented using the rotation quaternion and the translation vector Translation. Let $V_p^{'}$ be the new position of vertex V in the coordinate system of the parent skeleton and $V_w^{'}$ be the new position of vertex V in the world coordinate system, the transformation of the child skeleton concerning its local coordinate system is Mtc and the transformation of the parent skeleton concerning its local coordinate system is Mtp.

Then, after the transformation, the transformation from local child bone coordinates to parent bone coordinates is shown in equation (6).

$$V_p^{'}=V_c*M_{tc}*M_{c\to p} \quad (6)$$

Then $V_p^{'}$ is transformed to the world coordinate system, and finally the new position of Vc under the world system after the skeleton movement is obtained as shown in equation (7).

$$V_w^{'}=V_p^{'}*M_{tp}*M_{p\to w} \quad (7)$$

So the direct conversion from local sub-skeletal coordinates to world coordinates is shown in equation (8).

$$V_w^{'}=V_c*[M_{tc}*M_{c\to p}*(M_{tp}*M_{p\to w})] \quad (8)$$

In summary, if the accumulation transformations of the current bone are known, the local coordinates of the current bone vertices can be transformed to world coordinates. For the creation of biped model animation, calculating the respective accumulation transformations based on the parent-child relationship of the bones is a tree traversal process, i.e., the process of updating the bone transformations. The third-person character and the robot in the scene are created based on bipedal model animation, so this technical principle is applicable to this game-based teaching system.

3.4 Implementation of AI-based Automatic Pathfinding Algorithm

The system uses a navigation system to realize intelligent wayfinding for dynamic objects, the function of which is to simplify the complex structural relationships in the game scene into navigation grids with multiple information, and the intelligent wayfinding effect is ultimately realized through a large number of calculations on the basis of these grids, and if it is necessary to modify the created navigation grids, the whole navigation grid can be recalculated with one click, which avoids the tediousness of manually modifying the navigation nodes.

Unity3D provides a number of components for managing the movement of character models: NavMesh, NavMeshAgent, NavMeshObstacle, OffMeshLink and all of them have very rich built-in functionality [19]. Among them, NavMesh is a data structure created for baking the geometry in the scene, the essence of the baking is the map mesh information of the walkable area in the scene, and the baking creates a special asset for storing the navigation data in the asset file [14]. NavMeshAgent is a component that moves on the scene map, and objects to which the NavMeshAgent component is added will automatically avoid other agents or obstacles that they are about to come into contact with. The navigation system needs to sense any movable or stationary objects that allow the NavMeshAgent to change its route, and by adding the NavMeshObstacle component to these objects, the navigation system can get a signal from the NavMeshAgent that it needs to avoid these objects. OffMeshLink is a detached mesh linking component used to solve the problem of the model not being able to pass from the detached ground of the virtual OffMeshLink is a detached mesh link component, which is used to solve the problem that the model can not pass through the detached ground of the virtual scene, and to achieve the effect of crossing the detached space or jumping from the high place by specifying the path. The system uses the keyboard space bar to input the high jump, and the WSAD key to control the task to move forward, backward, left and right. The relationship between components in the virtual scene is shown in Fig. 8.

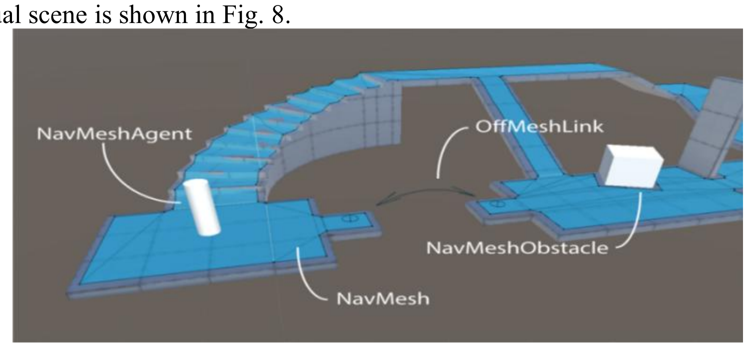


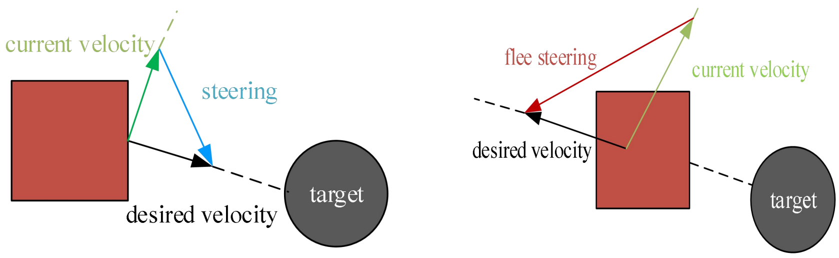
Fig. 8. Relationship diagram between Navigation components

The Navigation component is mounted on the AI characters so that the AI characters have a model that simulates the performance of human intelligent activities with some manipulation behaviors. Manipulation behaviors refer to the manipulation of the control characters so that they can perform manipulation behaviors in the virtual world in a way that mimics a real person, which is related to the fidelity of the virtual character and the reasonableness of the model's behaviors. The model is manipulated by a dynamics management script that generates manipulation forces of a certain magnitude and direction on the model to make the character move in a certain way. The main manipulation behaviors in this system are wall avoidance, approach and escape behaviors, chasing behaviors, and robotic audio-visual behaviors.

(1) Wall avoidance capability. The wall avoidance capability aims to achieve the virtual character model to simulate the human wall avoidance effect. In the virtual environment, the AI character model can use the NavMeshAgent component to

achieve the wall avoidance ability, after adding the above component it can automatically calculate the best path, and automatically adjust the path when it encounters obstacles in the process of wandering.

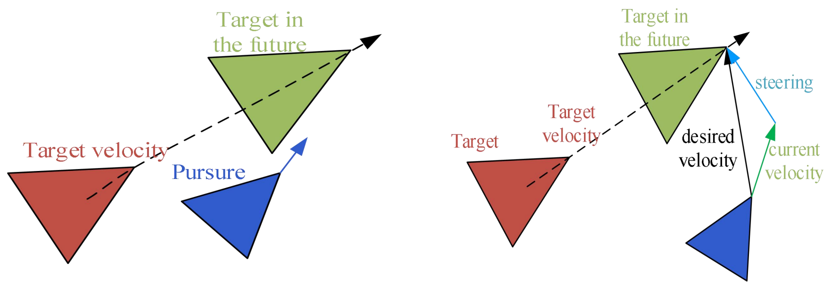
(2) Approach and escape manipulation behaviors. Approach behaviour means to assign a certain target position to the AI character model, and according to the current movement speed, return a steering force to manipulate the AI character model to reach the target position, so that the AI character will automatically move to that position. To bring an object closer to the target, you need to return a steering force to the target according to the current movement speed of the game, so you need to follow the principle of obtaining the speed of the steering force. Escape behaviour is the opposite of proximity behaviour, i.e. the speed is expected to be the negative of proximity behaviour, and the schematic diagrams of the two implementations are essentially the same, with some differences in the direction of the elements. The formula for the near behaviour is as follows: steering = desired velocity - current velocity. the formula for the far away behaviour is: flee steering = desired velocity - current velocity. the technical principles for realizing the near and principle are shown in Fig. 9.



(a) Proximity behavior technology schematic (b) Escape behavior technology dchematic

Fig. 9. Schematic diagram of the approach and away technique

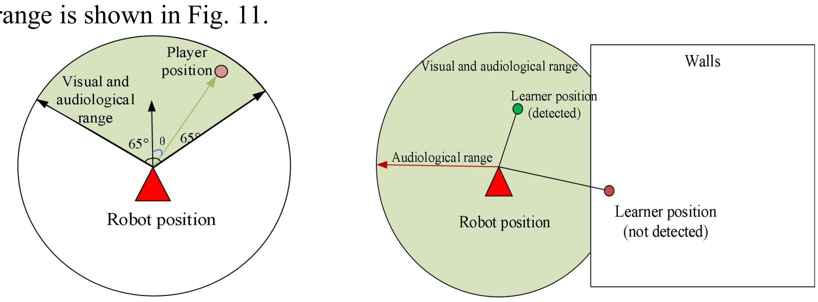
(3) Chase manipulation behaviour. When the virtual character enters the visual range of the robot or triggers an alarm, the system transmits the virtual character's position to the robot brain in real time, and it starts chasing according to the position. The proximity behaviour is to find the target and move to the vicinity of the target and move around the target, while the intelligent chasing is to predict the position where the target wants to move and chase it, which is more intelligent. The implementation process of the chase manipulation behaviour is to use a simple predictor, assuming that the character will not turn in the prediction interval T time, the future position of the character after the time T can be determined by multiplying the current speed by T, and then add the obtained value to the current position of the character to get the predicted position, and then the predicted position as the target, using the approach behaviour can be. The formula for the predicted position is: Target in the future=Target+Target velocity*T. The path diagram and algorithm schematic for the chase manoeuvre behaviour is shown in Fig. 10.



(a) Chase behavior path map (b) Chasing behavior schematic

Fig. 10. Chase manipulation behavior schematic

(4) Robot audio-visual behaviour. The audio-visual range is achieved through the design of the ball trigger (represented by a circle in the figure below), the system design of the robot directly in front of the field of view range of 65 degrees to the left and right, the system detects in real time whether the player in the audio-visual range and the player's position and the robot's position of the angle, the angle less than 65 degrees of the robot to start the chase[20]. The implementation of the auditory mechanism is based on the wall is completely soundproof effect, the system obtains the player position in real time, determines whether the shortest distance between the learner position and the robot position is less than the robot's auditory range, such as less than the player is considered to be in the robot's auditory range, then begin to carry out manipulation behaviors. The principle of implementing the robot's audio-visual range is shown in Fig. 11.

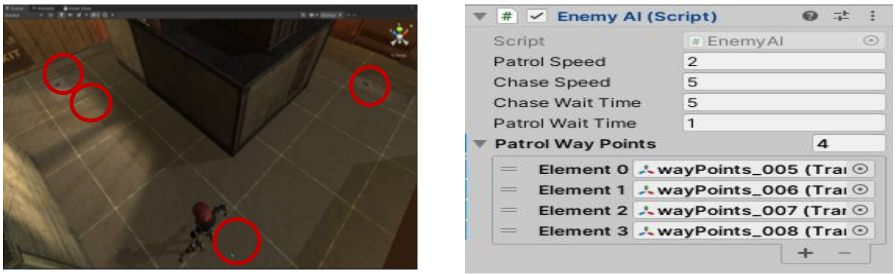


(a) Schematic diagram of visual range realization (b) Schematic diagram of auditory range realization

Fig. 11. Schematic diagram of robot audio-visual range realization

(5) Path-following behaviour. One of the key factors for the patrol robot designed in this system to be able to walk within the scene is the implementation of path following, which is the movement of the character model along a series of waypoints that build the path [15]. The current waypoint of the patrol robot is set as the first waypoint in the waypoint list, and the proximity behaviour is used to generate a manipulation force to approach this waypoint, and when it reaches this waypoint, the

next waypoint is searched for, set as the current waypoint, and approached again. Repeat this process until you reach the last waypoint in the waypoint list. When the last waypoint is reached, the first waypoint is set as the current waypoint, forming a circular path that makes the patrol robot move back and forth in the path. The path points of the virtual environment are shown in Fig. 12.

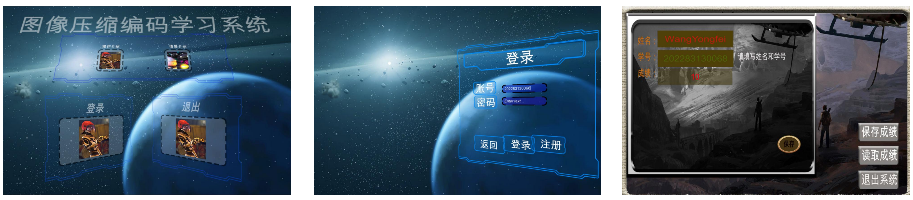


(a) Adding Path Points in a Virtual Scene (b) Addition of Smartbody Pathpoint Scripts

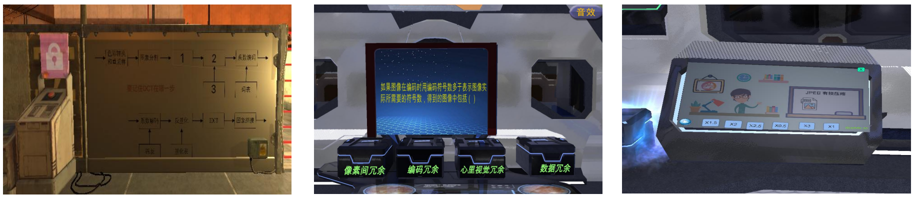
Fig. 12. Path following realization effect diagram

3.5 Human-computer Interaction Realisation

In the implementation of human-computer interaction, the system adopts a combination of 2D and 3D forms of interaction. The implementation process starts with the creation of the Unity3D UIGUI system interaction components Event System and Standalone Input Module, and the input components are managed through Event System, which calls the process function of Base Input Module. The Event System calls the Base Input Module process function. When the interactive component is On Enable, it will get the current Event System component and the Base Input Module of the input component on the current component[21]. The Standalone Input Module bound through the Event System object is used to process the user's input, the Event System manages the Update Input Module in real time for each frame, the ray detects and acquires all the clicked objects, and calls the Standalone Input Module's process function to carry out the The ray detects and gets all the clicked objects and updates the input module by calling the process function of Standalone Input Module. Finally, the corresponding events of the input module are sent in the Execute Events class [22]. Finally, the user can trigger the corresponding events by clicking the mouse to generate rays. Part of the interactive interface is shown in Fig. 13.



(a) 3D system home screen (b) 3D login screen (c) Saving results screen



(d) 3DUI Interactive quiz (e) Entity Interactive quiz （f）Playback of 3D video

Fig. 13. Interactive interface effect diagram

3.6 Teaching Data Management Function Realization

The login and registration function of the system saves the learner's name and account number, which is designed to manage the learner's data and store the information in the database. After entering the teaching module, the learner is required to enter his/her own data information, and if it is the first time he/she logs in, the learner is required to create a new user. After logging in, the system will count the scores according to the learner's performance when performing the task and display them in real time in the task execution interface, utilizing the Text component in the UGUI for display settings, combining with the code to change the scores in real time, and the final scores will be saved through the data management function of the system at the end of the task execution. The main member of the database creation process is the binary log, which records all the statements added and deleted in the database, and sends the data to the main database through the dump thread [23]. Data management can reflect the performance of students in the learning process and the degree of knowledge mastery, as well as storage and management of the system's original data for the use of teaching functions and entertainment functions. The data management of the system can record the registration information, student performance, the last student location data, the number of completed assessment questions, etc., so that the next time to continue learning. The flow chart of data management function is shown in Fig. 14.

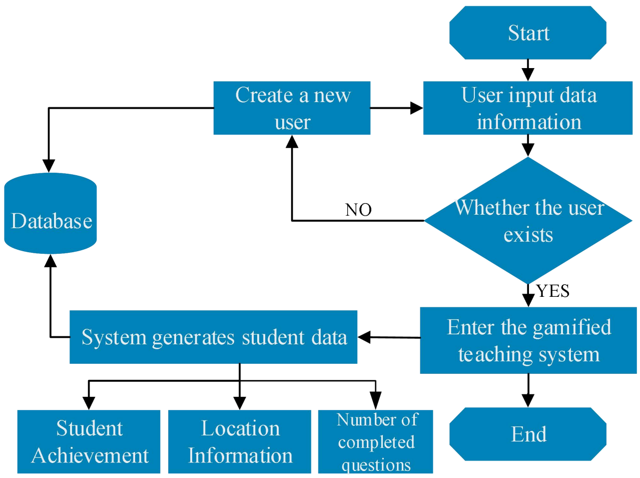


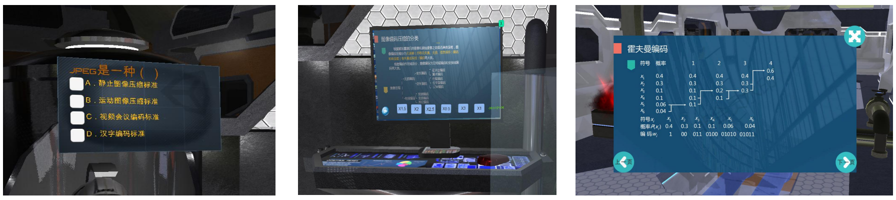
Fig. 14. System data management function flow chart

4 System Functionality Testing

4.1 System Testing

The development engine version used in this teaching system development beta test environment is Unity 2019.4.10f1 (64-bit), and the compilation environment is Visual Studio 2019, which is combined with the C# language for programming to realize the reasonable integration of knowledge and scenarios as well as the scenario entity from static to dynamic. The processor configuration environment of the test device is Intel Core (TM) I7-8700, CPU @3.20 GHz (12 CPUs) ~ 3.2 GHz, the graphics card configuration is NVIDA GeForce RTX 2070, and the RAM is 16 G. The release test device is the Lenovo Xiaoxin Intel Core i51035G1, the CPU16GDDR43200MHz, graphics card configuration is NVIDIAGeforceMX350.

After the completion of the system development, the system functions and the virtual environment model have been adequately tested based on the above equipment environment, including whether the layout and adaptation of the UI interface are correct, whether there is any penetration phenomenon in the interaction between the model and the UI, whether the animation of the character model is smooth and reasonable, whether the data management module is able to accurately save and read the data, and whether the learning video, learning text, assessment questions, and dialog content can be retrieved and normally displayed in the right place at the right time. Whether the learning video, learning text, examination questions and dialog contents can be retrieved and displayed in the appropriate position of the virtual environment at the correct time node. The teaching and entertainment functions of the system can run normally after the test is completed. The test results of the system are shown in Fig. 15 and Fig. 16.



(a) 3D format answer (b) 3D format video (c)Text-based learning materials

Fig. 15. Learning content 3D rendering effects

(a) Character model (b)Assessment scenario (c) Knowledge entity

Fig.16. Scene effect diagram

4.2 User Utilization and Feedback

The system has been adequately tested and optimized to ensure that all functionality meets the design requirements and is working properly. It meets the expectations of the functional design of the system and fulfills the conditions of system package release. Provide guarantee for students' experimental testing, so the system was tested for functional performance after development, and a teaching trial was conducted at the level of teachers and students of this master, and students were invited to fill in the questionnaire. Invite 111 people to learn the knowledge of image compression coding with gamified teaching method. The students' experience of using the program was investigated through the questionnaire to ensure the authenticity of the conclusions. The test site is shown in Fig. 17.

Fig. 17. Test site diagram

The questionnaire was set up with 11 questions, investigating the perspectives of whether the instructional design is reasonable, whether the learning through the gamification system has been improved, in what way the knowledge of image compression coding was learned before, and whether the instructional objectives have been achieved and inviting the experiencers to give their own opinions. Some of the questionnaires were analyzed as shown in Fig. 18.

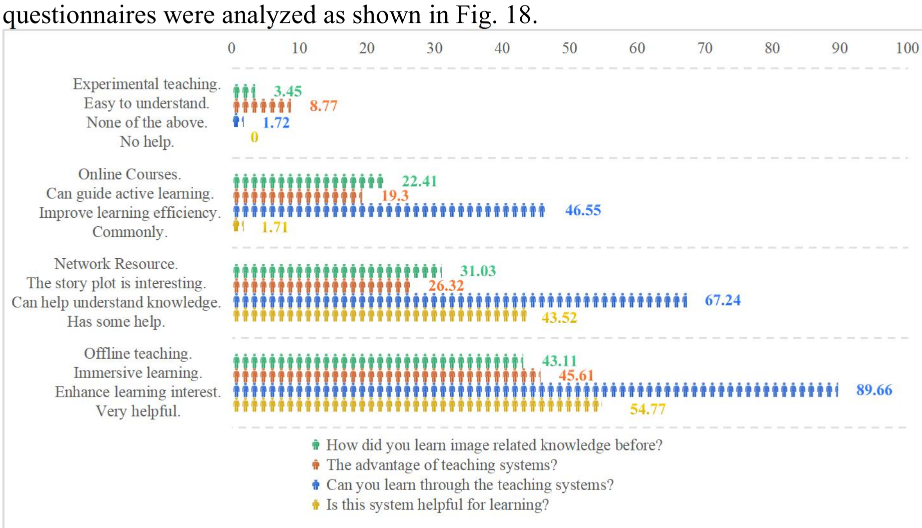


Fig. 18. Partial questionnaire analysis chart

5 Conclusion

This system is based on the main line of accomplishing the teaching objectives of image compression coding, designing storyline and virtual scene according to the knowledge characteristics of image compression coding, reasonably integrating the knowledge of image compression coding with the storyline, utilizing virtual reality technologies such as UGUI technology, particle system, animation system and dialogue system, and combining Unity3D to define the entities in the virtual scene to carry out intuitive teaching. The system is designed with experimental training to understand the programming of various methods of image compression coding, and the assessment mechanism is designed to verify the learning effect, which is in line with the complete teaching process and the teaching law, and the design of the system can satisfy our teaching objectives and the original design intention. However, the system only considers the mouse and keyboard to operate the system, the next work can study the smart wearable devices, as the peripheral of the system to interact with the system.

References

[1] Junge Han,Bingshuang Han.Research on the Design and Application of Virtual Reality Interaction Mode for Multimedia Speech Rooms in Colleges and Universities[J].China Now Educational Equipment,2024(05):19-21.

[2] Hui Zhao,Xiaojun Wang,Yun Wang,Zhe Hao,Xiuzai Zhang,Xining Wang.Experimental Design of Image Compression Encoding Teaching Based on FPGA[J].Information Technology and Informatization,2021(11):225-227.

[3] Siying Chen.Teaching Design of 3D Animation Course Based on Virtual Reality Technology[J].Computer knowledge and technology,2024,20(03):114-115.

[4] Qing Jin,Ping Lu.Exploration of Teaching Scheme for Digital Image Processing Case Library Based on VR[J].Chinese literary and artistic figures,2019(01):189-191.

[5] Xuekong Duan,Tong Li,Xundong Zhu.Design of Character Animation in Unity3D Game Scenes[J].Computer knowledge and technology,2019,15(09):199-200.

[6] Zhouyu Zhang. Research and Implementation of Skeletal Animation Subsystem Based on Neural Network Action Synthesis[D].University of Chinese Academy of Sciences (School of Artificial Intelligence, Chinese Academy of Sciences),2020.

[7] Hu X. Design of Virtual Experiment Teaching of Inorganic Chemistry in Colleges and Universities based on Unity3D[J]. International Journal of Advanced Computer Science and Applications, 2023.

[8] Guoyan Zhang.Research on using virtual reality technology to create simulation animations[J].Journal of Wuhu Vocational and Technical College,2022,24(04):67-71.

[9] Zhigang Wang.Research on Automatic Pathfinding Navigation System Based on Unity3D[J].Computer knowledge and technology,2019,15(36):209-211.

[10] Weiping Shao,Dongyang Li.Design and Implementation of Autonomous Pathfinding for Flapping Wing Aircraft Based on Unity3D[J].Group Technology and Modernization of Production,2021,38(03):30-33.

[11] Baoming Shi,Lan Li.Design of Unity3D Random Pathfinding Algorithm[J].Journal of Jilin Normal University (Natural Science Edition),2022,43(01):128-133.

[12] Yang Bo, Huadong Liu. Erratum to “Research on the Teaching Method of Environmental

Landscape Design Practice Based on IoT Network and Digital VR Technology" [J]. Security and Communication Networks, 2023, 21(10): 9891230.
[13] Minghe Huang, Cuixiang Zhong. An improvement to EZW codec of digital image processing teaching[J], 2018: 252-256.
[14] Michael Filsecker. A multilevel analysis of the effects of external rewards on elementary students' motivation, engagement and learning in an educational game. Computers & Education, 2014, 136-148.
[15] Tan L, Zeng Y, Zhang W. Research on image compression coding technology[C]. Journal of Physics: Conference Series. IOP Publishing, 2019, 1284(1): 012069.
[16] Di Tang.Design and Implementation of First Person Shooting Game Based on Unity3D Engine[D].University of Electronic Science and Technology of China,2021.
[17] Tao Dong,Yin Zhang.Design and implementation of a third-party shooting mobile game based on Unity3D[J].Communication World,2019,26(11):98-99.
[18] Hong Li,Peipei Lu.Design of Virtual Animation System Based on Unity3D[J].Modern electronic technology,2021,44(08):164-168.
[19] Qiao Li.Research on Dynamic Routing Problems Based on NavMesh Automatic Routing Algorithm[D].Huazhong University of Science and Technology,2019.
[20] Qinghua Liu,Di Wang.Research on Robot Simulation and Teleoperation System Based on Unity3D[J].Measurement and testing technology,2024,51(01):39-43.
[21] Jingyu Chen,Jin Xun,Rong Luo.A 3D human-computer interaction system based on gesture recognition[J].Modern Information Technology,2023,7(22):88-91.
[22] Jianqi Sun,Yuhan Nan,Junzhi Chen.Based on Unity3D virtual reality rehabilitation training system[J].Automation and Instrumentation,2024,(01):172-174.
[23] Zhike Wang,Changwei Huang.The application of loading and rendering technology for large-scale urban scene data in Unity 3D[J].Electricity and Energy,2023,44(03):254-259.